Stereo Depth Estimation (SDE)

Aakar Sharma

Indian Instituite of Technology Jammu

2016ucs0066@iitjammu.ac.in

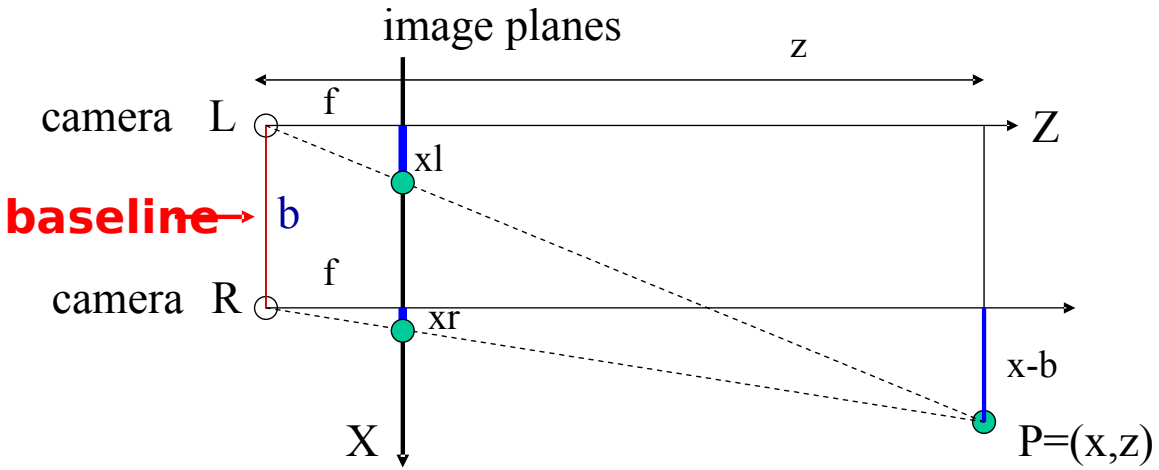
October 1, 2018

Introduction

Stereo Depth Estimation is an techinique by which we determine the depth of objects in a stereo pair images. Stereo pair images is defined as images of an object from two different angles. The Depth of objects will be calculated just lke trhe way our eyes estimate the depth of the object they see. The two eyes forms two streo pair which helps our brain to calculate the depth of the objects.

Projection for Stereo Images

Simple Model: Optic axes of 2 cameras are parallel



$$\frac{z}{f} = \frac{x}{xl} \qquad \frac{z}{f} = \frac{x-b}{xr} \qquad \frac{z}{f} = \frac{y}{yl} = \frac{y}{yr}$$

Y-axis is perpendicular to the page.

(from similar triangles)

Algorithm of SDE

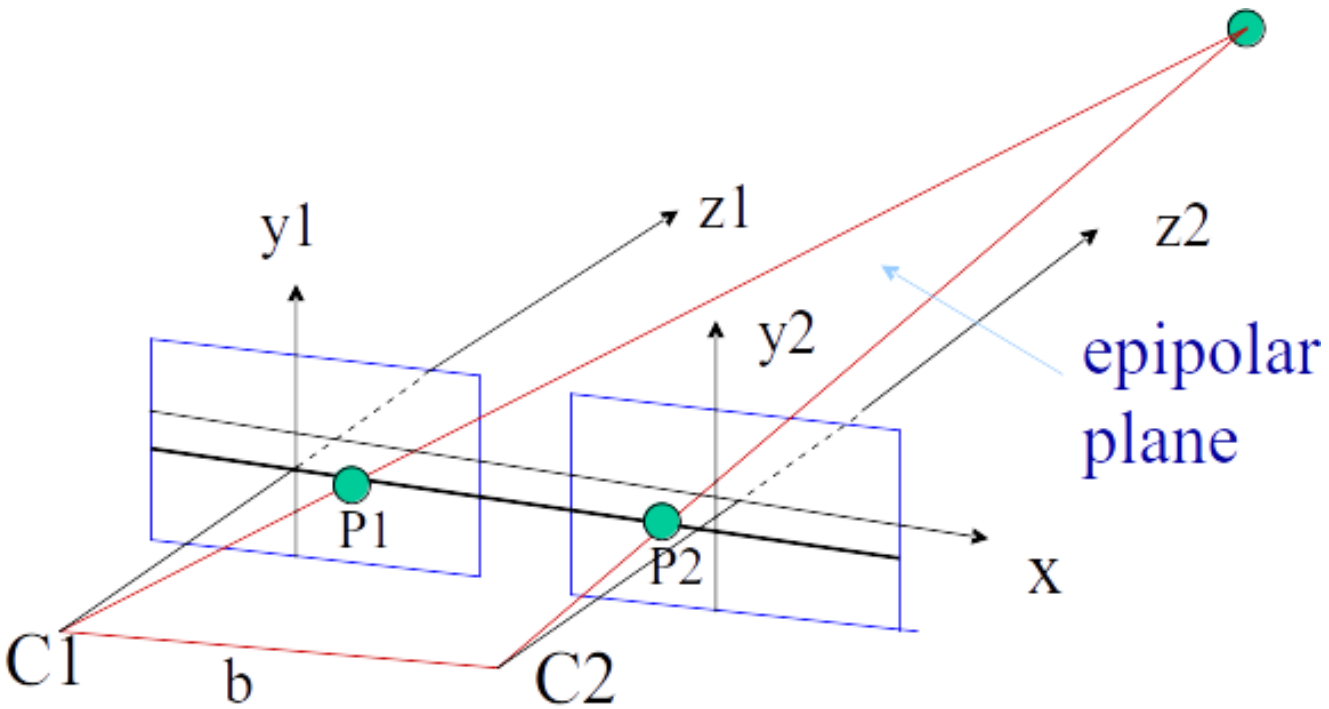
By considering the two similar triangles, we can deduce the result of Disparity in the stereo images. This result is for one identical pixel of both the images. To find this identical pixel we use three different methods.

