EE381V: Active 3D Reconstruction by Using Ego-centric Camera

Mingyo Seo
EID: ms84662
mingyo@utexas.edu

Zhou Fang
EID: zf2727
fazhou@utexas.edu

Abstract

In this project, we aim at an active 3D reconstruction of a given object using an ego-centric posed camera. 3D reconstruction from ego-centric image frames requires various viewpoints to achieve enough accuracy. Therefore, an inexperienced user may experience difficulty taking images in proper viewpoints without any instructions. To solve the issue, we reconstruct the 3D shapes of a target object while computing the certainty of the explored areas on the reconstructed shape. With the spatial certainty on the reconstructed shape, the most preferred direction of the viewpoint that the user should move to is predicted. For implementing the 3D reconstruction module, we applied NeuralRecon, a real-time 3D shape reconstruction model, which is trained by the ScanNet dataset. NeuralRecon outputs reconstructed shapes of target objects as well as the confidence of estimation. Then, reconstructed shapes are down-sampled to process fast computation of normal vectors on the surface, and the spatial integration of normal vectors weighted by confidence on the surface is used as an index of the most preferred direction of future viewpoints. We tested this method by using an Apple iPhone 12 which equips with an HD camera and an inertial measurement unit to get image frames labeled with the position and the orientation of the camera. We considered reconstructing the shape of a single piece of furniture as a target episode and presented the results of the reconstructed shapes and active viewpoint instructions. Attachments for this project such as codes and videos are accessible at this link.

1. Introduction

Constructing an object's 3D shape has been studied in many fields for different purposes. In augmented reality, reconstructed models transfer information of the surrounding physical environments to AR and enable realistic effects. As an application for AR purposes, 3D reconstruction can produce 3D shapes of objects without additional human efforts, such as building models by using computer-aided design (CAD) [4]. It can also help people understand the virtual objects more straightforward [2]. To enhance the effect of realistic effects, there have been many studies on how to make rendered virtual objects react to users' motion and changes in environments, by using sensors and analyzing spatial information surrounding users [10].

In robotics, spatial relations between robots and environments provide important cues to define and analyze the robots' interaction. In particular, the 3D shape of environments, including terrains and objects can be considered as motion constraints for the robot kinematics and dynamics. As an example, 3D SLAM (simultaneously localization and mapping) actively uses 3D reconstruction methods to generate surrounding environments and geometric relations between the robot and the environment[6]. In addition, constructing 3D models of unknown objects gives more information about the objects, which helps motion planning tasks [5, 9, 11].

Due to its importance in many fields, there have been many works on 3D reconstruction from image frames. Successful works like [15, 17] 3D reconstruct realistic representations of random objects such as texture and reflection. These methods construct 3D shapes of target objects and generate scenes at unseen view points by interpolating features from given scenes of the objects. However, they are computationally expensive and very slow, and they strictly require environments such as constant light. Therefore, these methods cannot be easily applied to real environments where dynamic changes in light conditions and limited ranges of available viewpoints worsen the performance of the model.

On the other hand, [21, 25] allows 3D reconstruction by using a monocular camera to generate the models of humans and animals. These methods take advantage of pre-trained models from a large scale of data and allow predicting 3D shapes of target objects without further image inputs at a wide range of viewpoints. However, these works assume the target of 3D reconstruction is limited to features in training datasets. They also require additional modules for image segmentation for capturing the areas of the target objects and usually use pre-trained masks for the image segmentation.

The above methods have shown successful deployment on specified tasks and remarkable performance, but they are not suitable for daily use. The strict requirements limit the fields where the methods can be applied and accompany complicated setups on hardware setup and deployment environments. Therefore, it is necessary to study a more general method for daily use though it compromises the performance of reconstruction. In addition, as a matter of user experience, active instruction for users would be a huge help for people who are not experts in the 3D reconstruction field.

2. Problem Statement

The goal of this project is to build an active 3D reconstruction model that reconstructs the 3D shape of certainty objects in real-time. We consider the target object of 3D reconstruction has a closed surface and each image frame from the camera is labeled with the camera's pose information. For the experiments, we used Apple iPhone which has an in-build inertial measurement unit along with cameras, and estimated the camera's pose by Apple's AR kit. When an object is given, the model starts real-time 3D reconstruction with a pre-trained model and will finish when the surface of the reconstructed shape is closed. Simultaneously, the model computes the most uncertain direction of view based on the reconstructed shapes and outputs instructions for a user to move the camera view for data collection for uncertain areas.

