Explorative 3D Reconstruction

Zhengyuan Dong
University of Michigan
Ann Arbor, MI
dongzy@umich.edu

Cameron Husted
University of Michigan
Ann Arbor, MI
cehusted@umich.edu

Nathan Louis
University of Michigan
Ann Arbor, MI
natlouis@umich.edu

Hanwen Miao
University of Michigan
Ann Arbor, MI
hwmiao@umich.edu

Zeyu Sun
University of Michigan
Ann Arbor, MI
zeyusun@umich.edu

Abstract

Obtaining three-dimensional information from two-dimensional visual data with no inherent depth component is a fundamental problem of computer vision and has been a challenge for decades. 3D reconstruction plays a crucial role in vision and robotics tasks such as detection, navigation and manipulation, and has thus become an increasingly popular area of study in recent years. While most existing algorithms focus on reconstruction based on one or more fixed input views, it is normal for an embodied agent to move around and choose its observations of the object. We therefore address the problem of intelligently choosing views for optimal 3D reconstruction where an agent understands which positions it would need to view a novel object from in order to best understand its shape. We designed a framework that achieves this in an iterative manner. At each time step, it predicts the areas of high uncertainty in the current reconstruction, then obtains a new observation from this area and updates the reconstruction. We demonstrate the effectiveness of our method on the watercraft class of the ShapeNet dataset in two experiments. We hope our model can quantitatively reason about the importance of different views to reconstruction quality and inspire the design of next-generation agent-aware benchmarks.

1. Introduction

3D shape reconstruction aims to infer the 3D geometry and structure of objects from single or multiple images. It has been a fundamental vision problem that finds its way into many important applications such as object detection and robot navigation. Although tremendous progress has been made to tackle this problem, especially after the advent of deep learning, performing 3D reconstruction in an active manner still remains a difficult problem as it involves the viewpoint planning on top of the reconstruction.

In this paper, we present a learning-based framework that tackles the 3D multi-view reconstruction problem in a dynamic, explorative manner. Our approach is motivated by the observation that humans tend to gravitate towards the uncertain parts to learn the shape of an object. If we consider an agent learning to reconstruct an object, we would imagine it will benefit from collecting more images on regions that require additional information. From this intuition, we would like to train an agent to reconstruct a 3D model by making intelligent decisions on which views of the object it needs. To plan a camera path to learn a 3D shape efficiently, the agent must learn which perspective/view of the target object is most necessary to increase its confidence of its reconstructed object, which we denote as the Next Best View (NBV).

This work can be divided into two primary components: multi-view 3D reconstruction and NBV prediction. For the multi-view 3D reconstruction part, our work is motivated by three recent papers: Mesh R-CNN[4], Pix2Vox[13] and Pixel2Mesh++[12]. Mesh R-CNN[4] can do mesh reconstruction from one single image. Pix2Vox[13] can do voxel reconstruction from one or more images. For this work we use those existing works unchanged and limit our experiments in optimizing our next best view module.

Regarding the prediction of the Next Best View, our work is motivated by two most similar works to this problem, by Ramakrishnan et al. [8] and Seifi et al [10]. [8] uses reinforcement learning to decide which viewpoints are most informative for 3D scene and object reconstruction tasks. [10] learns which viewpoint in a 360° image is most infor-

