Zero-shot Prompt-based Partial 3D Point Cloud Creation of the Specified Object from an Unlabeled 2D Image

Sagar Manglani
Stanford University
Stanford, CA
sagarm@stanford.edu

Abstract

This project aims to generate RGBD point clouds from specific objects in unlabeled 2D images using zero-shot, prompt-based methods. By integrating advanced machine learning models for object detection, image segmentation, and depth estimation, it creates accurate 3D models from 2D data. Leveraging the recent advancements in computer vision, this approach utilizes textual prompts to perform tasks without prior explicit training. The model inputs a 2D image and outputs a partial RGBD point cloud, effectively transforming 2D images into 3D representations.

The primary goal is to demonstrate the feasibility and effectiveness of this zero-shot framework. Objectives include achieving effective detection in prompt-based object detection, high accuracy in segmentation, and creating high-quality depth maps. The integration of these technologies aims to produce detailed and reliable 3D outputs suitable for various real-world applications. The generated 3D models have significant potential to enhance virtual reality environments, build new 3D print models, and expand training datasets in areas where data is limited - by synthetically generating data with 3D models of underrepresented objects, thereby reducing bias in machine learning applications.

1. Introduction

Transforming two-dimensional images into three-dimensional point clouds presents a significant challenge. Traditionally, 3D models have been derived from depth data obtained through LiDAR, and converting these data points into refined 3D models often involves extensive manual intervention due to the complexity of cleaning up 3D point clouds.

Recent advancements in deep learning for computer vision have enabled effective monocular depth estimation, which deduces depth from a single image. Furthermore, the integration of vision-language models leverages contextual information from textual prompts to facilitate zero-shot learning—allowing models to perform tasks without prior explicit training. This capability is complemented by modern zero-shot segmentation models, which accurately isolate specific parts of an image, representing a significant step forward in computer vision technology.

This project capitalizes on these innovations by integrating advanced models for object detection, segmentation, and depth estimation to create 3D point clouds for specified objects from unlabeled 2D images, driven solely by textual prompts.

1.1. Problem Statement

This project aims to employ a combination of zero-shot learning models that do object detection, segmentation, and depth estimation from single, unlabeled 2D images to create accurate 3D point clouds. The input to the model during inference is one 2D image and the output of the model during inference is one partial RGBD point cloud of the object in the scene. The input to the model during fine-tuning is a small set of image and depth pairs. This is used to reduce scale variancy of the depth output and the actual depth.

1.2. Goals

The primary goal of this project is to demonstrate the feasibility and effectiveness of a zero-shot framework for generating 3D point clouds. The specific objectives include:

- Achieving effective detection in prompt-based object detection from general 2D images.
- Attaining high accuracy in segmentation using the bounding box generated from object detection.
- Creating high-quality depth maps using monocular depth estimation models.
- Integrating these technologies to produce detailed and reliable 3D outputs suitable for various real-world applications.

2. Related Work

Recent developments in computer vision have greatly enhanced our ability to interpret and reconstruct complex scenes from 2D images. Historically, techniques such as stereo vision and structure from motion were prevalent for depth estimation [4]. With the advent of deep learning, the focus has shifted toward monocular depth estimation. Methods like those developed by Godard et al. [3] employ convolutional neural networks to generate depth maps from single images. The Depth Anything model [9] represents the current state-of-the-art in monocular depth estimation and has been trained on an extensive dataset, including 1.5 million labeled depth images and over 62 million unlabeled images, providing it with significant reliability in depth estimation.

In parallel, advancements in Open Vocabulary Object Detection [10] have bridged textual and visual data, enhancing zero-shot learning capabilities. The YOLO (You Only Look Once) World model [2] represents the forefront of this technology.

Furthermore, Meta's Segment Anything [5] model marks foundational progress in image segmentation using self-supervised learning. The Segment Anything model (SAM) operates under a zero-shot framework, which means it can segment objects it has not seen during training. Although faster models such as Fast SAM [12] and Mobile SAM [11] exist, they rely on a CNN encoder and generally exhibit lower performance compared to the original SAM.

Despite these advancements, the development of a unified framework for 3D reconstruction of objects from single images remains underexplored. This project builds upon these technologies to craft a more integrated and efficient approach to 3D modeling from 2D images.

Notably, while ZeroNVS by Sargent et al. [8] focuses on reconstructing entire 3D scenes, this project distinctively concentrates on prompt-based 3D object reconstruction, offering a more targeted and accurate 3D reconstruction.

3. Methods

3.1. Object Detection and Refinement

The YOLO (You Only Look Once) World is utilized as the primary mechanism for object detection in this project. This model is chosen for its effectiveness in detecting objects based on a prompt like "sitting person", "floor lamp", "chair", etc. At this moment, the model is limited to one detection per image, so there is only one 3D model generated per image. To ensure precision in detection, the algorithm controlling the model selects the detection with the highest confidence and only considers that detection for subsequent operations. If there are no detections within the image, the operation is skipped for that image. Consequently, the confidence threshold of the model is set low, as all non-prime detections are discarded.

