3D Photography
using
Context-aware Layered Depth Inpainting

Submitted in partial fulfilment of the requirements of the degree of

BACHELOR OF COMPUTER ENGINEERING

by

Omkar Chougule (19102025)
Devansh Katheria (19102027)
Abhishek Singh (19102061)
Kshitiz Jain (19102015)

Guide:
Prof. Sachin Malave

Department of Computer Engineering

A. P. SHAH INSTITUTE OF TECHNOLOGY, THANE
(2022-2023)

A. P. SHAH INSTITUTE OF TECHNOLOGY

CERTIFICATE

This is to certify that the Project entitled “***3D Photography using Context-aware Layered Depth Inpainting”*** is a bonafide work of ***“Omkar Chougule, Kshitiz Jain, Abhishek Singh and Devansh Katheria*”** submitted to the University of Mumbai in partial fulfilment of the requirement for the award of the degree of ***Bachelor of Engineering*** *in* ***Computer Engineering*****.**

____

Guide
Prof. Sachin Malave

____

Project Coordinator
Prof. Rushikesh Nikam

____

Head of Department
Prof. Sachin Malave

____

Principal
Dr. Uttam Kolekar

Date:

A. P. SHAH INSTITUTE OF TECHNOLOGY

Project Report Approval for BE

This Mini project report entitled ***3D Photography using Context-aware Layered Depth Inpainting*** by ***Omkar Chougule, Kshitiz Jain, Abhishek Singh and Devansh Katheria*** is approved for the degree of ***Bachelor of Engineering*** in ***Computer Engineering***, ***2022-23***.

Examiner Name Signature

1. ______________________________

2. ______________________________

Date:

Place:

Declaration

We declare that this written submission represents my ideas in my own words and where others' ideas or words have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. I understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Omkar Chougule, 19102025

Kshitiz Jain, 19102015

Abhishek Singh, 19102061

Devansh Katheria, 19102027

Date:

Abstract

3D images can add a whole new dimension to photography. However, creating such parallax effects using conventional reconstruction and rendering techniques requires a complex setup and special hardware, which is not always feasible. By taking two clicks of the same scene, one barely off-center from the other, a 3D image can be produced. Your brain will be fooled into thinking you are seeing an image with depth by into believing you are viewing an image with depth by this small difference. Recent advanced cellphone cameras, such as a camera with two lenses, make it possible to capture depth information. Such images have extra parameter depth along with regular RGB parameters. We have studied the previous work '3D photography' and identified several problems. The first is that hair-like structures are omitted during image rendering. Second, if the image has low contrast or the background is similar to the object, the rendering will be blurry due to improper depth estimation and edge detection. And third, when the foreground layer contains thin objects that are rendered separately those are part of the object. We propose a method to transform a single image into a 3D photo, solving the second problem by generating depth map images using the pre-trained model MiDasV3.0 DPT Large and then applying edge sharpening techniques (bilateral and morphological filters). The results show correct 3D rendering with blur even if an image has low contrast or the background is similar to the foreground objects.

CONTENTS

List of Figures

List of Tables

Abbreviation

LDI	Layered Depth Image
BMD	Boosting Monocular Depth
SLIDE	Single Image 3D Photography with Soft Layering and Depth-aware Inpainting
DPT	Dense Prediction Transformer
SSMR	Structured Similarity Indexing Method
ORB	Oriented FAST and Rotated BRIEF
PSNR	Peak Signal-to-Noise Ratio with blocking
RMSE	Normalized Root Mean Square Error

CHAPTER 1

Introduction

3D pictures can take your photography to a whole new dimension. [3] With traditional rendering techniques, the time required to render an image increase with the geometric complexity of the scene. The rendering time also grows as the requested shading computations (such as those requiring global illumination solutions) become more ambitious. However, creating such parallax effects with classical reconstruction and rendering techniques requires elaborate setup and specialized hardware, which is not always feasible. [1] Depth is the most important aspect of 3D photography. A 3D image can be created by taking two shots of the same scene, where one is a little offset to the other. This slight difference is enough to trick your brain into thinking you are looking at an image with depth. Recent advancements in cell phone cameras, like dual-lens camera, enable capturing depth information. The resulting image is an RGB-D (color and depth) image. In an attempt to generate a lifelike view from this RGB-D image, occlusions created by parallax must be nullified. LDI, which contains potentially multiple depth pixels at each discrete location in the image. [3] Instead of a 2D array of depth pixels (a pixel with associated depth information), we store a 2D array of layered depth pixels. A layered depth pixel stores a set of depth pixels along one line of sight sorted in front-to-back order. The front element in the layered depth pixel samples the first surface seen along that line of sight; the next pixel in the layered depth pixel samples the next surface seen along that line of sight, etc. [1] The authors have proposed a method of converting a single image into a 3D photo. They have used Layered Depth Image (LDI) as an underlying representation and have presented a learning-based inpainting model that can synthesize new color and depth in the occluded region. A color single RGB-D input image is used as input and the 3D photo is generated in MP4 format. 3D photo generation is based on the multi-layer representation where these layers contain hallucinated color and depth structures in regions occluded in the original view. The library uses Layered Depth Image (as input) with explicit pixel connectivity as underlying representation and present a learning-based inpainting model that iteratively synthesizes new local color-and-depth content into the occluded region in a spatial context-aware manner. The final 3D photo result image can be efficiently rendered with motion parallax using standard graphics engines.

We have studied and identified some issues in the previous work on ‘3D photography’. The first one is it misses hair-like structures from the image during rendering. Secondly, if the image is having low contrast or the background is similar to the object then rendering is blurred because of improper depth estimation and edge detection. And thirdly, if the front layer has thin objects like a thin stick, then the thin part of the object is rendered separately which is part of the object.

We are proposing a method of converting a single image into a 3D photo. We have solved the second problem by using depth map image generation with pre-trained model MiDasV3 and then edge sharpening techniques (bilateral and morphological filters). Results show proper 3D rendering even if an image has thin objects. The problem of hair line structure visibility is solved by the previous work “SLIDE” in which foreground and background is separated and foreground rendered on background rendering [2].

The main challenge was these are already state-of-the-art methods, which we have first grasped the procedure the ways they have done and improving the same is bit challenging. We tried different depth estimation models, different edge sharpening methods and filters and baselined the method which is given the best performance.

CHAPTER 2

Literature Survey

Layered Depth Image, or LDI, that contains potentially multiple depth pixels at each discrete location in the image. Instead of a 2D array of depth pixels (a pixel with associated depth information), we store a 2D array of layered depth pixels. A layered depth pixel stores a set of depth pixels along one line of sight sorted in front to back order. The front element in the layered depth pixel samples the first surface seen along that line of sight; the next pixel in the layered depth pixel samples the next surface seen along that line of sight, etc. When rendering from an LDI, the requested view can move away from the original LDI view and expose surfaces that were not visible in the first layer. The previously occluded regions may still be rendered from data stored in some later layer of a layered depth pixel. There are a set of efficient image based rendering methods capable of rendering multiple frames per second on a PC. The first method warps Sprites with Depth representing smooth surfaces without the gaps found in other techniques. A second method for more general scenes performs warping from an intermediate representation called a Layered Depth Image (LDI). An LDI is a view of the scene from a single input camera view, but with multiple pixels along each line of sight. The size of the representation grows only linearly with the observed depth complexity in the scene. Moreover, because the LDI data are represented in a single image coordinate system [3].

It is possible to create edge maps that look eerily similar to human sketches by performing element-wise multiplication on Canny and HED edge maps. EdgeConnect, a new deep learning model for image inpainting tasks. EdgeConnect comprises of an edge generator and an image completion network

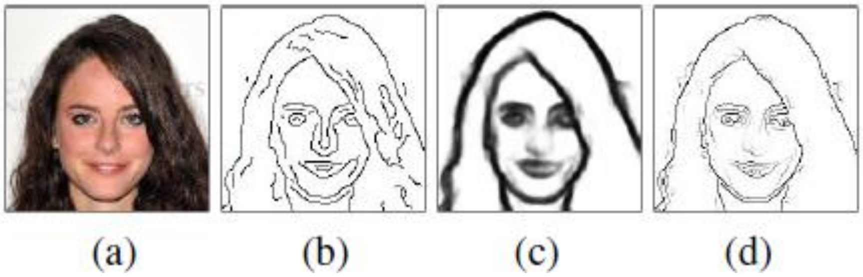


Fig. 1. Edge Connect (a) Image. (b) Canny. (c) HED. (d) Canny_HED

By using a new randomized algorithm for finding approximate nearest neighbour matches between image patches. This algorithm provides substantial improvements over the previous work and making it use in interactive image editing tools. With the help of random sampling, we can find some good path matches and natural coherence in the imagery which allows us to propagate such matches quickly to surrounding areas. Theoretical and practical evidence are provided for its high quality and performance. This can be used as many tools such as image retargeting, completion and reshuffling which can be used together in high level image editing tools. Result: Marking out mask regions, users can interactively fill nontrivial holes. Reshuffling tools can quickly fix the architectural dimensions and layout of ancient monuments. This is because architecture often contains repeating patterns. The nature of this algorithm bears some superficial similarity to LBP and Graph Cuts algorithm often used to solve Markov Random Fields on an image grid, but there are differences as this algorithm is designed to optimize as energy function without any neighbourhood term. This algorithm has no explicit generative model, but uses coherency in the data to prune the search for likely solution to a simpler parallel search problem. This algorithm does has some failure cases, in extreme edits to an image can sometimes produce 'ghosting' or 'feathering' artifacts where the algorithm simply cannot escape a large local minimum basin. [4]

Inconsistency between color edges in guidance color images and depth discontinuities on depth map are the most challenging issues in color guided depth map restoration. Due to this the restored depth map suffer from texture copy artifacts and blurring depth discontinuities. New complex algorithms which are based on guidance color images and heuristically make use of the bicubic interpolation of the input depth map are developed to handle this situation. But bicubic interpolated depth map can blur depth discontinuities when the upsampling factor is large and the input depth map contains large holes and heavy noise. This paper propose a robust optimization technique for color-guided depth map restoration. This method is robust against the inconsistency between color edges and depth discontinuities even when we use simple guidance weight. Moreover, the proposed system works well in suppressed textured copy artifacts. It can preserve sharp depth discontinuities than previous heuristic weighting methods [5].

Here they consider image transformation problems, where an input image is transformed into an output image. Feed-forward convolutional neural networks typically train images using a per-pixel loss. they show results on image style transfer, where a feed-forward network is trained to solve the optimization problem in real-time. In this paper they have combined the benefits of feed-forward image transformation tasks and optimization-based methods for image generation by training networks with perceptual loss functions. they have applied this method to style transfer where we achieve comparable performance and drastically improved speed compared to existing methods, and to single-image super-resolution [6].

Current methods for combining two different images produce artifacts when the sources have very different textures and structures. New method synthesizes a transition region between two source images, such that inconsistent color, texture, and structural properties all change gradually from one source to the other. They propose a new energy based on mixed L2/L0 norms for colors and gradients that produces a gradual transition between sources without sacrificing texture sharpness. They enrich the patch search space with additional geometric and photometric transformations. They integrate image gradients into the patch representation and replace the usual color averaging with screened Poisson equation solver. In several cases, our unified method outperforms previous state-of-the-art methods [7].

Recent deep learning based approaches have shown promising results for the challenging task of inpainting large missing regions in an image. These methods can generate visually plausible image structures and textures, but often create distorted structures or blurry textures inconsistent with surrounding areas. Motivated by these observations, we propose a new deep generative model-based approach which can not only synthesize novel image structures but also explicitly utilize surrounding image features as references during network training to make better predictions. The model is a feed-forward, fully convolutional neural network which can process images with multiple holes at arbitrary locations and with variable sizes during the test time. Experiments on multiple datasets including faces (CelebA, CelebA-HQ), textures (DTD) and natural images (ImageNet, Places2) demonstrate that our proposed approach generates higher-quality inpainting results than existing ones [8].

The dense prediction transformer (DPT) is a dense prediction architecture that is based on an encoder-decoder design that leverages a transformer as the basic computational building block of the encoder. Dense vision transformers, are an architecture that leverages vision transformers in place of convolutional networks as a backbone for dense prediction tasks. The transformer backbone processes representations at a constant and relatively high resolution and has a global receptive field at every stage. These properties allow the dense vision transformer to provide finer-grained and more globally coherent predictions when compared to fully convolutional networks. This architecture yields substantial improvements on dense prediction tasks, especially when a large amount of training data is available [11]

Here they consider image transformation problems, where an input image is transformed into an output image. Feed-forward convolutional neural networks typically train images using a per-pixel loss. they show results on image style transfer, where a feed-forward network is trained to solve the optimization problem in real-time. In this paper they have combined the benefits of feed-forward image transformation tasks and optimization-based methods for image generation by training networks with perceptual loss functions. they have applied this method to style transfer where we achieve comparable performance and drastically improved speed compared to existing methods, and to single-image super-resolution. [8]

Current methods for combining two different images produce artifacts when the sources have very different textures and structures. New method synthesizes a transition region between two source images, such that inconsistent color, texture, and structural properties all change gradually from one source to the other. They propose a new energy based on mixed L2/L0 norms for colors and gradients that produces a gradual transition between sources without sacrificing texture sharpness. They enrich the patch search space with additional geometric and photometric transformations. They integrate image gradients into the patch representation and replace the usual color averaging with screened Poisson equation solver. In several cases, our unified method outperforms previous state-of-the-art methods [9].

Recent approaches combine monocular depth networks with inpainting networks to achieve compelling results. A drawback of these techniques is the use of hard depth layering, making them unable to model intricate appearance details such as thin hair-like structures. Authors introduced SLIDE (Soft-Layering and Inpainting that is Depth-aware) technique which is modular and unified system for single image 3D photography that uses a simple and effective soft layering. The strategy that enables synthesizing intricate appearance details such as hair-like structure, which preserves appearance details in novel views. They have also proposed a novel depth-aware training strategy for the inpainting module. The SLIDE approach is modular which enables the use of other components such as segmentation and matting for improved layering. SLIDE uses a simple two-layer decomposition of the scene and efficient layered depth formulation that requires only a single forward pass through the components networks which produce high-quality 3D photos [2].

