LEARNING VIEWER-CENTERED PROJECTIONS FOR 3D SHAPE COMPLETION

BY

DAEYUN SHIN

THESIS

Submitted in partial fulfillment of the requirements
for the degree of Master of Science in Computer Science
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2017

Urbana, Illinois

Adviser:

Associate Professor Derek Hoiem

ABSTRACT

The goal of this study is to determine the effectiveness of different 3D shape representations in learning to generate volumetric shapes using deep neural networks. We propose to automatically reconstruct a 3D model from a single-view image of an object by synthesizing multiple depth images and inferring the volume through multi-view 3D reconstruction. The final output is a 3D mesh inferred without seeing voxels in the training process. This is similar to the intuition that humans remember (and inherently reproduce) 3D shapes without ever "seeing through" the underlying volume – we think of objects as seen from certain viewpoints and 3D structure is a derived concept. Most previous studies have focused on directly learning the voxel representations, deforming exemplars, or utilizing user interaction. In this paper, we want to learn category-independent object shape representations by simultaneously predicting multiple incomplete surfaces in relation to the viewer with the complete 3D structure in mind. Instead of predicting voxels which typically need to be in low resolution, we hypothesize that learning a representation that can consistently produce partial surfaces in a multi-task learning model enables inter-category 3D shape transfer. We perform shape completion in novel categories and evaluate quantitatively using voxel I/U and surface distance metrics. We also report that the learned representation improves 3D shape classification.

ACKNOWLEDGMENTS

I cannot express enough how deeply grateful I am for the support and encouragement I received from my advisor Derek Hoiem. I would not have had the opportunity to pursue graduate school and grow as a researcher without his guidance and understanding. I would also like to thank the members of the UIUC vision group for discussion and feedback – in particular Jason Rock and Tanmay Gupta, for the help in getting started in vision research. I am also thankful to my parents for their love and encouragement.

TABLE OF CONTENTS

CHAPTER 1: INTRODUCTION

1.1 OVERVIEW

The human visual system allows us to effortlessly construct a picture of the world from very little input, often in the form of inferring higher dimensions from lower dimensional representations. We understand 2D surfaces as lines, 3D structure as surfaces, and motion and spatial relations through 3D structures. The motivation for this paper is to automatically infer the spatial context outside the image domain by bridging the gap between such representational dualities in our visual world — in particular, learning to generate 3D shapes through projective supervision. We consider the problem of recovering the 3D shape of an object from a single depth image and silhouette under the assumption that the underlying volume is not available at training time.

Reconstructing the complete 3D shape of an object from one or a few images (e.g. predicting the four legs of a table when only the top portion is visible) is considered ill-posed in general without strong contextual cues due to self-occlusions and ambiguities in the object's pose and scale. A common formulation is to predict binary labels in a volumetric grid using synthetic datasets. In contrast, humans rely on view-based observations to derive 3D shapes without ever "seeing through" the underlying volume. We are interested in evaluating the two aforementioned representations on volumetric shape transfer across unseen object views, instances, and categories.

Studies on the human visual system suggest that humans process 3D information through viewer-centered representations by remembering many different viewpoints of an object and representing each view as a 2.5D sketch. To apply this idea to 3D shape generation, we train a neural net to output 3D shapes represented as multiple depth images and silhouettes. In addition, we observe the general problem that 3D pose alignment is not always uniquely defined across categories. As

