Group 3
Stereo Vision and Depth Analysis

K Sai Niranjan Reddy • Radhika Deshpande • Sanjan Prakash
ES14BTECH11011 • ES14BTECH11006 • EP14BTECH11007

Project Outline

Motivation

- Set Up - first with 2 cameras and then with 1
- Calibration and Rectification
- Disparity by SGBM
- 3D reconstruction
- Object detection by Graph Cut (mask + rect)
- Depth measurement
- Errors and future work

Motivation

Stereo vision is modeled based on our eyes. Potential applications of stereo Vision are -

- SLAM
- Automated Vehicles
- Gaming Industry
- Virtual Reality
- Artificial vision for the blind

Setup

Our main aim was to build our very own pair of stereo cameras.

Initially, we began with two camera side by side with 6.5cm distance between the lenses.

However, we got better disparity results with one camera (shifting by 5 cm for the stereoscopic complement image)

Calibration and Rectification

$$K = \begin{pmatrix} f_x & s & x_0 \\ 0 & f_y & y_0 \\ 0 & 0 & 1 \end{pmatrix}$$

- Used Stereo calibrated app to get the intrinsic parameters and extrinsic parameters of the camera.
- Tried rectification both through calibrated results and through uncalibrated.
- Rectification procedure.
- Calibrated and uncalibrated way
- However for both the ways undistortion using radial and tangential coefficients was done.

Rectification

- Used intrinsic matrices and rotation and transformation matrix to get rotation matrix for first and second camera.
- We then use the individual rotation matrices to rotate the images so that their epipolar lines become parallel.
- In the uncalibrated way, we used SIFT to get feature points and matched them using Flann Matcher.
- Once we get at least 8 point correspondence from the two images. We can use 8 point algorithm to get fundamental matrix. X’FX =0
- Then we used cv2.stereoRectifyUncalibrated to get rectification matrices of both left image and right image.

Disparity

- Refers to the difference in location of an object in corresponding two (left and right) images as seen by the left and right eye which is created due to parallax (eyes' horizontal separation)

- Sum of Absolute Differences (SAD) for a pixel over a window/block W is given by :

$$\sum_{(i,j)\in W} |I_1(i,j) - I_2(x+i,y+j)|$$

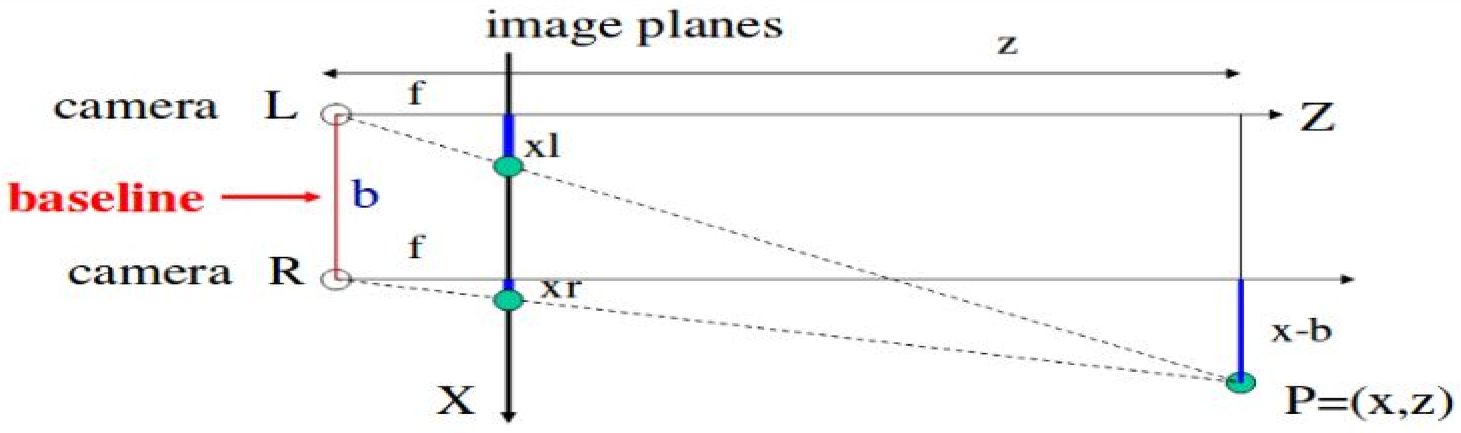
- For each shift value of the right image, there is an SAD map equal to the size of the image. It leads to a 3D space, which contains "*disp12MaxDiff*" number of SAD maps