	Paper Name	**Strengths**	**Research Gap**
[1]	3D Photography using Context-	3D Reconstruction: It can generate high-quality 3D reconstructions of scenes	The first is that hair-like structures are omitted during image rendering. Second, if

	aware Layered Depth Inpainting	from single 2D images, making it useful for a variety of applications like virtual and augmented reality, gaming, and film.	the image has low contrast or the background is similar to the object, the rendering will be blurry due to improper depth estimation and edge detection. And third, when the foreground layer contains thin objects that are rendered separately those are part of the object.
[2]	SLIDE: Single Image 3D Photography with Soft Layering and Depth-aware Inpainting	SLIDE uses an efficient layered depth formulation that only requires a single forward pass through the component networks to produce high quality 3D photos. The model does not miss model intricate appearance details such as thin hair-like structures.	Limited view points: SLIDE can only generate 3D models from a single viewpoint, which means that the models may not be accurate from other viewpoints.
[3]	Layered Depth Images	Efficient representation: The LDI technique provides an efficient way to represent and manipulate images that contain multiple layers of depth information. This makes it useful for a variety of applications, such as	Limited to discrete layers: The LDI technique is limited to representing depth information in discrete layers. This can result in loss of detail and accuracy, particularly for complex scenes that contain objects

		computer graphics, augmented and virtual reality, and image and video processing. Flexibility: The LDI technique can handle a variety of image types, including photographs, computer-generated images, and videos. High-quality rendering: The LDI technique allows for high-quality rendering of images, even when they contain complex depth information.	with overlapping depth information. Requires pre-processing: The LDI technique requires pre-processing of the input images to generate the LDI representation. This can be time-consuming and computationally expensive, particularly for large datasets or videos.
[4]	EdgeConnect - Generative Image Inpainting with Adversarial Edge Learning	High-quality results: EdgeConnect can generate high-quality image inpainting results that blend seamlessly with the surrounding image content, making it useful for a variety of applications, such as photo restoration, image editing, and image completion.	

[5]	Robust Color Guided Depth Map Restoration	Robust Color Guided Depth Map Restoration can generate high-quality depth maps that are visually appealing and accurate. This makes it useful for a variety of applications, such as 3D reconstruction and virtual reality.	Limited to single depth map restoration: Robust Color Guided Depth Map Restoration is designed to restore a single depth map at a time, which may limit its applicability in scenarios where multiple depth maps need to be restored simultaneously.
[6]	Perceptual Losses for Real-Time Style Transfer	Perceptual Losses for Real-Time Style Transfer can generate stylized images in real-time, making it useful for a variety of applications, such as video streaming and live broadcasting.	Perceptual Losses for Real-Time Style Transfer is limited to a specific set of style transfer options that have been trained on a particular dataset. If a new style is desired, the method requires retraining with the new style images.
[7]	Image Melding Combining Inconsistent Images using Patch-based Synthesis	Image Melding can generate high-quality consistent images that blend the input images seamlessly, making it useful for a variety of applications, such as image editing, image compositing, and image retouching.	Limited to two or three input images: Image Melding is designed to work with two or three input images only, and may not be suitable for combining larger sets of images.

[8]	Generative Image Inpainting with Contextual Attention	Generative Image Inpainting with Contextual Attention can generate high-quality inpainted images that are visually appealing and realistic. This makes it useful for a variety of applications, such as image editing, photo restoration, and image completion.	Generative Image Inpainting with Contextual Attention can be computationally intensive, particularly when processing high-resolution images or when inpainting large areas of an image. This can limit its applicability in real-time or low-end devices.
[9]	Midas 3 DPT	MiDaS 3DPT can estimate accurate depth maps even for challenging scenes with complex geometry and texture, making it useful for a variety of applications, such as virtual reality, augmented reality, and 3D reconstruction.	MiDaS 3DPT can be computationally intensive, particularly when processing large sets of images or when estimating depth maps for high-resolution images. This can limit its applicability in real-time or low-end devices.
[10]	Towards Robust Monocular Depth Estimation: Mixing Datasets for Zero-Shot Cross-Dataset Transfer	By training on a mix of datasets, Towards Robust Monocular Depth Estimation can improve the generalization of depth estimation models to new and unseen data, making it useful for a variety of	By mixing datasets, there is a risk of introducing dataset bias and reducing the generalization of the depth estimation model. This may require additional pre-processing or fine-tuning to ensure that the model

		applications where robust and accurate depth estimation is critical.	performs well on new datasets.
[11]	3D Moments from Near-Duplicate Photos	3D Moments from Near-Duplicate Photos is a non-intrusive method that does not require specialized equipment or sensors, making it easy and cost-effective to implement. It can estimate accurate 3D structure from a small number of photos, making it useful for applications where precise 3D information is required.	Limited to near-duplicate photos: This method requires a set of near-duplicate photos, which can limit its applicability in scenarios where only a single photo or a small number of photos are available.
[12]	Boosting Monocular Depth Estimation Models to High-Resolution	Improved Spatial Accuracy: Boosting monocular depth estimation models to high-resolution can improve the spatial accuracy of depth maps. High-resolution depth maps contain more details, which can help improve the accuracy of depth estimation.	Increased Computational Cost: Boosting monocular depth estimation models to high-resolution increases the computational cost of depth estimation. This can be a significant issue in real-time applications such as robotics or autonomous driving. Larger Model Sizes: High-resolution depth estimation

		Enhanced Depth Perception: High-resolution depth maps can provide enhanced depth perception, making it easier to distinguish between objects located at different depths. This can be particularly useful in applications such as autonomous driving or robotics.	models tend to be larger in size than low-resolution models. This can make it more challenging to deploy models on resource-constrained devices such as mobile phones or embedded systems.
[13]	Object-Driven Multi-Layer Scene Decomposition From a Single Image	Object-driven multi-layer scene decomposition can accurately segment objects in an image, which can be useful in applications such as object recognition and tracking. Improved Depth Estimation: Multi-layer scene decomposition can also improve the accuracy of depth estimation by separating objects into different layers and estimating their depth separately.	Computational Complexity: Object-driven multi-layer scene decomposition can be computationally complex, especially when dealing with complex scenes with multiple objects. This can make it challenging to deploy the technique in real-time applications. Limited Applicability: The accuracy of this technique can be limited when dealing with objects that have similar colors or textures, as it may be difficult to separate them into different layers.

[14]	Learning to Recover 3D Scene Shape from a Single Image	More Information for Scene Understanding: By estimating the 3D shape of the scene, this technique can provide more information for scene understanding, such as the layout of objects in the scene and the relationships between them.	Learning to recover 3D scene shape from a single image can be challenging, and the accuracy of the technique is limited by the quality of the input image and the complexity of the scene.

Table 2.1: Literature Survey Table

CHAPTER 3

Limitations of Existing System

3D rendering is a process of generating a 2D image from a 3D model by applying complex mathematical algorithms. The output of 3D rendering can be stunning, but it can also be blurred or improperly in-painted in certain cases. This can occur when an image contains hair-like structures, objects with similar colors as the background or when there is less contrast between objects, and when the front layer has thin objects like thin sticks.

Hair-like structures are challenging to render because they are usually small and irregularly shaped. 3D rendering software may not be able to accurately represent such fine details, resulting in blurred or distorted images. In such cases, special techniques such as the use of particle systems or advanced hair rendering algorithms can be used to create more accurate representations of hair-like structures.

Similarly, if an object has a similar color to the background or if there is little contrast between objects, the 3D rendering software may not be able to differentiate between them properly, leading to poorly rendered images. To overcome this limitation, depth mapping techniques can be used to improve the accuracy of object placement and differentiation.

In the case of thin objects like thin sticks, the 3D rendering software may have difficulty accurately representing their shape and position in space, resulting in improper in-painting or blurring. Techniques like texture filtering can be used to enhance the accuracy of the in-painting and improve the overall quality of the rendered image.

In conclusion, while 3D rendering is a powerful tool for generating realistic images, it can still face challenges in accurately rendering fine details or differentiating between similar objects or colors. However, with the use of specialized techniques and algorithms, it is possible to overcome these limitations and create stunning 3D renderings that accurately represent the real world.

CHAPTER 4

Problem Statement, Objectives and Scope

4.1 Problem Statement

1. Convert a single 2D RGB image into a 3D photo — a multi-layer representation for novel view synthesis that contains hallucinated color and depth structures in regions occluded in the original view.

2. Demonstrate proper rendering without any blur during 3D rendering even if the image is having low contrast or the background is similar to the object. The previous work 3D photography shows blur during rendering in such cases.

4.2 Objectives:

- We propose a method for converting a single 2D RGB input image into a 3D photo — a multi-layer representation for novel view synthesis that contains hallucinated color and depth structures in regions occluded in the original view.

- The resulting 3D photos can be efficiently rendered with motion parallax using standard graphics engines.

- 3D photography capturing views of the world with a camera and using image-based rendering techniques for novel view synthesis is a fascinating way to record and reproduce visual perception.

- It provides a dramatically more immersive experience than old 2D photography: almost lifelike in Virtual Reality, and even to some degree on normal flat displays when displayed with parallax.

- Improve the methodology to solve the issue in the previous work. The issue is if an image is having low contrast, or the background is similar to the object then rendering is blurred because of improper depth estimation and edge detection.

4.3 Scope:

- Convert stereo 2D (RGB-D) images into following 3D views with improved depth inpainting than earlier inpainting algorithms available.
  - Circular 3D view
  - Swing 3D view
  - Zoom-in 3D view

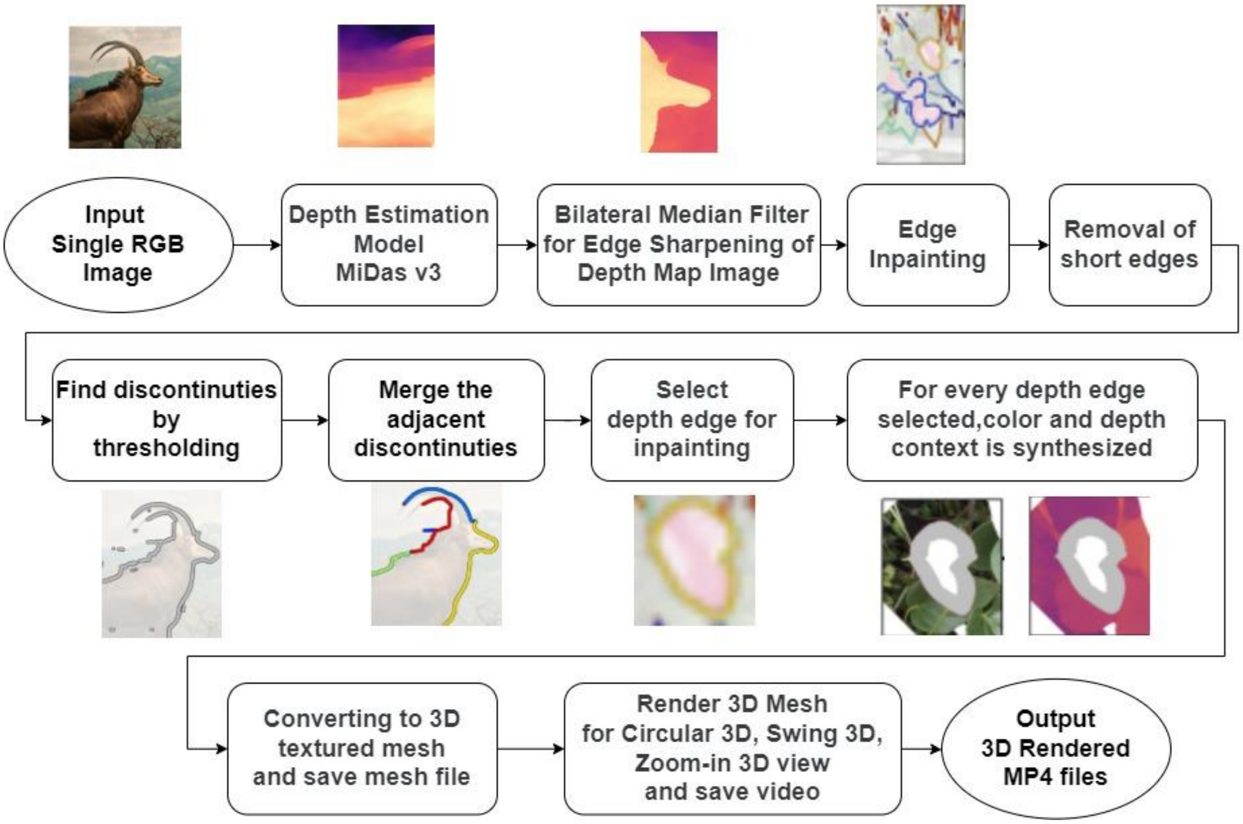
- Resolve the issue during 3D Photography rendering in the case, if an image is having low contrast or the background is similar to the object then rendering is blurred.

CHAPTER 5

Proposed System

5.1 Methodology:

Following are the major steps followed in the proposed system.



Input Single RGB Image
Depth Estimation Model MiDas v3
Bilateral Median Filter for Edge Sharpening of Depth Map Image
Edge Inpainting
Removal of short edges
Find discontinuties by thresholding
Merge the adjacent discontinuties
Select depth edge for inpainting
For every depth edge selected,color and depth context is synthesized
Converting to 3D textured mesh and save mesh file
Render 3D Mesh for Circular 3D, Swing 3D, Zoom-in 3D view and save video
Output 3D Rendered MP4 files

Fig. 2. Methodology

- **Input RGB Image:** The input to this method is **single RGB-D** image. The depth information can come from a mobile dual camera or can be estimated from a single RGB image.

- **Depth Estimation:** If a image if only RGB image taken from single camera then we have used a pretrained depth estimation model MiDaS v3 to compute depth from the input image with only color information. Thus, the proposed method applies to any image.

  We normalize the depth channel, by mapping the min and max disparity values (i.e., 1 / depth) to 0 and 1, respectively.

- **Sharpen the edges:** However, the depth map from the dual camera or depth estimation has blurry discontinuities across multiple pixels. To sharpen it, we used a bilateral median filter. This step thus ensures easy localization of the edges.

- **Edge Inpainting:** We adopt the architecture provided by Edge Connect.

- **Short Edge Removal:** There are few more sub-steps like thresholding and removal of short edges (<10 pixels).

- **Find discontinuities by thresholding:** After sharpening the depth map, we find discontinuities by thresholding the disparity difference between neighbouring pixels.

- **Merge the adjacent discontinuities:** Merge adjacent discontinuities into a collection of “Linked depth edges”.