3. Related Work

3.1. 3D Reconstruction from Multiple Viewpoint Images

By taking advantage of utilizing GPU accelerated computation, there have been many works conducted on reconstructing shapes or predicting unseen scenes of 3D objects from multiple image frames. Neural Radiance Fields (NeRF) and extended works from this are considered as successful examples of 3D feature estimation by using interpolating on given multiple image frames [17]. The original NeRF work achieves not only successful 3D reconstruction but also a realistic rendering of the reconstructed shape, including light reflection and target objects' textures, by using neural networks. However, it suffers from expensive computation during inference steps, and the target environment must be strictly controlled, such as static target objects and constant light conditions. To overcome the limits of the work, many methods were considered as an extension of NeRF. [15] tackles the constraints of constant light conditions and successfully applies the NeRF method to objects in outdoor environments where light condition changes depending on the viewpoints. Also, [20, 19, 12, 24] improved the NeRF method and applied it into reconstruction of dynamic features. On the other hand, [8] extended the NeRF method to focus more on constructing object features rather than extracting only shapes, and the output of the network can represent multiple object shapes from sequential image frames. [26] focuses on that reconstructed 3D shapes provide information to estimate the camera's position and orientation at the image frame and built a framework of the camera pose by using NeRF outputs.

3.2. 3D Reconstruction from a Posed Camera

Other than the methods of interpolating multiple images from different viewpoints that require expensive computation, labels of camera position and orientation can provide more direct information to merge multiple image frames in real-time. The hardware setup for such a camera with measurements for the camera pose information is called a *posed camera*. In particular, due to the advance of technology in Micro-electromechanical systems (MEMS), many commercial camera products, such as smartphones, are usually equipped with inertial measurement unit (IMU) sensors at reasonable prices. Therefore, the methods to extract data of scenes from posed cameras can be applied to many areas and have been actively studied. [13] successfully estimates spatial depth information from 2D monocular camera images and evaluated the certainty of the depth estimation. Assuming access to the camera's pose information, we can extend the range of 3D reconstruction targets, not only limited to a certain point of view. In particular, by using the images taken multiple views, [18, 14, 23] 3D reconstruct a wide range of open environments. The above works do not aim at applications for an entire single target object, but we assumed that the method of using a posed camera can be applied to our target task.

3.3. Shape Representations

There are multiple ways to represent 3D shapes, such as voxels, meshes, and point clouds. However, these conventional representations are discrete, which have lots of limits when resolution increase. For example, the methods of voxels and meshes are expensive for storage. In addition, for meshes representation, it requires a non-trivial post-process to generate the final 3D shapes. In contrast, implicit neural representations are free from the above issues because it uses a continuous function to represent any formats, including 3D coordinates [16]. Recently, many works have shown the advantage of neural implicit representation with their neural network architecture for constructing 3D shapes [18, 23, 22]. Due to the above advantages, we use an implicit neural network to generate an occupancy map and improve the resolution of the reconstructed shapes in this project.

4. Methods

The pipeline developed in this project is presented in Figure 1. The pipeline consists of the posed camera hardware, the 3D reconstruction model, the rendering module for user visualization, and the viewpoint planner. In the following subsections, the details of each module are illustrated.

4.1. Posed Camera Hardware

We assume that a user can access an ordinary smartphone, including an HD-resolution camera and an IMU sensor. In particular, we used Apple's iPhone 12, which can record image frames up to 30 frames per second and equips with a 10-degree-of-freedom IMU, including a 3-axis accelerometer, a 3-axis gyro sensor, a 3-axis magnetometer, and a barometer [3]. The IMU measures spatial acceleration and angular velocity of the camera and estimates the orientation of the camera by fusing sensor data of gyro and magnetometer sensors. Even only with IMU data, we can estimate the camera orientation precisely enough. However, estimation of a spatial position of the camera suffers from drifts that occur when double integrating acceleration with noise. Therefore, we used Apple's ARKit to estimate the camera's position by a visual odometry package that synchronizes the spatial odometry of the camera with image frames [1]. Due to the limit of computing performance of the Apple iPhone, we cannot process onboard computing for inference of NeuralRecon's 3D reconstruction model. In addition, Apple ARKit limits real-time data transmission between the iPhone and a GPU machine to process the model's inference. Therefore, we collected data without running our active 3D reconstruction module and evaluated the module by executing it with offline data of posed image frames provided by the iPhone and ARKit.

