Constructing 3D models from 2D images using monocular depths

H.W.R.L Hendawitharana#1, S. Suthakar#2,

#Department of Computer Science, University of Jaffna

Jaffna, Sri Lanka

[1]rumithwitharana5@gmail.com

[2]sosuthakar@univ.jfn.ac.lk

Abstract

This study focuses on improving the DVR method through 3D reconstruction with fewer input data. The target is to generate the neighboring views of 2D images to reconstruct a 3D shape with fewer input images. This work proposes a deep neural network that predicts the depth maps from an encoder-decoder structure. Here we compare monocular depth estimation techniques.Open3D is an open -source library that deals with 3D data. Here we used Open3D for generating point Clouds.

Introduction

Before the invention of 3D scanning, many scientists used physical models to store 3D information.

Information storage is now in a digital stage, so instead of cabinets, we use computers.

Finding a 3D model of a particular thing is normally easy, but making one that is accurate is quite difficult

In this study, we focus on how to construct the more accurate 3D shapes with fewer 2D images

Objective

The objective of this research project is to construct 3D shape with real data using convolutional neural network (CNN). Even though there are a lot of 3D reconstruction systems using 2D images. So there is a scope to implement a better reconstruction approach using emerging technologies.

So I came up with the idea of improving the DVR method for real data.

Data Set

DTU

- This data set (freely available) is aimed at Multiple View Stereo (MVS) evaluation.
- The data set consist of 124 different scenes.
- The data set contain RGB images, binary mask for each image, camera parameters, depth map of each image.

Differentiable Volumetric Rendering approach

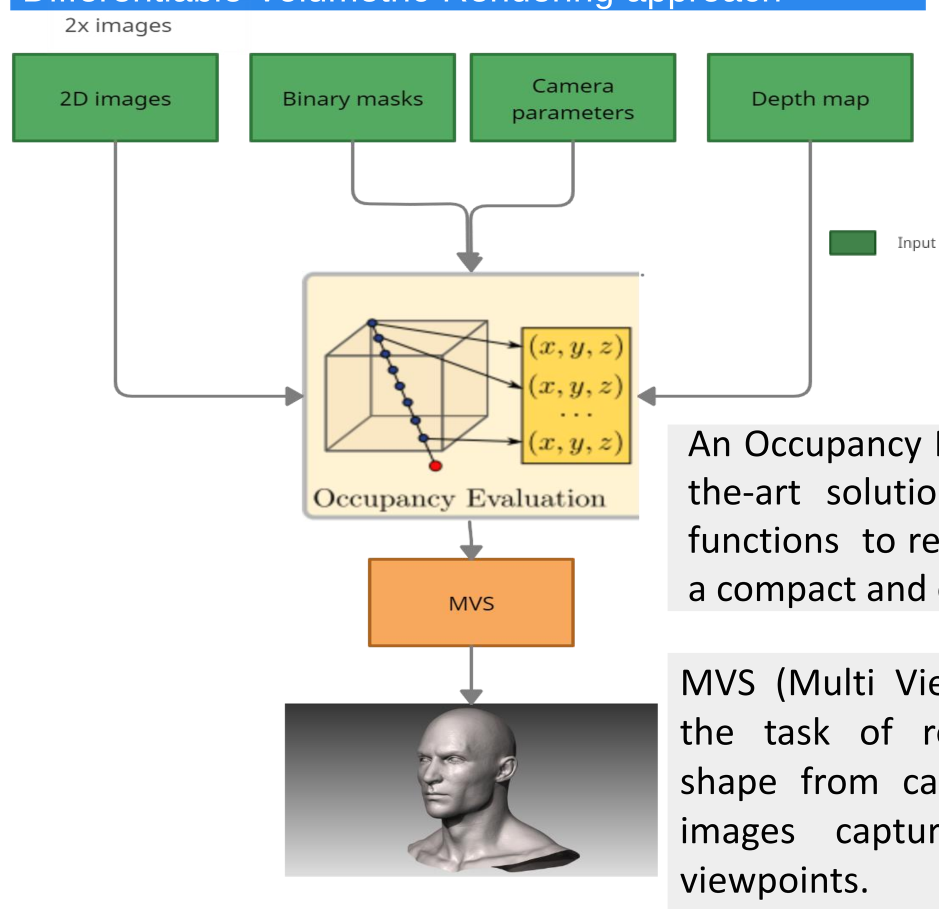


Fig. 1 Differentiable Volumetric Rendering approach

An Occupancy Network is a state-of-the-art solution that uses implicit functions to represent 3D objects in a compact and expressive manner.

MVS (Multi View Stereo) refers to the task of reconstructing a 3D shape from calibrated overlapping images captured from different viewpoints.

Methodology

Detect the Foreground

- DeepLabv3+ is a state-of-the-art deep learning model for semantic image segmentation (Google's latest and best performing Semantic Image Segmentation model)
- DeepLabv3+ have achieved a much more detailed segmentation map by employing an encoder-decoder network architecture.
- The segmentation map generated by the DeepLabv3+ model internally creates a binary mask.
- In this approach, we used this mask to detect the foreground of the images.

Estimate the Depth

- An encoder-decoder architecture is developed for estimating depths.
- The encoder is based on the pre-trained model ResNet-18.
- ResNet-18, a convolutional neural network with 18 layers, is used as a backbone for the depth estimation part.
- With this architecture, additionally we provided the Depth hints using a mono-trained model for better solution.
- The encoder outputs several feature maps with high level features.
- The output of the encoder was fed into the decoder, then the output gradually improved, refined, and finally provided a high-resolution depth map as output.
- It is composed of the best performing ELU activation function, and sigmoid function.
- The architecture of CNN, which has 5 blocks with 14 convolutional layers, respectively.
- The decoder part, which is often termed as a “deconvolution”, is used to gradually upsample the feature maps obtained by the encoder into a semantically segmented output image (depth image).

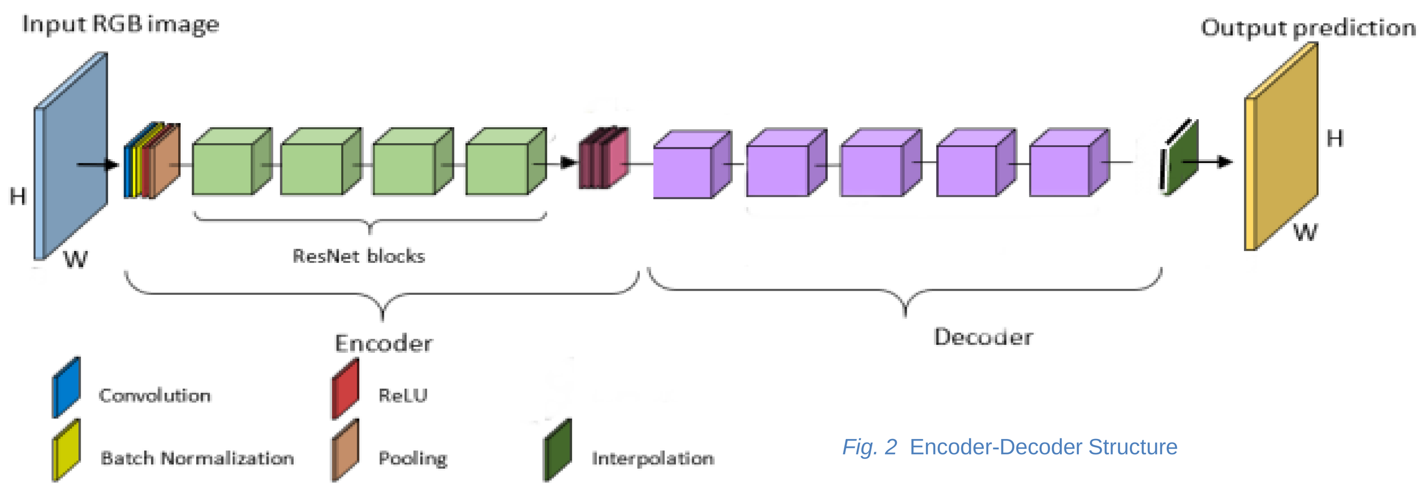
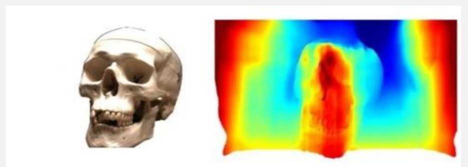


Fig. 2 Encoder-Decoder Structure

The encoder starts with a convolution layer, batch normalization, an activation layer, and a max-pooling layer. Next, It follows four ResNet blocks, then decoder follows 5 blocks (14 convolution layers) and interpolations respectively.

Generate Point Clouds

- Obtain the x, y ,z coordinates from depth image that got from the encoder-decoder architecture and get the colour information from the original input image.



Obtain the neighbor view

- Finally this approach uses a transformation matrix to rotate the pointCloud.
- Then with the x, y coordinates and colours, the neighbour view of the input RGB image is generated.

