

2013 高教社杯全国大学生数学建模竞赛

承 诺 书

我们仔细阅读了《全国大学生数学建模竞赛章程》和《全国大学生数学建模竞赛参赛规则》(以下简称为“竞赛章程和参赛规则”,可从全国大学生数学建模竞赛网站下载)。

我们完全明白,在竞赛开始后参赛队员不能以任何方式(包括电话、电子邮件、网上咨询等)与队外的任何人(包括指导教师)研究、讨论与赛题有关的问题。

我们知道,抄袭别人的成果是违反竞赛章程和参赛规则的,如果引用别人的成果或其他公开的资料(包括网上查到的资料),必须按照规定的参考文献的表述方式在正文引用处和参考文献中明确列出。

我们郑重承诺,严格遵守竞赛章程和参赛规则,以保证竞赛的公正、公平性。如有违反竞赛章程和参赛规则的行为,我们将受到严肃处理。

我们授权全国大学生数学建模竞赛组委会,可将我们的论文以任何形式进行公开展示(包括进行网上公示,在书籍、期刊和其他媒体进行正式或非正式发表等)。

我们参赛选择的题号是(从 A/B/C/D 中选择一项填写): B

我们的参赛报名号为(如果赛区设置报名号的话):

所属学校(请填写完整的全名): 西南交通大学

参赛队员(打印并签名): 1. 唐隽玉

2. 胡文伟

3. 孙雪清

指导教师或指导教师组负责人(打印并签名): 徐跃良

(论文纸质版与电子版中的以上信息必须一致,只是电子版中无需签名。以上内容请仔细核对,提交后将不再允许做任何修改。如填写错误,论文可能被取消评奖资格。)

日期: 2013 年 9 月 16 日

赛区评阅编号(由赛区组委会评阅前进行编号):

2013 高教社杯全国大学生数学建模竞赛

编 号 专 用 页

赛区评阅编号（由赛区组委会评阅前进行编号）：

赛区评阅记录（可供赛区评阅时使用）：

评阅人										
评分										
备注										

全国统一编号（由赛区组委会送交全国前编号）：

全国评阅编号（由全国组委会评阅前进行编号）：

碎纸片的拼接复原

摘 要

本文运用左右边界匹配、图片特征匹配、上下边界匹配等方法研究单页打印纵切纸片、单页打印横、纵切纸片以及双页打印横、纵切纸片的拼接与复原问题。

针对问题一，首先对图像进行数据处理，读取图片的灰度信息，构建灰度矩阵，并将灰度矩阵转化为 0-1 矩阵，从而将二维图片数值化。接着，提取出 0-1 矩阵的第一列与最后一列，存储在图片的左右边界矩阵中，通过建立两张图片的左右边界匹配度模型，探究图片的左右邻接关系。计算结果为：汉字图片从左到右依次为：008、014、012、015、003、010、002、016、001、004、005、009、013、018、011、007、017、000、006，英文的排序结果为：003、006、002、007、015、018、011、000、005、001、009、013、010、008、012、014、017、016、004。

问题二，采用二层筛选的方法，第一层做行位置筛选，读取图片的前 100 个像素行，存入图片的特征列向量中，并将此列向量作为行特征的唯一标识，建立图片的特征匹配模型，将列向量元素差异最小的图片聚类，中文确定出 15 类，英文归为 16 类。然后通过人为干预，实现类的合并，使每类中的图片个数相同，将中英文都聚成 11 类，每一类包含 19 张图片。构建行内图片的左右边界匹配模型，最终确定出每类内部图片的排序；第二层做列位置筛选，建立每行上下边界匹配模型，得出在各行的上下位置序列，经过两层筛选，得出原文件图片序列。最后，视人工干预后的最终结果为正确答案，检验未加入人工干预计算机排序结果，得到中文的拼接正确率为 90.4%，英文的拼接正确率为 65.1%。

对于问题三，建立两次特征匹配模型将图片聚类，即首先任取一碎片的一面依次与其他碎片的两个面分别作第一次特征匹配，寻得与该面特征匹配程度高的另一碎片的一面，再将这两个碎片的另一面做第二次特征匹配，在两者匹配很好的前提下，探求出两碎片的确定面属于同一类。加入人工干预，对类的个数降维，并保证每类中图片的数量相同。再利用问题二中的模型构建方法，通过左右边界匹配模型的求解、上下边界匹配模型的构建方法，完成了本问的研究。最后，我们从问题二的模型多增加一层特征匹配约束可得到问题三模型这一角度出发，得出了模型三的拼接精度更高这一结论。

本文综合各种匹配方法，根据问题的深入，对匹配模型加以不断的改进，结合 matlab 编程、word 拼图等手段，对碎纸片的拼接复原做了逐步深入分析，并给出了基于边界灰度、图片行特征灰度的匹配模型。在文章的最后对模型的适用范围做出了推广，在实际应用中有一定的参考价值。

关键词：左右边界匹配 特征匹配 上下边界匹配 matlab 两层筛选

一、问题重述

破碎文件的拼接在司法物证复原、历史文献修复以及军事情报获取等领域都有着重要的应用。传统上，拼接复原工作需由人工完成，准确率较高，但效率很低。特别是当碎片数量巨大，人工拼接很难在短时间内完成任务。随着计算机技术的发展，人们试图开发碎纸片的自动拼接技术，以提高拼接复原效率。请讨论以下问题：

1. 对于给定的来自同一页印刷文字文件的碎纸机破碎纸片（仅纵切），建立碎纸片拼接复原模型和算法，并针对附件 1、附件 2 给出的中、英文各一页文件的碎片数据进行拼接复原。如果复原过程需要人工干预，请写出干预方式及干预的时间节点。复原结果以图片形式及规定的表格形式表达。

2. 对于碎纸机既纵切又横切的情形，请设计碎纸片拼接复原模型和算法，并针对附件 3、附件 4 给出的中、英文各一页文件的碎片数据进行拼接复原。如果复原过程需要人工干预，请写出干预方式及干预的时间节点。复原结果表达要求同上。

3. 上述所给碎片数据均为单面打印文件，从现实情形出发，还可能有双面打印文件的碎纸片拼接复原问题需要解决。附件 5 给出的是一页英文印刷文字双面打印文件的碎片数据。请尝试设计相应的碎纸片拼接复原模型与算法，并就附件 5 的碎片数据给出拼接复原结果，结果表达要求同上。

二、问题分析

碎片的拼接复原，通常的做法是人工识别碎片边缘的字迹断线、和理解碎片内文字含义，这样利用人工智能的方法虽然准确度高，但是当碎片的数量很大时，人工的效率就显得低，而且出错率会明显提高；而计算机拼接与复原图像，虽不及人工识别智能，但能充分发挥其运算量大，运算速度快的特点。

故本问题的目标就是利用附件中给的碎片数据，分单页纵切，单页横纵切，双页打印横纵切三种情况，把拼接复原问题抽象成一个明确完整的数学模型，利用计算机，并加以人工干预，复原出原图表。

问题一的分析

问题一要求仅考虑单面纵切，建立来自同一页印刷文字文件的碎纸机破碎的纵切纸片拼接复原模型和算法。通过对附件 1 和附件 2 给出的碎片数据图的观察，发现本题的碎片图像具有相对文字（汉字、英文）方向纵向规则剪开的特征，所以不适合基于碎片的边缘线建模，也不适合基于两幅图片的重合度建模。我们可以根据打印文件的每行文件具有前后连续性，考虑先从读取文件数据入手，存储每幅图片对应的灰度值矩阵。依靠得到的灰度值矩阵转化为 0-1 二值矩阵，并利用相邻接左右边界差异不大这一特性作为依据来建立左右边界匹配模型，来解决此问题，复原出图片的原始序列。

问题二的分析

此题加入了横向切割，使得切割方式更加多样化和更接近实际。它相对于第一问而言，图片的信息量更小，图片的个数增多了一倍。图片总体不仅在纵向具有无序性，而且在横向也具有无序性。若仅采用问题一中的方法，定位约束太少，可能会出现一个图片与多个图片最小差异度相等，导致该图片与多个图片相联系，从而增加问题求解的难

度。通过观察图片的平行切割特点，发现来自原文件同一行的文字切割后的图片一般在相同的行位置上。所以可以考虑，先进行行位置筛选，通过构建图片的特征列向量作为唯一标识，建立特征匹配模型，得到具有相同行特征的图片，聚成同一类。考虑到每类包含的图片个数不一致，可加入人工干预，对类的个数降维，使得行集合包含的碎片个数一致。而利用左右边界匹配模型可以确定同一行的图片的序列；可采用相同的原理，建立上下边界匹配模型来解决纵向图片的定序问题。这样一来，可以拼接出本问的原文文件，完成问题二的求解。

问题三的分析

问题三在前两问的基础上，加入了双面打印这一条件。本问中图片的个数相较于问题二增大了一倍，达 $2 \times 11 \times 19 = 418$ 个，较前两问复杂度最高。由于从单面看问题二和问题三没有任何区别，所以可以采取相似的方法对问题三求解。但我们思考总结出如下两方面：一方面不能思维定势，也就是说所有编号中带有 a 的图不一定都来自同一面，即有可能是碎纸片的正面也有可能是碎纸片的反面。另一方面如果采用问题二中相同的处理方法对附件 5 中所有的图片排序的话，可能会发生一个图片的匹配图片过多，或者出现将一个碎纸片的正反面归为同一类的错误。综合以上两方面的思考，问题三的求解过程的特点在于：先对一张碎纸片构建其对应的特征匹配模型，若得到另外一张碎纸片与这张碎纸片匹配，则随后对它们的反面进行匹配以检验。

三、模型假设

1. 假设附件中每张碎纸片都是大小相等的矩形，切割边缘光滑；
2. 假设附件中编号为 000 的图片为第一张图片，编号为 001 的图片为第二张图片，依此类推；
3. 假设附件中每张图片无倾斜，即底边水平；
4. 假设附件中的每张图片是无噪的，仅考虑图像的拼接无须考虑图像的修补；
5. 假设每一附件为同一页纸的碎片数据；
6. 假设包含 000a 图片的那页为原文件的正面

四、符号说明与名称解释

4.1 符号说明

- $A^{(k)}$ ：第 k 张图片的灰度值矩阵
 $a_{i,j}^{(k)}$ ：第 k 张图片的灰度值矩阵的第 i 行第 j 列的元素
 $C^{(k)}$ ：第 k 张图片的灰度值矩阵转化的 0-1 矩阵
 $c_{i,j}^{(k)}$ ：第 k 张图片 0-1 矩阵的第 i 行第 j 列的元素
 $B^{(k)}$ ：第 k 张图片左、右边界线上的 0-1 边界矩阵
 $P_{k,s}^r$ ：第 k 张图片右边界与第 s 张图片的左边界的边界匹配值
 $P_{k,s}^l$ ：第 k 张图片左边界与第 s 张图片的右边界的边界匹配值
 P_k^* ：第 k 张图片左右边界匹配时最优的匹配值
 $D^{(k)}$ ：存入特征灰度信息的特征列向量
 $d_i^{(k)}$ ：第 k 张图片灰度信息特征列向量
 $W_{k,s}$ ：反映图片 k 及图片 s 的特征信息吻合程度的特征值

4.2 名称解释

1. 原文件：每个附件中所有图片拼接复原图；

- 2.图片行：以附件中各个图片为单位组成的行；
- 3.文字行：以图片内部文字为单位组成的行；
- 4.像素行：图片内部像素矩阵的行；
- 5.行集合：具有相同行特征的碎片组成的图片行。

五、模型的建立与求解

5.1 问题一的模型建立与求解

问题一要求拼接复原来自同一页纵切的破碎纸片。这个问题仅在纵向的维度对碎纸片的拼接复原提出了要求，对此本文从以下三个步骤进行回答：

步骤一：读取每张图片文件的数据，其目的是将附件中给的 bmp 格式的碎纸片图以灰度值矩阵的形式存储。再将灰度值矩阵转化为 0-1 矩阵，来得到模型的数据基础；

步骤二：基于上述 0-1 矩阵，提取每幅图片左右边界的 0-1 值，建立左右边界匹配模型，确定出图片的序列；

步骤三：根据上面的步骤，将附件图片拼接，以图片和表格形式展现。

5.1.1 图像的数据处理

Step1.灰度值矩阵的获取^[1]

附件中无论印有汉字还是英文的碎纸片均以 bmp 格式的图片形式给出。先将附件中的图片以元胞矩阵的形式存入 matlab 中

为建立模型，必须得到数字依据。

所以用 matlab 的 imread 函数读取图片的灰度信息，将第 k 张图片的灰度信息分别存入灰度值矩阵 $A^{(k)}$ 中 ($k=1,2\cdots 19$)：

$$A^{(k)} = \begin{bmatrix} a_{1,1}^{(k)} & a_{1,2}^{(k)} & \cdots & a_{1,72}^{(k)} \\ a_{2,1}^{(k)} & a_{2,2}^{(k)} & \cdots & a_{2,72}^{(k)} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m,1}^{(k)} & a_{m,2}^{(k)} & \cdots & a_{m,n}^{(k)} \end{bmatrix}$$

其中，第 k 个图片的灰度信息以 0~255 的灰度值存储在矩阵 $A^{(k)}$ 中，颜色越深，灰度值越大。

Step2. 0-1 矩阵的建立

由于 matlab 在计算时，为防止灰度值溢出，会将值限制在 0~255 的范围内。在此模型的计算中，为保证灰度匹配模型中绝对值的和不受这个约束的影响，同时简便计算，需将灰度值矩阵 $A^{(k)}$ 进行转化为 0-1 矩阵 $C^{(k)}$ ($k=1,2\cdots 19$)，具体转化操作如下：

若 $A^{(k)}$ 中某个元素灰度值 $a_{i,j}^{(k)}$ 小于 255，则 $C^{(k)}$ 中相同位置的元素值 $c_{i,j}^{(k)}$ 记为 0，否则记为 1。即：

$$c_{i,j}^{(k)} = \begin{cases} 0, & a_{i,j}^{(k)} < 255 \\ 1, & a_{i,j}^{(k)} = 255 \end{cases} \quad (5.1)$$

于是，建立了 0-1 矩阵 $C^{(k)}$ ：

$$C^{(k)} = \begin{bmatrix} c_{1,1}^{(k)} & c_{1,2}^{(k)} & \cdots & c_{1,n}^{(k)} \\ c_{2,1}^{(k)} & c_{2,2}^{(k)} & \cdots & c_{2,n}^{(k)} \\ \vdots & \vdots & \ddots & \vdots \\ c_{m,1}^{(k)} & c_{m,2}^{(k)} & \cdots & c_{m,n}^{(k)} \end{bmatrix}$$

Step3: 获取左、右边界矩阵

根据 0-1 矩阵 $C^{(k)}$ ，将第 k 张图片的左、右边界处的元素分别存于 $B^{(k)}$ ($k=1,2\cdots 19$) 矩阵的第一列、第二列中。得到保存左、右边界线上的 0-1 值的 $B^{(k)}$ 边界矩阵；

$$B^{(k)} = \begin{bmatrix} c_{1,1}^{(k)} & c_{1,n}^{(k)} \\ c_{2,1}^{(k)} & c_{2,n}^{(k)} \\ \vdots & \vdots \\ c_{m,1}^{(k)} & c_{m,n}^{(k)} \end{bmatrix}$$

5.1.2 边界匹配模型的建立

为了得到每个附件中各幅图片的邻接关系，采用边界匹配法。它是基于纵向规则切割特性：两相邻图片中第一幅图片的右边界上的文字和第二幅图片左边界上文字多数来自同一个汉字或英文字母，即两邻接图片的边界的差异度小。

边界匹配模型的建立具体步骤如下：

1. 右边界匹配模型的构建

将第 k 个图片的右边界与第 s 张 ($s=1,2\cdots 19$ 且 $s \neq k$) 图片的左边界进行右边界匹配，即求第 k 个图片边界矩阵的第二列与第 s 张边界矩阵的第一列对应行元素的差，再求差的绝对值的和—— $P_{k,s}^r$ ：

$$P_{k,s}^r = \sum_{i=1}^m |c_{i,n}^{(k)} - c_{i,1}^{(s)}| \quad (5.2)$$

将第 k 个图片的右边界依次与其余的任意一张图片的左边界进行右边界匹配，得到 n 个值： $P_{k,1}^r, P_{k,2}^r, \cdots, P_{k,n}^r$ 。通过比较，取这 n 个值中最小值，作为右边界匹配值。

$$P_{k,s}^r = \min\{P_{k,1}^r, P_{k,2}^r, \cdots, P_{k,n}^r\} \quad (5.3)$$

2. 左边界匹配模型的构建

将第 k 个图片的左边界分别与第 s 张 ($s=1,2\cdots 19$ 且 $s \neq k$) 图片的右边界进行左边界匹配，即求第 k 个图片边界矩阵的第一列与第 s 张边界矩阵的第二列对应行元素的差，再求差的绝对值的和—— $P_{k,s}^l$ ：

$$P_{k,s}^l = \sum_{i=1}^m |c_{i,1}^{(k)} - c_{i,n}^{(s)}| \quad (5.4)$$

将第 k 个图片的左边界依次与其余的任意一张图片的右边界进行左边界匹配，得到 n 个值： $P_{k,1}^l, P_{k,2}^l, \cdots, P_{k,n}^l$ 。通过比较，取这 n 个值中最小值，作为左边界匹配值。

$$P_{k,s}^l = \min\{P_{k,1}^l, P_{k,2}^l, \cdots, P_{k,n}^l\} \quad (5.5)$$

3. 最佳边界匹配模型的建立

取第 k 个图片，先与任意一张图片 s ($s=1,2\cdots 19$ 且 $s \neq k$) 图片依次进行右边界匹配，求得 $P_{k,s}^r$ ，再与任意一张图片 s ($s=1,2\cdots 19$ 且 $s \neq k$) 图片依次进行左边界匹配，得 $P_{k,s}^l$ ，取两值之间的最小值：

$$P_k^* = \min\{P_{k,s}^l, P_{k,s}^r\} \quad (5.6)$$

P_k^* 对应的匹配方式即为第 k 张图片与第 s 张图片的最佳匹配方式。若 $P_k^* = P_{k,s}^r$ ，则说明第 k 张图片的右边接于第 s 张图片的左边，记做 $k \rightarrow s$ ； $P_k^* = P_{k,s}^l$ 时，说明第 k 张图片的左边与第 s 张图片的右边相连，记做 $s \rightarrow k$ 。

综上，我们的左右边界匹配模型可以总结为：

$$P_k^* = \min\{P_{k,s}^l, P_{k,s}^r\}$$

$$\begin{cases} P_{k,s}^r = \min\{P_{k,1}^r, P_{k,2}^r, \dots, P_{k,n}^r\} \\ P_{k,s}^r = \sum_{i=1}^m |c_{i,n}^{(k)} - c_{i,1}^{(s)}| \\ P_{k,s}^l = \min\{P_{k,1}^l, P_{k,2}^l, \dots, P_{k,n}^l\} \\ P_{k,s}^l = \sum_{i=1}^m |c_{i,1}^{(k)} - c_{i,n}^{(s)}| \end{cases} \quad (5.7)$$

5.1.4 模型的求解

1.求解算法

Step1 取附件中第*i*张碎片,将其依次与第 1 张碎片,第 2 张碎片,第*i*-1张碎片,...,第*i*+1张碎片, ..., 第 *n* 张碎片进行右边界匹配,得到右边界匹配值;

Step2 取附件中第*i*张碎片,将其依次与第 1 张碎片,第 2 张碎片,第*i*-1张碎片,...,第*i*+1张碎片, ..., 第 *n* 张碎片进行左边界匹配,得到左边界匹配值;

Step3 比较右边界匹配值与左边界匹配值的大小,选择两者之间的最小值对应的匹配方式,将两张图片按匹配方式结合,视为一体。

Step4 重复进行 Step1→Step3,直至确定出附件原文件的图片序列。

2.编程实现

我们利用 matlab 求解 (具体程序见附录),得到如下:

附件 1,2 分别包含了 19 张图片,通过 matlab 中“imread”命令,得到灰度值矩阵 $A^{(k)}$,考虑到本题中 $A^{(k)}$ 的维度为 1980×72,且这是中间过程的结果,故不予以展示。

同样地,由于 0-1 矩阵 $C^{(k)}$ 的大小为 1980×72,边界矩阵 $B^{(k)}$ 的维度为本题中它的维数为 1980×2,不在正文中显示矩阵的具体值。

本文仅列举右边界匹配值 $P_{k,s}^r$ 与邻接关系,附件 1、2 的情况一并显示在表 1 中:

表 1 问题一的邻接关系与 $P_{k,s}^*$ 值

附件 1 (汉字)		附件 2 (英文)	
邻接关系	$P_{k,s}^r$ 值	邻接关系	$P_{k,s}^r$ 值
000→006	282	000→005	281
001→004	284	001→009	228
002→016	333	002→007	135
003→010	332	003→006	172
004→005	279	004→003	101
005→009	296	005→001	185
006→008	169	006→002	185
007→017	318	007→015	212
008→014	238	008→012	180
009→013	375	009→013	200
010→002	330	010→008	198
011→007	259	011→000	145
012→015	212	012→014	230
013→018	71	013→010	178
014→012	262	014→017	169
015→003	372	015→018	46
016→001	336	016→004	143
017→000	322	017→016	195
018→011	327	018→011	192

复原结果：附件 1 对应的复原表格见表 2，附件 2 对应的复原表格见表 3。按照表 2、3 的序列，把图片设置成相同的高和宽的统一大小，将图片复制，粘贴进 word，即用 word 拼图，得到复原图，依次见附录 1.4、1.5。.

