

## Generation of 3D Model from Images

Akash Chaudhari<sup>1</sup>, Aditya Deo<sup>1</sup>, Mahesh Badhe<sup>1</sup>, Ritesh Patidar<sup>1</sup>, Sunil Rathod<sup>2</sup>

<sup>1</sup>Student, Department of Computer Engineering, Dr. D. Y. Patil School of Engineering, Lohegaon, Savitribai Phule Pune University, Pune, Maharashtra, India

<sup>2</sup>Professor, Department of Computer Engineering, Dr. D. Y. Patil School of Engineering, Lohegaon, Savitribai Phule Pune University, Pune, Maharashtra, India

### ABSTRACT

3D models are used in various fields such as video games, films, architecture, illustration, engineering and commercial advertising. We have seen significant progress in 3D model generation and reconstruction in recent years. In this paper, we discussed how to convert a 2D image into a 3D model. Creating a 3D Model takes lots of effort starting from scratch in software like Maya or Blender. We are proposing a tool that allows you to generate a 3D Model from a single 2D image. The tool uses a pre-trained machine learning model in the background to generate a 3D Model. The pre-trained model is based on Hierarchical Surface Prediction (HSP). HSP uses Convolutional Neural Network (CNN) which is good at processing visual data like images, 3D Models with low computational power.

**Keywords :** HSP, CNN, 3D Model, DIB-R

### I. INTRODUCTION

The 3D modelling is a computer graphics technique that makes it possible to produce a 3D digital representation of any object or surface. The 3D model is a collection of points/vertex, edges, and faces. The 3D models are used in various media, including video games, movies, architecture, illustration, engineering and commercial advertising. This is what engineers and architects use to plan and design their work. Animators and game developers use 3D modelling to turn their ideas into reality. Due to their value, a vast market for 3D models is available online.

The 3D modelling is an accurate workflow that often involves the careful placement of individual vertices

to obtain the correct contours of the desired object. There are several tools available to accelerate the modelling process. Most programs include a mirror technique that allows the artist to build an asymmetric model by working on only half or even a quarter of the object.

So to solve the problem of manual work like changing vertex position, or adding a new vertex, and many more things the 3D modelling is required as a technique to do it using algorithm. we have developed an application that allows generating 3D models automatically. The system takes an input image that holds an object for which the user wishes to generate the 3D model. Next, it processes the image and makes it appropriate for a subsequent process. After

completion of the process, the generated 3D model is automatically saved on the local disk in .obj format.

A Hierarchical Surface Prediction (HSP) algorithm is used for the generation of models. It uses Convolutional Neural Network (CNN) to detect important characteristics of the input image and generate a 3D model matching the object displayed in the image. Hierarchical surface prediction predicts the surface of a 3D model, rather than the entire volume, or whatever is within the surface of the desired 3D model.

Our application provides an interactive User Interface (UI) to end-users which allows them to generate a model of isometric images, view the previously generated images and search for the queries in the FAQ and well-written user manual.

Our goal is to automate the process of creating 3D models and increase productivity by reducing the time it takes to manually generate models.

## II. RELATED WORK

Most of the 3D Generation is done using Generative Adversarial Networks. There are various approaches proposed for 3D Model generation in the field of GANs also. Some of them are "Learning a Probabilistic Latent Space of Object Shapes via 3D Generative-Adversarial Modeling (3D GAN)" [1], "Learning to Predict 3D Objects with an Interpolation-based Differentiable Renderer (DIB-R)" [2]. GAN needs a lot of computation power.

DIBR predicts and renders a whole 3D Model. It makes rendering pipeline differential which allows it to use Machine Learning in the process. DIB-R also predicts the texture and light details of a 3D Model. It needs a lot of computation power.

Recently Hierarchical Surface Prediction (HSP)[3] has shown that it is possible to predict 3D Models using Convolutional Neural Network (CNN) [4]. The HSP focuses on the surface of an object, which is what we only need to render a 3D Model. The surface of a 3D

Model predicted by HSP is high resolution and smooth, although it doesn't predict texture, and lighting (shadows, reflections, etc.) like DIBR.

HSP seems efficient than others because it predicts a high-resolution 3D Model with low relative computation power. Our tool uses its pre-trained model for the generation of 3D Models.

## III. SYSTEM ARCHITECTURE

The FLASK Light-weighted web app for the complete user interface for the end-user. The web application includes different features like generating a 3D model, history, FAQ, and user manual to manage the web application in a particular way.

Our system takes a single 2D image, and upstream server (Colab Server) URL as inputs. Upstream server is responsible for generation of model. The Upstream server accepts this image via the flask server and saves the received file in the Google Drive folder. The saved file is passed as an argument with the training data set already available on the drive for 3D model generation. The algorithm generates a 3D model and stores it in the output folder using the .obj extension. The colab flask server sends the 3D model by the algorithm in response to a received 2D image.

WebGL (Web Graphics Library) is a JavaScript API for rendering high-performance interactive 3D, 2D graphics received 3D models over the web app.

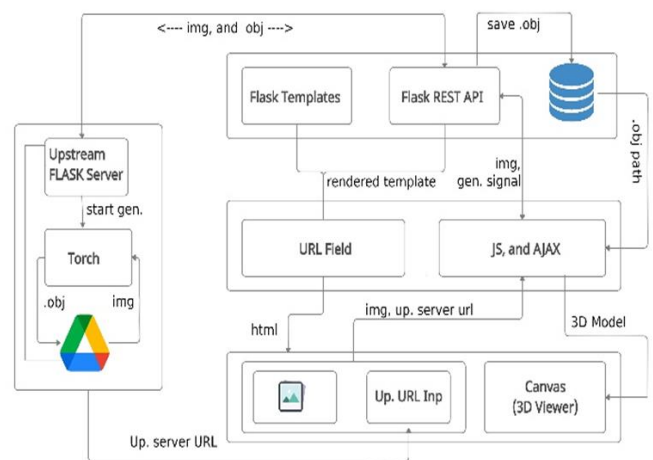


Figure-1: System Architecture

In this system architecture, we have the following modules

1. Flask (Lite version of Django )  
Flask Server is used to view the complete user interface and manage file transfer from the local machine to the server (Colab)
2. WebGL (Web Graphics Library)  
WebGL (Web Graphics Library) is a JavaScript API for rendering high-performance interactive 3D and 2D graphics in any compatible web browser with no plug-ins. It is for previewing the 3D template on canvas from the colab server.
3. JavaScript and AJAX  
JS Function `get3DModel` uses AJAX to send 2-D images and receives the 3D model generated by the colab server in the background. It validates the URL field input by the user.
4. Upstream Flask Server  
Colab server makes the upstream server live on the Internet to temporarily using ngrok python library. It receives the 2D image sent by the local flask server and sends the generated 3D object file as a reply.
5. Colab  
3D Model Generation Algorithm (CNN) is running behind whenever the 2D image came from the server it passes as an argument for algorithm along with training dataset
6. Google Drive  
Bundle of training data, 2D images, and 3D-Model stored on Google Drive.

#### IV. ALGORITHM

1. Define input image size, and give it to the input layer of CNN.
2. Define n Convolution layers(Conv + Pooling).
  - define weight, and give number of filter, kernels, and other hyperparameters

- connect each with previous layer, if first then connect with input layer
3. Define fully connected layer.
    - output of fully connected layer will be 3D model data(vertices, edges, faces).
    - connect fully connected layer to the last convolutional layer.
  4. Train the model using Backprop, and gradient descent by providing input images.
  5. Test the ML model by feeding test images.
  6. If output of ML model gives 90% accuracy, then save the trained model as `t7`, or `h5` file for future use.
  7. If output accuracy is low, then train the model again with desired changes

#### V. RESULTS

Figure-1 is a home screen. Using the deployment server, the application is launched, which may be obvious from the address in the address bar. It contains the following four features: Generate a 3d model, History/Records, FAQ and User Manual.

Figure-2 is Generate 3D Model. The user selects the image to upload by choosing the file option, then adds the upstream server URL to the text field and click Continue. In the rendered view of the 3D model, the generated model is shown.

Figure-3 is History/Records. All the models previously generated can be viewed in the history function. Clicking on the images from the list on the left allows the user to view their respective models.

Figure-4 is FAQ. The questions are based on different categories and by clicking on any category, the user will be able to see the questions below.

Figure-5 is User Manual. A well-defined process server as a guide simplifies the use of the application for the new user.