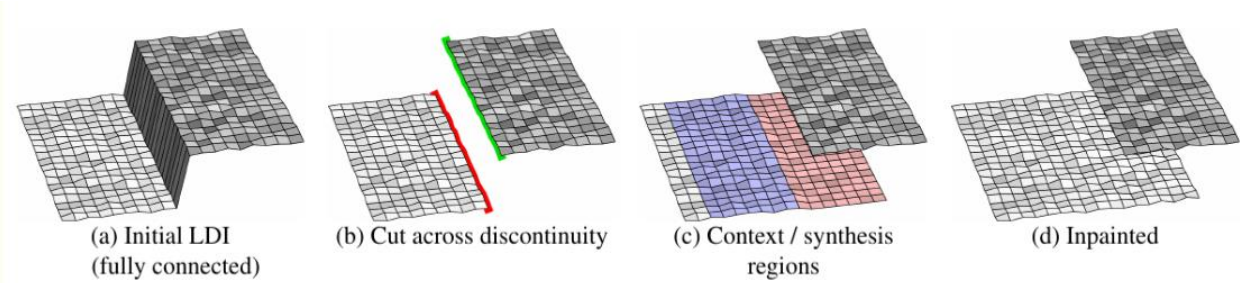
- **Randomly select Depth Edges:** We randomly select one of the edges as a subproblem.

- **Color and Depth Synthesized:** For the depth and color inpainting networks, we use a standard U-Net architecture with partial convolution.

- **Converting to 3D Textured Mesh:** We form the 3D textured mesh by integrating all the inpainted depth and color values back into the original LDI. Using mesh representations for rendering allows us to quickly render novel views, without the need to perform per-view inference step.

- **Render the 3D Mesh:** 3D mesh structure of the image is now can easily be rendered using standard graphics engines on edge devices.
- **Save the Output 3D Rendered MP4 files:** Save videos in MP4 files for Circular 3D, Swing 3D, Zoom-in 3D novel view.

5.2 Context and Synthesis Regions:



(a) Initial LDI (fully connected)

(b) Cut across discontinuity

(c) Context / synthesis regions

(d) Inpainted

Fig. 3. Context and Synthesis Regions

Initial Fully connected LDI (Fig a):

Our inpainting algorithm operates on one of the previously computed depth edges at a time. Given one of these edges, the goal is to synthesize new color and depth content in the adjacent occluded region.

Cut across discontinuity (Fig b):

We start by disconnecting the LDI pixels across the discontinuity. We call the pixels that became disconnected (i.e., are now missing a neighbor) silhouette pixels. We see in Figure b that a foreground silhouette (marked green) and a background silhouette (marked red) forms.

Context / synthesis regions (Fig c):

Only the background silhouette requires inpainting. We are interested in extending its surrounding content into the occluded region. We start by generating a synthesis region, a contiguous region of new pixels (pink region in fig. c)

Inpainting the synthesis region (Fig d):

We initialize the color and depth values in the synthesis region using a simple iterative flood-fill like algorithm.

Context aware color and depth inpainting:

Given the context and synthesis regions, our next goal is to synthesize color and depth values. Even though we perform the synthesis on an LDI, the extracted context and synthesis regions are locally like images, so we can use standard network architectures designed for images. Specifically, we build our color and depth inpainting models upon image inpainting methods in [13,14,15]

5.3 Depth Estimation Model Selection:

We have studied and identified some issues in the previous work on '3D photography'. The first one is it misses hair-like structures from the image during rendering. Secondly, if the front layer has thin objects like a thin stick, then the thin part of the object is rendered separately which is part of the object.

Therefore, we have reviewed different following state-of-the-art depth estimation models:

1. MiDaS 2.1 + BMD
2. MiDaS v2 Large + BMD
3. MiDaS v2.1 Large + BMD
4. MiDaS v3.0 DPT-Large + BMD
5. MiDaS v3.0 DPT-Large

We found MiDaS v3.0 DPT-Large gives the better result of depth estimation.

5.4 Comparison of Depth Estimation Models:

1) Depth estimation using MiDaS v2.1:

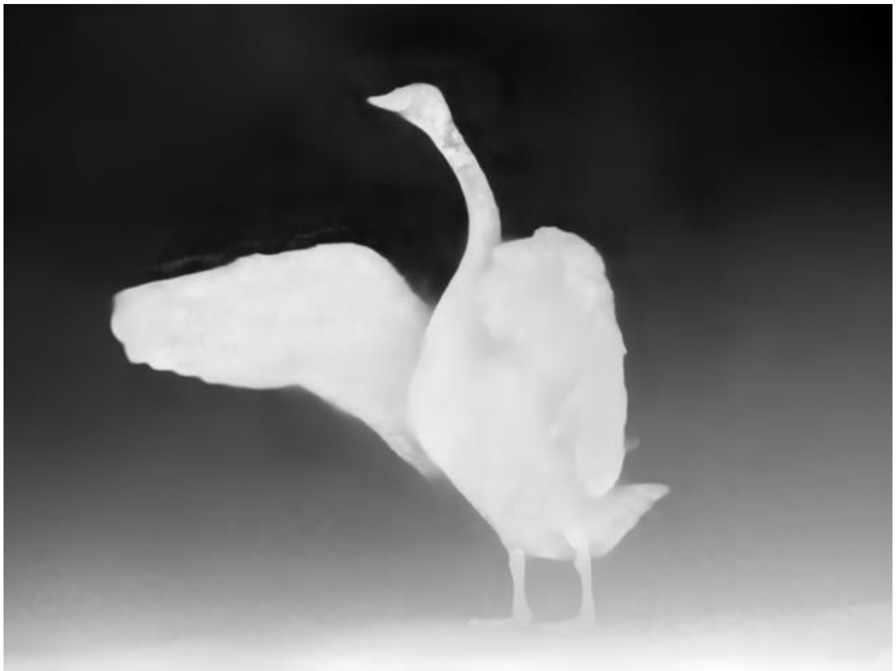


Fig. 4. Depth estimation using MiDaS v2.1

2) Depth estimation using MiDaS V2.1 + BMD:

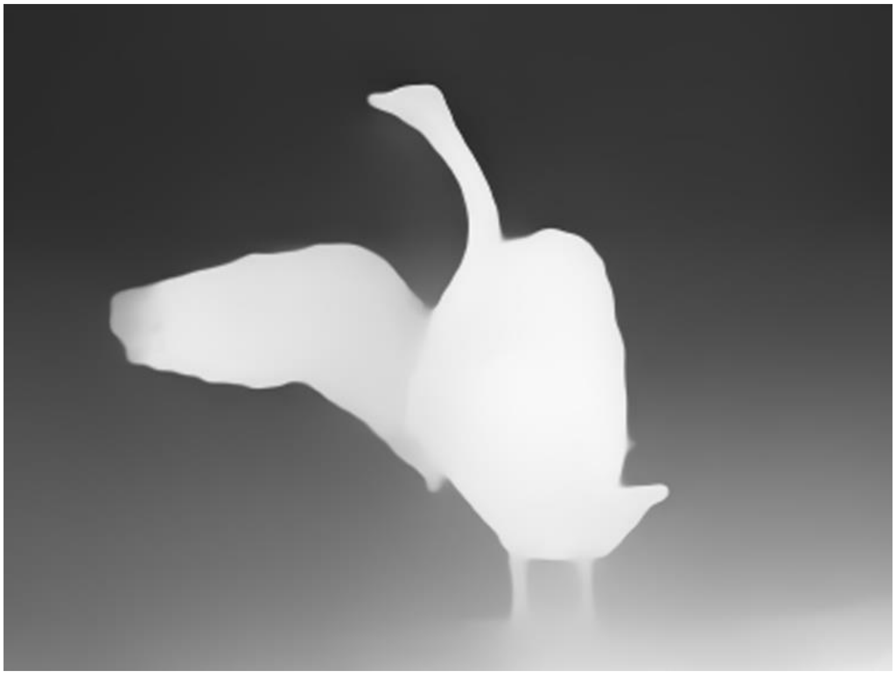


Fig. 5. Depth estimation using MiDaS v2.1 + BMD

3) Depth estimation using MiDaS v3 DPT Large + Edge Sharpening:

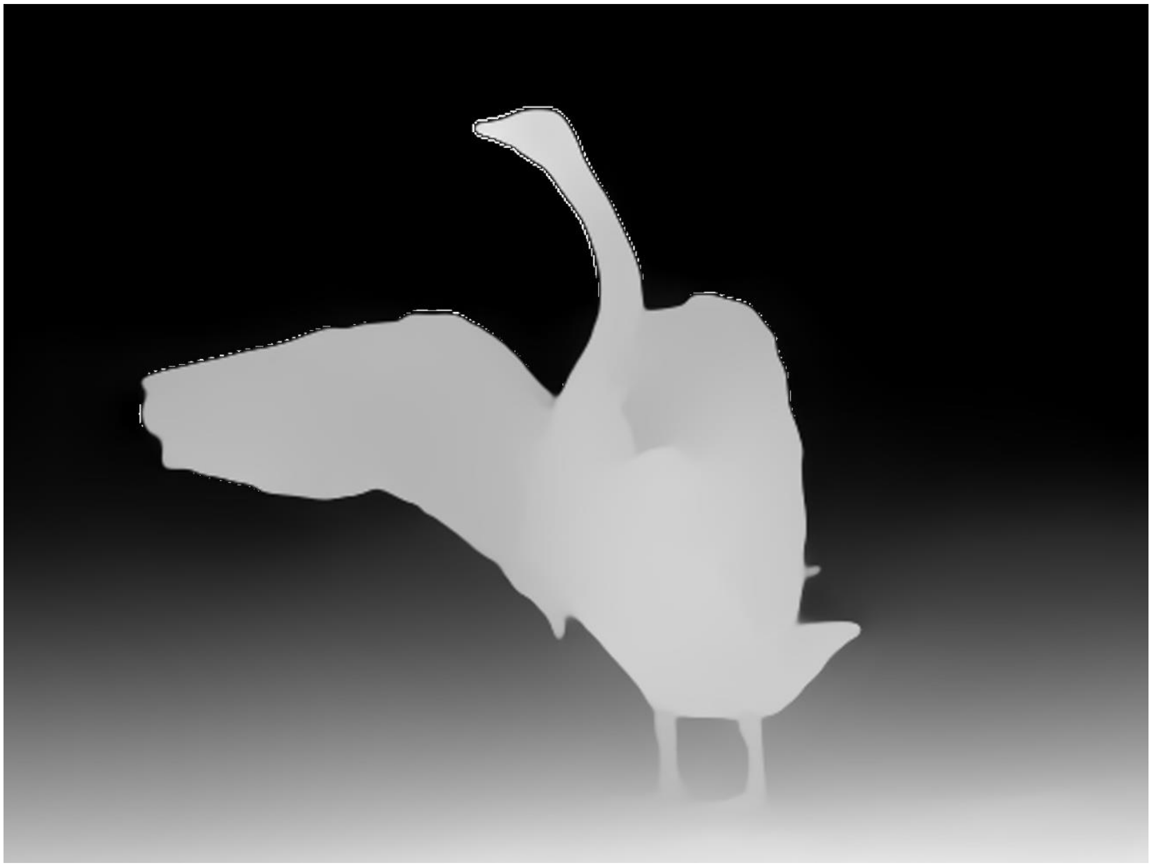


Fig. 6. Depth estimation using MiDaS v3.0 DPT Large + Bilateral Median Filter

Also, we have checked with parameter tuning of the bilateral filter which is used to sharpen the edges of the depth map. Based lined the highest performance bilateral filter.

Our project code is a modification of the previous 3D-Photo-Inpainting [2] code. We have changed the depth estimation model MiDaS v2.1 with MiDaS v3.0 DPT-Large and parameter tuning in the bilateral median filter.

CHAPTER 6

Experimental Setup

6.1 Hardware Requirement

- **Nvidia GPU:**
    - GPUs break complex problems into thousands or millions of separate tasks and work them out at once parallel.
    - That makes them ideal for graphics, where textures, lighting and **the rendering of shapes have to be done at once to keep images flying across the screen**.
    - Architecturally, the CPU is composed of just a few cores with lots of cache memory that can handle a few software threads at a time. In contrast, a GPU is composed of hundreds of cores that can handle thousands of threads simultaneously which gives high throughput.

- **VRAM 8 GB:** 8 GB RAM refers to a computer's random access memory that can temporarily store and quickly access data for efficient system performance.

- **i5 processing unit:** The i5 processor is a mid-range central processing unit (CPU) developed by Intel. It is known for its balance of performance and cost-effectiveness, making it a popular choice for many mainstream computer applications.

6.2 Software Requirement

- **Python:** Python is one of the widely used programming languages for building systems that indulge in Image Processing as well as Machine Learning. Python provides amazingly powerful libraries and tools that help us in achieving the tasks efficiently.
- **Anaconda:** Anaconda is a distribution of the Python and R programming languages for scientific computing (data science, machine learning applications, large-scale data processing, predictive analytics, etc.), that aims to simplify package management and deployment.
- **Matplotlib:** matplotlib is the plotting library which provides functions for plotting various kinds of data
- **Tensorflow:** TensorFlow is a free and open-source software library for machine learning and artificial intelligence. It can be used across a range of tasks but has a particular focus on training and inference of deep neural networks.
- **Nvidia Drivers Installed :** This package contains the NVIDIA graphics driver. A graphics or video driver is the software that enables communication between the graphics card and the operating system, games, and applications.
- **Operating System – Windows 10**

CHAPTER 7

Results

The below table shows metrics when different depth estimation models are used.

