Project 3

Maryam Darei

Computational Photography
Dr. Pless

Image Mosaicking

Background

Image stitching or photo stitching is the process of combining multiple photographic images with overlapping fields of view to produce a segmented panorama or high-resolution image. Most approaches to image stitching require nearly exact overlaps between images and identical exposures to produce seamless results, although some stitching algorithms actually benefit from differently exposed images by doing high-dynamic-range imaging in regions of overlap.

The purpose of this project is to stitch two or more images to create one panorama image. In this project, we need to capture multiple such images.

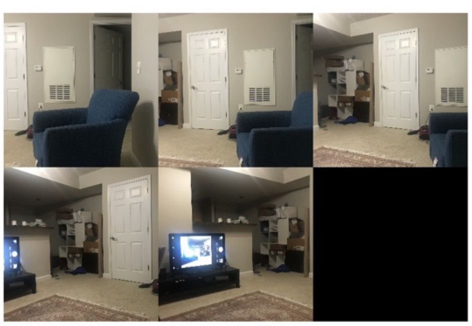
Data Description:

Data Characteristics:

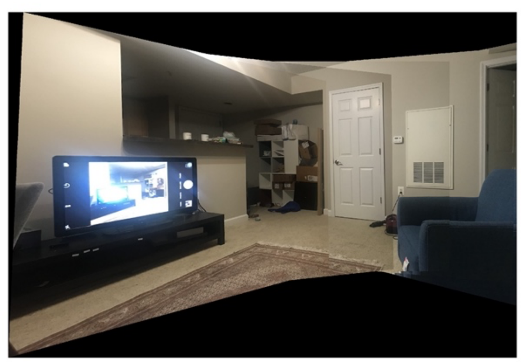
- We have some RGB images as an input which would have the potential to stitch to.

Sample of input and output:

Input 1



Input 2



Project description:

In order to estimate image alignment, algorithms are needed to determine the appropriate mathematical model relating pixel coordinates in one image to pixel coordinates in another. Algorithms that combine direct pixel-to-pixel comparisons with gradient descent (and other optimization techniques) can be used to estimate these parameters.
Distinctive features can be found in each image and then efficiently matched to rapidly establish correspondences between pairs of images. When multiple images exist in a panorama, techniques have been developed to compute a globally consistent set of alignments and to efficiently discover which images overlap one another.
A final compositing surface onto which to warp or projectively transform and place all of the aligned images is needed, as are algorithms to seamlessly blend the overlapping images, even in the presence of parallax, lens distortion, scene motion, and exposure differences.

Image stitching issues

1- Differences in images' light and background changing:

   Since the illumination in two views cannot be guaranteed to be identical, stitching two images could create a visible seam. Other reasons for seams could be the background changing between two images for the same continuous foreground.

2- Other major issues to deal with are the presence of parallax, lens distortion, scene motion, and exposure differences.
3- In a non-ideal real-life case, the intensity varies across the whole scene, and so does the contrast and intensity across frames.
4- Additionally, the aspect ratio of a panorama image needs to be taken into account to create a visually pleasing composite.

Key point detection
Feature detection is necessary to automatically find correspondences between images. Robust correspondences are required in order to estimate the necessary transformation to align an image with the image it is being composited on. **Corners**, **blobs**, **Harris corners**, and **differences of Gaussians of Harris corners** are good features since they are repeatable and distinct.

SIFT and SURF are recent key-point or interest point detector algorithms but a point to note is that these are patented and their commercial usage restricted. Once a feature has been detected, a descriptor method like SIFT descriptor can be applied to later match them.

Feature detection is a low-level image processing operation. That is, it is usually performed as the first operation on an image and examines every pixel to see if there is a feature present at that pixel. If this is part of a larger algorithm, then the algorithm will typically only examine the image in the region of the features
Occasionally, when feature detection is computationally expensive and there are time constraints, a higher level algorithm may be used to guide the feature detection stage, so that only certain parts of the image are searched for features.

For panoramic stitching, the ideal set of images should have a reasonable amount of overlap (at least 15–30%) to overcome lens distortion and have enough detectable features. The set of images will have consistent exposure between frames to minimize the probability of seams occurring.

Explain some terms:
1- Image registration is **an image processing technique used to align multiple scenes into a single integrated image**. It helps overcome issues such as image rotation, scale, and skew that are common when overlaying images.

Image registration methods:
feature detection, feature matching, mapping function design, and image transformation and resampling.

2- Image warping is the process of digitally manipulating an image such that any shapes portrayed in the image have been significantly distorted. Warping may be used for correcting image distortion as well as for creative purposes like morphing.

