3D RECONSTRUCTION OF OBJECTS FROM RGB IMAGES AND DEPTH INFORMATION USING DEEP LEARNING

Tharindu Dananjaya Karunanayaka

189327E

This dissertation submitted in partial fulfillment of the requirements for the Degree of MSc in Computer Science specializing in Data Science and Analytics

Department of Computer Science and Engineering
Faculty of Engineering

University of Moratuwa
Sri Lanka

October 2022

DECLARATION

I, K.N.T.D.Karunanayaka, hereby declare that this is my own work and this report does not incorporate without acknowledgement any material previously submitted for the degree or diploma in any other university or institute of higher learning and to the best of my knowledge and belief it does not contain any material previously published or written by another person except where the acknowledgement is made in the text.

Also, I hereby grant University of Moratuwa the non-exclusive right to reproduce and distribute my dissertation, in whole or in part in print, electronic or other medium. I retain the right to use this content in whole or part in future works (such as articles or books).

Signature: Date:

The above candidate has carried out research for the Master's thesis under my supervision. I confirm that the declaration made above by the student is true and correct.

Name of the supervisor: Dr. Charith Chithraranjan

Signature: Date:

ACKNOWLEDGMENTS

I owe my deepest gratitude to my supervisor, Dr. Charith Chithraranjan of Department of Computer Science and Engineering, Faculty of Engineering, University of Moratuwa for the supervision and advice given throughout to make this research a success.

Further, my sincere appreciation goes to my family for the continuous support and motivation given to make this thesis a success. I am also thankful to the management and the staff of LiveRoom (Pvt.) Ltd for supporting me in balancing my workload which allowed me to spend time on this research.

Moreover, I would like to thank all my colleagues for their help in finding relevant research material, sharing knowledge and experience and for their encouragement. Last but not least, I also thank my friends who supported me in this whole effort.

ABSTRACT

Object reconstruction is the manner of producing a computer model of the 3D appearance of an object from two-dimensional photos. It's the opposite procedure of obtaining 2D photos from 3D scenes. 3D reconstruction of objects from their digital pictures is a time-efficient and convenient manner of analysing the structural features of the item being modelled. Currently there may be an essential need for 3D content for computer graphics, virtual reality and communication, triggering an alternate emphasis for the requirements. Many present methods for constructing 3D objects are built round specialized hardware resulting in a high fee, information scanning barriers due to environment conditions which can't satisfy the requirement of its new programs. The art of three-dimensional reconstruction of objects and scenes has been a broadly researched topic.

In this Master's thesis, I proposed to address the above problems by developing a Deep Learning approach to reconstruct the object. This type of approach does not depend too much on the environment condition and the cost is low. However, the proposed method mostly targets the reconstruction of objects other than reconstruction of scenes. This research attempts to develop a Deep Learning based 3D reconstruction method for objects to avoid the limitations of the current 3D reconstruction approaches.

TABLE OF CONTENTS

LIST OF FIGURES

LIST TABLES

LIST OF EQUATION

LIST OF ABBREVIATIONS

SFM	Structure from Motion
MVS	Multi View Stereo
SLAM	Simultaneous Localization and Mapping
DIP	Dots per Inch
CNN	Convolutional Neural Network
DVR	Differentiable Volumetric Rendering
3D-R2N2	3D recurrent Reconstruction Neural Network
DRC	Differentiable Ray Consistency
PSR	Poisson Surface Reconstruction
MLP	Multilayer Perceptron
NV	Neural Volumes
SRN	Scene Representation Networks
LLFF	Local Light Field Fusion
2D-CNN	2D Convolutional Neural Network
LSTM	Long Short Term Memory
3D-LSTM	3D Convolutional Long Short Term Memory
3D-DCNN	3D De-Convolutional Neural Network
FID	Frechet Inception Distance
AR	Augmented Reality
GAN	Generative Adversarial Network
3D-R2N2	3D Recurrent Reconstruction Neural Network
IoU	Intersection over Union
SDF	Signed Distance Function
FC	Fully Connected
DVR	Differentiable Volumetric Rendering
ED	Euclidean Distance
SPSR	Screened Poisson surface reconstruction

1. INTRODUCTION

1.1 Motivation

3D reconstruction of objects is about getting information from the scenes or objects and generating a 3D model that looks closer to the original. It is the fastest and easiest technique to create a 3D model. There are many applications related to this. In the last few years, 3D reconstruction has evolved dramatically. Much research has been carried out to solve the problem of 3D object reconstruction. A 3D scan is a three-dimensional image of an object or a scene. The result of the 3D scanning is a mesh. A mesh is consisting of a texture and geometry. Geometry is created using polygons. Texture information is used to represent the appearance of the object.

There are three main components of a 3D scanner. They are sensors, light sources and a system that had to do the analysis. For any 3D scanning method above three are common. Up to now, there are lots of 3D reconstruction methods have been proposed. Each has its own advantages and disadvantages. Some are cheap and do not produce good results, some are so much expensive that a person could not bear the price. With the emerging technologies like AR, VR, and computer graphics, there is a high demand the 3D reconstruction. There are wide applications in 3D reconstruction. Currently, there are lots of research going on how to build a system with low cost but with greater accuracy. 3D reconstruction is really challenging due to the uniqueness of the problem. Each new reconstruction can be considered a new problem and based on that certain measures need to be taken. Due to this, 3D reconstruction has become a challenging problem.

There is a high demand for novel 3D reconstruction method with better performance, low cost and easy scanning. Today there are lots of researches going on in this area to cater the above requirements. Digital 3D product models are starting to replace mainstream 2D image visualization within online stores. This immersive technological advancement provides the shoppers with an interactive experience that delivers more details about the product which could only be achieved with 3D models. By using the 3D models one can better analyse a system. Using a 3D model instead of a 2D image provides a better understanding of the objects. In the future, there will be a trend toward 3D objects rather than classical images in e-commerce. It is predicted that in the future many people prefer 3D objects to better analyse the product they wish to buy. 3D Scanners can be used to create digital models for video games and virtual cinematography. It can scan objects instead of creating the models from scratch. So there are lots of benefits for the entertainment industry. Another main use case of 3D scanning is AR. The combination of augmented reality and 3D scanning is a natural fit. Currently, there is a trend towards building metaverse. 3D reconstruction will take a prominent place in building metaverse like systems. So there is a wide variety of applications in 3D reconstruction technology.

There are two main techniques used to generate the 3D model from real-world information. One is photogrammetry and the other is laser based scanning. Each of these methods has its own advantages and disadvantages. Laser-based solutions

provide accurate results. But the cost of using this system is very high. The cost is too much so the individual finds it difficult to bear it. Laser-based scanning use dedicated software and hardware to get the final result. This method can generate fine-tune information of objects as well.

When considering the photogrammetry solutions, the initial investment is low, but the results are not accurate as a laser-based system. This solution is mainly affected by the environment and its lighting conditions. Photogrammetry provides the best result when there are textures in the object. It is difficult to generate the areas of the object with low textures from this method. So the main drawback of this technology is it can only reconstruct objects with high textures only. For the less textured objects, this method does not work and does not give good results. And also the meshes obtained from this method are not clean and a 3D artist has to modify them with considerable effort before it is used by anyone. These methods have long been blamed for their limitations. 3D data generation often requires expensive data collection as well.

Currently, there is a trend for the 3D scanners in such a way that the solution should be low cost and the quality of the result should be good. Also the scanning should be easy as well. Both the photogrammetry and laser scanning techniques cannot achieve both of these requirements. So a novel solution is needed to achieve the above requirement.

1.2 Problem Statement

The Problem that this study tries to address can be stated as;

3D reconstructs objects from RGBD information using Deep Learning

This project hopes to use a Deep Learning based approach to predict the geometry. Because this method does not depend too much on texture and lighting conditions where traditional methods do.

1.3 Objectives and Output

To deal with the research problem, I need to accomplish the below objectives.

- To conduct a comprehensive literature survey on technologies used
- Data acquisition
- Predict the geometry of the scan
- Texture the geometry
- Evaluate the overall performance of the proposed method

1.4 Outline

The document is structured as below. Chapter 2 contains theoretical aspects of different methods used for the 3D reconstruction. Chapter 3 provides the method for the 3D reconstruction. Finally, chapter 4 contains the result and analysis.

2. LITERATURE REVIEW

Section 2.1 describes the basics of 3D scanning. It discusses the types of scanners and methods used for 3D reconstruction. This section also has discussed their advantages as well as their limitations. Section 2.2 focuses on Deep Learning based approaches. This section summarizes the researches that has been carried out in 3D reconstruction related to deep learning.

2.1 3D Scanning Technology

In 3D scanning there are two main steps, data acquisition and data processing. When capturing the data there are several factors to be considered. Accurate, fast, reliable and cost efficient are the main factors. Data processing involves after the data capturing process. The software should be able to process the raw data acquired from the hardware and further process it. At the software it will remove the noise and distorted data and do the final processing to get the 3D model.

There are different kinds of 3D reconstruction methods are available.

- Laser triangulation 3D scanning technology
- Structured light 3D scanning technology
- Photogrammetry
- Contact based 3D scanning technology
- Laser pulse based and phase shift 3D scanners
- Deep Learning based methods

2.1.1 Laser Triangulation 3D Scanning Technology

This is a non-contact, non-destructive technology. These scanners expose beams to the scanning object after that the sensors will catch their reflections. There can be single or multiple sensors placed to catch the reflections. Most of the scanners have a ready-made setup so the information can capture with good accuracy. The angle of reflection, the distance between the scanning object and sensors are important modules in this scanning. Here the scanning device is easy to handle. User has to capture the whole surface of the object. Large amount of points can be collected in a short amount of time. So this method can generate meshes with fine details. This method can generate the point cloud while scanning. After that it is converted to mesh using provided software.

These types of scanners provide accurate results. The results can be obtained within a short period of time. Laser scanners can be used to reconstruct complex geometries. Also it can be used to reconstruct object which have shiny, dark, tough surfaces. Since these types of scanners are operating in dedicated locations/studios, the effects due the environment and its lighting is minimal.

2.1.2 Structured Light 3D Scanning Technology

These types of scanners projects different light pattern to the scanned object. It uses a sensor system to measure the changes for the light pattern and based on that it measures the shape of the object.

Structure light scanning requires a minimal setup to start the scanning process. This has the capability to automate the entire process, that is from the data acquisition to final mesh generation. Scanning system projects different light patterns to the object. Camera system identified all the the patterns and their changes due to the object. These changes helps to identify the shape of the object. This method also provides accurate results with in short period of time. There are two major methods of stripe pattern generation that have been established. Those are interference and projection. These types of 3D scanners can be handheld or tripod mounted. The stability of the scanning device is very import.

These scans are very fast and can scan a larger area. High resolution structured light 3D scanners are also available. There are two types of scanners. One is stationary scanner and other is handheld scanners. Stationary scanners need to keep still when scanning while handheld scanners can move around the object to do the scanning. Handheld scanners provide accurate result because it can scan the entire object while moving.

2.1.3 Photogrammetry

This is the most widely used method to generate meshes. Here the user has to take images of the object from different viewing angles. Then those images are feed to a special build software package and it will output the 3D model. This is a low cost solution and also provide comparable results as well. But this method has several drawbacks. Photogrammetry can generate object which has high textures. The results are also depending on the environment, lighting conditions and etc. There are softwares available for photogrammetry. Colmap [4], OpenMVG, Meshroom, VisualSFM and Reality Capture are some of them. This software will calculate the camera intrinsic, extrinsic sparse point cloud using SFM [3]. After that it uses MVS [7] to reconstruct the geometry of the 3D model. Finally, it will texture the mesh to get the colored mesh.

There are two main types of photogrammetry solutions based on the camera location when scanning. They are close range photogrammetry and aerial photogrammetry. Photogrammetry solutions provide compatible results. To have a good outcome the resolution of the scanning images should be high and the images should capture in good lighting conditions. The downside of this method is that it takes some time to run and it can only create objects with high texture information. Low texture objects will not get a good final outcome.

2.1.4 Contact Based 3D Scanning Technology

Physical contact of a probe onto the surface of the object being scanned is the method of this scanning. The scanning is done in a controlled environment. So there will not be much effect from environment lighting. In some systems an artificial arm is used to move the probe along the surface of the object. This will provide better results than handheld scans. These probes capture the information from different angles. These types of scanners can capture objects and their surfaces with high precision. To create an accurate model, a sufficient amount of points on the surface need to be captured. Here the object should not move and it should keep still. Contact based scanners can capture the complex surface information accurately.

One of the main advantages using this method is that, reflective and transparent surfaces can be accurately scanned. This method can provide accurate results when compared to other related technologies. The probe takes some time to move all over the object. So slow speed is a disadvantage with contact based 3D scanning. The are several applications in contact based scanning. This technique is used in fabrication industry for quality control. This method can be used to find any damages occurred to the fabric. Contact based scanning is a widely used technique in 3D scanning.

2.1.5 Laser Pulse Based and Phase Shift 3D Scanners

There are two major formats of Long range 3D scanners. They are;

- Pulse based
- Phase shift

These scanners are mainly used to capture the large objects (buildings, aircrafts and etc). Pulse based and phase shift scanners are used to capture medium range objects (automobiles, generators and etc) as well.

2.1.6 Laser Pulse Based 3D Scanners

This types of scanners also knows as time of flight scanners. The accuracy of these scanners are high. Here scanner project a light beam, when the beam collided with the object the beam is reflected. There is a sensor system that capture the reflected beam. Then based on the time to arrive the beam, system can calculate the object shape. This scanner can scan up to full 360 degrees around itself by rotating the laser.