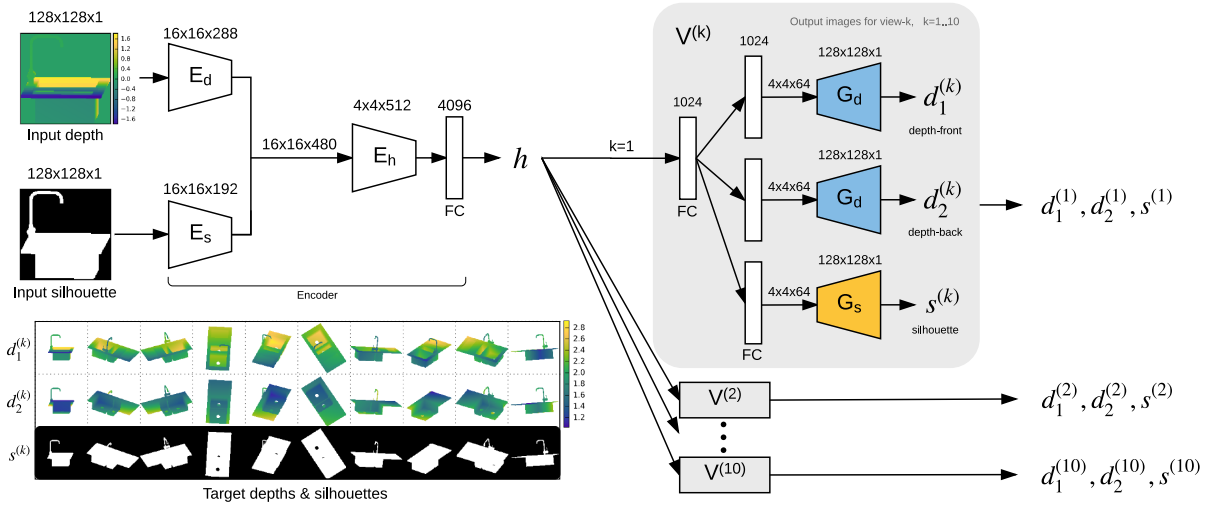


Figure 1.1: Network architecture: Encoders E_d, E_s, E_h learn view-specific shape features h extracted from the input depth and silhouette. h is used by the 10 output branches $V^{(k)}$, k=1..10 each outputting one silhouette and two, front and back, depth images. They have their own fully connected layers, but the up-convolutional decoders G_d, G_s have parameters shared by all output branches.

addressed in the work of Tulsiani *et al.* [1], cross-category pose induction is a difficult task on its own, so we choose a viewer-centered frame of reference for the 3D representation (rather than an object-centered or shared coordinate system). This is also related to biological findings that the human visual reference frame is tied to viewer-centered coordinates [2]. As a result, our network learns a view-specific 3D representation which can potentially generalize better to novel object categories because new shapes can be interpolated from alignment-independent representations. In summary, our formulation is viewer-centered in that: (1). we learn to generate 3D shapes represented as 2.5D projections (2). in a reference frame relative to the viewer.

1.2 RELATED WORK

Although we mainly focus on learning depth images, our approach is inspired by the success of texture-based image generation techniques.

1.2.1 Generating images

Dosovitskiy *et al*. [3] showed that CNNs can be used to interpolate new images from high-level descriptions such as object instance, viewpoint, and transformation parameters. Their network jointly predicts an RGB image and its segmentation mask using two up-convolutional output branches sharing a high-dimensional hidden representation. The decoder in our network learns the segmentation for each output view in a similar manner to this paper. A related problem is single-view depth prediction, however in our setting, we predict multiple unseen depth images given an observed depth image as input.

1.2.2 Volumetric representations

Generating volumetric object shapes from one or a few images has been explored in many recent studies [4, 5, 6, 7, 8]. Wu *et al*. [4] proposed a convolutional deep belief network for learning 3D representations using volumetric supervision and evaluated applications in various recognition tasks – while other studies also quantitatively evaluated 3D reconstruction results. Metrics include voxel I/U [6, 7, 8], mesh distances [5, 6], and depth map errors [5]. Some [5, 6] have taken template deformation approaches using surface rigidity [5, 6] and symmetry [6] priors, while others [4, 7, 8] have done deep representation learning using encoder-decoder networks. In our approach, we use encoder-decoder networks to generate segmented depth images and then obtain the final volume using existing multi-view reconstruction pipelines.

1.2.3 Multi-view representations

Representing 3D shape as a set of 2D projections has been successful in discriminative tasks. The seminal work by Chen *et al*. [9] proposed a 3D shape descriptor based on the silhouettes rendered from the 20 vertices of a dodecahedron surrounding the object. In a more recent

