Unsupervised Learning for 3D Reconstruction and Blocks World Representation

Tejas Khot
June, 2019

The Robotics Institute
Carnegie Mellon University
Pittsburgh, PA

Thesis Committee:
Martial Hebert, Chair
Abhinav Gupta
Adam Harley

CMU-RI-TR-19-29

Submitted in partial fulfillment of the requirements for the degree of Master of Science in Robotics.

To my family, for their boundless love and support.

Abstract

Recovering the dense 3D structure of a scene from its images has been a long-standing goal in computer vision. Recent years have seen attempts of encoding richer priors into the geometry-based pipelines with the introduction of learning based methods. We argue that the form of 3D supervision required by such methods is too onerous, is not naturally available, and it is therefore of both practical and scientific interest to pursue solutions that do not rely on such 3D supervision.

In this thesis, we attempt to bridge the worlds of geometric modeling and deep learning – how to use geometric constraints for obtaining supervisory signal for the task of reconstructing and representing the 3D world efficiently. We first present an unsupervised learning based approach for 3D reconstruction, based on a novel robust photometric consistency objective, the output of which is a 3D point cloud. When trained with our proposed learning objective, deep multi-view stereo models produce significantly better 3D reconstructions. The proposed objective allows implicitly overcoming lighting changes and occlusions across multiple views.

In order to represent the reconstructions efficiently, we draw inspiration from Larry Roberts' famous Blocks World of 1965. We introduce a deep learning framework that enables representing 3D point clouds as an assembly of blocks giving way to a lightweight representation with a several orders of magnitude reduction in memory. We describe how geometric relationships between points and surfaces along with physical priors can be utilized to provide supervisory signal for training deep models. We also present a synthetic-to-real transfer learning setup with a differentiable matching loss that facilitates supervised learning of such blocks world representations.

Acknowledgments

This thesis is a product of the continuous support and guidance of many people without whom I could not have written a sixty-page research document. I would like to use this opportunity to thank them.

I am deeply indebted to Martial Hebert for being an incredible advisor. Martial taught me the discipline for conducting good research, asking the right questions and not losing sight of the destination while taking baby steps. He allowed me the freedom to play with ideas that interested me and was unequivocally supportive through my stumbles, guiding me just enough to enable me to discover solutions and mature as a problem solver.

This journey was enriched by the company of Wentao Yuan who has been an amazing lab mate and collaborator. The countless deep discussions we had helped shape my research lens.

I am glad that Shubham Tulsiani decided to move to Pittsburgh and collaborate with us. Observing him, I have learned how to crisply and clearly formulate problems and deal with negative results calmly.

I am grateful to Christoph Mertz for graciously providing lab space and funding to support my research. His constructive criticism and sharp remarks helped make all writing that extra bit better. I would also like to thank Lynnetta Miller for patiently handling all our logistics.

The student community at CMU has been a great force in making my stay a fantastic experience and making me a better version of myself. I would like to thank the many people who spent generous hours brainstorming ideas with me.

I am extremely thankful to Divyansh Kaushik in whom I found a great friend. Playing devil's advocate with him and indulging in profound discussions on topics ranging from politics to policy, over food and coffee, has been a highlight of my life at CMU. I would also like to thank Bhavan Jasani for being a good friend and partner in the daily food excursions. I am deeply thankful to Katarzyna (Kate) Olszewska for being the best friend and support system one could ask for. Kate has been instrumental in helping me get going when things got tough, often believing in me more when I had doubts.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

Reconstructing 3D scenes from photographs has been among the earliest recognized goals in Computer Vision, keeping researchers occupied for decades. With applications ranging from mapping and navigation for robotics to 3D printing, augmented and mixed reality, video games, or cultural heritage archival, the interest in this task is only growing. We are now finally at a point where our methods are slowly becoming mature enough to operate in the real world, evolving from the controlled lab environments.

There exists a vast body of work on multi-view stereo reconstruction with increasingly accurate results at scale. The dominant line of attack has relied on leveraging stereopsis, or more specifically epipolar geometry, which expresses geometric relationships between points in 3D, their projections on to 2D images, and the camera positions and configurations. We are now seeing how such techniques can be married with tools from machine learning to produce improved results.

The advent of deep learning has brought immense success to computer vision tasks with an increasing reliance on obtaining direct supervision for training neural networks to yield impressive results. However, this paradigm of supervised learning is not a friendly one for 3D inference as the acquisition, labeling and processing of 3D data is expensive, painstaking and not easily scalable. This is also reflected in the size of 3D datasets which contain orders of magnitude lower number of examples than their 2D counterparts. Going forward, an important obstacle in the way for our learning techniques to scale to more

diverse scenarios is this onerous requirement of supervision. Bypassing the need for explicit 3D supervision can enable 3D learning techniques to enjoy success similar to 2D.

An important aspect of reconstruction is the choice of representation for the produced 3D geometry. The reconstructions that stereo techniques produce are often represented as dense point clouds before processing them to other forms such as meshes or voxel grids. While these are easy to incorporate in the processing pipelines, such dense representations (eg. millions of points in a 3D point cloud) are often an overkill for the task at hand. For many tasks such as building maps of regions, one might prefer a more lightweight representation that can capture the underlying shapes of structures to a certain degree of approximation without enforcing fine granularity. This is akin to human reasoning which relies on exploiting repeating structures in visual patterns to obtain compositional representations — eg. a chair is made of 4 legs which resemble 4 vertical blocks.

This thesis addresses the challenges regarding learning based 3D reconstruction and obtaining sufficient representations of scenes while relaxing the requirement for having explicit 3D supervision.

1.2 Overview of Contributions

This thesis offers the following contributions to the research community:

1. A framework to learn multi-view stereopsis in an unsupervised manner, using only images from novel views as supervisory signal.
2. A robust multi-view photometric consistency loss for learning unsupervised depth prediction that allows implicitly overcoming lighting changes and occlusions across training views.
3. Demonstration on how the proposed learning scheme can be used to fine-tune other supervised learning models in addition to being a standalone trained model that generalizes well to new datasets.
4. An unsupervised learning framework for learning compositional representations of 3D shapes by encoding 3D point clouds as collections of geometric primitive shapes by posing this as a primitive fitting task.
5. A technique for programmatically generating a synthetic dataset of man-made buildings composed of simple parameterized shapes that can be used for supervised

learning of primitive fitting models.

6. A synthetic-to-real transfer learning formulation for primitive fitting, specific to the case of man-made buildings, that includes a differentiable matching loss utilizing the Hungarian optimal assignment algorithm.

1.3 Thesis Outline

This thesis will be organized as follows. In Chapter 2, we introduce an unsupervised learning scheme which enables training of deep multi-view stereo models resulting in 3D reconstructions by implicitly overcoming lighting changes and occlusions across multiple views. We show quantitative and qualitative results on multiple real datasets along with extensive ablation studies. In Chapter 3, we present a learning based model for representing dense 3D point clouds as collections of primitive shapes leading to a compact and sufficient representation of shapes. We describe methods to learn such a model with and without supervision and show results on synthetic and real data. In Chapter 4, we provide concluding remarks and highlight some important open problems for future work.

Chapter 2

Unsupervised Multi-View Stereopsis

In this chapter, we present a learning based approach for multi-view stereopsis (MVS). While current deep MVS methods achieve impressive results, they crucially rely on ground truth 3D training data, and acquisition of such precise 3D geometry for supervision is a major hurdle. Our framework instead leverages photometric consistency between multiple views as supervisory signal for learning depth prediction in a wide baseline MVS setup. However, naively applying photo consistency constraints is undesirable due to occlusion and lighting changes across views. To overcome this, we propose a robust loss formulation that: a) enforces first order consistency and b) for each point, selectively enforces consistency with some views, thus implicitly handling occlusions. We demonstrate our ability to learn MVS without 3D supervision using a real dataset, and show that each component of our proposed robust loss results in a significant improvement. We qualitatively observe that our reconstructions are often more complete than the acquired ground truth, further showing the merits of this approach. Lastly, our learned model generalizes to novel settings, and our approach allows adaptation of existing CNNs to datasets without ground-truth 3D by unsupervised finetuning.

2.1 Motivation

Recovering the dense 3D structure of a scene from its images has been a long-standing goal in computer vision. Several approaches over the years have tackled this multi-view stereopsis (MVS) task by leveraging the underlying geometric and photometric constraints

– a point in one image projects on to another along the epipolar line, and the correct match is photometrically consistent. While operationalizing this insight has led to remarkable successes, these purely geometry based methods reason about each scene independently, and are unable to implicitly capture and leverage generic priors about the world *e.g.,* surfaces tend to be flat, and therefore sometimes perform poorly when signal is sparse *e.g.,* textureless surfaces.

To overcome these limitations, an emergent line of work has focused on learning based solutions for the MVS task, typically training CNNs to extract and incorporate information across views. While these methods yield impressive performance, they crucially rely on ground-truth 3D data during the learning phase. We argue that this form of supervision is too onerous, is not naturally available, and it is therefore of both practical and scientific interest to pursue solutions that do not rely on such 3D supervision.

We build upon these recent learning-based MVS approaches that present CNN architectures with geometric inductive biases, but with salient differences in the form of supervision used to train these CNNs. Instead of relying on ground-truth 3D supervision, we present a framework for learning multi-view stereopsis in an *unsupervised* manner, relying only on a training dataset of multi-view images. Our insight that enables the use of this form of supervision is akin to the one used in classical methods – that the correct geometry would yield photometrically consistent reprojections, and we can therefore train our CNN by minimizing the reprojection error.

While similar reprojection losses have been successfully used by recent approaches for other tasks *e.g.,* monocular depth estimation, we note that naively applying them for learning MVS is not sufficient. This is because different available images may capture different visible aspects of the scene. A particular point (pixel) therefore need not be photometrically consistent with all other views, but rather only those where it is not occluded. Reasoning about occlusion explicitly to recover geometry, however, presents a chicken-and-egg problem, as estimates of occlusion depend on geometry and vice-versa. To circumvent this, we note that while a correct estimate of geometry need not imply photometric consistency with all views, it should imply consistency with at least *some* views. Further, the lighting changes across views in an MVS setup are also significant, thereby making enforcing consistency only in pixel space undesirable, and our insight is to additionally enforce gradient-based consistency. We present a robust reprojection loss that enables us to capture these two insights, and allow learning MVS with the desired form

of supervision. Our simple, intuitive formulation allows handling occlusions without ever explicitly modeling them. Our setup and sample outputs are depicted in Figure 2.1. Our model, trained without 3D supervision, takes a collection of images as input and predicts per-image depth maps, which are then combined to obtain a dense 3D model.

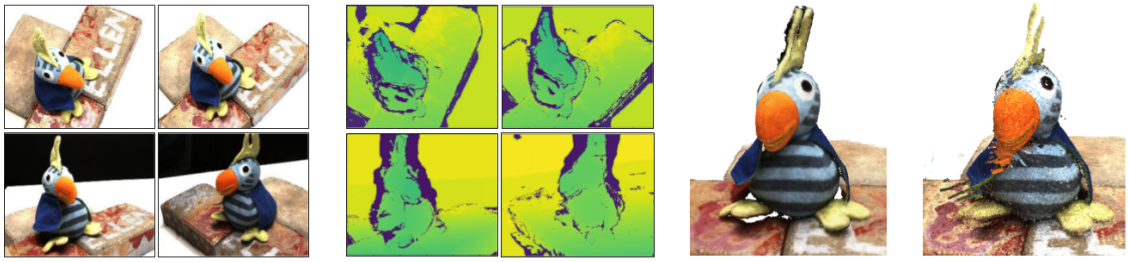

Figure 2.1: Our model consumes a collection of calibrated images of a scene from multiple views and produces depth maps for every such view. We show that this depth prediction model can be trained in an unsupervised manner using our robust photo consistency loss. The predicted depth maps are then fused together into a consistent 3D reconstruction which closely resembles and often improves upon the sensor scanned model. Left to Right: Input images, predicted depth maps, our fused 3D reconstruction, ground truth 3D scan.

2.2 Related Work

Multi-view Stereo Reconstruction. There is a long and rich history of work on MVS. We only discuss representative works here and refer the interested readers to [11, 46] for excellent surveys. There are four main stages in an MVS pipeline: view selection, propagation scheme, patch matching and depth map fusion. Schemes for aggregating multiple views for each pixel have been studied in [12, 16, 28, 28, 43, 58], and our formulation can be seen as integrating some of these ideas via a loss function during training.

The seminal work of PatchMatch[4] based stereo matching replaced the classical seed-and-expand[10, 16] propagation schemes. PatchMatch has since been used for multi-view stereo[12, 43, 58] in combination with iterative evidence propagation schemes, estimation of depth and normals. Depth map fusion[22, 25, 43, 47, 54] combines individual depth maps into a single point cloud while ensuring the resulting points are consistent among multiple views and incorrect estimates are removed. Depth representations continue to dominate MVS benchmarks [2, 45] and methods seeking depth images as output thus

decouple the MVS problem into more tractable pieces.

Learning based MVS. The robustness of features learned using CNNs makes them a natural fit for the third step of MVS: matching image patches. CNN features have been used for stereo matching [19, 55] while simultaneously using metric learning to define the notion of similarity [18]. These approaches require a series of post-processing steps [20] to finally produce pairwise disparity maps. There are relatively fewer works that focus on learning all steps of the MVS pipeline. Volumetric representations encode surface visibility from different views naturally which has been demonstrated in [27, 29]. These methods suffer from the common drawbacks of this choice of representation making it unclear how they can be scaled to more diverse and large-scale scenes. In [31], a cost volume is created using CNN features and disparity values are obtained by regression using a differentiable soft argmin operation. Combining the merits of above methods and borrowing insights from classical approaches, recent works [23, 50, 53] produce depth images for multiple views and fuse them to obtain a 3D reconstruction. Crucially, all of the above methods have relied on access to 3D supervision and our work relaxes this requirement.

Unsupervised depth estimation. With a similar motivation of reducing the requirement of supervision, several recent monocular [13, 15, 38] or binocular stereo based [59] depth prediction methods have leveraged photometric consistency losses. As supervision signal, these rely on images from stereo pairs [13, 15, 38] or monocular videos [52, 60] during training. As means for visibility reasoning, the network is made to predict an explainability [60], invalidation [57] mask or by incorporating a probabilistic model of observation confidence [35]. These methods operate on a narrow baseline setup with limited visual variations between frames used during training, and therefore do not suffer significantly due to occlusions and lighting changes. As we aim to leverage photometric losses for learning in an MVS setup, we require a robust formulation that can handle these challenges.

2.3 Approach

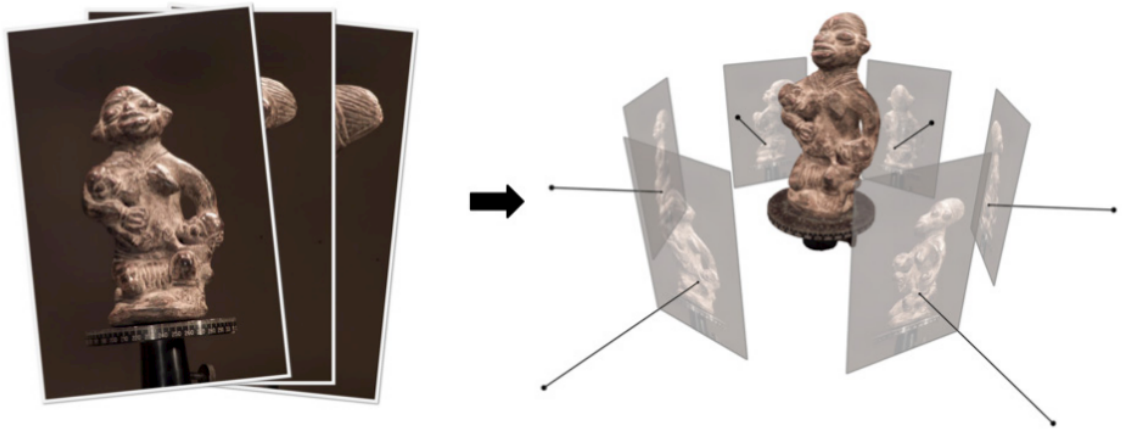

Figure 2.2: The multi-view stereo setup. Given a set of images of a scene (left), the goal of multi-view image based 3D reconstruction is to estimate the most likely 3D shape that explains those images under the assumptions of known viewpoints. Image taken from [11].

The goal in the MVS setup is to reconstruct the dense 3D structure of a scene given a set of input images, where the associated intrinsics and extrinsics for these views are known – these parameters can typically be estimated via a preceding Structure-from-Motion (Sfm) step (see Figure 2.2). While there are several formulations of the MVS problem focused on different 3D representations [10, 11, 29], we focus here on depth-based MVS setup. We therefore infer the per-pixel depth map associated with each input, and the dense 3D scene is then obtained via back-projecting these depth maps into a combined point cloud.

