

# 15-663 HW6

Sajal Maheshwari  
sajalm@andrew.cmu.edu

December 14, 2020

## 0 Collaborations

I collaborated with my flatmate Shamit Lal (shamitl) for the capture of the video for the self captured images. We also borrowed the setup from Sachit Mahajan(sachitma) to capture the images.

## 1 Implementing structured-light triangulationo

### 1.1 Video Processing

#### 1.1.1 Spatial shadow edge localization

The spatial edges for the various objects can be seen in Figure 1.

#### 1.1.2 Temporal shadow edge localization

The temporal edges for the various objects can be seen in Figure 2

### 1.2 Intrinsic and extrinsic calibration

The intrinsic and the extrinsic calibration matrix values can be seen in the Figure 3 and Figure 4 respectively.

### 1.3 Reconstruction

The reconstructed outputs for some of the examples along with some representative images can be seen in Fig 5 and Fig 6 respectively. There are various steps taken to ensure a good reconstruction. These are as follows :

- Threshold the initial images so that all regions that are the ones belonging to the shadow regions can be identified easily.
- After getting the zero-crossing points for the vertical and horizontal lines, running a RANSAC in order to not create lines consisting of outlier zero-coordinates.

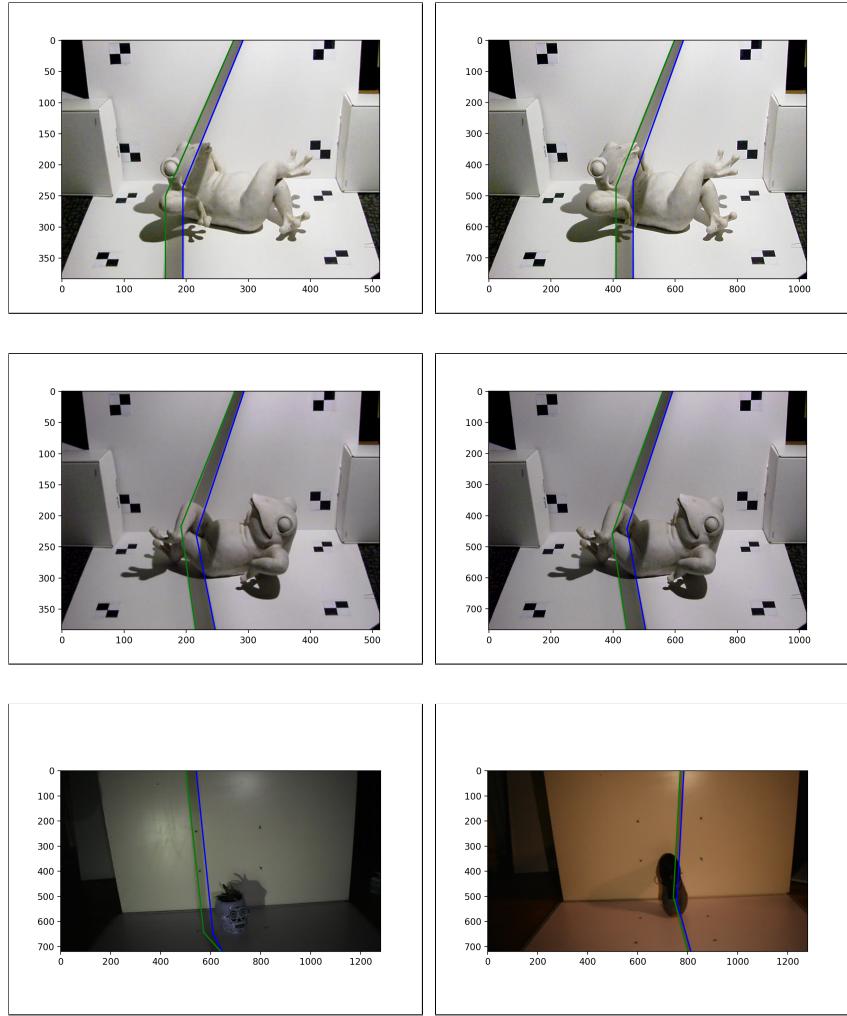


Figure 1: The spatial edges for some examples. The first column shows the low-resolution(left) and high resolution(right) outputs for the object frog in one orientation.The second column shows the low-resolution(left) and high resolution(right) outputs for the object frog in second orientation.The third column shows the outputs for the objects captured.