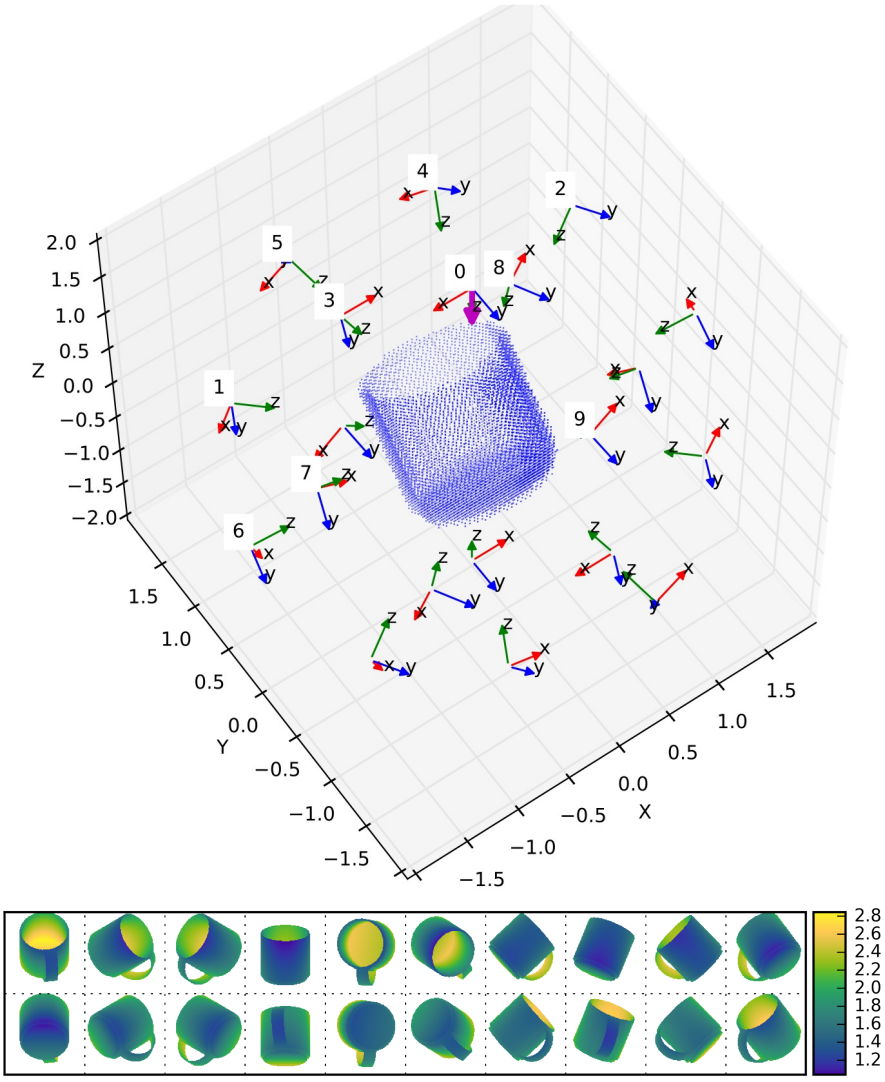


Figure 1.2: Illustration of our dataset. Top: GT mesh voxelized in camera coordinates; used in the voxel baseline experiment. The numeric labels around it are the indices of the viewpoints from which the multi-view depth maps (shown above) were rendered, left to right. Viewpoint-0, whose camera is located at the origin, *always* corresponds to the input view, and the relative transformation from Viewpoint-0 to all the other viewpoints are constant throughout all experiments in our work. Bottom: Multi-view projections. This method does not require alignment between the 3D models and allows the network to be trained in an unsupervised manner — at the cost of increased dataset size.

line of work, Su *et al*.and Qi *et al*.[10, 11] train CNNs on 2D renderings of 3D mesh models for classification. Qi *et al*. [11] compared CNNs trained on volumetric representations versus CNNs trained on multi-view representations. Although both representations encode similar amounts of information, they showed that multi-view representations significantly outperform volumetric representations for 3D object classification. Unlike our approach, these approaches use projections as input rather than output.

