

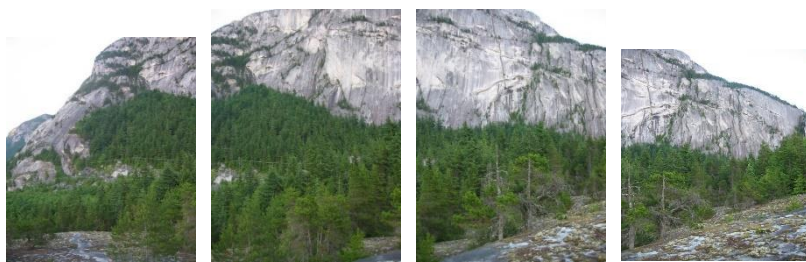
Project Report of Image Mosaic

Author: 赵文浩 23020201153860 (计算机科学系)

I. Task of Image Mosaic

1.1 What is Image Mosaic?

图像拼接是指将多幅图像拼接在一起,形成一张完整的全景图(Panorama)。拼接任务的难点在于待处理的图像可能具有不同的方向(Direction/Angle)和尺度(Scale)度,重点在于如何图像之间的方向和尺度差异[1]。以图 1-1 (a)展示的待拼接图像为例, Image Mosaic 的目标是得到图 1-1 (b)所示的全景图。



(a) Origin Images



(b) Panorama Image

图 1-1 Image Mosaic

1.2 The Principle of Image Mosaic

对于两张图片的拼接任务,要得到较好的拼接结果,首先需要以其中一张图片为标准模板,将另外一张图片映射到该图片上,使得相同的图像内容重合,因此拼接任务具体可划分为投影映射(Projection)和图像缝合(Stitching)两个基本步骤,其中 Projection 可以通过求解单应性矩阵(Homography Matrix)实现。

1.2.1 Homography Matrix

单应性矩阵包含图像投影变换的参数,说明了一张图片的像素点坐标如何变换到另一张图片对应的位置上。对于三维图像,利用 Homography Matrix 做投影

变换的过程如公式 1-1 所示。

$$\begin{bmatrix} x_2 \\ y_2 \\ z_2 \end{bmatrix} = \begin{bmatrix} H_{11} & H_{12} & H_{13} \\ H_{21} & H_{22} & H_{23} \\ H_{31} & H_{32} & H_{33} \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix} \Leftrightarrow X_2 = HX_1 \quad (1-1)$$

在非齐次坐标下，投影后的坐标可表示为：

$$x_2' = \frac{H_{11}x_1 + H_{12}y_1 + H_{13}z_1}{H_{31}x_1 + H_{32}y_1 + H_{33}z_1}, \quad y_2' = \frac{H_{21}x_1 + H_{22}y_1 + H_{23}z_1}{H_{31}x_1 + H_{32}y_1 + H_{33}z_1}$$

对于二维平面，我们将 z 轴坐标设置为 1 ($z_1 = 1$)，重新排列上式得到：

$$x_2' (H_{31}x_1 + H_{32}y_1 + H_{33}) = H_{11}x_1 + H_{12}y_1 + H_{13}$$

$$y_2' (H_{31}x_1 + H_{32}y_1 + H_{33}) = H_{21}x_1 + H_{22}y_1 + H_{23}$$

这里不介绍单应性矩阵 H 的求解，但从以上变换过程能够发现，单应性矩阵元素 H_{13} 、 H_{23} 分别代表图像在水平和竖直方向上的平移变换，其他元素的组合形成了图像在尺度和方向上的变换。

1.2.2 Coordinate Conversion

得到单应性矩阵 H 后，即可将 image1 的各个像素点坐标转换到 image2 的对应位置，坐标变换的过程如公式 1-2 所示。需要注意的是，待处理的图像属于二维平面，因此变换后的坐标需要将 z 轴归一化为 1。

$$\begin{bmatrix} x_2 \\ y_2 \\ 1 \end{bmatrix} = \begin{bmatrix} H_{11} & H_{12} & H_{13} \\ H_{21} & H_{22} & H_{23} \\ H_{31} & H_{32} & H_{33} \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix} \Leftrightarrow X_2 = HX_1 \quad (1-2)$$

$$x_2' = \frac{H_{11}x_1 + H_{12}y_1 + H_{13}}{H_{31}x_1 + H_{32}y_1 + H_{33}}, \quad y_2' = \frac{H_{21}x_1 + H_{22}y_1 + H_{23}}{H_{31}x_1 + H_{32}y_1 + H_{33}}$$