2.1.6.1 Laser Phase Shift 3D Scanners

This is also a time of flight 3D scanner. The working principle of this method is similar to laser pulse based scanners. The results from this scanner is very accurate. There are several benefits in this scanner.

- Fast scanning
- Portable
- Can scan large amount of point per second
- High accuracy
- Can scan high scale objects
- Non-contact scanning

2.2 Deep Learning Approaches

To solve computer vision tasks researchers have used Deep Learning methods. Many researchers are involved in Deep Learning based methods to provide solutions to different problems. 3D reconstruction is also a similar problem that the researchers are trying to solve using Deep Learning. Since the problem is in its early stage there are many opportunities. There are many solutions available for 3D reconstruction using Deep Learning from multiple images and single images.

Currency, CNN models provide best results for most of the image related problems. Background removal, segmenting images and other vision problems. These same basics can be used for the 3D reconstruction as well. Since 3D reconstruction is a complex problem, Neural Networks based solutions will provide good results. Also this will help to mitigate issues due to the environment conditions. There have been several projects developed to address the 3D reconstruction problem. So it is really important to research and study about those methods. In the next section several deep learning approaches and their usage is discussed.

There are several applications on 3D reconstruction. Car driving, design cities, metaverse, AR and VR are some of the applications that are currently in high demand. Therefore, there will be good demand for these kinds of applications in the future. Extraction of 3D information is in high demand. Many researches have been carried out by academics as well as industrial organizations to solve the problem of extracting 3D information. The task is extremely challenging due to dimensionality, environment related issues etc. There have been two main types of experiments,

- Multi view reconstruction
- Single view reconstruction

When considering these two each one has its own advantages and disadvantages. Multi view reconstruction can provide better results than single view reconstruction. But the result from single view reconstruction is faster than the multi view reconstruction.

Below are the major Deep Learning researches carried out related to 3D reconstruction.

2.2.1 Perspective Transformer Nets

In this method authors have developed a neural network which can predict the shape of an object using a RGB image [11]. Here, they focus on learning 3D format regardless of color and texture. They also simplified the problem by assuming that the scene is a clean white background and that light is a constant source of natural light. They use a 3D volume representation where each voxel is a binary unit. Here they proposed a prediction method to generate the 3D mesh. For that they have used convolutional network. The encoder network learns the hidden representation of the vision that is then used by the decoder network to produce volume. The mask (2D silhouette) of the 2D images of the object can be obtained easily due to the white background. Researchers have obtained the 2D silhouette by using the 3D model and camera parameters. They propose a silhouette-based volumetric loss function.

The basis of their approach is the 2D silhouette produced under certain camera views should match ground truth 2D silhouette in image views. They think that the volume produced should be same as visual hull representation class of the respective GT volume. They have defined a loss function for 2D silhouette as well. Here they used only the image to predict the geometry of the object. If the IoU between original silhouette and generated are close to one, then we have a good prediction.

For this training ground-truth volumes are not needed. For each point in the 3D world frame, they compute the corresponding point in the camera frame with the help of camera extrinsic parameters. To get 2D silhouettes from 3D model, they suggest an easy method using max operator which flatten a large 3D space output across all dimensions. They use a max operator only for the prediction. Here the volume is represented as a cube in which only have binary values zero and one. If the volume has voxels it has a value of one and if it has empty voxels then the value is zero. In this experiment, they assumed that the conversion matrix was always provided as an input which is parameterized by the viewpoint. This is known as the camera extrinsic.

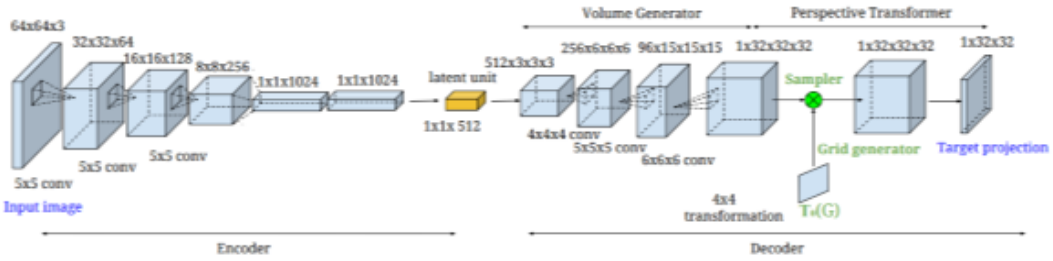


Figure 2.1: Transformer Nets network architecture [11]

To learn the 3D object representation multiple images of the objects from different views are needed. The more information you have, the more accurate the results will be. The researchers have implement a Deep Learning method to predict the geometry. It includes two methods. t the beginning they learn the 3D representation using the encoder network. Then they train a volumetric decoder with perspective transformer networks. To predict the 3D shape accurately, they also include viewpoints from neighbouring projections. Their Network structure consists of three parts, perspective transformer network, 2D convolutional encoder and 3D up-convolutional decoder. Below is an illustration of network architecture.

For stochastic optimization, authors have used the ADAM solver. They train the neural network model based on three loss functions. Authors use the ShapeNetCore database. This has more than fifty thousand scanned objects with more than fifty different model types. They use only projection loss, volume loss, and combined loss training models. They trained three models under two test settings. One is multiple categories and second is single category.

To evaluate the results, they generate one volume per view image. After that they compute the IoU value and the mean IoU value over all volumes. From the visualization results which they have provided, all three models predict volumes reasonably well. This method provides good results in generating the 3D object, even though only the images are used. Also in this project no 3D volume is used as the supervision. So this method can be scalable.

2.2.2 3D-R2N2

Christopher B. Choy and team have proposed a method to achieve 3D reconstructions of an object and they have proposed a novel RNN architecture which is called 3D-R2N2 [10]. It maps the images of objects to their underlying 3D shape by using arbitrary viewpoints of an object and outputs a 3D occupied grid. 3D-R2N2 doesn’t need any image annotation or label of the object class for training and as well as for testing processes.

Figure 2.2: 3D-R2N2 high-level architecture [10]

Main key features of the 3D-R2N2,

- Introduced RRN Network which is based on LSTM framework
- Integrate single and multi-view reconstructions
- Supervision needed is minimal for both testing and training
- Single view reconstructions are better that state of art methods
- Perform well for objects where SFM fails

The network is made up of three components: novel architecture called 3D-LSTM,2D-CNN, and 3D-DCNN. The input to the system can be a single image or set of images. First 2D-CNN encode its input to features. Those features are low dimensional features. Then 3D-DCNN predict the LSTM units. After that it will predict the voxel representation of the interested object. For the training it requires lots of images. Therefore, synthetic data has been used for the training (used CAD models with no background). The input image is square and its size is hundred and twenty-seven pixels (both vertically and horizontally). The reconstruction quality is low as well. This network has been trained for sixty thousand iterations with a batch size of thirty-six. They have used theano framework and Adam as the optimizer.

For the training authors have used lots of datasets. Not only for training they used those for validation of the model as well. A subset of Shape Net data set is used. It includes fifty thousand models with thirteen categories. Also they have used PASCAL 3D dataset. It has more than twenty thousand products. This database does not have 3D CAD models. This database is therefore only used for qualitative evaluation. The authors tested the network for five configurations. They trained and tested the network for Shapenet data set and used PASCAL dataset for the evaluation. Below table shows the evaluation results for the 3D-LSTM with different configurations.

From the mentioned five variants Res3D-GRU-3 method performs well. It provides the best results when using real images. It is a good outcome. Also they try to input as many images as possible for the training. They noted that when they increase the image count the results are getting better. So when the network see more images of

	Encoder	Recurrence	Decoder	Loss	IoU
3D-LSTM-1	simple	LSTM	simple	0.116	0.499
3D-GRU-1	simple	GRU	simple	0.105	0.540
3D-LSTM-3	simple	LSTM	simple	0.106	0.539
3D-GRU-3	simple	GRU	simple	0.091	0.592
Res3D-GRU-3	residual	GRU	residual	**0.080**	**0.634**

Table 2.1: 3D-R2N2 Performance

the scans, reconstruction results get better in Res3D-GPU-3 method.

Then they have compared their approach with MVS reconstructions which has different textures with different number of views. They have used high quality images of CAD models with augmentation. MVS method failed completely when view count is less than 20. This model works regardless of the degree of texture of the material where the MVS method failed to generate geometry when the object has low texture levels.

There are some limitations in 3D-R2N2. This method does not perform well in reconstructing objects which have a low degree of textures. The details in the MVS is high when compared to the 3D-R2N2. But this model need only small number of images to generate a 3D reconstruction. But upon increasing the number of images model will provide better results. This method is an alternative to the issues in MVS based reconstructions.

So we can conclude that 3D-R2N2 provide comparable result to other methods. Authors have used evaluation techniques to prove this. The proposed novel 3D-R2N2 architecture works well. Researchers have tested the system for real images. To increase the performance of the system they try to add more images as possible. They have perform five experiments by changing the configurations in the proposed network. From those Res3D-GPU-3 methods provide best results. Its IoU is high and loss is low.

2.2.3 DeepSDF

The researchers of the paper introduced a method called DeepSDF[13]. This method is able to generate mesh even without the full information of the object. It learned a representation called SDF. DeepSDF provides comparable results to other 3D reconstruction methods.

For the DeepSDF, they have used a neural network. The network is consist of eight layers. In this neural network, there is a skip connection as well. SDF is a function

that provides the length of the nearest point in the mesh to a given point. They have used normalization techniques to get better results. The researchers have used relu as the activation function. Also, they have applied twenty percent of dropin for the eight layers. By doing this optimization they were able to get good results.

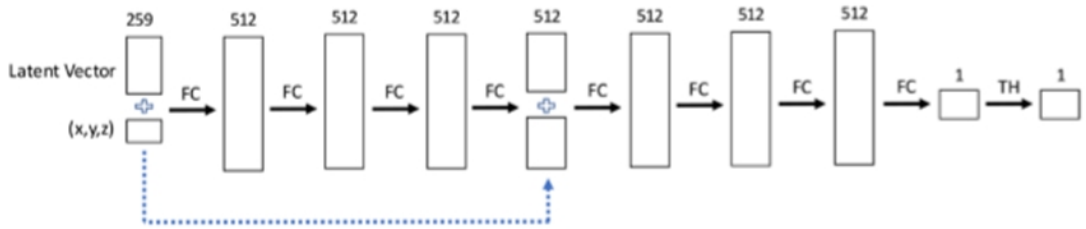


Figure 2.3: DeepSDF network architecture [13]

Researchers obtained training meshes first. After that, they collected the SDF samples from those meshes. They created a training data set for the format they required to input to the system. They render each mesh and obtained the images from different angles. Before this authors normalize each shape. Here they also used the KD tree. They used this to get a better output. For each shape, they generated SDF samples. The authors took many random points in the mesh, this is directly in line with the number of triangular faces of the mesh. Researchers got an approximate distance from the KD tree for the surface. They have used a different method to increase data points. They have applied a noise to every point. The result for this method gets better when using more samples. They have learned the shape of the object as a decision boundary.

Initially, they trained a network that fit each category. But with the introduction of the latent space, they were able to train a common network that helps to get the desired results. When training authors have used various techniques to get the best results. The input to the system was SDF values and 3D points of each mesh. Initially, they started with small latent vectors. Also, they have set the decoder learning rate as small as possible. As the optimizer, they have initially used Adam optimizer. But after that, they moved to gauss newton as it gives accurate results. When inferencing they provide a point cloud (to model the real-world scenario they used a noise point cloud). Providing a point cloud gives lots of information to the system rather than proving an image. Researchers trained the network for eight hours with thousand epochs. By adding this optimization, they were able to get accurate results.

This method does not consume much memory. At inferencing, input to the system is a noisy point cloud. Due to that much information is received at inferencing. Authors have done lots of experiments to show how well DeePSDF performs with other methods. These results show comparable results to other methods at that time.

Here the prediction can be done using any number of SDF samples. This is the main benefit of this proposed method. Due to this, completion of shape is at a good level. The authors carry out lots of experiments to exhibit the representational power of

DeepSDF using different methods. This method is evaluated for known shapes as well as for unseen shapes.

Figure 2.4: Result Comparison - DeepSDF vs AtlasNet [13]

2.2.4 A Point Set Generation Network

In this research a point cloud is generated when an image is provided. So called the point set generation network [15]. The input to the system is a RGB image and output is a point cloud. From the point cloud user has to create the mesh. This can be done using various methods. The authors designed the architecture, loss function and learning paradigms. Those are effective as well as novel. This solution is able to predict multiple point clouds from a single RGB image.

Here authors want to generate a complete 3D representation using a single RGB image. Researchers were able to do it to some extent. They present the object using point cloud. But they faced three issues. They have to decide the architecture of the point set generator network. For the proposed architecture there were two branches. First one mainly targeted on capturing the complex structures and second one mainly targeted on continuity of the geometry. A H-glass structure was introduced to increase the representation ability. The next problem was the loss function to compare the point set. They introduced two metrics for points sets. (1) Chamfer distance (2) Earth Mover distance. Next they have to deal with GT. They try to solve this by using an auto encoder.

This conditional generative network has a forecasting stage and encoder stage. This encoder matches the two images with a vector. It predicts a matrix, where a single line can be map to a point.

Generative network has two branches. One is FC branch and other is de-convolution branch. They have also done experiments with H-glass. They introduced H-glass to get the best performance. This generative network provides better results; it can represent 3D reconstruction well.

Designing a good loss function is a critical challenge as they have to compare the predicted cloud and the ground truth. So they proposed two methods. (1) Chamfer Distance, (2) EM Distance. The network trained with CD, generates better results

than the network trained with EMD. The first network results has a better shape. In the second network the shape of the object does not reconstruct well.

