Computer Vision 2
Assignment 3

Martín de la Riva, Nuno Neto de Carvalho Mota
11403799, 11413344

June 2018

1 - Introduction

This assingment consists of 3 parts. In the first part we analyze the advantages and disadvantages between models for 3D reconstruction; more specifically a depth based method (ICP) and a texture based method (SfM). Both methods have been implemented in previous assignments. In the next part, another two methods are introduced and analyzed; Poisson Surface Reconstruction [1] and Marching Cubes [2]. They are tested to reconstruct the 3D structure of a person given by 8 different frames, where each contains information about depth, RGB and camera values (such as focal length and camera pose). This information is used to merge the point clouds of all frames into one representation, which is then made into a mesh using one of the two previously mentioned methods. This will show the 3D structure of the model without colour. In the last section, colour and texture is added to the mesh.

2 - Depth-based and Texture-based 3D Reconstruction Comparison

In Iterative Closest Point (ICP) the goal is to match two point clouds by finding appropriate translation and rotation matrices, minimizing the distance between the two point clouds. It requires a proper initialization of the structure of the body from which the matrices are calculated on. If it is not the case, the model may get stuck in a local minimum. Also, updating the matrices is computationally expensive as it has to calculate the distance between all points in order to find the closest points. To overcome this, sampling is performed. Of course random sampling is not ideal, and to ensure proper results some type of informational sampling is required. Generally, spatial sampling is chosen, maintaining the overall structure while eliminating points that are close to each other. Concluding, the drawbacks of this method are the need for point clouds (only image information is not enough), the proper initialization and computational expense.

In Structure from Motion (SfM), a 3D scene is reconstructed using a set of 2D images. In order to do this, it must find the corresponding points between images. It is required to use epipolar theory to find these correspondences, from which the RANSAC algorithm is applied to ignore bad matches. An assumption is made that either only the camera moves, or the objects move rigidly. After applying the algorithm in several frames the result is a pointview matrix denoting which features are present in which images, and, from it, dense blocks are selected (features that appear in numerous frames) to recreate the transformation matrix. With this the 3D structure can be calculated and the points outside of the dense blocks added to this 3D model. The main disadvantage is that the assumption makes it restrictive for many problems.

Summarizing, ICP is a method to combine point clouds while SfM tries to regenerate 3D scenes out of 2D images. One way to mix this two methods is, detect a set of point using SfM out of 2D images and generate set of point clouds with it using the dense blocks in the pointview matrix. Then, using these point clouds ICP would be applied to mix them all together into a 3D structure.

3 - 3D Meshing and Watertighting

Using 8 consecutive point clouds, we created a 3D polygon mesh reconstruction using the methods Marching Cubes [2] and Poisson Surface Reconstruction [1].

Marching Cubes is known for being computationally cheap thanks to it's computation of 14 triangulations per cube (instead of the possible 256); but sometimes having some problems such as creation of holes in it's models. Since it's creation in 1987 many algorithms built upon Marching Cubes have appeared, in terms of both efficiency and quality.
In Figures 1, 2, 3 and 4 we implement Marching Cubes and compute the 3D reconstruction of a person, as previously mentioned. From the parameters in the algorithm we find that the grid size of the cubes is the one that affects the most the final result. Using a too low grid size results in very rude and poorly detailed results (1). Increasing it results in more detailed results (2), although if it is too high it can generate wrinkled results, as well as added noise (3, 4). Also, it can be seen how the aforementioned problem with holes is present in this model, as it always appears in the top of the head.
Note: in the following images the views are always (front, back, left, right and top).

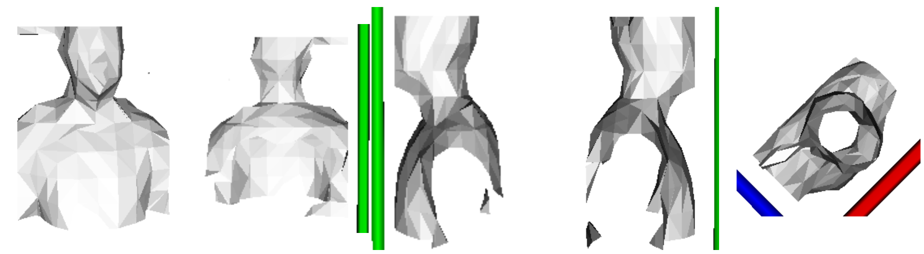

Figure 1: Marching Cubes reconstruction using Grid size = 10.

