CS223B Homework Assignment 1

Sebastian Thrun, Dan Maynes-Aminzade, Mitul Saha
cs223b@gmail.com
Due Monday, January 23, 11:59 PM

Assignment Overview

This assignment is worth 10% of your total grade. It is graded out of a total of 100 points. It consists of three problems: one programming problem, and two written problems.

What to Hand In

You will electronically submit the following items:

- All of your source code for Problem 1. Please make an effort to submit neat, commented code so that we can give you partial credit if your results are not good.
- A `README` file containing instructions for compiling and running your source code for Problem 1. This file should be in plain ASCII text.
- A `WRITEUP` file containing a description of your approach. State your methods for solving the problem and explain what worked and didn't work. Include the names and email addresses of all of your group members. This file should be in plain ASCII text or PDF format.
- An `OUTPUT` directory containing a series of 100 output images that have been produced by your code, one for each source image. We will use these images to grade the correctness of your solution.
- An `ANSWERS` file containing the answers to the written portion of this assignment (Problems 2 and 3). Include the names and email addresses of all of your group members. This should be in plain ASCII text or PDF format.

To submit your completed assignment, email a single ZIP or TAR archive containing all of the above components to cs223b+submit@gmail.com. You must include all of the components; we will not accept partial submissions.

1: Programming a Feature Detector (65 Points)

Getting Started

In this problem, you will be asked to identify cars on a road in a typical driving situation. You will be given a data set of images taken from a car driving down a highway, and your code will produce a monochromatic image in which each pixel is tagged as CAR (white) or NON-CAR (black).

You are free to use Matlab, OpenCV, or any other computer vision libraries. In your write-up, please list any library functions you used and describe what you used them for.

Files You Need

The images that you will be working with are in the directory

`/afs/cs.stanford/edu/class/cs223b/www/homework/hw1/`

You can also find them on the web at

`http://cs223b.stanford.edu/homework/hw1/`

The source images are in JPG format and the labeled images are in PNG format. Both formats can be read directly into Matlab or OpenCV.

There are two directories of images:

- The `SOURCE` directory contains the collection of 100 source images that your program will use as input. The images are numbered in sequence from `source00.jpg` to `source99.jpg`.
- The `LABELED` directory contains 10 hand-labeled monochrome images corresponding to the first 10 source images, numbered in sequence from `labeled00.png` to `labeled09.png`. You can compare your results against these hand-labeled images to judge the correctness of your solution. We do not provide hand-labeled images for the other 90 source images, but we will examine your results on these images as well when grading your solution.

What Your Code Should Do

Given an input image, your code should produce an output image of the same pixel resolution. All of the pixels in the output image should be either black or white, with white pixels representing CAR and black pixels representing NON-CAR.

Run your code on each of the images in the `SOURCE` directory. For each source image `sourceXX.jpg`, write out a corresponding output image in PNG format named `outputXX.png`. Note that your code should produce PNG rather than JPG images; the lossy compression of JPG-formatted images will result in pixels that are not pure black or pure white. Put all 100 of these output images in a directory called `OUTPUT` and include them in your assignment submission.

Grading

We will grade your solution based on two metrics:

1. *Pixel-by pixel comparison to hand-labeled image.* We will compute the norm of the difference between the output image that your code produces and the hand-labeled solution image. The smaller this value, the better your score. In other words, your output image should differ from the hand-labeled image in as few pixels as possible.
2. *Documentation and approach.* Because the first metric does not always accurately represent the quality of your work (for example, it weights nearby cars more heavily than distant cars), we reserve a portion of your grade based on more subjective factors. In particular, we will consider how reasonable your approach is, the efficiency and correctness of your code, and how well you document your solution.

2: Perspective Geometry (20 Points)

Consider a scene that contains 5 collinear features, A, B, C, D, and E, where collinearity is defined in 3D coordinates. We know that under perspective projection, the projected features are also collinear in the 2D image plane.

(a) Suppose A, B, C, D, and E are all equidistant. Will the *order* of the projected points (along the projected line) always be the same in the camera image? If yes, argue why this must be the case. If no, provide a counterexample.

(b) Will the observed features also be equidistant in the image plane? If yes, argue why. If not, argue why not.

(c) What are the implications of the previous answer for stereo vision? Specifically, suppose we find five collinear features a, b, c, d, e in the left image, will they also be collinear in the right image? If yes, argue why; if not, provide a counterexample.

(d) Suppose the projected points a, b, c, d, e are collinear in both stereo images. Can we expect that the order is preserved? Again, argue the correctness or provide a counterexample.

3: Hough Transform (15 Points)

Suppose you use the Hough Line Transform (described in Section 5.2 textbook) to map the edges of a regular planar hexagon in the (ρ, θ) space. Assume the hexagon is centered at the origin.

(a) What will be the arrangement of the "peaks" in the (ρ, θ) space?

(b) How does the arrangement change when you rotate the hexagon about the origin?

(c) How does your answer in (a) compare with the answer for 6 lines all intersecting at the origin?