College of Engineering Chengannur
Department of Computer Engineering
M. Tech. Computer Science (Image Processing)
03CS7903 Seminar II
Abstract of Proposed Seminar Topic

3D Photography using Context-aware Layered Depth Inpainting

18/MCS/2019 CHN19CSIP02 Anushree Santosh

September 8, 2020

Keywords: RGB-D image,occuled region,motion parallex,LDI method

Abstract

A method for converting a single RGB-D 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. Use a Layered Depth Image with explicit pixel connectivity as underlying representation, and present a learning-based inpainting model that synthesizes new local color-and-depth content into the occluded region in a spatial context-aware manner. The resulting 3D photos can be efficiently rendered with motion parallax.

The most salient features in rendered novel views are the disocclusions due to parallax: 1.naive depth-based warping techniques either produce gaps (1a) or stretched content (1b). 2. Facebook 3D Photos use a layered depth image (LDI) representation, which is more compact due to its sparsity, and can be converted into a light-weight mesh representation for rendering. The color and depth in occluded regions are synthesized using heuristics that are optimized for fast run time on mobile devices. In particular it uses a isotropic diffusion algorithm for inpainting colors, which produces overly smooth results and is unable to extrapolate texture and structures

In this work ,a new learning-based method that generates a 3D photo from an RGB-D input. The depth can either come from dual camera cell phone stereo, or be estimated from a single RGB image . Use the LDI representation (similar to Facebook 3D Photos) because it is compact and allows us to handle situations of arbitrary depth-complexity. Unlike the "rigid" layer structures above, explicitly store connectivity across pixels in representation. However, as a result it is more difficult to apply a global CNN to the problem, because in this topology is more complex than a standard tensor.

Instead, break the problem into many local inpainting sub-problems, and solve iteratively. Each problem is locally like an image, so can apply standard CNN.Here use an inpainting model that is conditioned on spatially adaptive context regions, which are extracted from the local connectivity of the LDI. After synthesis,fuse the inpainted regions back into the LDI, leading to a recursive algorithm that proceeds until all depth edges are treated.

The result of this algorithm are 3D photo with synthesized texture and structure in occluded regions. Unlike most previous approaches , do not require predetermining a fixed number of layers. Instead this algorithm adapts by design to the local depth-complexity of the input and generates a varying number of layers across the image.

Layered depth image method takes as input an RGBD image (i.e., an aligned color-and-depth image pair) and generates a Layered Depth Image (LDI) within painted color and depth in parts that were occluded in the input. 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. Unlike the original LDI work, we explicitly represent the local connectivity of pixels: 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. LDIs are a useful representation for 3D photography,because (1) they naturally handle an arbitrary number of layers, i.e., can adapt to depth-complex situations as necessary,and (2) they are sparse, i.e., memory and storage efficient and can be converted into a lightweight textured mesh representation that renders fast. The quality of the depth input to method does not need to be perfect, as long as discontinuities are reasonably well aligned in the color and depth channels.

Given an input RGB-D image, this method proceeds as follows.Firstly, initialize a trivial LDI, which uses a single layer everywhere and is fully 4-connected. In a preprocess,detect major depth discontinuities and group them into simple connected depth edges. These form the basic units for main algorithm below. In the core part of algorithm, iteratively select a depth edge for inpainting and then disconnect the LDI pixels across the edge and only consider the background pixels of the edge for inpainting. And then extract a local context region from the "known" side of the edge, and generate a synthesis region on the "unknown" side. The synthesis region is a contiguous 2D region of new pixels, whose color and depth values are generated from the given context using a learning-based method.Once inpainted, merge the synthesized pixels back into the LDI. This method iteratively proceeds in this manner until all depth edges have been treated.

The only input to this method is a single RGB-D image. Every step of the algorithm below proceeds fully automatically. Normalize the depth channel. All parameters related to spatial dimensions below are tuned for images with 1024 pixels along the longer dimension, and should be adjusted proportionally for images of different sizes.And start by lifting the image onto an LDI, i.e., creating a single layer everywhere and connecting every LDI pixel to its four cardinal neighbors. Since , goal is to inpaint the occluded parts of the scene, need to find depth discontinuities since these are the places where we need to extend the existing content. In most depth maps produced by stereo methods (dual camera cell phones) or depth estimation networks, discontinuities are blurred across multiple pixels, making it difficult to precisely localize them.Therefore, sharpen the depth maps using a bilateral median filter.

After sharpening the depth map,find discontinuities by thresholding the disparity difference between neighboring pixels. This results in many spurious responses, such as isolated speckles and short segments dangling off longer edges.And clean this up as follows: First,create a binary map by labeling depth discontinuities as 1(and others as 0). Next,use connected component analysis to merge adjacent discontinuities into a collection of "linked depth edges". To avoid merging edges at junctions,separate them based on the local connectivity of the LDI. Finally,remove short segments (<10 pixels), including both isolated and dangling ones.

In Context and synthesis regions,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. And start by disconnecting the LDI pixels across the discontinuity . and also call the pixels that became disconnected (i.e.,are now missing a neighbor) silhouette pixels Only the background silhouette requires inpainting. But usually are interested in extending its surrounding content into the occluded region.And then start by generating a synthesis region, a contiguous region of new pixel. These are essentially just 2D pixel coordinates at this point. Then initialize the color and depth values in the synthesis region.using a simple iterative flood-fill like algorithm. It starts by stepping from all silhouette pixels one step in the direction where they are disconnected. These pixels form the initial synthesis region.

One important difference to this work is that these image holes were always fully surrounded by known content,which constrained the synthesis. In this case, however, the inpainting is performed on a connected layer of an LDI pixels, and it should only be constrained by surrounding pixels that are directly connected to it. Any other region in the LDI, for example on other foreground or background layer,is entirely irrelevant for this synthesis unit, and should not constrain or influence it in any way.

.Given the context and synthesis regions,next goal is to synthesize color and depth values. Even though , perform the synthesis on an LDI, the extracted context and synthesis regions are locally like images, so that can use standard network architectures designed for images. The inpainted depth map, however, may not be well-aligned with respect to theinpainted color. To address this issue, design color and depth inpainting network and break down the inpainting tasks into three sub-networks: (1) edge inpainting network, (2) color inpainting network, and (3)depth inpainting network.

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 algorithm can easily be rendered using standard graphics engines on edge devices

References

[1] Connelly Barnes, Eli Shechtman, Adam Finkelstein, and Dan B Goldman. Patchmatch: A Randomized Correspondence Algorithm for Structural Image Editing. *ACM Transactions on Graphics*, 28:24, 2009.

[2] Ziyang Ma, Kaiming He, Yichen Wei, Jian Sun, and Enhua Wu. Constant Time Weighted Median Filtering for Stereo Matching and Beyond. In *Proceedings*

of IEEE International Conference on Computer Vision, pages 49–56, 2013.

[3] Guilin Liu, Fitsum A Reda, Kevin J Shih, Ting-Chun Wang, Andrew Tao, and Bryan Catanzaro. Image Inpainting for Irregular Holes using Partial Convolutions. In *ECCV*, page 2, 2018.