1. Cross correlation or SSD using small windows(Batch).
2. Symbolic feature matching, usually using segments/corners.
3. Use the newer interest operators, e.g., SIFT.

Algorithm of SDE

Epipolar Constraint for Correspondence

A plane is drawn joining the points **P** and camera **L** and **R** is called as Epipolar plane. This Epipolar plane cuts through image planes forming an epipolar line in each plane. We use this line to search for similar pixels in the two images rather than searching all the images.



Implementation

The Implementation of this concept is done in python. The database used is from this link. A straight forward implemenntation is done, after which results are saved. The block size used is 8. Some filtering is also done to make the image clear.

Program

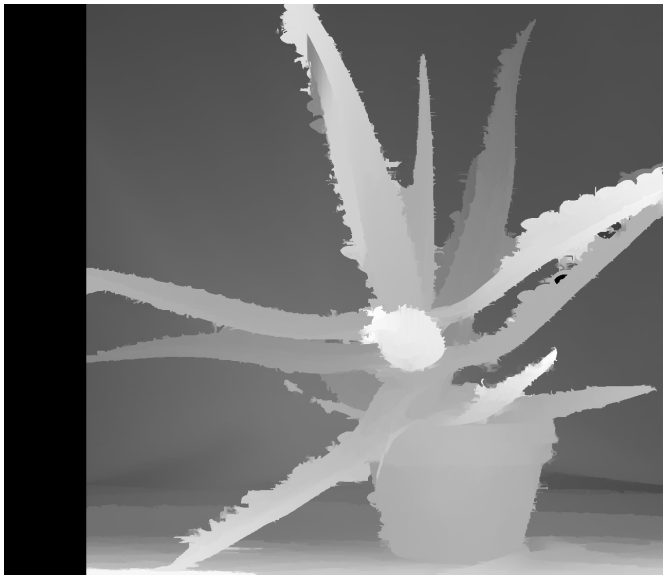
```
import numpy as np
import cv2
import os

arr = os.listdir('./dataset')
window_size = 7                    # wsize default 3; 5; 7 for SGBM reduced size image; 15 for SGBM full size image (1300px and above); 5 Works nicely

left_matcher = cv2.StereoSGBM_create(
    minDisparity=0,
    numDisparities=160,             # max_disp has to be dividable by 16 f. E. HH 192, 256
    blockSize=8,
    P1=8 * 3 * window_size ** 2,    # wsize default 3; 5; 7 for SGBM reduced size image; 15 for SGBM full size image (1300px and above); 5 Works nicely
    P2=32 * 3 * window_size ** 2,
    disp12MaxDiff=1,
    uniquenessRatio=15,
    speckleWindowSize=0,
    speckleRange=2,
    preFilterCap=63,
    mode=cv2.STEREO_SGBM_MODE_SGBM_3WAY
)
right_matcher = cv2.ximgproc.createRightMatcher(left_matcher)

# FILTER Parameters
lmbda = 80000
sigma = 1.2
visual_multiplier = 1.0
wls_filter = cv2.ximgproc.createDisparityWLSFilter(matcher_left=left_matcher)
wls_filter.setLambda(lmbda)
wls_filter.setSigmaColor(sigma)

for i in arr:   # Loop through every element in the database
    imgL = cv2.imread('dataset/'+i+'/view1.png',-1)
    imgR = cv2.imread('dataset/'+i+'/view5.png',-1)

    displ = left_matcher.compute(imgL, imgR)  # .astype(np.float32)/16
    dispr = right_matcher.compute(imgR, imgL)  # .astype(np.float32)/16
    displ = np.int16(displ)
    dispr = np.int16(dispr)
    filteredImg = wls_filter.filter(displ, imgL, None, dispr)  # important to put "imgL" here!!!
    filteredImg = cv2.normalize(src=filteredImg, dst=filteredImg, beta=0, alpha=255, norm_type=cv2.NORM_MINMAX);
    filteredImg = np.uint8(filteredImg)
    cv2.imwrite("./Result/"+i+".png", filteredImg)
```

