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

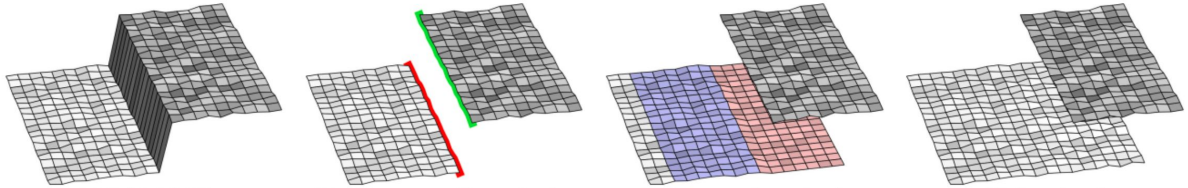
1. The project aims at converting a single RGB-D input image into a 3D photo - a multi-layer representation for novel view synthesis that contains hallucinated colour and depth structures in regions occluded in the original view.
2. It uses the Layered Depth Image with explicit pixel connectivity as 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 our method is validated on a wide range of challenging everyday scenes and show fewer artifacts when compared with the state-of-the-arts.

Method Overview

1. Generating a Layered-Depth-Image and preprocessing the image
    a. Normalizing the depth channel
    b. Lifting the image onto an LDI
    c. Finding depth-discontinuities
2. Finding Context and Synthesis Regions
    a. Generation of Synthesis Region (a contiguous region of new pixels)
        i. Using flood-fill like algorithm
        ii. Iterative expansion
        iii. Synthesis Region remains in the occluded part of the image
    b. Generation of Context Region
        i. Using flood-fill like algorithm
        ii. Select LDI pixels and follow connection links
        iii. Halts at silhouettes

Method Overview

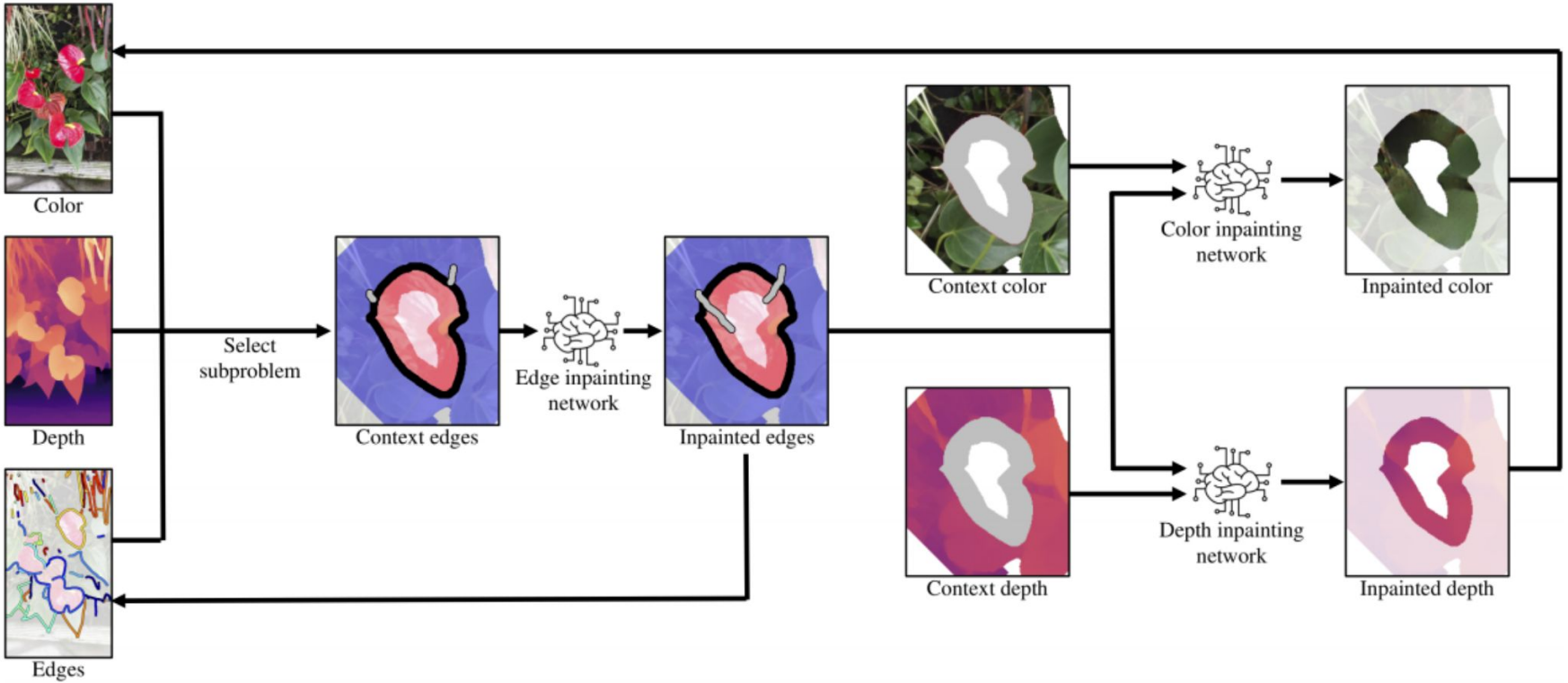
3. Context aware color and depth inpainting