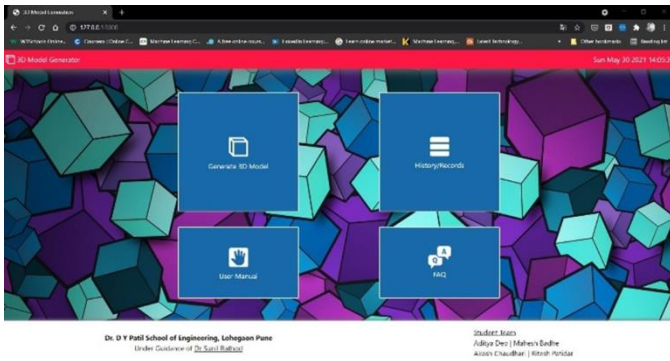


Figure-1: Home Screen

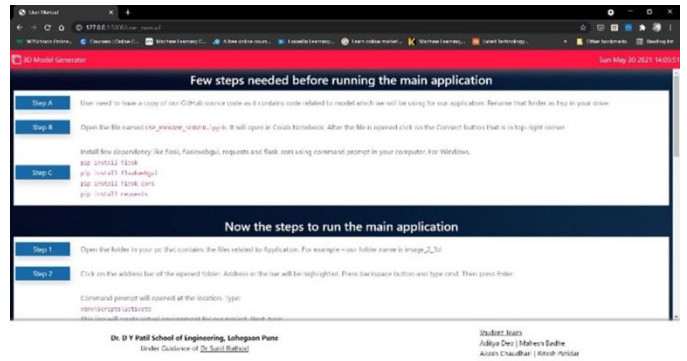


Figure-5: User Manual

## VI. CONCLUSION

In this paper, a tool with a user interface is proposed which allows user to generate a 3D Model using a single 2D image. The tool also allows user to see previously generated 3D Models, and data related to them. It uses a pre-trained Machine Learning model to generate a 3D Model. The ML algorithm/approach used here is called Hierarchical Surface Prediction (HSP), which uses CNN. Although 3D Models generated in this tool are not as perfect as manually created 3D Models by artists, combining them with human efforts can allow users to create 3D Models faster. Also, this area of auto 3D model generation is getting the attention of researchers. So, in future, we may see better 3D Models generated with the new algorithms. Our tools are scalable; therefore new algorithms can be embedded in it also.

## VII. REFERENCES

- [1]. Jiajun Wu\*, Chengkai Zhang\*, Tianfan Xue\*, William T. Freeman\*, Joshua B. Tenenbaum\*, Learning a Probabilistic Latent Space of Object Shapes via 3D Generative-Adversarial Modelling, NIPS 2016
- [2]. Wenzheng Chen, Jun Gao,, Huan Ling, Edward J. Smith, Jaakko Lehtinen, Alec Jacobson, Sanja Fidler, Learning to Predict 3D Objects with an

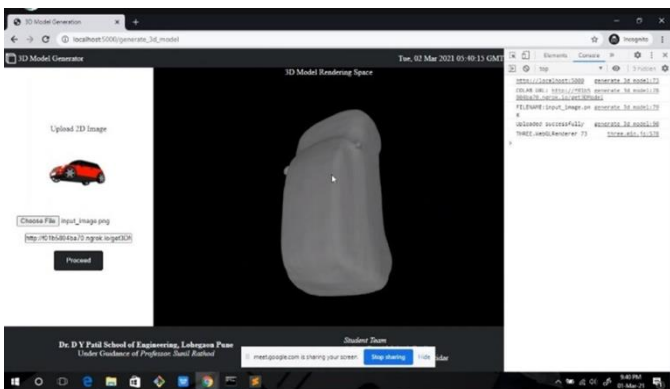


Figure-2: Generate 3D Model

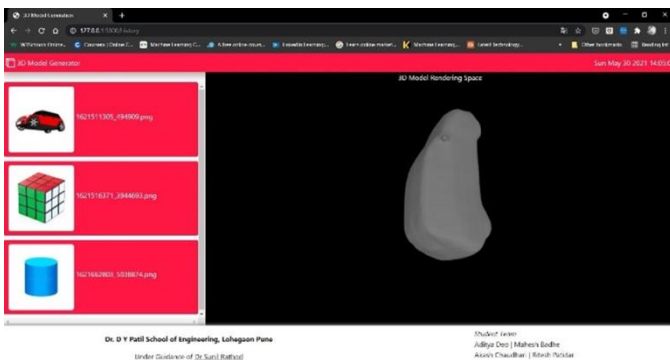


Figure-3: History/Records

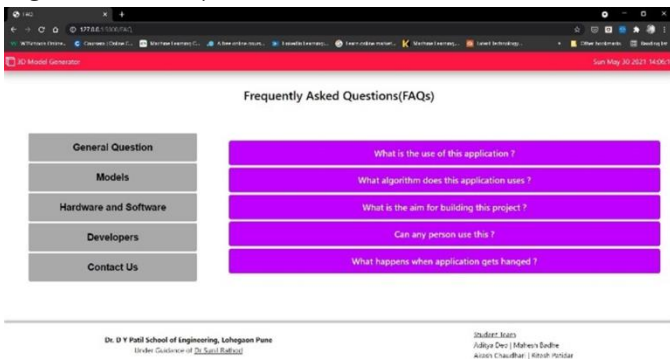


Figure-4: FAQ

Interpolation-based Differentiable Renderer,  
NeurIPS 2019

- [3]. Christian Hane, Shubham Tulsiani, Jitendra Malik, Hierarchical Surface Prediction for 3D Object Reconstruction, Nov 2017
- [4]. Deepu R, Murali S, 3D Reconstruction from Single 2D Image, January 2016 ISSN: 2454-5031