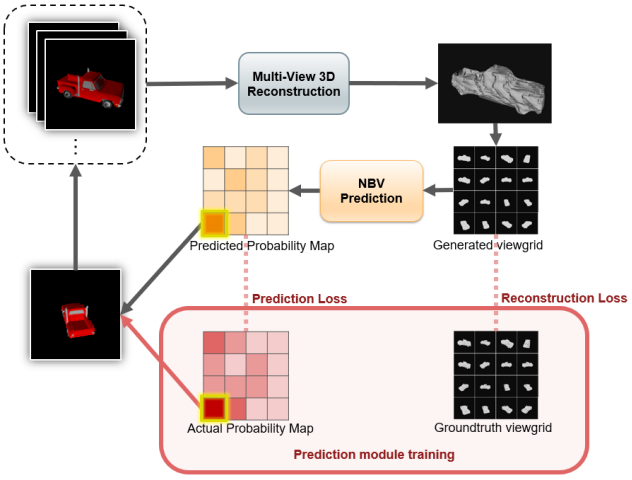


Figure 1: We start with a 3D reconstructed mesh from a randomly sampled image viewpoint of the object. We then render a viewgrid of the predicted mesh from a predetermined set of viewpoints for the model. We generate our actual probability map from the reconstruction loss between the predicted view grid and a corresponding ground-truth viewgrid. The viewpoint with the highest loss is taken as the Next Best View and becomes the viewpoint sampled next in the reconstruction. Our NBV prediction module is supervised using the actual probability map, and the predicted probability map is utilized during evaluation.

mative or useful for whole scene reconstruction. We differ by targeting 3D object reconstruction without reinforcement learning, constrained by a static set of viewpoints. Within this project, we focus on demonstrating the benefits of incorporating the NBV into a multi-view reconstruction pipeline, compared against any other randomly selected viewpoint.

The contributions of this paper can be summarized as follows:

- We present an active vision learning framework for 3D object reconstruction, which has the potential to jointly learn the 3D shape and the navigation strategy.
- We adapt the viewgrid technique used in [8] to calculate the silhouette loss. Instead of using natural images with RGB channels, we use their silhouette renderings to put the emphasis on the object outline.
- We implement a supervised learning strategy for Next Best View prediction. Experimental results show that our proposed model shows improvement on IoU scores as compared to a random selection strategy.

2. Related Works

2.1. 3D Reconstruction

A large amount of literature has been produced over the decades in the field of image-based 3D reconstruction. Traditional methods like Structure from Motion (SfM) suffers from the prohibiting feature matching procedure and fails with insufficient views. Remarkable improvement of reconstruction quality upon those traditional methods has been achieved by deep learning-based approaches. However, most of the state-of-the-art approaches are built upon one or a fixed set of input images and leave no freedom for the algorithm to choose viewpoints.

3D-R2N2, proposed by Choy et al. in [2], generates voxel reconstructions from multiple images using a novel 3D convolutional LSTM layer coupled with an encoder-decoder framework. The Learnt Stereo Machine (LSM) proposed in [6] encodes images with known camera poses to 2D features maps and then unprojects them to 3D for end-to-end learning. Another work, Pix2Vox, attempts to overcome the limitations of RNN-based methods like 3D-R2N2 and LSM by using an autoencoder structure to produce both single- and multi-view reconstructions, the latter achieved by generating (sans-RNN) coarse voxel reconstructions for each viewpoint and fusing them together [13].

Pixel2Mesh [11] comes up with an approach to reconstruct 3D meshes from a single image. It gradually deforms an ellipse surface into the final model, which has the drawback of performing poorly when dealing with objects with holes. This work is followed by Pixel2Mesh++ [12], where the authors introduce a MultiView Deformation Network to pool perceptual features from multiple views. Gkioxari et al. [4] propose Mesh R-CNN, which extends upon their Mask-RCNN [5] pipeline. They introduce a voxel branch that takes a single image and produces voxel reconstructions, which are converted into a mesh ("cubified") and then refined with a mesh branch to achieve finer details. Our model uses Mesh-RCNN framework as the main structure and replaces the voxel branch with Pix2Vox to be able to take arbitrary images for continuous reconstruction.

2.2. Next Best View

Active vision has been a popular area in the computer vision community. Jayaraman et al. [3] introduce the concept of a viewgrid to embed 3D shape information into a single-view image representation. They use a ShapeCode feature extractor to embed viewpoints into a ShapeCode representation. This representation is then used to produce the viewgrid which generates a 2D image of the object from multiple viewpoints. We use this concept to help us choose the NBV.

Seifi et al. [10] addresses the problem of active visual exploration of a large 360° input. It uses an attention module to decide the next location to attend. Ramakrishnan et

al. [8] proposes a reinforcement learning to let an agent learn efficient exploratory behaviors to acquire informative visual observations. Mendoza et al. [7] uses a 3D-CNN to directly predict the NBV based on supervised deep learning. Our method is also based on supervised learning, but instead of predicting the NBV directly from 3D reconstruction, we predict it from a viewgrid.