Our work is similar to recent studies [12, 13, 14, 15, 8] that generate multi-view projections of 3D objects. The multi-view perceptron by Zhu *et al*. [15] is a network that generates one random view at a time, given an RGB image and a random vector as input. Another way to formulate this is through recurrent rotation [14], inspired by the mental rotation ability in human vision — Yang *et al*. [14] proposed a recurrent encoder-decoder network that outputs RGB images rotated by a fixed angle in each time step along a path of rotation, given an image at the beginning of the rotation sequence as input. They disentangle object identity and pose by sharing the identity unit weights across all time steps. Their experiments did not include 3D reconstruction. We observe that, for the purpose of 3d reconstruction, it is important to be able to see the object from certain viewpoints – e.g. classes such as cup and bathtub need at least one view from the top to cover the concavity. Our proposed method therefore predicts evenly spaced views around the object. Similarly to [9]'s 3D shape descriptor, we place the cameras on the 20 vertices of a dodecahedron. Unlike our setting, [12, 13, 8] parametrized the output image as (x, θ) where x is the input image and θ is the desired viewpoint. The recent work of Yan *et al*. [8] concurrent to ours introduces a formulation that indirectly learns to generate voxels through silhouettes using multi-view projective constraints. They report that voxel I/U performance was better when the network was trained to minimize projection loss alone, compared to when jointly trained with volumetric

loss. Unlike their formulation, our proposed approach uses multi-view reconstruction pipelines to obtain the volume, treating any inconsistencies in the output images as if they were observational noise.

Our formulation differs in that we learn a view-specific representation, and the object's shape identity is enforced by simultaneously predicting multiple views.

1.2.4 Reconstruction

We convert the predicted depth images to a triangle mesh using Poisson Reconstruction [16] and FSSR [17]. Both methods are widely used in computer graphics for surface reconstruction from oriented points. In a typical setting, the input points and their oriented normals would be derived from naturally observed depth images, such as the ones captured by depth sensors. Our experiments are unique in that surface reconstruction methods are used to deal with noise in images generated by neural networks rather than observational data noise. In addition, we also experiment with using both the observed surface and predicted surface as input.

CHAPTER 2: APPROACH

2.1 OVERVIEW

Given a 2.5D observation of an object, we want to predict the complete 3D shape represented as multiple 2.5D views. The proposed approach is a multi-task learning model where predicting each view is treated as a different task. A single training example in our dataset consists of orthographic depth images from 20 viewpoints around a sphere centered at the origin, to serve as the ground truth, in addition to one normalized input depth image. Viewpoint and category are assumed unknown.

Each view in our setting has a corresponding segmented silhouette and another view on the opposite side, thus only 10 out of the 20 silhouettes need to be predicted due to symmetry. The network therefore outputs a silhouette and corresponding front and back depth images $\{s^{(i)}, d_f^{(i)}, d_b^{(i)}\}$ in the i-th output branch.

Note that the observed viewpoint is not known in our setting, however the relative transformations to all of the output viewpoints are predefined. This is so that we can control the predicted shapes to be interpolated solely based on the 2.5D shape input in a setting where contextual cues are not available. The network can be trained in an unsupervised manner across multiple categories because it does not require alignment between 3D models.

2.2 COORDINATES

The input depth image is normalized so that the bounding box of the silhouette fits inside an orthographic viewing frustum ranging from $\langle$-1,-1$\rangle$ to $\langle 1,1\rangle$ and the depth values are centered at the centroid.

The output 3D shape is represented as depth images with evenly spaced viewpoints around the object. Similarly to the setting in the Light Field Descriptor [9], we place the cameras at the 20 vertices $\{\mathbf{v}_0,..,\mathbf{v}_{19}\}$ of a dodecahedron centered at the origin, and consequently, the camera positions can be parametrized as a rotation on the vertex coordinates $\langle\pm1,\pm1,\pm1\rangle$, $\langle 0,\pm1/\phi,\pm\phi\rangle$, $\langle\pm1/\phi,\pm\phi,0\rangle$, $\langle\pm\phi,0,\pm1/\phi\rangle$ where $\phi\approx 1.618$ is the golden ratio. In order to determine the 20 camera parameters, we rotate the vertices by spherical angles (ϑ,φ) so that vertex $\mathbf{v}_0=\langle 1,1,1\rangle$ aligns with the input viewpoint in the object's model coordinates. The up-vectors point along the z-axis and are rotated accordingly.

2.3 GENERATING MULTI-VIEW DEPTHS AND SILHOUETTES

As illustrated in Figure 1.1, we train a CNN to encode the depth image and silhouette into features h. Note that the input viewpoint $\mathbf{v}_0$ is not known in our setting, but the relative transformations from $\mathbf{v}_0$ to all of the output viewpoints are known and fixed.

