E27 Computer Vision - Final Project: Creating Panoramas
David Nahmias, Dan Spagnolo, Vincent Stigliani
Professor Zucker
Due 5/10/13

Sources
Brown, M.; Lowe, D.G., "Recognising panoramas," *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on* , vol., no., pp.1218,1225 vol.2, 13-16 Oct. 2003
doi: 10.1109/ICCV.2003.1238630

Motivation and Goals:

In this project we aimed to implement a now popular feature in modern day cameras, panorama image stitching (also known as image mosaicing). Usually when this is done on a camera, either several pictures are taken manually which are then stitched together or the view is panned through a scene while the camera automatically takes images which will then be stitched together.

For this project we implemented a version of panorama image stitching where the program searches through a folder of images (for example, a folder of summer vacation photos), and determines if there are appropriate matches for some of the images. If there exist panoramas in the folder, the program outputs all possible panoramas that can be validly stitched together.

While stitching together two images known to overlap is a relatively simple problem (with one challenge being dynamically matching points in multiple images), the task of mosaicing multiple images and the problem of determining which images depict overlapping scenes from a folder of images taken at many different times and places is a much more complex challenge. This essentially requires extracting and matching keypoints from each image and every other image in the folder (because any 2 images *could* overlap at *any* point in the scene), and only constructing panoramas for image matches with valid homographies. Another challenge is to smooth the discontinuities of image boundaries, a task known as image splining. The task of image splining for arbitrarily overlapping, multi-image panoramas is an open problem in computer vision (Brown & Lowe, pg 5).

Implementation:

The first step of the project involved implementing the simpler panorama problem: stitching images of know overlap. To this end, with the assistance of some OpenCV feature detection algorithms, we used the scale invariant feature transform (SIFT) algorithm, a popular image feature detection method put forth by David Lowe. This algorithm outputs an ordered list of the best SIFT feature matches, such as those depicted in Image 1.

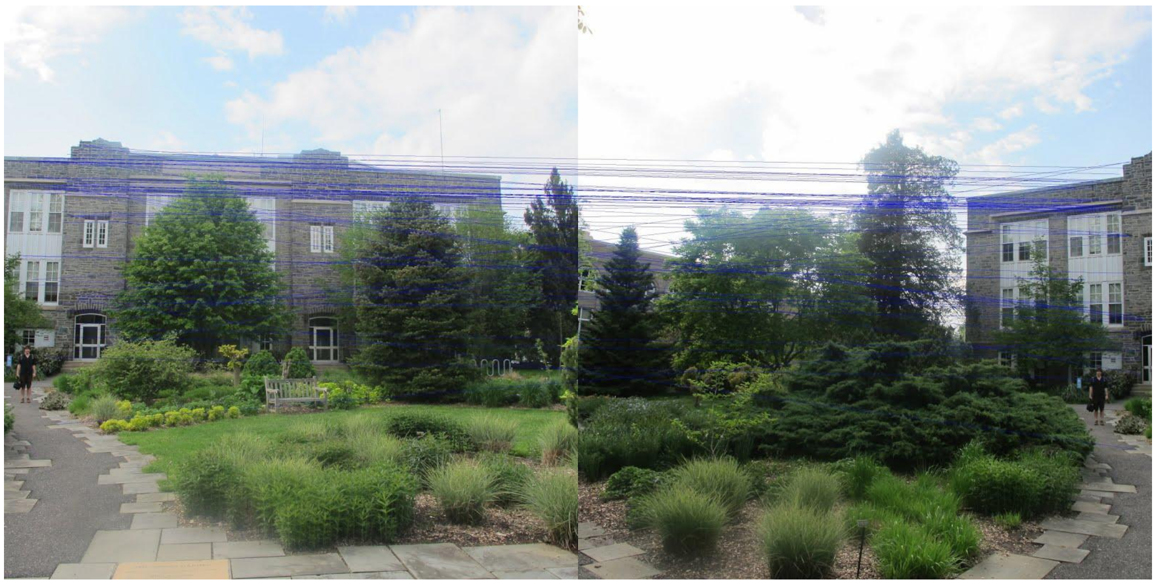
Image 1: SIFT feature match of overlapping images of Hicks

As the name suggests, the main benefit of using SIFT features is that scale is invariant for these features. Another feature detection scheme could be using Harris corner extraction with cross-correlation matching to find corresponding features. However, this method is neither invariant to scale nor rotation. This means that if the user is moving in or out of the scene (which would change the scene scale), or if the user's camera is rotating, that the Harris corner extraction method is going to perform poorly. In practice, users are not able to prevent their hand from creating these rotations or differences of scale, which is why SIFT features are preferred for dynamic feature matching in this scenario (Brown & Lowe, pg 1).

With this set of keypoint matches, developing a panorama simply required warping the perspective of one to the other (and determining the minimum bounding box of the entire panorama). The minimum bounding box of the entire panorama requires finding each of the individual bounding boxes of the warped panorama images and using overlap information to determine a bounding box for the whole image.

To dynamically choose which potential keypoint matches should be used in the homography, we used the RANSAC method. This method is ideal for this situation (likely, rather than hand-selected keypoint matches) because it dynamically adds more points to improve a potential homography if those points are a close match in the initial homography. From the keypoints shown in **Image 1**, we produced the stitched scene shown in **Image 2**.