表 2 附件 1（汉字）复原的碎片序号

008	014	012	015	003	010	002	016	001	004	005	009	013	018	011	007	017	000	006
-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

表 3 附件 2（英文）复原的碎片序号

003	006	002	007	015	018	011	000	005	001	009	013	010	008	012	014	017	016	004
-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

5.1.4 模型结果的分析

本问中，建立边界匹配模型，利用 matlab 编程进行求解，可以得到附件 1（汉字）、附件 2（英文）的图片排列顺序，在 word 中拼出图片见附录 2.1、2.2。观察图片，通过观察碎片边缘的字迹断线的连接、理解碎片内文字含义，检验了拼接复原结果正确性。也说明边界匹配模型在不需要人工干预的情况下很好地解决了问题一。

5.2 问题二的模型建立与求解

问题二要求在碎纸机既纵切又横切的情形下，对碎纸片进行拼接复原。我们认为由以下四个步骤组成：

步骤一：进行模型的准备，用问题一的方法对图像进行预处理，分别构建反映中英文文章行特征的特征向量以及确定需要扫描像素行的行数；

步骤二：通过分别建立特征匹配模型，左右边界匹配模型，上下边界匹配模型三个模型，完成单页打印横纵切纸片匹配模型的构建；

步骤三：对模型进行求解。特别的，特征匹配模型求解后加入人工干预；

步骤四：对求解结果进行分析。

5.2.1 模型的准备

1.图像的预处理

对于图像的预处理，我们采用问题一中的方法，首先用 matlab 读取每张图片的灰度信息，再将灰度信息转换为 0-1 矩阵。

2.构建特征灰度条向量

(1) 构建中文特征灰度条向量

特征灰度条是指记录图片中文字的横方向信息的列向量 $D^{(k)}$ ，建造特征灰度条的方法为：对于预处理后的图像，建造一个与碎片的图像行数一致的列向量 $D^{(k)}$ 。对图像中像素行进行扫描，若此行中有像素值为 0 的点，则将列向量 $D^{(k)}$ 中相同行处的值设为 0，否则设为 1。图 000.bmp 的特征灰度条如图 2 所示

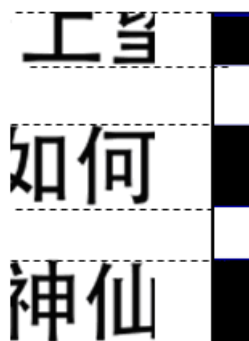


图 2 图 000.bmp 及其灰度条

$$d_i^{(k)} = \begin{cases} 0, & \text{第}k\text{张图片中第}i\text{行出现了}0 \\ 1, & \text{第}k\text{张图片中第}i\text{行全为}1 \end{cases} \quad (5.8)$$

特征灰度条的列向量 $D^{(k)}$ 为:

$$D^{(k)} = (d_1^{(k)}, d_2^{(k)}, \dots, d_m^{(k)})^T$$

根据此法,便可得到每张碎纸片的特征灰度条。若某两张碎片的灰度条相似程度达到精度要求,则它们就具有相同的图像行特征,位于原文件的同一行。

(2) 构建英文特征灰度条向量

英文与汉字的结构不同。各个英文字母在上、中、下三格的分布情况不同。有些图片中的某一行只含有中格字母,如附件四中的 011.bmp 图片第二行只含有 o、c、r 和 a,而与它相邻图片的对应行同时含有中上格、中下格字母,例如位于 011.bmp 图片右边的 154.bmp 图片第二行含有中上格字母“t”。如果用汉字特征灰度条的构建方法来制作这两个英文图片的特征灰度条,则 154.bmp 的特征灰度条在文字第二行区域的黑色段会比 011.bmp 的要长。在第一轮筛选过程中,会因为这个黑色段长度的差异而将两者归入不同的行集合中。为了使第一轮筛选结果尽量准确,必须要避免这种情况。通过分析英文字母的结构特性,发现所有的英文字母的主体部分都在中格,所以在构建特征灰度条时,可以只考虑每行英文字母的中格部分,这样就规避了因字母在上、中、下三格分布情况不同而造成筛选不准确的情况。具体方法如下:

将预处理后的附件四图像,建造一个与碎片 0-1 矩阵行数相同的列向量。逐次扫描图像的像素行。若某行元素之和小于 M ,说明该像素行中黑点较多,是英文字母的主体部分,位于此文字行的中格,则在列向量中相同行处的元素记为 0,否则记为 1。多次试验后,我们选取合适的值为 56。以 011.bmp 图片为例构建的特征灰度条如图 3 所示:

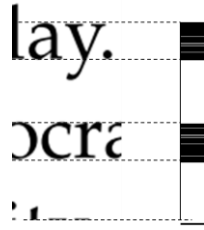


图 3 图 011.bmp 及其灰度条

$$d_i^{(k)} = \begin{cases} 0, & \text{第}k\text{张图片中第}i\text{行元素之和} < M \\ 1, & \text{第}k\text{张图片中第}i\text{行元素之和} \geq M \end{cases} \quad (5.9)$$

特征灰度条的列向量 $D^{(k)}$ 为:

$$D^{(k)} = (d_1^{(k)}, d_2^{(k)}, \dots, d_m^{(k)})^T$$

根据此法,便可得到附件四中每张碎纸片的特征灰度条。若某两张碎片的灰度条相似程度达到精度要求,则它们就具有相同的图像行特征,位于原文件的同一行。

3. 确定扫描像素行的行数^[1]

扫描一幅图片的像素点,其自然的思路就是对第 k 张图像的每一行进行扫描,得到每行的像素值。但在构造特征灰度条矩阵的时候,发现了一些特殊情况,如下图所示:



图 1 特殊情况举例

编号为 170 和编号为 205 的图片人为可以判断他们是左右相邻的，两幅图刚好位于一篇文章的最后三行文字的位置，其中编号为 170 的图片位置靠前，所以图片截到了第三行末尾的句号，但编号为 205 的图片位置靠后，图中第三文字行已经到了换行符之后，是空白的。两者的特征灰度条会在下边界处出现明显偏差，原本应归入同一集合的两张碎片会被分开。

这种情况的出现，会加重计算的复杂度，而且造成结果冗余而不准确。为了得到准确的结果，必须要削弱这种情况所带来的干扰。结合附件中图片特征，发现文字大多集中在图片三分之二高度的区域内，所以，构建特征灰度条时，没必要每一行都扫描到。为了得到合适的扫描行数，分别试验了扫描前 100 行、前 120 行和前 140 行的方案，得到的效果如下（附件三和附件四中每张图片总共有 180 个像素行）：

表 4 关于图像扫描行数确定的几次尝试

扫描像素行数	行集合的个数
100	15
120	16
140	19

通过观察结果可以看出，扫描前 100 行得到的碎片总行数是最少的，很接近题目中切割源文件的行数 11。所以，本文仅对每张图像的前 100 个像素行进行扫描，并提高匹配的精度要求，这样更有利于解决问题。

5.2.2 建立横纵切纸片匹配模型

(1) 建立特征匹配模型

将碎片 k 与碎片 s 进行特征比较 ($s = 0, 1, \dots, 208$ 且 $s \neq k$)，即求碎片 k 的特征列向量 $D^{(k)}$ 与碎片 s 的特征列向量 $D^{(s)}$ 对应元素的差的绝对值，再求和，得到特征值 $W_{k,s}$ ：

$$W_{k,s} = \sum_{i=1}^m |d_i^{(k)} - d_i^{(s)}| \quad (5.10)$$

考虑到每个汉字或者每个英文字母结构的差异性，位于同一行文字的高度可能会出现微小的偏差，很难出现特征灰度条相同（即 $W_{k,s} = 0$ ）的情况，若取 $W_{k,s} = 0$ 作为判断原则，那么原本位于同一行的两张图片可能因为这微小的偏差而归于不同的行集合中。

取一个合适小的置信区间 $[a, b]$ ，若 $W_{k,s} \in [a, b]$ ，则认为碎片 k 与碎片 s 来自原文件的同一行。

(2) 建立左右边界匹配模型

本问中的左右边界匹配模型相对于第一问中的边界匹配模型而言，差异性在于问题一中的边界匹配模型是 19 个纵向大长条，信息量大；而本问中是 19 个纵向小长条，信息量小。

也就是说，问题一中的左右边界矩阵为

$$B^{(k)} = \begin{bmatrix} c_{1,1}^{(k)} & c_{1,72}^{(k)} \\ c_{2,1}^{(k)} & c_{2,72}^{(k)} \\ \vdots & \vdots \\ c_{19801}^{(k)} & c_{198072}^{(k)} \end{bmatrix}$$

而问题二中的左右边界矩阵为

$$B^{(k)} = \begin{bmatrix} c_{1,1}^{(k)} & c_{1,72}^{(k)} \\ c_{2,1}^{(k)} & c_{2,72}^{(k)} \\ \vdots & \vdots \\ c_{180,1}^{(k)} & c_{180,72}^{(k)} \end{bmatrix}$$

其余模型的建立步骤同问题一，即：

1. 构建右边界匹配模型；
2. 构建左边界匹配模型；
3. 构建最佳匹配模型。

(3) 建立上下边界匹配模型

本模型的基本思路与第一问中边界匹配模型的构建一致，因为问题一中是一长列的边界匹配，是纵向的边界匹配，本模型是一长行的边界匹配，是横向的边界匹配。但是不同之处在于：问题一中匹配的 19 条纵列的左右边界匹配模型，而本模型为 11 条行的上下边界匹配模型。将第 k 张图片的上、下边界处的元素分别存于 $E^{(k)}$ ($k=1,2\cdots 11$) 矩阵的第一行、第二行中。即上下边界匹配模型中第 k 行的上下边界矩阵为：

$$E^{(k)} = \begin{bmatrix} c_{1,1}^{(k)} & c_{1,2}^{(k)} & \cdots & c_{1,n}^{(k)} \\ c_{m,1}^{(k)} & c_{m,2}^{(k)} & \cdots & c_{m,n}^{(k)} \end{bmatrix}$$

故模型的构建如下：

1. 上边界匹配模型的构建

将第 k 行的上边界与第 s ($s=1,2\cdots 11$ 且 $s \neq k$) 行的下边界进行上边界匹配，即求第 k 行的边界矩阵的第一行与第 s 行的边界矩阵的第二行对应列元素的差，再求差的绝对值的和—— $Q_{k,s}^u$ ：

$$Q_{k,s}^u = \sum_{j=1}^n |c_{1,j}^{(k)} - c_{m,j}^{(s)}| \quad (5.11)$$

将第 k 行的上边界依次与其余的任意一行的下边界进行上边界匹配，得到 n 个值： $Q_{k,1}^u, Q_{k,2}^u, \cdots, Q_{k,n}^u$ 。通过比较，取这 n 个值中最小值，作为上边界匹配值。

$$Q_{k,s}^u = \min\{Q_{k,1}^u, Q_{k,2}^u, \cdots, Q_{k,n}^u\} \quad (5.12)$$

2. 下边界匹配模型的构建

将第 k 行的下边界与第 s ($s=1,2\cdots 11$ 且 $s \neq k$) 行的上边界进行下边界匹配，即求第 k 行边界矩阵的第二行与第 s 行边界矩阵的第一列对应列元素的差，再求差的绝对值的和—— $Q_{k,s}^d$ ：

$$Q_{k,s}^d = \sum_{j=1}^n |c_{m,j}^{(k)} - c_{1,j}^{(s)}| \quad (5.13)$$

将第 k 个行的下边界依次与其余的任意一行的上边界进行下边界匹配，得到 n 个值： $Q_{k,1}^d, Q_{k,2}^d, \cdots, Q_{k,n}^d$ 。通过比较，取这 n 个值中最小值，作为下边界匹配值。

$$Q_{k,s}^d = \min\{Q_{k,1}^d, Q_{k,2}^d, \dots, Q_{k,3}^d\} \quad (5.14)$$

3.最佳边界匹配模型的建立

取第 k 行, 先与任意一行 s ($s=1,2,\dots,11$ 且 $s \neq k$) 依次进行上匹配, 求得 $Q_{k,s}^u$, 再与任意一行 s ($s=1,2,\dots,11$ 且 $s \neq k$) 依次进行下匹配, $Q_{k,s}^d$, 取两值之间的最小值:

$$Q_k^* = \min\{Q_{k,s}^u, Q_{k,s}^d\} \quad (5.15)$$

Q_k^* 对应的匹配方式即为第 k 行与第 s 行的最佳匹配方式。若 $Q_k^* = Q_{k,s}^u$, 则说明第 k 行上边接于第 s 行的下边; $Q_k^* = Q_{k,s}^d$ 时, 说明第 k 行的下边与第 s 行的上边相连。

