Hufflepuff

Project: 3D Photography using Context-aware Layered Depth Inpainting (link)

Shivansh (2018102007)
Utkarsh Mishra (2018102020)
Aryan Agarwal (2018102024)
Subodh Sondkar (2018101064)

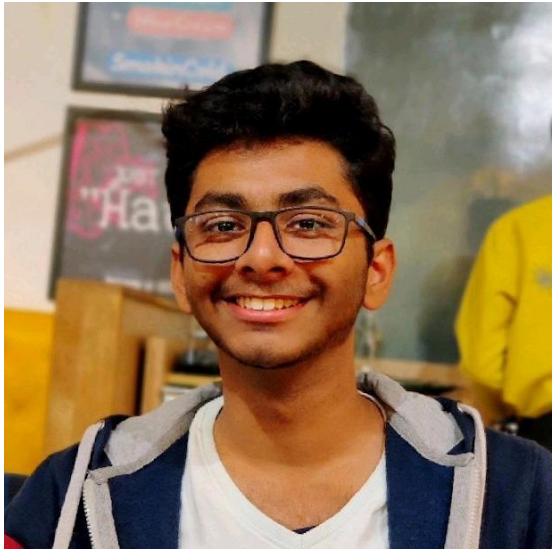
GOAL

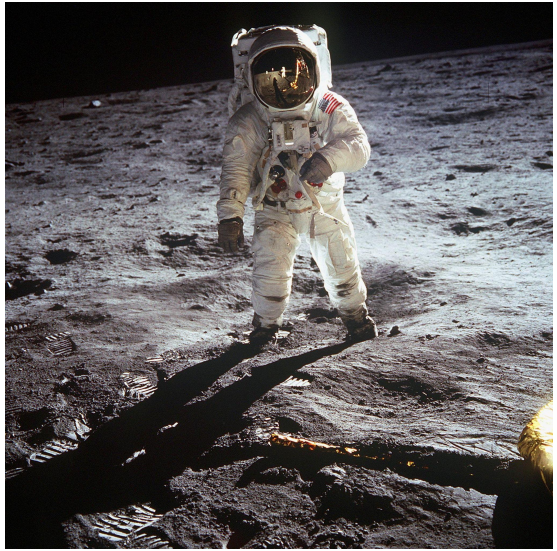
The project aims at converting a single RGB-D input image into a 3D photo - a multi-layer representation for novel view synthesis that contains hallucinated colour and depth structures in regions occluded in the original view.

Objectives

1. The project aims at converting a single RGB-D input image into a 3D photo - a multi-layer representation for novel view synthesis that contains hallucinated colour and depth structures in regions occluded in the original view.
2. It uses the Layered Depth Image with explicit pixel connectivity as underlying representation, and presents a learning-based inpainting model that iteratively synthesizes new local color-and-depth content into the occluded region in a spatial context-aware manner.
3. The resulting 3D photos can be efficiently rendered with motion parallax using standard graphics engines.
4. The effectiveness of this method is validated on a wide range of challenging everyday scenes and show fewer artifacts when compared with the state-of-the-arts.

Input Images





Input Images

Method Overview

1. Generating Depth Map from input RGB image

We use MiDaS to generate a depth map from input RGB image.

MiDaS - Modular Interactive Data Acquisition System.
Aim : Robust Monocular depth estimation

Problems in other related works:

- Various architectural innovations have been proposed to enhance prediction accuracy. These methods need ground-truth depth for training, which is commonly acquired using RGB-D cameras or LiDAR sensors.
- Others leverage existing stereo matching methods to obtain ground truth for supervision. These methods tend to work well in the specific type of scenes used to train them, but do not generalize well to unconstrained scenes, due to the limited scale and diversity of the training data.
- Garg Et al. proposed to use calibrated stereo cameras for self-supervision. While this significantly simplifies the acquisition of training data, it still does not lift the restriction to a very specific data regime. Since then, various approaches leverage self-supervision, but they either require stereo images or exploit apparent motion and are thus difficult to apply to dynamic scenes.

