

UC Irvine

UC Irvine Electronic Theses and Dissertations

Title

Objects, Scenes, and People: from Observation to Reconstruction

Permalink

<https://escholarship.org/uc/item/9xg0v169>

ISBN

9798241656568

Author

Shin, Daeyun

Publication Date

2026-04-14

Peer reviewed|Thesis/dissertation

UNIVERSITY OF CALIFORNIA,
IRVINE

Objects, Scenes, and People: from Observation to Reconstruction

DISSERTATION

submitted in partial satisfaction of the requirements
for the degree of

DOCTOR OF PHILOSOPHY

in Computer Science

by

Daeyun Shin

Dissertation Committee:
Professor Charless C. Fowlkes, Chair
Professor Alexander C. Berg
Professor Erik B. Sudderth

Portions of Chapters 2 and 3 © 2018, 2019 IEEE
Portions of Chapter 4 © 2020 Springer
All other materials © 2026 Daeyun Shin

TABLE OF CONTENTS

	Page
LIST OF FIGURES	v
LIST OF TABLES	ix
ACKNOWLEDGMENTS	xi
VITA	xii
ABSTRACT OF THE DISSERTATION	xiv
1 Introduction	1
2 Pixels, Voxels, and Views	6
2.1 Introduction	7
2.2 Related work	10
2.3 Viewer-centered 3D shape completion	13
2.4 Network architectures for shape prediction	14
2.4.1 Generating multi-surface depth and silhouettes	15
2.4.2 Reconstructing multi-surface representations	16
2.4.3 Generating voxel representations	16
2.5 Experiments	18
2.5.1 Datasets	18
2.5.2 Evaluation metrics and processes	22
2.6 Discussion	23
2.7 Conclusion	25
3 3D Scene Reconstruction with Multi-Layer Depth and Epipolar Trans-	
formers	27
3.1 Introduction	28
3.2 Related Work	32
3.3 Reconstruction with Multi-Layer Depth	34
3.3.1 Multi-Layer Depth Maps from 3D Geometry	35
3.3.2 Predicting Multi-Layer Depth Maps	36
3.4 Epipolar Feature Transformer Networks	37
3.5 Experiments	40

3.5.1	Generation of Training Data	41
3.5.2	Model Architecture and Training	41
3.5.3	Evaluation	43
3.6	Conclusion	49
4	Predicting Camera Viewpoint Improves Cross-Dataset Generalization for 3D Human Pose Estimation	51
4.1	Introduction	51
4.2	Related Work	52
4.3	Analysis of Viewpoint Bias Across Datasets	54
4.4	Method	57
4.5	Experiments	59
4.5.1	Cross-Dataset Evaluation	59
4.5.2	Ablation Studies	60
4.5.3	Generalization to Other Architectures	61
4.6	Discussion	61
5	Egocentric Multi-Person Human Motion Estimation	63
5.1	Introduction	63
5.2	Related Work	65
5.3	Problem Formulation	67
5.4	Egocentric and Exocentric Estimation	69
5.4.1	EgoAllo: Head-Motion-Conditioned Body Estimation	70
5.4.2	CoMotion: Third-Person Visual Tracking	70
5.5	Method: Observation-Conditioned Diffusion	72
5.5.1	Imputation-Based Conditioning	73
5.5.2	Noise-Aware Conditioning Schedule	74
5.5.3	Per-Modality Handling	75
5.5.4	Training with Cutoff Schedule	75
5.5.5	Post-Inference Refinement	77
5.6	Experiments	77
5.6.1	Experimental Setup	77
5.6.2	Evaluated Methods	78
5.6.3	Main Results	80
5.6.4	Ablation Studies	82
5.6.5	Verification on Synthetic Data and No-Degradation Analysis	83
5.6.6	Analysis	85
5.7	Discussion	90
6	Conclusion	93
6.1	Summary	93
6.2	Limitations	95
6.3	Future Directions	96
	Bibliography	97

Appendix A Additional Material for Chapter 2	108
Appendix B Additional Material for Chapter 3	124

LIST OF FIGURES

	Page
2.1 We investigate the problem of predicting the 3D shape of an object from a single depth or RGB image (illustrated above). In particular, we examine the impacts of coordinate frames (viewer-centered vs. object-centered), shape representation (volumetric vs. multi-surface), and familiarity (known instance, novel instance, novel category).	8
2.2 We compare view-centered vs. object-centered and volumetric vs. multi-surface formulations of shape prediction. In view-centered prediction, the shape is predicted relative to the viewpoint of the input image, which requires encoding both shape and pose. In object-centered prediction, the shape is predicted in a canonical pose, which is standardized across training and prediction evaluation. For volumetric prediction, the shape is modeled as a set of filled 3D voxels. For multi-surface prediction, the shape is modeled as depth maps from multiple different viewpoints which tile the viewing sphere.	8
2.3 Network architecture: Encoders E_d , E_s , E_h learn view-specific shape features h extracted from the input depth and silhouette. h is used by the 10 output decoder branches $V^{(k)}$, $k=1..10$ which each synthesize one silhouette and two, front and back, depth images. The branches have independently parameterized fully connected layers, but the up-convolutional decoders G_d , G_s share parameters across all output branches.	13
2.4 Illustration of ground truth generation. Top: GT mesh voxelized in camera coordinates which are used in the viewer-centered voxel baseline experiment. The numeric labels around the model are the indices of the viewpoints from which the multi-surface depth images (shown above) were rendered, left to right. Viewpoint-0, whose camera is located at the origin, <i>always</i> corresponds to the input view, and the relative transformation from Viewpoint-0 to all the other viewpoints is constant throughout all our experiments. Bottom: Multi-surface projections are offset relative to the input view. This viewer-centered approach does not require alignment between the 3D models and allows the network to be trained in an unsupervised manner on synthetic data.	17

2.5	Multi-surfaces vs. Voxels. “Novel View” means other views of that shape instance were seen during training. “Novel Model” means other instances from the same class were seen in training. “Novel Class” means that no instances from that category were seen during training. Under multi-surface, a subset of the predicted depth maps are shown, as well as the complete reconstructed shape. The multi-surface approach tends to generalize better for novel classes.	19
2.6	RGB-based shape prediction examples: On left, is the input image. We show predicted depth maps and silhouettes from three views and a merged point cloud from all views, produced by the networks trained with object-centered coordinates and with viewer-centered coordinates. Viewer-centered tends to generalize better while object-centered sometimes produces a model that looks good but is from entirely the wrong category. In viewer-centered, the encoder learns to map inputs together if they correspond to similar shapes in similar poses, learning a viewpoint-sensitive representation. In object-centered, the encoder learns to map different views of the same object together, learning a viewpoint-invariant representation.	22
3.1	Given a single input view of a scene (top left), we would like to predict a complete geometric model. Depth maps (top right) provide an efficient representation of scene geometry but are incomplete, leaving large holes (e.g., the wardrobe). We propose multi-layer depth predictions (bottom left) that provide complete view-based representations of shape, and introduce an epipolar transformer network that allows view-based inference and prediction from virtual viewpoints (like overhead views, bottom right).	29
3.2	Overview of our system for reconstructing a complete 3D scene from a single RGB image. We first predict a multi-layer depth map that encodes the depths of front and back object surfaces as seen from the input camera. Given the extracted feature map and predicted multi-layer depths, the epipolar feature transformer network transfers features from the input view to a virtual overhead view, where the heights of observed objects are predicted. Semantic segmentation masks are inferred and inform our geometry estimates, but explicit detection of object instances is not required, increasing robustness.	30
3.3	Epipolar transfer of features from the input image to a virtual overhead view. Given multi-layer depth predictions of surface entrances and exits, each pixel in the input view is mapped to zero, one, or two segments of the corresponding epipolar line in the virtual view.	35
3.4	Single image scene reconstruction via multi-layer depth maps. Estimates of the front (green) and back (cyan) surfaces of objects, as seen from the input view, are complemented by heights estimated by a virtual overhead camera (dark green) via our epipolar feature transform. Room envelope estimates are rendered in gray.	44
3.5	Precision and recall of scene geometry as a function of match distance threshold.	45

3.6	Qualitative comparison of our viewer-centered, end-to-end scene surface prediction (left) and the object-based detection and voxel shape prediction of [88] (right). Object-based reconstruction is sensitive to detection and pose estimation errors, while our method is more robust.	46
3.7	A volumetric visualization of our predicted multi-layer surfaces and semantic labels on SUNCG. We project the center of each voxel into the input camera, and the voxel is marked occupied if the depth value falls in the first object interval (D_1, D_2) or the occluded object interval (D_3, D_4)	47
3.8	Evaluation of 3D reconstruction on the NYUv2 [69] dataset. Tulsiani <i>et al.</i> [88] are sensitive to the performance of 2D object detectors, and their voxelized output is a coarse approximation of the true 3D geometry.	48
4.1	Body-centered coordinate frame used to quantify camera viewpoint relative to the body. The origin is at the pelvis, with up (U) pointing toward the midpoint of the shoulders, front (F) defined by the cross product of the pelvis-to-shoulder vectors, and right (R) completing the orthonormal basis. The camera coordinate axes (x, y, z) are shown at left.	55
4.2	Azimuth distribution of camera viewpoints relative to the body-centered coordinate frame across five datasets. H36M shows four distinct peaks from its fixed cameras. 3DPW is heavily biased toward frontal views. SURREAL is nearly uniform (synthetic).	56
4.3	Elevation distribution across the same five datasets. H36M and GPA show positive elevation bias (cameras above subject height). 3DPW clusters near zero elevation (handheld camera at subject height). 3DHP has the widest elevation spread, including high angles from ceiling-mounted cameras.	56
4.4	Reduction in per-joint position error from the viewpoint branch as a function of azimuth (left) and elevation (right), for a model trained on 3DHP and tested on H36M. The viewpoint distributions of the training set (3DHP, black) and test set (H36M, blue) are shown alongside the error reduction (red). Error reduction is largest in viewpoint regions where 3DHP has coverage but H36M does not.	57
4.5	Two-branch architecture. A shared ResNet backbone feeds a pose branch (3D joint coordinates via heatmap regression) and a viewpoint branch (body-to-camera rotation classified into quaternion clusters). The viewpoint branch is discarded at test time and serves purely as a training regularizer.	58
5.1	Overview. In a multi-person egocentric setting, each person’s head-mounted camera provides third-person observations of others, while their own body motion is inferred from head trajectory alone. Can we fuse these complementary signals to enable global scene understanding and recover full-body 3D motion for all participants?	65

5.2	Architecture. Video streams from other participants' cameras are processed by a 3D pose model (CoMotion) to produce per-frame body parameter estimates. At each denoising step, these estimates are replaced into the noisy state x_t via imputation, with an observation mask encoder signaling which dimensions have observations. The diffusion transformer (based on EgoAllo [106]), conditioned on the egocentric SLAM trajectory via cross-attention, denoises all dimensions jointly.	69
5.3	Mean per-joint position error as a function of observation fraction, the fraction of frames within each 128-frame window where at least one camera detects the subject. Three baselines are shown: the egocentric motion prior (EgoAllo, dashed line at approximately 136mm), visual tracking with device alignment (CoMotion+SLAM), and zero-shot fusion (pretrained EgoAllo with observation replacement but no imputation training). Two trained fusion variants use the same replacement cutoff ($t=50$) at inference: +imputation training (100k steps with clean observations) and +cutoff training (10k additional steps with the cutoff schedule applied during training). The prior is approximately flat, confirming that observation fraction does not correlate with intrinsic scene difficulty. Both trained fusion variants improve over all baselines at every coverage level, and each training stage produces further gains across all bins. Gray bars show the number of evaluation windows per bin.	71
5.4	Comparison of egocentric and exocentric estimates. Left: The egocentric model (EgoAllo) provides full temporal coverage with calibrated global positioning, but limited body pose accuracy since head motion alone constrains limb positions loosely. Right: The exocentric model (CoMotion) provides more accurate relative body pose when people are visible, but has incomplete temporal coverage and unreliable absolute positioning.	72

LIST OF TABLES

	Page
2.1 3D shape prediction from a single depth image on the SHREC'12 dataset used by [61], comparing results for voxel and multi-surface decoders trained to produce models in a viewer-centered coordinate frame. Rock <i>et al.</i> [61] also predicts in viewer-centered coordinates.	20
2.2 Voxel IoU of predicted and ground truth values (mean, higher is better), using the voxel network. Trained for 45 epochs with batch size 150, learning rate 0.0001.	20
2.3 Silhouette IoU, using the 6-view multi-surface network (mean, higher is better).	20
2.4 Depth error, using the 6-view multi-surface network (mean, lower is better).	21
2.5 Per-category voxel IoU on PASCAL 3D+ using our multi-surface network and the voxel-based 3D-R2N2 network [5]. Although the network trained to produce object-centered (OC) models performs slightly better quantitatively (for multi-surface), the viewer-centered (VC) model tends to produce better qualitative results sometimes with misaligned pose (see Appendix A for additional examples).	21
3.1 Scene surface coverage (recall) of ground truth depth layers with a 5cm threshold. Our predictions cover 93% of the scene geometry inside the viewing frustum.	34
3.2 Augmenting the frontal depth prediction with the predicted virtual view height map improves both precision and recall (match threshold of 5cm).	45
3.3 We quantitatively evaluate the synthetic-to-real transfer of 3D geometry prediction on the ScanNet [7] dataset (threshold of 10cm). We measure recovery of true object surfaces and room layouts within the viewing frustum.	49
4.1 Cross-dataset 3D pose estimation (MPJPE in mm, lower is better). Each cell shows the error when training on the column dataset and testing on the row dataset. Bold: same-dataset (diagonal). The viewpoint branch reduces total cross-dataset error by 9–20mm without degrading same-dataset performance.	60

5.1	Results on EgoHumans (132 sequences, 7 activity categories). The first two rows are standalone baselines using a single signal each. The remaining rows show progressive improvement of the observation-conditioned diffusion approach: adding imputation-aware training, noise-aware conditioning, and post-inference optimization. MPJPE: mean per-joint position error in the world frame (mm). PA-MPJPE: after Procrustes alignment (mm). MPJVE: mean per-joint velocity error (mm/s). Accel: acceleration error (m/s ²). All rows without post-inference refinement except the last.	81
5.2	Per-category MPJPE (mm) on EgoHumans. The egocentric motion prior provides a baseline unaffected by observation coverage. Visual tracking + device alignment depends on mutual visibility: activities with close interaction (fencing, tagging) have better coverage and lower error, while activities with more distance (volleyball, tennis) or object focus (lego assembly) have worse coverage. The observation-conditioned diffusion approach improves on both baselines across all categories.	81
5.3	Cross-evaluation of model variants across observation conditions and datasets (MPJPE, mm). “EgoHumans + obs” uses each model’s best inference configuration with CoMotion observations: plain replacement for the pretrained model (a), per-modality cutoff and shape stabilization for trained variants (b, c). “EgoHumans no obs” drops all observations, reducing to CPF-conditioned prior only. “AMASS + obs” uses clean ground-truth observations at $f=1.0$ without cutoff (since clean observations help at all denoising steps). “AMASS no obs” is the original EgoAllo evaluation setting ($f=0$). No post-inference refinement.	83
5.4	AMASS evaluation at varying observation fractions (MPJPE, mm). Clean ground-truth observations, no replacement cutoff. The cutoff-schedule-trained model (c) matches or improves on the imputation-trained model (b) at all fractions, including $f=0$	84
5.5	Percentage of person-frames with no exocentric observation, by activity category. Activities with close interaction (fencing, tagging) have high mutual visibility, while activities with greater distance (volleyball, tennis) or shared object focus (lego assembly) have more missing frames.	87
5.6	Body pose accuracy (PA-MPJPE, mm) restricted to the 265,361 person-frames where CoMotion detected the subject from at least one observer. CoMotion’s direct visual estimate achieves better relative body pose than the diffusion model on these frames; the diffusion model’s advantage lies in temporal gap-filling and global positioning rather than refining already-observed poses. Lower is better.	88

ACKNOWLEDGMENTS

I thank my advisor, Professor Charless Fowlkes, for his patience, encouragement, and perspective throughout my PhD. He has been endlessly supportive and has a way of making hard problems feel approachable. His guidance has shaped how I think. I could not have asked for a better advisor.

I thank my committee members, Professor Charless Fowlkes, Professor Alexander Berg, and Professor Erik Sudderth, for their questions and engagement with this work. I also thank Professor Derek Hoiem at UIUC for advising my master’s thesis.

I thank my labmates in the computer vision lab for their company and for making grad school a lot more enjoyable: Yunhan Zhao, Phuc Nguyen, Zhe Wang, Bailey Kong, Shu Kong, Samia Shafique, Minhaeng Lee, Viktor Liviniuk, Alina Liviniuk, Shaurya Rathore, Linfeng Wang, Dongxu Zhao, Xinxuan Lu, Edward Li, Arka Pathak, Sahil Khose, and Mengqi Zhang.

Finally, I thank my family for their love and support through all of this.

Portions of this dissertation are adapted from previously published material with coauthors. Professor Charless C. Fowlkes directed and supervised the research which forms the basis for this dissertation. In particular:

- The text of Chapter 2 is an adaptation of the material as it appears in Daeyun Shin, Charless Fowlkes, and Derek Hoiem. Pixels, voxels, and views: A study of shape representations for single view 3D object shape prediction. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3061–3069, 2018, used with permission from IEEE.
- The text of Chapter 3 is an adaptation of the material as it appears in Daeyun Shin, Zhile Ren, Erik Sudderth, and Charless Fowlkes. 3D scene reconstruction with multi-layer depth and epipolar transformers. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2172–2182, 2019, used with permission from IEEE.
- The text of Chapter 4 is an adaptation of the material as it appears in Zhe Wang, Daeyun Shin, and Charless Fowlkes. Predicting camera viewpoint improves cross-dataset generalization for 3D human pose estimation. In *European Conference on Computer Vision Workshops (ECCVW)*, pages 523–540, 2020, used with permission from Springer.

This research was supported by NSF grants IIS-1618806, IIS-1253538, and CNS-1730158, a gift from Qualcomm, and a hardware donation from NVIDIA.

VITA

Daeyun Shin

EDUCATION

Doctor of Philosophy in Computer Science University of California, Irvine	2026 <i>Irvine, California</i>
Master of Science in Computer Science University of Illinois at Urbana-Champaign	2017 <i>Urbana, Illinois</i>
Bachelor of Science in Computer Science University of Illinois at Urbana-Champaign	2015 <i>Urbana, Illinois</i>

RESEARCH EXPERIENCE

Software Engineering Intern Google LLC	2025 <i>Mountain View, California</i>
Graduate Research Assistant University of California, Irvine	2018–2022, 2024–2026 <i>Irvine, California</i>
Research Intern Google LLC	2021 <i>Irvine, California</i>
Research Intern Snap Inc.	2019 <i>Santa Monica, California</i>
Research Assistant University of Illinois at Urbana-Champaign	2016–2017 <i>Urbana, Illinois</i>
Software Engineering Intern Google LLC	2016 <i>New York, New York</i>

TEACHING EXPERIENCE

University of California, Irvine	2017–2018
University of Illinois at Urbana-Champaign	2015–2016

PUBLICATIONS

Jason Rock, Tanmay Gupta, Justin Thorsen, Junyoung Gwak, Daeyun Shin, and Derek Hoiem. Completing 3D object shape from one depth image. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2484–2493, 2015

Daeyun Shin, Charless Fowlkes, and Derek Hoiem. Pixels, voxels, and views: A study of shape representations for single view 3D object shape prediction. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3061–3069, 2018

Daeyun Shin, Zhile Ren, Erik Sudderth, and Charless Fowlkes. 3D scene reconstruction with multi-layer depth and epipolar transformers. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2172–2182, 2019

Yunhan Zhao, Shu Kong, Daeyun Shin, and Charless Fowlkes. Domain decluttering: Simplifying images to mitigate synthetic-real domain shift and improve depth estimation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3330–3340, 2020

Zhe Wang, Daeyun Shin, and Charless Fowlkes. Predicting camera viewpoint improves cross-dataset generalization for 3D human pose estimation. In *European Conference on Computer Vision Workshops (ECCVW)*, pages 523–540, 2020

Zhe Wang, Liyan Chen, Shaurya Rathore, Daeyun Shin, and Charless Fowlkes. Geometric pose affordance: Monocular 3D human pose estimation with scene constraints. In *European Conference on Computer Vision Workshops (ECCVW)*, pages 3–18, 2022

ABSTRACT OF THE DISSERTATION

Objects, Scenes, and People: from Observation to Reconstruction

by

Daeyun Shin

Doctor of Philosophy in Computer Science

University of California, Irvine, 2026

Professor Charless C. Fowlkes, Chair

Recovering the three-dimensional structure of objects, scenes, and people from images requires reasoning beyond what is directly visible—behind occluding surfaces, outside the field of view, and across time. This thesis develops methods for reconstructing shapes and motion across these domains, from static objects and scenes to human pose in dynamic settings. A recurring theme is fusing complementary representations: coordinate frame choices, perceptual and generative approaches, egocentric and exocentric perspectives. We begin by studying how coordinate systems and viewpoint choices shape learned representations for object shape prediction and pose estimation. We then develop multi-layer depth representations as a viewer-centered alternative to detection-based scene reconstruction. Combining viewpoint prediction with pose estimation produces representations that transfer across datasets. Building on these foundations, we integrate visual tracking with generative motion modeling to recover global multi-person motion by fusing egocentric and exocentric views.

Chapter 1

Introduction

Every image records what is visible from a single viewpoint. The backs of objects, surfaces hidden behind other surfaces, and people who have left the frame are all absent from the observation but present in the world. A depth sensor aimed at a coffee mug captures one surface; the rest of the mug, and the table beneath it, go unrecorded. A head-mounted camera sees the world in front of the wearer but not the wearer’s own body, while providing a third-person view of everyone else in the room. Recovering complete 3D structure in any of these cases requires reasoning beyond what is directly observed.

Learning-based methods for 3D reconstruction have made substantial progress on this challenge. Given a single image, neural networks can now predict volumetric occupancy [98, 5], point clouds [13], mesh vertices [93], or depth maps from multiple viewpoints [83] for the complete 3D shape of an object, including surfaces not visible in the input. At the scene level, methods predict geometric completions from single depth images [75] or recover layered representations separating occluders from background [89]. These methods share a common structure: an encoder processes the input image, and a decoder produces 3D output in some chosen representation. Most operate in object-centered or canonical coordinates, where the

3D shape is predicted in a standardized pose regardless of the input viewpoint. A front view and a side view of the same car should produce the same 3D output, which simplifies the prediction task but requires aligned training data. Generalization to new object categories is limited because a canonical pose must be defined for each category.

A similar pattern holds in 3D human pose estimation. Methods predicting joint positions [45] or body model parameters [31] achieve strong results on individual benchmarks but transfer poorly across datasets with different capture environments. Whether the gap comes from body pose, background appearance, or camera configuration has received less systematic attention. In egocentric vision, the problem takes a different form: head-mounted cameras capture the world from the wearer’s first-person perspective, but the wearer’s own body falls almost entirely outside the field of view [108]. Recent work estimates body pose from head motion alone using diffusion-based priors [106], but accuracy remains limited by the weak coupling between head motion and limb positions. Across these domains, the camera’s geometric relationship to the subject is typically treated as a preprocessing step: something to normalize out before the real modeling begins.

This thesis develops methods for 3D inference under partial observability across four settings: single objects, indoor scenes, human bodies observed by external cameras, and human bodies observed by head-mounted cameras worn by the subjects themselves. The work spans static reconstruction from a single image to dynamic estimation from multi-person video. Across these settings, the geometric relationship between observer and subject plays a recurring role, whether as the coordinate frame for predictions, the camera’s viewpoint relative to the body, or the distinction between first-person and third-person perspectives.

Chapter 2 examines coordinate frame and shape representation choices for single-view 3D object prediction. We provide a controlled comparison of viewer-centered and object-centered prediction using a shared encoder-decoder architecture, evaluating across three levels of familiarity: novel viewpoints, novel instances, and novel categories. Viewer-centered prediction

generalizes better to unfamiliar objects; object-centered prediction favors familiar instances. The two frames induce qualitatively different learned representations: viewer-centered models encode shape and pose jointly and transfer across categories, while object-centered models rely more on implicit category recognition. Surface-based representations outperform voxels for novel categories and encode higher-resolution geometry with fewer parameters.

Chapter 3 extends viewer-centered prediction from single objects to full indoor scenes. Scene reconstruction has traditionally relied on pipelines that detect objects, retrieve or reconstruct each individually, and compose them into a scene [26, 20]. We take a different approach: predict the complete 3D scene directly from a single input image, without object detection or category-specific models. We introduce a multi-layer depth representation that records multiple surface intersections along each camera ray, including front and back surfaces of foreground and occluded objects and the room envelope. An Epipolar Feature Transformer uses perspective geometry to transfer features to virtual camera viewpoints, even for surfaces nearly tangent to the input viewing direction. The five-layer representation covers over 90% of scene geometry within the camera frustum.

The first two chapters address static subjects observed from a single viewpoint, where the missing information is geometric: what surfaces are hidden. The later chapters shift to human subjects, where the 3D structure changes over time, observations arrive from cameras with different and varying configurations, and coverage is intermittent.

Chapter 4 turns to 3D human pose estimation, where the observer–subject relationship manifests as dataset bias. The major benchmarks (Human3.6M [23], GPA [95], SURREAL [90], 3DPW [91], MPI-INF-3DHP [48]) differ substantially in camera configuration: number of cameras, viewing angles, and height relative to the subject. Standard preprocessing normalizes camera intrinsics and subject distance but leaves camera orientation unnormalized. We show that this residual viewpoint variation, not body pose variation, is the dominant source of cross-dataset generalization failure: a classifier trained on body-centered, size-normalized

poses cannot identify the source dataset above chance. The datasets differ not in what bodies do, but in how cameras see them. Adding a viewpoint prediction branch, an auxiliary task that classifies the rotation from a body-centered frame to the camera frame, reduces total cross-dataset error by 9–20mm across five benchmarks, with minimal effect on same-dataset performance. Coarse viewpoint classification regularizes more effectively than precise viewpoint regression; the value lies not in accurately predicting the viewpoint but in forcing the shared representation to encode the camera–body geometry. The auxiliary branch is discarded at inference.

Chapter 5 addresses the most challenging setting: reconstructing the full-body motion of multiple people, each wearing a head-mounted camera, as they move freely in a shared environment. Recent benchmarks capture multi-person interactions from egocentric perspectives [35, 19]. The setting introduces a structural duality: each person’s first-person camera simultaneously provides third-person observations of others, while their own body must be inferred from indirect cues. Two complementary signals are available for reconstructing any individual’s motion. An egocentric motion prior [106], conditioned on head trajectory alone, generates full-body motion with complete temporal coverage but limited pose accuracy, since head motion constrains limb positions only loosely. Exocentric visual observations from other cameras [50] provide more accurate body pose when available, but temporal coverage is incomplete (approximately 24% of person-frames have no observation from any camera), and absolute positioning suffers from monocular scale ambiguity. Neither source alone is sufficient.

We develop a conditional diffusion model that fuses these signals through imputation-based conditioning, replacing observed dimensions of the body state during the iterative denoising process. Noisy observations help during the early, coarse stages of denoising but corrupt the result during fine-grained refinement. When observation noise exceeds the diffusion noise at a given step, replacement adds net noise rather than net information. A noise-aware

conditioning schedule that stops observation replacement below this crossover point improves both positional accuracy and temporal smoothness. Per-modality handling further refines the approach by exploiting the fact that some body parameters (shape) are more reliably estimated than others (joint rotations). The model trains entirely on single-person motion capture data with simulated observations; no multi-person training data is needed. On the EgoHumans benchmark [35], mean per-joint position error drops from 136mm (motion prior alone) to 102mm with the full approach.

This thesis makes four contributions. First, a systematic comparison of coordinate frame and shape representation choices for single-view 3D object prediction, establishing that viewer-centered and object-centered prediction induce qualitatively different learned representations with different generalization properties (Chapter 2). Second, a multi-layer depth representation for full indoor scenes that covers both visible and hidden surfaces without object detection or category-specific models (Chapter 3). Third, the identification of camera viewpoint as the primary source of cross-dataset generalization failure for 3D human pose estimation, with an auxiliary viewpoint prediction task that reduces cross-dataset error by 9–20mm across five benchmarks (Chapter 4). Fourth, a conditional diffusion framework for multi-person egocentric motion estimation that fuses egocentric priors with exocentric observations through noise-aware imputation-based conditioning, with a per-window analysis decomposing performance by observation conditions (Chapter 5).

Chapter 2

Pixels, Voxels, and Views: A Study of Shape Representations for Single View 3D Object Shape Prediction

This chapter examines the simplest setting in this thesis: predicting the complete 3D shape of an object from a single image, where the hidden structure is determined by the shape itself and the camera’s viewpoint. The central question is how the choice of coordinate frame (viewer-centered versus object-centered) and shape representation (volumetric versus surface-based) affects what models learn and how they generalize.

The goal of this chapter is to compare surface-based and volumetric 3D object shape representations, as well as viewer-centered and object-centered reference frames for single-view 3D shape prediction. We propose a new algorithm for predicting depth maps from multiple viewpoints, with a single depth or RGB image as input. By modifying the network

© 2018 IEEE. Reprinted, with permission, from D. Shin, C. Fowlkes, and D. Hoiem, “Pixels, Voxels, and Views: A Study of Shape Representations for Single View 3D Object Shape Prediction,” in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

and the way models are evaluated, we can directly compare the merits of voxels vs. surfaces and viewer-centered vs. object-centered for familiar vs. unfamiliar objects, as predicted from RGB or depth images. Among our findings, we show that surface-based methods outperform voxel representations for objects from novel classes and produce higher resolution outputs. We also find that using viewer-centered coordinates is advantageous for novel objects, while object-centered representations are better for more familiar objects. Interestingly, the coordinate frame significantly affects the shape representation learned, with object-centered placing more importance on implicitly recognizing the object category and viewer-centered producing shape representations with less dependence on category recognition.

2.1 Introduction

Shape is arguably the most important property of objects, providing cues for affordance, function, category, and interaction. This chapter examines the problem of predicting the 3D object shape from a single image (Fig. 2.1). The availability of large 3D object model datasets [1] and flexible deep network learning methods has made this an increasingly active area of research. Recent methods predict complete 3D shape using voxel [32, 5] or octree [84] volumetric representations, multiple depth map surfaces [83], point cloud [13], or a set of cuboid part primitives [118]. However, there is not yet a systematic evaluation of important design choices such as the choice of shape representation and coordinate frame.

In this chapter, we investigate two key issues, illustrated in Fig. 2.2. First, is it better to represent shape *volumetrically* or as *multiple 2.5D surfaces* observed from varying viewpoints? The earliest (albeit still recent) pattern recognition approaches to shape prediction use volumetric representations (e.g. [61, 32]), but more recent works have proposed surface-based representations [83]. Qi *et al.* [55] finds an advantage for surface-based representations for 3D object classification, since surfaces can encode high resolution shapes with fewer param-

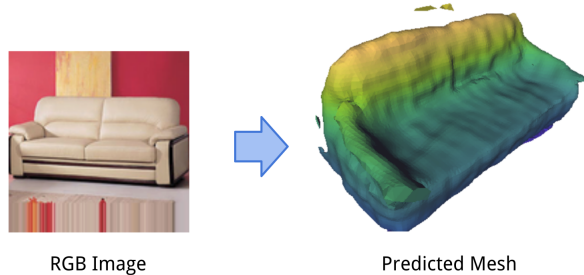


Figure 2.1: We investigate the problem of predicting the 3D shape of an object from a single depth or RGB image (illustrated above). In particular, we examine the impacts of coordinate frames (viewer-centered vs. object-centered), shape representation (volumetric vs. multi-surface), and familiarity (known instance, novel instance, novel category).

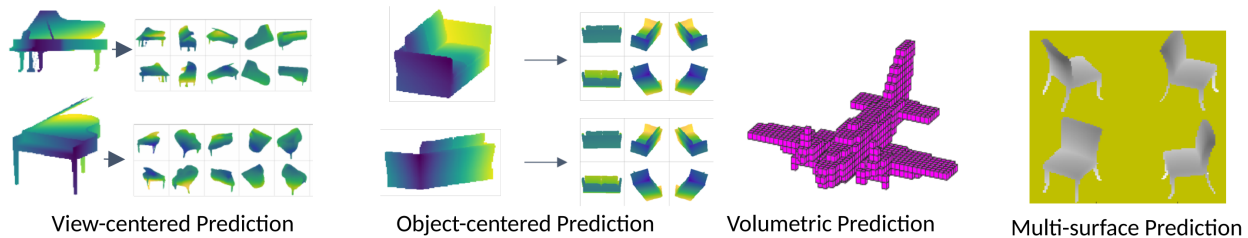


Figure 2.2: We compare view-centered vs. object-centered and volumetric vs. multi-surface formulations of shape prediction. In **view-centered** prediction, the shape is predicted relative to the viewpoint of the input image, which requires encoding both shape and pose. In **object-centered** prediction, the shape is predicted in a canonical pose, which is standardized across training and prediction evaluation. For **volumetric** prediction, the shape is modeled as a set of filled 3D voxels. For **multi-surface** prediction, the shape is modeled as depth maps from multiple different viewpoints which tile the viewing sphere.

eters. Rendered surfaces have fewer pixels than there are voxels in a high resolution mesh. However, generating a complete shape from 2.5D surfaces creates an additional challenge, since the surfaces need to be aligned and fused into a single 3D object surface.

Second, what is the impact of object-centered vs. view-centered coordinate frames for shape prediction? Nearly all recent 3D shape generation methods use *object-centered coordinates*, where the object’s shape is represented in a canonical view. For example, shown either a front view or side view of a car, the goal is to generate the same front-facing 3D model of the car. Object-centered coordinates simplify the prediction problem, but suffer from several practical drawbacks: the viewer-relative pose is not recovered; 3D models used for training must be aligned to a canonical pose; and prediction on novel object categories is difficult due

to lack of predefined canonical pose. In *viewer-centered coordinates*, the shape is represented in a coordinate system aligned to the viewing perspective of the input image, so a front-view of a car should yield a front-facing 3D model, while a side-view of a car should generate a side-facing 3D model. This increases the variation of predicted models, but also does not require aligned training models and generalizes naturally to novel categories.

We study these issues using a single encoder-decoder network architecture, swapping the decoder to study volume vs. surface representations and swapping the coordinate frame of predictions to study viewer-centered vs. object-centered. We examine effects of familiarity by measuring accuracy for novel views of known objects, novel instances of known categories, and objects from novel categories. We also evaluate prediction from both depth and RGB images. Our experiments indicate a clear advantage for surface-based representations in novel object categories, which likely benefit from the more compact output representations relative to voxels. Our experiments also show that prediction in viewer-centered coordinates generalizes better to novel objects, while object-centered performs better for novel views of familiar instances. Further, models that learn to predict in object-centered coordinates seem to learn and rely on object categorization to a greater degree than models trained to predict viewer-centered coordinates.

In summary, our main contributions include:

- We introduce a new method for surface-based prediction of object shape in a viewer-centered coordinate frame. Our network learns to predict a set of silhouette and depth maps at several viewpoints relative to the input image, which are then locally registered and merged into a point cloud from which a surface can be computed.
- We compare the efficacy of volumetric and surface-based representations for predicting 3D shape, showing an advantage for surface-based representations on unfamiliar object categories regardless of whether final evaluation is volumetric or surface-based.

- We examine the impact of prediction in viewer-centered and object-centered coordinates and showing that networks generalize better to novel shapes if they learn to predict in viewer-centered coordinates (which is not currently common practice), and that the coordinate choice significantly changes the embedding learned by the network encoder.

2.2 Related work

Our approach relates closely to recent efforts to generate novel views of an object, or its shape. We also touch briefly on related studies in human vision.

Volumetric shape representations: Several recent studies offer methods to generate volumetric object shapes from one or a few images [98, 32, 61, 5, 102, 84]. Wu *et al.* [98] proposes a convolutional deep belief network for learning 3D representations using volumetric supervision and evaluate applications to various recognition tasks. Other studies quantitatively evaluate 3D reconstruction results, with metrics including voxel intersection-over-union [61, 5, 102], mesh distance [32, 61], and depth map error [32]. Some follow template deformation approaches using surface rigidity [32, 61] and symmetry priors [61], while others [98, 5, 102] approach the problem as deep representation learning using encoder-decoder networks. Fan *et al.* [13] proposes a point cloud generation network that efficiently predicts coarse volumetric object shapes by encoding only the coordinates of points on the surface. Our voxel and multi-surface prediction networks use an encoder-decoder network. For multi-surface prediction, the decoder generates multiple segmented depth images, pools depth values into a 3D point cloud, and fits a 3D surface to obtain the complete 3D shape.

Multi-surface representations: Multi-surface representations of 3D shapes are popular for categorization tasks. The seminal work by Chen *et al.* [2] proposes a 3D shape descrip-

tor based on the silhouettes rendered from the 20 vertices of a dodecahedron surrounding the object. More recently, Su *et al.* [76] and Qi *et al.* [55] train CNNs on 2D renderings of 3D mesh models for classification. Qi *et al.* [55] compares CNNs trained on volumetric representations to those trained on multiview representations. Although both representations encode similar amounts of information, they showed that multiview representations significantly outperform volumetric representations for 3D object classification. Unlike our approach, these approaches use multiple projections as input rather than output.

To synthesize multi-surface output representations, we train multiple decoders. Dosovitskiy *et al.* [11] show that CNNs can be used to generate images from high-level descriptions such as object instance, viewpoint, and transformation parameters. Their network jointly predicts an RGB image and its segmentation mask using two up-convolutional output branches sharing a high-dimensional hidden representation. The decoder in our network learns the segmentation for each output view in a similar manner.

Our work is related to recent studies [83, 38, 104, 116, 102, 74, 43] that generate multiview projections of 3D objects. The multiview perceptron by Zhu *et al.* [116] generates one random view at a time, given an RGB image and a random vector as input. Inspired by the mental rotation ability in humans, Yang *et al.* [104] proposed a recurrent encoder-decoder network that outputs RGB images rotated by a fixed angle in each time step along a path of rotation, given an image at the beginning of the rotation sequence as input. They disentangle object identity and pose by sharing the identity unit weights across all time steps. Their experiments do not include 3D reconstruction or geometric analysis.

Our proposed method predicts 2.5D surfaces (depth image and object silhouette) of the object from a set of fixed viewpoints evenly spaced over the viewing sphere. In some experiments (Table 2.1), we use 20 views, as in [2], but we found that 6 views provide similar results and speeds training and evaluation, so 6 views are used for the remainder. Most existing approaches [83, 38, 102] parameterize the output image as (x, θ) where x is the

input image and θ is the desired viewpoint relative to canonical object-centered coordinate system. Yan *et al.* [102] introduce a formulation that indirectly learns to generate voxels through silhouettes using multi-surface projective constraints, but interestingly they report that voxel IoU performance is better when the network is trained to minimize projection loss alone, compared to when jointly trained with volumetric loss.

Our approach, in contrast, uses multiview reconstruction techniques (3D surface from point cloud) as a post-process to obtain the complete 3D mesh, treating any inconsistencies in the output images as if they were observational noise. Our formulation also differs in that we learn a view-specific representation, and the complete object shape is produced by simultaneously predicting multiple views of depth maps and silhouettes. In this multi-surface prediction, our approach is similar to Soltani *et al.*'s [74], but our system does not use class labels during training. When predicting shape in object-centered coordinates, the predicted views are at fixed orientations compared to the canonical view. When predicting shape in viewer-centered coordinates, the predicted views are at fixed orientations compared to the input view.

Human vision: In experiments on 2D symbols, Tarr and Pinker [82] found that human perception is largely tied to viewer-centered coordinates; this was confirmed by McMullen and Farah [47] for line drawings, who also found that object-centered coordinates seem to play more of a role for familiar exemplars. Note that in the human vision literature, “viewer-centered” usually means that the object shape is represented as a set of images in the viewer’s coordinate frame, and “object-centered” usually means a volumetric shape is represented in the object’s coordinate frame. In our work, we consider both the shape representation (volumetric or surface) and coordinate frame (viewer or object) as separate design choices. We do not claim our computational approach has any similarity to human visual processing, but it is interesting to see that in our experiments with 3D objects, we also find a preference for object-centered coordinates for familiar exemplars (i.e., novel view

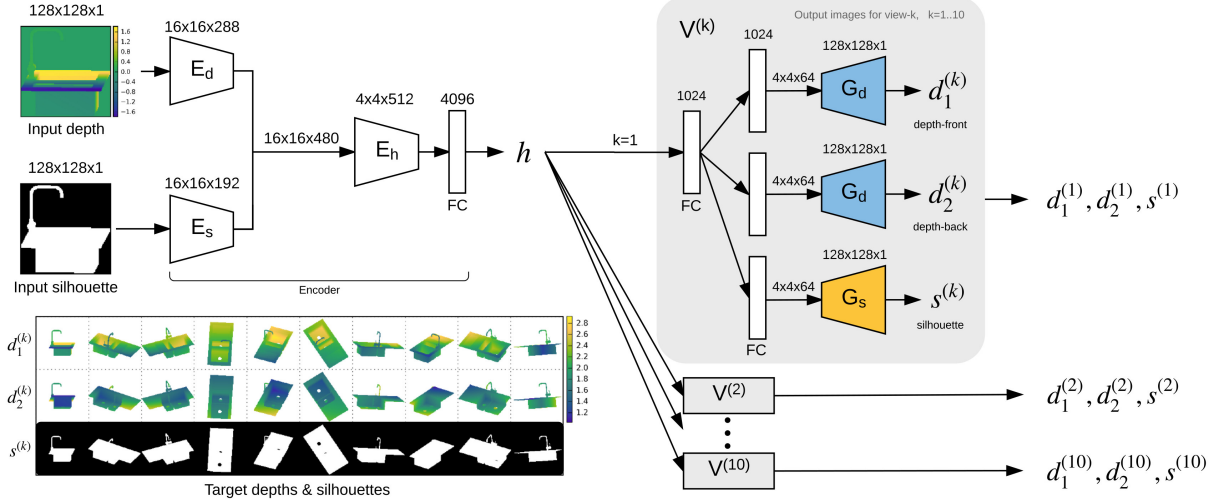


Figure 2.3: Network architecture: Encoders E_d , E_s , E_h learn view-specific shape features h extracted from the input depth and silhouette. h is used by the 10 output decoder branches $V^{(k)}$, $k=1..10$ which each synthesize one silhouette and two, front and back, depth images. The branches have independently parameterized fully connected layers, but the up-convolutional decoders G_d , G_s share parameters across all output branches.

of known object) and for viewer-centered coordinates in other cases.

2.3 Viewer-centered 3D shape completion

Given a single depth or RGB image as input, we want to predict the complete 3D shape of the object being viewed. In the commonly used object-centered setting, the shape is predicted in canonical model coordinates specified by the training data. For example, in the ShapeNet-Core dataset, the x-axis or $(\phi_{az} = 0^\circ, \theta_{el} = 0^\circ)$ direction corresponds to the commonly agreed upon front of the object, and the relative transformation parameters from the input view to this coordinate system is unknown. In our viewer-centered approach, we supervise the network to predict a pre-aligned 3D shape in the input image’s reference frame — e.g. so that $(\phi_{az} = 0^\circ, \theta_{el} = 0^\circ)$ in the output coordinate system always corresponds to the input viewpoint. Our motivation for exploring these two representations is the hypothesis that networks trained on viewer-centered and object-centered representations learn very different

information. A practical advantage of the viewer-centered approach is that the network can be trained in an unsupervised manner across multiple categories without requiring humans to specify intra-category alignment. However, viewer-centered training requires synthesizing separate target outputs for each viewpoint input which increases training data storage cost.

In all experiments, we supervise the networks only using geometric (or photometric) data without providing any side information about the object category label or input viewpoint. The only assumption is that the gravity direction is known (fixed as down in the input view). This allows us to focus on whether the predicted shapes can be completed/interpolated solely based on the 2.5D geometric or RGB input stimuli in a setting where contextual cues are not available. In the case of 2.5D input, we normalize the input depth image so that the bounding box of the silhouette fits inside an orthographic viewing frustum ranging from $\langle -1, -1 \rangle$ to $\langle 1, 1 \rangle$ with the origin placed at the centroid.