3.2. Segmentation with Segment Anything Model

Following successful detection, the bounding box is passed to the Segment Anything Model (SAM). The bounding box from the YOLO model is used by SAM to segment the image into two parts: the object within the box and the image outside it. This workflow is critical for maintaining the accuracy and efficiency of the process, paving the way for the subsequent depth estimation and 3D modeling stages. This approach ensures that each part of the system contributes effectively to the overall goal of transforming 2D images into detailed 3D point clouds.

3.3. Depth Estimation and Enhancement

The Depth Anything model [9], a monocular depth estimation model, is utilized to determine the depth information of the scene, although it outputs relative depth. To refine these depth outputs and align them with the NYU Depth v2 dataset, the Depth Anything model is fine-tuned using the NYU Depth v2 dataset. For this purpose, we use the toolkit from ZoeDepth [1]. This step aims to calibrate the model's depth perception to better match the standardized depth values provided in this dataset. All the model training and evaluations are done with PyTorch [7].

3.4. Integration and 3D Reconstruction

The segmented image and the depth map are merged to produce a segmented depth map. This map explicitly highlights the depth of the detected object while minimizing background interference. Using the camera calibration data, this segmented depth map is converted into a 3D point cloud using the transformations described below.

Given a depth image where each pixel (u, v) contains a depth value Z, and the camera's intrinsic parameters including the focal lengths f_x, f_y along the x and y axes, and the optical center (c_x, c_y), the conversion to 3D point cloud coordinates can be performed using the following equations:

1. Convert pixel coordinates to camera coordinates:

$$X_c = \frac{(u - c_x) \cdot Z}{f_x}, \tag{1}$$

$$Y_c = \frac{(v - c_y) \cdot Z}{f_y}, \tag{2}$$

$$Z_c = Z. \tag{3}$$

2. Transform camera coordinates to world coordinates:

   Using the extrinsic parameters (rotation matrix R and translation vector $\mathbf{t}$), the coordinates in the world coordinate system can be computed as:

$$\begin{bmatrix} X_w \\ Y_w \\ Z_w \end{bmatrix} = R \cdot \begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix} + \mathbf{t}. \tag{4}$$

This results in a partial set of points representing the point cloud based on the information available in the original image. These point clouds are exported into a point cloud file. All the visualizations and 3D model exports are done with Open3D [13].

3.5. Project Contributions

This project leverages existing methods for prompt-based object detection, image segmentation, and depth estimation. It also leverages an existing method for fine-tuning the depth estimation. This project contributes in the following ways:

1. Created a new dataset from the NYU Depth v2 dataset, featuring manually labeled binary segmentation and associated prompts while preserving the original depth and image information.
2. Combined object detection and semantic segmentation to achieve accurate pixel-level segmentation for the prompted objects.
3. Integrated the resulting segmentation with monocular depth estimation and fine-tuned the depth estimation to produce clean RGB depth maps.
4. Applied computer vision techniques to convert RGB depth maps into RGB point clouds.
5. Performed several experiments to evaluate the viability and effectiveness of this approach.

4. Dataset

This project creates a new dataset from the NYU Depth Dataset V2 [6], a comprehensive collection of RGB and depth images captured from indoor scenes. NYU Depth v2 features approximately 1,400 labeled pairs from various indoor environments, with around 600 labeled pairs in the test dataset. We created a new dataset consisting of 331 images from the test portion, along with associated prompts, segmentation, and depth. The segmentation is a binary mask of the object and the object alone. The dataset prompts are highly diverse, featuring 142 different prompts describing objects across the 331 images. We used the test dataset to ensure minimal bias in the results, as some models have been trained on the NYU Depth v2 training dataset. Additionally, the prompts are crafted to be part of natural language rather than curated to work well with word embeddings.

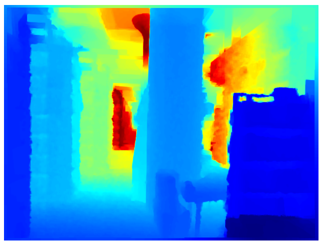
To build this dataset, an annotation tool was developed to specify the prompt and select the associated binary segmentation of that object. For example, if a guitar is selected in the image, the same pixel location is used to read the segmentation value from the segmented image data. Then, a connected-component analysis is performed on the image using that pixel value until all pixels representing that segmentation are selected. The segmented pixels are saved as a binary mask image, and the prompt is saved as a text file. Figure 1 shows some samples of the dataset, with the prompt for each image displayed below it as a caption.