MiDaS uses the experimental protocol of **zero-shot cross-dataset** transfer. That is, it trains a model on certain datasets and then test its performance on other datasets that were never seen during training. The intuition is that zero-shot cross-dataset performance is a more faithful proxy of “real world” performance than training and testing on subsets of a single data collection that largely exhibit the same biases.

Training on a single dataset leads to good performance on the corresponding test split of the same dataset (same camera parameters, depth annotation, environment),but may have limited generalization capabilities to unseen data with different characteristics. Instead, it proposes to train on a collection of datasets, and demonstrate that this approach leads to strongly enhanced generalization by testing on diverse datasets that were not seen during training.

Training Datasets of MiDaS:

a) **ReDWeb** is a small,heavily curated dataset that features diverse and dynamic scenes with ground truth that was acquired with a relatively large stereo baseline.
b) **MegaDepth** is much larger, but shows predominantly static scenes. The ground truth is usually more accurate in background regions since wide-baseline multi-view stereo reconstruction was used for acquisition.
c) **WSVD** : consists of stereo videos obtained from the web and features diverse and dynamic scenes. This dataset is only available as a collection of links to the stereo videos.
d) **DIML Indoor** : is an RGB-D dataset of predominantly static indoor scenes,captured with a Kinect v2.
e) **3D movies** are also introduced as a new datasource.

Test datasets of MiDaS:

a) **DIW** : is highly diverse but provides ground truth only in the form of sparse ordinal relations.
b) **ETH3D**: features highly accurate laser-scanned ground truth on static scenes.
c) **Sintel** : features perfect ground truth for synthetic scenes.
d) **KITTI & NYU**: are commonly used datasets with characteristic biases.
e) **TUM dataset** : we use the dynamic subset that features humans in indoor environments.