2.4 Network architectures for shape prediction

Our multi-surface shape prediction system uses an encoder-decoder network to predict a set of silhouettes and depth maps. Figure 2.3 provides an overview of the network architecture, which takes as input a depth map and a silhouette. We also perform experiments on a variant that takes an RGB image as input. To directly evaluate the relative merits of the surface-based and voxel-based representations, we compare this with a volumetric prediction network by replacing the decoder with a voxel generator. Both network architectures can be trained to produce either viewer-centered or object-centered predictions.

2.4.1 Generating multi-surface depth and silhouettes

We observe that, for the purpose of 3d reconstruction, it is important to be able to see the object from certain viewpoints – e.g. classes such as cup and bathtub need at least one view from the top to cover the concavity. Our proposed method therefore predicts 3D object shapes at evenly spaced views around the object. We place the cameras at the 20 vertices $\{\mathbf{v}_0, \dots, \mathbf{v}_{19}\}$ of a dodecahedron centered at the origin. A similar setup was used in the Light Field Descriptor [2] and a recent study by Soltani *et al.* [74].

In order to determine the camera parameters, we rotate the vertices so that vertex $\mathbf{v}_0 = \langle 1, 1, 1 \rangle$ aligns with the input viewpoint in the object’s model coordinates. The up-vectors point towards the z-axis and are rotated accordingly. Note that the input viewpoint $\mathbf{v}_0$ is not known in our setting, but the relative transformations from $\mathbf{v}_0$ to all of the output viewpoints are known and fixed.

As illustrated in Figure 2.3, our network takes the depth image and the silhouette in separate input branches. The encoder units (E_d, E_s, E_h) consist of bottleneck residual layers. E_d and E_s each take in a depth image and a silhouette. They are concatenated in the channel dimension at resolution 16 and the following residual layers E_h output the latent vector h from which all output images are derived simultaneously. An alternate approach is taking in a two-channel image in a single encoder. We experimented with both architectures and found the two-branch network to perform better.

We use two generic decoders (Table 2.3) to generate the views, one for all depths and another for all silhouettes. Each view in our setting has a corresponding segmented silhouette and another view on the opposite side, thus only 10 out of the 20 silhouettes need to be predicted due to symmetry (or 3 out of 6 if predicting six views). The network therefore outputs a silhouette and corresponding front and back depth images $\{s^{(i)}, d_f^{(i)}, d_b^{(i)}\}$ in the i -th output branch. Similarly to Dosovitskiy *et al.* [11], we minimize the objective function

$$\mathbf{L}_{\text{proj}} = k\mathbf{L}_s + (1 - k)\mathbf{L}_d$$

where $\mathbf{L}_s$ is the mean logistic loss over the silhouettes and $\mathbf{L}_d$ is the mean MSE over the depth maps whose silhouette label is 1. We use $k = 0.2$ in our experiments.

2.4.2 Reconstructing multi-surface representations

In our study, we use the term “reconstruction” to refer to surface mesh reconstruction in the final post-processing step. We convert the predicted multiview depth images to a single triangulated mesh using Floating Scale Surface Reconstruction (FSSR) [14], which we found to produce better results than Poisson Reconstruction [34] in our experiments. FSSR is widely used for surface reconstruction from oriented 3D points derived from multiview stereo or depth sensors. Our experiments are unique in that surface reconstruction methods are used to resolve noise in predictions generated by neural networks rather than sensor observations. We have found that 3D surface reconstruction reduces noise and error in surface distance measurements.

2.4.3 Generating voxel representations

We compare our multi-surface shape prediction method with a baseline that directly predicts a 3D voxel grid. Given a single-view depth image of an object, the “Voxels” network generates a grid of 3D occupancy mappings in the camera coordinates of viewpoint $\mathbf{v}_0$. The cubic window of length 2 centered at $\langle 0, 0, 1 \rangle$ is voxelized after camera transformation. The encoded features h feed into 3D up-convolutional layers, outputting a final $48 \times 48 \times 48$ volumetric grid. The network is trained from scratch to minimize the logistic loss $\mathbf{L}_v$ over the binary voxel occupancy labels.

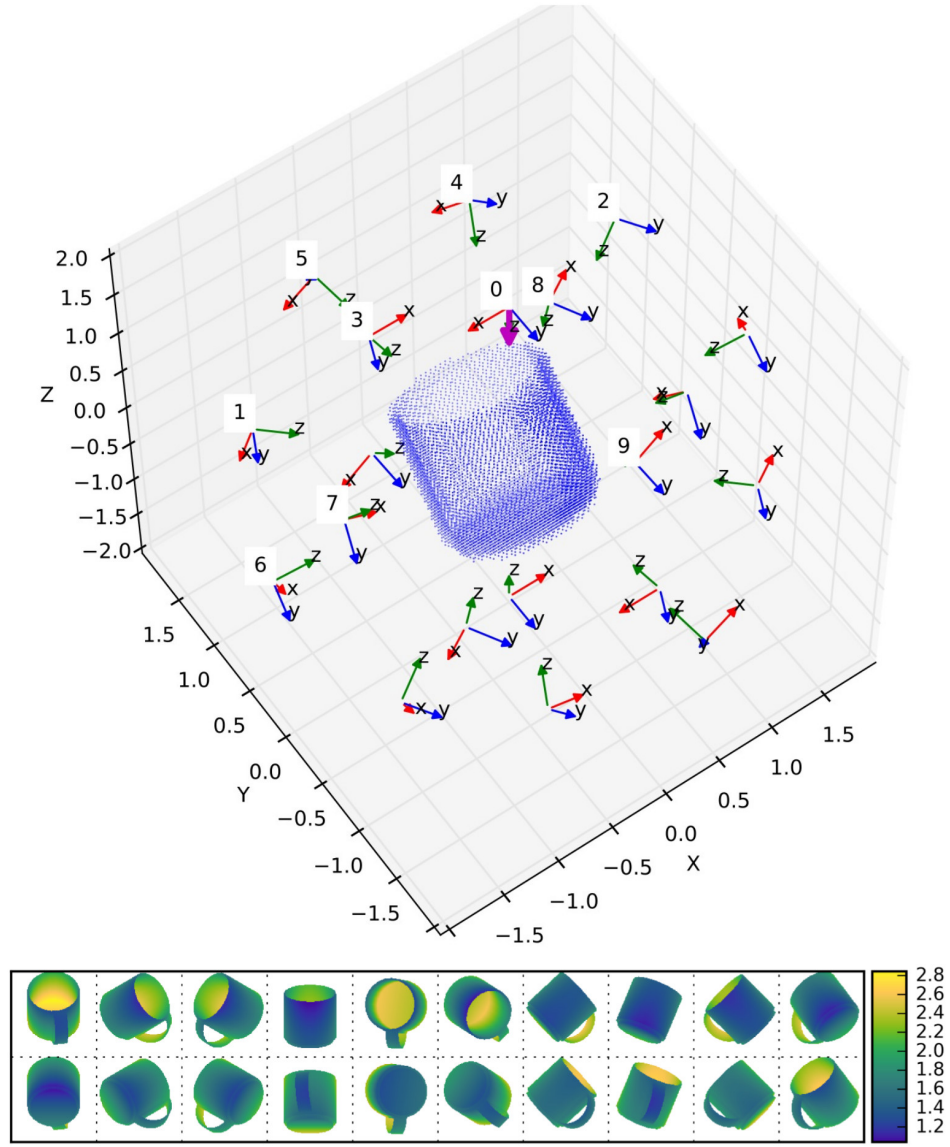


Figure 2.4: Illustration of ground truth generation. Top: GT mesh voxelized in camera coordinates which are used in the viewer-centered voxel baseline experiment. The numeric labels around the model are the indices of the viewpoints from which the multi-surface depth images (shown above) were rendered, left to right. Viewpoint-0, whose camera is located at the origin, *always* corresponds to the input view, and the relative transformation from Viewpoint-0 to all the other viewpoints is constant throughout all our experiments. Bottom: Multi-surface projections are offset relative to the input view. This viewer-centered approach does not require alignment between the 3D models and allows the network to be trained in an unsupervised manner on synthetic data.

2.5 Experiments

We first describe the datasets (Sec. 2.5.1) and evaluation metrics (Sec. 2.5.2), then discuss results in Section 2.6. In all experiments, we train the networks on synthetically generated images. A single training example for the multi-surface network is the input-output pair $(x_d, x_s) \rightarrow \{(s^{(k)}, d_f^{(k)}, d_b^{(k)})\}_{k=0..9}$ where (x_d, x_s) is the input depth image and segmentation, and the orthographic depth images $(s^{(k)}, d_f^{(k)}, d_b^{(k)})$ serve as the output ground truth. The k -th ground truth silhouette has associated front and back depth images. Each image is uniformly scaled to fit within 128x128 pixels. Training examples for the voxel prediction network consist of input-output pairs $(x_d, x_s) \rightarrow V$, where V is a grid of ground truth voxels (size 48x48x48 for the input depth experiments, and 32x32x32 for the input RGB experiments).

2.5.1 Datasets

3D shape from single depth We use the SHREC’12 dataset for comparison with the exemplar retrieval approach by Rock *et al.* [61] on predicting novel views, instances, and classes. Novel views require the least generalization (the same shape is seen in training), and novel classes require the most (no instances from the same category seen during training). This dataset has a training set consisting of 22,500 training + 6,000 validation examples and has 600 examples in each of the three test evaluation sets, using the standard splits [61]. The 3D models in the dataset are aligned to each other, so that they can be used for both viewer-centered and object-centered prediction. Results are shown in Fig. 2.5 and Tables 2.1, 2.2, 2.3, and 2.4.

3D shape from real-world RGB images We also perform novel model experiments on RGB images. We use RenderForCNN’s [78] rendering pipeline and generate 2.4M synthetic

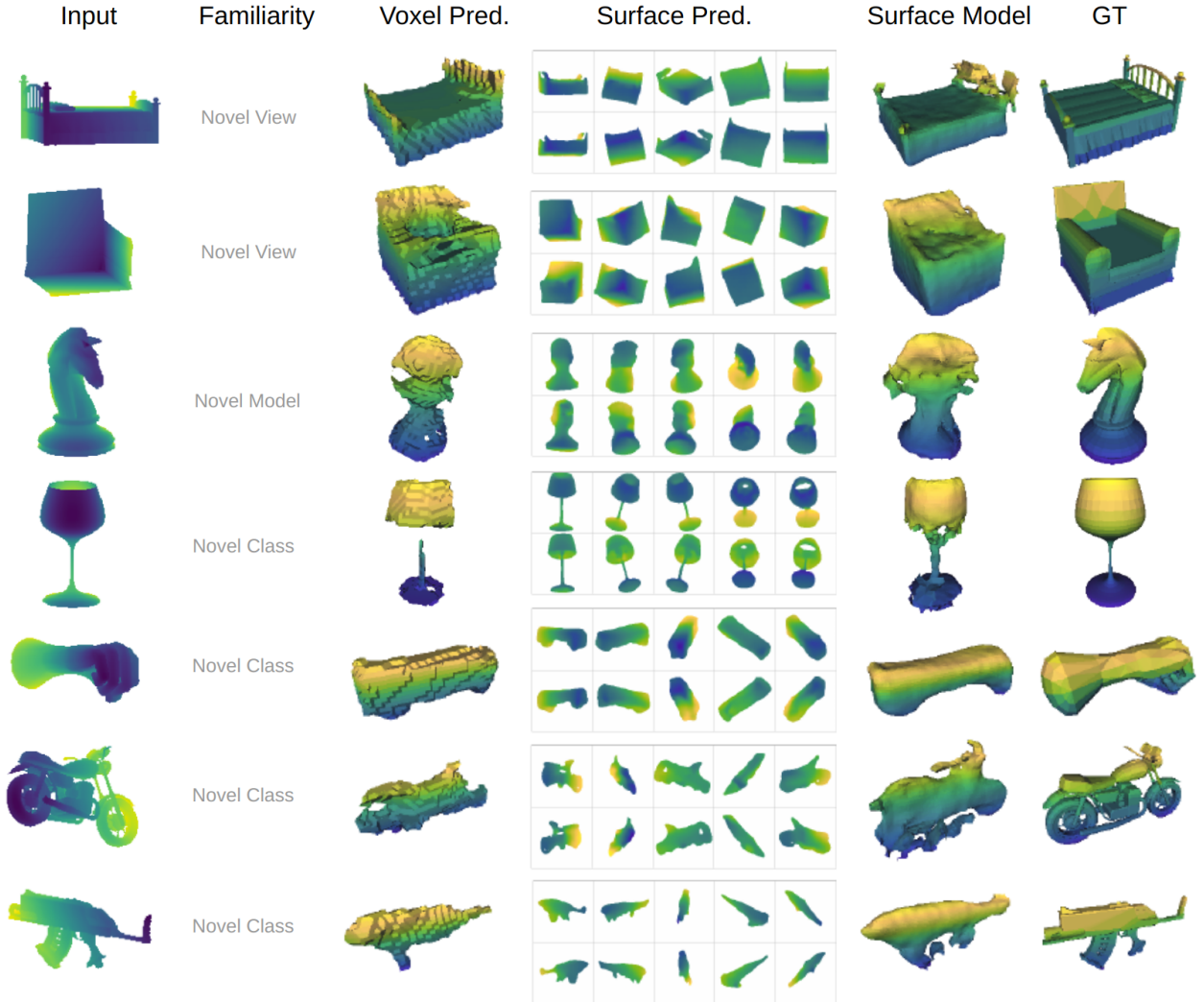


Figure 2.5: Multi-surfaces vs. Voxels. “Novel View” means other views of that shape instance were seen during training. “Novel Model” means other instances from the same class were seen in training. “Novel Class” means that no instances from that category were seen during training. Under multi-surface, a subset of the predicted depth maps are shown, as well as the complete reconstructed shape. The multi-surface approach tends to generalize better for novel classes.

training examples using the ShapeNetCore dataset along with target depth and voxel representations. In this dataset, there are 34,000 3D CAD models from 12 object categories. We perform quantitative evaluation of the resulting models on real-world RGB images using the PASCAL 3D+ dataset [99]. We train 3D-R2N2’s network [5] from scratch using the same dataset and compare evaluation results. The results we report here differ from those in the original paper due to differences in the training and evaluation sets. Specifically, the

Mean	Surface Distance			Voxel IoU		
	NovelClass	NovelModel	NovelView	NovelClass	NovelModel	NovelView
Voxels	0.0950	0.0619	0.0512	0.4569	0.5176	0.6969
Multi-surfaces	0.0759	0.0622	0.0494	0.4914	0.5244	0.6501
Rock <i>et al.</i> [61]	0.0827	0.0604	0.0639	0.5320	0.5888	0.6374

Table 2.1: 3D shape prediction from a single depth image on the SHREC’12 dataset used by [61], comparing results for voxel and multi-surface decoders trained to produce models in a viewer-centered coordinate frame. Rock *et al.* [61] also predicts in viewer-centered coordinates.

	NovelView	NovelModel	NovelClass
View-centered	0.714	0.570	0.517
Obj-centered	0.902	0.474	0.309

Table 2.2: Voxel IoU of predicted and ground truth values (mean, higher is better), using the voxel network. Trained for 45 epochs with batch size 150, learning rate 0.0001.

results reported in [5] are obtained after fine-tuning on the PASCAL 3D+ dataset, which is explicitly discouraged in [99] because the same 3D model exemplars are used for train and test examples.

Thus, we train on renderings and test on real RGB images of objects that may be partly occluded and have background clutter – a challenging task. Results are shown in Tables 2.5 and in Figure 2.6.

	NovelView	NovelModel	NovelClass
View-centered	0.807	0.706	0.670
Obj-centered	0.921	0.586	0.416

Table 2.3: Silhouette IoU, using the 6-view multi-surface network (mean, higher is better).

	NovelView	NovelModel	NovelClass
View-centered	0.011	0.016	0.0207
Obj-centered	0.004	0.035	0.0503

Table 2.4: Depth error, using the 6-view multi-surface network (mean, lower is better).

Category	[5] (OC)	[5] (VC)	Ours (OC)	Ours (VC)
aero	0.359	0.201	0.362	0.289
bike	0.535	0.106	0.362	0.272
boat	0.366	0.236	0.331	0.259
bottle	0.617	0.454	0.643	0.576
bus	0.387	0.273	0.497	0.556
car	0.462	0.400	0.566	0.582
chair	0.325	0.221	0.362	0.332
d.table	0.081	0.023	0.122	0.118
mbike	0.474	0.167	0.487	0.366
sofa	0.602	0.447	0.555	0.538
train	0.340	0.192	0.301	0.257
tv	0.376	0.164	0.383	0.397
mean	0.410	0.240	0.414	0.379

Table 2.5: Per-category voxel IoU on PASCAL 3D+ using our multi-surface network and the voxel-based 3D-R2N2 network [5]. Although the network trained to produce object-centered (OC) models performs slightly better quantitatively (for multi-surface), the viewer-centered (VC) model tends to produce better qualitative results sometimes with misaligned pose (see Appendix A for additional examples).

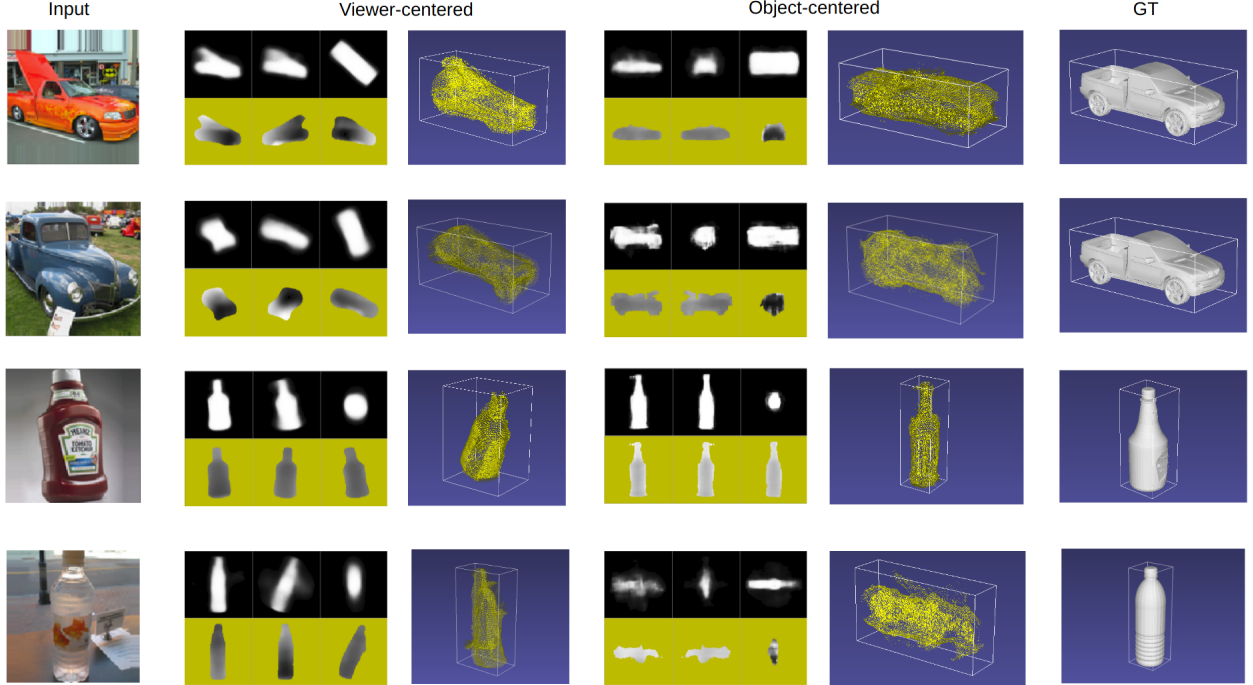


Figure 2.6: **RGB-based shape prediction examples:** On left, is the input image. We show predicted depth maps and silhouettes from three views and a merged point cloud from all views, produced by the networks trained with object-centered coordinates and with viewer-centered coordinates. Viewer-centered tends to generalize better while object-centered sometimes produces a model that looks good but is from entirely the wrong category. In viewer-centered, the encoder learns to map inputs together if they correspond to similar shapes in similar poses, learning a viewpoint-sensitive representation. In object-centered, the encoder learns to map different views of the same object together, learning a viewpoint-invariant representation.

2.5.2 Evaluation metrics and processes

Voxel intersection-over-union Given a mesh reconstructed from the multi-surface prediction (which may not be watertight), we obtain a solid representation by voxelizing the mesh surface into a hollow volume and then filling in the holes using ray tracing. All voxels not visible from the outside are filled. Visibility is determined as follows: from the center of each voxel, we scatter 1000 rays and the voxel is considered visible if any of them can reach the edge of the voxel grid. We compute intersection-over-union (IoU) with the corresponding ground truth voxels, defined as the number of voxels filled in both representations divided by the number of voxels filled in at least one.

Surface distance We also evaluate with a surface distance metric similar to [61], which tends to correspond better to qualitative judgments of accuracy when there are thin structures. The distance between surfaces is approximated as the mean of *point-to-triangle* distances from i.i.d. sampled points on the ground truth mesh to the closest points on surface of the reconstructed mesh, and vice versa. We utilize a KD-tree to find the closest point on the mesh. To ensure scale invariance of this measure across datasets, we divide the resulting value by the mean distance between points sampled on the GT surface. The points were sampled at a density of 300 points per unit area. To evaluate surface distance for voxel-prediction models, we use Marching Cubes to obtain the mesh from the prediction.

Image-based measures For multi-surface experiments, in addition to voxel IoU and surface distance, we also evaluate using silhouette intersection-over-union and depth error averaged over the predicted views. Sometimes, even when the predictions for individual views are quite accurate, slight inconsistencies or oversmoothing by the final surface estimation can reduce the accuracy of the 3D model.

2.6 Discussion

Multi-surface vs. voxel shape representations: Table 2.1 compares performance of multi-surface and voxel-based representations for shape prediction. Quantitatively, multi-surface outperforms for novel class and performs similarly for novel view and novel instance. We also find that the 3D shapes produced by the multi-surface model look better qualitatively, as they can encode higher resolution.

We observe that it is generally difficult to learn and reconstruct thin structures such as the legs of chairs and tables. In part this is a learning problem, as discussed in Choy *et al.* [5]. Our qualitative results suggest that silhouettes are generally better for learning and

predicting thin object parts than voxels, but the information is often lost during surface reconstruction due to the sparsity of available data points. We expect that improved depth fusion and mesh reconstruction would likely yield even better results. As shown in Fig. 2.6, the multi-surface representation can more directly be output as a point cloud by skipping the reconstruction step. This avoids errors that can occur during the surface reconstruction but is more difficult to quantitatively evaluate.

Viewer-centered vs. object-centered coordinates When comparing performance of predicting in viewer-centered coordinates vs. object-centered coordinates, it is important to remember that only viewer-centered encodes pose and, thus, is more difficult. Sometimes, the 3D shape produced by viewer-centered prediction is very good, but the pose is misaligned, resulting in poor quantitative results for that example. Even so, in Tables 2.2, 2.3, and 2.4, we observe a clear advantage for viewer-centered prediction for novel models and novel classes, while object-centered outperforms for novel views of object instances seen during training. For object-centered prediction, two views of the same object should produce the same 3D shape, which encourages memorizing the observed meshes. Under viewer-centered, the predicted mesh must be oriented according to the input viewpoint, so multiple views of the same object should produce different 3D shapes (which are related by a 3D rotation). This requirement seems to improve the generalization capability of viewer-centered prediction to shapes not seen during training.

In Table 2.5, we see that object-centered prediction quantitatively slightly outperforms for RGB images. In this case, training is performed on rendered meshes, while testing is performed on real images of novel object instances from familiar categories. Qualitatively, we find that viewer-centered prediction tends to produce much more accurate shapes, but that the pose is sometimes wrong by 15-20 degrees, perhaps as a result of dataset transfer.

Qualitative results support our initial hypothesis that object-centered models tend to corre-

respond more directly to category recognition. We see in Figure 2.6, that the object-centered model often predicts a shape that looks good but is an entirely different object category than the input image. The viewer-centered model tends not to make these kinds of mistakes and, instead, errors tend to be overly simplified shapes or slightly incorrect poses.

Implications for object recognition While not a focus of our study, we also trained an object classifier using the 4096-dimensional encoding layer of the viewer-centric model as input features for a single hidden-layer classifier. The resulting classifier outperformed by 1% a Resnet classifier that was trained end-to-end on the same data. This indicates that models trained to predict shape and pose contain discriminative information that is highly useful for predicting object categories and may, in some ways, generalize better than models learned to predict categories directly. More study is needed in this direction.

2.7 Conclusion

Recent methods to produce 3D shape from a single image have used a variety of representation for shape (voxels, octrees, multiple depth maps). By utilizing the same encoder architecture for volumetric and surface-based representations, we are able to more directly compare their efficacy. Our experiments show an advantage for surface-based representations in predicting novel object shapes, likely because they can encode shape details with fewer parameters. Nearly all existing methods predict object shape in object-centered coordinates, but our experiments show that learning to predict shape in viewer-centered coordinates leads to better generalization for novel objects. Further improvements in surface-based prediction could be obtained through better alignment and fusing of produced depth maps. More research is also needed to verify whether recently proposed octree-based representations [84] close the gap with surface-based representations. In addition, the relationship between object categorization and shape/pose prediction requires further exploration. Novel view prediction

and shape completion could provide a basis for unsupervised learning of features that are effective for object category and attribute recognition.

Viewer-centered prediction transfers more robustly to novel objects, while object-centered prediction works better when the category is familiar. The next chapter takes the viewer-centered approach to a harder setting: full indoor scenes.

Chapter 3

3D Scene Reconstruction with Multi-Layer Depth and Epipolar Transformers

The previous chapter established that viewer-centered prediction generalizes better to unfamiliar objects than object-centered prediction. Here we apply the same approach to full indoor scenes, where the hidden geometry includes not only the backs of individual objects but also surfaces occluded by other objects and the room envelope. Rather than detecting and reconstructing objects individually, we predict the complete scene geometry directly from the input viewpoint.

We tackle the problem of automatically reconstructing a complete 3D model of a scene from a single RGB image. This challenging task requires inferring the shape of both visible and occluded surfaces. Our approach utilizes viewer-centered, multi-layer representation of scene

© 2019 IEEE. Reprinted, with permission, from D. Shin, Z. Ren, E. Sudderth, and C. Fowlkes, “3D Scene Reconstruction with Multi-Layer Depth and Epipolar Transformers,” in *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019.

geometry adapted from recent methods for single object shape completion. To improve the accuracy of view-centered representations for complex scenes, we introduce a novel “Epipolar Feature Transformer” that transfers convolutional network features from an input view to other virtual camera viewpoints, and thus better covers the 3D scene geometry. Unlike existing approaches that first detect and localize objects in 3D, and then infer object shape using category-specific models, our approach is fully convolutional, end-to-end differentiable, and avoids the resolution and memory limitations of voxel representations. We demonstrate the advantages of multi-layer depth representations and epipolar feature transformers on the reconstruction of a large database of indoor scenes.

3.1 Introduction

When we examine a photograph of a scene, we not only perceive the 3D shape of visible surfaces, but effortlessly infer the existence of many invisible surfaces. We can make strong predictions about the complete shapes of familiar objects despite viewing only a single, partially occluded aspect, and can infer information about the overall volumetric occupancy with sufficient accuracy to plan navigation and interactions with complex scenes. This remains a daunting visual task for machines despite much recent progress in detecting individual objects and making predictions about their shape. *Convolutional neural networks* (CNNs) have proven incredibly successful as tools for learning rich representations of object identity which are invariant to intra-category variations in appearance. Predicting 3D shape rather than object category has proven more challenging since the output space is higher dimensional and carries more structure than simple regression or classification tasks.

Early successes at using CNNs for shape prediction leveraged direct correspondences between the input and output domain, regressing depth and surface normals at every input pixel [12]. However, these so-called 2.5D representations are incomplete: they don’t make predictions

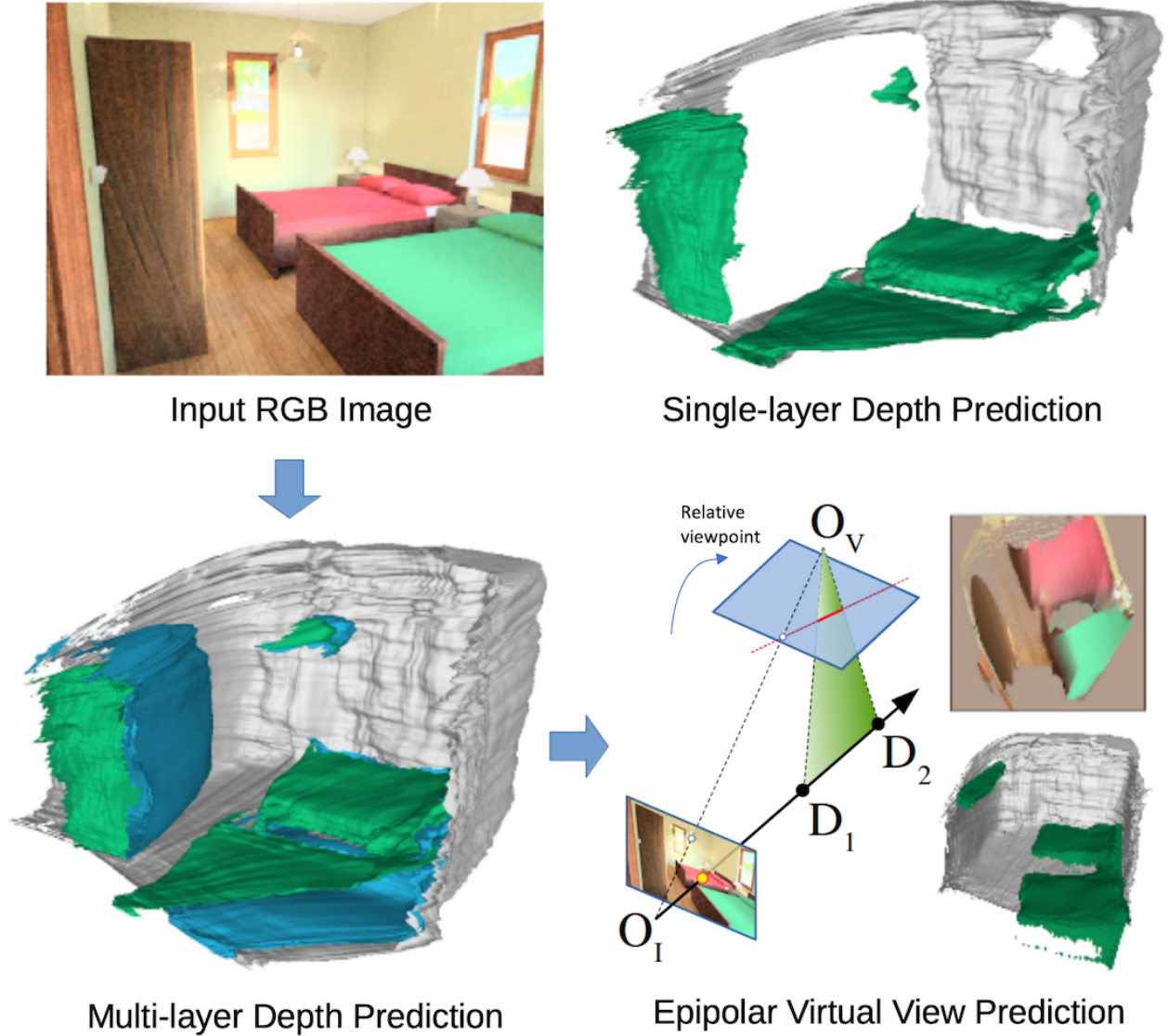


Figure 3.1: Given a single input view of a scene (top left), we would like to predict a complete geometric model. Depth maps (top right) provide an efficient representation of scene geometry but are incomplete, leaving large holes (e.g., the wardrobe). We propose multi-layer depth predictions (bottom left) that provide complete view-based representations of shape, and introduce an epipolar transformer network that allows view-based inference and prediction from virtual viewpoints (like overhead views, bottom right).

about the back side of objects or other occluded surfaces. Several recent methods instead manipulate voxel-based representations [75] and use convolutions to perform translation-covariant computations in 3D. This provides a more complete representation than 2.5D models, but suffers from substantial storage and computation expense that scales cubically

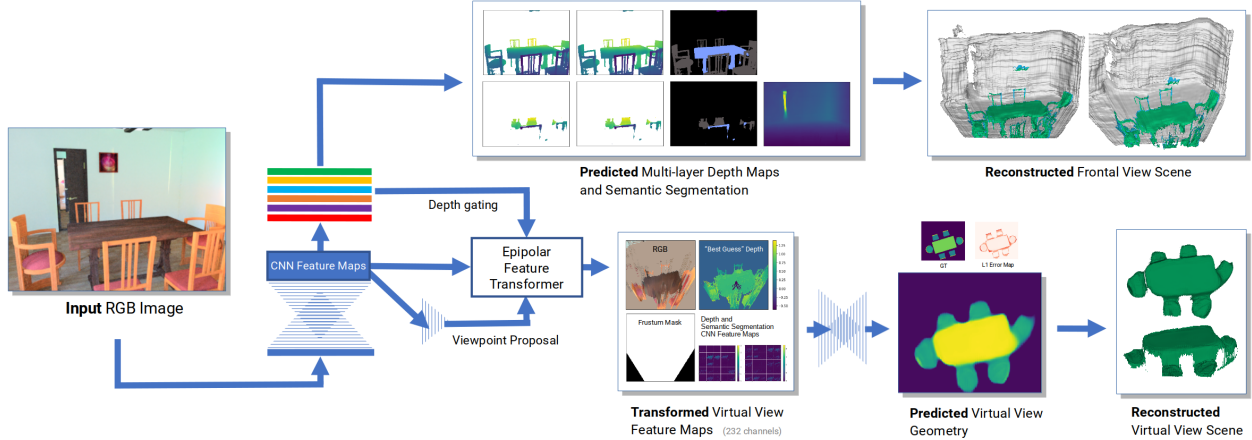


Figure 3.2: Overview of our system for reconstructing a complete 3D scene from a single RGB image. We first predict a multi-layer depth map that encodes the depths of front and back object surfaces as seen from the input camera. Given the extracted feature map and predicted multi-layer depths, the epipolar feature transformer network transfers features from the input view to a virtual overhead view, where the heights of observed objects are predicted. Semantic segmentation masks are inferred and inform our geometry estimates, but explicit detection of object instances is not required, increasing robustness.

with resolution of the volume being modeled (without specialized representations like oc-trees [59]). Other approaches represent shape as an unstructured point cloud [56, 77], but require development of suitable convolutional operators [16, 93] and fail to capture surface topology.

In this chapter, we tackle the problem of automatically reconstructing a *complete* 3D model of a scene from a single RGB image. As depicted in Figures 3.1 and 3.2, our approach uses an alternative shape representation that extends view-based 2.5D representations to a complete 3D representation. We combine *multi-layer* depth maps that store the depth to multiple surface intersections along each camera ray from a given viewpoint, with *multi-view* depth maps that record surface depths from different camera viewpoints.

While multi-view and multi-layer shape representations have been explored for single object shape completion, for example in Chapter 2 [67], we argue that multi-layer depth maps are particularly well suited for representing full 3D scenes. *First*, they compactly capture high-resolution details about the shapes of surfaces in a large scene. Voxel-based representations

allocate a huge amount of resources to simply modeling empty space, ultimately limiting shape fidelity to much lower resolution than is provided by cues like occluding contours in the input image [75]. A multi-layer depth map can be viewed as a run-length encoding of dense representations that stores only transitions between empty and occupied space. *Second*, view-based depths maintain explicit correspondence between input image data and scene geometry. Much of the work on voxel and point cloud representations for single object shape prediction has focused on predicting a 3D representation in an object-centered coordinate system. Utilizing such an approach for scenes requires additional steps of detecting individual objects and estimating their pose in order to place them back into some global scene coordinate system [88]. In contrast, view-based multi-depth predictions provide a single, globally coherent scene representation that can be computed in a “fully convolutional” manner from the input image.

One limitation of predicting a multi-layer depth representation from the input image viewpoint is that the representation cannot accurately encode the geometry of surfaces which are nearly tangent to the viewing direction. In addition, complicated scenes may contain many partially occluded objects that require a large number of layers to represent completely. We address this challenge by predicting additional (multi-layer) depth maps computed from virtual viewpoints elsewhere in the scene. To link these predictions from virtual viewpoints with the input viewpoint, we introduce a novel *Epipolar Feature Transformer* (EFT) network module. Given the relative poses of the input and virtual cameras, we transfer features from a given location in the input view feature map to the corresponding epipolar line in the virtual camera feature map. This transfer process is modulated by predictions of surface depths from the input view in order to effectively re-project features to the correct locations in the overhead view.

To summarize our contributions, we propose a view-based, multi-layer depth representation that enables fully convolutional inference of 3D scene geometry and shape completion. We

also introduce EFT networks that provide geometrically consistent transfer of CNN features between cameras with different poses, allowing end-to-end training for multi-view inference. We experimentally characterize the completeness of these representations for describing the 3D geometry of indoor scenes, and show that models trained to predict these representations can provide better recall and precision of scene geometry than existing approaches based on object detection.

3.2 Related Work

The task of recovering 3D geometry from 2D images has a rich history dating to the visionary work of Roberts [60].

Monocular Object Shape Prediction Single-view 3D shape reconstruction is challenging because the output space is under-constrained. Large-scale datasets like ShapeNet [1, 98] facilitate progress in this area, and recent methods have learned geometric priors for object categories [32, 97], disentangled primitive shapes from objects [18, 118], or modeled surfaces [21, 67, 110]. Other work aims to complete the occluded geometric structure of objects from a 2.5D image or partial 3D scan [61, 9, 96, 103]. While the quality of such 3D object reconstructions continues to grow [33, 93], applications are limited by the assumption that input images depict a single, centered object.

3D Scene Reconstruction We seek to predict the geometry of full scenes containing an unknown number of objects; this task is significantly more challenging than object reconstruction. Tulsiani *et al.* [88] factorize 3D scenes into detected objects and room layout by integrating separate methods for 2D object detection, pose estimation, and object-centered shape prediction. Given a depth image as input, Song *et al.* [75] propose a volumetric reconstruction algorithm that predicts semantically labeled 3D voxels. Another general ap-

proach is to retrieve exemplar CAD models from a large database and reconstruct parts of scenes [26, 119, 20], but the complexity of CAD models may not match real-world environments. While our goals are similar to Tulsiani *et al.*, our multi-layered depth estimates provide a denser representation of complex scenes.

Representations for 3D Shape Prediction Most recent methods use voxel representations to reconstruct 3D geometry [5, 75, 71, 87, 70], in part because they easily integrate with 3D CNNs [98] for high-level recognition tasks [46]. Other methods [13, 40] use dense point clouds representations. Classic 2.5D depth maps [12, 4] recover the geometry of visible scene features, but do not capture occluded regions. In Chapter 2, we empirically compared these representations for object reconstruction. We extend these ideas to whole scenes via a multi-view, multi-layer depth representation that encodes the shape of multiple objects.

Learning Layered Representations Layered representations [92] have proven useful for many computer vision tasks including segmentation [17] and optical flow prediction [80]. For 3D reconstruction, decomposing scenes into layers enables algorithms to reason about object occlusions and depth orderings [24, 72, 95]. Layered 2.5D representations such as the two-layer decompositions of [89, 10] infer the depth of occluded surfaces facing the camera. Our multi-layer depth representation extends this idea by including the depth of back surfaces (equiv. object thickness). We also infer depths from virtual viewpoints far from the input view for more complete coverage of 3D scene geometry. Our use of layers generalizes [58], who used multiple intersection depths to model non-convexities for constrained scenes containing a single, centered object. Concurrently to our work, [52] predicts object-level thicknesses for volumetric RGB-D fusion and [15] estimates 3D human shape.

Multi-view Shape Synthesis Many classic 3D reconstruction methods utilize multi-view inputs to synthesize 3D shapes [25, 73, 7]. Given monocular inputs, several recent methods explore ways of synthesizing object appearance or image features from novel viewpoints [114,

102, 28, 5, 53, 79]. Other work uses unsupervised learning from stereo or video sequences to reason about depths [113, 29]. Instead of simply transferring the pixel colors associated with surface points to novel views, we transfer whole CNN feature maps over corresponding object volumes, and thereby produce more accurate and complete 3D reconstructions.

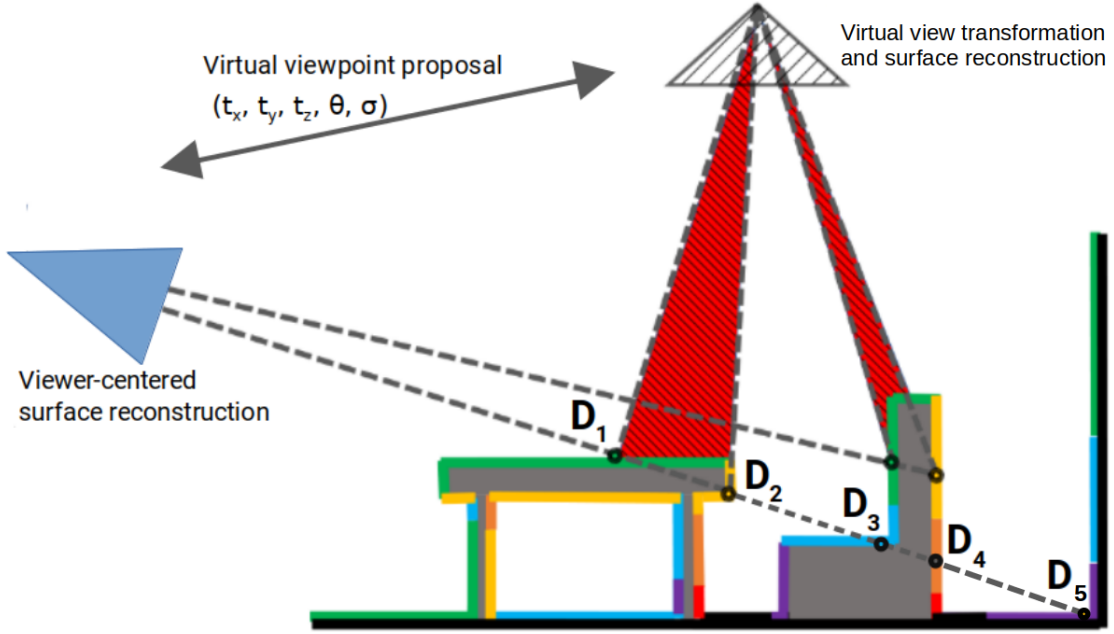
3.3 Reconstruction with Multi-Layer Depth

Traditional depth maps record the depth at which a ray through a given pixel first intersects a surface in the scene. Such 2.5D representations of scene geometry accurately describe visible surfaces, but cannot encode the shape of partially occluded objects, and may fail to capture the complete 3D shape of unoccluded objects (due to self-occlusion). We instead represent 3D scene geometry by recording multiple surface intersections for each camera ray. As illustrated in Figure 3.3(a), some rays may intersect many object surfaces and require several layers to capture all details. But as the number of layers grows, multi-layer depths completely represent 3D scenes with multiple non-convex objects.

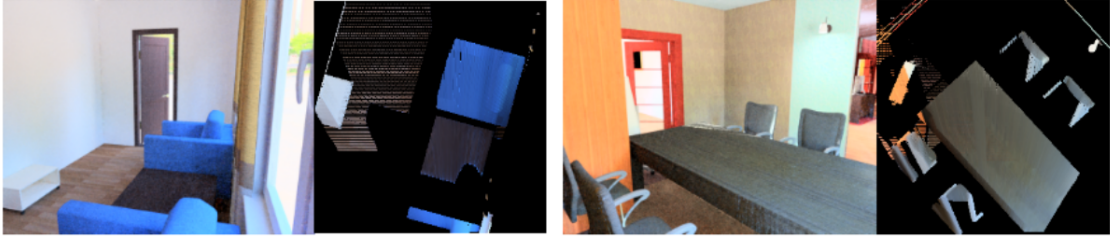
We use experiments to empirically determine a fixed number of layers that provides good coverage of typical natural scenes, while remaining compact enough for efficient learning and prediction. Another challenge is that surfaces that are nearly tangent to input camera rays are not well represented by a depth map of fixed resolution. To address this, we introduce an additional virtual view where tangent surfaces are sampled more densely (see Section 3.4).

$\bar{D}_1$	$\bar{D}_{1,2}$	$\bar{D}_{1,2,3}$	$\bar{D}_{1..4}$	$\bar{D}_{1..5}$	$\bar{D}_{1..5} + \text{Ovh.}$
0.237	0.427	0.450	0.480	0.924	0.932

Table 3.1: Scene surface coverage (recall) of ground truth depth layers with a 5cm threshold. Our predictions cover 93% of the scene geometry inside the viewing frustum.



