Robust Panoramic Image Stitching

CS231A Final Report

Harrison Chau
Department of Aeronautics and Astronautics
Stanford University
Stanford, CA, USA
hwchau@stanford.edu

Robert Karol
Department of Aeronautics and Astronautics
Stanford University
Stanford, CA, USA
robkarol@stanford.edu

Abstract— Creation of panoramas using computer vision is not a new idea, however, most algorithms are focused on creating a panorama using all of the images in a directory. The method which will be explained in detail takes this approach one step further by not requiring the images in each panorama be separated out manually. Instead, it clusters a set of pictures into separate panoramas based on scale invariant feature matching. Then uses these separate clusters to stitch together panoramic images.

Keywords—Panorama; SIFT; RANSAC; Homography; Keypoint; Stitching; Clustering

I. INTRODUCTION

You travel often for work or pleasure but you have had a passion for photography since childhood. Over the years, you take many different photographs at each of your destination. Over the years, you compiled a giant collection of images in your personal photograph database. But later in your life you regret not creating panoramas from each of your images to decorate your new house. Unfortunately, you do not have the pictures organized well, and so you can't remember which images were taken on each trip. You do not know how to sort the photos or create the panoramas. The developed algorithm solves both of these problems. Figure 1 shows the flow of system.

Figure 1: Diagram of a system

These problems are important to solve in the automation of image processing for the future. Automation of many features is becoming more common place and expected. This project addresses this issue by taking a procedure which used to be done manually, and making it automatic. It allows more users to become interested in photography without the need for the technical knowledge to create panoramas.

The methods used to solve this problem involve a combination of machine learning and computer vision algorithms combined in a novel way. First, an image database is created. This is done manually by taking pictures from a variety of locations around the world. SIFT keypoints are then found for each image. The algorithm then sorts through the images to find keypoint matches between the images. A clustering algorithm is run which separates the images into separate panoramas, and the individual groups of images are stitched and mosaicked together to form separate panoramas.

II. RELATED WORK

A. Prior Algorithms

Algorithms which have been used to create panoramas have been developed in the past and implemented many times. Some of these algorithms have even been optimized to run in near real time applications on mobile processors. Additionally, there are some well-known machine learning concepts which have been used to classify data into separate groups based on common characteristics. This section will outline some of the prior work which has been done in these two areas.

B. Panorama stitching

Using images to create panoramas is a well studied problem. In particular we focused on the work done by Lowe (2004) and Szeliski (2004). Each of these papers focused on creating a high quality panorama from a large number of input pictures. The methods which these papers were based on, including the geometry of image recognition using camera matrices and creating a holography between two images are well known concepts extended with more advanced features for blurring and interpolation between pixels for a final higher quality image. There are two main methods used for image alignment and stitching, direct, and feature based. Direct is a brute force method which uses all the data in an image. It is more accurate since it takes in all the available information but requires inputs from a human operator to define the matching image pairs. It is also significantly slower from a computational standpoint. The feature based method does not require user input as it focuses on matching specific points in an image automatically, and assumes the rest of the image is consistent around those points. For this work since the goal was a fully automated system, the feature based method was chosen.

C. Clustering algorithms

There are a number of different classification algorithms which are well known and widely implemented to solve a number of different problems. In particular many of these algorithms come from machine learning techniques. Some examples include support vector machines, and k-means clustering. These algorithms all need to be tweaked for the given task. This paper describes the changes made in order to get well known algorithms to work for our purpose.

D. Current algorithm

The current method proposed by this paper uses invariant feature based approach, SIFT, to automate image sorting and panorama stitching. The use of SIFT allows robust matching of pictures in the image database irrespective of camera zoom, rotation, and illumination. This method discovers the relationship between the matching images so it allows the image database to be broken down into individual panorama datasets.

The current method takes in an image database and sorts the images based on relationships using SIFT, keypoint matching, RANSAC, and clustering. Then the panorama image datasets are stitched together linearly.

III. Technical Solution

This section details the method developed to cluster and stitch together pictures from a database to get the images shown in section IV.