(a) Prompt: Yoga mat on Table

(b) Prompt: Guitar

(c) Yoga mat on Table Seg

(d) Guitar Segmentation


(e) Yoga mat on Table Depth GT

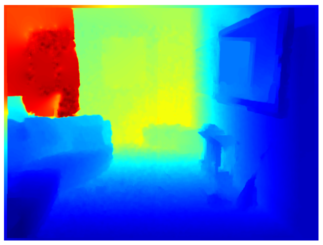

(f) Guitar Depth GT

Figure 1: Custom Dataset from NYU Depth v2 Test dataset

5. Results

The dataset we created includes an image, the ground-truth segmentation, and the ground-truth depth. Figure 2 shows results for a test sample. The image in (a) is processed with the prompt ”taller bin.” We first estimate the bounding box using YOLO World, as shown in (b). To obtain the predicted segmentation, we pass the bounding box to the Segment Anything model, resulting in (c). This predicted segmentation can be compared with the ground-truth segmentation shown in (d).

Upon closely comparing (c) and (d), we observe that the quality of segmentation in the prediction seems slightly better than the ground truth. For instance, the bottom of the bin is clipped off in a straight line in (d), while it closely follows the actual boundary in (c).

Next, we perform monocular depth estimation on the original image, shown in (e), and obtain the predicted relative depth, shown in (f). The depth in (f) might not reveal detailed features of the bin due to the reduced resolution for visualization (int8), whereas the real depth from the model is in float32 format, providing a higher resolution. After fine-tuning the model, the predicted depth, as seen in (g), appears similar to the ground-truth depth in (h).

Comparing (g) and (h) closely, we notice that the depth in (g) appears smoother than in the ground truth (h). For example, the overhead lights near the ceiling have better detail in (g) compared to (h). This difference is likely because the ground-truth depth is collected from a Microsoft Kinect sensor for NYU Depth v2, which typically has lower resolution and some noise due to its structured light sensor.

Finally, we combine the image, depth, and mask to generate a partial 3D point cloud, shown in (i) and (j). The 3D point clouds are purposefully turned to a slightly isometric view to better understand the 3D object. The result in (i) combines the image, predicted segmentation mask, and predicted fine-tuned depth, while the result in (j) combines the image, ground-truth segmentation mask, and ground-truth depth.

Looking closely at (i) and (j), we observe much smoother surfaces on the predicted 3D model compared to the ground truth. Additionally, the lid of the bin in the predicted model has a smooth curved contour, whereas the ground truth shows several distorted points around the lid.

5.1. Quantitative results

Confidence	TP	FP	FN	P	R	mIOU
0.01	156	16	175	0.906	0.471	0.829
0.001	255	23	76	0.917	0.770	0.830
0.0001	298	28	33	0.914	0.900	0.829

Table 1: Object Detection Results on Custom Dataset

Table 1 shows the results of the object detection task, presenting True Positives (TP), False Positives (FP), and False Negatives (FN). We observe that as the confidence threshold decreases, the number of false negatives drops significantly. While the number of false positives increases with the lower confidence threshold, the increase in true positives maintains overall precision, even though recall decreases. Precision and recall are defined as follows:

$$Precision = \frac{TP}{TP+FP}$$

$$Recall = \frac{TP}{TP+FN}$$

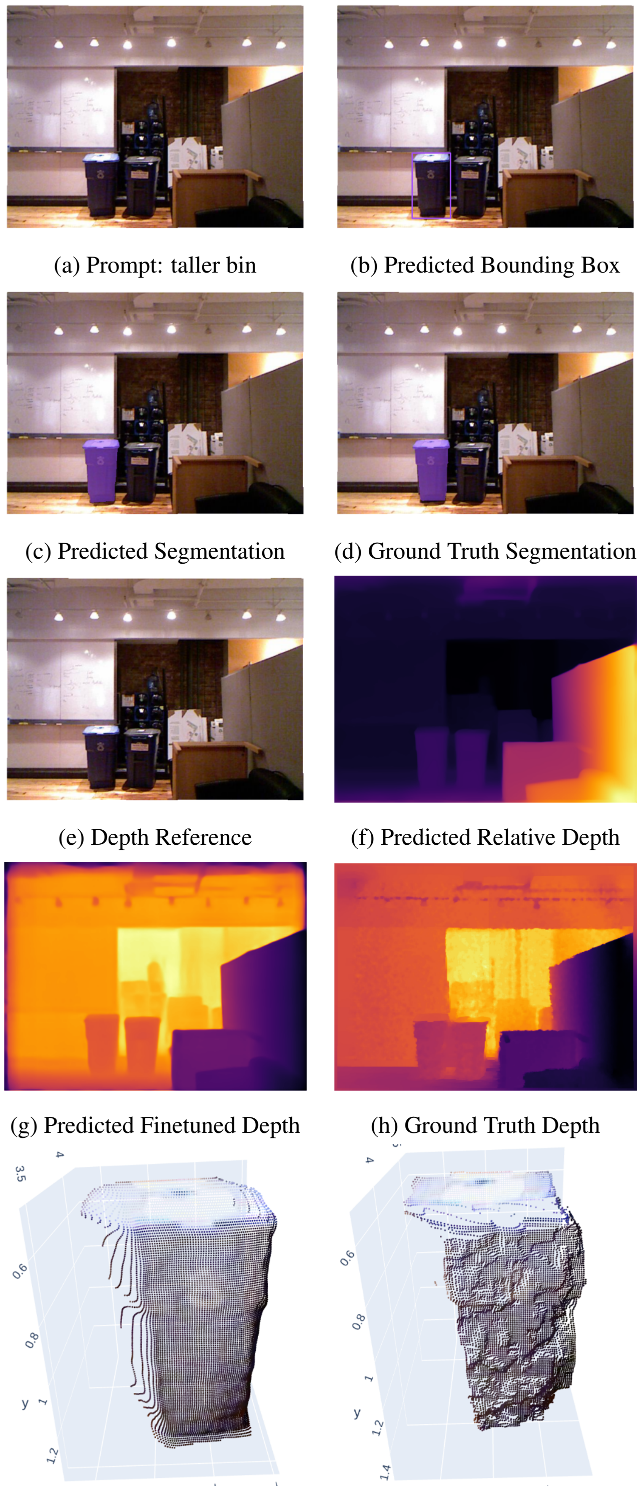

(a) Prompt: taller bin (b) Predicted Bounding Box (c) Predicted Segmentation (d) Ground Truth Segmentation (e) Depth Reference (f) Predicted Relative Depth (g) Predicted Finetuned Depth (h) Ground Truth Depth (i) Predicted 3D Model (j) Ground Truth 3D Model

Figure 2: Detailed results on a sample image

This results in a trend where the best results are observed at the lowest confidence levels. It is important to note that the problem of low confidence is not as severe in our use case, as we only consider one detection with the highest confidence. Therefore, we will continue evaluating all downstream tasks with a confidence level of 0.0001.

Confidence	mIOU	Dice Coeff.	Pix. Acc.
0.0001	0.829	0.898	0.992

Table 2: Segmentation Results on Custom Dataset

In table 2, we present the results of the segmentation task, evaluated using mean Intersection over Union (mIoU), Dice Coefficient, and Pixel Accuracy. These metrics are defined as follows:

$$mIoU = \frac{1}{N}\sum_{i=1}^{N}\frac{|A_i \cap B_i|}{|A_i \cup B_i|}$$

$$Dice = \frac{2|A \cap B|}{|A|+|B|}$$

$$PixelAccuracy = \frac{\text{Number of correctly predicted pixels}}{\text{Total number of pixels}}$$

These metrics are calculated where there is a minimum overlap between the two bounding boxes. If there is no overlap, false positive and false negative counts are increased, indicating a bad detection. We observe a strong performance with the Segment Anything model, suggesting its effectiveness. However, it is important to note that the images in the NYU Depth Dataset often have borders between segments, which do not perfectly cover edges. This leads to issues like those observed in Figure 2 parts (c) and (d). This is likely why we observe such high pixel accuracy in the model.

Model	MAE	RMSE	SILog
Finetuned	0.236	0.383	0.130

Table 3: Despth Estimation Results on Custom Dataset

In table 3, we observe strong depth performance with the Depth Anything model across Mean Absolute Error (MAE), Root Mean Square Error (RMSE), and Scale-Invariant Logarithmic Error (SILog). They are defined as follows:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|d_i^{\text{pred}} - d_i^{\text{gt}}$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(d_i^{\text{pred}} - d_i^{\text{gt}})^2}$$

$$SILog = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\delta_i)^2 - \left(\frac{1}{n}\sum_{i=1}^{n}\delta_i\right)^2}$$

where $\delta_i = \log d_i^{\text{pred}} - \log d_i^{\text{gt}}$

The evaluations are performed with the unit being meters, so an MAE of 0.236 means the error in depth across all scenes is approximately 0.236 meters. Additionally, a higher RMSE suggests that some outliers in the dataset are exponentially increasing the error. SILog is a metric that is particularly useful when the absolute scale of the depths is not as important as the relative differences between them. This metric represents the relative shape between ground truth depth and predicted depth.

5.2. Qualitative Results

Figure 3 illustrates the previously shown qualitative results of object detection, segmentation, and depth estimation performed on the NYU Depth v2 dataset, alongside the related ground truth (GT) and associated prompts. Here, the results of the fine-tuned model are also shared in the original JET colorspace.