The Approach

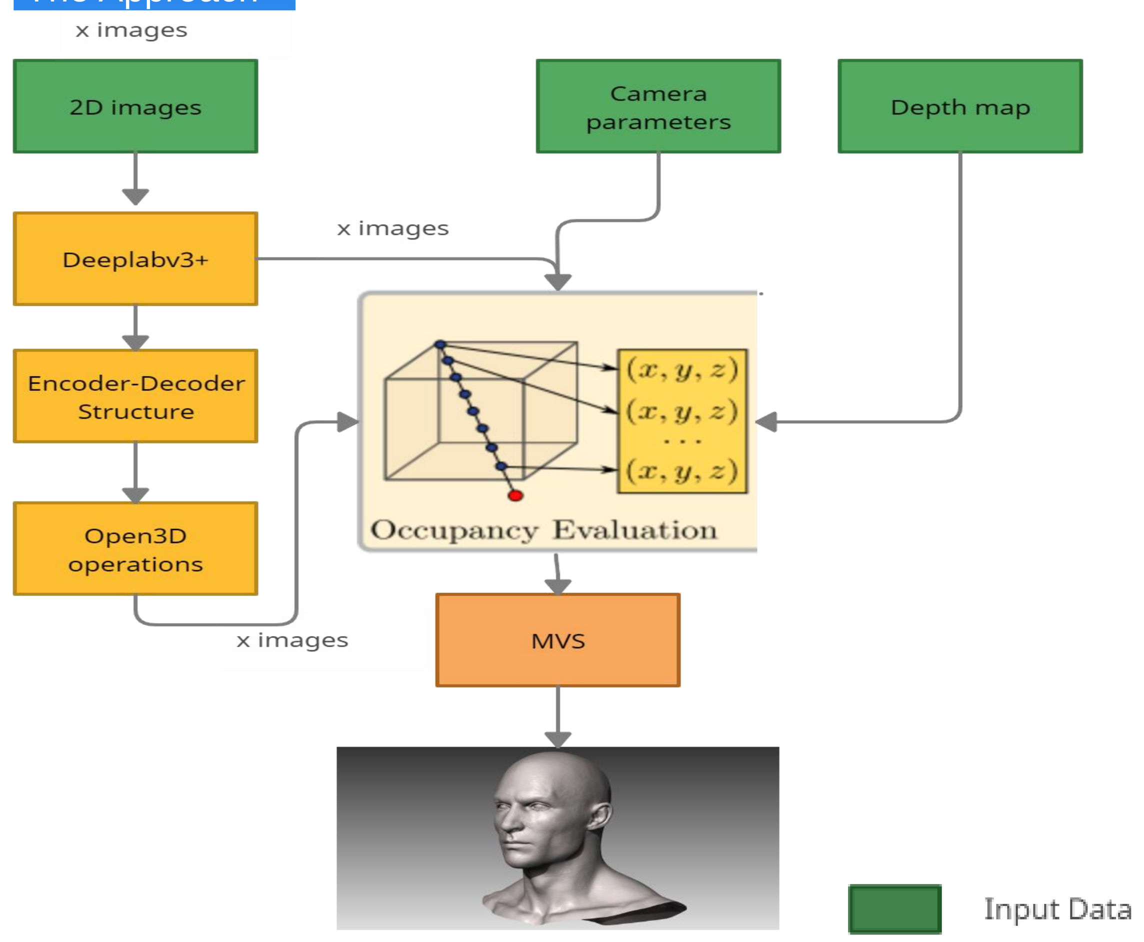


Fig. 3 The approach

Testing results

The table shows the accuracy of data regards to with or without depth hints.

	Error Rates				Accuracies		
	Abs Rel	Sq Rel	RMSE	RMSE log	δ<1.25	δ<1.25^2	δ<1.25^3
ResNet-18 Without Depth hints	0.112	0.953	5.007	0.207	0.862	0.949	0.976
ResNet-18 With Depth hints	0.112	0.929	4.960	0.204	0.867	0.951	0.976

So, ResNet – 18 with depth hints is better.

RMSE - Root Mean Square Error

- **Accuracies:** % of d_i s.t. $\max\left(\frac{d_i}{d_i^*}, \frac{d_i^*}{d_i}\right) = \delta < thr$,

In here di is the predicted depth value of pixel i, di∗ is the ground truth of depth, thr is the threshold value.

Testing results

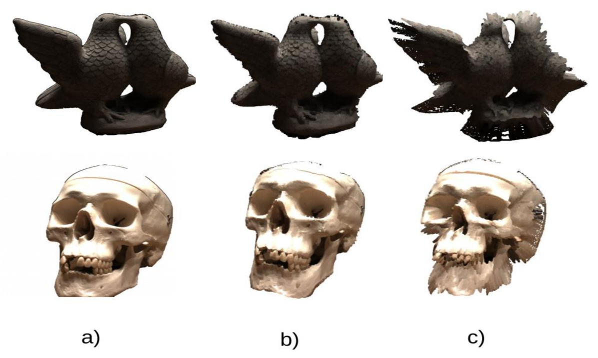
There are many models which can be used to identify depth hints.

