2D-to-3D and Parallax Scrolling :
A Depth-Based Multimodal Image Layering System

Rong-Lin Jian, Yu-Zhe Pien, Fang-Ying Lin

College of Artificial Intelligence
National Yang Ming Chiao Tung University

***Abstract* - This project presents a depth-based multimodal image layering system that generates dynamic, depth-aware visual effects from a single 2D image. The system consists of two specialized modules: a 2D-to-3D pop-out rendering module and a parallax scrolling module. Semantic object segmentation is performed using SAM3 to obtain accurate foreground masks, while depth information is estimated using Depth-Anything. For 2D-to-3D rendering, foreground objects are divided into 100 depth layers using percentile-based slicing, and transparency-controlled RGBA composition is applied to achieve smooth pseudo-3D motion. Premultiplied alpha blending and a reference support layer are introduced to reduce edge artifacts and occlusion gaps. For parallax scrolling, depth-guided foreground, midground, and background layers are refined through morphological processing to ensure stable motion. Experimental results demonstrate convincing pseudo-3D effects and consistent depth perception from single images. The code is available at https://github.com/fanglin02/MMIP-Final-Project_Group10.git**

Index Terms - Multimodal Image, Parallax Scrolling, Semantic Segmentation, Depth Estimation, Image Layering

I. Introduction

The motivation for this project is inspired by two classic visual representations: side-scrolling game maps and papercraft. In side-scrolling games, maps consist of foreground, midground, and background layers moving at different speeds to produce parallax effects, giving the illusion of depth in a 2D environment. Papercraft, on the other hand, uses stacked layers with partial occlusion to create 3D structure and spatial depth even in static images. Both approaches rely on image layering and depth ordering to convey 3D effects. This project aims to explore the automatic extraction of semantic layers and depth information from a single 2D image to produce dynamic, depth-aware visual effects through two distinct modules: a 2D-to-3D module for object-level pop-out rendering and a parallax scrolling module for scene-level motion synthesis.

A. Repository and Setup

1) Repository URL: https://github.com/fanglin02/MMIP-Final-Project_Group10.git

2) How to Run:

a) Install required dependencies: pip install -r requirements.txt

b) Run the 2D-to-3D module: python 2dto3d.py

c) Run the Parallax Scrolling module: python parallax_scrolling.py

3) Requirements:

a) Python 3.10 or higher

b) opencv-python

c) Depth-Anything-3

d) sam3

e) Additional packages as listed in requirements.txt

II. Methodology

A. 2D-to-3D Pop-Out Rendering

The 2D-to-3D pop-out rendering module converts a single 2D image into a layered pseudo-3D representation by extracting semantic and depth information from the foreground objects. The workflow consists of five key steps: semantic segmentation, foreground extraction, depth estimation, layer generation, and animation composition. Images illustrating each step are shown in Fig. 1.

1) Semantic Segmentation: The input image is processed using the SAM3 model guided by text prompts to generate precise object masks. Each mask is overlaid on the original image to distinguish foreground elements from the background, providing the structural foundation for subsequent processing. While initial attempts using DINOv3 semantic maps to segment the scene into foreground, midground, and background yielded unsatisfactory results due to coarse boundaries, the adoption of SAM3 provided the high-fidelity segmentation required for clean object isolation. Images are shown in Fig. 1(b).

2) Foreground Extraction: Using the generated masks, foreground objects are cropped to create RGBA images with transparency. This step separates the foreground from the background, ensuring that depth estimation and layer generation are not affected by background pixels. Images are shown in Fig. 1(c).

3) Depth Estimation: The isolated RGBA foreground is processed via the Depth-Anything model. To prevent the model from assigning extreme or erroneous depth values to transparent regions, these areas are temporarily filled with neutral gray. After the model outputs a grayscale depth map, the original alpha channel is reapplied, and depth normalization is performed exclusively on the foreground pixels. Images are shown in Fig. 1(d).

4) Layer Generation: Foreground objects are divided into 100 discrete layers using percentile-based slicing. Unlike linear depth intervals, this method ensures that pixels are distributed equally across layers, maintaining stable visual density. Each layer retains the original RGBA data, but visibility is constrained to a specific depth range. By using alpha transparency to define these boundaries, this ensures smooth transitions between layers and prevents visual cracks or discontinuities at object edges.

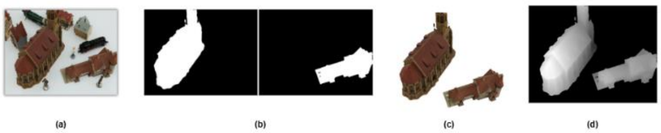

Fig. 1: 2D-to-3D pop-out rendering workflow. (a) Original input image, (b) Semantic segmentation, (c) Foreground extraction, (d) Depth estimation.