We leverage a learning based system for the step of predicting a depth map, and learn a CNN that takes as input an image with associated neighboring views, and predicts a per-pixel depth map for the central image. Unlike previous learning based MVS methods which also adopt a similar methodology, we only rely on the available multi-view images as supervisory signal, but do not require a ground-truth 3D scene. Towards leveraging this supervision, we build upon insights from classical methods, and note that the accurate geometry prediction for a point (image pixel) should yield photometrically consistent predictions when projected onto other views. We operationalize this insight and use a photometric consistency loss to train our depth prediction CNN, penalizing discrepancy between pixel intensities in original and available novel views. However, we note that

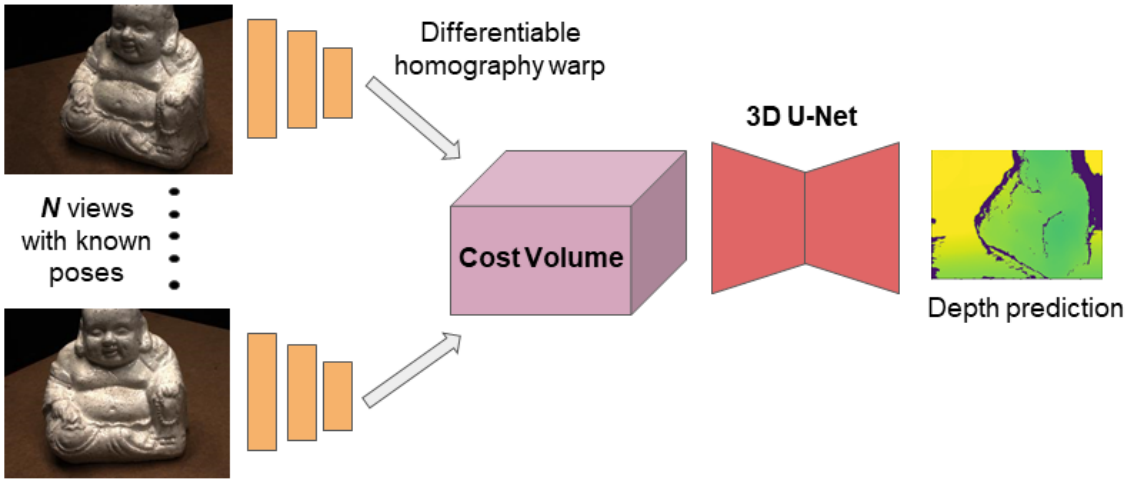

Figure 2.3: **Overview of our network.** We take as input N images of a scene. Image features are generated using a CNN. Using differentiable homography, a cost volume is constructed by warping image features over a range of depth values. The cost volume is then refined using a 3D U-Net style CNN. The final output is a depth map at a downsized resolution. Details of the network architecture can be found in the supplemental.

the assumption of photometric consistency is not always true. The same point is not necessarily visible across all views. Additionally, lighting changes across views would lead to further discrepancy between pixel intensities. To account for possible lighting changes, we add a first-order consistency term in the photometric loss and therefore also ensure that gradients match in addition to intensities. We then implicitly deal with possible occlusions by proposing a robust photometric loss, which enforces that a point should be consistent with *some*, but not necessarily all views.

We describe the architecture of the CNN used to infer depth in Section 2.3.1, and present in Section 2.3.2 the vanilla version of photometric loss that can be used to learn this CNN in an unsupervised manner. We then present our robust photometric loss in Section 2.3.3, and describe the overall learning setup, additional priors and implementation details in Section 2.3.5. While we primarily focus on the learning of the depth prediction CNN in this section, we briefly summarize how the learned CNN is integrated in a standard MVS setup at inference in Section 2.3.6.

2.3.1 Network Architecture

The unsupervised learning framework we propose is agnostic to network architecture. Here, we adopt the model proposed in [53] as a representative network architecture while noting that similar architectures have also been proposed in [23, 50]. The network takes as input N images, extracts features using a CNN, creates a plane-sweep based cost volume and infers a depth map for every reference image. A sketch of the architecture is given in Figure 2.3.

Every input image is first passed through a feature extraction network having shared weights for all images. For this network, we use use an 8-layer CNN having batch-normalization and ReLU after every convolution operation till the penultimate layer. The last layer produces a 32 channel downsized feature map for every image. Using the differentiable homography formulation, the feature maps are warped into different fronto-parallel planes of the reference camera at 128 depth values to form one cost volume per non-reference image. The homography between the i^{th} feature map and the reference image feature map at a candidate depth d is given by $H_i(d)$ as below.

$$\mathbf{H}_i(d) = \mathbf{K}_i \cdot \mathbf{R}_i \cdot \left(\mathbf{I} - \frac{(\mathbf{t}_{ref} - \mathbf{t}_i) \cdot \mathbf{n}_{ref}^T}{d}\right) \cdot \mathbf{R}_{ref}^T \cdot \mathbf{K}_{ref}^T. \tag{2.1}$$

Here K, R, t, n represent the camera intrinsics, rotations, translations and principle axis of the reference camera respectively.

All such cost volumes are aggregated into a single volume using a variance-based cost metric. Note that MVSNet uses 256 depth values for the cost volume during training. In our experiments, we find the model trained with coarser depth resolution to work better, which is potentially due to insufficient regularization when allowed to predict in high depth ranges. This reduction does play a role in the output reconstruction quality, but we leave this optimization for future work since our contributions hold nonetheless.

In order to refine the cost volume and incorporate smooth variations of the depth values, we use a three layer 3D U-Net. An initial estimate of the predicted depth map can be obtained by performing a soft $argmin$ operation along the depth channel. Unlike the $winner - take - all$ approach which requires the non-differentiable $argmax$ operation, such a soft aggregation of volumes allows for sub-pixel accuracies while being amenable to training due to its differentiability. Thus, in spite of the discretization of depth value for

constructing the cost volume, the resulting depth map follows a continuous distribution. Thus, the expected value along the depth direction is obtained by the weighted sum (by probability) over all hypotheses and can be computed as:

$$\mathbf{D} = \sum_{d=d_{min}}^{d_{max}} d \times \mathbf{P}(d) \tag{2.2}$$

Here, $\mathbf{P}$ is the pixel-wise $winner - take - all$ probability volume and $\mathbf{P}(d)$ represents the probability estimations for all pixels at a candidate depth d.

The resulting probability distribution which the output volume represents is likely to be containing outliers and would not necessarily contain a single peak. To account for this, a notion of depth estimate quality is established wherein the quality of estimate at any pixel is defined to be the sum of the probabilities over the four nearest depth hypotheses. This estimate is then filtered at a threshold of 0.8 and applied as a mask to the output volume. The predicted depth map is then concatenated to the reference image and passed through a four-layer CNN to output a depth residual map. The final depth map is obtained by adding the residual map to the initial estimated depth map. For complete details of the hyperparameters, we refer the reader to the MVSNet[53] paper and it's corresponding supplemental.

The emphasis of our work is on a way to train such CNNs in an unsupervised manner using a robust photometric loss, as described in the following sections.

2.3.2 Learning via Photometric Consistency

We now describe how our depth prediction network can be trained effectively without requiring ground truth depth maps. The central idea is to use a warping-based view synthesis loss, that has been quite effective in the stereo and monocular depth prediction tasks [39, 60] though hasn't been explored for unstructured multi-view scenarios. Given an input image I_s, and additional neighboring views, our CNN outputs a depth map D_s. During training, we also have access to M additional novel views of the same scene $\{I_v^m\}$, and use these to supervise the predicted depth D_s.

For a particular pair of views (I_s, I_v^m) with associated intrinsic/relative extrinsic (K, T) parameters, the predicted depth map D_s allows us to "inverse-warp" the novel view to the source frame using a spatial transformer network [24] followed by differentiable bilinear

sampling to yield $\hat{I}_s^i$. For a pixel u in the source image I_s, we can obtain its coordinate in the novel view with the warp:

$$\hat{u} = K\ T(D_s(u)\ \cdot K^{-1}\ u) \tag{2.3}$$

The warped image can then be obtained by bilinear sampling from the novel view image around the warped coordinates:

$$\hat{I}_s^m(u) = I_v^m(\hat{u}) \tag{2.4}$$

Alongside the warped image, a binary validity mask V_s^m is also generated, indicating "valid" pixels in the synthesized view as some pixels project outside the image boundaries in the novel view. As previously done in context of learning monocular depth estimation [60], we can then formulate a photo-consistency objective specifying that the warped image should match the source image. In our scenario of a multi-view system, this can naively be extended to an inverse-warping of all M novel views to the reference view, with the loss being:

$$L_{photo} = \sum_{m}^{M} ||(I_s - \hat{I}_s^m) \odot V_s^m|| \tag{2.5}$$

This loss allows us to learn a depth prediction CNN without ground-truth 3D, but there are several issues with this formulation *e.g.,* inability to account for occlusion and lighting changes. While similar re-projection losses have been successfully used in datasets like KITTI[14] for monocular or stereo reconstruction, there is minimal disocclusion and lighting change across views in these datasets. However in MVS datasets, self-occlusion, reflection and shadows (see Figure 2.4) are a much bigger concern. We therefore extend this photometric loss and propose a more robust formulation appropriate for our setup.

Due to low baselines in datasets like KITTI, and consistent lighting between the different views, this loss works reasonable well for the monocular or stereo depth prediction setup in such datasets.

2.3.3 Robust Photometric Consistency

Our proposed robust photometric loss formulation is based on two simple observations – image gradients are more invariant to lighting changes than intensities, and that a point need only be photometrically consistent with some (and not all) novel views.

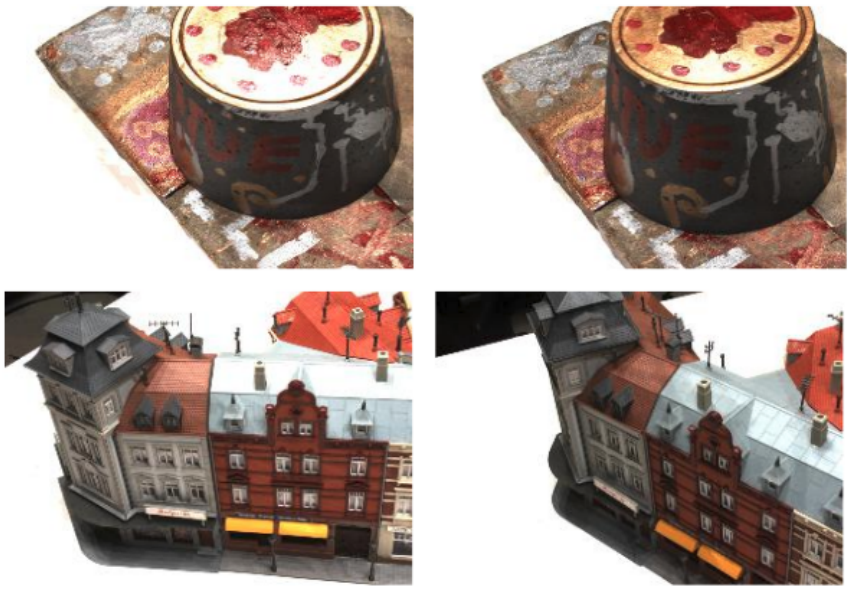

Figure 2.4: Some neighboring pairs from DTU MVS dataset. Note that despite the same light setting, there are several local variations caused by reflections and shadows. There are also complex objects which lead to abundant self-occlusion.

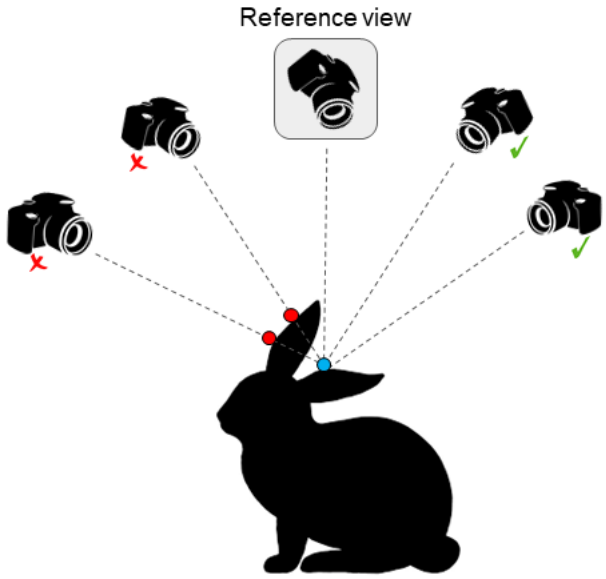

Figure 2.5: For a set of images of a scene, a given point in a source image may not be visible across all other views.

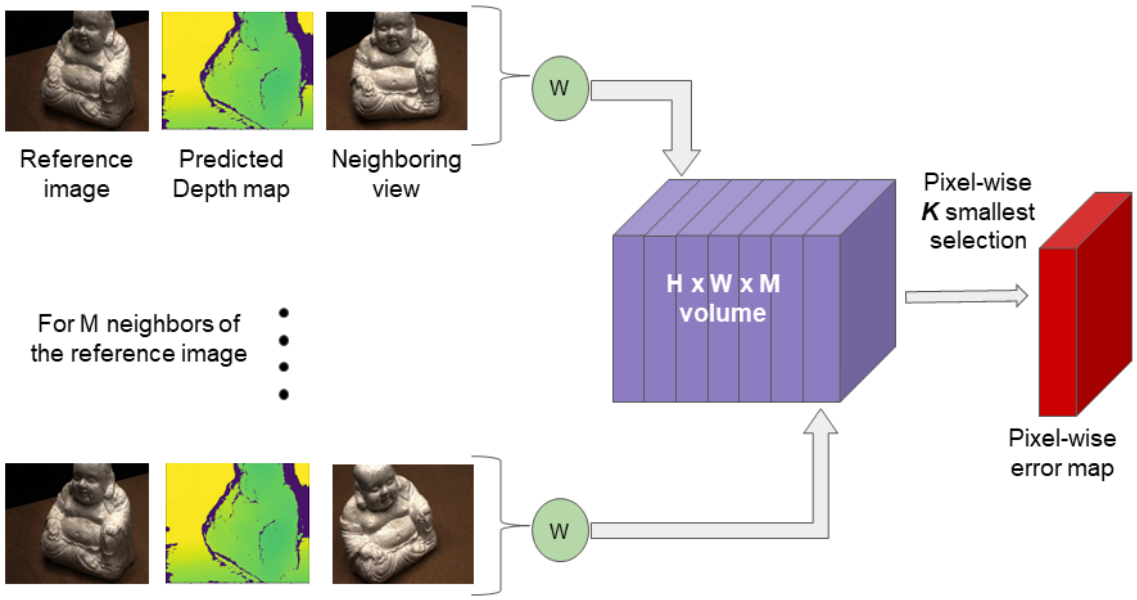

Figure 2.6: Visualization of the robust pixel-wise aggregation loss used for training. The predicted depth map from the network, along with the reference image are used to warp and calculate a loss map for each of M non-reference neighboring views, as given in eqn 2.6. These M loss maps are then concatenated into a volume of dimension $H \times W \times M$, where H and W are the image dimensions. This volume is used to perform a pixel-wise selection to pick the K "best" (lowest loss) values, along the 3rd dimension of the volume (i.e. over the M loss maps), using which we take the mean to compute our robust photometric loss.

The first modification we make in fact leverages insights developed over many years of MVS research [12], where a number of conventional approaches have found that a matching cost based on both the absolute image intensity and the difference of image gradients works much better than just the former. We also found that due to the large variations in pixel intensities between images, it is important to take a huber loss for the absolute image difference term. The inverse-warping based photometric loss of eqn 2.5 is therefore modified to reflect this:

$$L_{photo} = \sum_{m=1}^{M} ||(I_s - \hat{I}_s) \odot V_s^m||_\epsilon + ||(\nabla I_s - \nabla \hat{I}_s^m) \odot V_s^m|| \tag{2.6}$$

We refer to this as a first-order consistency loss.

We next address the issues raised by occlusion of the 3D structure in the different images. The loss formulations discussed above enforce that each pixel in the source image should be photometrically consistent with *all* other views. As shown in Fig 2.5, this is undesirable as a particular point may only be visible in a subset of novel views due to occlusion.

Our key insight is to enforce per-pixel photo-consistency with only top-K (out of M) views. Let $L^m(u)$ denote the first-order consistency loss for a particular pixel u w.r.t a novel view I_m^i. Our final robust photometric loss can be formulated as:

$$L_{photo} = \sum_{u} \min_{\substack{m_1,\cdots m_K \\ m_i \neq m_j \\ V_s^{m_k}(u)>0}} \sum_{m_k} L^{m_k}(u) \tag{2.7}$$

The above equation simply states that for each pixel u, among the views where the pixel projection is valid, we compute a loss using the best K disjoint views. An illustration of this is shown in Fig 2.6. To implement this robust photometric loss, we inverse-warp the M novel-view images to the reference image and compute a per-pixel first order consistency "loss-map". All M loss-maps are then stacked up into a 3D loss volume of dimensions $W \times H \times M$. For each pixel, we find the K least value entries with valid mask, and sum them to obtain a pixel-level consistency loss.

2.3.4 Learning Objective

While minimizing the photometric consistency loss obtained by view synthesis is our primary objective, we make use of two additional ingredients in the loss objective to improve model performance. We augment our robust photometric loss with two additional losses, namely image-patch level structured similarity loss (L_{SSIM}) and an image-aware smoothness loss (L_{Smooth}), as suggested by [39] for monocular depth prediction task, on the depth map's gradients. The smoothness loss enforces an edge-dependent smoothness prior on the predicted disparity maps. The SSIM loss is a higher order reconstruction loss on the warped images, but as it is based on larger image patches, we do not apply our pixel-wise selection approach for the robust photometric loss here. Instead, the two neighboring views with the highest view selection score are used to calculate SSIM loss.

