Converting 2D Images to 3D Anaglyphs using Deep Learning Models

VARICH BOONSANONG and IVY DING, University of Washington

***Abstract* – Technological innovation in the realm of virtual reality (VR) has progressed significantly with devices as ubiquitous as smartphones capable of showing 3D content. Since the advent of camera technology, we have over a century of 2D media material at our disposal, which we believe has untapped potential for future 3D-generated content. Our project focuses on leveraging the advancement in Depth Estimation Deep Learning Models to approximate the depth of an object(s) in a 2D image in order to convert it into a 3D anaglyph.**

1 INTRODUCTION

Research into depth estimation primarily focuses on applications in self-driving technology and robotics, typically requiring the use of multiple specialized cameras to generate a 3D reconstruction of a particular scene. Our goal is to inexpensively estimate depth and generate 3D content from 2D source material. As such, our use cases are limited to generating image/video anaglyphs or image/video files compatible with 3D displays.

There are multiple related works in this area of 2D to 3D depth estimation, which we can separate into two categories: (1) using classical computer vision algorithms to interpolate depth, and (2) using neural networks. Our work falls under the second category. For example, [Kha20]'s work focused on generating 3D models of shoes based on a 2D side-view image. [Alc18]'s work is particularly relevant in that it focused on generating a 3D reconstruction of human faces from 2D images. More recently, research into using pre-trained models, such as [Cal23]'s stable diffusion 2D to 3D video synthesis, has become more popular for generating the depth map.

It's worth noting that most of these projects have not implemented the software layer to support VR. Only [Xie16]'s 2D to 3D video conversion with CNN's project supports using anaglyphs and VR, but it used inferior neural networks and an old dataset. Our project can be best compared to Google's Starline project although it is much cheaper to execute and requires less specialized equipment and resources. We seek to expand upon this new area in 3D reconstruction and support 3D real-time video conversion using anaglyphs and VR.

1.1 Contributions

We developed a fully automatic pipeline for generating 3D anaglyphs from 2D image inputs.

- We have shown that we can use an inexpensive 2D to 3D conversion pipeline to provide an immersive 3D experience using commodity hardware and off-the-shelf anaglyph glasses.
- We have further improved classical stereo rendering algorithms with anaglyphs by accelerating the process on GPU.

Authors' address: Varich Boonsanong, varicb@cs.washington.edu; Ivy Ding, iding@cs.washington.edu, University of Washington.

2 RELATED WORK

Past works mentioned earlier usually revolve around converting a particular object, such as a shoe or human faces, into 3D. Our project, however, is intended to be more general purpose as well as branching into real-time videos.

We were inspired by [Law+21] Google's Starline project which involves ground-breaking research in computer vision, machine learning, and real-time compression techniques. There is a significant hardware aspect to it as well that includes both visual and audio aspects, although we will focus on the visual technology only. Currently, Starline operates in a 3D video chat booth that contains "capture pods" which capture both color and depth data to output three depth maps. Four additional tracking cameras are used to generate four color perspectives, combining with the depth maps for a total of seven video streams. Four high-end Nvidia GPUs are used to process this data. Three depth maps from each of the left and right eye are rendered through their novel "image-based fusion" raycasting algorithm. The four color texture streams are projected onto this fused surface and blended using normal-based texture blending. The resulting left and right images are displayed on their special 3D light display screen for an optimal 3D render.

We are limited in the amount of hardware and computational power we can use and thus we will be using pre-trained models and simpler but adequate stereo rendering algorithms to generate our left and right images.

3 METHOD

Our method can be used on commodity hardware, only requiring the use of a web camera, single GPU, and anaglyph glasses. As such, we will be focusing heavily on the software aspect. Although we were not able to access a 3D display, our research can still be applicable for its use, namely that a 3D display simplifies the pipeline by essentially doing the post-processing step for us. A diagram of the full process is illustrated in Figure 7.

3.1 Monocular Depth Estimation

Monocular depth estimation is the process of estimating depth values for each pixel given a single (monocular) RGB image. Our core process for estimating these depth values involves a Dense Prediction Transformer (DPT) model by [RBK21] which takes in a 2D RGB image and outputs a grayscale (1D) depth map. DPT is a new architecture that leverages a Vision Transformer (ViT) instead of CNNs which has been shown to improve performance and quality of output. It has primarily been used in autonomous vehicles and has been trained on 1.4 million images produced from self-driving car radars. As such, the data it has been trained on consists mainly of images taken outdoors, although we believe it will generalize well enough on human-focused image inputs without too much noise or estimation error.