3- Resampling is the technique of manipulating a digital image and transforming it into another form. This manipulation could be for various reasons - change of resolution, change of orientation, i.e. rotation, change of sampling points, etc.
You can increase or decrease the amount of data in the image (resampling). Or, you can maintain the same amount of data in the image (resizing without resampling). When you resample, the image quality can degrade to some extent.
When you RESAMPLE AN IMAGE, you are actually changing the image's file size by adding or deleting pixels within the image. This is determined by changing either the image dimension OR its resolution change. You can also change the resolution to match the output media type.
4- Compositing is the process of combining multiple images to form a single, cohesive image. It's a common visual technique in photography and film.
Homographies are a transformation that we can use to transform one image into another when both images are pictures of the same plane, but from a different perspective

Method

We create a panorama using feature-based image registration techniques. I use detect blob as a feature and mosaic images based on those.

Data pre-processing and Feature Engineering:

The image set used in this example contains pictures of a living room. These were taken with an uncalibrated smartphone camera by sweeping the camera from Right to left along the horizon.

I used the interior image. I guess it would have the better result as there are minimal differences in scene light.

As seen below, the images are relatively unaffected by any lens distortion, so camera calibration was not required. However, we need to preprocess our data, if lens distortion is present (the camera should be calibrated, and the images undistorted prior to creating the panorama and we can use camera calibrator applications).

Code description:
The code is in **MATLAB**.

First attempt: In this approach, it used the SURFF feature.

- First step:

It should be loaded with your images. We need the grayscale 2-D grayscale input image for detectSURFFeatures() function.

- Second Step:

In this method find and set features in all of your images and then match these features between the last image and the previous one (Image(n) and Image(n−1)). Then estimate the geometric transformation, T(n), that maps Image(n) to Image(n−1). And in 3rd step compute the transformation that maps I(n) into the panorama image as T(n)∗T(n−1)∗...∗T(1). That it initializes all the transforms to the identity matrix.

It is used from detectSURFFeatures(grayImage) function for features detecting. This function detects SURF features and returns SURFPoints object. This function finds blob features.
A Blob, is anything that is considered a large object or anything bright in a dark background, in images, we can generalize it as a group of pixel values that forms a somewhat colony or a large object that is distinguishable from its background. Using image processing, we can detect such blobs in an image.

The main interest of the SURF approach lies in its fast computation of operators using box filters, thus enabling real-time applications such as tracking and object recognition.
The SURF method (Speeded Up Robust Features) is a fast and robust algorithm for local, similarity invariant representation and comparison of images. Similarly to many other local descriptor-based approaches, interest points of a given image are defined as salient features from a scale-invariant representation.
The SURF feature detector works by applying an approximate Gaussian second derivative mask to an image at many scales. Because the feature detector applies masks along each axis and at 45 deg to the axis it is more robust to rotation than the Harris corner.
It is composed of two steps:
1- Feature Extraction: The approach for interest point detection uses a very basic Hessian matrix approximation.
2- Feature Description : The creation of SURF descriptor takes place in two steps. The first step consists of fixing a reproducible orientation based on information from a circular region around the keypoint.

Then, construct a square region aligned to the selected orientation and extract the SURF descriptor from it.

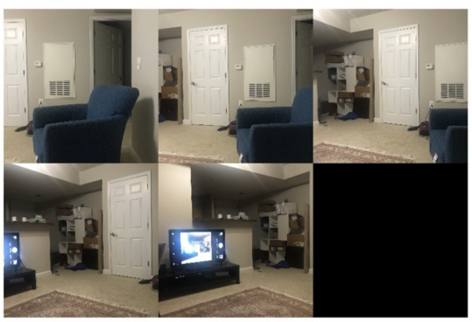
$$H(f(x,y)) = \begin{bmatrix} \frac{\partial^2 f}{\partial x^2} & \frac{\partial^2 f}{\partial x \partial y} \\ \frac{\partial^2 f}{\partial x \partial y} & \frac{\partial^2 f}{\partial y^2} \end{bmatrix}$$

