The impact of dense layer compression of MPIs on image quality

Author: Ilker Kiliç
Supervisor: Petr Kellnhofer

TU Delft

1. The Problem

- Multiplane images: Simple to render 3D scenes
- Images are stacked in front of each other to create sense of depth
- Multiplane images are very efficient to render
- But take up a lot of memory due to the large number of image layers
- MPI layers are sparse and often contain duplicate information
- Compression required for viability on low-power platforms like the web and VR which stand to gain most from MPI's rendering efficiency.

Figure 1: Visualization of MPI

2. Dense layer compression

Dense layers: A potential avenue for MPI compression

Merge every layer down, based on simple rules:

- Is the lower pixel empty? Replace with upper pixel.
- Is the upper pixel empty? Skip this pixel.
- Is the lower pixel similar enough in color? Replace with upper pixel.
- Are there any resulting layers that are nearly empty? Discard that layer.

Result is a set of 'dense layers'. Alpha values are stored seperately.
By saving indices to dense layers instead of color data, we can save space!
But what does that do with the image quality?

Figure 2: Visualization of dense layer compression

3. The research question

How does dense layer compression affect the image quality of MPIs?

Subquestions

1. How might dense layer compression be used to reduce the memory footprint of MPIs?
2. What potential artifacts may result from dense layer compression of MPIs?
3. How does dense layer compression affect the perceived image quality of MPIs?

4. Methodology

1. Implement layer merging and reconstruction of dense layer compression in Python using numpy, PIL and scikit-image libraries.
2. Use implementation to compress and decompress MPIs of custom 3D scenes synthesized using Local Light Field Fusion, a method developed by Mildenhall et al. [1], **in a parameter sweep**.
3. Use image quality metrics (PSNR, SSIM, LPIPS) to compare decompressed composites to uncompresed composites for each parameter combination in executed sweep. (See figure 3 and 5)
4. Observe decompressed results for compression artifacts. (See figure 3 and 4)

Three parameters are used in this method**:**

- c_c - Color similarity threshold: We determine similarity by converting colors to L*a*b* color space and taking the Euclidean distance between color components.
- c_s - Surface threshold: If the fraction of non-empty pixels vs empty pixels in a dense layers is beneath this threshold, we discard the dense layer as it will barely contribute to the overall image.
- c_a - Alpha threshold: If the alpha value of a pixel is below this threshold, we consider the pixel 'empty'.

5. Results

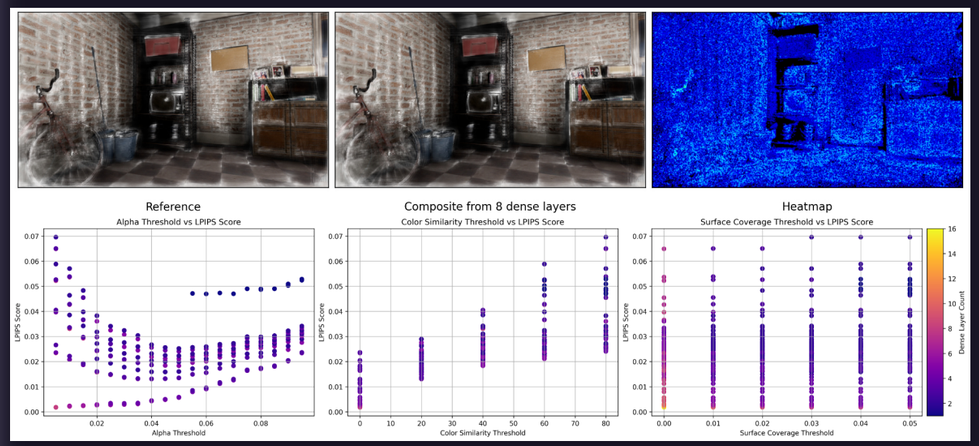

Figure 3: Difference heatmap and parameter sweep results for 'Props' MPI

Figure 4: Visible image artifacts in compressed layer stacks

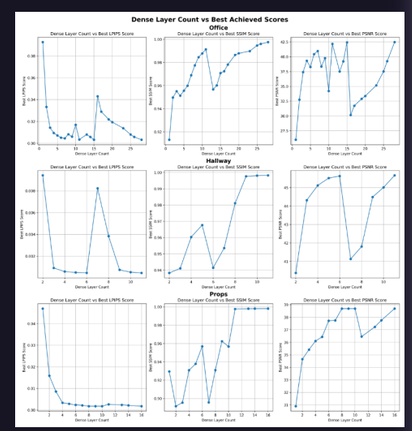

Figure 5: Best achieved quality scores in parameter sweep for each achieved dense layer count

Running our implementation on three custom MPIs of 32 image layers shows we can condense these down into just a few dense layers. Sometimes even into only 2. Depending on the MPI's composition, quality loss at a very low dense layer count of 2 or 3 may be negligiby low, or may display several visible artifacts. At dense layer counts above 6, obvious image artifacts usually disappear. Difference heatmaps show very frequent slight color changes across the image, but these are essentially unnoticeable with the naked eye.

6. Conclusion

Dense layer compression is a promising technique that allows us to condense an MPI's image layer stack down to only a couple of images with often minimal image quality loss. Further research is warranted into evaluating the method with more MPIs of different compositions. Our research shows that dense layer compression is a promising universal method for compressing MPIs that warrants further research. Our implementation only results in a set of dense layers and a map of indices and alpha values. To turn our method into a full compression algorithm, future research should involve determining how to compress the index and alpha map.

[1] Ben Mildenhall, Pratul P. Srinivasan, Rodrigo Ortiz-Cayon, Nima Khademi Kalantari, Ravi Ramamoorthi, Ren Ng, and Abhishek Kar. Local light field fusion: Practical view synthesis with prescriptive sampling guidelines. ACM Transactions on Graphics (TOG), 2019.