- The disparity map is a 2D map reduced from the 3D space above. The disparity of a pixel is equal to the shift value that leads to minimum SAD for that pixel

Disparity (with cv2.createStereoSGBM())

- SADWindowSize
- numberOfDisparities
- minDisparity
- uniquenessRatio
- speckleWindowSize
- speckleRange
- disp12MaxDiff
- P1
- P2

Depth evaluation from disparity maps

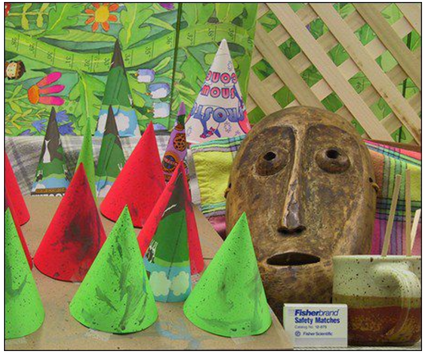


$$\boxed{\frac{z}{f} = \frac{x}{xl}} \qquad \boxed{\frac{z}{f} = \frac{x-b}{xr}}$$

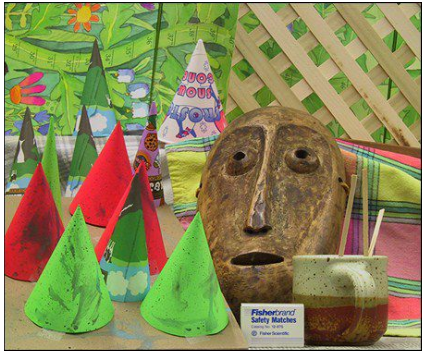
(from similar triangles)