SSIM: We take cues from recent works[15, 39, 60] showing the effectiveness of perceptual losses for evaluating the quality of image predictions. Similarly, we also use the structured similarity (SSIM) as a loss term for training. The SSIM similarity betwen two image patches is given by :

$$SSIM(x, y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x + \sigma_y + c_2)} \tag{2.8}$$

Here, μ and σ are the local mean and variance respectively. We compute μ_x, μ_y, σ_x, σ_y using average pooling and set $c_1 = 0.01^2$ and $c_2 = 0.03^2$. Since higher values of SSIM are desirable, we minimize its distance to 1 which is the highest attainable similarity value. The SSIM loss for an image pair then becomes :

$$L_{SSIM} = \sum_{ij} \left[1 - SSIM(I_s^{ij}, \hat{I_s^{ij}})\right] M_s^{ij} \tag{2.9}$$

Here, M_s^{ij} is a mask which excludes all pixels whose projections after inverse warping lie outside the source image. As observed in [39], ignoring such regions improves depth predictions around the boundaries. We apply the SSIM loss only between the reference image and two nearest images ranked by view selection score.

Depth Smoothness loss: In order to encourage smoother gradient changes and allow

sharp depth discontinues at pixels corresponding to sharp changes in the image, it is important to regularize the depth estimates. Similar to [39], we add an L_1 penalty on the depth gradients.

$$L_{Smooth} = \sum_{ij} ||\nabla_x D^{ij}||e^{-||\nabla_x I^{ij}||} + ||\nabla_y D^{ij}||e^{-||\nabla_y I^{ij}||} \tag{2.10}$$

2.3.5 Learning Setup

During training, the input to our depth prediction network comprises of a source image and $N = 2$ additional views. However, we enforce the photometric consistency using a larger set of views ($M = 6, K = 3$). This allows us to extract supervisory signal from a larger set of images, while only requiring a smaller set at inference.

Our final end-to-end unsupervised learning objective is a weighted combination of the losses described previously:

$$L = \sum \alpha L_{photo} + \beta L_{SSIM} + \gamma L_{Smooth} \tag{2.11}$$

For all our experiments, we use $\alpha = 0.8$, $\beta = 0.2$ and $\gamma = 0.0067$. The network is trained with ADAM[32] optimizer, learning rate of 0.001 and a 1st moment decay factor of 0.95. We use Tensorflow [3] to implement our learning pipeline. As also noted by [53], the high GPU memory requirements of the network imply that it is efficient to use a smaller image resolution and coarser depth steps at training, while a higher setting can be used for evaluation. We note the image resolutions used in the Experiments section 2.4.

2.3.6 Inference using Learned Depth Prediction

At test time, we take a set of images of a 3D scene, and predict the depth map of each image through our network. This is done by passing one reference image and 2 neighboring images through the network, which are chosen on the basis of the camera baselines or a view selection score if available. The set of depth images are then fused to form the point cloud. We use Fusibile [12], an open source utility, for the point cloud fusion.

2.4 Experiments

We now describe the evaluation of our proposed models. The primary dataset of evaluation is the DTU MVS dataset [26]. In section 2.4.1 we describe the DTU dataset and our training and evaluation setup, and discuss our results, qualitatively and quantatively. Next, we perform rigorous ablation studies on the effects of various components of the robust loss function we propose (Section 2.5). We also show in Section 2.4.2 that our method can allow us to adapt pretrained models to datasets without using ground-truth, by finetuning using our robust photometric consistency loss. Lastly, we demonstrate the generalization of our model to another dataset without finetuning (Section 2.4.3).

2.4.1 Benchmarking on DTU

The DTU MVS dataset contains scans of 124 different scenes with 3D structure and high-resolution RGB images captured using a robotic arm. For each scene, there are 49 images whose camera poses are known with high accuracy. We use the same train-val-test split as used in SurfaceNet [27] and MVSNet [53].

As in MVSNet, for a given reference image of one scan, its neighboring images for input to the network (N) are selected using a view-selection score [56], which uses the sparse point cloud and camera baselines to pick the most suitable neighboring views for a given reference view. We similarly use neighboring M views during training for self-supervision with the top-K loss.

We evaluate our models on the test split of the DTU dataset[26] using the officially prescribed metrics:

- *Accuracy* : The mean distance of the reconstructed points from the ground truth.
- *Completion* : The mean distance of the ground truth points to the reconstruction.
- *Overall* : the mean of accuracy and completion.

Additionally, we report the percentage metric and f-score (which measures the overall accuracy and completeness of the point cloud) as used in the Tanks and Temples benchmark [36].

Training Setup

For training, we scale the DTU images to 640×512 resolution. All of our networks are trained with $N = 3$, such that during each iteration, one reference view and 2 novel views are used for predicting a depth map. For our top-K aggregation based robust photometric loss, we use $M = 6$ and $K = 3$. Thus, 6 neighboring views are used to calculate the photometric loss volume, and per pixel the best 3 are selected. We later discuss the effect of varying K.

For evaluation on the test set, depth maps are generated at image resolution 640×512. The d_{min} and d_{max} for the plane sweep volume generation in the network is set to $425mm$ and $935mm$ respectively.

Quantitative Results

We quantitatively evaluate three unsupervised models, namely:

- *Photometric*: This model uses a combination of the naive photometric image reconstruction loss as in Equation 2.5, along with SSIM and Smooth loss.
- *Photometric + first order loss*: We replace the naive photometric loss with our proposed first order gradient consistency loss of Equation 2.6, which makes the network much more robust to local lighting variations (denoted as Photometric + G in Table 2.1).
- *Robust:* Our best model, which combines both the first order gradient consistency loss and the top-K view aggregation scheme.

To place our results in context, in addition to the unsupervised photometric setting, we compare our models against two classical methods (Furukawa et. al. and Tola et al.) [10, 48] , and two more recent deep learning methods that are *fully-supervised*, SurfaceNet [27] and MVSNet [53]. To the best of our knowledge, we are not aware of any other existing deep-learning based models that learn this task in an unsupervised manner.

We find that for our model, the one with the robust loss, significantly outperforms the variants without it across all metrics. In order to characterize the performance of our model, we further compute the percentage metrics for distance thresholds up to $10mm$ and report the f-score plot in Figure 2.13. As reported in Table 2.1, while our model struggles at a high resolution ($< 1mm$), we outperform all other methods (except the fully-supervised MVSNet model) on increasing resolutions. This indicates that while some classical methods

are more accurate compared to ours in very low thresholds, our approach produces fewer outliers. The quantitative results from Table 2.1 and qualitative visualizations of the errors in Figure 2.8 show that our robust model leads to higher quality reconstructions. Figure 2.14 shows superior performance of our model in low-texture regions.

Table 2.1: Quantitative results on *DTU*'s evaluation set [2]. We evaluate two classical MVS methods (top), two learning based MVS methods (bottom) and three unsupervised methods (naive photometric baseline and two variants of our robust formulation) using both the distance metric [2] (lower is better), and the percentage metric [36] (higher is better) with respectively $1mm$, $2mm$ and $3mm$ thresholds.

	Mean Distance (mm)			Percentage ($<1mm$)		
	Acc.	Comp.	Overall	Acc.	Comp.	*f-score*
Furu [10]	0.612	0.939	0.775	69.37	57.97	63.16
Tola [48]	0.343	1.190	0.766	88.96	53.88	67.12
Photometric	1.565	1.378	1.472	46.90	42.16	44.40
Ours (Photometric+G)	1.069	1.020	1.045	55.98	45.24	50.04
Ours (Robust: G + top-K)	0.881	1.073	0.977	61.54	44.98	51.98
SurfaceNet[27]	0.450	1.043	0.746	75.73	59.09	66.38
MVSNet[53]	0.444	0.741	0.592	82.93	62.71	71.42
	Percentage ($<2mm$)			Percentage ($<3mm$)		
	Acc.	Comp.	Overall	Acc.	Comp.	*f-score*
Furu [10]	77.30	64.06	70.06	79.77	66.27	72.40
Tola [48]	92.35	60.01	72.75	93.46	62.29	74.76
Photometric	71.68	55.90	62.82	81.92	60.56	69.64
Ours (Photometric+G)	81.11	60.70	69.43	87.03	64.36	74.00
Ours (Robust: G + top-K)	85.15	61.08	71.13	89.47	64.26	74.80
SurfaceNet[27]	79.44	63.87	70.81	80.50	66.54	72.86
MVSNet[53]	88.58	68.70	77.38	89.85	70.11	78.76

The DTU dataset's evaluation script measures the reconstruction quality in terms of accuracy and completeness while also reporting their median values and variances. We list these results for two classical (top), two supervised learning based (bottom), and three unsupervised learning (middle) methods in Table 2.2. As noted by [53], SurfaceNet[27] used their own script for evaluation. However, we use the released DTU evaluation benchmark scheme for reporting results from all the methods.

Additionally, we show the comparison of two components of the percentage metric, precision and recall, that make up the f-score reported in the main paper, in Figure 2.7.

Table 2.2: Quantitative results on the *DTU*'s evaluation set [2]. We evaluate two classical MVS methods (top), two learning based MVS methods (middle) and three variants of our unsupervised method using the distance metrics. For all columns, lower is better.

	Accuracy			Completeness			Overall
	Mean	Median	Variance	Mean	Median	Variance	
Furu [10]	0.612	0.324	1.249	0.939	0.463	3.392	0.775
Tola [48]	0.343	0.210	0.385	1.190	0.492	5.319	0.766
Photometric	1.565	1.041	3.683	1.378	0.694	4.964	1.472
Ours(Photometric+G)	1.069	0.759	1.883	1.020	0.595	2.779	1.045
Ours(Robust: G + top-K)	0.881	0.673	1.075	1.073	0.617	3.418	0.977
SurfaceNet[27]	0.450	0.254	1.270	1.043	0.285	5.594	0.746
MVSNet[53]	0.444	0.307	0.436	0.741	0.399	2.501	0.592

Qualitative Results

Qualitative results for the remaining 17 instances of the DTU test set are presented in Figures 2.8, 2.9, 2.10.

2.4.2 Fine-tuning on ETH3D

We also test the effectiveness of the robust consistency formulation as a means of fine-tuning pretrained models on unseen datasets without available annotations. For this, we evaluate our method on the recent ETH3D benchmark[44] which is divided into *low-res* and *high-res* scenes, and provides ground truth depth maps for MVS training of supervised techniques. On the low-res many view dataset of the ETH3D benchmark, we compare results of a pretrained MVSNet model with one that is fine-tuned on the train split of ETH3D. For fusion of depth maps, we run Fusibile[12] with identical hyperparameters for each.

Experiment details and Results

The results in Table 2.3 demonstrate that fine-tuning with our proposed loss, in the absence of available 3D ground truth annotations, improves performance.

Table 2.3: Effect of fine-tuning on the low-res many view ETH3D dataset. All metrics are represented as (%) and higher is better.

Method	F1 score	Accuracy	Completeness
Pretrained MVSNet	16.91	17.51	19.59
Fine-tuned MVSNet	**17.31**	**18.31**	**19.68**

2.4.3 Generalization on Tanks and Temples

We perform an experiment to check the generalization ability of our network. Since the network has explicitly been trained to match image correspondences rather than memorize scene priors, our hypothesis is that it should generalize well to completely unseen data. We select the Tanks and Temples dataset for this purpose, which contains high-res images of outdoor scenes of large objects. We use our model trained on DTU on images from this dataset, without any fine-tuning. We downscale the images to 832×512 resolution and use 256 depth intervals for the plane-sweep volume. The results are visualized in Figure 2.11. More extensive results are provided in the supplemental. However, we do note that the very high depth range of scenes in open-world datasets like Tanks and Temples are not amenable to the current deep architectures for MVS, as they all rely on some sort of volume formulation. Thus to sample depths at a finer resolution for higher quality reconstructions becomes extremely computationally expensive, and is perhaps a promising direction for future work.

2.5 Ablation Studies

This section analyzes the influence of several design choices involved in our system, and further highlights the importance of the robust losses in our training setup.

2.5.1 Top-K Selection Frequency

In order to characterize the top-K choice selection, we visualize the frequency with which pixels from different views are selected for photo-consistency. We run the trained model on the training and validation datasets for 50 iterations and store frequency counts of top-K operations which are shown in Figure 2.12. We can observe two things: 1) A view's

selection frequency is directly proportional to its view selection score. This validates that the view-selection criterion used for picking image sets for training corresponds directly to photo-consistency, 2) More than 50% of selections are from views ranked lower than 2 which explains why adding the flexibility of accumulating evidence from additional images leads to better

2.5.2 Top-K Selection Threshold.

We ablate the effect of varying K in our robust loss formulation. As can be seen from Table 2.5, using $K = 3$ i.e. 50% of the non-reference images has a substantially better validation accuracy. Note that for validation, we use accuracies against the ground truth depth maps. We report percentages of pixels where the absolute difference in depth values is under 3%.

Table 2.4: Ablation study of models with various combinations of loss functions, in terms of validation accuracy against ground truth depth maps of DTU MVS datasets. B signifies the naive baseline photometric loss, as given in eqn. 2.5. G is the first order gradient consistent loss. Our robust model is a combination of G, SSIM, Smooth and top-K aggregation.

Loss used	L1 error	$\% < 1mm$	$\% < 3mm$
B + SSIM	6.57	30.93	55.07
B + Smooth	5.92	42.52	63.05
B + Smooth + SSIM (our baseline)	4.98	49.37	72.92
G + Smooth + SSIM	5.33	61.92	77.29
G+Smooth+SSIM w/ top k (our robust)	**4.06**	**65.33**	**81.08**

Table 2.5: Performance comparison as the K in our robust photo-consistency loss varies. Results for using best 25%, 50% and 100% of warping losses per-pixel.

Method (M=6)	K=1	K=3	K=6
Validation Accuracy ($\%$)	75.59	**81.08**	77.99

2.5.3 Impact of Loss Terms

We perform ablations to analyze the different components of our robust photometric loss. Although our models are trained in an unsupervised manner, we use the ground truth depth maps of the validation set of DTU to evaluate their performances. In particular, we evaluate the methods on 3 metrics : 1) Absolute difference between predicted and ground truth depths (in mm, lower is better) 2) Percentage of predicted depths within $1mm$ of ground truth (higher is better) 3) Percentage of predicted depths within $3mm$ of ground truth. The detailed quantitative results are provided in Table 2.4. We observe that both the proposed modifications over the naive baseline yield significant improvements.

2.6 Discussion

We presented an unsupervised learning based approach for multi-view stereopsis, and proposed robust photometric losses to learn effectively in this setting. This is however, only an initial attempt, and further efforts are required to realize the potential of unsupervised methods for this task. We are however optimistic, as an unsupervised approach is more scalable as large amounts of training data can be more easily acquired. In addition, as our experiments demonstrated, these unsupervised methods can be used in conjunction with, and further allow us to improve over supervised methods, thereby allowing us to leverage both, the benefits of supervision along with the scalability of unsupervised methods. Lastly, we also hope that the proposed robust photometric loss formulation would be more broadly applicable for unsupervised 3D prediction approaches.

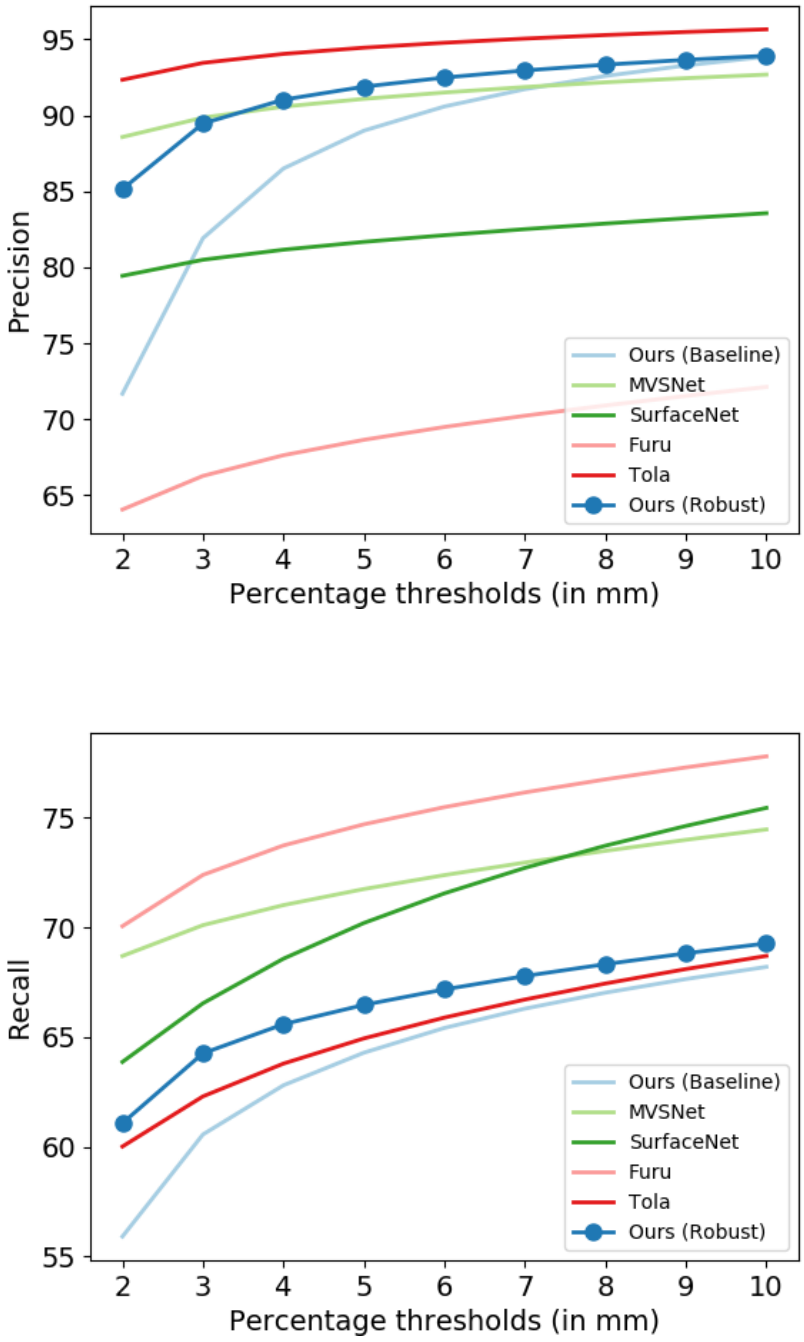

Figure 2.7: Percentage metrics on the DTU datasets as proposed in [36]. The f-scores reported in Table 2.1 are computed using precision and recall which are displayed here.

