**2-D IMAGE TO 3-D MODEL CONVERSION**

**KGISL INSTITUTE OF TECHNOLOGY-COIMBATORE**

**MENTOR: MS.JANHANI.D (2nd YEAR)**

**DEPARTMENT OF COMPUTER SCIENCE AND BUSINESS SYSTEM**

**HARI KRISHNA.M[1], THARANEVEL.G.A[2], HARISHWA.S[3], PRANESH.C.S[4]**

[**harikrishna\_23me16@kgkite.ac.in**](mailto:harikrishna_23me16@kgkite.ac.in)

[**tharanevel\_23me54@kgkite.ac.in**](mailto:tharanevel_23me54@kgkite.ac.in)

[**pranesh\_23me38@kgkite.ac.in**](mailto:pranesh_23me38@kgkite.ac.in)

[**harishwa\_23me18@kgkite.ac.in**](mailto:harishwa_23me18@kgkite.ac.in)

**DEPARTMENT OF MECHANICAL ENGINEERING**

**INTRODUCTION**

To develop a computer solution to reconstruct a 3d model from 2d image. The goal is to generate accurate and detailed 3d representations of objects or scenes captured in 2d image. We used PIL (pillow) library from python to accept images in 2-D format. This Python code is a simple script that converts a 2D grayscale image into a basic 3D model represented in the Wavefront .obj file format. We convert the image to greyscale and then modelize it to 3-D.

Reviving Legacy Content: Converting classic or older 2D content to 3D can breathe new life into it, attracting audiences who may be drawn to the novelty of experiencing familiar content in a new dimension.

Adapting to New Technologies: With the rise of 3D displays, virtual reality (VR), augmented reality (AR), and 3D printing, there's a growing demand for 3D content. Converting 2D images, videos, and graphics to 3D allows content creators to leverage these emerging technologies and platforms.

Improving User Engagement: In industries such as entertainment, advertising, and gaming, 3D content often garners more attention and engagement from audiences. By converting 2D content to 3D, businesses can enhance their marketing efforts and create more captivating experiences for consumers.

***ABSTRACT***

Transforming 2D content into 3D not only enriches the viewing experience but also enhances comprehension across various domains. By imbuing depth into formerly flat images or scenes, viewers are transported into a more immersive and captivating visual realm.

This heightened engagement can be especially valuable in fields like architecture, engineering, medicine, and education, where precise spatial understanding is paramount. Moreover, the transition from 2D to 3D fosters a new realm of artistic exploration and expression.

Artists and designers are empowered to breathe life into existing artwork, infuse illustrations with newfound depth, and elevate animations and visual effects to unprecedented levels of realism and intrigue.

In essence, the conversion from 2D to 3D not only expands the horizons of perception but also unlocks boundless possibilities for creativity and innovation.

**THEORY**

**Aim of the Project:**

Creating a computer vision system to generate detailed 3D models from a set of 2D images involves utilizing sophisticated methodologies like structure-from-motion and multi-view stereo.

This approach aims to reconstruct objects accurately by analysing images captured from various angles, resulting in comprehensive 3D representations.

In related research, Clément Godard and colleagues introduced MonoDepth2, a self-supervised framework for monocular depth estimation. This method employs geometric and photometric constraints to train deep neural networks for predicting depth, eliminating the need for ground truth annotations. Additionally, Shikun Liu et al.

proposed DeepVoxels, a technique for learning persistent 3D feature embeddings from 2D images. Their approach utilizes convolutional neural networks to predict voxelized representations of 3D scenes from 2D images, enabling tasks such as 3D reconstruction and object manipulation.

These advancements represent significant contributions to the field of computer vision and hold promise for enhancing the capabilities of 3D modelling systems.

`

**Existing Project**

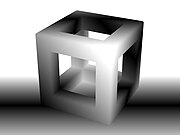
Alpha3D is revolutionizing the landscape of 3D content creation through its cutting-edge AI-powered platform. Regardless of prior experience in 3D modeling, users can swiftly convert text prompts or 2D images into intricate 3D digital assets within mere minutes.

With the surging popularity of extended reality (XR), consumers are increasingly drawn to immersive environments, prompting innovative brands to meet this demand head-on. However, the arduous, costly, and time-intensive nature of traditional 3D asset creation poses a significant hurdle.

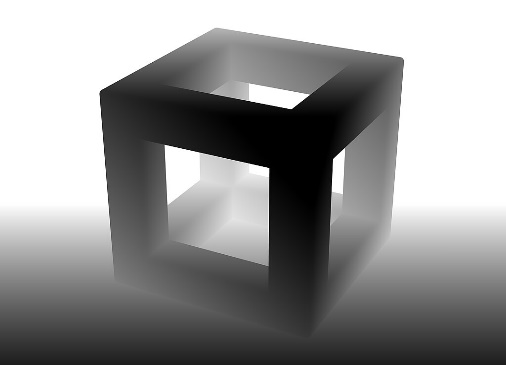
Crafting just one digital asset often consumes weeks of effort and thousands of dollars, acting as a bottleneck for XR platforms and hindering scalability. Alpha3D addresses this challenge, streamlining workflows and unlocking unprecedented potential for 3D content creation at scale.

**2-D IMAGE TO 3-D**

Depth Map Generation: The first step is to generate a depth map from the 2D image. This map assigns a depth value to each pixel in the image, indicating how far away that point is from the camera. Various techniques can be used for depth estimation, such as stereo vision (using multiple images from different viewpoints), structured light, or depth sensors like LiDAR.