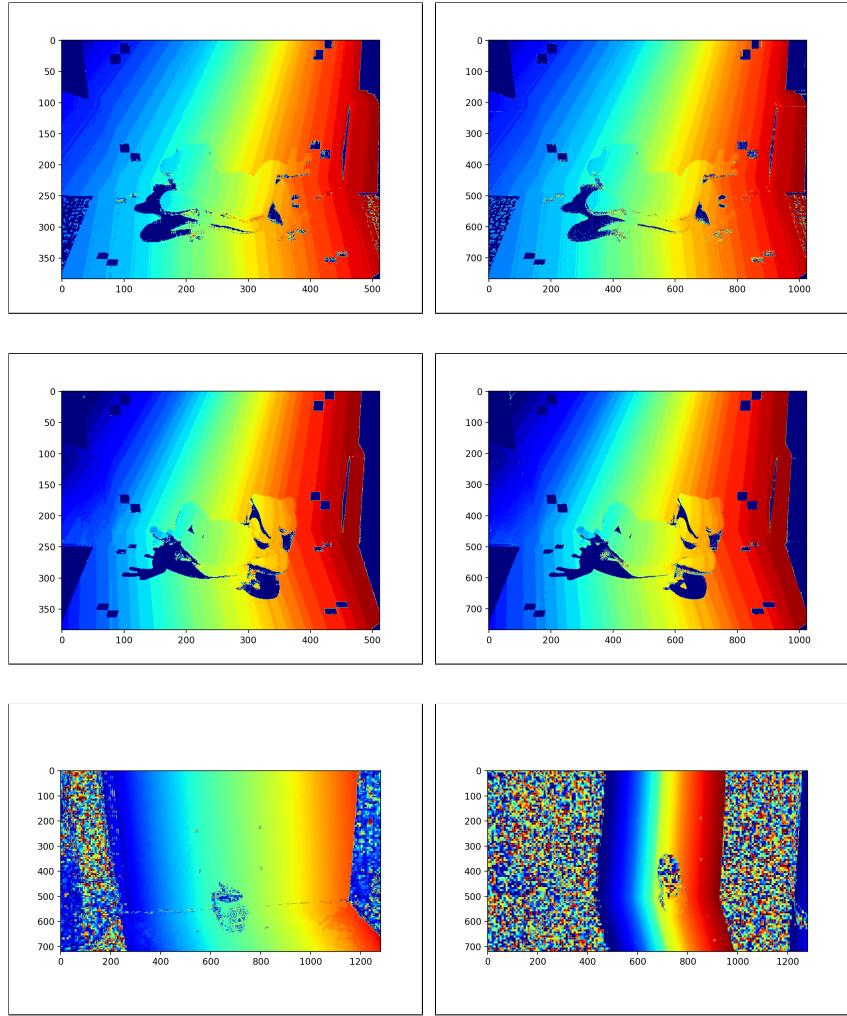


Figure 2: The timestamp edges for some examples. The first column shows the low-resolution(left) and high resolution(right) outputs for the object frog in one orientation. The second column shows the low-resolution(left) and high resolution(right) outputs for the object frog in second orientation. The third column shows the outputs for the objects captured.

```

>>> intrinsic['mtx']
array([[975.54361318, 0., 270.83006141],
       [0., 972.40460106, 142.73159318],
       [0., 0., 1.]]])
>>> intrinsic['dist']
array([-4.53908894e-01, 6.37141181e+00, -4.05808838e-03,
       2.42890564e-04, -6.79687332e+01]]))

>>> intrinsic['mtx']
array([[2.0625973e+03, 0.00000000e+00, 5.68921854e+02],
       [0.00000000e+00, 2.06726817e+03, 4.30545808e+02],
       [0.00000000e+00, 0.00000000e+00, 1.00000000e+02]])
>>> intrinsic['dist']
array([-1.64146999e-01, 1.64429888e-01, 6.58748153e-03,
       4.01261335e-03, 1.16492442e+01]]))
>>> = None

>>> intrinsic['mtx']
array([[2.0625973e+03, 0.00000000e+00, 5.68921854e+02],
       [0.00000000e+00, 2.06726817e+03, 4.30545808e+02],
       [0.00000000e+00, 0.00000000e+00, 1.00000000e+02]])
>>> intrinsic['dist']
array([-1.64146999e-01, 1.64429888e-01, 6.58748153e-03,
       4.01261335e-03, 1.16492442e+01]]))

>>> intrinsic['mtx']
array([[1.03614137e+03, 0.00000000e+00, 4.59745974e+02],
