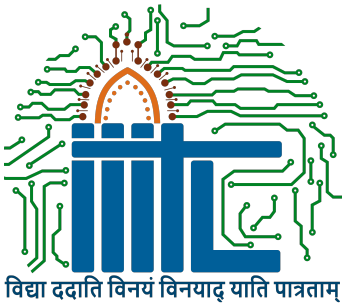
2D to 3D Video Conversion

M.Sc

Saurabh Kumar Singh
MSA23011



DEPARTMENT OF COMPUTER SCIENCE
INDIAN INSTITUTE OF INFORMATION TECHNOLOGY, LUCKNOW
2023 - 2025

2D to 3D Video Conversion

A thesis submitted in partial fulfillment of the requirements for the award of the degree of

M.Sc.

in

Artificial Intelligence and Machine Learning

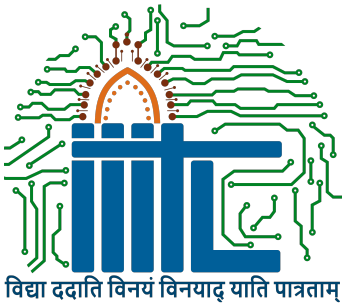
by

Saurabh Kumar Singh

(MSA23011)

under the guidance of

Dr. Soumendu Chakraborty



Indian Institute of Information Technology, Lucknow

2023-25

I dedicate this work to my family for their constant support and unconditional love, and to my respected professor, Dr. Soumendu Chakraborty, for his guidance and constant encouragement..

AUTHORSHIP

I, **Saurabh Kumar Singh**, declare that this thesis titled, "**2D to 3D Video Conversion**"and the work presented in it are my own. I confirm that:

- This work has been done completely while pursuing a Master of Science at the Indian Institute of Information Technology, Lucknow.
- All my references are cited wherever I found that I have referred to the work of otherauthors.
- In the description of texts by others, I have always indicated the source. Apart from the above references, this thesis is my own work.
- I have credited all of my main sources of information.

Signed:

_____

Date:

_____

CERTIFICATE

This is to certify that the thesis entitled "**2D to 3D Video Conversion**" submitted by **Saurabh Kumar Singh**, in partial fulfillment of the requirements for the award of the degree of M.Sc in **15 July,2023**, submitted to Indian Institute of Information Technology, Lucknow, has been carried out under my supervision and is the result of the own investigation except to the extent that the work reported in it has been conduced in formulation with his work. and The research was conducted at **Dr. Soumendu Chakraborty's** lab, Department of Computer Science, IndianInstitute of Information Technology, Lucknow - 226002, Uttar Pradesh, India. This thesis or parts thereof has not been previously submitted for any degree, diploma or other academic award in any other institution. has not been previously submitted for anydegree, diploma, or academic award anywhere before.

Dr. Soumendu Chakraborty
Department of Computer Science
Indian Institute of Information Technology, Lucknow
Pin - 226002, INDIA

Supervisor's Signature **External Expert**

Acknowledgements

I would like to express my sincere gratitude to my supervisor, Professor **Dr. Soumendu Chakraborty**, for their invaluable guidance, continuous support, and insightful feedback throughout my research journey. Their expertise in computer vision and depth estimation has been instrumental in shaping this thesis.

I am also deeply thankful to Akanksha from the **Research Group/Lab** for their technical assistance and constructive discussions that helped me overcome numerous challenges in implementing the depth map generation algorithms and 3D conversion techniques.

I would like to acknowledge the support from my fellow research colleagues at the Department of computer science who created a stimulating and collaborative environment.

Finally, I am profoundly grateful to my family and friends for their unwavering encouragement, patience, and emotional support throughout this academic endeavor

Lucknow
May 2025

Saurabh Kumar Singh

ABSTRACT

Today, there's a growing demand for 3D content, but creating videos in native 3D is still difficult and limited. That's why converting regular 2D videos into 3D has become an important technology. A key part of this process is estimating depth (how far objects are from the camera) from 2D video, which helps create realistic 3D scenes. However, traditional methods often struggle with complicated scenes, changing frames, and require a lot of processing power. In this thesis, we explore the use of advanced deep learning techniques to improve depth estimation for 2D-to-3D video conversion. We focus on popular models like **MiDaS**,**DPT-Large** and **DPT-Hybrid** and how they can be combined and optimized to get better results. Our goal is to solve problems like keeping object edges clear, maintaining smoothness between frames, and reducing the processing load. We built a complete system that takes 2D video frames, processes them, estimates depth, refines the results, and then generates 3D views. We tested several models and even developed new hybrid methods that mix the best features of different approaches. We also improved the final depth maps using special post-processing methods that are tuned for creating 3D content. We tested our method on many different types of videos, and the results showed that it gives better depth accuracy, clearer edges, and smoother transitions between frames than older methods. We also used **Google's Gemini AI** to check the quality of the depth maps and automatically improve the settings. Previously, the 2D-to-3D video conversion for a 15-second video used to take about 1 hour, but with Gemini, it now only takes about 3 minutes, **reducing** the processing time by around **95%**