(a) **3D volume inference** through multi-layer depth images



(b) **Input image and transformed color features** using $\bar{D}_1$ and $\bar{D}_2$.

Figure 3.3: Epipolar transfer of features from the input image to a virtual overhead view. Given multi-layer depth predictions of surface entrances and exits, each pixel in the input view is mapped to zero, one, or two segments of the corresponding epipolar line in the virtual view.

3.3.1 Multi-Layer Depth Maps from 3D Geometry

In our experiments, we focus on a five-layer model designed to represent key features of 3D scene geometry for typical indoor scenes. To capture the overall room layout, we model the room envelope (floors, walls, ceiling, windows) that defines the extent of the space. We define the depth D_5 of these surfaces to be the *last* layer of the scene.

To model the shapes of observed objects, we trace rays from the input view and record the

first intersection with a visible surface in depth map D_1 . This resembles a standard depth map, but excludes the room envelope. If we continue along the same ray, it will eventually exit the object at a depth we denote by D_2 . For non-convex objects the ray may intersect the same object multiple times, but we only record the *last* exit in D_2 . As many indoor objects have large convex parts, the D_1 and D_2 layers are often sufficient to accurately reconstruct a large proportion of foreground objects in real scenes. While room envelopes typically have a very simple shape, the prediction of occluded structure behind foreground objects is more challenging. We define layer $D_3 > D_2$ as the depth of the next object intersection, and D_4 as the depth of the exit from that second object instance.

We let $(\bar{D}_1, \bar{D}_2, \bar{D}_3, \bar{D}_4, \bar{D}_5)$ denote the ground truth multi-layer depth maps derived from a complete 3D model. Since not all viewing rays intersect the same number of objects (e.g., when the room envelope is directly visible), we define a binary mask $\bar{M}_\ell$ which indicates the pixels where layer ℓ has support. Note that $\bar{M}_1 = \bar{M}_2$, and $\bar{M}_3 = \bar{M}_4$, since D_2 (first instance exit) has the same support as D_1 . Experiments in Section 3.5 evaluate the relative importance of different layers in modeling realistic 3D scenes.

3.3.2 Predicting Multi-Layer Depth Maps

To learn to predict five-channel multi-layer depths $\mathcal{D} = (D_1, D_2, D_3, D_4, D_5)$ from images, we train a standard encoder-decoder network with skip connections, and use the Huber loss $\rho_h(.,.)$ to measure prediction errors:

$$L_d(\mathcal{D}) = \sum_{\ell=1}^5 \left(\frac{\bar{M}_\ell}{\|\bar{M}_\ell\|_1} \right) \cdot \rho_h(D_\ell, \bar{D}_\ell). \quad (3.1)$$

We also predict semantic segmentation masks for the first and third layers. The structure of the semantic segmentation network is similar to the multi-layer depth prediction network, except that the output has 80 channels (40 object categories in each of two layers), and errors

are measured via the cross-entropy loss. To reconstruct complete 3D geometry from multi-layer depth predictions, we use predicted masks and depths to generate meshes corresponding to the front and back surfaces of visible and partially occluded objects, as well as the room envelope. Without the back surfaces [65], ground truth depth layers $\bar{D}_{1,3,5}$ cover only 82% of the scene geometry inside the viewing frustum (vs. 92% including back surfaces of objects, see Table 3.1).

3.4 Epipolar Feature Transformer Networks

To allow for richer view-based scene understanding, we would like to relate features visible in the input view to feature representations in other views. To achieve this, we transfer features computed in input image coordinates to the coordinate system of a “virtual camera” placed elsewhere in the scene. This approach more efficiently covers some parts of 3D scenes than single-view, multi-layer depths.

Figure 3.2 shows a block diagram of our *Epipolar Feature Transformer* (EFT) network. Given features F extracted from the image, we choose a virtual camera location with transformation mapping T and transfer weights W , and use these to warp F to create a new “virtual view” feature map G . Like *spatial transformer networks* (STNs) [27] we perform a parametric, differentiable “warping” of a feature map. However, EFTs incorporate a weighted pooling operation informed by multi-view geometry.

Epipolar Feature Mapping Image features at spatial location (s, t) in an input view correspond to information about the scene which lies somewhere along the ray

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = z \mathbf{K_I}^{-1} \begin{bmatrix} s \\ t \\ 1 \end{bmatrix}, \quad z \geq 0,$$

where $\mathbf{K}_I \in \mathbb{R}^{3 \times 3}$ encodes the input camera intrinsic parameters, as well as the spatial resolution and offset of the feature map. z is the depth along the viewing ray, whose image in a virtual orthographic camera is given by

$$\begin{bmatrix} u(s, t, z) \\ v(s, t, z) \end{bmatrix} = \mathbf{K}_V \left(z \mathbf{R} \mathbf{K}_I^{-1} \begin{bmatrix} s \\ t \\ 1 \end{bmatrix} + \mathbf{t} \right), \quad z \geq 0.$$

Here $\mathbf{K}_V \in \mathbb{R}^{2 \times 3}$ encodes the virtual view resolution and offset, and $\mathbf{R}$ and $\mathbf{t}$ the relative pose.⁰ Let $T(s, t, z) = (u(s, t, z), v(s, t, z))$ denote the forward mapping from points along the ray into the virtual camera, and $\Omega(u, v) = \{(s, t, z) : T(s, t, z) = (u, v)\}$ be the pre-image of (u, v) .

Given a feature map computed from the input view $F(s, t, f)$, where f indexes the feature dimension, we synthesize a new feature map G corresponding to the virtual view. We consider general mappings of the form

$$G(u, v, f) = \frac{\sum_{(s, t, z) \in \Omega(u, v)} F(s, t, f) W(s, t, z)}{\sum_{(s, t, z) \in \Omega(u, v)} W(s, t, z)},$$

where $W(s, t, z) \geq 0$ is a gating function that may depend on features of the input image.¹ When $\Omega(u, v)$ is empty, we set $G(u, v, f) = 0$ for points (u, v) outside the viewing frustum of the input camera, and otherwise interpolate feature values from those of neighboring virtual-view pixels.

Choice of the Gating Function W By design, the transformed features are differentiable with respect to F and W . Thus in general we could assign a loss to predictions from the virtual camera, and learn an arbitrary gating function W from training data. However, we

⁰For a perspective model the righthand side is scaled by $z'(s, t, z)$, the depth from the virtual camera of the point at location z along the ray.

¹For notational simplicity, we have written G as a sum over a discrete set of samples Ω . To make G differentiable with respect to the virtual camera parameters, we perform bilinear interpolation.

instead propose to leverage additional geometric structure based on predictions about the scene geometry produced by the frontal view.

Suppose we have a scene depth estimate $D_1(s, t)$ at every location in the input view. To simplify occlusion reasoning we assume that relative to the input camera view, the virtual camera is rotated around the x -axis by $\theta < 90^\circ$ and translated in y and z to sit above the scene so that points which project to larger t in the input view have larger depth in the virtual view. Setting the gating function as

$$W_{\text{surf}}^1(s, t, z) = \delta[D_1(s, t) = z] \prod_{\hat{t}=0}^{t-1} \delta \left[D_1(s, \hat{t}) \neq \frac{t/f - \tan \theta}{\hat{t}/f - \tan \theta} \cdot z \right]$$

where f is the focal length, yields an epipolar feature transform that *re-projects* each feature at input location (s, t) into the overhead view via the depth estimate D_1 , but only in cases where it is not occluded by a patch of surface higher up in the scene. In our experiments we compute W_{surf}^ℓ for each D_ℓ , $\ell \in \{1, 2, 3, 4\}$, and use $W_{\text{surf}} = \max_\ell W_{\text{surf}}^\ell$ to transfer input view features to both visible and occluded surfaces in the overhead feature map. We implement this transformation using a z-buffering approach by traversing the input feature map from bottom to top, and overwriting cells in the overhead feature map.

Figure 3.3(b) illustrates this feature mapping applied to color features using the ground-truth depth map for a scene. In some sense, this surface-based reprojection is quite conservative because it leaves holes in the interior of objects (e.g., the interior of the orange wood cabinet). If the frontal view network features at a given spatial location encode the presence, shape, and pose of some object, then those features really describe a whole volume of the scene behind the object surface. It may thus be preferable to instead transfer the input view features to the entire expected volume in the overhead representation.

To achieve this, we use the multi-layer depth representation predicted by the frontal view to define a range of scene depths to which the input view feature should be mapped. If $D_1(s, t)$

is the depth of the front surface and $D_2(s, t)$ is the depth at which the ray exits the back surface of an object instance, we define a volume-based gating function:

$$W_{\text{vol}}(s, t, z) = \delta[z \in (D_1(s, t), D_2(s, t))].$$

As illustrated in Figure 3.3(a), volume-based gating copies features from the input view to entire segments of the epipolar line in the virtual view. In our experiments we use this gating to generate features for (D_1, D_2) and concatenate them with a feature map generated using (D_3, D_4) .

Overhead Viewpoint Generation For cluttered indoor scenes, there may be many overlapping objects in the input view. Overhead views of such scenes typically have much less occlusion and should be simpler to reason about geometrically. We thus select a virtual camera that is roughly overhead and covers the scene content visible from the reference view. We assume the input view is always taken with the gravity direction in the y, z plane. We parameterize the overhead camera relative to the reference view by a translation (t_x, t_y, t_z) which centers it over the scene at fixed height above the floor, a rotation θ which aligns the overhead camera to the gravity direction, and a scale σ that captures the radius of the orthographic camera frustum.

3.5 Experiments

Because we model complete descriptions of the ground-truth 3D geometry corresponding to RGB images, which is not readily available for natural images, we learn to predict multi-layer and multi-view depths from physical renderings of indoor scenes [111] provided by the SUNCG dataset [75].

3.5.1 Generation of Training Data

The SUNCG dataset [75] contains complete 3D meshes for 41,490 houses that we render to generate our training data. For each rendered training image, we extract the subset of the house model that is relevant to the image content, without making assumptions about the room size. We transform the house mesh to the camera’s coordinate system and truncate polygons that are outside the left, top, right, bottom, and near planes of the perspective viewing frustum. Objects that are projected behind the depth image of the room envelope are also removed. The final ground truth mesh that we evaluate against contains all polygons from the remaining objects, as well as the true room envelope.

For each rendered training image, we generate target multi-depth maps and segmentation masks by performing multi-hit ray tracing on the ground-truth geometry. We similarly compute ground-truth height maps for a virtual orthographic camera centered over each scene. To select an overhead camera viewpoint for training that covers the relevant scene content, we consider three heuristics: (i) Convert the true depth map to a point cloud, center the overhead camera over the mean of these points, and set the camera radius to 1.5 times their standard deviation; (ii) Center the overhead camera so that its principal axis lies in the same plane as the input camera, and offset in front of the input view by the mean of the room envelope depths; (iii) Select a square bounding box in the overhead view that encloses all points belonging to objects visible from the input view. None of these heuristics worked perfectly for all training examples, so we compute our final overhead camera view via a weighted average of these three candidates.

3.5.2 Model Architecture and Training

As illustrated in Figure 3.2, given an RGB image, we first predict a multi-layer depth map as well as a 2D semantic segmentation map. Features used to predict multi-layer depths are

then mapped through our EFT network to synthesize features for a virtual camera view, and predict an orthographic height map. We then use the multi-layer depth map, semantic segmentation map, and overhead height map to predict a dense 3D reconstruction of the imaged scene.

We predict multi-layer depth maps and semantic segmentations via a standard convolutional encoder-decoder with skip connections. The network uses dilated convolution and has separate output branches for predicting each depth layer using the Huber loss specified in Section 3.3.2. For segmentation, we train a single branch network using a softmax loss to predict 40 semantic categories derived from the SUNCG mesh labels (see Appendix B for details).

Our overhead height map prediction network takes as input the transformed features of our input view multi-layer depth map. The overhead model integrates 232 channels (see Figure 3.2) including epipolar transformations of a 48-channel feature map from the depth prediction network, a 64-channel feature map from the semantic segmentation network, and the RGB input image. These feature maps are extracted from the frontal networks just prior to the predictive branches. Other inputs include a “best guess” overhead height map derived from frontal depth predictions, and a mask indicating the support of the input camera frustum. The frustum mask can be computed by applying the epipolar transform with $F = 1$, $W = 1$. The best-guess overhead depth map can be computed by applying an unnormalized gating function $W(s, t, z) = z \cdot \delta[D_1(s, t) = z]$ to the y -coordinate feature $F(s, t) = t$.

We also train a model to predict the virtual camera parameters which takes as input feature maps from our multi-depth prediction network, and attempts to predict the target overhead viewpoint (orthographic translation (t_x, t_y) and frustum radius σ) chosen as in Section 3.5.1. While the EFT network is differentiable and our final model can in principle be trained end-to-end, in our experiments we simply train the frontal model to convergence, freeze it, and

then train the overhead model on transformed features without backpropagating overhead loss back into the frontal-view model parameters. We use the Adam optimizer to train all of our models with batch size 24 and learning rate 0.0005 for 40 epochs. The Physically-based Rendering [111] dataset uses a fixed downward tilt camera angle of 11 degrees, so we do not need to predict the gravity angle. At test time, the height of the virtual camera is the same as the input frontal camera and assumed to be known. We show qualitative 3D reconstruction results on the SUNCG test set in Figure 3.4, and more qualitative examples in our supplementary video (see Appendix B).

3.5.3 Evaluation

To reconstruct 3D surfaces from predicted multi-layer depth images as well as the overhead height map, we first convert the depth images and height maps into a point cloud and triangulate vertices that correspond to a 2×2 neighborhood in image space. If the depth values of two adjacent pixels is greater than a threshold $\delta \cdot a$, where δ is the footprint of the pixel in camera coordinates and $a = 7$, we do not create an edge between those vertices. We do not predict the room envelope from the virtual overhead view, so only pixels with height values higher than 5 cm above the floor are considered for reconstruction and evaluation.

Metrics We use precision and recall of surface area as the metric to evaluate how closely the predicted meshes align with the ground truth meshes, which is the native format for SUNCG and ScanNet. Coverage is determined as follows: We uniformly sample points on surface of the ground truth mesh then compute the distance to the closest point on the predicted mesh. We use sampling density $\rho = 10000/\text{meter}^2$ throughout our experiments. Then we measure the percentage of inlier distances for given a threshold. *Recall* is the coverage of the ground truth mesh by the predicted mesh. Conversely, *precision* is the coverage of the predicted mesh by the ground truth mesh.

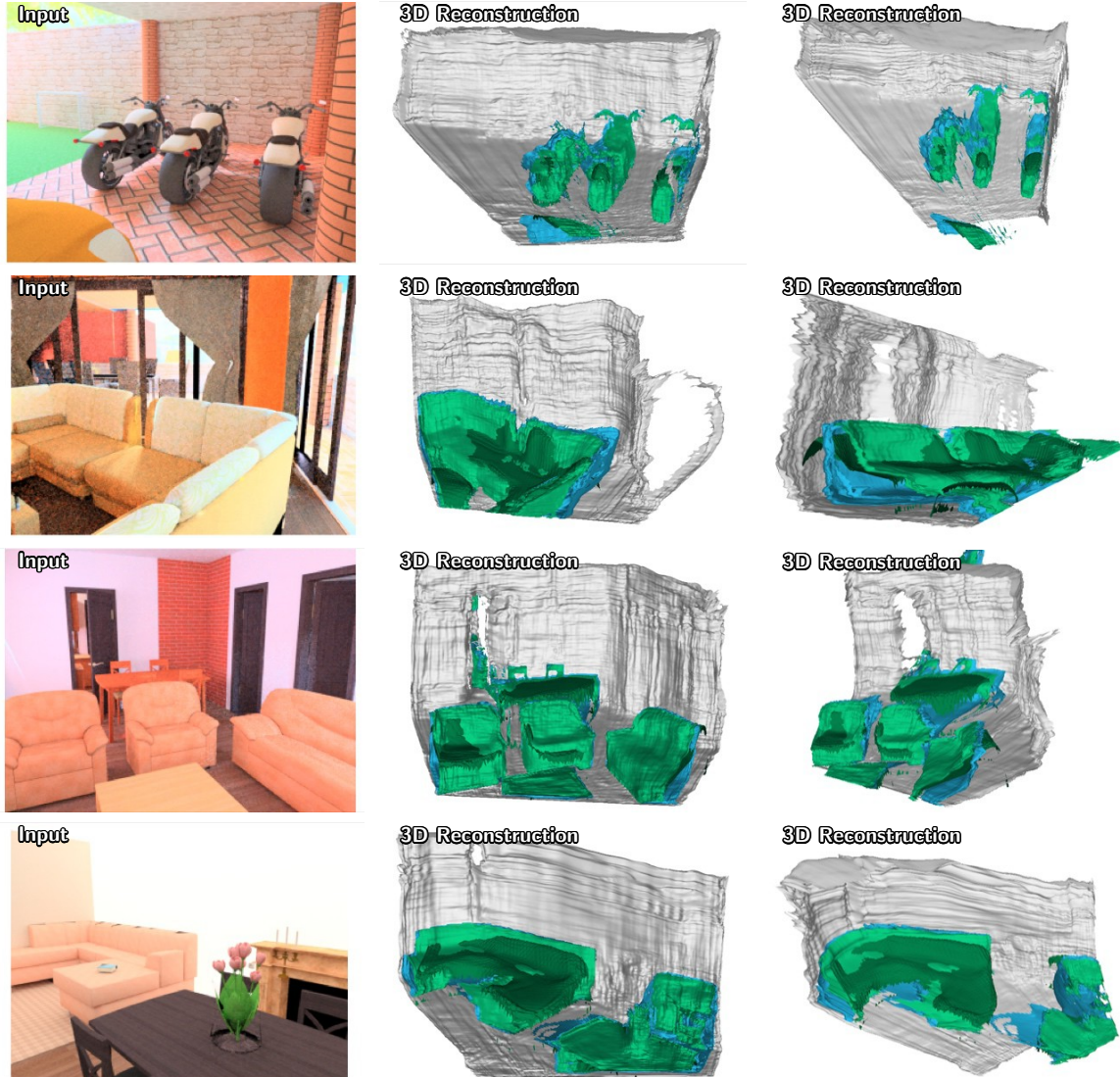


Figure 3.4: Single image scene reconstruction via multi-layer depth maps. Estimates of the front (green) and back (cyan) surfaces of objects, as seen from the input view, are complemented by heights estimated by a virtual overhead camera (dark green) via our epipolar feature transform. Room envelope estimates are rendered in gray.

3D Scene Surface Reconstruction To provide an upper-bound on the performance of our multi-layer depth representation, we evaluate how well surfaces reconstructed from ground-truth depths cover the full 3D mesh. This allows us to characterize the benefits of adding additional layers to the representation. Table 3.1 reports the coverage (recall) of the ground-truth at a threshold of 5cm. The left panels of Figure 3.5 show a breakdown of the precision and recall for the individual layers of our model predictions along with the upper

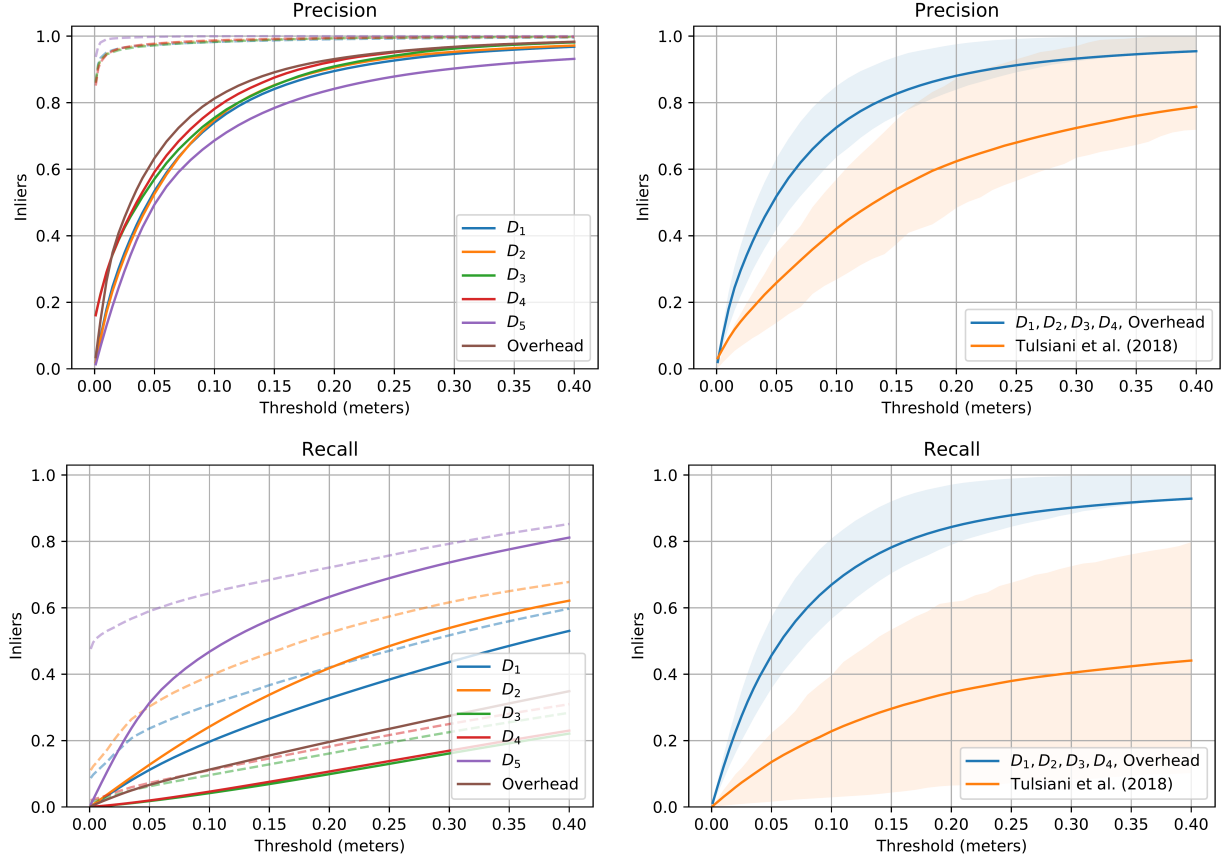


Figure 3.5: Precision and recall of scene geometry as a function of match distance threshold. *Left*: Reconstruction quality for different model layers. Dashed lines are the performance bounds provided by ground-truth depth layers ($\bar{D}_1, \bar{D}_2, \bar{D}_3, \bar{D}_4, \bar{D}_5$). *Right*: Accuracy of our model relative to the state-of-the-art, evaluated against objects only. The upper and lower band indicate 75th and 25th quantiles. The higher variance of Tulsiani *et al.* [88] may be explained in part by the sensitivity of the model to having the correct initial set of object detections and pose estimates.

	Precision	Recall
D_1	0.525	0.212
D_1 & Overhead	0.553	0.275
$D_{1,2,3,4}$	0.499	0.417
$D_{1,2,3,4}$ & Overhead	0.519	0.457

Table 3.2: Augmenting the frontal depth prediction with the predicted virtual view height map improves both precision and recall (match threshold of 5cm).

bounds achievable across a range of inlier thresholds.

Since the room envelope is a large component of many scenes, we also analyze performance for objects (excluding the envelope). Results summarized in Table 3.2 show that the addition

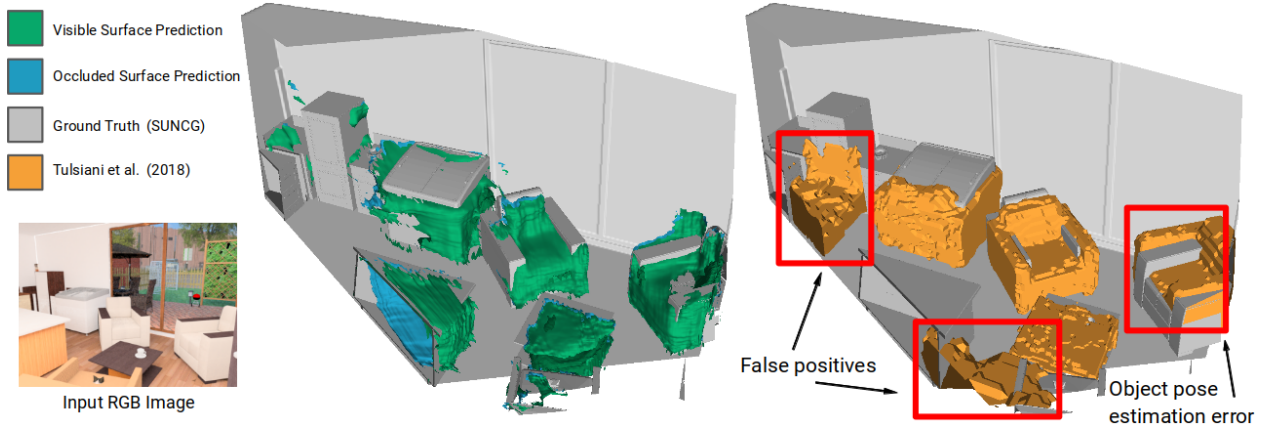


Figure 3.6: Qualitative comparison of our viewer-centered, end-to-end scene surface prediction (left) and the object-based detection and voxel shape prediction of [88] (right). Object-based reconstruction is sensitive to detection and pose estimation errors, while our method is more robust.

of multiple depth layers significantly increases recall with only a small drop in precision, and the addition of overhead EFT predictions boosts both precision and recall.

Ablation Study on Transformed Features To further demonstrate the value of the EFT module, we evaluate the accuracy of the overhead height map prediction while incrementally excluding features. We first exclude channels that correspond to the semantic segmentation network features and compare the relative pixel-level L1 error. We then exclude features from the depth prediction network, using only RGB, frustum mask and best guess depth image. This baseline corresponds to taking the prediction of the input view model as an RGB-D image and re-rendering it from the virtual camera viewpoint. The L1 error increases respectively from 0.132 to 0.141 and 0.144, which show that applying the EFT to the whole CNN feature map outperforms simple geometric transfer.

Comparison to the State-of-the-art Finally, we compare the scene reconstruction performance of our end-to-end approach with the object-based Factored3D [88] method using their pre-trained weights, and converting voxel outputs to surface meshes using marching cubes. We evaluated on 3960 examples from the SUNCG test set and compute precision and recall on objects surfaces (excluding envelope). As Figure 3.5 shows, our method yields

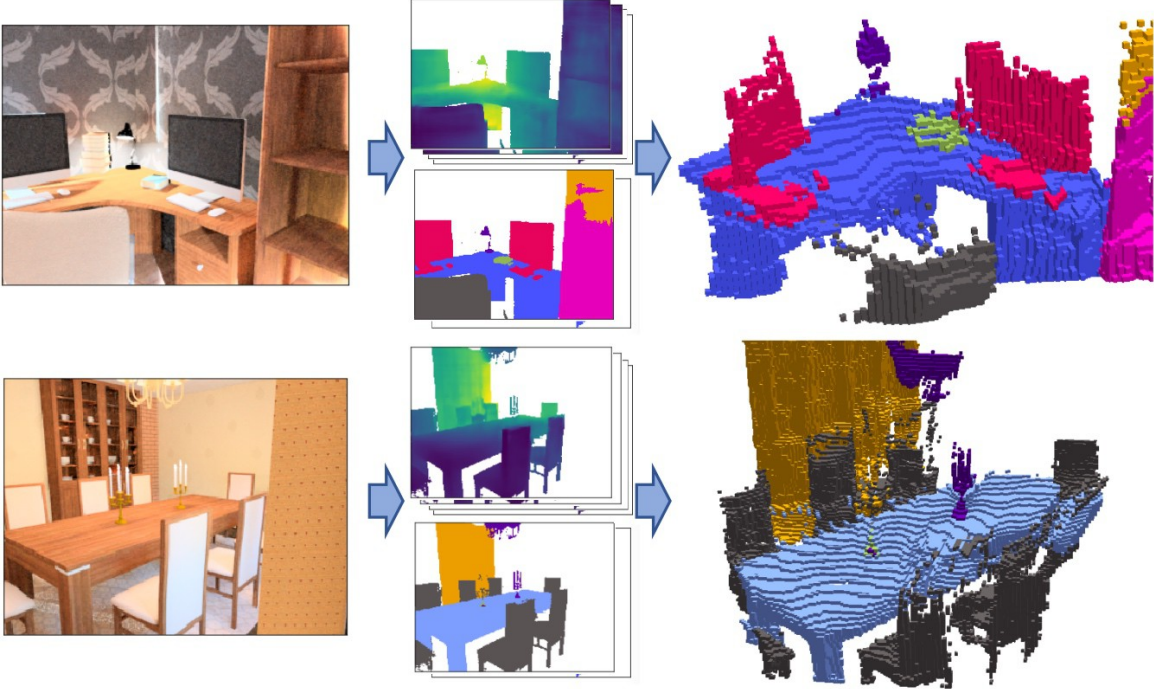


Figure 3.7: A volumetric visualization of our predicted multi-layer surfaces and semantic labels on SUNCG. We project the center of each voxel into the input camera, and the voxel is marked occupied if the depth value falls in the first object interval (D_1, D_2) or the occluded object interval (D_3, D_4).

roughly 3x improvement in recall and 2x increase in precision, providing estimates which are both more complete and more accurate. Figure 3.6 highlights some qualitative differences. To evaluate with an alternative metric, we voxelized scenes at 2.5cm^3 resolution (shown in Figure 3.7). Using the voxel intersection-over-union metric, we see significant performance improvements over Tulsiani *et al.* [87] (0.102 to 0.310) on objects (see Appendix B for details).

Reconstruction on Real-world Images Our network model is trained entirely on synthetically generated images [111]. We test the ability of the model to generalize to the NYUv2 dataset [69] via the promising comparison to Tulsiani *et al.* [88] in Figure 3.8.

We additionally test our model on images from the ScanNetv2 dataset [7]. The dataset contains RGB-D image sequences taken in indoor scenes, as well as 3D reconstructions produced by BundleFusion [8]. For each video sequence from the 100 test scenes, we randomly sample 5% of frames, and manually select 1000 RGB images to compare our algorithm to

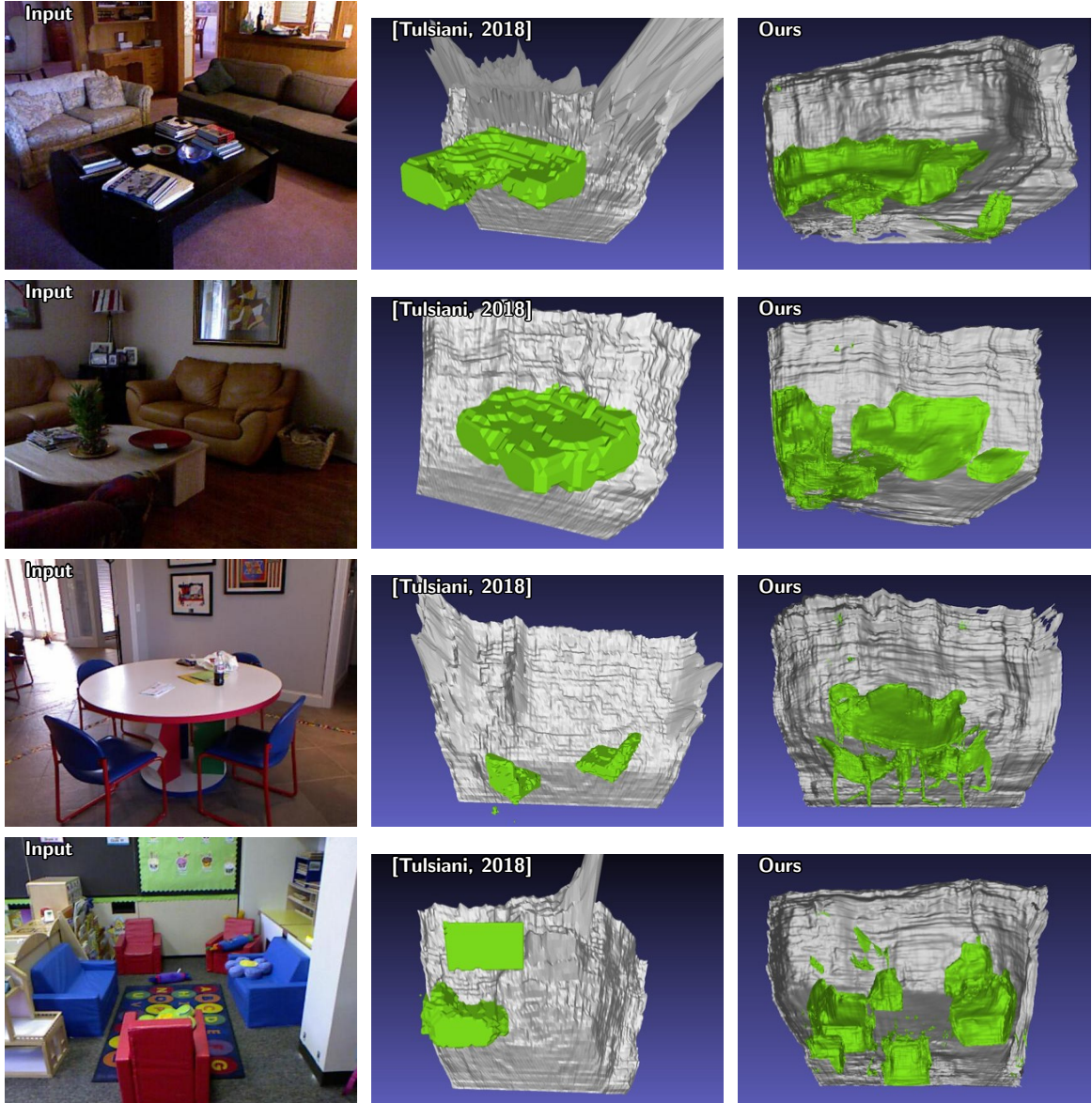
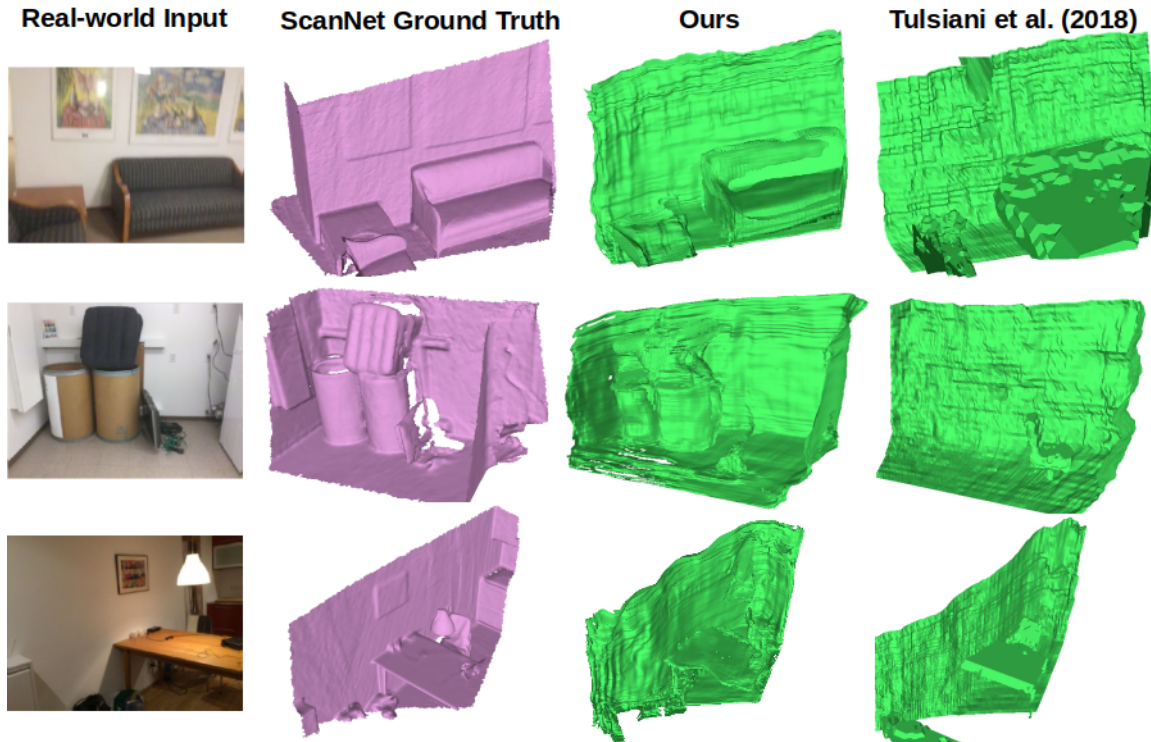


Figure 3.8: Evaluation of 3D reconstruction on the NYUv2 [69] dataset. Tulsiani *et al.* [88] are sensitive to the performance of 2D object detectors, and their voxelized output is a coarse approximation of the true 3D geometry.

Tulsiani *et al.* [88]. We select images where the pose of the camera is almost perpendicular to the gravity orientation, the amount of motion blur is small, and the image does not depict a close-up view of a single object. We treat the provided 3D reconstructions within each viewing frustum as ground truth annotations. As summarized in Table 3.3, our approach has significantly improved precision and recall to Tulsiani *et al.* [88].



	Precision	Recall
$D_{1,2,3,4,5}$ & Overhead	0.221	0.358
Tulsiani <i>et al.</i> [88]	0.132	0.191

Table 3.3: We quantitatively evaluate the synthetic-to-real transfer of 3D geometry prediction on the ScanNet [7] dataset (threshold of 10cm). We measure recovery of true object surfaces and room layouts within the viewing frustum.

3.6 Conclusion

Our novel integration of deep learning and perspective geometry enables complete 3D scene reconstruction from a single RGB image. We estimate multi-layer depth maps which model the front and back surfaces of multiple objects as seen from the input camera, as well as the room envelope. Our epipolar feature transformer network geometrically transfers input CNN features to estimate scene geometry from virtual viewpoints, providing more complete coverage of real-world environments. Experimental results on the SUNCG dataset [75] demonstrate the effectiveness of our model. We also compare with prior work that predicts voxel representations of scenes, and demonstrate the significant promise of our multi-view and multi-layer depth representations for complete 3D scene reconstruction.

The multi-layer depth representation developed here was later used as a geometric prior for human pose estimation under occlusion [95]. The next two chapters shift from static scenes to human bodies, where the observer–subject relationship takes a different form: camera viewpoint relative to a moving person.

Chapter 4

Predicting Camera Viewpoint Improves Cross-Dataset Generalization for 3D Human Pose Estimation

4.1 Introduction

3D pose estimation models trained on one dataset often generalize poorly to others, even when the underlying body poses are similar. We identify a geometric factor that accounts for much of this gap: the camera’s orientation relative to the body, a variable that standard preprocessing leaves unnormalized.

The standard preprocessing pipeline normalizes several nuisance variables: camera intrinsics

Adapted, with permission, from Z. Wang, D. Shin, and C. Fowlkes, “Predicting Camera Viewpoint Improves Cross-Dataset Generalization for 3D Human Pose Estimation,” in *European Conference on Computer Vision Workshops (ECCVW)*, 2020. © 2020 Springer.

are accounted for via known projection parameters, and subject distance is normalized by centering the root joint. But camera orientation relative to the human body is not normalized, and this variable differs systematically across datasets.

We find that viewpoint variation is a major source of cross-dataset bias. When viewpoint and skeleton scale are factored out, the pose distributions across five major datasets are nearly indistinguishable—a classifier trained to identify the source dataset from body-centered poses achieves only chance-level accuracy. The datasets differ not in what bodies do, but in how cameras see them.

To address this, we make the model aware of the viewpoint relationship. We add a viewpoint prediction branch that estimates the rotation from a body-centered coordinate frame to the camera frame, trained jointly with the pose estimation network. The viewpoint branch serves only as a training signal and is not needed at inference. By jointly learning to predict viewpoint, the backbone becomes sensitive to the observer–subject geometry. Total cross-dataset error drops by 9–20mm across five training datasets, without degrading same-dataset performance.

4.2 Related Work

3D human pose estimation. Methods for single-image 3D human pose estimation range from simple lifting networks to end-to-end model-fitting approaches. Martinez et al. [45] reported that a simple fully-connected network lifting 2D detections to 3D achieves competitive accuracy. Volumetric approaches [54] predict per-joint heatmaps in discretized 3D space, while integral regression [81] applies a soft-argmax operation to these volumes for end-to-end differentiable training. Simple Baselines [100] provides strong results with a straightforward deconvolution-based architecture. End-to-end methods such as HMR [31] regress body model

parameters directly from images, jointly estimating shape and pose. Weakly-supervised approaches [115] reduce the reliance on paired 3D annotations. These methods are typically trained and evaluated on a single dataset, and their cross-dataset performance receives less attention.

Cross-dataset generalization. Torralba and Efros [86] observed that visual recognition datasets carry systematic biases that enable classifiers to identify the source dataset from images alone. In 3D pose estimation, the major benchmarks differ substantially in capture environment and camera configuration: Human3.6M [23] uses four fixed cameras in a controlled indoor studio, SURREAL [90] renders synthetic subjects with near-uniform viewpoint sampling, 3DPW [91] uses a single moving camera in outdoor scenes, and 3DHP [48] provides 14 cameras at varying heights. Standard preprocessing normalizes camera intrinsics and subject distance by centering the root joint, but does not normalize camera orientation relative to the body. Section 4.3 shows that this residual viewpoint variation is a significant source of cross-dataset performance differences.

Coordinate frames in 3D prediction. The prediction coordinate frame is not merely a design choice: it determines generalization behavior. In Chapter 2, we find that viewer-centered and object-centered prediction lead to qualitatively different learned representations for single-view 3D shape prediction [67]. For human pose, Rhodin et al. [57] learn geometry-aware representations by training with multi-view consistency, implicitly encoding viewpoint information. Zimmermann and Brox [117] separately estimate viewpoint and pose in a canonical hand-centered coordinate frame, then combine the two to recover pose in the camera frame. Our approach differs: rather than changing the prediction coordinate frame, we add an auxiliary task that predicts the rotation between a body-centered frame and the camera frame. The prediction itself remains in camera coordinates, but the shared representation learns to capture the viewpoint relationship explicitly.

Auxiliary supervision as regularization. The viewpoint prediction branch regularizes the backbone to capture the camera–body geometric relationship. The auxiliary task does not need to produce accurate viewpoint estimates: classification with coarse cluster assignments outperforms direct quaternion regression (Section 4.5.2), suggesting that its value lies in the training signal it provides rather than in the viewpoint prediction itself.

Benchmark design and evaluation. Most work in 3D human pose estimation reports same-dataset performance. Cross-dataset evaluation is less common, and when studied, the focus is typically on domain adaptation techniques [112] rather than on diagnosing why the gap exists. Our contribution is diagnostic: we characterize the geometric factor (camera viewpoint) separately from other domain factors (appearance, lighting, body proportions) and show that correcting for viewpoint bias yields substantial cross-dataset improvements. This complements work on synthetic-to-real transfer, where the domain gap includes both appearance and geometry differences.

4.3 Analysis of Viewpoint Bias Across Datasets

Standard 3D pose estimation preprocessing normalizes camera intrinsics and root joint position, but does not normalize camera orientation relative to the body. We examine five datasets: Human3.6M [23] (H36M), Geometric Pose Affordance [95] (GPA), SURREAL [90], 3D Poses in the Wild [91] (3DPW), and MPI-INF-3DHP [48] (3DHP), and find that their camera viewpoint distributions differ systematically.

To quantify this, we define a body-centered coordinate frame with origin at the pelvis (Figure 4.1). The up vector points from the pelvis to the midpoint of the left and right shoulders. The front vector is the cross product of the pelvis-to-left-shoulder and pelvis-to-right-shoulder vectors. The right vector completes the orthonormal basis. The rotation between

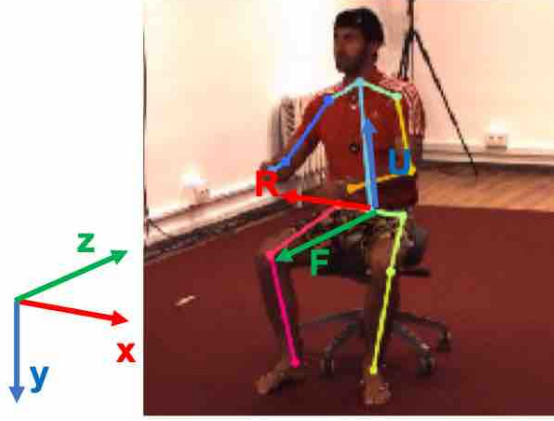


Figure 4.1: Body-centered coordinate frame used to quantify camera viewpoint relative to the body. The origin is at the pelvis, with up (U) pointing toward the midpoint of the shoulders, front (F) defined by the cross product of the pelvis-to-shoulder vectors, and right (R) completing the orthonormal basis. The camera coordinate axes (x, y, z) are shown at left.

this body-centered frame and the camera frame, represented as a unit quaternion, captures the camera viewpoint relative to the body.

