Report: Shape from Silhouettes

Rishabh Singh: 19-953-793

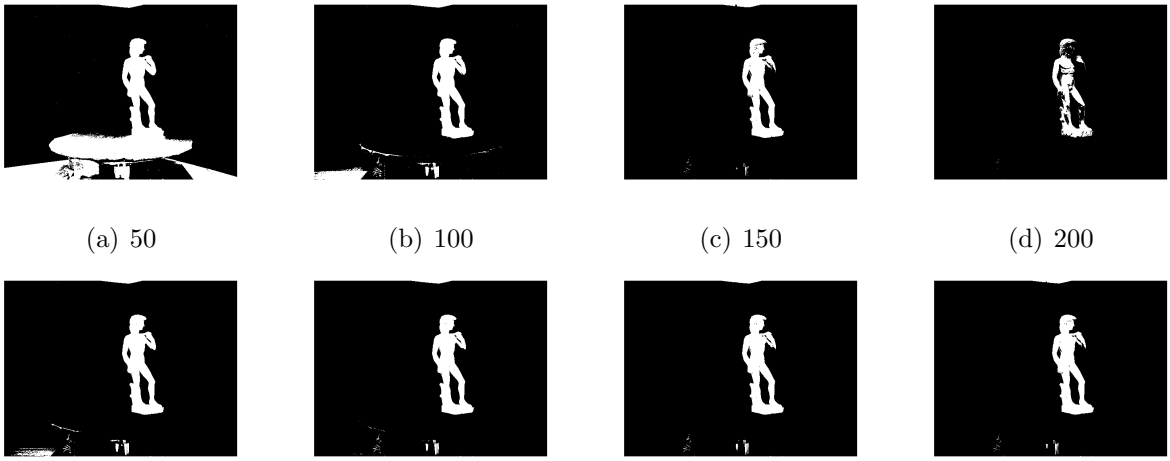
December 6, 2019

1 Silhouette extraction

The objective of this exercise is to reconstruct a 3D object using multiple calibrated images by silhouette extraction. The silhouettes are extracted by a simple binary thresholding method. If the intensity of a pixel is greater than certain threshold, *thresh*, the value assigned to the corresponding location in the binary image will be 1, else 0. As evident, the choice of threshold is an important step for accurate silhouettes to get extracted. The values of *thresh* were varied and the extracted silhouettes were analysed critically as shown in Fig.1. As the values were increased, erosion of the object of interest was observed. For instance, for values equal to and greater than 120, the left arm of the object shrunk in width. Similarly, for values below 90, a lot of information from background was present in the segmented image. Based on the quality of final reconstructed shape, **thresh value of 100 was chosen**. In order to automate this step and choose an intelligent threshold value based on image properties, I went through the paper [1] and implemented it, which is discussed broadly in section 4 on improvements.

2 Volume of Interest

Choosing the volume of interest was a hit and trial method, where the minimum and maximum values of the coordinates of the bounding boxes were chosen based on the final output. Initially the values were kept quite optimistic and then later refined to make the bounding box tighter and get better resolution. The final values were chosen as: **minX = 0.25, minY = -0.1, minZ = -1.8, maxX = 2.1, maxY = 1.1, maxZ = 2.5**, as shown in Fig.2.
The resolution of the grid was initially kept coarse, but later on increased to a larger size. The results for higher resolutions appear more finer as shown in Fig.3. The resolution value used in generating results was **64x64x128**.



(a) 50 (b) 100 (c) 150 (d) 200

(e) 110 (f) 120 (g) 130 (h) 140

Figure 1: Silhouettes extracted at different threshold values. While the lower values tend to keep background information, higher ones lead to erosion of the object in question too.

3 Visual Hull

In order to calculate the visual hull, we consider each voxel in the grid we initialised above and compute its centre point. The resultant point is transformed from volume to world coordinates using the transformation as mentioned in the code provided.These are then projected from world to image coordinates using individual Projection matrices provided. We homogenise these points. We then add 1 to the voxel if the projection of its centre lies within a silhouette region. We finally extract iso-surface from the volume obtained using threshold value of 17 as shown in Fig 4.