For the training data, researchers have used the ShapeNet dataset. This data set includes 3D CAD models. So what they have done is, they rendered 2D images from CAD models. After that they use these images for training and testing. They rendered these 2D images according to the Blinn-Phong model. They have selected the environment map randomly. To reduce the calculation time authors have used simple model for lighting. Researchers have used the upper hemisphere to render the views.

The proposed network architecture works with RGB images. FC branch generate two hundred and fifty-six points and de-convolutional branch generate seven hundred and sixty-eight points. The proposed network has convolutional layers. Each convolutional layer has sixteen maps. Here they have not used max pooling, but used stride convolution. Here the adam optimizer is used as the optimizer. The programme is developed using tensorflow. They used RELU as activation function. Then mentioned that they have not used any normalization techniques. Finally, researchers have compared the performance of the network with 3D-R2N2.

Authors faced difficulties in generating 3D point cloud, namely how to represent the data and how to handle the data in machine learning. They have done lots of experiment to obtain the satisfactory results. The main implementations of this method are point set generation network and loss functions for point set comparison. In point set prediction network there is an encoding stage and predictor stage. Since there were not much public data available for training and testing they had to prepare their own datasets. The randomness in their network enables prediction of different shapes given the same input image.

Researchers have used several matrix to compare the results with state of methods. They have use chamfer distance, earth movers's distance and etc. Also they have compared the result visually. This method provide good results. Since the input to this is just a single image this solution has become famous among the developers and researchers. This method does not generate a mesh. That is one of the downside of this proposed method.

They have applied this method to both synthetic and real world data sets. Also they have identified that more complex network gives better results. They compare the results of the real world data visually and use the results from synthetic data to compare and analysis the result mathematically.

Table 2.2 shows the performance of the system. They have compared the results of each category with 3D-R2N2. These categories are from the Shapenet dataset. Author's main target was to develop a network that can predict 3D point cloud using a single RGB image. They have achieved this target and it outperform the state of art method at that time. Table 2.2 shows the IoU with point set generation network vs 3D-R2N2.

category	Ours	3D-R2N2		
	1 view	1 view	3 views	5 views
plane	**0.601**	0.513	0.549	0.561
bench	**0.550**	0.421	0.502	0.527
cabinet	0.771	0.716	0.763	**0.772**
car	0.831	0.798	0.829	**0.836**
chair	0.544	0.466	0.533	**0.550**
monitor	0.552	0.468	0.545	**0.565**
lamp	**0.462**	0.381	0.415	0.421
speaker	**0.737**	0.662	0.708	0.717
firearm	**0.604**	0.544	0.593	0.600
couch	**0.708**	0.628	0.690	0.706
table	**0.606**	0.513	0.564	0.580
cellphone	0.749	0.661	0.732	**0.754**
watercraft	**0.611**	0.513	0.596	0.610
mean	**0.640**	0.560	0.617	0.631

Table 2.2: Result Comparison - Point Set Generation Network vs 3D-R2N2 [15]

According to the above table we can conclude that the proposed method provides better results than 3D-R2N2 in single view reconstruction. Many categories provide better results than 3D-R2N2 (using 5 number of views)

The team identified the failure situations of this method on the verification dataset. In the first set of failures, the neural network provides a state that has no sense at all. Then networks try to explain inputs with something similar but incorrect. Sometimes the predicted output gives bad results. The neutral network tries to predict multiple objects than one. Here the output is not converged well.

2.2.5 Pix3D

Xingyuan Sun and his team have proposed the Pix3D [17], which includes datasets and methods for generating 3D objects from a single image. This has many applications related to shape fitting. The dataset provided here is large and can be used for other researchers to do their work. Pix3D provided a model that does the 3D reconstruction. It also provides a method to identify the post estimation as well.

When considering their data set, it is a rich data set and anyone can use it for research purposes. This data set include RGB images, camera parameters, 3D models, masks and etc. First, they research through the internet to find suitable 3D model repositories. They have collected the RGB images as well. For the creation of

the data set, they have used the Ikea data set. By acquiring the 3D model, they rendered 3D models in different environments with different lighting conditions to generate a rich dataset. This data set has both real images as well as rendered images.

Building this kind of dataset is beneficial for both the team as well as the other researchers. They obtained the mask for each of the rendered images they obtained. Authors need to generate camera parameters for each of viewing angle as well. They used their own method to solve the problem. They used 3D model and 2D/3D key points to solve this problem. The researchers have used various optimization techniques to get the best results. After doing all the above-mentioned steps they were able to build a rich data set for 3D reconstruction.

In the dataset, there are nearly four hundred shapes that exist. It also has more than nine types of objects. This data set has more than ten thousand shape image pairs and their masks. So we can consider this a rich dataset. In this project, the authors have to use IoU, CD, and EMD as matrices.

To compare the above metrics, the team has run three shape reconstruction algorithms on randomly selected images, and using human rating they have discovered that the EMD and CD perform better.

The team has proposed a system that can reconstruct the object as well as the pose estimation. They trained and evaluated using their own dataset. There are four main modules in Pix3D.

- 2.5D Sketch Estimator
- 2.5D Sketch Encoder
- 3D Shape Decoder
- View Estimator

2.5D Sketch Estimator is an encoder-decoder network. It takes RGB images and outputs normal, silhouette, and depth maps. The output sends to the 2.5 sketch encoder. It provides a latent vector as an output. Then the results are sent to the 3D shape decoder and view estimator. 3D shape decoder output a vowelized shape. View estimator provides the pose estimation of the object. Figure 2.5 shows the diagram of the main modules.

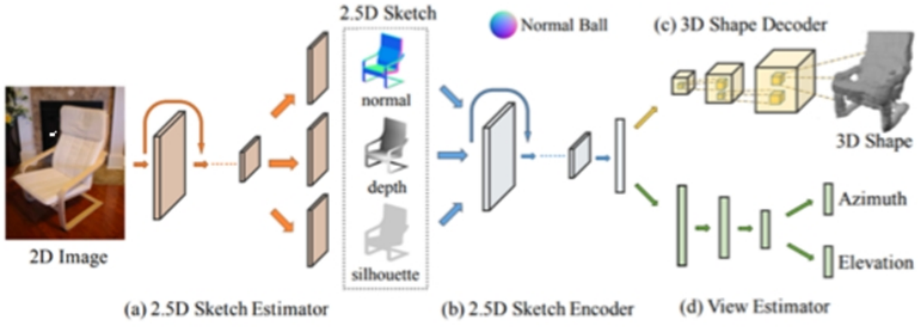


Figure 2.5: Pix3D high-level architecture [17]

For the training team used ShapeNet data set as well as SUN database. They rendered 3D models to generate RGB images. Authors also used image augmentation to increase the data set. They simulated different environmental conditions as well as different lighting conditions when generating the data set. When training first they trained the individual components one by one and finally they trained all together to get the best results.

The researchers have done several experiments to compare the results. For the result comparison, they have used several matrices. Here the output is a voxel mesh. They used CD, EMD, and IoU. IoU is used in many 3D reconstruction projects to compare and analyze the result. Since the authors had a voxel mesh they had to convert it to a point cloud before calculating the CD and EMD. First team compared the results with pose estimation and without pose estimation.

Pix3D can predict the 3D shape from a single RGB image. It also has the ability to estimate the pose as well. Pix3D also has their own dataset. This method does not predict texture information. This project also has a rich dataset that can be used for 3D reconstruction experiments.

The team compared the pose estimation results as well as the shape prediction results with other methods with that time. The proposed method provides comparable results to other methods.

2.2.6 Differentiable Volumetric Rendering

In this work, Michael Niemeyer and his team introduced differential volumetric rendering [20] which has the ability to represent both texture and geometry. They learned this using a neural network. Authors represent the texture and geometry in function space. The input to the system was RGB images, masks, and camera parameters. Also, they have introduced reconstruction from a single image and as well as reconstruction from multiple images which provide accurate results in many cases.

When it comes to the representation of the shape and texture the team represents the 3D geometry explicitly using occupancy network. The occupancy network assigns an occupancy probability to all the points in the 3D space. There is an encoder network when predicting geometry with a single image. They define 3D object texture using a DVR, another 3D surveillance system that allows them to learn both occupancy probability and texture field in 2D images and depth information. When using the DVR they use an automatic differentiation as the base of their method. During the training, the team assumed they were given n-images and compatible camera intrinsic/extrinsic parameters, and object masks. As these tests show, this method works with just one image of each object.

In addition, this method can use depth information if available. They used different types of methods to find losses. For RGB loss, in each point they find the predicted depth and define photo-consistency loss for the points. They use depth loss for the predicted surface depth using given depth values. If no depth is predicted, freespace

losses are applied to the random sample point on the ray.

The mentioned Occupancy Loss helps the network to obtain the 3D space along the ray which can be used by Depth Loss and RGB Loss to fine tune the initial occupancy. By regularizing surface normal authors have incorporated a smoothness prior. This is especially useful when predicting for the real-world data set.

The authors of the DVR have designed the neural network with five FC ResNet blocks. Here Each ResNet block has five hundred and twelve hidden dimensions. They used relu as the activation function. In this model, there are four outputs. Three outputs refer to the color values and one output refers to the probability of occupancy. Color values present the texture prediction results. The team has used an encoder network to support the reconstruction using a single image.

The proposed architecture support reconstruction using a single image and reconstruction using multiple images. Here, the authors have fine-tuned different parameters to get the best results. The training was carried out with sixty-four images with thousand and twenty-four pixels each. Increasing the sampled pixel will provide good results, but it will take more time to finish a single iteration. Fine-tuning these parameters are based on the network architecture. Initially, the team started with a learning rate and it decreased with the epoch count.

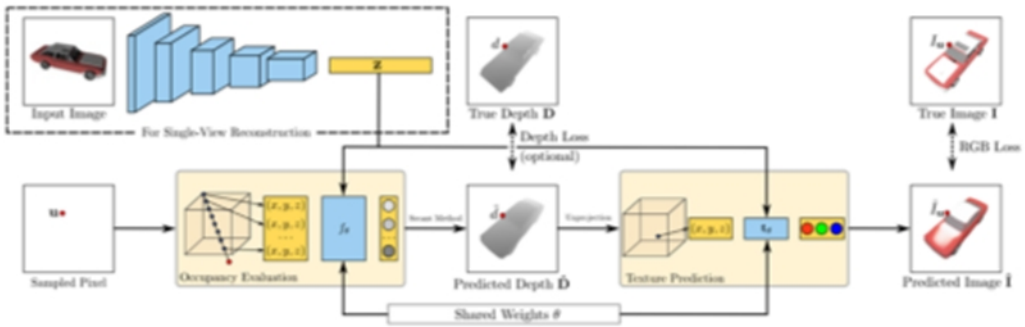


Figure 2.6: DVR network architecture [20]

Here they did two different types of tests to confirm this method. First, they explored how this technique can generate texture and shape using an RGB image while training in a large set of datasets. Next, they evaluate for reconstruction using multiple images. For this authors have to train a different model for different objects. After that they predict the geometry using empty input. This method provides accurate results most of the time. They mainly used this method for real-world objects. This is a very challenging task. For single view reconstruction, researchers use Shape Net data set for 2.5D and 3D supervised methods. For the multi view reconstruction team has used the DTU dataset.

The researchers have compared their method with other approaches which generate meshes: Pixel2Mesh (mesh-based), 3DR2N2 (voxel-based) and ONet (obscure representation). They also compare the results with Soft Rasterizer (mesh based) and

DRC (voxel-based) with respect to single and multi-view reconstructions.

They evaluate both Multi-view supervision and single-view supervision using chamfer-L1 distance. In multi-view supervision, this method works best when training with 2D GT and as well as 3D GT. It will provide high quality result when training with 3D GT. When using depth information, the result is getting better. This method used function based approach to describe the mesh, therefore this will provide accurate results than the mesh based methods. The authors have found a method to represent the texture as function as well. From it they can accurately represent the texture information along with the geometry. This approach can predict texture as well as geometry given a single image as input. This requires lots of training and fine tuning.

The authors ran their method for the DTU dataset. It is a real world data set captured using a robotic arm. They have compared the results obtained with the results from MVS, SFM methods. The results were better than the classical photogrammetry techniques. So this method can be used to generate meshes from multiple images. The main benefit of this method is it will provide a clean mesh, if a 3D artist wants to fix issues after a scan he will be able to do it easily from these results.

2.2.7 NeRF

NeRF generates new views from a given scene. The input to the NeRF[22] is images of the scene and it renders the whole scene by interpolating between scenes. Here they used a MLP (multilayer perceptron) network. The propose network used ReLU as the activation function. It has eight FC layers. The output layer consists of hundred and twenty-eight features. Final result of the NeRF is accurate. But the main issue with this method is that this does not provide a clean mesh. The main task of the NeRF is generation of novel views. Later this method has extended to generate the mesh with textures.

The neural radiance field presents the scene as a radiance field at all the locations in the world. It additionally takes into account the extent density. The color of the point is also estimated using the radiance. For this they have to use the techniques in different rendering methods. This method supports both complex and simple scenes.

To improve the performance of the network they encode the coordinates which are used by the MLP. These enhancements are related to the frequency functions. The next improvement to the system is the change of sampling method. They used hierarchical method which helps them to better sample the information. Researchers have optimized two branches of the network. They are known as fine and coarse. They carried out the optimization simultaneously for the two branches. First they used fine network to get the positions and later they used coarse network to test the positions. By considering the output from coarse network they try to generate more samples near the beam. Overall NeRf produce goodresults when compared with the other methods. It has used several optimization techniques to get the best results.