The encoder units (E_d, E_s, E_h) consist of bottleneck residual layers (Table **??**). We use two generic decoders (Table A.2) to generate the views, one for all depths and another for all silhouettes. ReLU and batch normalization layers are not shown in the figures. Similarly to Dosovitskiy *et al*. [3], we minimize the objective function

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

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

Shape completion requires fusing the observed surface and the predicted 3D shape into a common coordinate system. After predicting the images. We transform the local 2.5D points into

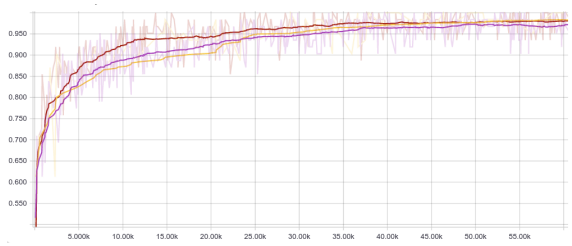
the coordinates of the observed viewpoint $\mathbf{v}_0$ to obtain the oriented points P_{pred} used as input to the reconstruction pipeline.

Observing that a considerable portion of the shape is already observed and accurately known, we also experiment with replacing the known portion of P_{pred} with the observed points. Note however that the alignment is not exactly known because the input depth image is normalized, as described in 2.2. So we need to predict the scale and (x,y,z) offsets used in the normalizing transformation. We achieve this by predicting the centroid of the input depth image in the output coordinate system as well as the mean distance from the centroid. After finding the alignment, any overlapping points in the prediction P_{pred} are replaced. An observed point $\mathbf{p}_i$ replaces any point $\mathbf{p}_j \in P_{\text{pred}}$ if $|\mathbf{p}_i - \mathbf{p}_j|_2 < \theta_d$ and $\mathbf{n}_j \cdot \mathbf{v} > 0$ where $\mathbf{n}_j$ is the inner surface normal at $\mathbf{p}_j$, $\mathbf{v}$ is the viewing direction, and θ_d is a constant threshold 0.05.

2.4 GENERATING VOXELS

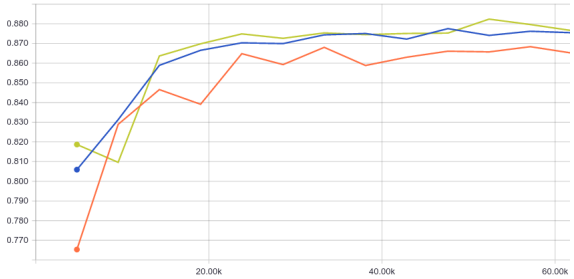
We compare our multi-view reconstruction method with a baseline that directly predicts a 3D voxel grid. Given a single-view depth image of an object, the "Voxels" baseline learns to generate a grid of 3D occupancy mappings in the camera coordinates of viewpoint $\mathbf{v}_0$. The cubic window of length 2 centered at $\langle 0,0,1 \rangle$ is voxelized after camera transformation. As described in Table A.3, the encoded features h feed into 3D up-convolutional layers outputting a single $48 \times 48 \times 48$ volumetric grid. The network is trained from scratch to minimize the logistic loss $\mathbf{L}_v$ over the binary voxel occupancy labels. Both networks are implemented using TensorFlow.

Training accuracy



Multi-view (red), Voxels (yellow), Classification-only (purple)

Test accuracy



Multi-view (green), Voxels (blue), Classification-only (orange)

Model	Accuracy
Classifier-only	86.5%
Multi-view + Classifier	87.7%
Voxels + Classifier	87.6%

Figure 2.1: 2.5D shape classification results using projective and volumetric supervision on ModelNet40 dataset.

2.5 2.5D SHAPE CLASSIFICATION

We evaluate the usefulness of the learned features (h in Figure 1.1) on single-view shape recognition tasks. In this experiment, the output of the encoder h is treated as a constant and used as input to an MLP classifier with one hidden layer, as described in Table A.4). We compare this to a baseline trained from scratch, using the same encoder units outputting h, followed by the classification layer.