Here we observe similar facts as the first test sample. The segmentation (seg) shown in (e) and (f) are the segmentations directly from the dataset. There are substantial gaps in the segmentation and it does not appear to be pixel accurate. We observe a similar pattern in depth where the ground truth (GT) depth seems a bit rougher than the fine-tuned depth, but some features are lost in the fine-tuned depth - for example, the whiteboard in the image (j). Next, we'll look at the generated 3D shapes for qualitative analysis.

5.3. 3D objects

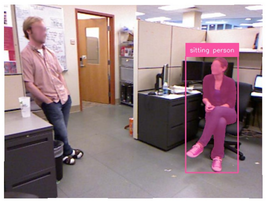
Figure 4 shows some of the 3D models extracted from the NYU Depth v2 Dataset. All the 3D models are rotated to display a slightly isometric view. Here, we can observe that 3D shape extraction works well for objects that occupy a significant number of pixels or are distinct from other objects. The NYU Depth v2 dataset is particularly challenging in this case because the images within the dataset are only 640x480 pixels. Convex objects like toys tend to perform better than concave objects such as a 'kitchen sink.' Although there are noticeable gaps in the models, resulting in partial point clouds, this information is still valuable. It can be utilized to create complete 3D models since these 3D objects convey significant information about shape and color.

Figure 5 presents a comparison between sample 3D models and their corresponding ground truth representations. Since the dataset does not include actual 3D models, the ground truth 3D structures are derived similarly to the predicted 3D objects. However, they utilize the segmentation and depth information directly from the dataset. Here, you can observe that in the first three examples, the model's predictions appear cleaner than the ground truth. However, in the fourth example, there are noticeable gaps in the predicted 3D model of the range hood. These gaps are likely due to segmentation errors caused by lighting conditions on the range hood in the image.

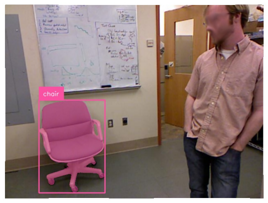
Note: You can find almost 300 3D models generated from the prompt dataset and their ground truth from the

(a) Prompt: Sitting person

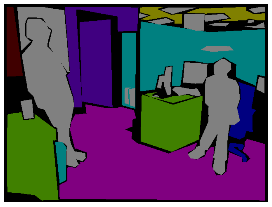
(b) Prompt: Chair

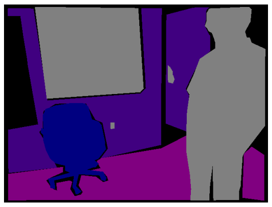


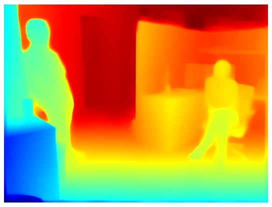
(c) SP Segmentation

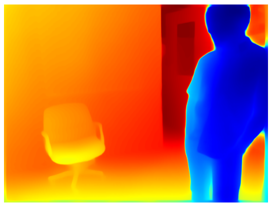


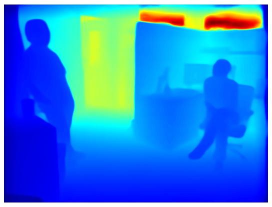
(d) Chair Segmentation

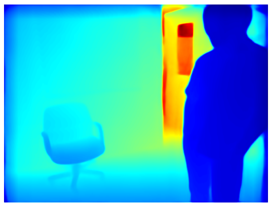

(e) SP Seg GT

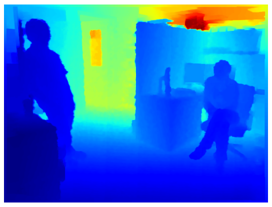

(f) Chair Seg GT


(g) SP Depth Estimation


(h) Chair Depth Estimation


(i) SP Depth Finetuned


(j) Chair Depth Finetuned


(k) Sitting Person Depth GT

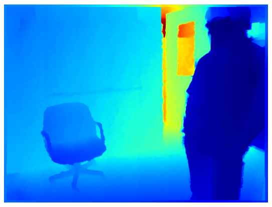
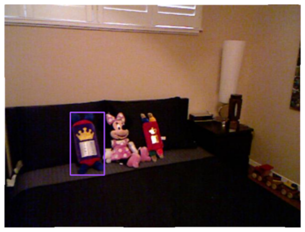
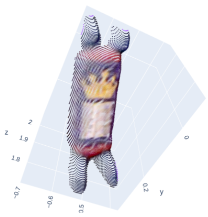
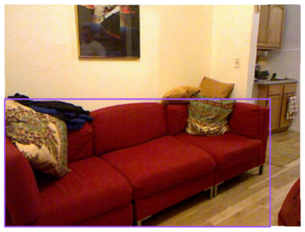

(l) Chair Depth GT

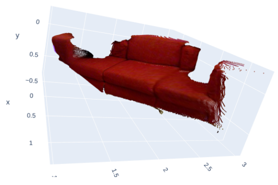
Figure 3: Results on NYU Depth v2 Test dataset

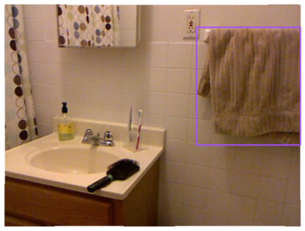
dataset at the Google Drive link shared below.