II. Process of Image Mosaic

这里以两张图像的拼接任务为例，说明图像拼接的过程，需要说明的是，缝合的图像需要进行仿射变换，而被缝合的图像保持不变。假设需要将图像 A 缝合到图像 B 上，拼接过程如下。

Step1. 利用 SIFT/SURF/ORB 等方法提取图像 A 和 B 中具有判别性的关键点；

Step2. 在两张图像的关键点之间找寻匹配良好的关键点对 (Keypoint-Pairs)；

Step3. 基于 RANSAC 等配准算法计算点对之间的单应性矩阵 H ；

Step4. 利用单应性矩阵 H 将图像 A 投影到图像 B 上，投影后的图像记为 A^P ;

Step5. 组合图像 A^P 和 B，得到全景图 $\text{Panorama}=(A^P,B)$ 。

III. Several Tips for Image Mosaic

3.1 Projection Order

投影顺序对于图像拼接的结果至关重要，它定义了需要对哪张图片进行仿射变换。这里假设给定的原始图像序列是连续的，且左图的右侧与右图则左侧是重叠区域，如图 3-1 所示。

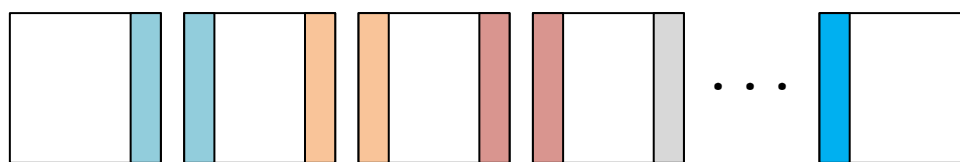


图 3-1 Origin Images

基于以上假设，对于两张图片的拼接任务，有以下两种投影顺序可供选择。

3.1.1 Left to Right

“Left to Right”的投影顺序代表保持右图形态不变，将左图进行仿射变换，投影到右图对应的位置，这种方式下得到的全景图的右半部分是正常非扭曲的。以图 3-2 (a)所示的两张图片为例，该操作下得到的全景图如图 3-2 (b)所示。



(a) Origin Images

(b) Panorama Image

图 3-2 Left to Right

3.1.2 Right to Left

类似地，“Right to Left”代表保持左图形态不变，将右图进行仿射变换，投影到左图对应的位置，这种方式下得到的全景图的左半部分是正常非扭曲的。以图 3-3 (a)所示的两张图片为例，该操作下得到的全景图如图 3-3 (b)所示。



图 3-2 Right to Left

3.1.3 Different Ways of Stitching

从上面的描述中可以看出，投影的顺序对拼接的结果有较大的影响。基于以上两种 **Projection Order**，这里提出以下三种图像缝合的方式。

A. Forward Stitching

前向缝合是指在拼接的过程中，依次执行 **Left to Right** 的投影顺序，即将左侧图像投影到右侧图像上，将拼接得到的图像再次投影到下一张图像上，这种拼接方式适用于仅进行平移变换的图像。

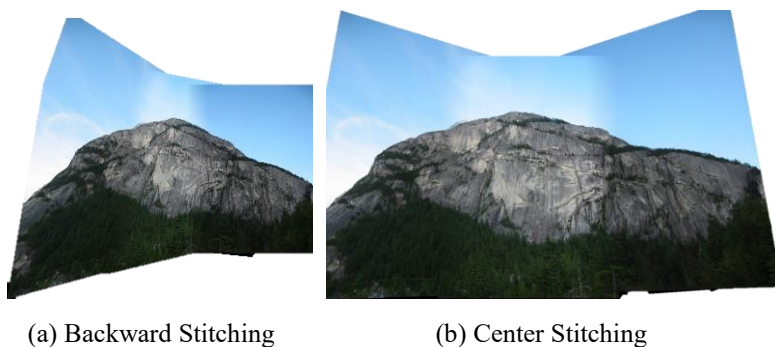
B. Backward Stitching

与前向缝合相反，**Backward Stitching** 指在拼接的过程中，依次执行 **Right to Left** 的投影顺序，即将右侧图像投影到左侧图像上，将下一张图像投影到拼接得到的图像上，这种方式同样对于仅进行平移变换的图像有较好的拼接效果。

