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 (HOD)
Department of Computer Engineering
A.P. Shah Institute of Technology
Thane (M.H.), India 400615
sachinmalave@apsit.edu.in

***Abstract*— 3D images can add a whole new dimension to your photography. However, creating such parallax effects using conventional reconstruction and rendering techniques requires a complex setup and special hardware, which is not always feasible. A 3D image can be created by taking two shots of the same scene, one slightly offset from the other. That small difference is enough to trick your brain into believing you are seeing an image with depth. 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 found several problems. The first is that hair-like structures are omitted when rendering the image. 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 and are part of the object. We propose a method to transform a single image into a 3D photo, solving the second problem by generating using 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 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. With conventional rendering techniques, the time required to render an image increases with the scene's geometric complexity. Rendering time also increases as the required shading calculations become more sophisticated ambitions. However, creating such parallax effects with classic classical reconstruction and rendering techniques requires elaborate setup and special specialized hardware, which is not always feasible [3]. Depth is the most significant aspect of 3D photography. A 3D image is created when you take two shots of the same scene, one slightly offset from the other. This slight difference is enough to fool your brain into thinking you are seeing an image with depth. 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. When attempting to create a realistic view from this RGB-D image, occlusions caused by parallax must be removed. LDI contains hypothetically many depth pixels at each distinct point in the image [1]. Instead of a 2D array of depth pixels (a pixel with associated depth information), a 2D array of layered depth pixels can be used. A layered depth pixel stores an array of depth pixels along a line of sight, sorted from front to back. In the layered depth, there are n number of layers. First pixels are seen the closer once; the next pixel in the layered depth pixel the next farther, and so on [3]. 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 photo is generated in video format. The generation of 3D photos is based on a layered representation, where these layers contain hallucinatory colors and depth structures in areas hidden in the original view. The library uses a layered depth image (input) with explicit pixel connectivity as the underlying representation and features a learning-based painting model that iteratively synthesizes new local color and depth content in context-sensitive space for hidden regions. The end result of a 3D photo can be efficiently rendered with motion parallax using typical graphics engines [1].

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 propose a method for converting a single image into a 3D photo, 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 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 [2].

II. Literature Survey

Layered Depth Image or LDI, which potentially contains 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 an array of depth pixels along a line of sight, sorted in order from front to back. When rendering from an LDI, the requested view may move away from the original LDI view and expose surfaces that were not visible in the first layer. The previously hidden regions can still be rendered from data stored in a later layer of a layered depth pixel. There are many efficient frame-based rendering methods that can render multiple frames per second on a PC. The first method distorts the sprites with a depth that makes the surfaces smooth without the gaps found with other techniques. The second method for more general scenes is deformation using an intermediate representation called a Layered Depth Image (LDI). It is a scene with a single camera, but with multiple pixels along each line of sight. The presentation size increases proportionate to the depth of complexity observed in the scene. Additionally, LDI data is presented in a single image coordinate system [3].

Edgemaps that look suspiciously like human sketches can be created by multiplying elements on Canny and HED edgemaps. EdgeConnect, a new deep learning model for painting tasks. EdgeConnect consists of an Edge Builder and an Image Completion Network [4].

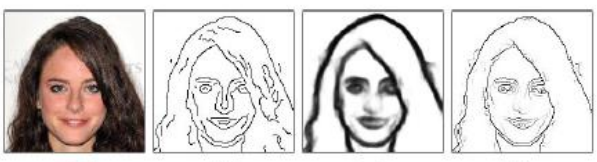

Fig. 1. Edge Connect (a) Image. (b) Canny. (c) HED. (d) Canny HED [1]

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 CNNs 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. The new process synthesizes the transition area between two source images so that inconsistent colors, textures, and textural properties gradually change from source to source. They offer a 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].

Current methods of combining two different images produce artifacts when the sources have very different textures and structures. The new process synthesizes the transition area between two source images so that inconsistent colors, textures, and textural properties gradually change from source to source. 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. They integrate image gradients with patch representation and replace the usual color averaging with a filtered Poisson solver. In many cases, our standardized method surpasses previous state-of-the-art methods [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. This architecture leads to significant improvements in dense forecasting tasks, especially when large amounts of training data are available[11].

Here the authors address image transformation problems in which an input image is transformed into an output image. Convolutional feedback neural networks typically form lossy images per pixel. They show the results of an image style transfer that trains a feedback network to solve an optimization problem in real-time. In this work, they combined the advantages of feedback image transformation tasks and optimization-based methods to generate image overtraining networks with perceptual loss functions. They applied this method to style transfer, where we get comparable efficiency, significantly improved speed over existing methods, and excellent resolution of individual photos. [8].