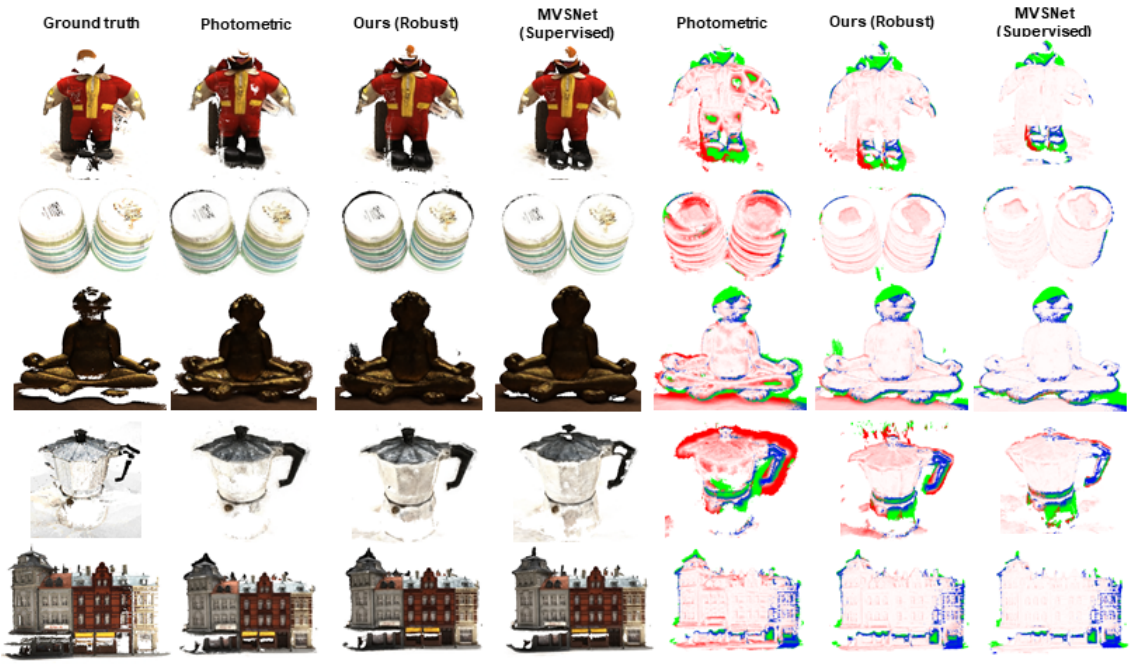

Figure 2.8: Left to right: a) Ground truth 3D scan, b) result with baseline photo-loss, c) result with robust photo-loss, d) result using a supervised approach (MVSNet [53]), e-g) corresponding error maps. For the error maps, points marked blue/green are masked out in evaluation. Magnitude of error is represented by variation from white-red in increasing order. Note how our method reconstructs areas not captured by the ground truth scan- doors and walls for the building in last row, complete face of statue in third row. Best viewed in color.

Figure 2.9: Predictions for the remaining instances of the DTU test set. Left to right: a) Ground truth 3D scan, b) result with baseline photo-loss, c) result with robust photo-loss, d) result using a supervised approach (MVSNet [53]), e-g) corresponding error maps. For the error maps, points marked blue/green are masked out in evaluation. Magnitude of error is represented by variation from white-red in increasing order. Best viewed in color.

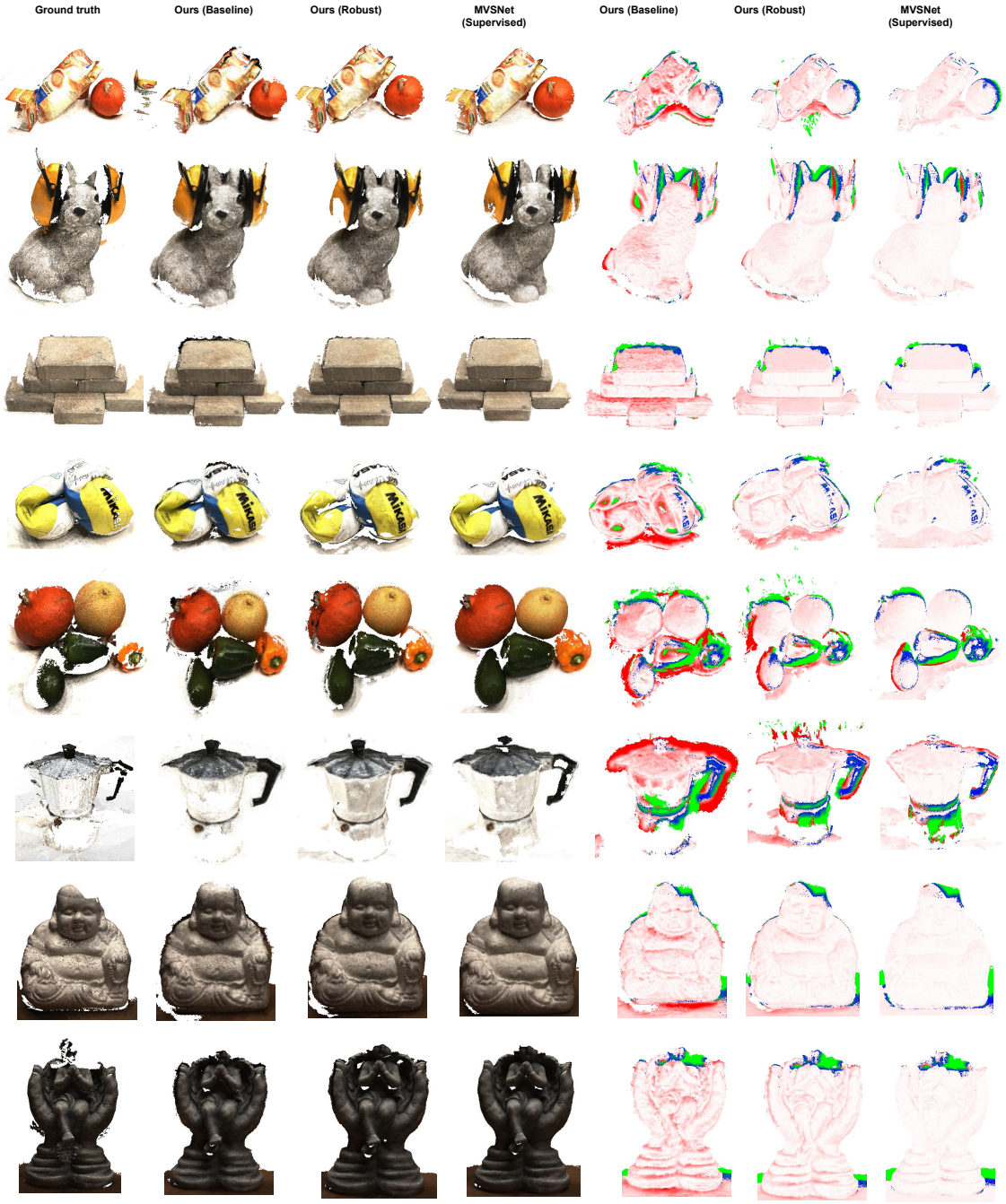

Figure 2.10: Predictions for the remaining instances of the DTU test set. Left to right: a) Ground truth 3D scan, b) result with baseline photo-loss, c) result with robust photo-loss, d) result using a supervised approach (MVSNet [53]), e-g) corresponding error maps. For the error maps, points marked blue/green are masked out in evaluation. Magnitude of error is represented by variation from white-red in increasing order. Best viewed in color.

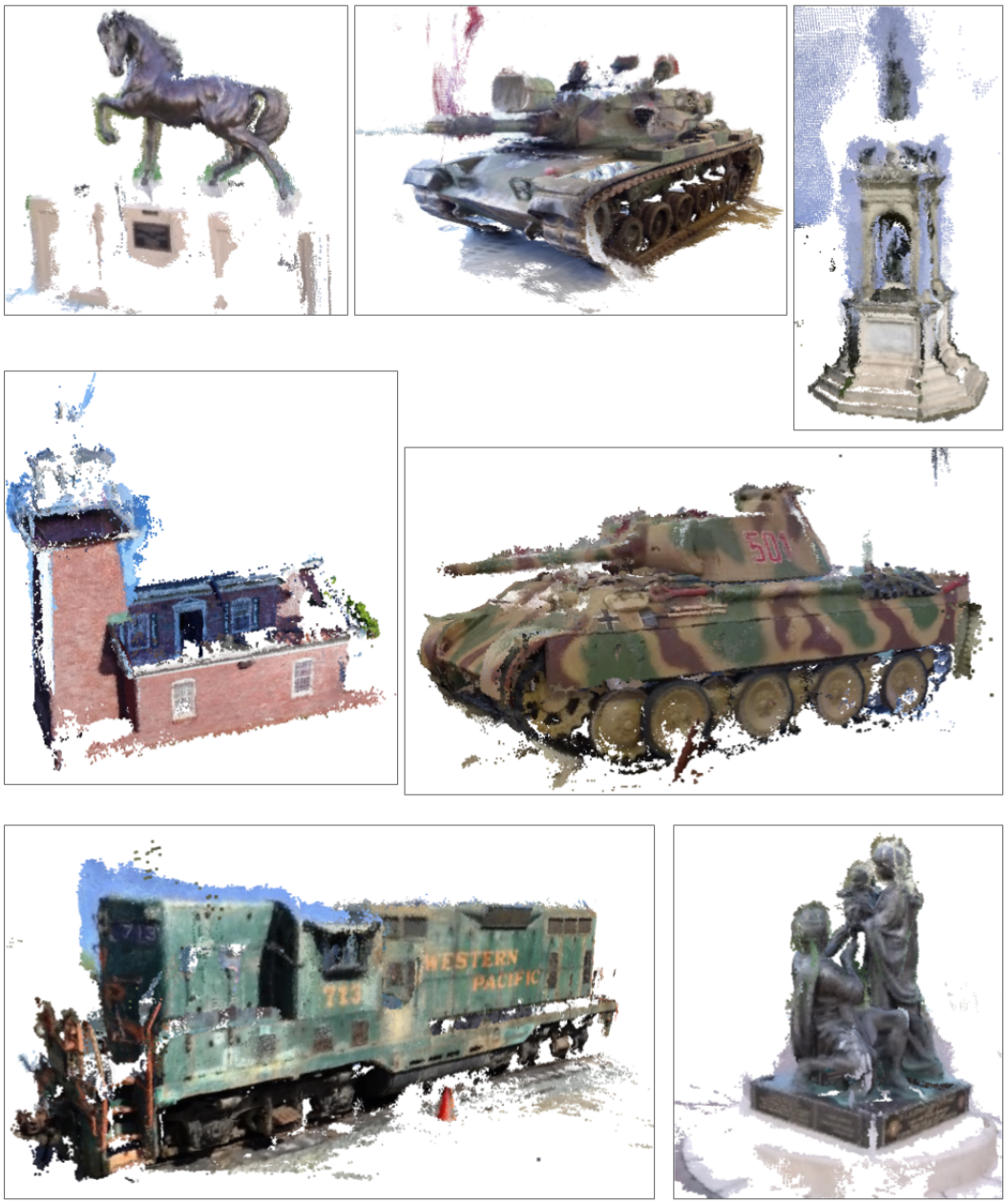

Figure 2.11: Generalization result of our robust model on the Tanks and Temples[36] dataset. Without any finetuning, our robust model provides reasonable reconstructions.

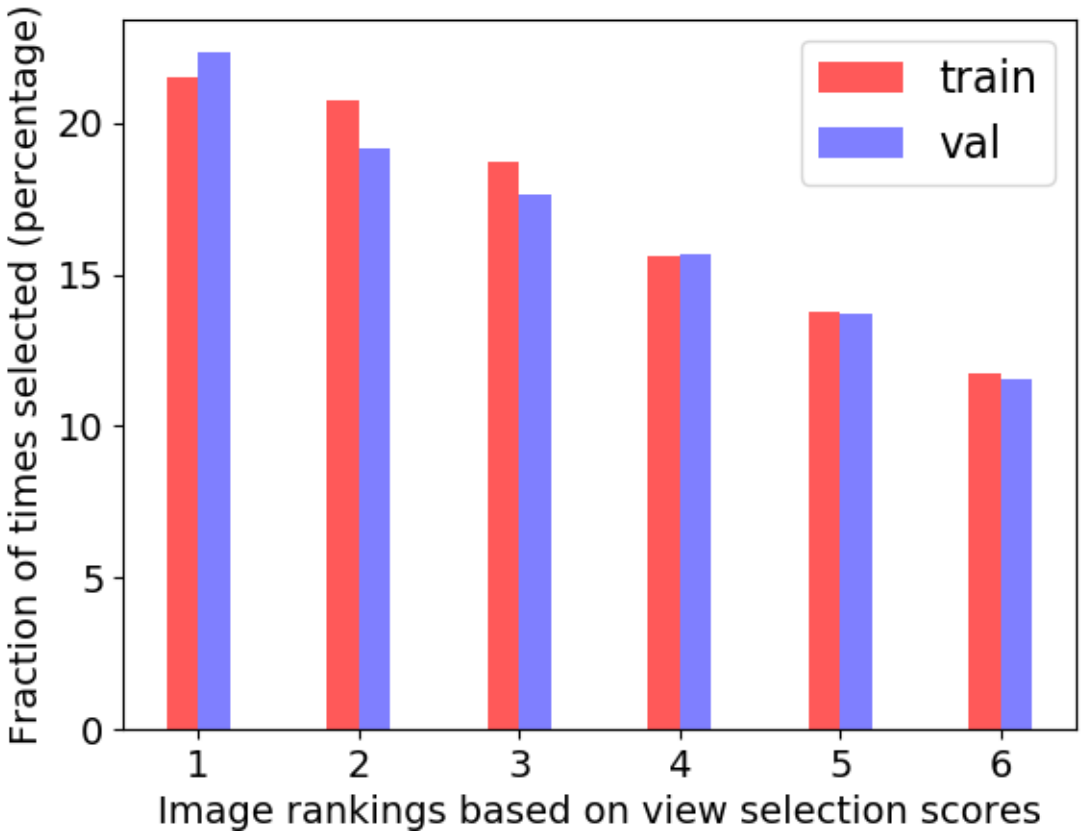

Figure 2.12: Frequency with which pixels from differently ranked images are picked as valid contributors to the top-K photo-loss. The input images are ranked based on the view selection scores as detailed in Section 2.4.1.