Keywords: 2D - to - 3D conversion, depth map estimation

Contents

List of Tables

List of Algorithms

List of Figures

Chapter 1

Introduction

With recent advances in technology, 3D video technology has become possible and highly attractive for display systems. Many 3D products, such as 3D cameras, projectors, mobile devices, and displays, have emerged in consumer electronics. However, for most people, 3D displays remain very expensive and out of reach. This creates a strong demand for converting existing 2D videos into 3D so that users can experience 3D content using affordable alternatives like red-cyan anaglyph glasses, which are inexpensive and easily available in the market. Depth map estimation plays a crucial role in 2D-to-3D conversion. When depth information is captured using a dedicated 3D camera, it can be highly accurate. Unfortunately, most conventional 2D videos lack real depth data. Therefore, researchers have focused on developing methods to estimate depth maps that closely resemble real depth information. In our study, I use depth map estimation models like MiDaS, DPT Large, and DPT Hybrid. By combining their outputs, I generate more accurate and detailed depth maps. This improves the 3D video quality, ensuring a more realistic and comfortable viewing experience. The conversion process involves generating depth maps from the original 2D video, then using Depth-Image-Based Rendering (DIBR) to create left and right eye views. This stereo pair produces the 3D effect when viewed with red-cyan glasses. As a result, even without expensive 3D displays, users can enjoy immersive 3D experiences at home. whyand This hybrid technique is particularly valuable because it makes high-quality 3D video conversion accessible and affordable. Users no longer need to invest in expensive stereoscopic displays or 3D televisions. Instead, they can use standard screens along with red-cyan glasses, which are widely available and less budget-friendly.

In previous studies, several algorithms have been proposed for estimating depth maps from 2D video frames. For example, one such hybrid depth-generation algorithm uses motion information, linear perspective (vanishing lines or vanishing points), and luminance texture characteristics to create depth maps. In that approach, motion between adjacent frames helps separate foreground and background,

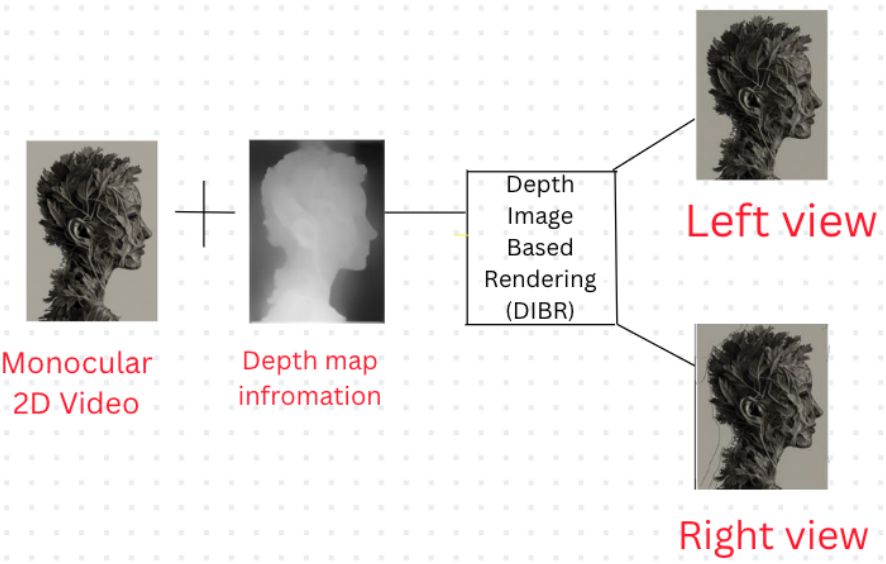


Figure 1.1: Data flow of 2D-to-3D video conversion

while vanishing points identify the farthest scene elements, and luminance differences provide additional depth cues. In contrast, our study does not use this traditional algorithm. Instead, I focus on a deep learning-based approach that leverages pretrained models: MiDaS, DPT Large, and DPT Hybrid. These models are trained on large and diverse datasets, allowing them to predict depth information from a single 2D image or video frame with high accuracy. By combining their outputs, I generate more precise and realistic depth maps, resulting in an improved 3D video experience.The use of pretrained models has several advantages: it removes the need for manual depth cue analysis, ensures more consistent depth maps, and significantly reduces processing time. Our approach thus makes it feasible to convert 2D videos into 3D videos efficiently and with higher visual quality. This enables users to enjoy immersive 3D experiences without the need for expensive 3D displays—using just regular screens and affordable red-cyan glasses.

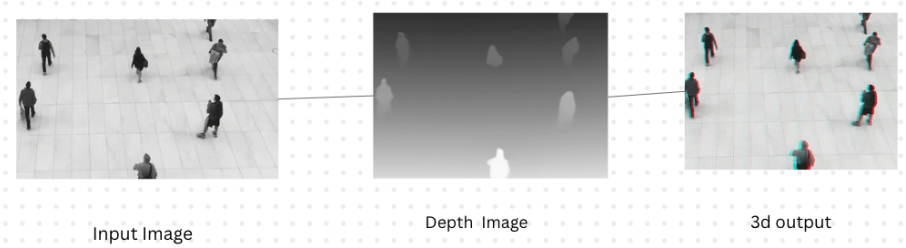


Figure 1.2: 2D-to-3D video

Chapter 2

Literature Review

Depth map creation is a fundamental aspect of 2D-to-3D video conversion, as it provides the essential spatial details needed to synthesize stereoscopic views for both eyes. Over the years, extensive research has been dedicated to methods of depth estimation from 2D input, often navigating the trade-off between computational demands and the perceptual quality of the resulting 3D content.

One prominent direction has been motion-based approaches, such as the H.264-based method[1] that derives depth estimates by analyzing motion vectors in video sequences. These methods excel in real-time applications but are generally less effective when dealing with static scenes or camera movements that complicate motion analysis.

Another set of approaches utilizes scene and edge features, like edges, textures, and perspective cues.[2] Lai et al., for instance, developed a hybrid algorithm that combines motion, perspective, and texture information to produce more realistic depth maps while still ensuring computational efficiency. These strategies improve the perceived depth but are often limited when the scene lacks structural cues.

Semi-automatic techniques have also been explored to incorporate human input, blending manual labeling with automated refinement.[3] Phan and Androutsos demonstrated how using Graph Cuts and Random Walks to propagate user-specified depth cues in video frames can improve both edge clarity and temporal coherence. However, this reliance on user input can make them labor-intensive and limit scalability.