       [0.00000000e+00, 1.03574820e+03, 3.4001131e+02],
       [0.00000000e+00, 0.00000000e+00, 1.00000000e+02]])
>>> intrinsic['dist']
array([-0.05520554, -0.35619148, -0.00221603, 0.0049767, 0.91131365]]))

>>> intrinsic['mtx']
array([[1.03614137e+03, 0.00000000e+00, 4.59745974e+02],
       [0.00000000e+00, 1.03574820e+03, 3.4001131e+02],
       [0.00000000e+00, 0.00000000e+00, 1.00000000e+02]])
>>> intrinsic['dist']
array([-0.05520554, -0.35619148, -0.00221603, 0.0049767, 0.91131365]]))

```

Figure 3: The intrinsic parameters. The first column shows the low-resolution(left) and high resolution(right) outputs for the object frog in one orientation. The second column shows the low-resolution(left) and high resolution(right) outputs for the object frog in second orientation. The third column shows the outputs for the objects captured.

- Handling cases of slope tending to infinite especially, by taking the mean of all the values as constant and a placeholder for the slope.

## 2 Building your own 3D scanner

The intrinsic and the extrinsic parameters of the setup are reported in Figure 3 and 4 respectively. The temporal and the spatial edge localization are reported in Figure 2 and Fig 1 respectively. The reconstructed outputs along with some input examples can be seen in Figure ?? and Figure 7 respectively. The setup for the capture is also illustrated in Figure 8