The five datasets exhibit markedly different viewpoint distributions in both azimuth and elevation (Figures 4.2 and 4.3). H36M, collected in a controlled indoor setting with four fixed cameras, produces four distinct azimuth peaks as subjects rotate during capture, with elevation biased positive (cameras above subject height). SURREAL, being synthetic, has a nearly uniform viewpoint distribution in both dimensions. 3DPW, filmed in the wild with a handheld or tripod-mounted camera, shows strong bias toward frontal views (azimuth -30° to 60°) and near-zero elevation (camera roughly at subject height). GPA concentrates between -60° and 90° azimuth, while 3DHP has the most uniform distribution among real datasets, owing to its 14 cameras at varying heights, including ceiling-mounted cameras that produce high positive elevations absent from other datasets.

To test whether the datasets differ in pose or viewpoint, we trained a multilayer perceptron to classify which dataset a body-centered, size-normalized pose came from. The classifier achieves only 20% accuracy, chance level for five classes. Once viewpoint and skeleton scale

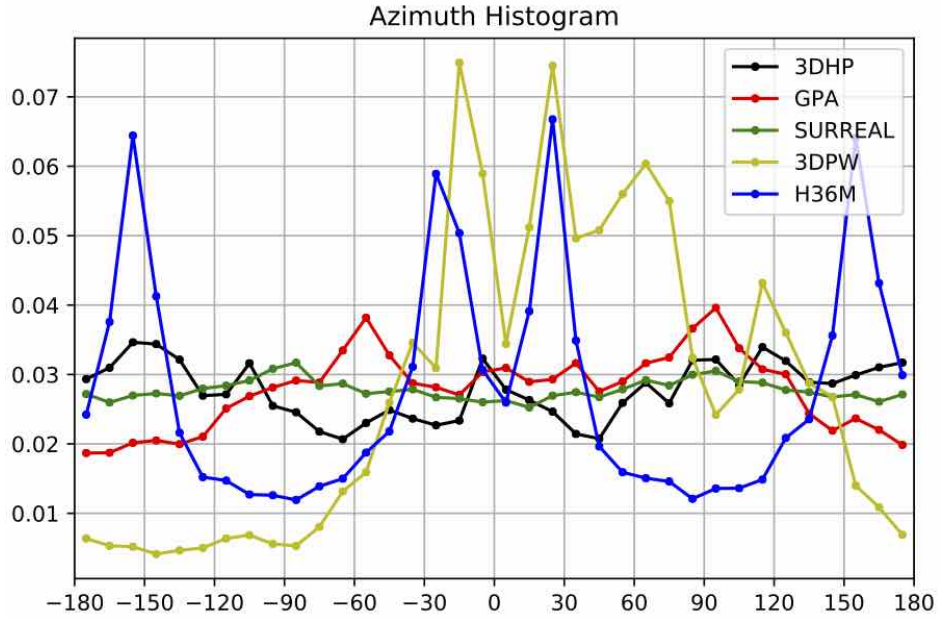


Figure 4.2: Azimuth distribution of camera viewpoints relative to the body-centered coordinate frame across five datasets. H36M shows four distinct peaks from its fixed cameras. 3DPW is heavily biased toward frontal views. SURREAL is nearly uniform (synthetic).

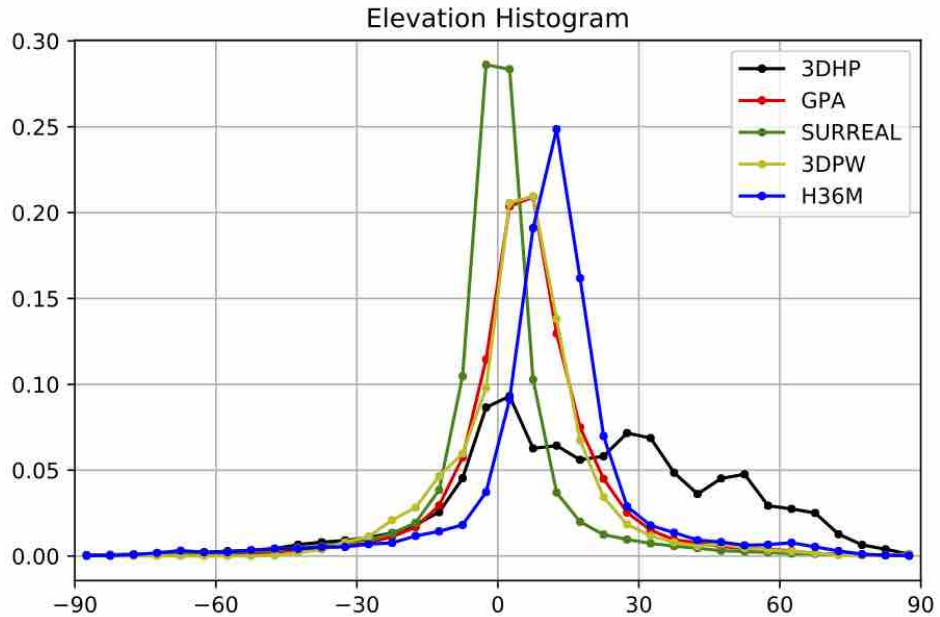


Figure 4.3: Elevation distribution across the same five datasets. H36M and GPA show positive elevation bias (cameras above subject height). 3DPW clusters near zero elevation (handheld camera at subject height). 3DHP has the widest elevation spread, including high angles from ceiling-mounted cameras.

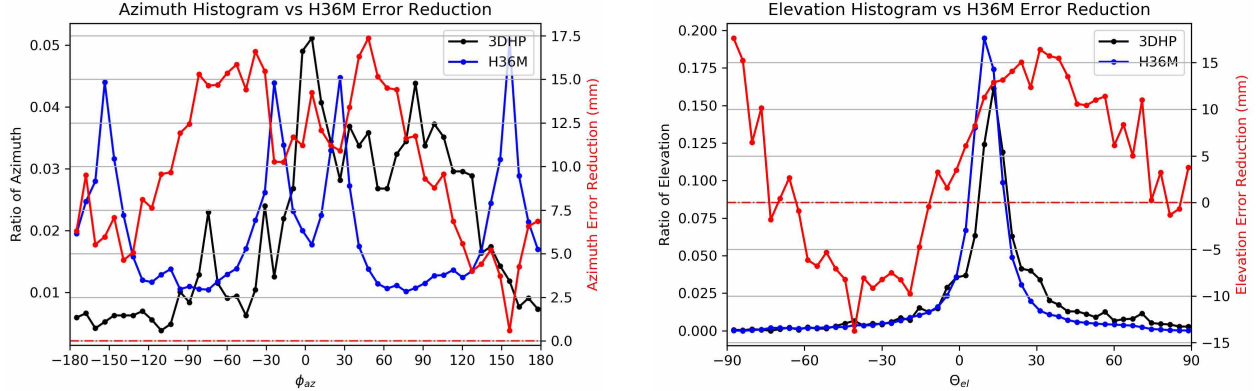


Figure 4.4: Reduction in per-joint position error from the viewpoint branch as a function of azimuth (left) and elevation (right), for a model trained on 3DHP and tested on H36M. The viewpoint distributions of the training set (3DHP, black) and test set (H36M, blue) are shown alongside the error reduction (red). Error reduction is largest in viewpoint regions where 3DHP has coverage but H36M does not.

are factored out, the pose distributions across datasets are nearly indistinguishable. Camera viewpoint, not body pose, is the primary source of cross-dataset variation.

These distribution differences have a direct effect on prediction error. Figure 4.4 shows the reduction in per-joint position error when the viewpoint branch is added, plotted as a function of azimuth and elevation for a model trained on 3DHP and tested on H36M. Error reduction concentrates in viewpoint regions where the training set (3DHP) has coverage but the test set (H36M) does not. The auxiliary viewpoint task helps most where it can leverage training-set diversity that the test set lacks.

4.4 Method

We add a viewpoint prediction branch alongside the standard pose estimation network (Figure 4.5). Both branches share a ResNet-50 [22] backbone trained on 256×256 cropped image patches. The pose branch follows a standard design: three deconvolutional layers produce 3D heatmaps, and a soft-argmax operation extracts joint coordinates in camera-relative

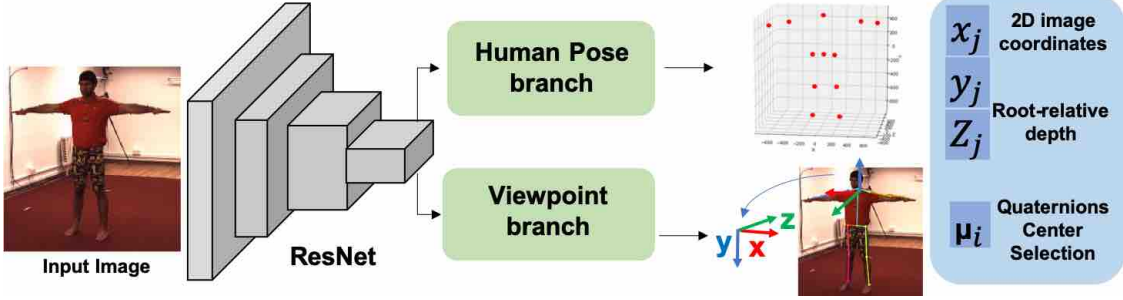


Figure 4.5: Two-branch architecture. A shared ResNet backbone feeds a pose branch (3D joint coordinates via heatmap regression) and a viewpoint branch (body-to-camera rotation classified into quaternion clusters). The viewpoint branch is discarded at test time and serves purely as a training regularizer.

coordinates.

The viewpoint branch predicts the quaternion rotation from the body-centered frame to the camera frame. We test three formulations:

- **Classification:** k -means clustering of quaternions from all training sets produces k cluster centers $\{\mu_1, \dots, \mu_k\}$. The branch outputs a probability distribution over clusters via softmax: $p(c | q) = \exp(-\mu_c^\top q) / \sum_{i=1}^k \exp(-\mu_i^\top q)$, trained with cross-entropy loss.
- **Regression:** The branch directly predicts the quaternion q and is trained with squared Euclidean loss $\|q - q^*\|^2$.
- **Combined:** The branch first classifies into a cluster, then regresses the residual rotation from that cluster center.

The total loss is $L = \lambda L_{\text{viewpoint}} + L_{\text{pose}}$, with $\lambda = 0.5$. The viewpoint prediction is *discarded at test time*; only the pose branch output is used. The prediction remains in camera coordinates; the viewpoint branch acts purely as a training-time regularizer, encouraging the shared representation to encode the geometric relationship between camera and body.

4.5 Experiments

We evaluate on all five datasets using 14 joints common across datasets, reporting MPJPE (mean per-joint position error in mm). The core experiment is a 5×5 cross-dataset evaluation: train on one dataset, test on all five.

Training details. The ResNet-50 backbone is initialized from ImageNet pretraining and fine-tuned with the Adam optimizer (initial learning rate 10^{-3} , reduced by $10\times$ at epoch 17) for 25 epochs per dataset with batch size 128. Input images are 256×256 cropped image patches. Data augmentation includes rotation, horizontal flip, color jittering, and synthetic occlusion. In addition, we mix 2D in-the-wild images from the MPII dataset during training to improve backbone robustness. To reduce redundancy in high-framerate datasets, 3DHP and SURREAL are downsampled to 5 FPS and H36M to 10 FPS. For H36M, we follow Protocol 2 (subjects 1, 3, 5, 7, 8 for training; 9, 11 for testing). Depth values are normalized from $(-z_{\max}, +z_{\max})$ to $(0, 63)$ with $z_{\max} = 2400\text{mm}$ across all datasets.

4.5.1 Cross-Dataset Evaluation

Table 4.1 shows the full 5×5 cross-dataset evaluation. Adding the viewpoint classification branch ($k=100$ global clusters) reduces total cross-dataset error by 9–20mm, with the largest reductions when training on GPA (18.6mm) and 3DHP (20.4mm). Same-dataset error drops minimally by 0.1 to 1.5mm for four of five datasets, with 3DPW showing a larger 10.9mm improvement.

	Test \ Train	H36M	GPA	SURREAL	3DPW	3DHP
Baseline	H36M	53.2	110.5	107.1	125.1	108.4
	GPA	105.2	53.9	86.8	111.7	90.5
	SURREAL	118.6	103.2	37.2	120.8	108.2
	3DPW	108.7	116.4	114.2	100.6	113.3
	3DHP	111.8	123.9	120.3	139.7	91.9
+ Viewpoint	H36M	52.0	102.5	103.3	124.2	95.6
	GPA	98.3	53.3	85.6	110.2	91.3
	SURREAL	114.0	101.2	37.1	113.8	107.2
	3DPW	109.5	112.0	112.2	89.7	105.9
	3DHP	111.9	119.7	118.2	136.0	90.3
Same-dataset reduction		1.2	0.6	0.1	10.9	1.5
Cross-dataset reduction		10.6	18.6	9.1	13.1	20.4

Table 4.1: Cross-dataset 3D pose estimation (MPJPE in mm, lower is better). Each cell shows the error when training on the column dataset and testing on the row dataset. Bold: same-dataset (diagonal). The viewpoint branch reduces total cross-dataset error by 9–20mm without degrading same-dataset performance.

4.5.2 Ablation Studies

Classification vs. regression. Classification reduces MPJPE to 89.7mm compared to 94.0mm for regression and 93.2mm for the combined approach. Classification provides stronger regularization despite being less accurate at predicting the exact quaternion.

Cluster configuration. Global clusters (computed from all five datasets) outperform local clusters (computed per-dataset) by 3.4mm (89.7 vs. 93.1mm). Global clustering avoids bias toward any single dataset’s viewpoint distribution. The number of clusters $k = 100$ is optimal in a sweep over $\{10, 24, 50, 100, 200, 500\}$, and results are robust to k -means initialization (range of ± 0.4 mm across four random seeds).

Canonical pose prediction. Adding a third branch that predicts body pose in the body-centered coordinate frame (canonical pose) provides no improvement (100.3mm vs. 100.6mm baseline). Camera-coordinate prediction can directly exploit the correspondence between 2D

keypoint locations and 3D joint positions, so the canonical branch adds little useful signal.

4.5.3 Generalization to Other Architectures

The viewpoint prediction branch is not specific to a particular pose estimation architecture. Applying the same approach to the architecture of Zhou et al. [115] produces consistent improvements: cross-dataset error reductions of 1.9–12.3mm, with the same pattern of large cross-dataset gains and small same-dataset changes. This suggests the benefit comes from the auxiliary task rather than from interactions with the specific architecture.

4.6 Discussion

The viewpoint branch helps most when formulated as classification rather than regression (89.7mm vs. 94.0mm on 3DPW). The benefit comes not from encoding viewpoint precisely, but from requiring features to distinguish broad geometric configurations—coarse cluster assignments suffice.

The per-dataset results suggest why the improvement varies across training sets. Training on 3DHP produces the largest total cross-dataset reduction (20.4mm), consistent with 3DHP having the most uniform viewpoint distribution among real datasets. Its 14 cameras at varying heights provide the broadest coverage of the azimuth-elevation space. Conversely, training on SURREAL produces the smallest total cross-dataset reduction (9.1mm). SURREAL’s viewpoint sampling is already uniform, so the baseline has little viewpoint bias for the auxiliary task to correct.

Limitations. The viewpoint distributions studied here are static properties of each dataset’s capture setup; in deployment, viewpoint may vary continuously as cameras move. The

method does not address other sources of dataset bias (background appearance, lighting, body proportions), which contribute to the residual cross-dataset gap that remains after the viewpoint branch is added. The 9–20mm total improvement is substantial relative to the total cross-dataset gap but far from closing it entirely.

Connections. In Chapter 2, we showed that the choice between viewer-centered and object-centered coordinate frames influences what a model learns about 3D shape. Here, camera viewpoint plays a related role: it is a geometric variable that standard preprocessing leaves unnormalized, and modeling it explicitly encourages representations that transfer better across datasets.

Chapter 5

Egocentric Multi-Person Human Motion Estimation

5.1 Introduction

Viewer-centered and scene-centered representations capture different aspects of 3D structure, and combining them can recover information that neither provides alone. For static objects and scenes, this combination can be computed from a single image, for instance by combining viewer-centered depth with scene-centered shape. For dynamic scenes with moving people, the problem is harder: observations are partial, intermittent, and distributed across time and viewpoints. What a given camera can see changes from moment to moment.

Wearable camera platforms such as head-mounted displays and smart glasses are making egocentric video increasingly common in multi-person settings. When multiple people in a shared environment each wear a camera, a natural structure emerges: each person’s first-person view provides third-person observations of others, while their own body falls almost entirely outside their field of view. The same person is simultaneously an unobserved sub-

ject from their own perspective and an observed target from others’. This ego–exo duality (Figure 5.1) creates complementary signals for body estimation that few existing methods exploit jointly.

Existing methods have explored full-body pose estimation from egocentric or exocentric observations. A head-motion prior (such as EgoAllo [106]) estimates the wearer’s body pose from their camera trajectory alone, a viewer-centered signal, using the statistical relationship between head motion and body movement learned from motion capture data. This provides an estimate at every timestep with good absolute positioning (the body is attached to the known camera location) but limited accuracy in relative body pose, since head motion constrains limb positions only loosely. Alternatively, a third-person tracker (such as CoMotion [50]) can detect and estimate 3D body pose for people visible in each camera’s video stream, a scene-centered signal. When a person is visible, this provides accurate relative body pose, but coverage is incomplete (people move in and out of each other’s field of view) and absolute positioning is limited by the observer’s viewpoint.

Neither source alone is sufficient. The head-motion prior has full temporal coverage but limited pose accuracy. Visual observations of the body have good pose accuracy but sparse temporal coverage (approximately 24% of person-frames have no observation from any camera in our evaluation setting). Combining these egocentric and exocentric estimates, retaining the coverage of one and the accuracy of the other, remains an open problem.

We develop a conditional diffusion model that integrates both signals through imputation-based conditioning. At each step of the iterative denoising process, observed dimensions of the body state are replaced with values from the visual tracker, while the diffusion model fills in unobserved dimensions guided by the head-motion prior. We find that noisy observations from the visual tracker help during the early, coarse stages of denoising but introduce harmful noise during fine-grained refinement. Stopping observation replacement at low noise levels improves both positional accuracy and temporal smoothness.

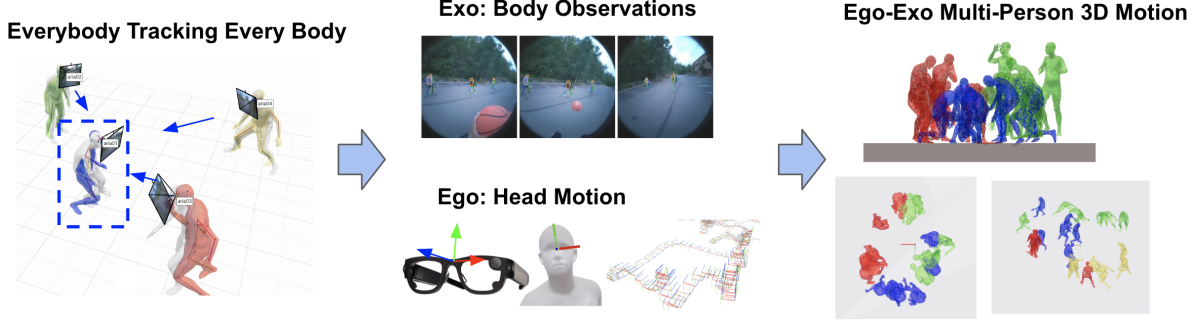


Figure 5.1: Overview. In a multi-person egocentric setting, each person’s head-mounted camera provides third-person observations of others, while their own body motion is inferred from head trajectory alone. Can we fuse these complementary signals to enable global scene understanding and recover full-body 3D motion for all participants?

The model trains entirely on single-person motion capture data (AMASS [44]) with randomly masked ground-truth observations simulating varying levels of coverage. No multi-person training data is required. On the EgoHumans benchmark, mean per-joint position error drops from 136mm (motion prior alone) to 102mm with our full approach.

5.2 Related Work

Egocentric body estimation. Estimating the body pose of a camera wearer from egocentric video is challenging because the wearer’s body is largely outside the field of view. Early work inferred the camera wearer’s pose from interaction cues, such as the visible body of a social partner [51]. EgoBody [108] introduced a benchmark for egocentric body estimation during social interactions. EgoPoser [30] estimates full-body pose in real time from sparse head and hand tracking signals on VR/AR devices. EgoAllo [106] takes this further by conditioning a diffusion model on the head trajectory alone, learning a motion prior from large-scale motion capture data. Our work uses a pretrained EgoAllo model as the egocentric motion prior and introduces imputation-based conditioning to incorporate exocentric observations from other cameras.

Multi-person 3D pose estimation. Recovering 3D pose for multiple people from monocular video is a long-standing problem. Most methods operate in a third-person setting with a static camera [31, 37]. When cameras are moving, methods like GLAMR [107] and SLAHMR [105] jointly optimize for camera motion and human body pose to maintain global trajectories. CoMotion [50] provides concurrent multi-person 3D tracking from monocular video, producing per-frame SMPL estimates with confidence scores. In our setting, CoMotion is applied to each person’s egocentric video stream to obtain third-person observations of other people. The key difference from standard multi-person estimation is that our cameras are worn by the subjects being tracked, creating the ego–exo duality that motivates our fusion approach.

Diffusion models for human motion. Diffusion models have become the dominant approach for human motion generation and prediction. MDM [85] applies denoising diffusion to human motion sequences conditioned on text or action labels. PriorMDM [66] uses a pretrained motion diffusion model as a prior for downstream tasks including long-sequence generation and multi-person motion synthesis. HumanMAC [3] frames motion prediction as masked completion, conditioning a diffusion model on observed past frames to generate future frames. OmniControl [101] enables spatial control of any joint at any time through learned guidance. Our work differs in that the conditioning signal (body pose observations from a visual tracker) is noisy and intermittent rather than clean and user-specified. This noise-in-conditioning setting requires the noise-aware conditioning schedule described in Section 5.5.2.

Conditional diffusion and imputation. Imputation-based conditioning replaces observed dimensions of the noisy state during denoising, an approach used for image inpainting by RePaint [42] and adapted to motion by CondMDI [6]. CondMDI replaces observed joint trajectories with clean observation values at each denoising step and provides a binary mask to the model, enabling flexible conditioning for motion in-betweening. We follow

CondMDI’s per-step replacement paradigm, adapted to SMPL body model parameter space. Saadatnejad et al. [64] apply diffusion models to pose forecasting from noisy observations. RoHM [109] uses diffusion models to reconstruct human motion from noisy and occluded monocular RGB(-D) video. Both address the noisy-observation setting that we also face; our work differs in the multi-person egocentric setting and the imputation-based fusion of egocentric and exocentric signals.

5.3 Problem Formulation

We consider N people moving in a shared environment, each wearing a head-mounted camera (Meta Aria glasses in our experiments, with $N = 3$ or 4). All camera feeds and pose estimates are synchronized and expressed in a common world coordinate frame (z -up, floor at $z = 0$). For each person i at time t , we have access to:

- An RGB video stream I_t^i from their egocentric camera,
- A camera pose estimate C_t^i from visual-inertial odometry (VIO/SLAM).

The goal is to estimate the body state $S_t^i = (\theta_t^i, \beta_t^i)$ and root-joint pose $L_t^i = (R_t^i, t_t^i)$ for every person at every timestep. We represent the body using the SMPL-H model [41, 62], whose state is a 496-dimensional vector consisting of 16 body shape parameters (β), 189 body rotation parameters (21 non-root joints $\times$ 9 entries of flattened 3×3 rotation matrices), 21 joint-floor contact flags, and 270 hand rotation parameters (15 joints per hand $\times$ 9).

We relate the egocentric camera pose to the body root-joint pose through a deterministic mapping $C_t^i = \Phi_{S_t^i}(L_t^i)$ and its inverse $L_t^i = \Phi_{S_t^i}^{-1}(C_t^i)$. This mapping is defined via the Central Pupil Frame (CPF), the midpoint between the pupil center vertices of the posed SMPL mesh, with the same rotation as the SMPL head joint. The CPF serves as the kinematic

root of the body: instead of the standard SMPL pelvis root, the body “hangs” from this head-anchored frame. During training on AMASS (which has no wearable devices), the CPF is computed from the ground-truth body mesh following EgoAllo’s derivation [106]. At inference, the Aria device tracks the head trajectory using visual-inertial sensors, which does not correspond exactly to this virtual, mesh-derived CPF, due to variation in head shape and device placement. This difference is captured by a per-person rigid-transform prior estimated from calibration data. The prior assumes left-right symmetry and is held fixed across each sequence. Obtaining ground-truth body parameters in multi-person ego-centric settings is itself difficult. While large-scale datasets such as Ego-Exo4D [19] capture simultaneous first- and third-person perspectives, they focus on a single camera wearer and lack the dense SMPL parameters required for our evaluation. EgoHumans [35] provides the closest available benchmark, with body parameters triangulated from a dense camera array and refined through post-hoc optimization rather than direct motion capture, which can introduce inherent person- or scene-specific biases.

We model each person’s body state independently, conditioned on the observations from all cameras:

$$P(S, L \mid I, C) = \prod_i P(S^i, L^i \mid I^1, \dots, I^N, C^1, \dots, C^N).$$

This independence assumption simplifies the problem but ignores inter-person interactions. In practice, the primary coupling between people comes through their mutual visibility: person j ’s camera may observe person i , providing an exocentric observation $(\hat{S}_t^{ij}, \hat{L}_t^{ij})$ when i is visible in camera j at time t .

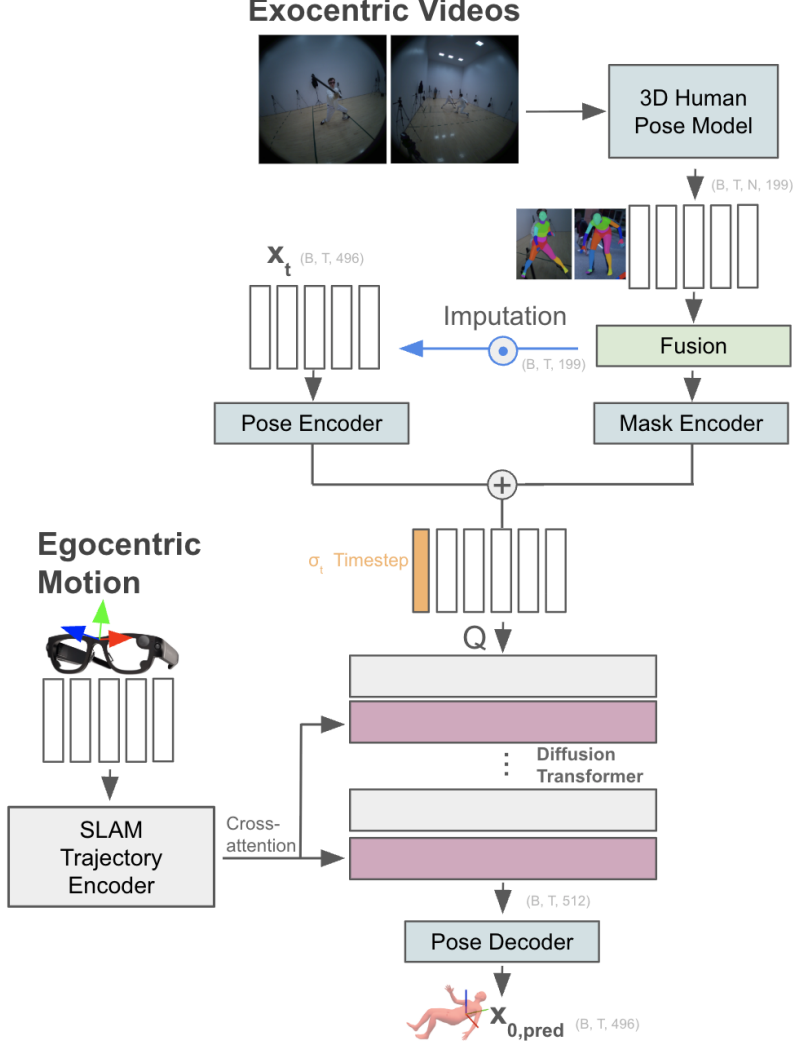


Figure 5.2: Architecture. Video streams from other participants’ cameras are processed by a 3D pose model (CoMotion) to produce per-frame body parameter estimates. At each denoising step, these estimates are replaced into the noisy state x_t via imputation, with an observation mask encoder signaling which dimensions have observations. The diffusion transformer (based on EgoAllo [106]), conditioned on the egocentric SLAM trajectory via cross-attention, denoises all dimensions jointly.

5.4 Egocentric and Exocentric Estimation

Our approach builds on two existing estimation systems, one egocentric and one exocentric, that provide complementary signals for body pose. Combining their outputs recovers information that neither captures alone; Section 5.5 describes how we integrate them.

5.4.1 EgoAllo: Head-Motion-Conditioned Body Estimation

EgoAllo [106] is a diffusion model that estimates full-body pose from a single person’s head trajectory. Given the CPF trajectory C^i as conditioning, it samples from a learned distribution $p_\theta(S^i | C^i)$ over body states. The root-joint pose L^i is then recovered deterministically via $\Phi_{S^i}^{-1}(C^i)$.

The model architecture consists of 7 transformer layers with CPF conditioning applied via cross-attention at every layer. It directly predicts the clean state x_0 rather than the noise, and was pretrained on AMASS, a large archive of single-person motion capture data, for 3 million steps. This pretraining provides a learned prior over the relationship between head motion and body pose.

EgoAllo provides complete temporal coverage (every timestep has an estimate) and calibrated global positioning, since the body is directly attached to the camera trajectory. Its weakness is relative body pose accuracy: head motion alone constrains the body only loosely, particularly for the limbs. Two people with identical head trajectories but different arm positions would produce identical estimates.

5.4.2 CoMotion: Third-Person Visual Tracking

CoMotion [50] is a 3D human detection and tracking system that we apply to each observer’s egocentric video stream. CoMotion was trained on third-person video datasets and applied directly to EgoHumans without fine-tuning, which partly explains the noise level of its predictions in our setting. For each detected person in camera j ’s view, it outputs SMPL body parameters $\hat{S}_t^{ij}$ with two associated scores: a detection confidence (sigmoid-activated output from the detection head, reflecting detection difficulty) and a tracking match score (geometric consistency between the tracked 3D pose and the corresponding detection, computed via

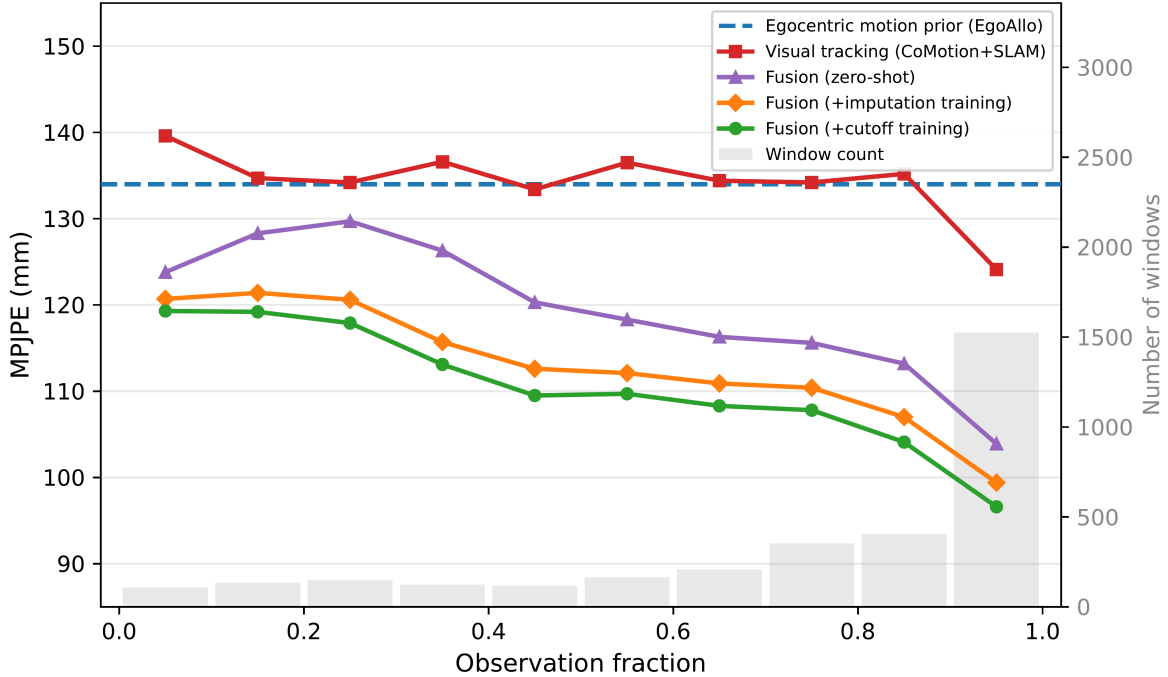


Figure 5.3: Mean per-joint position error as a function of observation fraction, the fraction of frames within each 128-frame window where at least one camera detects the subject. Three baselines are shown: the egocentric motion prior (EgoAllo, dashed line at approximately 136mm), visual tracking with device alignment (CoMotion+SLAM), and zero-shot fusion (pretrained EgoAllo with observation replacement but no imputation training). Two trained fusion variants use the same replacement cutoff ($t=50$) at inference: +imputation training (100k steps with clean observations) and +cutoff training (10k additional steps with the cutoff schedule applied during training). The prior is approximately flat, confirming that observation fraction does not correlate with intrinsic scene difficulty. Both trained fusion variants improve over all baselines at every coverage level, and each training stage produces further gains across all bins. Gray bars show the number of evaluation windows per bin.

a modified Object Keypoint Similarity metric). These estimates are in camera coordinates; we transform them to the world frame using the known camera pose C_t^j .

Data association (determining which detection in camera j corresponds to person i) projects person i ’s camera location into camera j ’s image plane and matches the nearest detection, subject to thresholds on 2D distance, depth consistency, angular deviation, detection confidence, and tracking score. When multiple cameras observe the same person, we select the single observation with the highest confidence score (“best-score” fusion).

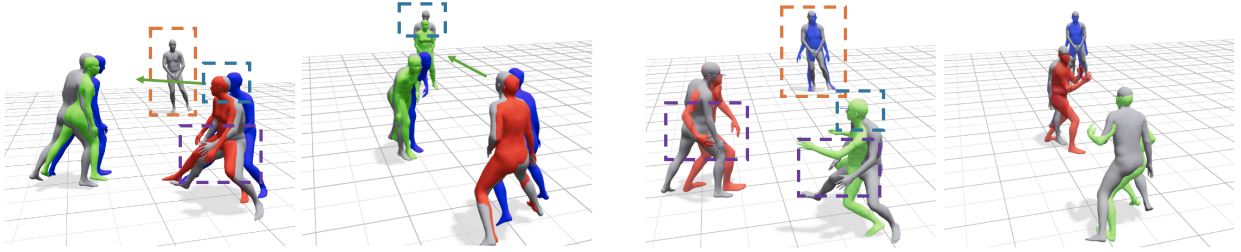


Figure 5.4: Comparison of egocentric and exocentric estimates. **Left:** The egocentric model (EgoAllo) provides full temporal coverage with calibrated global positioning, but limited body pose accuracy since head motion alone constrains limb positions loosely. **Right:** The exocentric model (CoMotion) provides more accurate relative body pose when people are visible, but has incomplete temporal coverage and unreliable absolute positioning.

When a person is visible, CoMotion provides accurate relative body pose that is substantially better than what head motion alone can achieve (Figure 5.4). However, it has three limitations. First, coverage is incomplete: temporal coverage varies from 26% (tennis, where players are far apart) to 89% (fencing, with close face-to-face interaction) across activity categories in our evaluation dataset. Second, absolute positioning is unreliable due to monocular scale ambiguity in the per-frame 3D estimates. Third, the predicted body rotations are noisy, with a mean geodesic error of approximately 17° per joint relative to ground truth, and the spine2 joint alone contributes disproportionately to this error.

5.5 Method: Observation-Conditioned Diffusion

The key challenge is fusing the egocentric motion prior with noisy, intermittent exocentric observations from CoMotion. We use imputation-based conditioning: at each step of the iterative denoising process, observed dimensions of the noisy state are replaced with values from the visual tracker. This approach requires no changes to the base model architecture and no multi-person training data (Figure 5.2).

5.5.1 Imputation-Based Conditioning

We condition the diffusion model on external observations by replacing observed dimensions of the noisy state x_t at each DDIM denoising step, following the per-step replacement paradigm of CondMDI [6]. Concretely, at step t :

1. Replace the observed dimensions of x_t with the corresponding observation values:
$$x_t[\text{obs}] \leftarrow x_0^{\text{obs}}[\text{obs}].$$
2. Encode the modified x_t and pass it to the model along with an observation mask.
3. The model denoises all dimensions jointly, producing a prediction $\hat{x}_0$.
4. Compute x_{t-1} via the standard DDIM update.

The observation values come from CoMotion’s SMPL parameter estimates. Because SMPL local joint rotations (joints 1–21) are frame-invariant and identical regardless of the world or camera coordinate system, CoMotion’s predictions can be used directly as replacement values without coordinate transformation. For frames with a valid observation, we replace the betas (16 dims) and body rotations (189 dims) provided by CoMotion, imputing 205 of the 496 dimensions. Replacement is all-or-nothing at the frame level: frames without a valid observation receive no replacement and are generated entirely by the diffusion prior.

To signal which dimensions have observations at each timestep, we add an observation mask encoder to the model: a 3-layer MLP ($496 \rightarrow 512 \rightarrow 512 \rightarrow 512$) that takes a binary mask as input and produces a 512-dimensional vector. This vector is summed into the encoded input alongside the existing modality-specific encoders (for betas, body rotations, contacts, and hand rotations). The observation mask encoder is necessary for the model to distinguish frames with observations from frames without; removing it degrades performance.

5.5.2 Noise-Aware Conditioning Schedule

The imputation approach described above assumes that observations are reliable. When they are (as with synthetic ground-truth observations on AMASS), replacement at every denoising step works well. But CoMotion’s predictions are noisy ($\sim 17^\circ$ mean geodesic error per joint), and we find that their effect on the denoising process depends on the noise level t .

At high t (early denoising steps), the diffusion noise in x_t is large. The observation noise is small by comparison, so replacing observed dimensions with CoMotion values provides useful structural guidance, pushing the trajectory toward the right neighborhood of body poses. At low t (late denoising steps), the diffusion noise is small and the model performs fine-grained refinement. Here, the observation noise dominates: replacing dimensions with noisy values corrupts the structure that earlier steps established.

More generally, in any imputation-based conditioning scheme where the conditioning signal is noisy, there exists a crossover point in the denoising schedule where the diffusion noise drops below the observation noise. Above this point, replacement provides net information; below it, replacement adds net noise. This relates to the principle underlying SDEdit [49], where the noise level controls the tradeoff between fidelity to the input and realism of the output; in our setting, the analogous tradeoff is between fidelity to noisy observations and coherence of the learned motion prior. The practical implication is simple: stop replacing observed dimensions below the crossover. We set this replacement cutoff at $t = 50$. Due to the quadratic DDIM timestep schedule (30 steps from $t = 1000$ to $t = 0$), this means observations are used for the first ~ 23 steps ($t \geq 64$) and the model denoises freely for the final ~ 7 steps ($t \leq 49$).

In our experiments, adding the replacement cutoff improves both positional accuracy and temporal smoothness (Section 5.6.3, Table 5.1). Without the cutoff, replacing observa-

tions at every denoising step increases velocity error from 487 to 595 mm/s, worse than the observation-free prior, because frame-independent CoMotion predictions create temporal discontinuities. With cutoff-schedule training and the replacement cutoff, velocity error drops to 413 mm/s, well below the prior. We found no benefit from graduated or smooth ramps over a hard cutoff. The cutoff hurts performance when observations are clean (noise-free), suggesting that it specifically addresses observation noise rather than providing a general improvement.

5.5.3 Per-Modality Handling

Different components of the SMPL state have different noise characteristics in CoMotion’s predictions. Body shape parameters (β , 16 dimensions) are relatively reliable, since shape is a global property that CoMotion estimates stably. Body joint rotations (189 dimensions) are noisier, with high variance from frame to frame.

We exploit this by applying different cutoff thresholds per modality. Body shape parameters are replaced at all timesteps (cutoff $t = 0$), while body rotations follow the standard cutoff at $t = 50$. Additionally, since body shape is constant for a given person, we stabilize the shape estimate by replacing per-frame CoMotion betas with their per-subject mean across all observed frames. Per-frame variation in shape estimates is noise; averaging removes it.

These refinements reduce MPJPE by a further 1.6mm (1.3mm from per-modality cutoff, 0.3mm from shape stabilization).

5.5.4 Training with Cutoff Schedule

The techniques described above are applied only at inference time. The model itself was trained with observations present at all noise levels, creating a mismatch: at low t , the

model expects observations that inference withholds. We address this by fine-tuning with the cutoff schedule applied during training.

The full training proceeds in three stages:

1. **Pretraining** (EgoAllo): The base diffusion model is trained on AMASS for 3 million steps, with CPF head trajectory conditioning only.
2. **Imputation fine-tuning**: The observation mask encoder is added and the model is fine-tuned for 100,000 steps with clean observation replacement. For each training example, an observation fraction is sampled uniformly in $[0, 1]$, and that fraction of frames is randomly selected for replacement with the ground-truth SMPL parameters from AMASS. This simulates varying levels of observation coverage without requiring multi-person data or explicit camera simulation. A 10% observation dropout rate (zeroing the entire mask for a training sample) ensures the model can still generate motion with no observations.
3. **Cutoff-schedule fine-tuning**: Starting from the imputation-trained checkpoint, the model is further fine-tuned for 10,000 steps with the replacement cutoff applied during training. When the sampled noise level t falls below the cutoff threshold, the observation mask is zeroed out, matching the inference-time behavior. Learning rate is 10^{-4} with 1,000 warmup steps.

In our evaluation, cutoff-schedule training reduces MPJPE by an additional 2.9mm (from 107.6mm to 104.7mm) on top of the 5.4mm gain from imputation training. Verification on AMASS with clean observations shows that this training does not degrade performance: the cutoff-trained model is 1.0–1.5mm *better* than the imputation-only model across all observation fractions.

5.5.5 Post-Inference Refinement

The diffusion model produces plausible trajectories but does not explicitly enforce physical constraints such as foot-floor contact or temporal smoothness. We apply a post-inference optimization step adapted from EgoAllo [106]: a Levenberg-Marquardt optimizer that refines joint rotations against a weighted sum of physical objectives. These include foot skating prevention (using the model’s predicted contact labels to pin feet during ground contact), first- and second-order angular smoothness of joint rotations, smoothness of changes relative to the initial diffusion output, and a regularization term that keeps optimized poses close to the diffusion prediction. The optimizer runs 20 iterations per trajectory, operating on joint rotation parameters while holding body shape and root translation fixed. Each person’s trajectory is refined independently.

We considered applying optimization at different points in the pipeline. Running the optimizer between DDIM steps (“inner guidance”) would allow enforcing physical constraints throughout denoising. We found, however, that this degrades MPJPE by approximately 8mm, suggesting that external modifications to x_t during denoising disrupt the learned trajectory, consistent with the noise-aware cutoff finding that late-stage modifications are harmful. Applying optimization after the diffusion process completes avoids this conflict.

5.6 Experiments

5.6.1 Experimental Setup

Dataset. We evaluate on EgoHumans [35], a multi-person egocentric benchmark with 132 sequences across 7 activity categories: badminton, basketball, fencing, lego assembly, tagging, tennis, and volleyball. Each sequence involves 3–4 participants wearing Meta Aria

glasses. Ground truth SMPL body parameters are obtained from multi-view triangulation and optimization using a dense external camera array. The original 20 FPS data is resampled to 30 FPS to match AMASS and EgoAllo’s training frame rate.