4.2. 3D Reconstruction Model

To take advantage of GPU accelerated real-time construction of 3D shapes, we extended the model design of NeuralRecon, a data-driven model for generating 3D shapes [23]. The model consists of two parts of neural networks for merging a sequence of egocentric view images to infer 3D shapes and converting shape representation at the previous neural network into a representation of spatial occupancy. In particular, the first network outputs shape representations in the format of Truncated Signed Distance Function (TSDF), due to the easiness of converting data to other formats. More details of the 3D reconstruction network is shown at Figure 2 We used the model trained by dataset [7], which consists of multiple 3D scenes of indoor environments. Though the dataset focuses on training a model to reconstruct a 3D shape of indoor environments rather than a single object, we assumed the model can be trained to extract the 3D shapes of arbitrary features by the ScanNet. Therefore, we used a pre-trained model from a large scale of data in the ScanNet dataset, instead of generating a new dataset that would be specified for our tasks.

4.3. Computing the direction of most uncertain view point

Using the output surface and the uncertainty information, we know which area needs to be explored next. Given an object, we first take a video of this object, and input the video into a pre-trained Neural Recon network, as described in the previous sections. Neural Recon network will automatically generate an incomplete mesh model of the object. In order to achieve computational efficiency, we extract point clouds information from the constructed mesh model. Due to the output of NeuralRecon covering unnecessary areas from the raw image frames, we bound pointclouds that are used for computation. The size of the bounding box is given manually corresponding to the rough size of the object. The center of the box is given as the *center point* of the object, which is initially given manually but recursively updated as the spatial average point of the points of the objects, which would be described later in this subsection. Denote the set of those point clouds in the bounding box as $P = \{p_i\}_{i=1}^{N}$

For each p_i, we find k nearest points of p_i. In this paper, we choose $k = 3$. Let's denote 3 nearest points of p_i as $a^i = (a^i_x, a^i_y, a^i_z)$, $b^i = (b^i_x, b^i_y, b^i_z)$, and $c^i = (c^i_x, c^i_y, c^i_z)$, and thus we have a matrix

$$M = \begin{bmatrix} a^i_x & a^i_y & a^i_z \\ b^i_x & b^i_y & b^i_z \\ c^i_x & c^i_y & c^i_z \end{bmatrix}$$

then, applying Singular Value Decomposition to M,

$$M = U\Sigma V^*$$

the last column of U is normal vector of the best fitting plane of these three points a^i, b^i, and c^i. Normalizing the normal vector, and we get a unit normal vector, denote as n_i.

In this paper, we assume that objects to be constructed have "convex" shape, and the captured video also contains reasonably enough information. Therefore, we can assume that the mass center of point clouds points is inside of the object. Denote the mass center of $P = \{p_i\}_{i=1}^{N}$ as m. Thus, for each point p_i, if dot product $n_i \cdot (p_i - m) \leq 0$, change sign of normal vector n_i, if the dot product is positive, then nothing changes. We compute the directional vector at this point by the following formula

$$\frac{n_i \cdot (p_i - m)}{|||p_i - m|||^2}(p_i - m)$$

add all these direction vectors on the object together, and take the opposite, we get the real directional vector, which