综上, 我们构建的横纵切纸片匹配模型为:

$$Q_k^* = \min\{Q_{k,s}^u, Q_{k,s}^d\} \quad \left\{ \begin{array}{l} W_{k,s} = \sum_{i=1}^m |d_i^{(k)} - d_i^{(s)}| \in [a, b] \\ Q_{k,s}^u = \min\{Q_{k,1}^u, Q_{k,2}^u, \dots, Q_{k,n}^u\} \\ Q_{k,s}^u = \sum_{j=1}^n |c_{1,j}^{(k)} - c_{m,j}^{(s)}| \\ Q_{k,s}^d = \min\{Q_{k,1}^d, Q_{k,2}^d, \dots, Q_{k,n}^d\} \\ Q_{k,s}^d = \sum_{j=1}^n |c_{m,j}^{(k)} - c_{1,j}^{(s)}| \end{array} \right. \quad (5.16)$$

5.2.3 模型的求解

1.求解步骤

Step1 取附件中第 i 张碎片, 将其依次与第 1 张碎片, 第 2 张碎片, 第 $i-1$ 张碎片, ..., 第 $i+1$ 张碎片, ..., 第 n 张碎片进行特征匹配, 将匹配成功的碎片存入同一行;

Step2 重复 **Step1**, 得到具有相同行特征的行集合;

Step3 人工识别碎片边缘的字迹断线、理解碎片内文字含义的方式对相邻接的图片, 即加入人工干预, 将得到的类的个数降维;

Step4 取同一行中第 i 张碎片, 将其依次与第 1 张碎片, 第 2 张碎片, 第 $i-1$ 张碎片, ..., 第 $i+1$ 张碎片, ..., 第 n 张碎片先进行右边界匹配, 得到右边界匹配值, 再依次进行左边界匹配, 得到左边界匹配值。比较右边界匹配值与左边界匹配值的大小, 选择两者之间的最小值对应的匹配方式, 将两张图片按匹配方式结合, 视为一体。

Step5 重复进行 **Step4**, 直至确定出每行内部图片的排列顺序。

Step6 取第 i 行, 将其依次与第 1 行, 第 2 行, 第 $i-1$ 行, ..., 第 $i+1$ 行, ..., 第 n 行先进行上边界匹配, 得到上边界匹配值, 再依次进行下边界匹配, 得到下边界匹配值。比较上边界匹配值与下边界匹配值的大小, 选择两者之间的最小值对应的匹配方式, 将两张图片按匹配方式结合, 视为一体。

Step7 重复进行 **Step6**, 直至确定出每行的上下位置, 从而得到图片的原始序列。

2. 算法实现

(1) 结合特征匹配模型, 利用 matlab 编程 (见附录), 得到每行集合的碎片个数, 附件 3(中文)对应的 15 个行集合中碎片个数, 如表 5 所示:

表 5 15 行中每行的碎片个数

行集合的编号	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
碎图片的个数	19	19	19	18	15	19	19	19	3	2	18	16	18	1	4

为了直观表现，我们将表 5 的数据绘制成条形图：

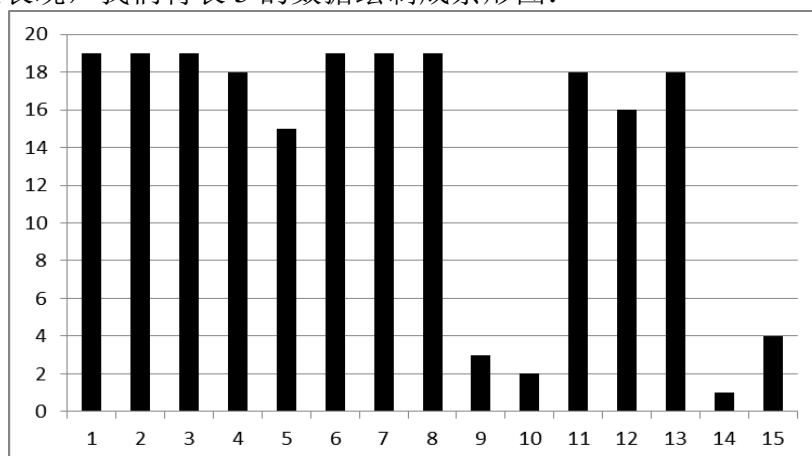


图 2 15 行包含的碎纸图个数

由表 5、图 2 知道：包含 19 个碎图片行集合的个数有 6 个，编号分别为：1、2、3、6、7、8，这 6 个可以唯一对应原文件的 6 个图像行；行集合的包含的图片个数大于 10 且小于 19 的行的个数有 5，分别的标号为：4、5、11、12、13。该行集合中包含于原文件中图片行。而行集合中元素个数小于 10 的行的个数有 4 个，行集合编号分别为：9、10、14、15。

由于附件 3 中的图片是 11×19 个，所以需要将 4 个行集合中的图片分配到 5 个行集合中去。

(2) 此外还得到了每行所包含的图片编号，考虑到数据过多，本文仅列举其中三行，见表 6。

表 6 特征匹配模型求解结果示例

	行集合 1			行集合 2			行集合 3		
碎	0	7	32	1	18	23	2	11	22
图	45	53	56	26	30	41	28	49	54
片	68	70	93	50	62	76	57	65	91
的	126	137	138	86	87	100	95	118	129
编	153	158	166	120	142	147	141	143	178
号	174	175	196	168	179	191	186	188	190
	208			195			192		

(3) 人工干预

由上文且结合附件 3、4 是 11×19 规模的文件，在特征匹配模型求解完成后加入第一次人工干预，即将表 5 中 4 个行集合中的图片元素通过与对应的 5 个行集合的图片进行人工配对，来人为生成与原文件相同的 11 个行集合。具体的人工配对方法如下：

Step1.按行集合编号的顺序选择 9 集合中的一个图片，通过文义匹配和边缘断线匹配的方法，将其与集合 4、5、11、12、13 中各抽取出一个图片共 5 张图片，进行依次比较，如果与某行集合中的那个图片匹配成功，就将这个图片存入该行集合中；

Step2: 循环进行 Step1 步骤，直至将集合 9、10、14、15 中共计 10 个图片分别归于各自的行集合中。

在最不利的情况下，需要匹配 $5 \times 5 + 5 + 4 + 3 + 2 = 39$ 次，而在最有利的情况下只需配对9次。可以见得人工干预的复杂度不高，从而充分发挥了人工智能的准确度高的优点，而且很好的规避了人效率不高的缺点。

通过人工干预，将未分配的图片归入对应的行中之后，利用左右边界匹配模型对附件三中每一行图片进行横向排序，使行内图片排列固定下来。附件3中的各类图片的排列顺序如表7：

表7 附件3 各类图片的排列顺序

类1	168	100	076	062	142	030	041	023	147	191	050	179	120	086	195	026	001	087	018
类2	125	013	182	109	197	016	184	110	187	066	106	150	021	173	157	181	204	139	145
类3	061	019	078	067	069	099	162	096	131	079	063	116	163	072	006	177	020	052	036
类4	094	034	084	183	090	047	121	042	124	144	077	112	149	097	136	164	127	058	043
类5	007	208	138	158	126	068	175	045	174	000	137	053	056	093	153	070	166	032	196
类6	038	148	046	161	024	035	081	189	122	103	130	193	088	167	025	008	009	105	074
类7	014	128	003	159	082	199	135	012	073	160	203	169	134	039	031	051	107	115	176
类8	029	064	111	201	005	092	180	048	037	075	055	044	206	010	104	098	172	171	059
类9	089	146	102	154	114	040	151	207	155	140	185	108	117	004	101	113	194	119	123
类10	049	054	065	143	186	002	057	192	178	118	190	095	011	022	129	028	091	188	141
类11	071	156	083	132	200	017	080	033	202	198	015	133	170	205	85	152	165	027	060

再根据上、下匹配模型，得到行与行之间的邻接关系为：8→5；1→6；6→11；7→4；10→3；2→8；3→1；11→7；4→2；5→9。进而便可得到行的排列顺序为：10→3→1→6→11→7→4→2→8→5→9。

同理，对于附件四采用相同的办法得到行内图片的排列如表8所示：

表8 附件4 各行内图片的排列顺序

类1	020	041	108	116	136	073	036	207	135	015	076	043	199	045	173	079	161	179	143
类2	132	181	095	069	167	163	166	188	111	144	206	003	130	034	013	110	025	027	178
类3	086	051	107	029	040	158	186	098	024	117	150	005	059	058	092	030	037	046	127
类4	208	021	007	049	061	119	033	142	168	062	169	054	192	133	118	189	162	197	112
类5	171	042	066	205	010	157	074	145	083	134	055	018	056	035	016	009	183	152	044
类6	159	139	001	129	063	138	153	053	038	123	120	175	085	050	160	187	097	203	031
类7	191	075	011	154	190	184	002	104	180	064	106	004	149	032	204	065	039	067	147
类8	070	084	060	014	068	174	137	195	008	047	172	156	096	023	099	122	090	185	109
类9	081	077	128	200	131	052	125	140	193	087	089	048	072	012	177	124	000	102	115
类10	019	194	093	141	088	121	126	105	155	114	176	182	151	022	057	202	071	165	082
类11	201	148	170	196	198	094	113	164	078	103	091	080	101	026	100	006	017	028	146

根据上、下匹配模型，得到行与行之间的邻接关系为：5→9；7→11；10→6；8→2；11→3；6→1；4→8；1→4；2→5；3→10；由此得到行与行之间的排列顺序：7→11→3→10→6→1→4→8→2→5→9。

3.复原结果

将已经得到的行内图片排列顺序以及行与行之间的排列顺序相结合，便可以得到各个碎图片在原文件中的位置，从而将原文件复原。结合用 word 拼图，使复原的结果分别以表格和图像的形式表现。其中附件三的复原表格见表9，复原图见附录2.7，附件四的复原表见表10，复原图见附录2.8。

表 9 附件 3（汉字）复原的碎片序号

049	054	065	143	186	002	057	192	178	118	190	095	011	022	129	028	091	188	141
061	019	078	067	069	099	162	096	131	079	063	116	163	072	006	177	020	052	036
168	100	076	062	142	030	041	023	147	191	050	179	120	086	195	026	001	087	018
038	148	046	161	024	035	081	189	122	103	130	193	088	167	025	008	009	105	074
071	156	083	132	200	017	080	033	202	198	015	133	170	205	085	152	165	027	060
014	128	003	159	082	199	135	012	073	160	203	169	134	039	031	051	107	115	176
094	034	084	183	090	047	121	042	124	144	077	112	149	097	136	164	127	058	043
125	013	182	109	197	016	184	110	187	066	106	150	021	173	157	181	204	139	145
029	064	111	201	005	092	180	048	037	075	055	044	206	010	104	098	172	171	059
007	208	138	158	126	068	175	045	174	000	137	053	056	093	153	070	166	032	196
089	146	102	154	114	040	151	207	155	140	185	108	117	004	101	113	194	119	123