xl - xr = disparity
Therefore, depth = z = (f*b)/disparity

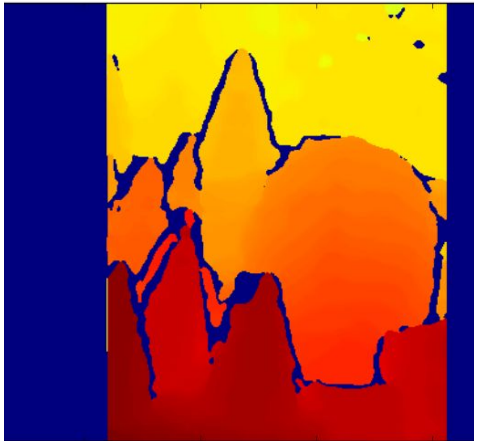
3D Reconstruction



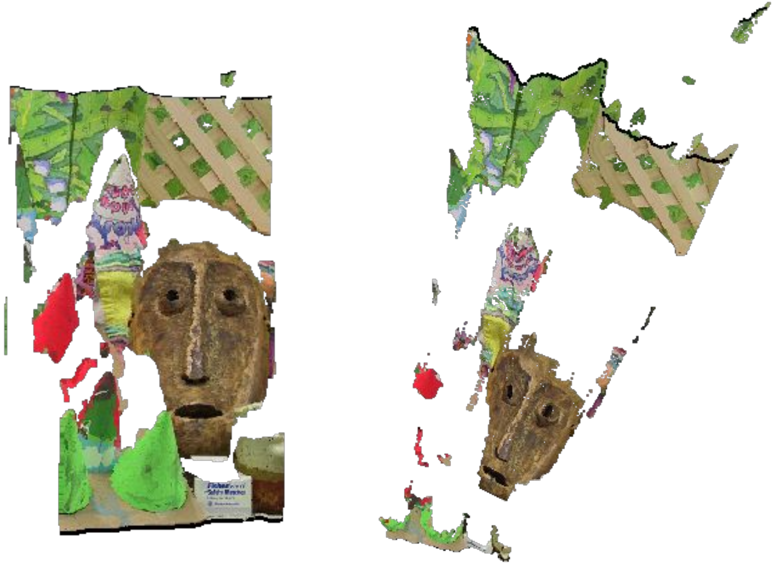
Left



Right

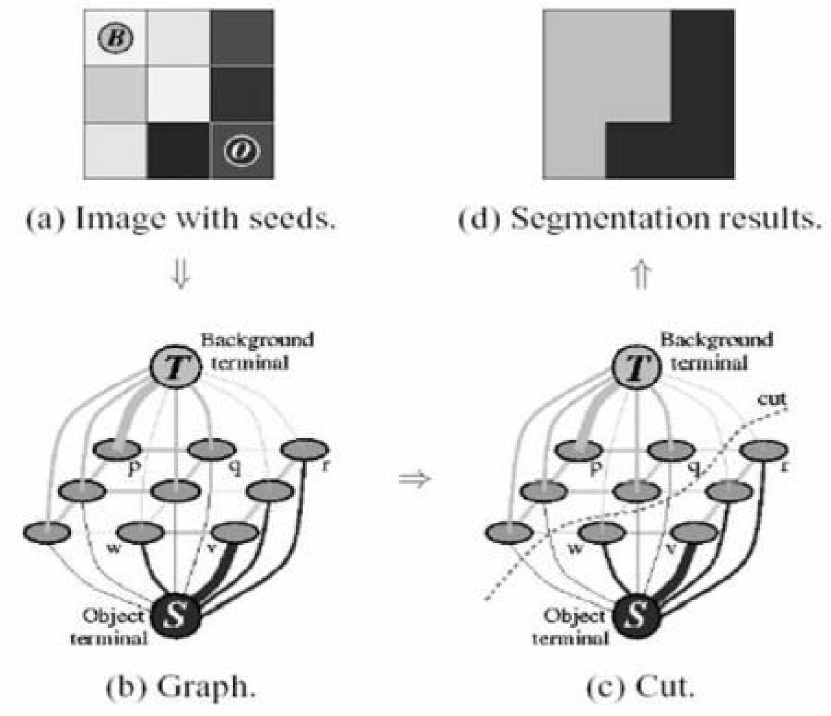


Disparity



3D point cloud
from different
angles

Object Detection - Graph Cut



(a) Image with seeds.

