**AI ALGORITHMS**

**AIDI-1002-01**

**JEFFREY ARUNASALAM**

**100783993**

**Executive Summary:**

The general proposal is creating an AI web client that Create a 3-dimensional equivalent object from analyzing the 2-dimensional image inputted.

The scope of the project is to analyze inputted images from a user and create 3-dimensional object from said image (ex. Square -> Cube). Dimensions for image (object) can be included in the image, in which the ai can use computer vision to detect and Appropriately created the scaled object.

Out of scope (limitations) of the project is relaying/recreating textures (texture mapping) from the image onto the 3D object, attempting to create weight paint and bones (animation recreation purposes).

**Rationale Statement:**

2D to 3D conversions are usually paywalled, on sketchy sites, or not that reliable (non-payed, advertising sites). The goal of the project is to make this type of stuff more accessible to everyone.

**Problem:**

Website client needs to be created for users to start accessing the website.

3D Renderer to be created, or imported (based of another 3d renderer, like blender)

**Data Requirements:**

(1). Image Reference

A user inputted image by uploading to the server.

*\*\*Main source of data collection for training the AI, besides other research findings on 2d to 3d computer vision imaging*

*\*\*Images are deleted after, not saved/kept for privacy, legal, and storage reasons.*

(2). Dimensions

The user would have to declare the dimensions of the image, so it can be accurately recreated as an 3D object.

*\*\*user must manually input the dimensions*

(3). Different Axis Views of the image (multiple photographs) (x, y, z axis)

The image will be easier to create with different angle of views.

*\*\*This is for if the image that is not symmetric (one half is completely different from the other)*

*\*\*Most cases images are not really symmetrical, so this option is expected to be used more often and collect in more data.*

(4). Point Cloud Representations

A way to get properties from the image and store information from collected from the data.

(5). Degree of Rotation

Rotation parameter of the object recreated from the image

\*\*User inputted or automatically detect, measured in pi, radians, degrees.

(6). 3D Renderer

Custom 3D renderer to show the created object from the inputted images.

(7). Surface parametrization

Discrete Differential geometry, helpful with 2D to 3D mapping.

(9). File Format Input

Input File formats that will be recognized: JPEG, PNG.

*\*\*More file formats to be added as there is more progress.*

(10). File Format Output

The final product will be a 3D object in the final format of the user choices of OBJ, FBX.

*\*\*More file formats to be added as there is more progress.*

**Data & Model/Architecture Approach:**

Main approach for creating a 3D object from a 2D image by using multi-view images (1 image for more simpler objects) for creating the 3d object. I will most likely use this because it is so far the simplest way of creating a conversion without getting too much advanced and adding more time to the project. Though if more advanced algorithms are deemed to more feasible, the projection duration increased will be worth it.

There are multiple 3d representations available such as depth map, volumetric, polygonal mesh, point cloud, and primitive-based CAD models. Microsoft Kinect algorithms also play a role in dissecting the image, its 3D imaging algorithm can be used for static imaging. Earlier Microsoft Kinect algorithms also mainly focused on 2d imaging.

[**http://graphics.stanford.edu/courses/cs468-17-spring/LectureSlides/L13%20-%203d%20deep%20learning%20on%20point%20cloud%20representation.pdf**](http://graphics.stanford.edu/courses/cs468-17-spring/LectureSlides/L13%20-%203d%20deep%20learning%20on%20point%20cloud%20representation.pdf)

**https://arxiv.org/pdf/1711.10669.pdf**

[**https://jhonykaesemodel.com/publication/3dv2017/**](https://jhonykaesemodel.com/publication/3dv2017/)

**https://arxiv.org/pdf/1604.00999.pdf**

[**https://github.com/Amir-Arsalan/Synthesize3DviaDepthOrSil**](https://github.com/Amir-Arsalan/Synthesize3DviaDepthOrSil)

[**https://www.photomodeler.com/products/how-it-works/**](https://www.photomodeler.com/products/how-it-works/)

[**https://www.kaggle.com/imoore/2d-and-3d-plotting-tutorial-in-python**](https://www.kaggle.com/imoore/2d-and-3d-plotting-tutorial-in-python)

[**https://explore.openaire.eu/search/publication?articleId=od\_\_\_\_\_\_\_\_18::a6cdac48e8bca44502bbeb4c96ef8197**](https://explore.openaire.eu/search/publication?articleId=od________18::a6cdac48e8bca44502bbeb4c96ef8197)

<https://openaccess.thecvf.com/content_cvpr_2017/papers/Soltani_Synthesizing_3D_Shapes_CVPR_2017_paper.pdf>

<https://github.com/timzhang642/3D-Machine-Learning#3d_synthesis>

**Project Plan:**

\*\*Times are general estimates

* Create web client to host the ai (something similar to other ai websites, example: <https://icons8.com/upscaler>) [1 h] (Deadline: 1 Day)
* Create a basic user interface [2 h] (Deadline: 1 Day)
* Offering options to create object from image ui input (1 image, or multiple from different views, axis) [1-3 h] (Deadline: 1 Day)
* Create a 3D engine renderer [10 – 15 h] (Deadline: 1 - 2 week)
* Create image uploader [3 – 4h] (Deadline: 2 Day)
* Create image file recognizer (which image file being uploaded will be recognized) (3 – 5h) (Deadline: 2-3 Day)
* Create first basic algorithms (basic 2d image to 3d object conversion, square to cube) [2 – 5 h] (Deadline: 2-3 Day)
* Create file outputs as FBX, OBJ (3D Object) [2 - 5 h] (Deadline: 2-3 Day)
* Tune basic algorithm [1-3 h] (Deadline: 1 Day)
* Add in more algorithms for more advanced images [10 - 15 h] (Deadline: 1 – 2 week)
* Test the program by uploading series of pictures and comparing with the 3D output, ai training [1 -3 h] (Deadline: 1 Day)
* Evaluate the program [1 -2 h] (Deadline: 1 Day)
* Refine the model [1 – 3h] (Deadline: 1 Day)