Metrics → **Depth Estimation Model ↓**	**ORB Similarity**	**SSIM Similiary**	**PSNR**	**RMSE**
Midas v2	0.7939	0.8296	0.8362	24.6507
MiDaS V2.1 + BMD	0.8444	0.8349	0.8349	24.5293
MiDaS 3 DPT Large	0.8738	0.8358	0.8358	24.2790

Table 7.1: Quantitative Comparison

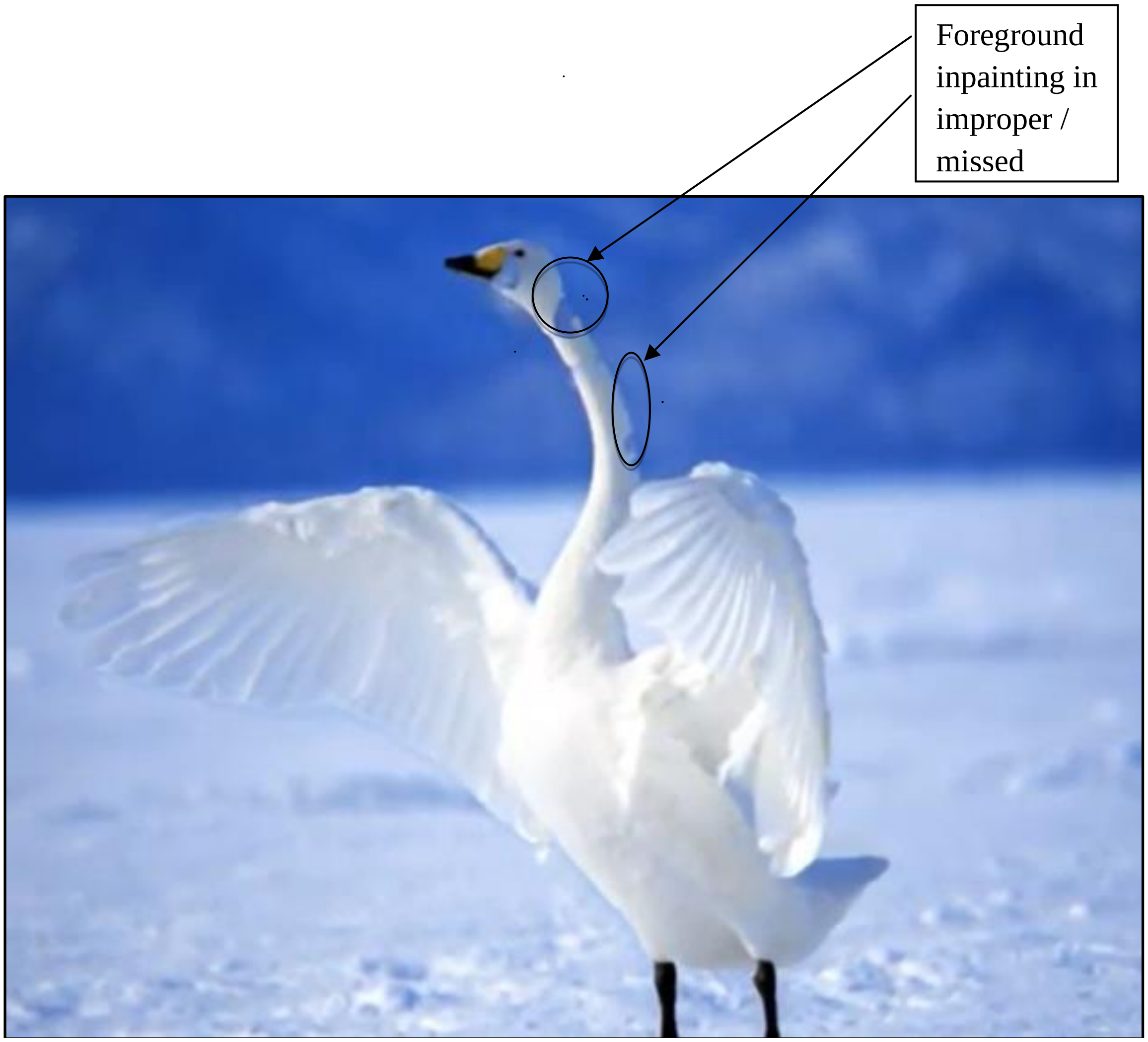
The above table shows metrics when different depth estimation models are used for 3D Photography and compared output images with the original image to be rendered. ORB Similarity and SSIM Similarity show that MiDaS V3.0 DPT Large with Bilateral Filter has good results compared to MiDaS V2.0 and MiDaS V2.1 + BMD. Also, PSNR is high and RMSE is low for MiDaS V3.0 DPT Large when compared with other models.

Please refer https://3d-photography.netlify.app/landing-page for the Demo of our results

1. Zoom-in: MiDaS 2.1

Zooming in during foreground inpainting is a common technique used to fill in missing or damaged parts of an image, especially in cases where the foreground object is partially obstructed. However, when zooming in, the edges of the object can become blurred or distorted, which can lead to errors in depth estimation and an inability to identify depth discontinuities. In turn, this can cause issues with the foreground inpainting process, as the algorithm may not be able to accurately identify which parts of the image need to be filled in and which should be left alone.

During zoom-in foreground inpainting is missed as depth estimation edges were not sharpened in turn depth discontinuity is not identified.



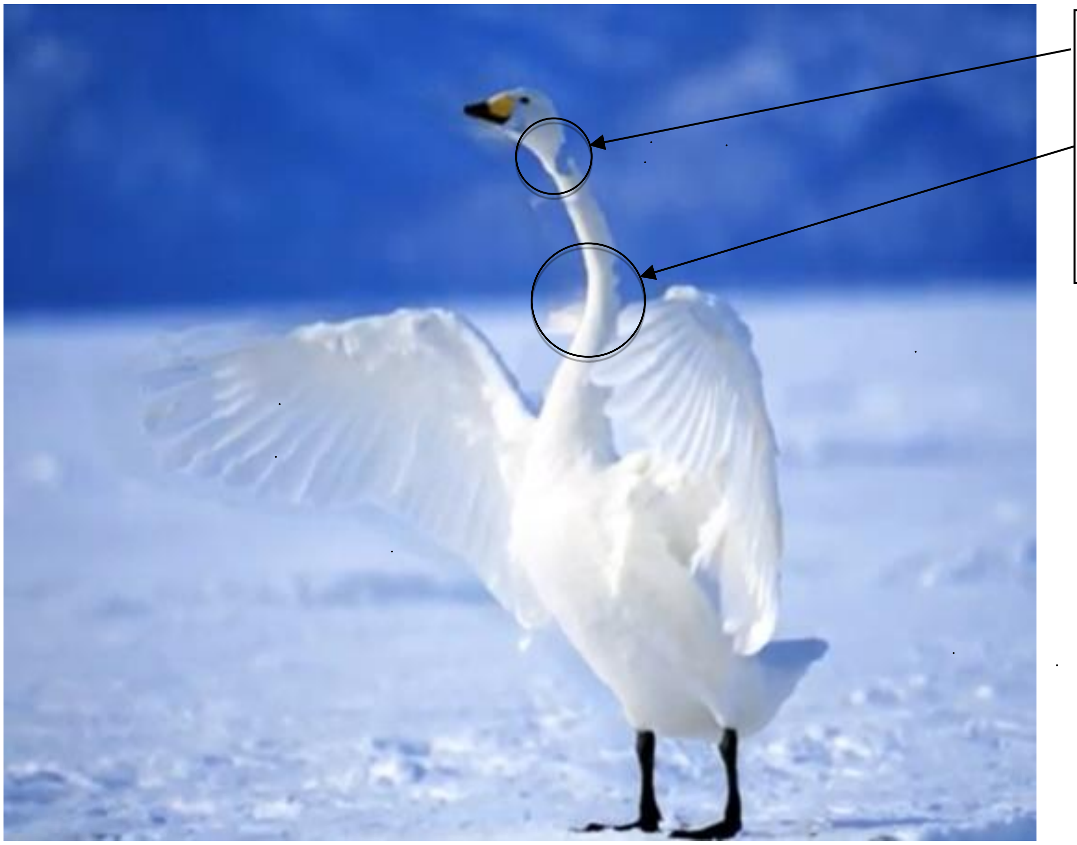
Foreground inpainting in improper / missed

Fig. 7. Rendering Snapshot – Zoom-in – Midas v2.1

2. Swing: MiDaS 2.1

Swing rendering is a technique used to create a 3D representation of an object or scene by rendering multiple images from different angles and then combining them into a single 3D image or video. However, during the rendering process, edges of the object can become blurred or distorted, which can lead to errors in depth estimation and an inability to identify depth discontinuities. In turn, this can cause issues with the inpainting process, as the algorithm may not be able to accurately identify which parts of the image need to be filled in and which should be left alone.

During swing rendering inpainting is blurred as during depth estimation edges were not sharpened in turn depth discontinuity is not identified.

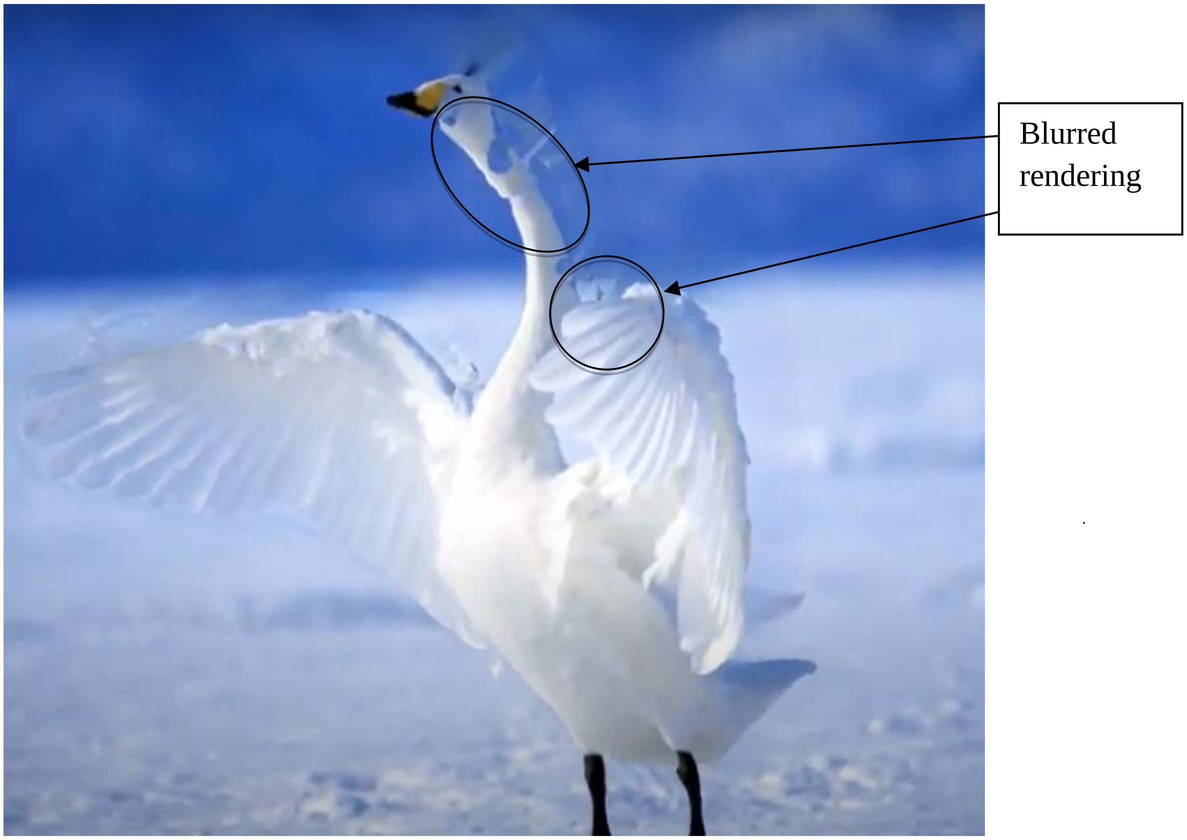


Foreground inpainting in improper / missed and blurred rendering

Fig. 8. Rendering Snapshot – Swing – Midas v2.1

3. Swing: MiDaS 2.1

During circular rendering, the foreground layer is blurred.



Blurred rendering

Fig. 9. Rendering Snapshot – Swing – Midas v2.1

4. Rendering results using MiDaS 2.1 + BMD:

MiDaS 2.1 + BMD is a combination of two depth estimation techniques: MiDaS 2.1 and Boosting Monocular Depth (BMD). MiDaS 2.1 is a deep learning-based approach to depth estimation that uses a neural network to predict depth maps from 2D images.

Here results are improved compared to MiDaS 2.1 but still showing blurred rendering in some cases.

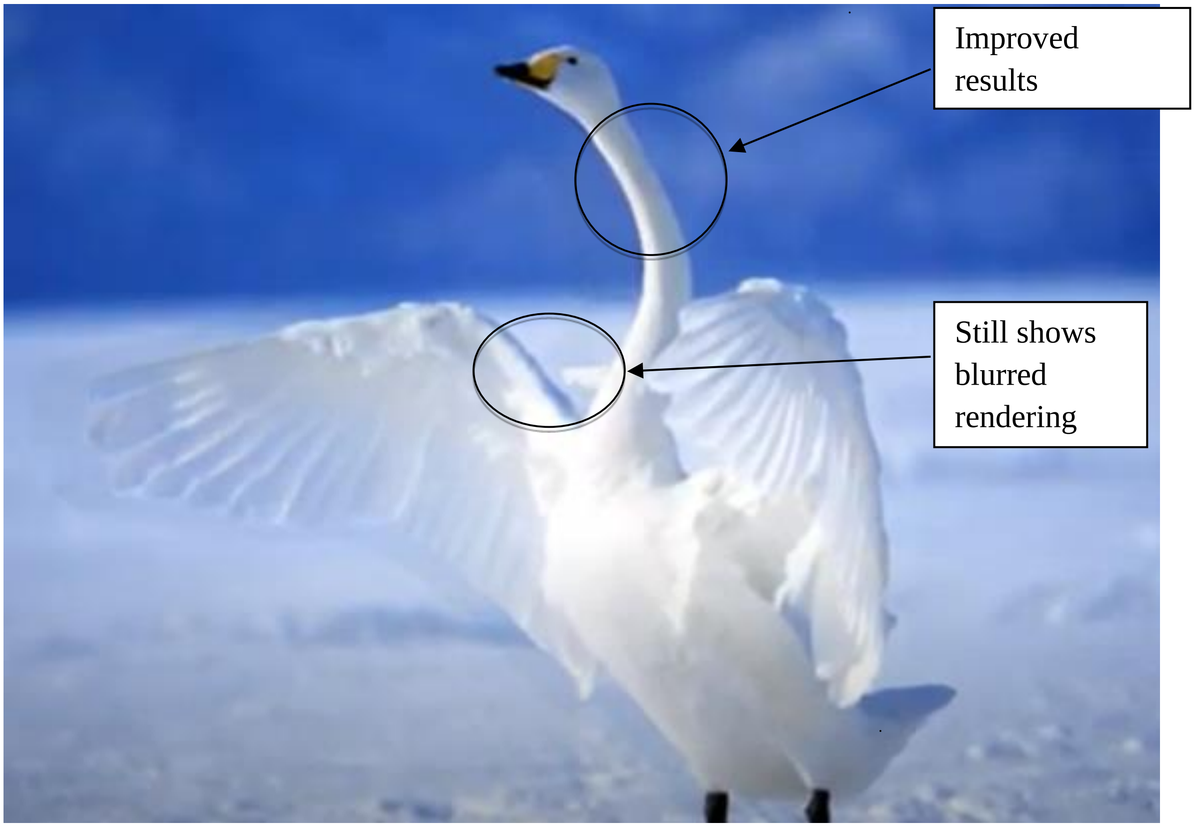


Fig. 10. Rendering Snapshot – Swing – Midas v2.1 + BMD

5. Improved Rendering results using MiDaS v3.0 DPT Large:

The snapshot below shows clear results without any blur and no missing foreground layer.

