Chapter 5
Image Stitching

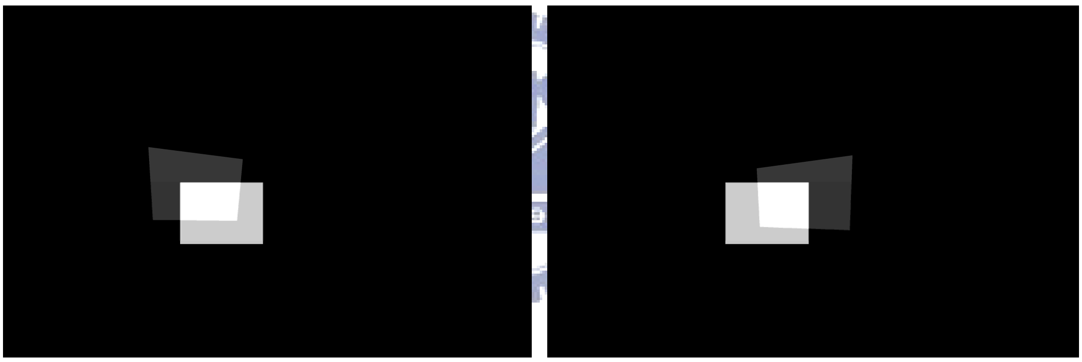
5.1 Introduction

From the previous chapters, the relationship between source and target images can be found and the source image can be mapped to match up the target image. However, no matter how similar the overlap regions are in the mapped source image and target image, it is unavoidable that some artifacts such as color difference and position mismatch still exist between the overlap regions of these two images. Thus, an image stitching technique is required to process the overlap region and reduce artifacts as possible while combining two images. In this chapter, the preprocessing technique for the image stitching process to save the calculation time called rectangular boundary determination is proposed and the previous researches of image stitching techniques are introduced.

The boundary determination technique is used to determine the rectangular boundaries of the mapped source image, target image and overlap region by establishing the rectangular boundaries of the source region, the target region and the overlap region from the combined region in V_{cb}. After the rectangular boundary determination, two previous researches of image stitching techniques called minimum error boundary cut [3] and optimal partition [9] are introduced to divide the overlap region into two parts for the mapped source image and the target image respectively by creating a low artifact cut line as a border to obtain the combined image.

5.2 Rectangular Boundary Determination

From Section 4.4, the rectangular canvases V_{cs}, V_{ct} and V_{cb} have been created to respectively place the mapped source image, target image and combined region as shown in Figure 4.1. In order to achieve the goal of stitching numbers of images from arbitrary directions, the size of these canvases should be the same and large enough. For example, the size of all the canvases in this thesis is set to be 2000×3000. However, the combined region needing to be processed is just a small part of V_{cb} at the beginning, as shown in Figure 5.1. Hence, the rectangular boundary determination is a useful technique to extract the rectangular boundaries for the mapped source image, target image and the overlap region such that the process on the whole canvas V_{cb} is not needed.



(a) (b)

Fig 5.1 The combined region with different gray-level values in V_{cb}.

The pixels of the combined region on V_{cb} have three pixel values, 255, 55, and 200, corresponding to the overlap region, the source region, and the target region respectively. Since the overlap region is treated as part of the source region and part of the target region, the pixels with pixel values 255 or 55 form the source region and the pixels with pixel values 255 or 200 form the target region. In order to determine the rectangular boundaries for these regions, it is required to find out the edges of each region and then the four corners. Based on the corners, a set of rectangular boundaries is resulted for each region. Figure 5.2 shows the flowchart of the rectangular boundaries determination technique for the source region following the rules stated above. The canvas $\boldsymbol{V}_{cb}$ of size 2000×3000 is the input to be scanned and the scanning direction expressed by *VScan* is initially set to be 0 for horizontal scanning to determine the edges whose pixel values change from 255 or 55 to another, and then extract the corners in the vertical direction of the source region. After horizontal scanning, *VScan* is changed to be 1 for vertical scanning on the $\boldsymbol{V}_{cb}$ to find the edges of the source region and extreme corners in the horizontal direction. Similarly, the extreme corners in the vertical and horizontal directions of the target region can be obtained by determining the edges as the change of pixel values from 255 or 200 to another of $\boldsymbol{V}_{cb}$, and those of the overlap region can be obtained by determining the edges as the change of pixel values from 255 to another. As shown in Figure 5.3, the red lines in the left, middle and right images represent the rectangular boundaries of the source regions, the target regions and the overlap regions from the combined region on canvas $\boldsymbol{V}_{cb}$ in Figure 5.1 which is obtained according to the four extreme corners of each region.