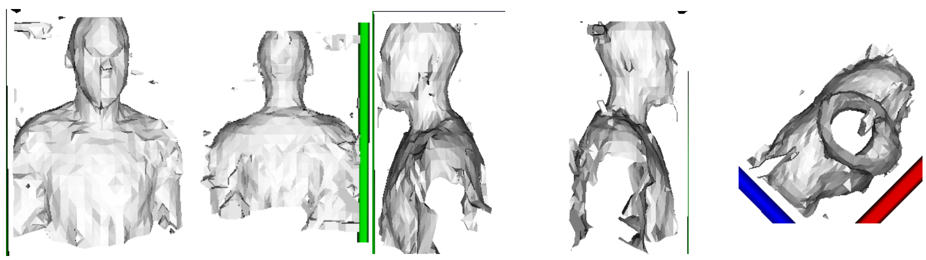

Figure 2: Marching Cubes reconstruction using Grid size = 25.

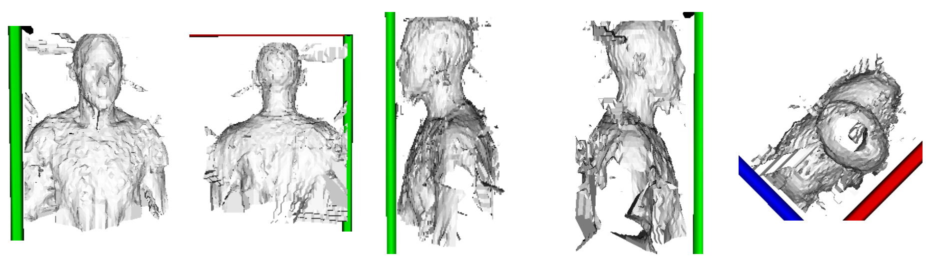

Figure 3: Marching Cubes reconstruction using Grid size = 50.

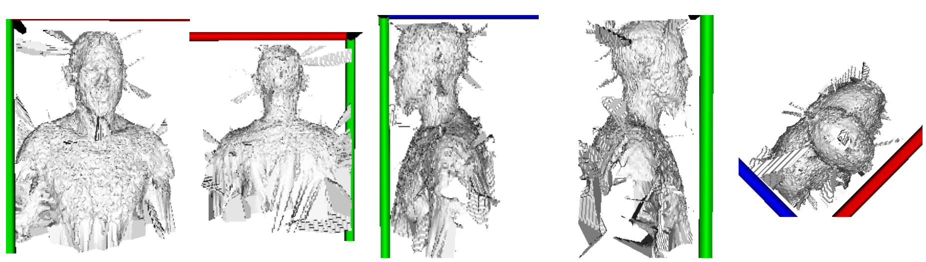

Figure 4: Marching Cubes reconstruction using Grid size = 100.

Poisson Surface Reconstruction was introduced in 2006. This algorithm is more computationally costly than Marching Cubes, but is way better at handling areas with limited information, achieving to make them smooth. It also has a parameter (*poisson depth*) that handles the amount of detail and therefore the smoothness on the final model. Although it is a good feature it can sometimes result in a drawback, as it may connect points in the model that should not be connected, as stated by Kazhdan and Hoppe in the original paper.
As with Marching Cubes, we implemented Poisson Surface Reconstruction and tested it on the person 3D model; as it can be seen in Figures 5, 6, 7 and 8. This model completely outperforms the previous one, being not only more detailed and correct, but also ignoring noise around the 3D structure. We tested the parameters Depth for the smoothness of the model and Minimum Samples for the number of points that can go into an octree node. For the first we found that the smaller the depth is the less detailed the end model is, being nearly a simple body shape, when using a depth of 4 (5), and how it gets more detailed when increasing it (6 and 7). However, if too high this parameter strongly affects the computational cost. For the Minimum Samples parameter, an improvement in smoothness can be noticed when the Depth is high (8).

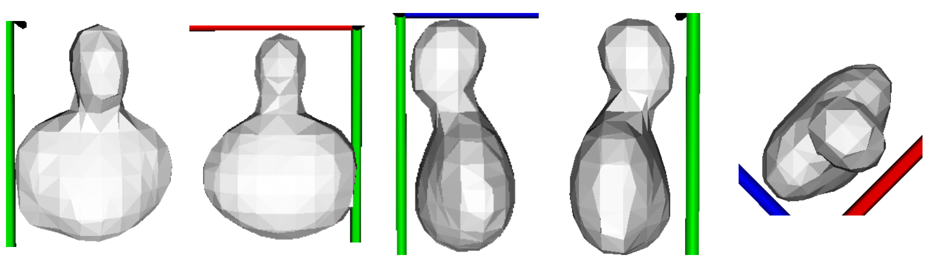

Figure 5: Poisson Surface Reconstruction reconstruction using Depth = 4 and Minimum Samples = 10.