C. Center Stitching

Center Stitching 是两种缝合方式的结合。假设待拼接图像数量为 N ，对于前 $N/2$ 张图片，采用 **Forward Stitching**，对于之后的图片，采用 **Backward Stitching**。

以上三种缝合方式对于不同类型的待拼接图像会产生不同的拼接效果，且最优的拼接效果对应的缝合方式也存在差异。图 3-3 展示了采用以上三种缝合方式的拼接结果。





(c) Backward Stitching

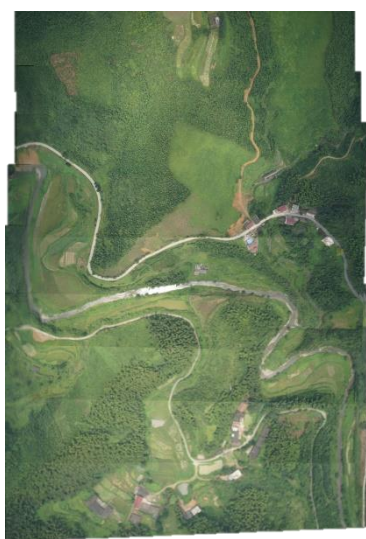
图 3-3 Mosaic Result in Different Stitching Ways

3.2 Orientation Adjustment

图像拼接的结果在一定程度上也受原始图像方向的影响，本次课程设计主要针对水平方向重合的图片集（如图 3-1 所示）。若给定图片集的重合区域处于图片的下方和上方，则需要先对原始图像进行旋转操作，从而得到更好的拼接效果。对于图 3-4 (a)所示的待拼接图像，直接采用 Forward 缝合的方式得到的拼接结果如图 3-4 (b)所示；先对图像进行旋转操作后得到的拼接结果如图 3-4 (c)所示。



(a) Origin Images



(b) No Rotation



(c) With Rotation

图 3-4 Mosaic Result on Different Orientations

从图 3-4 中可以看出，未经旋转直接拼接得到的全景图存在横向衔接缝，虽然衔接缝可以通过融合算法消除，但相比之下，经过旋转后再进行拼接得到的结果更好，也省去了消除接缝的操作。

IV. Seam Elimination and Optimization

无论采用哪种投影方式，拼接得到的结果都会在两张图片的衔接处产生缝隙（Seam），这是因为算法只进行了图像的拼接，未考虑到不同图像直接存在的亮度差异。全景图上的缝隙会带来较差的视觉效果，因此有效地去除图像之间的缝隙至关重要。另外，从图 3-3 (c)和图 3-4 (b)可以看出，拼接过程中产生的缝隙可能是垂直或水平的，也可能两种缝隙同时存在。

实际上，消除缝隙的本质是将待拼接的两张图像在缝隙周围进行亮度融合，目前常用的方法有距离加权融合和金字塔融合（Pyramid Blending），这里以实现起来较为简单的距离加权融合算法为例，分别介绍垂直和水平缝隙的消除过程。