In this section, the rectangular boundary determination technique decides the boundaries of the source region, the target region, and the overlap region from combined region in $\boldsymbol{V}_{cb}$ according to their different decision pixel values. After this preprocessing step, image stitching techniques proposed in the following sections can easily choose the region they want to process from $\boldsymbol{V}_{cs}$ and $\boldsymbol{V}_{ct}$ by these boundary information and save a lot of calculation time.

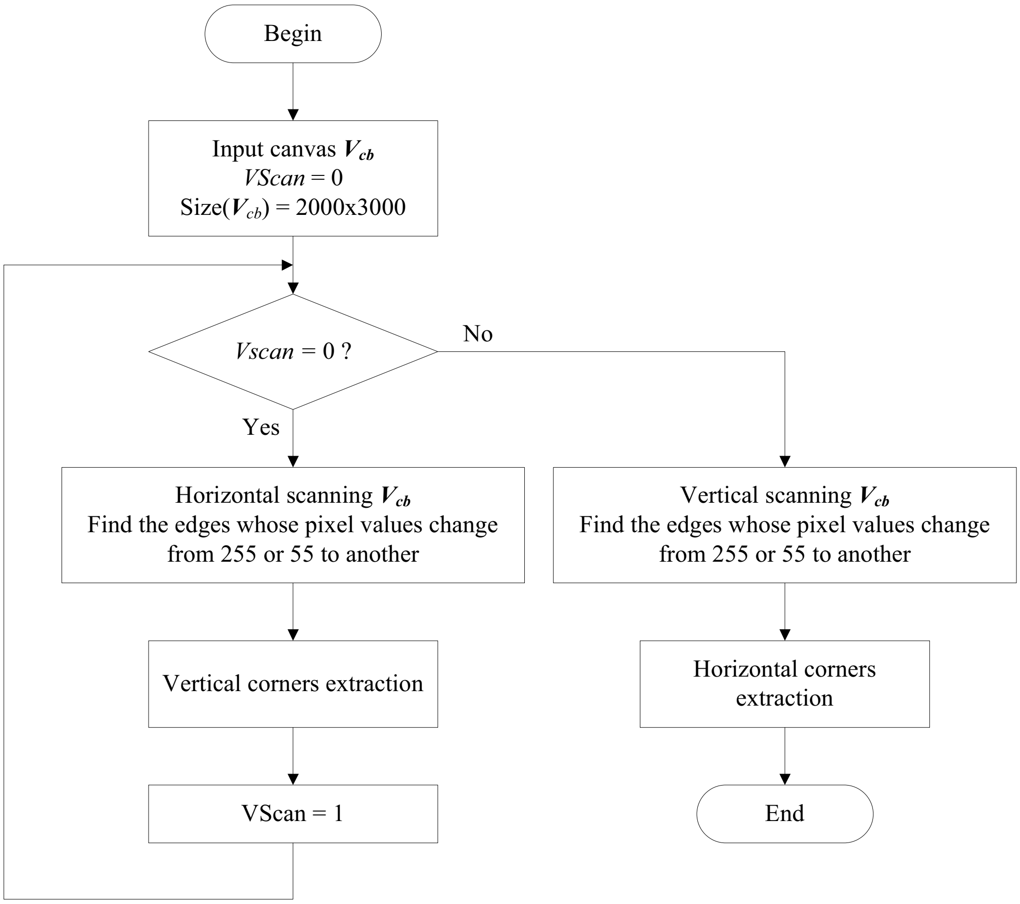
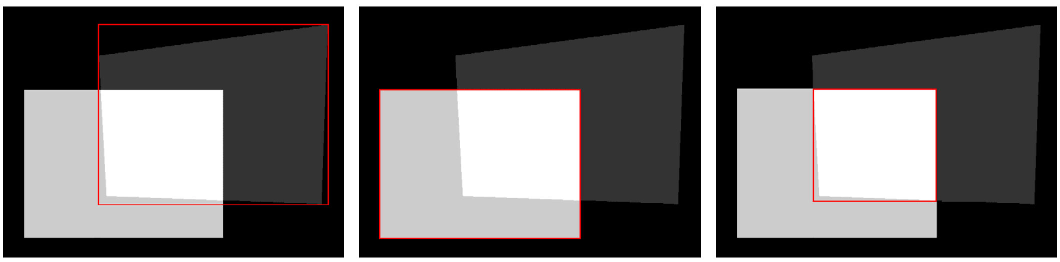
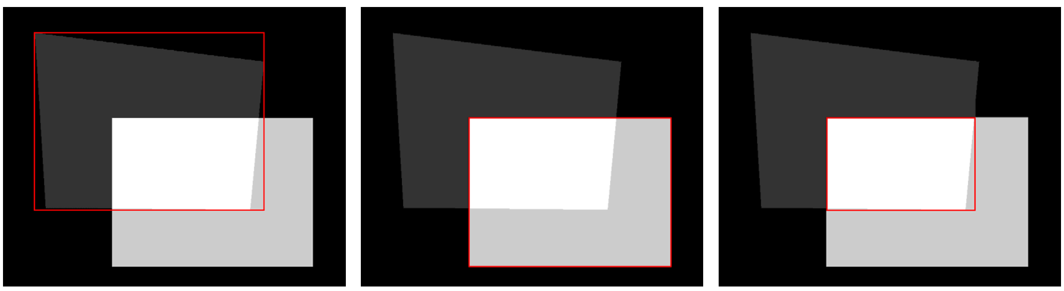