Fig. 1. Sample image for illustrative purposes.

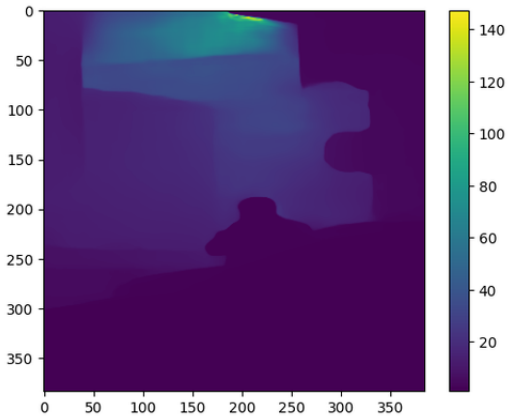

Fig. 2. A depth map outputted as a 1-channel image. The image does not necessarily need to be grayscale.

A depth map for the original image presented above (Figure 1) is seen in Figure 2. Essentially, a depth map is all that is needed to render a full 3D anaglyph, however we go further by allowing for selective 3D effects.

3.2 Image Segmentation

For selective 3D anaglyph images, where only a particular object(s) is given a 3D effect, we employ image segmentation to determine the object(s) of focus. We use the Masked-attention Mask Transformer (Mask2Former) by [Che+21] that takes in a 2D image and outputs a segmentation map. Current research focuses on developing models specialized for specific tasks, however, the Mask2Former model generalizes well to any image segmentation task. A segmentation map of the original image is seen in Figure 3. This is done separately from the DPT model, however, both outputs then serve as inputs for the post-processing step.

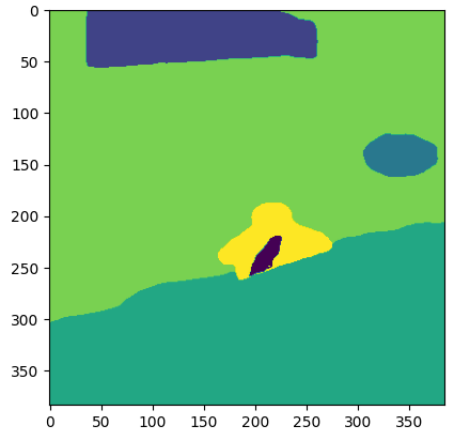

Fig. 3. A segmentation map outputted as a 1-channel image. We manually mapped different values into different colors to visually show the classes as represented by the segmentation map.

3.3 Anaglyph Formation

Anaglyphs are a common and inexpensive way of generating 3D images. We generate two additional images from each left and right eye to superimpose onto the original image, creating the resulting 3D anaglyph. The left and right eye images are generated by shifting every pixel by a specific amount determined by the normalized depth map. In general, the left and right images will have a negative horizontal parallax. This is especially relevant when discussing selective anaglyphs.

We use a classical anaglyph algorithm to determine how much to shift the pixels of an image [Cal23]. In addition to the original image and depth map as inputs, we also take in a divergence parameter to adjust the degree to which we want a 3D effect. Let D be the divergence parameter where $D = \pm 0.025w$ where w is the width of the 2D input image. D is negative for the left image while positive for the right image. We define the coefficient 0.025 to be the measure of the 3D effect (i.e. being 2.5% of the width of the image) as it provided the best results. For a pixel, the resulting offset is then:

$$\text{offset} = \lfloor (1 - n^2) * D \rfloor \tag{1}$$

where n is the depth value associated with that pixel in the normalized depth map. Pixels that appear farther away (have depth values close to 1.0) would have little to no offset, making them appear flat and further away. The intended effect is to have higher offsets for objects that appear closer to us. In the left/right image, we shift each pixel to the right/left by the offset calculated. Once the left and right images are created, we superimpose them onto the original to create a full-image anaglyph. An example result can be seen in Figure 4

With a segmentation map, we can adjust the values in the depth map so that the object of interest is "brought closer" while everything else appears further away. This effect compounds when calculating the offsets as the pixels associated with the object(s) of focus are the only ones that appear "shifted." This alternative implementation

is used for generating selective anaglyphs. An example result can be seen in Figure 5.

