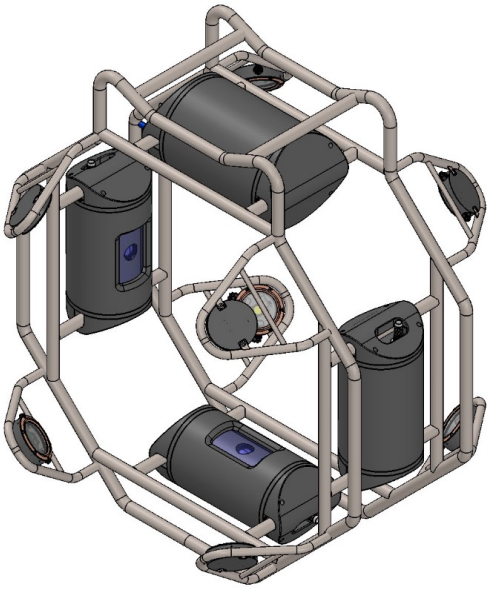
CS 423/523 Introduction to Computer Vision Project

Due Date: 10 Jan 2024 @ 23:55

Volume Reconstruction (with Python + OpenCV + Matplotlib)

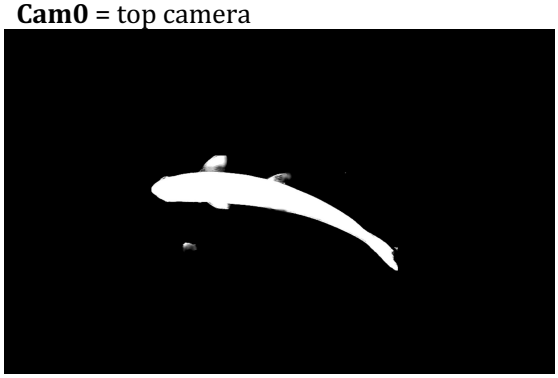


In this assignment, your objective is to reconstruct a 3D world within a closed system observed by four distinct cameras. Visualize the device as described here: it includes four separate compartments, each housing a camera that is oriented towards the device's origin. Select any camera as a reference; the subsequent camera in a clockwise direction is positioned at a precise 90-degree angle around the axis. All cameras are meticulously calibrated to focus directly on the origin. Additionally, the device is equipped with lights strategically placed outside the field of view to avoid obstructing the cameras' vision. Objects passing through this tunnel-like device are captured in real time. Furthermore, the cameras are time-synchronized to ensure they capture images simultaneously. The camera sensor and lens parameters are given to you:

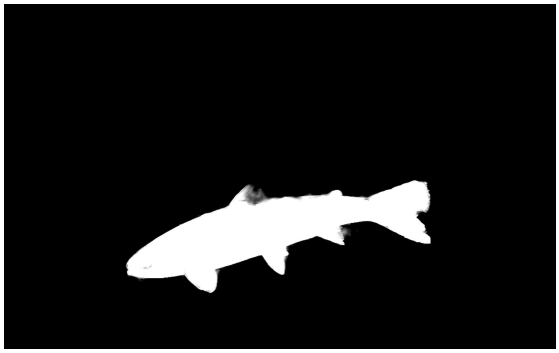
Sensor size in world coordinates:	11.3 mm x 7.1 mm
Focal length of the camera lens:	8.2 mm
Camera distance to the origin:	420 mm
Sensor resolution:	1920x1200

Assume a pin-hole camera model, and a distortion free lens setup.
(20 pts) Create a 3d voxel grid between the cameras.
(20 pts) For each voxel, trace a light ray through the pinhole of a camera and find where the light coming from the voxel will hit the sensor in 3d camera coordinates.
(20 pts) Convert 3d camera coordinates of the light ray hitting on the sensor to the image pixel coordinates.
(10 pts) Do this for all four cameras and check with their supplied object masks. If most of the cameras say that there is an object in that voxel, then assume the voxel is filled by the object.
(10 pts) Calculate the volume of the object as cm^3.
(10 pts) Visualize the 3d voxel grid.
(10 pts) Write a report for the findings and elaborate on possible enhancements on this algorithm.
(20 pts bonus) Write your program with matrixes (without any nested for-loops)
You'll be given the silhouettes of the object within the device from 4 different cameras as below in the order of **cam0**=top, **cam1**=front, **cam2**=back, **cam3**=bottom cams:

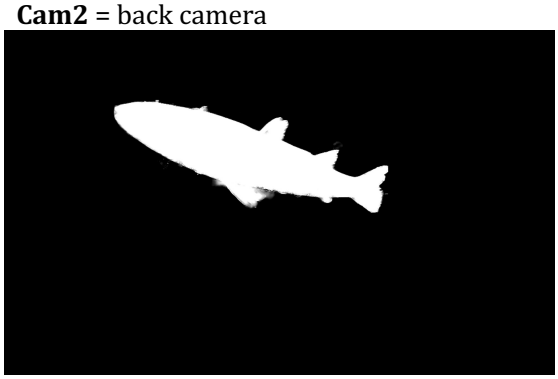
Cam0 = top camera



Cam1 = front camera



Cam2 = back camera



Cam3 = bottom camera