5) Animation Composition: Each foreground layer is translated horizontally according to its depth to produce a pseudo-3D parallax effect, with closer layers moving faster than distant layers. During initial compositing, edge artifacts such as white seams may appear due to improper handling of semi-transparent pixels. If RGB values are directly overlaid without accounting for alpha, semi-transparent regions are treated as fully opaque, causing bright edges. To address this, premultiplied alpha blending is applied in (1):

$$RGB_{premul} = RGB \times \alpha \tag{1}$$

where RGB represents the original color values and α is the alpha channel. Each pixel's color is multiplied by its transparency before translation and compositing, preventing the bright-edge effect. Additionally, a complete object reference layer is placed beneath all slices to fill small gaps and ensure consistent edge coverage. This approach produces smooth, depth-aware motion while minimizing visual artifacts in the final animation.

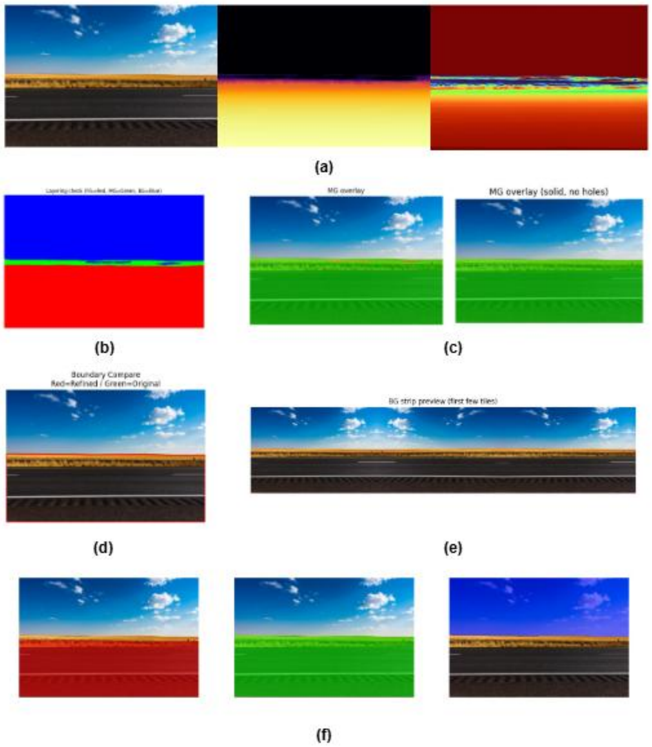
(a)
Layering check (FG=Red, MG=Green, BG=Blue)
(b)
MG overlay
MG overlay (solid, no holes)
(c)
Boundary Campare
Red=Refined / Green=Original
(d)
BG strip preview (first few tiles)
(e)
(f)

Fig. 2: Parallax scrolling workflow. (a) Input RGB image, depth map, and depth confidence heatmap, (b) Depth-based layer segmentation, (c) Closing to fill small holes, (d) Corrected midground boundaries, (e) Seamless mirrored strips, (f) Final RGBA overlays for foreground, midground, and background.

B. *Parallax Scrolling*

The parallax scrolling module generates depth-aware scene animations from a single 2D image by leveraging multimodal depth information. The processing pipeline consists of five main steps: depth generation, depth-based layer segmentation, seamless tiling, visual enhancement, and video output. Images illustrating each step are shown in Fig. 2.

1) Depth Generation: Since a single RGB image lacks explicit spatial coordinates, we utilize Depth-Anything-V3 to generate a high-density depth map. To enhance reliability, the system simultaneously computes a Depth Confidence Heatmap. These three modalities (RGB, Depth, and Confidence) are aligned in scale to provide a robust foundation for layering, effectively preventing misclassification of foreground and background elements. The three modalities (RGB, Depth, and Confidence) images are shown in Fig. 2(a).

2) Depth-Based Layer Segmentation: Depth maps often contain noise, making it difficult to separate foreground (FG), midground (MG), and background (BG). Percentile-based thresholds are applied to define depth boundaries in (2):

$$t_{\text{bg}} = \text{Percentile}(D, 5), \;\; t_{\text{fg}} = \text{Percentile}(D, 53) \tag{2}$$

The midground mask is defined as in (3):

$$mask_{MG2} = mask_{MG} \cup mask_{FG} \tag{3}$$

Layers are color-coded as red for foreground, green for midground, and blue for background. The mask overlay is shown in Fig. 2(b). Morphological processing, including dilation and erosion (closing), is applied to eliminate small holes and ensure stable RGBA layers. The results of morphological closing are shown in Fig. 2(c). Overlapping foreground pixels are removed from the midground layer to prevent occlusion conflicts. A combined support layer of FG and MG is created to provide structural integrity. To refine the midground boundaries, Canny edge detection is applied to extract object contours. The mask is divided into known foreground, known background, and unknown regions. The Watershed algorithm is then used to expand the unknown regions, aligning the midground mask with actual object edges. The refined midground boundaries are presented in Fig. 2(d). Corrected boundaries are highlighted in red, while original boundaries are shown in green.

3) Seamless Tiling: To produce continuous horizontal motion, each layer is mirrored and concatenated to create seamless long strips. Layer-specific scrolling speeds are set according to depth: background layers move slowly, midground layers move at medium speed, and foreground layers move fastest. RGBA layers are composited using alpha blending to maintain visual consistency. Images are shown in Fig. 2(e).

4) Visual Enhancement: To enhance realism and motion perception, horizontal motion blur is applied to foreground layers to simulate rapid movement, overall lighting is smoothed to mimic environmental illumination changes, and subtle camera-like shaking is added to emulate terrain undulations.

5) Video Output: The processed layers are combined to generate the final parallax video, with separate overlays stored for foreground, midground, and background. Images are shown in Fig. 2(f). This approach produces depth-consistent, dynamic scene animations akin to side-scrolling game backgrounds.

III. Results and Ablation Study

A. Experimental Results

The proposed system was evaluated using diverse single 2D images containing various foreground objects and environmental scenes. Since the outputs of both the 2D-to-3D Pop-Out Rendering and Parallax Scrolling modules are dynamic animations, experimental results are provided as video demonstrations hosted on the project's GitHub repository for full visual inspection. The generated videos demonstrate that the system effectively synthesizes depth-aware motion from static 2D inputs. Both modules produce smooth transitions and maintain consistent spatial layering.

B. Ablation Study

To evaluate the contributions of key components in the parallax scrolling module, we conducted an ablation study focusing on two factors: seamless tiling and support layer integration.

1) Seamless Tiling: Layered images are mirrored and concatenated to create continuous horizontal motion. When seamless tiling is applied, foreground, midground, and background layers move continuously without visible discontinuities. Without tiling, horizontal edges appear misaligned, resulting in noticeable jumps or breaks in the animation.

2) Support Layer (Foreground + Midground Merging): To maintain structural integrity, the midground and foreground layers are combined into a support layer. With this layer, small holes or gaps in the midground are filled, ensuring smooth occlusion transitions. When omitted, midground layers exhibit large gaps where they overlap with the foreground, causing visual artifacts and broken continuity.

Due to the dynamic nature of the outputs, these effects are best observed in the final videos, which are available in the project repository. Both ablation conditions can be compared through the demonstration videos for direct visual inspection.

IV. Conclusion and Future Work

This project presents a depth-based multimodal image layering system capable of generating dynamic, depth-aware visual effects from single 2D images. The system consists of two specialized modules: a 2D-to-3D Pop-Out Rendering module for object-level pseudo-3D effects and a Parallax Scrolling module for scene-level motion synthesis. Experimental results demonstrate that the system produces smooth, visually convincing animations, while the ablation study highlights the critical roles of seamless tiling and support layer integration in maintaining visual continuity and structural integrity.

Future work includes improving depth estimation for highly complex or cluttered scenes, extending the system to handle video inputs directly, and exploring real-time implementation for interactive applications. Additionally, enhancing semantic segmentation to better separate fine details and investigating alternative compositing strategies may further reduce residual artifacts and improve the overall realism of generated animations.

Acknowledgment

We would like to express our sincere gratitude to our course instructor, Chih-Chung Hsu, for the insightful teaching and continuous guidance provided throughout the semester. The concepts and techniques taught in this course were fundamental to the successful implementation of this project. We also thank our teammates for their dedicated collaboration, effective communication, and joint efforts in the development and testing of both modules. Finally, we acknowledge the developers of the open-source tools and pre-trained models, specifically SAM3 and Depth-Anything, which served as the core technological pillars of this work.

References

[1] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, 4th ed., Hoboken, NJ, USA: Pearson, 2018.

[2] Nicolas Carion, Laura Gustafson, Yuan-Ting Hu, Shoubhik Debnath, Ronghang Hu, Didac Suris, Chaitanya Ryali, Kalyan Vasudev Alwala, Haitham Khedr, Andrew Huang, et al. Sam 3: Segment anything with concepts. *arXiv preprint arXiv:*2511.16719, 2025.

[3] Haotong Lin, Sili Chen, Junhao Liew, Donny Y Chen, Zhenyu Li, Guang Shi, Jiashi Feng, and Bingyi Kang. Depth anything 3: Recovering the visual space from any views. *arXiv preprint arXiv:*2511.10647, 2025.