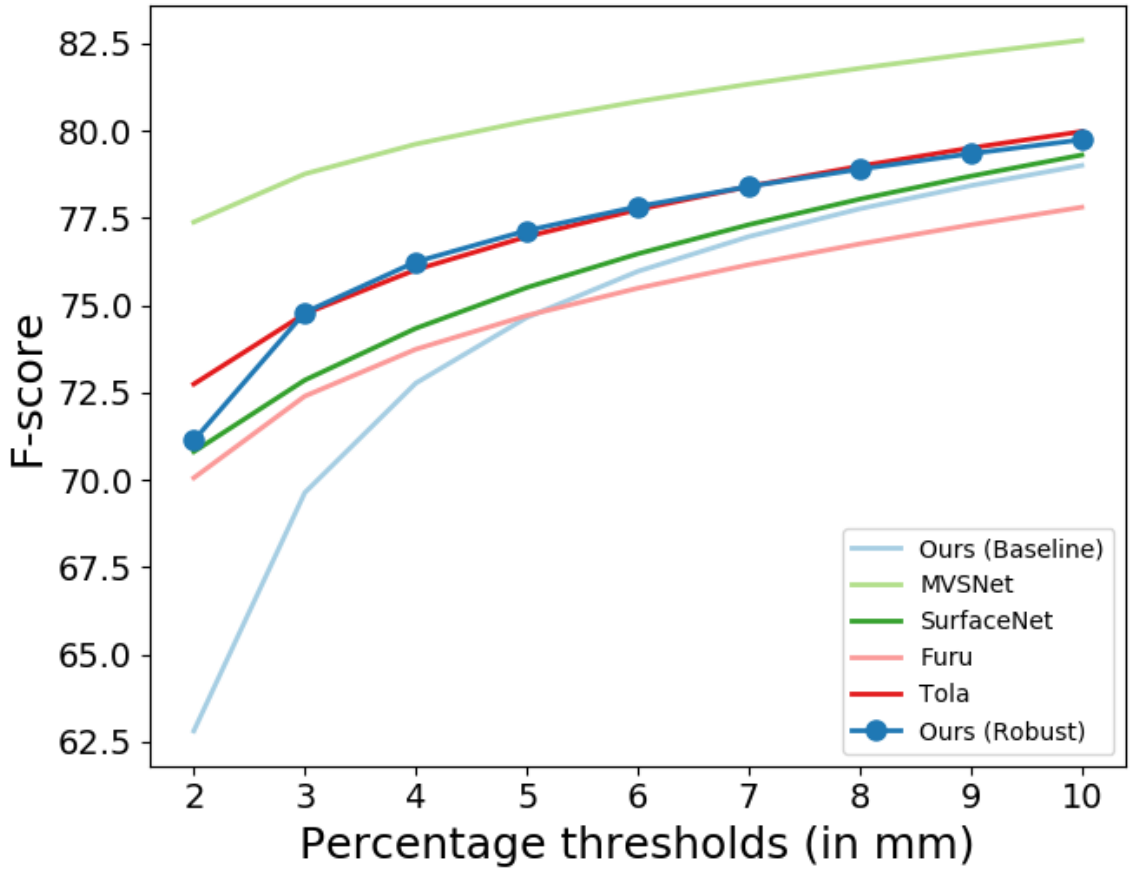

Figure 2.13: Comparison of different models on the *DTU*'s evaluation set [2] using the F-score metric proposed in [36]. We see that our model trained with robust loss consistently outperforms the baseline and several classical methods.

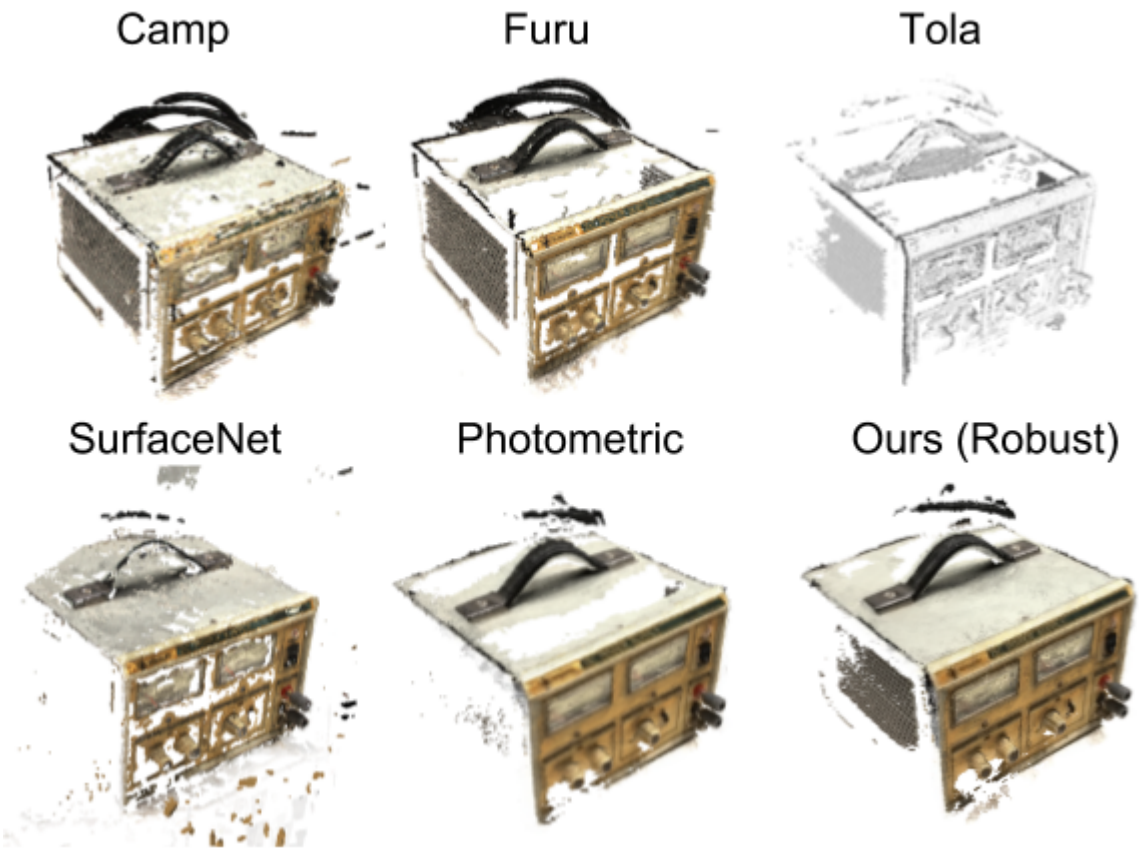

Figure 2.14: An example of how our proposed technique improves completeness over other methods in low-texture regions. Our result is a smooth dense reconstruction with significantly fewer holes or missing regions.

Chapter 3

Learning Blocks World Representation

In the previous chapter, we demonstrated how we can learn to reconstruct 3D scenes in the form of dense point clouds without using any 3D supervision. The resulting reconstructions often contain millions of points that can be interpreted as a dense sampling of the 3D surface. Given the extent of repeating patterns in our visual world, it is natural to ask if such an expensive representation is always needed. Often the representations of data are task dependent and what we seek is a *sufficient* representation that is not necessarily densely complete and detailed. There is a rich history of work on representing shapes as collections of simpler entities, called visual primitives, dating several decades back starting with the work of Binford in 1971 that introduced generalized cylinders[6]. Even earlier in 1965, Larry Roberts proposed the famous Blocks World[42] which was an attempt at complete scene understanding in a toy world composed of only textureless polyhedral shapes referred to as blocks. More recently, learning based methods[49] have been used for obtaining such volumetric representations. We refer interested readers to [17, 49] for an excellent survey of previous work.

How can we represent our 3D reconstructions parsimoniously? How can prior knowledge of the scene structure and its regularities be exploited to obtain an abstract, parameterized model of the scene? These are the questions that motivate the task of *primitive fitting* to point clouds. The primitive fitting task involves the use of a 3D point cloud as input with the goal of producing as output a collection of primitive shapes that best fit the input points. In line with the principle of parsimony, we seek to obtain the best fit using the least number of primitives. Such a system can be used for obtaining a decomposition of a scene into

its constituent objects or of an object into its constituent parts. It may also be used as a regularizing constraint for 3D reconstructions to increase completeness. We might expect some tasks such as recognition or synthesis to also benefit from such representations.

More specifically, in this chapter we introduce learning based techniques that consume 3D point clouds and produce as output a variable number of parameterized shapes that best represent the points. For the scope of this thesis, we limit the range of shapes to be rigid transforming cuboids and the input point clouds to be from man-made structures such as airplanes, chairs, tables, and buildings.

3.1 Motivation

Unlike images in 2D, there is no single 3D representation that works best for all scenarios. However, for the reconstructions produced by MVS methods, there are some distinct advantages of a primitive based representation over polygonal meshes. We enumerate some of them below.

1. **Incorporating prior knowledge.** Depending on the task, we might want to inject specific rules and exploit regularities in the man-made structures. Dealing with simple geometric primitives allows us to explicitly encode a number of structural priors into our learning scheme in the form of constraints.
2. **Parsimonious representation.** Primitive based representations contain orders of magnitude lower number of parameters than the dense point clouds or polygonal meshes which often have millions of points against the few dozen for primitives.
3. **Symbolic representation.** The assembly of primitives into objects and consequently of objects to form a scene provides a symbolic representation that is easy to manipulate and operate on enabling high-level reasoning such as visibility, etc.

3.2 Challenges

There are three closely entangled sub-problems that make the task of primitive fitting challenging. They are:

1. How many primitives to use?
2. Which primitives to use?

3. How to determine the best fit?

Seen as such, the task is akin to model selection. At its core, it demands learning the arrangement of a variable number of primitives while simultaneously determining the types, counts, shapes and locations of them. This can be seen as a hierarchical process wherein the complex structure is partitioned into sub-structures for which we determine the appropriate primitives and their fit. For the purpose of this thesis, we assume the input point clouds to be pre-segmented objects from a scene. The primitive fitting is conducted on individual objects and the resulting models can then be projected back into the scene as shown in Figure 3.14 and Figure 3.15.

3.3 Unsupervised Primitive Fitting

We formulate the problem of obtaining a blocks representation of a output structure O, given a set of input points P as that of predicting (up to) M distinct shapes (parts) which are then composed to assemble the final shape. Towards this, we learn an encoder-decoder style model f_θ, parameterized by θ, which consumes 3D point clouds, represented as a collection of (x, y, z) coordinates, and outputs a primitive based representation. We do not have any annotations for the primitives that best represent the underlying surface from which the input points can be thought to be sampled from. Hence the task of learning such a model is an unsupervised one. While there is no direct supervision on the parameters on the primitive shapes, we can measure the *goodness of fit* of the complete assembled structure to the available point set and use this estimate as a guiding signal to train the model. This insight suggests that we would be optimizing for M parameter settings that would represent shapes which are the most likely surfaces containing the available input points P.

3.3.1 Parameterized Shape Representation

The output structure O is represented as an assembly of M primitive shapes. We follow the representation scheme suggested in [49]. Each primitive shape (block) is encoded as a vector of length 10 and is of the form (z, t, q) where z represents the dimensions of the shape in a canonical frame and (t, q) represent the spatial tranformation (translation, orientation). In 3D, z is a 3-tuple denoting the dimensions of the block (height, width, breadth), t is a 3-tuple denoting the coordinates of the center of the block (x, y, z) and q is a

4-tuple encoding the orientation as a unit-quaternion. As demonstrated in [30], the choice of quaternions to represent orientation is motivated by the ease of mapping arbitrary 4-tuples to valid rotations by normalizing them to unit length. This poses a simpler optimization problem than the orthonormalization required by rotation matrices, while also avoiding the wrap-around issue with Euler angles. The structure O predicted by the model f_θ can therefore be written as below.

$$f_\theta(P) = \{(z_m, t_m, q_m) | m = 1, \ldots, M\} \tag{3.1}$$

This representation explicitly models the compositionality of sub-structures and exploits the independence of the shape formation (z) and shape placement (t, q). There is a natural interpretation to the training signal that follows from the entanglement of the parameters — should the dimensions of the shape be altered or simply its placement such the resulting fit improves.

3.3.2 Network Architecture

The design of a neural network that operates directly on 3D point clouds is challenging for a number of reasons. First, a point cloud is an unordered set, which means permutations of the points do not change the geometry they represent. This necessitates the design of a feature extractor and a loss function that are permutation invariant. Second, there is no clear definition of local neighbourhoods in point clouds, making it difficult to apply any convolutional operation. Third, the number of input points can vary drastically. Taking these into account, we build our encoder model inspired by the recently proposed PointNet[41] architecture. The PointNet model includes two spatial transformer sub-networks called T-Net which provide invariance under tranformations like rotation and translation. Such spatial invariance is desirable for tasks like classification or shape recognition, wherein rotating a point cloud does not change its object class identity. However, for our task of regression, we want the network to be sensitive to all changes in the input and learn to make predictions accordingly. Hence we remove this spatial transformer component and let the network be sensitive to spatial changes in the input.

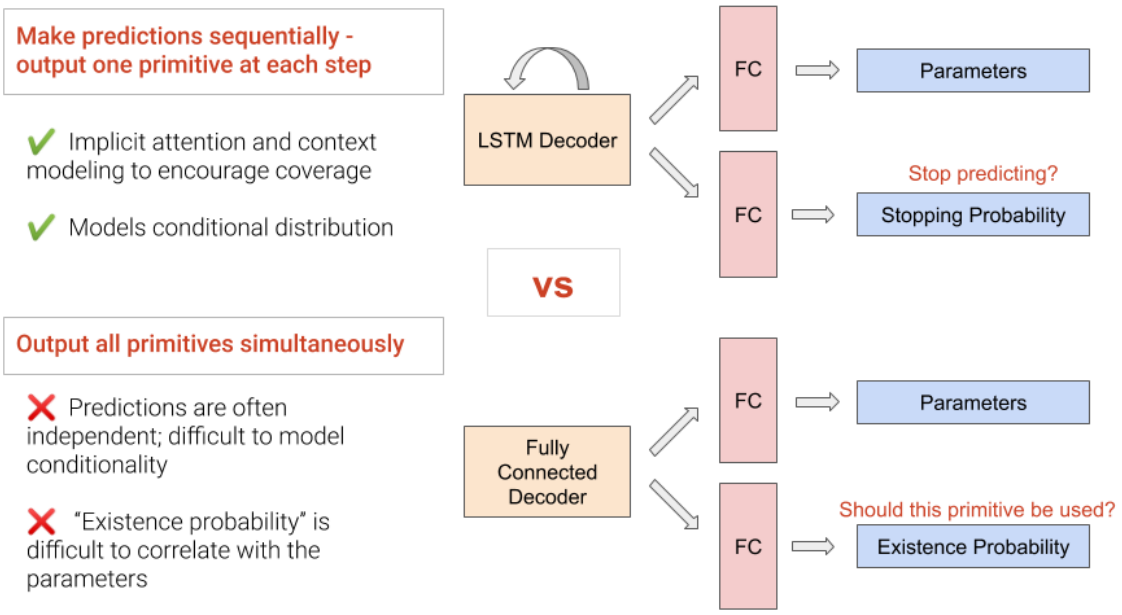

Figure 3.1: Two candidate choices for a decoder which would predict the final shape parameters from the point cloud feature representation produced by the encoder model.

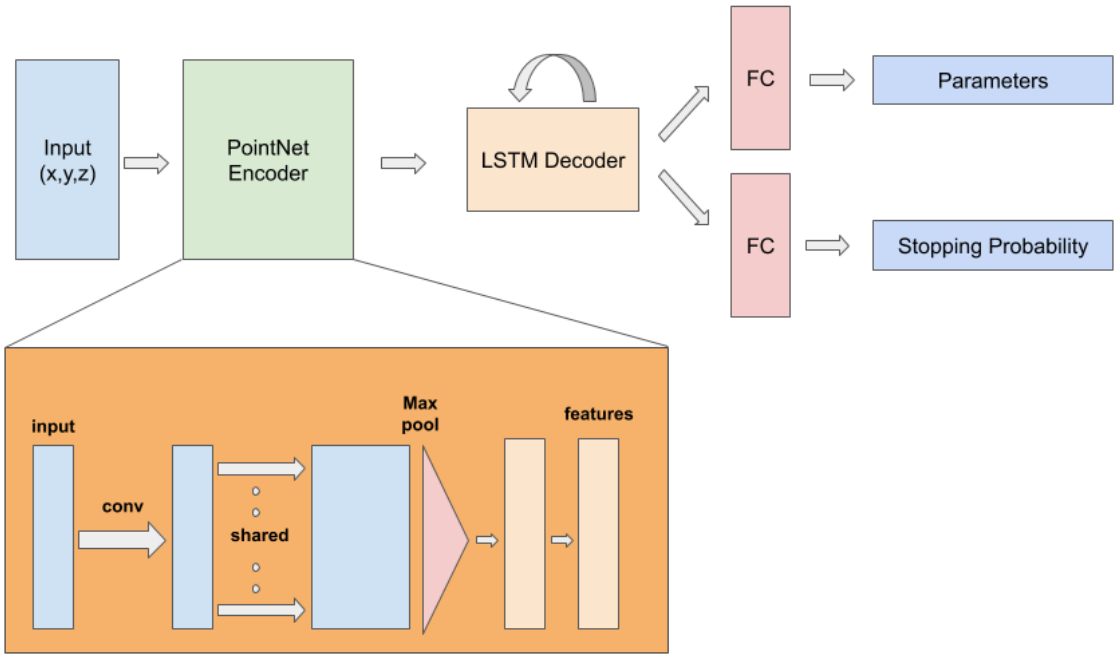

Figure 3.2: A sketch of our model architecture that consumes a 3D point cloud as input and produces a variable number of parameterized primitive shapes as output.

We experiment with two choices for a decoder — MLP (a sequence of fully connected

layers) and LSTM[21] based. The comparison of the two models is shown in Figure 3.1. In the case of the MLP decoder, we predict the probability of existence of a primitive, p_m, along with the shape parameters to allow us to model variable counts of primitives. The predicted probability p_m is treated as the parameter of a Bernoulli distribution according to which the predicted shape is sampled (used). During training, the gradients of this distribution parameter are computed using the REINFORCE algorithm[51] as described in [49]. On the other hand, the LSTM decoder outputs a stopping probability which can be thresholded to determine the number of predictions to make.

3.3.3 Learning Objective

In order to train the model described above, we would like to define a differentiable loss function $L((z_m, t_m, q_m), P)$ between the predicted structure (z_m, t_m, q_m) and the input point set P. This is challenging because the two 3D representations the loss function is to operate on are different — one is a 3D point cloud and the other is a collection of 3D blocks. To overcome this, we leverage the fact that the point cloud is a discrete sampling of the underlying 3D surfaces and hence a geometric fit between the two representations would imply high fidelity reconstruction of the true underlying surface. This allows us to define some complimentary losses which together capture the *goodness of fit* by minimizing the discrepancy between the predicted and ground-truth shape. The *Surface Distance Loss* enforces the predicted structure to contain the input point set by minimizing the geometric distance of every point to its nearest surface. The *Chamfer Distance Loss*, on the other hand, enforces a similarity between the input point set and a discrete uniform sampling of the predicted structure. Additionally, the *Intersection Loss* encourages predicted primitives to be non-overlapping. For dealing with variable number of primitives, we incorporate the *Parsimony Loss*.

