

STEREO VISION

CODE:

```
In [1]: import cv2 as cv
import matplotlib.pyplot as plt
import numpy as np
```

```
In [2]: ply_header = '''ply
format ascii 1.0
element vertex %(vert_num)d
property float x
property float y
property float z
property uchar red
property uchar green
property uchar blue
end_header
'''

def write_ply(fn, verts, colors):
    verts = verts.reshape(-1, 3)
    colors = colors.reshape(-1, 3)
    verts = np.hstack([verts, colors])
    with open(fn, 'w') as f:
        f.write(ply_header % dict(vert_num=len(verts)))
    np.savetxt(f, verts, '%f %f %f %d %d %d')
```

```
In [61]: img1 = input("Enter the Left Image: ")
img2 = input("Enter the Right Image: ")
img_l = cv.imread(img1)
img_r = cv.imread(img2)

img_l_gray = cv.cvtColor(img_l, cv.COLOR_BGR2GRAY)
img_r_gray = cv.cvtColor(img_r, cv.COLOR_BGR2GRAY)

block_size = 5
min_disp = 16
max_disp = 112

num_disp = max_disp - min_disp

uniquenessRatio = 10

speckleWindowSize = 100

speckleRange = 2
disp12MaxDiff = 0

stereo = cv.StereoSGBM_create(
    minDisparity=min_disp,
    numDisparities=num_disp,
    blockSize=block_size,
    uniquenessRatio=uniquenessRatio,
    speckleWindowSize=speckleWindowSize,
    speckleRange=speckleRange,
    disp12MaxDiff=disp12MaxDiff,
    P1=8 * 1 * block_size * block_size,
    P2=32 * 1 * block_size * block_size,)
disparity_SGBM = stereo.compute(img_l, img_r)
```

```

disparity_SGBM = cv.normalize(disparity_SGBM, disparity_SGBM, alpha=255, beta=0, norm_type=cv.NORM_MINMAX)
disparity_SGBM = np.uint8(disparity_SGBM)

h, w = img_l.shape[:2]
f = 0.8*w
Q = np.float32([[1, 0, 0, -0.5 * w],
                [0, -1, 0, 0.5 * h],
                [0, 0, 0, -f],
                [0, 0, 1, 0]])
points = cv.reprojectImageTo3D(disparity_SGBM, Q)
colors = cv.cvtColor(img_l, cv.COLOR_BGR2RGB)
mask = disparity_SGBM > disparity_SGBM.min()
out_points = points[mask]
out_colors = colors[mask]
out_fn = 'out.ply'
# write_ply('out1ply', out_points, out_colors)
# print ('%s saved' % 'out1.l')
write_ply('out3.ply', out_points, out_colors)
print ('%s saved' % 'out3.ply')

cv.imshow("Left Image", img_l)
cv.imshow("Right Image", img_r)
cv.imshow("Disparity", disparity_SGBM)
#plt.imshow(disparity_SGBM, 'gray')
#plt.axis('off')
#plt.show()

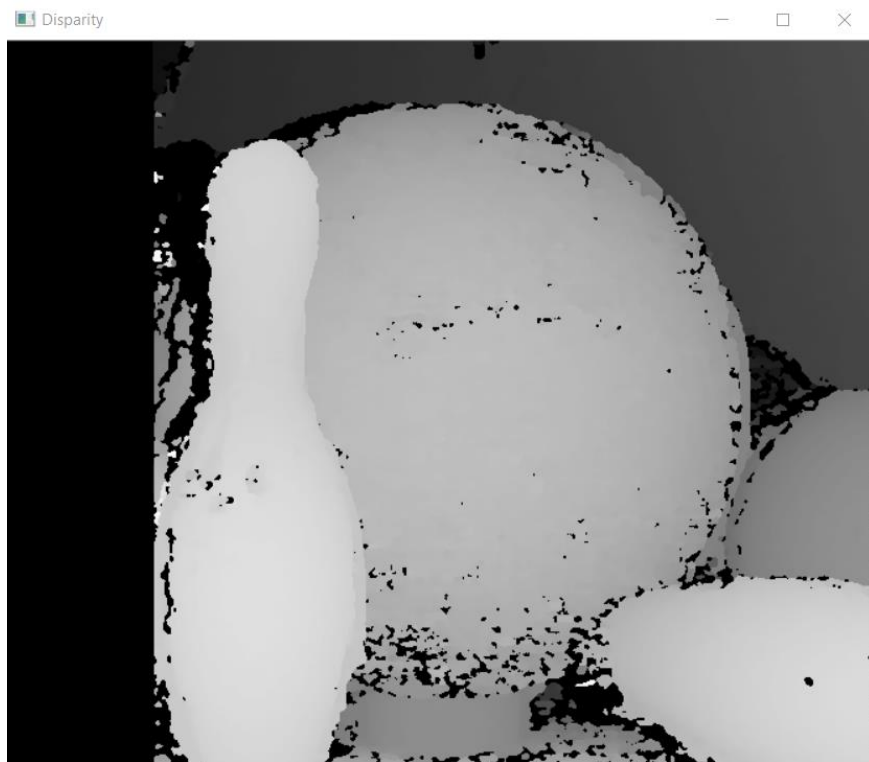
cv.waitKey()
cv.destroyAllWindows()

