FrameToVolume
2D-to-3D Image-to-Gif Conversion

Name: Dhivyashree Siva Prakasam

UBIT: dsivapra – 50560688

Project Overview:

This project is all about transforming a single 2D photograph into rich 3D representations. 3D visualization like point clouds, meshes and animation like GIFs are generated from ordinary images that user gives. As a user you can input any image and this project outputs a depth map, a coloured point cloud, a reconstructed mesh, and a parallax-view GIF. The state of the art monocular depth estimation models are used especially transformer based models that demonstrates robust depth prediction from single images. These state of the art models such as MiDAS, DepthAnything and ZeoDepth that are trained on millions of images and capture fine geometry without requiring multiple views. Hence, our system inputs one image and produces multiple 3D outputs (depth map, point cloud, mesh, GIFs) using contemporary single image depth estimation techniques.

Approach:

In this project, I have implemented the pipeline in Python and related libraries. The three recent monocular depth models MiDaS DPT-Hybrid, Depth Anything V2 (Small) and ZoeDepth N were selected and used for the 3D conversion. For MiDaS (DPT-Hybrid) the Hugging Face Transformers implementation (Intel/dpt-hybrid-midas) is used. I loaded it by using the Hugging Face pipeline and the standard preprocessing is done as documented in its examples. Depth Anything V2 (Small) was obtained from its GitHub/Hugging Face repository. I used it through the Python API and ran inference on each image. For ZoeDepth, I specifically used the ZoeD_N model (fine-tuned on indoor NYU-Depth data).

After loading each model, I obtained the raw depth outputs for the image that is given as the input. Then the simple depth normalization is done so that the metrics and comparisons will be uniform for all modelsand the images. The depth output resolution is also aligned with the original image size if any model subsampled internally.

The 3D geometry is generated. From each normalized depth map the 3D point cloud is built. I used Open3D for this. The Poisson surface reconstruction is applied on the point cloud that was generated in the previous step. Then a mesh is generated and also the Poisson reconstruction on the point cloud smooths and fills surfaces from noisy depth points. The colourful point cloud was generated using the colours that are in the original image given. For animated output, a short parallax GIF is created by virtually shifting the camera position left to right and compositing the reprojected image from the depth map. To create the animation, I have implemented a custom module using Open3D and imageio that processes the point cloud and converts it into a mesh. The result is a compelling side-by-side comparison of the reconstructed 3D scene quality from each model.

Experimental Protocol:

The pipeline is evaluated on six different images that covers different scenes like shadow, outdoor, low light, etc. All three models were ran for each image and the depth maps are produced and saved as map_midas, map_depth_anything and map_zoe_depth. Then the metrics are calculated on each depth map. The Laplacian Variance and the edge sharpness along with the Depth Range are calculated. Then the point cloud, mesh and the 3D GIFs were generated for each image and stored.

The processing was done in VSCode with AMD Radeon GPU and CPU. So the standard that is followed is that each image was loaded, depth was inferred by each model, normalized, and then used to compute metrics and build the 3D outputs.

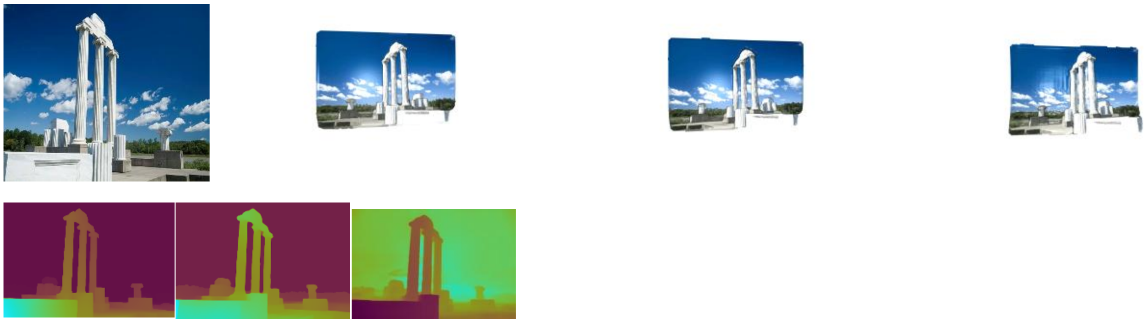
Results:

The results are the GIFs generated for all the six images. A detailed per image comparison of the three depth models is shown below (except point clouds and meshs which are unable to be attached) and are in the format:

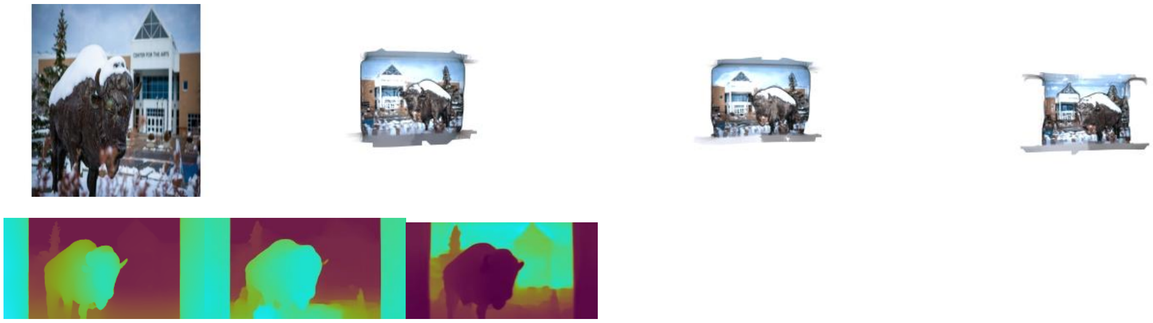
INPUT IMAGE :: DEPTH_ANYTHING_GIF :: MiDAS_GIF :: ZOE_DEPTH_GIF

DEPTH_ANYTHING_DEPTH_MAP :: MiDAS_DEPTH_MAP :: ZOE_DEPTH_DEPTH_MAP

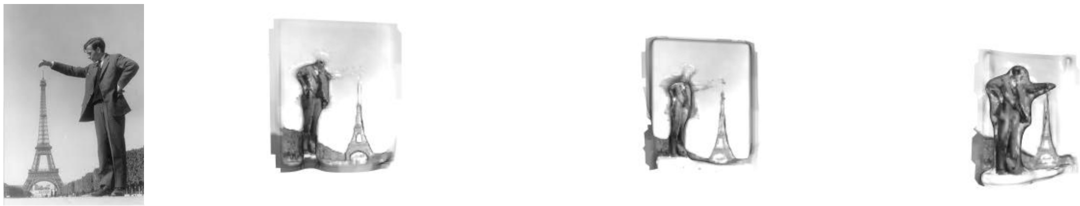
Bairdpoint.jpg (outdoor image, well lighted)

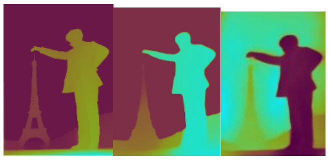


Bull.png (have more depth image)

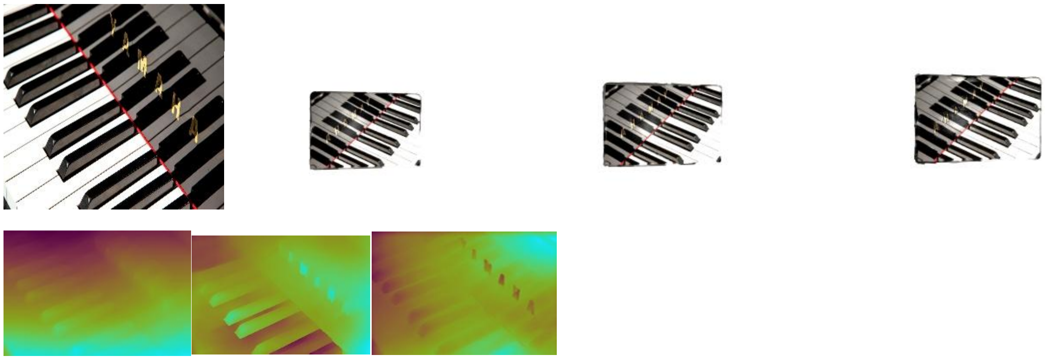


Paris.jpeg (black & white, depth image)