表 10 附件 4（英文）复原的碎片序号

191	075	011	154	190	184	002	104	180	064	106	004	149	032	204	065	039	067	147
201	148	170	196	198	094	113	164	078	103	091	080	101	026	100	006	017	028	146
086	051	107	029	040	158	186	098	024	117	150	005	059	058	092	030	037	046	127
019	194	093	141	088	121	126	105	155	114	176	182	151	022	057	202	071	165	082
159	139	001	129	063	138	153	053	038	123	120	175	085	050	160	187	097	203	031
020	041	108	116	136	073	036	207	135	015	076	043	199	045	173	079	161	179	143
208	021	007	049	061	119	033	142	168	062	169	054	192	133	118	189	162	197	112
070	084	060	014	068	174	137	195	008	047	172	156	096	023	099	122	090	185	109
132	181	095	069	167	163	166	188	111	144	206	003	130	034	013	110	025	27	178
171	042	066	205	010	157	074	145	083	134	055	018	056	035	016	009	183	152	044
081	077	128	200	131	052	125	140	193	087	089	048	072	012	177	124	000	102	115

5.2.4 结果分析

同理，视人工干预后的最终结果为正确答案，检验未加入人工干预计算机排序结果。因为本问中人工干预只选择图片的首序列，对图片的排列顺序没有任何影响，所以附件 3（中文）的排序正确率为 90.4%，附件 4（英文）的排序正确率为 65.1%。

由此可得，三模型两筛选的方法能很好的解决中文单页打印横纵切纸片的拼接复原问题，而对于英文单页打印横纵切纸片的拼接复原效果不理想。

5.3 问题三的模型建立与求解

问题三要求在双面打印横纵切割的情况下，对碎纸片进行拼接复原。由于问题三相较于问题二，仅加入了双面打印一个新的条件，故可知问题三的基本求解思路与问题二一致。

5.3.1 模型的准备

1. 图像的数据处理

利用 matlab 读取每张图片的灰度信息，再将灰度信息转换为 0-1 矩阵。

2. 构建正、反面特征矩阵

利用问题二中英文灰度条的构建方法，先得到图片 k 的 a 面特征灰度条，再扫描特征灰度条，得到 a 面的特征列向量：

$$D_a^{(k)} = (d_{1,a}^{(k)}, d_{2,a}^{(k)}, \dots, d_{m,a}^{(k)})^T$$

$$\text{其中 } d_{i,a}^{(k)} = \begin{cases} 0, & \text{第 } k \text{ 张图片中第 } i \text{ 行元素之和} < M \\ 1, & \text{第 } k \text{ 张图片中第 } i \text{ 行元素之和} \geq M \end{cases}$$

同理，得到 b 面的特征列向量：

$$D_b^{(k)} = (d_{1,b}^{(k)}, d_{2,b}^{(k)}, \dots, d_{m,b}^{(k)})^T$$

$$\text{其中 } d_{i,b}^{(k)} = \begin{cases} 0, & \text{第 } k \text{ 张图片中第 } i \text{ 行元素之和} < M \\ 1, & \text{第 } k \text{ 张图片中第 } i \text{ 行元素之和} \geq M \end{cases}$$

5.3.3 建立双面横纵切纸片匹配模型

(1) 建立两次特征匹配模型

参考问题二的求解思路，需要进行两次特征匹配：

第一次特征匹配：

将碎片 k 的 a 面与碎片 s ($s=0,1\cdots 208$ 且 $s \neq k$) 的 a 面进行特征比较，即求碎片 k 的 a 面特征列向量 $D_a^{(k)}$ 与碎片 s 的 a 面特征列向量 $D_a^{(s)}$ 的对应元素之差，再对差的绝对值求和，得到特征值 $R_{k,s}^{a,a}$ ：

$$R_{k,s}^{a,a} = \sum_{i=1}^m |d_{i,a}^{(k)} - d_{i,a}^{(s)}| \quad (5.17)$$

再将碎片 k 的 a 面与碎片 s ($s=0,1\cdots 208$ 且 $s \neq k$) 的 b 面进行特征比较，即求碎片 k 的 a 面特征列向量 $D_b^{(k)}$ 与碎片 s 的 b 面特征列向量 $D_b^{(s)}$ 的对应元素之差，再对差的绝对值求和，得到特征值 $R_{k,s}^{a,b}$ ：

$$R_{k,s}^{a,b} = \sum_{i=1}^m |d_{i,a}^{(k)} - d_{i,b}^{(s)}| \quad (5.18)$$

$$R_k^1 = \min\{R_{k,s}^{a,a}, R_{k,s}^{a,b}\} \quad (5.19)$$

取一个合适小的置信区间 $[c,d]$ ，若 $R_k^1 \in [c,d]$ ，则进行第二次特征匹配：

情况一 ($R_k^1 = R_{k,s}^{a,a}$)：将碎片 k 的 b 面与碎片 s ($s=0,1\cdots 208$ 且 $s \neq k$) 的 b 面进行特征比较，即求碎片 k 的 b 面特征列向量 $D_b^{(k)}$ 与碎片 s 的 b 面特征列向量 $D_b^{(s)}$ 的对应元素之差，再对差的绝对值求和，得到特征值 $R_{k,s}^{b,b}$ ：

$$R_{k,s}^{b,b} = \sum_{i=1}^m |d_{i,b}^{(k)} - d_{i,b}^{(s)}| \quad (5.20)$$

若取一个合适小的置信区间 $[e,f]$ ，若 $R_{k,s}^{b,b} \in [e,f]$ ，则认为碎片 k 与碎片 s 的匹配方式为 k 的 a 面与 s 的 a 面处于一面的同一行， k 的 b 面与 s 的 b 面处于另一面的同一行。

情况二 ($R_k^1 = R_{k,s}^{a,b}$)：将碎片 k 的 b 面与碎片 s ($s=0,1\cdots 208$ 且 $s \neq k$) 的 a 面进行特征比较，即求碎片 k 的 b 面特征列向量 $D_b^{(k)}$ 与碎片 s 的 a 面特征列向量 $D_b^{(s)}$ 的对应元素之差，再对差的绝对值求和，得到特征值 $R_{k,s}^{b,a}$ ：

$$R_{k,s}^{b,a} = \sum_{i=1}^m |d_{i,b}^{(k)} - d_{i,a}^{(s)}| \quad (5.21)$$

若取一个合适小的置信区间 $[e,f]$ ，若 $R_{k,s}^{b,a} \in [e,f]$ ，则认为碎片 k 与碎片 s 的匹配方式为 k 的 b 面与 s 的 a 面处于一面的同一行， k 的 a 面与 s 的 b 面处于另一面的同一行。

(2)建立左右边界匹配模型

本问中此模型的构建方式同于问题二的思路。

(3)建立上下边界匹配模型

本问中上下边界匹配模型的建立与问题二的构建过程类似。

5.3.4 模型的求解

(1) 算法

Step1 读取图片数据，构建 0-1 矩阵；

Step2 任取碎片 i 依次与其他碎片 s 进行二次特征匹配，确定出 i 与 s 的特定面来自原文件的同一行；

Step3 重复 Step2，将附件中的图片聚类，将相同特征的图片放入同一行；

Step4 加入同问题二中的人工干预方式，将类的个数降维，并使得每类的图片个数相同。

Step5 取同一行中第 i 张碎片, 将其依次与第 1 张碎片, 第 2 张碎片, 第 $i-1$ 张碎片, ..., 第 $i+1$ 张碎片, ..., 第 n 张碎片先进行右边界匹配, 得到右边界匹配值, 再依次进行左边界匹配, 得到左边界匹配值。比较右边界匹配值与左边界匹配值的大小, 选择两者之间的最小值对应的匹配方式, 将两张图片按匹配方式结合, 视为一个整体。

Step6 重复进行 **Step5**, 直至确定出每行内部图片的排列顺序。

Step7 取第 i 行, 将其依次与第 1 行, 第 2 行, 第 $i-1$ 行, ..., 第 $i+1$ 行, ..., 第 n 行先进行上边界匹配, 得到上边界匹配值, 再依次进行下边界匹配, 得到下边界匹配值。比较上边界匹配值与下边界匹配值的大小, 选择两者之间的最小值对应的匹配方式, 将两张图片按匹配方式结合, 视为一体。

Step8 重复进行 **Step7**, 直至确定出每行的上下位置, 从而得到图片的原始序列。

(2) 算法的实现

根据上述模型的分类原则, 通过 **matlab** 编程(见附录), 将附件五中的 2×209 张图片进行了归类。程序计算出来的结果中总共含有 36 类, 由于数据太多, 这里只选取其中三类的元素展示在表 11 中:

表 11 特征匹配后的聚类结果

类 1					类 2					类 3				
0	7	30	45	69	1	2	37	65	70	6	24	26	57	91
84	85	86	105	121	88	107	115	139	151	96	99	100	103	106
135	141	148	176	185	162	166	170	180	191	109	112	113	134	196
204	80	126	187		203									

在这 36 类中, 每一类的元素个数参差不齐。这时需要加入人工干预, 先从元素较少的类中挑出元素与其他类的图片进行比较, 通过对比图中文字到上边界的距离、文字含义等特征来将这些图片归入其他元素较多的类别, 直至使正、反两面都有 11 类, 每类 19 个元素。

利用左右边界匹配模型, 对每一类中各个碎片进行横向排序, 得到各类中图片的排列顺序。再利用上下边界匹配模型, 求出类与类之间的排列顺序。结合碎片在行内的排列顺序以及类与类之间的排列顺序, 就可以得到每个碎片的正反面在原文件中的位置, 进而复原出原文件。

为了结果表述的方便, 而且由于正反面的地位相同, 也就是一张纸既可以说正面的后面是反面, 也可以说反面的后面是正面。设包含 000a 的那一页为原图片的正面。

本文结合 word 拼图方式, 将复原出的正反两面的信息分别以表格和图片的形式给出, 其中正面的碎片序列表格见表 12, 其图片见附录 3.3, 反面的碎片序列表格见表 13, 其图片见附录 3.4。

表 12 附件五正面的复原的碎片序号

136a	047b	020b	164a	081a	189a	029b	018a	108b	066b	110b	174a	183a	150b	155b	140b	125b	111a	078a
005b	152b	147b	060a	059b	014b	079b	144b	120a	022b	124a	192b	025a	044b	178b	076a	036b	010a	089b
143a	200a	086a	187a	131a	056a	138b	045b	137a	061a	094a	098b	121b	038b	030b	042a	084a	153b	186a
083b	039a	097b	175b	072a	093b	132a	087b	198a	181a	034b	156b	206a	173a	194a	169a	161b	011a	199a
090b	203a	162a	002b	139a	070a	041b	170a	151a	001a	166a	115a	065a	191b	037a	180b	149a	107b	088a
013b	024b	057b	142b	208b	064a	102a	017a	012b	028a	154a	197b	158b	058b	207b	116a	179a	184a	114b
035b	159b	073a	193a	163b	130b	021a	202b	053a	177a	016a	019a	092a	190a	050b	201b	031b	171a	146b
172b	122b	182a	040b	127b	188b	068a	008a	117a	167b	075a	063a	067b	046b	168b	157b	128b	195b	165a
105b	204a	141b	135a	027b	080a	000a	185b	176b	126a	074a	032b	069b	004b	077b	148a	085a	007a	003a
009a	145b	082a	205b	015a	101b	118a	129a	062b	052b	071a	033a	119b	160a	095b	051a	048b	133b	023a
054a	196a	112b	103b	055a	100a	106a	091b	049a	026a	113b	134b	104b	006b	123b	109b	096a	043b	099b

