

数字图像处理

大作业一

书页图片矫正与拼接

学 号	2017011589
姓 名	吾尔开西
班 级	自 76

目录

- 一、总述 3
- 二、书页图片矫正 3
 - 1、获得原图点和目标图点 3
 - 2、TPS 薄板样条插值 4
 - 3、矫正结果 5
- 三、图片拼接 6
 - 1、SIFT 算法 6
 - 2、变换矩阵 6
 - 3、渐变权重 10
 - 4、多频段融合 12
- 四、图片二值化 13
- 五、自动选点 17
- 六、运算结果 21
 - 1、手动选点矫正拼接 21
 - 2、拼接图片二值化 21
 - 3、自动选点矫正拼接 22
- 七、总结 23
- 八、参考资料 23

一、总述

本次大作业要求矫正两张自然弯曲的书页图片，并将这两张有公共部分的图片拼接起来，另外还要将结果进行二值化。

为了矫正书页图片，我使用了 TPS 薄板样条插值的方法，分别用手动选关键点和自动选关键点的方式进行，其中自动选点的算法将在报告最后一部分详述。

拼接矫正后的书页图片时，本项目采用了 SIFT 算法查找关键点，用修改后的 RANSAC 算法匹配关键点并计算拼接用的变换矩阵，由于本次任务的特殊性，在这过程中分两次对图片进行调整，以便更好地匹配。得到变换矩阵后，用多频段融合算法进行拼接。

将图片二值化时，由于光照、折叠等原因，图片不同区域的灰度值差异较大，所以需要先从横向和纵向两个方向将图片进行均衡化，再用空域二值化的方法进行二值化。

二、书页图片矫正

作业要求矫正书页图片，即将自然弯曲的书页图片展平，这一点无法用线性的仿射变换实现，因为在仿射变换下，原图中位于一条直线上的点在变换后的图片中仍然在一条直线上，然而本任务要求将原来不在一条直线上的点（如某一行字），变换到一条水平直线上来。此外，描述书页弯曲的三维函数难以写出，因此很难自己设计变换方法。查阅资料后，我淘汰了一系列变换方法，比如畸变矫正（矫正由摄像机镜头工艺带来的畸变），最后决定选用 TPS 薄板样条插值的方法进行矫正。TPS 算法需要输入多个原图点坐标和目标图点坐标，本项目用了手动和自动两张方式来获得原图点坐标，这部分先介绍手动选点的方式，自动获取算法将在最后一部分进行介绍。

1、获得原图点和目标图点

对应代码：undistort.m

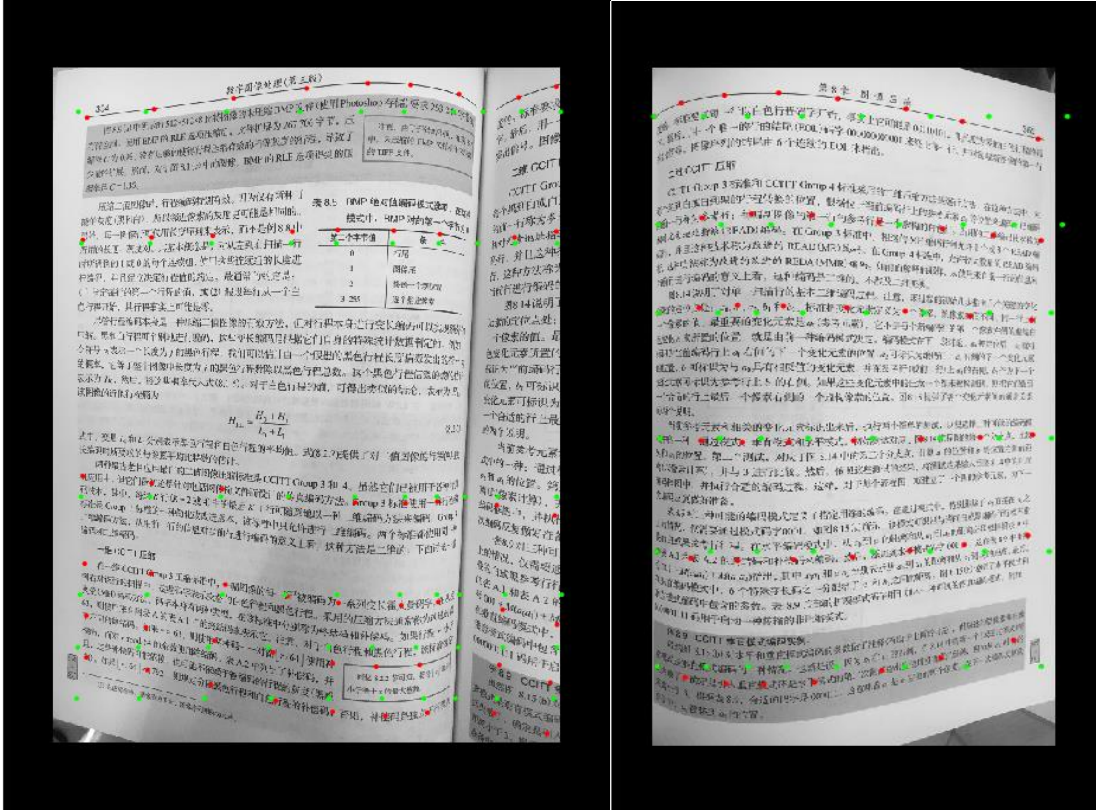
这部分介绍用手动选点的方式获取原图点，并计算其对应目标图点的方法。

书页图片矫正，需要将原图中呈弯曲状的每一行字矫正为水平直线，因此需要在原图中选取位于同一行字的一系列点坐标，并计算出它们对应的目标图中点的坐标，将这些点坐标输入 TPS 函数。

实际操作时，我在第一张图片中选了 6 行字，每一行字 10 个点；第二张图片选了 6 行字，每一行字 8 个点。

计算原图点坐标对应目标图点坐标时，每一行点是独立的。以第一张图片为例，首先从该行 10 个点的中心点（第 8 个点）向左出发，中心点的目标点坐标与它本身相同，中心点左边第一个点的纵坐标与中心点相同，横坐标的差是二者距离差。计算中心点左边第二个点横坐标是以中心点左边第一个点为准。同理计算中心点右边的点。

这种计算方法有一个缺陷，那就是原图中两个点之间的距离其实并不等于实际距离，所以需要在弯曲程度大点附近进行距离补偿。此外，由于算出来的目标点很可能超出原图坐标范围，所以需要在原图四周填充黑边。下图是两张图的选点结果，红点是原图中点，绿点是目标图中对应的点。



2、TPS 薄板样条插值

对应代码：rbfwarp2d.m

TPS 薄板样条插值是一种插值算法，在本项目中可以把矫正问题看作是一个二维插值问题，已知 n 个特征点对（目标图中点的坐标） (x_i, y_i) 和它们对应的函数值（原图中对应点的坐标） $f(x_i, y_i) \in D^2, (i = 1, 2, \dots, n)$ ，求一个插值函数，并用这个插值函数求得目标图中所有点对应的原图中点的坐标，求出的原图中点坐标可能不是整数，所以还需要用线性插值求得这个点的灰度值。

关键是如何求得这个插值函数，在 TPS 算法中(Bookstein & intelligence, 1989)，插值函数形式为：

$$f(x, y) = a_0 + a_1x + a_2y + \sum_{i=1}^n w_i U(|(x_i, y_i) - (x, y)|)$$

其中， $U(t) = t^2 \log t$ 。 a_0, a_1, a_2, w_i 均为参数，维度均为 2×1 ，满足 $\sum_{i=1}^n w_i = \sum_{i=1}^n w_i x_i = \sum_{i=1}^n w_i y_i = 0$ 。

将上面的条件和插值条件写成矩阵形式：

$$\begin{bmatrix} K & P \\ P^T & 0 \end{bmatrix} \begin{bmatrix} w \\ a \end{bmatrix} = \begin{bmatrix} V \\ 0 \end{bmatrix}$$