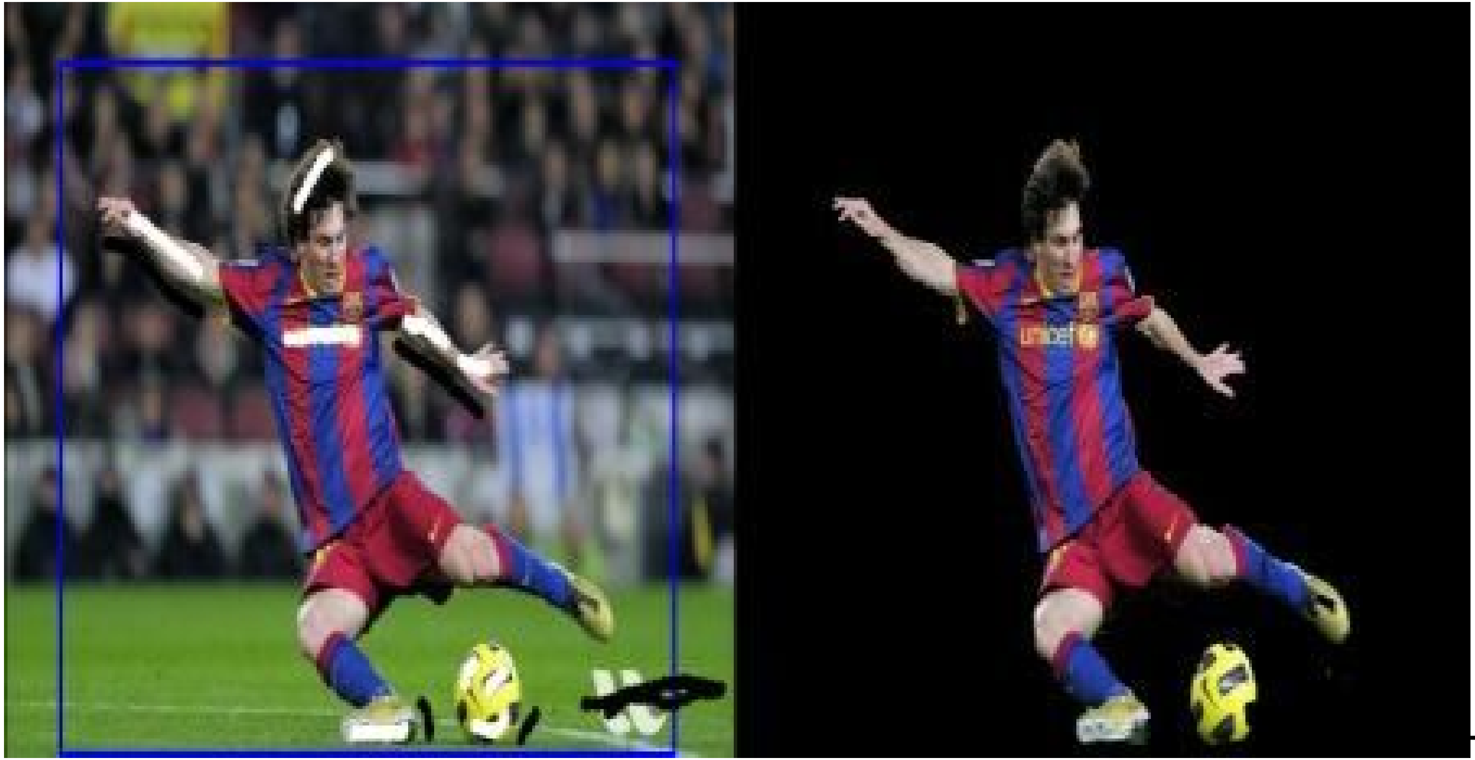
(d) Segmentation results.

(b) Graph.

(c) Cut.

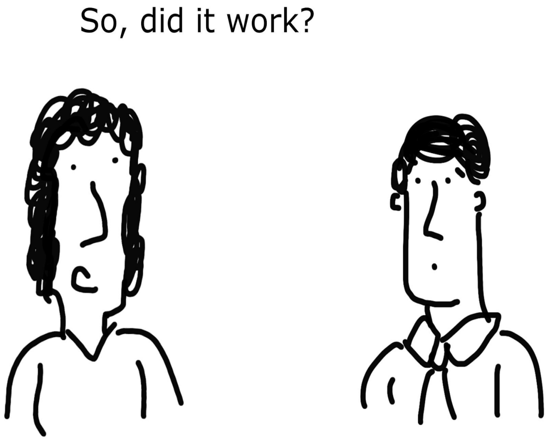
(Image Courtesy: http://www.cs.ru.ac.za/research/g02m1682/)

Opencv. Grabcut()