When developing the model they fine-tune a separate continuous neural network to represent the volume. For this input parameters to the system and intermediate results are needed. Intrinsic and extrinsic camera parameters, photos are used here. At each iteration they randomly selected pixels from the inputs and cast camera rays. When generating novel views team use volume rendering to get the color. They used ADAM as the optimizer. Authors have decide to use the multi rate leaning rate. That is start the leaning rate with an initial value and then reduce the leaning rate with each iteration. To produce a single scene, it took more than hundred thousand iterations in a NVIDIA V100 GPU with a single GPU. Initially this method was developed to render new views, but later it extended to generate the mesh with textures. The result of the NeRF is comparable to the previous method specified here.

NeRF provide good results in novel view generation. The main goal of the NeRF is the novel view generation. After providing the multiple images to the system, it can generate novel views by interpolation between the views. Later this has been extent to generate the mesh. They also generate datasets for the training.

Most of the time they used DeepVoxels data set for the training and validation. They generate data set for eight different meshes with complex geometry. They use realistic non-lambertian as material

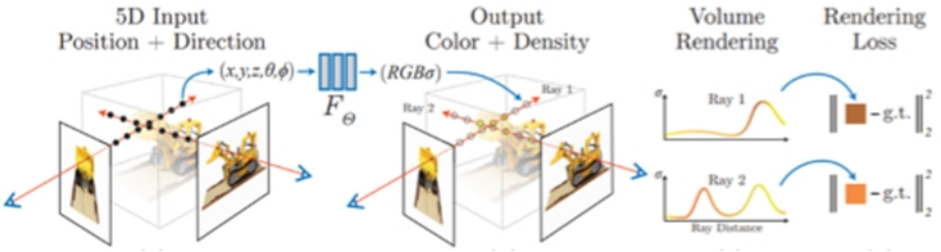


Figure 2.7: NeRF scene representation [22]

They compare their results with techniques such as SRN, NV, LLFF. These are top-performing techniques for view synthesis. The NV method produced detailed appearance and geometry, The SRN method provided smoothed texture and geometry. This method provides comparable results to above top performing techniques.

Although this method out performed all the other methods, it has some limitations and future developments to be done. Despite the fact that they have used an advanced sampling method to make rendering more accurate, much research and development is still needed. This method provides comparable results when compared with other methods.

NeRf provides a manner of representing complex geometric scenes. This is known as 5D representation. The main building block of the network is MLP. This method provides best results for most of the scenes. To get the best result the input images should be captured with an empty background. Since this method does not require mask of the object that we are interested in, we can provide the captured input. The

paper suggests to remove any blur images in the input. The advantages using this method is that generating the input to the system can be done without much effort. The main drawback of the system is that; the results obtain from this method are not clean. It will take considerable effort for a 3D artist to clean the object before doing any modification.

2.2.8 Soft Rasterizer

The team has proposed a method to generate color meshes using single or multi images. The inputs to the system are RGB images, silhouette etc. SoftRas[23] uses shape deformation technique to generate the mesh .Along with the technique to generate the mesh , this project provide a renderer as well. This renderer can be used for other projects related to 3D. The paper indicates that this method can generate meshes with acceptable accuracy. The main advantage of this method is, generated meshes are clean. This project has proposed to generate meshes using single image as well as using multiple images. For the training they have rendered 3D models using their own render. This method provides good results when compared with other related technologies and related works. The project authors have rendered twenty-four images with masks per each scan, which they used for the supervision. They train scans in the canonical pose.

The researchers compare their approach with the methods that are mentioned below: Pixel2Mesh (mesh-based), 3DR2N2 (voxel-based) and ONet (obscure representation). They also compare the results with Soft Rasterizer (mesh based) and DRC (voxel-based) with respect to single and multi-view reconstructions.

They evaluate both multi-view supervision and single-view supervision using chamfer-L1 distance. Multi-view supervision works best between 2D supervision and competes with high quality 3D supervision methods. This project used NMR dataset for training and testing. This is a differentiable render framework. This project provides a method to generate mesh from a single RGB image. Also, the authors have implemented mesh deformation techniques to generate the mesh. Here the input to the system is multiple silhouette images and output is a mesh.

Softras team has compared their results with other multi-view methods. There are two methods used in softras to generate meshes. The first method used a trained model to predict the mesh. It uses a single RGB image to predict. In the second method, they used mesh deformation to generate the mesh. The input to the mesh deformation system is a set of silhouettes of different angles. Here they also have provided camera parameters as well. Softras has the ability to model the features of the object being scan accurately. They used spherical coordination for this system. This method provide comparable results when compared with other methods.

There are three main modules in the softras.

- Soft Rasterizer
- Color Generator
- Shape Generator

First, the input images are fetched to the shape and color generators. The outputs from those are fetched to the SoftRasterizer. Then using the outputs of these, losses are calculated. There are three kinds of losses in sofas. They are loss due to color, loss due to geometry and loss due to silhouette. Finlay they have accumulated these losses to get the final loss. Softras Deep Leaning approach can predict both the geometry as well as the texture. The authors have used novel methods to generate the color information of the object. Below figure describes the method they used to predict the color.

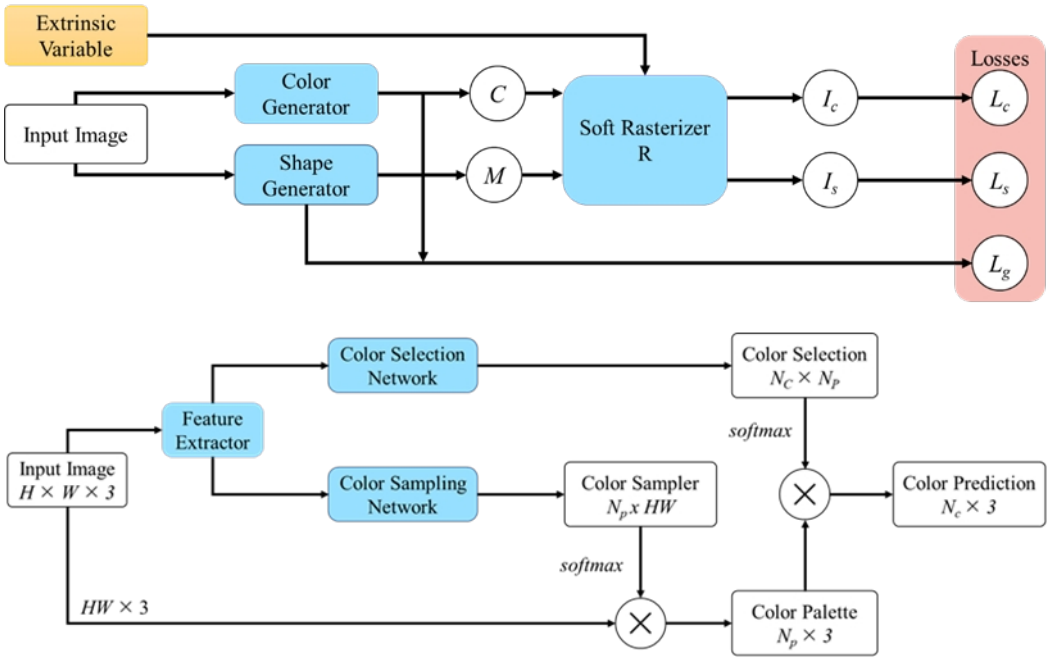


Figure 2.8: Network architecture for soft rasterizer

Softras does not only provide solution for reconstruction but also provide pose optimization and non-rigid shape fitting as well. This project also provides a render. It provides realistic rendering results. Soft Rasterizer is known as a differentiable render. These are used to calculate the loss of rendered images. So the loss can back propagate and can train the network. For the single view mesh experiments, they train and predict the network for each category.

Softras used ShapeNet dataset for training. They created their own data set for training and testing. First, they rendered each object in the ShapeNet data set in twenty for different angles. The researchers have used thirteen object types to generate the dataset. The size of a training image is small (sixty-four pixels wide). Due to this, training is faster but the accuracy is not that much good. The authors

used adam as the optimizer. They trained the network and predict results per category. They have trained a single category for twelve hours. For the implementation, they have used pytorch framework. They have not provided any 3D-related supervision for the training. The authors have done several experiments to compare the results. For the result comparison, they have mainly used IoU. This approach can generate fine details of the object as well.

Softras can generate both geometry of the object as well as the color information of the object. Even though the resolution of the trained images are low, still able to predict color information with acceptable accuracy. They did two main experiments under two different configurations. First they trained the network using only the loss due to silhouette and compare the results. After that they used both silhouette and shading loss for the training. Second configuration provides the best results and surpasses results from other methods in all categories.

2.2.9 Generating 3D Models from Single 2D Image without Rendering

Nikola Zubi'c and the team proposed 3D-based reconstruction using a single image, a method to predict the geometry of the object by learning from 2D supervision [26]. Then to generate the texture a GAN is used. The final output is created using the geometry and the texture. Although modern methods require a rendering step that requires significant computational power, this method has not used any rendering step.

Due to its compactness, cloud points are considered as one of the three most popular 3D presentations. There are several advantages in using point clouds. It will take less time to generate the point cloud and also the operations can be done to the point cloud without much computational power. When doing result comparisons, point cloud is the preferred 3D data format. The proposed method is able to generate an accurate point cloud from the given image. Generating a point cloud is more effective than predicting a mesh itself.

A point cloud has a finite number of 3D points. The point cloud can represent any 3D mesh effectively. After the point cloud is generated, then converted to a mesh. Then predicted texture is applied to get the final result.

The team has developed and trained the network which learns to construct a point cloud using a single image. For that, they have trained the network using images from different viewing angles. That means here authors used 2D supervision. No ground truth model is used. Here they have used silhouettes of different viewing angles for training. The loss incurred in the training tries to fit the predicted model inside the silhouette. The training method involved two main processes.

First they start with a random point cloud. Then with the training, they fined tune it to be the targeted mesh. The team fine-tuned the network so that projections will occur only on the silhouette. After that, they try to cover the full silhouette as much as possible. By following these processes, they were able to generate the point cloud with texture information. The reconstruction results from this method provide

comparable results to other methods.

The researchers introduced loss to predict the geometry. For this they have used two loss functions. First loss helps to do the projections better. The second loss helps to predict the mesh better. The final loss function is a combination of these two. To obtain better results authors have to concentrate more on the second loss than on the first loss. But the loss one is important as well. So final loss function is simply not the addition of loss one and loss two. This approach helps the researchers to converge the model better.

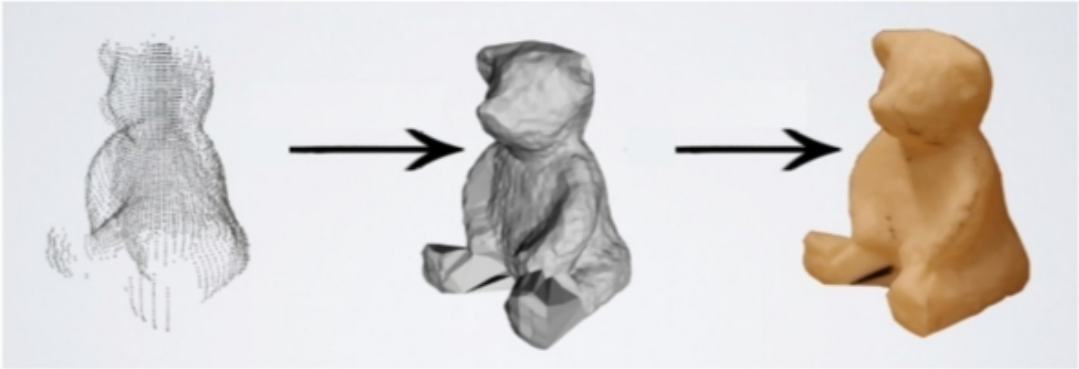


Figure 2.9: Generate Mesh from point clouds [26]

When you select a loss function, there are several factors that you have to pay attention to. Here selecting the best loss function helps to estimate the points in the 3D space. This project has defined two loss functions. The final loss function is a combination of these two. The final result of this project is a point cloud. After that, it is converted to a mesh. The researchers have used GAN to predict the texture. They predict the texture using the provide RGB images. They predict the point cloud using a single image. Therefore, they have to use lots of images for the training. There are several methods to generate the mesh using point cloud. They used 'Poisson surface reconstruction' to convert the point cloud to mesh. They were also able to generate displacement maps.

This project uses a single RGB image to generate the point cloud. So they train the network using many input images. They have used CUB-200-2011 data sets for training and testing. Not only that they have used Pascal 3D and ShapeNet dataset as well. To evaluate the result, researchers have used CD between point clouds. Not only that researchers have also used FID to evaluate the results. They have evaluated both the texture and geometry. The results show that this implementation provides comparable results to all the previous reconstruction methods. When you increase the resolution of the images, the CD is getting improved.

The results of this method provide comparable results to all the previous 3D reconstruction methods. The renders are better than voxel renders. There are strengths and weaknesses of this system. This takes less time in the geometry prediction as well as texture generation.

3. METHODOLOGY

This chapter discusses the high level architecture of the proposed system, main steps that are involving in the project and their results.

3.1 High-Level Architecture

3D reconstruction of objects has become a widely researched topic. There are lots of applications in 3D reconstruction. There are two main methods used by users to scan and reconstruct objects. Those are laser-based scanning and photogrammetry. Each technique has its own advantages and disadvantages. Photogrammetry is a low-cost solution, but it cannot provide accurate results for every object. The result depends on the object type, object texture, environment, lighting and etc. Since the cost is low many people use this technique to generate the results. Colmap and reality capture tools use photogrammetry to generate the mesh. Laser scanning can provide accurate results, but the initial cost is too much. This solution is a good one for a company but not for an individual.