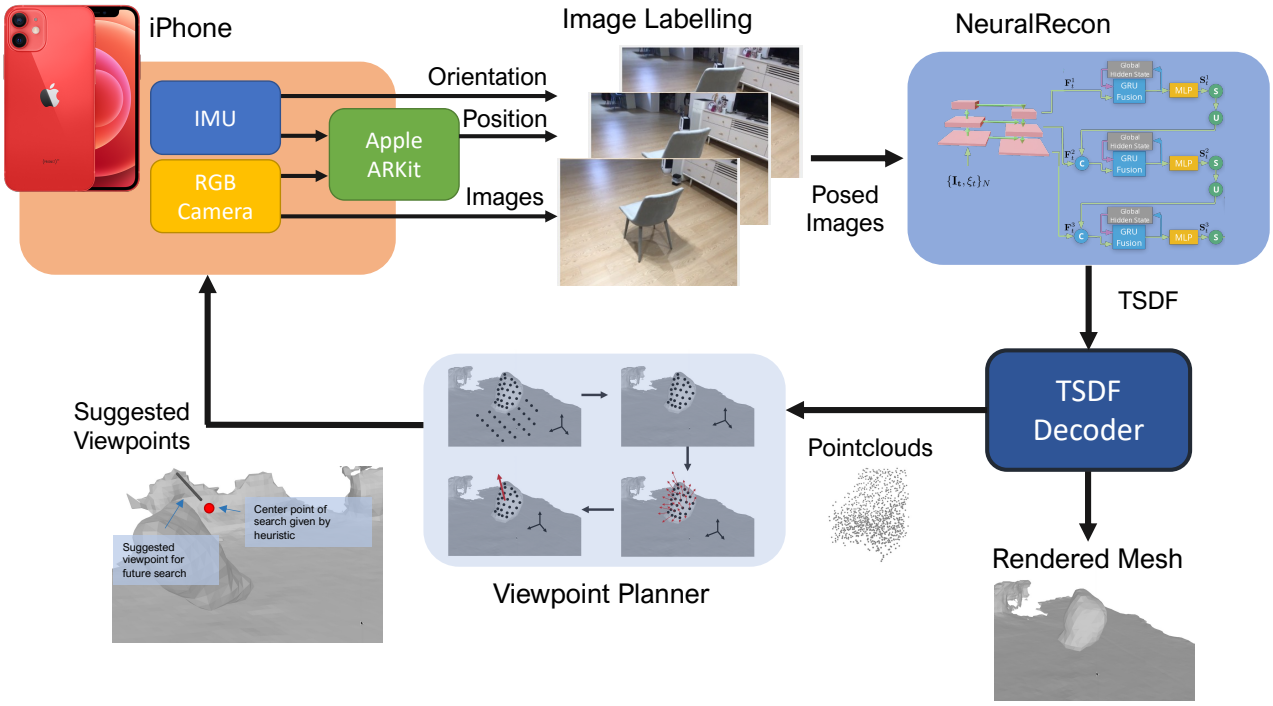

Figure 1. Illustration of the pipeline for active object 3D reconstruction: Apple iPhone captures RGB images of target objects and records sensor data of the camera's orientation, angular velocity, and acceleration. Orientation data are directly used for labeling image frames. To compute the camera's position, Apple ARKit integrates the IMU data and the image frames by using the visual odometry package, which is used for the labels of the camera position at image frames. Then, the 3D reconstruction network of the NeuralRecon model computes reconstructed 3D shapes in the form of TSDF from the posed images. The TSDF outputs are decoded as rendered meshes and pointclouds: rendered meshes visualize the reconstructed shapes for users, and pointclouds are inputs for computing suggested viewpoints for active instruction. In the pipeline, users move the camera by following the instruction of camera viewpoints, but, in this project, we do not implement real-time feedback between the user and the pipeline due to the limits of hardware and Apple ARKit, as described in 4.1. Instead, we used recorded posed image data as inputs of the pipeline and considered episodes that users do not follow the instructions. Though we used the offline data, the entire system runs in real-time, and we expect this pipeline can be applied in real-time feedback for active 3D reconstruction when the limits of ARKit can be resolved.

points to the direction to which user should move camera.

$$\sum_{i=1}^{N} \frac{n_i \cdot (p_i - m)}{|||p_i - m|||^2} (p_i - m)$$

One technical problem is that the output of the NeuralRecon network also includes lots of points on the ground. Because these points are not on the parts of the target objects and can affect the final directional vector significantly, we should not consider computing active instruction. Therefore, we need to exclude these points to compute the final directional vector. Considering we only 3D reconstruct one object, so, roughly speaking, points can only be on the ground or on the object, which implies there are two clusters of points. We then use the K-means algorithm (k=2) to find the centers of these two clusters of points. In this way, we can find the mass center of the object, and make the directional vector more accurate.

An initial surface of the 3D shape before the 3D reconstruction can be given as any random shape, such as a sphere. Uncertainty values of the unexplored area are initiated to the maximum value. The values are updated when the reconstruction of the area is processed from newly received image inputs.

One thing worth mentioning is for an extreme case if the object is too large, it is very likely the incomplete mesh model constructed for the first few times will not be convex, which means the mass center of the incomplete mesh model is outside of the object. In this case, we think there may not be possible to find a proper heuristic method to identify where is the inside of the object. More data-driven methods might be applied to help the situation like texture recognition.