Surface Distance Loss. Given a set of points and a collection of blocks, the surface loss computes the L_2 distance between each point and the nearest surface of the nearest block. With the choice of primitive parameterization, such a loss can be computed efficiently in the form of a distance field as described in [49]. For an origin-centered block, the distance field $dist(\cdot; z)$ can be computed by,

$$dist(p; z)^2 = (|p_x| - w)_+^2 + (|p_y| - h)_+^2 + (|p_z| - d)_+^2 \tag{3.2}$$

where $z \equiv (w, h, d)$ denotes the extent of the block in the three dimensions. If we denote the rotation and translation of the block by R and t respectively, then the distance field at a point p w.r.t. the transformed block is the same as that at a point p' w.r.t. the canonical origin-centered block where $p' = R^{-1}(p - t)$. This observation allows us to transform all the input points P to the canonical frame of an origin centered block and easily compute distances to all sides of block. The final loss is thus defined as the loss over all points.

$$SD(O, P') = \sum_{i=1}^{N} \min_{O_m \in O} dist(p'_i; O_m) \tag{3.3}$$

Chamfer Distance Loss. We would like our predicted output structure O to be such that a discrete uniform sampling of its surfaces would generate a point cloud that closely resembles the input point cloud P. For this we use the permutation invariant Chamfer distance loss introduced in [9]. In order to uniformly sample points on the surfaces of the predicted structure, we build upon the observation described above for obtaining transformed points. Similarly, we can first sample points p on a block in the canonical origin-centered reference frame and then apply the predicted rotation and translation transformations to the sampled points to obtain a point cloud P' in the same reference frame as the input points P. The distance between these two point sets is then given as below.

$$CD(P, P') = \frac{1}{|P|} \sum_{x \in P} \min_{y \in P'} \|x - y\|_2 + \frac{1}{|P'|} \sum_{y \in P'} \min_{x \in P} \|y - x\|_2 \tag{3.4}$$

CD (3.4) calculates the average closest point distance between the output point cloud P and the ground truth point cloud P'. We use the symmetric version of CD where the first term forces output points to lie close to ground truth points and the second term ensures the ground truth point cloud is covered by the output point cloud. Note that P and P' need not be the same size to calculate CD.

In order to facilitate gradient computation of the predicted parameters under this loss, we use the reparameterization trick[34] which decouples the parameters from random sampling. Similar to [49], this is done by sampling $u \sim [-1, 1]$ and then scaling the coordinate by multiplying with it to obtain an intermediate value. For example, in order to sample

the x-coordinate, instead of randomly picking a value $x \sim [-w, w]$, we now use $x = uw$ which allows us to compute $\frac{\partial x}{\partial w}$ and enables training.

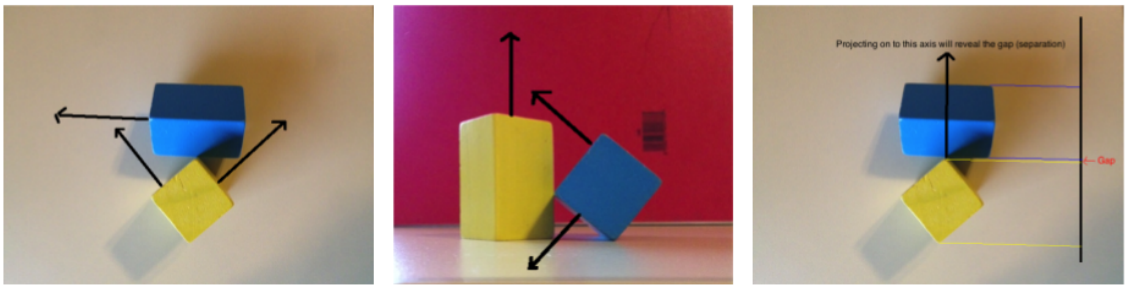

Figure 3.3: An illustration of collision detection using the Separating Axis Theorem. Images taken from [1].

Intersection Loss. We would like to obtain a parsimonious primitive configuration that is physically plausible and reasonable. A key constraint to enforce towards this goal is to have the predicted structure contain non-overlapping shapes. This constraint can be posed as a collision avoidance objective. For our purposes, we use the Separating Axis Theorem (SAT)[8] for detecting intersections between pairs of blocks (cuboids). The theorem is stated below.

Theorem 3.3.1 (Separating Axis Theorem) *Two nonintersecting convex polyhedra can be separated by a plane that is either parallel to a face of one of the polyhedra or that contains an edge from each of the polyhedra.*

Examination of the intersections of the projections of the polyhedra on lines that are perpendicular to the planes is both a necessary and sufficient condition to determine if two convex polyhedra are intersecting. The polyhedra would be intersecting if and only if the minimal intervals containing their projections onto one of the axes intersect. Such an axis is called a separating axis. Hence, the collision/overlap detection amounts to processing each of the potential separating axes one by one by projecting the blocks onto an axis and testing the intersection of the minimal intervals containing the projections. In 3D, we have separating planes that separate the bounding volumes and are perpendicular to their corresponding separating axes. We can stop the tests early if a separating axis is found. We implement the intersection loss as a pairwise collision detection scheme and relax the hard choices of selecting minimum intervals to be a soft operation which makes the overall formulation differentiable.

Parsimony Reward. In order to encourage the MLP decoder model to use fewer primitives to describe the shape, we add an additional *parsimony reward* when computing the gradients for predicted probabilities using REINFORCE as described above. For the LSTM based decoder, we simply add the reward scaled inversely by the number of predictions made by the model such that fewer predictions (≥ 1) are rewarded.

3.3.4 Implementation Details

We perform our experiments primarily using proprietary aerial scans of buildings and using the ShapeNet[7] dataset. In both cases, we use point clouds with 2048 points as input; points are either upsampled or downsampled as needed in order to obtain the fixed size. All experiments used the Adam[33] optimizer with an exponential learning rate schedule starting from 0.003 and scaling by 0.75 times every 75 epochs. All our models were trained to predict up to 20 primitive shapes. We found it challenging to accurately train the counting aspect of the model i.e. variable number of primitives. In our experiments, the models would often collapse to not use any primitives (fixed by setting minimum to 1) or by using all primitives, with no strong signal of parsimony. Tuning the parsimony reward led to some success, but we observed better results by predicting a fixed number of primitives and consequently pruning the ones overlapping with others (more than 50% overlap) in a post-processing step.

3.3.5 Results on ShapeNet Dataset

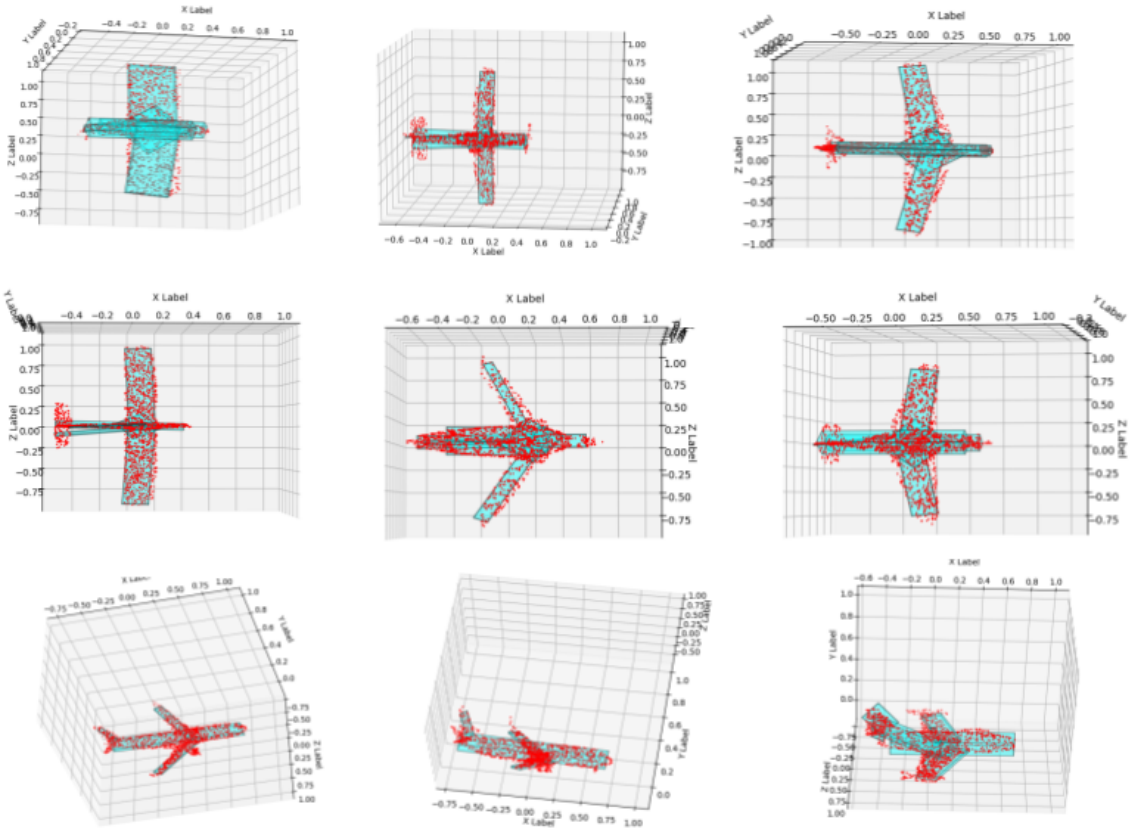

Figure 3.4: Representing airplanes from the ShapeNet dataset with primitives.

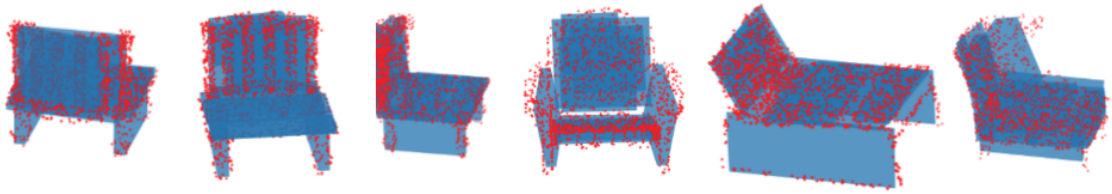

Figure 3.5: Representing chairs from the ShapeNet dataset with primitives.

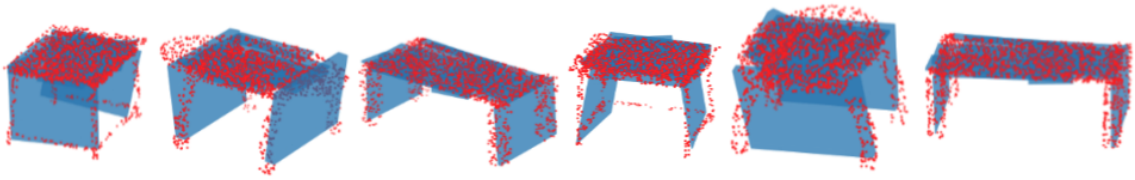

Figure 3.6: Representing tables from the ShapeNet dataset with primitives.

We picked 500 models each for the airplane, chair and table classes of the ShapeNet

dataset for training and sample points uniformly on the mesh surfaces to obtain point clouds. There is no ground truth model available for these scans for comparison of our results and hence we present qualitative results here. Note that while we have access to the synthetic CAD models, their decomposition into shape primitives (blocks) is unavailable. We show qualitative results on unseen instances of these classes in Figure 3.4, Figure 3.5, Figure 3.6.

3.4 Supervised Sim-to-Real Approach

For a class of man-made structures such as buildings, we, in fact, possess much stronger structural priors which can help significantly reduce the search space for learning primitive fitting. For example, buildings are often located on planar surfaces which means we need not predict complete 3D rotations but can instead get by with only in-plane rotations. As a result, unlike the 4-vector quaternion used for representing rotation earlier, we can now represent rotation as a single scalar value which is the deviation from an in-plane axis. This is permitted due to the symmetry of the primitive shape we consider which is a cuboid.

The regularities in such structures can be captured programmatically to a reasonable extent enabling us to simulate buildings with code. We show next how such simulations can be used to create large scale training data for models to be trained in a supervised fashion. Such supervised training can also help the models learn counting more effectively.

3.4.1 Synthetic Data Generation

We follow a simple technique based on recursive sub-division to simulate buildings programmatically as illustrated in Figure 3.7. We start with a fixed sized rectangular region and repeatedly divide it into sub-regions by sampling random points within it. A few iterations of such divisions lead to a configuration of non-overlapping rectangles. A subset of these rectangles are sampled and adjacent rectangles with completely shared edges are merged together producing a layout as shown. In order to capture non-rectangular structures, some of the boundary rectangles are replaced with a triangular or elliptical shape of appropriate dimensions. In order to emulate real sensor noise, we distort the edges of the complete layout with Gaussian noise. The layout up to this point is entirely in 2D.

We next assign heights to the regions such that the same elevation is assigned to an

entire region. Just like the boundaries, we also distort the elevations values with some Gaussian noise to emulate sensor noise. This gives us a 2.5 height map of a building which can be projected to a 3D point cloud as shown in Figure 3.8. For this projection, we first copy all elevated points to the terrain and add some Gaussian noise to distort them. Next, we interpolate sides using Poisson reconstruction and sample points uniformly. The final point cloud is obtained after aggressive downsampling. Examples of the resulting 3D point clouds are shown in Figure 3.9. We note that while the eventual point clouds represent non-cuboidal structures, the ground truth values we regress to are still the parameters of the original rectangles along with their corresponding elevation.

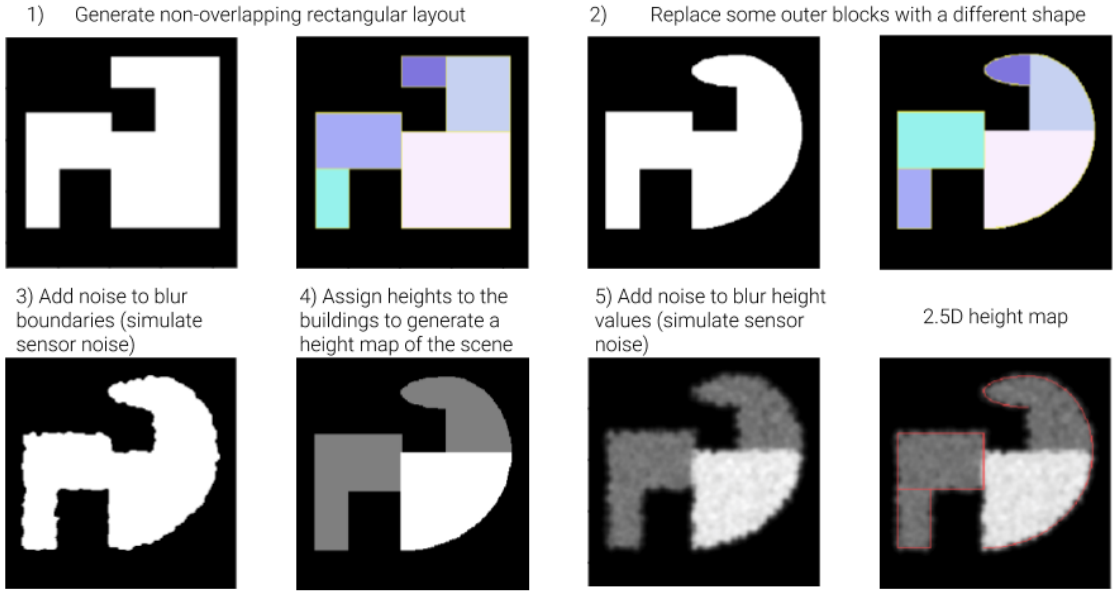

Figure 3.7: An illustration of the synthetic data generation process.

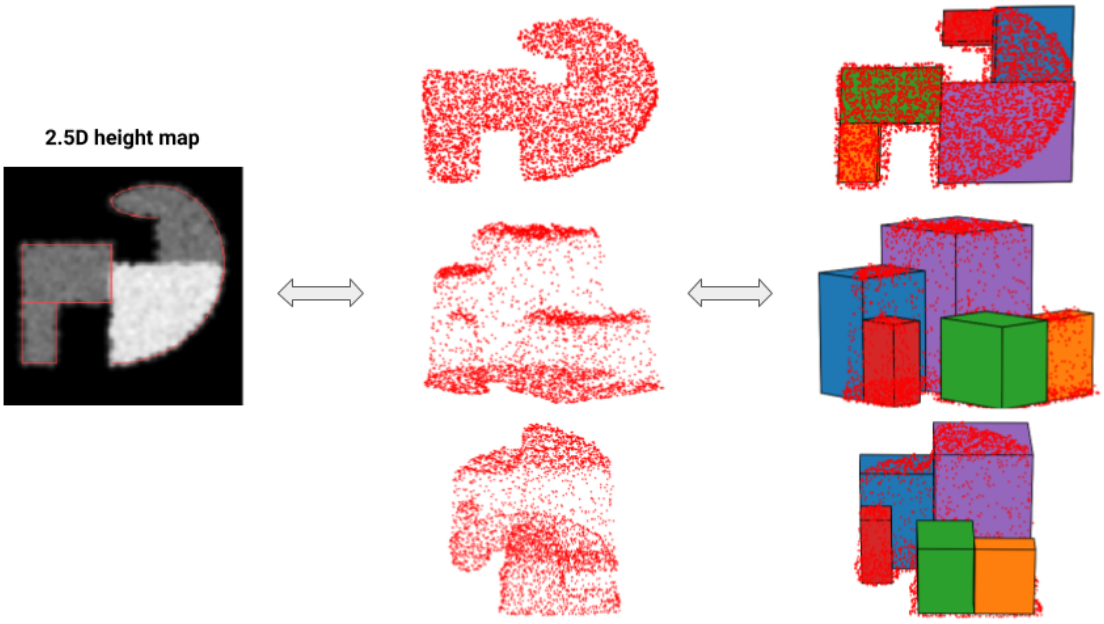

Figure 3.8: An illustration of a synthetically generated 2.5D depth map, its projection in 3D and the set of 3D primitive shapes (cuboids) that are the ground truth parsimonious representation to learn.