Implementation

The Implementation of this concept is done in python. The database used is from this link. A straight forward implemenntation is done, after which results are saved. The block size used is 8. Some filtering is also done to make the image clear.

Results

Aloe

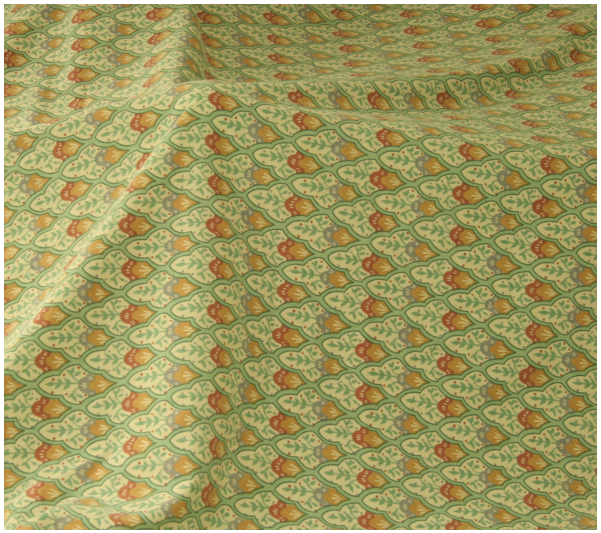


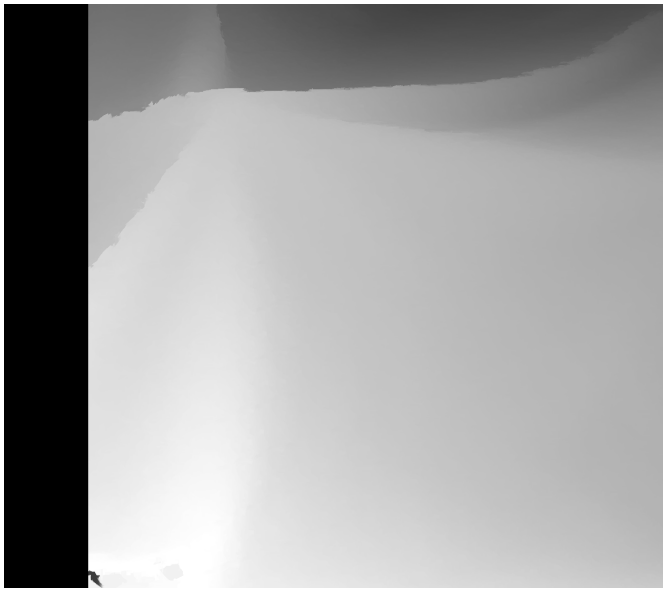
Results

Baby1

Results

Cloth1



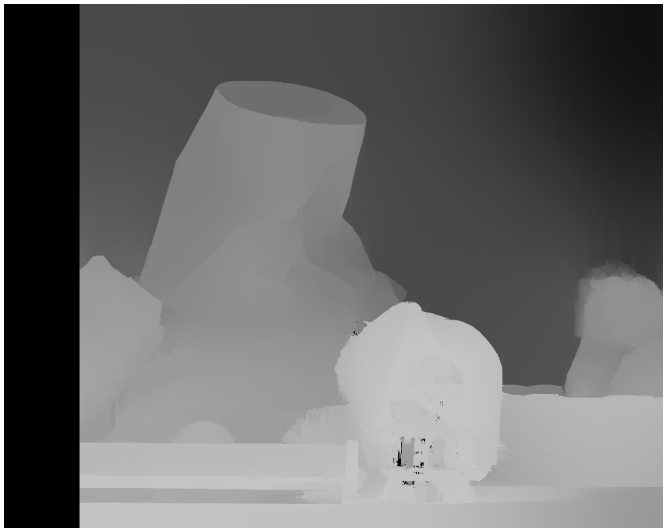


Results

Lampshade1

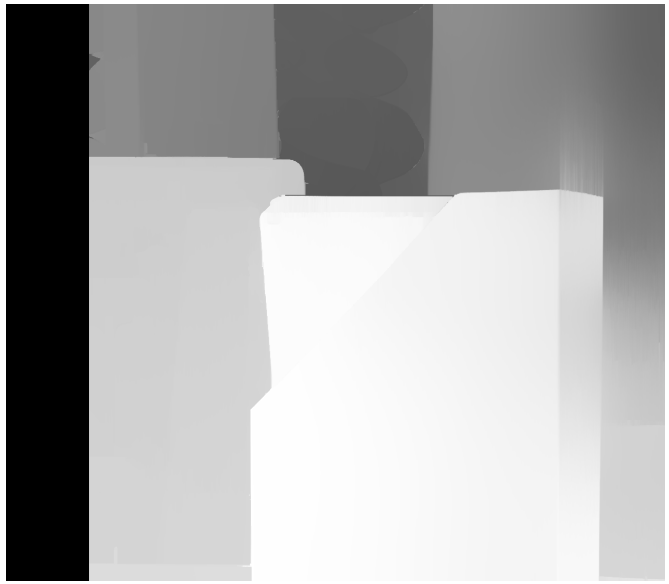
Results

Midd1



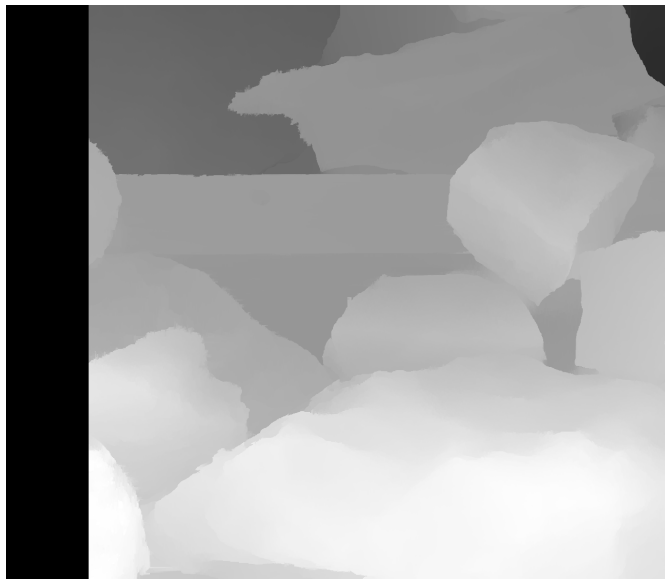