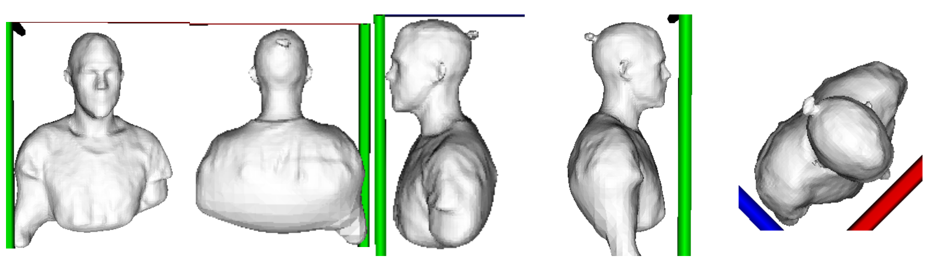

Figure 6: Poisson Surface Reconstruction reconstruction using Depth = 7 and Minimum Samples = 10.

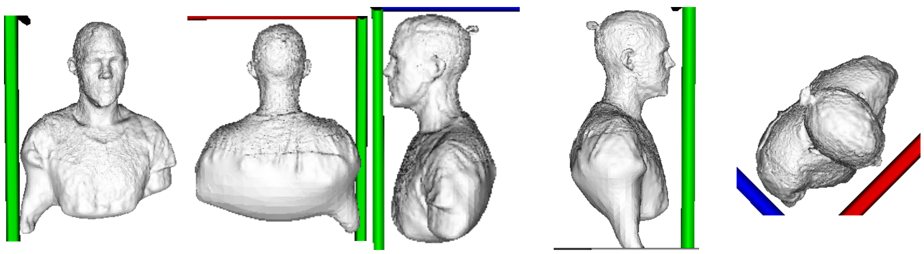

Figure 7: Poisson Surface Reconstruction reconstruction using Depth = 14 and Minimum Samples = 10.

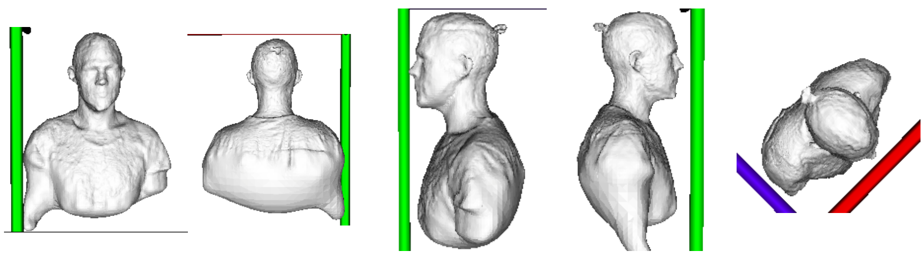

Figure 8: Poisson Surface Reconstruction reconstruction using Depth = 14 and Minimum Samples = 20.

One important aspect to note is that the Poisson reconstruction method achieves an extremely good watertight (hole filling) result, by extracting a continuous isosurface [1].

4 - Coloring 3D Model

In this section we color the 3D reconstructions of the aforementioned models. We now have a mesh of polygons and the information of the point clouds per frame. In order to color the mesh we must determine the RGB values of each vertex in the polygon. As we have 8 different frames with this information we must find a method to decide how to choose the final color in the 3D model. We also must take into account whether the vertex appears on the frame. That is, whether it is visible (it is inside the RGB image) and if it is, whether it is occluded by other polygons. We do the former by obtaining the UV coordinates of the frame and checking if they are within the RGB image. For the latter, we detect if the vertex is occluded using an Octree.
To determine the color of each polygon vertex, we save it's RGB values for each of the frames in which it is non-occluded and visible. Now, as you may have very differed values, we find for each channel the 3 most similar ones. So, let's say we have for the Red channel values (1, 30, 35, 49, 90). We would pick the values (30, 35 and 49). Then, we finalize it's final channel value as the average

of the three.

We test this coloring model on our already implemented 3D reconstruction models, as it can be seen in Figures 9, 10, 11 and 12. As it can be seen even in the models with low resolution you can see the color of the clothes and even the shadows of the 3D model (9). The more detailed the model is the better is the final result. Our best result (12) actually shows some interesting details like the lighting in it's shoulders and neck bones, although it leaves some room for improvement as some areas can be seen grayish or blueish, like in the neck and the sides of the head, although we do not know if this may be due to some noise in the original images.

