UNIVERSITY OF AMSTERDAM

Assignment 3 Report

Luca Simonetto - 11413522
Edgar Schönfeld - 11398272
Computer Vision 2

June 6, 2017

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

An advantage of a depth-based algorithm like ICP over a texture-based algorithm like SFM is that it does not require feature extraction and matching, which come with their own problems. For example, objects with no texture can deliver only few informative feature points, while repetitive texture provoke mismatches. Specular highlights can cause trouble to SFM as well. SFM on the other hand handles occlusions better, since the algorithm can work with incomplete measurement matrices. Apart from this, "sea urchin" or "planetoid" shaped objects can make the ICP algorithm fail [1]. Other (dis)advantages concern the data gathering: Depth cameras with structured illumination work perfectly in low light conditions. Yet, the resolution of depth-only cameras is not very fine-grained, and large outliers in the reconstructed point cloud cause problems to the ICP [1]. RGB-D cameras have a much finer resolution, as they use color-consistency to refine the depth-image. The additional color information could be used to combine both algorithms. If the frame-to-frame transformations of the two methods are translated in a common mathematical representation, the difference between both transformations could provide information about the validity of a transformation. Similarly, texture consistency could be used as an additional constrained for merging point clouds. Furthermore, depth information may guide or improve the correspondence search of the SFM, by limiting the amount of candidate regions further.

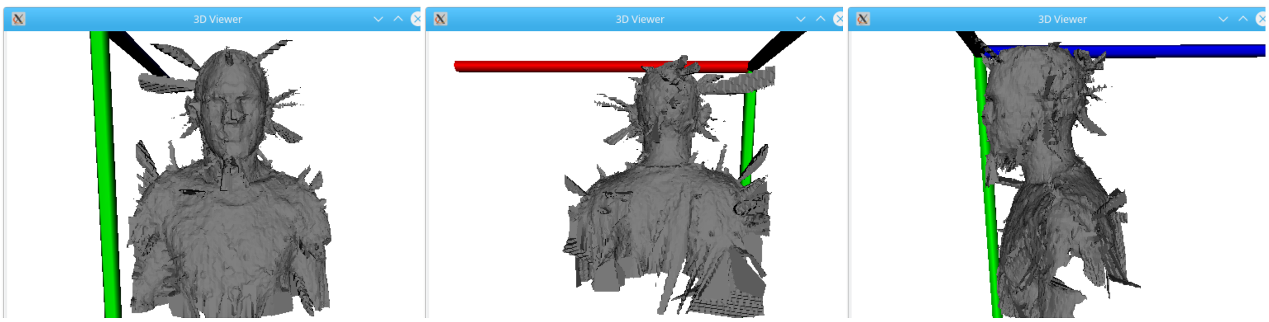
2 3D meshing

The first step in this project is to extract and combine the point clouds originated from the provided data, implementing the algorithm provided. As no particular changes or implementation choices have been made, the implementation will not be discussed.

Having a complete point cloud of the input model, the next step is to generate the corresponding mesh by using one of the available meshing algorithm: in this case, Poisson and marching cubes.

Marching cubes meshing

The marching cubes meshing algorithm has been the first method tested and immediately discarded, given the required parameter tuning, watertighting and error correction required for a smooth mesh generation. The main parameters to be tuned, grid size and ISO level, has been chosen to be respectively (100, 100, 100) and 0.005. The first parameter determines the granularity of the reconstruction, and higher values translate in a denser grid used when determining the polygons, while the ISO level regulates the density threshold at which the generated mesh should be [1]. Figure 1 shows the resulting mesh, generated using the marching cubes algorithm.



(a) (b) (c)

Figure 1: Generated mesh using the marching cubes meshing algorithm.