4 Improvements

4.1 Silhouette Extraction

The key step in the performance of this method is how accurately the silhouette is extracted from the actual image. In our case, the object got sufficiently separated from the background using basic binary thresholding, but this may not be the case where objects have a lot of interactions with the environment. Also, manually selecting threshold based on the appearance of silhouette is rather subjective and time consuming process. I read the paper on learning an intelligent threshold from an image itself [1] and implemented it in the function *select_thresh*. It is intended to calculate the best threshold value on a grayscale image. From the grayscale image, *I(x,y)*, the histogram function could be formed. The histogram function was then differentiated. Differentiation of a histogram with bins of $(2^8 - 1)$ will result in $(2^8 - 1) - 1$ bins. For the last value $g(255)$, it is initialised to 0. We now apply *signum* function on $g(x)$ to assign value 1 to positive values, -1 to negative values and 0

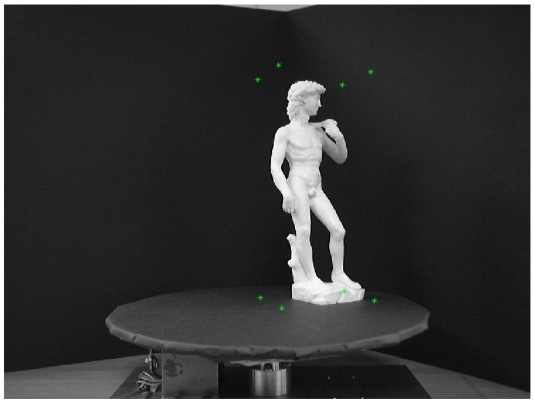


Figure 2: The corners of the bounding box shown in green

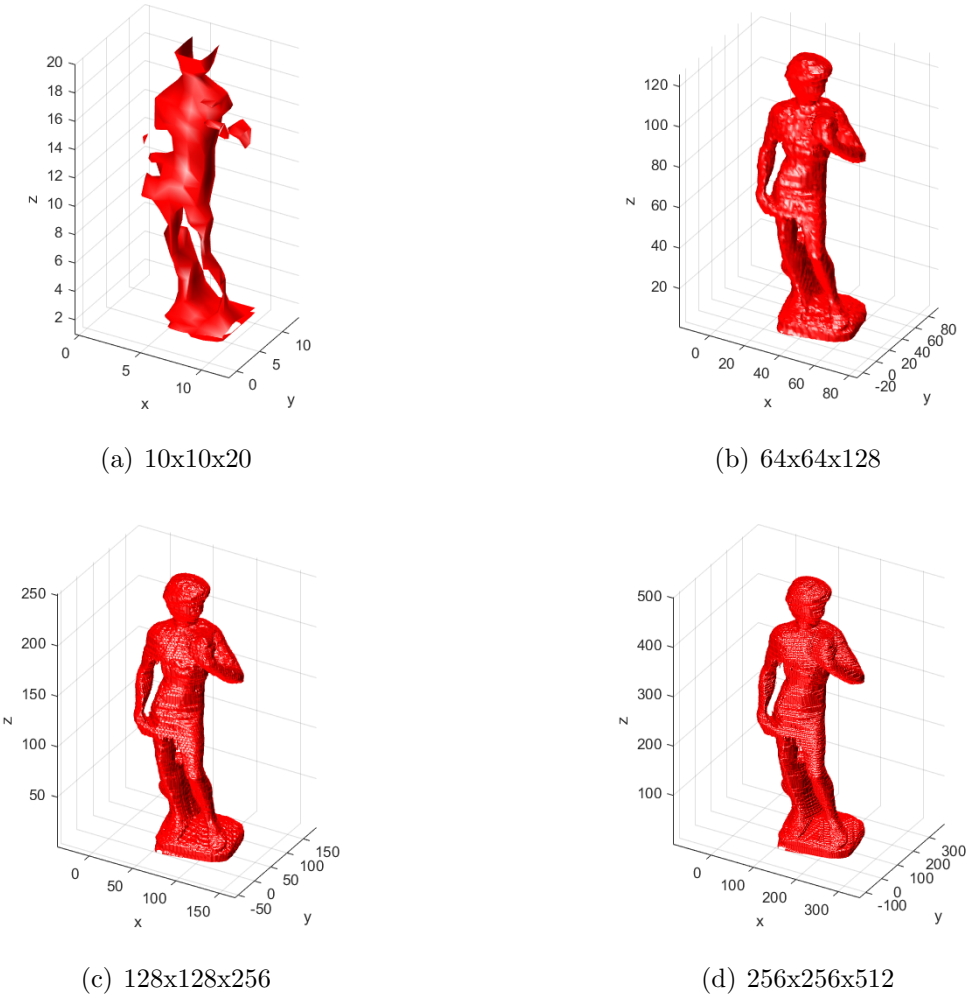
otherwise. Let's say this function is $g_2(x)$. We then identify the number of peaks and valleys based on the logic as stated in [1]. The mean of histogram values at the identified valley points are calculated. A distance metric as defined in [1] takes into account the deviation of histogram values from the calculated mean plus a differentiation term. The argument value for which this distance is minimised is taken as threshold value. I observed that the square of the differentiation term in the distance metric was leading to eroded segmentation and thus, replaced it with absolute value. This generated more plausible threshold results. The final threshold over the set of images clicked by each camera could either be the maximum, minimum or the average of these individual thresholds. Eventually, mean value was chosen based on experimental observations (Fig. 5).