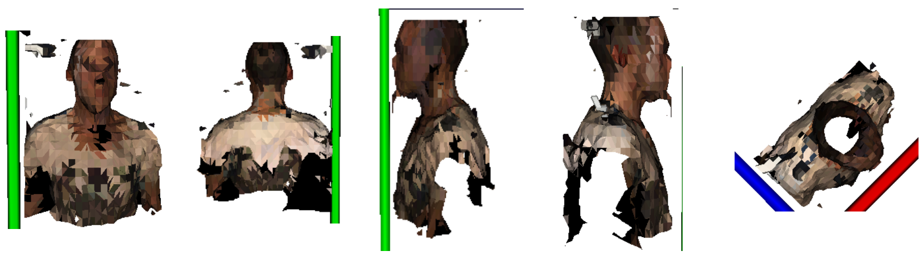

Figure 9: Colored Marching Cubes reconstruction using Grid size = 25.

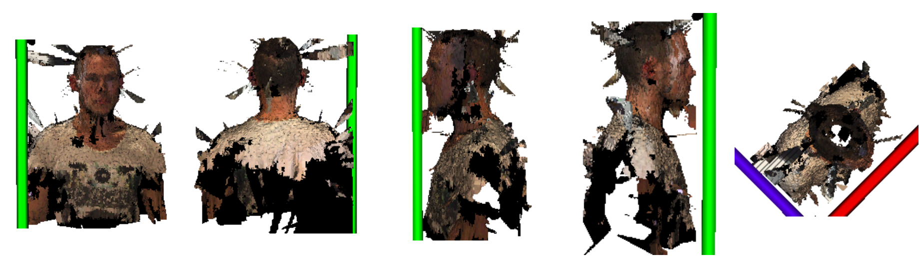

Figure 10: Colored Marching Cubes reconstruction using Grid size = 100.

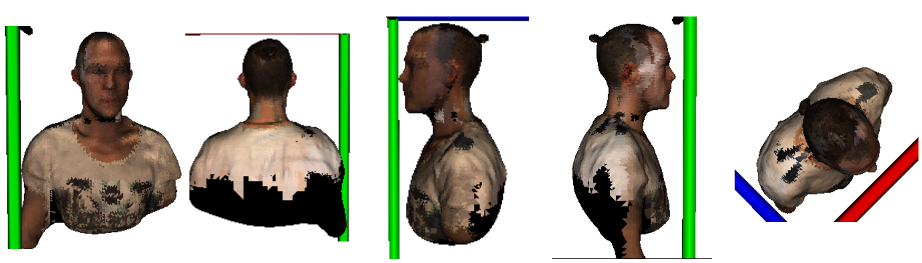

Figure 11: Colored Poisson Surface Reconstruction reconstruction using Depth = 7 and Minimum Samples = 10.

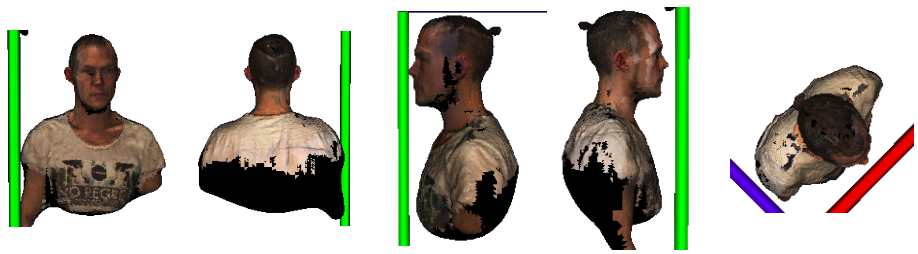

Figure 12: Colored Poisson Surface Reconstruction reconstruction using Depth = 14 and Minimum Samples = 20.

5 - Conclusion

With this assignment we got to learn about 2 different surface reconstruction methods and how to texture those reconstructions.
Marching cubes revealed itself as being an acceptable algorithm, given it's shortcoming as severe artifacts, holes and relatively low robustness to noise, yet extremely fast. Its results are still quite good and impressive.
On the other hand, Poisson surface reconstruction yields much better results, at the expense of more expensive computations. However, the advantages are obvious, yielding smooth surfaces that are significantly more resilient to noise, yet can still tune in on high frequency detail. The quality of producing watertight results is also admirable. It is obvious it far surpasses the Marching Cubes Algorithm.
Finally, we observe that the texturing of the surfaces is also of relatively good quality.

References

[1] Kazhdan, M., Hoppe, H. (2013). Screened poisson surface reconstruction. ACM Transactions on Graphics (ToG), 32(3), 29.

[2] Lorensen, W. E., Cline, H. E. (1987, August). Marching cubes: A high resolution 3D surface construction algorithm. In ACM siggraph computer graphics (Vol. 21, No. 4, pp. 163-169). ACM.