In Photogrammetry solutions, the user has to take images of the object being scan in different angles. The quality of the final results depends on the quality of the images, number of images and etc. But the main issue in this technology is, it can reconstruct objects with high textures only. For the less textured objects, this method does not work and does not give good results. And also the meshes obtained from this method are not clean and a 3D artist has to modify with considerable effort before it is used by anyone. These methods have long been blamed for their limitations. 3D data generation often requires expensive data collection as well. So a new method is needed to address the above issues. This research used a deep learning based approach to predict the geometry. Since this method does not depend too much on texture and lighting conditions this project gives good results. This project has been tested for both synthetic data sets as well as for the real world data sets. To capture the synthetic data blender is used. Blender is a 3D computer graphics software. It is used by people around world to create 3D models, animation movies and etc. To get the real world data, a kinect is used. Here after, an obtained single data set is called as a scan. There can be errors due to noise of the depth information. But when using the pre-processed RGBD data error can be mitigated.

I calculated camera extrinsic parameters based on SLAM. After that I used bundle adjustment to further improve the camera extrinsic parameters. For the prediction of the geometry, this research initialized with DVR multi view reconstruction approach. DVR is a Deep Learning method to predict mesh using RGBD and camera information which is based on occupancy network. The input to the DVR requires RGBD information, camera information and masks of the RGBD images. RGBD information is available in the obtained dataset. I used SLAM to find the camera parameters for the dataset. To generate the mask first, I create the point cloud using RGBD information and camera parameters. Then clean the point cloud manually and remove the outliers from KD Tree based clustering. Here the outliers refer to the small 3D point clusters that are left with the object but not belong to the object. Then

project the 3D point cloud back to 2D image. After that using the alpha shape and FBA matting I generate the masks for the scan.

When training the DVR it requires to cast the rays from camera origin through pixel to the world. Then evaluate the occupancy probabilities along the ray at every sampled point in the world. This operation can perform faster if you wrapped the interested area in a unit cube as per the paper. I introduced a clustering technique which automatically identifies this unit cube for the scan, so the process can run without manual intervention.

Since DVR becomes slow at texture projection, I modified the DVR network so the geometry prediction will be faster. Also I have introduced new loss function, IoU at training. This loss helps to generate the shape of the geometry better. I removed the losses related to texture prediction. Not only that I had to fine tune several parameters to get the best results. The texture projection in this research is based on MVS method.

3.2 Data Collection

In order to carry out this project, I used two methods. One is generating 3D mesh from synthetic data and other is generating 3D mesh from real work dataset. For each of this method I had to collection data from different resources.

3.2.1 Collect data from synthetic data set

Blender is a tool used in 3D content creation. This is used in several other applications as well. Since this is an open source tool, many 3D artists tend to use this tool to generate their contents. There are lots use cases in blender. Blender supports a number of 3D mesh formats. Also there are freely available plugins that can be used for various tasks. It supports for various kinds of materials as well. There are three main types of rendering engines are available. Cycle rendering engine is used in production.

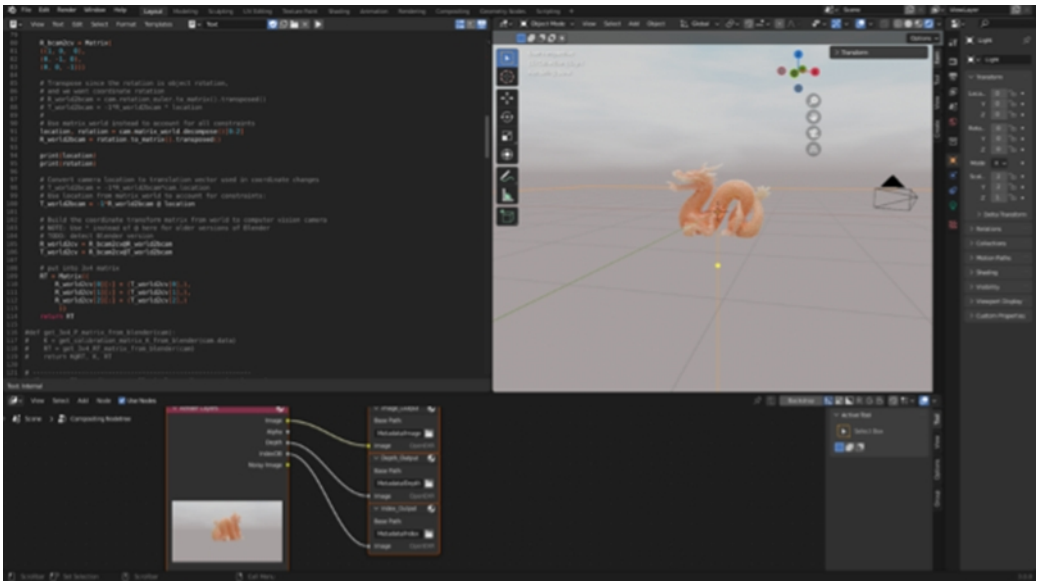


Figure 3.1: Blender generate data set

It is known as physically based rendering. In this project I have also used Cycle rendering to generate the images and also used light sources to light up the environment.

Using the blender nodes, I extracted the rendered output into images and depth maps. The render has generated three outputs in openexr format. Openexr format can store raw values. This format is used in a variety of projects to store the information without any information loss. This mainly stores numerical data which has very high range. Then I converted the openexr format to png for images and kept the openexr format for depth maps. All the outputs are taken from pre-defined camera locations. Those camera locations are fixed for every dataset.

Here I have captured sixty images and depth maps for each 3D object. Below shows a data sample.

Figure 3.2: Object image

Figure 3.3: Object depth map

Since the depth maps from the blender is accurate, can use those to generate the masks for the data set. This is possible because there is only a single object in the entire scene. Below shows the generated mask for this sample.

So using the predefined camera locations; images, depth maps and masks are generated. Since I am using blender, can find the intrinsic camera parameters from the blender itself. Using the predefined camera locations can calculate the extrinsic matrix for each location. Both intrinsic and extrinsic camera parameters are necessary for the training.

Figure 3.4: Object mask

3.2.2 Collect data for real world data set

For real word data set there are several options.

- TUM Data set
- DTU data set
- Capture your own data set with kinect or any other camera

For this project I used a TUM dataset and data captured using a kinect. In TUM data set a kinect is used to capture the information of the scanned object. By using the kinect it captures RGB and depth information as a video. Using that information and timestamp I was able to generate rgb images and depth images. I was also able to captured image and depth information of object using a kinect. First the kinect has to calibrate, then has to attached a kinect to a tripod. Then move the kinect around the object to capture depth and rgb information. Then using the RGBD information I calculated the extrinsic matrices from SLAM.

3.2.2.1 Calculate extrinsic matrices

In this project I used SLAM to calculate the extrinsic matrices. SLAM generate 3D map as well as the extrinsic matrices. One drawback of generating a extrinsic using SLAM is that it depend the features of the environment. There are different implementations of SLAM. To generate the extrinsic, images and well as the depth maps are used.

In this project ORB-SLAM2 is used. Using the depth images and RGB images this can generate 3D map as well as the camera parameters. The input to the system is intrinsic matrix, rgb images and depth images. Here ORB is used to detect the features. The results are based on the features of the environment and the accuracy of the depth map. This provide good results when there are rich features.

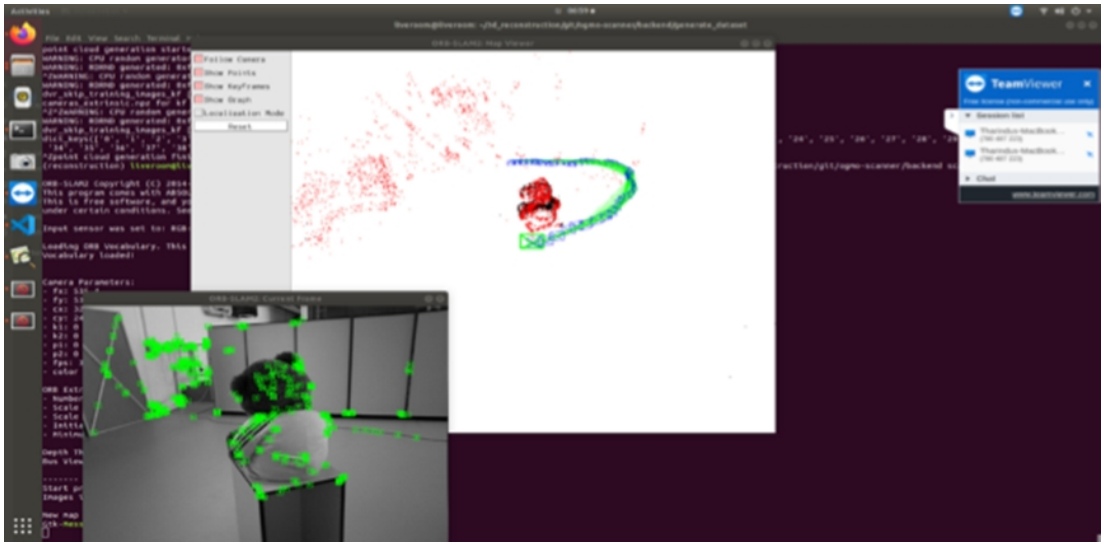


Figure 3.5: ORM-SLAM2 - Keyframes

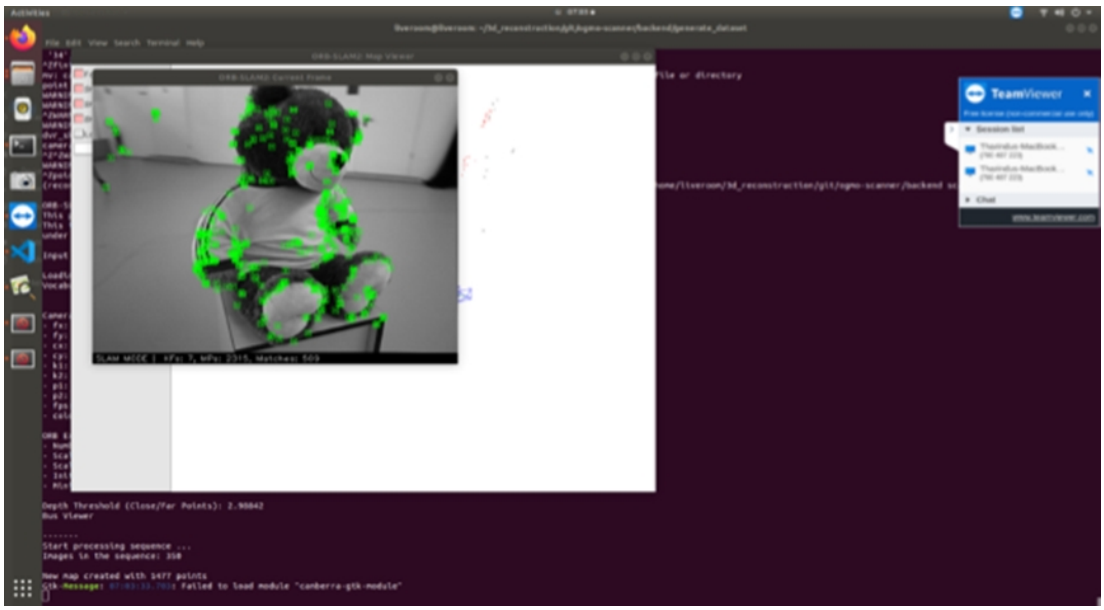


Figure 3.6: ORM-SLAM2 - Feature Points

After running the ORB-SLAM2 I have a set of key frames with their extrincs. For those key frames removed all the blurred images, so it will help later on training. After that depth maps are filtered according to the distance.

Figure 3.7: Panda point cloud - front view

Figure 3.8: Panda point cloud - side view

Finally, I have a set for images/depth maps with intrinsic and extrinsic parameters. In the synthetic data set, obtained extrinsic matrices are very accurate. But in this case we cannot guarantee that the generated extrinsic parameters are accurate. To check the accuracy of the data set created a point cloud of the object with the help of obtained extrinsic parameters by using filtered depth maps, images, intrinsic and extrinsic parameters.

Above images demonstrate the point cloud from different views. With this result it can be concluded that extrinsic parameters have been generated with the acceptable accuracy. So these extrinsic parameters can be used for the training. To get a good extrinsic parameter, the object or the environment should have good feature set. If

the surrounding environment does not have good features, I placed a marker so there can be features around the object.

3.2.2.2 Generate Masks

In synthetic data generation, depth maps are used to generate mask. But in real world data set depth map is not accurate and it cannot be used to generate the mask (the result is not accurate). Therefore, a new approach need to be introduced. Since I have the point cloud, first I cleaned the point cloud manually. Then used KD tree based clustering technique to identify any other outlier exists and remove them. After that I projected the point cloud in to the camera plane based on the projection matrix. Then alpha shape is created by wrapping the projected points. The alpha shape is created based on the following object category.

- Simple Objects - Solid objects with no holes - considered as a single object in the process
- Complex Objects - Hollow objects with multiple holes - concatenate multiple parts to create the final object in the process

Then shrunk the alpha shape to generate the trimap.