The first thing that can be seen is that the reconstruction is quite poor, with multiple polygons extruding from the subject where there shouldn't be any. This problem can be solved by going through every polygon and determining its correctness by checking the neighbor normals. Another major problem is that the watertight property is absent and it should be added but, as this was not required, it hasn't been implemented. On top of the aforementioned issues, the required time for mesh reconstruction grows a lot when increasing the grid size, translating in a balance problem between quality of the reconstruction and elapsed time. Overall, this method gives poor results if fine tuning is absent, requiring a lot of specialized work.

A possible explanation for the poor results is that the used point cloud is not dense enough, so the tangent planes estimation fails to produce the correct results. This is one of the main disadvantages of this algorithm, reducing its applicability on specific problems [2].

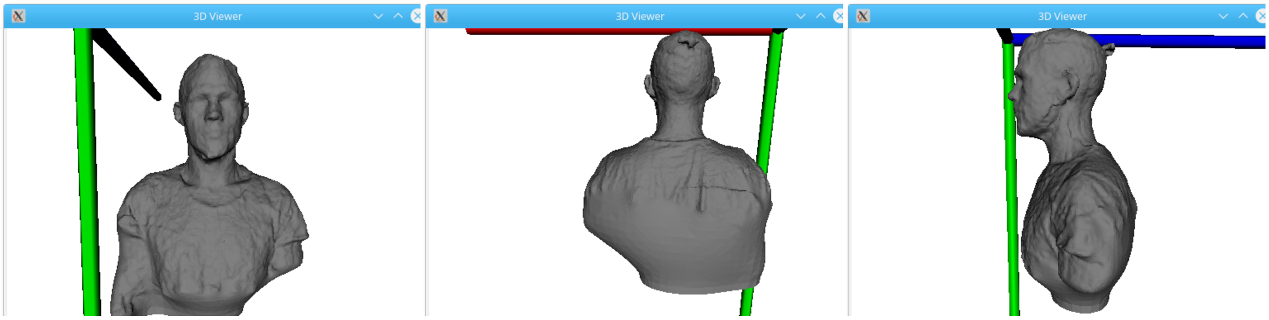
Poisson meshing

The second method tested, Poisson meshing, produced much better results, while integrating the watertight property and lower elapsed time. The main principle driving this method is the generation of a Poisson equation, whose solution gives the best fit surface of a dense point cloud [2]. The used functions didn't require any parameter tuning, resulting in a straightforward implementation, resulting in the mesh shown in Figure 2.

The result is clearly better when compared to the marching cubes one, and it has been chosen for the subsequent texturing. One major drawback of the Poisson meshing algorithm, is that it gives watertight meshes by default, and there could be instances where this is must be avoided, requiring the use of other algorithms. In this case, Poisson meshing is ideal.

Having created the mesh corresponding to the input point cloud, the next step is to apply texturing, resulting in a fully colored 3D model.

[1] `https://www.reddit.com/r/VoxelGameDev/comments/2ny8ys/need_help_understanding_marching_cubes_isolevel/`



(a) (b) (c)

Figure 2: Generated mesh using the Poisson meshing algorithm.

3 Coloring the 3D model

In this section, the procedure followed for the 3D colorization of the mesh is discussed, along with problems encountered and visualizations of various approaches used.

3.1 Occlusion detection

As multiple camera views have been used to record the 3D appearance of the model, the major problem to overcome is detecting, given the camera pose of a particular frame, if a certain point in the point cloud is occluded or not. Multiple options have been analyzed, and the final approach chosen was to iterate through every polygon of the mesh and calculating its normal w.r.t. the camera, in order to be able to know if it's visible or not. This condition is met when the z coordinate of the polygon's normal is negative, meaning that it is pointing in the hemisphere where the camera is located. In this project, however, setting the condition to be less than a small negative value instead of just zero improved the final results (in our case -0.3). In order to calculate the normal vector, 3 points composing the polygon have been checked (as 3 is the minimum number of points necessary for determining a plane) and the vectors that connect them are calculated. With these vectors, determining the normal is trivial, as a simple cross product is required. For each visible polygon the standard procedure described in the provided assignment has been followed, in order to extract the color associated with a certain point.

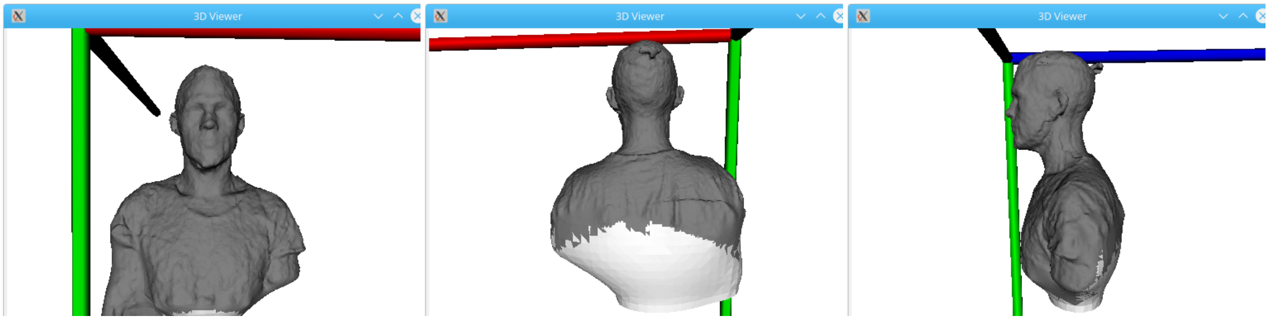
3.2 Coloring

The second major issue encountered in this section was the actual color to associate to each point: as multiple frames can see the same model part, multiple colors can be applied. In the case of homogeneous illumination this should not be an issue, as just the first frame that sees a point could be used, but this dataset lacks this feature. Instead, multiple illumination conditions are present in different parts of the subject, and different angles result in very different color values for each point.

To capture different colors associations a custom structure has been implemented, consisting of four vectors that will store the values given for the r, g, and b color channel from each frame, along with the total intensity (calculated as the sum of the three color values).

As the used mesh is produced with the Poisson method, watertight is automatically added, and consequently there will not be any color associations made with parts of the mesh not visible in the beginning. This parts consist of the lower bust of the subject, as shown in Figure 3.

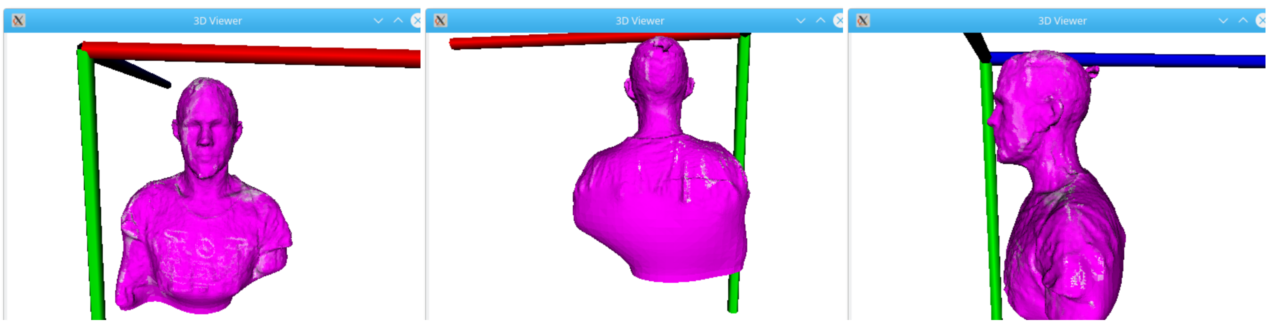
To show how different color values are assigned to the same point, Figure 4 shows the 3d mesh, overlaid with a variance map showing the areas subject to the highest changes in color values: low variance is indicated



(a) (b) (c)

Figure 3: Uncolored parts of the mesh.

as magenta, whereas higher variance shifts the color towards white.



(a) (b) (c)

Figure 4: Variance map of the colors.