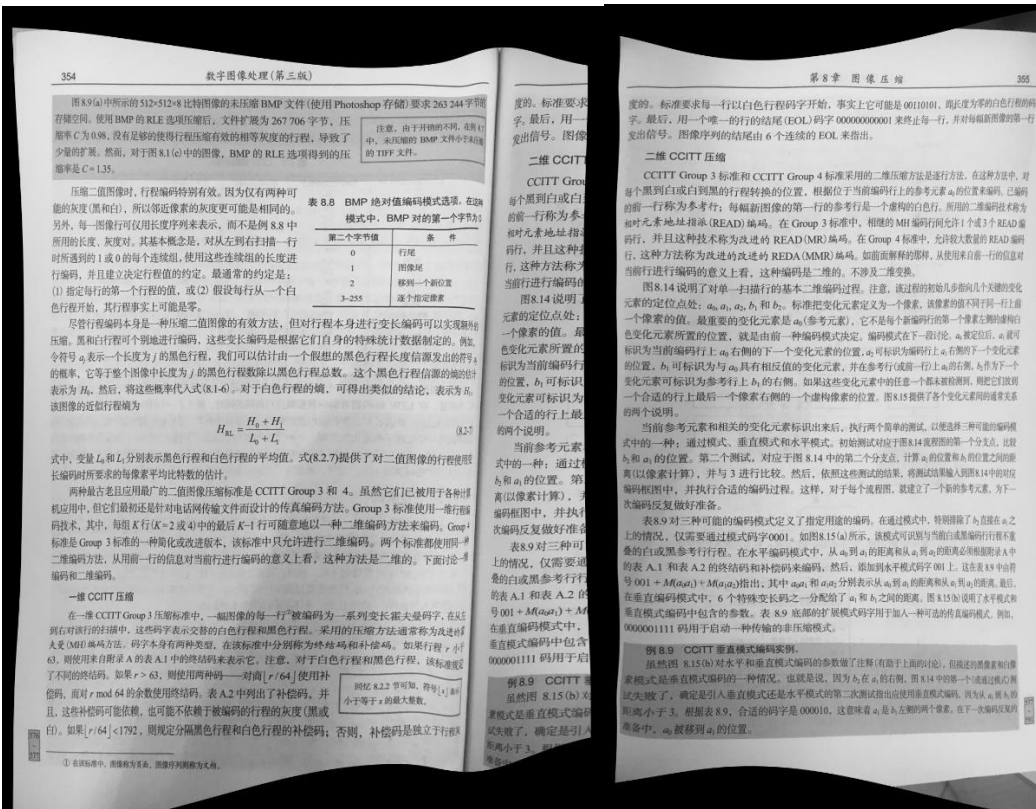
其中, $K \in D^{n \times n}$, $K_{ij} = U(|(x_i, y_i) - (x_j, y_j)|)$; $P \in D^{3 \times n}$, $P_i = [1 \ x_i \ y_i]$; $w \in$

$$D^{n \times 2}; a = \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix}; V \in D^{n \times 2}, V_i = f(x_i, y_i)$$

设 $L = \begin{bmatrix} K & P \\ P^T & 0 \end{bmatrix}$, 则 $\begin{bmatrix} w \\ a \end{bmatrix} = L^{-1} \begin{bmatrix} V \\ 0 \end{bmatrix}$, 也就求得了插值函数, 再用这个插值函数求得目标图中每个点对应原图中点的坐标, 并用线性插值求得目标图中点的灰度值即可。

3、矫正结果

下图是矫正结果, 由于没有标定太多选择点, 所以图一有少量未能矫正的部分以及少量畸变。



三、图片拼接

拼接矫正后的书页图片时，本项目采用了 SIFT 算法查找关键点，用修改后的 RANSAC 算法匹配两张图片的关键点并计算拼接用的变换矩阵，由于本次任务的特殊性，在这过程中有两次对图片进行的调整，以便更好地匹配。得到变换矩阵后，用多频段融合算法进行拼接。

1、SIFT 算法

代码：sift.m

SIFT 算法的代码实现使用了 MATLAB 的 vlfeat 工具包(<http://www.vlfeat.org/install-matlab.html>)

SIFT 算法是一种特征点提取的算法([Cheung & Hamarneh, 2007](#))，提取结果受光照、角度等因素影响较小。为了检测特征点，首先需要对不同尺度的原图片进行高斯滤波，滤波时用的高斯核方差有多个，从而形成高斯金字塔，每一层代表某一尺度的滤波结果，每一层有多张图片，对应不同方差高斯核的滤波结果。

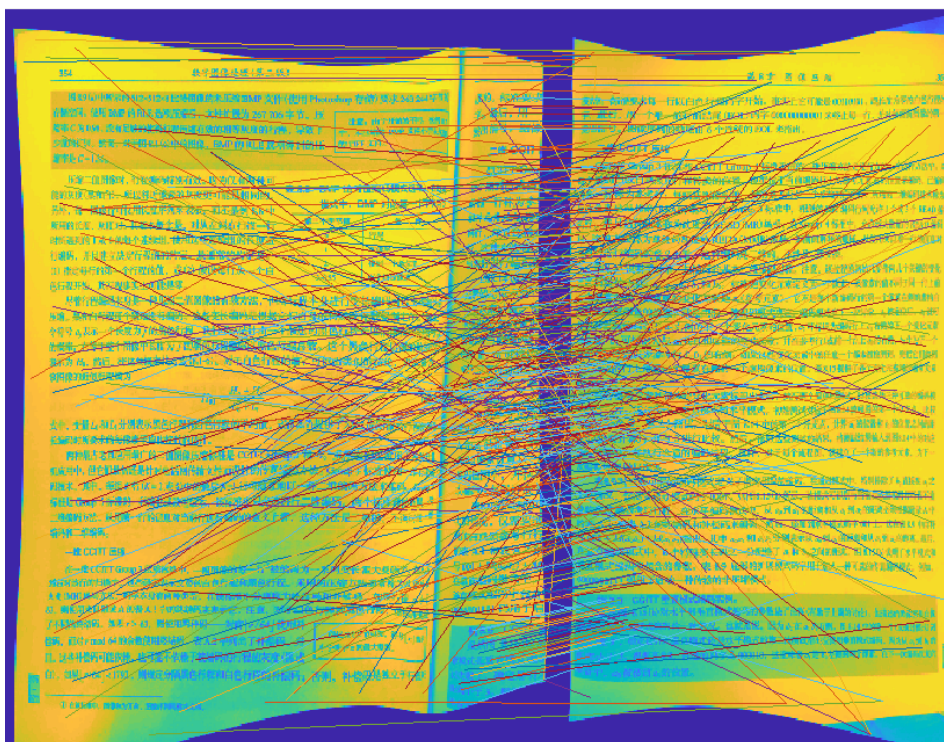
得到高斯金字塔后再将其相邻两层相减，得到高斯差分金字塔。之后在高斯差分金字塔中检测局部极值点，这些极值点是 DOG 空间中的，因为查找时不止在它二维平面上的领域查找，也要和这个点在相邻图片上的领域进行比较。得到的极值点再经过平滑处理、消除边缘效应，便能得到最终的特征点，除此之外还有这些特征点的方向信息。最后再用特征点领域内的梯度信息获得 128 维的特征描述子，该特征描述子有方向不变性。

2、变换矩阵

代码：stitch.m

要将两张图片进行拼接，最重要的得到一个 3×3 的变换矩阵，用该矩阵将图 1 投影到空间某个位置后，两张图片相匹配的特征点将最大限度地重合。该变换矩阵其实是用两张图片中相匹配的特征点进行计算，即将图片 1 的特征点进变换矩阵投影后与图片 2 中的特征点重合，具体过程如下。

用 sift 算法得到两张图片的特征点和它们的特征描述子后，可以用最近邻匹配的方法将两张图片的所有特征点匹配起来。下图是第一次匹配的结果，两张图中相匹配的特征点用不同一根彩线连接起来，可以看出有很多错误匹配的特征点。



接下来，我决定采用 RANSAC 算法得到变换矩阵和最佳匹配的特征点。由于本项目的特殊性，两张书页图片可以视作在同一平面上的矩形，只需要上下左右平移来进行拼接。此外，弯曲矫正时并没有把两张图片完全展平，因此我修改了 RANSAC 算法的计算过程，并在计算过程中对图片 2 进行了两次矫正，以便更好地进行拼接。

RANSAC 算法的思路是：从匹配的特征点中随机选择 n （如 $n=4$ ）对特征点，用这 n 对特征点计算变换矩阵，即将图一中的特征点投影到图二中对应特征点的位置，所

用的矩阵。设图一中的特征点为 $x_i = \begin{bmatrix} X_i \\ Y_i \\ 1 \end{bmatrix}, i = 1, 2, 3, 4$ ，变换矩阵为 H ，图二中对应特征

点为 x'_i ，则 $x'_i = k_i H x_i$ ，由于 x'_i 要归一化为 $\begin{bmatrix} X'_i \\ Y'_i \\ 1 \end{bmatrix}$ 的形式，因此 k_i 并不重要，上式子等同于