Coordinate systems. All estimates and evaluations use a shared world coordinate frame (z -up, floor at $z = 0$). The rigid transform between the device trajectory and the training CPF is estimated per person per sequence from calibration data. Despite this calibration, residual mismatch between the two contributes approximately 90mm of irreducible global MPJPE across all evaluation windows.

Inference. DDIM sampling uses 30 steps with a quadratic timestep schedule, deterministic sampling ($\eta = 0$), and the per-modality cutoff and shape stabilization described above. For sequences longer than the 128-frame model window, we use a sliding window with 32-frame overlap, blending predictions from adjacent windows linearly in SMPL parameter space at each DDIM step.

Metrics. We report global accuracy metrics (MPJPE: mean per-joint position error), relative pose metrics after Procrustes alignment (PA-MPJPE), and temporal smoothness metrics (MPJVE: mean per-joint velocity error; Accel: acceleration error). MPJPE captures both body pose and absolute positioning; PA-MPJPE isolates body pose accuracy by removing global translation, rotation, and scale.

5.6.2 Evaluated Methods

We evaluate four configurations that differ in what signals they use and how they combine egocentric and exocentric information.

Egocentric motion prior. Each subject’s device provides a continuous head trajectory via SLAM/VIO, from which full-body pose can be estimated. We use EgoAllo [106] (Section 5.4.1), a diffusion model pretrained on motion capture data that operates on local body parameters (joint rotations and shape). The device’s SLAM trajectory serves as the kinematic root, giving this estimate full temporal coverage with calibrated global positioning. Despite using no visual information, head trajectory alone produces surprisingly plausible full-body poses [106, 39] because head and body motion are strongly biomechanically coupled. Full-body pose accuracy is nonetheless limited because head motion leaves limb positions underdetermined, with many plausible body configurations compatible with a single trajectory.

Visual tracking. Body pose can also be estimated through exocentric perception when the subject is visible in another person’s camera, complementing the egocentric motion prior. In our setting, each person’s egocentric camera provides third-person views of other participants. We apply CoMotion [50] (Section 5.4.2), a multi-person 3D pose estimator, to these video streams. Estimates are produced in the observer’s camera coordinates and transformed to the world frame using the observer’s known camera pose. Because these estimates come from direct observation of the body, the relative joint rotations are better grounded than what can be inferred from head motion alone. However, an estimate is available only when the subject appears in at least one observer’s field of view, creating temporal gaps that vary with activity and scene geometry. Frames without any valid observation are filled by interpolating body rotations between the nearest observed frames. Even when visible, estimate quality depends on the observation viewpoint. Without device-based alignment, the resulting global MPJPE is 598mm. The relative body pose is accurate (PA-MPJPE of 51mm, identical to the aligned variant), but global positioning is catastrophic, illustrating why the alignment described in the next paragraph is necessary.

Visual tracking with SLAM alignment. Even after transforming to the world frame, the tracked body’s global position depends on the observer’s estimate of where the subject is. Because each subject in our setting wears their own device with SLAM/VIO, their device trajectory provides a more direct source of global position. As in the egocentric motion prior, this trajectory serves as the kinematic root, but here the body pose comes from exocentric visual tracking rather than the head-motion prior. We anchor the tracked body to the device trajectory (via the per-person prior in Section 5.3), analogous to the root-aligned evaluation standard in the pose estimation literature but using the wearable device rather than the pelvis. The result combines visually grounded body pose with device-derived global positioning, without learned motion priors. This is straightforward when multiple participants wear devices; most prior egocentric work involves a single wearer.

Observation-conditioned diffusion. Our full approach (Section 5.5) integrates both signals through conditional diffusion. At each denoising step, body parameters from the visual tracker are replaced into the noisy state via imputation. The diffusion model generates estimates for all timesteps, filling temporal gaps with learned motion priors rather than interpolation, and refining the observed poses themselves. A noise-aware schedule stops replacement at low noise levels, where tracker noise would corrupt the fine structure that earlier denoising steps established.

5.6.3 Main Results

Table 5.1 compares the evaluated methods. The two standalone baselines each capture a different aspect of body estimation: the egocentric motion prior provides full temporal coverage with calibrated global positioning (135.5mm MPJPE) but poor relative body pose (92.1mm PA-MPJPE), while visual tracking with device alignment achieves much better relative pose (50.8mm PA-MPJPE) but comparable global error (131.6mm) and catastrophic

Configuration	MPJPE↓	PA-MPJPE↓	MPJVE↓	Accel↓
Egocentric motion prior (no observations)	135.5	92.1	487	8.1
Visual tracking + device alignment	131.6	50.8	497	7.8
Observation-conditioned diffusion (pretrained EgoAllo)	113.0	57.2	595	15.5
+ Imputation training	107.6	57.8	431	7.9
+ Cutoff-schedule training	104.7	57.9	413	7.5
+ Post-inference refinement	102.3	58.6	384	6.7

Table 5.1: Results on EgoHumans (132 sequences, 7 activity categories). The first two rows are standalone baselines using a single signal each. The remaining rows show progressive improvement of the observation-conditioned diffusion approach: adding imputation-aware training, noise-aware conditioning, and post-inference optimization. MPJPE: mean per-joint position error in the world frame (mm). PA-MPJPE: after Procrustes alignment (mm). MPJVE: mean per-joint velocity error (mm/s). Accel: acceleration error (m/s²). All rows without post-inference refinement except the last.

Category	n	Motion prior	Tracking + align	Ours
badminton	61	135.7	133.3	104.6
basketball	13	141.4	127.1	108.9
fencing	14	130.8	115.5	81.9
legoassemble	6	134.1	147.7	108.7
tagging	14	123.2	126.8	97.9
tennis	13	130.8	132.8	119.0
volleyball	11	155.9	143.4	119.5
ALL	132	135.5	131.6	104.7

Table 5.2: Per-category MPJPE (mm) on EgoHumans. The egocentric motion prior provides a baseline unaffected by observation coverage. Visual tracking + device alignment depends on mutual visibility: activities with close interaction (fencing, tagging) have better coverage and lower error, while activities with more distance (volleyball, tennis) or object focus (lego assembly) have worse coverage. The observation-conditioned diffusion approach improves on both baselines across all categories.

leg positioning (263mm leg translation error, due to SLERP interpolation for the $\sim 24\%$ of frames without observations).

The observation-conditioned diffusion approach combines both signals. Even without training for imputation, replacing CoMotion observations into the pretrained model’s denoising process reduces MPJPE to 113.0mm. This comes at a cost: velocity error increases from 487 to 595 mm/s, as noisy frame-by-frame replacements create temporal discontinuities. Training the model for imputation (Section 5.5.4) brings MPJPE to 107.6mm and sharply reduces velocity error to 431 mm/s, below the observation-free prior. The cutoff-schedule training

stage reduces MPJPE further to 104.7mm with the best temporal metrics (413 mm/s velocity, 7.5 m/s² acceleration). Post-inference refinement (Section 5.5.5) adds a 2.4mm MPJPE improvement and further reduces velocity error to 384 mm/s, with foot skating dropping from 6.1 to 3.1 mm/frame.

These gains are strictly additive: none of the imputation-related training stages degrade the model’s no-observation performance (Section 5.6.5, Table 5.3). Relative body pose accuracy (PA-MPJPE) plateaus at ~ 57 – 59 mm across all observation-conditioned variants, including the pretrained model. The progressive improvements concentrate in global positioning and temporal smoothness rather than relative body pose. Table 5.2 shows that the improvement holds across all activity categories.

5.6.4 Ablation Studies

Cutoff threshold. The replacement cutoff at $t = 50$ was selected based on a sweep over thresholds. Both $t = 0$ (no cutoff, equivalent to standard imputation) and $t = 100$ (too aggressive, discards most observation guidance) perform worse. The optimum lies where the transition from observation-helpful to observation-harmful occurs in the denoising trajectory.

Classifier-free guidance. CFG increases MPJPE by 0.1–1.4mm at all guidance scales above 1.0. In our setting, where the conditioning signal is noisy, CFG appears to amplify noise alongside signal. Additionally, the base model was pretrained with zero conditioning dropout, so there is no learned unconditional mode to contrast against.

Per-joint selective masking. Excluding the noisiest joint (spine2, 24.6° mean rotation error) from the observation mask increases error by 5.0mm. The observation mask encoder was trained only on all-or-nothing masks; partial masks are out of distribution.

Model variant	EgoHumans (MPJPE↓)		AMASS (MPJPE↓)	
	+ obs	no obs	+ obs ($f=1$)	no obs ($f=0$)
(a) Pretrained EgoAllo	113.0	135.5	35.4	116.8
(b) + Imputation fine-tuning (100k)	107.6	133.1	34.5	113.4
(c) + Cutoff-schedule training (10k)	104.7	131.6	33.5	112.0

Table 5.3: Cross-evaluation of model variants across observation conditions and datasets (MPJPE, mm). “EgoHumans + obs” uses each model’s best inference configuration with CoMotion observations: plain replacement for the pretrained model (a), per-modality cutoff and shape stabilization for trained variants (b, c). “EgoHumans no obs” drops all observations, reducing to CPF-conditioned prior only. “AMASS + obs” uses clean ground-truth observations at $f=1.0$ without cutoff (since clean observations help at all denoising steps). “AMASS no obs” is the original EgoAllo evaluation setting ($f=0$). No post-inference refinement.

Training duration. The cutoff-schedule training curve is non-monotonic, with the best MPJPE at 10,000 steps and a secondary minimum at 45,000 steps. Extended training beyond this shows no consistent improvement.

Stochastic DDIM. In our experiments, stochastic DDIM sampling ($\eta > 0$) provides no improvement over deterministic sampling ($\eta = 0$).

5.6.5 Verification on Synthetic Data and No-Degradation Analysis

Table 5.1 shows progressive improvement on EgoHumans with CoMotion observations. A natural concern is whether these gains come at the cost of degrading performance when observations are absent, or whether they are specific to CoMotion’s noise characteristics. We evaluate each model variant on two axes: with and without observations, on both EgoHumans and AMASS (Table 5.3).

No-observation performance is preserved. The “no obs” columns show that fine-tuning for imputation and adding the cutoff schedule do not degrade the model’s ability to

Model variant	$f=0$	$f=0.1$	$f=0.25$	$f=0.5$	$f=1.0$
(a) EgoAllo pretrained	116.8	95.5	72.5	50.6	35.4
(b) + Imputation fine-tuning	113.4	77.0	54.7	41.9	34.5
(c) + Cutoff-schedule training	112.0	75.9	53.2	40.6	33.5

Table 5.4: AMASS evaluation at varying observation fractions (MPJPE, mm). Clean ground-truth observations, no replacement cutoff. The cutoff-schedule-trained model (c) matches or improves on the imputation-trained model (b) at all fractions, including $f=0$.

estimate body pose from head motion alone. On AMASS at $f=0$ (the original EgoAllo evaluation setting with no observations), the pretrained model produces 116.8mm, imputation fine-tuning produces 113.4mm, and cutoff-schedule training produces 112.0mm. The later training stages marginally *improve* the motion prior. The same pattern holds on EgoHumans without observations: 135.5mm $\rightarrow$ 133.1mm $\rightarrow$ 131.6mm. The imputation-related training stages add capability for exploiting observations without sacrificing the base model’s motion prior.

AMASS verification with clean observations. The “AMASS + obs” column evaluates each variant with clean ground-truth observations at full coverage ($f=1.0$), without the replacement cutoff (since clean observations help at all denoising steps). Imputation fine-tuning (b) improves over the pretrained model (a), and cutoff-schedule training (c) improves further. This pattern holds across all observation fractions (Table 5.4), suggesting that the training stages improve the model’s ability to use observations in general, not just its robustness to CoMotion-specific noise.

Cutoff is specifically a remedy for observation noise. A diagnostic asymmetry supports this interpretation: applying the cutoff at inference time on clean AMASS observations *hurts* performance substantially (up to 24mm increase at full observation coverage). When observations are noise-free, they provide useful information at all denoising steps, and cutting them off discards that information. The cutoff improves results only when observation

noise exceeds diffusion noise at low t , which holds for CoMotion’s predictions but not for ground-truth observations. This is consistent with the crossover interpretation from Section 5.5.2.

5.6.6 Analysis

The results above report aggregate metrics over all 132 sequences. Aggregate MPJPE, however, averages over windows with frequent observations and windows with no observations at all. To understand where the system’s improvement concentrates, we decompose the evaluation into windows, 128-frame segments (~ 4.3 seconds, matching the model’s input size), and analyze error as a function of observation conditions.

Evaluation landscape. The 3,290 resulting (subject, window) pairs span a wide range of observation conditions. Only 3.3% of windows have near-zero exocentric observations ($\text{obs_frac} < 0.1$), while 46% have coverage above 0.9; the median observation fraction lies in the range $[0.7, 0.9]$. This distribution is a property of the evaluation setting, not the method: it reflects the geometry of multi-person egocentric interaction, where mutual visibility depends on spatial arrangement, the camera’s limited field of view ($\sim 110^\circ$ for Aria glasses), and activity structure. Because the distribution is skewed toward high coverage, the aggregate MPJPE is dominated by the system’s behavior when observations are plentiful.

Error and observation coverage. Figure 5.3 shows mean MPJPE as a function of observation fraction. The egocentric motion prior ($\sim 136\text{mm}$, dashed) is approximately flat, confirming that observation fraction does not correlate with intrinsic scene difficulty. Visual tracking with device alignment (CoMotion+SLAM) remains near the prior for most observation levels and improves only at the highest coverage, where dense observations reduce the impact of SLERP interpolation over temporal gaps.

All three fusion variants, which share the same inference-time replacement cutoff ($t=50$) and differ only in training, improve over both baselines at every coverage level. Even the zero-shot variant, which was never trained for imputation, benefits from the cutoff: observations provide useful guidance during early denoising while the cutoff prevents noise corruption during refinement. Each successive training stage (imputation training, then cutoff-schedule training) produces further improvement across all bins, with the cutoff-trained model improving monotonically from 119mm to 97mm.

Performance bounds. When `obs_frac` exceeds 0.9, the system achieves 97mm MPJPE. A persistent ~ 90 mm floor from the training-to-device CPF mismatch (Section 5.6.1) affects all windows equally regardless of observation conditions. PA-MPJPE, which removes this global offset, reaches 51mm under favorable conditions, providing a cleaner measure of actual body pose accuracy under the best achievable observation coverage.

At zero coverage, the system produces ~ 132 mm MPJPE, slightly better than the base prior (136mm), reflecting the improved motion model from fine-tuning (Table 5.3). This is expected: with no observations, the observation mask is all zeros and the model relies on CPF conditioning alone.

Activity categories. Performance varies across activity categories (Table 5.2), partly explained by differences in mutual visibility. Temporal coverage, the fraction of person-frames with at least one exocentric detection, varies substantially by activity (Table 5.5): activities with close interaction (badminton, tagging) have high observation rates, while activities where participants are far apart (volleyball, tennis) or focused on a shared task (lego/assembly) have more missing frames.

Activities with higher observation rates tend to have lower MPJPE, but temporal coverage alone does not explain all variation: fencing achieves the lowest error (82mm) at 89% cover-

Category	Missing (%)
badminton	13.2
basketball	18.0
fencing	11.0
legoassemble	40.1
tagging	11.5
tennis	74.0
volleyball	25.8
ALL	23.7

Table 5.5: Percentage of person-frames with no exocentric observation, by activity category. Activities with close interaction (fencing, tagging) have high mutual visibility, while activities with greater distance (volleyball, tennis) or shared object focus (lego assembly) have more missing frames.

age, while lego assembly (109mm at 60% coverage) outperforms volleyball (120mm at 74% coverage), suggesting that motion type and observation viewpoint quality also contribute.

Observer diversity. When multiple cameras observe the same person, their observations are fused heuristically before conditioning. Whether more principled multi-observer fusion could exploit angular diversity across viewpoints, beyond simply increasing temporal coverage, remains to be characterized.

Observation refinement on visible frames. To isolate the diffusion model’s effect on frames where observations were available, we evaluate PA-MPJPE restricted to observed frames only (Table 5.6). We use PA-MPJPE rather than global MPJPE because the model’s root position derives from CPF head tracking, not from CoMotion.

On the 76% of person-frames where CoMotion detects the subject, CoMotion’s best-score selection achieves 45.1mm PA-MPJPE, better than the diffusion model’s 53.9mm on the same frames. Post-inference refinement does not help (54.8mm), consistent with the physics-based constraints not targeting relative body pose. The diffusion model’s overall advantage in global MPJPE (Table 5.1) comes from filling the 24% of frames with no observations

Method	PA-MPJPE↓
CoMotion (best-score selection)	45.1
Ours (DDIM output, before refinement)	53.9
Ours (DDIM + post-inference refinement)	54.8

Table 5.6: Body pose accuracy (PA-MPJPE, mm) restricted to the 265,361 person-frames where CoMotion detected the subject from at least one observer. CoMotion’s direct visual estimate achieves better relative body pose than the diffusion model on these frames; the diffusion model’s advantage lies in temporal gap-filling and global positioning rather than refining already-observed poses. Lower is better.

and from device-derived global positioning, not from improving body pose on frames where visual evidence is available.

Temporal metrics and post-inference refinement. Replacing observations at every denoising step *increases* velocity error from 487 to 595 mm/s (Table 5.1): CoMotion produces frame-independent estimates, so consecutive predictions can differ substantially even for smooth motion. Imputation training reduces velocity error to 431 mm/s, and cutoff-schedule training brings it to 413 mm/s, both below the observation-free prior (487 mm/s). Post-inference refinement further reduces velocity error to 384 mm/s and foot skating from 6.1 to 3.1 mm/frame, slightly below the ground-truth skating rate of 3.3 mm/frame. These gains come from physics-based constraints (foot-floor contact, angular smoothness) rather than from observation information. The same optimizer improves temporal metrics by a comparable amount even when no observations are used.

Joint-specific angular accuracy. PA-MPJPE plateaus at ~ 57 – 58 mm across all observation-conditioned variants, which might suggest that training stages improve only global positioning. Per-joint angular errors tell a different story. Mean per-joint angular error (MPJAE) for legs decreases progressively: 24.7° (prior only) $\rightarrow 21.8^\circ$ (pretrained EgoAllo) $\rightarrow 20.3^\circ$ (imputation training) $\rightarrow 19.8^\circ$ (cutoff-schedule training). Since MPJAE measures rotation accuracy independent of global translation, this confirms that the training stages improve

local body pose for legs, not just root positioning. Arms show the opposite pattern: Co-Motion’s direct observations (26.6°) outperform all diffusion variants ($27.2\text{--}28.3^\circ$), even in the global frame. The head-motion prior provides little information about arm configuration (arms move largely independently of the head), so the diffusion model regularizes arms toward average configurations. For root orientation, the diffusion model (17.4°) substantially outperforms the tracking baseline (22.3°), where SLERP interpolation of unobserved frames degrades orientation estimates.

Noise-aware cutoff by observation condition. If noisy observations help at high denoising noise but hurt at low noise, we would expect a specific interaction between the cutoff and observation density. At high `obs_frac`, the cutoff should matter most: more observation frames mean more opportunities for noise to corrupt fine-grained refinement. At low `obs_frac`, fewer frames are replaced and the cutoff has less to prevent.

Comparing the pretrained model to the cutoff-trained model by `obs_frac` bin, the improvement is smallest at near-zero coverage (5mm) and peaks at intermediate coverage (12mm at `obs_frac` in $[0.3, 0.5)$), then narrows to 7mm at `obs_frac` ≥ 0.9 . At near-zero coverage, few frames have observations to cut off. At intermediate coverage, noisy observations enter the denoising process frequently enough to affect the trajectory but not densely enough for noise to average out across frames. At the highest coverage, observations are dense and the gap between all-step and truncated replacement is smaller.

Detection confidence. Detection confidence from the visual tracker correlates with prediction accuracy at the frame level (Spearman $\rho = -0.434$ between detection confidence and per-frame MPJPE).

Observation noise characteristics. Analysis of 180,931 observed frames reveals that CoMotion’s body rotation predictions have a mean geodesic error of 17.1° per joint. Approximately 49% of this error is systematic bias (constant offset within temporal windows), and the noise is temporally correlated (lag-1 autocorrelation of 0.826). The high autocorrelation means that observation errors persist across frames rather than averaging out. Simple temporal averaging of consecutive observations would reduce i.i.d. noise but has limited effect on correlated errors. The spine2 joint dominates both rotation error (24.6°) and its downstream effect on joint positions (43.8mm via forward kinematics), suggesting that targeted improvement of trunk estimation in monocular trackers would have disproportionate impact on downstream fusion. Body shape estimates are more reliable than rotations, motivating the per-modality handling described in Section 5.5.3.

5.7 Discussion

Summary. We present an approach for multi-person pose estimation from egocentric cameras, integrating visual tracking with generative motion modeling. Each participant’s head-mounted device provides a head-motion prior, and body estimates from other participants’ cameras complement this with visually grounded pose. Anchoring these exocentric estimates to the subject’s own device trajectory via SLAM alignment improves their global positioning. A conditional diffusion model fuses both signals through imputation-based conditioning, filling temporal gaps with learned priors and refining observed poses. In our experiments, noisy visual observations improve the diffusion model’s coarse pose structure but corrupt fine details when the remaining diffusion noise drops below the observation noise. With a noise-aware conditioning schedule that stops replacement at this crossover, the approach reduces MPJPE from 136mm to 102mm on EgoHumans.

Limitations. The model trains entirely on single-person motion capture data (AMASS), which differs in distribution from the multi-person activities encountered at evaluation time. Evaluation is limited to EgoHumans, currently the only multi-person egocentric benchmark with dense ground truth and device calibration. How well the approach generalizes to other settings, with different activity types, participant counts, or device configurations, remains to be established.

Future directions. *Inter-person modeling.* In the current formulation, people in the scene are coupled through the observer-subject relationship: when one person’s camera captures another, those observations inform the subject’s pose estimate. A natural extension is to also model physical and social interactions between the estimated trajectories themselves. People collide, coordinate, and react to one another: a basketball player’s motion depends on the defender’s position, fencers make contact, and conversational partners adjust their body language in response to each other. Incorporating these dependencies, through inter-person attention or collision constraints within the diffusion and guidance processes, could improve plausibility, especially in close-interaction settings [36].

Observation quality and fusion. Multi-observer fusion in the current pipeline is heuristic. Integrating information from multiple concurrent observers more deeply, accounting for detection confidence and geometric consistency across viewpoints, could improve body pose estimates. Because the diffusion model conditions on SMPL body parameters from the tracker, the framework is not tied to any particular tracking or prior model. As more capable models become available, they can be used directly. Fine-tuning the current tracker for the egocentric domain would also be a straightforward improvement.

Noise-adaptive conditioning. The fixed replacement cutoff at $t = 50$ treats all observations identically regardless of their reliability. Conditioning the cutoff on per-frame detection confidence or per-joint noise estimates would allow the model to use high-confidence observations

deeper into denoising while discarding unreliable ones earlier.

Training data. The model trains on AMASS, a curated collection of single-person motion capture that differs in distribution from the multi-person activities in EgoHumans. Training on larger and more diverse motion data, including multi-person recordings, would reduce this distribution gap. Multi-person motion capture currently requires dense instrumentation, careful synchronization across devices and external camera arrays, and significant manual effort in post-processing. More automated pipelines for extracting training data from raw recordings would help make larger-scale data collection practical.

Chapter 6

Conclusion

6.1 Summary

This thesis has studied 3D inference under partial observability across four settings: single objects, indoor scenes, human bodies observed by external cameras, and dynamic multi-person settings where each person observes the others while all are in motion.

In Chapter 2, we compared viewer-centered and object-centered coordinate frames for single-view 3D shape prediction. Viewer-centered prediction generalizes better to unfamiliar objects; object-centered prediction favors familiar instances. The coordinate frame determines whether a model learns transferable shape-pose representations or relies on category recognition. Surface-based representations outperform voxels for novel categories.

In Chapter 3, we extended viewer-centered prediction from objects to full scenes using multi-layer depth, a representation that records multiple surface intersections along each camera ray. An Epipolar Feature Transformer transfers learned features to virtual viewpoints via perspective geometry, and the resulting system reconstructs complete scenes without object

detection or category-specific models.

The first two chapters address static subjects where the missing structure is geometric: surfaces hidden from the input viewpoint. The later chapters turn to human subjects, where observations depend on camera configuration and vary over time.

In Chapter 4, we identified camera viewpoint as the dominant source of cross-dataset generalization failure in 3D human pose estimation. Predicting viewpoint as an auxiliary task during training improves cross-dataset accuracy by 9–20mm, without changing the prediction coordinate frame. The auxiliary task acts as a regularizer: it encourages the learned representation to encode the geometric relationship between camera and body, reducing overfitting to dataset-specific viewpoint distributions.

In Chapter 5, we developed a conditional diffusion model for multi-person egocentric motion estimation. Head-mounted cameras create a duality: each person’s first-person view provides third-person observations of others. We fused an egocentric motion prior with noisy exocentric visual observations through imputation-based conditioning. We observe that noisy observations help during coarse denoising but harm fine-grained refinement; a noise-aware conditioning schedule addresses this, reducing mean per-joint position error from 136mm to 102mm while training only on single-person motion capture data.

A recurring theme across these chapters is that the geometric relationship between observer and subject matters for 3D inference.

In Chapter 2, the choice between viewer-centered and object-centered coordinate frames determines generalization behavior. In Chapter 4, camera viewpoint is not part of the prediction target, but modeling it explicitly as an auxiliary task regularizes the learned representation and improves cross-dataset transfer. In Chapter 5, the observer–subject relationship is at the center of the problem itself: each camera wearer is simultaneously an observer of others and an unobserved subject, and the availability of observations depends on who can

see whom at any given moment.

What is unobserved also changes across chapters. In Chapter 2, the hidden structure is the back of an object, static and determined by a single viewpoint. In Chapter 3, it is the hidden surfaces of an entire scene. In Chapters 4 and 5, the subjects are people who move, and the observations are intermittent over time. The underlying problem is shared, but the settings range from static objects to dynamic scenes with multiple moving people.

The nature of partial observability also differs. In Chapters 2 and 3, the input is clean and the uncertainty is structural: which surfaces are hidden. In Chapter 4, the inputs are clean but carry hidden bias from camera viewpoint. In Chapter 5, observations are noisy and intermittent, and this noise interacts with the inference process.

6.2 Limitations

Chapters 2 and 3 train on synthetic data (ShapeNet and SUNCG, respectively). The multi-layer depth representation assumes a fixed number of surface layers, limiting its applicability to scenes with more complex occlusion structure.

Chapter 4 does not address other sources of dataset bias beyond viewpoint, including background appearance, lighting, and body proportions. The regularization effect of the auxiliary task is empirically demonstrated but the mechanism is not fully explained.

Chapter 5 trains on single-person motion capture (AMASS), which differs in distribution from the multi-person activities in EgoHumans. Evaluation is on a single benchmark; generalization to other multi-person egocentric settings remains to be established.

6.3 Future Directions

Representation choices in modern 3D reconstruction. Chapters 2 and 3 show that how the output is parameterized affects what a model can learn and how well it generalizes. Since then, 3D reconstruction methods have changed substantially in both architecture and scale, and new types of 3D representations have broadened the space of parameterization choices. Whether these choices carry the same weight at larger scale, or whether other factors dominate, is an open question. These are empirical questions that could be studied with the methods and data available today.

Recovering motion from diverse observations. Chapter 5 fuses a learned motion prior with visual observations that are noisy and temporally sparse. Head motion and visual tracking are natural first choices because they are readily available from current devices and complement each other well. Individual modalities for human motion generation, including text, audio, and scene context, have each been studied extensively on their own. The challenge is integrating them with visual observations in dynamic, in-the-wild settings with multiple interacting people. Each source differs in what it reveals, how often it is available, and how reliable it is, and these differences interact with the generative process. As wearable devices grow more capable, the set of available signals will expand, but the core difficulty remains the same: fusing observations that differ in quality, coverage, and temporal density into a coherent estimate. In-the-wild data with diverse modalities is increasingly available, but typically lacks precise body annotations. Obtaining accurate ground truth currently requires either controlled studio capture or extensive manual post-processing, neither of which scales easily to the volume and variety of available data. More automated annotation pipelines and methods that can learn from fragmented, partially labeled data would both help close this gap. Beyond observation fusion, modeling the physical and social interactions between people is another open direction that per-person estimation does not capture.

Bibliography

- [1] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015.
- [2] Ding-Yun Chen, Xiao-Pei Tian, Yu-Te Shen, and Ming Ouhyoung. On visual similarity based 3D model retrieval. In *Computer Graphics Forum*, volume 22, pages 223–232. Wiley Online Library, 2003.
- [3] Ling-Hao Chen, Jiawei Zhang, Yewen Li, Yiren Pang, Xiaobo Xia, and Tongliang Liu. HumanMAC: Masked motion completion for human motion prediction. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9544–9555, 2023.
- [4] Weifeng Chen, Zhao Fu, Dawei Yang, and Jia Deng. Single-image depth perception in the wild. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 730–738, 2016.
- [5] Christopher B Choy, Danfei Xu, JunYoung Gwak, Kevin Chen, and Silvio Savarese. 3d-r2n2: A unified approach for single and multi-view 3d object reconstruction. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2016.
- [6] Setareh Cohan, Guy Tevet, Daniele Reda, Xue Bin Peng, and Michiel van de Panne. Flexible motion in-betweening with diffusion models. In *ACM SIGGRAPH 2024 Conference Papers*, 2024.
- [7] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.
- [8] Angela Dai, Matthias Nießner, Michael Zollhöfer, Shahram Izadi, and Christian Theobalt. BundleFusion: Real-time globally consistent 3D reconstruction using on-the-fly surface reintegration. *ACM Transactions on Graphics*, 36(3):24:1–24:18, 2017.
- [9] Angela Dai, Charles Ruizhongtai Qi, and Matthias Nießner. Shape completion using 3d-encoder-predictor cnns and shape synthesis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6545–6554, 2017.

- [10] Helisa Dharmo, Keisuke Tateno, Iro Laina, Nassir Navab, and Federico Tombari. Peek-ing behind objects: Layered depth prediction from a single image. *Pattern Recognition Letters*, 125:333–340, 2019.
- [11] Alexey Dosovitskiy, Jost Tobias Springenberg, and Thomas Brox. Learning to generate chairs with convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1538–1546, 2015.
- [12] David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 2366–2374, 2014.
- [13] Haoqiang Fan, Hao Su, and Leonidas Guibas. A point set generation network for 3d object reconstruction from a single image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2463–2471, 2017.
- [14] Simon Fuhrmann and Michael Goesele. Floating scale surface reconstruction. *ACM Transactions on Graphics (TOG)*, 33(4):46, 2014.
- [15] Valentin Gabeur, Jean-Sébastien Franco, Xavier Martin, Cordelia Schmid, and Gregory Rogez. Moulding humans: Non-parametric 3d human shape estimation from single images. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2019.
- [16] Matheus Gadelha, Rui Wang, and Subhransu Maji. Multiresolution tree networks for 3d point cloud processing. In *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018.
- [17] Soumya Ghosh and Erik B Sudderth. Nonparametric learning for layered segmentation of natural images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2272–2279, 2012.
- [18] Rohit Girdhar, David F Fouhey, Mikel Rodriguez, and Abhinav Gupta. Learning a predictable and generative vector representation for objects. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 484–499. Springer, 2016.
- [19] Kristen Grauman, Andrew Westbury, Lorenzo Torresani, Kris Kitani, Jitendra Malik, Triantafyllos Afouras, Kumar Ashutosh, Vijay Baiyya, Siddhant Bansal, Bikram Boote, et al. Ego-Exo4D: Understanding skilled human activity from first- and third-person perspectives. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 19383–19400, 2024.
- [20] Saurabh Gupta, Pablo Arbeláez, Ross Girshick, and Jitendra Malik. Aligning 3d models to rgb-d images of cluttered scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.
- [21] Christian Häne, Shubham Tulsiani, and Jitendra Malik. Hierarchical surface prediction for 3d object reconstruction. In *IEEE International Conference on 3D Vision (3DV)*, pages 412–420, 2017.

- [22] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.
- [23] Catalin Ionescu, Dragos Papava, Vlad Olaru, and Cristian Sminchisescu. Human3.6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 2014.
- [24] Phillip Isola and Ce Liu. Scene collaging: Analysis and synthesis of natural images with semantic layers. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 3048–3055, 2013.
- [25] Shahram Izadi, David Kim, Otmar Hilliges, David Molyneaux, Richard Newcombe, Pushmeet Kohli, Jamie Shotton, Steve Hodges, Dustin Freeman, Andrew Davison, et al. Kinectfusion: real-time 3d reconstruction and interaction using a moving depth camera. In *Proceedings of the 24th annual ACM symposium on User interface software and technology*, pages 559–568. ACM, 2011.
- [26] Hamid Izadinia, Qi Shan, and Steven M Seitz. Im2cad. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.
- [27] Max Jaderberg, Karen Simonyan, Andrew Zisserman, and Koray Kavukcuoglu. Spatial transformer networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2015.
- [28] Dinghuang Ji, Junghyun Kwon, Max McFarland, and Silvio Savarese. Deep view morphing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.
- [29] Huaizu Jiang, Gustav Larsson, Michael Maire, Greg Shakhnarovich, and Erik Learned-Miller. Self-supervised relative depth learning for urban scene understanding. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018.
- [30] Jiaxi Jiang, Paul Streli, Manuel Meier, and Christian Holz. EgoPoser: Robust real-time egocentric pose estimation from sparse and intermittent observations everywhere. In *European Conference on Computer Vision (ECCV)*, pages 277–294, 2024.
- [31] Angjoo Kanazawa, Michael J. Black, David W. Jacobs, and Jitendra Malik. End-to-end recovery of human shape and pose. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7122–7131, 2018.
- [32] Abhishek Kar, Shubham Tulsiani, Joao Carreira, and Jitendra Malik. Category-specific object reconstruction from a single image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1966–1974, 2015.
- [33] Hiroharu Kato, Yoshitaka Ushiku, and Tatsuya Harada. Neural 3D mesh renderer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

- [34] Michael Kazhdan and Hugues Hoppe. Screened Poisson surface reconstruction. *ACM Transactions on Graphics (TOG)*, 32(3):29, 2013.
- [35] Rawal Khirodkar, Aayush Bansal, Lingni Ma, Richard Newcombe, Minh Vo, and Kris Kitani. Egohumans: An egocentric 3D multi-human benchmark. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023.
- [36] Rawal Khirodkar, Jyun-Ting Song, Jinkun Cao, Zhengyi Luo, and Kris Kitani. Harmony4D: A video dataset for in-the-wild close human interactions. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.
- [37] Muhammed Kocabas, Nikos Athanasiou, and Michael J. Black. VIBE: Video inference for human body pose and shape estimation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5253–5263, 2020.
- [38] Tejas D. Kulkarni, William F. Whitney, Pushmeet Kohli, and Joshua Tenenbaum. Deep convolutional inverse graphics network. In *Advances in Neural Information Processing Systems 28*, pages 2539–2547, 2015.
- [39] Jiaman Li, C. Karen Liu, and Jiajun Wu. Ego-body pose estimation via ego-head pose estimation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.
- [40] Chen-Hsuan Lin, Chen Kong, and Simon Lucey. Learning efficient point cloud generation for dense 3d object reconstruction. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2018.
- [41] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. SMPL: A skinned multi-person linear model. *ACM Transactions on Graphics (Proc. SIGGRAPH Asia)*, 34(6):248:1–248:16, 2015.
- [42] Andreas Lugmayr, Martin Danelljan, Andrés Romero, Fisher Yu, Radu Timofte, and Luc Van Gool. Repaint: Inpainting using denoising diffusion probabilistic models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11451–11461, 2022.
- [43] Zhaoliang Lun, Matheus Gadelha, Evangelos Kalogerakis, Subhransu Maji, and Rui Wang. 3D shape reconstruction from sketches via multi-view convolutional networks. In *2017 International Conference on 3D Vision (3DV)*, 2017.
- [44] Naureen Mahmood, Nima Ghorbani, Nikolaus F. Troje, Gerard Pons-Moll, and Michael J. Black. AMASS: Archive of motion capture as surface shapes. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 5442–5451, 2019.
- [45] Julieta Martinez, Rayat Hossain, Javier Romero, and James J. Little. A simple yet effective baseline for 3d human pose estimation. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2017.

- [46] Daniel Maturana and Sebastian Scherer. Voxnet: A 3d convolutional neural network for real-time object recognition. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 922–928, 2015.
- [47] Patricia A. McMullen and Martha J. Farah. Viewer-centered and object-centered representations in the recognition of naturalistic line drawings. *Psychological Science*, 2(4):275–278, 1991.
- [48] Dushyant Mehta, Helge Rhodin, Dan Casas, Pascal Fua, Oleksandr Sotnychenko, Weipeng Xu, and Christian Theobalt. Monocular 3d human pose estimation in the wild using improved cnn supervision. In *International Conference on 3D Vision (3DV)*, 2017.
- [49] Chenlin Meng, Yutong He, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. SDEdit: Guided image synthesis and editing with stochastic differential equations. In *International Conference on Learning Representations (ICLR)*, 2022.
- [50] Alejandro Newell, Peiyun Hu, Lahav Lipson, Stephan R. Richter, and Vladlen Koltun. Comotion: Concurrent multi-person 3D motion. In *International Conference on Learning Representations (ICLR)*, 2025.
- [51] Evonne Ng, Donglai Xiang, Hanbyul Joo, and Kristen Grauman. You2me: Inferring body pose in egocentric video via first and second person interactions. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9890–9900, 2020.
- [52] Andrea Nicastro, Ronald Clark, and Stefan Leutenegger. X-section: Cross-section prediction for enhanced RGB-D fusion. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2019.
- [53] Eunbyung Park, Jimei Yang, Ersin Yumer, Duygu Ceylan, and Alexander Berg. Transformation-grounded image generation network for novel 3d view synthesis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.
- [54] Georgios Pavlakos, Xiaowei Zhou, Konstantinos G. Derpanis, and Kostas Daniilidis. Coarse-to-fine volumetric prediction for single-image 3D human pose. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.
- [55] Charles R. Qi, Hao Su, Matthias Nießner, Angela Dai, Mengyuan Yan, and Leonidas J. Guibas. Volumetric and multi-view CNNs for object classification on 3D data. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.
- [56] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

- [57] Helge Rhodin, Mathieu Salzmann, and Pascal Fua. Unsupervised geometry-aware representation for 3d human pose estimation. In *European Conference on Computer Vision (ECCV)*, 2018.
- [58] Stephan R. Richter and Stefan Roth. Matryoshka networks: Predicting 3D geometry via nested shape layers. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1936–1944, 2018.
- [59] Gernot Riegler, Ali Osman Ulusoy, and Andreas Geiger. Octnet: Learning deep 3d representations at high resolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6620–6629, 2017.
- [60] Lawrence G Roberts. *Machine perception of three-dimensional solids*. PhD thesis, Massachusetts Institute of Technology, 1963.
- [61] Jason Rock, Tanmay Gupta, Justin Thorsen, Junyoung Gwak, Daeyun Shin, and Derek Hoiem. Completing 3D object shape from one depth image. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2484–2493, 2015.
- [62] Javier Romero, Dimitrios Tzionas, and Michael J. Black. Embodied hands: Modeling and capturing hands and bodies together. *ACM Transactions on Graphics (Proc. SIGGRAPH Asia)*, 36(6):245:1–245:17, 2017.
- [63] Samuel Rota Bulò, Lorenzo Porzi, and Peter Kotschieder. In-place activated Batch-Norm for memory-optimized training of DNNs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.
- [64] Saeed Saadatnejad, Ali Rasekh, Mohammadreza Mofayez, Yasamin Medghalchi, Sara Rajabzadeh, Taylor Mordan, and Alexandre Alahi. A generic diffusion-based approach for 3D human pose prediction in the wild. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2023.
- [65] Jonathan Shade, Steven Gortler, Li-wei He, and Richard Szeliski. Layered depth images. In *ACM SIGGRAPH*, 1998.
- [66] Yonatan Shafir, Guy Tevet, Roy Kapon, and Amit H. Bermano. Human motion diffusion as a generative prior. In *International Conference on Learning Representations (ICLR)*, 2024.
- [67] Daeyun Shin, Charless Fowlkes, and Derek Hoiem. Pixels, voxels, and views: A study of shape representations for single view 3D object shape prediction. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3061–3069, 2018.
- [68] Daeyun Shin, Zhile Ren, Erik Sudderth, and Charless Fowlkes. 3D scene reconstruction with multi-layer depth and epipolar transformers. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2172–2182, 2019.

- [69] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from rgb-d images. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2012.
- [70] Edward Smith, Scott Fujimoto, and David Meger. Multi-view silhouette and depth decomposition for high resolution 3d object representation. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 6479–6489, 2018.
- [71] Edward J. Smith and David Meger. Improved adversarial systems for 3d object generation and reconstruction. In *Conference on Robot Learning (CoRL)*, pages 87–96, 2017.
- [72] Paul Smith, Tom Drummond, and Roberto Cipolla. Layered motion segmentation and depth ordering by tracking edges. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(4):479–494, 2004.
- [73] Noah Snavely, Steven M. Seitz, and Richard Szeliski. Modeling the world from internet photo collections. *International Journal of Computer Vision (IJCV)*, 80(2):189–210, 2008.
- [74] Amir Arsalan Soltani, Haibin Huang, Jiajun Wu, Tejas D. Kulkarni, and Joshua B. Tenenbaum. Synthesizing 3D shapes via modeling multi-view depth maps and silhouettes with deep generative networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.
- [75] Shuran Song, Fisher Yu, Andy Zeng, Angel X Chang, Manolis Savva, and Thomas Funkhouser. Semantic scene completion from a single depth image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.
- [76] Hang Su, Subhransu Maji, Evangelos Kalogerakis, and Erik Learned-Miller. Multi-view convolutional neural networks for 3D shape recognition. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 945–953, 2015.
- [77] Hang Su, Varun Jampani, Deqing Sun, Subhransu Maji, Evangelos Kalogerakis, Ming-Hsuan Yang, and Jan Kautz. SPLATNet: Sparse lattice networks for point cloud processing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2530–2539, 2018.
- [78] Hao Su, Charles R. Qi, Yangyan Li, and Leonidas J. Guibas. Render for CNN: View-point estimation in images using CNNs trained with rendered 3D model views. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2015.
- [79] Hao Su, Fan Wang, Li Yi, and Leonidas J. Guibas. 3D-assisted feature synthesis for novel views of an object. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2015.
- [80] Deqing Sun, Erik B Sudderth, and Michael J Black. Layered segmentation and optical flow estimation over time. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1768–1775, 2012.

- [81] Xiao Sun, Bin Xiao, Fangyin Wei, Shuang Liang, and Yichen Wei. Integral human pose regression. In *European Conference on Computer Vision (ECCV)*, 2018.
- [82] Michael J. Tarr and Steven Pinker. When does human object recognition use a viewer-centered reference frame? *Psychological Science*, 1(4):253–256, 1990.
- [83] Maxim Tatarchenko, Alexey Dosovitskiy, and Thomas Brox. Multi-view 3D models from single images with a convolutional network. In *European Conference on Computer Vision*, pages 322–337. Springer, 2016.
- [84] Maxim Tatarchenko, Alexey Dosovitskiy, and Thomas Brox. Octree generating networks: Efficient convolutional architectures for high-resolution 3D outputs. In *Proceedings of the IEEE International Conference on Computer Vision*, 2017.
- [85] Guy Tevet, Sigal Raab, Brian Gordon, Yonatan Shafir, Daniel Cohen-Or, and Amit H. Bermano. Human motion diffusion model. In *International Conference on Learning Representations (ICLR)*, 2023.
- [86] Antonio Torralba and Alexei A. Efros. Unbiased look at dataset bias. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011.
- [87] Shubham Tulsiani, Tinghui Zhou, Alexei A Efros, and Jitendra Malik. Multi-view supervision for single-view reconstruction via differentiable ray consistency. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.
- [88] Shubham Tulsiani, Saurabh Gupta, David Fouhey, Alexei A Efros, and Jitendra Malik. Factoring shape, pose, and layout from the 2d image of a 3d scene. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.
- [89] Shubham Tulsiani, Richard Tucker, and Noah Snavely. Layer-structured 3d scene inference via view synthesis. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018.
- [90] Gül Varol, Javier Romero, Xavier Martin, Naureen Mahmood, Michael J. Black, Ivan Laptev, and Cordelia Schmid. Learning from synthetic humans. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.
- [91] Timo von Marcard, Roberto Henschel, Michael J. Black, Bodo Rosenhahn, and Gerard Pons-Moll. Recovering accurate 3d human pose in the wild using imus and a moving camera. In *European Conference on Computer Vision (ECCV)*, 2018.
- [92] J. Y. A. Wang and E. H. Adelson. Representing moving images with layers. *IEEE Transactions on Image Processing (TIP)*, 3(5):625–638, September 1994.
- [93] Nanyang Wang, Yinda Zhang, Zhuwen Li, Yanwei Fu, Wei Liu, and Yu-Gang Jiang. Pixel2mesh: Generating 3d mesh models from single rgb images. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018.

