

EE 702 Project 2 : Stereo Vision

Bino and Multi-ocular Stereo Vision

Group members:

- Meet Shah (13D070003)
- Yash Bhalgat (13D070014)

Abstract

This code repository aims at implementing stereo vision using patch-based similarity measures to estimate the disparity and hence generate the depth maps. The patch similarity measures used include:

- Sum of absolute differences SAD
- Sum of squares of differences SSD
- Normalized Cross Correlation NCC

Multi-ocular stereo is also implemented using 4 images of the same object with linearly increasing distances between the 1st image.

Code files:

- stereo_SSD.m to implement binocular stereo using SSD.
- stereo_NCC.m to implement binocular stereo using NCC.
- stereo_SAD.m to implement binocular stereo using SAD.
- dataScript.m to run all stereo methods over the Middlebury dataset.
- multiocular.m to run multiocular stereo over the Middlebury dataset.

Result files:

The result files for each image in the dataset has been saved in the individual folders with names :

- depthSAD.png
- depthNCC.png
- depthSSD.png
- depthMultiOcularNCC.png
- depthMultiOcularSAD.png
- depthMultiOcularSSD.png

Observations:

SSD

- SSD measure is an unsigned similarity measure and hence it only reflects the difference between the left and right patches and does not depend on the patch order i.e. $ssd(p_1, p_2) = ssd(p_2, p_1)$. Thus the occluded and foreshortened regions in the images are enhanced by black spots, as seen in the image. One way to avoid these black patches is to use adaptive windows for patch similarity with smaller patches at edges and larger patches at similar textures.

SAD

- This is a computationally very cheap (~ 7sec/imagepair) but it is more prone to intensity variation and intensity spikes among the image pairs. Since it is a signed measure of pixel dissimilarity, it is able to recognize directional (signed) edges very effectively. Hence it does not lead to black patches at edges and the occluded regions (which generally happen to be at the edges). The intensity variation in images affects the disparity and discontinuity detection in the images and can be avoided by using rank/consensus transforms (essentially variants of local contrast limited adaptive histogram equalization for each patch) before calculating the cost.

NCC

- NCC seems to perform better on almost the entire dataset and can be seen as a more general pixel similarity measure for stereo vision. Since NCC calculates the normalized cross correlation, it seems to overcome the shortcomings of SSD (unsigned measure)

and SAD (prone to intensity variations and spikes). It is computationally expensive and does not detect discontinuities as accurately as SAD and similarities as accurately as SSD. This can be avoided by using semi-global optimization as suggested by Hirschmuller et. al.

General Issues

- Since the patch matching uses pixel similarity, it becomes almost impossible to distinguish between patches that have almost the same texture besides having very little differences in depth. These patches lead to convolved zero-boundary features on the texture plane as observed in many result images having planes with similar textures. These iso-depth and iso-color contours lead to patches in the disparity. A lucid explanation of the same can be that : since all patches in the neighbourhood look the same (i.e. have the same cost matching) the algorithm fails to identify the corresponding right patch for a left patch and hence it leads to random disparity values for textures with same depth and same color/texture.

Multi-ocular

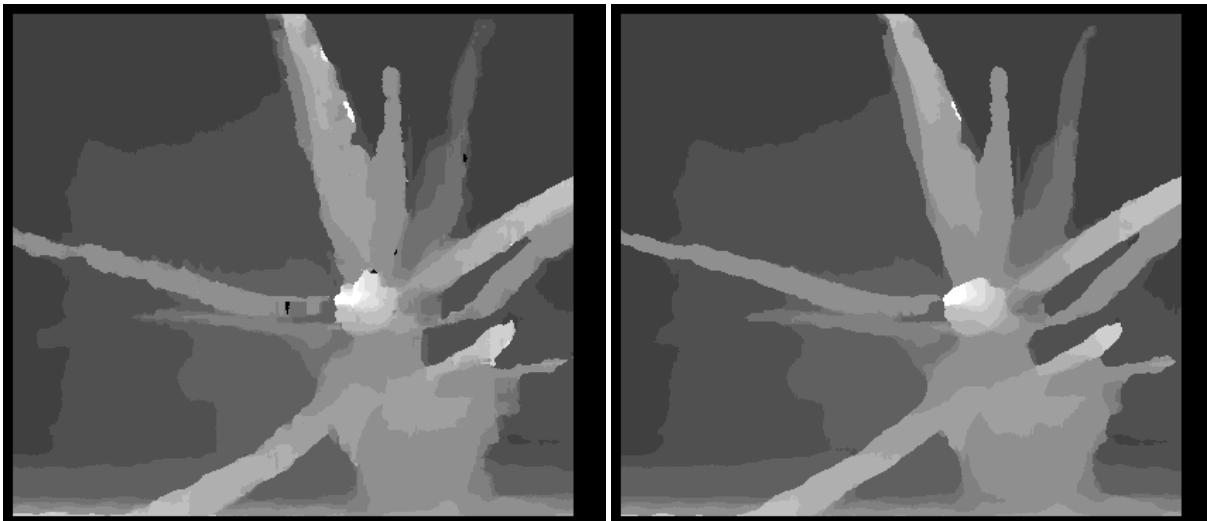
- In using multiple images for depth estimation, the accuracy goes up as we have more data to estimate the depth and the false positive values and matches are reduced significantly. We look at the image from a larger field of view (4 images covering almost 120 degrees) and hence the occluded regions in the object in the image reduce significantly and hence we have lesser patches/pixels whose disparity has to be estimated from the neighbourhood.