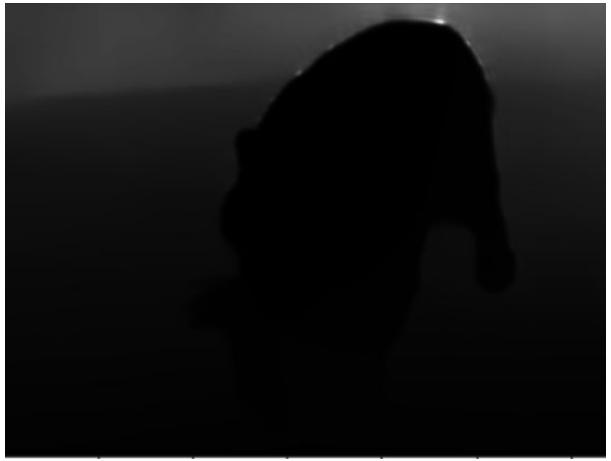
MiDaS Depth Map Results

MiDaS Depth Map Results

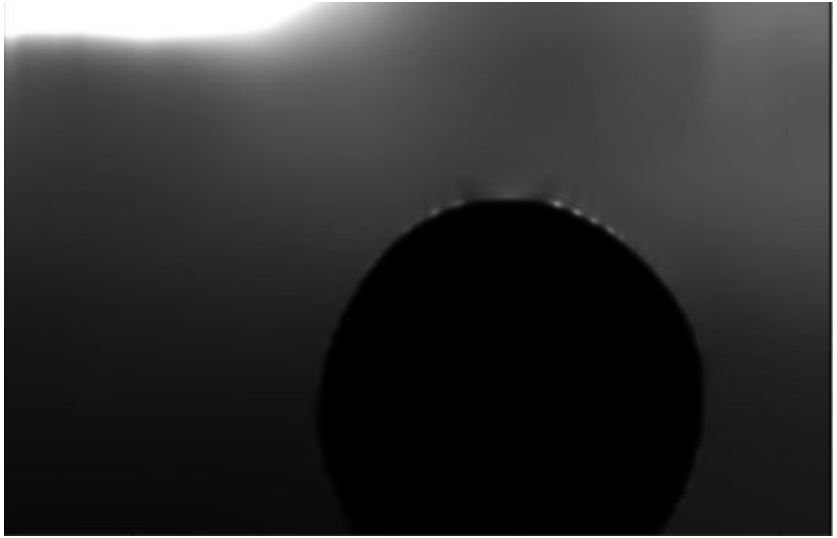
Method Overview

2. Normalizing Depth Channel of the RGB-D image

We normalize the depth map that we computed for the input image. This is done by mapping the maximum and minimum disparity values to 1 and 0 respectively.



Normalised Depth Map Results



Method Overview

3. Lifting Normalized RGB-D image onto LDI (Layered Depth Image)

An LDI is similar to a regular 4-connected image, except at every position in the pixel lattice it can hold any number of pixels, from zero to many. Each LDI pixel stores a color and a depth value. Each pixel stores pointers to either zero or at most one direct neighbor in each of the four cardinal directions (left, right, top, bottom). LDI pixels are 4-connected like normal image pixels within smooth regions, but do not have neighbors across depth discontinuities.

So this creates a graph for the image where the nodes are the pixels and edges denote connection between neighboring pixels.

It is not possible to show the LDI as it is like a mesh or a graph and not an image.

Method Overview

4. Sharpening Depth Map using Bilateral Median Filter

We sharpen the depth map, or in other words, sharpen the depth values of LDI pixels in LDI, using bilateral median filter. This process is done to avoid blurring of depth discontinuities across multiple pixels as we are interested in depth discontinuities. We have used a bilateral median filter with the following properties:

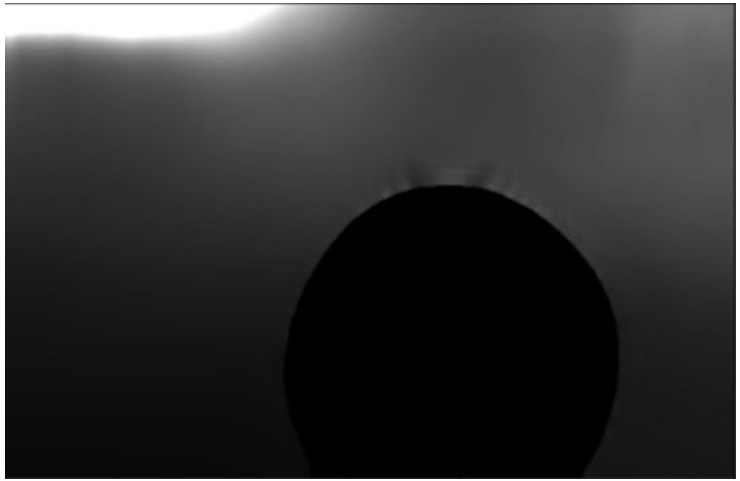
a) Window size = 7 x 7

b) $\sigma_{spatial} = 4.0$

c) $\sigma_{intensity} = 0.5$

Filtered Depth Map Results

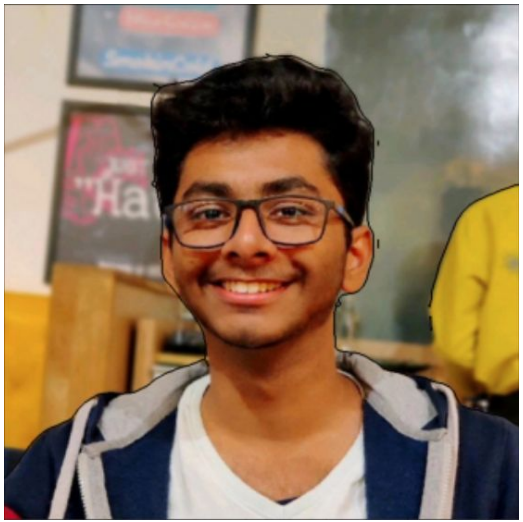
Filtered Depth Map Results

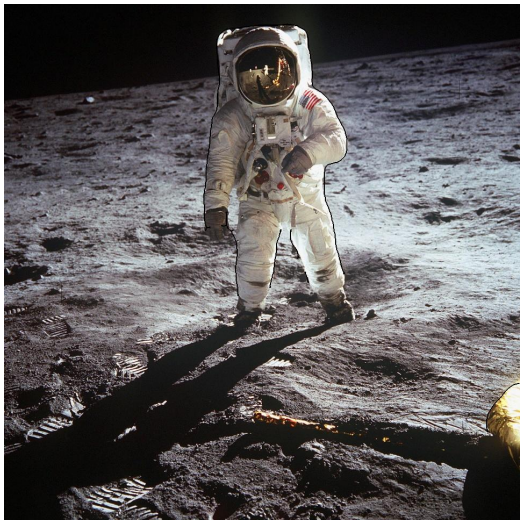


Method Overview

5. Finding Depth Discontinuities

We find depth discontinuities by finding disparity difference between neighboring pixels. This could lead to many spurious responses. To avoid this, we use connected component analysis to merge adjacent collection of depth discontinuities into a collection of linked depth edges.



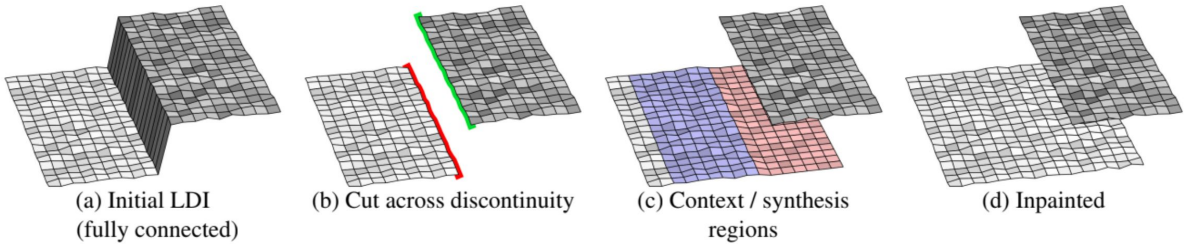


Depth Discontinuities Results

Method Overview

6. Finding Context and Synthesis Regions

We disconnect pixels across depth discontinuities and call them silhouette pixels. The green pixels denote foreground silhouette and the red pixels denote background silhouette in figure (b). We generate synthesis region (red region in figure (c)). We initialise some color and depth values for synthesis region using iterative flood-fill like algorithm. Using a similar algorithm, context region (blue region in figure(c)) is generated. As we are working with LDI, we don't cross the silhouette pixels to expand context and synthesis regions.



Context and Synthesis Regions Results

NOTE : Context and Synthesis Regions are found out in LDI (graph), thus we have not shown those regions in the form of an image. We have shown a sample for one of our input images, how it's context and synthesis regions would look like (after looking at LDI after generation of context and synthesis regions).