- [94] Zhe Wang, Daeyun Shin, and Charless Fowlkes. Predicting camera viewpoint improves cross-dataset generalization for 3D human pose estimation. In *European Conference on Computer Vision Workshops (ECCVW)*, pages 523–540, 2020.
- [95] Zhe Wang, Liyan Chen, Shaurya Rathore, Daeyun Shin, and Charless Fowlkes. Geometric pose affordance: Monocular 3D human pose estimation with scene constraints. In *European Conference on Computer Vision Workshops (ECCVW)*, pages 3–18, 2022.
- [96] Jiajun Wu, Yifan Wang, Tianfan Xue, Xingyuan Sun, William T. Freeman, and Joshua B. Tenenbaum. MarrNet: 3D shape reconstruction via 2.5D sketches. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 540–550, 2017.
- [97] Jiajun Wu, Chengkai Zhang, Xiuming Zhang, Zhoutong Zhang, William T. Freeman, and Joshua B. Tenenbaum. Learning Shape Priors for Single-View 3D Completion and Reconstruction. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018.
- [98] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3D ShapeNets: A deep representation for volumetric shapes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1912–1920, 2015.
- [99] Yu Xiang, Roozbeh Mottaghi, and Silvio Savarese. Beyond PASCAL: A benchmark for 3D object detection in the wild. In *IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2014.
- [100] Bin Xiao, Haiping Wu, and Yichen Wei. Simple baselines for human pose estimation and tracking. In *European Conference on Computer Vision (ECCV)*, 2018.
- [101] Yiming Xie, Varun Jampani, Lei Zhong, Deqing Sun, and Huaizu Jiang. Omnicon-trol: Control any joint at any time for human motion generation. In *International Conference on Learning Representations (ICLR)*, 2024.
- [102] Xinchun Yan, Jimei Yang, Ersin Yumer, Yijie Guo, and Honglak Lee. Perspective transformer nets: Learning single-view 3d object reconstruction without 3d supervision. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 1696–1704, 2016.
- [103] Bo Yang, Stefano Rosa, Andrew Markham, Niki Trigoni, and Hongkai Wen. Dense 3D object reconstruction from a single depth view. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2018.
- [104] Jimei Yang, Scott E. Reed, Ming-Hsuan Yang, and Honglak Lee. Weakly-supervised disentangling with recurrent transformations for 3D view synthesis. In *Advances in Neural Information Processing Systems*, pages 1099–1107, 2015.
- [105] Vickie Ye, Georgios Pavlakos, Jitendra Malik, and Angjoo Kanazawa. Decoupling human and camera motion from videos in the wild. In *Proceedings of the IEEE/CVF*

- Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 21222–21232, 2023.
- [106] Brent Yi, Vickie Ye, Maya Zheng, Yunqi Li, Lea Müller, Georgios Pavlakos, Yi Ma, Jitendra Malik, and Angjoo Kanazawa. Estimating body and hand motion in an ego-sensed world. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7072–7084, 2025.
 - [107] Ye Yuan, Umar Iqbal, Pavlo Molchanov, Kris Kitani, and Jan Kautz. GLAMR: Global occlusion-aware human mesh recovery with dynamic cameras. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11038–11049, 2022.
 - [108] Siwei Zhang, Qianli Ma, Yan Zhang, Zhiyin Qian, Taein Kwon, Marc Pollefeys, Federica Bogo, and Siyu Tang. EgoBody: Human body shape and motion of interacting people from head-mounted devices. In *European Conference on Computer Vision (ECCV)*, pages 180–200, 2022.
 - [109] Siwei Zhang, Bharat Lal Bhatnagar, Yuanlu Xu, Alexander Winkler, Petr Kadlecek, Siyu Tang, and Federica Bogo. RoHM: Robust human motion reconstruction via diffusion. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14606–14617, 2024.
 - [110] Xiuming Zhang, Zhoutong Zhang, Chengkai Zhang, Joshua B. Tenenbaum, William T. Freeman, and Jiajun Wu. Learning to reconstruct shapes from unseen classes. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 2263–2274, 2018.
 - [111] Yinda Zhang, Shuran Song, Ersin Yumer, Manolis Savva, Joon-Young Lee, Hailin Jin, and Thomas Funkhouser. Physically-based rendering for indoor scene understanding using convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.
 - [112] Yunhan Zhao, Shu Kong, Daeyun Shin, and Charless Fowlkes. Domain decluttering: Simplifying images to mitigate synthetic-real domain shift and improve depth estimation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3330–3340, 2020.
 - [113] Tinghui Zhou, Matthew Brown, Noah Snavely, and David G. Lowe. Unsupervised learning of depth and ego-motion from video. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.
 - [114] Tinghui Zhou, Richard Tucker, John Flynn, Graham Fyffe, and Noah Snavely. Stereo magnification: Learning view synthesis using multiplane images. *ACM Transactions on Graphics (SIGGRAPH)*, 37(4):65, 2018.
 - [115] Xingyi Zhou, Qixing Huang, Xiao Sun, Xiangyang Xue, and Yichen Wei. Towards 3d human pose estimation in the wild: A weakly-supervised approach. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2017.

- [116] Zhenyao Zhu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Multi-view perceptron: a deep model for learning face identity and view representations. In *Advances in Neural Information Processing Systems*, pages 217–225, 2014.
- [117] Christian Zimmermann and Thomas Brox. Learning to estimate 3d hand pose from single rgb images. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2017.
- [118] Chuhan Zou, Ersin Yumer, Jimei Yang, Duygu Ceylan, and Derek Hoiem. 3d-prnn: Generating shape primitives with recurrent neural networks. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 900–909, 2017.
- [119] Chuhan Zou, Ruiqi Guo, Zhizhong Li, and Derek Hoiem. Complete 3d scene parsing from an rgb-d image. *International Journal of Computer Vision (IJCV)*, 127(2):143–162, 2019.

Appendix A

Additional Material for Chapter 2

This appendix contains additional qualitative results and network architecture details for the experiments in Chapter 2. Sections A.1–A.2 show qualitative prediction examples; Section A.3 provides full network parameters.

A.1 RGB-based Shape Prediction

A.1.1 Synthetic Training Examples

Figure A.1 shows sample RGB training images generated using the RenderForCNN [78] pipeline. Our dataset has 2.4M renderings of 34,000 3D CAD models from 12 object categories. In the RGB multi-surface experiment, each example has six output ground truth depth images rendered on the faces of a cube: front, back, left, right, top, bottom. Figure A.2 shows the corresponding viewer-centered and object-centered output ground truth depth images.

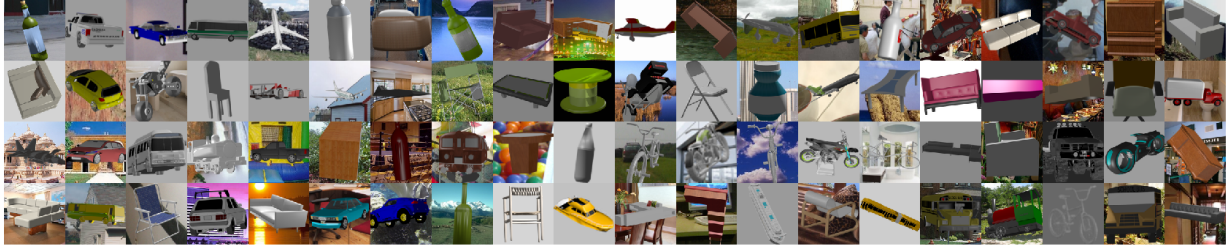


Figure A.1: Sample RGB training images generated using RenderForCNN. Each example has six output ground truth depth images rendered on the faces of a cube.

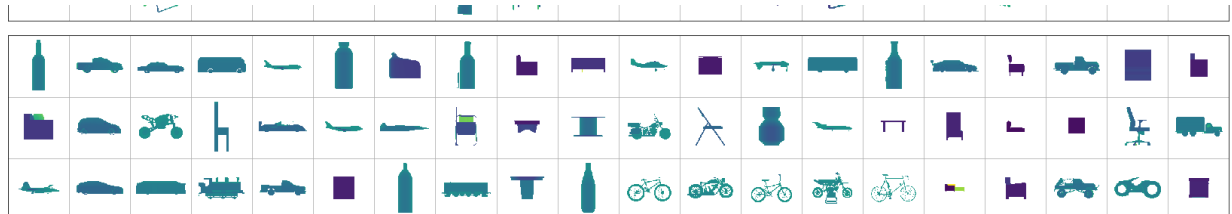


Figure A.2: Viewer-centered (top) and object-centered (bottom) output ground truth depth images for the corresponding input images in Figure A.1. Only showing the first view in each example.

A.1.2 Viewer-centered and Object-centered Coordinates

Does object-centered representation make shape prediction more of a categorization problem? Qualitative results support our initial hypothesis that object-centered models tend to correspond more directly to category recognition. The object-centered model often predicts a shape that looks good but is in an entirely different object category than the input image (Section A.1.2). The viewer-centered model tends not to make these kinds of mistakes; instead, errors tend to be overly simplified shapes or incorrect poses (Section A.1.2).

Object-centered Failure Examples

Figures A.3 and A.4 show examples where the object-centered model predicts shapes in the wrong category, while the viewer-centered model produces a closer match. Each row shows the input image (left), multi-surface prediction rendered from the viewer-centered and object-centered models.

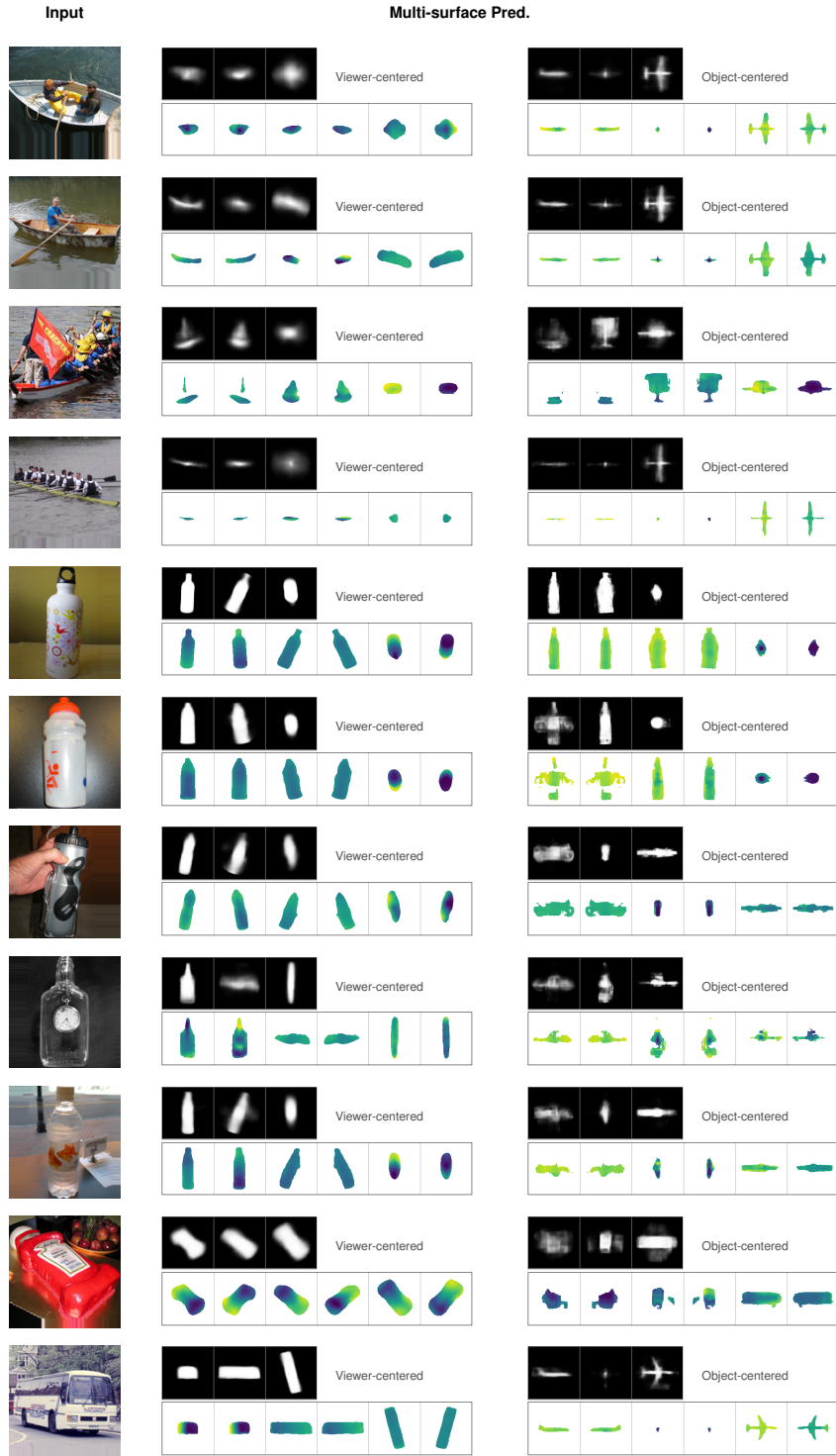


Figure A.3: Object-centered failure examples (part 1). The object-centered model often predicts a plausible shape from the wrong category.

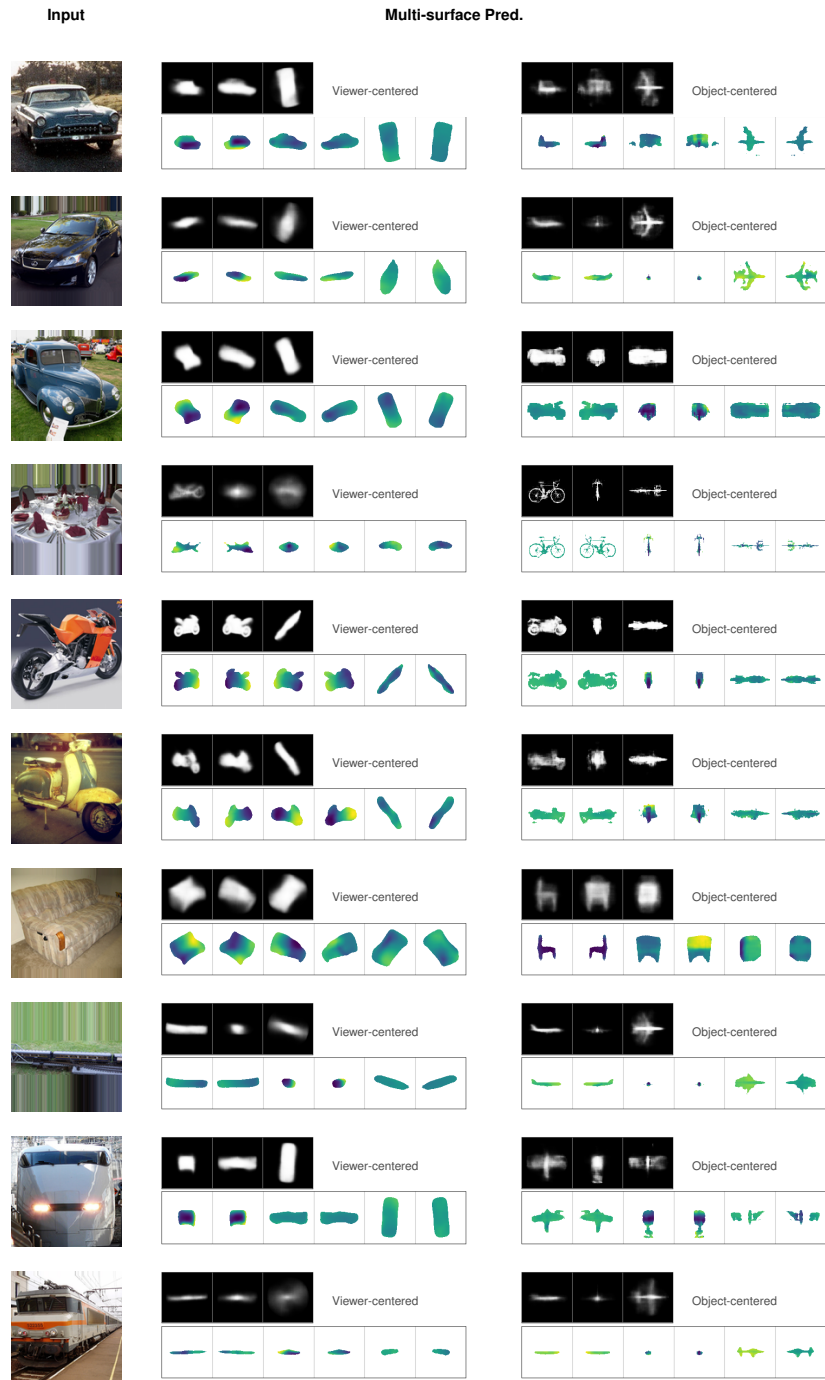


Figure A.4: Object-centered failure examples (part 2).

Viewer-centered Failure Examples

Figure A.5 shows examples where the viewer-centered model produces incorrect predictions. Unlike the object-centered failures, these errors typically involve overly simplified shapes or incorrect poses rather than category mismatches.

Successful Examples

Figures A.6 and A.7 show selected examples where both coordinate systems produce successful predictions.

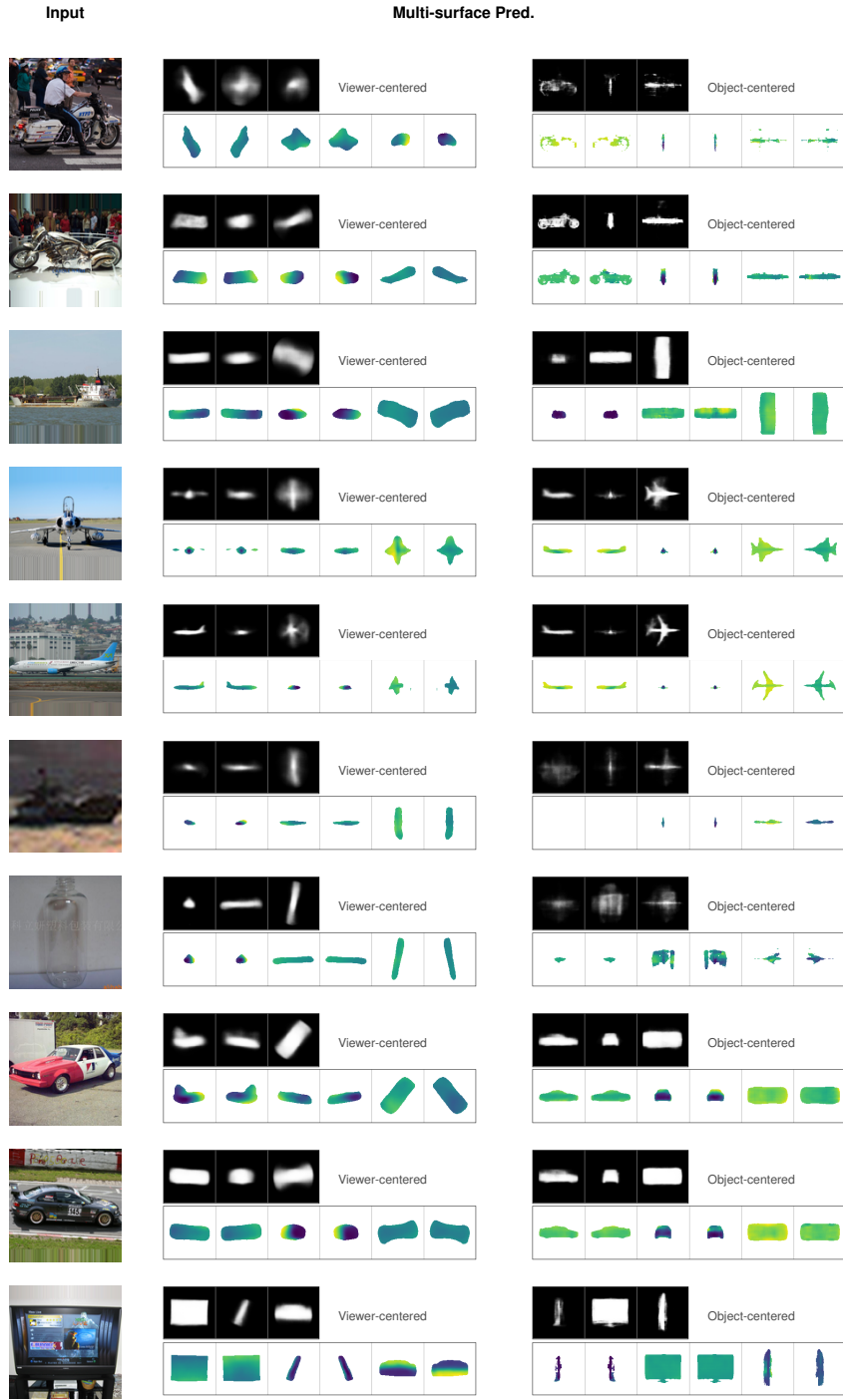


Figure A.5: Viewer-centered failure examples. Errors tend to be oversimplified shapes or incorrect poses, not category mismatches.

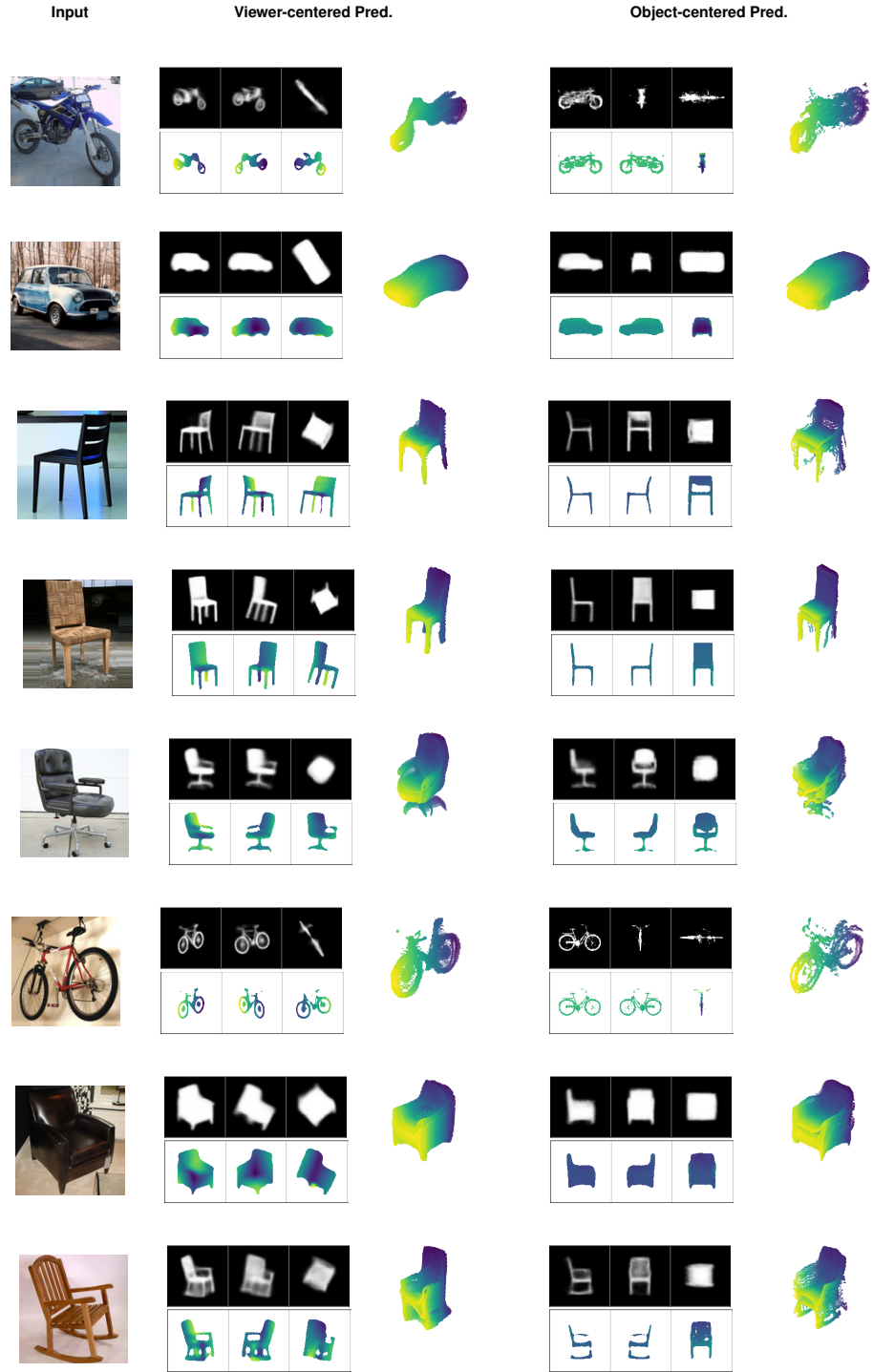


Figure A.6: Successful prediction examples (part 1). Both viewer-centered and object-centered models produce accurate reconstructions.

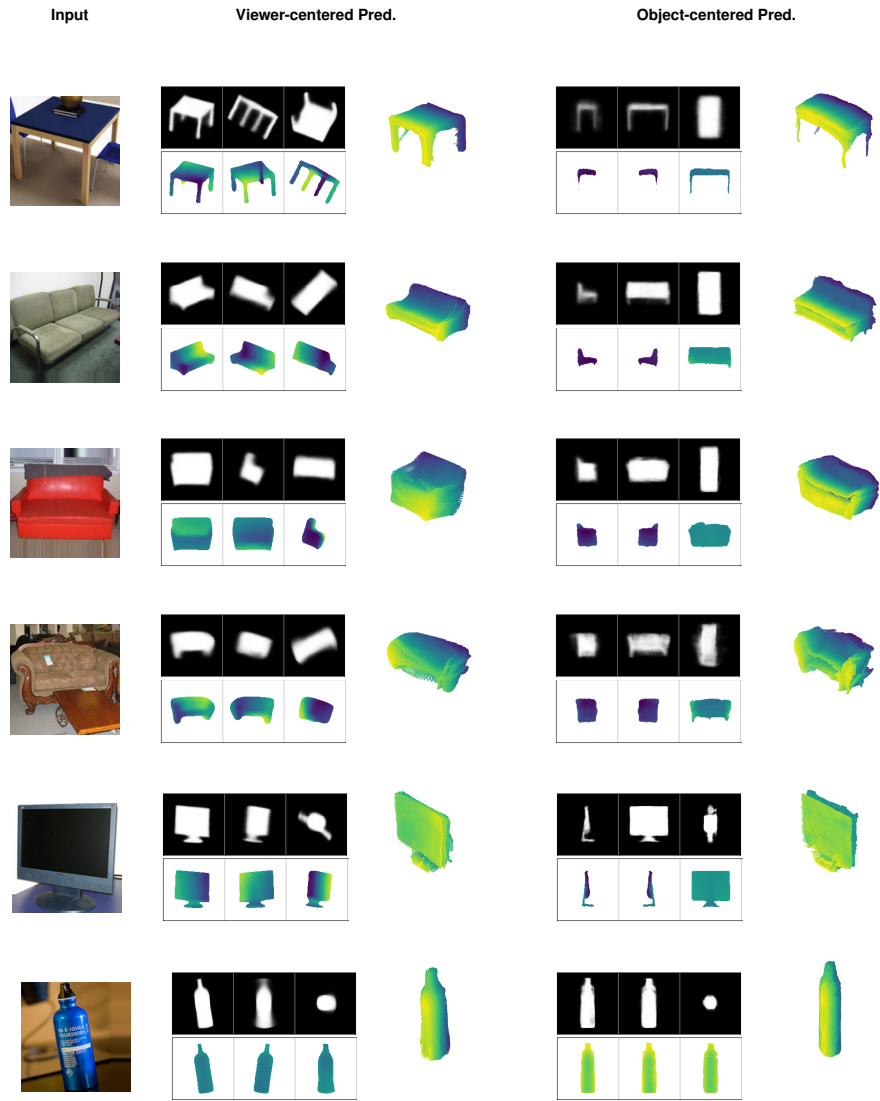


Figure A.7: Successful prediction examples (part 2).

A.2 Depth-based Shape Prediction

A.2.1 Multi-surface vs. Voxel Shape Representations

We evaluate on the SHREC’12 dataset with the following category splits:

- **Train+val:** bed, biplane, bookset, bookshelf, cellphone, city, classicpiano, computer, computerkeyboard, desklamp, door, face, glasses, guitar, handgun, helicopter, militaryvehicle, monitor, monoplane, mug, plier, quadruped, rectangletable, roundtable, singlehouse, skyscraper, sofa, spoon, tree, violin, bicycle, biped, fish, floorlamp, flyinginsect, bird, bottle, chess, deskphone, drum, humanhead, sword, train, truck, wheelchair
- **Test (NovelClass):** apartmenthouse, bus, car, cup, hand, homeplant, knife, motorcycle, nonflyinginsect, nonwheelchair, pianoboard, rocket, ship, submachinegun, trucknoncontainer

Figures A.8–A.12 show qualitative reconstruction results at three levels of familiarity: NovelView (same model, unseen viewpoint), NovelModel (same category, unseen model), and NovelClass (unseen category at test time). Each row shows the input depth image, the familiarity condition, voxel prediction, multi-surface depth prediction, the mesh reconstructed from the surface prediction, the result of Rock et al. [61], and the ground truth mesh.

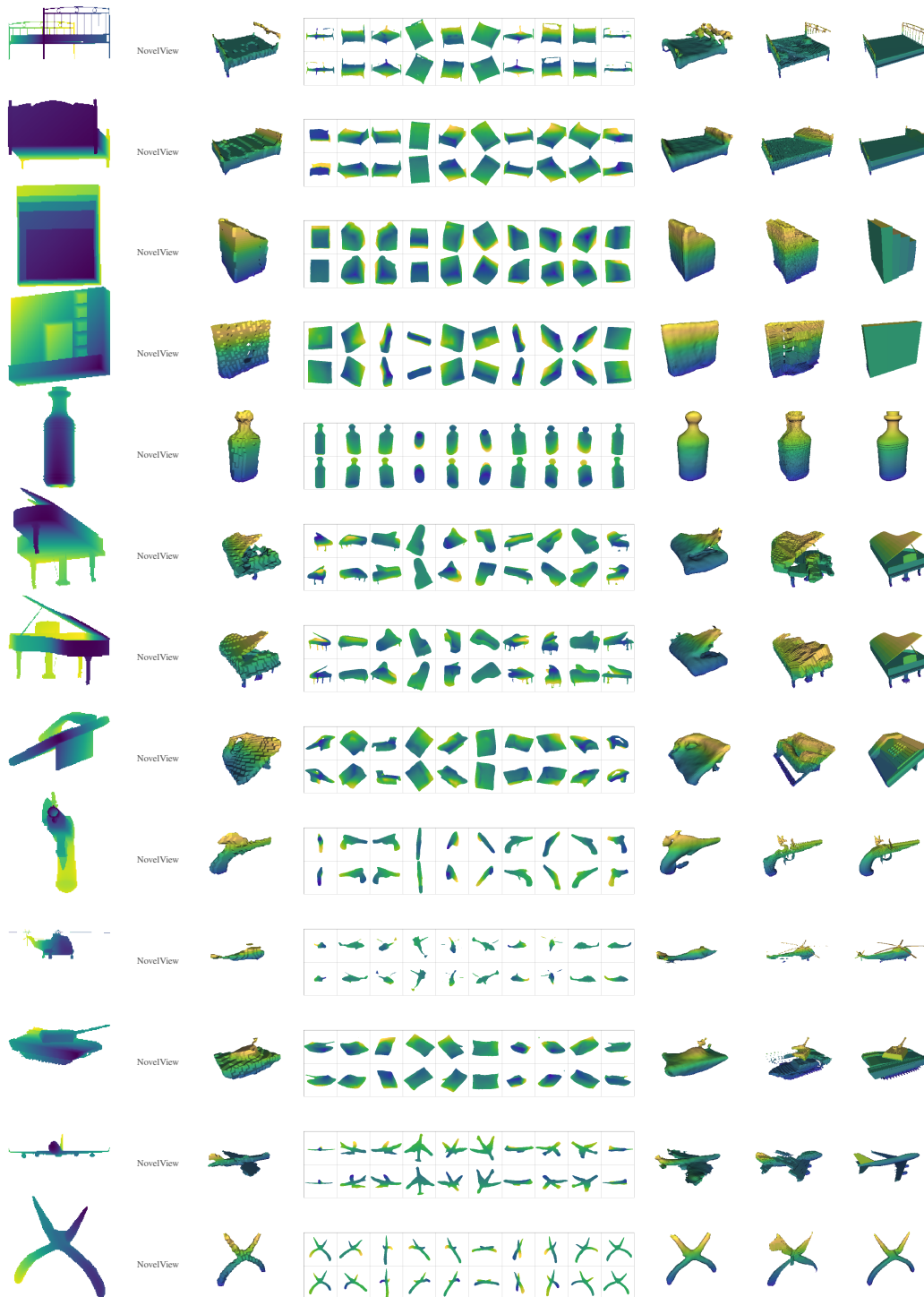


Figure A.8: Qualitative reconstruction results on SHREC'12 (part 1, NovelView examples).

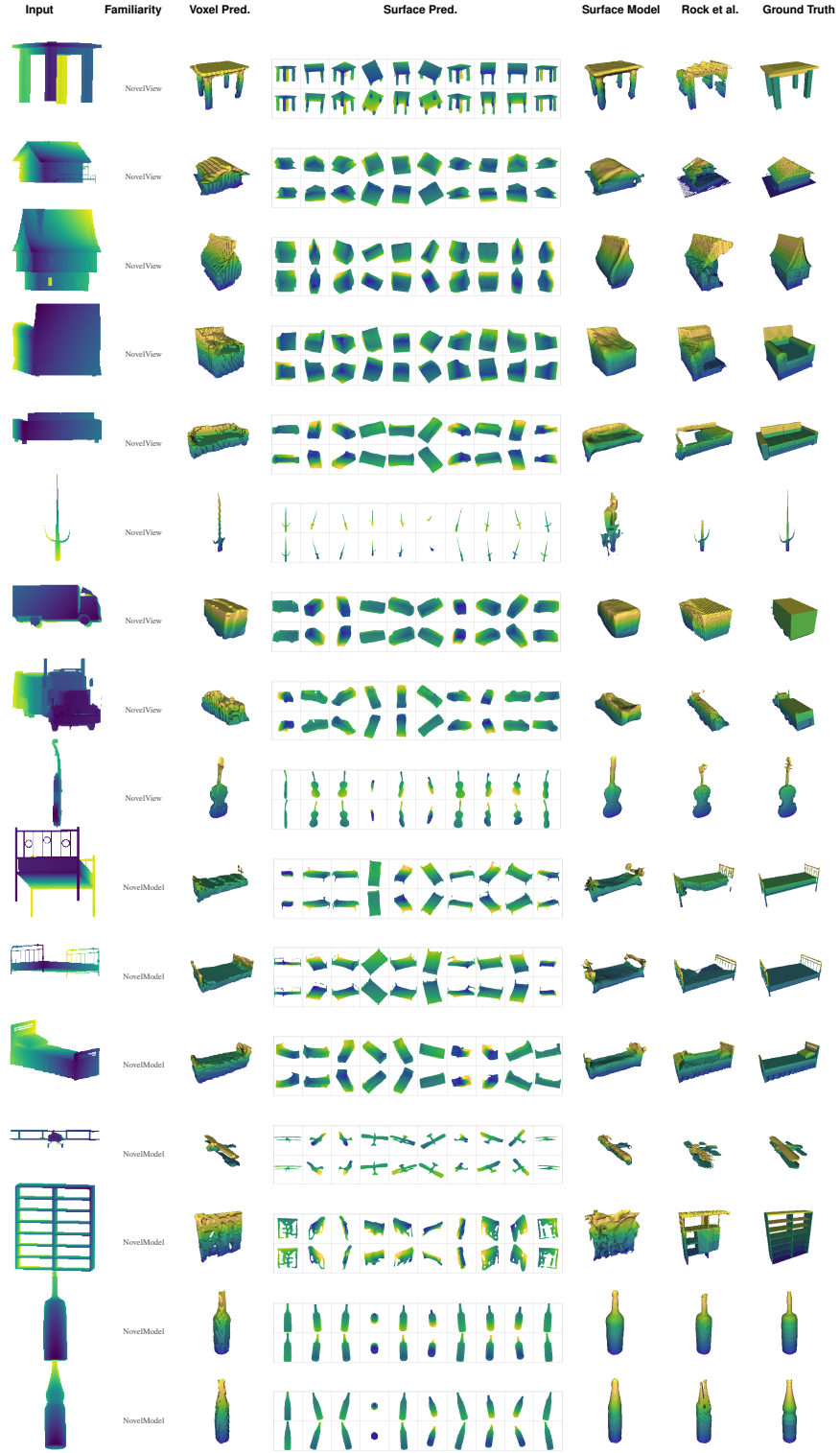


Figure A.9: Qualitative reconstruction results (part 2, NovelView and NovelModel examples).

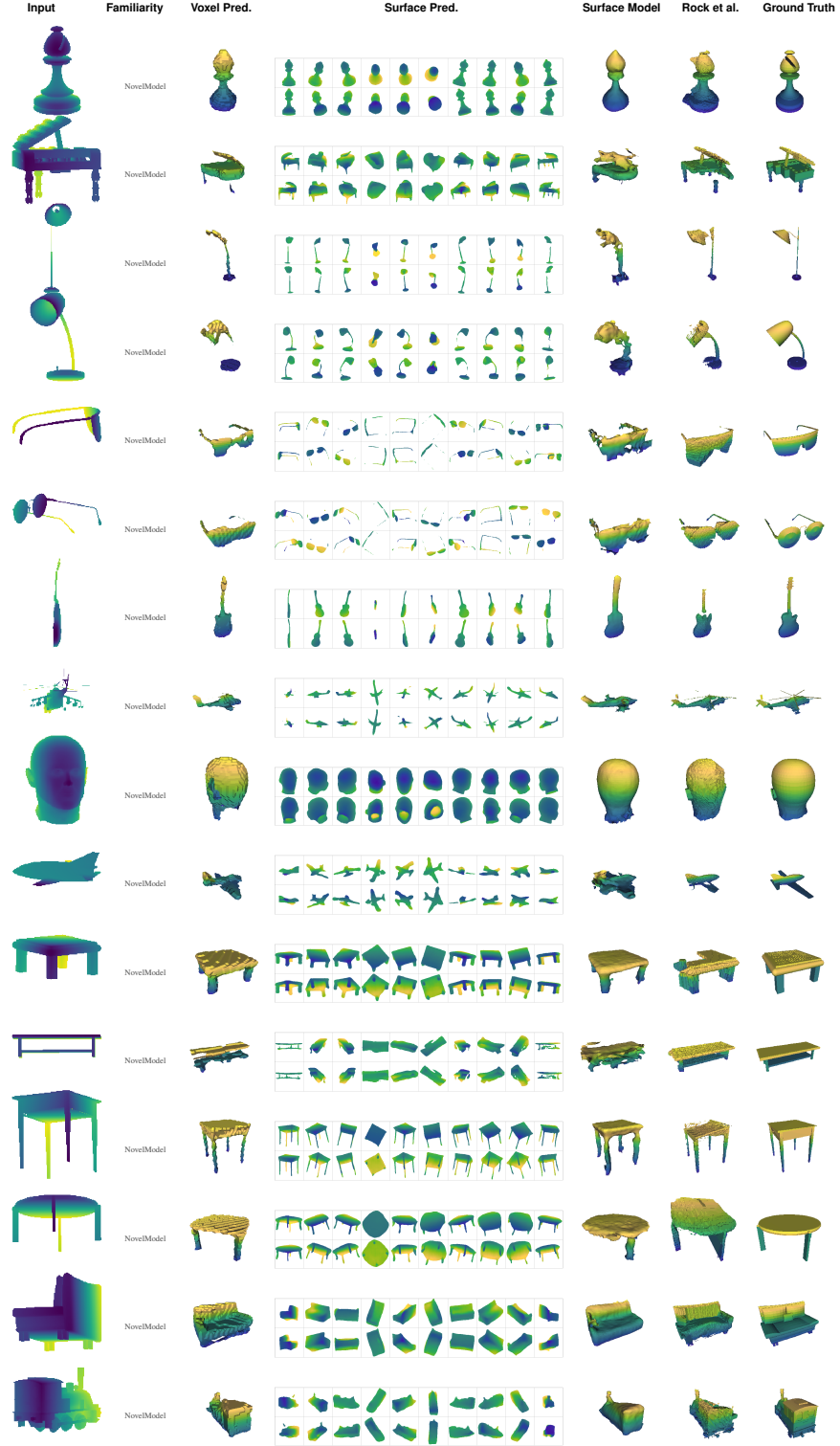


Figure A.10: Qualitative reconstruction results (part 3, NovelModel examples).