4.2 Number of cameras

According to [2], this method gives good results only when the object lies within the intersection cone of camera projections. Thus, the performance depends heavily on the number of cameras and their relative placements. We used 18 images in our case and thus the results obtained were agreeable. Reducing the number of cameras lead to inferior performance as shown in Fig.6. This limitation has been addressed in [2] where they rely on the key idea of using a subset of cameras when deciding if a 3D point represents a foreground object as opposed to Shape from Silhouettes which uses the complete set of cameras.

4.3 Ghost particles

The intersection of visual cones of cameras can also occur in regions where there is no actual object present, as seen in Fig.7 where some ghost objects were spotted below the arm of the actual object. This limitation is addressed in [2] where they find the subset of the connected components of the Visual Hull that contain real objects only.



(a) 10x10x20

(b) 64x64x128

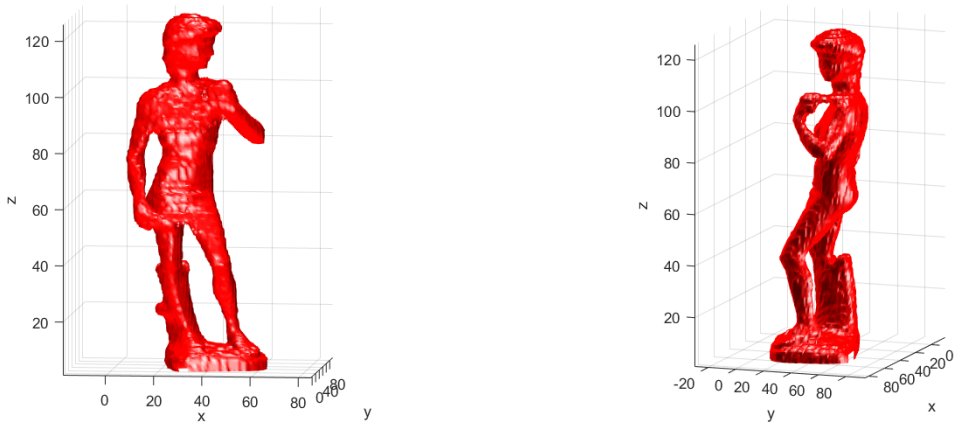
(c) 128x128x256

(d) 256x256x512

Figure 3: 3D reconstruction for different scale values. Higher resolutions appear more finer

4.4 Concave(Hidden) regions

Some regions can be hidden in the silhouette extracted because of the illumination effect. Regions such as cavities on the face and thorax will always be invisible in the silhouette, no matter the angle of the cameras. We need additional information in the image such as depth information (like disparity map for each image), shading, texture, albedo of the surface, reflectance map of each image, illumination conditions etc. in order to reconstruct these hidden regions. Since the example discussed here is of grayscale nature, shading can in particular give some cues on such regions. The results can be further improved by using a prior information based on the contour of the surface instead of a rectangular bounding box.