5. Results

To evaluate the feasibility and the performance of the pipeline, we tested 3D reconstruction of the chair as shown

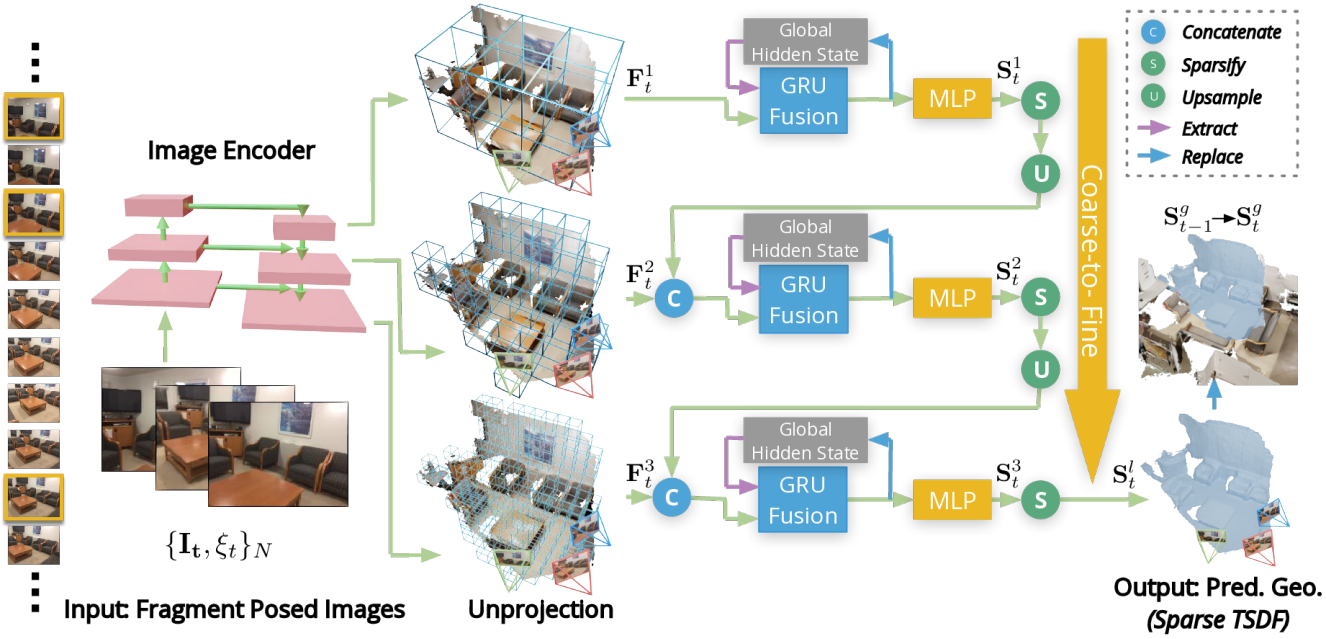

Figure 2. Illustration of the NeuralRecon method that was used as the 3D reconstruction module in this project: NeuralRecon predicts 3D shapes from image frames with a three-level coarse-to-fine approach that gradually increases the resolution. Images labeled with the camera pose are first passed through the image backbone to extract the multi-level features. These features are back-projected along with each level and integrated into a 3D feature volume. [23].