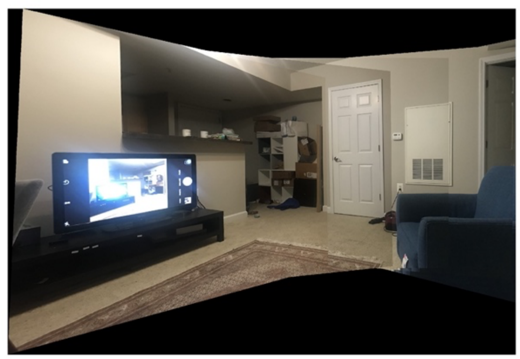
In fact, here are all the transformations in tforms compared to the first image, which was a much easier way to register images because it is easier to process consecutive images from beginning to end. While this method is aesthetically difficult and does not result in the most beautiful panoramic image. The most beautiful image is formed when the middle image has the least distortion .This is accomplished by inverting the transform for the center image and applying that transform to all the others.
For solving this problem you can use the projective2d outputLimits method that encapsulates a 2-D projective geometric transformation.You can create a projective2d object using estimates a geometric transformation that maps pairs of control points between two images. Then, compute the average in the horizontal direction for each transform and find the image that is in the center. It is because the scene is known to be horizontal. If another set of images is used, both the horizontal and vertical may need to be used to find the center image. In the last step, apply the center image's inverse transform to all the others. At this point, a blank panoramic image is created, and all images are mapped to it. To sketch the images in this empty panorama, imwarp() and to overlap the images on top of each other vision.AlphaBlender () has been used.

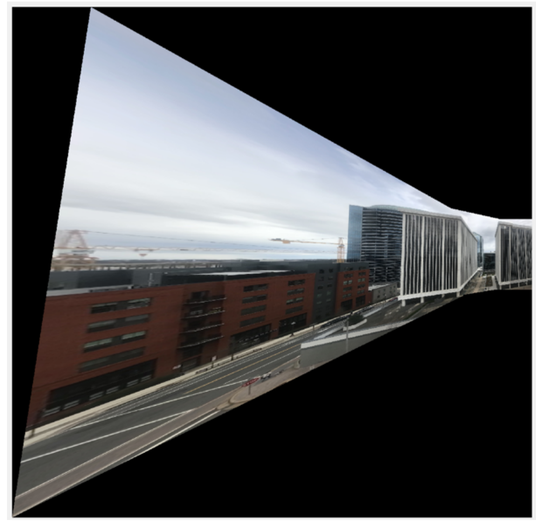
Input and output:

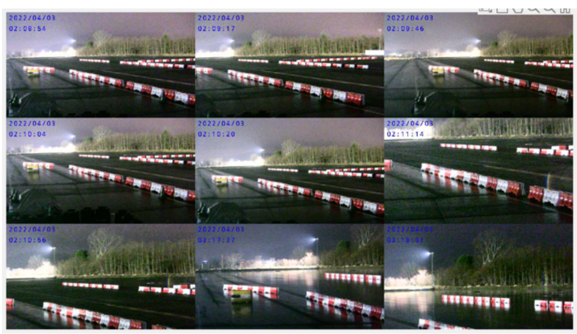
Input images

Panorama image

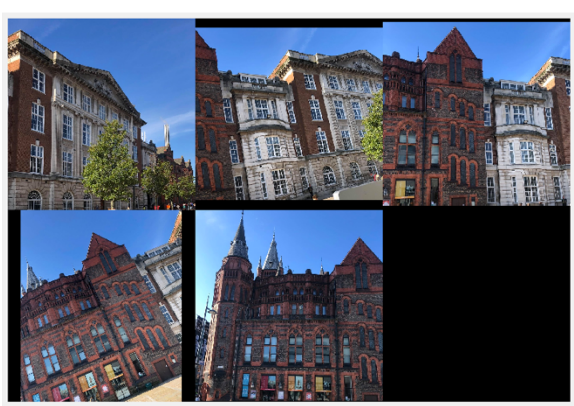









2022/04/03
02:10:20



Second attempt:

In the second approach, it is used of SIFT function.
SIFT helps locate the local features in an image, commonly known as the 'keypoints' of the image. These keypoints are scale & rotation invariant that can be used for various computer vision applications, like image matching, object detection, scene detection, etc.

In general, SIFT algorithm can be decomposed into four steps: Feature point (also called keypoint) detection. Feature point localization. Orientation assignment

First step:
The images were loaded from the data folder into the algorithm and then converted to single, greyscale images. Each of the images is then passed through a SIFT detector to find the keypoint indices of each image.

Second step:
each of the images key points needed to be matched using their SIFT descriptors. RANSAC is then used to find each of the transformations between images, it is used because it finds the most likely transformation by removing outliers. This transformation is then applied to the next image, and the two are stitched together.
In order for the outputted image to not look distorted, each image needed to be stitched in a particular order.

- If there is an odd number of images:
  - The starting point is the exact middle image.
  - This is then stitched with the image to its right.
  - The two stitched images are then stitched with the image to its left.
  - An alternating pattern of right and left is then used to stitch the remaining images.

- If there is an even number of images:
  - The starting point is the middle-left image.
  - This is then stitched with the image to its left.
  - The two stitched images are then stitched with the image to its right.
  - An alternating pattern of left and right is then used to stitch the remaining images.