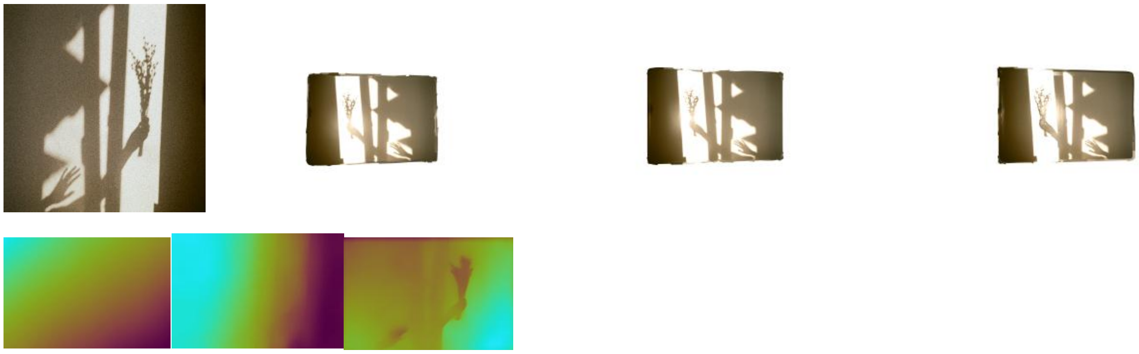




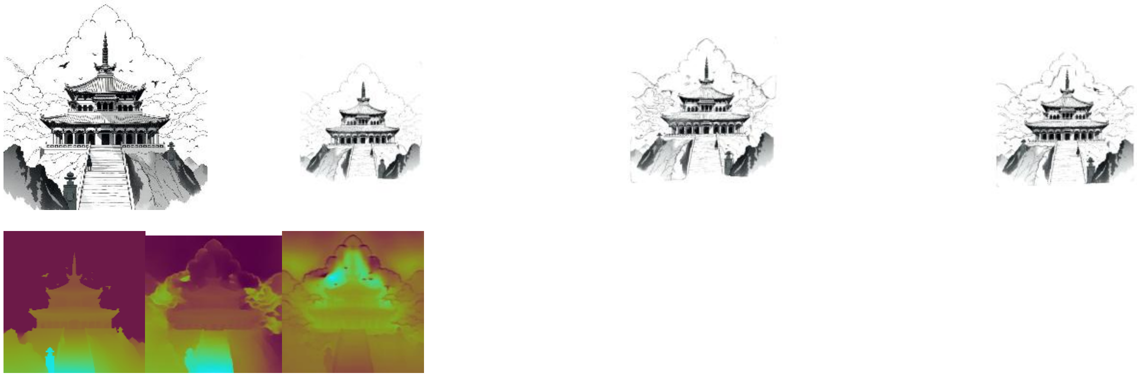
Piano.jpeg (easily confused based on reflection)



Shadow.jpeg (indoor, shadow)

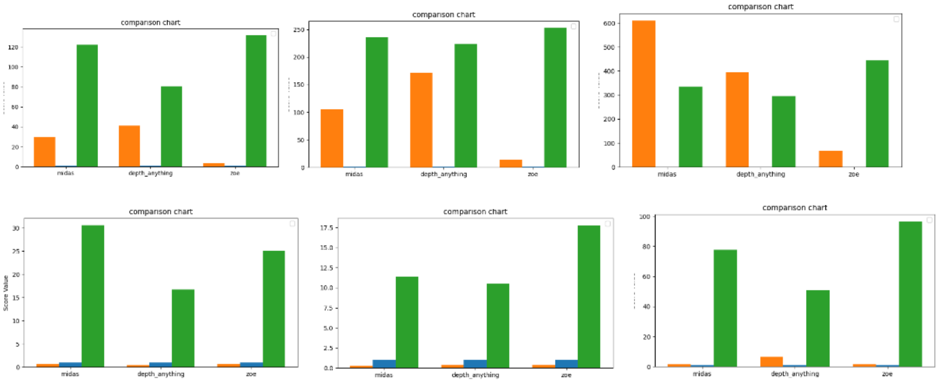


Temple.webp (symmetric, art)



The parallex gif, point clouds and mesh results are also generated for each image and each model. The different images make the models to work differently on different models.

Analysis:



Comparison Plots of Bairdpoint, Bull, Paris, Piano, Shadow, Temple

For the first 3 images (Bairdpoint, Bull, Paris)the results both Laplacian variance and edge sharpness are good when compared to the other 3 images(Piano, Shadow, Temple). The first 3 images are outdoor images and thus gives better results than indoor images.

In bairdpoint image we can see depthanything had high LV while the ZoeDepth has the sharpest edges but with minimal texture detail. MiDAS has average LV and sharpness and thus good with rugged llandscapes.

In bull image DepthAnything woks well both LV and sharpness. In this also ZoeDepth lacks in texture details while having high sharpness. MiDAS is again average player.

In paris image because of the architectural image, the LV of MiDAS is very high with a good edge sharpness. Comparitively, Depth Anything is also good but ZoeDepth is again missing the detailed texture of the image in the depth map.

In piano image the LV is very low for all the 3 models while MiDAS performs well in edge sharpness and because of low contrast DepthAnything and ZoeDepth are not able to perform well.

The shadow image is where all models fail drastically. It is clearly visible that the models are not performing well when it comes to shadow and ZoeDepth is having sharpness a little more when compared to others.

Since the temple image is an art and symmetric zeoDepth again leads in sharpness while all have very low LV.

The models get confused when there is low contrast, low light or complex illumination.

So, for rich texture scenes MiDAS has maximum LV followed by DepthAnything. The sharp silhouttes are for ZoeDepths which is best in edge definition. MiDAS and DepthAnything are slightly more generalized and consistent while ZoeDepth excels in edge sharpness.

Discussion and Lessons Learned:

Creating this project helped me understand the depth estimation, point clouds, mesh and practical usage of these. I have learnt about these models and created curiosity in working on this. I have learnt to use Blender and Open3D for this project.

For future extension we can do a custom model for depth extimation and integrating more advanced rendering. Also tune so that models perform well in most complex illuminations also.

Bibliography:

Ranftl et al. (2021) – “Vision Transformers for Dense Prediction” (ICCV 2021). Introduced the DPT (Dense Prediction Transformer) architecture used by MiDaS

Bhat et al. (2023) – “ZoeDepth: Zero-shot Transfer by Combining Relative and Metric Depth” (CVPR 2023). Describes the ZoeDepth models (ZoeD-N, etc.)

Yang et al. (2024) – “Depth Anything V2: A More Capable Foundation Model for Monocular Depth Estimation” (NeurIPS 2024, arXiv:2406.09414)

Zhou et al. (2018) – “Open3D: A Modern Library for 3D Data Processing” (CoRR abs/1801.09847)