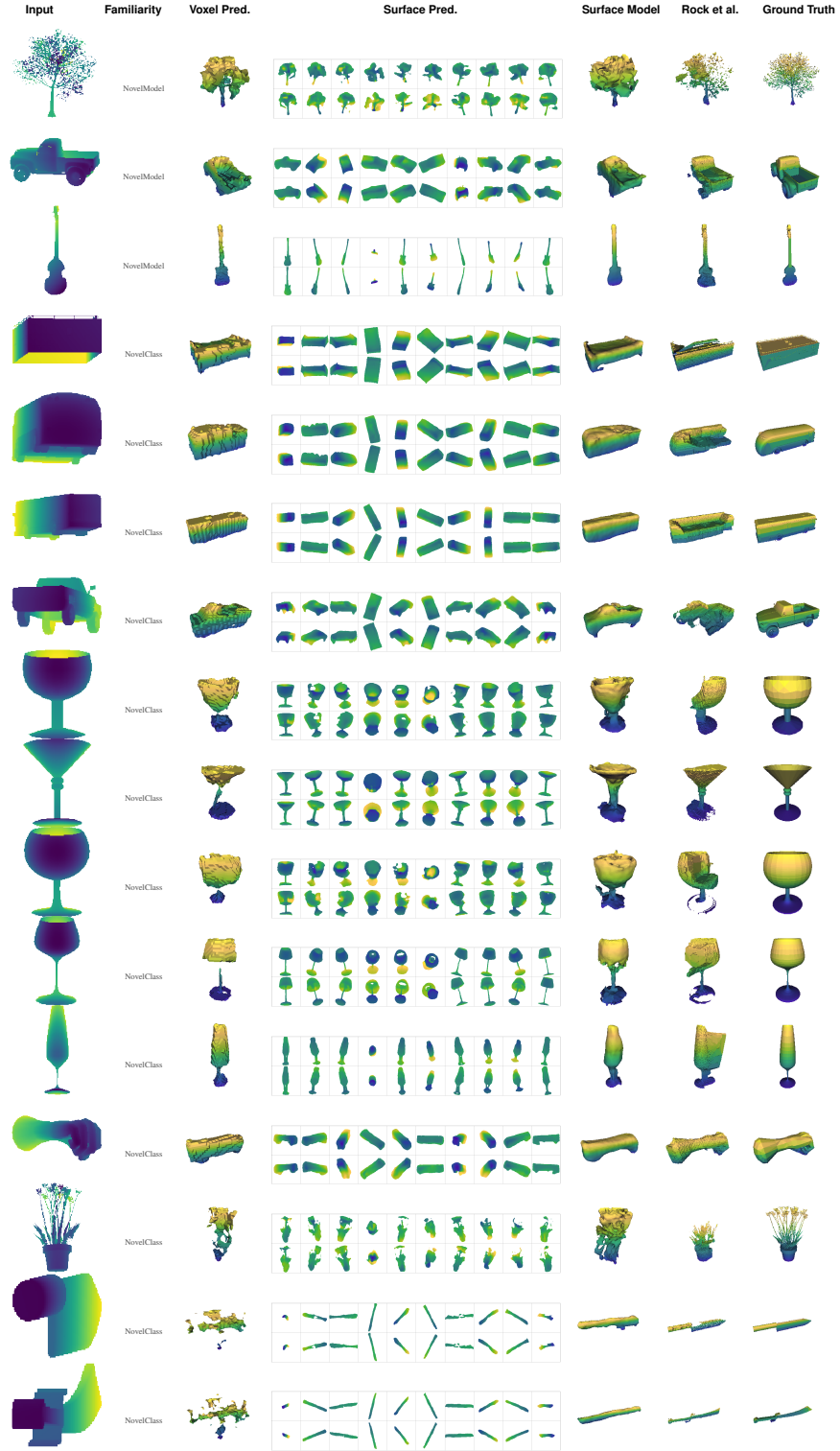
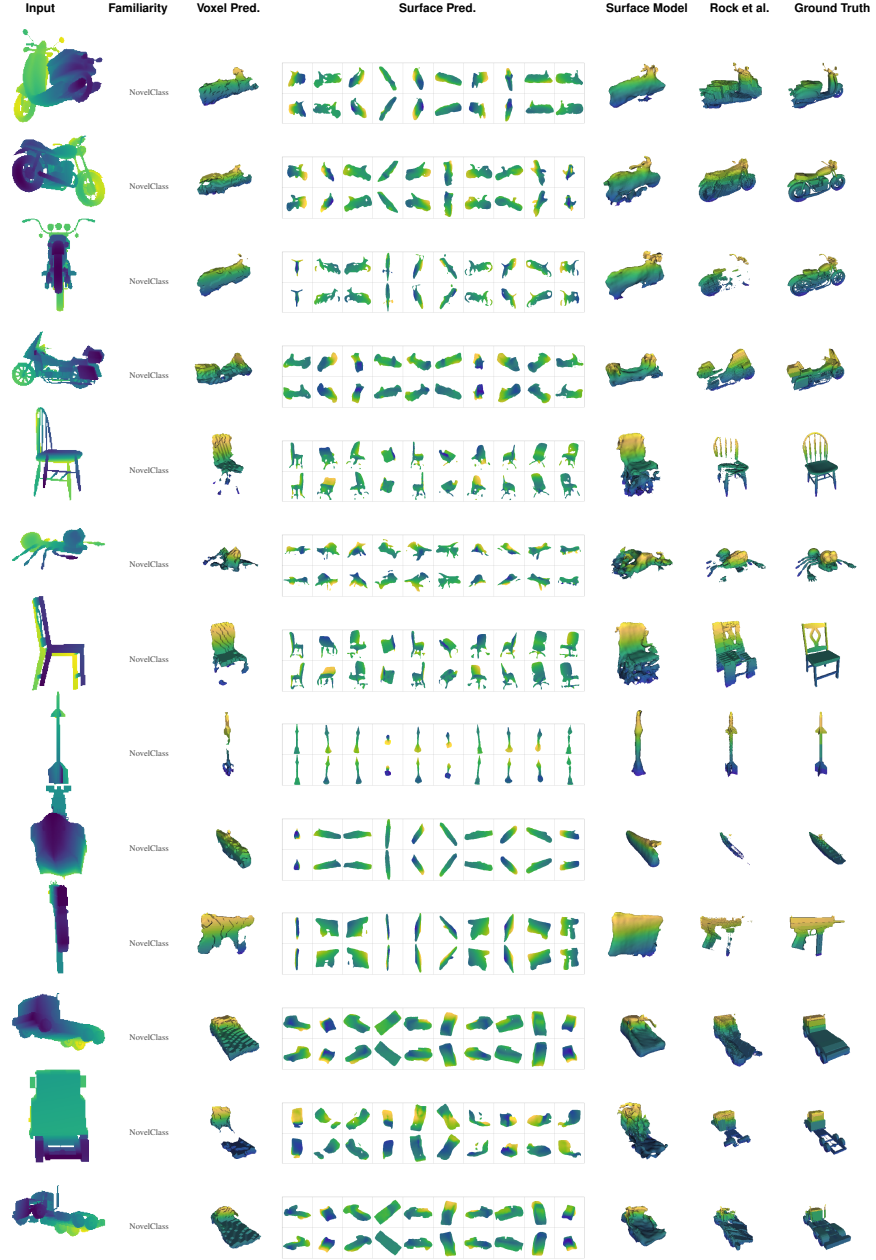


Figure A.11: Qualitative reconstruction results (part 4, NovelModel and NovelClass examples).



A.3 Network Architecture

Tables A.1–A.7 provide full network parameters for all architectures used in Chapter 2. All batch normalization and ReLU layers are omitted for brevity.

Table A.1: Network parameters for encoders E_d , E_s , E_h . ReLU and batch normalization layers are not shown.

<i>Depth encoder E_d:</i>				
Layer	Output size	Kernel	Stride	Repeats
Input depth	$128 \times 128 \times 1$			1
conv2d	$64 \times 64 \times 48$	7	2	1
residual unit	$64 \times 64 \times 48$	3	1	3
residual unit	$32 \times 32 \times 144$	3	2	1
residual unit	$32 \times 32 \times 144$	3	1	3
residual unit	$16 \times 16 \times 288$	3	2	1

<i>Silhouette encoder E_s:</i>				
Layer	Output size	Kernel	Stride	Repeats
Input silhouette	$128 \times 128 \times 1$			1
conv2d	$64 \times 64 \times 32$	7	2	1
residual unit	$64 \times 64 \times 32$	3	1	3
residual unit	$32 \times 32 \times 96$	3	2	1
residual unit	$32 \times 32 \times 96$	3	1	3
residual unit	$16 \times 16 \times 192$	3	2	1

<i>Combined encoder E_h:</i>				
Layer	Output size	Kernel	Stride	Repeats
residual unit	$16 \times 16 \times 480$	3	1	10
residual unit	$8 \times 8 \times 256$	3	2	1
residual unit	$8 \times 8 \times 256$	3	1	4
residual unit	$4 \times 4 \times 512$	3	2	1
residual unit	$4 \times 4 \times 512$	3	1	4
FC $\rightarrow h$	4096			1

Table A.2: Network parameters for multi-surface decoders G_d , G_s .

Layer	Output size	Kernel	Stride	Repeats
upconv2d	$8 \times 8 \times 256$	5	2	1
upconv2d	$16 \times 16 \times 128$	5	2	1
upconv2d	$32 \times 32 \times 64$	5	2	1
upconv2d	$64 \times 64 \times 32$	5	2	1
upconv2d	$128 \times 128 \times 1$	5	2	1

Table A.3: Network parameters for the volume decoder used in the voxel network.

Layer	Output size	Kernel	Stride	Repeats
upconv3d	$6 \times 6 \times 6 \times 512$	5	2	1
upconv3d	$12 \times 12 \times 12 \times 256$	5	2	1
upconv3d	$12 \times 12 \times 12 \times 128$	5	1	1
upconv3d	$24 \times 24 \times 24 \times 64$	5	2	1
upconv3d	$48 \times 48 \times 48 \times 1$	5	2	1

Table A.4: Network parameters for the classification layer in the 2.5D shape classification experiment. The classification experiment uses 370K renderings of 3D CAD models from the ModelNet40 dataset and is evaluated using an 80/20 train/test model split ratio.

Layer	Output size	Repeats
$h \rightarrow \text{FC}$	256	1
FC, softmax	40	1

Table A.5: Network parameters for the 6-view network encoder.

Layer	Output size	Kernel	Stride	Repeats
Input image	$128 \times 128 \times 3$			1
conv2d	$64 \times 64 \times 64$	7	2	1
max pool	$32 \times 32 \times 64$	3	2	1
residual unit	$32 \times 32 \times 256$	3	1	4
residual unit	$16 \times 16 \times 768$	3	2	7
residual unit	$8 \times 8 \times 2048$	3	2	4
average pool	$1 \times 1 \times 2048$	8	8	1

Table A.6: Network parameters for multi-surface decoders G_d , G_s used in the 6-view network.

Layer	Output size	Kernel	Stride	Repeats
upconv2d	$4 \times 4 \times 512$	4	2	1
upconv2d	$8 \times 8 \times 256$	4	2	1
upconv2d	$16 \times 16 \times 128$	4	2	1
upconv2d	$32 \times 32 \times 64$	4	2	1
upconv2d	$64 \times 64 \times 32$	4	2	1
upconv2d	$128 \times 128 \times 1$	4	2	1

Table A.7: Network parameters for the volume decoder used in the voxel network (corresponds to Tables A.5 and A.6).

Layer	Output size	Kernel	Stride	Repeats
upconv3d	$4 \times 4 \times 4 \times 512$	4	2	1
upconv3d	$8 \times 8 \times 8 \times 256$	4	2	1
upconv3d	$16 \times 16 \times 16 \times 128$	4	2	1
upconv3d	$32 \times 32 \times 32 \times 1$	4	2	1

Appendix B

Additional Material for Chapter 3

B.1 System Overview

We provide an overview of our 3D reconstruction system and additional qualitative examples in our **video**.

B.2 Training Data Generation

As we describe in Section 3.5.1, we generate the target multi-layer depth maps by performing multi-hit ray tracing on the ground-truth 3D mesh models. If an object instance is completely occluded (i.e. not visible at all from the first-layer depth map), it is ignored in the subsequent layers. The Physically-based Rendering [111] dataset ignores objects in “person” and “plant” categories, so those categories are also ignored when we generate our depth maps. The complete list of room envelope categories (according to NYUv2 mapping) are as follows: wall, floor, ceiling, door, floor_mat, window, curtain, blinds, picture, mirror, fireplace, roof, and whiteboard. In our experiments, all room envelope categories are merged into a single

“background” category. In Figure B.1, we provide a layer-wise 3D visualization of our multi-layer depth representation. Figure B.3 illustrates our surface precision-recall metrics.

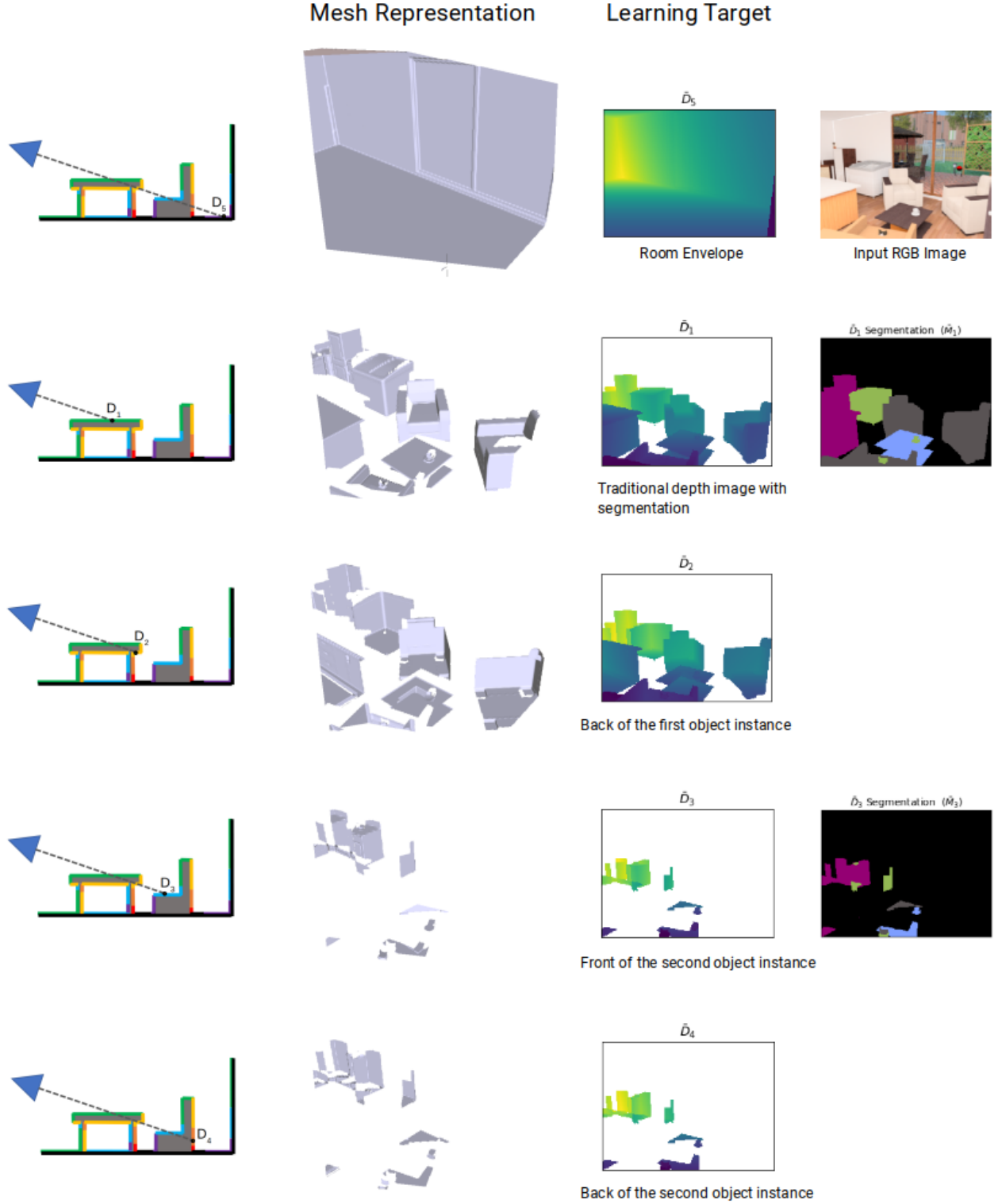


Figure B.1: Layer-wise illustration of our multi-layer depth representation in 3D. Table 3.1 reports an empirical analysis which shows that the five-layer representation ($\bar{D}_{1,2,3,4,5}$) covers 92% of the scene geometry inside the viewing frustum.

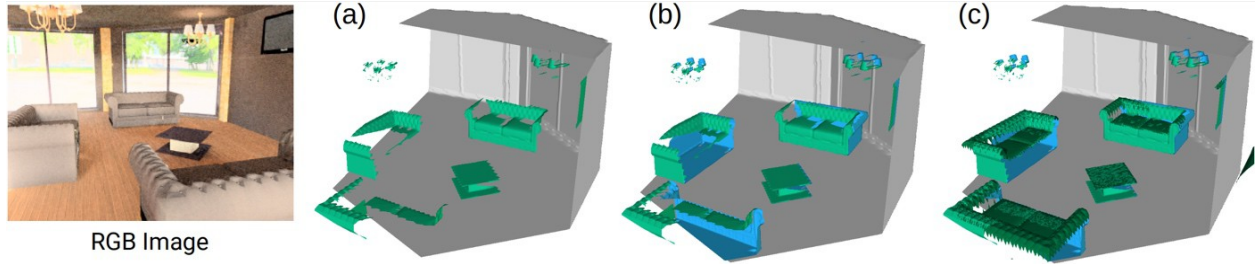


Figure B.2: Illustration of ground-truth depth layers. (a, b): 2.5D depth representation cannot accurately encode the geometry of surfaces which are nearly tangent to the viewing direction. (b): We model both the front and back surfaces of objects as seen from the input camera. (c): The tangent surfaces are sampled more densely in the additional virtual view (dark green). Table 3.2 shows the effect of augmenting the frontal predictions with the virtual view predictions.

B.3 Representing the Back Surfaces of Objects

Without the back surfaces, ground truth depth layers ($\bar{D}_{1,3,5}$) cover only 82% of the scene geometry inside the viewing frustum (vs. 92% including frontal surfaces — refer to Table 3.1 for full comparison). Figure B.2(a) visualizes $\bar{D}_{1,3,5}$, without the back surfaces. This representation, *layered depth image* (LDI) [65], was originally developed in the computer graphics community [65] as an algorithm for rendering textured depth images using parallax transformation. Works based on prediction of LDI or its variants [89, 114] therefore do not represent the invisible back surfaces of objects. Prediction of back surfaces enables volumetric inference in our epipolar transformation.

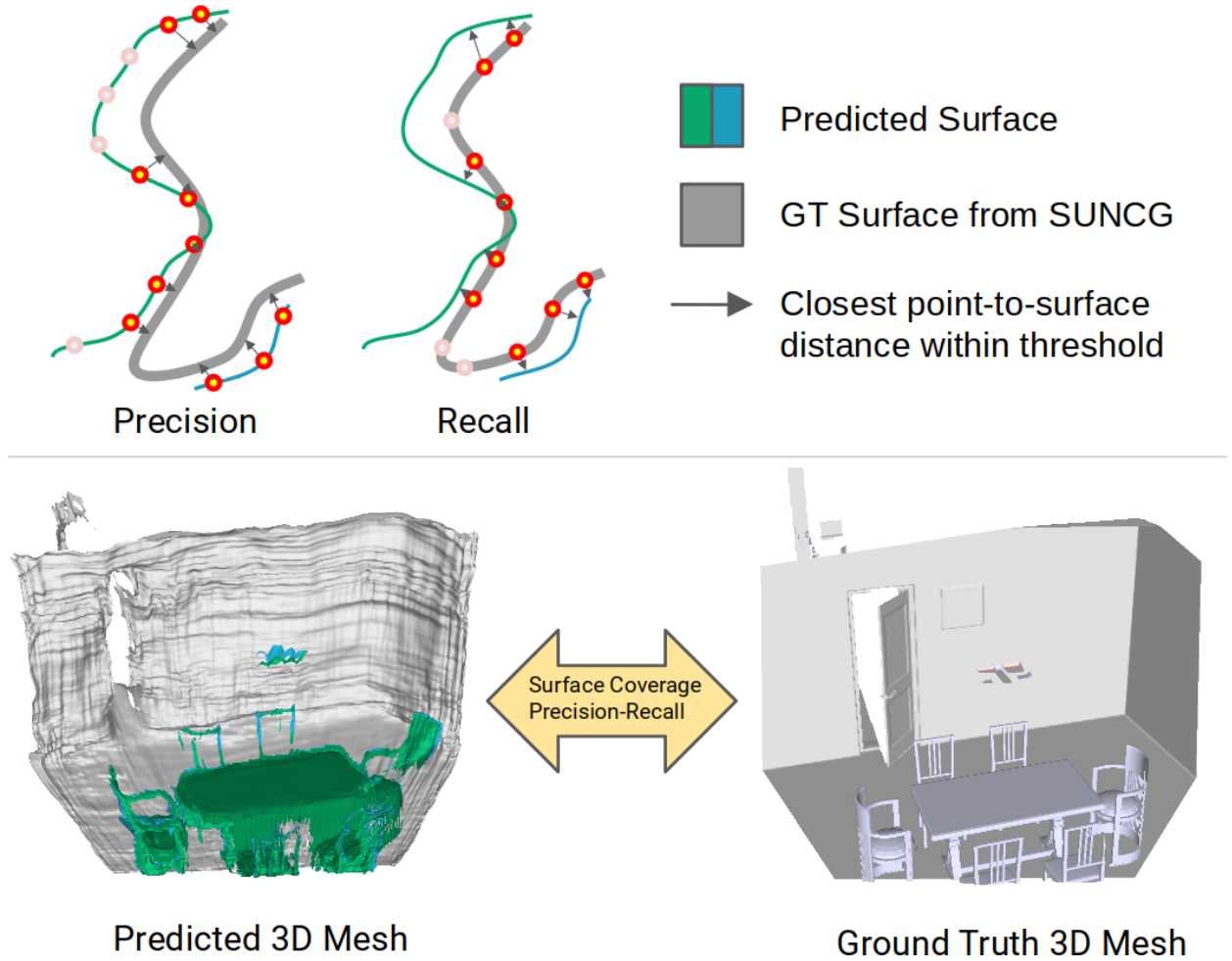


Figure B.3: Illustration of our 3D precision-recall metrics. *Top*: We perform a bidirectional surface coverage evaluation on the reconstructed triangle meshes. *Bottom*: The ground truth mesh consists of all 3D surfaces within the viewing frustum and in front of the room envelope. We take the union of the predicted meshes from different views in world coordinates. This allows us to perform a layer-wise evaluation (e.g. Figure 3.5).

B.4 Multi-layer Depth Prediction

See Figure B.9 for network parameters of our multi-layer depth prediction model. All batch normalization layers have momentum 0.005, and all activation layers are Leaky ReLUs layers with $\alpha = 0.01$. We use In-place Activated BatchNorm [63] for all of our batch normalization layers. We trained the network for 40 epochs.

B.5 Multi-layer Semantic Segmentation

See Figure B.10 for network parameters of multi-layer semantic segmentation. We construct a binary mask for all foreground objects, and define segmentation mask M_l as all non-background pixels at layer l . As mentioned in Section 3.3.1, D_1 and D_2 have the same segmentation due to symmetry, so we only segment layers 1 and 3. The purpose of the foreground object labels is to be used as a supervisory signal for feature extraction F_{seg} , which is used as input to our Epipolar Feature Transformer Networks.

B.6 Virtual-view Prediction

See Figure B.12 and B.13 for network parameters of our virtual-view height map prediction and segmentation models. The height map prediction network is trained to minimize foreground pixel losses. At test time, the background mask predicted by the segmentation network is used to zero out the floor pixels. The floor height is assumed to be zero in world coordinates. An alternate approach is minimizing both foreground and background losses and thus allowing the height map predictor to implicitly segment the floor by predicting zeros. We experimented with both architectures and found the explicit segmentation approach to perform better.

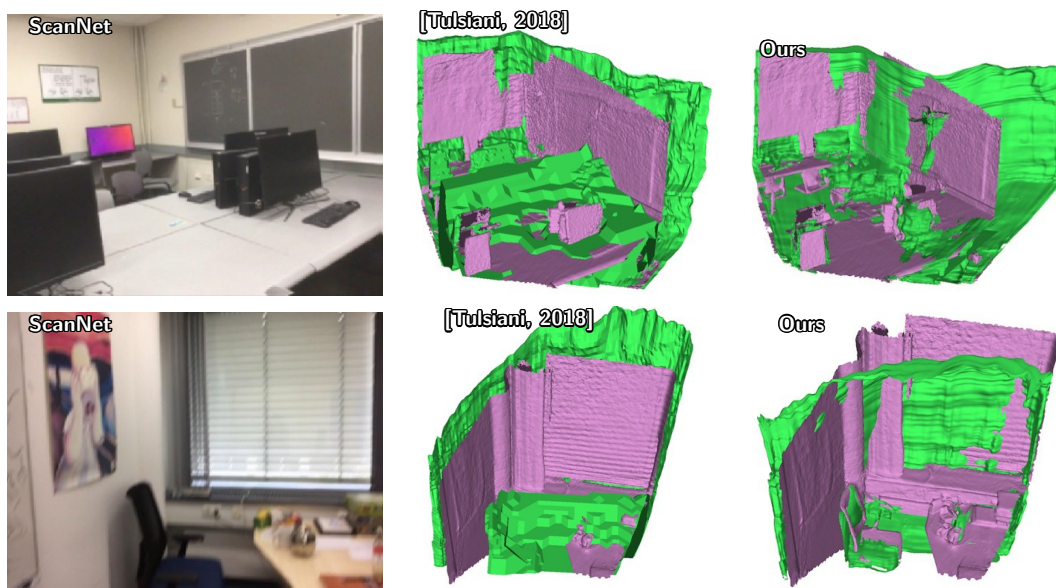


Figure B.4: Evaluation of 3D reconstruction on the ScanNet [7] dataset, where green regions are detected objects and pink regions are ground truth.



Figure B.5: Volumetric evaluation of our predicted multi-layer depth maps on the SUNCG [75] dataset.

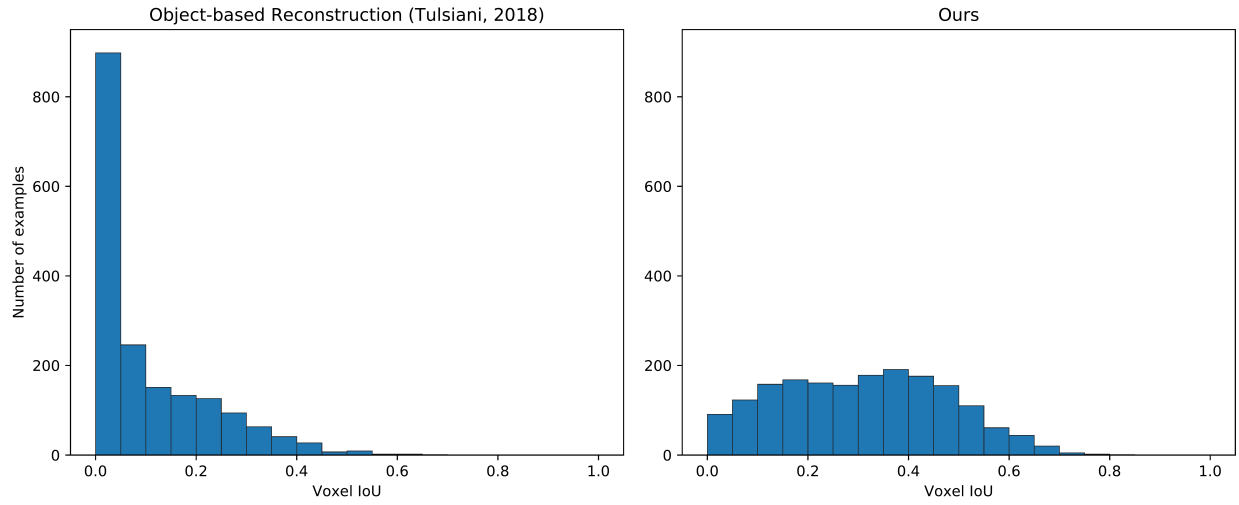


Figure B.6: Distribution of voxel intersection-over-union on SUNCG (excluding room layouts). We observe that object-based reconstruction is sensitive to detection failure and misalignment on thin structures.

B.7 Voxelization of Multi-layer Depth Prediction

Given a 10m^3 voxel grid of resolution 400 (equivalently, 2.5cm^3) with a bounding box ranging from $(-5,-5,-10)$ to $(5,5,0)$ in camera space, we project the center of each voxel into the predicted depth maps. If the depth value for the projected voxel falls in the first object interval (D_1, D_2) or the occluded object interval (D_3, D_4) , the voxel is marked occupied. We evaluate our voxelization against the SUNCG ground truth object meshes inside the viewing frustum, voxelized using the Binvex software which implements z-buffer based carving and parity voting methods. We also voxelize the predicted Factored3D [88] objects (same meshes evaluated in Figure 3.5) using Binvex under the same setting as the ground truth. We randomly select 1800 examples from the test set and compute the intersection-over-union of all objects in the scene. In addition to Figure 3.7, Figure B.5 shows a visualization of our voxels, colored according to the predicted semantic labeling. Figure B.6 shows a histogram of voxel intersection-over-union values.

B.8 Predictions on NYU and SUNCG

Figures B.8 and B.7 show additional 3D scene reconstruction results.

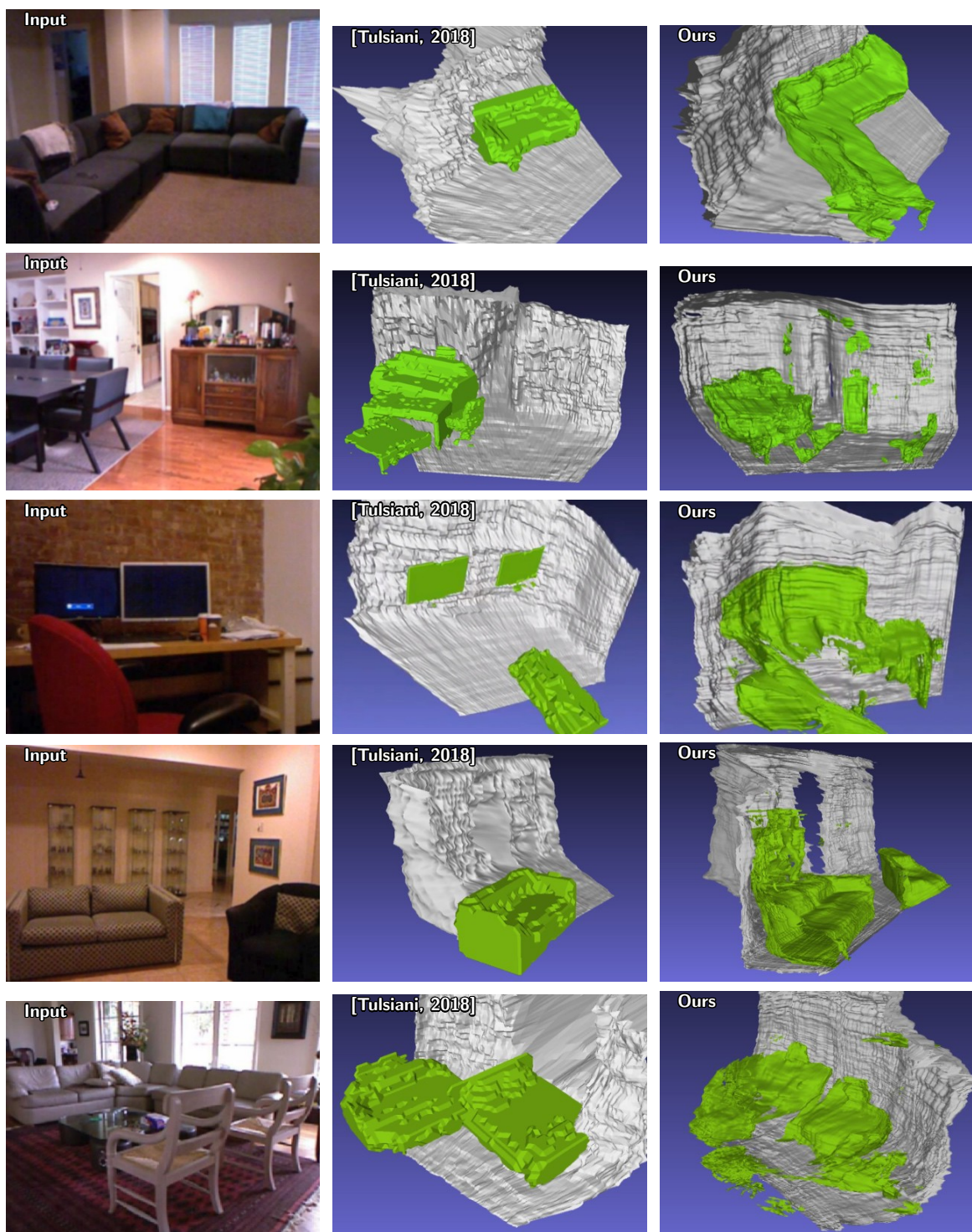


Figure B.7: Evaluation of 3D scene reconstruction on the NYUv2 [69] dataset.

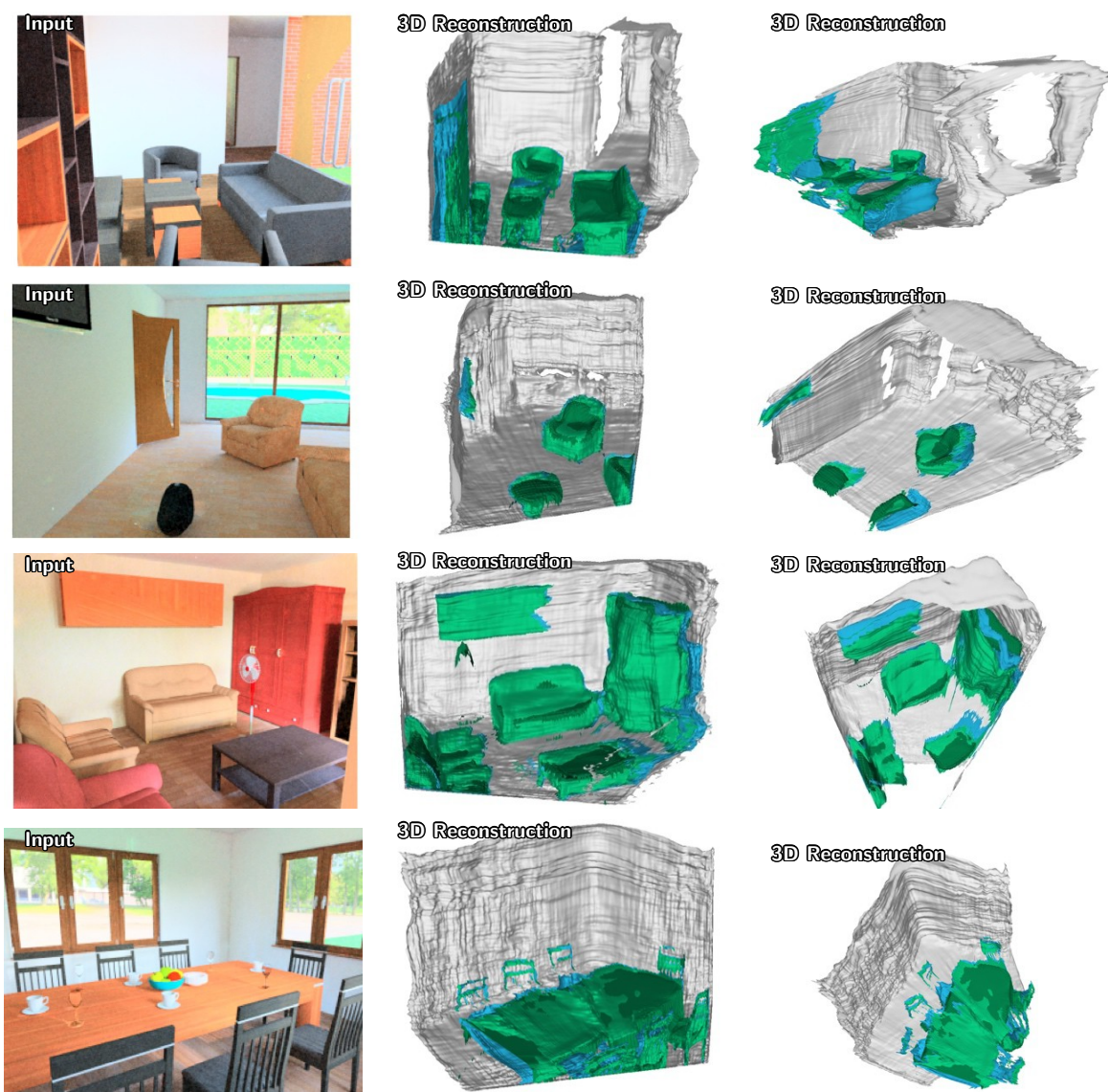
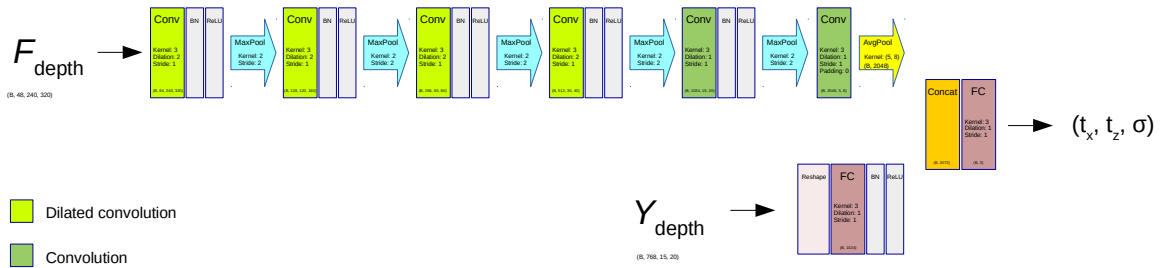
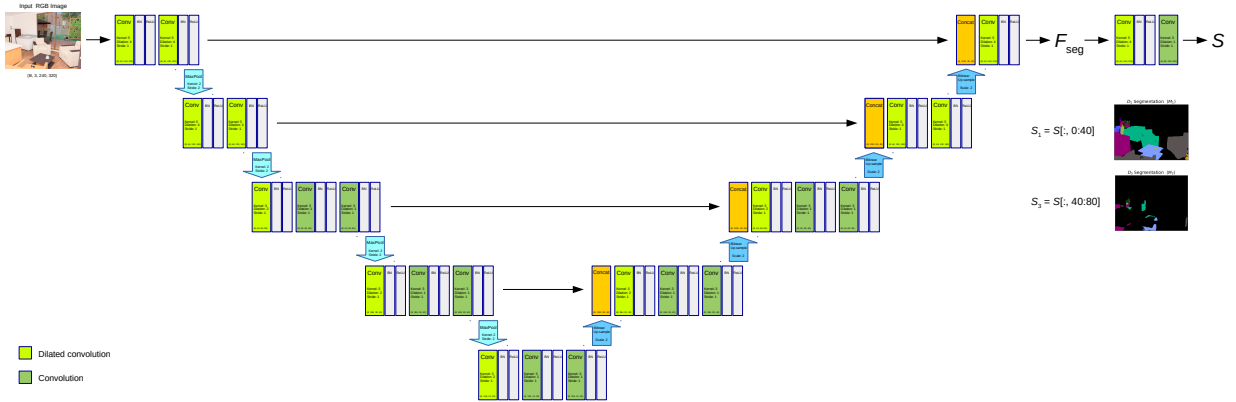
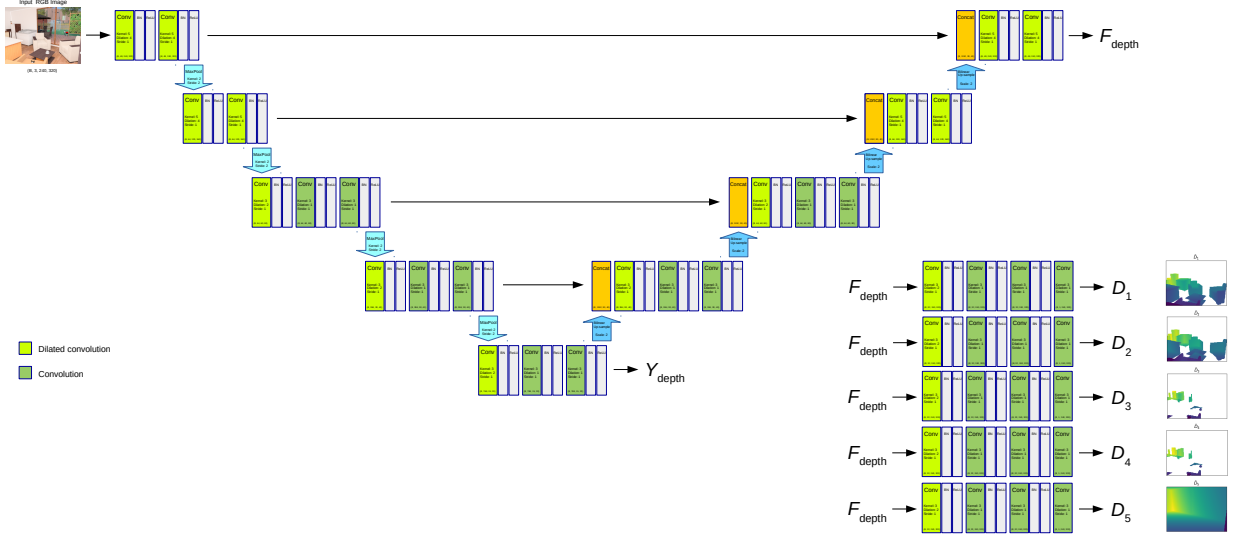


Figure B.8: Evaluation of 3D scene reconstruction on the SUNCG [75] dataset.



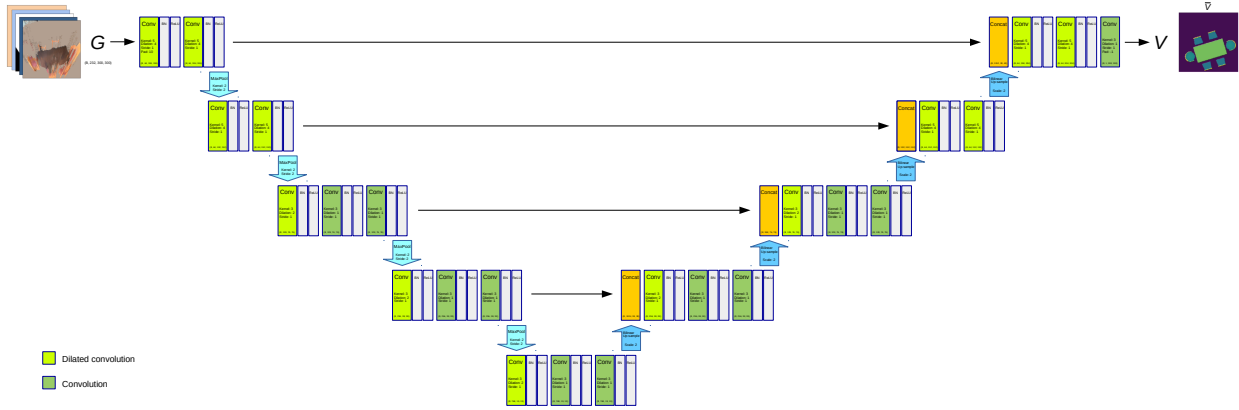


Figure B.12: Network architecture for virtual view surface prediction network. (Best viewed in color and on screen)

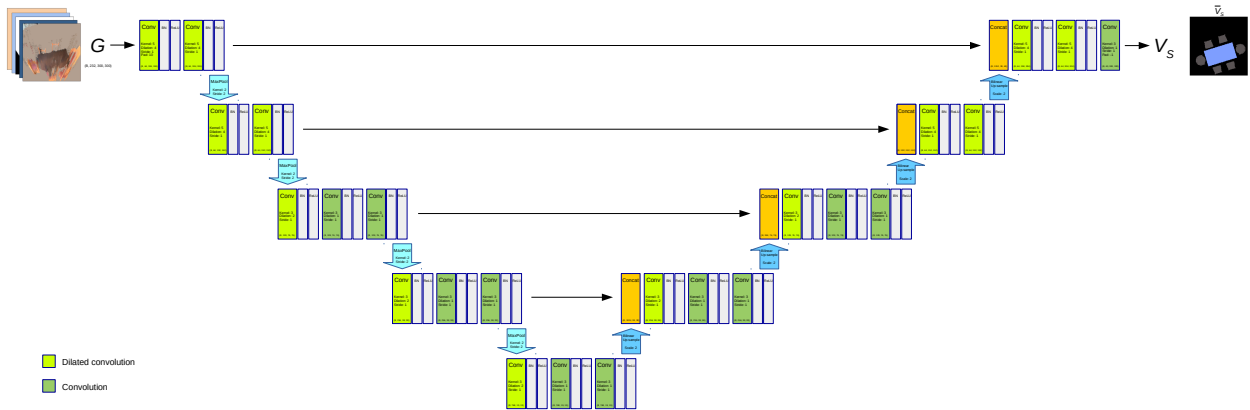


Figure B.13: Network architecture for virtual view segmentation network. (Best viewed in color and on screen)

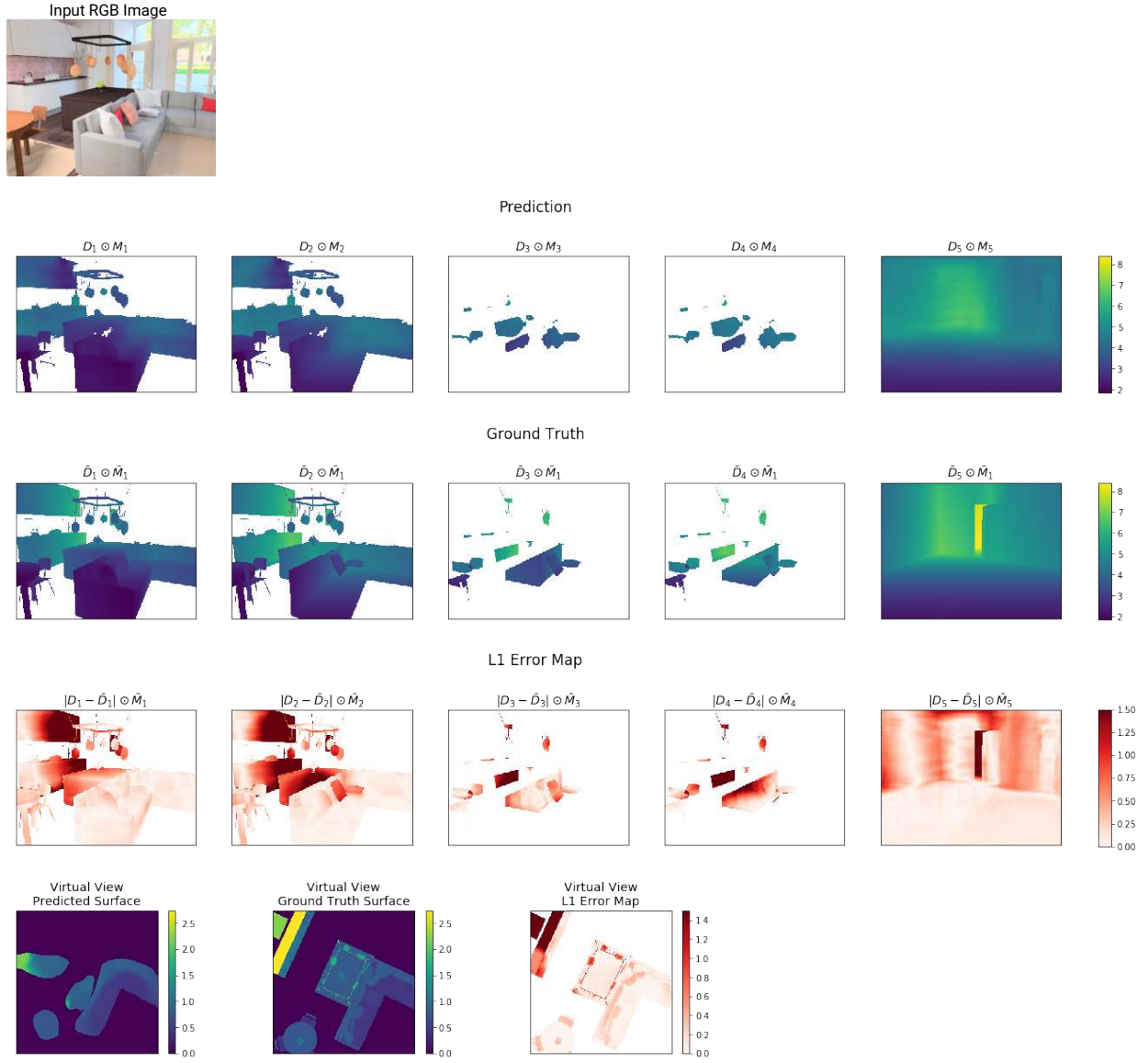


Figure B.14: Multi-layer depth prediction on SUNCG (example 1 of 12).