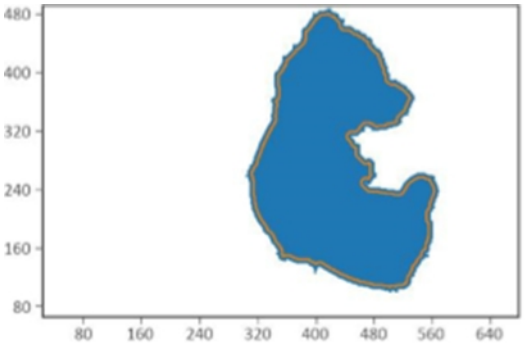


Figure 3.9: Alpha shape

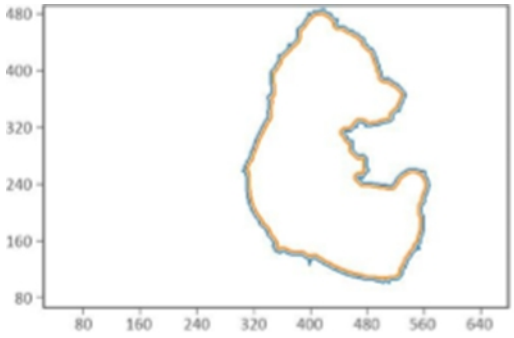


Figure 3.10: Shape with shrinking

Using the shape after shrinking I created the trimap. Then used the FBA Matting to generate the mask. To apply FBA Matting, first need to generate a trimap FBA Matting is developed using the alpha matting network. This project differentiates background and foreground from a RGB image. FBA matting provides new loss functions which helps to predict the result better. Currently this provides accurate result. The input to the system is an image and the trimap. This method provides good results when compared with other methods. Here I have not trained the FBA matting, but have used the provided pre-trained model that has been trained with Adobe Image Matting Dataset.

Using the pre-trained model, trimap and image I predicted the mask. Then repeated these steps for all the images and able to get a masks for the dataset with acceptable accuracy.

Figure 3.11: Object trimap

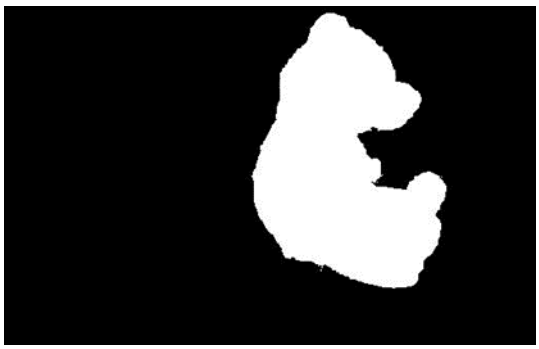


Figure 3.12: Object mask

3.3 Euclidean Clustering for point cloud

In this project point cloud clustering is used in two stages.

- Remove outliers from point cloud before projecting the point cloud onto the camera plane
- Wrap the point cloud to the unit cube this will speed up the training.

Voxel filtering is down sampling, reducing the number of points in a point cloud using a voxelized grid approach. Voxel Grid is a 3D voxel grid over the input point cloud data. Then, in each voxel, all the points present will be approximated with their centroid. This method is generally used when down sampling any point cloud.

$$Distance_{(searc\ hpoint\ ,\ \ currentpoint\ )}\sqrt{(x - searchpoint(x))^2 + (y - searchpoint(y))^2 + (z - searchpoint(z))^2}$$

Equation 3.1: Distance between search point and current node point in ED

Before doing the clustering, this data has to be stored in a data structure. Since there are lots of data and processing need to be done, all the points are stored in K-D Tree. K-D tree is binary search tree. Due to the above mentioned feature we can process the data fast and organized the data well. K-D tree is used in many applications due to its structure and performance. This is normally used with high dimensional data. Using a K-D tree I was able to structure the data to a format that I can do the processing.

K-D Tree was used because of its speed and performance characteristics for handling large amounts of data and inherent data partitioning nature which is very useful while calculating the nearest neighbour points. It is a structure of nodes which are interlinked sequentially together based on their properties. The arrangements of nodes is similar to that of a Tree structure in K dimensions. Here find the nodes which are near to each other based on their Euclidean distance (straight-line distance between two points).

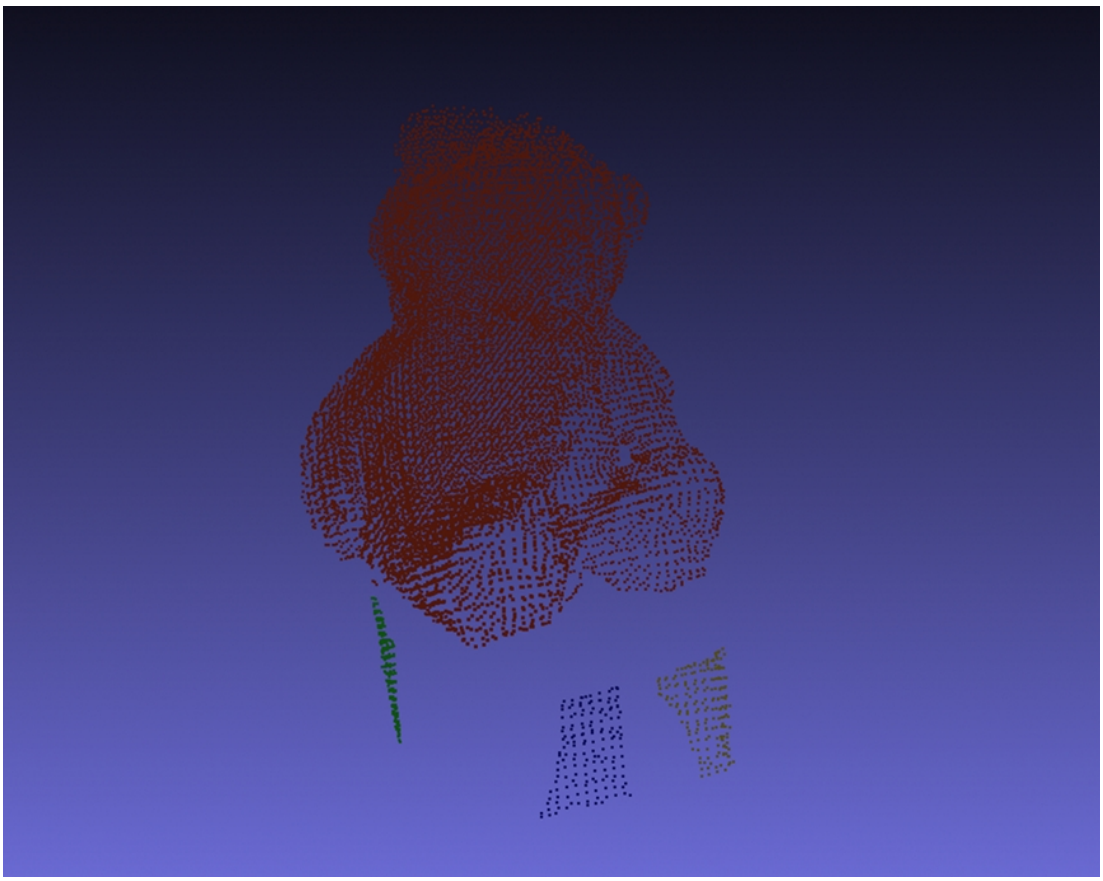


Figure 3.13: Clustering of the point cloud

If the Euclidean distance is within the distance threshold limit I add this point as a near point in K-D tree search results. Then I recursively iterate through the left and the right node of the current node to find all the related points. If the point is not processed before, it's assigned as a new cluster and all the nearby points to the cluster is found by searching. All the points returned are then recursively iterated again, to check if there are part of another cluster and to verify if the nearby points have not been processed before. Using this method, I process all the points.

In this project I used clustering technique to remove the outliers that the point cloud has before further processing. Above image shows the result of clustering. It has four clusters. Since 3D scan happens targeting the object, I kept the largest cluster. Then we remove other outliers. Finally, I have a point cloud without outliers. After that using the point cloud I project to a 2D plane and generate the masks. And also I wrapped the point cloud to the unit cube, which will reduce the training time. Since the K-D Tree has the ability to handle large amounts of data, I have obtained a good result in removing outliers from the point cloud.

3.4 Reconstruction

Deep neural networks have revolutionized computer vision over the last decade. There have been lots of projects related to identification assets/objects, optical flow prediction, or segmentation of assets/objects. All those are 2D based vision tasks. However, our world is not two dimensional, but three dimensional. In modern day,

there have been novel projects which are based on 3D reconstruction. Also they have obtained accurate results. Some projects are able to obtain good results from a single image. Most projects do the reconstruction using synthetic data. But there have been few projects that involve real world data. This project covers generating 3D models using both synthetic data as well and real world data.

This project uses differentiable volumetric rendering as the neural network model. In DVR , they have mainly used two functions, an occupancy network fθ : R 3 → [0 , 1]. Here what they have done is, assign a probability to all the points in space. There is a binary classifier which used as a decision boundary to identify the geometry of the scan. Same way object of the texture can be predicted using the texture field. In DVR they have assigned RGB value to all points in the space. DVR includes both texture field and occupancy network in a one network.

This project used differential volumetric rendering for mesh generation only. For every point in the 3D space DVR calculate occupancy probability and texture (RGB). The main bottleneck in DVR is texture generation. Meaning that it will slow at texture projection.

So this project use only use geometry generation from DVR. To generate the texture, I use MVS Texturing. To predict only the geometry, I modified the network.

DVR predict the occupancy probability and texture field as a single network. It takes a batch of Np 3D points as input and outputs both one-dimensional occupancy probabilities and three dimensional texture vectors in RGB space. DVR first pass the point coordinates (p1, p2, p3) via a FC layer. Then pass its output to five consecutive ResNet blocks. The original implementation has a 512 of hidden dimensions. Then final results of this system is a batch of Np one-dimensional occupancy values and three-dimensional RGB color values. To predict only the occupancy probabilities effectively I modified the network as Figure 3.14.

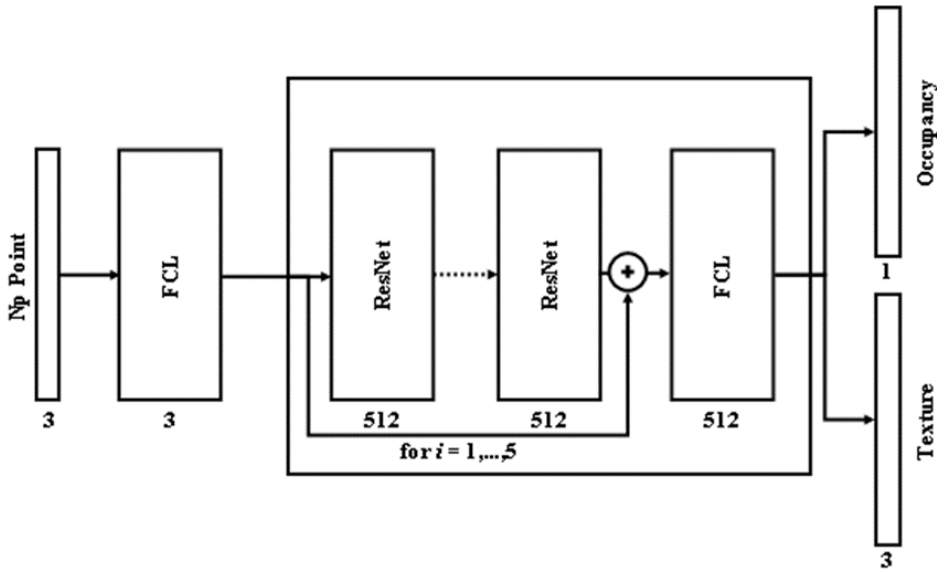


Figure 3.14: DVR network architecture

In the modified network there is only one output that is only the occupancy probability. Here I first pass the point coordinates (p1, p2, p3) through a fully connected layer (hidden dimension X) with ReLU activation. Then pass the output to Y consecutive ResNet blocks with ReLU activation and a hidden dimension of X. The final output is a batch of Np one-dimensional occupancy values.

I tested the modified network with three different configurations by changing the X and Y. The different configurations are as follows.

- X=512 and Y = 3
- X=256 and Y=5
- X=256 and Y=5

I tested with these configurations, to find which gives the best result and least time to converge. All the configurations show acceptable results. But X=256 and Y=5 gives the best results.

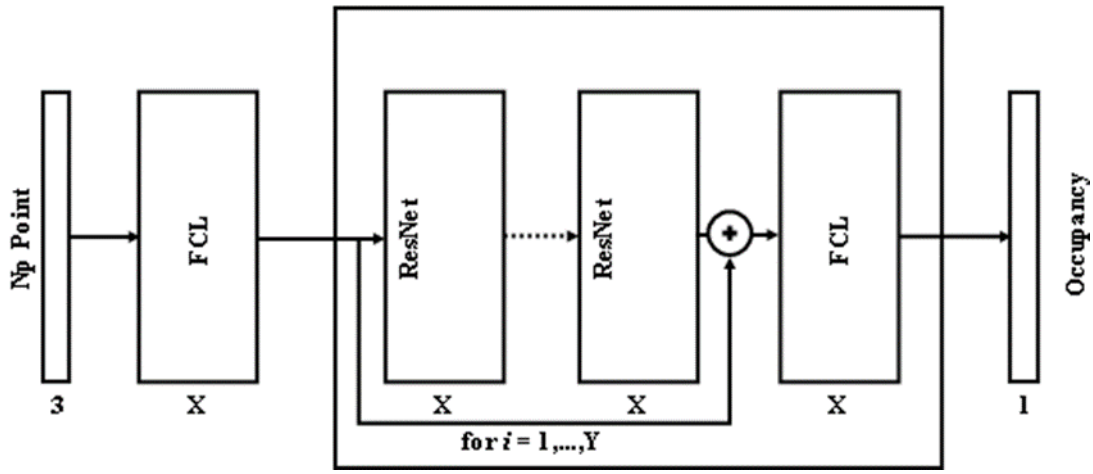


Figure 3.15: Modified network architecture

There are several losses defined and used in differential volumetric rendering for training.