CHAPTER 3: EXPERIMENTS

3.1 DATASET

A training example in the Multi-view dataset is the pair $(x_d, x_s), \{(s^{(k)}, d_f^{(k)}, d_b^{(k)})\}_{k=0..9}$ where (x_d, x_s) is the observed segmented depth image and $(s^{(k)}, d_f^{(k)}, d_b^{(k)})$ is the k-th ground truth silhoouette with the associated front and back depth images. The "Voxels" baseline dataset has an example pair $(x_d, x_s), V$ where V is the 48x48x48 grid of ground truth voxels.

We use the SHREC'12 dataset for comparison with the exemplar retrieval approach by Rock *et al*. [6] on predicting novel views, models, and classes. This dataset has a shared training set which consists of 28500 (22500 training + 6000 validation) examples and has 600 examples in each of the evaluation sets. The splits are equivalent to the ones used by Rock *et al*. [6].

We also perform novel model, novel class, and 2.5D classification experiments using the ModelNet40 dataset which consists of 191820 training examples rendered from 6394 3D models in 40 object categories.

3.2 RECONSTRUCTION

In order to quantitatively evaluate 3D shape prediction, we use two metrics: voxel intersection-over-union and surface distances.

3.2.1 Voxel I/U

Given a mesh reconstructed from the multi-view prediction, we obtain a solid representation by voxelizing the mesh surface into a hollow volume and then filling in the holes using 3D flood fill and ray tracing; all voxels not visible from the outside are filled. Then we compute intersection-over-union with the corresponding ground truth voxels.

Mean	Surface Distance			Voxel I/U		
	NovelClass	NovelModel	NovelView	NovelClass	NovelModel	NovelView
Voxels (Marching Cubes)	0.0950	0.0619	0.0512	0.4569	0.5176	**0.6969**
Multi-view (FSSR)	**0.0759**	0.0622	**0.0494**	0.4914	0.5244	0.6501
Multi-view (Poisson)	0.0842	0.0685	0.0550	0.4921	0.5164	0.6538
Rock *et al*.	0.0827	**0.0604**	0.0639	**0.5320**	**0.5888**	0.6374

Median	Surface Distance			Voxel I/U		
	NovelClass	NovelModel	NovelView	NovelClass	NovelModel	NovelView
Voxels (Marching Cubes)	0.0856	0.0550	0.0438	0.4605	0.5103	**0.7297**
Multi-view (FSSR)	**0.0687**	**0.0496**	**0.0419**	0.4881	0.5133	0.6692
Multi-view (Poisson)	0.0777	0.0578	0.0499	0.4986	0.5095	0.6824
Rock *et al*.	0.0766	0.0529	0.0580	**0.5560**	**0.6076**	0.6582

Std.	Surface Distance			Voxel I/U		
	NovelClass	NovelModel	NovelView	NovelClass	NovelModel	NovelView
Voxels (Marching Cubes)	0.0405	0.0328	0.0304	0.1930	0.2105	0.1654
Multi-view (FSSR)	0.0311	0.0477	0.0332	0.1772	0.1932	0.1592
Multi-view (Poisson)	0.0334	0.0479	0.0351	0.1769	0.2037	0.1698
Rock *et al*.	0.0340	0.0377	0.0419	0.1594	0.1823	0.1867

Table 3.1: Comparison of single-depth 3D shape completion methods on the SHREC'12 dataset.

Mean	Surface Distance			Voxel I/U		
	NovelClass	NovelModel	NovelView	NovelClass	NovelModel	NovelView
Voxels (Marching Cubes)	**0.0289**	**0.0248**	0.0284			
Multi-view (FSSR)	0.0302	0.0322	**0.0241**	**0.8734**	**0.8596**	**0.9016**
Multi-view (Poisson)	0.0384	0.0493	0.0426	0.8197	0.7389	0.8313

Median	Surface Distance			Voxel I/U		
	NovelClass	NovelModel	NovelView	NovelClass	NovelModel	NovelView
Voxels (Marching Cubes)	0.0249	0.0210	0.0243			
Multi-view (FSSR)	**0.0150**	**0.0122**	**0.0143**	**0.9266**	**0.9067**	**0.9406**
Multi-view (Poisson)	0.0290	0.0282	0.0275	0.8782	0.8141	0.9102

Std.	Surface Distance			Voxel I/U		
	NovelClass	NovelModel	NovelView	NovelClass	NovelModel	NovelView
Voxels (Marching Cubes)	0.0129	0.0135	0.0147			
Multi-view (FSSR)	0.0493	0.0648	0.0334	0.1453	0.1614	0.1008
Multi-view (Poisson)	0.0395	0.0719	0.0567	0.1722	0.2402	0.1745

Table 3.2: Comparison of 3D reconstruction methods (from GT depths and voxels) on the SHREC'12 dataset. This is the expected upper bound due to reconstruction methods for shapes reconstructed from neural network predictions. This can also be seen as a measure of relative reconstruction difficulty for each test set.