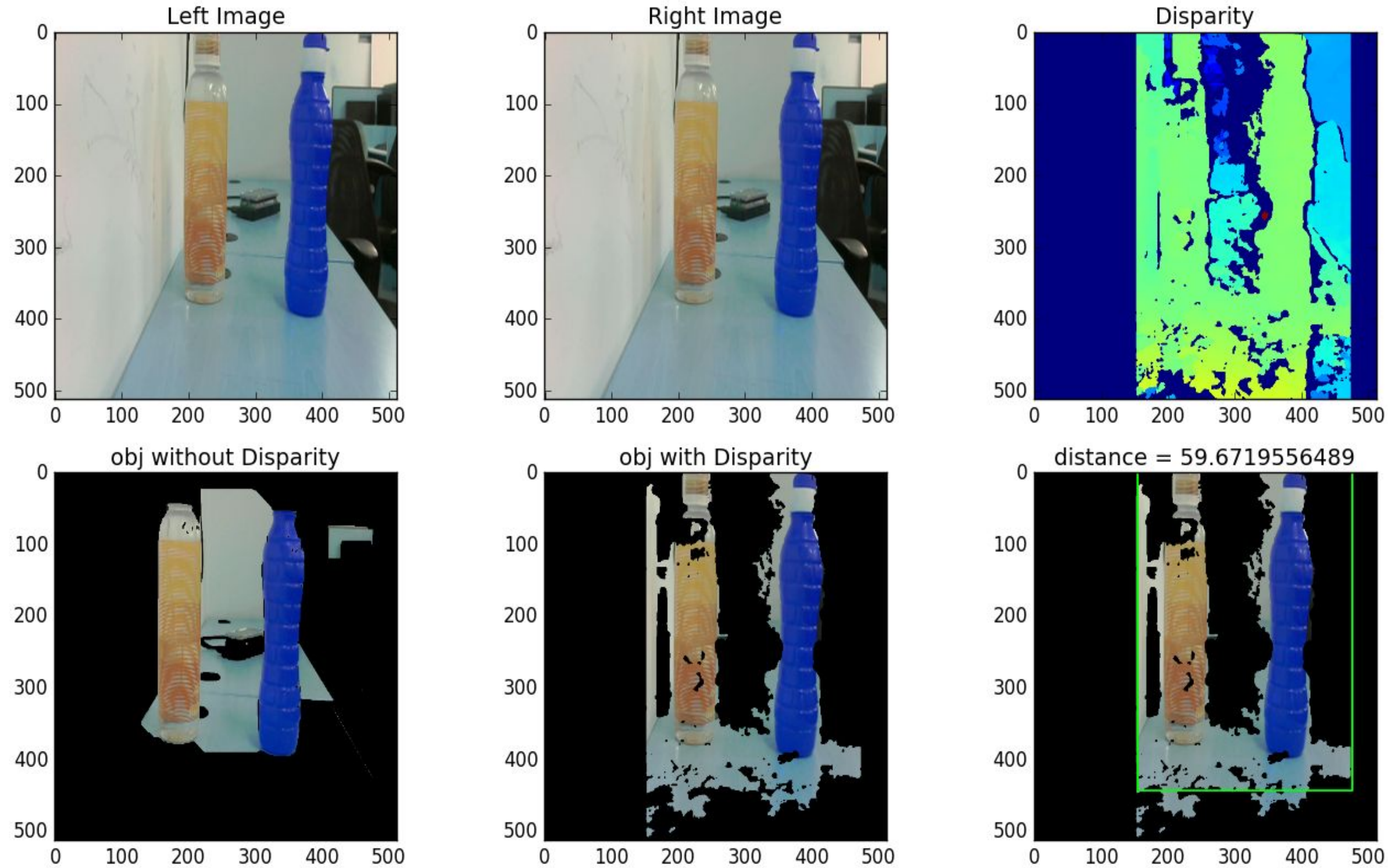


How can we use stereo vision for better object detection?

