
Exercise 5 - Shape from Silhouettes

Computer Vision

Alberto Montes (malberto@student.ethz.ch)

November 18, 2016

Silhouette extraction

The first task consists on setting the best threshold to extract the image silhouette. With the given images, the best threshold found was 100 which extract all the figure including the darkest zones because of shadows. It also extract some other places of the environment, but with the bounding box, the noisy extracted parts disappear. On Figure 1 there a snapshot of the silhouette extraction.



Figure 1: Silhouette extraction

Volume of interest

Once the silhouette is extracted, it is required to specify the bounding box that will define the volume of interest for the computation of the Visual Hull. After some iterations, the final values for the bounding box to fit the object has been:

$$\begin{bmatrix} x_{min} & y_{min} & z_{min} \\ x_{max} & y_{max} & z_{max} \end{bmatrix} = \begin{bmatrix} 0 & -0.5 & -2 \\ 2.5 & 2 & 3 \end{bmatrix} \quad (1)$$

On Figure 2 there is a snapshot of the bounding box represented over the silhouette.

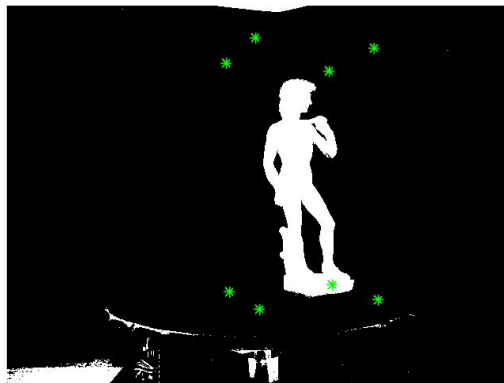


Figure 2: Volume of Interest

Visual Hull

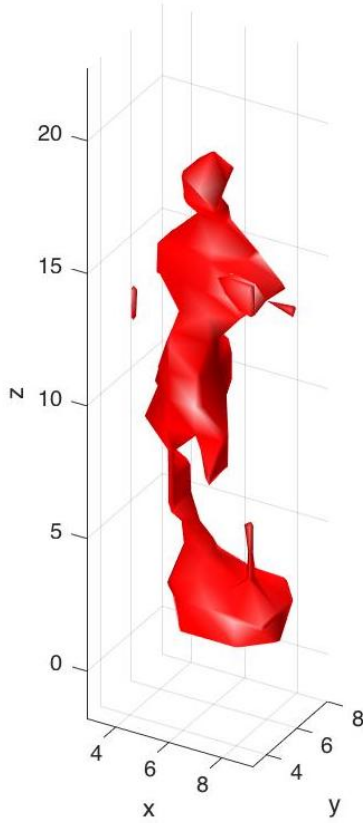
The final step to achieve a 3D reconstruction of the original object at the images, is to compute the Visual Hull. To do so, it has been divided the bounding box into discrete voxels. Then, for each of the voxels, it has been computed the projection of this voxel into each of the captured image and compute in how many images, the voxel is projected into the silhouette. Finally all the voxels with at least `thresholdVolume` are considered as iso-surface and recreate the original object in 3D.

The implementation of this task is on Listing 1 and a pair of details has been taken into account. The first one is that for each voxel it has been computed the projection of the mid point of the voxel, reason why on line 12, on each axis is subtracted 0.5. The other detail is taking into account the case where the projected 3D points lays out of the 2D image. In this case we have to ensure the continuity of the program execution.

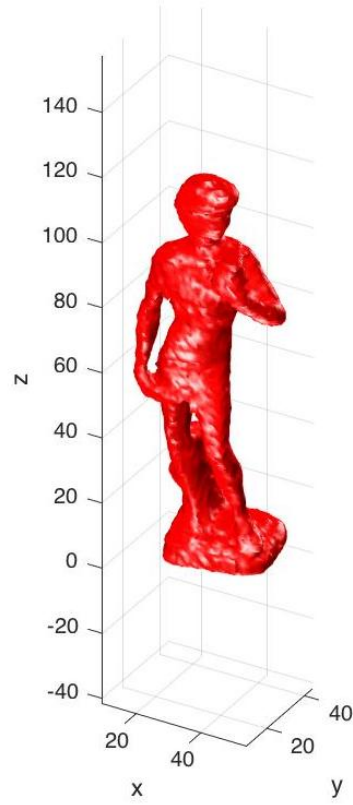
```
1 if computeVisualHull
2     % Define volume.
3     volume = zeros(volumeX,volumeY,volumeZ);
4
5     % Visual hull computation
6     % Task 7.3 Visual hull computation
7     % - add one to volume if projection is within silhouette region
8     for n = 1:numCameras
9         for i = 1:volumeX
10            for j = 1:volumeY
11                for k = 1:volumeZ
12                    p = [i-.5 j-.5 k-.5 1]';
13                    p_img = Ps{n} * T * p;
14                    p_img = round(p_img ./ p_img(3));
15
16                    % Check if the projected point ends out of the image
17                    if p_img(1) <= 0 || p_img(1) > size(sils{n}, 2) ||
...
18                        p_img(2) <= 0 || p_img(2) > size(sils{n}, 1)
19                            continue;
20                    end
21                    volume(i,j,k) = volume(i,j,k) + sils{n}(p_img(2),
p_img(1));
22                end
23            end
24        end
25    end
```

Listing 1: computeVisualHull

The results of the iso-surface are plot on Figure 3 where different resolutions have been used for dividing the volume of interest. In the first case (with a lower resolution) it can be seen how the arms and one leg are not distinguishable and the quality is improvable. On the other hand, increasing the resolution of the volume of interest, it can be observed a well defined object, with a very similar shape in comparison to the original.



(a) Volume size: (10, 10, 20)



(b) Volume size: (64, 64, 128)

Figure 3: Visual Hull for different resolutions

Improvements

The method that it has been already implemented consist from a set of images from different angles to the same object reconstruct its 3D model. To do so, first the silhouette of the object is extracted from each image captured. Then knowing the projection matrix of each camera, and a bounding box around the object, the bounding box is converted into voxels and each one is projected to each of the images with the silhouettes. If the voxel lie inside the silhouette for more images than a threshold, this voxel is considered as part of the volume. Doing so it is possible to find the 3D model finding all this voxels that lie inside enough silhouettes.

This method has the little problem that if the surface of the original object has concave surface, then this method won't be able to correctly identify.

In order to improve this method it can be exploited more information given by the images. This information is referring to the texture of the object. To exploit this information, at the time to compute the visual hull with the silhouettes, when a voxel is projected inside a silhouette, the color value of the projection at the image can be stored, and at the end, for each voxel compute the mean color value and plot it when representing the 3D model.