<pre>&gt;&gt;&gt; extrinsic['rmat_h'] array([[ 0.99909414,  0.02294029, -0.03584205],        [-0.02427705, -0.38449064, -0.92280961],        [-0.03495045,  0.92284381, -0.38358542]])  &gt;&gt;&gt; extrinsic['rmat_v'] array([[ 0.99969846, -0.02243172, -0.00999079],        [-0.01684437, -0.92247355,  0.38569264],        [-0.01786799, -0.38540805, -0.92257323]])  &gt;&gt;&gt; extrinsic['tvec_h'] array([[-302.98374887],        [ 320.19899762],        [1438.83926076]])  &gt;&gt;&gt; extrinsic['tvec_v'] array([[-295.20715875],        [ 70.58619888],        [1766.67518611]])</pre>	<pre>&gt;&gt;&gt; extrinsic['rmat_h'] array([[ 0.99984358,  0.00559032, -0.01677971],        [-0.01235181, -0.45832817, -0.8886972 ],        [-0.01265872,  0.88876546, -0.45818743]])  &gt;&gt;&gt; extrinsic['rmat_v'] array([[ 0.9995154 , -0.02580206, -0.0174298],        [-0.01427985, -0.87711794,  0.48006273],        [-0.02765984, -0.47958145, -0.87706132]])  &gt;&gt;&gt; extrinsic['tvec_h'] array([[-321.20927561],        [ 207.28278493],        [1560.1499446481]])  &gt;&gt;&gt; extrinsic['tvec_v'] array([[-320.35743326],        [ -62.44552159],        [1877.40497507]])</pre>
<pre>array([[ 0.99800294,  0.02408244, -0.05839659],        [-0.04504899, -0.37669861, -0.92523983],        [-0.04427994,  0.92602278, -0.37486143]])  &gt;&gt;&gt; extrinsic['rmat_v'] array([[ 0.99969632, -0.02246521, -0.01012849],        [-0.01680909, -0.92219345,  0.38636343],        [-0.01802016, -0.38607585, -0.92229191]])  &gt;&gt;&gt; extrinsic['tvec_h'] array([[-305.2215721 ],        [ 327.82384479],        [1426.80163607]])  &gt;&gt;&gt; extrinsic['tvec_v'] array([[-295.23566878],        [ 70.47696487],        [1766.78642605]])</pre>	<pre>&gt;&gt;&gt; extrinsic['rmat_h'] array([[ 0.9998473 ,  0.00542847, -0.01661036],        [-0.01227542, -0.45833068, -0.88869697],        [-0.01243731,  0.88876516, -0.45819406]])  &gt;&gt;&gt; extrinsic['rmat_v'] array([[ 0.99952989, -0.02550737, -0.0170101],        [-0.0142401 , -0.87759128,  0.47919804],        [-0.02715158, -0.47873083, -0.87754194]])  &gt;&gt;&gt; extrinsic['tvec_h'] array([[-321.12716035],        [ 207.28762487],        [1559.92971501]])  &gt;&gt;&gt; extrinsic['tvec_v'] array([[-320.34888508],        [ -62.48638853],        [1877.51260721]])</pre>
<pre>&gt;&gt;&gt; extrinsic['rmat_h'] array([[ 0.99792442,  0.03450801, -0.05436952],        [-0.04518672, -0.22629853, -0.97300932],        [-0.04588836,  0.97344654, -0.22426953]])  &gt;&gt;&gt; extrinsic['rmat_v'] array([[ 0.9943844 , -0.05145142, -0.09247929],        [-0.0173439 , -0.94127543,  0.33719395],        [-0.10439759, -0.33369645, -0.93688197]])  &gt;&gt;&gt; extrinsic['tvec_h'] array([[-75.15561915],        [ 329.34440261],        [1003.98774225]])  &gt;&gt;&gt; extrinsic['tvec_v'] array([[-76.12723955],        [ 24.00719107],        [1328.37961158]])</pre>	<pre>&gt;&gt;&gt; extrinsic['rmat_h'] array([[ 0.99792442,  0.03450801, -0.05436952],        [-0.04518672, -0.22629853, -0.97300932],        [-0.04588836,  0.97344654, -0.22426953]])  &gt;&gt;&gt; extrinsic['rmat_v'] array([[ 0.9943844 , -0.05145142, -0.09247929],        [-0.0173439 , -0.94127543,  0.33719395],        [-0.10439759, -0.33369645, -0.93688197]])  &gt;&gt;&gt; extrinsic['tvec_h'] array([[-75.15561915],        [ 329.34440261],        [1003.98774225]])  &gt;&gt;&gt; extrinsic['tvec_v'] array([[-76.12723955],        [ 24.00719107],        [1328.37961158]])</pre>

Figure 4: The timestamp edges for some examples. The first column shows the low-resolution(left) and high resolution(right) outputs for the object frog in one orientation. The second column shows the low-resolution(left) and high resolution(right) outputs for the object frog in second orientation. The third column shows the outputs for the objects captured.

