Assignment 2: Camera geometric calibration

Rick Droogh 6977170

This is a report for the extended second week's assignment of the course computer vision. In this report all parts of assignment 2 are shown. The code is available on a private Github repository at https://github.com/rickdr/Computer-vision, access could be granted.

I. Quality of intrinsics

To calculate the intrinsic matrices. First, we compare all of the frames, from each intrinsics.avi video, whether it has changed by more than a threshold of 250.000. Secondly, the frames that have surpassed the threshold are save for the next process. Finally, the saved frames are passed to the process of Assignment 1, in which the intrinsic matrices are calculated and saved in intrinsics.xml files. For the comparison we created a frame of the changes by using the ”absdiff” function over the previous saved frame and the new frame. The frame of changes then is compared by ”countNonZero” with the threshold. The threshold is found by trial and error of extracting a fare amount of frames.

Bellow, the intrinsic matrices of the different camera's are shown.

$$\begin{bmatrix} 493.43 & 0 & 332.48 \\ 0 & 496.56 & 219.56 \\ 0 & 0 & 1 \end{bmatrix}$$

Fig. 1. Intrinsic matrix cam 1

$$\begin{bmatrix} 487.58 & 0 & 330.194 \\ 0 & 489.58 & 229.89 \\ 0 & 0 & 1 \end{bmatrix}$$

Fig. 2. Intrinsic matrix cam 2

$$\begin{bmatrix} 487.40 & 0 & 320.00 \\ 0 & 486.22 & 237.26 \\ 0 & 0 & 1 \end{bmatrix}$$

Fig. 3. Intrinsic matrix cam 3

$$\begin{bmatrix} 490.67 & 0 & 343.61 \\ 0 & 492.97 & 241.09 \\ 0 & 0 & 1 \end{bmatrix}$$

Fig. 4. Intrinsic matrix cam 4

II. Quality of extrinsics

The extrinsic matrices are calculated using the corner click process. Attempts to automate this process have not been successful. The configuration files containing the matrices could be found in the delivery folder or in the Github repository.

III. Ingenuity of background subtraction method

In order to subtract the foreground from the background, we need to find the optimal HSV-values. To accompish this, we use the given code that splits the HSV-channels and calculates, for each of the channels, by using the ”Absdiff” function the difference of ... Then, we calculate the standard deviation of the difference by using the ”MeanStdDev” function and use it as the new value for the threshold of each channel. The V-value of the HSV-values controls the light of the HVS-channels, by multiplying the calculated standard deviation by 2, the shadow of the ”horse-man” is removed. Finally, after calculating the new HSV-values, we remove noise by using the ”Dilate” and ”Erode” functions over the foreground with a ”GetStructuringElement” kernel using ”MORPH_ELLIPSE” and size 2x2.

IV. Quality of background subtraction

The background substraction could be seen bellow, or in the video: https://youtu.be/Ct_52D9i3oo.

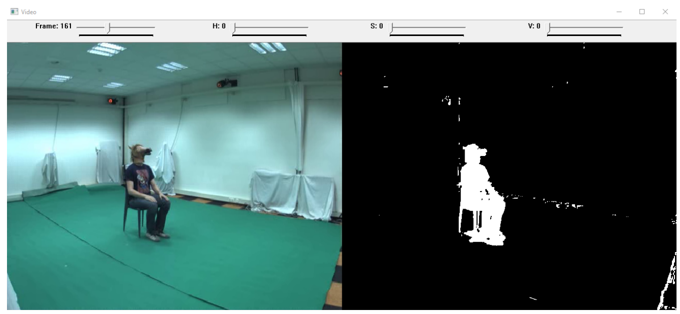

Fig. 5. Background subtraction example 1

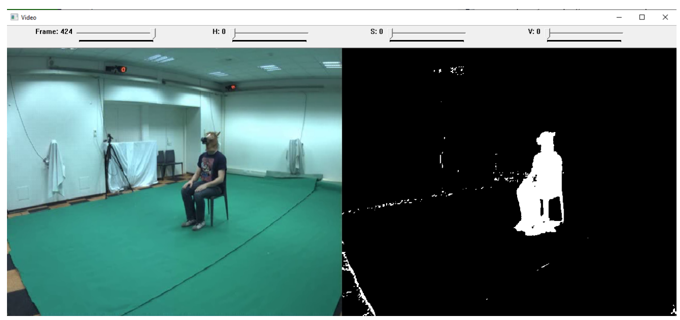

Fig. 6. background subtraction example 2

In both images, it could be observed that there is still some noise in the background. Also, although a attempt to remove the shadow is in place, part of the shadow is still there.

V. Quality of voxel model

The voxel model could be seen bellow, or in the video: https://youtu.be/2uy7jMY2qbI.

In image 1 the Voxel model is shown. In image 2, a problem in the camera could be observed. After a few times recalibrating the camera, this is the optimal view.

VI. Coloring the voxel model

Altough the voxels in the shown image, there is a theoretical attempt to color the voxels. In the ”initialize” function of the reconstructor the color of the foreground pixels is taken from

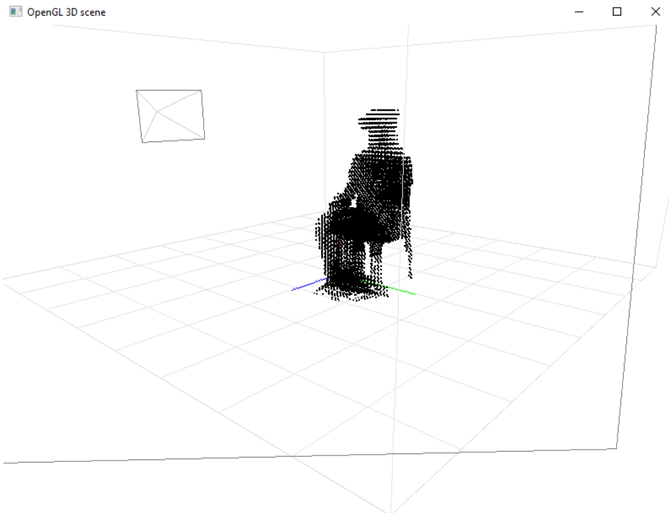

Fig. 7. Voxel model

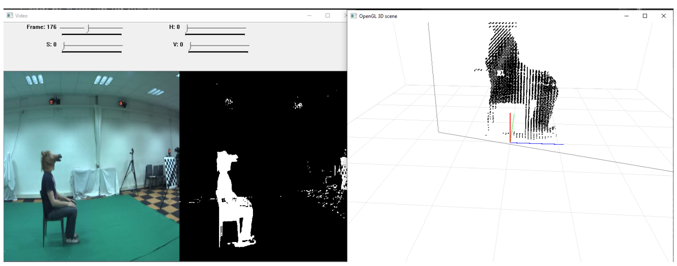

Fig. 8. Problem cam 3

the position of the projected voxel point. Next, the colors are saved in the voxel and in the Glut "DrawVoxels" function, the passed color is given to the voxel. The problem here is probably in the extrinsics, the camera's are unfortunately not correctly calibrated.