Assignment 1

Task 1. Find orientation of Pyrender and Augment YCB object properly.

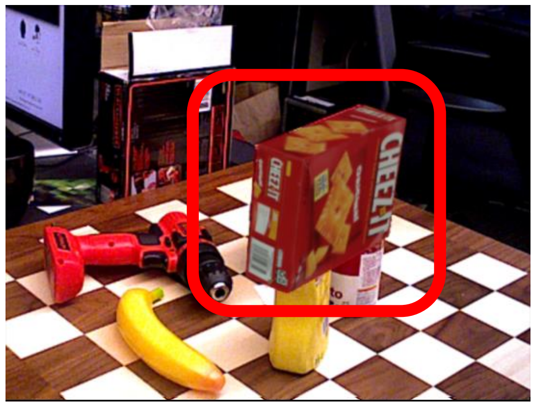
1. Check slide 25 to see how to find the orientation of the Pyrender (right)
2. Check slide 26 for YCB dataset orientation (right)
3. Check slide 29 for orientation conversion between right and right

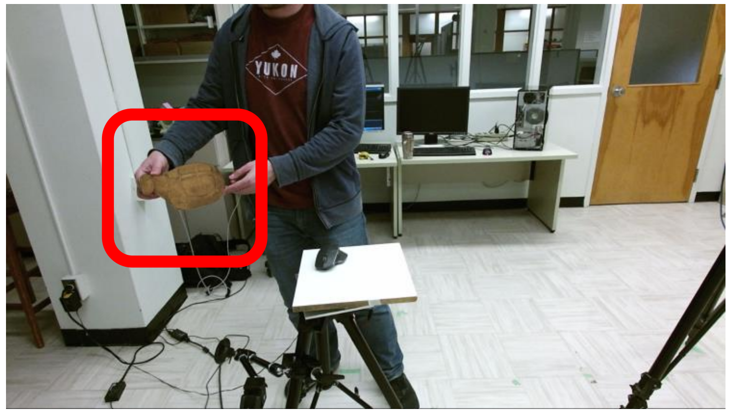
Task 2. Find orientation of Laval dataset (right) and augment the object properly.

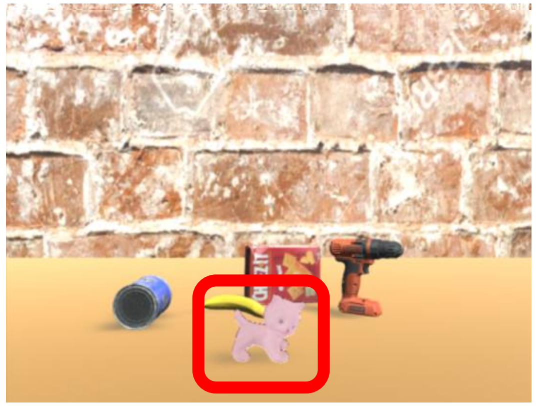
1. Check slide 27 to see how to find orientation of the unknown dataset
2. Check slide 29 for orientation conversion between right and right

Task 3. Try to convert Unity pose (left) into Pyrender pose and augment the object properly.

