EEE3032 – Computer Vision and Pattern Recognition

Lab Worksheet 9 – Multiview Geometry

Miroslaw Bober (m.bober@surrey.ac.uk)

Task and Learning Objective: The purpose of today's lab is to experiment with a calibrated multi-camera setup, using it to perform 3D reconstruction through a visual hull.

Resources: The coursework and these exercises will be performed using Matlab. Download the Matlab code for the labs, if you have not done so before, and unzip the code into a separate folder. Ensure the code folder is in the Matlab search path (File -> Set Path.. -> Add with subfolders).

Ex1: Visualise the Cameras *(5 minutes)*

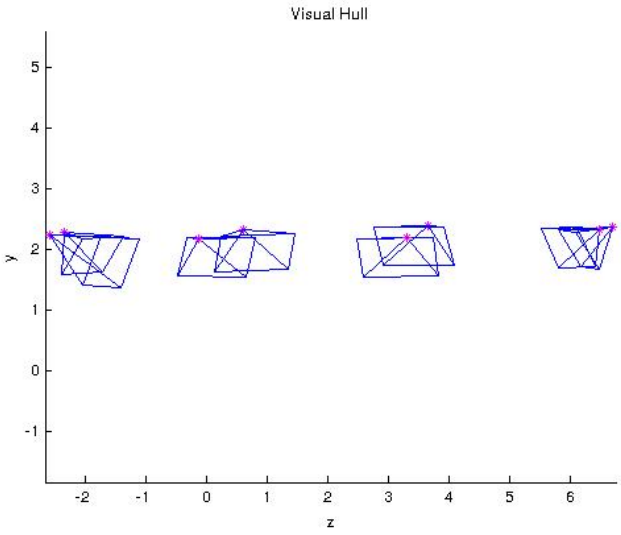
You change into the subfolder `mvdata` which is within the `vishull` folder of the lab code. This folder contains a text file `calibration.txt` containing the calibration data (the extrinsic and intrinsic parameters) for 8 cameras. There are also 8 images captured simultaneously of an object (a dancer dressed in yellow) from the 8 cameras.

Type:

```
>> C=loadCamera()
```

to load this information into variable C. You can inspect the settings for each of the 8 cameras e.g. `C(1)` gives you a struct containing the extrinsics (R and T) and the intrinsics (fx,fy,cx,cy) for camera 1.

You can visualize the camera positions (extrinsics) using the `plotCameras()` function.



Select the Matlab 3D rotation tool in the figure (the little circle of arrows) and drag the mouse to view this 3D figure from different viewpoints. The cameras are in a ring approx 2 metres off the floor (the x-z plane).

Ex2: Generate Masks *(15 minutes)*

Read in one of the multi-view images (cam1.png through cam8.png) and attempt to convert this colour image into a thresholded image (binary mask) exploiting the blue colour of the background. You may use any of the

techniques or code provided on the course. Try to pick a suitable threshold value. You should find a value that works well across all 8 images. A typical output is given below.

Using colour alone it will be impossible to remove some background elements such as the camera rig/gantry. This is not a problem for the purposes of creating a visual hull – why?

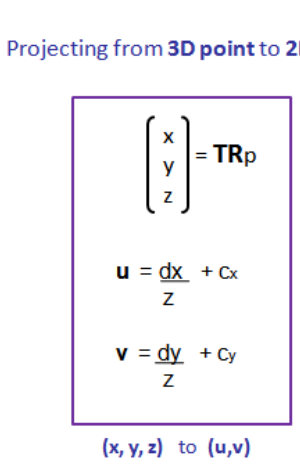
Ex3: Generate 3D Reconstruction *(25 minutes)*

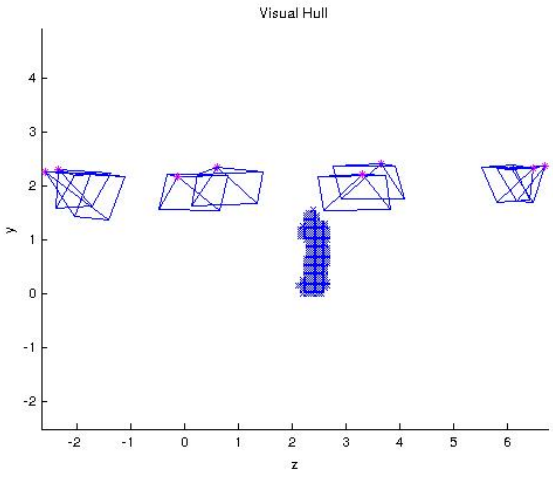
We will now combine the camera calibration data, and the thresholded images, to produce a 3D visual hull.

You should edit the program `vishulldemo.m` provided in the lab code. This program defines a 3D space x=[0,4], y=[0,2], z=[0,4] in metres and decimates this space into voxels of $5cm^3$.

The program iterates through each voxel and projects its (x,y,z) position to each of the 8 camera viewpoints, to obtain its projected 2D position (u,v) in that viewpoint.

However the code to perform this projection is missing (a few lines marked 'TODO'). You must write this code implementing the Maths from the lectures (reproduced below)





You should end up with a 3D plot depicting the 3D volume occupied by the dancer, as below.
When you have the program working, experiment with the various constants in CAPITALS at the top of the program. For example, try varying `VISHULL_THRESHOLD` to redefine the minimum number of cameras in which the (u,v) point must be white in order to fill a voxel (currently 5 of the 8 views).