An Exploration of Image Stitching as a Linear Optimization Problem

Initial Project Proposal

Thomas Pasfield and Madeline Gorman

Embry-Riddle Aeronautical University, Daytona Beach, FL.

March 8, 2024

1 Introduction

Image stitching provides a useful tool for both consumers and for industry. In consumer products, people generally encounter it in the form of panoramic images on their phones. In industry, it finds applications in anything from photogrammetry to remote sensing. While it isn't a particularly new technology, it still has broad application and has a simple entry-point. We feel that this makes it ideal as a student project.

In computational imaging, the most common representation of imaging is in the form of a linear system. In applications such as deblurring, edge detection, and other filtering, it is common to see a formulation such as Equation 1.

$$\vec{\mathbf{b}} = A\vec{\mathbf{x}} + \vec{\varepsilon} \tag{1}$$

In this form, $\vec{\mathbf{x}}$ and $\vec{\mathbf{b}}$ represent $mn \times 1$ sized vectors, equivalent to stacking the columns of the image atop each other sequentially. A represents the transformation matrix, which acts upon the range of the image. This must be of size $mn \times mn$. $\vec{\varepsilon}$ often represents the residual, or random noise. It is not adequate to represent an image as a linear transformation upon its range in this problem, as we need to warp and translate the images rather than filter them.

To better address this problem, we will approach the image processing in the domain of the image instead. We want to modify the coordinate space rather than the values at those coordinates. This new formulation is shown in Equation 2.

$$\vec{\mathbf{b}} = T\vec{\mathbf{x}} \tag{2}$$

Unlike the form in Equation 1, $\vec{\mathbf{x}}$ and $\vec{\mathbf{b}}$ represent 2×1 vectors of the coordinate, and the transformation matrix is simply represented by T, which is 2×2. This method is performed on each pixel individually, which allows for efficient parallelization. All the image warping and transformations which we must perform can be represented by this form with ease.

As we approach this problem, we expect to encounter transformations such as:

- Translation
- Rotation
- Scaling/Aspect
- Affine Transformation
- Perspective Transformation
- Barrel Distortion

2 Planned Approach and Implementation

We plan to approach this problem via optimization, where the results are scored by some metric, and we iterate to minimize this score. Metrics based on the overlap of pixels will certainly be computationally inefficient but will be quite simple to set up. This may make a good first iteration for our work, and we can expand upon this and implement a feature-detection based approach, with algorithms such as SIFT.

Our primary focus is on the optimization algorithms rather than the computational efficiency of other components. This problem will require multi-variable optimization, of which we don't yet know the difficulty of. At the very least we have parameters such as the x and y components for translation and scaling, an angle θ for rotation. Parameters for the affine and perspective transformations will be fairly similar, to the extent that affine may likely be replaced by perspective if we encounter too many variables. We expect a minimum of 4 additional parameters for each of these transforms that we must tune.

Our work will be performed using Python and managed via a GitHub repository. We will make extensive use of the Numpy and Scipy libraries. We will likely use `scipy.optimize` in our early implementation to ensure that our model functions as expected before we implement our own optimization. We may perform early image operations via the Python Image Library (PIL) or OpenCV if necessary.

If we reach our stretch goals related to blending the images, OpenCV will likely be utilized to perform these range transformations, as they are purely for our qualitative review, and are not a primary focus of this work.

3 Project Goals

The focus of this work is to implement an optimization-driven image stitching approach. It does not seek to develop any new methods.

Primary Goal

We seek to be capable of stitching at least 3 grayscale images, each of a size of 100×100 pixels or greater, with a very rough initial guess provided by the user. We will operate within at least a rectilinear coordinate space, and we seek to apply at least 3 types of optimization.

Stretch Goals

Provided that we complete our primary goal, we would like to pursue some additional features and capabilities. Many of these are qualitative in nature.

1. Add support for color images.
   We expect this to be fairly simple, but to triple the computations performed.
2. Add support for range transformations.
   Modifying exposure, contrast, white balance, and other range-based transformative properties, we can create a more visually appealing output.
3. Add support for more coordinate spaces.
   This allows for a much more accurate output, but is much more difficult for us to intuitively approach.

4 Data Availability and Sources

Computational imaging is a broad enough field that they have well-established standard test images and datasets. There are many resources out there which we can utilize to test out methods on, such as:

- `https://github.com/visionxiang/Image-Stitching-Dataset`
- `https://paperswithcode.com/dataset/udis-d`
- `https://paperswithcode.com/datasets?task=image-stitching&mod=images`

We could also utilize other public datasets such as archived images from NASA's Mars2020 Perseverance Rover. An example of an image stitch of this data can be seen in Figure 1.

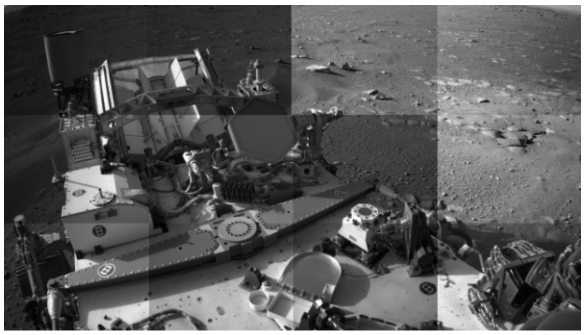

Figure 1: *A manually stitched image series of the Mars2020 Perseverance Rover. This is performed within rectilinear space with translation and scale operations. No range transformation was performed. This was performed by Thomas.*

It is also possible for us to use free open-source software such as Blender to generate our own ideal image data. This could allow us to create ideal data without distortion, camera shake, or other inconveniences. This process would be very simple and could be done within 2-3 days easily.