1. Unity is Left-handed system. check slide 30 to see how to deal with conversion btw left and right


Result from Task 1

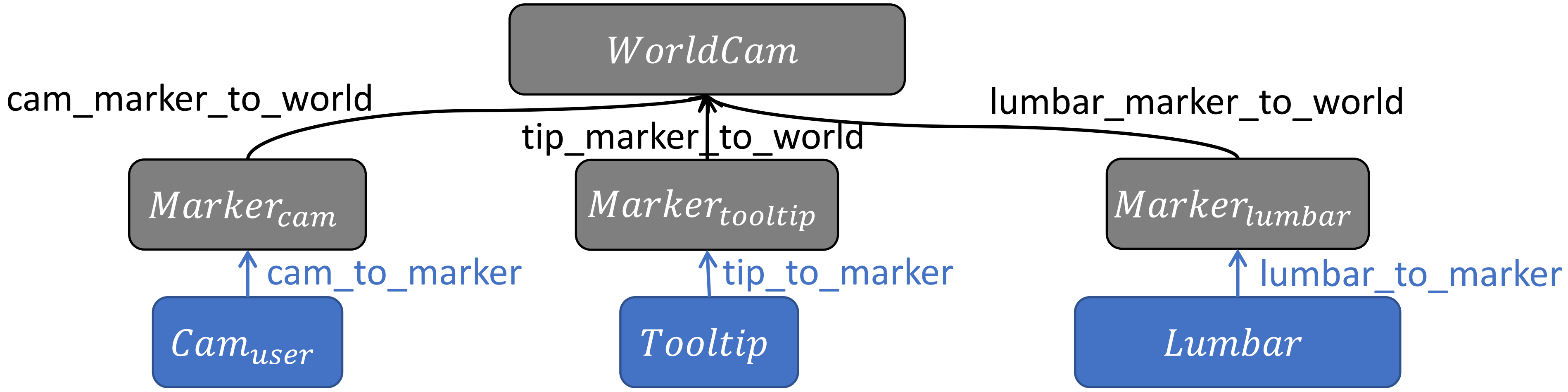

Result from Task 2


Result from Task 3

Assignment 2

Task 1. Cameras are not yet calibrated. Given checkerboard images taken from the user's camera, get camera intrinsic K (here we assume all cameras have same intrinsic)

1. Check slide 34 for details about camera calibration code.



Task 2. Given the scenario and relative pose graph, render the objects properly on world camera

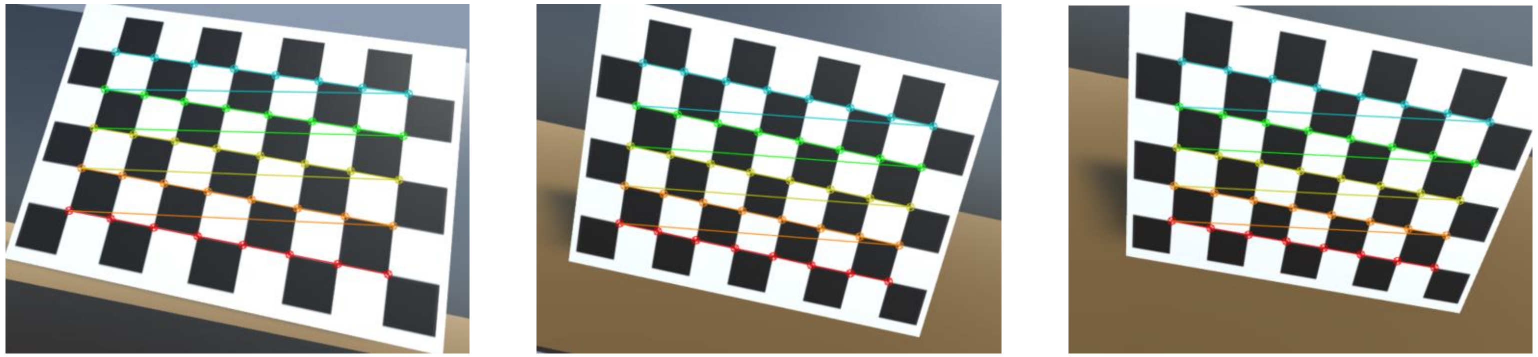
1. Try to augment tool tip and target on WorldCam using K from task1.

Task 3. With given pose graph, Now render the tool tip and target on user camera.