a. Heuristic based approach using standard network architectures designed for images.
b. Independent Colour and Agemap Inpainting
c. Since independent, the inpainted depth-image might not be well-aligned with the inpainted colour. Thus, we break down inpainting task into sub-networks:
    i. Edge inpainting network
    ii. Color inpainting network
    iii. Depth inpainting network
d. Multi-layer Inpainting to fill up discontinuities

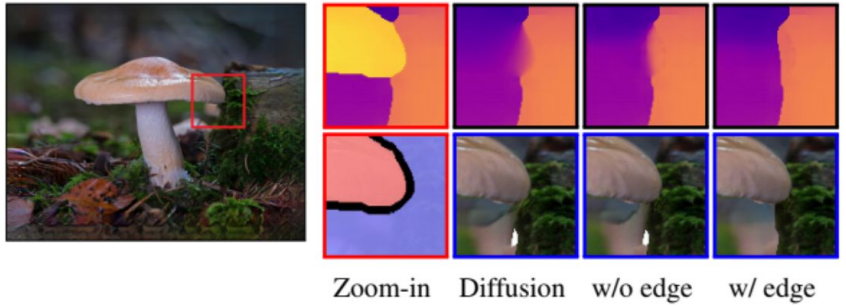
4. 3D textured mesh generation by integrating all inpainted values to original LDI.

Context-aware colour and depth inpainting overview

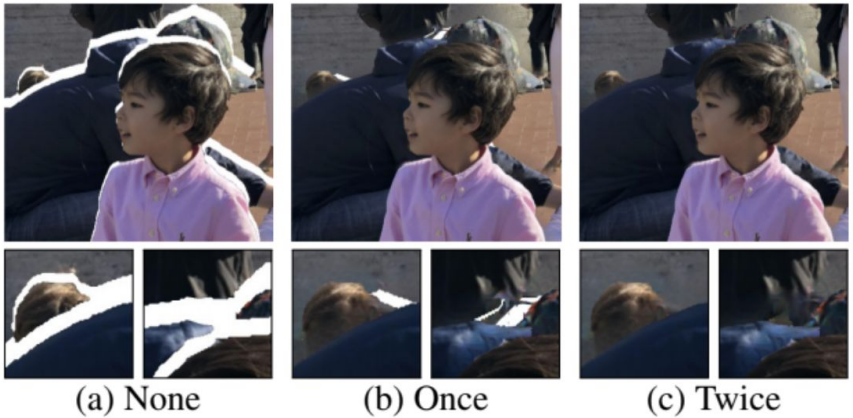


Method Overview (cont..)

Benefit of Edge-Guided Depth Inpainting:



Multi-layer inpainting:



Timeline

Mid-Eval Deliverables:

1. Generating a Layered-Depth-Image and preprocessing the image
2. Finding Context and Synthesis Regions

Final Eval Deliverables:

1. Context aware color and depth inpainting
2. 3D textured mesh generation by integrating all inpainted values to original LDI.
3. Testing and Final output generation