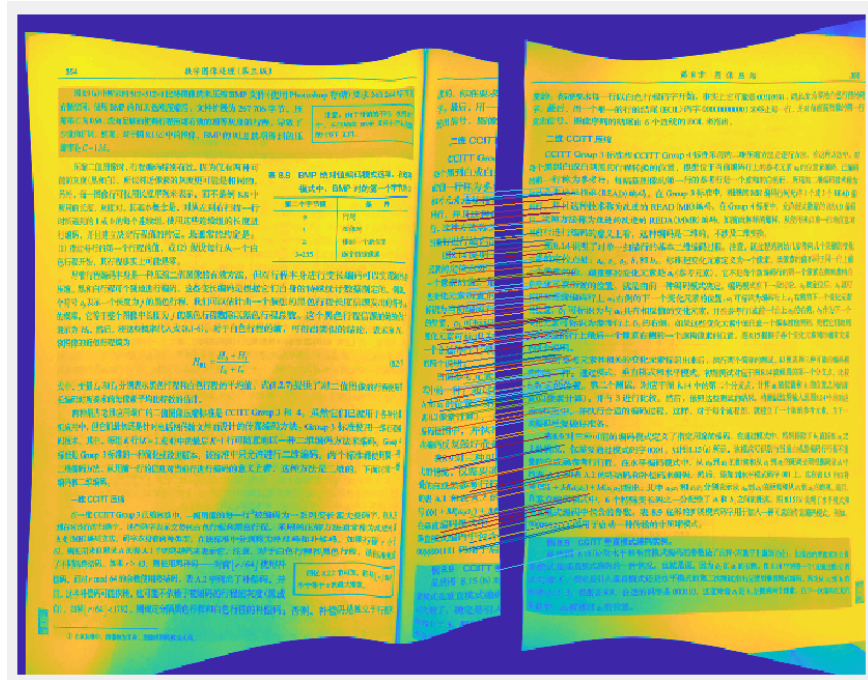
$x'_i \times H x_i = 0$ ，这其实是一个优化问题，最后化成奇异值分解的方法求得解 H 。得到 H 后对图 1 中所以特征点进行变换，看变换结果是否与它对应的图 2 中的特征点足够近（距离小于一个阈值），如果足够近，就将这对特征点记作最佳匹配特征点，每一轮中最佳匹配特征点的个数记作这个变换矩阵 H 的分数，循环足够多次（如 100 次），选择分数最高的 H 作为求得的变换矩阵，并返回它的最佳匹配特征点。

由于我们拼接的两张图片可以看作在同一个平面上，所以变换矩阵 H 只需两个未知量

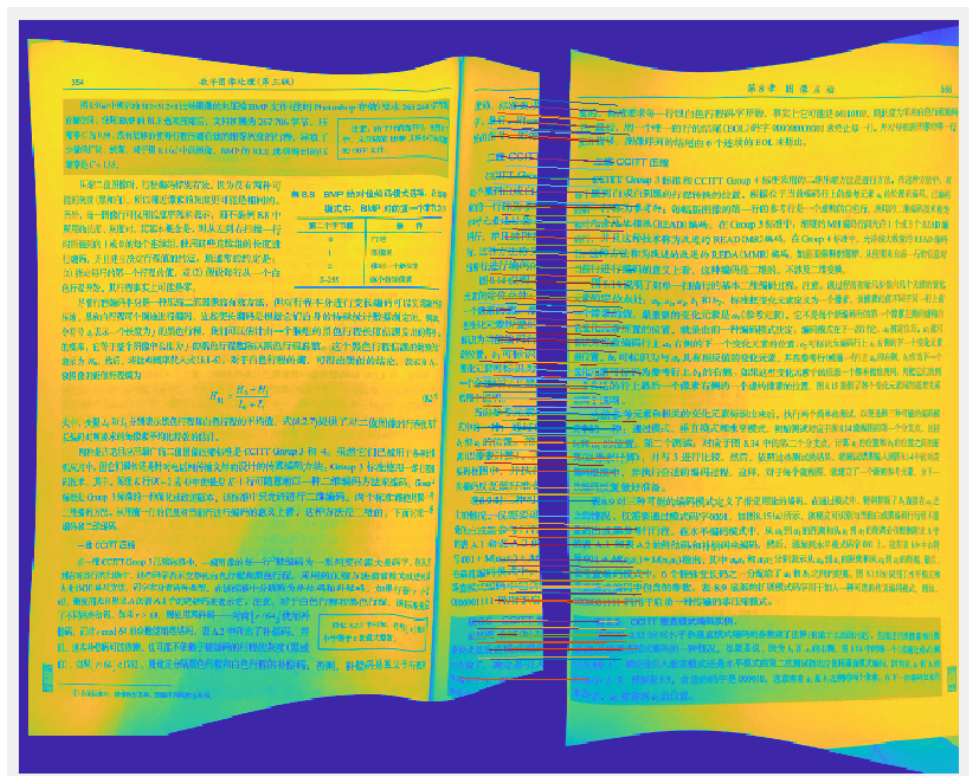
$$H = \begin{bmatrix} 1 & 0 & x_0 \\ 0 & 1 & y_0 \\ 0 & 0 & 1 \end{bmatrix}$$

随机取图 1 中的一个特征点 $x_1 = \begin{bmatrix} X_1 \\ Y_1 \\ 1 \end{bmatrix}$, Hx_i 将这个特征点上下左右平移。设 x_1 在图

二中对应的特征点为 x'_1 , 则 $x_0 = x'_1(1) - x_1(1)$, $y_0 = x'_1(2) - x_1(2)$ 。求解的其他流程与上面介绍的 RANSAC 算法相同, 下图是第一次用改版 RANSAC 算法得到的最佳匹配特征点, 可以看出匹配结果基本正确, 但匹配的对数较少。



接下来对图 2 进行第一次矫正, 由于两张图片中的纸张高度尺寸可能与现实的比例不同, 所以需要将两张图片的高度进行矫正, 做法是选取两对最高和最低的最佳匹配特征点, 根据双方的高度差比例来矫正第二张图片的高度。矫正后第二次用 sift 算法查找图 2 的特征点, 并用改版 RANSAC 查找最佳匹配点。下图是经第一次矫正和第二次匹配后的最佳匹配特征点, 可以看出匹配结果仍然正确, 且匹配对数较第一次更多。



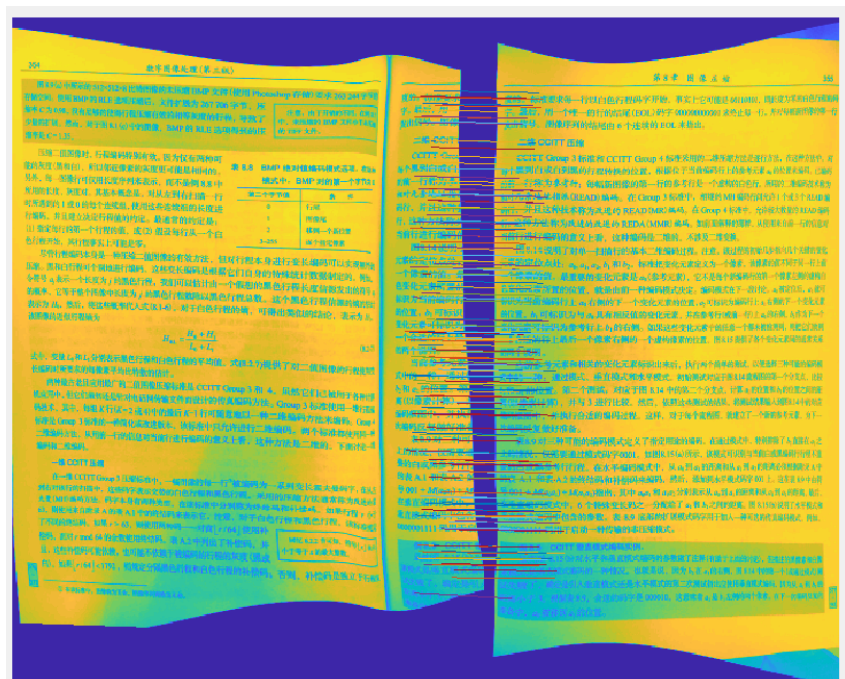
如果这时用得到的变换矩阵进行拼接，会得到一张质量不太好的拼接图，两张图片的公共部分中中间部位匹配的较好，而上面和下面的部位并没有完全重叠，会出现重影。从最佳匹配特征点的连接线上也可以看出，各个连接线的斜率并不相同。这是因为在前面一步将书页图片展平时，并未能将书页的每个部位完全展平，所以两张图片并不能直接拼接，否则会出现匹配了 A 部位，而 B 部位不能完全匹配的情况。

为了解决这个问题，我决定对图 2 进行第二次矫正，这次用（二）中介绍的 TPS 薄板样条插值。首先去除最佳匹配特征点中距离过近的点，再从中选择一个数量的特征对（如 60 对），让它们之间距离尽量大一些。TPS 插值需要输入原图点坐标和对应目标图点坐标，我们输入筛选后的特征对对图 2 的点坐标和对应图 1 中的点经变换矩阵

投影后的坐标，用符号来表示，记一对特征对为 $\{x_{i1}, x_{i2}\}$, $x_{i1} = \begin{bmatrix} X_{i1} \\ Y_{i1} \\ 1 \end{bmatrix}$ 是图 1 中特征点

坐标向量， x_{i2} 是对应图 2 中特征点坐标向量，则输入 TPS 的一对点中原图点坐标为 $[x_{i2}(1) \ x_{i2}(2)]$ ，目标图点坐标为 $[(Hx_{i1})(1) \ (Hx_{i1})(2)]$ 。

矫正后第三次用 sift 算法查找图 2 的特征点，并用改版 RANSAC 查找最佳匹配点，下面是匹配结果，可以看出最佳匹配点个数足够多，且连接线基本水平。



经过两次矫正三次匹配后，便可以用得到的变换矩阵 H 进行变换了，首先将两张图片扩大，保证变换后的图片都在范围内。变换时，虽然变换矩阵是按图 1 的特征点映射到图 2 特征点的方法所求，但图 1 保持不变。变换图 2，对目标图中的每个点坐标 (i, j) ，用 $H \begin{bmatrix} i \\ j \\ 1 \end{bmatrix}$ 计算图 2 原图坐标 (i', j') ，图 2 目标图中 (i, j) 的灰度值等于图 2 原图中 (i', j') 处的灰度值， (i', j') 可能不是整数，这时需要用线性插值的方法求得该点灰度值。

