# Panoramic Image Stitching

Alex Karantza
ask9900@rit.edu
Intro to Computer Vision 2011-1

## ABSTRACT

*The goal of this project was to implement a system in MATLAB that could search a collection of photographs and extract multi-image panoramas, automatically stitching them together where appropriate. The primary paper guiding this implementation was* (M. Brown). *The image registration aspects of this paper were successfully implemented and tested on real-world data, producing satisfying panoramas. The paper by Brown and Lowe goes further than the basic registration done here and touches on additional mechanisms for aesthetic improvement of the panoramas which were not implemented.*

## I. INTRODUCTION

A common image editing operation is to take multiple photographs of the same scene and stitch them together to achieve the effect of a higher resolution photograph, or a wider field of view than the camera can provide. This process of creating a panorama from images can be described as simply finding the transformation for each image to properly align it with its neighbors, and then to render out all the images with their transformations. In most implementations of panorama stitching, the user is required to determine these transformations manually (or some assumptions are made, such as that the camera is only panning, not rotating or altering the focal length).

The primary paper for this project (M. Brown) describes an algorithm for automatically identifying matching features between images and using these points to compute the transformation. The likelihood of two images being connected can also be easily determined, allowing the algorithm to run on a large set of images and being capable of only stitching together images where appropriate.

This project focuses on this aspect of panorama stitching - the automatic extraction of feature points, computing a homography between overlapping images, and determining connected sets of images to render panoramas. The paper also describes additional steps that were not implemented, such as bundle adjustment to compensate for rotation of the camera and gain compensation and blending to remove artifacts in the rendered panorama.

## II. PREVIOUS WORK

Panorama stitching is a well-researched topic with many implementations available of varying degrees of complexity. Most modern digital cameras or smartphones are capable of basic stitching. Implementations that perform this kind of completely automatic stitching are less common; the

authors of the primary paper have a product called AutoStitch (Brown), meant to run on smartphones, which implements this algorithm with a number of patented improvements for efficiency. Microsoft Research has the Image Composite Editor (Microsoft Corporation) as part of the Photosynth product which also performs this same level of automatic image stitching along with extraction of three dimensional scene detail based on movement of the camera.

In addition to these full panorama applications, the smaller components of the algorithm are useful topics of research themselves. The scale-invariant feature transform algorithm used to identify and match images for panorama stitching can also be used for any other feature detection application such as object recognition and tracking.

## III. IMPLEMENTATION

This project was implemented in MATLAB using limited external libraries. It essentially follows the algorithm laid out in sections 2 and 3 of (M. Brown), dealing with feature detection, image matching, and homography estimation.

### 1. MULTIPLE IMAGE LOADING

The first step in the implementation is to load the images. This is done by reading in the contents of every jpeg image in a particular directory. Since the algorithm is tolerant of stray images that do not contribute to panoramas, the user may place whatever images they have in this path without consequence. The images are stored in an array of structures; these structures store information such as the filename, the data of the original image, the grayscale data used for feature matching, the detected features, and additional information about pair-wise matching.

### 2. SIFT FEATURE EXTRACTION

Once the image data is available, the next step is to determine feature points in all these images that will be used to match them together and find a homography. This feature extraction is performed by the VLFeat (Andrea Vedaldi) library's implementation of SIFT. SIFT (Lowe) is an algorithm which finds corners of data in a Gaussian pyramid and stores them in terms of radial gradients. This allows the feature to be tolerant of changes in scale, rotation, and to some extent intensity (since the gradients are relative.) This is necessary since photographs taken of the same scene may well vary in roll of the camera, focal length, and exposure settings.

### 3. HOMOGRAPHY ESTIMATION

The next step is to attempt to match points in each image against points in every other image. If sufficient points are found in common, these points are run through the RANSAC algorithm (M. Fischler) to determine if they can be accurately described by a homography matrix. If this is also found to be the case, the two images are marked as connected in a binary adjacency matrix. The RANSAC algorithm with homography extraction is implemented by a Computer Vision and Image Processing library for MATLAB (Kovesi). This function returns both the homography transform as well as a list of inlier features.