Fig. 4. Full image anaglyph using only the depth map.

Fig. 5. Selective anaglyph on the person only with the depth map and segmentation map.

3.4 Real-time Videos

So far we have discussed the pipeline for a single 2D image input. As videos simply consist of numerous images strung together at a high frequency, we can extrapolate the process to be deployed in real-time video use. This involves rendering each frame of the video into an anaglyph. When using a webcam, the rendering is done between each consecutive image captured by the camera. Due to the relatively high computational power needed, we experience high latency issues which we discuss more in depth in Section 6.

3.5 3D Displays

Ideally we want a 3D display larger than 50 inches as larger displays would give a more immersive 3D experience, although any 3D display would work. A 3D display would perform the post-processing step for us, only needing to take the original image and depth map as inputs. A segmentation map can also be inputted to selectively choose an object(s) to be 3D (i.e. a person).

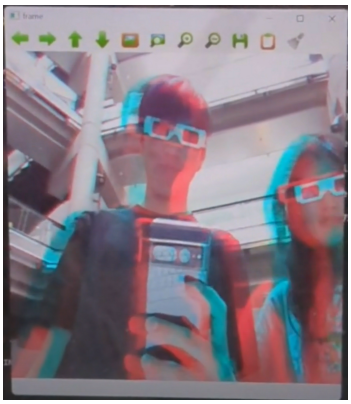

Fig. 6. A screenshot of our video anaglyph in action.

4 IMPLEMENTATION DETAILS

We only used commodity hardware and off-the-shelf anaglyph glasses for our project and presentation. For our hardware, we used a laptop with the 12th Gen Intel(R) Core(TM) i7-12700H CPU running at 3.30 GHz and an NVIDIA GeForce RTX 3050 Ti Laptop GPU. Both models were adjusted to run on the GPU to speed up performance for real-time videos while the pre- and post-processing steps were run on the CPU.

4.1 Software

Our pipeline uses two deep learning models: (1) the DPT model by [RBK21], and (2) the Mask2Former model by [Che+21]. In our pre-processing step, we simply resized the input RGB images to be 250x250 pixels before they are further pre-processed by the models mentioned above. The output of the DPT model is a depth map: a 1 channel image (250x250) where the value of each pixel represents the approximate depth. The output of the Mask2Former model is a segmentation map: a 1 channel image (also 250x250) where pixels classified into the same segment contain the same value. These two maps are then used as inputs to our post-processing step.

In our post-processing step, we normalize the depth map and then use the classical anaglyph algorithm from section 3.3 to calculate the offsets for each pixel for the left and right images. More specifically, we replace the offset pixels in the left and right images with the RGB value of the pixel from the original image such that they appear "shifted" from the original image. Once we have the left and right images, we take the original RGB image and replace the R values with the R values of the left image and the G and B values with the G and B values of the right image. The composite image (250x250) created is our resulting full-image anaglyph.

We can optionally use the segmentation map to create selective anaglyphs, which is used when deploying our method on our web-cam for real-time videos. Our post-processing step can be adjusted to accommodate this additional input. Instead of shifting every pixel of the image, we only shift the pixels associated with a particular segment, as determined by the segmentation map. For example, pixels only associated with a "person" segment would have their corresponding pixels in the depth map "brought forward." In our

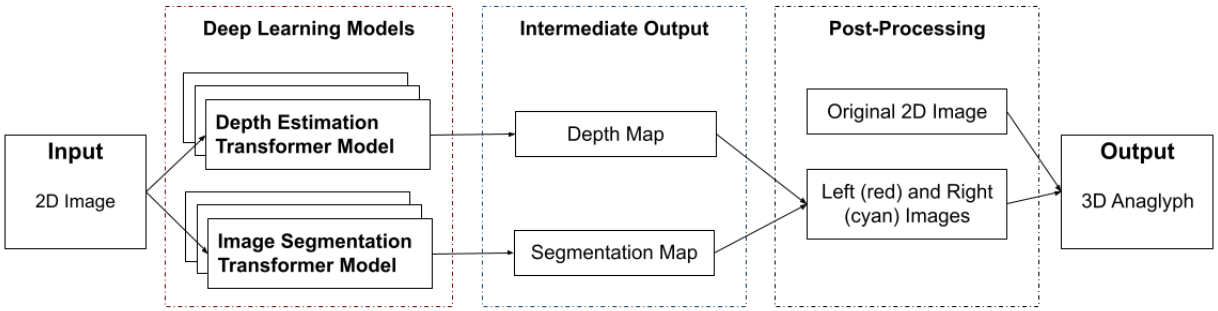

Fig. 7. Overview of the components and data flow in our system. In real-time videos, the entire pipeline is computed per frame.

implementation, we decreased the depth of pixels associated with the "person" segment by 20 meters while increasing the depth of all other pixels by 1000 meters (or essentially infinity) before normalizing the depth map. This results in the red and cyan shifts only appearing for the person while the rest of the image is in 2D.