We can see that there are a lot of points where there is high variance, meaning that different cameras associated different colors to the same element. This issue translated in high difficulty in creating a correctly and uniformly colored subject, and different strategies have been tested.

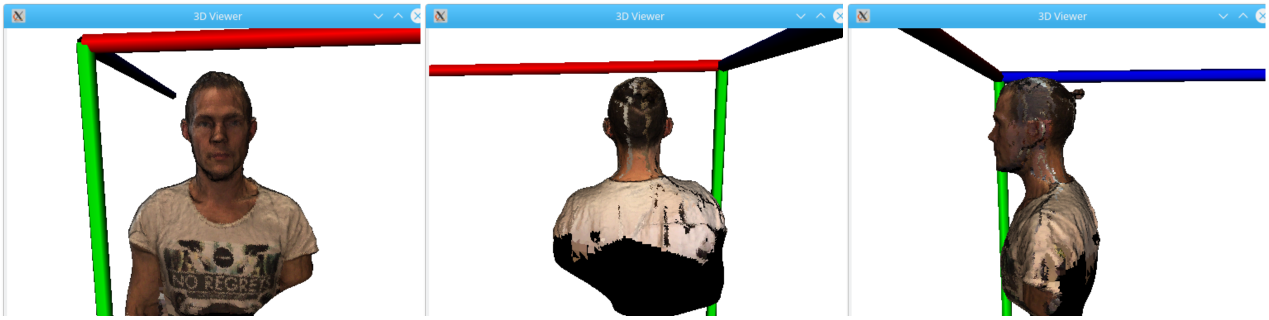
Picking the first color

The first color associated to each point has been used, without checking the values for all the other assignments. This method has the benefit of giving true colors to the points seen from the first frame and removing issues with multiple views, but ignores any other information arising from other cameras. Figure 5 shows the result obtained when using this method.

It can be seen that the frontal view of the model is perfect, with clear details especially in the face and the t-shirt, where words are clearly recognizable. Other views of the model show how this method lacks quality in every other frame, and overexposure is clearly visible in the back of the head. The next idea was to average each color association, and some initial testing has been done by averaging each channel.

Averaging the color for each channel

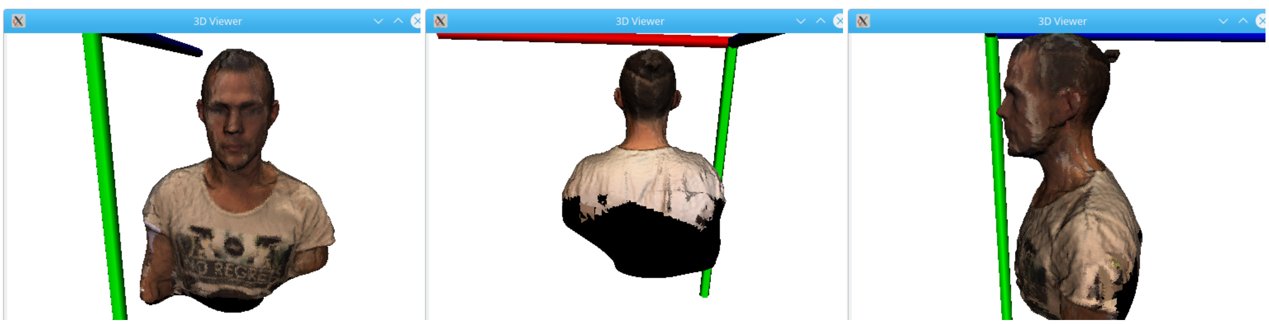
This method assigns, for each color channel, the average value obtained by checking all the association in order to smooth the imperfections of just using the first association. The results should in theory be bad,



(a) (b) (c)

Figure 5: Model colored with first association.

as averaging each channel separately should create very poor color combinations. Figure 6 shows the results obtained when using this method.