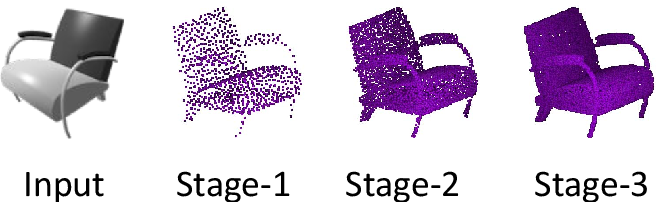
Original image depth mapped image



Depth mapped image (reversed)

Point Cloud Reconstruction:

Once the depth map is generated, it can be used to create a point cloud representation of the scene. A point cloud is a set of points in 3D space, where each point represents a surface point in the scene. Each point in the point cloud is associated with its corresponding pixel in the 2D image and its depth value from the depth map.



**Techniques used**

Image Processing with PIL (Python Imaging Library):

The script imports the Image module from PIL to handle image loading and manipulation.

Loading and Converting Image to Grayscale: The script loads a 2D image from the specified path using Image.open().

It converts the image to grayscale using the convert () method with the argument 'L'.

Height Map Generation: The script extracts pixel values from the grayscale image and arranges them into a 2D array representing a height map.

Each pixel's intensity represents the height of the corresponding point in the 3D model.

Extrusion to Create 3D Model: The script iterates through the height map to generate vertices for the 3D model.

For each pixel in the height map, a vertex is created with coordinates (x, y, intensity \* height\_factor), where intensity is the pixel value and height\_factor is a scaling factor.

Face Definition: The script defines faces by connecting vertices in a grid-like pattern.

It iterates through the height map and defines quad faces by referencing vertex indices.

For each grid cell, two quad faces are defined, resulting in a mesh that covers the entire height map.

OBJ File Generation: The script writes the generated vertices and faces to an OBJ file format. Vertices are written using the v prefix, and faces are written using the f prefix.

Note: OBJ files use 1-based indexing for vertex and face indices, so indices in the faces are incremented by 1.

Main Execution Block: The script defines a main execution block to execute the conversion process when the script is run as a standalone program.

It specifies the input image path, calls the convert\_to\_3d function to generate vertices and faces, and then saves the result to an OBJ file using the save\_obj\_file function.

Refinement and Post-processing: Finally, the generated 3D model may undergo refinement processes to improve its accuracy and visual quality. This can include noise reduction, smoothing, and optimization of the mesh or surface representation.

**LITERATURE SURVEY**

In the process of refining and post-processing generated 3D models, several techniques are applied to enhance their accuracy and visual quality.

This includes procedures such as noise reduction, smoothing, and optimizing the mesh or surface representation.

A literature survey outlines a method that takes an image file path and an optional height\_factor parameter.

It operates by opening the image, converting it to grayscale, and extracting pixel values to create a height map.

Through iteration, it generates vertices and faces for the 3D model, with vertices defined in 3D space (x, y, z) and z determined by the pixel intensity multiplied by the height\_factor.

Faces are formed using indices of vertices to create triangles comprising the surfaces of the 3D model. Additionally, a save\_obj\_file function writes the vertices and faces to a Wavefront OBJ file format. In the main section, the image path is specified, and the convert\_to\_3d function is called to generate vertices and faces. Finally, the save\_obj\_file function is invoked to save the resulting 3D model as an OBJ file.

This Python code effectively converts a 2D image into a simplified 3D model, where lighter areas in the image correspond to higher elevations in the 3D model.



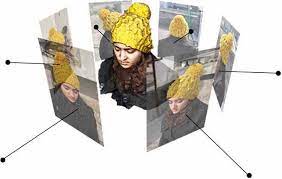
**Surface Reconstruction:**

Volumetric Reconstruction: Representing the scene as a volumetric grid and estimating surface properties from the occupancy or density of grid cells.

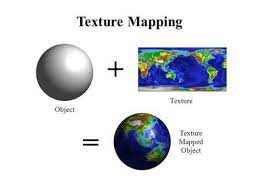
face Reconstruction: To create a solid 3D representation, the point cloud is typically processed further to reconstruct surfaces. This can involve techniques such as:

Meshing: Connecting neighboring points in the point cloud to create triangles, resulting in a mesh representation of the surfaces.

Implicit Surface Reconstruction: Generating a continuous surface representation from the point cloud using mathematical functions or algorithms like Poisson surface reconstruction.



volumetric grid and estimating surface properties f Texture Mapping: Once the 3D model is reconstructed, the texture from the original 2D image can be mapped onto the surfaces of the model to give it realistic appearance. This involves aligning the texture coordinates from the 2D image with the vertices of the 3D mesh.

rom the occupa ncy or

Texture mapping for reconstructed objects

**REFERENCES**

* <https://www.alpha3d.io/>
* <https://gemoo.com/tools/2d-to-3d-image-converter/>
* <https://imagetostl.com/>
* <https://www.leiapix.com/>

**CONCLUSION**

In summary, transforming 2D images into 3D models is a groundbreaking endeavor that revolutionizes visualization and comprehension. With cutting-edge techniques like structure-from-motion and deep learning driving progress in computer vision, this conversion achieves unprecedented levels of realism and precision. The impact spans numerous fields, spanning from immersive environments to scientific visualization and beyond, promising boundless potential for innovation and exploration.