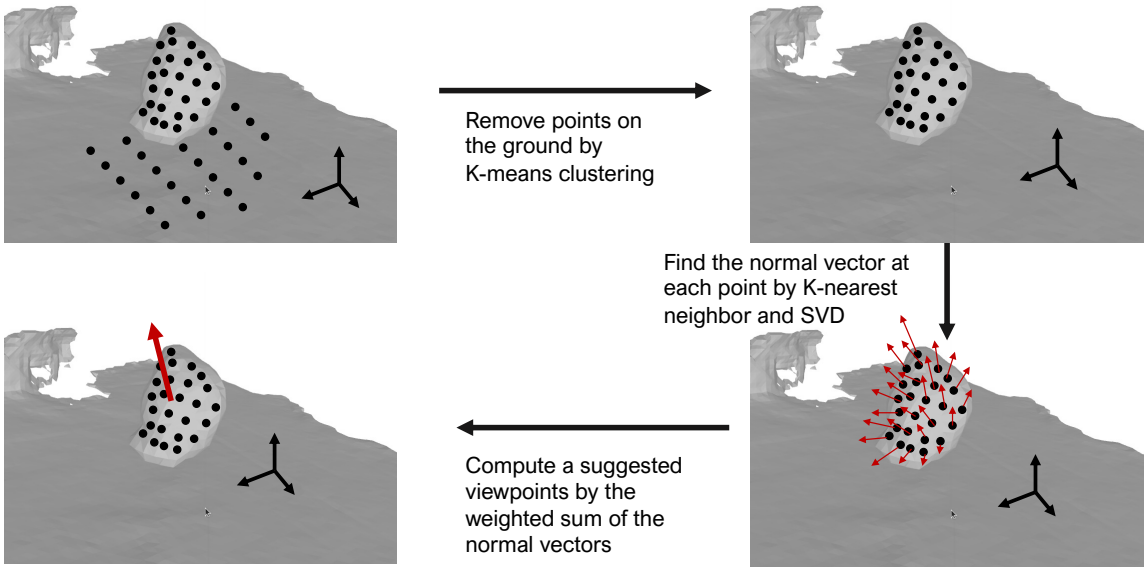

Figure 3. Illustration of the process of computing suggested viewpoint direction: Initially, we consider a downsampled number of points nearby the target object. Then the points on the ground are excluded by K-mean clustering method. Then, normal vectors at the filtered points on the object are computed by K-nearest neighbor and single value decomposition (SVD) method. Finally, the spatially weighted sum of the normal vectors at the points is output as a suggested viewpoint vector.

at 4, which has various features, including non-convex shapes and various thicknesses on its volume. Other than the chair, we also tested other simpler objects such as napkin boxes but did not include the detailed results from the

Figure 4. Target object of evaluating 3D reconstruction and viewpoint planing: we used an object which has various shapes, such as the main body's non-convex shapes and thin legs.

objects in this paper. Also, as described in 4.1, the iPhone 12 was used for collecting posed image frame data. For the evaluation, the data of 180 image frames were used for constructing 3D shapes and computing the most preferred future viewpoint direction. The images were taken while walking around the chair. The initial center point is given above the chair, and the size of the bounding box is given as $0.8 \times 0.8 \times 1.6\text{m}^3$. The results of the pipeline are displayed in 5, and evaluation and discussion on the results are presented in the following subsections.

5.1. 3D reconstruction

From the results of the reconstructed shapes at (1,2,3-a) of Figure 5, we can find that the reconstructed shapes include more details as more image frames were processed. However, the reconstructed shape could not include the shape of legs even after all the image frames were processed. Therefore, we could conclude that the pre-trained 3D reconstruction module can be applied to a single object but does not work well enough to include detailed features of small objects.

5.2. Active Viewpoint Planner

From the results of the suggested viewpoint directions at (1,2,3-b) of Figure 5, we can find that the viewpoint planner module outputs the downward directional vectors. Considering that we do not record the image frame dataset at a height lower than the chair, the areas below the chair may always have high uncertainty. Also, we could find that when the reconstructed shapes were not processed enough, the output vector changed continuously but jitters in the direction parallel to the ground plane at the end of the reconstruction. As a possible explanation of this issue, we can consider the case that the uncertainty in the areas below the chair remains higher than other areas reconstructed successfully. Among the higher uncertain areas, even small updates in the reconstructed meshes may result in significant noise, which would cause the jittering issue.

6. Conclusion

In this project, we developed an active 3D object reconstruction pipeline for inexperienced users with ego-centric cameras that have embedded IMUs. Sensor data from the embedded IMU are used directly or integrated with RGB image data for estimating the position and orientation at each image frame of the camera. The pipeline can reconstruct shapes of target objects from posed image data on target objects, and output the most preferred directions for users to move their cameras.

As for extension works, several methods can be applied to improve the performance of 3D reconstruction. For example, better estimation for camera pose and more camera viewpoints can provide more precise outputs. We can use a robot manipulator to receive precise pose information of the camera from the robot joint encoders. Also, we may be able to achieve better results in outdoor environments because a large scale of object and scale would not suffer from pose estimation errors and reconstruction resolution.

Contribution of Each Member

Mingyo Seo

- Writing of the milestone, the presentation, and the final paper
- Literature search
- Design of the pipeline
- Building the development environments
- Collecting posed image data
- Visualization of the results
- Discussion on the technical details of methods
- Cleaning up the codes and debugging

Zhou Fang

- Writing of the details on the subsection 4.3 in the final paper
- Formulation of the viewpoint planner module
- Implementation of the active viewpoint planner module

References

[1] Arkit overview - apple developer - augmented reality. https://developer.apple.com/augmented-reality/arkit/. Accessed: 2021-12-11. 3

[2] Augmented reality - apple. https://www.apple.com/augmented-reality/. Accessed: 2021-12-11. 1

[3] iphone 12 and iphone 12 mini key features - apple. 3

[4] Microsoft hololens — mixed reality technology for bussiness. https://www.microsoft.com/en-us/hololens. Accessed: 2021-12-11. 1