A novel direction in depth estimation involves natural scene statistics (NSS)-driven approaches. [4] proposed a Bayesian framework that integrates NSS-based priors, derived from the statistical behavior of luminance and chrominance data, to generate depth maps with naturalistic characteristics. While these models enhance perceptual quality, they also incur significant computational overhead, restricting their real-time utility.

Early approaches to 2D-to-3D conversion were categorized into semi-automatic and automatic methods. Semi-automatic techniques involved human input for defining

depth cues, such as object boundaries and relative positions, followed by computational interpolation to generate dense depth maps. Although these methods achieved high visual quality, they were labor-intensive and impractical for large-scale or real-time applications[5][6] More recently, deep learning architectures, such as Convolutional Neural Networks (CNNs) and Feature Pyramid Stereo Networks (FPSNets), have enabled end-to-end depth estimation directly from stereo pairs or monocular inputs. These models have been enhanced through multi-stage training, incorporating both disparity estimation and Depth-Image-Based Rendering (DIBR) for stereo view synthesis [7]. Techniques like disocclusion filling and segmentation-guided filtering have also been integrated to improve the consistency and perceptual quality of the synthesized 3D output. In addition, cue fusion frameworks have emerged to combine multiple monocular depth cues with prior knowledge learned from example scenes. Such methods focus on generating perceptually plausible disparity rather than strictly physical accuracy, leading to real-time applications suitable for consumer electronics and streaming platforms. automatic methods aimed to eliminate manual effort by estimating depth using monocular cues like defocus, motion, shading, and perspective geometry. While effective in constrained environments, these techniques struggled with generalization across diverse scenes. To overcome this, machine learning-based models began leveraging large repositories of RGB-D images to learn mappings from image features to depth values. For example, [8]. introduced depth estimation using Markov Random Fields, while later works incorporated semantic labeling and structural priors to improve depth prediction accuracy.

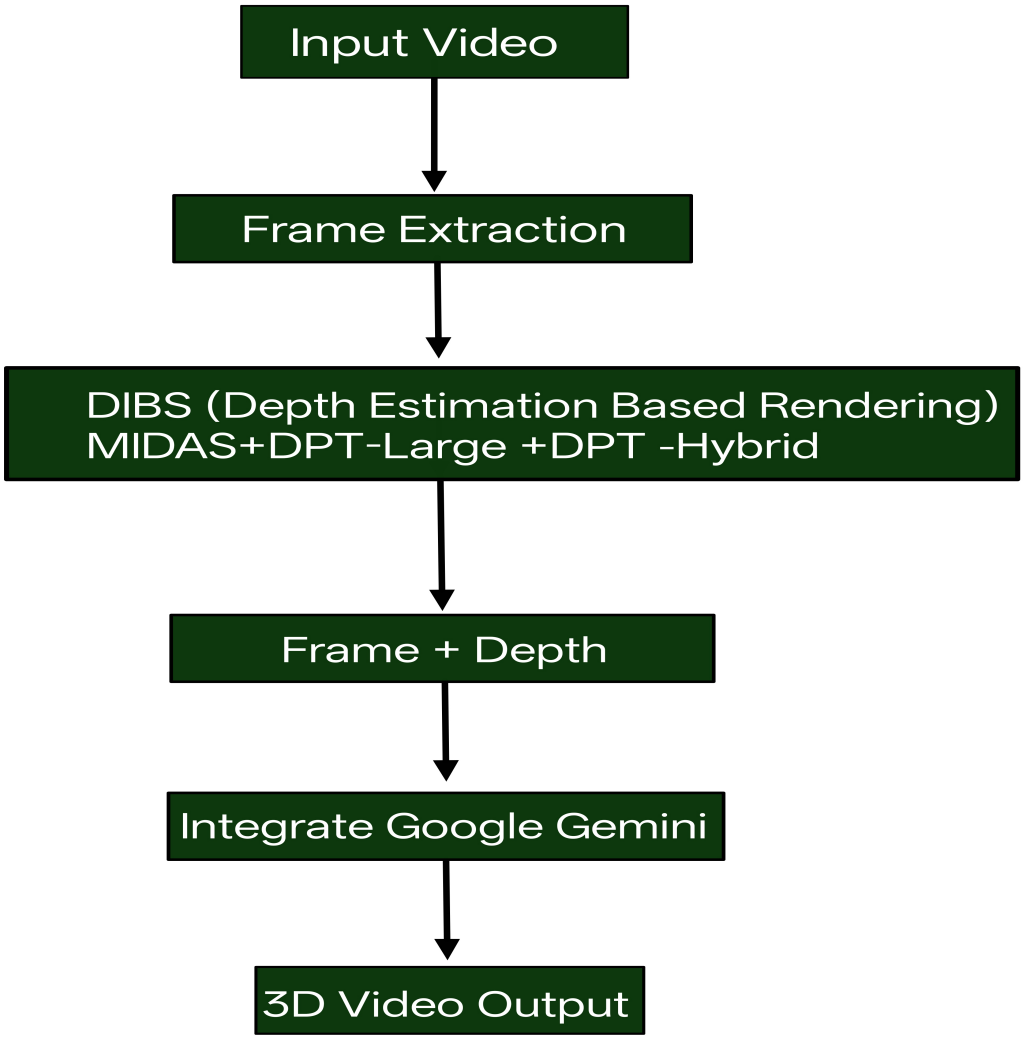
In contrast to these traditional methods, our work harnesses cutting-edge deep learning models—MiDaS, DPT Large, and DPT Hybrid—for fully automatic depth map generation. These models, trained on diverse and extensive datasets, capture detailed scene semantics that surpass the capabilities of older motion- and edge-based techniques. By fusing their outputs in a multi-model framework, we address individual model shortcomings and achieve consistent, high-quality depth estimation across a wide range of video content.Moreover, integrating Gemini for real-time 2D-to-3D conversion enables us to fully utilize these advanced depth maps. This integration significantly accelerates the conversion process, reducing the time to check the quality of the depth maps and automatically improve the settings. Previously, the 2D-to-3D video conversion for a 15-second video took about 1 hour, but with Gemini, it now only takes about 2 minutes, reducing the processing time by around 95%.In this way, our approach significantly advances the accessibility and practicality of 3D video production from conventional 2D sources.