将图 2 变换后，其实就是将其移到合适的位置后，便可以将两张图进行融合了。

3、渐变权重

代码: `get_stitch_mask.m`

两张图片将用多频段融合的方法进行融合，在这之前，需要求出一个渐变权重，这个权重矩阵需要在多频段融合中用到。

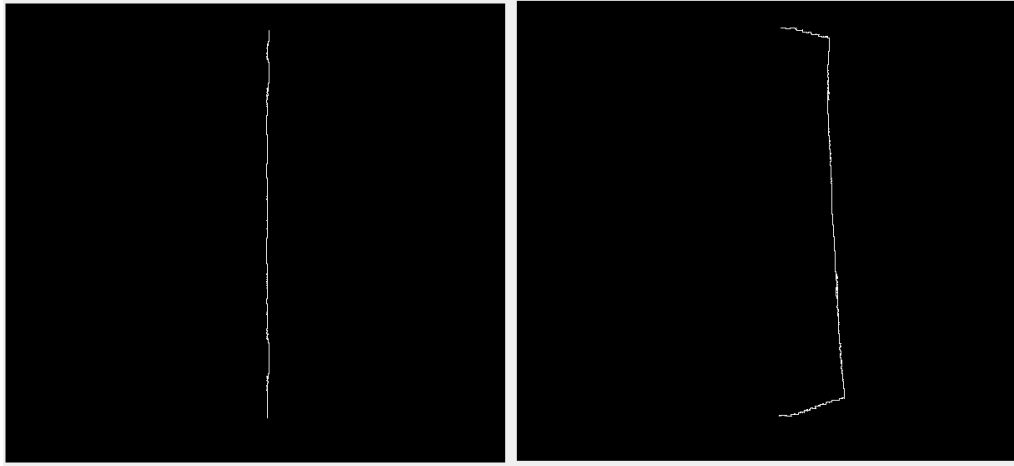
为表述方便，记 `mask_12` 为图 1 和图 2 公共部分的掩膜，即

$$mask_{12}(i, j) = \begin{cases} 1, & (i, j) \in Pic1 \text{ and } (i, j) \in Pic2 \\ 0, & otherwise \end{cases}$$

同理，`mask_n12` 表示无图 1 而有图 2 的部分，`mask_1` 表示有图 1 的部分。

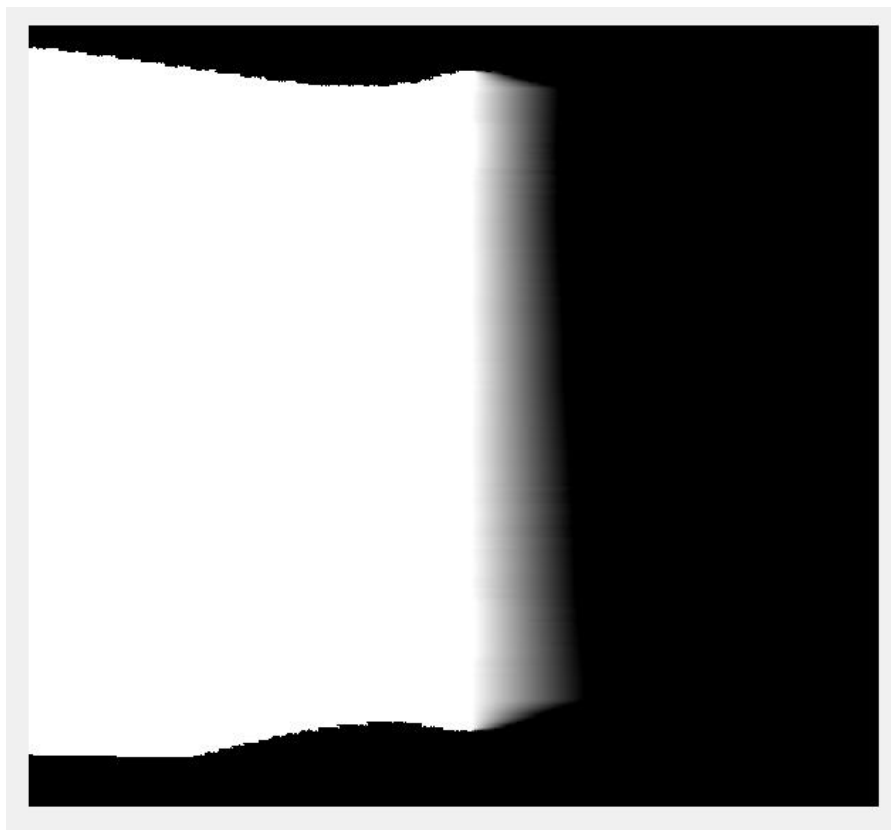
权重矩阵的大小与扩充后的两张图片的大小相同，矩阵值在 `mask_1n2=1` 的部分为 1，在 `mask_n12=1` 的部分为 0，在图 1 图 2 的公共部分从 1 渐变到 0。

为了求权重矩阵在 $\text{mask}_{12}=1$ 的部分的值，我们需要先找出 mask_{1n2} 和 mask_{n12} 的边界。以找 mask_{1n2} 的边界为例，我的方法是先将 mask_{1n2} 进行均值滤波，让滤波结果与 mask_{12} 逐元素相乘，即找出 mask_{1n2} 与 mask_{12} 的重叠部分，结果如下，两张图片的白色部分分别为 mask_{1n2} 和 mask_{n12} 的边界。



分别在两个边界上进行采样，每一行各取一个点，用这些点在中间部分进行一阶线性插值，其中 mask_{1n2} 上的点值为 1， mask_{n12} 上的点值为 0.

下图是得到的权重矩阵的结果，可以看出只有图片 1 的部分为 1，只有图片 2 的部分为 0，二者共有的部分从 1 渐变到 0。



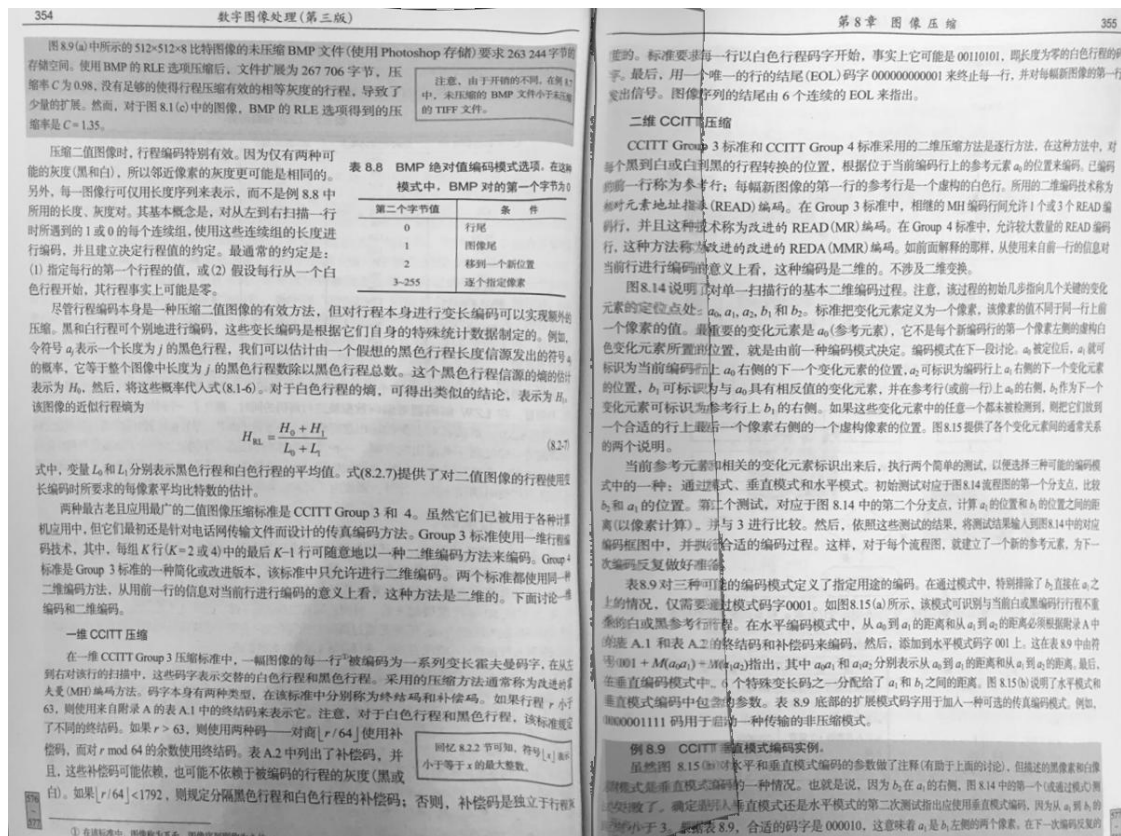
4、多频段融合

代码: multiFre_mix.m

将两张图片移到合适的位置后, 一种简单的融合办法是

$$result \ img = \frac{img1 + img2}{mask_1 + mask_2}$$

即图 1 和图 2 的公共部分相加以除以 2, 其他部分不变。下图是这种方法的结果, 可以看出两张图片的公共部分和其他部分有明显的割裂, 且色调也有差异, 为了解决这个问题, 我使用了多频段融合的方法。