Method Overview

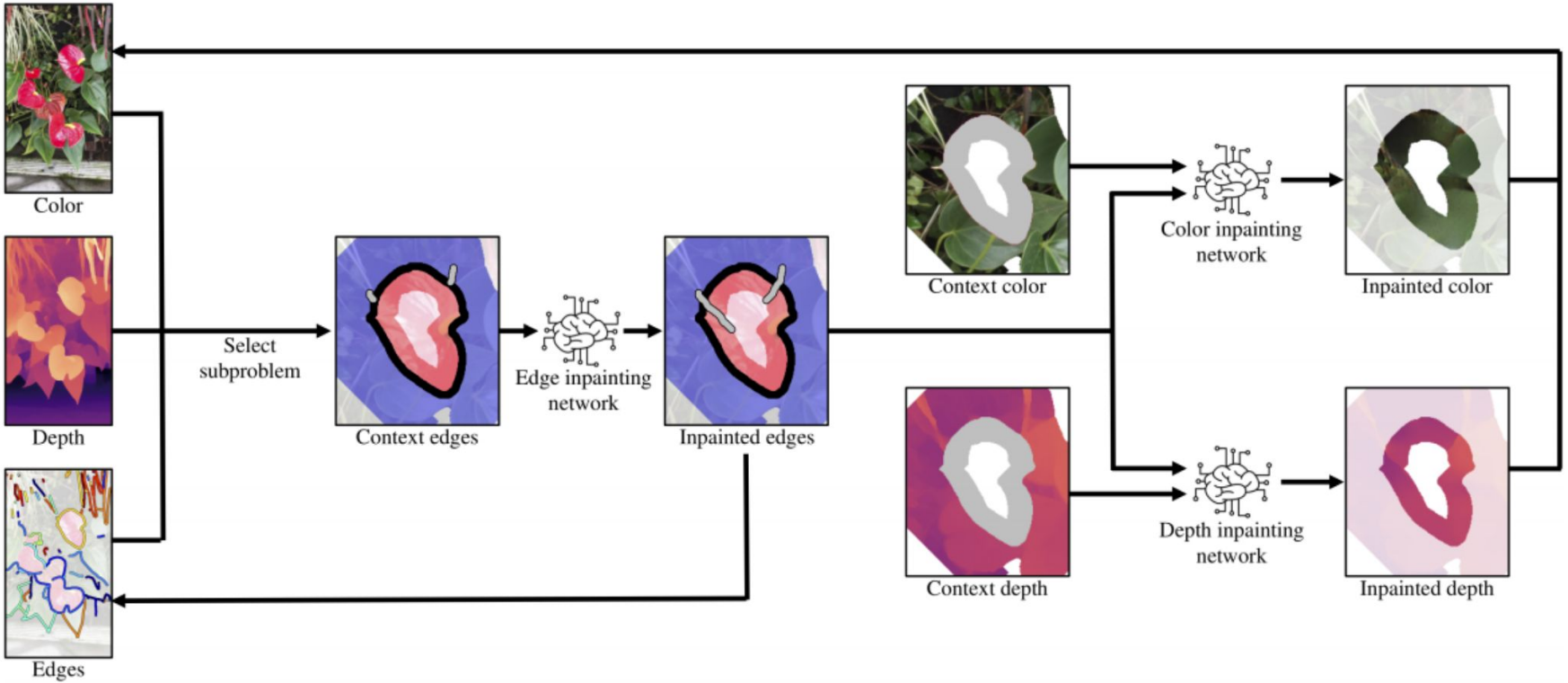
7. Context-aware Color and Depth inpainting

We use context and synthesis regions to generate color and depth values. We use three neural network models for this process :

a) Edge inpainting network
b) Color inpainting network
c) Depth inpainting network

Using context edges, the edge inpainting network predicts depth edges in synthesis regions. After generating inpainted edges, we know something about the structure of synthesis regions. We now perform color and depth inpainting using the other 2 networks. This would give us good values for synthesis regions, but the 3D effect to the photo is generated using further steps. Thus there is no specific output to be shown here.