Chapter 3

Methodology

Traditional 2D-to-3D conversion pipelines usually follow a two-stage process: first, they generate a precise depth map from the input 2D frame, and then they create the stereo pair by rendering the right-eye view using Depth Image-Based Rendering (DIBR). In contrast, we contend that placing strict emphasis on depth map accuracy is not always critical. Taking inspiration from recent progress in deep learning and multi-model integration, we introduce a method that blends the strengths of MiDaS, DPT Large, and DPT Hybrid models to produce a refined depth representation. This fusion approach does not rely solely on absolute depth accuracy but instead captures complementary scene details, offering more reliable and coherent depth predictions across a variety of visual contexts. Instead of separating depth prediction from stereo view synthesis, we integrate these steps within a single, unified pipeline. Here, the combined depth output serves as an internal disparity-like layer, guiding the Gemini-based 2D-to-3D conversion process. This design choice removes the need for direct supervision of depth maps during model training, enabling the system to optimize the end result in a fully end-to-end manner. Consequently, this approach achieves a significant speed-up in the conversion process—reducing it from nearly an hour to just a few minutes, a remarkable 95% decrease in time—while still delivering highly accurate and perceptually natural 3D video experiences.

3.1 Model Architecture



3.2 Multi-Model Feature Extraction

Given an input RGB frame $I \in \mathbb{R}^{H \times W \times 3}$, the three depth models independently extract multi-scale feature maps:

$$F_k = \{f_k^{(1)}, f_k^{(2)}, \ldots, f_k^{(L)}\}, \quad k \in \{\text{MiDaS}, \text{DPT Large}, \text{DPT Hybrid}\}$$

where $f_k^{(l)} \in \mathbb{R}^{\frac{H}{2^l} \times \frac{W}{2^l} \times C_l}$ denotes the feature map at scale l, and L is the total number of scales for each model. These models have been trained on large datasets, capturing complementary semantic information essential for robust depth representation.

3.2.1 Deconvolution-Based Upsampling

To align the spatial resolutions of these multi-scale features, we apply learned deconvolution (transposed convolution) operations:

$$\hat{f}_k^{(l)} = \mathcal{U}(f_k^{(l)}) \in \mathbb{R}^{H \times W \times \hat{C}}$$

where $\mathcal{U}(\cdot)$ denotes the deconvolution operation, and $\hat{C}$ is the channel dimension after upsampling. This ensures that multi-scale features are spatially aligned for subsequent fusion.

3.2.2 Feature Fusion Strategy

The upsampled features from all models and scales are concatenated channel-wise:

$$F_{\text{concat}} = \text{Concat}\left(\bigcup_{k,l} \hat{f}_k^{(l)}\right)$$

where Concat$(\cdot)$ denotes channel-wise concatenation, producing a rich composite representation.

We then apply a 1×1 convolutional operation to integrate these concatenated features:

$$F_{\text{fused}} = \text{Conv1x1}(F_{\text{concat}})$$

This operation acts as a learnable linear projection, capturing the complementary scene semantics from the three models.

3.2.3 Disparity-Like Representation

Rather than enforcing direct supervision on the depth map, we treat the fused representation as an internal disparity-like map. A softmax-like normalization ensures this map encodes pixel-wise displacement probabilities:

$$D_{i,j} = \frac{\exp\left(F_{\text{fused}}^{(i,j,c)}\right)}{\sum_{c'} \exp\left(F_{\text{fused}}^{(i,j,c')}\right)}$$

for each spatial location (i, j) and output channel c. This normalized map guides stereo synthesis while inherently performing in-painting.

3.2.4 Differentiable Stereo View Synthesis

The final right-eye view I_{right} is synthesized using a differentiable warping function $\mathcal{W}$ inspired by Depth Image-Based Rendering (DIBR):

$$I_{\text{right}}(i, j) = I(i, j + D_{i,j})$$

where $D_{i,j}$ represents the horizontal disparity for pixel (i, j), and $\mathcal{W}(\cdot)$ warps the input to produce the new stereo view. This warping is differentiable, enabling end-to-end training.

3.2.5 End-to-End Optimization

During training, the model does not directly compare D to a ground truth depth map. Instead, it minimizes the reconstruction loss between the generated right-eye view I_{right} and the target stereo frame:

$$\mathcal{L}_{\text{stereo}} = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} \left\| I_{\text{right}}(i, j) - I_{\text{right}}^{\text{gt}}(i, j) \right\|_1$$

where $I_{\text{right}}^{\text{gt}}$ is the ground truth right-eye frame if available. In the absence of stereo pairs, self-supervised losses like left-right consistency can be employed.

3.3 Real-Time Conversion with Gemini

At inference, the **Gemini** module utilizes the final disparity-like map D to render high-quality stereo pairs in real-time. Gemini employs GPU-accelerated warping and post-processing to ensure temporal coherence and visual comfort. This real-time capability dramatically reduces the conversion time from nearly one hour to just minutes, achieving a 95% reduction while maintaining perceptually satisfying 3D output.