表 13 附件五反面的复原的碎片序号

078b	111b	125a	140a	155a	150a	183b	174b	110a	066a	108a	018b	029a	189b	081b	164b	020a	047a	136b
089a	010b	036a	076b	178a	044a	025b	192a	124b	022a	120b	144a	079a	014a	059a	060b	147a	152a	005a
186b	153a	084b	042b	030a	038a	121a	098a	094b	061b	137b	045a	138a	056b	131b	187b	086b	200b	143b
199b	011b	161a	169b	194b	173b	206b	156a	034a	181b	198b	087a	132b	093a	072b	175a	097a	039b	083a
088b	107a	149b	180a	037b	191a	065b	115b	166b	001b	151b	170b	041a	070b	139b	002a	162b	203b	090a
114a	184b	179b	116b	207a	058a	158a	197a	154b	028b	012a	017b	102b	064b	208a	142a	057a	024a	013a
146a	171b	031a	201a	050a	190b	092b	019b	016b	177b	053b	202a	021b	130a	163a	193b	073b	159a	035a
165b	195a	128a	157a	168a	046a	067a	063b	075b	167a	117b	008b	068b	188a	127a	040a	182b	122a	172a
003b	007b	085b	148b	077a	004a	069a	032a	074b	126b	176a	185a	000b	080b	027a	135b	141a	204b	105a
023b	133a	048a	051b	095a	160b	119a	033b	071b	052a	062a	129b	118b	101a	015b	205a	082b	145a	009b
099a	043a	096b	109a	123a	006a	104a	134a	113a	026b	049b	091a	106b	100b	055b	103a	112a	196b	054b

5.4.4 模型的分析

从模型的构建过程及算法的实现过程来看，问题三的双面横纵切纸片匹配模型实质就是对问题二的横纵切纸片匹配模型增加了一层特征匹配程度约束。从而可知使得问题三模型的精确度比问题二中模型的精确度高。

六、模型的评价与推广

6.1 模型的评价

本文建立的模型简单易懂，建立过程自然，流畅。并随着问题的深入，而不断加以改进，通过对结果进行分析，可知本文的模型精确度较高，可以合理的解决规则边缘碎纸片的拼接复原问题。

但也存在缺点：对于灰度匹配模型，是以两图像左右对应的边界矩阵的相应元素的差异值越小为原则，此模型计算简单，运算时间复杂度低。但是，带来了两种可能对最终造成不利的影响。第一种：存在两张不相邻的碎片，一张图片的右边界上没有任何文字，另一幅碎纸片左边界上也空白，没有任何文字。通过灰度匹配模型计算，两张图片的配对指标值 $P_{k,s}^*$ 特别小，于是会犯将这两张图片归为相邻图片的错误；第二种不利影响是基于附件图片的观测得到，直接忽略了如下情况：存在四张图片 A, B, C, D ， A, B 图片邻接， C, D 邻接且 A, B 与 C, D 它们的碎片边缘字迹断线是一模一样的，这样会出现 A 与 D 邻接的误判断。

总体来说，本文建立的模型层层深入，可以很好的解决这类精确度要求不高的题目。

6.2 模型的推广^[2]

本文解决的平行或垂直规则切割的碎纸片拼接复原问题。但现实生活中，在扫描文档碎片的时候，会不是水平而是倾斜的扫描、或者切割文档文件的时候，倾斜切割，导致得到的碎纸片是倾斜的碎纸片。为了解决这类更为贴近现实的问题，我们对模型做了推广。我们可先将碎纸片方向进行调整，再利用本文建立的模型完成碎纸片的重构。故我们设计算法如下：

Step1 找到平行于碎片中文字的直线的斜率：找到图片 $1-x$ 列，每一列最上面像素值为 0 的点，从 x 个点中选出最上面的点。同理得到 $(m-x)-m$ （ m 为碎片图片的宽度）列中处于最上面像素值为 0 的点。使用这两个点得到平行于碎片中文字方向的直线；

Step2 根据找到直线的斜率对碎片进行碎片角度的调整；

Step3 根据本文建立的模型拼接复原碎纸片；

参考文献

- [1] 作者不详, matlab 批量读入数据文件的方法,
http://hi.baidu.com/ben_wf/item/57196ac8671f7c1db77a2424, 访问时间 (2013.9.13)
- [2] 张翠, 基于点线的文档图片数字水印与碎片拼接,
<http://www.cnki.net/KCMS/detail/detail.aspx?QueryID=1&CurRec=1&recid=&filename=1011229650.nh&dbname=CMFDLAST2012&dbcode=CMFD&pr=&urlid=&yx=&uid=WEEvREcwSIJHSlidTTGJhYlJWelhJSFpId0xaa3FJQzBSb1VxYWtGa0dRMEJnL1pWcXBnY1pubVZqZ1REcEJQVQ==&v=MTY1MTJyNUViUElSOGVYMUx1eFITN0RoMVQzcVRyV00xRnJDVVJMbWZidWRyRnl2Z1c3L0xWRjl2SDdHNkY5Zko=>, 访问时间 (2013.9.13)。

附录

附录清单:

1.问题一

- 1.1 将灰度值矩阵转换成 0-1 矩阵的函数
- 1.2 提取左右边界处 0-1 值的程序
- 1.3 对每张图片进行灰度匹配并排序的程序
- 1.4 附件1的复原图片
- 1.5 附件2的复原图片

2.问题二

- 2.1 对附件三提取数据以及构造图片边界矩阵的程序
- 2.2 将附件三中各图片做行归类的程序
- 2.3 将附件三中人工干预后得到的横条做行排序的程序
- 2.4 将附件四提取数据以及构造图片边界矩阵的程序
- 2.5 将附件四中各图片做行归类的程序
- 2.6 将附件四中人工干预后得到的横条做行排序的程序
- 2.7 附件3的复原图片
- 2.8 附件4的复原图片

3.问题三

- 3.1 针对附件五提取数据的程序
- 3.2 对附件五中各图片做行归类的程序
- 3.3 附件5正面的复原图片
- 3.4 附件五反面的复原图片

1.问题一:

注: 首先应该对应附件中的图片复制到 MATLAB 的当前目录下。

1.1 将灰度值矩阵转换成 0-1 矩阵的函数

文件名: **jz01zh.m**

```
function B=jz01zh(A)
```

```
%将灰度值矩阵A转换为0-1矩阵B
```

```
[rows cols]=size(A);
```

```
for i=1:rows
```

```
    for j=1:cols
```

```
        if A(i, j)<255
```

```
            B(i, j)=0;
```

```
        end
```

```
        if A(i, j)==255
```

```
            B(i, j)=1;
```

```
        end
```

```
    end
```

```
end
```

1.2 提取左右边界处 0-1 值的程序

文件名: **boundary.m**

```
clear;clc
```

```
m=1;n=1;
```

```
for k=0:18
```

```
    if k>=10
```

```
        t=strcat('0',int2str(k),'.bmp');
```

```
        a1{m}=jz01zh((imread(t)));
```

```
        m=m+1;
```

```
    end
```

```
    if k<10
```

```
        t=strcat('00',int2str(k),'.bmp');
```

```
        a2{n}=jz01zh((imread(t)));
```

```
        n=n+1;
```

```
    end
```

```
end
```

```
a=[a2,a1];
```

```
for k=1:19
```

```
    temp=a{k};
```

```
    [row col]=size(temp);
```

```
    br=temp(:, col);
```

```
    bl=temp(:, 1);
```

```
    bound{k}=[bl, br];
```

```
end
```

1.3对每张图片进行灰度匹配并排序的程序

文件名: **sortpicture.m**

```
org=bound{1};ord=[1];
```

```
while length(ord)<19
```

```

pfh1=10000;pfh2=10000;
[raw,col]=size(org);
for i=2:19
    temp1=bound{i};
    p1=sum(abs(org(:,col)-temp1(:,1)))*%right
    if p1<pfh1
        pfh1=p1;
        cn1=i;
    end
end
for j=2:19
    temp2=bound{j};
    p2=sum(abs(org(:,1)-temp2(:,2)))*%left
    if p2<pfh2
        pfh2=p2;
        cn2=j;
    end
end
if p1<=p2
    org=[org, bound{cn1}];ord=[ord, cn1];
end
if p1>p2
    org=[bound{cn2}, org];ord=[cn2, ord];
end
end
ord-1

```

1.4 附录 1 对应的复原图片

城上层楼叠嶂。城下清淮古汴。举手揖吴云，人与暮天俱远。魂断。魂断。后夜松江月满。簌簌衣巾莎枣花。村里村北响缲车。牛衣古柳卖黄瓜。海棠珠缀一重重。清晓近帘栊。胭脂谁与匀淡，偏向脸边浓。小郑非常强记，二南依旧能诗。更有鲈鱼堪切脍，儿辈莫教知。自古相从休务日，何妨低唱微吟。天垂云重作春阴。坐中人半醉，帘外雪将深。双鬟绿坠。娇眼横波眉黛翠。妙舞蹁跹。掌上身轻意态妍。碧雾轻笼两凤，寒烟淡拂双鸦。为谁流涕不归家。错认门前过马。

我劝髯张归去好，从来自己忘情。尘心消尽道心平。江南与塞北，何处不堪行。闲离阻。谁念萦损襄王，何曾梦云雨。旧恨前欢，心事两无据。要知欲见无由，痴心犹自，倩人道、一声传语。风卷珠帘自上钩。萧萧乱叶报新秋。独携纤手上高楼。临水纵横回晚鞦。归来转觉情怀动。梅笛烟中闻几弄。秋阴重。西山雪淡云凝冻。凭高眺远，见长空万里，云无留迹。桂魄飞来光射处，冷浸一天秋碧。玉宇琼楼，乘鸾来去，人在清凉国。江山如画，望中烟树历历。省可清言挥玉尘，真须保器全真。风流何似道家纯。不应同蜀客，惟爱卓文君。自惜风流云雨散。关山有限情无限。待君重见寻芳伴。为说相思，目断西楼燕。莫恨黄花未吐。且教红粉相扶。酒阑不必看茱萸。俯仰人间今古。玉骨那愁瘴雾，冰姿自有仙风。海仙时遣探芳丛。倒挂绿毛么凤。

俎豆庚桑真过矣，凭君说与南荣。愿闻吴越报丰登。君王如有问，结袜赖王生。师唱谁家曲，宗风嗣阿谁。借君拍板与门槌。我也逢场作戏、莫相疑。晕腮嫌枕印。印枕嫌腮晕。闲照晚妆残。残妆晚照闲。可恨相逢能几日，不知重会是何年。茱萸仔细更重看。午夜风翻幔，三更月到床。簾纹如水玉肌凉。何物与依归去、有残妆。金炉犹暖麝煤残。惜香更把宝钗翻。重闻处，余熏在，这一番、气味胜从前。菊暗荷枯一夜霜。新苞绿叶照林光。竹篱茅舍出青黄。霜降水痕收。浅碧鳞鳞露远洲。酒力渐消风力软，飕飕。破帽多情却恋头。烛影摇风，一枕伤春绪。归不去。凤楼何处。芳草迷归路。汤发云腴酽白，盏浮花乳轻圆。人间谁敢更争妍。斗取红窗粉面。炙手无人傍屋头。萧萧晚雨脱梧楸。谁怜季子敝貂裘。

1.5 附录 2 对应的复原图片

fair of face.