Fig 5.2 Flowchart of the rectangular boundary determination for the source region



(a)



(b)

Fig 5.3 The boundary determination of V_{cb} (a) for Fig 5.1(a), and (b) for Fig 5.1(b)

5.3 Previous Researches for Image Stitching

After boundary determination, it is required to choose an appropriate image stitching technique to combine two images together according to the relationship between them. Because the shape of the mapped source image is no longer in a rectangular form and change its original shape to match up the target image, as shown in Figure 4.2 and Figure 4.3. Therefore, no matter how similar the overlap regions of the two images are based on image mapping algorithm, the deformation of the same objects in each overlap region still exists. The traditional stitching methods such as taking average for all the pixels in overlap region or directly separate it into source and target image respectively with a straight cut line is unsuitable in general case of image stitching process.

5.3.1 Minimum Error Boundary Cut

Recently, Efros and Freeman have employed a useful stitching technique, minimum error boundary cut [3], to generate a synthesized image by stitching small patches of an existing image together. In Figure 5.4, the authors randomly choose many small patches from the input texture as quilting input with fixed size of rectangular overlap region. In order to find the minimal vertical cut through the overlap region, the error functions used to compute the cumulative minimum errors $E(j)$ for all paths is given as

$$\begin{cases} E^c_{1,j} = e_{1,j} \\ E^c_{i,j} = E^c_{i-1,j} + \min\left(e_{i,j-1}, e_{i,j}, e_{i,j+1}\right), & i = 2,3,\cdots,N. \end{cases} \quad (5.3\text{-}1)$$

where $e_{i,j}$ is the color difference between source and target images at the coordinate (i, j) on their overlap regions. Then, the cumulative minimum errors $E(j)$ are determined as

$$E(j) = E^c_{N,j} \quad (5.3\text{-}2)$$

Furthermore, find the color difference of the three candidate points in the next row and choose the one with the minimum color difference as a part of the cut line, and so on. After the cumulative minimum errors for all paths are found, the path with minimal cumulative error can be obtained as the boundary cut line to divide the overlap region into source and target image parts, as shown in Figure 5.4(c).