(a) front (b) side

Figure 4: 3D reconstruction for resolution 64x64x128

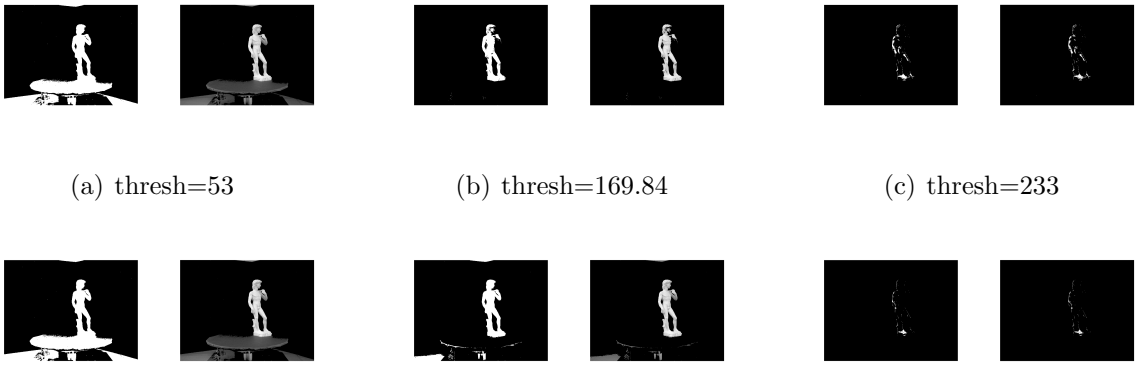
4.5 Depth-Discontinuities

Depth discontinuities can arise in any part of the object, not necessarily in the concave regions. In order to solve this, we can model this problem as a graph coloring algorithm where each voxel is treated as individual node. The voxel can be colored based on the RGB or grayscale value at that location using some wieghting function, where the weights are determined by the depth information. The depth information can be extracted from stereo-vision.

References

[1] A. Ismail and M. Marhaban, "A simple approach to determine the best threshold value for automatic image thresholding," in *2009 IEEE International Conference on Signal and Image Processing Applications*, pp. 162–166, IEEE, 2009.

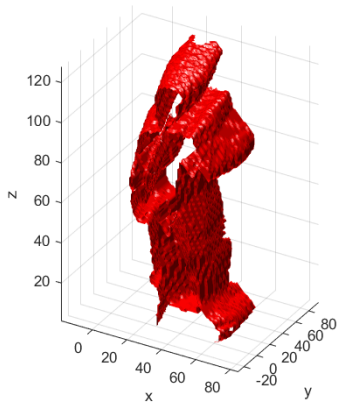
[2] B. Michoud, E. Guillou, H. M. Briceno, S. Bouakaz, *et al.*, "Silhouettes fusion for 3d shapes modeling with ghost object removal,"



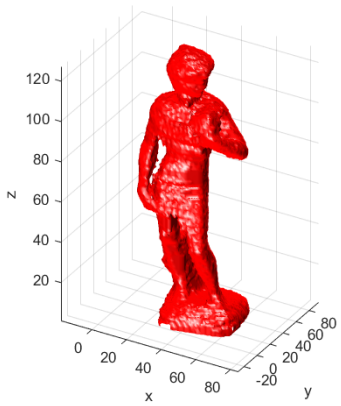
(a) thresh=53 (b) thresh=169.84 (c) thresh=233

(d) thresh=51 (e) thresh=97.2778 (f) thresh=241

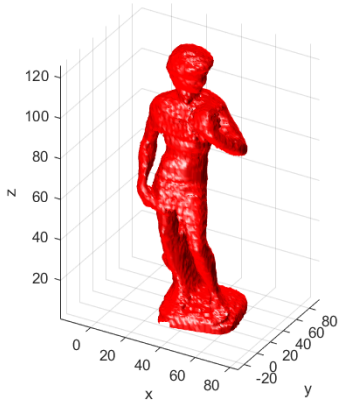
Figure 5: (top) the minimum, mean and maximum of threshold values over different camera images when square of differentiation of histogram values were used in distance function.(bottom) same, when absolute value of differentiation was used. Case (e) was selected as the final threshold measurement



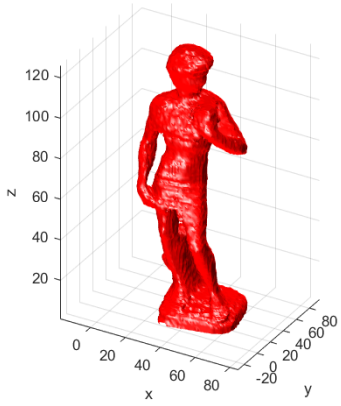
(a) 2 cameras



(b) 7 cameras



(c) 12 cameras



(d) 18 cameras

Figure 6: 3D construction obtained from images taken from different number of cameras. The final results are inferior in quality for lesser number of cameras.

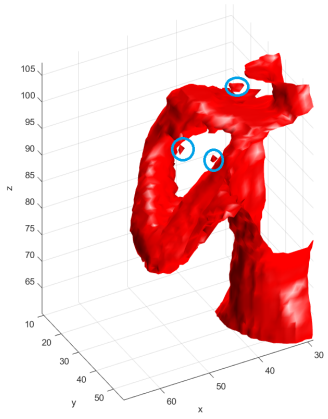


Figure 7: The objects in blue circles were not the part of the main object.