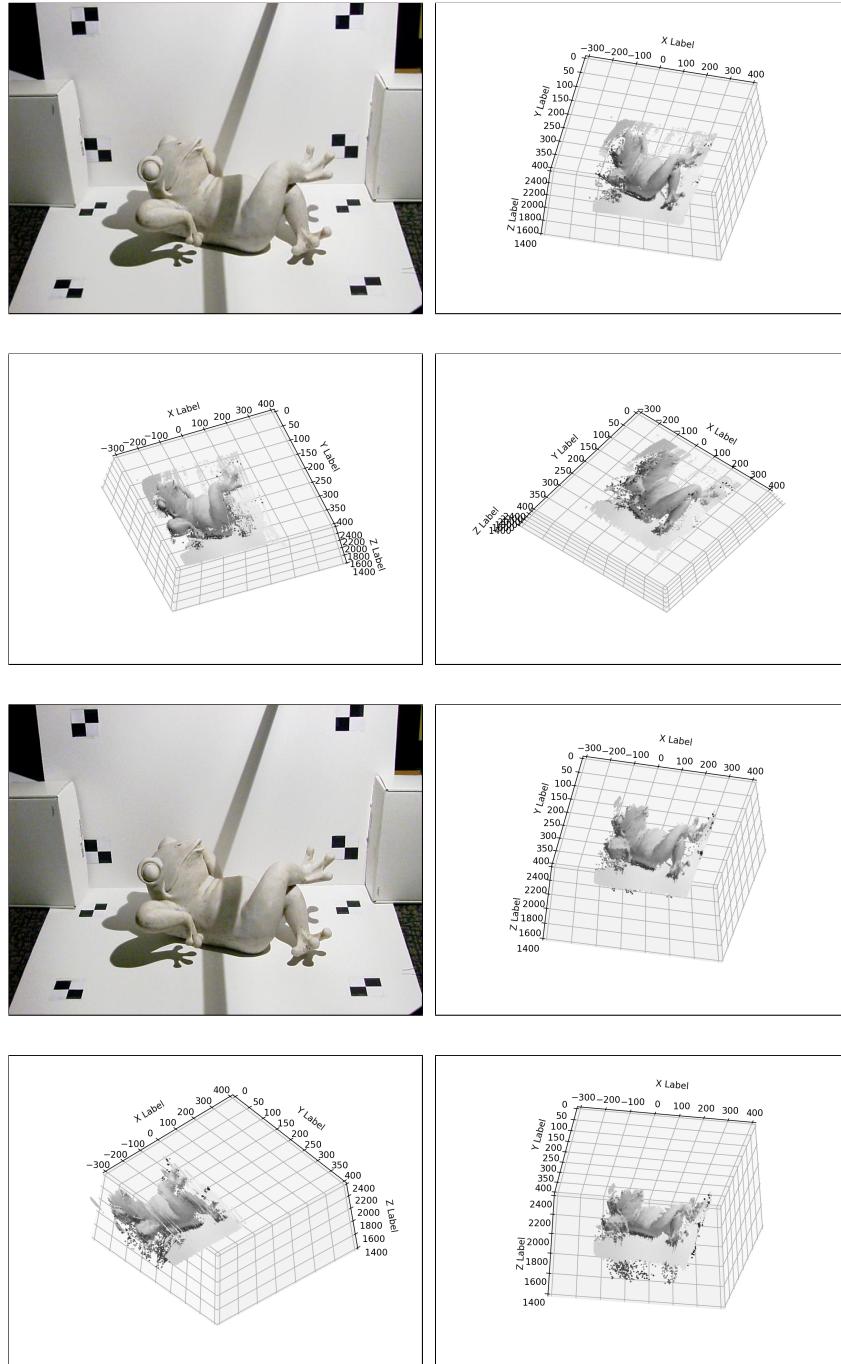


Figure 5: The reconstructed outputs for low-resolution (top) high-resolution (bottom) images for the frog in one orientation.