Results

Plastic



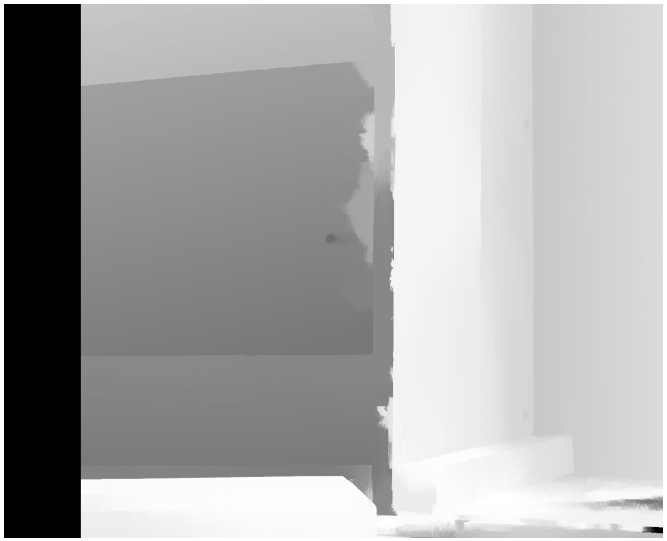
Results

Rocks1



Results

Wood1



Things Learned

From this assignment I have learned the theory behind Stereo Depth Estimation. A better and a clear understanding is achieved by implementing the same in python(using open-cv).

References

- https://courses.cs.washington.edu/courses/cse455/09wi/Lects/lect16.pdf
- http://www.cs.tut.fi/ suominen/SGN-1656-stereo/stereo_instructions.pdf
- http://vision.middlebury.edu/stereo/data/
- http://timosam.com/python_opencv_depthimage