- Disparity!
- Lower disparity - background
- Higher disparity - foreground
- **Some results….**



Demo



Left Image

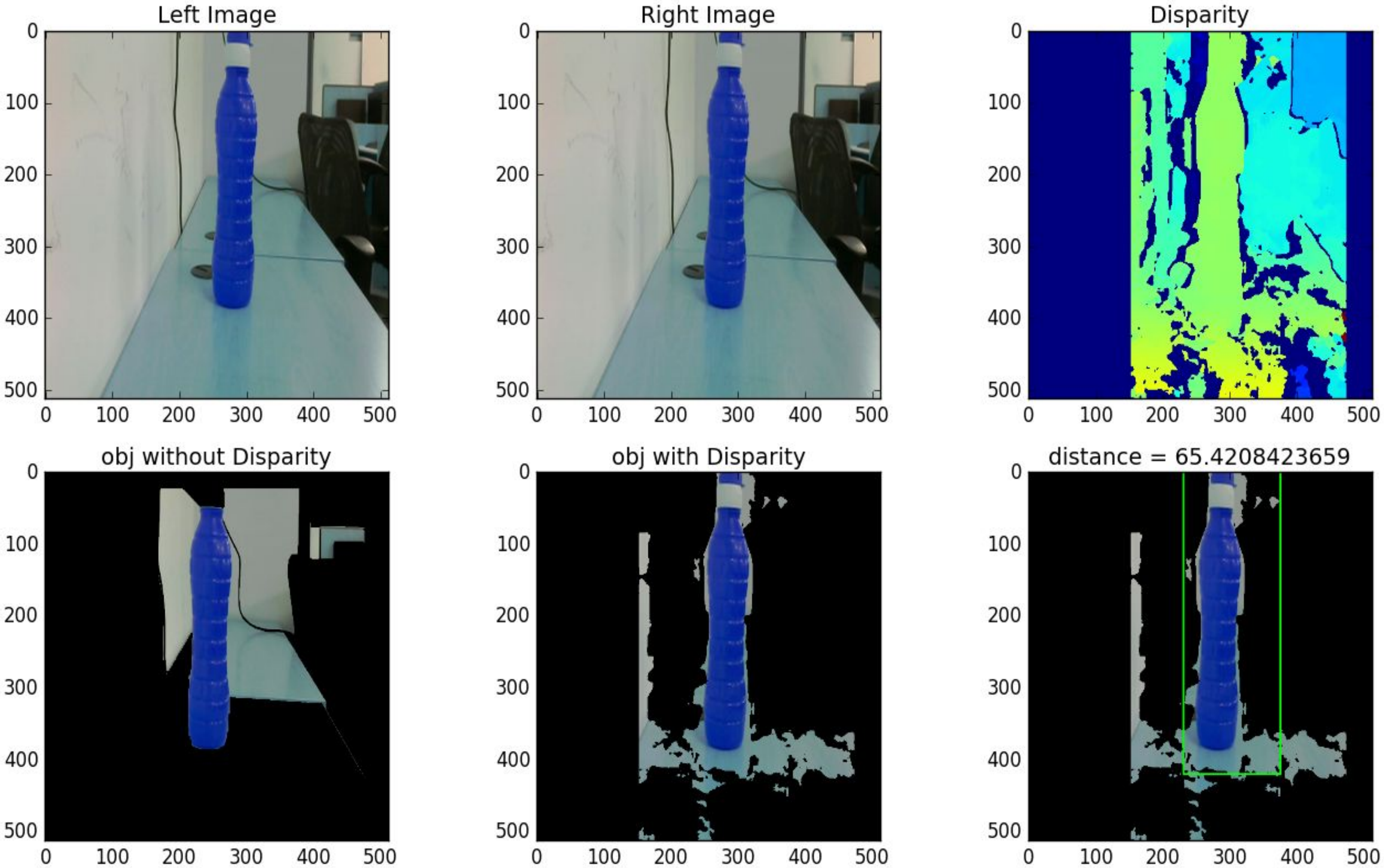
Right Image

Disparity

obj without Disparity

obj with Disparity

distance = 59.6719556489


Left Image
Right Image
Disparity
obj without Disparity
obj with Disparity
distance = 65.4208423659
0
100
200
300
400
500