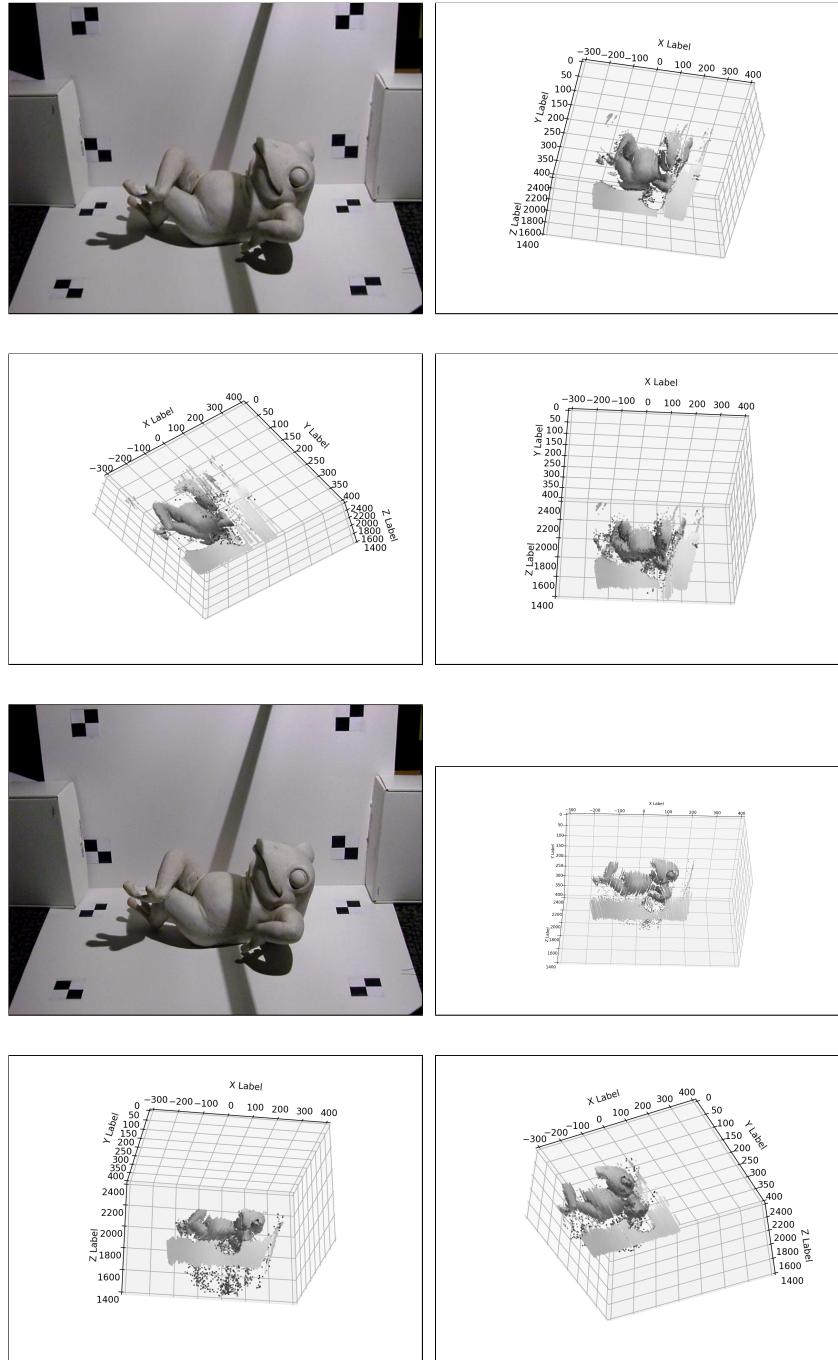


Figure 6: The reconstructed outputs for low-resolution (top) high-resolution (bottom) images for the frog in second orientation.