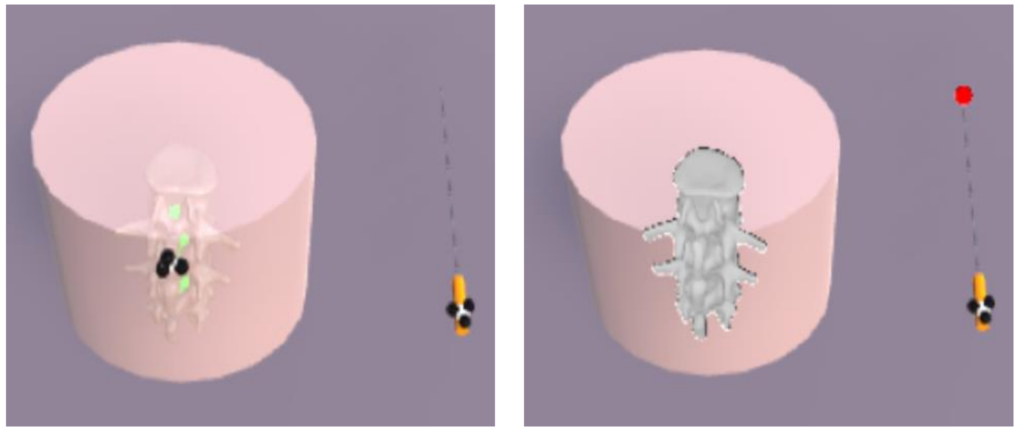
1. Details about traversing the graph is in slide 35
2. Try to augment the tip and target on Cam_{user}(use same K)

Assignment 2

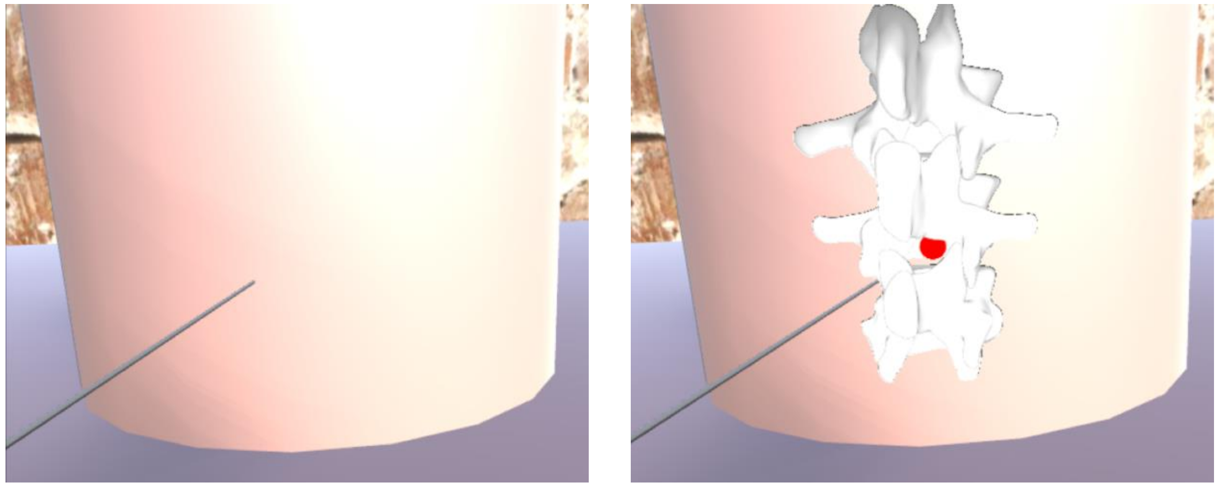
Example of correctly detected checkerboard (task1)



Augmented lumbar and tooltip



Task2 (seen from world camera)