Current methods of combining two different images produce artifacts when the sources have very different textures and structures. The new process synthesizes the transition area between two source images so that inconsistent properties of color, texture, and structure gradually change from source to source. They deliver new energy based on mixed L2/L0 standards for color and tonal transitions, enabling smooth transitions between sources without sacrificing the sharpness of textures. They enrich the change search space with additional geometric and photometric transformations. [9].

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. [2].

III. PROBLEM STATEMENT

3D photography, which captures views of the world with a camera and uses 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 the old 2D photography: almost lifelike in virtual reality and even to some degree on normal flat displays when displayed with parallax.

Convert a single RGB-D or 2D input image into a 3D photo - a multi-layered representation for a novel view synthesis that includes hallucinated color and depth structures in regions occluded 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. Previous work on 3D photography shows blur during rendering in such cases.

IV. METHODOLOGY

System Design

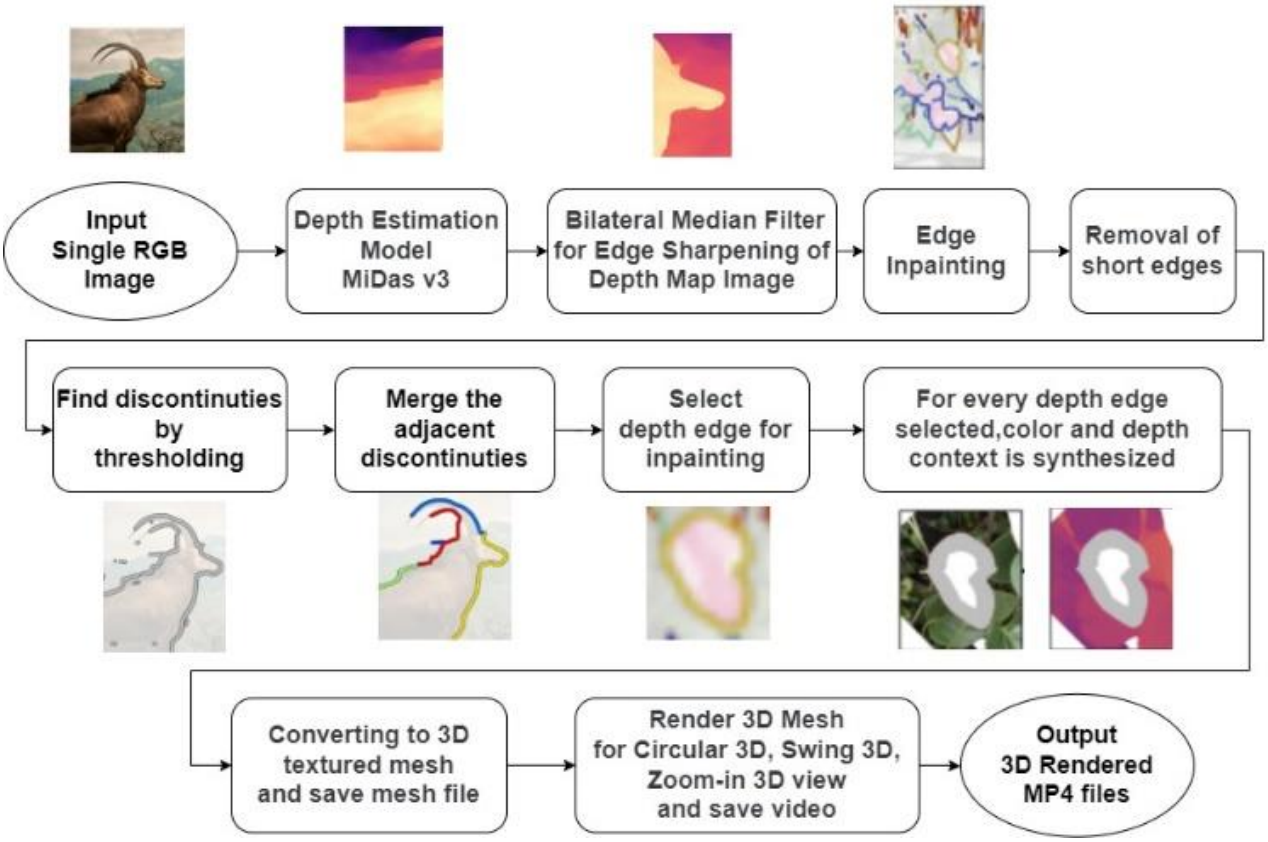

Fig. 2. System Design

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:

We adopt the architecture provided by Edge Connect.

e) Randomly select Depth Edges:

We randomly select one of the edges as a subproblem.

f) Color and Depth Synthesized:

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

g) 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 performing the per-view inference step.

h) Depth Estimation:

Render the 3D Mesh: 3D mesh structure of the image is now can easily be rendered using standard graphics engines on edge devices.