The customer is always right. East, west, home's best. Life's not all beer and skittles. The devil looks after his own. Manners maketh man. Many a mickle makes a muckle. A man who is his own lawyer has a fool for his client.

You can't make a silk purse from a sow's ear. As thick as thieves. Clothes make the man. All that glisters is not gold. The pen is mightier than sword. Is fair and wise and good and gay. Make love not war. Devil take the hindmost. The female of the species is more deadly than the male. A place for everything and everything in its place. Hell hath no fury like a woman scorned. When in Rome, do as the Romans do. To err is human; to forgive divine. Enough is as good as a feast. People who live in glass houses shouldn't throw stones. Nature abhors a vacuum. Moderation in all things.

Everything comes to him who waits. Tomorrow is another day. Better to light a candle than to curse the darkness.

Two is company, but three's a crowd. It's the squeaky wheel that gets the grease. Please enjoy the pain which is unable to avoid. Don't teach your Grandma to suck eggs. He who lives by the sword shall die by the sword. Don't meet troubles half-way. Oil and water don't mix. All work and no play makes Jack a dull boy.

The best things in life are free. Finders keepers, losers weepers. There's no place like home. Speak softly and carry a big stick. Music has charms to soothe the savage breast. Ne'er cast a clout till May be out. There's no such thing as a free lunch. Nothing venture, nothing gain. He who can does, he who cannot, teaches. A stitch in time saves nine. The child is the father of the man. And a child that's born on the Sab-

2.问题

2.1对附件三提取数据以及构造图片边界矩阵的程序

文件名: **tiqushuju.m**

```
clear;clc
m=1;n=1;q=1;
for k=0:208
    if k<10
        t=strcat('00',int2str(k),'.bmp');
        a1{m}=jz01zh((imread(t)));
        m=m+1;
    end
    if k>=10 && k<=99
        t=strcat('0',int2str(k),'.bmp');
        a2{n}=jz01zh((imread(t)));
        n=n+1;
    end
    if k>=100
        t=strcat(int2str(k),'.bmp');
        a3{q}=jz01zh((imread(t)));
        q=q+1;
    end
end
a=[a1,a2,a3];
for k=1:209
    temp=a{k};
    [row col]=size(temp);
    br=temp(:,col);
    bl=temp(:,1);
    bu=temp(1,:);
    bd=temp(row,:);
    bound1{k}=[bl,br];
    bound2{k}=[bu',bd'];
end
```

2.2将附件三中各图片做行归类的程序

```

文件名: jl.m
clc
jd=[];
for k=1:209
    temp=a{k};
    for i=1:180
        if sum(temp(i,:))~=72
            jd1(i)=0;
        end
        if sum(temp(i,:))==72
            jd1(i)=1;
        end
    end
    jd=[jd jd1'];
end

tp=[];m=1;
for i=1:209
    temp=[];
    temp(i)=10000;
    for j=1:209
        if j~=i
            temp(j)=sum(abs(jd(1:100,i)-jd(1:100,j)));
        end
    end
    jm=min(temp);
    pipei1=find(temp>=jm-5);
    pipei2=find(temp<=jm+5);
    pipei3=intersect(pipei1,pipei2);
    tp=[tp,pipei3];
    if isempty(find(tp==i))
        pipei3=[i,pipei3];
        julei{m}=pipei3;
        m=m+1;
    end
end
end

```

2.3将附件三中人工干预后得到的横条做行排序的程序:

文件名: **hangpaixu.m**

```

AA=[168 100 76 62 142 30 41 23 147 191 50 179 120 86 195 26 1 87 18
125 13 182 109 197 16 184 110 187 66 106 150 21 173 157 181 204 139 145
61 19 78 67 69 99 162 96 131 79 63 116 163 72 6 177 20 52 36
94 34 84 183 90 47 121 42 124 144 77 112 149 97 136 164 127 58 43
7 208 138 158 126 68 175 45 174 0 137 53 56 93 153 70 166 32 196
38 148 46 161 24 35 81 189 122 103 130 193 88 167 25 8 9 105 74
14 128 3 159 82 199 135 12 73 160 203 169 134 39 31 51 107 115 176
29 64 111 201 5 92 180 48 37 75 55 44 206 10 104 98 172 171 59
89 146 102 154 114 40 151 207 155 140 185 108 117 4 101 113 194 119 123
49 54 65 143 186 2 57 192 178 118 190 95 11 22 129 28 91 188 141
71 156 83 132 200 17 80 33 202 198 15 133 170 205 85 152 165 27 60
];

```

```

for i=1:11
    tp1=[];
    for j=1:19
        tt=AA(i,j);
        tp1=[tp1 a{tt}];
    end
    hang{i}=tp1;
end
%for i=1:11
%    tp1=[];
%    for j=1:19
%        tp1=[tp1 a{AA(i,j)}];
%    end
%    hang{i}=tp1;
%end
%tp1=hang{1};
%for i=1:180
%    if sum(tp1(i,:))<=56

```



```

%      AB(i)=0;
%      else AB(i)=1;
%      end
swxw=[];syxy=[];swxy=[];syxw=[];
for i=1:11
    temp=hang{i};
    sh=sum(temp(1,:));xh=sum(temp(180,:));
    if sh==1368
        if xh==1368
            swxw=[swxw i];
        end
    end
    if sh==1368
        if xh<1368
            swxy=[swxy i];
        end
    end
    if sh<1368
        if xh==1368
            syxw=[syxw i];
        end
    end
    if sh<1368
        if xh<1368
            syxy=[syxy i];
        end
    end
end
end

for i=1:3
    org=hang{swxy(i)};pfh=10000000;
    for j=1:11
        temp=hang{j};
        p=sum(abs(org(180,:)-temp(1,:)));
        if p<pfh
            pfh=p;
            cn=j;
        end
    end
    [swxy(i),cn,p]
end

for i=1:3
    org=hang{syxw(i)};pfh=10000000;
    for j=1:11
        temp=hang{j};
        p=sum(abs(org(1,:)-temp(180,:)));
        if p<pfh
            pfh=p;
            cn=j;
        end
    end
    [cn,syxw(i),p]
end

for i=1:5
    org=hang{syxy(i)};pfh=10000000;
    for j=1:11
        temp=hang{j};
        p=sum(abs(org(180,:)-temp(1,:)));
        if p<pfh
            pfh=p;
            cn=j;
        end
    end
    [syxy(i),cn,p]
end
end
2.4将附件四提取数据以及构造图片边界矩阵的程序
文件名: tiqushuju_4.m
clear;clc
m=1;n=1;q=1;
for k=0:208

```

```

if k<10
    t=strcat('00',int2str(k),'.bmp');
    a1{m}=jz01zh((imread(t)));
    m=m+1;
end
if k>=10 && k<=99
    t=strcat('0',int2str(k),'.bmp');
    a2{n}=jz01zh((imread(t)));
    n=n+1;
end
if k>=100
    t=strcat(int2str(k),'.bmp');
    a3{q}=jz01zh((imread(t)));
    q=q+1;
end
end
a=[a1,a2,a3];
for k=1:209
    temp=a{k};
    [row col]=size(temp);
    br=temp(:,col);
    bl=temp(:,1);
    bu=temp(1,:);
    bd=temp(row,:);
    bound1{k}=[bl,br];
    bound2{k}=[bu,bd];
end

```

2.5 将附件四中各图片做行归类的程序

文件名: **jl_4.m**

```

clc
jd=[];
for k=1:209
    temp=a{k};
    for i=1:180
        if sum(temp(i,:))<=56
            jd1(i)=0;
        end
        if sum(temp(i,:))>56
            jd1(i)=1;
        end
    end
    jd=[jd jd1'];
end
clear julei
tp=[];m=1;
for i=1:209
    temp=[];
    temp(i)=10000;
    for j=1:209
        if j~=i
            temp(j)=sum(abs(jd(1:100,i)-jd(1:100,j)));
        end
    end
    jm=min(temp);
    pipei1=find(temp>=jm-8);
    pipei2=find(temp<=jm+8);
    pipei3=intersect(pipei1,pipei2);

    for q=1:length(tp)
        AAA=find(pipei3==tp(q));
        pipei3(AAA)=[];
    end
    tp=[tp,pipei3];
    if isempty(find(tp==i))
        pipei3=[i,pipei3];
        julei{m}=pipei3;
        tp=[tp,julei{m}];
        m=m+1;
    end
end
end

```

2.6 将附件四中人工干预后得到的横条做行排序的程序

文件名: **hangpaixu_4.m**

```

AA=[21 42 109 117 137 74 37 208 136 16 77 44 200 46 174 80 162 180 144
133 182 96 70 168 164 167 189 112 145 207 4 131 35 14 111 26 28 179
87 52 108 30 41 159 187 99 25 118 151 6 60 59 93 31 38 47 128
209 22 8 50 62 120 34 143 169 63 170 55 193 134 119 190 163 198 113
172 43 67 206 11 158 75 146 84 135 56 19 57 36 17 10 184 153 45
160 140 2 130 64 139 154 54 39 124 121 176 86 51 161 188 98 204 32
192 76 12 155 191 185 3 105 181 65 107 5 150 33 205 66 40 68 148
71 85 61 15 69 175 138 196 9 48 173 157 97 24 100 123 91 186 110
82 78 129 201 132 53 126 141 194 88 90 49 73 13 178 125 1 103 116

```

```

20 195 94 142 89 122 127 106 156 115 177 183 152 23 58 203 72 166 83
202 149 171 197 199 95 114 165 79 104 92 81 102 27 101 7 18 29 147];

```

```

for i=1:11
    tp1=[];
    for j=1:19
        tp1=[tp1 a(AA(i,j))];
    end
    hang{i}=tp1;
end
%tp1=hang{1};
%for i=1:180
%    if sum(tp1(i,:))<=56
%        AB(i)=0;
%    else AB(i)=1;
%    end
swxw=[];syxy=[];swxy=[];syxw=[];
for i=1:11
    temp=hang{i};
    sh=sum(temp(1,:));xh=sum(temp(180,:));
    if sh==1368
        if xh==1368
            swxw=[swxw i];
        end
    end
    if sh==1368
        if xh<1368
            swxy=[swxy i];
        end
    end
    if sh<1368
        if xh==1368
            syxw=[syxw i];
        end
    end
    if sh<1368
        if xh<1368
            syxy=[syxy i];
        end
    end
end
end

for i=1:3
    org=hang{swxy(i)};pfh=10000000;
    for j=1:11
        temp=hang{j};
        p=sum(abs(org(180,:)-temp(1,:)));
        if p<pfh
            pfh=p;
            cn=j;
        end
    end
    [swxy(i),cn,p]
end

for i=1:3
    org=hang{syxw(i)};pfh=10000000;
    for j=1:11
        temp=hang{j};
        p=sum(abs(org(1,:)-temp(180,:)));
        if p<pfh
            pfh=p;
            cn=j;
        end
    end
    [cn,syxw(i),p]
end

for i=1:5
    org=hang{syxy(i)};pfh=10000000;
    for j=1:11
        temp=hang{j};
        p=sum(abs(org(180,:)-temp(1,:)));
        if p<pfh
            pfh=p;
            cn=j;
        end
    end
    [syxy(i),cn,p]
end

