Lab Session 6
NLOS Reconstruction with Transient Imaging

Computational Imaging. Universidad de Zaragoza.

Introduction

The goal of this lab session is implementing a method to reconstruct non-line-of-sight (NLOS) geometry based on time-resolved measurements at the relay wall. Your main task will be developing an NLOS reconstruction algorithm based on backprojection, testing it on different NLOS datasets [1], to analyze its performance and ability to reconstruct hidden scenes under different geometric configurations (see Figure 1) and capture topologies.

You will have to implement naive version of the backprojection NLOS reconstruction method [2]. Section 1 contains a brief overview of the principal aspects of the method. Please refer to the original article and the lecture notes for more details on the method. Then you will adapt your method to handle confocal measurements and compare the performance w.r.t. non-confocal measurements. Finally you will modify your method to mitigate part of the attenuation effects produced by distance falloff and cosine foreshortening and illustrate it in two scenes.

All the data and preliminary code required for this lab session are provided in the supplemental material. Note that you are not required to provide a high-performance implementation of the backprojection method. It's ok to nest multiple for loops, as long as the method provides the correct reconstruction.

1 Backprojection reconstruction (60 points)

In this first task you need to implement a backprojection method to reconstruct NLOS geometry using the transient datasets provided in the supplemental material. You have to implement the basic algorithm introduced by Velten et al. [2] to reconstruct a hidden scene from an arbitrary number of laser and SPAD points on a relay wall. In particular, for every voxel $\mathbf{x}_v$ of a hidden scene, the backprojection reconstruction $G(\mathbf{x}_v)$ of that voxel from a set of laser-SPAD measurements H on the relay wall is defined as

$$G(\mathbf{x}_v) = \sum_{\forall \mathbf{x}_l, \mathbf{x}_s} H(\mathbf{x}_l, \mathbf{x}_s, t_v) \tag{1}$$

where t_v corresponds to the time-of-flight of third-bounce light paths from the laser device to the relay wall positions $\mathbf{x}_l \in L$, to the reconstructed voxel $\mathbf{x}_v \in V$, back to SPAD positions on the wall $\mathbf{x}_s \in S$, and back to the SPAD device. You can find a schematic of these light paths in Figure 2.

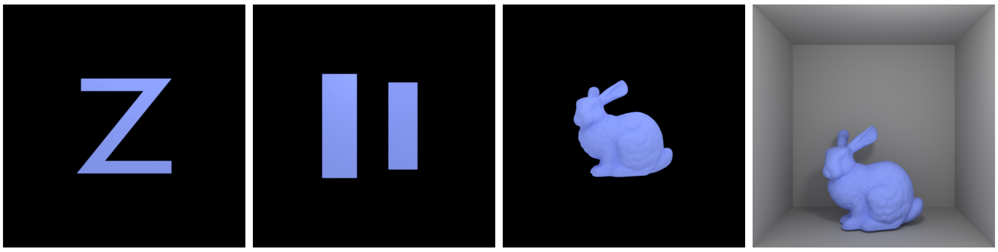

Figure 1: Illustration of the hidden scenes in the four datasets provided, from left to right: `Z_*`, `planes_*`, `bunny_*`, `bunnybox_*`.

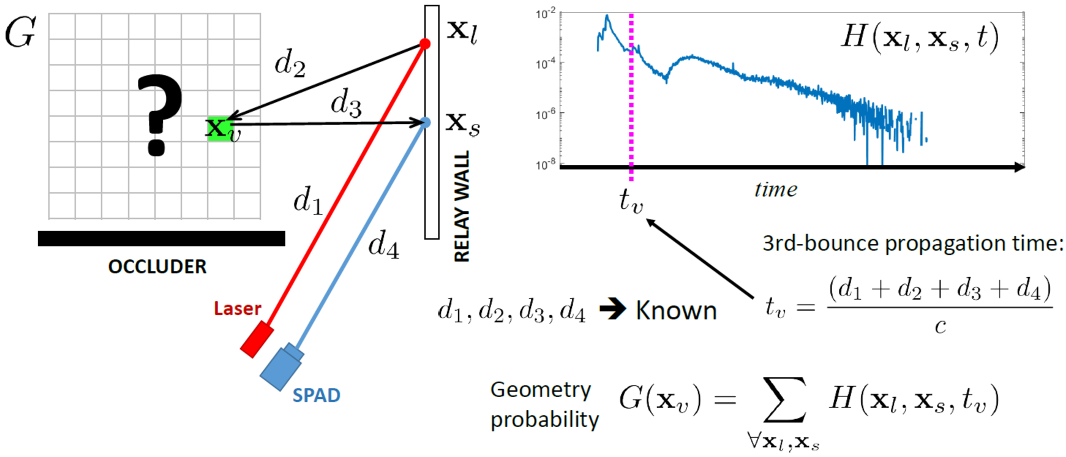

Figure 2: Schematic view of the backprojection reconstruction for a single voxel of the hidden scene.

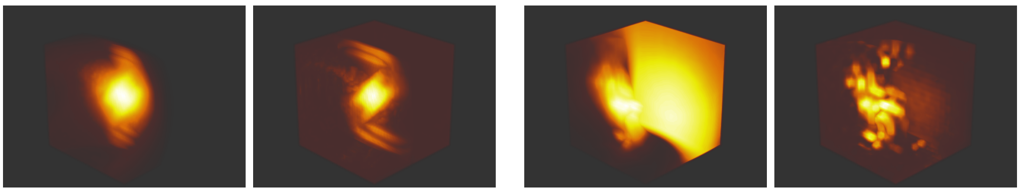

Figure 3: Examples of reconstructions in the Z and isolated bunny scenes.

Datasets The function `load_hdf5_dataset()` reads, processes, and loads the datasets contained in hdf5 files into a Matlab struct, which contains all the necessary information to implement the backprojection algorithm. You can find the description of the output struct fields in the comments of the Matlab file. Figure 3 shows the four different hidden scenes whose NLOS measurements are provided in the datasets. All the datasets have been obtained from [1].

Basic tests and filtering Test the implemented method over the `Z_d=0.5_l=[1x1]_s=[256x256].hdf5` dataset, and measure the time it takes reconstruct for different voxelization and capture grid resolutions. You can use the function `volshow` available on Matlab along with the view configuration provided in the file `volshow_config.mat` to correctly display the reconstructed volume as `volshow(G,volshow_config)`. You can also use the function `volumeViewer` of Matlab, setting maximum intensity projection and the "hot" color map to have extra control over the visualization thresholds. Apply a Laplacian filter to the reconstructed volumes to improve the quality of the reconstructions. You can use the following code snippet to apply this filtering, assuming G is the 3D volume reconstructed with your backprojection method.

```
f_lap = fspecial3('lap');
G_lap = imfilter(G, -f_lap, 'symmetric');
```

Non-planar geometry The Z dataset contains measurements for geometry contained within a single plane parallel to the relay wall. In contrast, the two bunny_* datasets contain measurements for an isolated non-planar geometry (the Stanford bunny) captured with different relay wall topologies: a single laser point (1x1) and a SPAD grid (256x256) in the file bunny_d=0.5_l=[1x1]_s=[256x256].hdf5, and an exhaustive scanning pattern with a laser grid (16x16) and a SPAD grid (16x16) in the file bunny_d=0.5_l=[16x16]_s=[16x16].hdf5. Additionally, the bunnybox_d=0.5_l=[16x16]_s=[16x16].hdf5 dataset contains measurements of a Stanford bunny within a Cornell box, creating significant global illumination effects. Apply your reconstruction method to the different datasets and answer the questions asked later.

In the report, include visualizations of the reconstructed volumes from different viewpoints, at different resolutions, with and without Laplacian filtering, and include the reconstruction time for each resolution you use. Please answer the following questions and illustrate them with the obtained results when you consider it necessary:

1. How does the reconstruction time scale with respect to the hidden scene voxelization and the relay wall capture resolution?
2. How does reconstruction of non-planar geometry (e.g. the bunny scene) differ when using a single laser position and a SPAD grid vs. combining laser and SPAD grids? What is the reason for the changes in the reconstructions?
3. Why do the outputs of backprojection reconstructions look blurry? What is the effect of applying a Laplacian filtering over this output and why does this happen?
4. What happens when reconstructing objects that are not isolated but surrounded by more geometry such as the bunny in a box? Try to explain why the reconstruction looks better or worse in this case, relating the resulting effects to the possible light paths with a specific time-of-flight.

Some aspects to consider when choosing reconstruction parameters:

- *Position and extent of the hidden geometry.* While in real applications the extent of the target hidden geometry is unknown, the datasets in the supplemental material provide the position and dimensions of the region where the hidden geometry is contained. Namely you can find this information in the dataset fields volumePosition and volumeSize. You can use this information to define the extent of your voxelization, and avoid reconstructing large empty regions.
- *Resolution of the voxelization and capture grids.* Choose a tractable volume resolution to test your algorithm. A reconstruction grid of size $4 \times 4 \times 4$ will be fast to compute, but will

yield a very coarse reconstruction; a reconstruction grid of size $1024 \times 1024 \times 1024$ will provide a finer reconstruction, but the algorithm may take forever to finish. A voxelization with resolution of $8 \times 8 \times 8$ or $16 \times 16 \times 16$ could be a good starting point to test that your algorithm works correctly. A resolution of $32 \times 32 \times 32$ may be a good trade-off between reconstruction quality and execution time, once you know your method works well. For testing purposes, you can also try skipping part of the laser and SPAD points on the relay wall (e.g. one every two, one every four, on each dimension), at the risk of affecting the reconstruction quality.

2 Backprojection on confocal measurements (20 points)

Confocal capture topologies synchronously co-locate lasers and SPADs when scanning the relay wall. Therefore each illuminated point only generates one corresponding time-resolved measurement at the same location. Modify your backprojection method to handle confocal measurements, and apply it to the confocal version of the bunny scene `bunny_d=0.5_c=[256x256].hdf5`. Please include comparisons between the confocal reconstruction, and the previous non-confocal reconstructions you made of the same scene. Is there any noticeable difference with reconstructions of non-confocal datasets? Try to explain the observable differences in terms of the light transport paths that may take place in each capture topology.

3 Attenuation and foreshortening correction (20 points)

You may have noticed the resulting reconstructions are attenuated towards regions far from the relay wall. This is caused partly by two factors. First, quadratic attenuation due to the different path distances leads to dimmer reconstructions at points of the hidden scene that are further from the relay wall. This is specially noticeable in the scene containing two planes at different distances `planes_d=0.5_l=[16x16]_s=[16x16].hdf5`. Second, the foreshortening due to the surfaces cosine term introduces significant radial attenuation, therefore points away from the relay wall center present dimmer reconstruction values. This is specially noticeable at the `Z` scene. In this final task, you will have to modify your backprojection algorithm to mitigate these attenuation effects given the known information about the reconstructed points and the normal at the relay wall for each laser and SPAD point (available in the datasets). Hint: think about how quadratic and cosine attenuation occur in forward propagation of light paths (see Figure 2), and invert these operations when accumulating radiance back onto the voxels (see Equation 1). Illustrate the results in the `planes` and `Z` scenes, and explain the motivation of the different factors you included in Equation 1 to mitigate the attenuation effects.

Deliverables

Your solution should be an archive (e.g., a ZIP file) that includes the following:

- All of your Matlab code with the proper comments on the principal steps, as well as a README file explaining the main design choices and instructions on how to run the code.
- A PDF report that includes the results required for each section, and responses to the different questions asked regarding the results you obtained.

References

[1] M. Galindo, J. Marco, M. O'Toole, G. Wetzstein, D. Gutierrez, and A. Jarabo. A dataset for benchmarking time-resolved non-line-of-sight imaging, 2019. URL `https://graphics.unizar.es/nlos`.

[2] A. Velten, T. Willwacher, O. Gupta, A. Veeraraghavan, M. G. Bawendi, and R. Raskar. Recovering three-dimensional shape around a corner using ultrafast time-of-flight imaging. *Nature Communications*, (3), 2012.