Mcgill University

Panoramic Image Stitching

by

Kai Wang

Pengbo Li

A report submitted in fulfillment for the
COMP 558 Final project

in the
Faculty of Computer Science

April 2013

Mcgill University

Abstract

Faculty of Computer Science

by Kai Wang
Pengbo Li

Panoramic image stitching is an active research topic and lots of methods was developed to achieve a nice stitched panorama under different limitation of raw pictures. For example, pictures need to be aligned properly, pictures need to has same exposure level, or pictures need to be horizontal or vertical in space. In this project, we followed the pipeline proposed by Matthew Brown and David G. Lowe[1] to implement construct a feature based method using SIFT feature, which is much less sensitive to rotation and scale. We will also implement two methods of blending methods: Laplacian pyramid method and Poisson blending to deal with the seam problem due to exposure difference. Compared with the method without blending technique, blending stitching images apparently shows better performance on the pictures with different exposure level than version without blending. Last but not least, we will provide a set of pictures with various transform, various exposure level, and try to create a nice panoramic image.

Keywords: panorama, invariant feature, image alignment, Laplacian pyramid, Poisson equation,exposure compensation

Contents

Chapter 1

Introduction

1.1 Panorama

A panorama is an wide-angle view or representation of a physical space, whether in painting, drawing, photography, film/video, or a three dimensional model. It is interesting to gain large perspective of the environment with a single image. It is easy to produce a panorama in painting , but it is relatively difficult to produce one in photography due to the limitation of field of view. We know that general camera that can only produce a picture with viewing angle of 60-90 degree. Panoramic photography soon came to create wide views for casual user, that is stitching multiple pictures together from different perspectives. Usually, panorama stitching result always undertakes mis-alignment due to randomness ,seam due to exposure difference, ghost due to moving object and other issues. Thus this topic is very attractive and challenging. The major goal of our project is to build a panorama stitching program with blending steps to generate a nice-looking panoramic pictures.

Our implementation consists of two major steps: alignment and blending. In alignment step, geometrical properties will be analyzed and compare to produce a correctly positioned picture. and In blending step, the exposure level is merged nicely together to produce a uniform exposure level across the overlap region of different pictures.

1.2 Alignment step

In the alignment step, we will extract the SIFT features in each input images. Those features will be matched and use RANSAC iteration to estimate homography.Then we choose one image as reference to align the other one according to homography to generate coarse stitching images and useful masks information for blending step.

1.3 Blending step

In the blending step,we blend aligned images with two different methods to produce a nicely merged image. Laplacian pyramid blending generate a band-pass pyramid for each images and combine them together. Poisson blending works in the gradient domain to solve Poisson equation with the Dirichlet conditions to generate seamless images.

Chapter 2

Alignment step

2.1 Feature Extraction

Our implementation will use be scale-invariant feature transform(SIFT) [2]. This algorithm can detect and describe local features in images. The SIFT enables reliable image matching with various orientation and zoom.The basic steps of extraction algorithm are Scale-space extrema detection, keypoint localization, orientation assignment and keypoint descriptor.

2.1.1 Scale-space extrema detection

In this step, keypoints will be determined suing a cascade filtering approach. At each scale, the difference of Gaussian function will be calculated with following formula:

$$D(x, y, \sigma) = (G(x, y, k\sigma) - G(x, y, \sigma)) * I(x, y) \tag{2.1}$$

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-(x^2+y^2)/2\sigma^2} \tag{2.2}$$

In this formula, $I(x, y)$ is the input image, and $G(x, y, \sigma)$ is the variable-scale 3 Gaussian. the maxima and minima of $D(x, y, \sigma)$ will be computed across multiple layers. Those maxima and minima will be the features on this image.

2.1.2 Keypoint localization

The feature points will then be filtered. There are two steps for filtering the feature points. First feature points with low contrast level will be filtered out with a threshold on minimum contrast. Then all the feature points will compare to surrounding feature point to remove edges produced by the noise.

2.1.3 Orientation assignment

In this step, orientation will be detected on all the feature points, and will be saved in the keypoint descriptor. The keypoint orientation will be calculated with the presence of surround 4 feature points. Those orientation will be useful when the target image has a rotation compare to the reference image.

2.1.4 Keypoint descriptor

In this step, all the useful information about the feature will be saved with corresponding points, including image location, scale, orientation of those keypoints. Those information will be enhanced locally to facilitate the comparison and to remove the in uence of noise.

2.2 Feature matching

After finding all the feature points in both images, those feature points need to match to each other. a data structure called k-d tree can be used for this step. This is a space-partitional data structure good for organizing points in a high dimensional space.For multiple pictures to align, this data structure will be easier to understand and implement. In each matching step, left and right child of current node will be a new k-d tree, and the location of current node will be the median of the keypoint's location.

With this data structure, matched keypoints will be easier to save and to retrieve, because the same feature point in different pictures will be at the same location in this structure.

2.3 Image matching

After matching feature points in input pictures, we will try to estimate homography using random sample consensus(RANSAC), which is an iterative method to estimate parameters of a mathematical model from a set of observed data which contains outlier. In our case, those observed data will be the feature points locations and the outliers will be those feature points which cannot be matched to another picture. RANSAC is a robust approach to find reasonable result after certain trials. With the alignment information, we choose one image as reference, then transform other image into local frame using homography and ready for blending step.

Figure 2.1: Input image 1

Figure 2.2: Input image 2

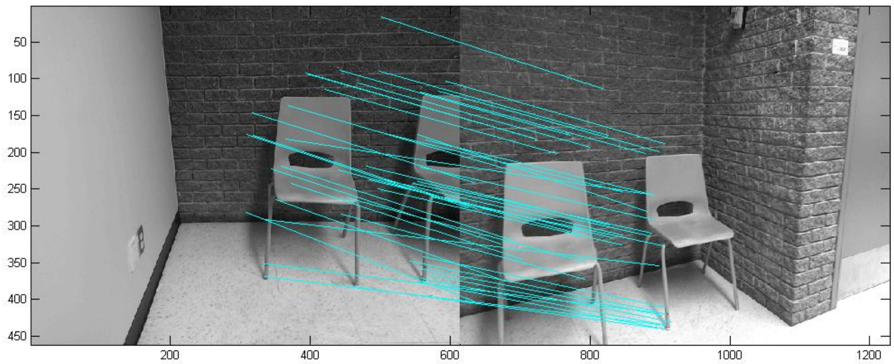

Figure 2.3: Result of SIFT feature finding and matching

Chapter 3

Pyramid Blending

3.1 Introduction

We first use pyramid blending to blend the images together[3]. The principle of this algorithm is to build Laplacian pyramid from input images. Each layer in pyramid will have half of the resolution of the previous layer then the pyramid will be merged at different layer separately. Finally the pyramid of the merged image will be collapsed into the final blended image.

3.2 Pyramid construction

Depending on how smooth the blending need to be, a specific number of layer of pyramid need to choose. The maximum number of layer is determine according to the size of input picture. The size of input picture need to be smaller than the 2^l pixel on the shorter side. At each level, the 4 pixel from previous level will be collapse into one single pixel and save at the location. So the picture size of level above will be half of the current level.

3.3 Mask computation

A mask of blending need to calculated from input image. Here I calculate separation line for the mask at the middle where the two image joined. In this way, during the merging step of pyramids from input images, those pixel at the edge of the picture will not be consider. The reason is that during the pyramid construction, pixels at the edge of picture will consider the area outside of the picture as part of the picture. And because pixel outside of the picture has 0 for its pixel value, which is black for color, the resulting blending result will be a darker area at the blending region.

3.4 Pyramid blending

We loop through all the level of pyramid, and blend the image using the re-sized mask at each level and construct a pyramid with combined pixel information at different level.

3.5 Final picture reconstruction

The resulting pyramid is then collapsed back to one image.

3.6 Result

After implementing the method and various test on the method, we have several interesting discovers:

FIGURE 3.1: Result after Pyramid blending with 8 levels

3.6.1 Pyramid layer

We tried blending with different pyramid layer number. Because our test image size is 577418, so the maximum number of layer is 8. When level is 1, the resulting picture has a sharp edge where the image was blending. when the level increase, the edge become smoother and smoother. When level number increase to 8, the edge is barely noticeable. However, for experimenting, I keep increase the level number. As soon as the level number is bigger than the maximum level allow, the lose of picture information is noticed, and it appear as the bottom part of the input image is not present in the resulting picture. This part increases in size as the level size keep increase until the entire result picture disappears.

3.6.2 Edge artifact

I've also notice an increasing size of artifact at the edge of picture. Because the input pictures was constructed as the original picture at its correct final position with black information around it. During the pyramid construction step, the black pixel information will be calculated into higher level pyramids. During the final picture reconstruction step, those information will be kept and result as a blurred edge.

Chapter 4

Poisson Blending

4.1 Introduction

Gradient domain processing has wide applications including blending, tone-mapping, and non-photorealistic rendering. The insight here is that people is often more sensitive to the gradient of an image than the overall intensity.Thus gradient methods can often make surprising performance.

Perez's work[4] shows how gradient reconstruction can be applied to do seamless object insertion. They do computation on the gradient domain of source image, rather than copy intensity of original pixels. For the corresponded area in the target image, the pixel intensity is reconstructed by solving a Poisson Equation that locally align the gradients while obeying the Dirichlet conditions at the seam boundary. Here the target pixels that maximally preserve the gradient of the source region without changing the background pixel. In our project, we extend this approach to stitching image blending, to solve the discrete Poisson equation to determine the final pixel value of overlapped part where always generates seam due to exposure difference and ghost due to moving object, which turned out to be very acceptable.

4.2 Implementation

Our objective can be formulated to a least square problem. Given the intensities of the source image "s" and the target image "t". In order to largely preserve property of source image, while nicely combine two pictures, we can solve for new intensity values "v" within the source region S, then we obtain the formation:

$$v = \sum_{i \in S, j \in N_i \cap S} ((v_i - v_j) - s_i - s_j))^2 + \sum_{i \in S, j \in N_i \cap -S} ((v_i - t_j) - (s_i - s_j))^2 \tag{4.1}$$

Where each "i" pixel in the source region S, and each "j" is 4-neighbor of pixel "i". Each summation guides the gradient values to match those of source region. In the first part, the gradient is over two neighbour variable pixels, and in the second part, one pixel is variable and one is in the fixed target region. We need to transform the least square constraints to a matrix form as $(Av - b)^2 = 0$ and our task becomes construct matrix a sparse matrix A and vector b, then solve for linear equation $Av = b$. Finally, the source region in the target image can be reconstruct d using solved vector v .

In our case, blending step is applied after warp images is constructed. We need to construct the source region mask according to overlap region of two images. Finally choose one image as source image and the other is target image. Then we step into most time consuming part of this implementation to construct sparse matrix A and vector b, because Matlab is very inefficient for loop, which can be speed up using Mex.file. Considering due time, I guess we can do test in the future work. For sparse matrix solving part, here is a tip is to initialize your sparse matrix with sparse() function, leading to ignore zero element while computing. To sum up, Poisson blending shows good performance to smooth the seem between two images.

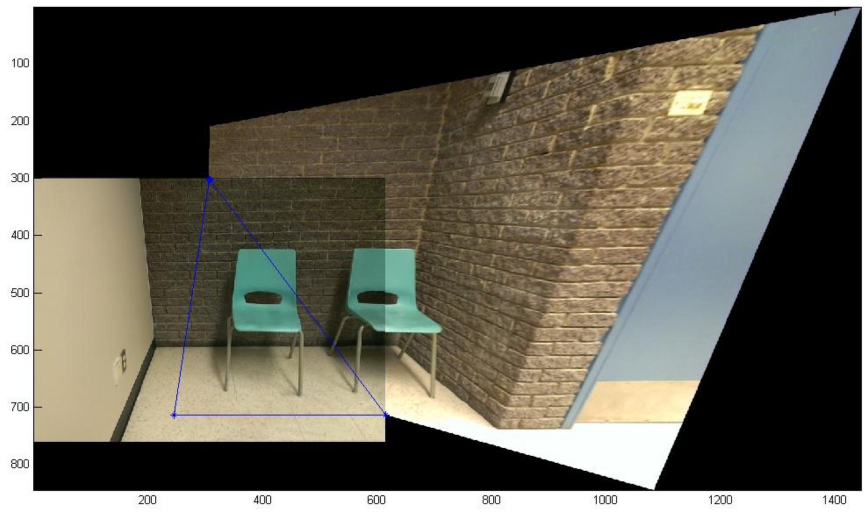

Figure 4.1: Poisson blending : source image and corresponded source region

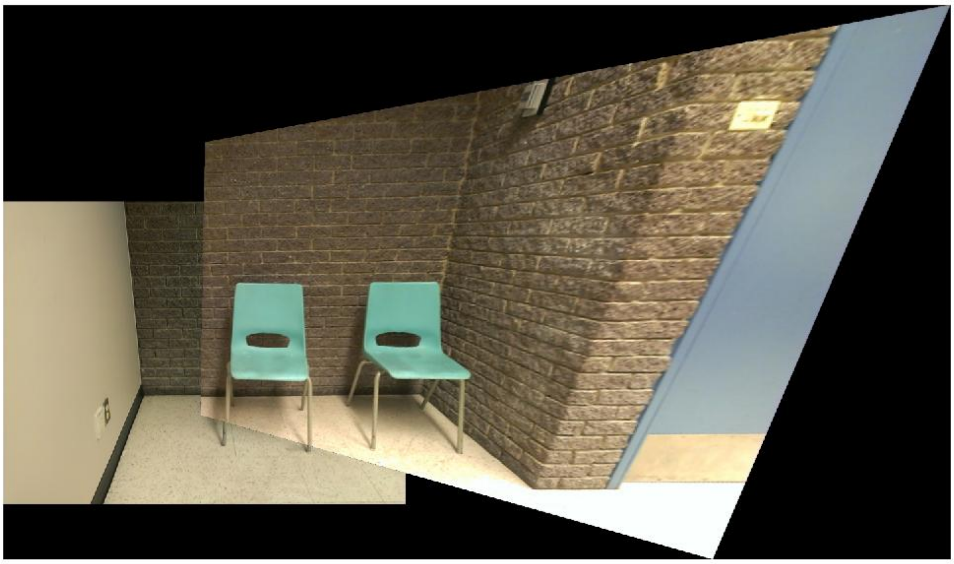

Figure 4.2: Poisson blending : target image

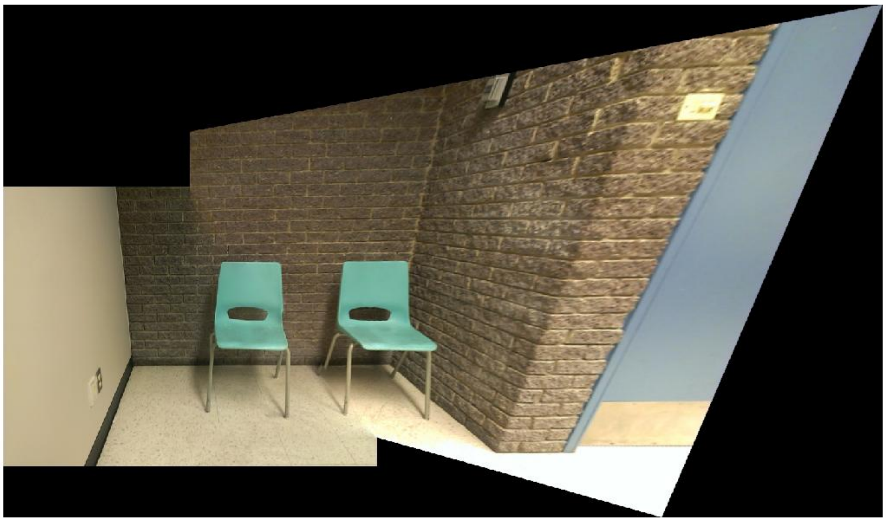

Figure 4.3: Poisson blending result

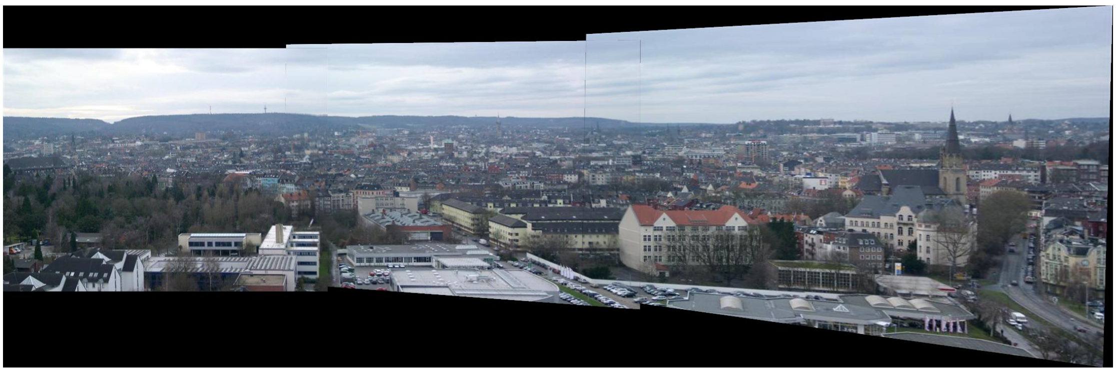

Figure 4.4: Test panorama without blending

FIGURE 4.5: Test panorama after blending

FIGURE 4.6: Roddick Gates, 6:00AM , April 26th, 2013

Chapter 5

Conclusion

As an extensive research topic, panorama image stitching have been developed to many engineers. In this project, we implemented a relatively completed panorama stitching system and generally explored the many aspects, including feature extraction and matching, homogaphy estimation and image blending. Comparing with two blending method, pyramid blending is relatively faster and show better performance on the whole picture exposure compensation, but it has edge artifacts with higher level. And Poisson blending can accurately remove the seam on the image, but could not globally compensate exposure, which can be seen in the figures of blending result.

Our algorithm of SIFT and RANSC was implemented based on existing works. We work together finish alignment step and the final report. Kai implement the part of Laplacian pyramid blending, and Pengbo take part in the Poisson blending. We do believe this is impressive corporation.

In this project, our job more focused on the blending part and do not take much effort to solve registration error like bundle adjustment, multiple warp option and more hybrid blending method.Hopefully, we can build up a perfect auto panoramic image stitching system in the future.

Bibliography

[1] Matthew Brown and David G Lowe. Automatic panoramic image stitching using invariant features. *International Journal of Computer Vision*, 74(1):59–73, 2007.

[2] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004.

[3] C. Allene, J.-P. Pons, and R. Keriven. Seamless image-based texture atlases using multi-band blending. In *Pattern Recognition, 2008. ICPR 2008. 19th International Conference on*, pages 1–4, 2008. doi: 10.1109/ICPR.2008.4761913.

[4] Patrick Pérez, Michel Gangnet, and Andrew Blake. Poisson image editing. *ACM Transactions on Graphics (TOG)*, 22(3):313–318, 2003.