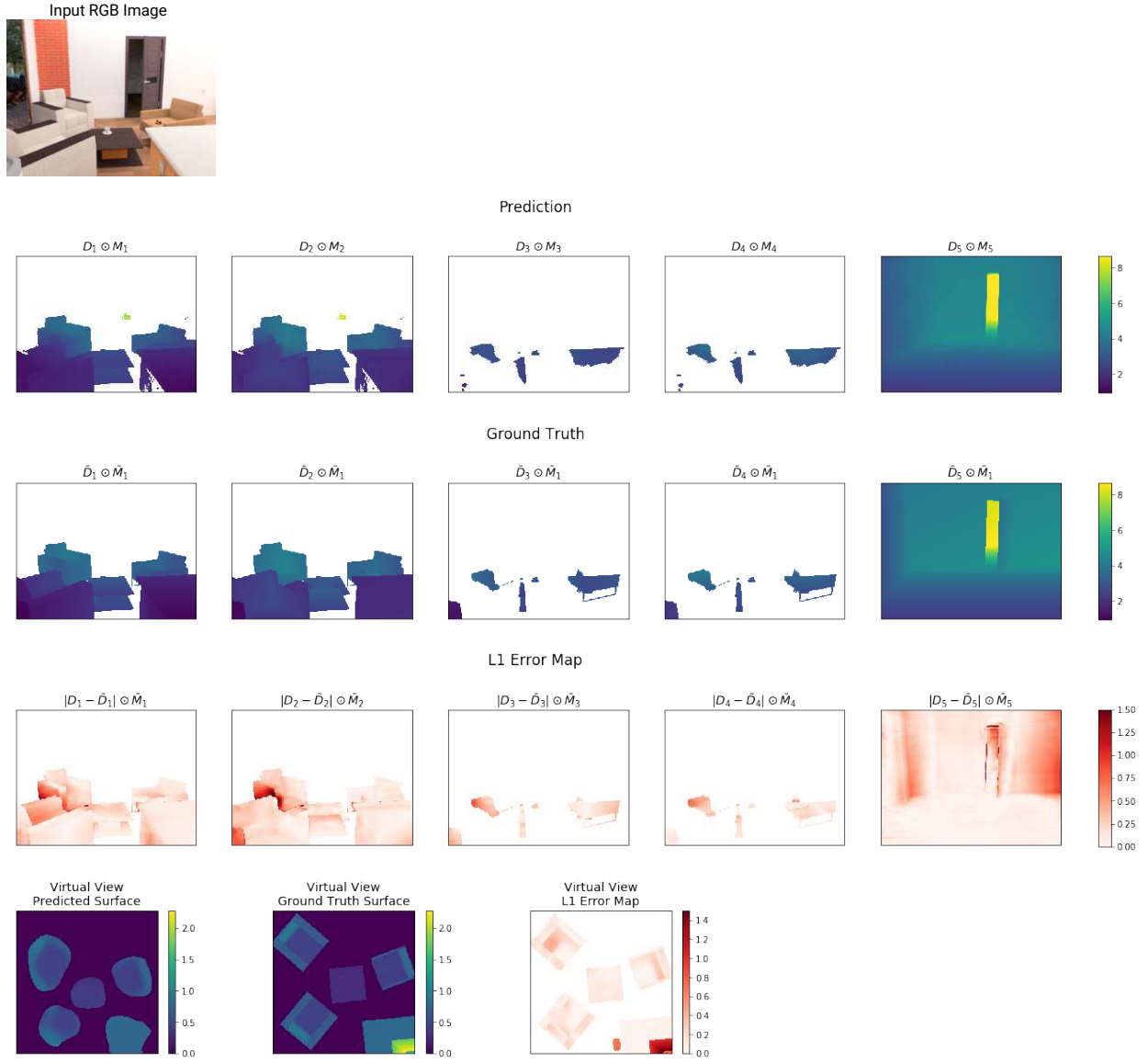


Figure B.15: Multi-layer depth prediction on SUNCG (example 2 of 12).

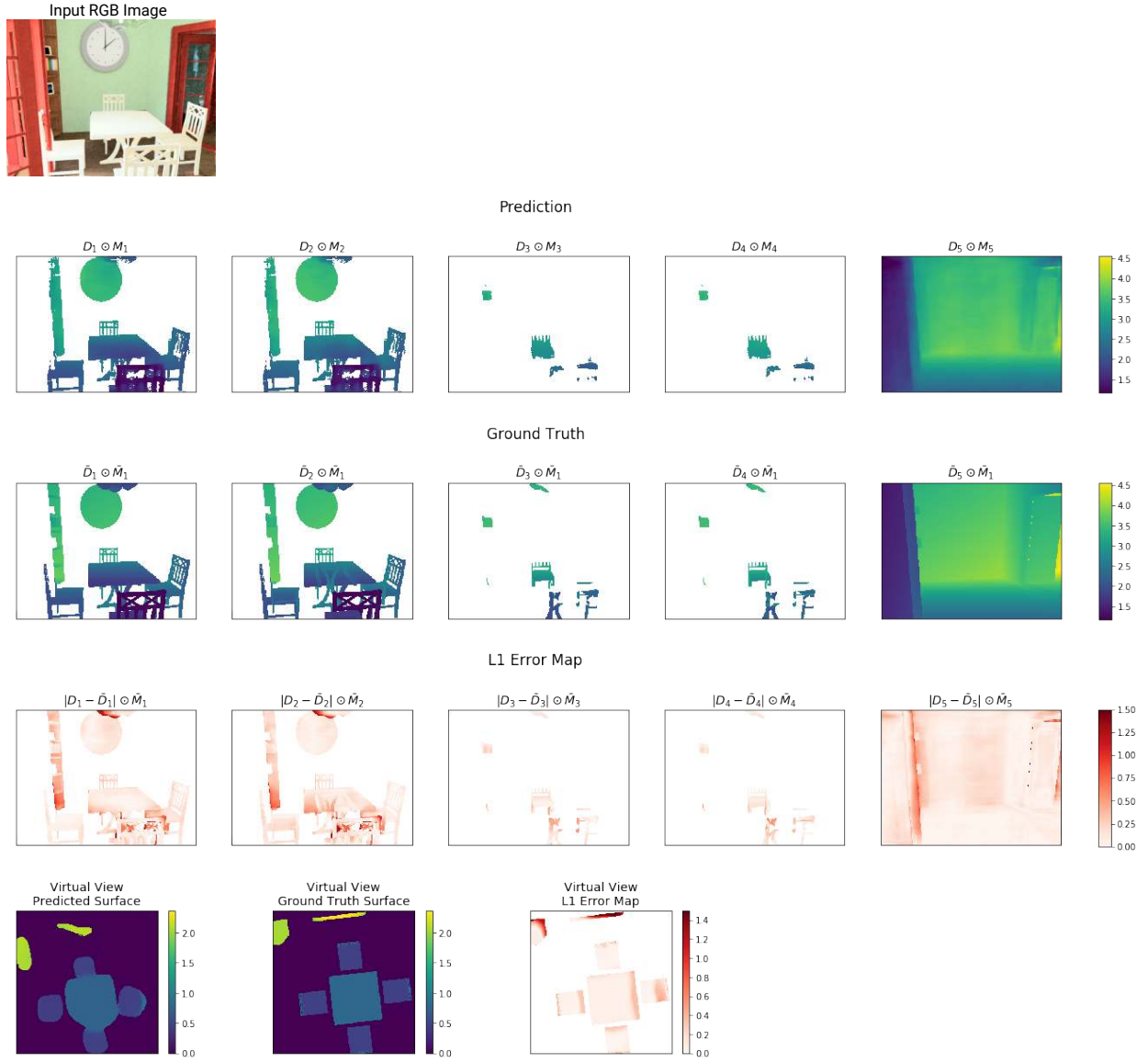


Figure B.16: Multi-layer depth prediction on SUNCG (example 3 of 12).

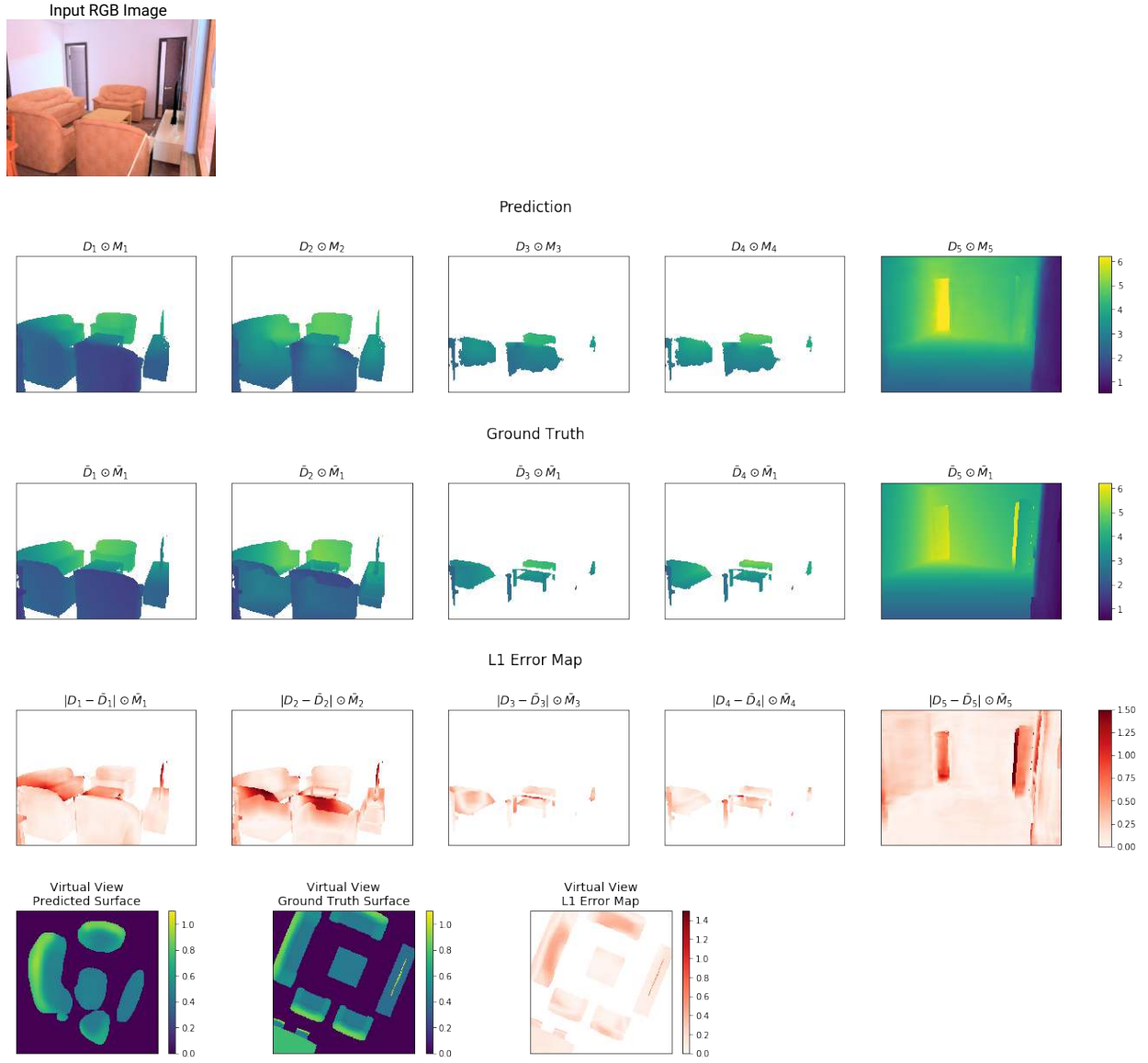


Figure B.17: Multi-layer depth prediction on SUNCG (example 4 of 12).

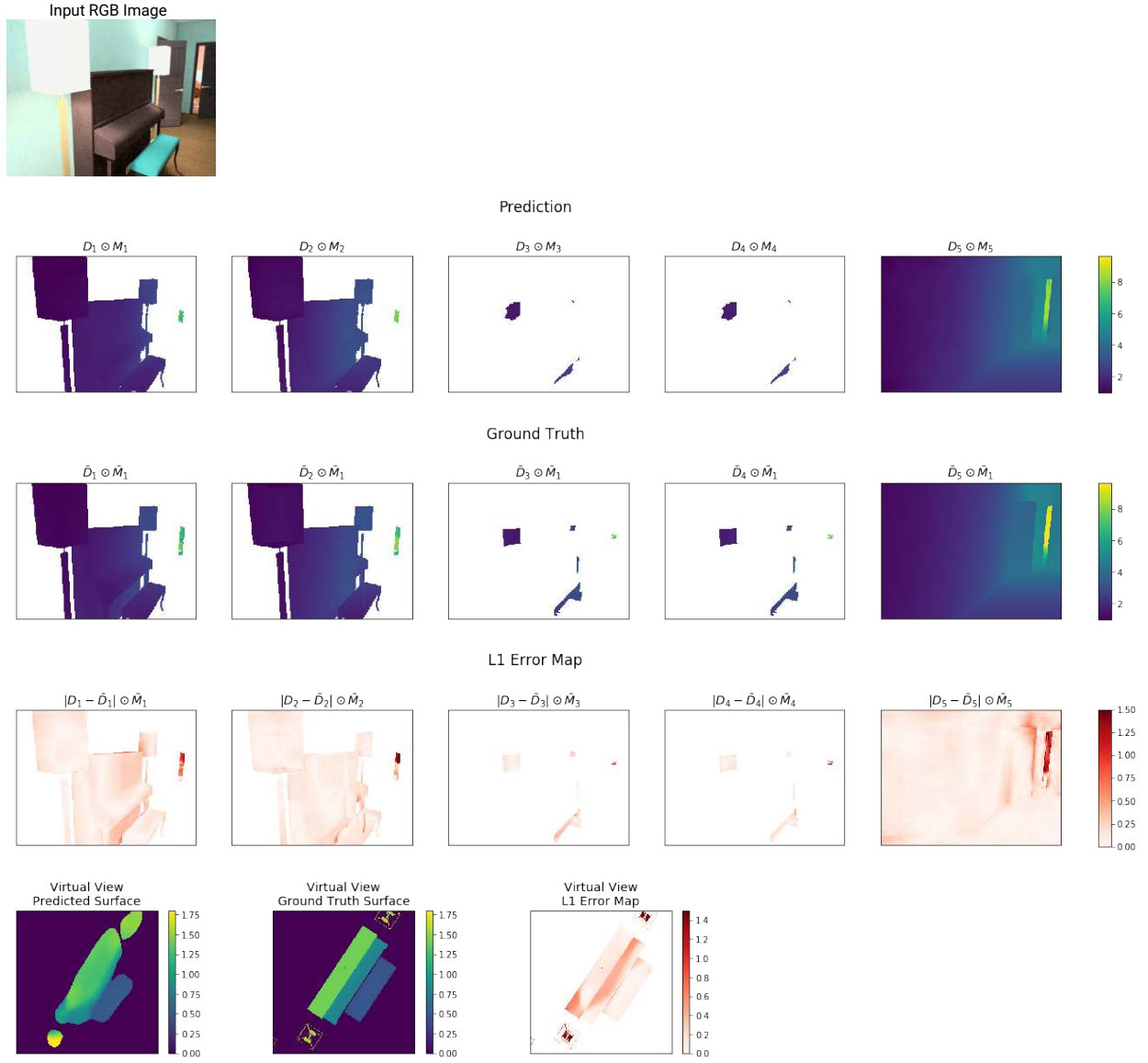


Figure B.18: Multi-layer depth prediction on SUNCG (example 5 of 12).

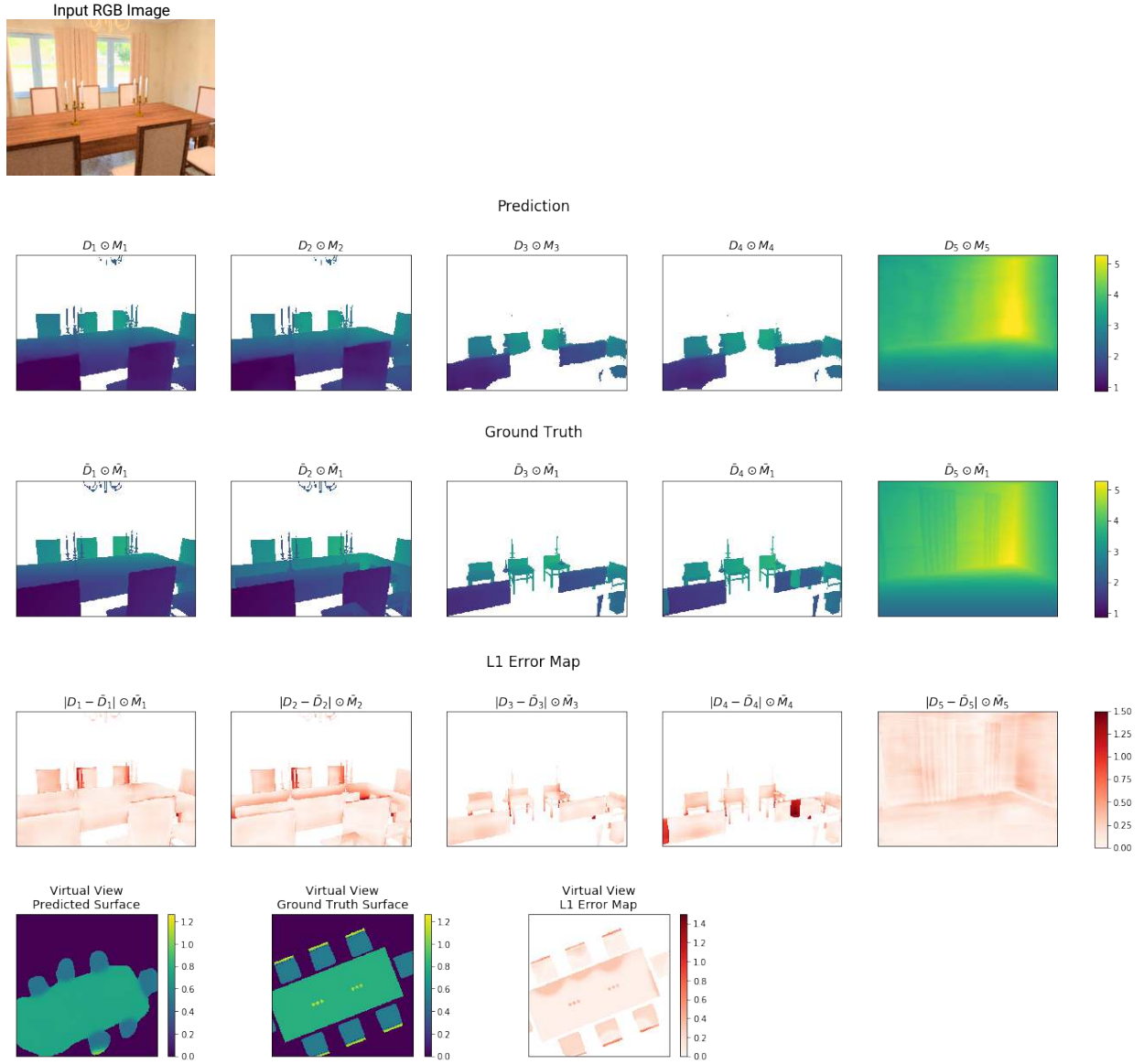


Figure B.19: Multi-layer depth prediction on SUNCG (example 6 of 12).

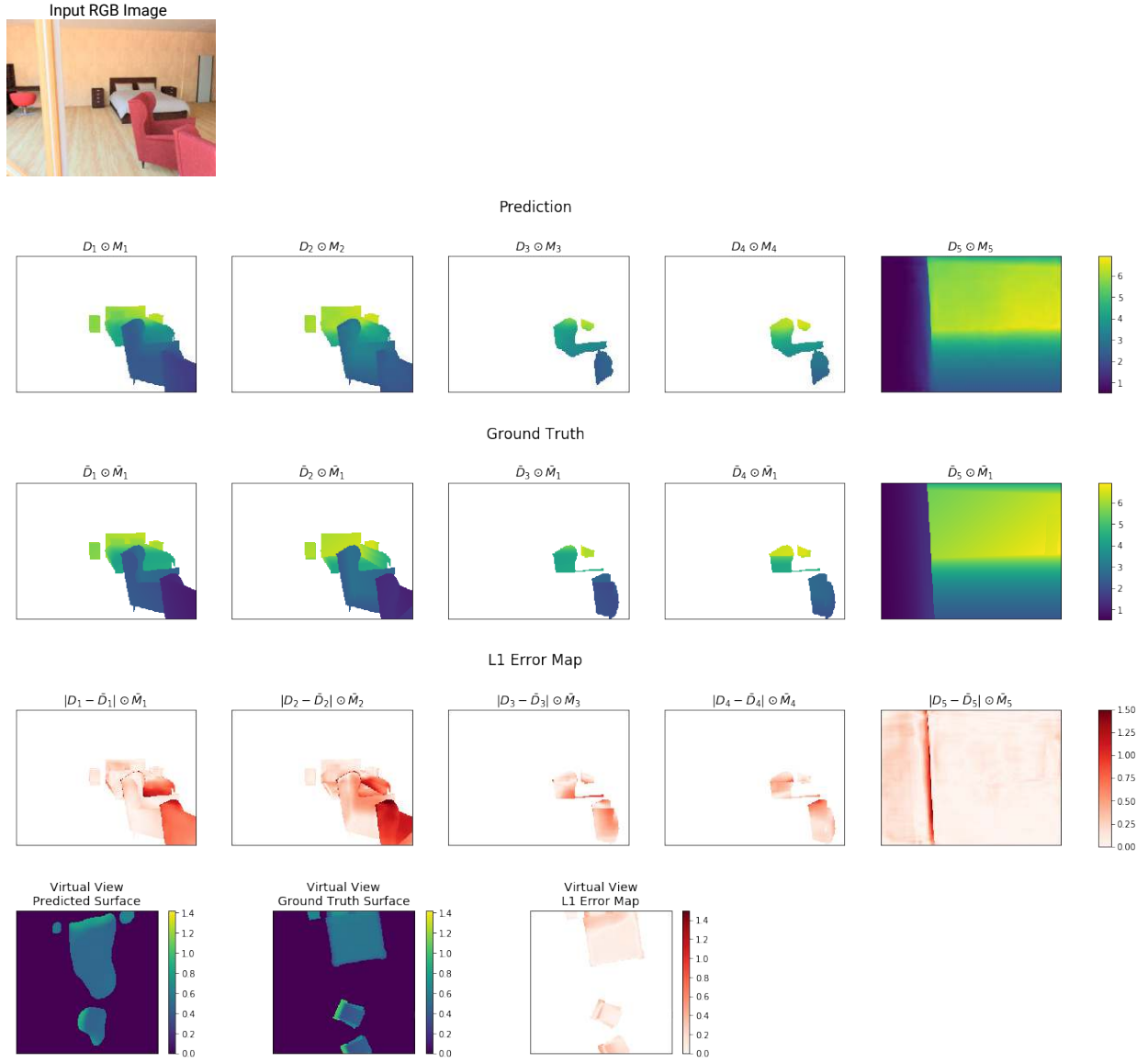


Figure B.20: Multi-layer depth prediction on SUNCG (example 7 of 12).

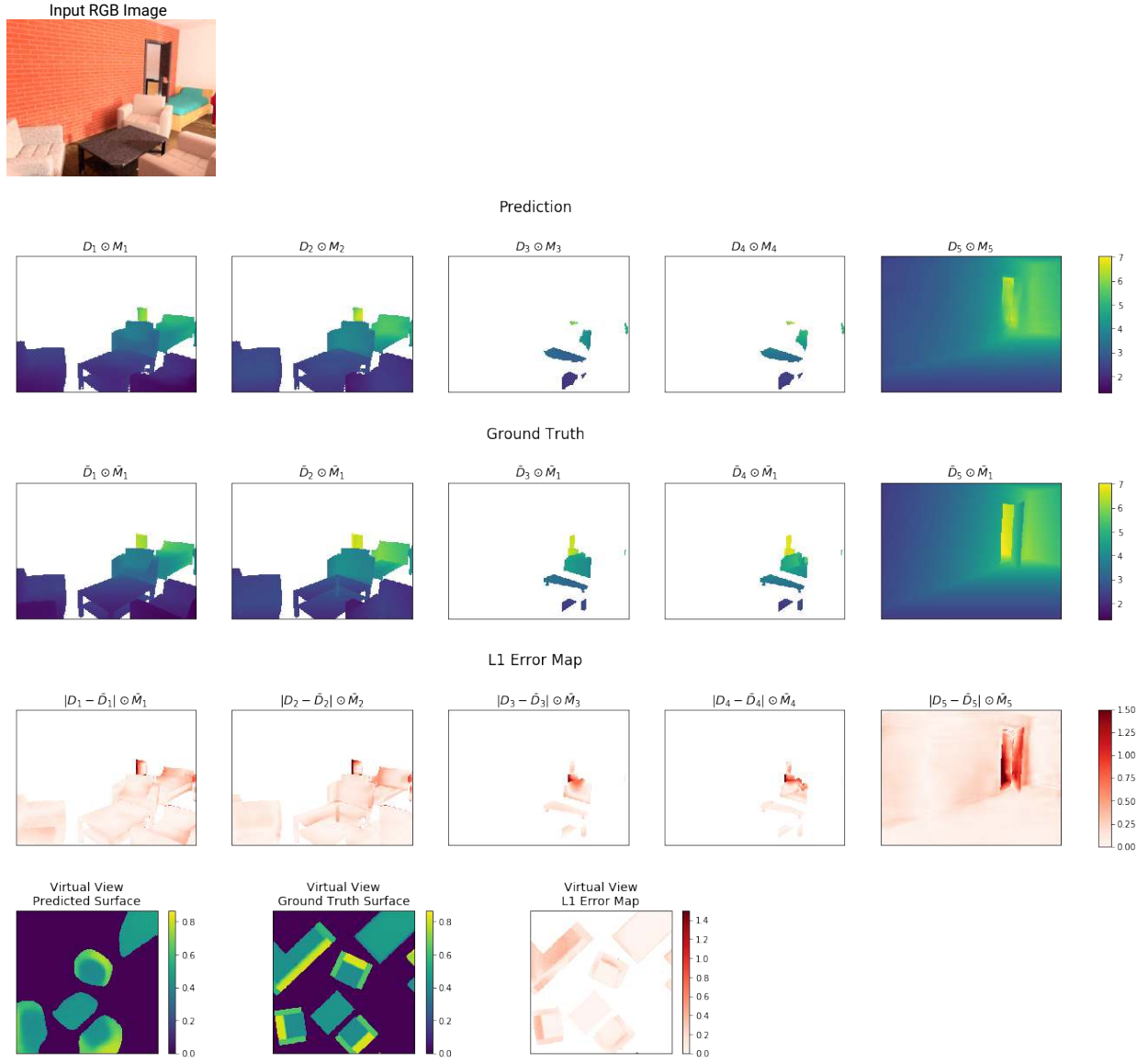


Figure B.21: Multi-layer depth prediction on SUNCG (example 8 of 12).

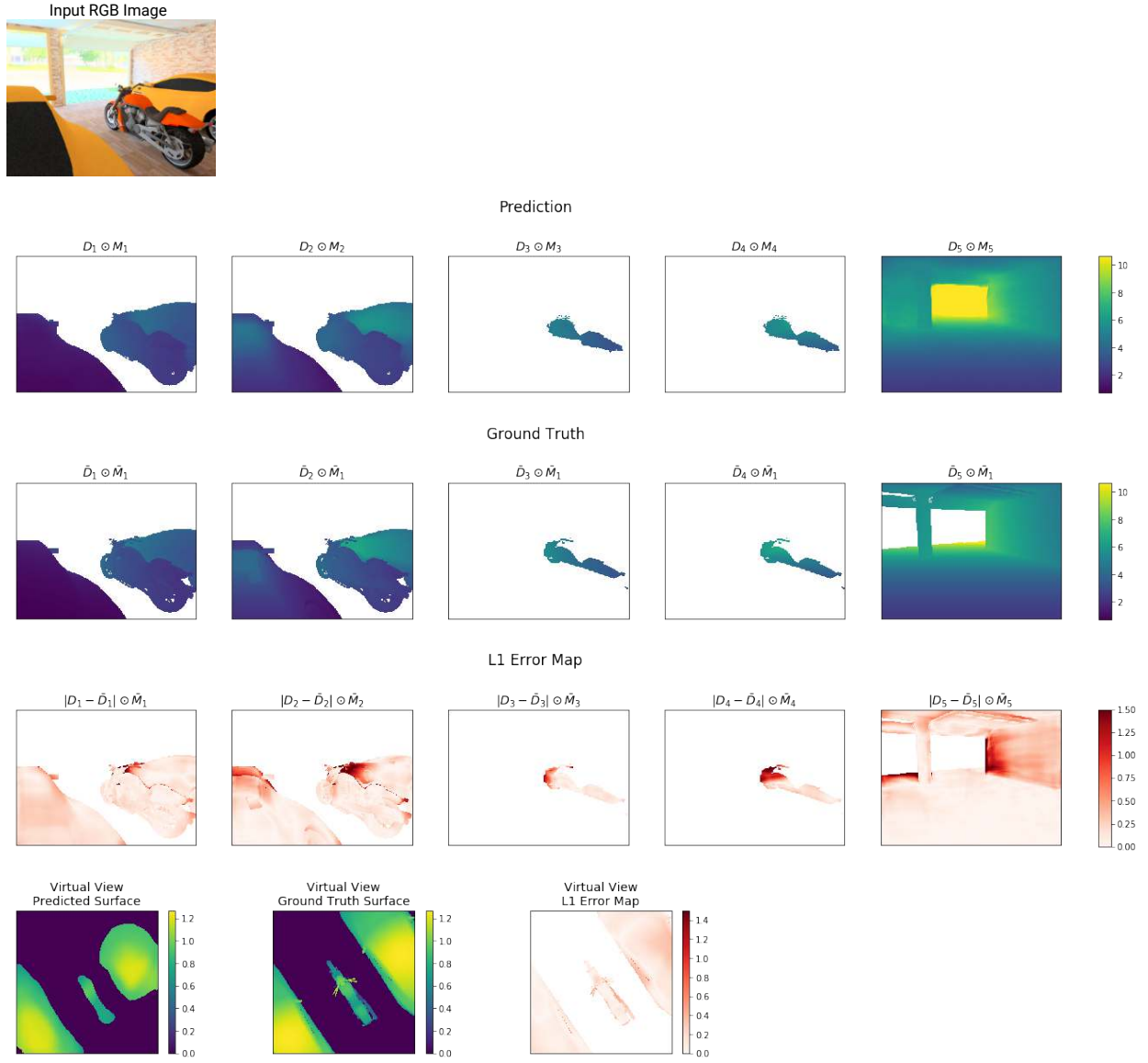


Figure B.22: Multi-layer depth prediction on SUNCG (example 9 of 12).

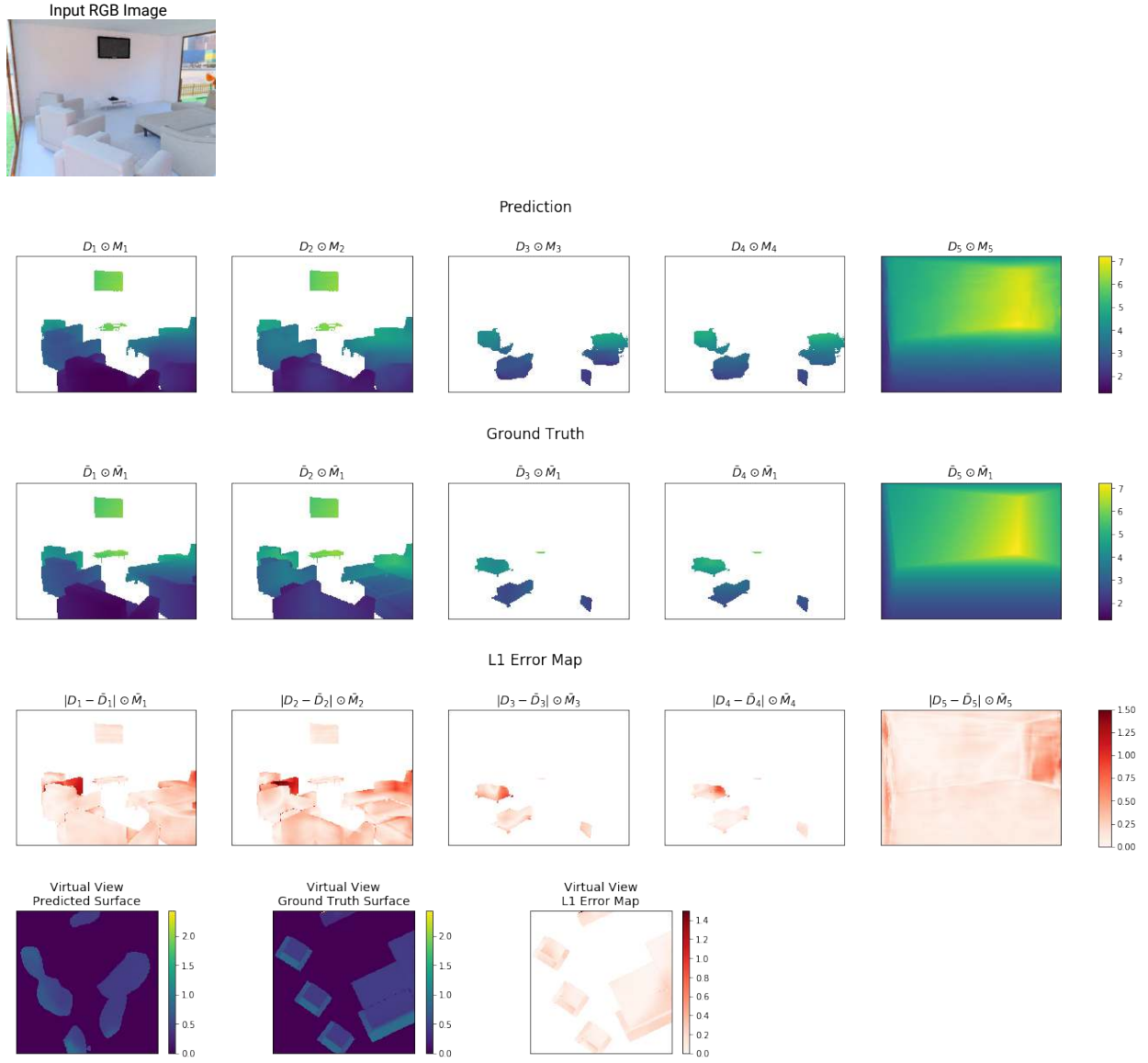


Figure B.23: Multi-layer depth prediction on SUNCG (example 10 of 12).

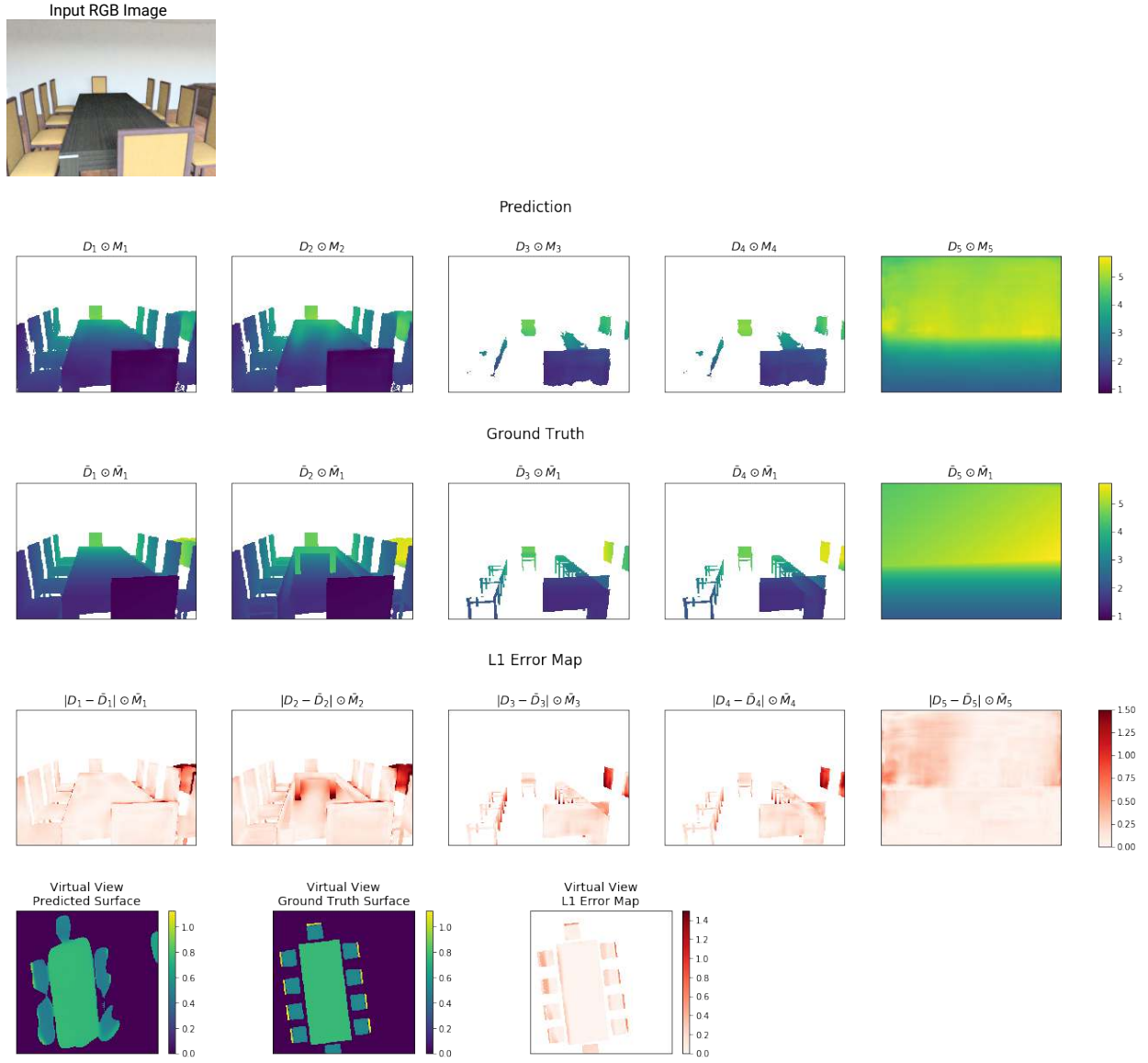


Figure B.24: Multi-layer depth prediction on SUNCG (example 11 of 12).

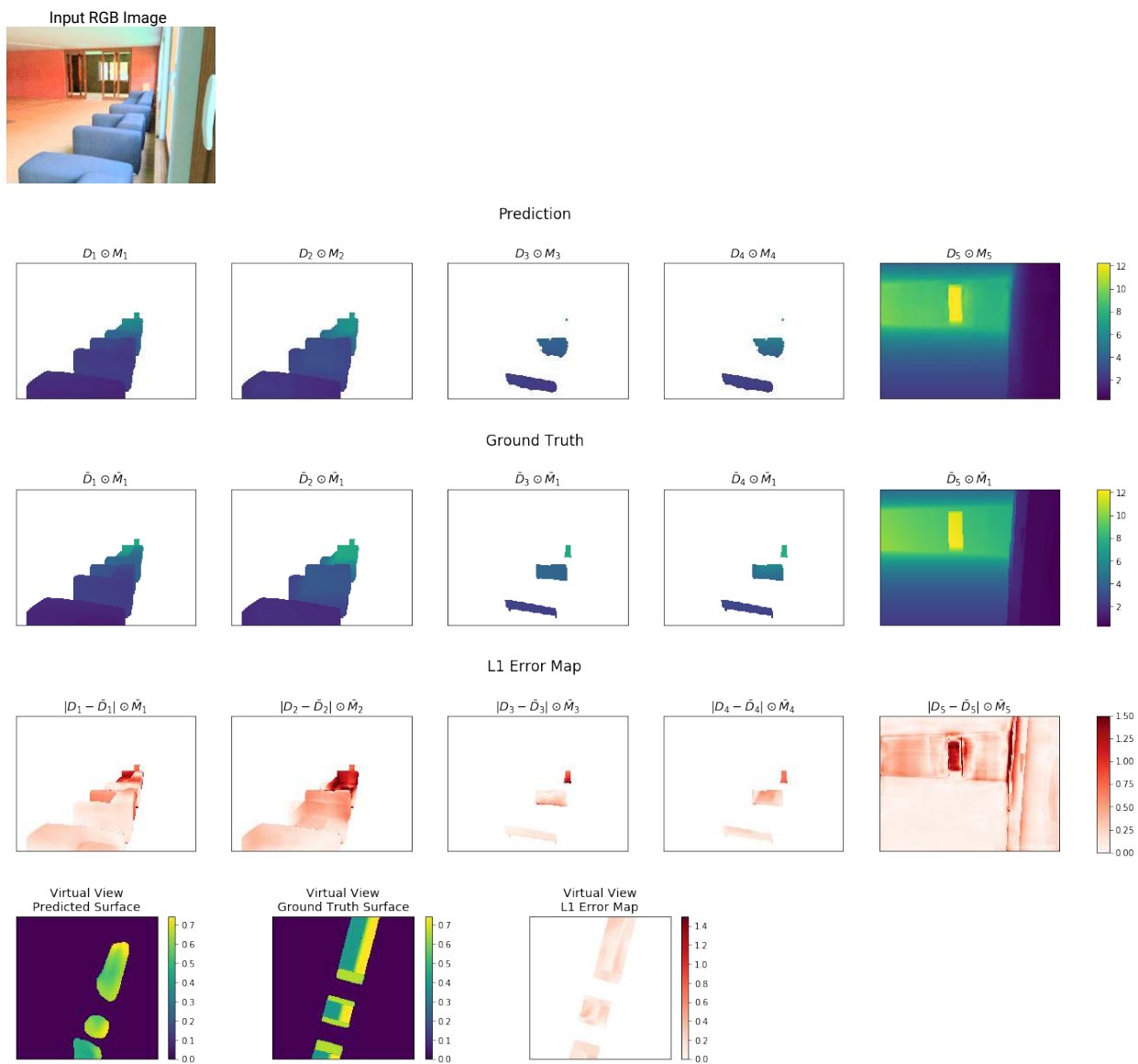


Figure B.25: Multi-layer depth prediction on SUNCG (example 12 of 12).