- RGB Loss
- Depth Loss
- Freespace Loss
- Occupancy Loss
- Normal Loss

This project does not predict the texture information, so removed the RGB loss in training. RGB loss mainly contributes for the prediction of the texture. Also I introduced a mask intersection loss at training. IoU can be defined as ratio of area of overlap to area of union. It measures the accuracy of the detection. This is a widely used evaluation method. This can be defined as below.

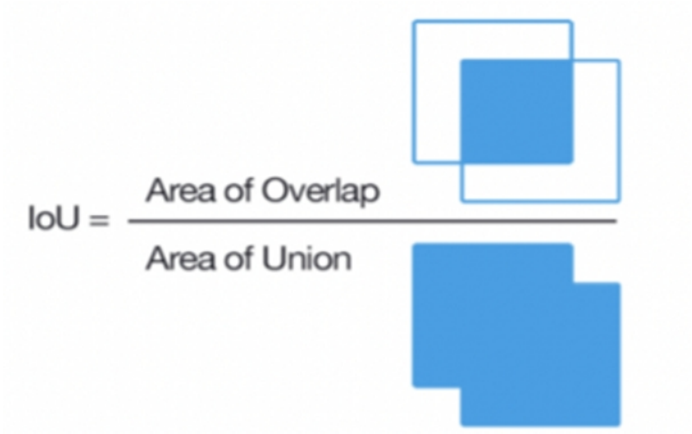


Figure 3.16: Intersection over Union

IoU is used by many researchers for evaluation purposes. This is a powerful way of evaluating detection results in Deep Learning projects. Mask intersection loss helps to generate the geometry shape better.

In order to obtain best results with the new losses and modified network I had to fine tune several parameters so that the model is generalized. Below show the parameters that have to fine-tune.

Multi step LR

The training starts with an initial learning rate. In Multi step LR, The learning rate reduces with the epoch count. The amount it reduces is determined by gamma value.

Here I have to fine tune the epoch counts as well as the gamma. I set the gamma as 0.25 and epoch count as 1000, 1500.

Sampling rate

Here I increase the sampling rate with the epoch. I set the initial value of sampling accuracy to 32 and increase it with the epoch count.

Depth loss factor

I made the depth loss contribute more to the final loss. Since it is possible to capture the depth information with acceptable accuracy, making depth loss contribute to the final loss helps to converge the model quickly.

Model validation steps

This parameter mainly involves in validation. To obtain the best result I set this parameter to 3000. Optimizing this parameter affect the final result.

Number of training points in an image

Since I made the model simpler (this is due to the prediction of geometry), I can increase the number of training points. This will help the training result better. I increased the number of training points per image to 4096 to get the best result.

Number of evaluation points in an image

Since I made the model simpler (this is due to the prediction of geometry), I can increase the number of evaluation points. Optimizing this parameter affect the final result. I increased the number of evaluation points per image to 20000 to get the best result.

When training the DVR it requires to cast the rays from camera origin through pixel to the world. Then evaluate the occupancy probabilities along the ray at every point in the world. This operation can perform faster if you wrap the interested area in a unit cube. For this I have introduced the K-D Tree based clustering technique. After clustering, keep the largest cluster and remove outliers. After that I resized the point cloud so that it fit to a unit cube (point cloud centroid will be at the {0,0,0}). This will help us on training. All the extrinsic camera matrixes will scale based on that. Using this clustering technique can identify the object and then automatically identifies unit cube for the scan. Therefore, I can run the process without manual intervention.

3.5 Texture Generation

For the texture generation I used MVS texturing method. This is a similar method that is used in SFM/MVS systems to generate the texture information. To generate the textures multiple images in different views had been used. This method provides good results if the provided images are not blurry and provided camera parameters are accurate. It is really important to capture images with best lighting conditions to get the best results.

Here we used generated intrinsic parameters, 3D model, RGB images and extrinsic parameters to generate the texture map. If the camera parameters are accurate, MVS texturing will give good results. Since the synthetic data are very accurate, texture projection gives accurate results. For the real word data set, this project provides acceptable results. In MVS Texturing the result depends mostly on the accuracy of the camera parameters. Overall this gives good results.

4. RESULTS AND ANALYSIS

4.1 Results

After adding model changes, adding/removing losses and fine tuning the related parameters, I obtained the following results at training. Below shows the training loss. It is the combination of all the losses described previously. Note that I made depth loss contributes more to the training loss.

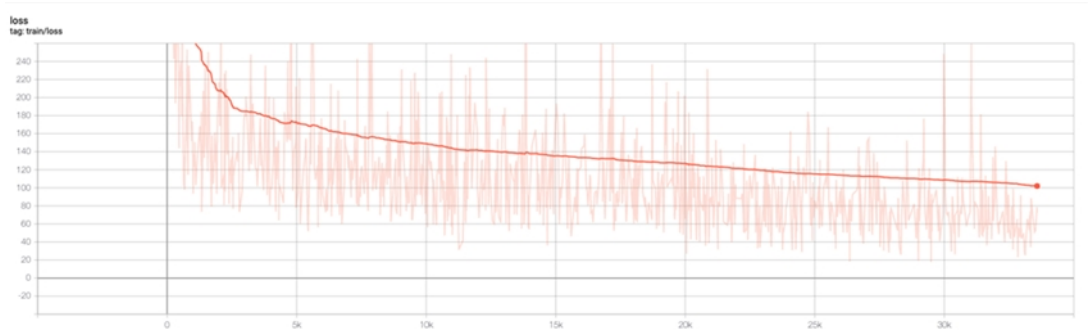


Figure 4.1: Training loss

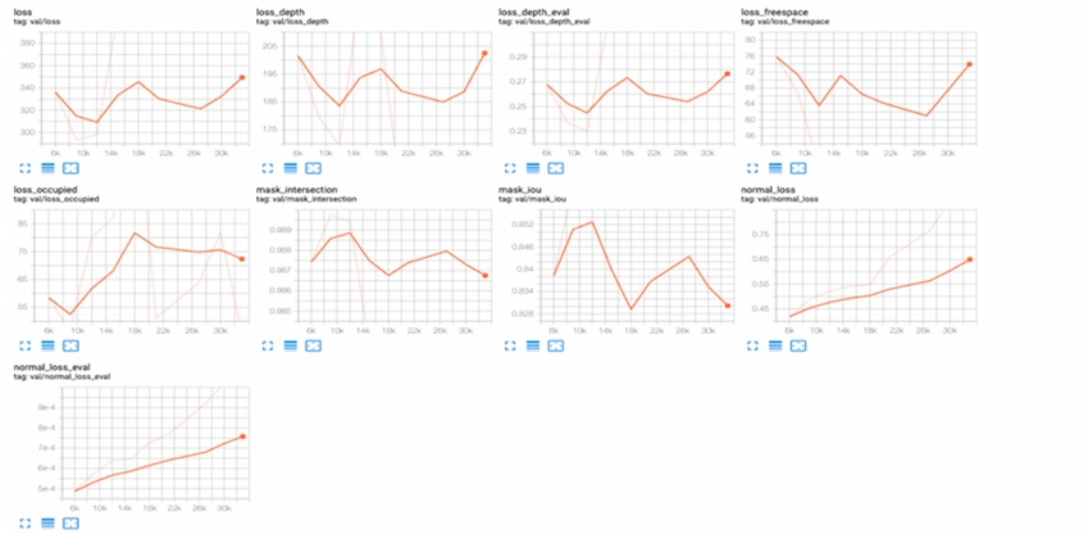


Figure 4.2: Validation loss

Above diagram shows the validation loss. Here the total validation loss is a combination of mask intersection loss and depth loss. This combination helps to generate the shape of the 3D model better.

4.2 Result Analysis

To evaluate the results, synthetic data sets can be used. By using the synthetic data set, can compare with respect to the ground truth shape models. Following matrices are used to evaluate the model.

Chamfer Distance

This metric can be used to evaluate two point clouds. It takes the distance of each points into account. For each point in each cloud, CD finds the nearest point in the other point set and sums the square of distance up.

The chamfer distance between point cloud S1 and S2 is defined as;

$$CD(S_1, S_1) = \frac{1}{|S_1|} \sum_{x \in S_1} \min_{y \in S_2} \|x - y\|_2^2 + \frac{1}{|S_2|} \sum_{y \in S_1} \min_{x \in S_1} \|x - y\|_2^2$$

Equation 4.1: Chamfer Distance

Hausdorff Distance

This metric measures how far two subsets of a metric space are from each other. If we have a two-point set called A = {a1, a2, . . ., an} and B = {b1, b2, . . ., bm}, the Hausdorff distance from A to B is defined as;

$$\widetilde{\delta_H}(A, B) = \max_{a \epsilon A} \min_{b \epsilon B} \|a - b\|$$