3. Method

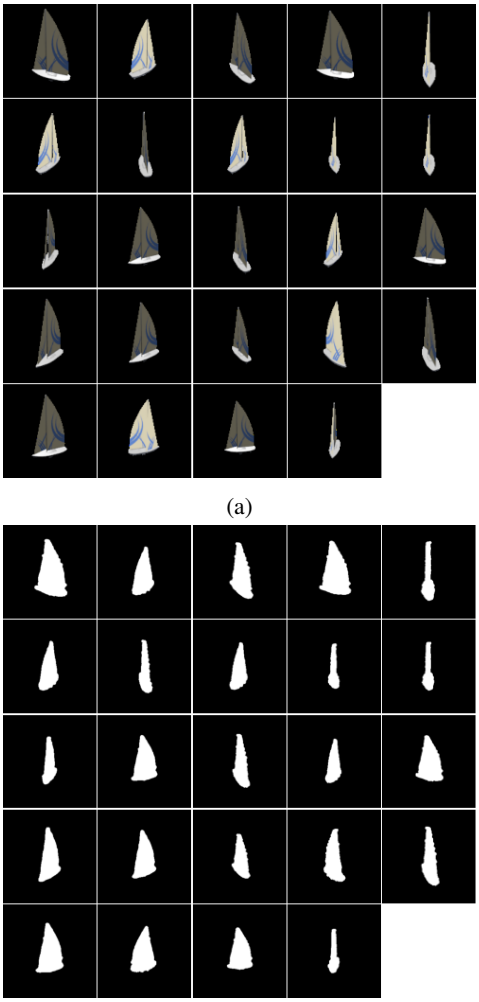
Our model consists of two primary parts: a multi-view 3D reconstruction module and a next-best-view prediction module. During the training phase, the 3D reconstruction module takes an image from a random viewpoint and reconstructs a 3D mesh. The NBV prediction module generates the viewgrid for the predicted mesh at the same viewpoints as ground-truth renderings, and then creates an actual probability map which is the normalized reconstruction loss between the rendering pairs. The viewpoint with the highest loss is the one with the highest uncertainty, and should be considered as the Next Best View used in the next reconstruction. During evaluation, the prediction module will determine which of the viewpoints is needed for the next reconstruction update. An overview of this pipeline is shown in Fig. 1. The following subsections discuss each major step in detail.

3.1. Multi-View 3D Reconstruction

Once we randomly select an initial viewpoint of an object and sample the image corresponding to that view, we use Pix2Vox to generate a 3D voxel of the object. Pix2Vox can reconstruct 3D models from one or multiple images. Chosen for its (relative) simplicity and lack of RNNs, it performs reconstruction by generating multiple single-view reconstructions which are fused together into a single voxel grid, which becomes the output and is converted into a mesh. Once the prediction module analyzes the viewgrid and selects the NBV, we use both the original image as well as the one corresponding to the NBV as inputs to Pix2Vox. This cycle can theoretically continue up through all $N = 24$ views, though we only selected a Next Best View once in this project.

3.2. Viewgrid Generation

We use viewgrids as the format for computing the reconstruction loss as well as evaluating the information gained from incorporating each subsequent view. Once we've received the reconstructed voxel from Pix2Vox and convert it into mesh, we use PyTorch3D [9] to create silhouette renderings of the mesh from a fixed set of viewpoints and form a viewgrid. We make use of 3D-R2N2-generated preprocessed renderings of the objects as our set of fixed viewpoints, which provides 24 views for per object. Using a



(a)

(b)

Figure 2: (a) Viewgrid using 3D-R2N2 provided renderings. (b) Viewgrid of silhouette renderings from mesh output, aligned to RGB images shown in (a).

fixed set of viewpoints ensures the locations in both viewgrids are the same. The viewgrids are essential to the NBV prediction module. Mean-squared error (MSE) is used as our reconstruction loss between the ground-truth viewgrid and the generated (predicted) viewgrid, as we will discuss in the next subsection.