A. Image Database

The following images are a sample of the about 45 images which were included in the initial database. These images in particular mainly correspond to 8 of the final panoramas while the entire database consists of many more images from a more varied set of locations. These images were taken around the Stanford University campus, in particular the engineering quad, the main quad, and inside the d.School. Figure 2 shows a sample of images from the picture database. Additional pictures taken at a nearby restaurant, London, United Kingdom, Bordighera, Italy, and Cannes, France have been included as well. The extraneous images that do not belong to a panorama have been included were taken at Stanford. The final image database size is 80 pictures after the addition of the images taken in Europe.

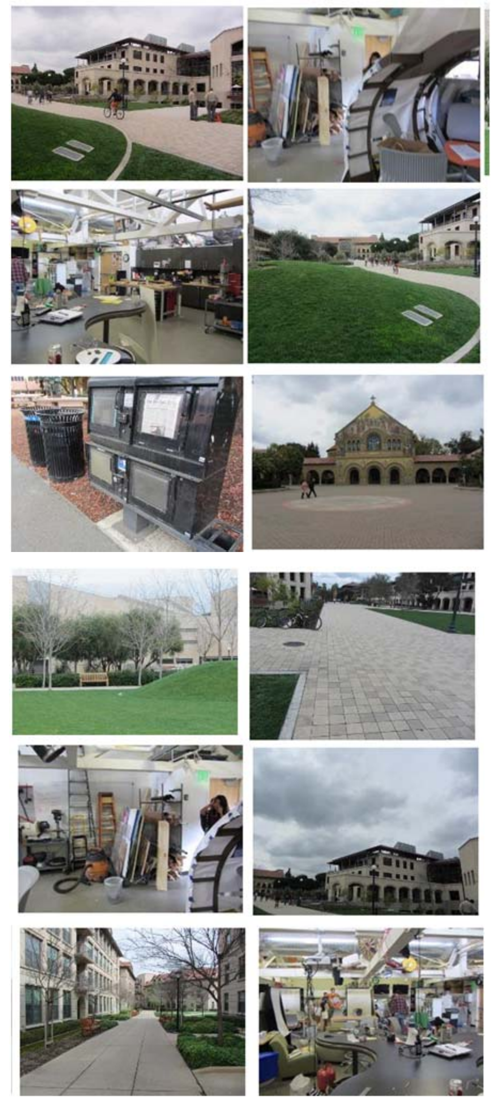

Figure 2: Sample images from the database

B. Technical Approach

This algorithm was developed using MATLAB, a rapid prototyping language which provided a intuitive framework for iterative testing. It also provided a convenient method for debugging problems with the algorithm to make rapid changes.

This approach did have a few drawbacks however. While the design and debugging process were simplified, the overhead of running a program which executes one line at a time rather than compiling the code significantly increased the amount of time the algorithm would take to run. This provided quite a few problems throughout development. In a future implementation this algorithm should be implemented in a lower level compiled algorithm to eliminate this bottleneck.

C. Keypoint Identification

A library within MATLAB was used in order to find SIFT features within each image in the database. This algorithm uses the VLFeat MATLAB library for the keypoint detection and stitching. These keypoints were saved for later reference as this process was time consuming and each set of keypoints needed to be used O(n) times. These keypoints were then compared to find matches between every possible combination of 2 images in order to find matches. This procedure took the longest time, and writing it in C or a similar low level language would provide significant advantages for computation time.

This procedure took O(n^2) time as each image had to be compared with every other image, and the matching was performed by finding the keypoint in image 2 which had the minimum Euclidian distance from each keypoint in image 1. This was repeated for each keypoint in image 1. Also an O(n^2) time operation. This operation provided a reasonable good set of matches between separate images which were then used for stitching.

D. Homography Calculations

After the matches are calculated, the algorithm attempts to form a homography matrix between the two images using 4 of the matching keypoints. This homography is then used to determine which keypoint matches are valid and which ones are not based on the error between the projection of a point in image 1 with its corresponding point in image 2. Using a predetermined threshold, all matches which are acceptable based on that threshold are considered inliers, and all others are outliers. The number of inliers determines the quality of the predicted homography matrix. This can be seen in figure 3.

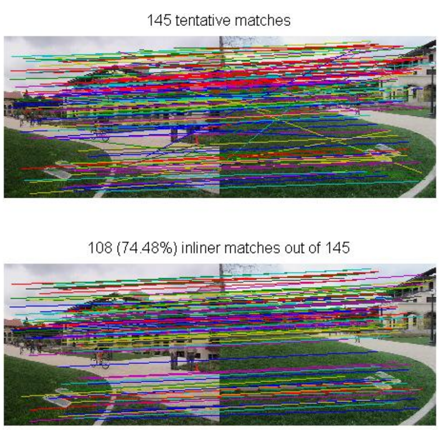

Figure 3: Keypoint matches and inliers after RANSAC

After randomly sampling 4 points 1000 times, computing the homography of each, and calculating the number of inliers each time, we are able to take the homography resulting in the most inliers as our best estimate of the true rotation between images.

E. Clustering algorithm

In order to determine which pictures corresponded to separate panorama's, a method for identifying similar images needed to be used. First, each pair of images is checked to determine the percent of inliers. While the homography calculation determined which points created the best homography, it didn't filter out images which had no good matches, or had a single good homography by chance. Step one, was to look for images which were likely to match by checking the percent of the keypoint matches which were inliers. Any images where more than 50% of the matches were inliers, were considered candidates for stitching together, and marked as such.

Using this information, an adjacency matrix was formed which contained a 1 wherever two images were considered a match, and a 0 whenever fewer than 50% of the keypoint matches were considered inliers. This adjacency matrix was very sparse and needed to be analyzed to determine which images truly belonged to the same panorama.

Using the adjacency matrix, the clusters were divided up by identifying strongly connected components within the adjacency matrix. In order to do this, the first image was used, and all images which it matched to according to the adjacency matrix were added to a stack. This procedure was applied recursively for each image currently in the stack, until all images in the stack had been used.

Through this procedure a single cluster was found corresponding to the images in the panorama which contains the first image. By removing these images and placing them into a new directory, this entire procedure can be recursively applied until no images are left in the image database. This finds all available panoramas, and all extraneous pictures have their own directory by themselves. This method could be extended to placing all panoramas with 1 or very few images into a folder called extraneous so that the extra pictures did not create so many separate directories without changing any of the underling properties of the algorithm based on personal preference.

F. Panorama Stitching

After a directory of all the images in a single panorama has been created, the two images with the largest number of keypoint matches inside this directory are used for stitching. The first step in this process is to calculate the best homography in a similar way to how it was being done before. Using this homography, one image is warped to be in the same frame as the other and a new image of all black pixels is created which can fit both images in the new frame. An example of this can be seen in figure 4.

Figure 4: Sample of two stitched images

Now that the images are aligned properly, they can be combined to form the stitched image. This does pose slightly problems in whitebalance and color saturation since the cameras adjusted based on the light environment in which they were taken. There are also blurring effect where a subject moved between two images.

After the images were combined this process was repeated using the new image, and one of the other images in the database which corresponds well to it. By recursively running this procedure each image is added one at a time to the panorama until all the images have been combined.

In order to create all of the panoramas, this algorithm was run once on each of the directories which had more than 2 images. Directories which only had a single image indicated that only a single image existed, or that multiple images were already combined into a panorama.

IV. RESULTS

The figures below shows the completed panorama stitching with the current algorithm. The database was a set of 80 images containing 8 separate panoramas with 10 extraneous images. The algorithm sorted the images into 18 directories using image matches. Then using the connected components, keypoints, of the images, the panoramas were outputted through stitching together similar images. Figures 5-11 show output panoramas from the algorithm.

This panorama was taken outside of the Huang engineering center at Stanford University. It worked quite well and contained elements of grass, a tiled surface, buildings and sky. As the images were taken at multiple images, a large distortion is evident.

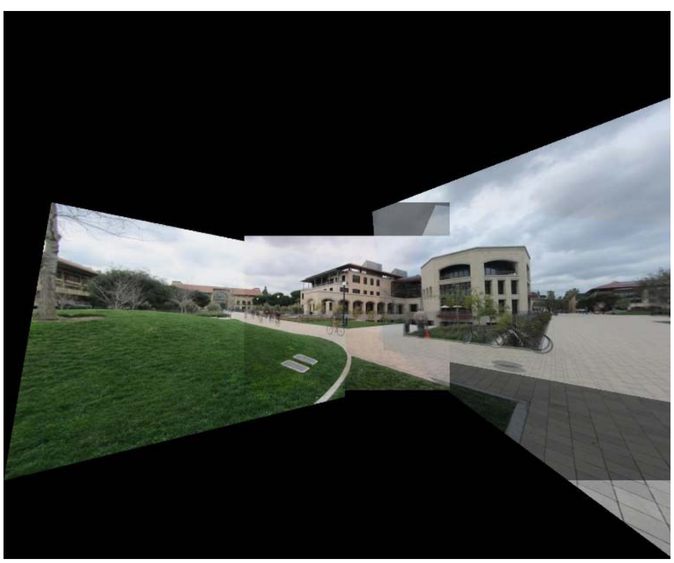

Figure 5: Stanford University Huang Engineering Center

Panorama of the Memorial Church at Stanford University this panorama worked quite well and included significant overlap between images.

Figure 6: Stanford University Memorial Church

This panorama was the only one taken indoors. The large number of objects in a cluttered environment made it difficult to calculate accurate homography matrices. This is most clear by the blurring near the center of images. The stretching in the images farthest right also makes this images look slightly stretched.

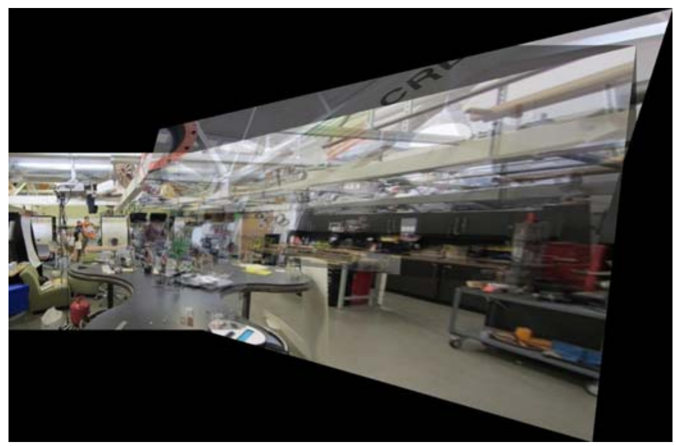

Figure 7: Stanford University ME310 Loft

The images in this panorama were more uniform then many of the others since the water and sky stretched through each. This did not prove a difficulty as the stitching worked quite well.

Figure 8: London, Thames River

Another fairly uniform image which worked quite well. The clustering was quite easy due to the uniformity of the panorama

.

Figure 9: Cannes, Port

This image had a bit more trouble than others due to the motion of people in the frame. The final image looks very stretched as the camera rotated and translated significantly between pictures.

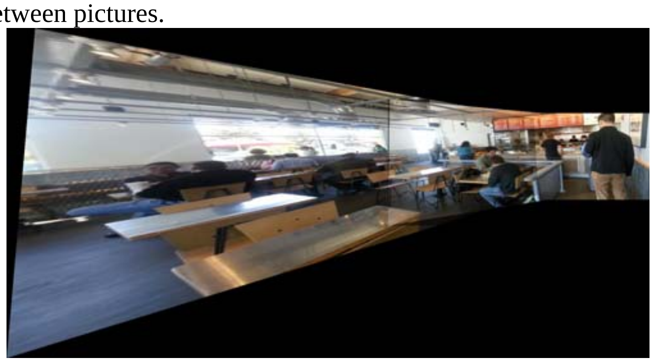

Figure 10: Chipotle

Here is one more example showing a successful cluster and panorama in a relatively easy setting with grass, sky, and distinctive structures.

Figure 11: Bordighera, Italy

The images were captured in various poses and illumination. Even though these are challenging situations for matching image correspondences, SIFT was able to robustly identify keypoints in the images. RANSAC was run 1000 times to find the best homography between two images and the highest number of inliers for both of them. Figure 3 shows potential keypoint matches between the two images. Then after running RANSAC only 74.48% of the keypoint matches were inliers.

There were cases where the algorithm had significantly more trouble. These issues usually arose in the stitching as the clustering was quite robust. The breaking point of the algorithm is when there are many keypoint matches between incorrect points. One such example is in crowds because it has trouble finding a correct homography as the people keep moving around and so keypoints are not consistent.

Another major failing point is when there is little overlap between the photographs as seen in Figure 12. In this case, it's hard for the algorithm to find enough matching points to determine if two images should be in the same panorama.

Figure 12: Cannes

Other failure points include when a person moves greatly between photos as seen in Figure 13. This has the potential for a single homography to be calculated based on the moving person or object rather than the background which causes problems when trying to calculate inliers. The algorithm also fails when the image quality is poor with many occlusions and illumination issues as seen in Figure 14.

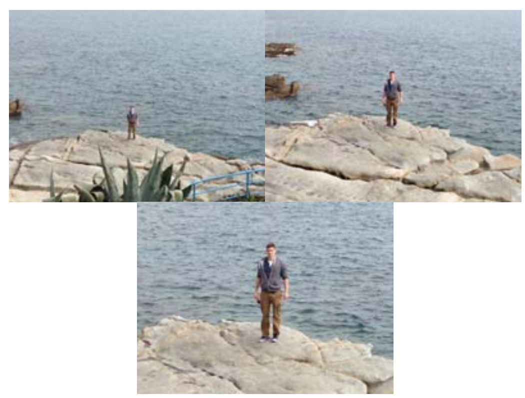

Figure 13: Images with few keypoint matches

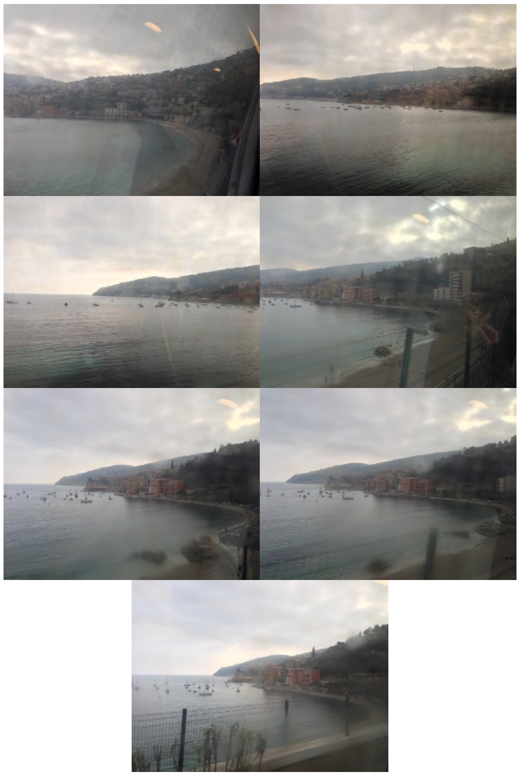

Figure 14: Landscapes with poor image quality

V. Conclusion

In this paper, we created a new framework to automatically sort an image database using clustering for the creation of panoramas. This method works for various scenes including images taken indoors and outdoors. The algorithm uses invariant local features for image matching. The method is robust for camera zoom, orientation of images, and changes in illuminations. The method fails when there are too many keypoint matches in a scene like a crowd of people, many occlusions, or poor image quality.

References

[1] Brown, M. and D. Lowe. *Automatic Panoramic Image Stitching using Invariant Features*. International Journal of Computer Vision. 2007.

[2] He, K., Chang, H, and J. Sun. *Rectangling Panoramic Images via Warping*. ACM Transactions on Graphics – SIGGRAPH 2013 Conference. 2013

[3] Ostiak, P. Implementation of HDR panorama stitching algorithm.

[4] Szeliski, R. *Image Alignment and Stitching: A Tutorial.*Microsoft Research. 2004

[5] Xiong, Y. and K. Pulli. *Fast Panorama Stitching for High-Quality Panoramic Images on Mobile Phones.* IEEE Transactions on Consumer Electronics. 2010.

[6] Xiong, Y. and K. Pulli.*Sequential Image Stitching for Mobile Panorams.* Seventh IEEE International Conference on Information, Communications, and Signal Processing. 2009.