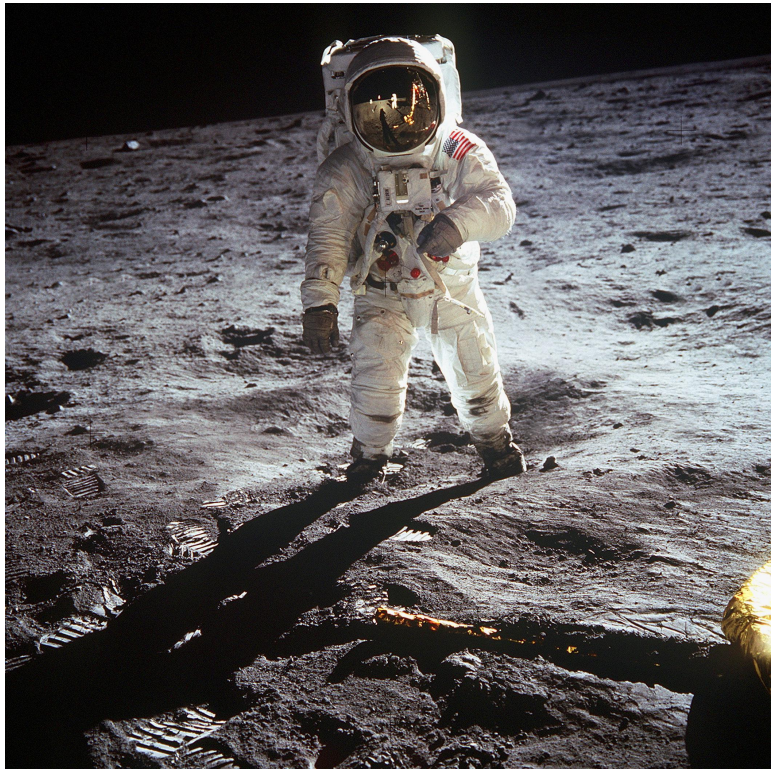
Layered-Depth Image (LDI) and Preprocessing

1. We have used MiDaS to get LDI image from the input image.
2. We then normalized the depth-map and added blur to smoothen the discontinuities in the depth-map.
3. We also applied bilateral median filter to sharpen the depth-map.
4. Then we used a threshold to find the discontinuities.
5. We merged adjacent discontinuities using connected component analysis and removed all the short segments to get the final output