Depth Estimation Map is improved due to MiDaS 3.0 DPT large. Along with this, parameters of bilateral median filter are tuned, and the depth map image is sharpened properly which helped proper 3D rendering with proper inpainting.

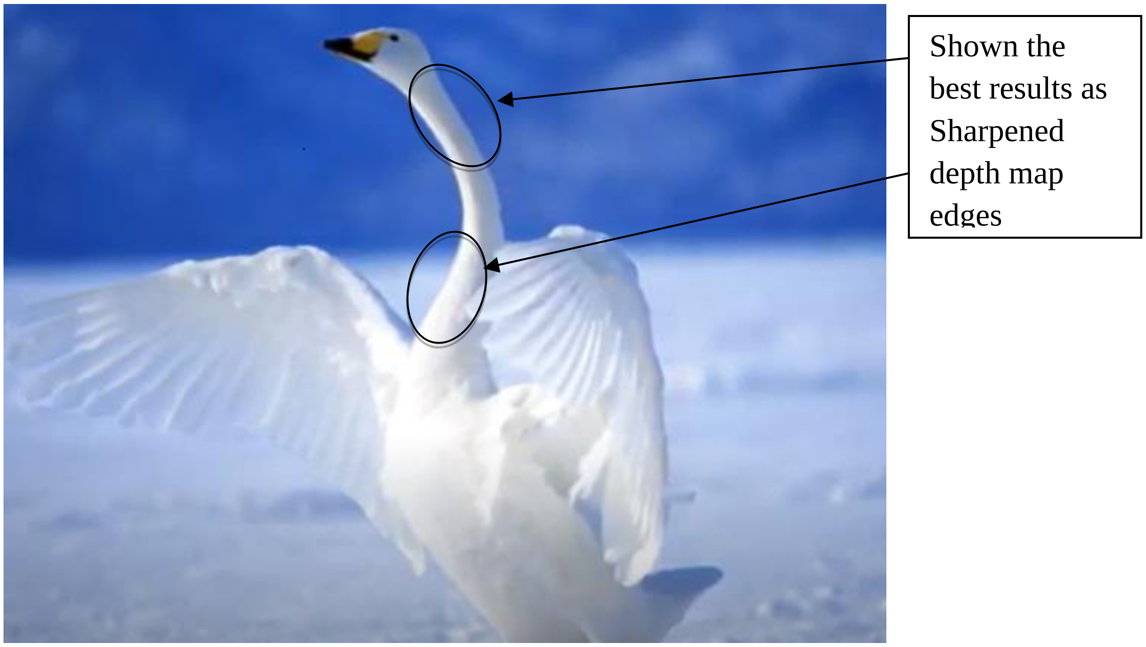
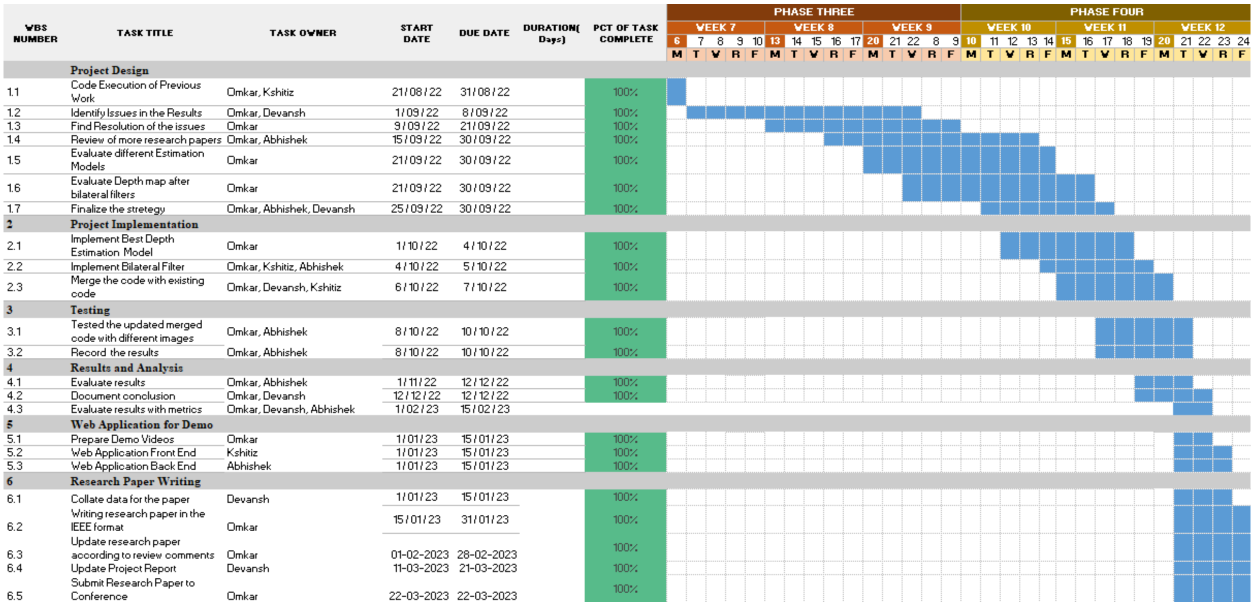


Fig. 11. Rendering Snapshot – Swing – Midas v3.0

CHAPTER 8

Project Planning (Gantt Chart)



WBS NUMBER	TASK TITLE	TASK OWNER	START DATE	DUE DATE	DURATION(Days)	PCT OF TASK COMPLETE
	Project Design					
1.1	Code Execution of Previous Work	Omkar, Kshitiz	21 / 08 / 22	31 / 08 / 22		100%
1.2	Identify Issues in the Results	Omkar, Devansh	1 / 09 / 22	8 / 09 / 22		100%
1.3	Find Resolution of the issues	Omkar	9 / 09 / 22	21 / 09 / 22		100%
1.4	Review of more research papers	Omkar, Abhishek	15 / 09 / 22	30 / 09 / 22		100%
1.5	Evaluate different Estimation Models	Omkar	21 / 09 / 22	30 / 09 / 22		100%
1.6	Evaluate Depth map after bilateral filters	Omkar	21 / 09 / 22	30 / 09 / 22		100%
1.7	Finalize the stretegy	Omkar, Abhishek, Devansh	25 / 09 / 22	30 / 09 / 22		100%
2	**Project Implementation**					
2.1	Implement Best Depth Estimation Model	Omkar	1 / 10 / 22	4 / 10 / 22		100%
2.2	Implement Bilateral Filter	Omkar, Kshitiz, Abhishek	4 / 10 / 22	5 / 10 / 22		100%
2.3	Merge the code with existing code	Omkar, Devansh, Kshitiz	6 / 10 / 22	7 / 10 / 22		100%
3	**Testing**					
3.1	Tested the updated merged code with different images	Omkar, Abhishek	8 / 10 / 22	10 / 10 / 22		100%
3.2	Record the results	Omkar, Abhishek	8 / 10 / 22	10 / 10 / 22		100%
4	**Results and Analysis**					
4.1	Evaluate results	Omkar, Abhishek	1 / 11 / 22	12 / 12 / 22		100%
4.2	Document conclusion	Omkar, Devansh	12 / 12 / 22	12 / 12 / 22		100%
4.3	Evaluate results with metrics	Omkar, Devansh, Abhishek	1 / 02 / 23	15 / 02 / 23		
5	**Web Application for Demo**					
5.1	Prepare Demo Videos	Omkar	1 / 01 / 23	15 / 01 / 23		100%
5.2	Web Application Front End	Kshitiz	1 / 01 / 23	15 / 01 / 23		100%
5.3	Web Application Back End	Abhishek	1 / 01 / 23	15 / 01 / 23		100%
6	**Research Paper Writing**					
6.1	Collate data for the paper	Devansh	1 / 01 / 23	15 / 01 / 23		100%
6.2	Writing research paper in the IEEE format	Omkar	15 / 01 / 23	31 / 01 / 23		100%
6.3	Update research paper according to review comments	Omkar	01-02-2023	28-02-2023		100%
6.4	Update Project Report	Devansh	11-03-2023	21-03-2023		100%
6.5	Submit Research Paper to Conference	Omkar	22-03-2023	22-03-2023		100%

Fig. 12. Gantt Chart for the project

CHAPTER 9

Conclusion

The proposed method of converting a single image into a 3D photo is where we have solved the issue of improper blurred rendering if an image is having low contrast, or the background is similar to the object. Identified the root cause of the issue causing improper depth estimation and edge detection. It is resolved using depth estimation pre-trained model MiDaS v3.0 DPT Large and then edges sharpening techniques (bilateral morphological filters). Results show proper 3D rendering even if an image has very low contrast.

CHAPTER 10

Future Scope

One of the main problems with the previous work is that for thin objects on the front layer of the image, such as a thin stick, the thin part of the object that is actually part of the object is rendered separately.

In the future, we hope to solve this problem.

There are some potential approaches that could help solve the problem of thin objects being rendered separately from the main object. Here are a few ideas we have researched:

1. Use a multi-layer approach: One potential solution would be to use multiple layers for the object, with each layer corresponding to a different thickness level. This could help ensure that thin parts of the object are not separated out from the rest of the object.
2. Incorporate more contextual information: Another possible solution would be to incorporate more contextual information into the rendering process. For example, if the rendering algorithm could detect that a thin object is part of a larger object, it could prioritize rendering it as part of the larger object rather than as a separate entity.

Eg. In below image suitcase handle is very thin compared to other parts of front layer.

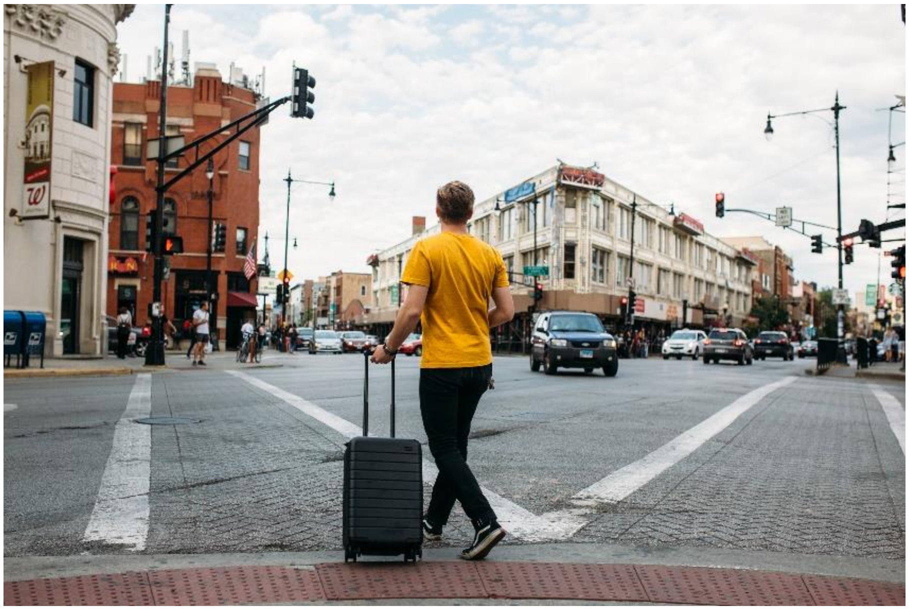
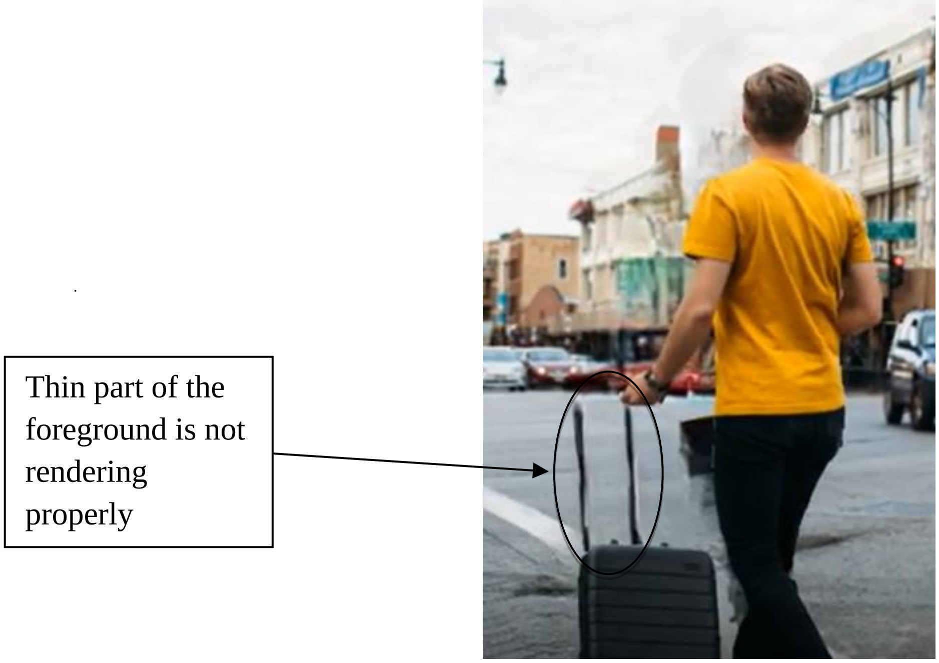


Fig. 13. Image with thin stick in the foreground

If we try to render this image, suitcase handle rendering is improper. It is miss placed during rendering and rendered separately which is part of the object.



Thin part of the foreground is not rendering properly

Fig. 14. Improper Rendering Snapshot – handle part is misplaced

References

[1] Meng-Li Shih, Shih-Yang Su, Johannes Kopf, Jia-Bin Huang; Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020

[2] Varun Jampani, Huiwen Chang, Kyle Sargent, Abhishek Kar, Richard Tucker, Michael Krainin, Dominik Kaeser, William T. Freeman, David Salesin, Brian Curless, Ce Liu. SLIDE: Single Image 3D Photography with Soft Layering and Depth-aware Inpainting Connelly Barnes, Eli Shechtman, Adam Finkelstein, and Dan B Goldman. Patchmatch: A randomized correspondence algorithm for structural image editing. In ACM Transactions on Graphics, volume 28, page 24, 2009.

[3] Shade, Jonathan, Steven J. Gortler, Li-wei He, and Richard Szeliski. 1998. Layered depth images. In Proceedings of the 25th annual conference on computer graphics and interactive techniques (SIGGRAPH 1998), July 19-24, 1998, Orlando, Flor., ed. SIGGRAPH and Michael Cohen, 231-242. New York, N.Y.: ACM Press.

[4] Nazeri, Kamyar & Ng, Eric & Joseph, Tony & Qureshi, Faisal & Ebrahimi, Mehran. (2019). EdgeConnect: Generative Image Inpainting with Adversarial Edge Learning.