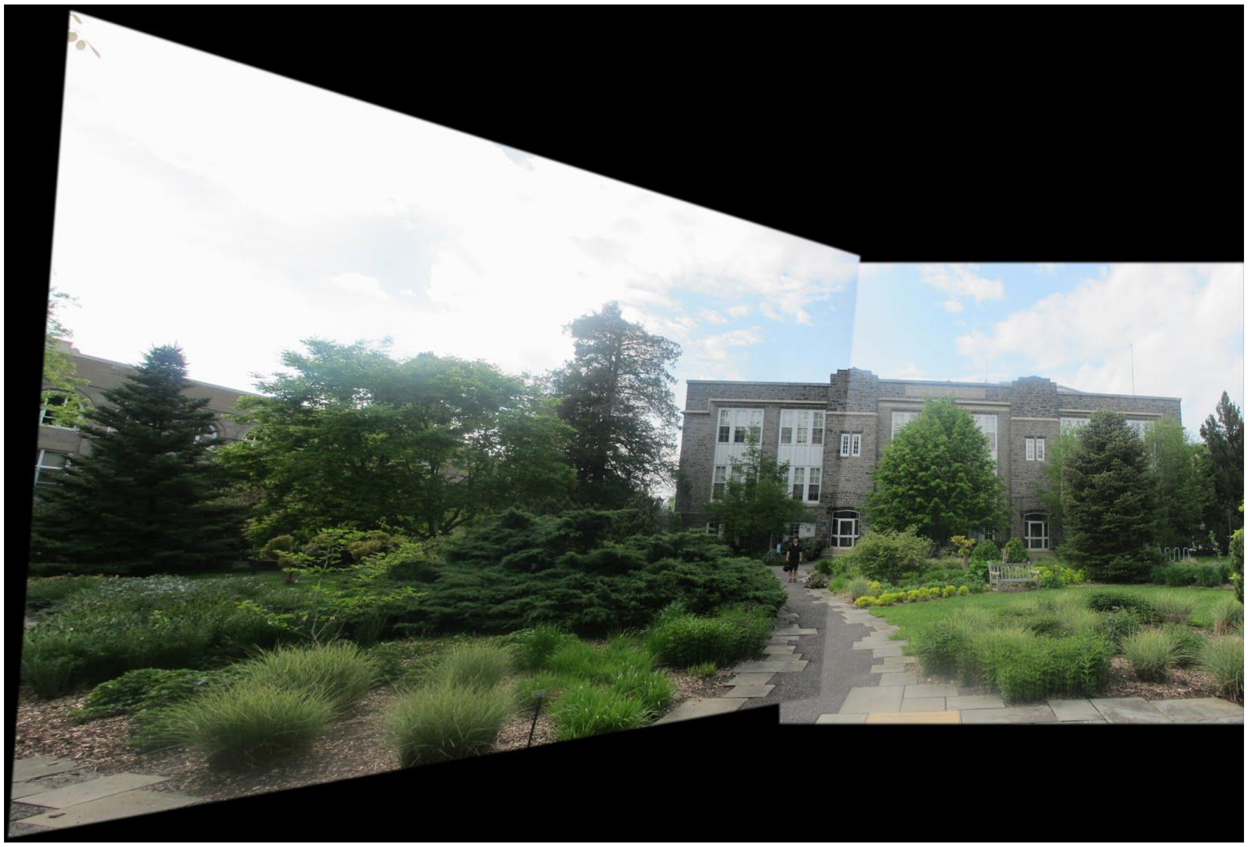
Image 2: Hicks after warping the perspective of one overlapping image to the plane of the other.

Finally there was the task of image splining. Originally, we attempted to use a multi-band spline defined by J. Burt and E. H. Adelson in their paper *A multiresolution spline with application to image mosaics*. However, we eventually settled for a more naive approach where a single image's pixels are weighted stronger at the image boundary. This approach seems to work well for our purposes.

The project was a very collaborative effort, most of the time all three members working together simultaneously on the project. However by breaking it down into the largest part we each contributed to: Vincent worked on coding of the SIFT keypoint matching, and RANSAC image matching. David worked on the top down design of the project as well as being able to process an arbitrary number of pictures and dynamically finding panoramas from a folder of images. Dan primarily worked on the bounding boxes for the image mosaicing, as well as SIFT keypoint matching with Vincent. All three lab partners wrote this report.

This project has a few directions it could go in. First off, the implementation of panning the camera through a scene would be interesting to implement via the computer webcam. This way the computer could be rotated around an area and a panorama could be created. Secondly, we could continue to work on the splining methods described by Burt and Adelson. Finally, we could use Bundle Adjustment, as defined in the Brown and Lowe paper, to validate our RANSAC image matches, leading to a more robust result (Brown & Lowe, pg. 3).

As a final direction, we believe the incorporation of a high-level scene descriptor classifier could greatly reduce the computational complexity of panorama stitching from a random folder. This could be used as preliminary clustering method to restrict the search for potential overlaps. In our knowledge-free approach, we assume every image could overlap with any other. With a classifier, though, we could recognize a given image as a particular scene (i.e. beach, forest, city, room), and restrict the search for correspondences to similarly classified images.

Sample Output: