Introduction

In this project, I implemented a disparity map estimation algorithm using graph cuts, a method known for its effectiveness in generating smooth and accurate depth maps from stereo images. The disparity map helps calculate depth by finding pixel correspondences between two images taken from slightly different viewpoints. This report covers the implementation of the algorithm, the effects of parameter changes on results, and a comparison with the region-growing method .

Parameter Exploration: λ and NCC Neighborhood Size :

1. Effects on λ:

- **Low λ :** values prioritize data fidelity, leading to disparity maps that closely match the actual intensity patterns of the images. However, this also results in a noisier output, with less smoothness.

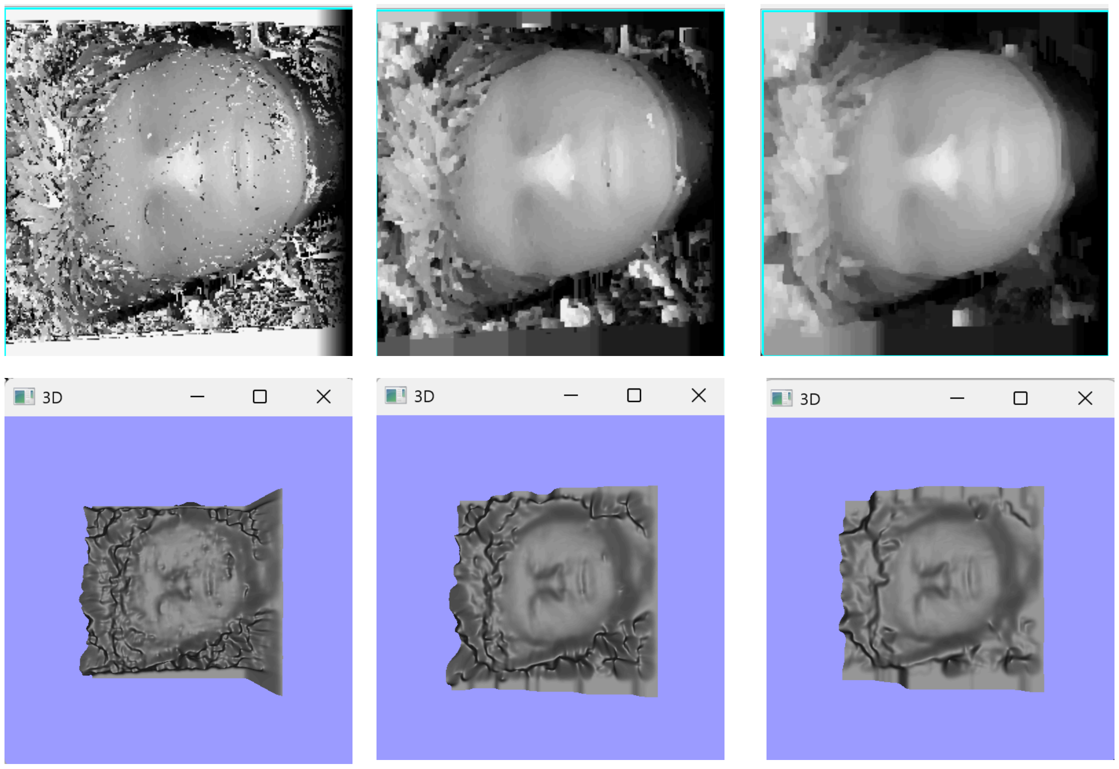

Figure 1 : varying λ for these values [0, 0.01, 0.05]

- **Higher λ** : values enforce more smoothness, reducing noise and making the disparity transitions smoother. However, excessive smoothing can oversimplify details, leading to a loss of fine structures in the depth map.