[5] W. Liu, X. Chen, J. Yang and Q. Wu, "Robust Color Guided Depth Map Restoration," in IEEE Transactions on Image Processing, vol. 26, no. 1, pp. 315-327, Jan. 2017, doi: 10.1109/TIP.2016.2612826.

[6] Johnson, J., & Alahi, A. (2016). Perceptual Losses for Real-Time Style Transfer and Super-Resolution. ArXiv. /abs/1603.08155.

[7] Darabi, Soheil & Shechtman, Eli & Barnes, Connelly & Goldman, Dan & Sen, Pradeep. (2012). Image Melding: Combining Inconsistent Images using Patch-based Synthesis. ACM Transactions on Graphics - TOG. 31. 10.1145/2185520.2185578..

[8] Yu, Jiahui & Lin, Zhe & Yang, Jimei & Shen, Xiaohui & Lu, Xin. (2018). Generative Image Inpainting with Contextual Attention. 5505-5514. 10.1109/CVPR.2018.00577.

[9] Ranftl, Rene & Bochkovskiy, Alexey & Koltun, Vladlen. (2021). Vision Transformers for Dense Prediction. 12159-12168. 10.1109/ICCV48922.2021.01196.

[10] R. Ranftl, K. Lasinger, D. Hafner, K. Schindler and V. Koltun, "Towards Robust Monocular Depth Estimation: Mixing Datasets for Zero-Shot Cross-Dataset Transfer," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 44, no. 3, pp. 1623-1637, 1 March 2022, doi: 10.1109/TPAMI.2020.3019967.

[11] Qianqian Wang, Zhengqi Li, David Salesin, Noah Snavely, Brian Curless, Janne Kontkanen "3D moments from near-duplicate photos" in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2022. pp. 3906-3915

[12] S. M. H. Miangoleh, S. Dille, L. Mai, S. Paris and Y. Aksoy, "Boosting Monocular Depth Estimation Models to High-Resolution via Content-Adaptive Multi-Resolution Merging," 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 2021, pp. 9680-9689, doi: 10.1109/CVPR46437.2021.00956.

[13] Helisa Dhamo, Nassir Navab, and Federico Tombari. Object-driven multi-layer scene decomposition from a single image. In ICCV, 2019.

[14] Yin, Wei & Zhang, Jianming & Wang, Oliver & Niklaus, Simon & Mai, Long & Chen, Simon & Shen, Chunhua. (2020). Learning to Recover 3D Scene Shape from a Single Image.

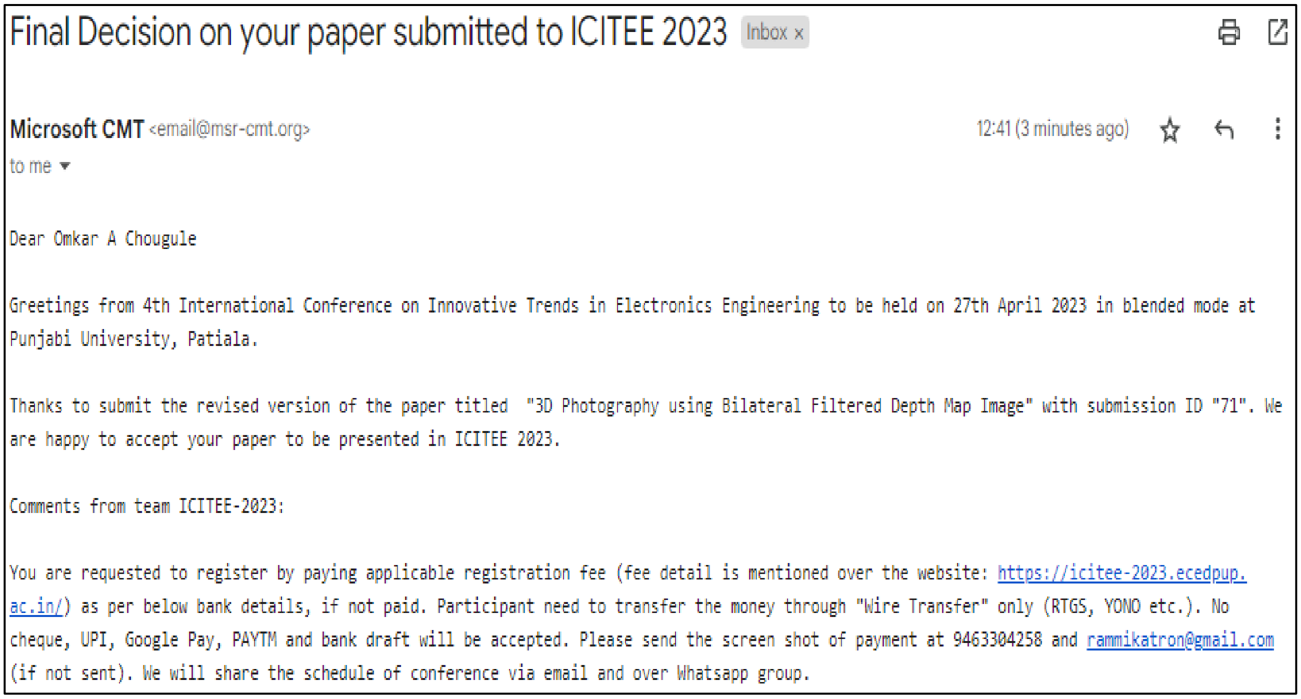
Publication

3D Photography using Bilateral Filtered Depth Map Image

Conferences:

[1] 4th International Conference on Innovative Trends in Electronics Engineering (ICITEE 2023).

Status: Paper Accepted



Final Decision on your paper submitted to ICITEE 2023

Microsoft CMT <email@msr-cmt.org>

12:41 (3 minutes ago)

Dear Omkar A Chougule

Greetings from 4th International Conference on Innovative Trends in Electronics Engineering to be held on 27th April 2023 in blended mode at Punjabi University, Patiala.

Thanks to submit the revised version of the paper titled "3D Photography using Bilateral Filtered Depth Map Image" with submission ID "71". We are happy to accept your paper to be presented in ICITEE 2023.

Comments from team ICITEE-2023:

You are requested to register by paying applicable registration fee (fee detail is mentioned over the website: https://icitee-2023.ecedpup.ac.in/) as per below bank details, if not paid. Participant need to transfer the money through "Wire Transfer" only (RTGS, YONO etc.). No cheque, UPI, Google Pay, PAYTM and bank draft will be accepted. Please send the screen shot of payment at 9463304258 and rammikatron@gmail.com (if not sent). We will share the schedule of conference via email and over Whatsapp group.

3D Photography using
Bilateral Filtered Depth Map Image

Omkar Chougule
Department of Computer Engineering
A.P. Shah Institute of Technology
Thane (M.H.), India 400615
chougule.omkar10@gmail.com

Abhishek Singh
Department of Computer Engineering
A.P. Shah Institute of Technology
Thane (M.H.), India 400615
singhabhishek07067@gmail.com

Kshitiz Jain
Department of Computer Engineering
A.P. Shah Institute of Technology
Thane (M.H.), India 400615
kshitiz.j15@gmail.com

Devansh Katheria
Department of Computer Engineering
A.P. Shah Institute of Technology
Thane (M.H.), India 400615
devanshkatheria8111@gmail.com

Prof. Sachin Malave
Department of Computer Engineering
A.P. Shah Institute of Technology
Thane (M.H.), India 400615
shmalave@apsit.edu.in

***Abstract*— 3D images can add a whole new dimension to photography. However, creating such parallax effects using conventional reconstruction and rendering techniques requires a complex setup and special hardware, which is not always feasible. By taking two clicks of the same scene, one barely off-center from the other, a 3D image can be produced. Your brain will be fooled into thinking you are seeing an image with depth by into believing you are viewing an image with depth by this small difference. Recent advanced cellphone cameras, such as a camera with two lenses, make it possible to capture depth information. Such images have extra parameter depth along with regular RGB parameters. We have studied the previous work '3D photography' and identified several problems. The first is that hair-like structures are omitted during image rendering. Second, if the image has low contrast or the background is similar to the object, the rendering will be blurry due to improper depth estimation and edge detection. And third, when the foreground layer contains thin objects that are rendered separately those are part of the object. We propose a method to transform a single image into a 3D photo, solving the second problem by generating depth map images using the pre-trained model MiDasV3.0 DPT Large and then applying edge sharpening techniques (bilateral and morphological filters). The results show correct 3D rendering with blur even if an image has low contrast or the background is similar to the foreground objects.**

Keywords* — *3D Photography, MiDaS v3.0 DPT-Large, Bilateral Median Filter

I. INTRODUCTION

3D images can add a whole new aspect to photography. When using typical rendering methods, the geometric complexity of the picture has an impact on how long it takes to render an image. As the necessary shading computations become more complex, rendering time also grows. However, creating such parallax effects with classical reconstruction and rendering techniques requires elaborate setup and special specialized hardware, which is not always feasible [1]. To develop 3D Photography the major important requirement is to know depth of 2D image. When you snap two pictures of the same scene, one marginally offset from the other, you can get a 3D image. Your brain will be tricked into trusting you are viewing an image with depth by this small difference. In recent years, advanced cellphone cameras have made it feasible to record depth information, like a camera with two lenses. Such images have extra parameter depth along with regular RGB parameters. When attempting to create a realistic view from this RGB-D image, occlusions caused by parallax must be removed. At each distinct point in the image, LDI includes a number of depth pixels [2]. A 2D array of layered depth pixels can be used in place of a 2D array of depth pixels (each of which has related depth information). A collection of depth pixels along a line of sight, arranged from front to back, are stored in a layered depth pixel. In the level of layers, there is n number of layers. First, pixels are seen the closer once; the next pixel in the layered depth pixel is the next farther, and so on [1]. The authors proposed a method of converting a single image into a 3D photo. Using a Layered Depth Image (LDI) as the underlying representation, they presented a learning-based painting model capable of synthesizing new colors and depths in the darkened region. A single RGB-D color input image is taken as input and a 3D image is generated in video format. The creation of 3D images is built on a representation in layers, where the layers include depth structures and colours that are distorted from the original view. The library employs a learning-based painting model that iteratively creates new local colour and depth content in context-sensitive space for hidden areas with the help of a layered depth picture with unambiguous pixel connectivity. Using common graphics engines gives output 3D photo after proper rendering [2].

Although rendering works well for many images, we have encountered some problems working on '3D photography' so far. First, during rendering, hair-like structures are missed from the images. Second, if the image has low contrast or the background is similar to the object, the rendering will be blurred due to incorrect depth estimation and edge detection. Third, if the front layer contains thin objects such as a thin stick, the thin part of the object that is part of the object is rendered separately.

We offer the method for developing a 3D photo with input single RGB image, where we solved the second problem by using depth map image generation with the pre-trained model MiDasV3 DPT Large and then using edge sharpening techniques (bilateral and morphological filters). The results show proper 3D rendering without blur even when an image has low contrast or the background is similar to the foreground objects.

We tried different depth estimation models, different edge sharpening methods, and filters and determined the method with the best performance.

The problem of visibility of hairline structures is solved by the previous work "SLIDE", in which foreground and background are separated and the foreground is rendered on the background [3].

II. Literature Survey

Layered Depth Image is a 2D array of pixels, in which every 1D array of depth pixels is at the line of sight of the unique pixel of an image. This 1D array starts with the front layer pixel and progresses toward the inward layers. During rendering, the view may move away from the original Layered Depth Image view and surfaces will be visible which were previously hidden. Due to data stored in a next layer of a layered depth pixel, the originally hidden regions can still be rendered. The depth image size depends on the depth complexity at the line of sight [1].

Edgemaps that look suspiciously like human sketches can be created by multiplying elements on Canny and HED edgemaps. EdgeConnect is a deep learning model for inpainting tasks. It contains main two tasks first one is Edge Builder where very short edges are removed and kept long edges which are actually edges in the object in the image and second one is an Image Completion Network which identifies and fills the gaps to complete or sharp the edges [4].

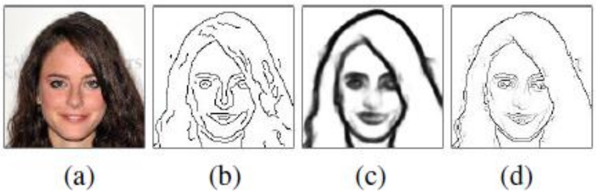


Fig. 1. Edge Connect (a) Image. (b) Canny. (c) HED. (d) Canny HED [2]

By using a new randomized algorithm for finding approximate matches between closely spaced image patches. This algorithm offers significant improvements over previous work and uses interactive image processing tools. Using random sampling, we can find some good path matches and natural coherence in the images, allowing us to quickly apply such matches to the surrounding areas. Theoretical and practical evidence of the high quality and performance of the system is provided. This method can be used for numerous tools such as image realignment, completion and reshaping, which can be used together in sophisticated image processing programs. Result: by selecting mask regions, users can interactively fill non-trivial holes. Reshaping tools allow you to quickly correct the architectural dimensions and layout of ancient monuments. This is because architecture often contains repeating patterns. This algorithm is superficially similar to the LBP and Graph Cuts algorithms commonly used to solve Markov random fields on an image grid. The difference, however, is that this algorithm is designed to optimize an energy function without a neighborhood term. This algorithm has some weaknesses. Extreme processing of an image can sometimes lead to 'ghosting' or 'feathering' artifacts, where the algorithm simply cannot get out of a large local minimum [4].

Inconsistencies between color edges in color-guided images and depth discontinuities in depth maps are the main challenges in restoring color-guided depth maps. Therefore, the recovered depth map suffers from texture copy artifacts and blurred depth discontinuities. New complex algorithms based on color-guided images and heuristically using bicubic interpolation of the input depth map have been developed to overcome this situation. But the bicubic interpolated depth map can blur depth discontinuities when the upsampling factor is large and the input depth map contains large holes and high noise. In this paper, a robust optimization technique for color-guided depth map recovery is proposed. This method is robust to the inconsistency between color edges and depth discontinuities even when we use simple guide weights. Moreover, the proposed system works well for suppressed textured copy artifacts [5].

Here the authors deal with image transformation problems in which an input image is transformed into an output image. Feed-forward Convolutional Neural Networks typically train images with a loss per pixel. In this work, they have combined the advantages of feed-forward image transformation tasks and optimization-based methods for image generation by training networks with perceptual loss functions. This is applied to style transfer so that they achieve comparable performance, dramatically improved speed compared to existing methods, and super-resolution of single images [6].