4.1 Vertical Junction Seam

以 Left to Right 投影顺序为例，图像 L 在 R 上的投影结果为 L^P ，接下来需要将图像 R 放置在图像 L^P 上方，如图 4-1 所示。

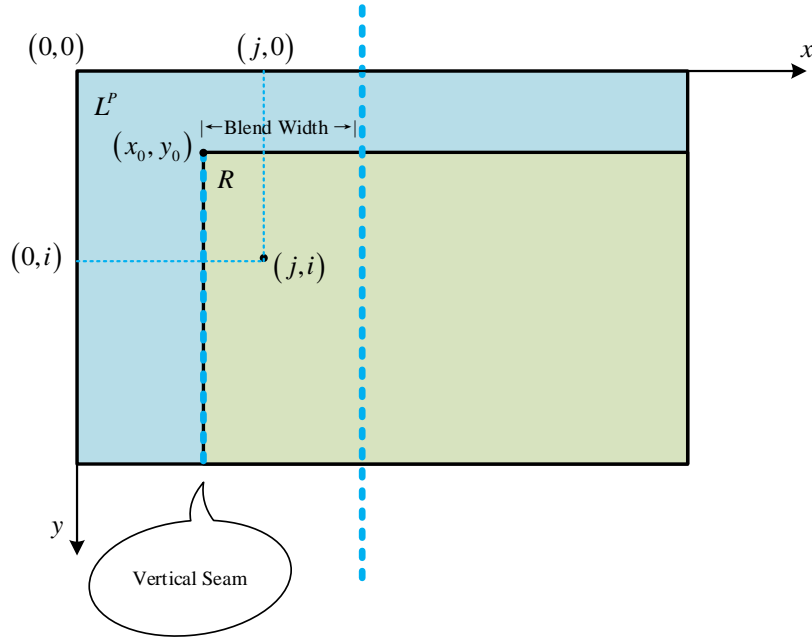


图 4-1 Sketch Map of Vertical Seam Elimination

图中 Blend Width 代表横向融合的宽度，记为 W_{blend} 。为了消除 Vertical Seam， $[x_0, x_0 + W_{blend}]$ 内的像素值由 L^P 和 R 共同决定，具体融合算法如公式 4-1 所示。

$$Dst(i, j) = \sigma \times R(i, j) + (1 - \sigma) \times L^P(i, j), \quad \sigma = (j - x_0) / W_{blend} \quad (4-1)$$

4.2 Horizontal Junction Seam

仍以 Left to Right 投影顺序为例，图 4-2 展示了水平缝隙的消除示意图。

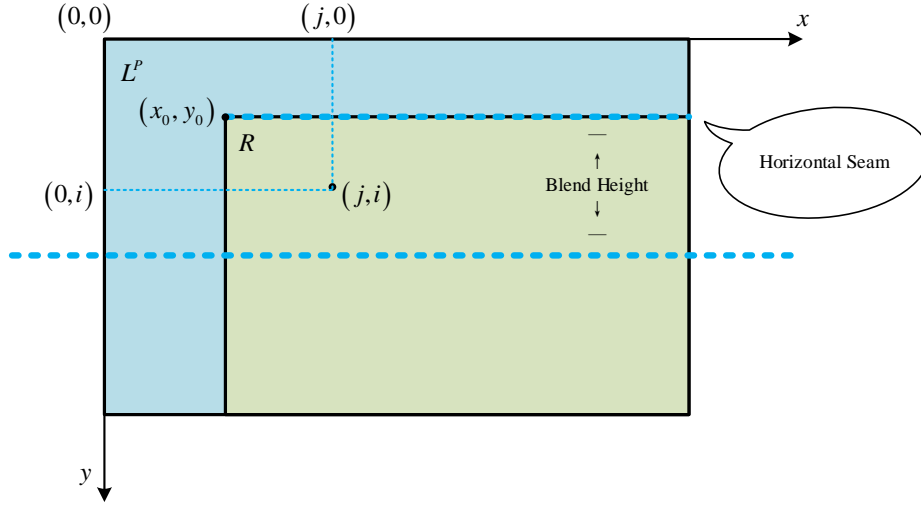


图 4-2 Sketch Map of Horizontal Seam Elimination

类似地，图中 Blend Height 代表纵向融合的高度，记为 H_{blend} 。为了消除 Horizontal Seam， $[y_0, y_0 + H_{blend}]$ 内的像素值同样由 L^p 和 R 共同决定，具体融合算法如公式 4-2 所示。

$$Dst(i, j) = \sigma \times R(i, j) + (1 - \sigma) \times L^p(i, j), \quad \sigma = (i - y_0) / H_{blend} \quad (4-2)$$

P.S. 对于不同投影顺序下得到的拼接结果，在缝隙消除操作中存在细微差异，具体实现见与报告一同提交的代码文件 *ImageMosaic.py*

V. Implementation of Image Mosaic

5.1 Algorithm Flow

本文设计的图像拼接流程如图 5-1 所示，图中的 *CenterIndex* 与选取的缝合方式有关。假设待拼接的图像数量为 N 当 *Stitch-Type*=Forward 时， $CenterIndex = N - 1$ ；当 *Stitch-Type*=Backward 时， $CenterIndex = 0$ ；当 *Stitch-Type*=Center 时， $CenterIndex = N/2$ 。

