University of Colorado at Boulder ECEN 4532

Instructor: Mike Perkins

Perspective Transformations and Motion Tracking

Lab 3
Lab report due on February 25, 2019

This is a Python only lab. Each student must turn in their own lab report and programs.

1 Introduction

This lab will continue our exploration of image processing with two new topics: correcting perspective distortions and motion tracking in video. As before, we will write our algorithms mostly from scratch.

1.1 Background

The lecture material covered in class explains how 3-D objects are projected onto a 2-D image plane. If a set of points in 3-D world space are known a-priori to lie in a plane, then the projections of those points onto two camera images taken from different perspectives are related to each other by a linear transformation. In Part I of this lab you will process images that clearly exhibit perspective distortion and remap them into images that (ideally) do not.

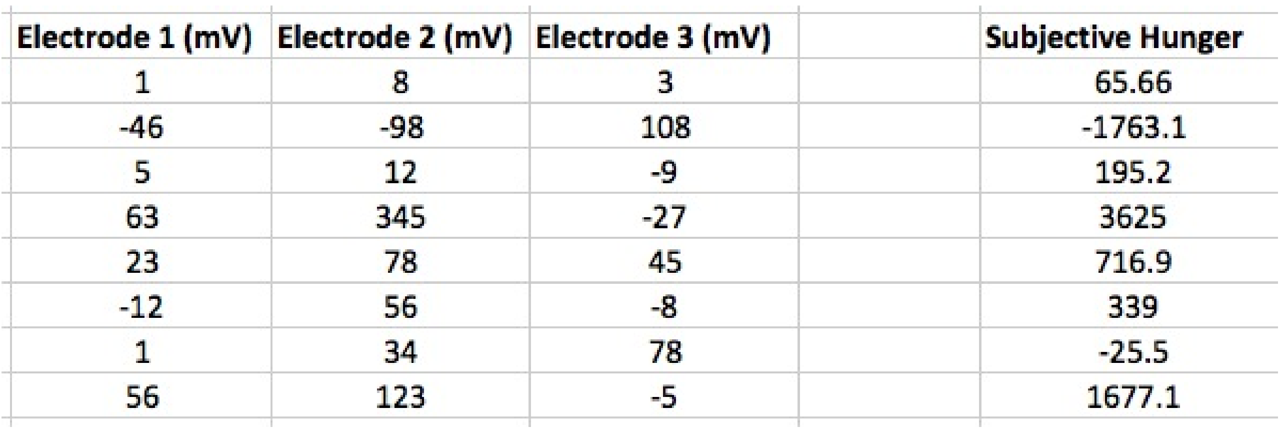
In Part II we will look at the problem of tracking moving objects between temporally adjacent images in a video sequence. The approach we use will be based on first building an image pyramid, and then using block-based motion estimation. Motion estimation is an integral part of modern video compression algorithms such as H.264 (the workhorse of .mp4)

2 Part I: Correcting Perspective Distortion

2.1 Linear Regression

We'll begin Part I by performing a linear regression using the pseudo inverse technique derived in class. Assume you are working for a company doing research on dieting. They believe they have figured out a way to predict how hungry people feel based on electrode measurements made at 3 locations in the brain.

The subjective hunger that people feel is self-reported on a scale of -2000 to 2000, where -2000 is feeling so stuffed that the thought of further food is nauseating, while 2000 means your stomach thinks your throat's been cut. You suspect that there is a linear relationship between the subject's subjectively reported hunger and the measured electrode voltages. The data at your disposal is in the following table



Electrode 1 (mV)	Electrode 2 (mV)	Electrode 3 (mV)		Subjective Hunger
1	8	3		65.66
-46	-98	108		-1763.1
5	12	-9		195.2
63	345	-27		3625
23	78	45		716.9
-12	56	-8		339
1	34	78		-25.5
56	123	-5		1677.1

Figure 1: Experimental data

Assignment

1. Perform a linear regression to predict the subjectively reported hunger based on the electrode data. Write the code to calculate the pseudo-inverse yourself, implementing the technique discussed in class. In other words, don't use a library call. Learning goal: learn how to use numpy to multiply matrices, find the transpose, compute a matrix inverse, etc. What values do you get for your prediction coefficients x_1, x_2, and x_3?
2. Add to your script a computation of the MSE between your predicted values and the measured values.

2.2 Perspective Correction

The lecture notes describe a method for processing a picture to change the apparent position from which the picture was acquired. In some situations, this can be used to remove, or at least reduce, unwanted perspective distortions.

Assignment

1. Select 20 points from test picture `PC_test_2.jpg` to use as the basis of your remapping calculation. Alternatively, you may use a picture you take yourself provided it shows visible perspective distortion (use your favorite tool to resize your picture to a size similar to that of `PC_test_2.jpg` if you go this route). Make sure all the points you select lie in a single plane. It is important that all the points not lie on the same line, although they can lie on two or more lines, and it will be easiest if they do. Include in your report a table of these 20 points, i.e. their `(row, col)` values, and the 20 points to which you are mapping them.
2. Write the code to process the image. Compute the pseudo-inverse yourself, don't use a library call. Include the original image and the processed image in your report.
3. Try a different mapping of your 20 input points to output points. See, for example, if you can zoom in or zoom out.
4. Describe any visual effects you see related to the apparent resolution of the processed image, and discuss how such phenomena come about. Do you notice anything about points that were not in the same plane as the points you selected and how they were remapped?

3 Part II: Tracking Motion in a Video Sequence

Coming soon to a Canvas module near you.