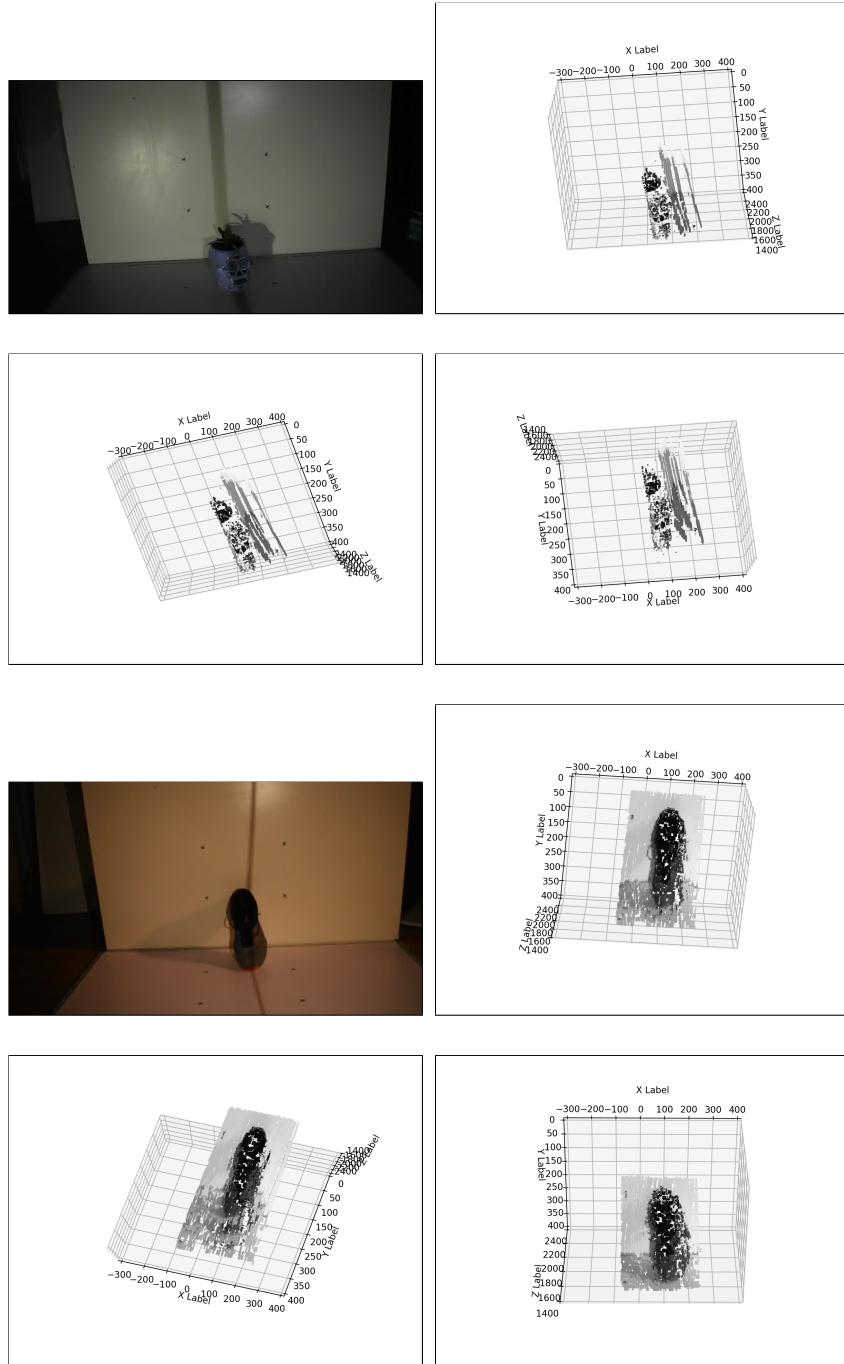


Figure 7: The reconstructed outputs for two different objects captured along with the representative images.



Figure 8: An illustration of the capture setup. The points marked for the extrinsic calibration, object and the camera can be seen.