

CSCI 4830/5722 Computer Vision, Fall 2016
Instructor: Fleming
Assignment 3, Due Monday, October 17th, by 11:55pm

Stereo and Segmentation

For Assignment 2, you will implement a few functions in a stereo pipeline example provided by mathworks.com

Provided files:

In your Matlab Command Window, type

```
>> edit DepthEstimationFromStereoVideoExample
```

What You Have to Do:

Task 1 (30 points): Calculate disparity using the SSD algorithm

Implement the SSD matching algorithm for a range of window sizes, and create a disparity image $D(x, y)$ such that $\text{Left}(x, y) = \text{Right}(x + D(x, y), y)$ when matching from left to right. Remember to reduce the search only over epipolar lines. You should also restrict the search using the disparity range (see line 56 in the original script).

Write a function to replace the disparity built-in function on line 54. Your function needs to accept as an input value the size of the window search. Call your function three times for the following window sizes: 1, 3, 5.

Create a 2x2 subplot and display the disparity map results from these three function calls, plus the one obtained using Matlab's built-in disparity function.

Note: when using a window size bigger than 1, it is common practice to use a Gaussian weighting of the window, so that the center has greater influence than the periphery. Convolve your window matrices with a Gaussian filter prior to computing the sum of squared differences.

Task 2 (30 points): Calculate disparity using the NCC algorithm

Implement the normalized cross correlation (NCC) matching algorithm for a range of window sizes, and create a disparity image $D(x, y)$ such that $\text{Left}(x, y) = \text{Right}(x + D(x, y), y)$ when matching from left to right. Similarly, remember to reduce the search only over epipolar lines and to restrict the search using the disparity range (see line 56 in the original script).

Information on how to compute NCC for image-processing applications here:

https://en.wikipedia.org/wiki/Cross-correlation#Normalized_cross-correlation

The inner product of vectors approach mentioned on Wikipedia, also here, slide 66:

http://www.gris.tu-darmstadt.de/teaching/courses/ws0910/cv2_ws0910/slides/l3-stereo-v1_0.pdf

Write a function to replace the disparity built-in function on line 54. Your function needs to accept as an input value the size of the window search. Call your function three times for the following window sizes: 3, 5 and 7.

Create a 2 x 2 subplot and display the disparity map results from these three function calls, plus the one obtained using Matlab's built-in disparity function.

Task 3 (30 points): Generate outliers map - Refine the disparity by performing a left-right consistency check.

The disparity map $d_{LR}(x)$ is acquired considering as reference image the left image of the stereo pair. If the right image is considered as reference, then the disparity map $d_{RL}(x)$ is acquired. The disparity maps $d_{LR}(x)$ and $d_{RL}(x)$ can be useful in detecting problematic areas, especially outliers in occluded regions and depth discontinuities. One strategy for detecting outliers is the Left-Right consistency check introduced by ¹. In this strategy, the outliers are disparity values that are not consistent between the two maps and therefore, they do not satisfy the relation:

$$|d_{LR}(x) - d_{RL}(x - d_{LR}(x))| \leq T_{LR}, \text{ where } T_{LR} \text{ is a user-defined threshold.}$$

Write a function that accepts as inputs two disparity maps (LR and RL) and a value for T_{LR} and returns a binary image where the outliers have the value 1 and the inliers have the value 0.

Call this function twice, passing as inputs the LR and RL disparity maps from SSD and NCC matching with window size of 3. Use a T_{LR} value of 1. With this value, pixels with difference equal to 1 in the Left-Right consistency check are not considered outliers. Plot the two resulting binary images side by side.

Task 4 (10 points): Compute depth from disparity

Now that we have a disparity image, computing depth at every pixel in the left image should be very straightforward. Use equation 11.1 from the Szeliski textbook.

Write a function to replace the `reconstructScene` built-in function on line 64. Your function should have the same input arguments as the built-in version and it should return a matrix of depth values for each pixel location from the left image.

Submitting the assignment:

Make sure each script or function file is well commented and it includes a block comment with your name, course number, assignment number and instructor name. Zip all the .m and image files together and submit the resulting .zip file through Moodle as Assignment 2 by Monday, October 17th, by 11:55pm.

¹ M. Humenberger, T. Engelke, W. Kubinger, *A census-based stereo vision algorithm using modified semi-global matching and plane-fitting to improve matching quality*, CVPR ECV Workshop, 2010.