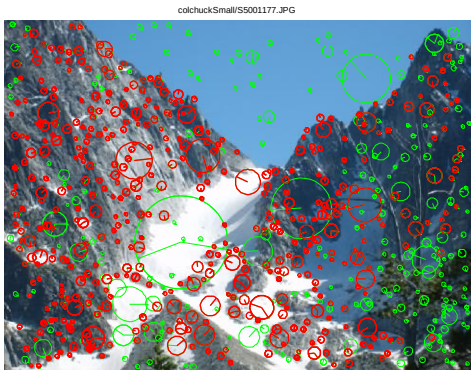


*Figure 1: An example image.*
*Red feature points were matched with other images.*

By comparing the ratio of inliers to the total number matched and thresholding against a suitable constant, we decide if these two images are actually connected.

The paper (M. Brown) discusses using kd-trees and early termination criteria to speed up this search, the basic implementation of which is $O(N^2)$. This project was not particularly concerned with efficiency, and so every image is compared against every other. For many large images with many (hundreds, thousands) of feature points, this can take a prohibitively long time. As a result, the examples given were computed on scaled copies of the input images.

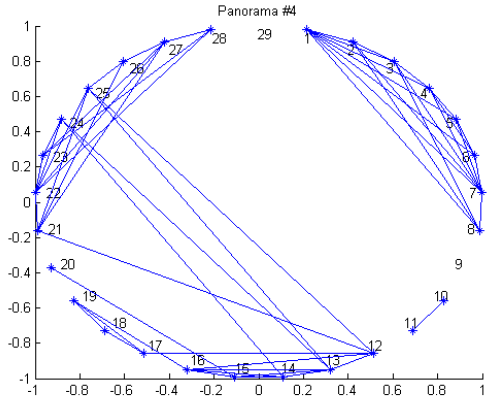


*Figure 2: The adjacency graph of a set of 29 images. You can see two large panoramas, one small panorama (images 10 and 11) and a few images that do not belong in any panoramas.*

## 4. Graph Analysis

The adjacency matrix found as a result of the RANSAC filtering can be used to determine which images belong to which panoramas, and which images are not associated with any others. Panoramas are represented as isolated connected components in the adjacency graph, and MATLAB provides functions for labeling and extracting these components. Images which have no connections to other images can be ignored for the panorama rendering stage. Images that do share connections to their neighbors are collected into sets for each identified panorama group.

## 5. Panorama Rendering

At this stage in the algorithm, we know the number of panoramas detected, which images belong in that panorama (and which connections they make to their neighbors), and the homographies along each of those connections. The reference paper uses a process of Bundle Adjustment to compute the overall best transformation by comparing feature points against every matching image and optimizing for camera rotation parameters. This is necessary for overcoming the errors introduced by a rotating, as opposed to merely translating, camera. This project's implementation does not do 3d bundle adjustment, but merely traverses the graph and concatenates the homographies. This has the benefit of being quick and easy and still producing acceptable images as long as the panorama does not span a wide field of view.

Once the graph is traversed (taking the image with the most matched points to be the center) each image is given a

transformation based on the concatenation of all previous homographies of images in the traversal path.

MATLAB's imtransform function is used to apply these transformations to the source images (along with a mask of 1s), and the panorama bounds are extended to include the new data.

The reference paper discusses many approaches for blending these images to reduce the unavoidable artifacts of image stitching. It is always possible that the camera has translated slightly, introducing parallax, or that some object in the scene has moved between photographs. These errors can be avoided by careful blending whenever two images overlap. This project's implementation did not seek to fix these particular errors (including the visible misregistration resulting from directly concatenating the homographies), and instead we directly average all overlapping images using the mask mentioned above. The rendered images do indeed contain some visible artifacts, which would be unacceptable for a practical implementation, but the motivation of this project is to demonstrate the practical application of image registration. Seeing the obvious overlapping of matching images, including the misregistrations, is a reasonable result.