多频段融合通过在不同的频段将两张图片加权求和, 从而得到比较好的融合效果。融合前需要先得到两张图片的高斯金字塔, 设图 1 为 $img1$, $kernel$ 是 5×5 的高斯卷积核, 方差为 1, 则

$$GA(0) = img1$$

$$GA(i) = \text{conv2}(GA(i-1), \text{kernel})$$

$$GA(i) = GA(i)(1:2:end, 1:2:end)$$

其中, $GA(i)(1:2:end, 1:2:end)$ 表示将 $GA(i)$ 降采样缩小两倍。用这种方法得到图 1 的六层高斯金字塔 GA , 同理可以得到图 2 的六层高斯金字塔 GB

接下来计算两张图片的拉普拉斯金字塔, 对图 1 来说:

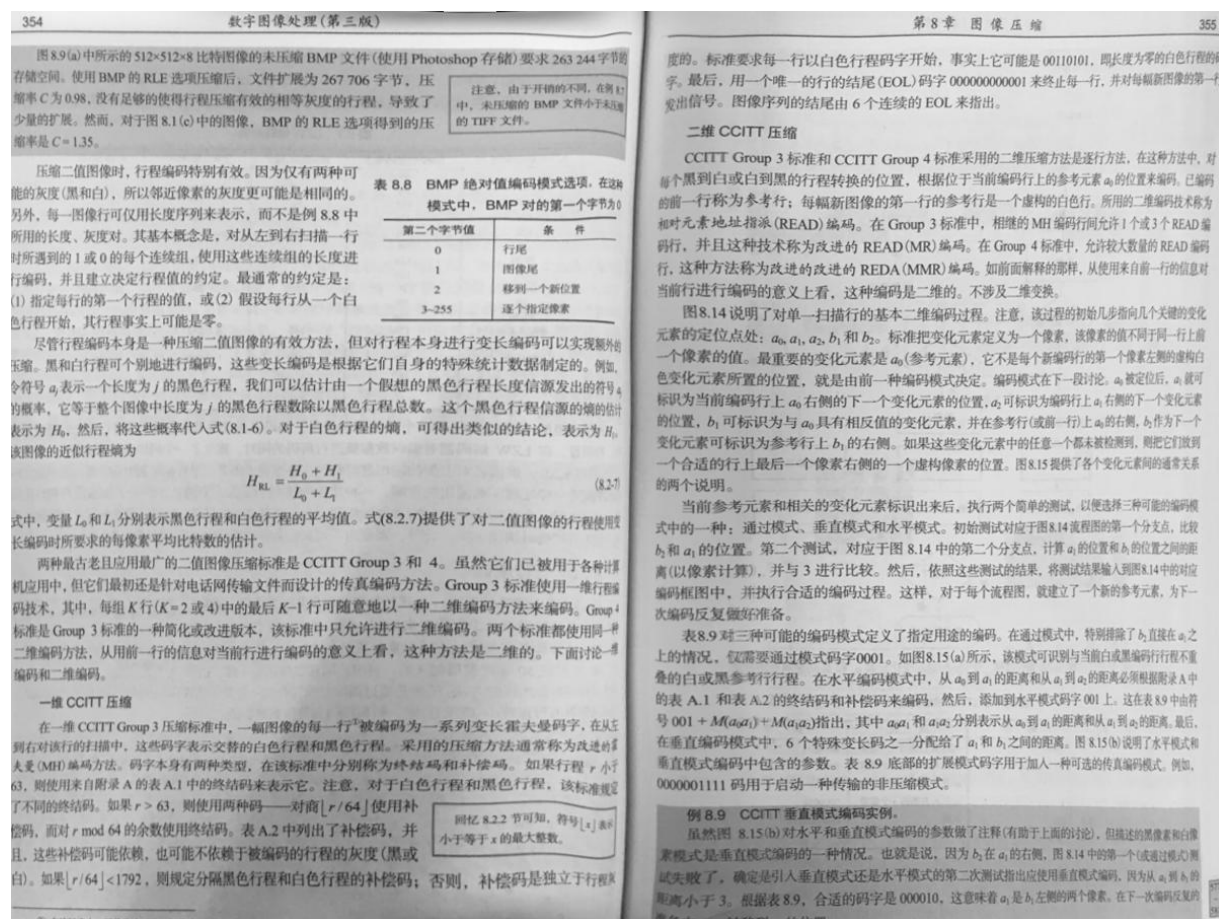
$$\text{LA}(5) = \text{GA}(5)$$

再接下来是将图 1 和图 2 对应的拉普拉斯金字塔层加权求和，设 (3、渐变矩阵) 中的得到的权重矩阵为 `mask`，则

$$LC(i) = LA(i) \circ \text{mask}(i) + LB(i) \circ (1 - \text{mask}(i))$$

最后再将LC(i)伸缩到原图同样大小后全部相加即可。

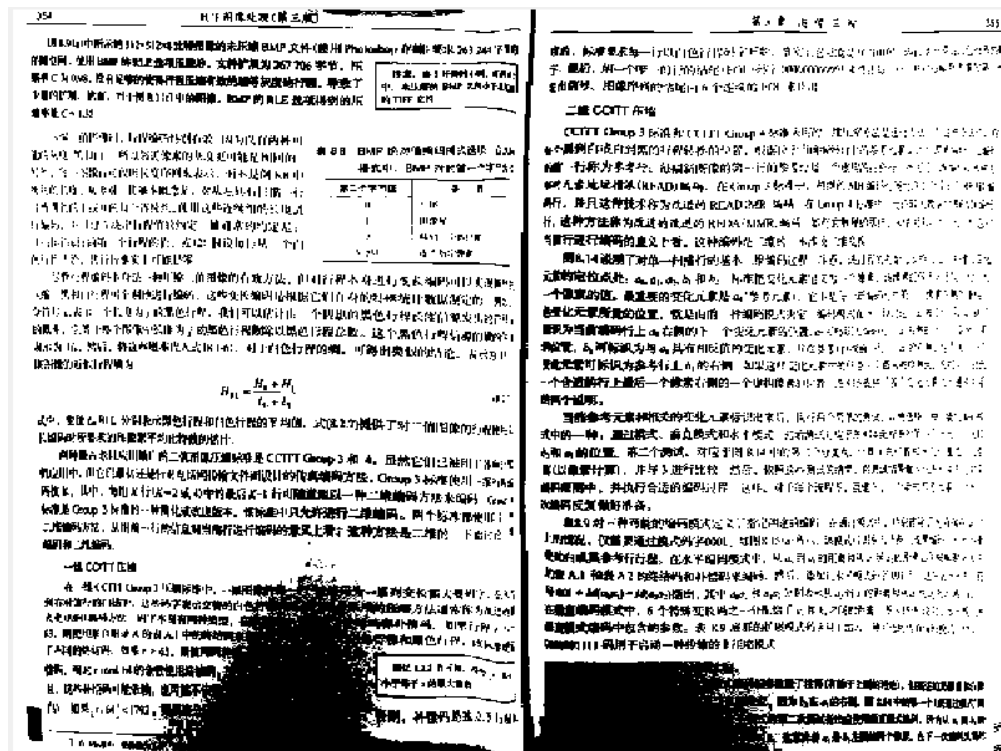
得到融合图像后去掉黑边，便能得到最终结果，如下图所示，几乎看不出拼接痕迹，过渡比较自然。



四、图片二值化

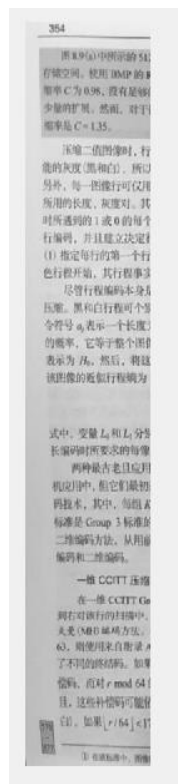
代码: myBinarize.m

图片二值化可以直接在空域进行简单的二值化,即定一个阈值,每个像素点大于该值为1,小于该阈值为0,但这种方法效果不佳,因为受光照等影响,图片每一部分文字的灰度值有一定差别,导致阈值很难选择,往往是有些文字太浅看不清,有些部分全黑。