Current methods of combining two different images produce artifacts when the sources have very different textures and structures. This new process combines the transition area between two source images therefore inconsistent colors, textures, and textural properties gradually change. They offer new energy based on mixed L2/L0 standards for color and tonal transitions, enabling a smooth transition between sources without sacrificing texture sharpness. They enrich the search space of the patches with further geometric and photometric transformations [7].

Deep learning methods are used for inpainting large absent regions in an image which are shown promising results. However, these generate visually plausible, distorted structures or blurry textures contradictory with surrounding areas. This is mainly due to the inefficiency of CNN in clearly borrowing information from distant locations. So, the authors introduced a deep generative model-based approach in which it is possible to produce novel image structures and utilize surrounding image features as references during network training to generate better predictions. The model is a fully CNN, feedforward. It can process images with multiple holes at arbitrary locations and also with variable test data sizes during the testing. This approach generates higher quality inpainting results than currently available [8].

The Dense Prediction Transformer (DPT) is a dense prediction architecture based on an encoder-decoder design, where the transformer is the main computational component of the encoder. Dense Vision Transformer is an architecture that uses Vision Transformers instead of Convolutional Networks as the basis for dense forecasting tasks. The transformer skeleton processes constant and relatively high-resolution representations and has an overall reception range in each phase. These features enable the Dense Vision Transformer to provide more accurate and globally consistent predictions than fully folded networks. The architecture gave substantial results in dense forecasting tasks when large training data is available [9].

The accuracy of Monocular Depth Estimation is dependent upon extensive and varied training sets. So, authors developed a variety of datasets with unique traits and biases. They created tools that make it possible to combine different datasets while training even when their annotators are conflicting. They experimented with five different training datasets, including a novel, substantial data source: 3D movies, using these tools. To show the generalization potential of the method, evaluations conducted using databases not used for training, authors call the method a zero-shot cross-dataset transfer. The tests demonstrate that monocular depth estimation is significantly enhanced by combining data from complementary sources. The method substantially beats rival methods on a range of datasets, setting a new benchmark for monocular depth estimation. This is the next version of monocular depth estimation MiDaS v3.1. It is trained on 12 different unique datasets. It is based on five different transformers such as BEiT, Swin2, Swin, Next-ViT, LeViT [10].

Here the authors have established a new computational photography effect called 3D Moments. If photos of moving objects are taken from the same angles, i.e., a pair of nearly duplicate photos, then a video can be produced that smoothly transitions the scene from the first photo to the second. A pair of feature-based layered depth images which are augmented with scene flow are used to get this effect. This representation allows for motion transition as well as independent control of the camera viewpoint. The system generates photorealistic space-time video, restoring regions that were occluded in the initial views. They performed massive experiments that showed exceptional performance compared to baseline solutions [11].

Boosting Monocular Depth (BMD) is a dual estimation method. It gives improved depth estimation for an image. By combining depth estimates at different resolutions, we can produce depth maps using a pre-trained model with a superior level of detail by taking advantage of both depths. The authors have shown that there is a tradeoff between consistent scene structure and high-frequency detail, and merge low and high-resolution estimates to exploit this duality using a simple depth fusion network [12].

Here the authors developed a method to meet the challenge of predicting the color and depth behind the visible content of an image. The goal is to create a Layered Depth Image (LDI) from a single RGB input. This is an efficient representation in which the scene is divided into layers, including the originally hidden regions. An adaptive scheme is used for layers along with semantic encoding for better detection of partially occluded objects. Moreover, our approach is object-oriented, which especially increases the accuracy of the hidden intermediate objects. The system consists of two steps. First, each object is completed individually in terms of color and depth while estimating the structure of the scene. Second, the scene is rebuilt based on the regressed layers and the reassembled image must resemble the structure of the original input [13].

LeReS: Even though substantial innovations in monocular depth estimation, latest methods cannot be used to regain accurate 3D scenes. This is due to the unknown depth shift caused by the loss of shift-invariant reconstruction when training depth prediction with mixed data, as well as the possibly unknown camera focal length. The authors studied this problem in detail and proposed a two-step procedure that first predicts depth to an unknown scale and displacement from a single monocular image and then “3D point cloud encoders” predict missing depth displacement and focal length, thus claiming an accurate 3D scene. Moreover, the authors proposed a “normalized image-level regression loss and a normal-based geometry loss to improve depth prediction models” trained on mixed datasets. After testing the LeReS model on nine unseen datasets, they achieve the best performance in generalizing zero-shot dataset generalization [14].

New approaches combine monocular depth reticles with lacquer reticles to achieve convincing results. The disadvantage of these techniques is the use of hard depth layers, making it impossible to model complex optical details such as fine hair-like structures. The authors introduced the Soft-Layering and Inpainting that is Depth-aware (SLIDE) technique, a modular and unified single-frame 3D photography system that uses simple and effective soft layers. The strategy synthesizes complex appearance details, such as B. hair-like structures, and preserves the appearance details in new views. They also proposed a new deep training strategy for the painting module. The SLIDE approach is modular, allowing other elements such as segmentation and meshing to be used to enhance layers. SLIDE uses a simple two-layer scene decomposition and an efficient layer-depth formula that requires only one pass through the component arrays that create high-quality 3D photos. [3].

III. Problem Statement

Convert a single RGB or 2D image into a 3D photo with proper color and depth structures in regions hidden in the original view. The objective is to improve the methodology to solve the problem in the previous work; The problem is when an image has low contrast or the background is similar to the object, then the rendering is blurry due to incorrect depth estimation and edge detection. Demonstrate correct rendering without blurring in 3D rendering, even if the image has low contrast or the background is similar to the object.

IV. Methodology

System Design

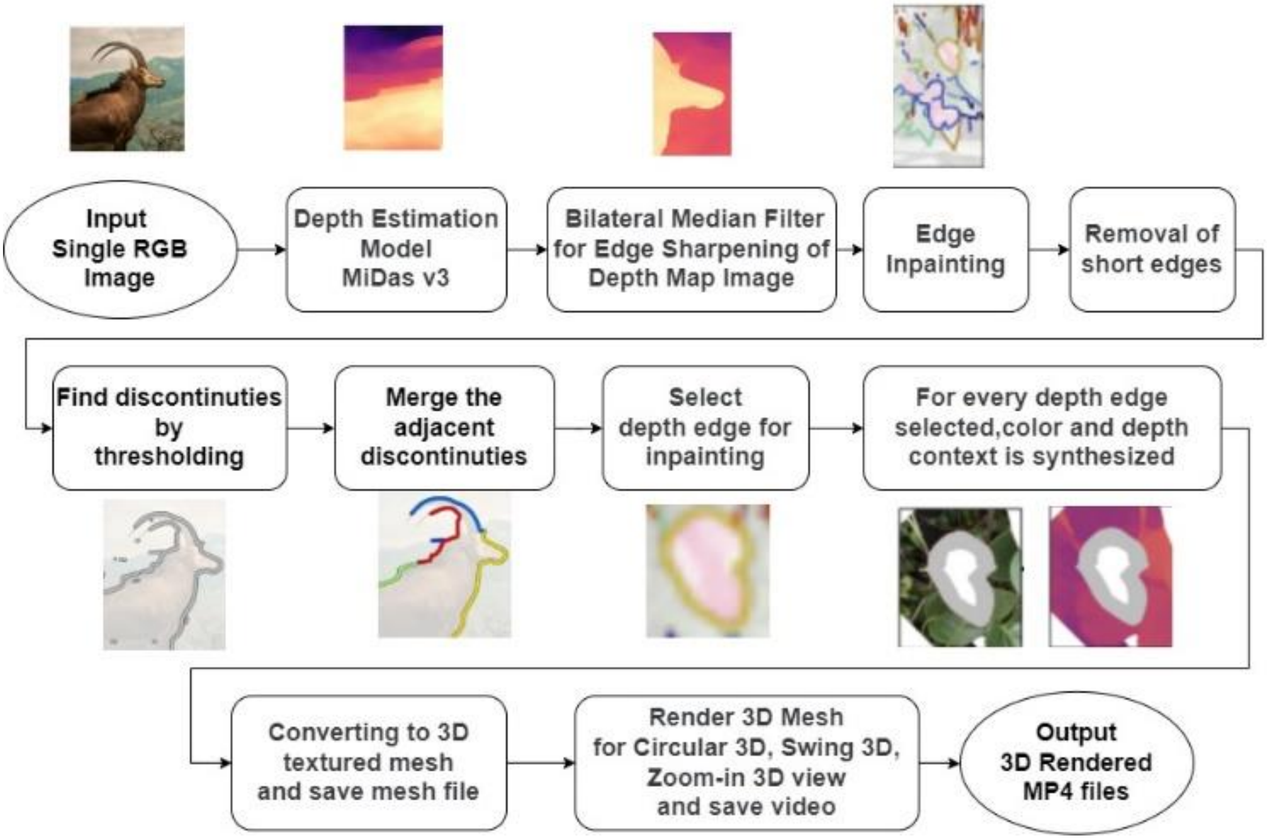

Input Single RGB Image
Depth Estimation Model MiDas v3
Bilateral Median Filter for Edge Sharpening of Depth Map Image
Edge Inpainting
Removal of short edges
Find discontinuties by thresholding
Merge the adjacent discontinuties
Select depth edge for inpainting
For every depth edge selected,color and depth context is synthesized
Converting to 3D textured mesh and save mesh file
Render 3D Mesh for Circular 3D, Swing 3D, Zoom-in 3D view and save video
Output 3D Rendered MP4 files

Fig. 2. System Design

a) Input RGB Image:

Input RGB Image: The input to this method is a single RGB-D image. The depth information can come from a mobile dual camera or can be estimated from a single image.

If an image is only an RGB image taken from a single camera then we have used a pre-trained depth estimation model MiDaS v3 DPT Large to compute depth from the input image with only color information. Thus, the proposed method applies to any image.

b) Sharpen the edges:

However, the depth map from the dual camera or depth estimation has blurry discontinuities across multiple pixels. To sharpen it, we used a bilateral median filter. This step thus ensures easy localization of the edges.

c) Short Edge Removal:

There are a few more sub-steps like thresholding and removal of short edges(<10 pixels).

d) Edge Inpainting:

Adopt the architecture provided by Edge Connect.

e) Select Depth Edges:

Select one of the edges as a subproblem.

f) Color and Depth Synthesized:

Use a standard U-Net architecture with partial convolution for the depth and color inpainting networks.

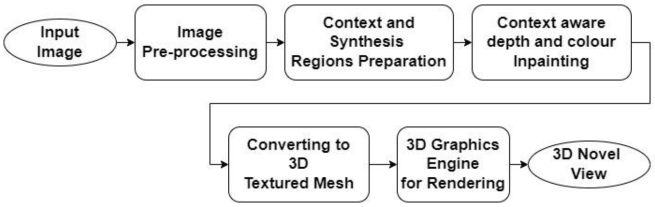
g) Develop 3D Textured Mesh:

Develop the 3D textured mesh with the help of inpainted depth and color information available added into the LDI. This is a crucial step as it helps to render quickly to get the novel view.

h) Render the 3D Mesh:

3D mesh structure of the image is used to render with the help of standard graphics engines.

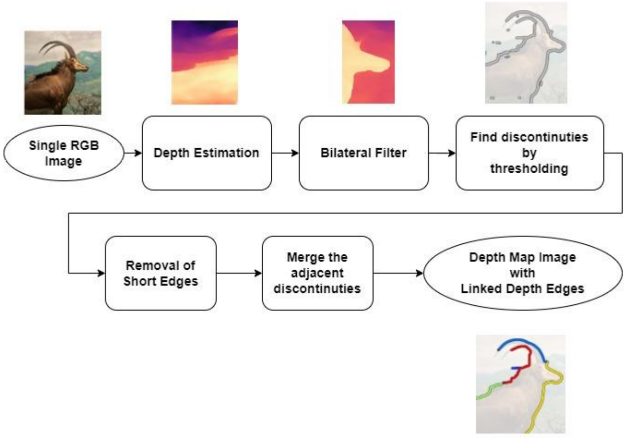
Major steps in the proposed system



Input Image
Image Pre-processing
Context and Synthesis Regions Preparation
Context aware depth and colour Inpainting
Converting to 3D Textured Mesh
3D Graphics Engine for Rendering
3D Novel View

Fig. 3. Major Steps in the proposed system

a) Image Pre-processing:



Single RGB Image
Depth Estimation
Bilateral Filter
Find discontinuities by thresholding
Removal of Short Edges
Merge the adjacent discontinuties
Depth Map Image with Linked Depth Edges

Fig. 4. Image Pre-processing

In this method, the input is a single-clicked single-camera RGB image.

First step is to Normalize the depth values between 0 and 1.

Next, convert each pixel connectivity to LDI but in a single layer.

Find the depth discontinuities since we need to inpainting the existing content but they are usually blurred by existing stereo methods (dual camera) and also depth estimation.

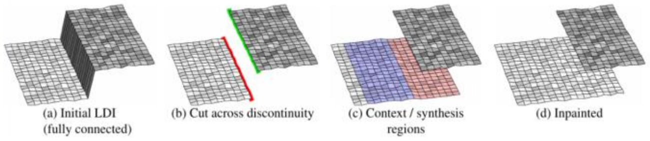
Sharpen the depth image with the help of a bilateral median filter.

Find discontinuities in the edges of depth by comparing neighboring pixels values.

Remove very short edges.

Merge adjacent discontinuities into a collection of “Linked depth edges”.

b) Context and Synthesis Regions:



(a) Initial LDI (fully connected)
(b) Cut across discontinuity
(c) Context / synthesis regions
(d) Inpainted

Fig. 5. Context / Synthesis Regions

Initial Fully connected LDI (Fig 5a): The inpainting algorithm works on current depth edges. Given one of these edges, the goal is to synthesize new color and depth values in the neighbouring occluded region.

Cut across discontinuity (Fig 5b): Next disconnect the LDI pixels at each discontinuity i.e they are missing a neighbour called silhouette pixels. As shown in Fig. 5b , a foreground silhouette marked in green and a background silhouette marked in red.

Context/synthesis regions (Fig 5c): As background silhouette information is missing, this region requires inpainting. Therefore, after this step, the synthesis region is generated to complete with new pixels marked with pink as shown in Fig. 5c.

Inpainting synthesis region (Fig 5d): Iterative flood-fill-like algorithm is used to fill the synthesis region with the colour and depth values.

c) Context-aware Color and Depth Inpainting:

As we have now the context and synthesis regions, next step is to synthesize colour and depth. Specifically, we develop colour and depth inpainting models upon the image inpainting method of previous work available [8].

d) Depth Estimation Model Selection:

We have studied and identified some issues in the previous work on ‘3D photography’. The first is that hair-like structures are missed when the image is rendered. Second, if the image has low contrast or the background is similar to the object, the rendering will be blurred due to improper depth estimation and edge detection. And third, if the foreground layer contains thin objects that are rendered separately and are part of the object.

Therefore, we have reviewed different following state-of-the-art depth estimation models:

1. MiDaS V2.1 + BMD
2. MiDaS V2 Large + BMD
3. MiDaS V2.1 Large+BMD
4. MiDaS V3.0 DPT-Large+BMD
5. MiDaS V3.0 DPT-Large

We found MiDaS v3.0 DPT-Large gives a better result of depth estimation.

Depth estimation using MiDaS v2.1:

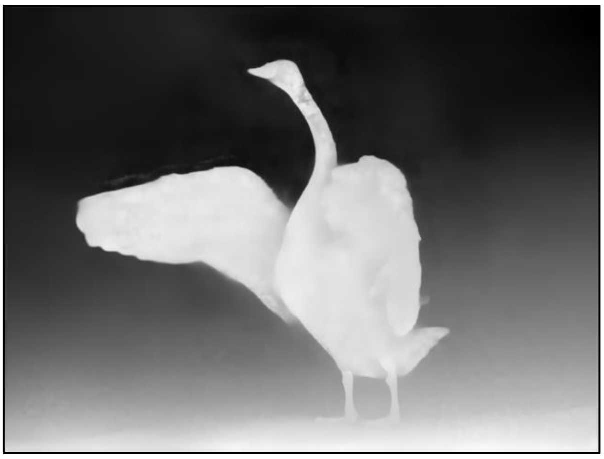


Fig. 6.Depth Estimation using MiDaS v2.1

Depth estimation using MiDaS v2.1 + BMD:

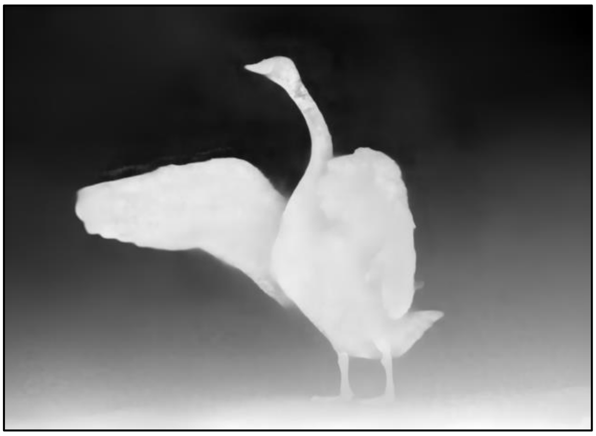


Fig. 7.Depth Estimation using MiDaS v2.1 + BMD

Depth estimation using MiDaS V3 DPT Large:

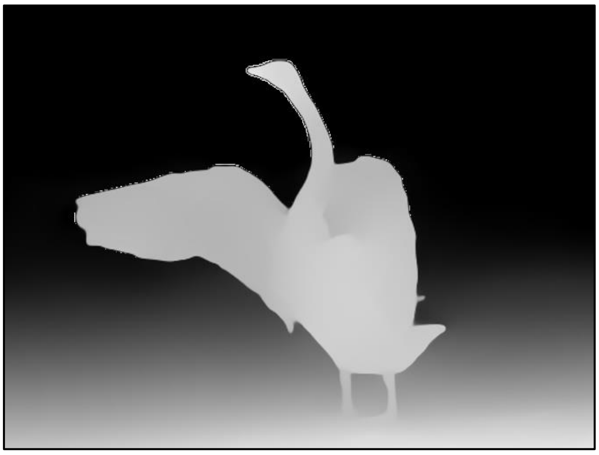


Fig. 8.Depth Estimation using MiDaS V3 DPT Large

Also, we have checked with parameter tuning of the bilateral filter which is used to sharpen the edges of the depth map. Based lined the highest performance bilateral filter.

Our project code is the modification of the previous 3D-Photo-Inpainting [3] code. We have replaced the depth estimation model MiDaS v2.1 with the MiDaS v3.0 DPT-Large and parameter tuning in the bilateral filter.

V. Results

The below table shows metrics when different depth estimation models are used for 3D Photography and compared output images with the original image to be rendered. ORB Similarity and SSIM Similarity show that MiDaS V3.0 DPT Large with Bilateral Filter has good results compared to MiDaS V2.0 and MiDaS V2.1 + BMD. Also, PSNR is high and RMSE is low for MiDaS V3.0 DPT Large when compared with other models.

TABLE I. MATRICS COMPARISON WHEN USED DIFFERENT DEPTH ESTIMATION MODELS

Metrics → Depth Estimation Model ↓	**ORB Similarity**	**SSIM Similiary**	**PSNR**	**RMSE**
MiDaS V2	0.7939	0.8296	0.8362	24.6507
MiDaS V2.1 + BMD	0.8444	0.8349	0.8349	24.5293
MiDaS V3.0 DPT Large	0.8738	0.8358	0.8358	24.2790

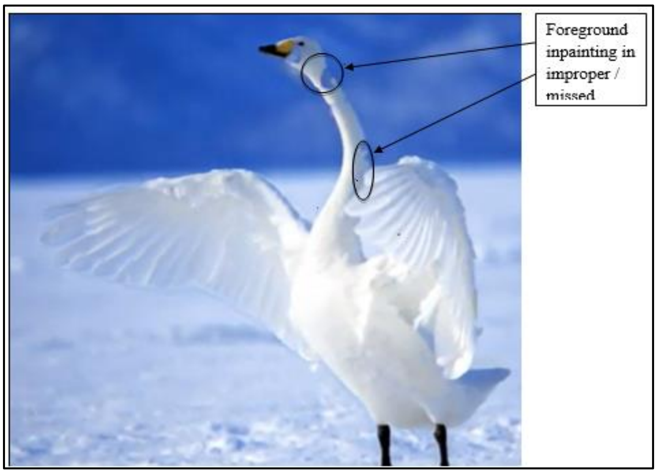
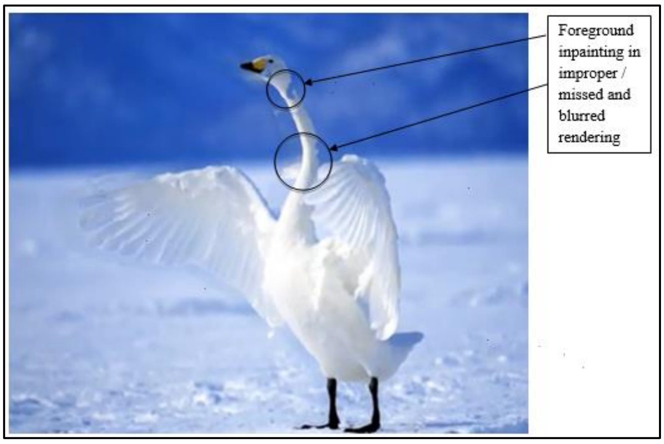
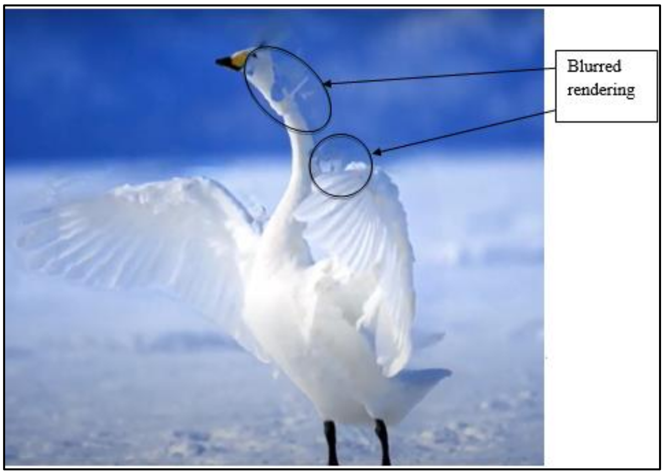


Fig. 9.Rendering Snapshot – Zoom in – MiDaS v2.1



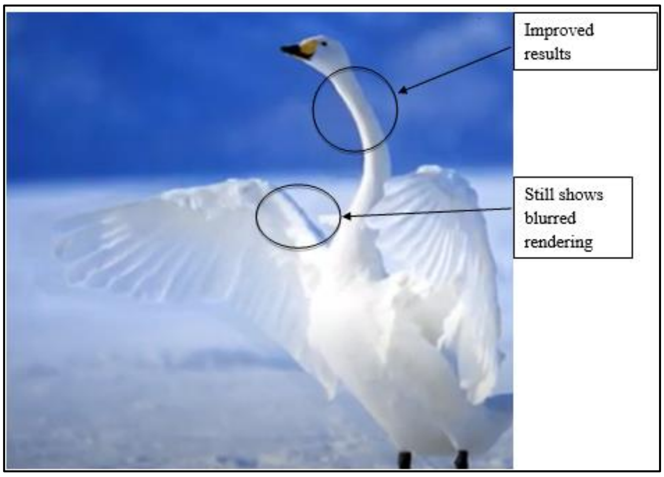
Foreground inpainting in improper / missed and blurred rendering

Fig. 10. Rendering Snapshot – Swing – Midas v2.1



Blurred rendering

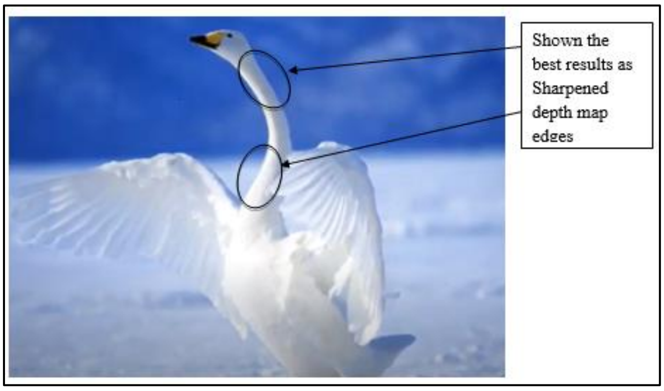
Fig. 11. Rendering Snapshot – Swing – Midas v2.1



Improved results

Still shows blurred rendering

Fig. 12. Rendering Snapshot – Swing – Midas v2.1 + BMD



Shown the best results as Sharpened depth map edges

Fig. 13. Rendering Snapshot – Swing – Midas v3.0 DPT Large

VI. Conclusion

The proposed method of converting a single image into a 3D photo is where we have solved the issue of improper blurred rendering if an image is having low contrast, or the background is similar to the object. Identified the root cause of the issue causing improper depth estimation and edge detection. It is resolved using depth estimation pre-trained model MiDaS v3.0 DPT Large and then edges sharpening techniques (bilateral morphological filters). Results show proper 3D rendering even if an image has very low contrast. Metrics such as ORB Similarity, SSIM Similarity, PSNR, and RMSE showed better results with MiDaS V3.0 DPT Large compared to other Depth Estimation Models.

VII. Future Scope

We would like to resolve the identified third issue so that if the front layer has thin objects like a thin stick, then the thin part of the object is rendered along with the object which is originally a part of the object.

References

[1] Shade, Jonathan, Steven J. Gortler, Li-wei He, and Richard Szeliski. 1998. Layered depth images. In Proceedings of the 25th annual conference on computer graphics and interactive techniques (SIGGRAPH 1998), July 19-24, 1998, Orlando, Flor., ed. SIGGRAPH and Michael Cohen, 231-242. New York, N.Y.: ACM Press.

[2] Meng-Li Shih, Shih-Yang Su, Johannes Kopf, Jia-Bin Huang; Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020

[3] Varun Jampani, Huiwen Chang, Kyle Sargent, Abhishek Kar, Richard Tucker, Michael Krainin, Dominik Kaeser, William T. Freeman, David Salesin, Brian Curless, Ce Liu. SLIDE: Single Image 3D Photography with Soft Layering and Depth-aware Inpainting Connelly Barnes, Eli Shechtman, Adam Finkelstein, and Dan B Goldman. Patchmatch: A randomized correspondence algorithm for structural image editing. In ACM Transactions on Graphics, volume 28, page 24, 2009.

[4] Nazeri, Kamyar & Ng, Eric & Joseph, Tony & Qureshi, Faisal & Ebrahimi, Mehran. (2019). EdgeConnect: Generative Image Inpainting with Adversarial Edge Learning.

[5] W. Liu, X. Chen, J. Yang and Q. Wu, "Robust Color Guided Depth Map Restoration," in IEEE Transactions on Image Processing, vol. 26, no. 1, pp. 315-327, Jan. 2017, doi: 10.1109/TIP.2016.2612826.

[6] Johnson, J., & Alahi, A. (2016). Perceptual Losses for Real-Time Style Transfer and Super-Resolution. ArXiv. /abs/1603.08155.

[7] Darabi, Soheil & Shechtman, Eli & Barnes, Connelly & Goldman, Dan & Sen, Pradeep. (2012). Image Melding: Combining Inconsistent Images using Patch-based Synthesis. ACM Transactions on Graphics - TOG. 31. 10.1145/2185520.2185578..

[8] Yu, Jiahui & Lin, Zhe & Yang, Jimei & Shen, Xiaohui & Lu, Xin. (2018). Generative Image Inpainting with Contextual Attention. 5505-5514. 10.1109/CVPR.2018.00577.

[9] Ranftl, Rene & Bochkovskiy, Alexey & Koltun, Vladlen. (2021). Vision Transformers for Dense Prediction. 12159-12168. 10.1109/ICCV48922.2021.01196.

[10] R. Ranftl, K. Lasinger, D. Hafner, K. Schindler and V. Koltun, "Towards Robust Monocular Depth Estimation: Mixing Datasets for Zero-Shot Cross-Dataset Transfer," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 44, no. 3, pp. 1623-1637, 1 March 2022, doi: 10.1109/TPAMI.2020.3019967.

[11] Qianqian Wang, Zhengqi Li, David Salesin, Noah Snavely, Brian Curless, Janne Kontkanen "3D moments from near-duplicate photos" in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2022. pp. 3906-3915

[12] S. M. H. Miangoleh, S. Dille, L. Mai, S. Paris and Y. Aksoy, "Boosting Monocular Depth Estimation Models to High-Resolution via Content-Adaptive Multi-Resolution Merging," 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 2021, pp. 9680-9689, doi: 10.1109/CVPR46437.2021.00956.

[13] Helisa Dhamo, Nassir Navab, and Federico Tombari. Object-driven multi-layer scene decomposition from a single image. In ICCV, 2019.

[14] Yin, Wei & Zhang, Jianming & Wang, Oliver & Niklaus, Simon & Mai, Long & Chen, Simon & Shen, Chunhua. (2020). Learning to Recover 3D Scene Shape from a Single Image.