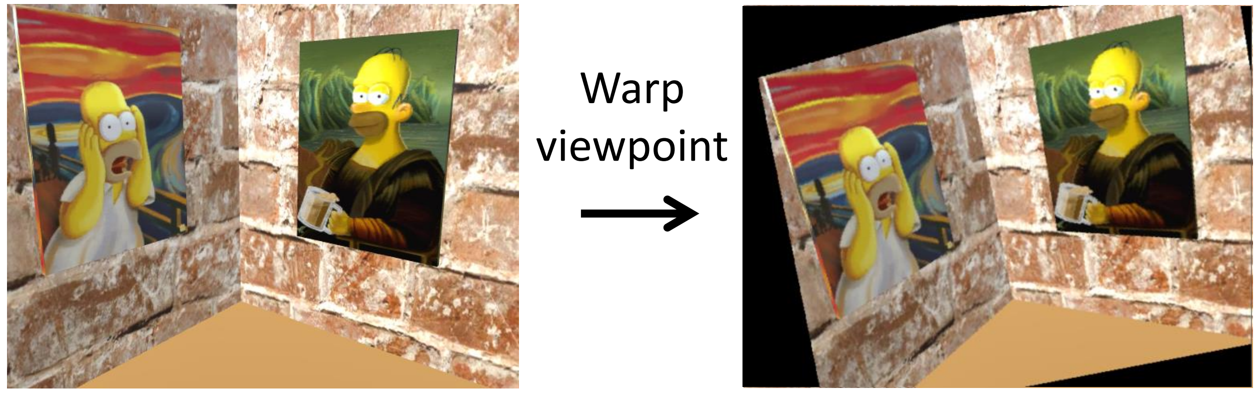
Task3 (seen from user camera)

Assignment 3

Task 1. Given the scenario and relative pose graph, implement 1) forward point warping of given pixels of camera0 (RGB_0) into camera1's viewpoint (RGB_1) using depth image of camera0 ($Depth_0$) and camera to world poses ($T_{c_0 \rightarrow World}, T_{c_1 \rightarrow World}$), 2) Bilinear interpolation

1. Check slide 40 for step-by step procedures of 3D warping
2. Check slide 41 for bilinear interpolation

$$\begin{tikzcd}
\boxed{Cam_0} \arrow[d, bend right, "Pose_0"'] & & \\
\boxed{World(0,0,0)} & & \boxed{Cam_1} \arrow[ll, bend right, "Pose_1"']
\end{tikzcd}$$



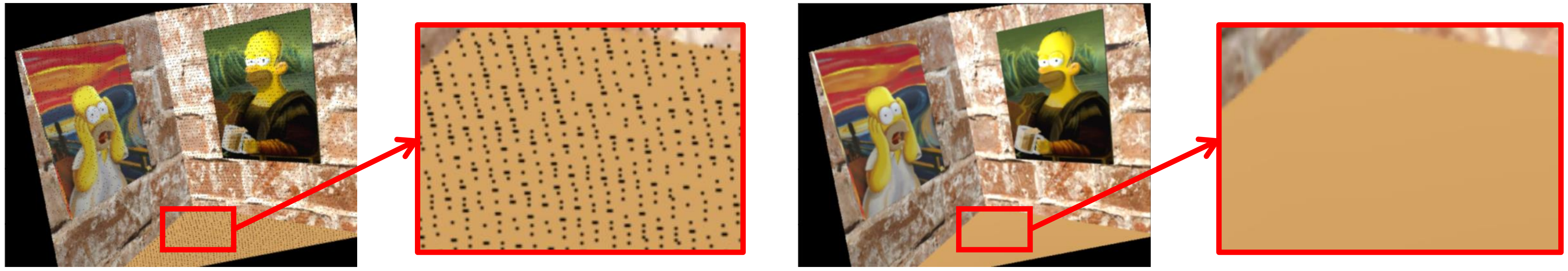
RGB_0 RGB_1

Task 2. Using pipeline and function from Task 1, implement both forward and inverse image warping with nested for loops and vectorized way (optional)

1. Check slide 9 for details of difference between forward and inverse warping
2. Once the pipeline works with for nested loops, try to replace them with vectorized pipeline

Assignment 3

Comparison between forward warping VS inverse warping (task2)



Comparison between nearest neighbor interpolation VS bilinear interpolation (task2)