```

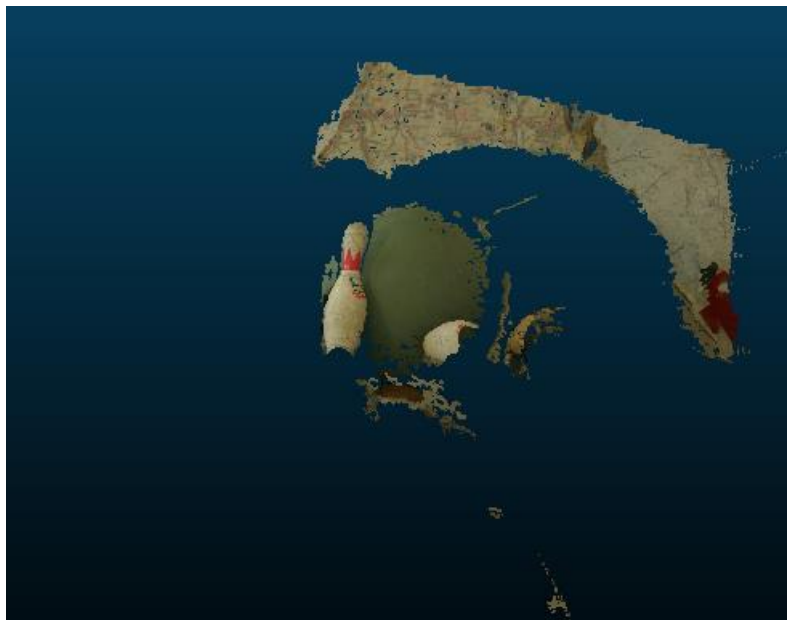
Enter the Left Image: 2.png
Enter the Right Image: 1.png
out3.ply saved

OUTPUT:

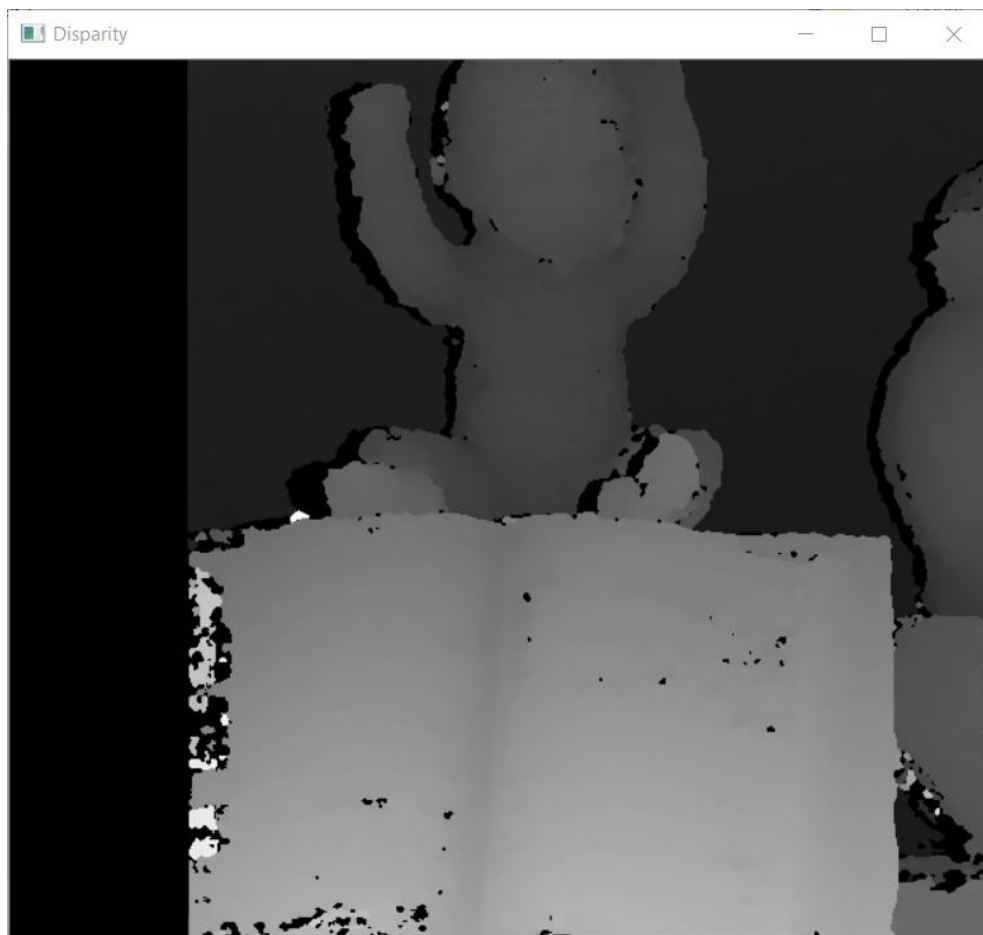
BALL:



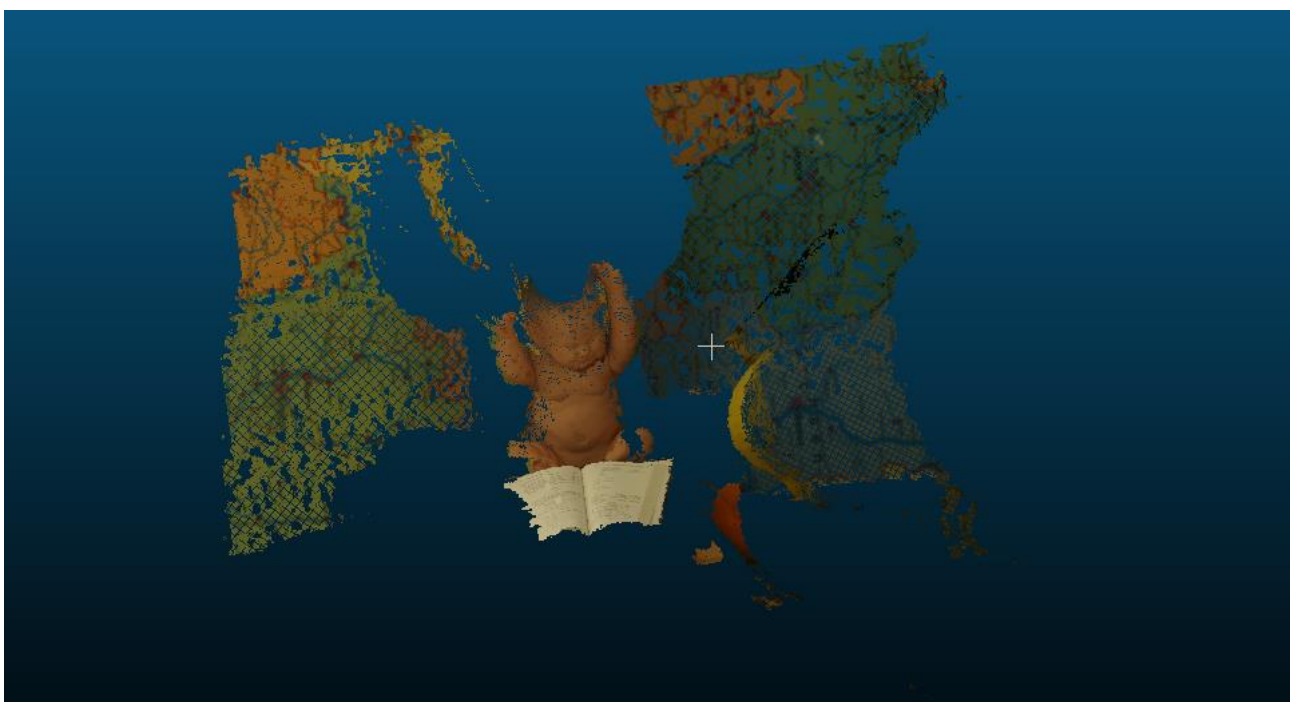
PLY FILE:



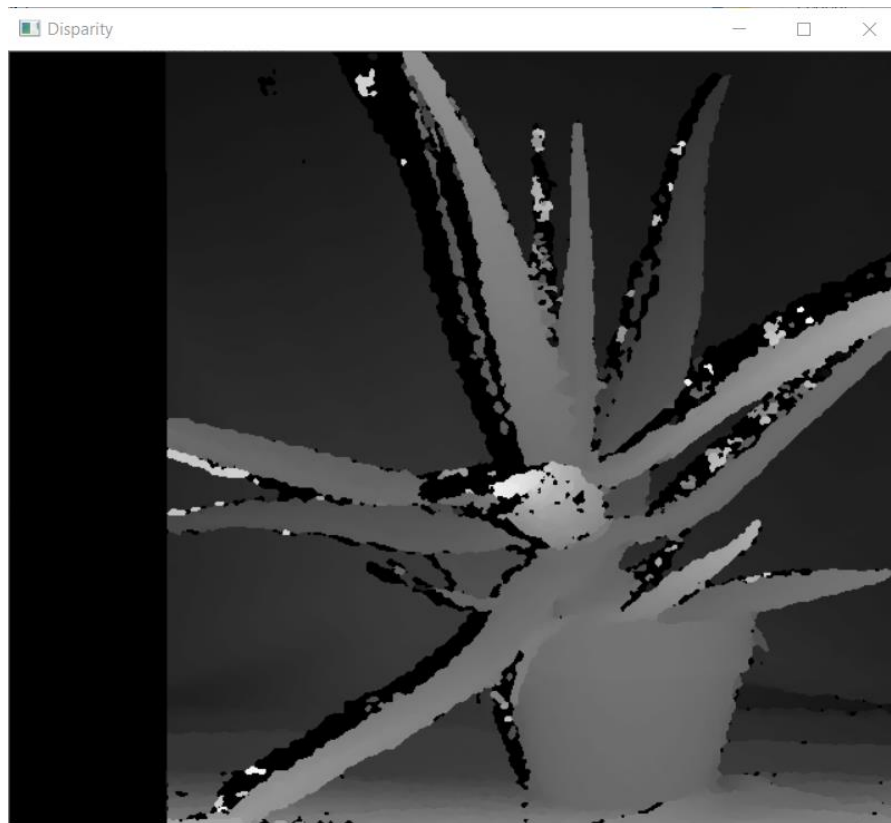
BABY:



PLY FILE:



PLANT:



PLY FILE:

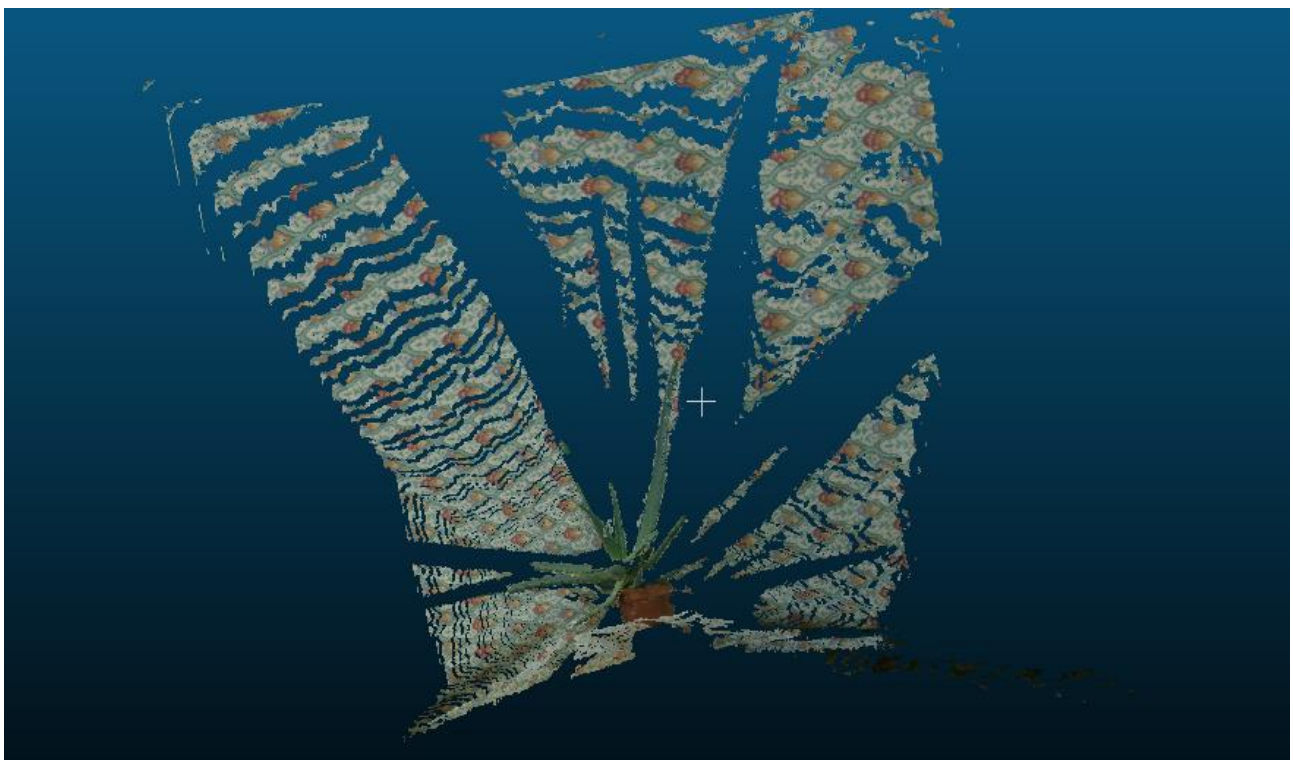
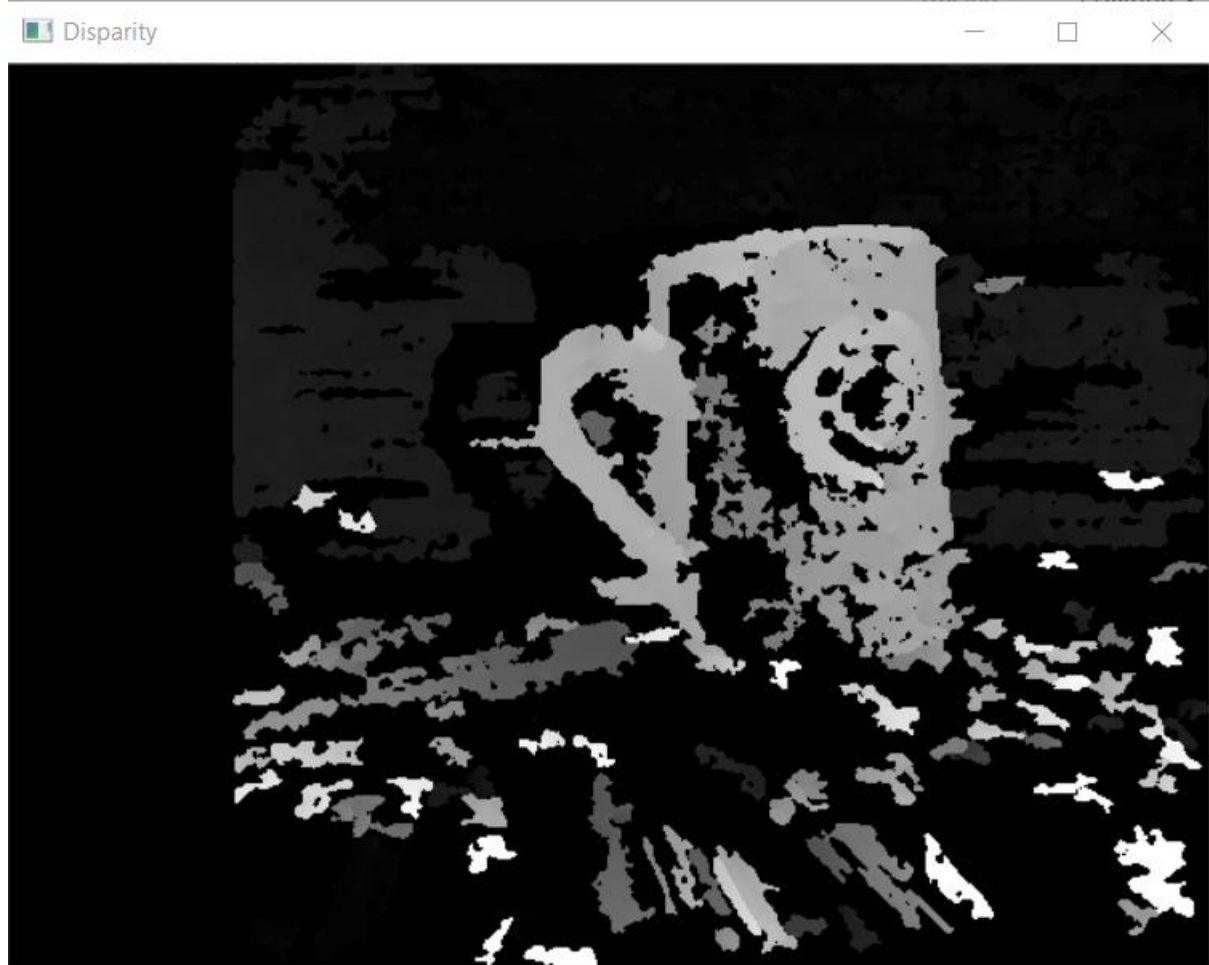