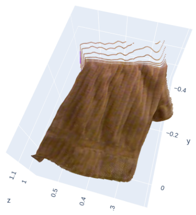

(a) Prompt: left toy

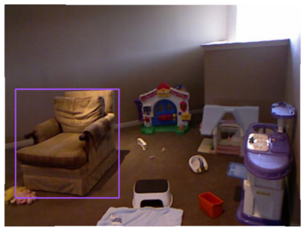

(b) 3D model

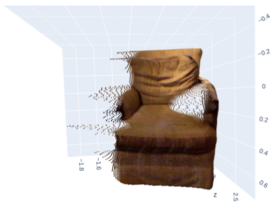

(c) Prompt: red sofa

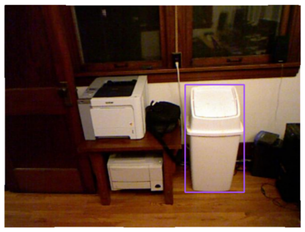

(d) 3D model


(e) Prompt: towel


(f) 3D model


(g) Prompt: chair


(h) 3D model


(i) Prompt: trash can

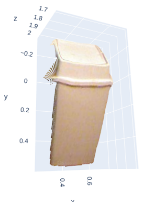
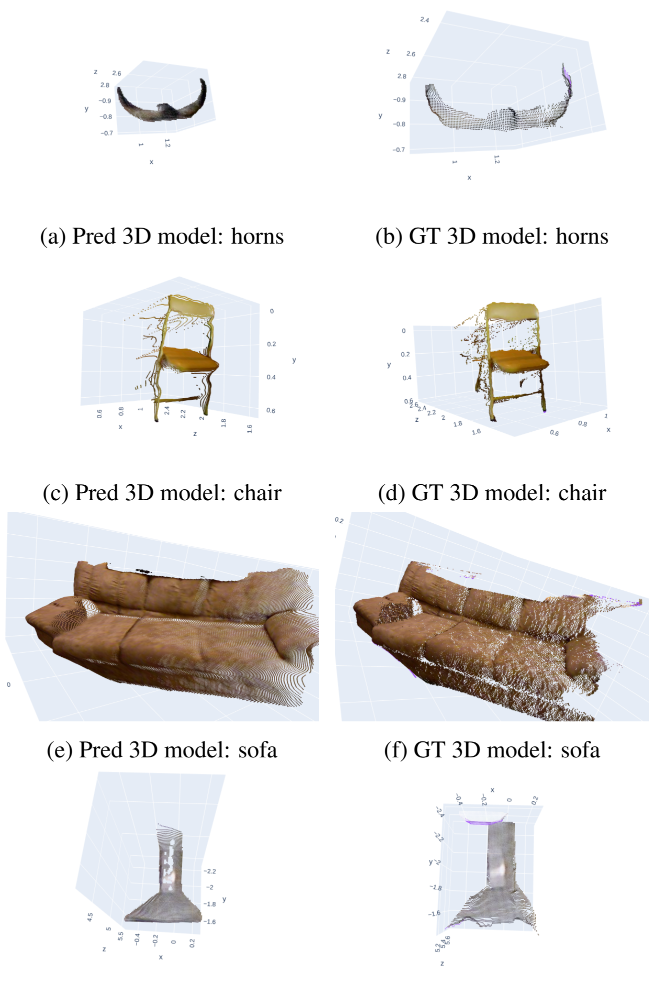

(j) 3D model

Figure 4: Generated 3D Objects with Source Images

5.4. Challenges

There are multiple types of challenges in the current approach. Let's review each of them:

1. No detections with YOLO World: In Figure 6, we see


(a) Pred 3D model: horns (b) GT 3D model: horns

(c) Pred 3D model: chair (d) GT 3D model: chair

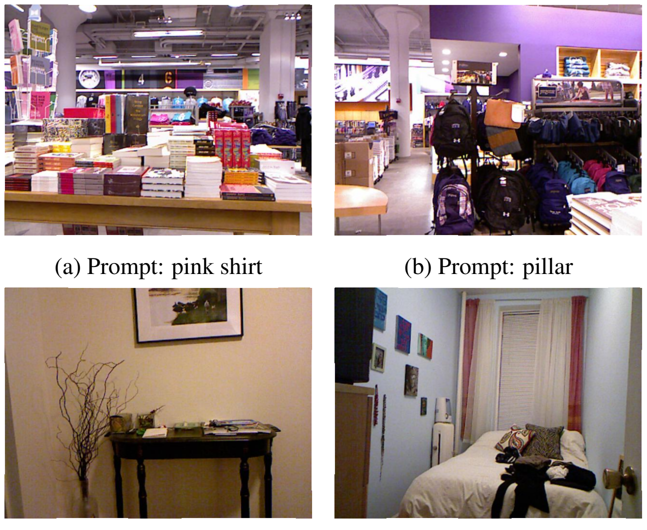
(e) Pred 3D model: sofa (f) GT 3D model: sofa

(g) Pred 3D model: range hood (h) GT 3D model: range hood

Figure 5: Generated 3D Objects vs. Ground Truth

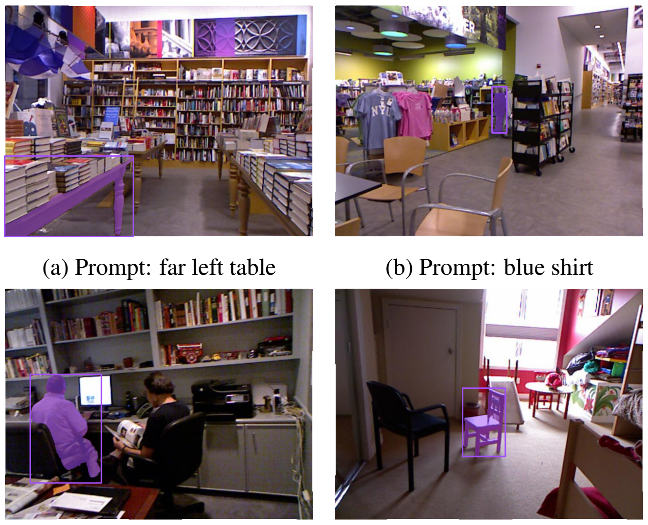
images and prompts where the model has failed to generate a detection. It can be assumed that some of the prompts, such as 'pillar' and 'door knob,' may be outside the vocabulary of the model. Additionally, if there are too many objects in the scene (as seen in the 'pink shirt' and 'table' images) and/or the object is not clearly visible, the model tends to perform worse.

2. Incorrect detection leading to incorrect segmentation: Figure 7 shows some sample cases where the object detection model makes incorrect detections. In the image (a), the model identifies the 'close left table' as the 'far left table,' indicating that the model has limited spatial understanding with respect to the text embeddings. In the image (b), the model selects a blue shirt that is farther away rather than the one nearby, suggesting a gap in the dataset where the prompt could be refined to specify a 'light blue shirt.' In the image (c), the model identifies a person in a pink shirt sit-


(a) Prompt: pink shirt (b) Prompt: pillar

(c) Prompt: table (d) Prompt: door knob

Figure 6: Cases with no detection


(a) Prompt: far left table (b) Prompt: blue shirt

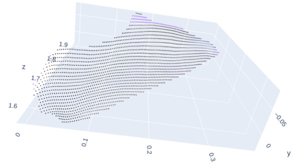
(c) Prompt: person in black (d) Prompt: blue chair

Figure 7: Cases with failed detection

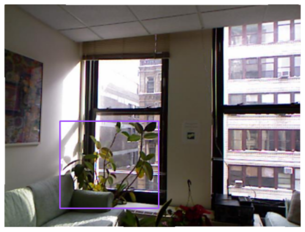
ting on a dark-brown chair as the 'person in black,' whereas the person next to him, who is wearing a black shirt and sitting in a black chair, is the correct match. For image (d), the red chair is selected instead of the blue one, which might be partially due to the challenging lighting conditions causing the chair to appear black.

3. Incorrect 3D models: Figure 8 shows failure cases for the 3D models. Here, we observe that the 3D model for the keyboard shown in (b) extracted from (a) is too sparse to be usable. In image (d), we observe a plant extracted from (c) that is mired in noise and raises questions on the usability

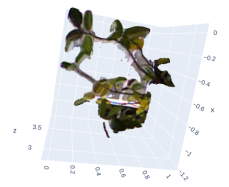
(a) Prompt: keyboard



(b) 3D model of keyboard



(c) Prompt: plant



(d) 3D model of plant

Figure 8: Cases with failed 3D models

of the same.

4. Extracting only one 3D shape per image: This is an artificially imposed limit within the code to allow users to precisely describe the 3D shape they are looking for in the image. This restriction can be lifted in a future iteration to allow for multiple 3D shape extractions when needed.