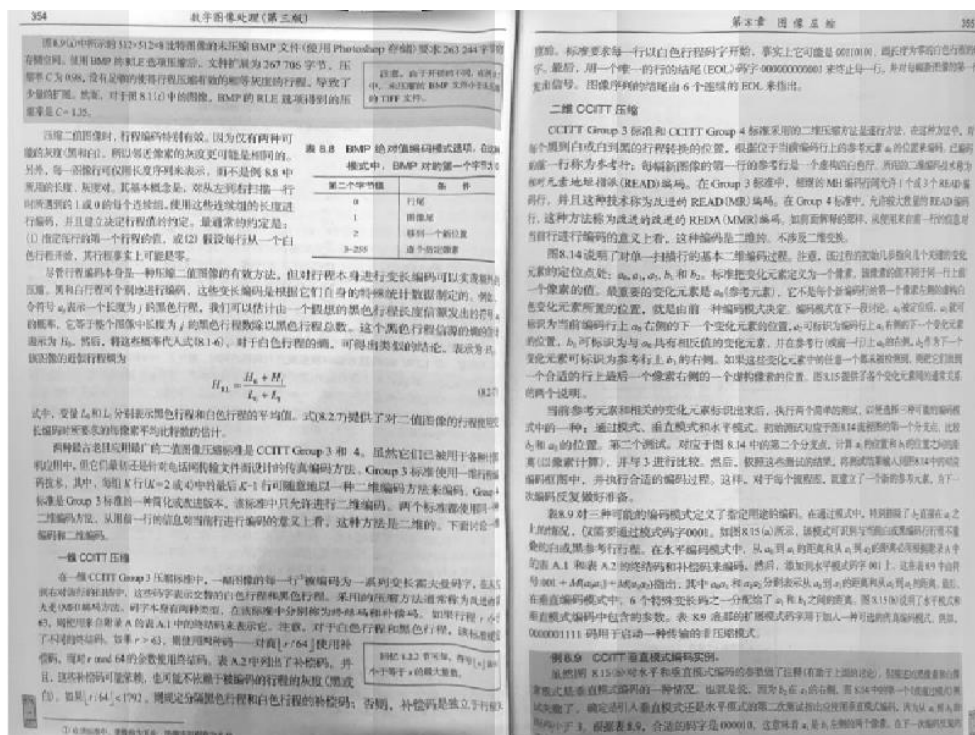


考虑到所要二值化的图片的特殊性,我决定使用空域均衡的方法解决这个问题。我所需要二值化的图片是一张单纯的书页图片,也就是说,选取该图片的一大块,其灰度值的平均值应该与其他部分的平均值差不多。而实际情况下,光照等因素的影响使各个部分的平均值往往有差异。因此,为了消除光照的影响,可以将图片各个部分的平均值均衡化。

首先进行横向均衡,取图片竖立条状的一部分,如下图。



求这一部分图片的平均值，再将书页其他部分划成同样宽度的条状部分，使它们的平均值与第一部分相同，结果如下。



再用同样的方法进行纵向均衡化，结果如下。

图8.9(a)中所示的512×512×8位图像文件的未压缩BMP文件(使用Photoshop存储)要求263 244字节的存储空间。使用BMP的RLE选项压缩后,文件扩展为267 706字节。压缩率C为0.98,没有足够的使得行程压缩有效的相等长度的行程,导致了少量的扩展。然而,对于图8.1(b)中的图像,BMP的RLE选项得到的压缩率是C=1.35。

行程二值图像时,行程编码特别有效,因为仅有两种可能的灰度(黑和白),所以邻近像素的灰度更可能是相同的。

另外,每一行程可以用仅用长度来表示,而不是图8.8中所需的长度、灰度。其基本概念是,从左到右扫描一行时所遇到的1或0的每个连续块,使用这些连续块的长度进行编码,并且建立决定行程值的约定。最通常的约定是:

(1)指定行程的第一个行程的灰,或(2)假设每行从一个白色行程开始,其行程至少可能是一。

尽管行程编码本身是一种压缩二值图像的有效方法,但对行程本身进行变长编码可以实现更好的压缩。黑和白行程可单独地进行编码,这些变长编码是根据它们自身的特殊统计数据制定的。例如,命令符 a_i 表示一个长度为 i 的黑色行程,我们可以估计出一个假想的黑色行程长度信息流发出的符号 a_i 的概率,它等于整个图像中长度为 i 的黑色行程数除以黑色行程总数。这个黑色行程信息的概率表示为 h_i 。然后,将这些概率代入式(8.1-6),对于白色行程的编码,可得出类似的结论,表示为 h_i 。因此,行程的近似编码为

$$H_{\text{RLE}} = \frac{H_0 + H_1}{L_0 + L_1} \quad (8.2-7)$$

式中,变量 L_0 和 L_1 分别表示黑色行程和白色行程的平均值。式(8.2-7)提供了对二值图像的行程编码和行程编码所需的每像素平均比特的估计。

两种最古老且应用最广的二值图像压缩标准是CCITT Group 3和4。虽然它们已被用于各种应用中,但它们最初还是针对电话传输文件而设计的传真编码方法。Group 3标准使用一维行程编码技术,其中每行以 $r=2$ 或 $r=4$ 中的中位数后一行可任意地以二种二维编码方法来编码。Group 4标准是Group 3标准的一种简化或改进版本,该标准中只允许进行二维编码。两个标准都使用行程二值编码,从用前一行信息对当前行程进行编码的意义上看,这种方法是二维的。下面讨论二值编码和行程编码。

一维CCITT压缩

在一维CCITT Group 3压缩标准中,一幅图像的每一行被编码为一组变长整数码字,在从左到右的扫描中,这些码字表示白色的行程和黑色行程。采用的编码方法通常称为改进的 READ(MR)编码方法。码字本身有两种类型,在标准中分别称为特殊码和补码。如果行程 $r < 63$,则使用来自表A.1中的码字来表示它。注意,对于白色行程和黑色行程,使用了不同的码字。如果 $r \geq 63$,则使用两种码——对 $r/64$ 使用补码,而对 $r \bmod 64$ 的余数使用特殊码。表A.2中列出了补码,并且,这些补码可能使用,也可能不依赖于被编码的行程的灰度(黑或白)。如果 $r/64 = 1792$,则规定分隔黑色行程和白色行程的补码;否则,补码是独立于行程的。

① 在标准中,行程码为灰色,行程码的码字为黑色。

表 8.8 BMP 绝对值编码模式选项,在BMP模式中, BMP 对第一个字节

第二个字节	解释
0	行程
1	特殊码
2	补码 全数行程
3-255	指定行程数

最初,标准要求每一行以白色行程码字开始,事实上它可能是0001001,即从左到右的白色行程的码字。最后,用一个唯一的行程的结尾(EOL)码字000000000000来终止每一行,并给每行新行程的第一行发出信号。图像序列的结尾由6个连续的行EOL来给出。

二维CCITT压缩

CCITT Group 3标准和CCITT Group 4标准采用的二维压缩方法是逐行方法,在这种方法中,对每个黑到白或白到黑的行程转换的位置,根据位于当前编码行上的参考元素 a_i 的位置来编码。已编码的行称为参考行;每幅新图像的第一行的参考行是一个虚设的白行,所用的虚设白行称为参考地地址码(READ)码。在Group 3标准中,相邻的MR编码行间允许1个或3个READ码行,并且这种技术称为改进的READ(MR)编码。在Group 4标准中,允许较大数量的READ码行,这种方法称为改进的改进的READ(MMR)编码。如前面所解释的那样,从使用目前一行的信息对当前行程进行编码的意义上看,这种编码是二维的,不涉及二维变换。

图8.14说明了对于每一扫描行的基本二维编码过程。注意,该过程的初始步是取几个连续变化元素的定位点处: $a_i, a_{i+1}, a_{i+2}, a_{i+3}$ 和 a_{i+4} 。标准把变化元素定义为一个像素,该像素的值不同于它上面一个像素的值。最重要的变化元素是 a_i (参考元素),它不是每个行程的第一个像素;黑到白或白到黑变化元素所置的位置,就是由前一种编码模式决定。编码模式在下一段讨论。 a_{i+1} 被定义为 a_i 可标识为当前编码行上 a_i 右侧的下一个变化元素的位置。 a_{i+2} 可标识为与 a_i 具有相反值的变化元素,并在参考行(或前一行)上 a_i 的右侧。 a_{i+3} 可标识为一个变化元素可标识为参考行上 a_i 的右侧。如果这些变化元素中的任何一个像素未被编码,则它应被一个合适的行上最后一个像素右侧的一个虚设像素的位置。图8.15给出了各个变化元素间的通常关系的两个说明。

当前参考元素和相关的变化元素标识出来以后,执行两个简单的测试,以便选择一种可能的编码模式中的一种:通过模式、垂直模式和水平模式。初始测试对应于图8.14流程图的第一个分支,比较 a_i 和 a_{i+1} 的位置。第二个测试,对应于图8.14中的第二个分支点,计算 a_i 的位置和 a_{i+2} 的位置(以像素计算),并与3进行比较。然后,依照这些测试的结果,将模式结果输入图8.14中的编码框图中,并执行合适的编码过程。这样,对于每个流程框,就建立了一个新的参考元素,并一次编码反复做好准备。