5 EVALUATION OF RESULTS

Given that there is no preestablished benchmark to assess the performance of our method as it's dependent on the specific hardware used, we want to dedicate this section to comparing the performance improvement that we made from running the models on GPU instead of the CPU. When running on the CPU, the process takes 7.134 seconds per frame render. On the other hand, with the pipeline partially utilizing the GPU, we reduce the time down to 0.5 seconds per frame.

6 BENEFITS AND LIMITATIONS

Given that we use anaglyphs as the medium to convey a 3D image, there is a limitation to the amount of parallax that it can create. It also deteriorates the image's color quality and requires the user to wear anaglyph glasses at all times. The color effect can be mitigated by using a VR headset, although it is definitely more cumbersome to wear than anaglyph glasses. On the one hand, the issue of wearing glasses can be resolved by using a 3D display instead.

Moreover, we can use knowledge distillation to reduce the classification of the segmentation model to just be either human or not human instead of the current output of 1000+ possible classes. With such a process, we will have smaller models that are more efficient to compute. Another aspect of our project that can be improved upon is to convert the model to ONNX browser format and test the feasibility of running the whole pipeline through the browser as a web application.

7 FUTURE WORK

Currently, our whole pipeline uses 250 x 250 image resolutions which is a relatively low screen resolution. Future work can consider adding image super-resolution models to our pipeline to upscale the image back to a particular display size. Of course, the computational limitation is still a factor to consider. As such, we can also consider investigating ways to speed up performance, such as delegating computation to the cloud and letting users transmit only the compressed embedded representation of the original images to the cloud server for processing. This would significantly improve the experience, especially on mobile devices.

8 CONCLUSION

In this paper, we delineated our methodology for generating selective 3D anaglyphs from 2D source material and applied it to real-time videos. We accomplished a practical pipeline of applying anaglyphs to webcam usage using only commodity hardware, entailing future potential for cheaper and more widespread applications in immersive 3D experiences.

REFERENCES

[Alc18] Marc Alcaraz. *MARCALCARAZF/realtime-2d-to-3d-faces: Reconstructing real-time 3D faces from 2D images using deep learning*. 2018. URL: https://github.com/marcalcarazf/realtime-2D-to-3D-faces.

[Cal23] Michael Callahan. *Mcallahan/stable-diffusion-2d-to-3d-video-synthesis: Automatic conversion of 2D video into 3D video by using stable diffusion depth map generation to create the secondary images*. 2023. URL: https://github.com/mcallahan/stable-diffusion-2D-to-3D-video-synthesis.

[Che+21] Bowen Cheng et al. "Masked-attention Mask Transformer for Universal Image Segmentation". In: *CoRR* abs/2112.01527 (2021). arXiv: 2112.01527. URL: https://arxiv.org/abs/2112.01527.

[Kha20] Nasir Khalid. *Nasirkhalid24/2Dto3D-shoes: Single View Shoe to 3D model using neural networks + Nvidia kaolin*. 2020. URL: https://github.com/NasirKhalid24/2Dto3D-Shoes.

[Law+21] Jason Lawrence et al. "Project Starline: A High-Fidelity Telepresence System". In: *ACM Trans. Graph.* 40.6 (Dec. 2021). ISSN: 0730-0301. DOI: 10.1145/3478513.3480490. URL: https://doi.org/10.1145/3478513.3480490.

[RBK21] René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. "Vision Transformers for Dense Prediction". In: *CoRR* abs/2103.13413 (2021). arXiv: 2103.13413. URL: https://arxiv.org/abs/2103.13413.

[Xie16] Eric Xie. *Piiswrong/deep3d: Automatic 2D-to-3d video conversion with cnns*. 2016. URL: https://github.com/piiswrong/deep3d.