3.3. Next View Prediction

Given the predicted and ground-truth viewgrids, we compute their reconstruction loss in Eq. 1 between each silhouette rendering of the predicted mesh, S_i^{pred}, and ground-truth mesh, S_i^{true}, where $i \in \{0, 1, 2, \cdots, 23\}$. This error grid, after being normalized by the maximum value for the sake of stability, will be passed to a soft-max operation in

Eq. 2 and becomes our ground-truth probability map, $\tilde{P}$. The higher the loss of a viewpoint, the more information the agent is missing from that viewpoint, and hence the more benefit the agent is going to get if it takes a peek from that view. Thus $\tilde{P}$ represents our belief how possible each viewpoint will be the next-best-view.

$$P_i = \frac{(S_i^{pred} - S_i^{gt})^2}{\max_j (S_j^{pred} - S_j^{gt})^2} \tag{1}$$

$$\tilde{P}_i = \frac{\exp P_i}{\sum_j \exp P_j} \tag{2}$$

Given the ground-truth silhouette renderings, we could simply treat the viewpoint with the highest loss in the viewgrid as the area with the highest uncertainty, and select that as our NBV, denoted as V_{i^*}, where $i^*(\theta) = \arg\max \tilde{P}_i(\theta)$, and θ is the parameters for the prediction module. However, during evaluation there's no such ground-truth available. Instead, we train a prediction module that learns to predict the ground-truth probability map. The prediction module architecture is ResNet-18 with a fully-connected layer added at the end, with weights initialized to ImageNet pre-trained weights. Taking a viewgrid as input, the prediction module outputs a predicted probability map, of which the area with the highest loss is selected as our next viewpoint. During training, we desire that the prediction module output a similar distribution as the ground truth probability map. We use a mean-squared error loss (Eq. 3) to train the prediction module from the normalized silhouette losses. But one could conceivably also use cross-entropy loss to treat this as a classification problem. In our experiments we found the silhouette losses had limited variability, so we settled on MSE.

$$L(\theta) = \frac{1}{N} \sum_n (\hat{P}_n(\theta) - \tilde{P}_n)^2 \tag{3}$$

4. Experiments

In order to best evaluate our model and the benefit of Next Best View , we conduct two main experiments. The first is an evaluation test of our prediction module, where we measure the accuracy of the module in selecting the Next Best View. The second experiment tracks the intersection over union (IoU) metric and aims to show that implementing the Next Best View at each time step provides maximum information gain.

4.1. Dataset

All experiments are performed on the ShapeNet dataset [1], specifically ShapeNetCoreV1. This is a subset of the full ShapeNet dataset which covers 55 common object categories with about 51,300 unique 3D models. From these 55 categories, we wanted to focus on just the 13 major classes, which still covers about 44,000 models. Unfortunately, do to uncommonly long training times, we decided to train and evaluate both experiments on just one class: the watercraft, comprised of 1,939 models. We decided to focus on this class in particular both because of its large intra-class variation as well as its relatively high saliency.

4.2. Implementation Details

All of the RGB images in both training and validation split are of size 137×137. As previously mentioned, we use the RGB renderings provided by 3D-R2N2 as our fixed viewpoints. We store the extrinsic matrix parameters for each viewpoint so we can later align the silhouette of the predicted mesh with the RGB renderings. As previously mentioned, our NBV prediction module uses a ResNet-18 backbone initialized with ImageNet pre-trained weights. The input to the prediction module is a stack of silhouette renderings of shape $N \times H \times W$. $N = 24$ images and the rendered silhouette dimensions are $H = W = 256$. Pix2Vox supplies a voxelized output, and to use this in our pipeline we use the `cubify` function supplied by PyTorch3D. This converts the output voxel into a Mesh object.

4.3. Accuracy of NBV Selection

For the first experiment, the prediction module's task is to select the Next Best View out of 24 possible options. Consequently, a natural baseline is random selection, which uniformly selects a random view with probability 4.17%.