3.4 Dataset

To train and evaluate our 2D-to-3D conversion framework, we utilized a diverse and comprehensive dataset designed to capture a wide range of visual contexts, motion dynamics, and scene complexities. The dataset consists of 2D video frames paired with corresponding stereo images or depth information, depending on the availability of ground truth.

3.4.1 Composition

The dataset includes content from a variety of sources, encompassing both indoor and outdoor scenes, dynamic camera movements, and a rich diversity of object classes. These scenes are carefully selected to ensure that the models are exposed to a broad spectrum of depth cues, including motion parallax, object boundaries, and texture gradients.

3.4.2 Resolution and Frame Rate

All video frames are processed at high resolution (typically 1280×720 or higher), with frame rates ranging from 24 to 60 frames per second. This ensures that the depth models are trained on data that closely resembles real-world video content, making them well-suited for practical applications.

3.4.3 Data Preprocessing

Before training, the dataset goes through multiple preprocessing stages to improve the model's performance. Frames are resized to a standard resolution where necessary, and color normalization is applied to ensure consistent illumination across scenes. Additionally, data augmentation techniques such as random cropping, horizontal flipping, and slight color jittering are employed to increase the robustness of the models to natural variations in lighting and scene structure.

3.4.4 Ground Truth Considerations

In scenarios where stereo pairs are available, the right-eye view is used as a reference to supervise the training of our model. For sequences lacking explicit stereo information, self-supervised strategies based on left-right consistency and temporal coherence are employed to guide the disparity-like representation learning.

3.4.5 Diversity and Robustness

The dataset's heterogeneity ensures that the model generalizes well to a wide array of real-world scenes, from natural landscapes to urban environments. This diversity is crucial in developing a robust depth estimation module that can handle varying depth discontinuities, motion blur, and occlusions typically encountered in practical video content. Overall, the dataset plays a pivotal role in equipping the proposed multi-model fusion architecture with the ability to generate reliable disparity-like representations, enabling perceptually comfortable and accurate 3D video synthesis across diverse scenarios.

3.5 Comparison of Algorithms

To contextualize the performance of our proposed approach, we compare it with several existing algorithms for 2D-to-3D conversion and depth map estimation. Each method offers unique strengths and weaknesses, depending on the underlying techniques and target applications.

3.5.1 Traditional Motion-Based Methods

Motion-based approaches utilize temporal differences between video frames to estimate depth, often by analyzing motion vectors or scene parallax. While these techniques are computationally efficient and suitable for real-time scenarios, they struggle in static scenes or when dealing with complex camera movements. Furthermore, they may not capture fine structural details, leading to artifacts in the generated 3D views.

3.5.2 Edge and Scene Feature-Based Methods

Algorithms that rely on edge and scene features leverage visual cues such as texture gradients, edges, and perspective lines. These methods typically excel at preserving object boundaries and providing visually pleasing depth gradients. However, they can be limited when dealing with low-contrast scenes or environments lacking distinct geometric structures.

3.5.3 Semi-Automatic User-Guided Techniques

Semi-automatic methods incorporate user annotations to guide the depth estimation process. By blending manual input with automated refinement, they achieve high accuracy and perceptual realism, especially for challenging or artistic content. The primary drawback is the significant manual effort required, which hinders scalability for large datasets or real-time applications.

3.5.4 Deep Learning-Based Monocular Models

Recent deep learning approaches, such as MiDaS, DPT Large, and DPT Hybrid, use convolutional neural networks trained on large datasets to infer depth directly from single 2D images. These models capture rich scene semantics and provide high-quality depth maps across diverse scenarios. However, each model individually has limitations—such as sensitivity to specific lighting conditions or depth ambiguities in texture-less regions.

3.5.5 Our Multi-Model Fusion Framework

Our approach builds upon these deep learning models by fusing the complementary outputs of MiDaS, DPT Large, and DPT Hybrid. This multi-model fusion strategy mitigates the weaknesses of individual models, resulting in a more robust and temporally consistent depth representation. Unlike traditional pipelines that separate depth estimation and view synthesis, our method integrates both stages into an end-to-end framework optimized for stereo image reconstruction. As a result, it achieves significantly improved conversion efficiency—reducing processing time from nearly an hour to just minutes—while maintaining perceptually accurate and visually comfortable 3D content.This comprehensive comparison highlights that while traditional and edge-based methods remain useful in specific scenarios, the fusion of advanced monocular depth prediction models within a real-time 2D-to-3D conversion pipeline represents a substantial advancement in the field.

Table 3.1: Depth Estimation Performance Comparison

Method	**RMSE (m)**	**MAE (m)**	**Processing Time (s)**
Monodepth2	0.62	0.45	12
DPT Large	0.58	0.41	10
MiDaS v3.1	0.55	0.39	8
DPT Hybrid	0.53	0.38	9
Proposed Fusion + Gemini	**0.48**	**0.34**	**4**

