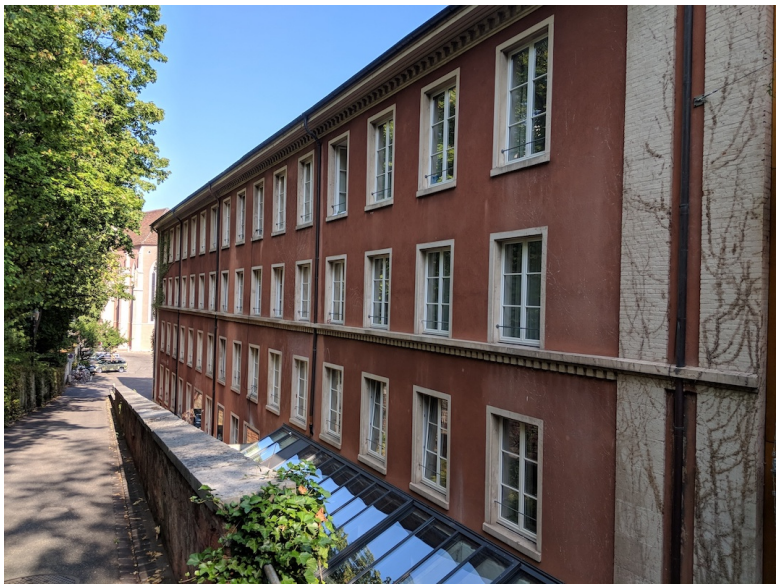
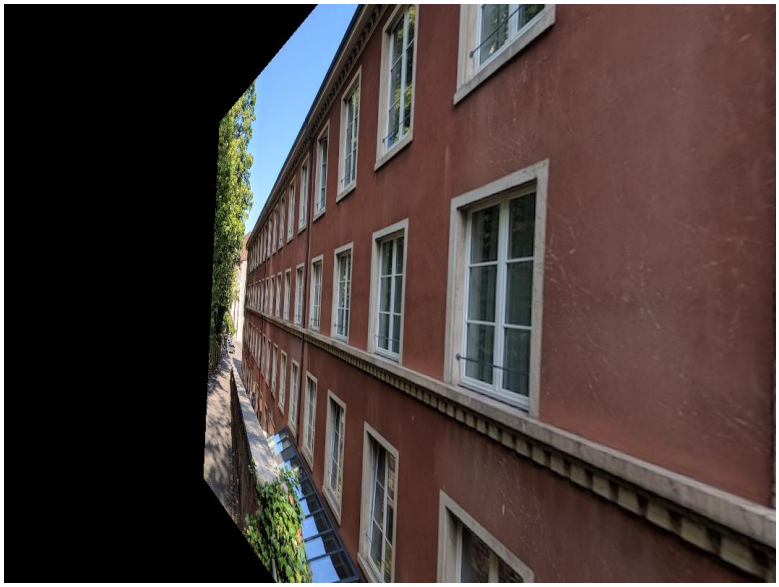
ECEN 4532 - Lab 3: Perspective Transformations and Motion Tracking

Andrew Teta

February 25, 2019





Contents

Introduction **3**

1 Introduction **3**
1.1 Background 3

2 Perspective Distortion Correction **3**
2.1 Linear Regression 3

1 Introduction

In this lab, we explore the low-level implementation of perspective distortion correction, and a simplified model of video motion tracking between frames in Python. We will mostly use `numpy` to manipulate N-D arrays of pixel information. Additionally, I used a free photo editing software called GIMP to select pixel indexes.

1.1 Background

Our algorithm for perspective distortion correction relies on the math theory of linear algebra, using a linear transformation matrix to map pixels from one image to another. Then, the process of motion tracking between frames will be based on a series of image pyramids, using block-based motion estimation to "search" for a matching region in the next frame.

2 Perspective Distortion Correction

2.1 Linear Regression

We begin by defining linear regression as one approach to modeling the relationship between a dependent and independent variable. This method is commonly used to fit a model to an **over-determined** set of data points. By over-determined, we mean there are more equations defining the data set, than there are variables, often occurring when many measurements are performed to estimate a small number of parameters.

Mathematically, we have a relationship of the form, $Ax = c$, where A is over-determined. This only has a solution if $\mathbf{c}$ lies in the column space of A, however there will not be an exact solution if A is over-determined and we instead seek an $\mathbf{x}$ that minimizes the mean-squared-error (MSE), defined $E = |A\mathbf{x} - \mathbf{b}|^2$. Ultimately, we want to find a "pseudo-inverse", A^+, such that $\mathbf{x}^\$ = A^+\mathbf{c}$