Introduction

The goal of this assignment is to reconstruct a dense 3D map from multiple scene views. We mainly utilize work from Gallup et al. 2007. A simplification is made by only considering the sweep direction orthogonal to the reference image plane. The pipeline consists of four steps. This report describes the most important implementation details of the system.

Part 1: Plane Sweep Homography

This part consists of finding a set of homographies that maps one image to the reference view where the views are related by $\mathbf{M} = \begin{pmatrix}\mathbf{R} & \mathbf{t}\end{pmatrix}$, such that $\begin{pmatrix}X & Y & Z\end{pmatrix}^\top = \mathbf{M}\begin{pmatrix}X_{ref} & Y_{ref} & Z_{ref} & 1\end{pmatrix}^\top$ for a set of inverse depths $\{d_i^{-1}\}_{i=1,2,\dots,D}$ Noting that the direction vector from the plane(s) to the reference view is $\mathbf{N} = \begin{pmatrix}0 & 0 & -1\end{pmatrix}^\top$ some simple calculations show that this homography is given by

$$\mathbf{H} = \mathbf{K}_{ref}\left(\mathbf{R} - \frac{\mathbf{t}\mathbf{N}^\top}{d}\right)\mathbf{K}^{-1}$$

where $\mathbf{K}_{ref}$ and $\mathbf{K}$ are the intrinsic of the reference view and the second view respectively. In our case they are the same. (Note that we are using the distance (d) here, not the inverse)

Part 2: Plane Sweep Stereo

This part is the heart of the plane Sweep algorithm. Following [Gallup et al. 2007] each of the views are transformed to the reference view for each homography induced by each of the planes. The absolute difference between a region of that pixel, and the same region of each other transformed view (there are $N \times D$ of them) is taken and summed over. The gain ratio is assumed 1 as the images are in an indoor environment with relatively equal light conditions. The depth estimates were obtained by minimizing over the depth dimension of this tensor.

In general this procedure is $\mathcal{O}(N(D + WH)$ where W is the width of the image and H is the height. During testing on the given dataset, after a longer optimization effort, the procedure took around an hour to complete on an regular laptop.

Lastly, parts of the reference view might not be visible in the other views, and so care must be taken to handle this properly. This was solved by setting these regions of the transformed image equal to the reference image, essentially driving that difference in the tensor from part 2 to 0 . This eliminates the needs of checks later on, reducing the runtime of the procedure.

Part 3: Unproject Depth Map

This part finds the world coordinate(in camera basis) corresponding to a pixel (x, y) in the reference view with depth d. A good way to frame this problem is just that - what point $\begin{pmatrix}X, Y, d, 1\end{pmatrix}^\top$ maps to (x, y) in the reference view? Again note the importance of the simplification of only considering fronto-parallel views. Some geometric, and algebraic reasoning yields

$$\mathbf{X} = \frac{d}{(\mathbf{K}^{-1})_3(x, y, 1)^\top}\mathbf{K}^{-1}(x, y, 1)^\top \tag{1}$$

where $(\mathbf{K}^{-1})_3$ is the third row of the inverse intrinsic matrix. Some investigation shows that this is actually $(0, 0, 1)^\top$ removing the entire vector product in the denominator.

Part 4: Post Processing

The goal of this part is enhancing both the depth estimate, and the resulting 3D model. Three mechanisms were used. Firstly a 3D median filter was applied to the tensor from part 2. A $9 \times 9 \times 9$ kernel size yielded good results. A second median smoothing of the depth image seemed to improve the results further - a smaller kernel size of 3×3 seemed sufficient. Secondly any points in the tensor where less than 5 views contributed to the value were ignored by setting it to infinity. Lastly, using the insight that depths around object borders - edges in the depth image - are in general difficult to estimate, these where detected using a simple Sobel operation. The magnitude of the gradients was thresholded with a value of $0.1 \cdot \bar{X}$ where $\bar{X}$ is the mean of the depth image. This procedure introduces several tuning variables, and a larger and more varied dataset is necessary to properly tune it. However the results demonstrate the feasibility of this approach. The results are shown bellow Figure 1.

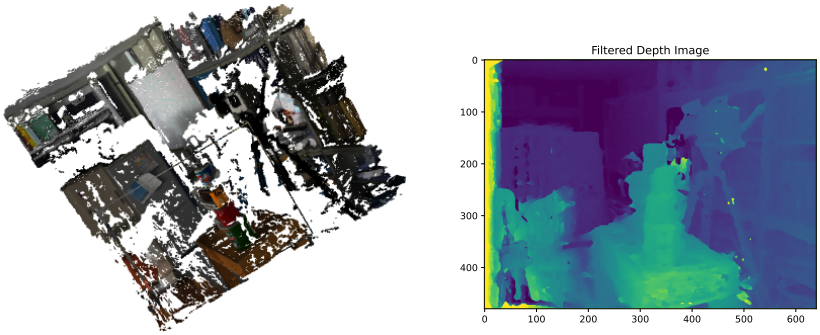

Figure 1: The final results after post processing. Left: the 3D reconstruction, right: the depth map

Bibliography

Gallup, David et al. (2007). "Real-Time Plane-Sweeping Stereo with Multiple Sweeping Directions". In: *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–8. DOI: 10.1109/CVPR.2007.383245.