表8.9对三种可能的编码模式定义了指定用途的编码。在通过模式时,特别强调了在图8.14上的情况,仅需要模式码字0001。如图8.15(a)所示,该模式可识别为当前白或黑行程的白或黑或参考行程。在水平编码模式中,从 a_i 到 a_{i+1} 的距离和从 a_i 到 a_{i+2} 的距离(从表A.1和表A.2的码字和补码来编码),然后,总距离水平模式码字001。这在表8.9中符号为001 + $M(a_{i+1}) + M(a_{i+2})$ 指出,其中 a_{i+1} 和 a_{i+2} 分别表示从 a_i 到 a_{i+1} 的距离和从 a_i 到 a_{i+2} 的距离。在垂直编码模式中,6个特殊变长码之一分配给了 a_i 和 a_{i+2} 之间的差。图8.15(b)显示了在垂直编码模式中包含的参数。表8.9流程图的编码模式码字用于输入一种可能的编码模式。例如,0000011111码字启动一种特殊的非压缩模式。

例8.9 CCITT垂直模式编码实例

虽然图8.15(b)对水平和垂直模式编码的参数做了注释(有助于上面的讨论),但垂直模式的编码实例是垂直模式编码的一种情况。也就是说,因为 a_i 在 a_{i+1} 的右侧,图8.14中的第一个分支测试失败了,确定输入垂直模式还是水平模式第二次测试指出应采用垂直模式编码,因为 a_i 和 a_{i+2} 间的距离 $r=3$,根据表8.9,合适的码字是000010,这意味着 a_i 是 a_{i+2} 右侧的第二个像素。在下一段的讨论中,我们将看到,合适的码字是000010,这意味着 a_i 是 a_{i+2} 右侧的第二个像素。在下一段的讨论中,我们将看到,合适的码字是000010,这意味着 a_i 是 a_{i+2} 右侧的第二个像素。

最后再将这个图片进行空域二值化,结果如下。

图8.9(a)中所示的512×512×8位图像文件的未压缩BMP文件(使用Photoshop存储)要求263 244字节的存储空间。使用BMP的RLE选项压缩后,文件扩展为267 706字节。压缩率C为0.98,没有足够的使得行程压缩有效的相等长度的行程,导致了少量的扩展。然而,对于图8.1(b)中的图像,BMP的RLE选项得到的压缩率是C=1.35。

行程二值图像时,行程编码特别有效,因为仅有两种可能的灰度(黑和白),所以邻近像素的灰度更可能是相同的。

另外,每一行程可以用仅用长度来表示,而不是图8.8中所需的长度、灰度。其基本概念是,从左到右扫描一行时所遇到的1或0的每个连续块,使用这些连续块的长度进行编码,并且建立决定行程值的约定。最通常的约定是:

(1)指定行程的第一个行程的灰,或(2)假设每行从一个白色行程开始,其行程至少可能是一。

尽管行程编码本身是一种压缩二值图像的有效方法,但对行程本身进行变长编码可以实现更好的压缩。黑和白行程可单独地进行编码,这些变长编码是根据它们自身的特殊统计数据制定的。例如,命令符 a_i 表示一个长度为 i 的黑色行程,我们可以估计出一个假想的黑色行程长度信息流发出的符号 a_i 的概率,它等于整个图像中长度为 i 的黑色行程数除以黑色行程总数。这个黑色行程信息的概率表示为 h_i 。然后,将这些概率代入式(8.1-6),对于白色行程的编码,可得出类似的结论,表示为 h_i 。因此,行程的近似编码为

$$H_{\text{RLE}} = \frac{H_0 + H_1}{L_0 + L_1} \quad (8.2-7)$$

式中,变量 L_0 和 L_1 分别表示黑色行程和白色行程的平均值。式(8.2-7)提供了对二值图像的行程编码和行程编码所需的每像素平均比特的估计。

两种最古老且应用最广的二值图像压缩标准是CCITT Group 3和4。虽然它们已被用于各种应用中,但它们最初还是针对电话传输文件而设计的传真编码方法。Group 3标准使用一维行程编码技术,其中每行以 $r=2$ 或 $r=4$ 中的中位数后一行可任意地以二种二维编码方法来编码。Group 4标准是Group 3标准的一种简化或改进版本,该标准中只允许进行二维编码。两个标准都使用行程二值编码,从用前一行信息对当前行程进行编码的意义上看,这种方法是二维的。下面讨论二值编码和行程编码。

一维CCITT压缩

在一维CCITT Group 3压缩标准中,一幅图像的每一行被编码为一组变长整数码字,在从左到右的扫描中,这些码字表示白色的行程和黑色行程。采用的编码方法通常称为改进的 READ(MR)编码方法。码字本身有两种类型,在标准中分别称为特殊码和补码。如果行程 $r < 63$,则使用来自表A.1中的码字来表示它。注意,对于白色行程和黑色行程,使用了不同的码字。如果 $r \geq 63$,则使用两种码——对 $r/64$ 使用补码,而对 $r \bmod 64$ 的余数使用特殊码。表A.2中列出了补码,并且,这些补码可能使用,也可能不依赖于被编码的行程的灰度(黑或白)。如果 $r/64 = 1792$,则规定分隔黑色行程和白色行程的补码;否则,补码是独立于行程的。

① 在标准中,行程码为灰色,行程码的码字为黑色。

表 8.8 BMP 绝对值编码模式选项,在BMP模式中, BMP 对第一个字节

第二个字节	解释
0	行程
1	特殊码
2	补码 全数行程
3-255	指定行程数

最初,标准要求每一行以白色行程码字开始,事实上它可能是0001001,即从左到右的白色行程的码字。最后,用一个唯一的行程的结尾(EOL)码字000000000000来终止每一行,并给每行新行程的第一行发出信号。图像序列的结尾由6个连续的行EOL来给出。

二维CCITT压缩

CCITT Group 3标准和CCITT Group 4标准采用的二维压缩方法是逐行方法,在这种方法中,对每个黑到白或白到黑的行程转换的位置,根据位于当前编码行上的参考元素 a_i 的位置来编码。已编码的行称为参考行;每幅新图像的第一行的参考行是一个虚设的白行,所用的虚设白行称为参考地地址码(READ)码。在Group 3标准中,相邻的MR编码行间允许1个或3个READ码行,并且这种技术称为改进的READ(MR)编码。在Group 4标准中,允许较大数量的READ码行,这种方法称为改进的改进的READ(MMR)编码。如前面所解释的那样,从使用目前一行的信息对当前行程进行编码的意义上看,这种编码是二维的,不涉及二维变换。

图8.14说明了对于每一扫描行的基本二维编码过程。注意,该过程的初始步是取几个连续变化元素的定位点处: $a_i, a_{i+1}, a_{i+2}, a_{i+3}$ 和 a_{i+4} 。标准把变化元素定义为一个像素,该像素的值不同于它上面一个像素的值。最重要的变化元素是 a_i (参考元素),它不是每个行程的第一个像素;黑到白或白到黑变化元素所置的位置,就是由前一种编码模式决定。编码模式在下一段讨论。 a_{i+1} 被定义为 a_i 可标识为当前编码行上 a_i 右侧的下一个变化元素的位置。 a_{i+2} 可标识为与 a_i 具有相反值的变化元素,并在参考行(或前一行)上 a_i 的右侧。 a_{i+3} 可标识为一个变化元素可标识为参考行上 a_i 的右侧。如果这些变化元素中的任何一个像素未被编码,则它应被一个合适的行上最后一个像素右侧的一个虚设像素的位置。图8.15给出了各个变化元素间的通常关系的两个说明。

当前参考元素和相关的变化元素标识出来以后,执行两个简单的测试,以便选择一种可能的编码模式中的一种:通过模式、垂直模式和水平模式。初始测试对应于图8.14流程图的第一个分支,比较 a_i 和 a_{i+1} 的位置。第二个测试,对应于图8.14中的第二个分支点,计算 a_i 的位置和 a_{i+2} 的位置(以像素计算),并与3进行比较。然后,依照这些测试的结果,将模式结果输入图8.14中的编码框图中,并执行合适的编码过程。这样,对于每个流程框,就建立了一个新的参考元素,并一次编码反复做好准备。

表8.9对三种可能的编码模式定义了指定用途的编码。在通过模式时,特别强调了在图8.14上的情况,仅需要模式码字0001。如图8.15(a)所示,该模式可识别为当前白或黑行程的白或黑或参考行程。在水平编码模式中,从 a_i 到 a_{i+1} 的距离和从 a_i 到 a_{i+2} 的距离(从表A.1和表A.2的码字和补码来编码),然后,总距离水平模式码字001。这在表8.9中符号为001 + $M(a_{i+1}) + M(a_{i+2})$ 指出,其中 a_{i+1} 和 a_{i+2} 分别表示从 a_i 到 a_{i+1} 的距离和从 a_i 到 a_{i+2} 的距离。在垂直编码模式中,6个特殊变长码之一分配给了 a_i 和 a_{i+2} 之间的差。图8.15(b)显示了在垂直编码模式中包含的参数。表8.9流程图的编码模式码字用于输入一种可能的编码模式。例如,0000011111码字启动一种特殊的非压缩模式。