Major steps in the proposed system

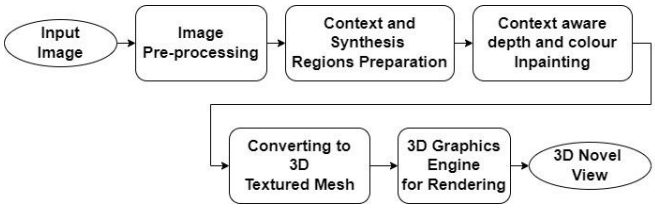

Fig. 3. Major Steps in the proposed system

a) Image Pre-processing:

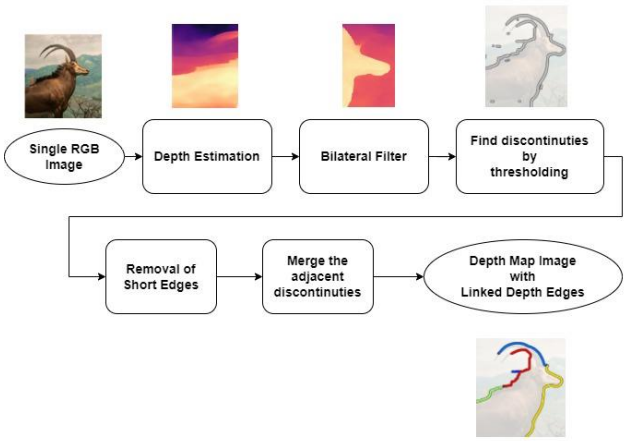

Fig. 4. Image Pre-processing

The only input to our method is a single RGB-D image.

We normalize the depth channel by mapping the min and max disparity values to 0 and 1, respectively.

We lift the image to LDI, by creating a single layer everywhere and connecting each pixel to its four cardinal neighbours.

We need to 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.

We sharpen the image, using a bilateral median filter.

After sharpening the depth map, we find discontinuities by thresholding the disparity difference between neighboring pixels.

Remove very short edges.

Merge adjacent discontinuities into a collection of "Linked depth edges".

b) ***Context and Synthesis Regions:***

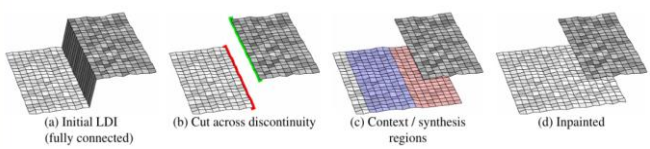

Fig. 5. Context / Synthesis Regions

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 neighbour) 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.

c) 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 [13,14,15].

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 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 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 [2] 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