3.2.2 Surface Distance

We use a surface distance metric similar to [6]. The distance between surfaces is approximated as the mean of *point-to-triangle* distances from i.i.d. sampled points on the ground truth mesh to the closest points on surface of the reconstructed mesh, and vice versa. In order to ensure scale invariance between datasets, we divide the resulting value by the mean distance between points sampled on the GT surface. The points were sampled at a density of 300 points per area. For the voxel prediction experiment, we use Marching Cubes to obtain the mesh.

3.3 TRAINING

We train our models, described in chapter 2 and Appendix A, using Adam optimizer with learning rate 0.001 and batch size 70 for 90 epochs. All of NovelClass, NovelModel, and NovelView experiments share the same training set.

CHAPTER 4: DISCUSSION

4.1 RECONSTRUCTION

We have found that FSSR produces both qualitatively and quantitatively better results for reconstructing 3D shapes from predicted depth images, and that the quality is often better reflected by surface distances than voxel I/U. As shown in Table 3.1, there is a distinct gap between the two methods in "surface distance", while the difference is insignificant in voxel I/U. However, our solid reconstructions are obtained by voxelizing the resulting mesh, so some loss of precision is expected. More reliable results may be obtained by extracting the isosurface from the octree and converting to voxels without going through mesh triangulation and hole filling.

We also observe that it is generally very difficult to learn and reconstruct thin object parts, such as the legs of chairs and tables. This problem is also mentioned as one of the difficulties in [7]. Reconstruction from oriented points usually requires surface normals from all surrounding directions, which is difficult to achieve in thin object parts due to the sparsity of available data points. This is also a learning problem because salient object parts are often quantitatively less significant compared to the rest of the object.

4.2 LEARNING

As shown in Table 3.1, "Multi-view" performs better or equivalent to "Voxels" in all three experiments using both metrics – with the exception of NovelView I/U. We observe that, in NovelClass, "Multi-view" outperforms "Voxels" by a larger margin compared to NovelView. Rock *et al.*'s exemplar deformation approach is shown to be most effective for NovelModel.

In Figure 2.1, training a classifier from the learned features h improved accuracy by around 1% compared to the baseline. The improvements were greater in our initial experiments, but using

a ResNet-based encoder improved the baseline results and reduced the gap.

4.3 CONCLUSION

We presented an approach for learning 3D volumetric shapes without seeing voxels in the training process. We have shown that our approach is effective in predicting the shapes of novel-class objects and performs better or is competitive to a voxel-based approach in novel-view and novel-model predictions.

One direction for future work is to jointly predict other image-based representations such as texture and surface normal maps. Other directions are to use RGB images as input, as well as incorporating an optimization step before running reconstruction to resolve conflicting predictions.

REFERENCES

[1] S. Tulsiani, J. Carreira, and J. Malik, "Pose induction for novel object categories," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 64–72, 2015.

[2] M. J. Tarr and S. Pinker, "When does human object recognition use a viewer-centered reference frame?," *Psychological Science*, vol. 1, no. 4, pp. 253–256, 1990.

[3] A. Dosovitskiy, J. Tobias Springenberg, and T. Brox, "Learning to generate chairs with convolutional neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1538–1546, 2015.

[4] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, and J. Xiao, "3d shapenets: A deep representation for volumetric shapes," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1912–1920, 2015.

[5] A. Kar, S. Tulsiani, J. Carreira, and J. Malik, "Category-specific object reconstruction from a single image," in *Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference on*, pp. 1966–1974, IEEE, 2015.

[6] J. Rock, T. Gupta, J. Thorsen, J. Gwak, D. Shin, and D. Hoiem, "Completing 3d object shape from one depth image," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2484–2493, 2015.

[7] C. B. Choy, D. Xu, J. Gwak, K. Chen, and S. Savarese, "3d-r2n2: A unified approach for single and multi-view 3d object reconstruction," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2016.