(a) (b) (c)

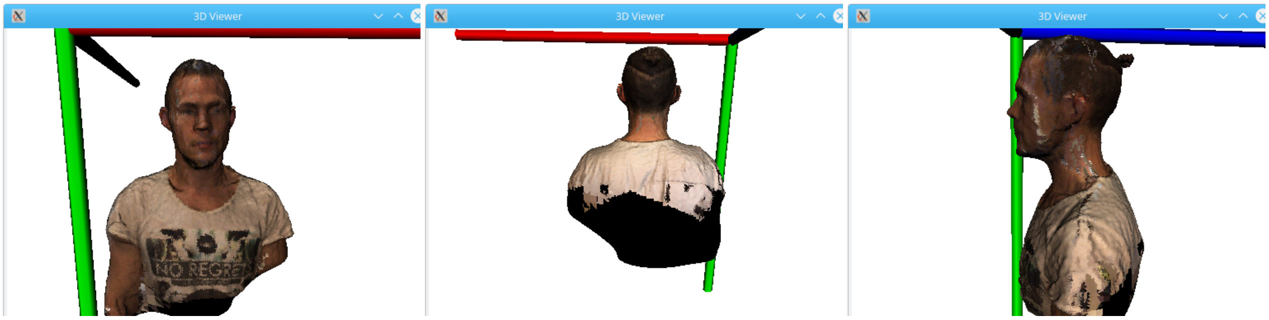
Figure 6: Model colored with the average value for each channel.

Interestingly, this method is not as bad as expected, and gave interesting results with somewhat recognizable details where present. The main improvement is at the back of the subject, where the color is now uniform and much closer to the true values. As expected, lot of the details vanish easily, as this method works similarly to a Gaussian filter over different views. Although the face of the subject now has white tones in the sides, this method revealed to be better than expected.

Color combination with average intensity

The next step was to use, instead of the channels as separate entities, the average calculated on the stored intensities: having the average value, the rgb combination chosen was the one having intensity that was closest to the average. This method should in theory be better than the simple per-channel averaging, and Figure 7 shows the results produced.

As expected, the results have better quality, and the details, especially on the t-shirt and the face, are now more clear. Although better, the frontal view is still not comparable to the first method implemented. When comparing the back view, it can be seen that is better than the first method and marginally better than the second, allowing average intensity to be the best method implemented up to this point.



(a) (b) (c)

Figure 7: Model colored with the rgb combination closer to the average intensity.

Median intensity

The final implemented method works by assigning the pixel color associated with the median intensity stored for every pixel: in this way, outliers obtained with overexposed pixel colors should vanish, resulting in a more accurate representation of the model and a more robust approach. When the associations for a certain pixel are only two, however, the rgb combination with lower intensity is chosen, in order to avoid having regions of the body too bright (especially in the face of the subject). Figure 8 illustrates the results obtained.

(a) (b) (c)

Figure 8: Model colored with the rgb combination having median intensity.

Overall, the main improvement can be seen in the back of the subject, where almost no artifacts can be seen, whereas in the right arm there is a visible white area, that appeared only when using this method.

Conclusions

The different methods exposed give very different results, and if constrained to the choice of one of them, average intensity seems the more reliable in all the body regions. If the frontal part of the subject is more important than the others, the first method is to be considered better: first association coloring in particular works best with the front view only because the order in which the reference frames are considered puts the front view first, so any part of the body could be colored with better accuracy by choosing first the view that has that part at the front. The quality of the results can be improved drastically by having a higher quality dataset, where illumination is constant for every camera angle and there are minor differences in coloring. Another approach could be implemented, by having better color values stored by removing outliers, but the proposed methods proved good enough for this project scope.

References

[1] Paul J. Bes, *"A Method for Registration of 3D Shapes"*, IEEE transactions on pattern recognition and machine intelligence, vol. 19, No. 2, February 1992

[2] Maiti, A., & Chakravarty, D.). *"Performance analysis of different surface reconstruction algorithms for 3D reconstruction of outdoor objects from their digital images"*, SpringerPlus, 5(1), 932. http://doi.org/10.1186/s40064-016-2425-9 (2016)