Due to time constraints, we trained and evaluated our NBV prediction module on just the watercraft class. We generated a mesh from a single image with the trained Mesh R-CNN model. For both the predicted mesh and the ground-truth mesh, we generated a viewgrid of 24 silhouetted images from PyTorch3D. In order to align the two viewgrids for comparison and to input the second rendered image, we retrieved and used as necessary the rotation and translation matrices from the 3D-R2N2 rendering metadata. We used PyTorch's built-in MSE loss (Eq. 3) to train our model, which evaluated the difference between the predicted probability map and the actual probability map.

4.4. Average Intersection over Union (IoU)

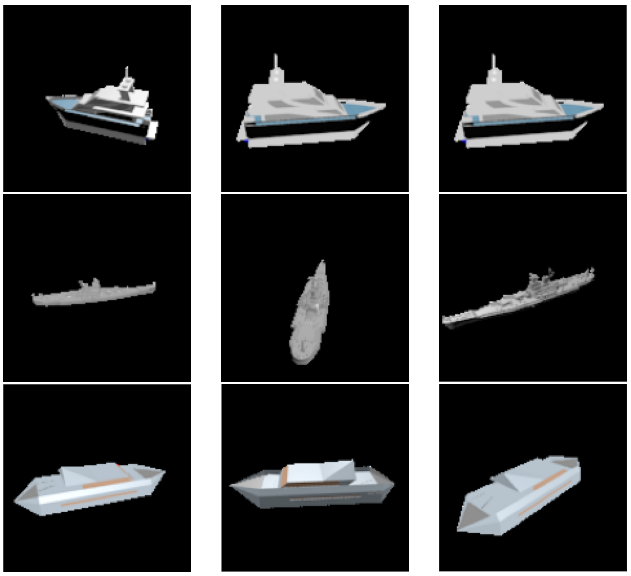
The second experiment is to calculate the average IoU of the voxelized reconstruction of the original viewpoint image and the Next Best View image. Thus, there are two simple baselines to make comparisons with. Our first baseline is to report the IoU of the single-view reconstruction from only the original viewpoint. The second baseline is to report the IoU of the multi-view reconstruction of the original viewpoint along with a randomly selected viewpoint. If the predicted Next Best View is truly the "best" view, it should result in an IoU greater than either of these baselines.

Prediction module	Training Accuracy	Validation Accuracy	Random Chance (N=24 views)
ResNet18 + FC + Softmax	39.689%	39.448%	4.17%
ResNet18 + FC	39.848%	37.771%	4.17%

Table 1: Prediction module performance on the watercraft class

Threshold	Single view	One + random	One + NBV
0.2	0.6521	0.6847	0.6861
0.3	0.6728	0.7047	0.7086
0.4	0.6851	0.7157	0.7210
0.5	0.6955	0.7288	0.7323

Table 2: IoU scores of voxel reconstruction with different threshold



(a) Input view (b) Ground-truth NBV (c) Predicted NBV

Figure 3: We provide three qualitative examples from our prediction module. Each row represents a different model from the watercraft class. The first column is the input view, the second column is the ground-truth next-best view i.e. view with the highest calculated loss, and the final column is the predicted next-best view.

4.5. Evaluation

4.5.1 Accuracy of NBV Selection

We treat the viewpoint with the highest value in the probability map as the Next Best View. The accuracy of NBV selection is calculated as the total correct predictions divided by the total predictions. As shown in Table 1, selecting the NBV by chance is around 4%, but our prediction module can correctly predict the NBV with about 40% accuracy. Our results state that the prediction module we trained is about 9.5 times better than random chance.

We provide qualitative examples in Fig. 3. Each row is a different model from the watercraft class. In the first row, our prediction module selects the same next view as the GT next best view. While the second and third rows select different views. If the RGB renderings provided consistent discrete viewpoints across all models, we could make more definitive statements. But intuitively it would seem that the next best view is either 90 or 180 degree rotation from the input view.

4.5.2 Average Intersection over Union (IoU)

To further see how the selected NBV improves the reconstruction quality, we calculated the average IoU of 100 randomly selected instances in the watercraft class. In this test, we go through the entire process of receiving the voxel output from Pix2Vox, converting it to a mesh, generating a viewgrid, and selecting the Next Best View. As usual, the NBV then becomes an additional input into Pix2Vox, which generates an updated reconstruction. But this time, we keep it in voxel format and calculate its IoU against the ground-truth voxel grid. We also computed its IoU with a voxel grid that used the original image and one randomly selected viewpoint as the NBV.