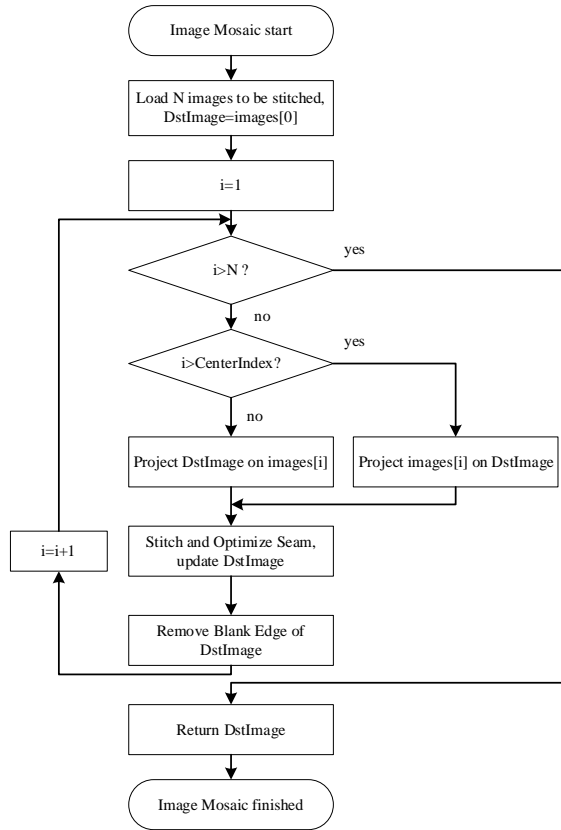


图 5-1 Algorithm Flow of Image Mosaic

5.2 Adjustable Parameters

对于本文设计的图像拼接算法，存在一些可调整的参数，这些参数对拼接结果有着不同程度的影响，如表 5-1 所示。

表 5-1 Adjustable Parameters in Image Mosaic

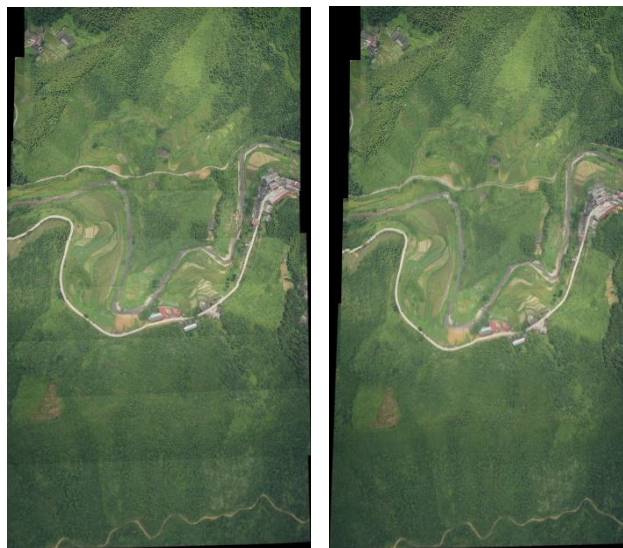
ID	Parameter	Description	Our Selection
1	DIST_THRESHOLD	获取图像匹配点对时的距离阈值	0.6
2	FEATURE_TYPE	选取的局部描述子类型	0.7
3	STITCH_TYPE	缝合类型（影响投影顺序）	Task Independent
4	BLEND_WIDTH	融合宽度（用于消除垂直缝隙）	100
5	BLEND_HEIGHT	融合高度（用于消除水平缝隙）	100

5.3 Results of Image Mosaic

基于以上图像拼接算法，本文对给定的图像数据集进行了拼接操作。给定的数据集包含两类，第一类为 **translation**，即待拼接的图像仅包含平移变换；第二类为 **rotate-scale**，代表待拼接的图像具有不同的尺度大小和角度方向。相比之下，**translation** 数据集更易于达到较好的拼接和融合效果。