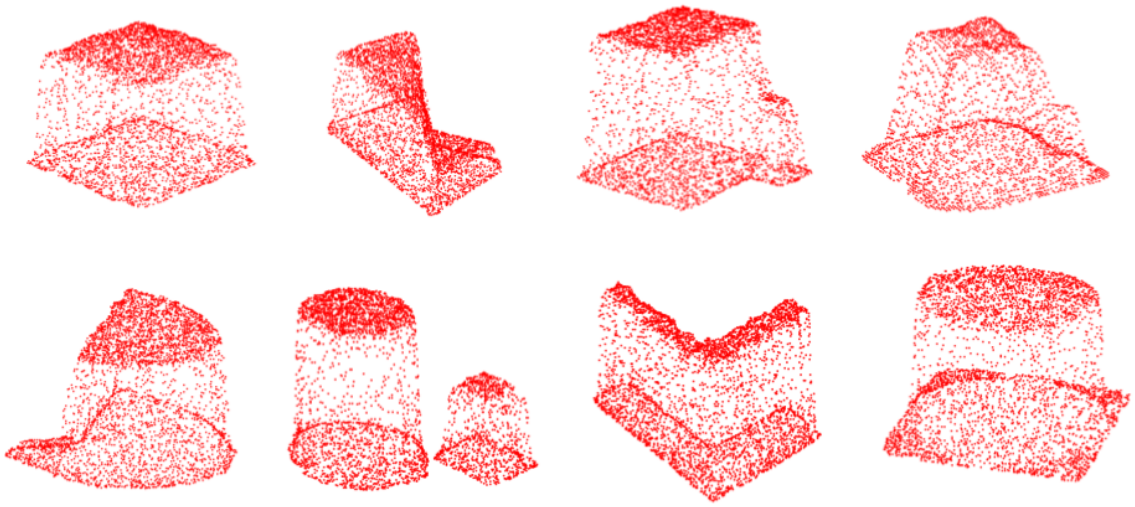

Figure 3.9: Examples of some 3D point clouds generated synthetically by sampling points uniformly on the surfaces of the primitive block shapes.

3.4.2 Network Architecture

The network architecture is same as the one described in the previous section for unsupervised learning. For the experiments here, we use the model with the LSTM decoder.

3.4.3 Learning Objective

The goal of the learning objective is similar to the one previously described. However, since we now have access to the correct ground truth counts and parameter values, we can use these during training and have the network regress to the true values directly, bypassing the indirect geometric relationships. We jointly optimize two loss terms, for counting and fitting. Since the LSTM outputs a stopping probability, we use the cross-entropy loss to train the model for true stopping time steps. The parameter predictions of the network are trained using the matching loss. During training we use all primitives predicted by the network as per ground truth counts and not the counts predicted. This helps attain more stable training.

Matching Loss. The model predicts a set of parameters that represent a variable number of primitive shapes which may or may not be as many as those in the ground truth. Additionally, since we care only about the assembled shape (placing all primitives together) and not individual placements in isolation, we need a mechanism for determining one to one correspondences between the predictions and ground truth.

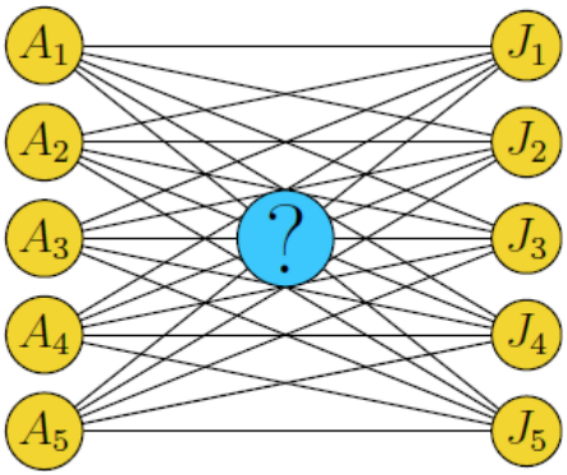

Figure 3.10: The optimal assignment problem.

As shown in Figure 3.10, the optimal assignment problems involves finding one-to-one correspondences between two sets of items by ranking all choices using some scoring function. Note that the two sets need not be of the same size and we may then only include matches according to the size of the smaller of the two sets. A classical technique for solving this problem in polynomial time is the Hungarian matching algorithm[37] which finds an one-to-one mapping between two sets. In its natural form, the Hungarian algorithm is non-differentiable and hence cannot be directly used for training a deep network. We can, however, get around the non-differentiability by first computing the cost matrix of all possible assignments in a differentiable manner and then using the optimal assignment mappings to index into the cost matrix. The loss value used during training is thus only for the optimal matches and the remaining entries do not contribute to the learning gradient. For simplicity, we use the L_1 norm distance between the ground truth and predicted parameters to compute the loss value.

3.4.4 Results on Synthetic Scans

Figure 3.11 shows some qualitative results of the trained model on a held out test set of 1000 synthetically generated scans. We can see that the model learns to precisely fit varying number of primitives to the input point cloud.

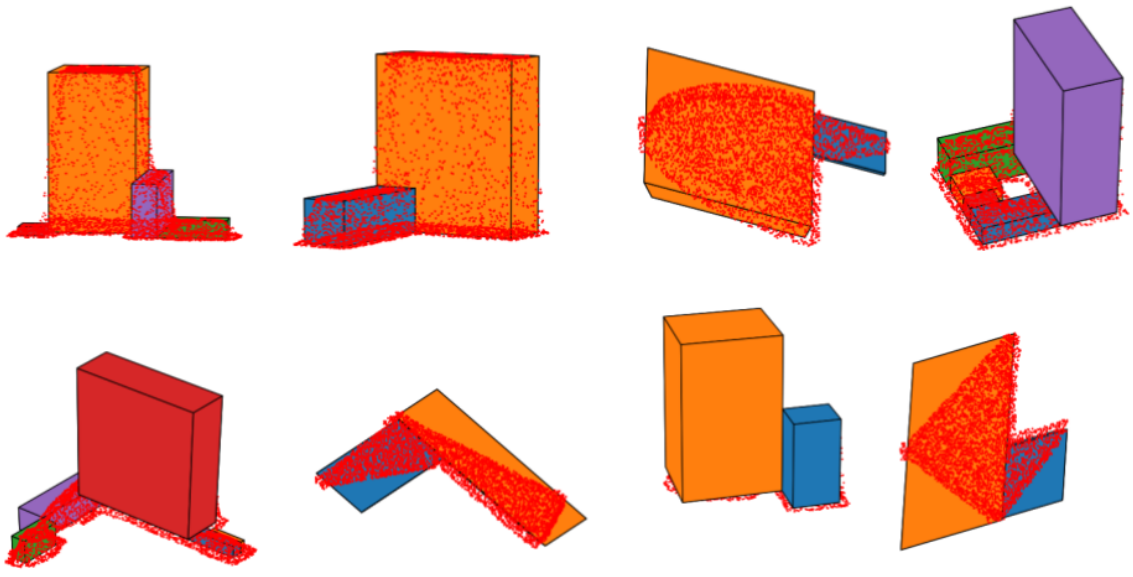

Figure 3.11: Qualitative results of the trained model on a held out test set of synthetically generated scans.

3.4.5 Results on Real Aerial Scans

The proprietary aerial scans dataset consists of 250 real scans of buildings that are segmented out from their surrounding.

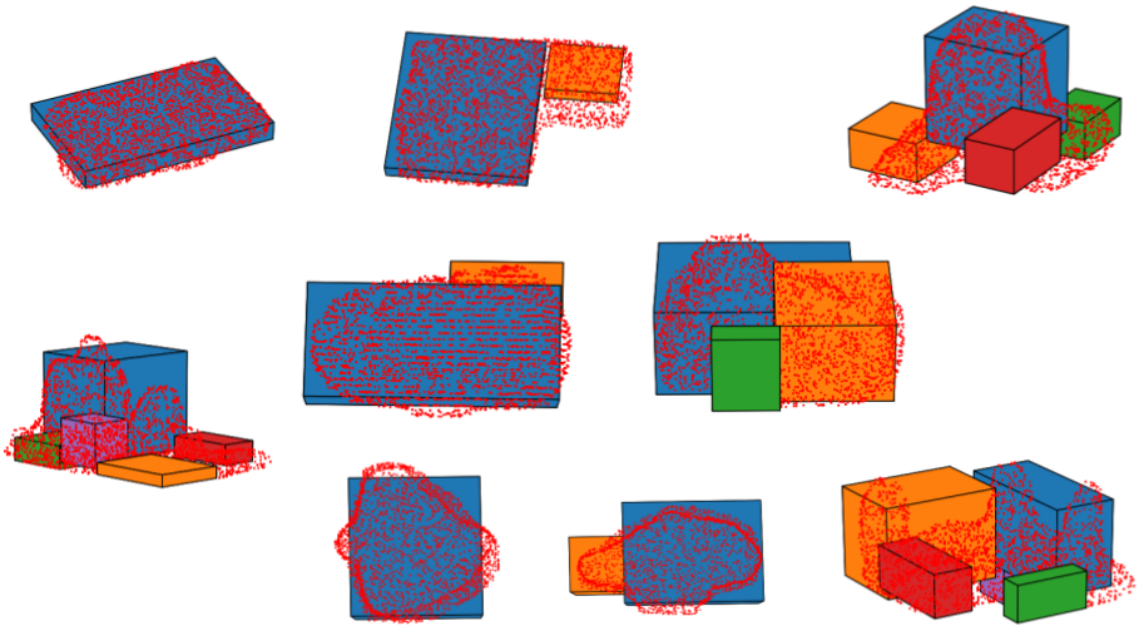

Figure 3.12: Qualitative transfer learning results of the trained model on some proprietary real world aerial scans.

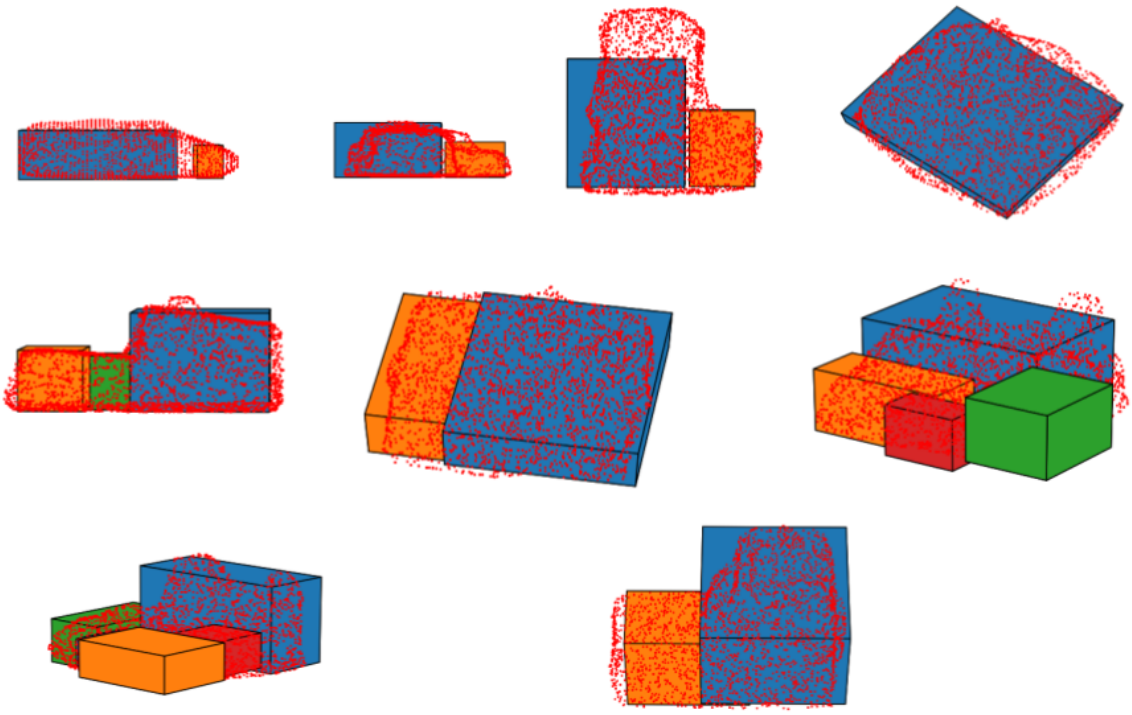

Figure 3.13: Qualitative transfer learning results of the trained model on some proprietary real world aerial scans.

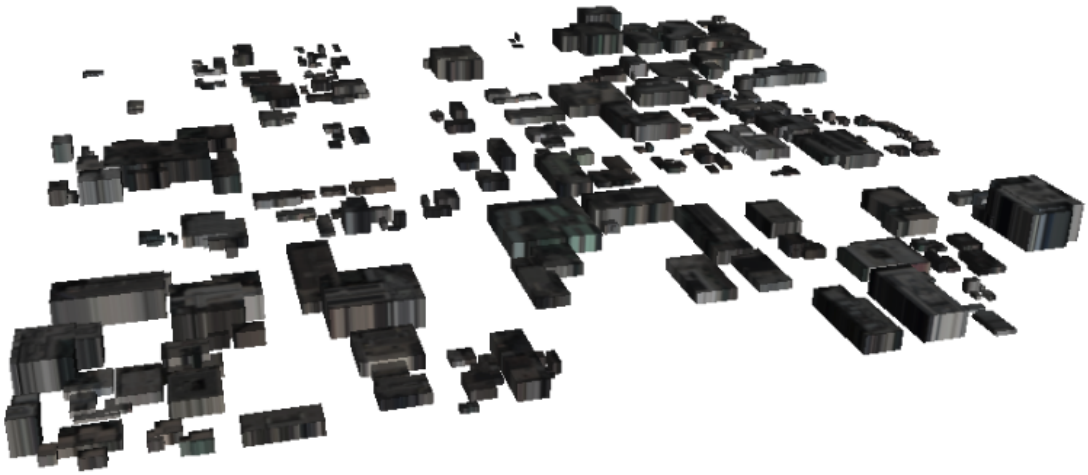

Figure 3.14: Putting all results together to form a region level map.

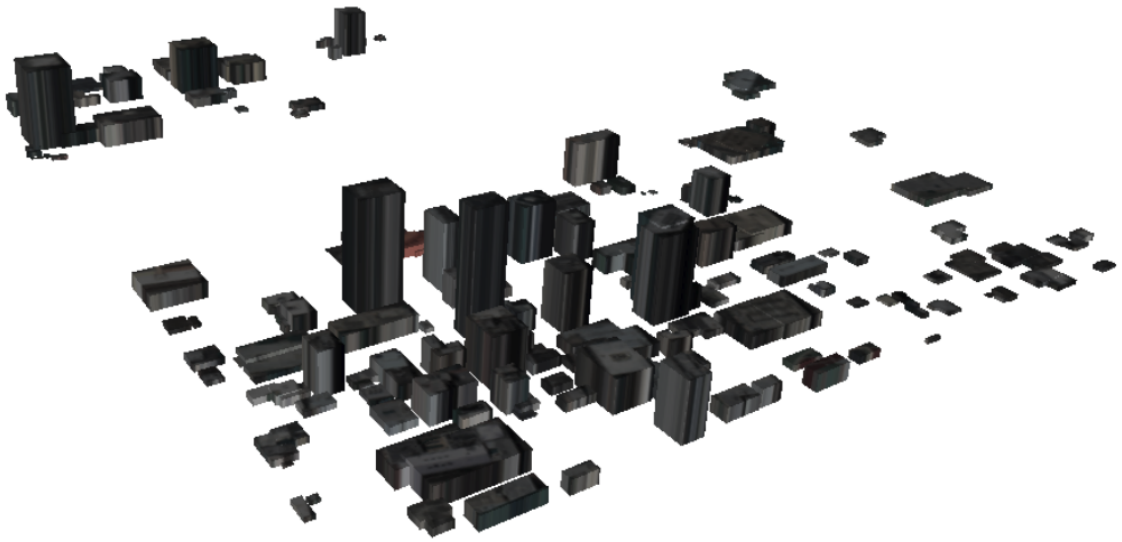

Figure 3.15: Putting all results together to form a region level map.

