

CS 223b: Introduction to Computer Vision

Assignment 2: Dense Stereo

Due date: **Thursday, January 20th 23:59 PST**

You may work in **teams of up to 3 persons**

Submission via email with subject “Assignment 2” to cs223b@gmail.com

1 Correlation based stereo (20 pts)

1.1 Basic Algorithm (10 pts)

Implement in MATLAB a correlation-based stereo algorithm that takes as input a pair of rectified stereo images and outputs a disparity map with the same dimensions as the input images. Your function should look like:

```
function disparity = stereo(imageL, imageR);
```

In order to perform this correlation, you will need to make at least the following choices:

- The **maximum disparity** maxd your algorithm will accept. The bigger this value is, the more computing time you need. A value that is too small, however, will prohibit your algorithm from detecting objects close to the camera.
- The **window size** of your comparison region. Determining how much a single pixel has moved in the right image compared to the left image is ambiguous, because other pixels might have the same grey value by accident or because of noise. Therefore, you need to compare whole image regions, whose size is called window size.
- The **similarity measure** for the windows to be compared. For every disparity $d \in [0..\text{maxd}]$, you need to compute this similarity measure between the original window in the left image and a window in the right image that is moved by d pixels. The value d for which you get the best similarity will be your disparity.

Some rectified testing images for your algorithm can be found at

<http://cs223b.stanford.edu/homework/hw2>

1.2 Algorithm optimizations (10 pts)

The algorithm you implemented should perform fine on the images `toy1LR.pgm` – `toy3LR.pgm`. In the Stanford Racing Team (www.stanfordracing.org), we are experimenting with stereo as a sensor for autonomous robot desert racing. These conditions are much harder, because cameras don't work as well in the desert and because there is less texture on the road. Try your algorithm on these images `real*.pgm`.

Come up with at least two improvements to your algorithm on these real world images, whose goal is to decrease the rate of mis-estimated disparities. If at one pixel your algorithm cannot reliably estimate a disparity, it is better to mark it “-1” (for missing) than to give a wrong estimate.



Figure 1: A typical stereo scene for desert racing. The left and right images are swapped so you can try human stereo vision by moving this paper close to your eyes.

Describe the idea of your optimizations briefly and write about conditions where it is expected to perform well and where you believe it is challenged hard.

Your solutions will be judged on novelty and smartness of the idea, as well as the quality of the resulting disparity map. For this quality, the amount of false disparity estimations will decrease the quality score more than the amount correct disparity estimations increases it.

Use the following function signature:

```
function disparity = stereoOpt(imageL, imageR);
```

2 Short Answers (5 pts)

1. **Camera Baseline (3 pts):** Assume you want to use two of the webcams from last week's assignment for stereo. Your project requirements state that you should be able to distinguish a point that is 10m from your camera from one 9m away. How far away do you have to place your two cameras horizontally if you want their viewing directions to be parallel?
2. **Depth recovery (2 pts):** Can depth be recovered from a stereo pair taken under the following circumstances? Briefly justify your answers.
 - (a) Two images taken by orthographic projection from cameras with parallel optical axes?
 - (b) Two images taken by a single perspective proj. camera rotated about its optical center?

Deliverables

- Your design choices for questions 1.1 and 1.2 as well as your answers for questions 2.1 and 2.2 in the body of your message.
- Your two MATLAB m-files for calibration as attachments called `stereo.m` and `stereoOpt.m`.
- The resulting disparity images called `toy1.bmp – toy3.bmp` and `real1.bmp – real10.bmp`.

Finally, please state the percent effort each individual member of your team has contributed to this assignment (numbers should add up to 100%). Submit everything with subject "Assignment 2" to `cs223b@gmail.com`.