Context-aware Colour and Depth inpainting using Edge-guided inpainting

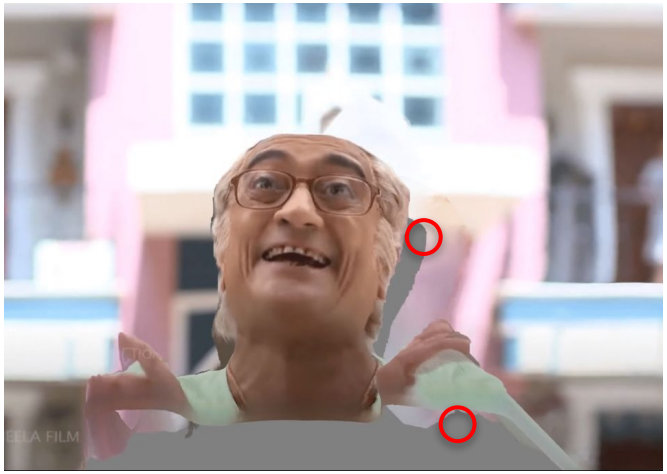


Method Overview

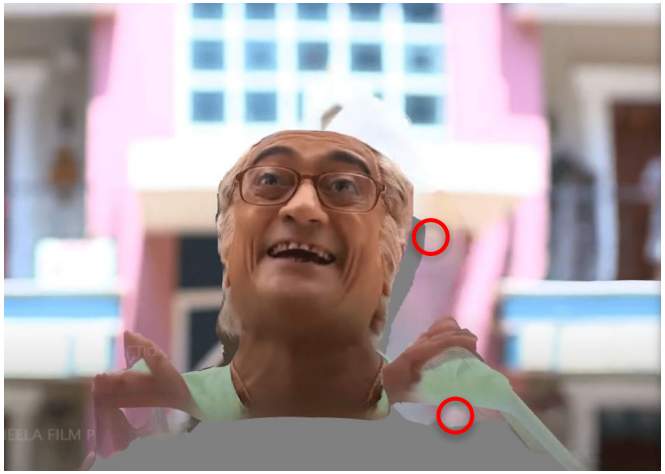
8. Multi-layer inpainting

In some complex images, running the inpainting model once does not give accurate results. We run our inpainting model once again to improve results. This fills the missing layers and gives better results.

Depth Inpainting once



Depth Inpainting twice



Method Overview

9. 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. Consequently, the 3D representation produced by this algorithm can easily be rendered using standard graphics engines on edge devices.

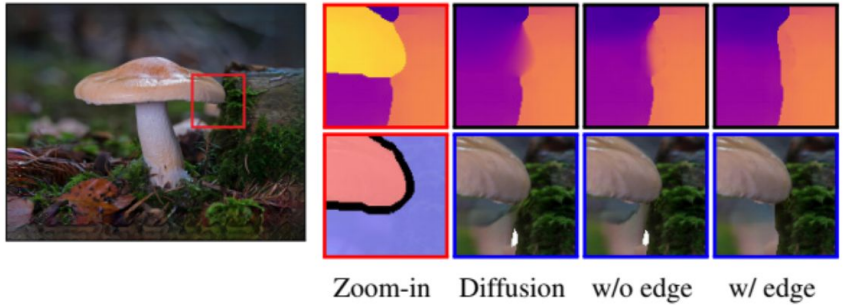
Then we create a video using a specified trajectory to realise the 3D representation.

Final 3D videos

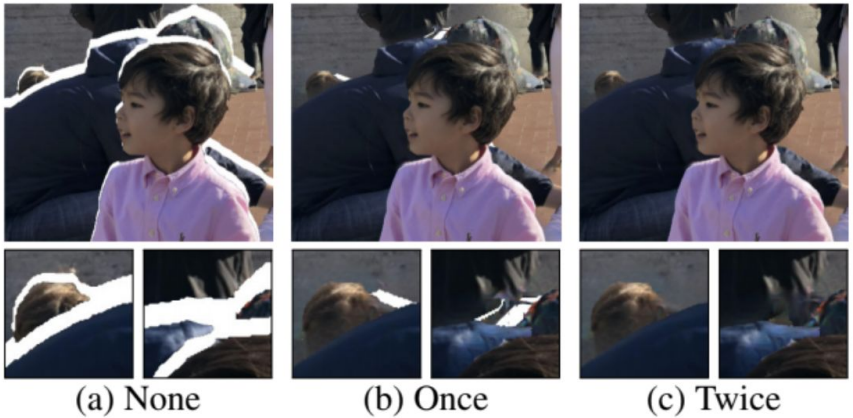
Final 3D videos

Observations in the reference paper

Benefit of Edge-Guided Depth Inpainting:



Multi-layer inpainting:



THANK YOU!