There is no ground truth model available for these scans for comparison of our results and hence we present qualitative results in Figure 3.12 and Figure 3.13. We observe that the model captures the spread of the buildings to an impressive extent and uses reasonable numbers of primitive shapes for the construction. We can place all resulting models back in the original world coordinates to form a compact region level map as shown in Figure 3.14 and Figure 3.15.

3.5 Discussion

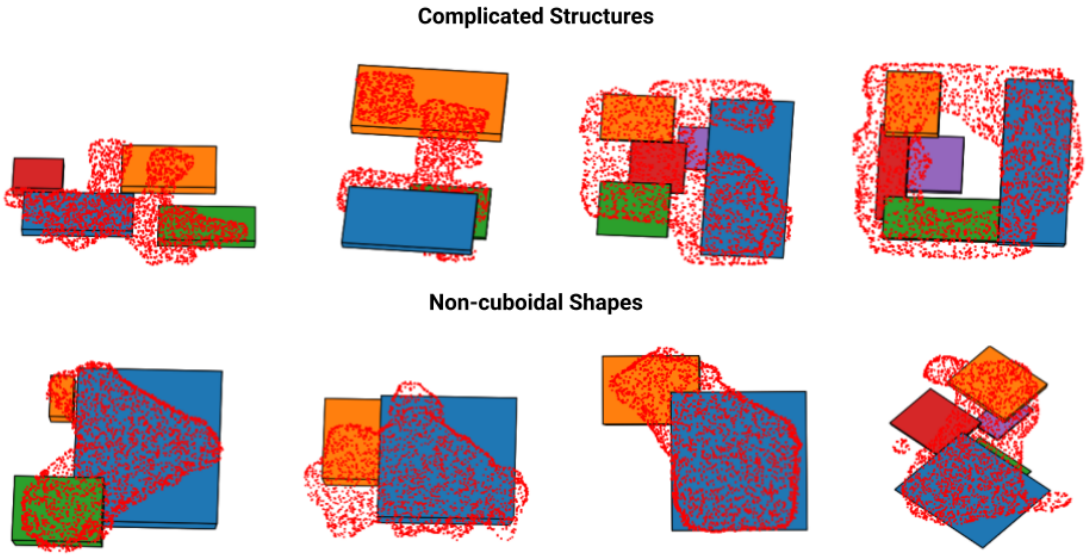

Figure 3.16: Some failure modes of the model when encountering challenging structures different from training data.

As can seen from the results in Figure 3.16 of some challenging real world scans, the network struggles to model complicated structures which are often composed of non-cuboidal base shapes. This is a limitation of the simplistic assumption made here of using only cuboids as the building blocks. Learning to leverage a larger library of primitive shapes (like the geons[5]) might enable better modeling of non-regular shapes. Also, the distribution of synthetic data used during training does not seem to sufficiently reveal the idiosyncracies revealed by real data. This mismatch of the data distributions is another reason for the model not being able to model the shape to a satisfactory level.

Chapter 4

Conclusion and Open Problems

In this thesis, we introduced techniques for leveraging geometry in the form of priors to enable learning based 3D reconstruction and consequently parsimonious representation. We first presented an unsupervised learning based approach for multi-view stereo reconstruction which utilized a novel robust photometric consistency objective to learn without access to 3D supervision. The proposed objective allows implicitly overcoming lighting changes and occlusions across training views, producing reasonably accurate depth maps. Next, we introduced an approach for representing complex 3D structures as collections of parameterized primitive shapes. We showed how such compositional representations can be learned in an unsupervised way by optimizing geometric fits. For certain man-made regular structures, we demonstrate how a synthetic-to-real supervised transfer learning approach can be utilized to encode priors efficiently.

While these are encouraging steps towards the goal of unsupervised learning of models for reconstructing and parsing the 3D world, this is only an initial attempt and further efforts are required to realize the potential of unsupervised methods for these tasks. We are however optimistic, as an unsupervised approach is more scalable as large amounts of training data can be more easily acquired. In addition, as our experiments demonstrated, these unsupervised methods can be used in conjunction with, and further allow us to improve over supervised methods, thereby allowing us to leverage both, the benefits of supervision along with the scalability of unsupervised methods. We also hope that the proposed loss formulations would be more broadly applicable for unsupervised 3D prediction approaches. We discuss some of the open problems below and highlight potentially interesting directions

for future work.

1. The promise of unsupervised learning is that we can exploit potentially infinite troves of data in order to learn stronger models. However, not sufficient data is publicly available in a form suitable to the models we used. Our MVS model assumed known camera parameters and, more importantly, the range of depth values of the scene of interest. Such assumptions prevent us from scaling the technique to free-form unstructured visual data. A better comparison benchmark would be when unsupervised methods are given access to significantly more data than the supervised methods. Curation of such a dataset and consequently training models at scale might reveal interesting characteristics of the unsupervised setup.
2. We restricted ourselves to learn blocks world representations where a block was defined to be strictly a cuboid. Our visual world is of course much richer and a larger library of primitive shapes is need in order to sufficiently represent it. Additionally, the inputs to the models we proposed assume the scene to be segmented apriori. Such precise segmentation is an open challenge in itself making it crucial to relax this assumption in future work.
3. While humans do not have access to 'ground truth' 3D models of the world, our visual learning systems function alongside our ability to conduct motor actions and interact with our environments to build mental models of the world over time. We often move and sense objects across a variety of positions, orientations and in a wide array of lighting conditions and locations. Such rich 3D data is currently unavailable in our benchmarking datasets. Learning visual representations by means of active exploration of the world is an exciting direction. While there has been some work on learning curiosity-driven agents[40] in the deep reinforcement learning literature, we are yet to it coupled with geometry for the tasks discussed in this thesis.

Bibliography

[1] How many and which axes to use for 3d obb collision with sat? https://gamedev.stackexchange.com/a/44501/116021. Accessed: 2019-05-20. (document), 3.3

[2] Henrik Aanæs, Rasmus Ramsbøl Jensen, George Vogiatzis, Engin Tola, and Anders Bjorholm Dahl. Large-scale data for multiple-view stereopsis. *International Journal of Computer Vision*, 120(2):153–168, 2016. (document), 2.2, 2.1, 2.2, 2.13

[3] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. Tensorflow: a system for large-scale machine learning. 2.3.5

[4] Connelly Barnes, Eli Shechtman, Adam Finkelstein, and Dan B Goldman. Patchmatch: A randomized correspondence algorithm for structural image editing. *ACM Transactions on Graphics (ToG)*, 28(3):24, 2009. 2.2

[5] Irving Biederman. Recognition-by-components: a theory of human image understanding. *Psychological review*, 94(2):115, 1987. 3.5

[6] I Binford. Visual perception by computer. In *IEEE Conference of Systems and Control*, 1971. 3

[7] 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. 3.3.4

[8] David Eberly. Dynamic collision detection using oriented bounding boxes, 1999. 3.3.3

[9] Haoqiang Fan, Hao Su, and Leonidas J 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*, pages 605–613, 2017. 3.3.3

[10] Yasutaka Furukawa and Jean Ponce. Accurate, dense, and robust multiview stereopsis. *IEEE transactions on pattern analysis and machine intelligence*, 32(8):1362–1376, 2010. 2.2, 2.3, 2.4.1, 2.1, 2.2

[11] Yasutaka Furukawa, Carlos Hernández, et al. Multi-view stereo: A tutorial. *Foun-*

dations and Trends® in Computer Graphics and Vision, 9(1-2):1–148, 2015. (document), 2.2, 2.2, 2.3

[12] Silvano Galliani, Katrin Lasinger, and Konrad Schindler. Massively parallel multiview stereopsis by surface normal diffusion. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 873–881, 2015. 2.2, 2.3.3, 2.3.6, 2.4.2

[13] Ravi Garg, G VijayKumarB., and Ian D. Reid. Unsupervised cnn for single view depth estimation: Geometry to the rescue. In *ECCV*, 2016. 2.2

[14] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3354–3361, 2012. 2.3.2

[15] Clément Godard, Oisin Mac Aodha, and Gabriel J Brostow. Unsupervised monocular depth estimation with left-right consistency. In *CVPR*, volume 2, page 7, 2017. 2.2, 2.3.4

[16] Michael Goesele, Noah Snavely, Brian Curless, Hugues Hoppe, and Steven M Seitz. Multi-view stereo for community photo collections. 2007. 2.2

[17] Abhinav Gupta, Alexei A Efros, and Martial Hebert. Blocks world revisited: Image understanding using qualitative geometry and mechanics. In *European Conference on Computer Vision*, pages 482–496. Springer, 2010. 3

[18] Xufeng Han, Thomas Leung, Yangqing Jia, Rahul Sukthankar, and Alexander C. Berg. Matchnet: Unifying feature and metric learning for patch-based matching. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3279–3286, 2015. 2.2

[19] Wilfried Hartmann, Silvano Galliani, Michal Havlena, Luc Van Gool, and Konrad Schindler. Learned multi-patch similarity. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 1595–1603. IEEE, 2017. 2.2

[20] Heiko Hirschmuller. Stereo processing by semiglobal matching and mutual information. *IEEE Transactions on pattern analysis and machine intelligence*, 30(2):328–341, 2008. 2.2

[21] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997. 3.3.2

[22] Xiaoyan Hu and Philippos Mordohai. Least commitment, viewpoint-based, multi-view stereo. In *3D Imaging, Modeling, Processing, Visualization and Transmission (3DIMPVT), 2012 Second International Conference on*, pages 531–538. IEEE, 2012. 2.2

[23] Po-Han Huang, Kevin Matzen, Johannes Kopf, Narendra Ahuja, and Jia-Bin Huang. Deepmvs: Learning multi-view stereopsis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2821–2830, 2018. 2.2, 2.3.1

[24] Max Jaderberg, Karen Simonyan, Andrew Zisserman, and Koray Kavukcuoglu. Spatial transformer networks. *CoRR*, abs/1506.02025, 2015. 2.3.2

[25] Michal Jancosek and Tomás Pajdla. Multi-view reconstruction preserving weakly-supported surfaces. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 3121–3128. IEEE, 2011. 2.2

[26] Rasmus Jensen, Anders Dahl, George Vogiatzis, Engil Tola, and Henrik Aanæs. Large scale multi-view stereopsis evaluation. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 406–413. IEEE, 2014. 2.4, 2.4.1

[27] Mengqi Ji, Juergen Gall, Haitian Zheng, Yebin Liu, and Lu Fang. Surfacenet: an end-to-end 3d neural network for multiview stereopsis. *arXiv preprint arXiv:1708.01749*, 2017. 2.2, 2.4.1, 2.4.1, 2.1, 2.4.1, 2.2

[28] Sing Bing Kang, Richard Szeliski, and Jinxiang Chai. Handling occlusions in dense multi-view stereo. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, volume 1, pages I–I. IEEE, 2001. 2.2

[29] Abhishek Kar, Christian Häne, and Jitendra Malik. Learning a multi-view stereo machine. In *Advances in neural information processing systems*, pages 365–376, 2017. 2.2, 2.3

[30] Alex Kendall, Matthew Grimes, and Roberto Cipolla. Posenet: A convolutional network for real-time 6-dof camera relocalization. In *Proceedings of the IEEE international conference on computer vision*, pages 2938–2946, 2015. 3.3.1

[31] Alex Kendall, Hayk Martirosyan, Saumitro Dasgupta, Peter Henry, Ryan Kennedy, Abraham Bachrach, and Adam Bry. End-to-end learning of geometry and context for deep stereo regression. *CoRR, vol. abs/1703.04309*, 2017. 2.2

[32] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014. 2.3.5

[33] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 3.3.4

[34] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013. 3.3.3

[35] Maria Klodt and Andrea Vedaldi. Supervising the new with the old: Learning SFM from SFM. In *ECCV (10)*, volume 11214 of *Lecture Notes in Computer Science*, pages 713–728. Springer, 2018. 2.2

[36] Arno Knapitsch, Jaesik Park, Qian-Yi Zhou, and Vladlen Koltun. Tanks and temples: Benchmarking large-scale scene reconstruction. *ACM Transactions on Graphics (ToG)*, 36(4):78, 2017. (document), 2.4.1, 2.1, 2.7, 2.11, 2.13

[37] H. W. Kuhn and Bryn Yaw. The hungarian method for the assignment problem. *Naval*

Res. Logist. Quart, pages 83–97, 1955. 3.4.3

[38] Yevhen Kuznietsov, Jörg Stückler, and Bastian Leibe. Semi-supervised deep learning for monocular depth map prediction. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2215–2223, 2017. 2.2

[39] Reza Mahjourian, Martin Wicke, and Anelia Angelova. Unsupervised learning of depth and ego-motion from monocular video using 3d geometric constraints. In *CVPR*, 2011. 2.3.2, 2.3.4, 2.3.4

[40] Deepak Pathak, Pulkit Agrawal, Alexei A Efros, and Trevor Darrell. Curiosity-driven exploration by self-supervised prediction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 16–17, 2017. 3

[41] 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*, pages 652–660, 2017. 3.3.2

[42] L Roberts. Machine perception of 3-d solids, optical and electro-optical information processing, 1965. 3

[43] Johannes L Schönberger, Enliang Zheng, Jan-Michael Frahm, and Marc Pollefeys. Pixelwise view selection for unstructured multi-view stereo. In *European Conference on Computer Vision*, pages 501–518. Springer, 2016. 2.2

[44] Thomas Schöps, Johannes L. Schönberger, Silvano Galliani, Torsten Sattler, Konrad Schindler, Marc Pollefeys, and Andreas Geiger. A multi-view stereo benchmark with high-resolution images and multi-camera videos. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 2.4.2

[45] Thomas Schöps, Johannes L Schönberger, Silvano Galliani, Torsten Sattler, Konrad Schindler, Marc Pollefeys, and Andreas Geiger. A multi-view stereo benchmark with high-resolution images and multi-camera videos. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2017, 2017. 2.2

[46] Steven M Seitz, Brian Curless, James Diebel, Daniel Scharstein, and Richard Szeliski. A comparison and evaluation of multi-view stereo reconstruction algorithms. In *null*, pages 519–528. IEEE, 2006. 2.2

[47] Shuhan Shen. Accurate multiple view 3d reconstruction using patch-based stereo for large-scale scenes. *IEEE transactions on image processing*, 22(5):1901–1914, 2013. 2.2

[48] Engin Tola, Christoph Strecha, and Pascal Fua. Efficient large-scale multi-view stereo for ultra high-resolution image sets. *Machine Vision and Applications*, 23:903–920, 2011. 2.4.1, 2.1, 2.2

[49] Shubham Tulsiani, Hao Su, Leonidas J. Guibas, Alexei A. Efros, and Jitendra Malik. Learning shape abstractions by assembling volumetric primitives. In *Computer Vision*

and Pattern Regognition (CVPR), 2017. 3, 3.3.1, 3.3.2, 3.3.3, 3.3.3

[50] Kaixuan Wang and Shaojie Shen. Mvdepthnet: Real-time multiview depth estimation neural network. In *2018 International Conference on 3D Vision (3DV)*, pages 248–257. IEEE, 2018. 2.2, 2.3.1

[51] Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3-4):229–256, 1992. 3.3.2

[52] Junyuan Xie, Ross B. Girshick, and Ali Farhadi. Deep3d: Fully automatic 2d-to-3d video conversion with deep convolutional neural networks. In *ECCV*, 2016. 2.2

[53] Yao Yao, Zixin Luo, Shiwei Li, Tian Fang, and Long Quan. Mvsnet: Depth inference for unstructured multi-view stereo. *CoRR*, abs/1804.02505, 2018. URL http://arxiv.org/abs/1804.02505. (document), 2.2, 2.3.1, 2.3.1, 2.3.5, 2.4.1, 2.4.1, 2.1, 2.4.1, 2.2, 2.8, 2.9, 2.10

[54] Christopher Zach. Fast and high quality fusion of depth maps. In *Proceedings of the international symposium on 3D data processing, visualization and transmission (3DPVT)*, volume 1. Citeseer, 2008. 2.2

[55] Jure Zbontar and Yann LeCun. Stereo matching by training a convolutional neural network to compare image patches. *Journal of Machine Learning Research*, 17(1-32): 2, 2016. 2.2

[56] Runze Zhang, Shiwei Li, Tian Fang, Siyu Zhu, and Long Quan. Joint camera clustering and surface segmentation for large-scale multi-view stereo. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2084–2092, 2015. 2.4.1

[57] Yinda Zhang, Sameh Khamis, Christoph Rhemann, Julien P. C. Valentin, Adarsh Kowdle, Vladimir Tankovich, Michael Schoenberg, Shahram Izadi, Thomas A. Funkhouser, and Sean Ryan Fanello. Activestereonet: End-to-end self-supervised learning for active stereo systems. *CoRR*, abs/1807.06009, 2018. 2.2

[58] Enliang Zheng, Enrique Dunn, Vladimir Jojic, and Jan-Michael Frahm. Patchmatch based joint view selection and depthmap estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1510–1517, 2014. 2.2

[59] Yiran Zhong, Yuchao Dai, and Hongdong Li. Self-supervised learning for stereo matching with self-improving ability. *CoRR*, abs/1709.00930, 2017. 2.2

[60] Tinghui Zhou, Matthew Brown, Noah Snavely, and David G. Lowe. Unsupervised learning of depth and ego-motion from video. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6612–6619, 2017. 2.2, 2.3.2, 2.3.2, 2.3.4