```

2.7 附录 3 对应的复原图片

便邮。温香熟美。醉慢云鬟垂两耳。多谢春工。不是花红是玉红。一颗樱桃樊素口。不爱黄金，只爱人长久。学画鸦儿犹未就。眉尖已作伤春皱。清泪斑斑，挥断柔肠寸。嗔人问。背灯偷搵拭尽残妆粉。春事阑珊芳草歇。客里风光，又过清明节。小院黄昏人忆别。落红处处闻啼鴂。岁云暮，须早计，要褐裘。故乡归去千里，佳处辄迟留。我醉歌时君和，醉倒须君扶我，惟酒可忘忧。一任刘玄德，相对卧高楼。记取西湖西畔，正暮山好处，空翠烟霏。算诗人相得，如我与君稀。约他年、东还海道，愿谢公、雅志莫相违。西州路，不应回首，为我沾衣。料峭春风吹酒醒。微冷。山头斜照却相迎。回首向来潇洒处。归去。也无风雨也无晴。紫陌寻春去，红尘拂面来。无人不道看花回。惟见石榴新蕊、一枝开。

九十日春都过了，贪忙何处追游。三分春色一分愁。雨翻榆荚阵，风转柳花球。白雪清词出坐间。爱君才器两俱全。异乡风景却依然。团扇只堪题往事，新丝那解系行人。酒阑滋味似残春。

缺月向人舒窈窕，三星当户照绸缪。香生雾縠见纤柔。搔首赋归欤。自觉功名懒更疏。若问使君才与术，何如。占得人间一味愚。海东头，山尽处。自古空槎来去。槎有信，赴秋期。使君行不归。别酒劝君君一醉。清润潘郎，又是何郎婿。记取钗头新利市。莫将分付东邻子。西塞山边白鹭飞。散花洲外片帆微。桃花流水鳜鱼肥。主人嗔小。欲向东风先醉倒。已属君家。且更从容等待他。愿我已无当世望，似君须向古人求。岁寒松柏肯惊秋。

水涵空，山照市。西汉二疏乡里。新白发，旧黄金。故人恩义深。谁道东阳都瘦损，凝然点漆精神。瑶林终自隔风尘。试看披鹤氅，仍是谪仙人。三过平山堂下，半生弹指声中。十年不见老仙翁。壁上龙蛇飞动。暖风不解留花住。片片著人无数。楼上望春归去。芳草迷归路。犀钱玉果。利市平分沾四坐。多谢无功。此事如何到得依。元宵似是欢游好。何况公庭民讼少。万家游赏上春台，十里神仙迷海岛。

虽抱文章，开口谁亲。且陶陶、乐尽天真。几时归去，作个闲人。对一张琴，一壶酒，一溪云。相如未老。梁苑犹能陪俊少。莫惹闲愁。且折

2.8 附录 4 对应的复原图片

bath day. No news is good news.

Procrastination is the thief of time. Genius is an infinite capacity for taking pains. Nothing succeeds like success. If you can't beat em, join em. After a storm comes a calm. A good beginning makes a good ending.

One hand washes the other. Talk of the Devil, and he is bound to appear. Tuesday's child is full of grace. You can't judge a book by its cover. Now drips the saliva, will become tomorrow the tear. All that glitters is not gold. Discretion is the better part of valour. Little things please little minds. Time flies. Practice what you preach. Cheats never prosper.

The early bird catches the worm. It's the early bird that catches the worm. Don't count your chickens before they are hatched. One swallow does not make a summer. Every picture tells a story. Softly, softly, catchee monkey. Thought is already is late, exactly is the earliest time. Less is more.

A picture paints a thousand words. There's a time and a place for everything. History repeats itself. The more the merrier. Fair exchange is no robbery. A woman's work is never done. Time is money.

Nobody can casually succeed, it comes from the thorough self-control and the will. Not matter of the today will drag tomorrow. They that sow the wind, shall reap the whirlwind. Rob Peter to pay Paul. Every little helps. In for a penny, in for a pound. Never put off until tomorrow what you can do today. There's many a slip twixt cup and lip. The law is an ass. If you can't stand the heat get out of the kitchen. The boy is father to the man. A nod's as good as a wink to a blind horse. Practice makes perfect. Hard work never did anyone any harm. Only has compared to the others early, diligently

3.问题三

3.1针对附件五提取数据的程序

文件名: **tiqushuju_5.m**

```
clear;clc
m=1;n=1;q=1;
for k=0:208
    if k<10
        t=strcat('00',int2str(k),'a.bmp');
        a1{m}=jz01zh((imread(t)));
        m=m+1;
    end
    if k>=10 && k<=99
        t=strcat('0',int2str(k),'a.bmp');
        a2{n}=jz01zh((imread(t)));
        n=n+1;
    end
    if k>=100
        t=strcat(int2str(k),'a.bmp');
        a3{q}=jz01zh((imread(t)));
        q=q+1;
    end
end
a=[a1,a2,a3];
m=1;n=1;q=1;
for k=0:208
    if k<10
        t=strcat('00',int2str(k),'b.bmp');
        b1{m}=jz01zh((imread(t)));
        m=m+1;
    end
    if k>=10 && k<=99
        t=strcat('0',int2str(k),'b.bmp');
        b2{n}=jz01zh((imread(t)));
        n=n+1;
    end
    if k>=100
        t=strcat(int2str(k),'b.bmp');
        b3{q}=jz01zh((imread(t)));
        q=q+1;
    end
end
b=[b1,b2,b3];
for k=1:209
    temp=a{k};
    [row col]=size(temp);
```

```

        br=temp(:,col);
        bl=temp(:,1);
        bu=temp(1,:);
        bd=temp(row,:);
        bounda1{k}=[bl,br];
        bounda2{k}=[bu',bd'];
    end
    for k=1:209
        temp=b{k};
        [row col]=size(temp);
        br=temp(:,col);
        bl=temp(:,1);
        bu=temp(1,:);
        bd=temp(row,:);
        boundb1{k}=[bl,br];
        boundb2{k}=[bu',bd'];
    end
end

```

3.2对附件五中各图片做行归类的程序

文件名: **fenhang.m**

```

clc;clear glei
jda=[];jdb=[];
jd=[];
for k=1:209
    temp=a{k};
    for i=1:180
        if sum(temp(i,:))~=72
            jd(i)=0;
        end
        if sum(temp(i,:))==72
            jd(i)=1;
        end
    end
    jda=[jda jd'];
end
jd=[];
for k=1:209
    temp=b{k};
    for i=1:180
        if sum(temp(i,:))~=72
            jd(i)=0;
        end
        if sum(temp(i,:))==72
            jd(i)=1;
        end
    end
    jdb=[jdb jd'];
end
nm=1;tep=[];
for j=1:209
    tl=[];xu=j;
    for i=1:209
        if i~=xu && isempty(find(tep==i))
            ch1=sum(abs(jda(:,xu)-jda(:,i)));ch2=sum(abs(jda(:,xu)-jdb(:,i)));
            ch3=sum(abs(jdb(:,xu)-jda(:,i)));ch4=sum(abs(jdb(:,xu)-jdb(:,i)));
            [chm,ind]=min([ch1,ch2,ch3,ch4]);
            if chm<=18
                if ind(1)==1
                    chch=abs(jdb(:,1)-jdb(:,i));
                end
                if ind(1)==2
                    chch=abs(jdb(:,1)-jda(:,i));
                end
                if ind(1)==3
                    chch=abs(jda(:,1)-jdb(:,i));
                end
                if ind(1)==4
                    chch=abs(jda(:,1)-jda(:,i));
                end
                if chch<=1
                    tl=[tl i];
                end
            end
        end
    end
    tep=[tep,glei{tt}];
end
end
if isempty(find(tep==xu))
    glei{nm}=[xu tl];
end

```

```
nm=nm+1;  
end  
end
```

3.3 附录 5 一面的复原图片

He who laughs last laughs longest. Red sky at night shepherd's delight; red sky in the morning, shepherd's warning. Don't burn your bridges behind you. Don't cross the bridge till you come to it. Hindsight is always twenty-twenty.

Never go to bed on an argument. The course of true love never did run smooth. When the oak is before the ash, then you will only get a splash; when the ash is before the oak, then you may expect a soak. What you lose on the swings you gain on the roundabouts.

Love thy neighbour as thyself. Worrying never did anyone any good. There's nowt so queer as folk. Don't try to walk before you can crawl. Tell the truth and shame the Devil. From the sublime to the ridiculous is only one step. Don't wash your dirty linen in public. Beware of Greeks bearing gifts. Horses for courses. Saturday's child works hard for its living.

Life begins at forty. An apple a day keeps the doctor away. Thursday's child has far to go. Take care of the pence and the pounds will take care of themselves. The husband is always the last to know. It's all grist to the mill. Let the dead bury the dead. Count your blessings. Revenge is a dish best served cold. All's for the best in the best of all possible worlds. It's the empty can that makes the most noise. Never tell tales out of school. Little pitchers have big ears. Love is blind. The price of liberty is eternal vigilance. Let the punishment fit the crime.

The more things change, the more they stay the same. The bread always falls buttered side down. Blood is thicker than water. He who fights and runs away, may live to fight another day. Eat, drink and be merry, for tomorrow we die.

3.4 附录五反面的复原图片

What can't be cured must be endured. Bad money drives out good. Hard cases make bad law. Talk is cheap. See a pin and pick it up, all the day you'll have good luck; see a pin and let it lie, bad luck you'll have all day. If you pay peanuts, you get monkeys. If you can't be good, be careful. Share and share alike. All's well that ends well. Better late than never. Fish always stink from the head down. A new broom sweeps clean. April showers bring forth May flowers. It never rains but it pours. Never let the sun go down on your anger.

Pearls of wisdom. The proof of the pudding is in the eating. Parsley seed goes nine times to the Devil. Judge not,

that ye be not judged. The longest journey starts with a single step. Big fish eat little fish. Great minds think alike. The end justifies the means. Cowards may die many times before their death. You can't win them all. Do as I say, not as I do. Don't upset the apple-cart. Behind every great man there's a great woman. Pride goes before a fall.

You can lead a horse to water, but you can't make it drink. Two heads are better than one. March winds and April showers bring forth May flowers. A swarm in May is worth a load of hay; a swarm in June is worth a silver spoon; but a swarm in July is not worth a fly. Might is right. Let bygones be bygones. It takes all sorts to make a world. A change is as good as a rest. Into every life a little rain must fall. A chain is only as strong as its weakest link.

Don't look a gift horse in the mouth. Old soldiers never die, they just fade away. Seeing is believing. The opera ain't over till the fat lady sings. Silence is golden. Variety is the spice of life. Tomorrow never comes. If it ain't broke, don't fix it. Look before you leap. The road to hell is paved with good