From Table 2 we can see that the voxel reconstruction using the original viewpoint and the NBV have the highest average IoU, meaning the predicted NBV provides greater additional information for 3D reconstruction than a randomly selected view.

5. Conclusion

We addressed the problem of 3D reconstruction through actively-chosen input views. We presented a learning-based

framework which incorporates a multi-view 3D reconstruction module and a next-best-view prediction module. We sampled a viewgrid of 24 discrete viewpoints per object for the agent to choose from and employed a supervised learning approach to predict the next best view. Qualitative and quantitative experiments showed that the proposed model outperforms the baseline methods and is able to dynamically choose observations and iteratively perform 3D reconstruction on novel objects.

6. Future Work

- Jointly training the entire model: at the current stage, we utilize a pre-trained Pix2Vox model for multi-view 3D reconstruction and focus on training the prediction module, but future work could incorporate the joint training of the two modules, allowing them to benefit from each other.
- Fixed viewgrid: although the views provided by 3D-R2N2 suffice as a set of fixed viewpoints, it would be interesting to expand our viewpoint selection and be able to render the object using a fixed discretized viewgrid on a sphere.
- Experiment with other multi-view reconstruction models: Pix2Vox is doing a satisfactory job for multi-view voxel reconstruction, but it would be interesting to integrate it into the state-of-the-art Mesh R-CNN framework so that we can do multi-view mesh reconstruction.
- A better way to evaluate the information gain with each extra input view: IoU, Chamfer Distance and F1 scores could be used to evaluate the reconstruction quality.

References

[1] A. X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su, J. Xiao, L. Yi, and F. Yu. ShapeNet: An Information-Rich 3D Model Repository. Technical Report arXiv:1512.03012 [cs.GR], Stanford University — Princeton University — Toyota Technological Institute at Chicago, 2015. 4

[2] 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 *European conference on computer vision*, pages 628–644. Springer, 2016. 2

[3] R. G. Dinesh Jayaraman and K. Grauman. Shapecodes: Self-supervised feature learning by lifting views to viewgrids. 2018. 2

[4] G. Gkioxari, J. Malik, and J. Johnson. Mesh r-cnn. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 9785–9795, 2019. 1, 2

[5] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017. 2

[6] A. Kar, C. Häne, and J. Malik. Learning a multi-view stereo machine. In *Advances in neural information processing systems*, pages 365–376, 2017. 2

[7] H. T. L. E. S. Miguel Mendoza, J. Irving Vasquez-Gomez and C. Reta. Supervised learning of the next-best-view for 3d object reconstruction. 2019. 3

[8] S. K. Ramakrishnan, D. Jayaraman, and K. Grauman. Emergence of exploratory look-around behaviors through active observation completion. *Science Robotics*, 4(30), 2019. 1, 2, 3

[9] N. Ravi, J. Reizenstein, D. Novotny, T. Gordon, W.-Y. Lo, J. Johnson, and G. Gkioxari. Pytorch3d. https://github.com/facebookresearch/pytorch3d, 2020. 3

[10] S. Seifi and T. Tuytelaars. Where to look next: Unsupervised active visual exploration on 360° input. *arXiv preprint arXiv:1909.10304*, 2019. 1, 2

[11] N. Wang, Y. Zhang, Z. Li, Y. Fu, W. Liu, and Y.-G. Jiang. Pixel2mesh: Generating 3d mesh models from single rgb images. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 52–67, 2018. 2

[12] C. Wen, Y. Zhang, Z. Li, and Y. Fu. Pixel2mesh++: Multi-view 3d mesh generation via deformation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1042–1051, 2019. 1, 2

[13] H. Xie, H. Yao, X. Sun, S. Zhou, and S. Zhang. Pix2vox: Context-aware 3d reconstruction from single and multi-view images. In *ICCV*, 2019. 1, 2