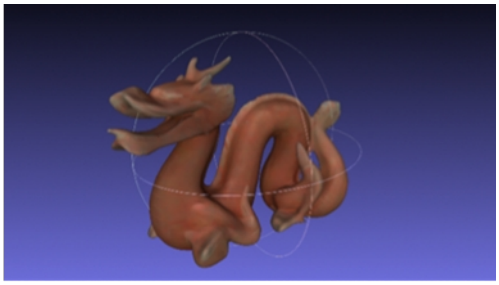
Equation 4.2: Hausdorff Distance

4.2.1 Synthetic Dataset

Scan 01

Original mesh

Predicted mesh



Chamfer Distance: 0.0028
Hausdorff Distance: 0.0119

Scan 02

Original mesh

Predicted mesh

Chamfer Distance: 0.0105
Hausdorff Distance: 0.0085

Scan 03

Original mesh

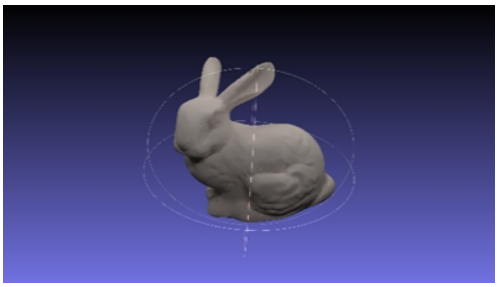
Predicted mesh

Chamfer Distance: 0.0011
Hausdorff Distance: 0.0034

Scan 04

Original mesh

Predicted mesh



Chamfer Distance: 0.0017
Hausdorff Distance: 0.0068

Scan 05

Original mesh

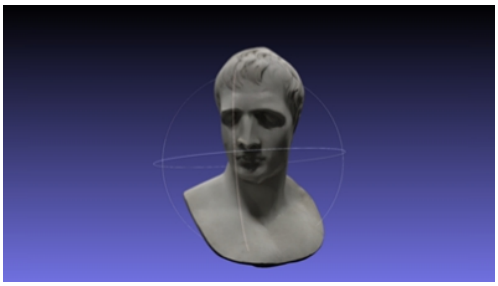
Predicted mesh

Chamfer Distance: 0.0274
Hausdorff Distance: 0.0813

Scan 06

Original mesh

Predicted mesh

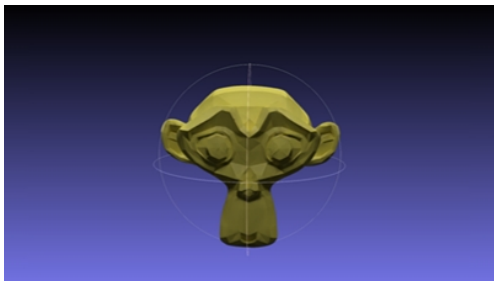


Chamfer Distance: 0.0037
Hausdorff Distance: 0.0125

Scan 07

Original mesh

Predicted mesh

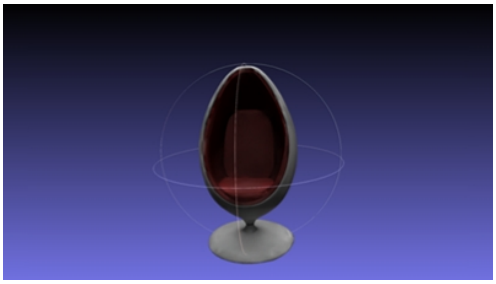


Chamfer Distance: 0.0031
Hausdorff Distance: 0.0086

Scan 08

Original mesh

Predicted mesh

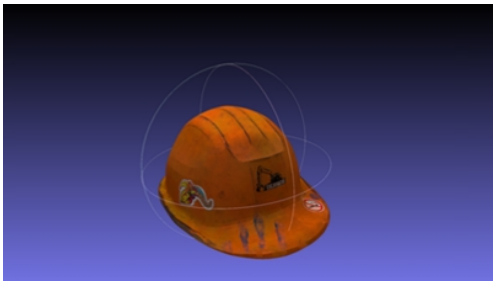


Chamfer Distance: 0.0118
Hausdorff Distance: 0.0397

Scan 09

Original mesh

Predicted mesh



Chamfer Distance: 0.0019
Hausdorff Distance: 0.0109

Scan 10

Original mesh

Predicted mesh

Chamfer Distance: 0.0121
Hausdorff Distance: 0.0089

Figure 4.3: Results of synthetic dataset

4.2.2 Real-World Dataset

Scan 11

Original mesh

Predicted mesh

Scan 12

Original mesh

Predicted mesh

Scan 13

Original mesh

Predicted mesh

Scan 14

Original mesh

Predicted mesh

Scan 15

Original mesh

Predicted mesh

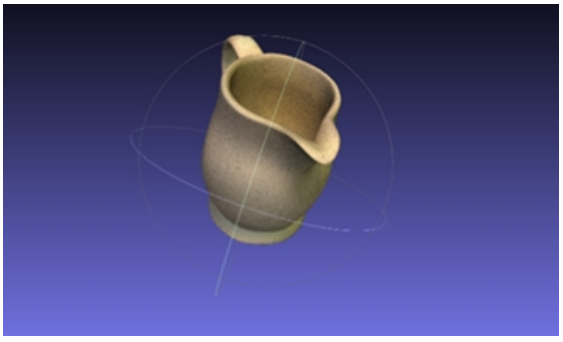


Figure 4.4: Results of real world dataset

4.2.3 Result Comparison

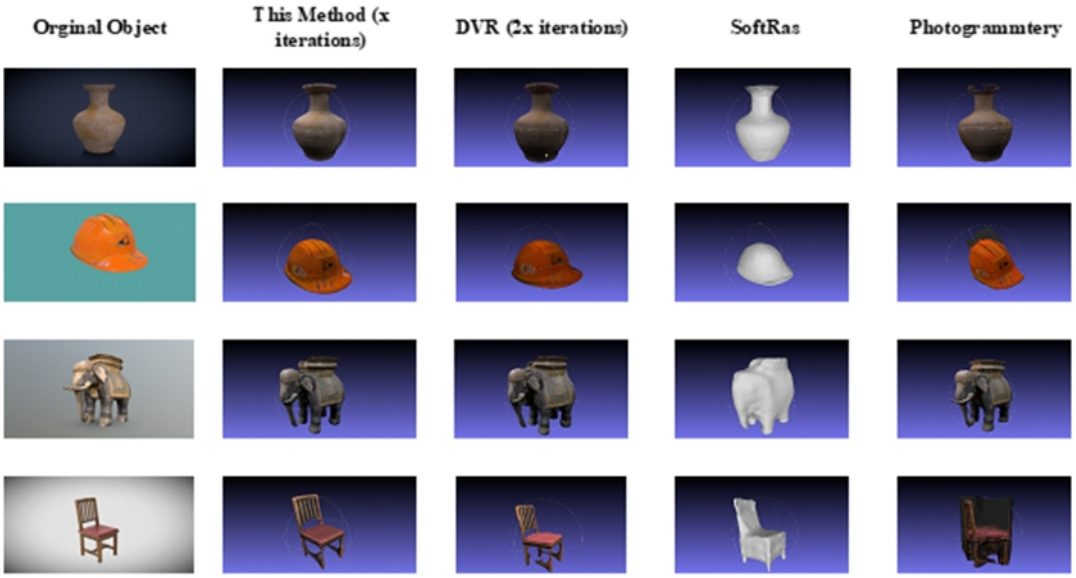


Figure 4.5: Results Comparison

	This Method	DVR	SoftRas	Photogrammtery
scan 01	0.0028	0.0047	0.0104	-
scan 02	0.0105	0.012	0.0112	-
scan 03	0.0011	0.002	0.0035	0.0158
scan 04	0.0017	0.0026	0.0043	-
scan 05	0.0274	0.0365	0.062	0.2576
scan 06	0.0037	0.0068	0.008	-
scan 07	0.0031	0.0042	0.0054	-
scan 08	0.0118	0.0149	0.0212	-
scan 09	0.0019	0.0027	0.0043	0.1016
scan 10	0.0121	0.0213	0.0306	0.0087
mean	**0.00761**	**0.01077**	**0.01609**	**0.095925**

Table 4.1: Result Comparison: chamfer distance wrt. to original mesh

Figure 4.5 shows results of this method with other baseline methods. Table 4.1 shows the chamfer distance with respect to the original mesh for synthetic data set results. Finally, I have calculated the mean for the chamfer distance for each method. The proposed method has the lowest mean value. So based on the results I can conclude that proposed method pro¬vide comparable results for both geometry and texture.

Screened Poisson surface reconstruction is another reconstruction method which use point cloud to generate the mesh. Since I have image, depth map, intrinsic and extrinsic parameters, can create the point cloud. Then I can use Screened Poisson reconstruction to convert the point cloud to mesh. If the point cloud is accurate, this

method can provide good results. So for the synthetic data, this gives good results. But when it comes to the real-world data set, it is highly unlikely that we can have a very accurate point cloud. When we apply the Screened Poisson reconstruction to a real-world data set it does not provide that much of good results. For both synthetic data and real world data there are some limitations with this method. for high textured objects, the texture information provided by this method is not good as well. Another major problem is that; the obtained reconstruction results are not clean. A 3D artist has to spend considerable time to clean the mesh before use for any other purpose. But the proposed method solves both of these problems.

Before generating the mesh, I have to estimate the normal for the point cloud. If the normal did not generate well, the output of the final mesh does not look good. As you see in the above image Screened Poisson reconstruction has some issues. The texture information has not generated well. So the proposed method provides good results when compared with the Screened Poisson surface reconstruction.

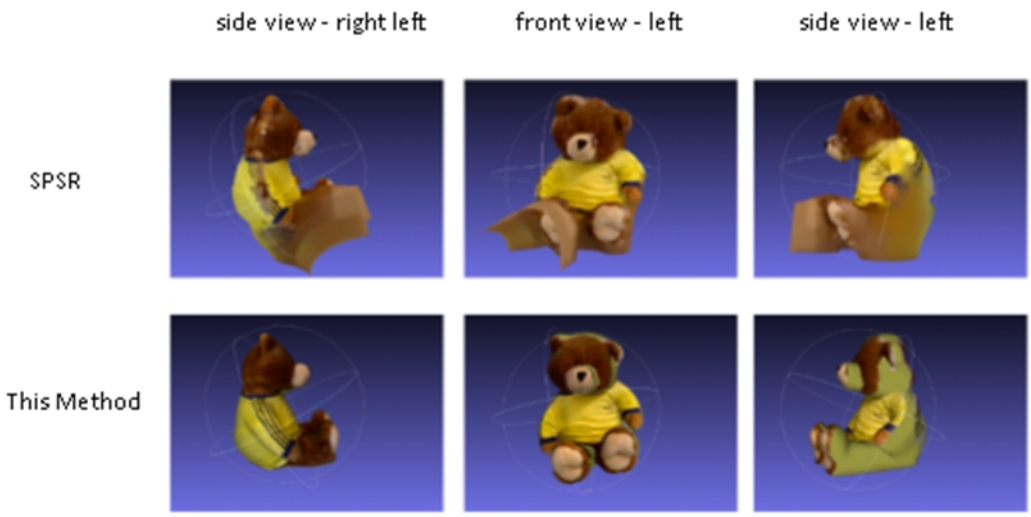


Figure 4.6: Result Comparison: SPSR vs This Method (scan 11)

4.3 Discussion

In this thesis, I have proposed a method to reconstruct objects from RGDB information. I predicted the geometry using Deep Leaning and generated the texture using MVS method. I used the chamfer distance and Hausdorff distance to compare the generated mesh with ground truth.

Then I compared the results of this method with other baseline methods like DVR, Softras, and photogrammetry. This method gives comparable results when compared with other reconstruction methods.

There are limitations to the proposed method as well. The main bottleneck of this method is the extrinsic generation. To generate the extrinsic high feature background is needed. For smaller objects, can use a marker to increase the features of the background. For large object we can do the same but there are practical issues. Due to this reason it is difficult to get good results for large objects. Depth sensors such as kinect fail to find the depth of transparent objects which makes 3D reconstruction of such objects a challenge. So this method does not provide good results when it comes to transparent objects.

5. CONCLUSION AND FUTURE WORKS

In this project, I have proposed a method to learn 3D reconstruction from multiple images. The presented method can also use synthetic data sets as well as real world data sets. This method also provides a way to generate masks and camera parameters, which will be useful for anyone who works with Deep Learning with 3D data. Anyone with rgb images and depth images of an object can use this method to generate 3D mesh with acceptable quality.

In the future, I hope to introduce a Deep Learning method to generate camera parameters. This will increase the performance of the system. Also I hope to introduce a method to capture the data with any mobile device that has the ability to obtain the depth information. This will help to grow 3D scanning, AR and VR technologies.

REFERENCES

[1] Mur-Artal, Raul & Tardos, Juan. (2016). ORB-SLAM2: an Open-Source SLAM System for Monocular, Stereo and RGB-D Cameras. IEEE Transactions on Robotics. PP. 10.1109/TRO.2017.2705103.

[2] D. J. Rezende, S. Eslami, S. Mohamed, P. Battaglia, M. Jaderberg, and N. Heess. Unsupervised learning of 3d structure from images. arXiv preprint arXiv:1607.00662, 2016

[3] Hahner, Martin & Varesis, Orestis & Bountouris, Panagiotis. (2017). Simulating Structure-from-Motion.

[4] Schönberger, Johannes & Frahm, Jan-Michael. (2016). Structure-from-Motion Revisited. 10.1109/CVPR.2016.445.

[5] H. Su, S. Maji, E. Kalogerakis, and E. Learned-Miller. Multi-view convolutional neural networks for 3d shape recognition. In ICCV, 2015

[6] Y. Yang, C. Feng, Y. Shen, and D. Tian. Foldingnet: Interpretable unsupervised learning on 3d point clouds. arXiv preprint arXiv:1712.07262, 2017

[7] Ducke, Benjamin. (2018). Multi□View Stereo. 1-4. 10.1002/9781119188230.saseas0398.

[8] Rock, J., Gupta, T., Thorsen, J., Gwak, J., Shin, D., Hoiem, D.: Completing 3d object shape from one depth image. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2015) 2484–2493

[9] R. Girdhar, D. F. Fouhey, M. Rodriguez, and A. Gupta. Learning a predictable and generative vector representation for objects. In ECCV, 2016.

[10] Choy, Chris & Xu, Danfei & Gwak, JunYoung & Chen, Kevin & Savarese, Silvio. (2016). 3D-R2N2: A Unified Approach for Single and Multi-view 3D Object Reconstruction. 9912. 628-644. 10.1007/978-3-319-46484-8_38.

[11] Yan, Xinchen & Yang, Jimei & Yumer, Ersin & Guo, Yijie & Lee, Honglak. (2016). Perspective Transformer Nets: Learning Single-View 3D Object Reconstruction without 3D Supervision.

[12] J. Gwak, C. B. Choy, M. Chandraker, A. Garg, and S. Savarese. Weakly supervised 3d reconstruction with adversarial constraint. In 3DV, 2017

[13] Park, Jeong & Florence, Peter & Straub, Julian & Newcombe, Richard & Lovegrove, Steven. (2019). DeepSDF: Learning Continuous Signed Distance Functions for Shape Representation.

[14] G. Riegler, A. O. Ulusoy, and A. Geiger. Octnet: Learning deep 3d representations at high resolutions. In CVPR, pages 6620–6629. IEEE, 2017

[15] Fan, Haoqiang & Su, Hao & Guibas, Leonidas. (2017). A Point Set Generation Network for 3D Object Reconstruction from a Single Image. 2463-2471. 10.1109/CVPR.2017.264.

[16] C. R. Qi, L. Yi, H. Su, and L. J. Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In NIPS, pages 5099–5108, 2017

[17] Sun, Xingyuan & Wu, Jiajun & Zhang, Xiuming & Zhang, Zhoutong & Zhang, Chengkai & Xue, Tianfan & Tenenbaum, Joshua & Freeman, William. (2018). Pix3D: Dataset and Methods for Single-Image 3D Shape Modeling. 2974-2983. 10.1109/CVPR.2018.00314.

[18] Sitzmann, V., Thies, J., Heide, F., Nießner, M., Wetzstein, G., Zollh¨ofer, M.: Deepvoxels: Learning persistent 3D feature embeddings. In: CVPR (2019)

[19] Niemeyer, Michael & Mescheder, Lars & Oechsle, Michael & Geiger, Andreas. (2019). Differentiable Volumetric Rendering: Learning Implicit 3D Representations without 3D Supervision.

[20] Knapitsch, J. Park, Q.-Y. Zhou, and V. Koltun. Tanks and temples: Benchmarking large-scale scene reconstruction. ACM TOG, 36(4):78, 2017

[21] Liu, Shichen & Chen, Weikai & Li, Tianye & Li, Hao. (2019). Soft Rasterizer: A Differentiable Renderer for Image-Based 3D Reasoning. 7707-7716. 10.1109/ICCV.2019.00780.

[22] Yiyi Liao, Simon Donne, and Andreas Geiger. Deep march- ´ ing cubes: Learning explicit surface representations. In Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), 2018

[23] Yiyi Liao, Simon Donne, and Andreas Geiger. Deep march- ´ ing cubes: Learning explicit surface representations. In Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), 2018

[24] David Stutz and Andreas Geiger. Learning 3d shape completion from laser scan data with weak supervision. In Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), 2018.

[25] Zubić, Nikola & Lio, Pietro. (2021). an Effective Loss Function for Generating 3D Models from Single 2D Image without Rendering.

[26] Thu Nguyen-Phuoc, Chuan Li, Stephen Balaban, and Yongliang Yang. Rendernet: A deep convolutional network for differentiable rendering from 3d shapes. arXiv preprint arXiv:1806.06575, 2018

[27] Angjoo Kanazawa, Shubham Tulsiani, Alexei A Efros, and Jitendra Malik. Learning category-specific mesh reconstruction from image collections. arXiv preprint arXiv:1803.07549, 2018

[28] Xu, Q., Wang, W., Ceylan, D., Mech, R., Neumann, U.: Disn: Deep implicit surface network for high-quality single-view 3d reconstruction (2019)

[29] Tulsiani, S., Efros, A.A., Malik, J.: Multi-view consistency as supervisory signal for learning shape and pose prediction (2018)

[30] A Kar, S. Tulsiani, J. Carreira, and J. Malik. Categoryspecific object reconstruction from a single image. In CVPR, 2015