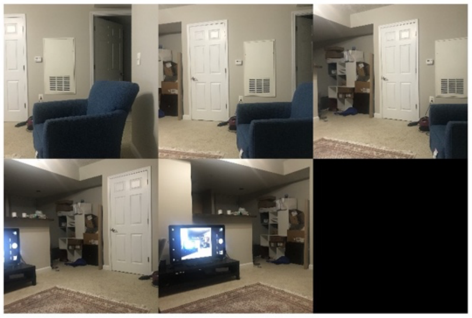
This way the algorithm can work for any number of images.

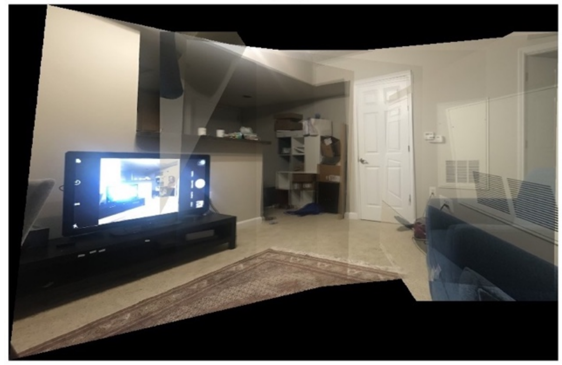
Below you can see that this approach is not. It is completely clear in the output panorama images.

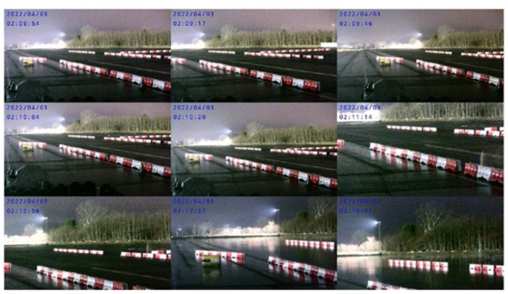
Input and output:

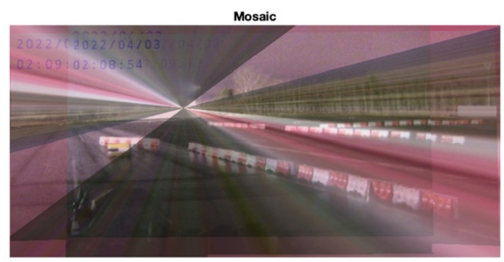
Input images

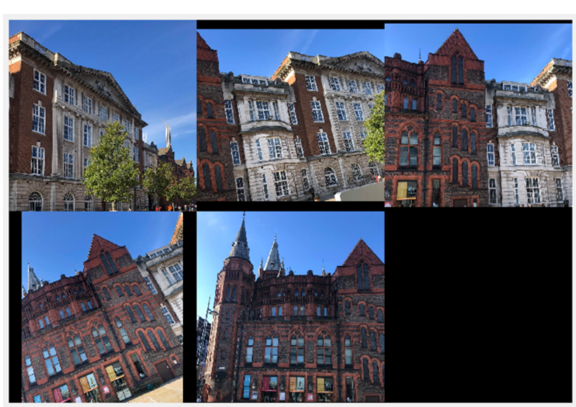
Panorama image











Bells & Whistles (Extra Points):

- **image geometric warping without using functions that warp the images:**
  We can do things manually. For example, adjusting the color and uniformity, color and light of images, cropping images so that the sharing of two images in a row have a good common border. Also, based on wrap definition, warp can be thought of as a mapping that sends each coordinate pair (u, v) into a corresponding pair (x, y). Any such map can be expressed as two. functions, X that determines the transformed x coordinate from (u, v), and. Y that determines the transformed y coordinate from (u, v). So, we can use it this definition and write a function based on that.

Another Bells & Whistles:

1- One of the things that can be done in this concept is calibration. We should preprocess our data if lens distortion is present (the camera should be calibrated and the images undistorted prior to creating the panorama and we can use camera calibrator applications).

2- non-in order image: In all of our approaches the images should be in-ordered. We should try to have some images as an input that are not in order and try to find the best matching to create the panorama. For this issue, maybe we can compare the features between every two images and decide which one is the next.

3- 2D or multi-row stitching is more difficult.

4- Another thing is related to noise. Based on my understanding noise can have a bad effect on finding the feature in images. We can pre-process our input images and cut the noise before finding the features in each image.

5- compensation for motion in the camera and scene, and more advanced modeling of the geometric and photometric properties of the camera is another issue that can be considered.

6- Horizontal and vertical images should be considered.