[8] X. Yan, J. Yang, E. Yumer, Y. Guo, and H. Lee, "Perspective transformer nets: Learning single-view 3d object reconstruction without 3d supervision," in *Advances In Neural Information Processing Systems 29*, 2016.

[9] D.-Y. Chen, X.-P. Tian, Y.-T. Shen, and M. Ouhyoung, "On visual similarity based 3d model retrieval," in *Computer graphics forum*, vol. 22, pp. 223–232, Wiley Online Library, 2003.

[10] H. Su, S. Maji, E. Kalogerakis, and E. Learned-Miller, "Multi-view convolutional neural networks for 3d shape recognition," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 945–953, 2015.

[11] C. R. Qi, H. Su, M. Nießner, A. Dai, M. Yan, and L. Guibas, "Volumetric and multi-view cnns for object classification on 3d data," in *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*, 2016.

[12] M. Tatarchenko, A. Dosovitskiy, and T. Brox, "Multi-view 3d models from single images with a convolutional network," in *European Conference on Computer Vision*, pp. 322–337, Springer, 2016.

[13] T. D. Kulkarni, W. F. Whitney, P. Kohli, and J. Tenenbaum, "Deep convolutional inverse graphics network," in *Advances in Neural Information Processing Systems 28*, pp. 2539–2547, 2015.

[14] J. Yang, S. E. Reed, M.-H. Yang, and H. Lee, "Weakly-supervised disentangling with recurrent transformations for 3d view synthesis," in *Advances in Neural Information Processing Systems*, pp. 1099–1107, 2015.

[15] Z. Zhu, P. Luo, X. Wang, and X. Tang, "Multi-view perceptron: a deep model for learning face identity and view representations," in *Advances in Neural Information Processing Systems*, pp. 217–225, 2014.

[16] M. Kazhdan and H. Hoppe, "Screened poisson surface reconstruction," *ACM Transactions on Graphics (TOG)*, vol. 32, no. 3, p. 29, 2013.

[17] S. Fuhrmann and M. Goesele, "Floating scale surface reconstruction," *ACM Transactions on Graphics (TOG)*, vol. 33, no. 4, p. 46, 2014.

APPENDIX A: NETWORK ARCHITECTURE

layer	output size	kernel	stride	repeats
Input depth	128x128x1			1
conv2d	64x64x48	7	2	1
residual unit	64x64x48	3	1	3
residual unit	32x32x144	3	2	1
residual unit	32x32x144	3	1	3
residual unit	16x16x288	3	2	1

layer	output size	kernel	stride	repeats
Input silhouette	128x128x1			1
conv2d	64x64x32	7	2	1
residual unit	64x64x32	3	1	3
residual unit	32x32x96	3	2	1
residual unit	32x32x96	3	1	3
residual unit	16x16x192	3	2	1

layer	output size	kernel	stride	repeats
residual unit	16x16x480	3	1	10
residual unit	8x8x256	3	2	1
residual unit	8x8x256	3	1	4
residual unit	4x4x512	3	2	1
residual unit	4x4x512	3	1	4
FC $\rightarrow h$	4096			1

Table A.1: Top to bottom: Network parameters for encoders E_d, E_s, E_h.

layer	output size	kernel	stride	repeats
upconv2d	8x8x256	5	2	1
upconv2d	16x16x128	5	2	1
upconv2d	32x32x64	5	2	1
upconv2d	64x64x32	5	2	1
upconv2d	128x128x1	5	2	1

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

layer	output size	kernel	stride	repeats
$h \rightarrow$ FC	3x3x3x64			1
upconv3d	6x6x6x512	5	2	1
upconv3d	12x12x12x256	5	2	1
upconv3d	12x12x12x128	5	1	1
upconv3d	24x24x24x64	5	2	1
upconv3d	48x48x48x1	5	2	1

Table A.3: Network parameters for the volume decoder used in the voxels baseline.

layer	output size	repeats
$h \rightarrow$ FC	256	1
FC, softmax	40	1

Table A.4: Network parameters for the classifier used in the 2.5D shape classification experiment. Unlike the experiments in [10, 11], the input is a single-view depth image and the whole 3D shape is only used at training time.