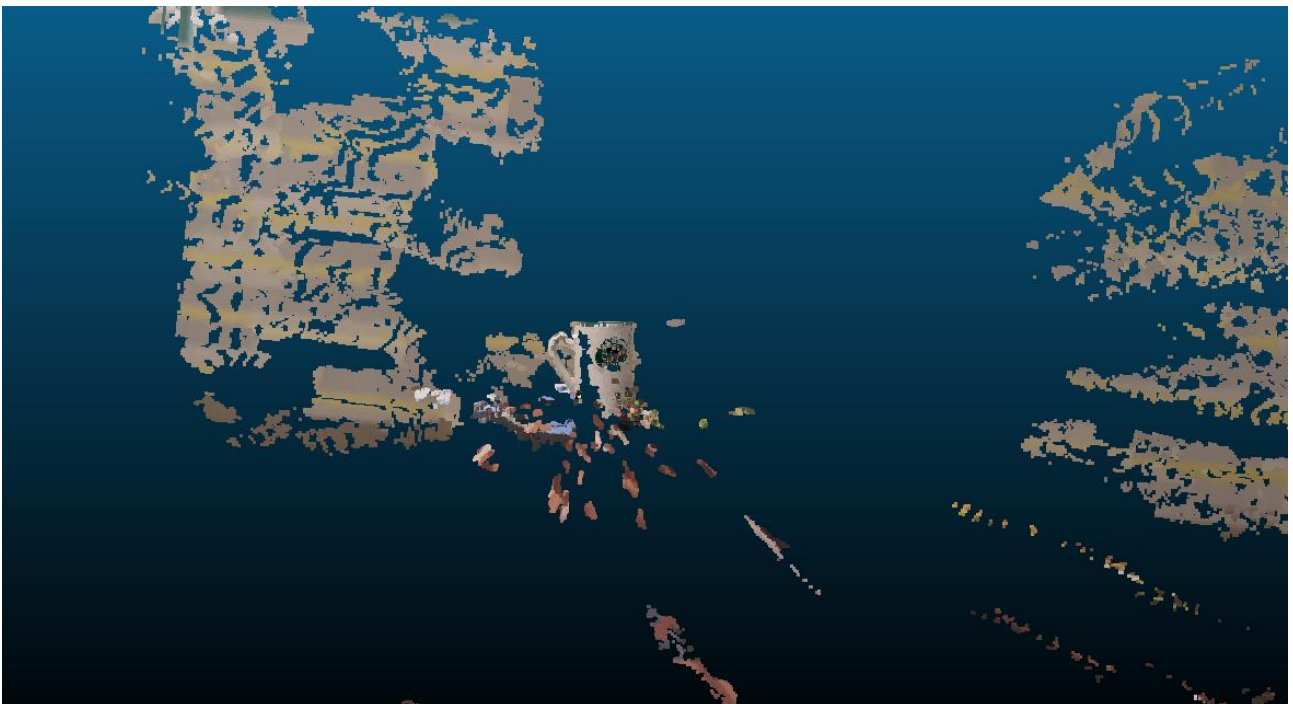
IMAGE FROM MOBILE PHONE:



DISPARITY MAP:



PLY FILE:



OPERATING SYSTEM: Windows 10

IDE Used: Jupyter Notebook

Number of Hours Spent: 11 Hours