5.5. Code, Dataset, and Implementations

All the implementations, dataset, and all supplementary tools are available at: Google Drive Link

Note to TAs: You can view the entire process, including the final 3D model of the object, without running anything in `zero_shot_prompt_3d.ipynb`. To run the notebook on a single sample, use `zero_shot_prompt_3d.ipynb`, or to process the entire dataset, use `process_dataset.ipynb`. Please mount the shared Google Drive link in the notebook to access the relevant models and data. If you have any questions, please reach out to sagarm@stanford.edu or manglanisagar@gmail.com.

6. Conclusion

The results of this project demonstrate the viability of extracting 3D shapes using the model's zero-shot capability. This indicates significant potential for generating detailed 3D representations from 2D images. The next objective over the summer is to develop this capability into a complete 3D model generator through several key steps.

First, we will build a comprehensive dataset that includes prompts, images, segmentation masks, depth information, and 3D point clouds. This dataset will serve as the foundation for training and validating our model. The prompts will be diverse and descriptive, and the images will be carefully curated to correspond to these prompts. Accurate segmentation masks will be annotated for each image, and depth information will be included to provide spatial context. Additionally, accurate 3D point clouds will be generated for the objects in the images, serving as ground truth for the training process.

Next, we will integrate the detection, segmentation, and depth estimation pipelines into a single, unified multi-task network. This integration will enhance the efficiency and accuracy of the model, allowing it to process inputs and produce comprehensive outputs more effectively. We will then train this unified network using the compiled dataset. The training process will focus on optimizing the network to accurately predict segmentation and depth information based on the given prompts and images.

Furthermore, we aim to expand the network's capabilities to predict full 3D RGBD point clouds. This involves expanding the multi-task network and training it with images, prompts, and 3D objects. This involves adding a generative aspect to the model to generate the unseen parts of the RGBD point cloud that are consistent in shape as well as accurate in color. This additional training will also ensure that the network can generate high-fidelity 3D models that accurately reflect the input data.

By following these steps, we aim to develop a robust and efficient 3D model generator that leverages the zero-shot capabilities of the current model. This will significantly enhance our ability to create detailed and accurate 3D representations from 2D images.

7. Acknowledgements

Several implementations were used in this paper. They are listed below:

1. NYU Depth v2 Dataset is used as a foundation to create the custom dataset.

2. NYU Python Toolkit is used to extract the NYU dataset, which is offered in Matlab format.

3. YOLO World is used for bounding box detection with open vocabulary.

4. Segment Anything Model is used for image segmentation.

5. Depth Anything is used for monocular depth estimation.

6. ZoeDepth is used for fine-tuning depth estimation.

References

[1] S. F. Bhat, R. Birkl, D. Wofk, P. Wonka, and M. Müller. Zoedepth: Zero-shot transfer by combining relative and metric depth, 2023.

[2] T. Cheng, L. Song, Y. Ge, W. Liu, X. Wang, and Y. Shan. Yolo-world: Real-time open-vocabulary object detection. *arXiv preprint arXiv:*, 2024.

[3] C. Godard, O. Mac Aodha, and G. J. Brostow. Unsupervised monocular depth estimation with left-right consistency. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 270–279, 2017.

[4] R. Hartley and A. Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003.

[5] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, et al. Segment anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4015–4026, 2023.

[6] P. K. Nathan Silberman, Derek Hoiem and R. Fergus. Indoor segmentation and support inference from rgbd images. In *ECCV*, 2012.

[7] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

[8] K. Sargent, Z. Li, T. Shah, C. Herrmann, H.-X. Yu, Y. Zhang, E. R. Chan, D. Lagun, L. Fei-Fei, D. Sun, et al. Zeronvs: Zero-shot 360-degree view synthesis from a single real image. *arXiv preprint arXiv:2310.17994*, 2023.

[9] L. Yang, B. Kang, Z. Huang, X. Xu, J. Feng, and H. Zhao. Depth anything: Unleashing the power of large-scale unlabeled data. *arXiv preprint arXiv:2401.10891*, 2024.

[10] A. Zareian, K. D. Rosa, D. H. Hu, and S.-F. Chang. Open-vocabulary object detection using captions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14393–14402, 2021.

[11] C. Zhang, D. Han, Y. Qiao, J. U. Kim, S.-H. Bae, S. Lee, and C. S. Hong. Faster segment anything: Towards lightweight sam for mobile applications. *arXiv preprint arXiv:2306.14289*, 2023.

[12] X. Zhao, W. Ding, Y. An, Y. Du, T. Yu, M. Li, M. Tang, and J. Wang. Fast segment anything. *arXiv preprint arXiv:2306.12156*, 2023.

[13] Q.-Y. Zhou, J. Park, and V. Koltun. Open3d: A modern library for 3d data processing. *arXiv preprint arXiv:1801.09847*, 2018.