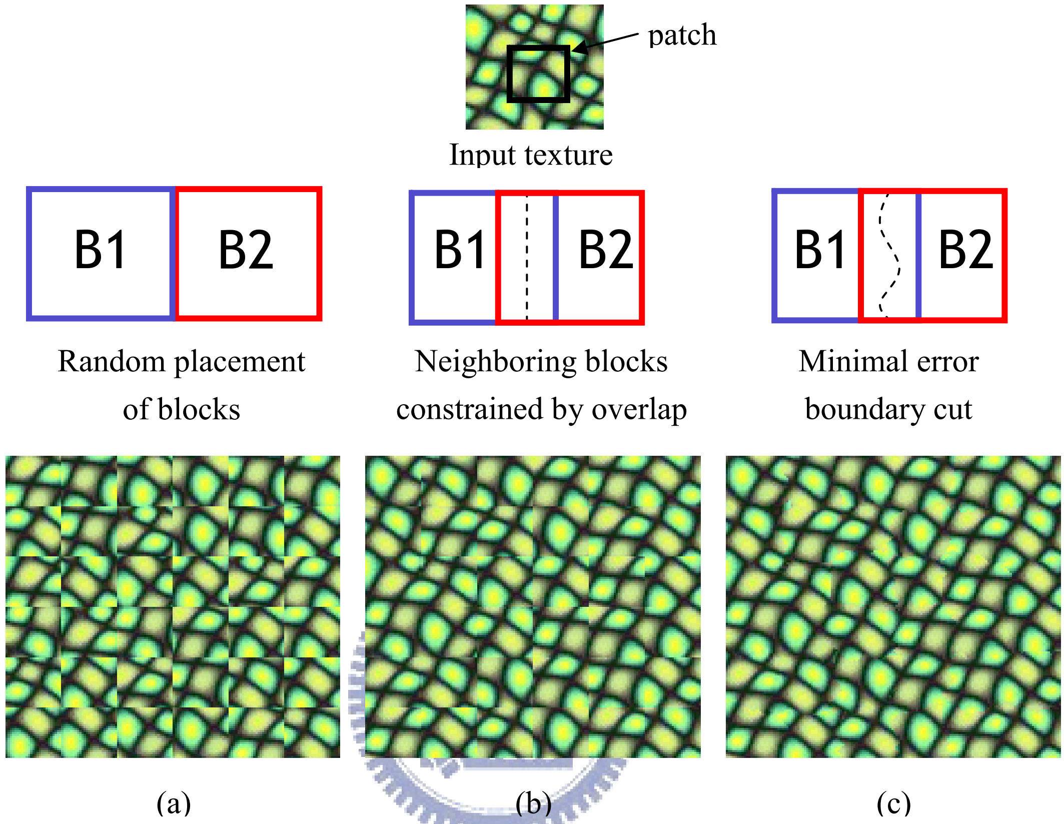


Fig 5.4 (a) Random placement of blocks, (b) Neighboring blocks constrained by overlap, and (c) image quilting based on minimum error boundary cut

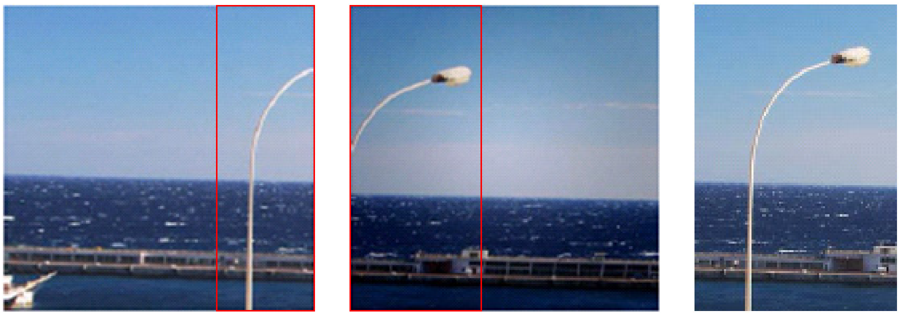
5.3.2 Optimal Partition

Recently, Jia and Tang adopted the so-called optimal partition [9] to deal with the division of overlap region, similar to but much better than the technique mentioned in Section 5.3.1. With the information of color difference and the image smoothness, they defined the gradient alignment cost $S(p,q)$ between any adjacent pixels p and q for the *RGB* color channels as

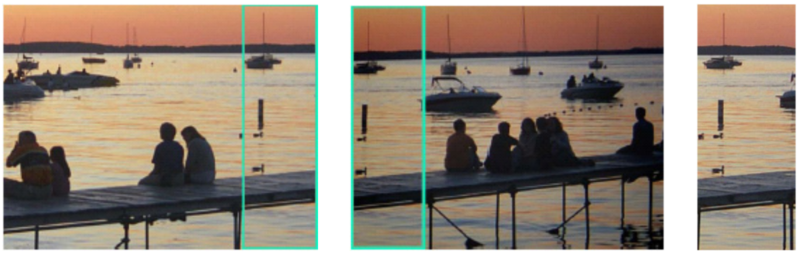
$$S(p,q) = \sum_{r,g,b} \left(\beta \cdot S_m + (1-\beta) \cdot S_d\right) \qquad (5.3\text{-}3)$$

where S_m and S_d are two costs respectively measuring the gradient smoothness by Sobel edge detection and similarity by color difference of all the neighboring pixels. Furthermore, β is a weight used to balance the relative influence of the two costs and is set to be larger than 0.5 to enhance the weighting of S_m. Afterward, the cumulative errors of all paths are calculated based on the error function (5.3-1) and then choose the one with minimal cumulative error as the cut line to divide the overlap region into two parts for source and target images like Figure 5.4(c). Finally, the feature extraction and feature matching methods are employed along the cut line and then use the deformation propagation to smoothly change the shape of images such that the matching pairs are matched along the cut line. Figure 5.5 shows the stitching results of two images with the overlap regions and Figure 5.6 shows the result of stitching algorithm applied to object insertion.

Although the above stitching methods can successfully separate the overlap region, some discontinuous edges still exist and are needed to be compensated. Next chapter will propose a novel image stitching technique to improve the defects in the previous researches for image stitching techniques with image artifacts almost invisible and then get a higher resolution combined image with good performance.



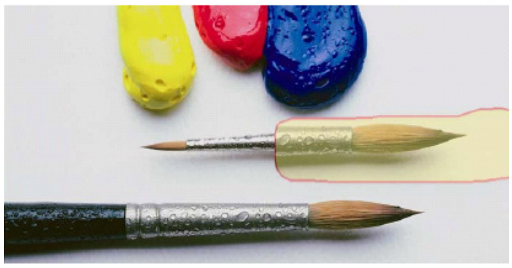
(a)



(b)

Fig 5.5 Source images (left images), target images(middle images), and the stitching results in overlap regions (right images)

(a)



(b)

(c)

Fig 5.6 (a) The original image, (b) the image with object insertion, and (c) the stitching results in the inserted region