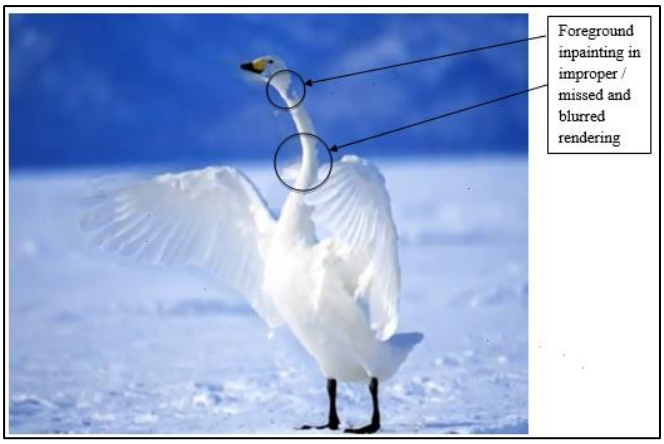

Fig. 10. Rendering Snapshot – Swing – Midas v2.1

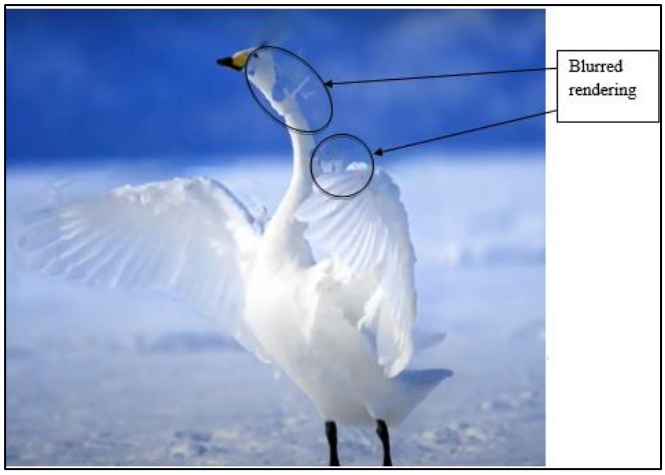

Fig. 11. Rendering Snapshot – Swing – Midas v2.1

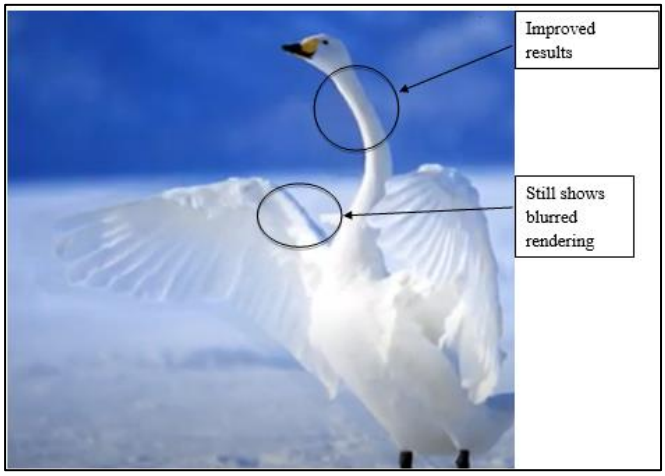

Fig. 12. Rendering Snapshot – Swing – Midas v2.1 + BMD

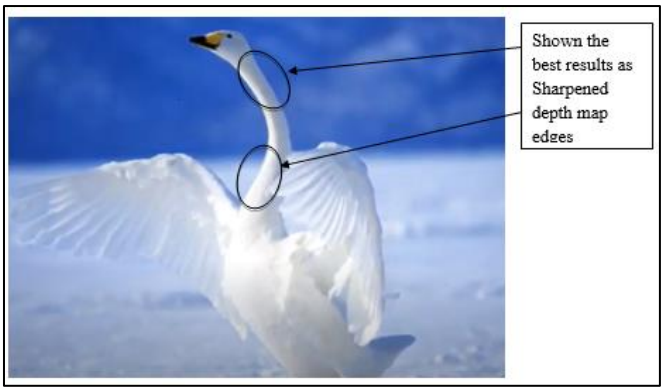

Fig. 13. Rendering Snapshot – Swing – Midas v3.0 DPT Large

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] 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] Chris Buehler, Michael Bosse, Leonard McMillan, Steven Gortler, and Michael Cohen. Unstructured lumigraph rendering. In Proceedings of the 28th annual conference on Computer graphics and interactive techniques, 2001.

[4] Weifeng Chen, Zhao Fu, Dawei Yang, and Jia Deng. Single-image depth perception in the wild. In NeurIPS, 2016

[5] Alexei A Efros and William T Freeman. Image quilting for texture synthesis and transfer. In Proceedings of the 28th annual conference on Computer graphics and interactive techniques, pages 341–346, 2001.

[6] David Eigen and Rob Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In ICCV, 2015.

[7] SM Ali Eslami, Danilo Jimenez Rezende, Frederic Besse, Fabio Viola, Ari S Morcos, Marta Garnelo, Avraham Ruderman, Andrei A Rusu, Ivo Danihelka, Karol Gregor, et al. Neural scene representation and rendering. Science, 360(6394):1204–1210, 2018.

[8] Peter Hedman, Julien Philip, True Price, Jan-Michael Frahm, George Drettakis, and Gabriel Brostow. Deep blending for free-viewpoint image-based rendering. ACM Transactions on Graphics, page 257, 2018.

[9] Jia-Bin Huang, Sing Bing Kang, Narendra Ahuja, and Johannes Kopf. Image completion using planar structure guidance. ACM Transactions on graphics, 33(4):129, 2014.

[10] Nima Khademi Kalantari, Ting-Chun Wang, and Ravi Ramamoorthi. Learning-based view synthesis for light field cameras. ACM Transactions on Graphics, 35(6):193, 2016.

[11] Ziyang Ma, Kaiming He, Yichen Wei, Jian Sun, and Enhua Wu. Constant time weighted median filtering for stereo matching and beyond. Proceedings of the 2013 IEEE International Conference on Computer Vision, pages 49–56, 2013.

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

[13] John Flynn, Ivan Neulander, James Philbin, and Noah Snavely. Deepstereo: Learning to predict new views from the world's imagery. In CVPR, 2016.

[14] Peter Hedman, Suhib Alsisan, Richard Szeliski, and Johannes Kopf. Casual 3d photography. ACM Transactions on Graphics, 36(6):234, 2017