5.3.1 Translation Transformation

对于仅进行平移变换的图像数据集,这里采用 Forward Stitching 的缝合方式,对旋转前后的数据集分别执行拼接操作,最终的拼接结果如图 5-2 所示。其中左图为未进行旋转操作得到的全景图像,右图为进行旋转操作后得到的全景图。



(a) Panorama of Translation-1



(b) Panorama of Translation-2



(c) Panorama of Translation-3 (Less Overlap)



(d) Panorama of Translation-4 (More Overlap)

图 5-2 Image Mosaic Result of Translation Dataset

5.3.2 Direction and Scale Transformation

相对于仅进行平移变换的图像，不同旋转角度和尺度大小的数据集更具有挑战性，这里采用 Center Stitching 的缝合方式，并对图像的衔接处进行了融合处理，最终的拼接结果如图 5-3 所示。其中右图相对于左图进行了垂直和水平方向上的缝隙消除处理。



(a) Panorama of Rotate-Scale-1



(b) Panorama of Rotate-Scale-2



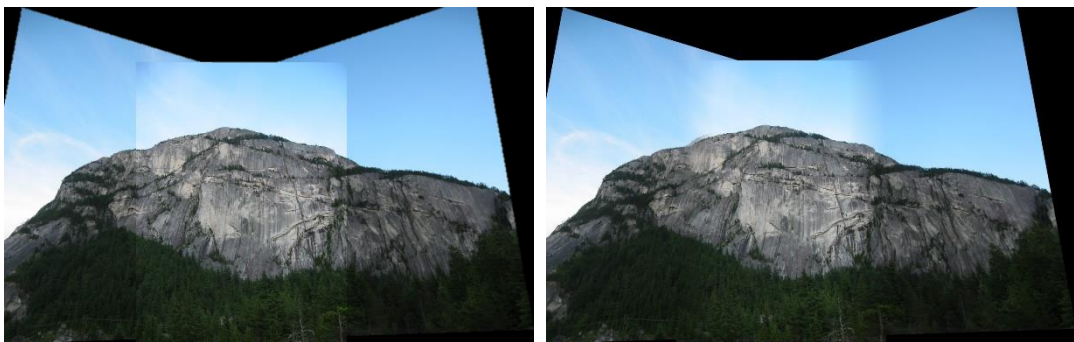
(c) Panorama of Rotate-Scale-3



(d) Panorama of Rotate-Scale-4



(e) Panorama of Rotate-Scale-5



(f) Panorama of Rotate-Scale-6

图 5-3 Image Mosaic Result of Rotate-Scale Dataset

VI. Conclusion and Ideas

为了使得拼接后的全景图更加美观，我们需要对黑色区域进行剪除操作。图 5-4 展示了部分图像裁剪优化后的结果。



(a) Panorama of Translation-1 (b) Panorama of Translation-4



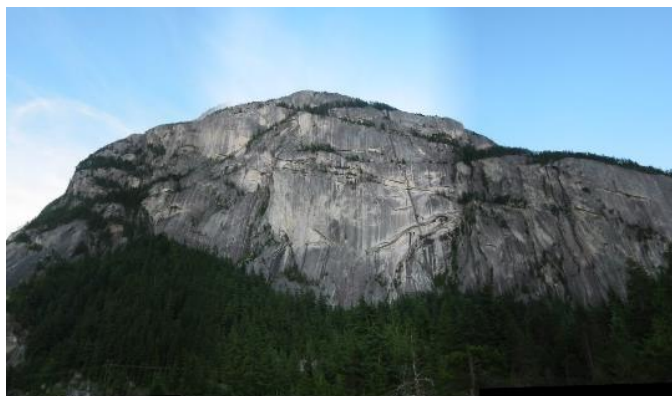
(c) Rotate-Scale-1



(d) Rotate-Scale-3



(e) Rotate-Scale-4



(f) Rotate-Scale-6

图 5-4 Cropped Panorama Images

参考文献

- [1] Brown, M., Lowe, D.G. Automatic Panoramic Image Stitching using Invariant Features. Int J Comput Vision 74, 59–73 (2007). <https://doi.org/10.1007/s11263-006-0002-3>