例8.9 CCITT垂直模式编码实例

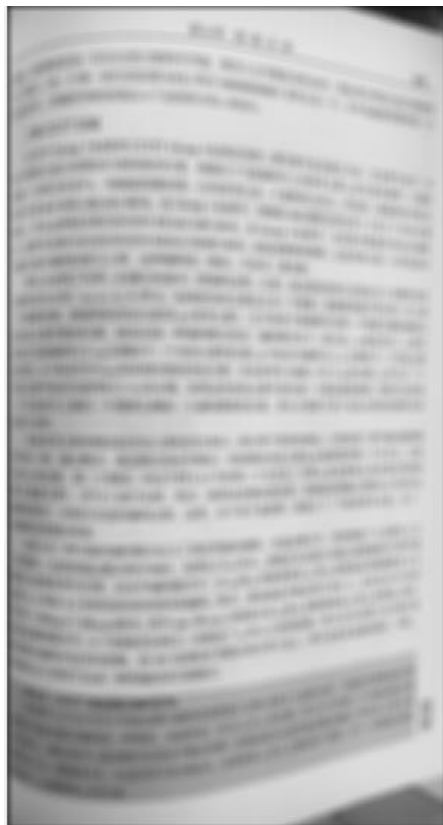
虽然图8.15(b)对水平和垂直模式编码的参数做了注释(有助于上面的讨论),但垂直模式的编码实例是垂直模式编码的一种情况。也就是说,因为 a_i 在 a_{i+1} 的右侧,图8.14中的第一个分支测试失败了,确定输入垂直模式还是水平模式第二次测试指出应采用垂直模式编码,因为 a_i 和 a_{i+2} 间的距离 $r=3$,根据表8.9,合适的码字是000010,这意味着 a_i 是 a_{i+2} 右侧的第二个像素。在下一段的讨论中,我们将看到,合适的码字是000010,这意味着 a_i 是 a_{i+2} 右侧的第二个像素。

五、自动选点

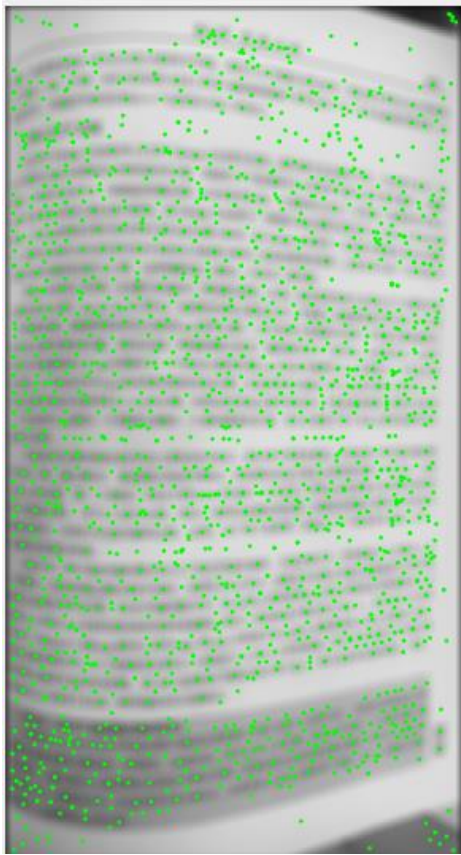
代码: `auto_point.m`

在用 TPS 进行书页图片弯曲矫正时, 需要得到若干原图点和它们对应的目标图点, 上述过程中是使用手动选点的方式来得到原图点, 下面介绍自动选点的方式。

自动选点最重要的是找到书页图片中文字的位置, 为了实现这个目的, 我将原图进行高斯模糊, 如下图。



图中的文字几乎都变成一个个的黑点, 接下来需要找到图中的局部极值点。方便起见, 我使用了第三部分介绍的 `sift` 算法找局部极值点。下图为结果, 可以看出找到的特征点很大一部分都文字。



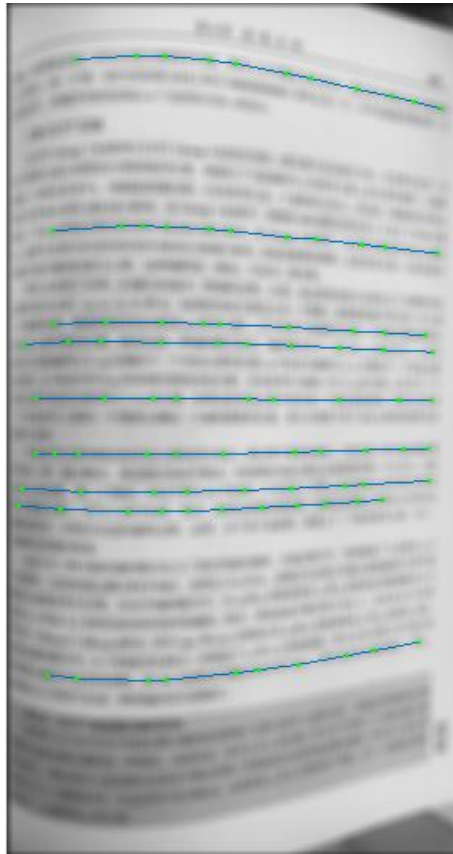
接下来对这些点进行筛选，先去除灰度值过高的点，再将剩下的点按灰度值排序，再去除距离过近的点，这样就能优先排除高灰度值的点。

接下来对剩下的点寻找它的右邻居，即在每个点右边一个方框内找与它的斜率绝对值最小的点，这个邻居最有可能和这个点在同一行文字上。结果如下，每个点与它的右邻居相连。

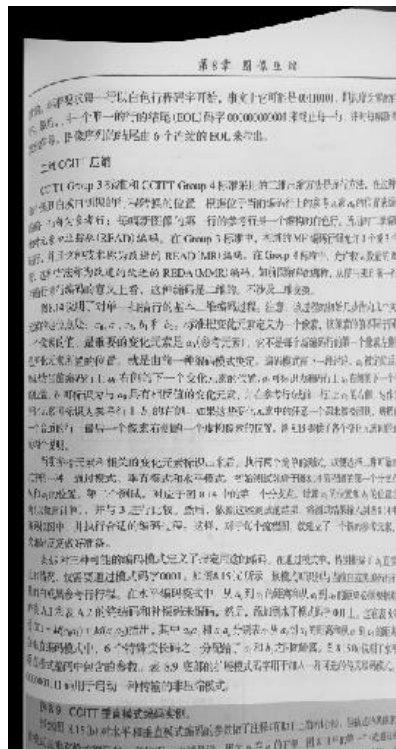


接下来找线，从最左边的点出发，顺次找每个点的右邻居，直到某个点没有右邻居为止，寻找过程中有角度条件和长度条件，少于 8 个点的线舍去，角度变化过大的线舍去。

将这些点按长度排序，再去除有重合部分的线，这样就能优先舍弃短线。最后从这些线中取出最长的若干条，下面是取出的线条。



可以看出找到的线条上的点确实是在同一行文字上，但线条的覆盖范围太小，没有均匀分布在整页上。用与第二部分相同的方法找出这些线条上的点对应的目标图中点坐标，输入 TPS 薄板样条插值，结果如下图。



2、拼接图片二值化

七、总结

在大作业布置后的一个星期多的时间内其实我是非常焦虑的，因为对如何将书页图片展平没有任何思路，在网上也找不到具体方法，一直在思考到底该怎么办，可以说这次大作业很好地锻炼了我的抗压能力。最后决定用 TPS 试试，因为数值分析课程大作业介绍了 TPS 算法，试过之后发现效果可以接受，于是才稍微松了一口气。但随着项目的推进，又遇到了一些困难，比如用 RANSAC 算法拼接图片时结果往往有很严重的畸变，于是我把这个项目放了几天，闲暇的时候便思考该怎么解决问题。最后想到根据这个问题的特殊性来修改 RANSAC 算法，将想法付诸实践后，没想到效果出其意料的好，之后的进程便比较顺利了。

最后自动选点矫正的结果虽然比不上手动选点的结果，但感觉在实现这个功能的过程中领悟到了数字图像处理的神奇之处，完成后也是很有成就感，毕竟在两周以前还认为这是不可能的任务。

在整个过程中也学到了很多新的知识，比如 sift 算法，多频段融合，RANSAC 算法，TPS 薄板样条插值等，将课上学过的一些知识应用于实践，加深了我的理解，锻炼我解决问题的能力 and 抗压能力，可以说是收获满满了。

八、参考资料

Bookstein, F. L. J. I. T. o. p. a., & intelligence, m. (1989). Principal warps: Thin-plate splines and the decomposition of deformations. *11*(6), 567-585.

Cheung, W., & Hamarneh, G. (2007). *N-sift: N-dimensional scale invariant feature transform for matching medical images*. Paper presented at the 2007 4th IEEE International Symposium on Biomedical Imaging: From Nano to Macro.

多频段融合方法——图像拼接: <https://blog.csdn.net/ccblogger/article/details/70665552>

SIFT 特征点提取: <https://blog.csdn.net/lingyunxianhe/article/details/79063547>

基于薄板样条插值图像配准的 Matlab 实现: <http://www.voidcn.com/article/p-pquporfk-du.html>