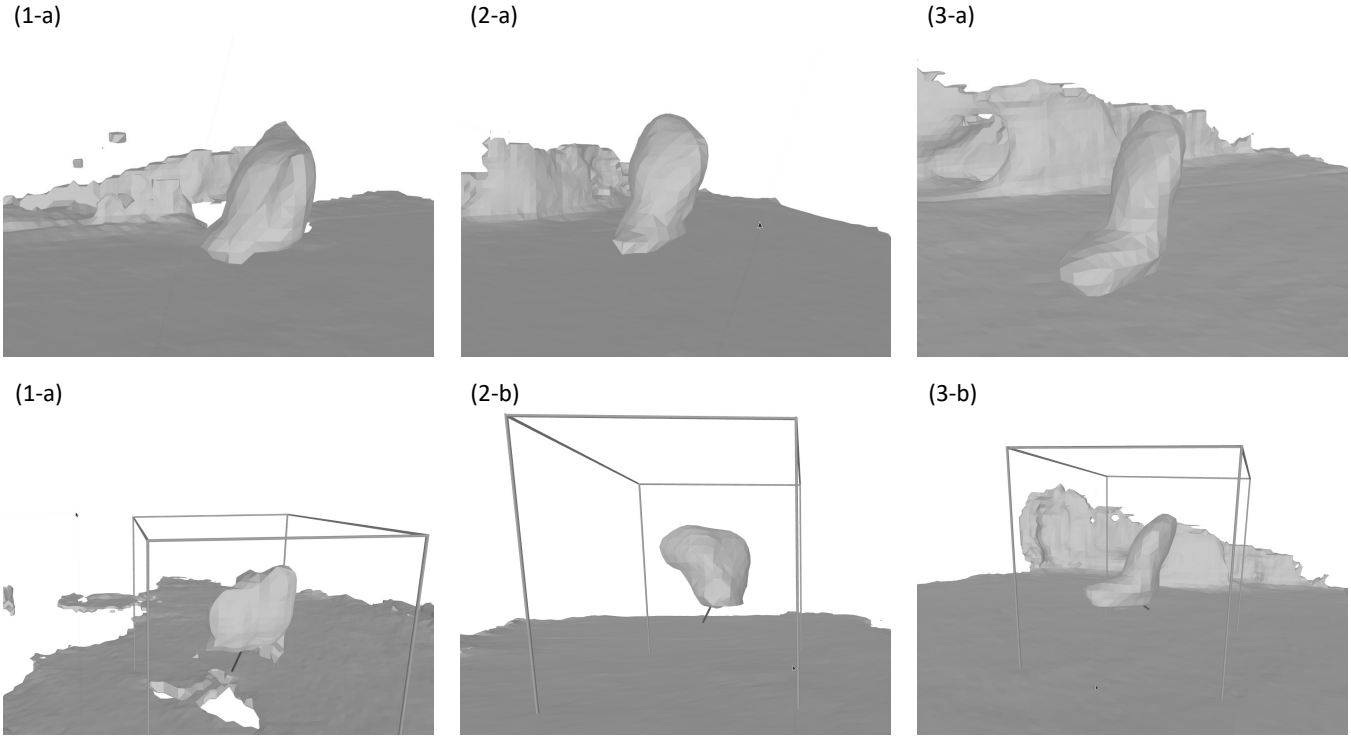

Figure 5. Results of 3D reconstruction and active viewpoint planner: the rendering images of the above row present the reconstructed shapes as more pose image frames are processed in the order from 1 to 3. The rendering images of the below row present the output results from the viewpoint planner corresponding to the time step at the above row. The gray box illustrates the bounding box to exclude unnecessary points that are located away from the object. The black segments display the output results of the spatial sum of the normal vectors at the viewpoint planner, which would be used as the suggested viewpoint direction for users.