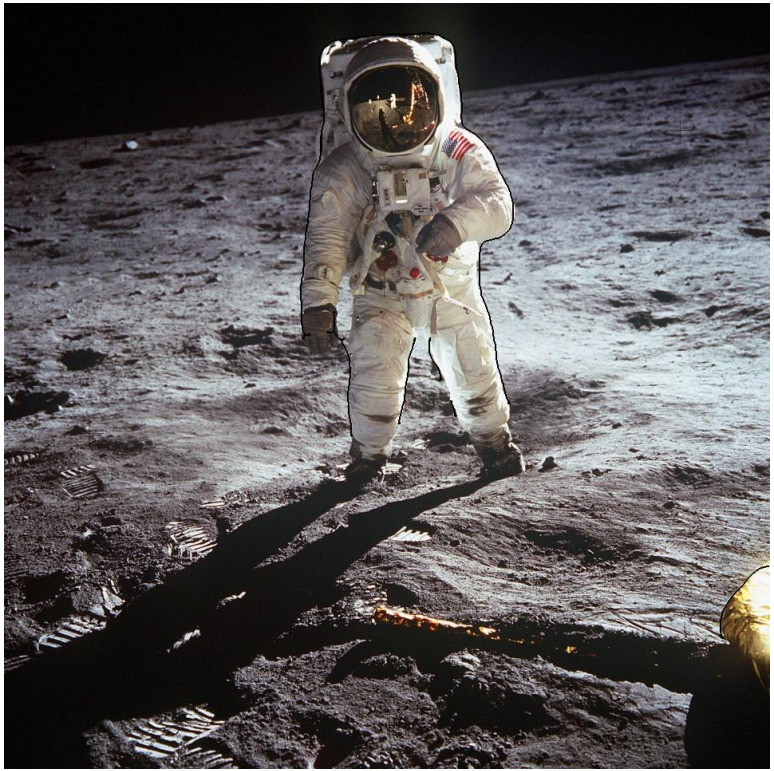
Input

LDI and pre-processed image

Input



LDI and pre-processed image



Input

LDI and pre-processed image

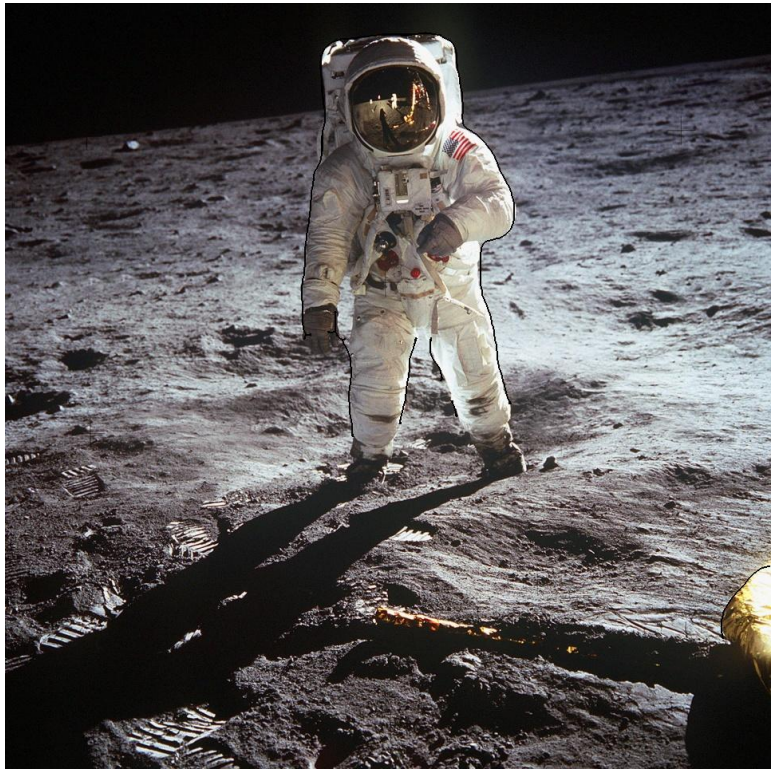
Context and Synthesis Regions

1. The disconnected pixels, which got disconnected in the previous part, are called silhouette pixels.
2. The synthesis regions are generated using flood-fill algorithm with 40 iterations.
3. Then we defined context regions using similar flood-fill algorithm. However, the difference in the algorithm is that it iterates on LDI pixels and halts on silhouette pixels.
4. Then we dilated the context regions to handle depth-discontinuity errors.

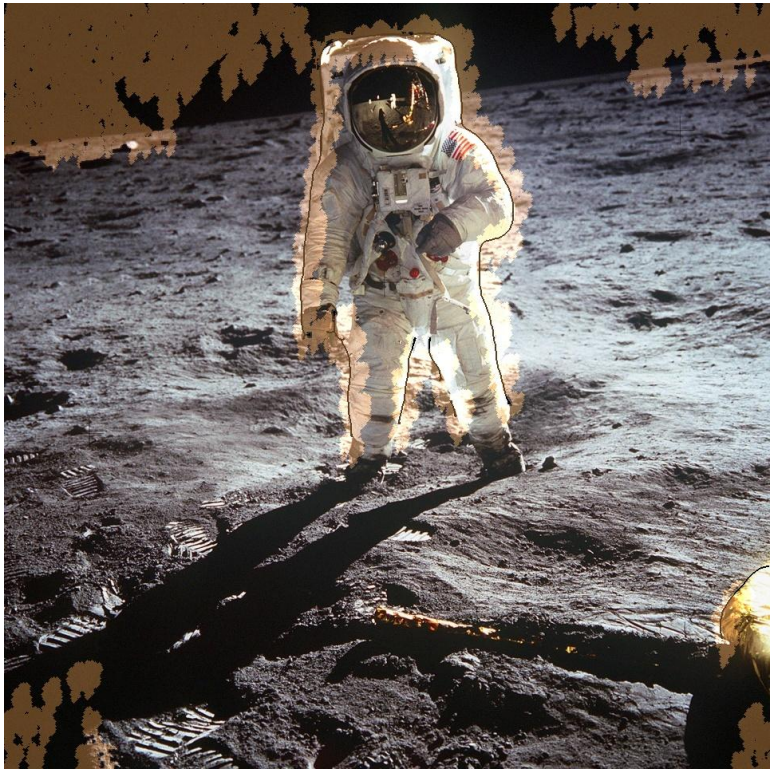
Input

Context and Synthesis Masking

Input



Context and Synthesis Masking



Input

Context and Synthesis Masking