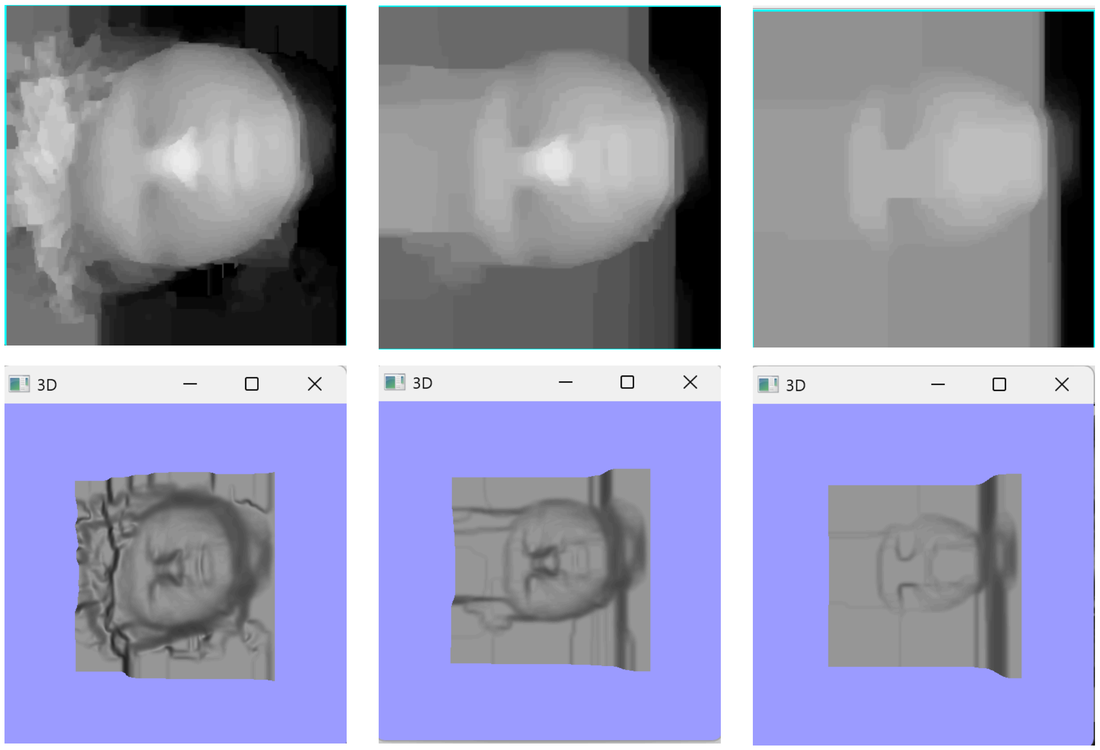

Figure 2 : varying λ for these values [0.1, 0.5 , 1]

2. Effect of NCC Neighborhood Size:

- Smaller neighborhood sizes result in more localized matches, potentially capturing finer details but also being more sensitive to noise and variations in texture.
- Larger neighborhood sizes provide a more robust similarity measure, resulting in smoother disparities. However, it makes the algorithm slower, because the ncc computation is larger.

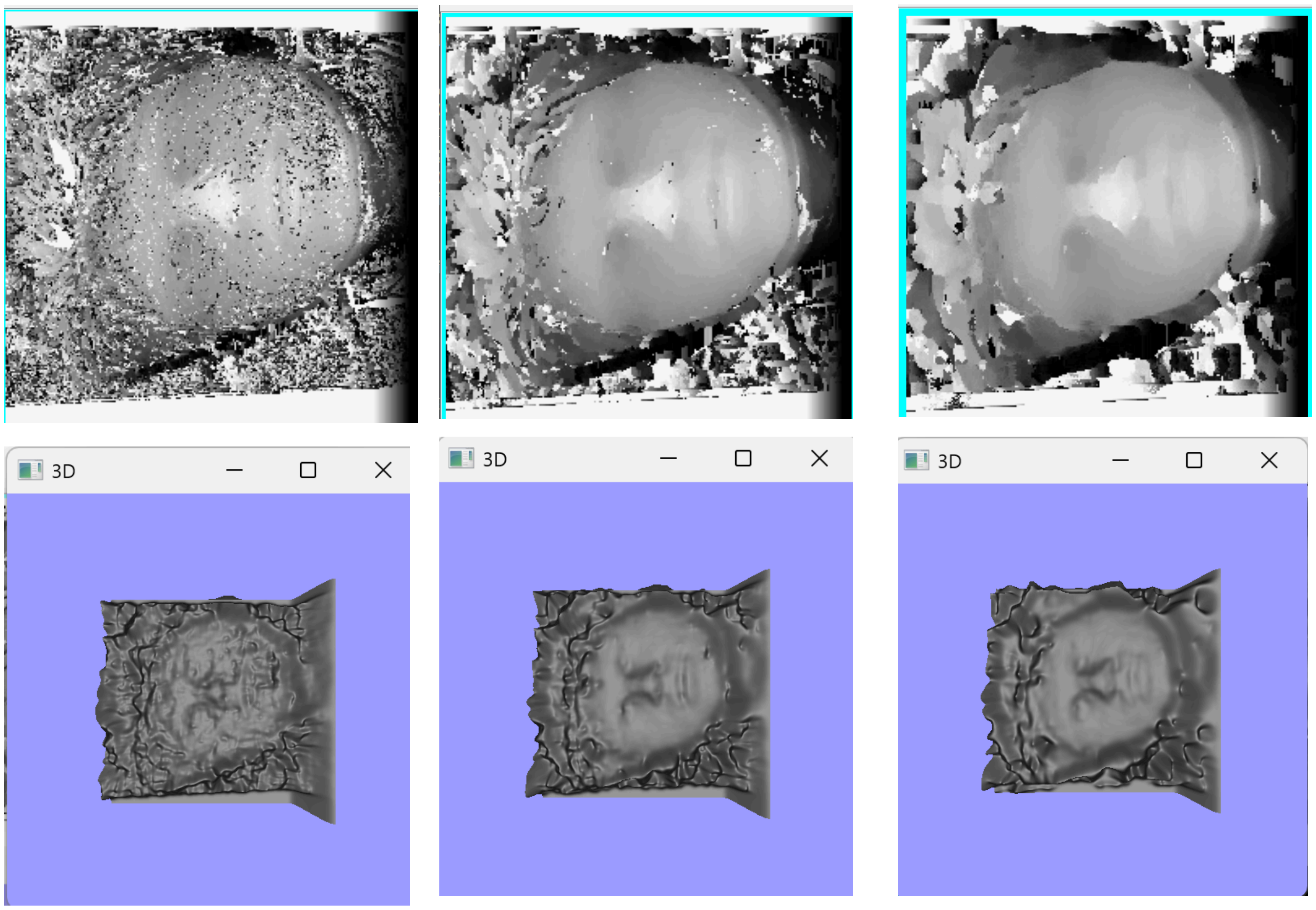

Figure 3 : varying win =[2,5,9]

3. Comparative Analysis with Region-Growing Method:

- **Precision**: Graph cuts produced more accurate disparity maps, especially in textured and edge regions, while region-growing struggled in areas with subtle depth changes.
- **Smoothness**: Graph cuts delivered smoother transitions due to its regularization term, whereas region-growing often showed blocky artifacts, especially on gradual depth changes.
- **Computation Time**: Graph cuts proved faster due to its efficient global optimization, whereas region-growing, being locally iterative, took longer to compute.