[5] Michel Breyer, Jen Jen Chung, Lionel Ott, Roland Siegwart, and Juan Nieto. Volumetric grasping network: Real-time 6 dof grasp detection in clutter. *arXiv preprint arXiv:2101.01132*, 2021. 1

[6] David M Cole and Paul M Newman. Using laser range data for 3d slam in outdoor environments. In *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.*, pages 1556–1563. IEEE, 2006. 1

[7] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5828–5839, 2017. 3

[8] Michelle Guo, Alireza Fathi, Jiajun Wu, and Thomas Funkhouser. Object-centric neural scene rendering. *arXiv preprint arXiv:2012.08503*, 2020. 2

[9] Jung-Su Ha, Danny Driess, and Marc Toussaint. Learning neural implicit functions as object representations for robotic manipulation. *arXiv preprint arXiv:2112.04812*, 2021. 1

[10] Shahram Izadi, David Kim, Otmar Hilliges, David Molyneaux, Richard Newcombe, Pushmeet Kohli, Jamie Shotton, Steve Hodges, Dustin Freeman, Andrew Davison, et al. Kinectfusion: real-time 3d reconstruction and interaction using a moving depth camera. In *Proceedings of the 24th annual ACM symposium on User interface software and technology*, pages 559–568, 2011. 1

[11] Zhenyu Jiang, Yifeng Zhu, Maxwell Svetlik, Kuan Fang, and Yuke Zhu. Synergies between affordance and geometry: 6-dof grasp detection via implicit representations. *arXiv preprint arXiv:2104.01542*, 2021. 1

[12] Zhengqi Li, Simon Niklaus, Noah Snavely, and Oliver Wang. Neural scene flow fields for space-time view synthesis of dynamic scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6498–6508, 2021. 2

[13] Chao Liu, Jinwei Gu, Kihwan Kim, Srinivasa G Narasimhan, and Jan Kautz. Neural rgb (r) d sensing: Depth and uncertainty from a video camera. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10986–10995, 2019. 2

[14] Xinzhu Ma, Zhihui Wang, Haojie Li, Pengbo Zhang, Wanli Ouyang, and Xin Fan. Accurate monocular 3d object detection via color-embedded 3d reconstruction for autonomous driving. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6851–6860, 2019. 2

[15] Ricardo Martin-Brualla, Noha Radwan, Mehdi SM Sajjadi, Jonathan T Barron, Alexey Dosovitskiy, and Daniel Duckworth. Nerf in the wild: Neural radiance fields for unconstrained photo collections. In *Proceedings of the IEEE/CVF*

Conference on Computer Vision and Pattern Recognition, pages 7210–7219, 2021. 1, 2

[16] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3d reconstruction in function space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4460–4470, 2019. 2

[17] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *European conference on computer vision*, pages 405–421. Springer, 2020. 1, 2

[18] Zak Murez, Tarrence van As, James Bartolozzi, Ayan Sinha, Vijay Badrinarayanan, and Andrew Rabinovich. Atlas: End-to-end 3d scene reconstruction from posed images. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part VII 16*, pages 414–431. Springer, 2020. 2

[19] Keunhong Park, Utkarsh Sinha, Jonathan T Barron, Sofien Bouaziz, Dan B Goldman, Steven M Seitz, and Ricardo Martin-Brualla. Nerfies: Deformable neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5865–5874, 2021. 2

[20] Albert Pumarola, Enric Corona, Gerard Pons-Moll, and Francesc Moreno-Noguer. D-NeRF: Neural Radiance Fields for Dynamic Scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020. 2

[21] Shunsuke Saito, Tomas Simon, Jason Saragih, and Hanbyul Joo. Pifuhd: Multi-level pixel-aligned implicit function for high-resolution 3d human digitization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 84–93, 2020. 1

[22] Vincent Sitzmann, Julien Martel, Alexander Bergman, David Lindell, and Gordon Wetzstein. Implicit neural representations with periodic activation functions. *Advances in Neural Information Processing Systems*, 33, 2020. 2

[23] Jiaming Sun, Yiming Xie, Linghao Chen, Xiaowei Zhou, and Hujun Bao. Neuralrecon: Real-time coherent 3d reconstruction from monocular video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15598–15607, 2021. 2, 3, 5

[24] Wenqi Xian, Jia-Bin Huang, Johannes Kopf, and Changil Kim. Space-time neural irradiance fields for free-viewpoint video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9421–9431, 2021. 2

[25] Gengshan Yang, Deqing Sun, Varun Jampani, Daniel Vlasic, Forrester Cole, Huiwen Chang, Deva Ramanan, William T Freeman, and Ce Liu. Lasr: Learning articulated shape reconstruction from a monocular video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15980–15989, 2021. 1

[26] Lin Yen-Chen, Pete Florence, Jonathan T Barron, Alberto Rodriguez, Phillip Isola, and Tsung-Yi Lin. inerf: Inverting neural radiance fields for pose estimation. *arXiv preprint arXiv:2012.05877*, 2020. 2