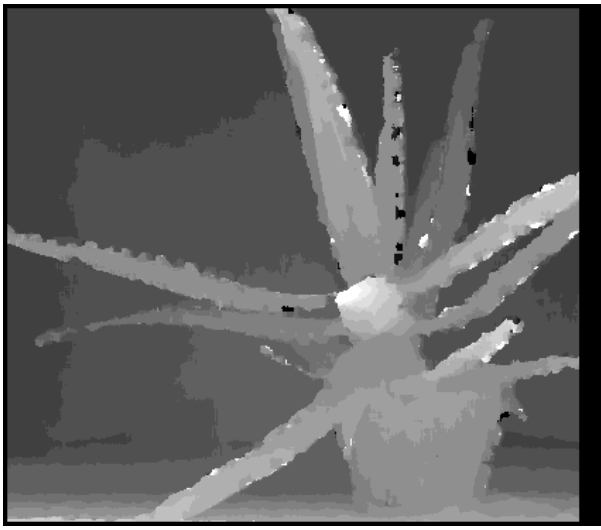
Results

Input



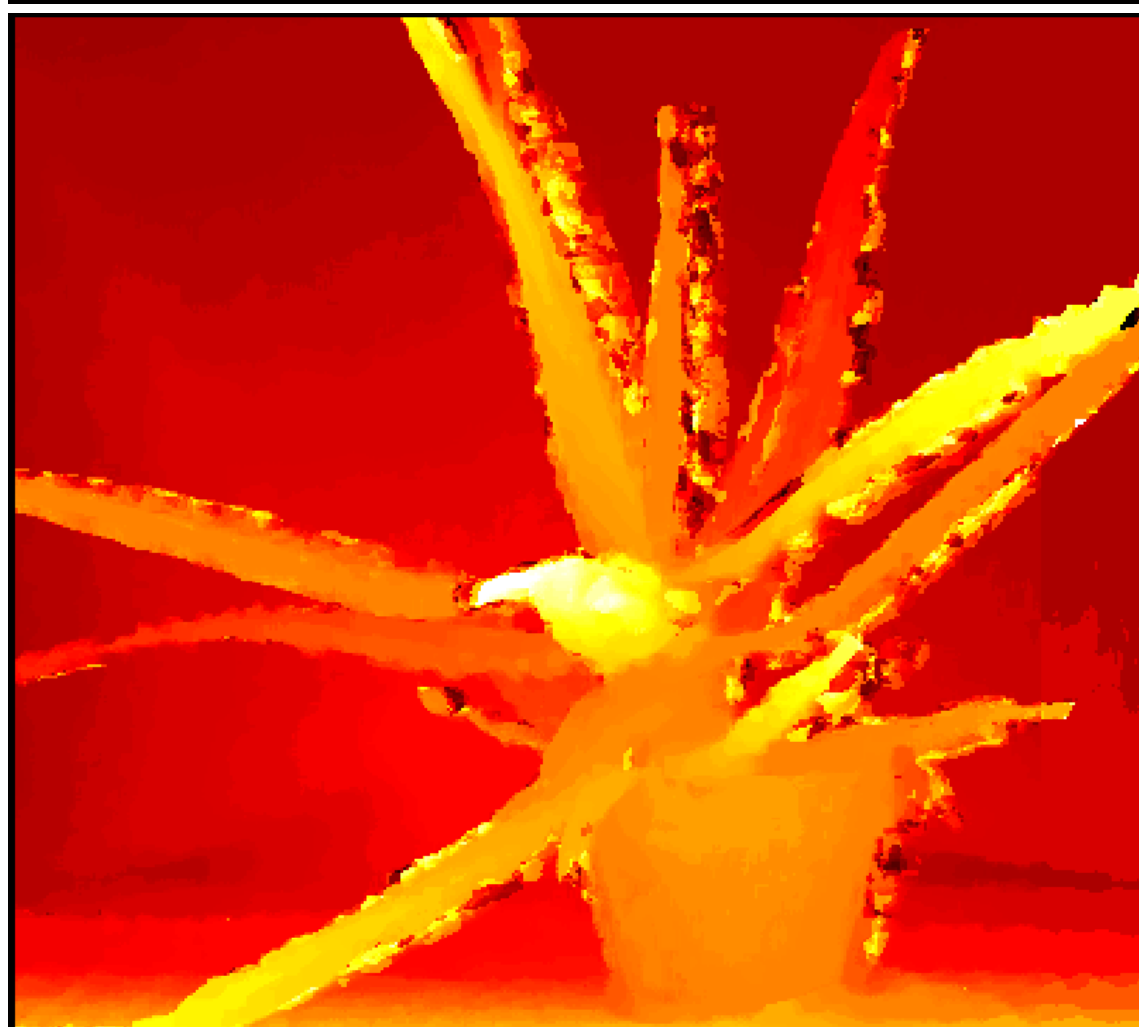
Binocular output





Multi Occular output





Future Work

- A lot of algorithms using dynamic programming, semi-global matching, scaled transformations and convolutional neural networks are being deployed to solve the stereo problem. I also reviewed a similar paper on the Application of convolutional neural networks for stereo matching which can be found [here](#).