a) Stereo-supervised model b) Mono-supervised model

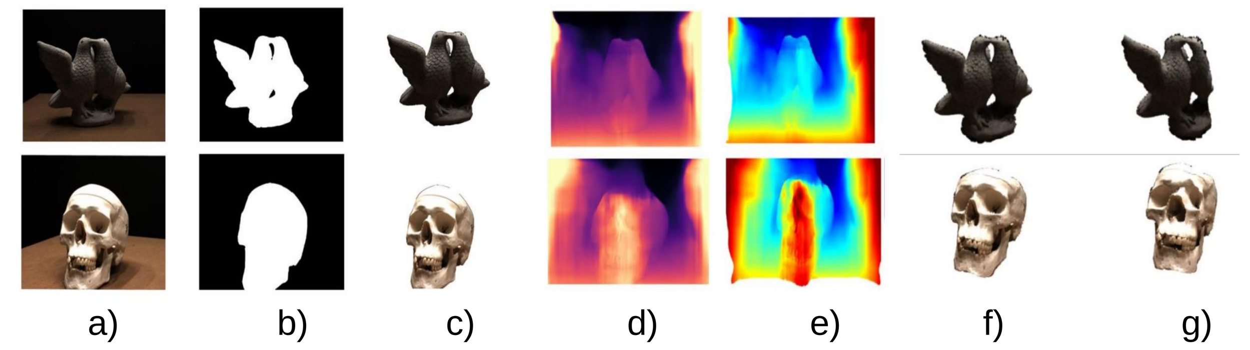
c) Mono-stereo-supervised model

Trained with ImageNet	Error Rates				Accuracies		
	Abs Rel	Sq Rel	RMSE	RMSE log	δ<1.25	δ<1.25^2	δ<1.25^3
ResNet-50 with stereo supervised	0.109	0.873	4.960	0.209	0.864	0.948	0.975
ResNet-50 with mono supervised	0.115	0.903	4.863	0.193	0.877	0.959	0.981
ResNet-50 with mono stereo supervised	0.106	0.818	4.750	0.196	0.874	0.957	0.979

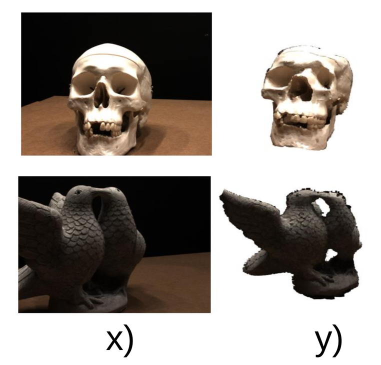
ResNet 18 has fewer layers than ResNet 50. So, training with a 50-layer ResNet takes longer training and test times. Therefore, This encoder is based on the pre-trained model, ResNet-18. With the above reasons, we have used the ResNet-18 with a mono-supervised (mono_1024x320) model.



a) RGB image (input image)
b) Open3D point cloud with depth hints
c) Open3D point cloud without depth hints



a) RGB image (input image) b) Binary Mask c) Detected Foreground image
d) Depth image e) pointCloud without colours f) pointCloud with colours
g) Rotated pointCloud (near view)



x) Actual near view with background
y) Predicted near view without background

Discussion & Conclusion

- This method have achieved depth estimation with an 18-layer encoder and a 14-layer decoder. This works introduced Depth Hints to help escape from local minima, to get a better overall solution.
- From the output of the decoder, This approach could calculate the x, y, z coordinates for the input RGB images.
- Then this approach have used Open3D to reconstruct the pointCloud. PointCloud.colors helps to map the colour into the pointCloud. So with the pointCloud transformation we can achieve the final goal.

References

1. Niemeyer, M., Mescheder, L., Oechsle, M., & Geiger, A. (2021). Differentiable Volumetric Rendering: Learning Implicit 3D Representations Without 3D Supervision. Retrieved 14 November 2021, from https://openaccess.thecvf.com/content_CVPR_2020/html/Niemeyer_Differentiable_Volumetric_Rendering_Learning_Implicit_3D_Representations_Without_3D_Supervision_CVPR_2020_paper.html
2. Laina, I., Rupprecht, C., Belagiannis, V., Tombari, F., & Navab, N. (2021). Deeper Depth Prediction with Fully Convolutional Residual Networks. Retrieved 14 November 2021, from https://arxiv.org/abs/1606.00373v2
3. Al-Kaff, A., Martín, D., García, F., Escalera, A., & María Armingol, J. (2018). Survey of computer vision algorithms and applications for unmanned aerial vehicles. Expert Systems With Applications, 92, 447-463. doi: 10.1016/j.eswa.2017.09.033
4. Liu, Y., & Zhao, X. (2020). Constrained Image Splicing Detection and Localization With Attention-Aware Encoder-Decoder and Atrous Convolution. IEEE Access, 8, 6729-6741. doi: 10.1109/access.2019.2963745