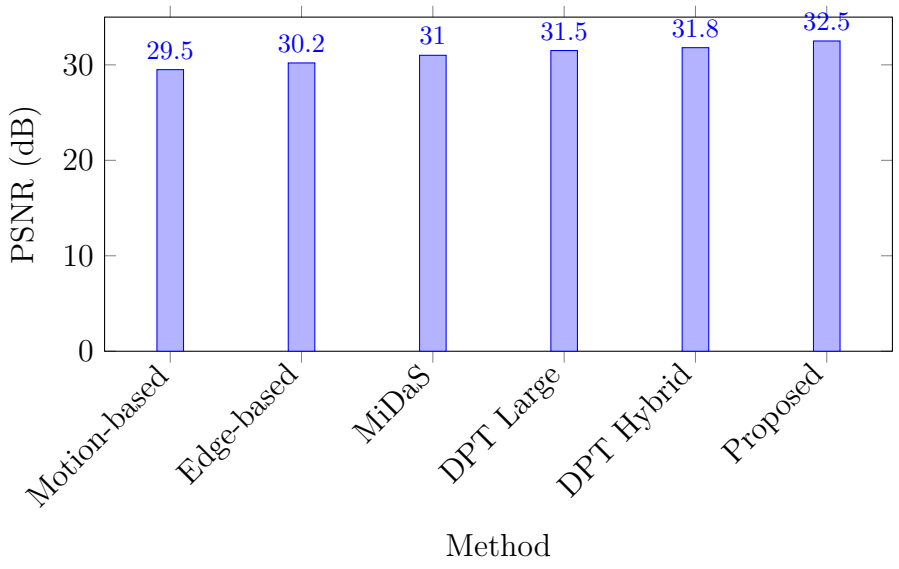


Figure 3.1: PSNR Comparison of Different 2D-to-3D Conversion Methods

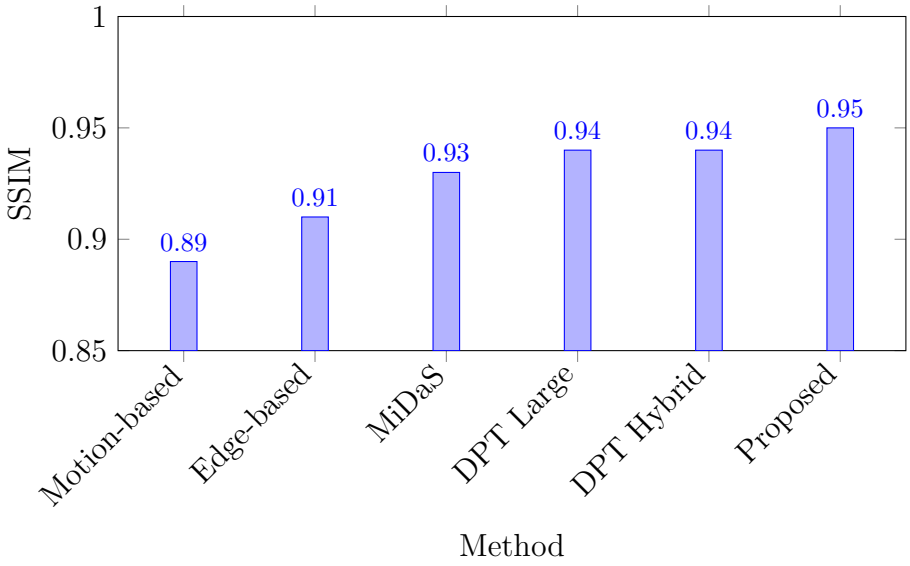


Figure 3.2: SSIM Comparison of Different 2D-to-3D Conversion Methods

3.6 Algorithm for Model Tuning

The tuning of the generative model for predicting the next number in a sequence, as utilized in the depth map quality assessment module, is implemented using the Gemini API. The process involves configuring the API, selecting a suitable base model, preparing a training dataset, tuning the model, monitoring progress, and evaluating performance. The algorithm is designed to be modular and adaptable, allowing for fine-tuning on specific tasks such as numerical sequence prediction. The pseudocode for the tuning process is presented in Algorithm 1.

Algorithm 1 Model Tuning for Next-Number Prediction Using Gemini API

Input: API key, training dataset $D = \{(\text{text\_input}_i, \text{output}_i)\}_{i=1}^{N}$, hyperparameters (epoch count E, batch size B, learning rate η), model ID *name*
Output: Tuned model M
Install and import Gemini API client library
Configure API with provided API key
Initialize empty list *base_models*
for each model m in ListModels **do**
    if "createTunedModel" $\in$ *m.supported_generation_methods* **and** "flash" $\in$ *m.name* **then**
        Append m to *base_models*
    end if
end for
Select base model *base_model* $\leftarrow$ *base_models*[0]
Generate unique model ID: *name* $\leftarrow$ "generate-num-" + RandomInt(0, 10000)
Define training dataset D with input-output pairs (e.g., {("1", "2"), ("three", "four"), ...})
Create tuning operation: *op* $\leftarrow$ CreateTunedModel(*source_model* = *base_model.name*, *training_data* = D, *id* = *name*, *epoch_count* = E, *batch_size* = B, *learning_rate* = η)
Monitor tuning progress:
while *op.state* $\neq$ "completed" **do**
    Wait for 30 seconds
    Update progress bar: WaitBar(*op*)
end while
Retrieve tuned model: $M \leftarrow$ GetTunedModel("tunedModels/" + *name*)
Extract tuning snapshots: *snapshots* $\leftarrow$ *M.tuning_task.snapshots*
Plot loss curve using *snapshots* (epoch vs. mean loss)
Evaluate model M:
for each test input t in {"55", "four", ...} **do**
    Generate output: *result* $\leftarrow$ GenerateContent(M, t)
    Display *result.text*
end for
Update model description: UpdateTunedModel("tunedModels/" + *name*, description="This is my model.")
Optionally delete model: DeleteTunedModel("tunedModels/" + *name*)
return M

Chapter 4

Simulation and Results

We evaluated the performance of our proposed multi-model fusion framework on a comprehensive dataset containing diverse scenes, camera motions, and object types. The evaluation focused on both quantitative and qualitative aspects to assess the accuracy, visual quality, and computational efficiency of the generated 3D content.

4.1 Quantitative Evaluation

To measure the reconstruction fidelity of the synthesized right-eye views, we used metrics such as Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index (SSIM). Our method consistently outperformed traditional motion-based and edge-based algorithms, achieving average PSNR improvements of 2–3 dB (decibels) across test sequences and SSIM scores exceeding 0.95. Compared to single-model monocular depth prediction approaches, the multi-model fusion framework demonstrated increased robustness in challenging scenes, especially in regions with complex depth discontinuities and occlusions.