## IV. RESULTS

The implementation was tested against a set of 100 images taken with a conventional consumer digital camera on a hiking trip. In this set of photos are a number of overlapping images intended to be stitched into panoramas, along with many other images that are isolated. The script successfully extracted six panoramas from the dataset (consisting of as few as two images, and as many as eighteen), and stitched them together with minimal errors. Some of the panoramas span a wide field of view, and the simple homography concatenation causes extreme warping of images at the edges. This effect would be eliminated by the use of more appropriate bundle adjustment.

Intermediate images were displayed as well to visualize the detected feature points, and which features were considered RANSAC inliers. This helped guide the selection of threshold parameters.

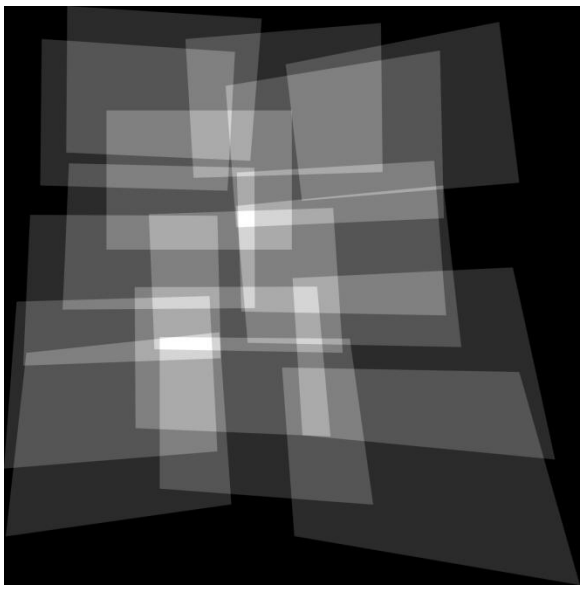

*Figure 3: The mask generated for the Aasgard Pass panorama showing areas of image overlap. Some points on the final panorama have up to six images contributing.*

## V. Conclusions

Overall the project successfully implemented the image registration aspects of the primary paper and produced a satisfactory result given real-world input. Additional features such as bundle adjustment, more sophisticated blending algorithms, and gain compensation could be added to the rendering stage to produce more natural panoramas with fewer registration glitches.

Performance could also be substantially improved. Running the algorithm on the full dataset at native resolution would take many hours to complete; even running at quarter resolution takes more than an hour. The MATLAB libraries used do not focus on performance, but rather on providing a simple and educational interface. If this implementation were intended for practical use, it would likely be rewritten in a more suitable language and use optimizations such as the kd-tree for matching, the early termination of homography estimates (since bundle adjustment will work fine with as few as three or four matching images) and other optimizations that were ignored in this project for the sake of simplicity and readability of the code.

## VI. References


Andrea Vedaldi, Brian Fulkerson. "VLFeat." 2007. <www.vlfeat.org>.

Brown, Matthew. "AutoStitch: A New Dimension in Automatic Image Stitching." 2011. <http://www.cs.bath.ac.uk/brown/autostitch/autostitch.html>.

Kovesi, Peter. "MATLAB and Octave Functions for Computer Vision and Image Processing." 2000. <http://www.csse.uwa.edu.au/~pk/research/matlabfns/>.

Lowe, D. "Distinctive image feature from scale-invariant keypoints." International Journal of Computer Vision (2004): 91-110.

M. Brown, D. Lowe. "Automatic Panoramic Image Stitching using Invariant Features." International Journal of Computer Vision (2007): 59-73.

M. Fischler, R. Bolles. "Random sample consensus: A paradigm for model fitting with application to image analysis and automated cartography." Communications of the ACM (1981): 24:381-395.

Microsoft Corporation. "Image Composite Editor." 2011. <http://research.microsoft.com/en-us/um/redmond/groups/ivm/ice/>.

*Figure 4: The largest panorama produced from the test image dataset, of Aasgard Pass and Colchuck Lake in Leavenworth, WA. Lots of overlapping images allowed for good registration, however images on the far extents experience excessive warping.*

*On closer inspection, slight errors in registration can be seen. Apparently some of the photographs of this mountain were taken from a slightly different position, causing parallax on foreground objects such as the tree on the left. Since the distant mountains produced more feature points than the foreground objects, the RANSAC algorithm discarded the matches on the tree in favor of properly aligning the background.*