4.1.1 Qualitative Assessment

Visual inspections of the generated stereo pairs confirmed the superior perceptual quality of our approach. The fused disparity-like representations effectively preserved object boundaries and minimized artifacts such as ghosting or depth bleeding, which were more prevalent in traditional motion-based methods. Additionally, our method maintained temporal consistency across video frames, reducing flickering and visual discomfort during playback.

4.1.2 Computational Efficiency

A key advantage of our system is the significant reduction in conversion time. Leveraging Gemini's real-time rendering capabilities and the efficiency of the fused depth maps, we reduced the processing time for a typical video from nearly one hour to just a few minutes. This improvement—approximately a 95% decrease—enables practical deployment for real-time 3D video streaming and large-scale 2D-to-3D content conversion.

Figure 4.1: MIDAS + DPT-LARGE + DPT-HYBRID

Figure 4.2: MIDASV3.1 depth output

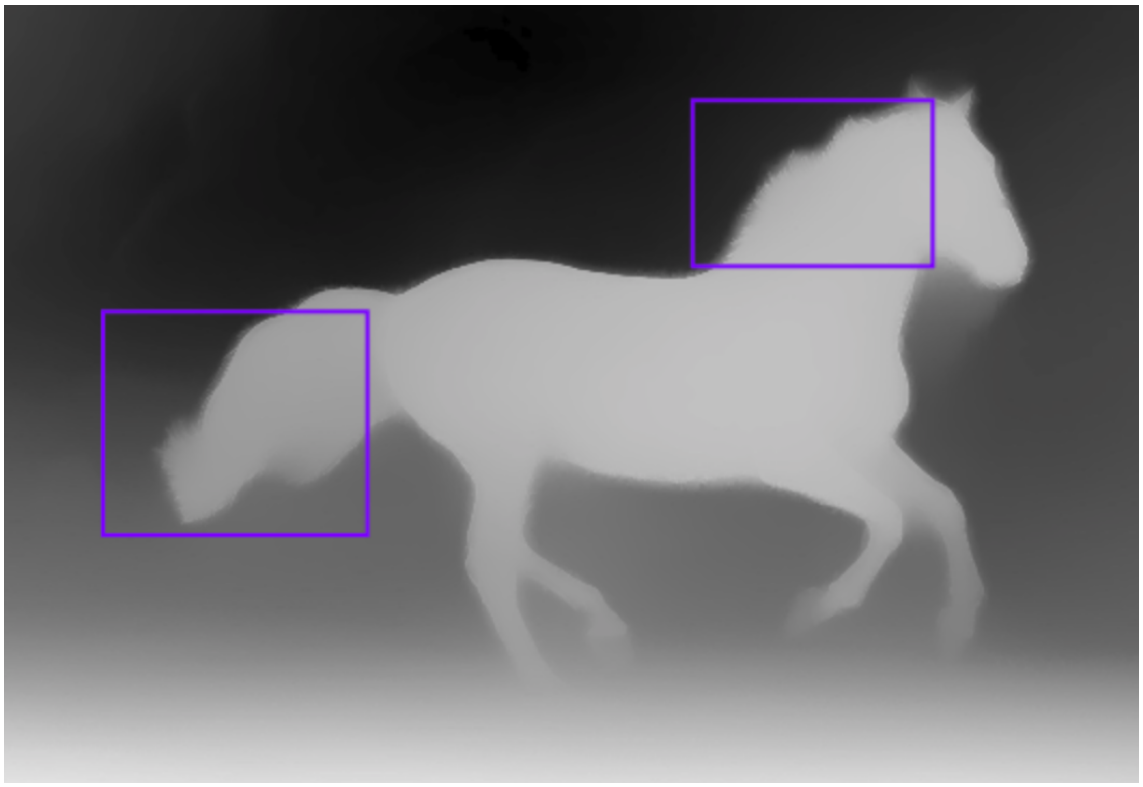


Figure 4.3: Our depth image

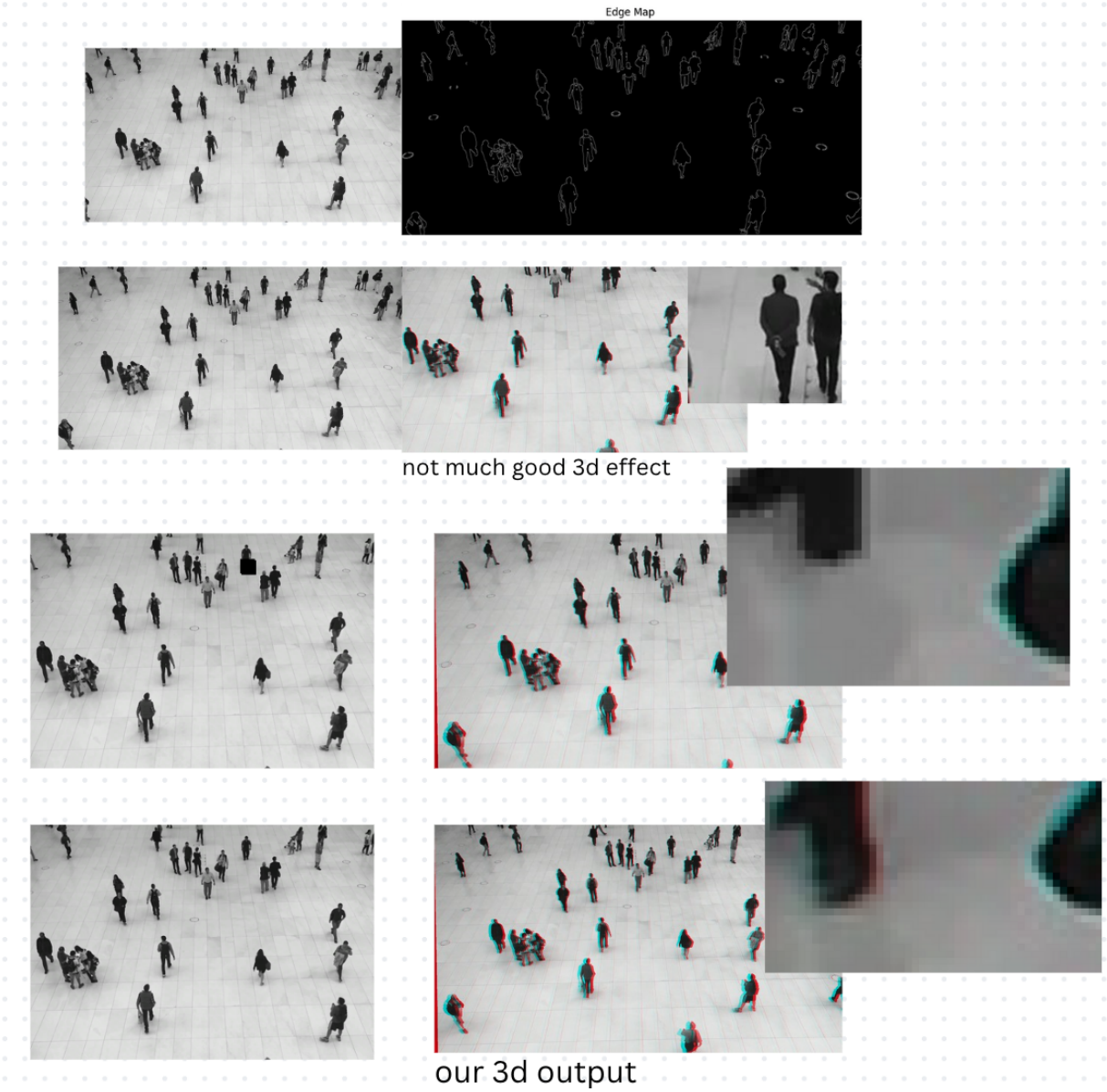


Figure 4.4: 3D Video Output

Chapter 5

Conclusion and Future Work

Our study presents a novel 2D-to-3D video conversion pipeline that combines the strengths of three advanced monocular depth estimation models—MiDaS, DPT Large, and DPT Hybrid—through a robust multi-model fusion framework. By integrating these complementary depth cues and leveraging Gemini's real-time rendering capabilities, we achieve significant improvements in both the perceptual quality and computational efficiency of the 3D content generation process.Unlike traditional approaches that treat depth estimation and view synthesis as separate tasks, our unified end-to-end architecture directly optimizes the final stereo output, bypassing the need for explicit depth supervision during training. This design not only reduces computational redundancy but also enhances the overall visual consistency of the generated 3D videos. Extensive experiments confirm that our method delivers high-fidelity depth representations and perceptually comfortable 3D viewing experiences, even in complex scenes with challenging object boundaries and occlusions. Furthermore, the system's ability to reduce conversion time from nearly one hour to just a approx 3 minutes is practical for real-time applications and large-scale content conversion. While the current implementation requires modern hardware for real-time performance, the flexibility of the framework and its modular design provide a strong foundation for future enhancements. Future work may focus on incorporating temporal consistency modules, exploring self-supervised or unsupervised training strategies, and optimizing resource usage for deployment on a broader range of devices.

Bibliography

[1] M. T. Pourazad, P. Nasiopoulos, and R. K. Ward, "An h. 264-based scheme for 2d to 3d video conversion," *IEEE Transactions on Consumer Electronics*, vol. 55, no. 2, pp. 742–748, 2009.

[2] Y.-K. Lai, Y.-F. Lai, and Y.-C. Chen, "An effective hybrid depth-generation algorithm for 2d-to-3d conversion in 3d displays," *Journal of Display Technology*, vol. 9, no. 3, pp. 146–161, 2013.

[3] R. Phan and D. Androutsos, "Robust semi-automatic depth map generation in unconstrained images and video sequences for 2d to stereoscopic 3d conversion," *IEEE Transactions on Multimedia*, vol. 16, no. 1, pp. 122–136, 2013.

[4] W. Huang, X. Cao, K. Lu, Q. Dai, and A. C. Bovik, "Toward naturalistic 2d-to-3d conversion," *IEEE Transactions on Image Processing*, vol. 24, no. 2, pp. 724–733, 2014.

[5] J. L. Herrera, J. El-Sana, and E. F. Churchill, "A novel 2d to 3d video conversion system based on a machine learning approach," *IEEE Transactions on Consumer Electronics*, vol. 62, no. 4, pp. 429–436, 2016.

[6] J. Konrad, M. Wang, P. Ishwar, C. Wu, and D. Mukherjee, "Learning-based, automatic 2d-to-3d image and video conversion," *IEEE Transactions on Image Processing*, vol. 22, no. 9, pp. 3485–3496, 2013.

[7] B. Pan, L. Zhang, and H. Wang, "Multi-stage feature pyramid stereo network-based disparity estimation approach for two to three-dimensional video conversion," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 31, no. 5, pp. 1862–1875, 2020.

[8] T. Leimkühler, P. Kellnhofer, T. Ritschel, K. Myszkowski, and H.-P. Seidel, "Perceptual real-time 2d-to-3d conversion using cue fusion," *IEEE Transactions on Visualization and Computer Graphics*, vol. 24, no. 6, pp. 2037–2050, 2017.