

2013 高教社杯全国大学生数学建模竞赛

承 诺 书

我们仔细阅读了中国大学生数学建模竞赛的竞赛规则。

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，严格遵守竞赛规则，以保证竞赛的公正、公平性。如有违反竞赛规则的行为，我们将受到严肃处理。

我们参赛选择的题号是（从A/B/C/D中选择一项填写）： B题

我们的参赛报名号为（如果赛区设置报名号的话）： B0129

所属学校（请填写完整的全名）： 厦门大学

参赛队员（打印并签名）： 1、 林质锐

2、 江艺烨

3、 陈孝蓉

指导教师或指导教师组负责人（打印并签名）： 谭忠

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

本文针对附件1至附件5中经过碎纸机破碎后的各类纸片，设计不同的模型和算法，复原碎片。主要利用碎片间差异度大的特征构造特征因子，来描述碎片的行列特征，用以比较、分类、匹配。

问题一，对仅纵切碎片提取左右边界差异。将碎片用矩阵表示，将边界列向量视为1980维空间中的点，在两点间定义绝对值距离用以描述碎片边缘的匹配程度。两点间的绝对值距离越大表示碎片匹配程度越低，两点间的绝对值距离越小表示碎片匹配程度越高。在此定义基础上建立最优化模型，寻找和待匹配碎片距离最小的碎片与之相邻。按照此法依次从左至右找到相邻碎片，最终复原碎片，并且不用人工干预。

问题二，经过横纵切后的碎片左右边界差异不如问题一明显，故构造新的特征因子记录碎片空白行的宽度和位置信息。先找出位于文章最左端的11个碎片，根据空白行的特征为余下碎片找到同行的18个碎片。从最左端的11个碎片开始利用图论中寻找权值最小哈密尔顿路径的相关理论以及最优化理论向右复原整行碎片，得到11条只有横切的碎片条。再根据上下端特征，利用与问题一相似方法，并配合少量的人工干预复原文。

问题三，先对英文碎片进行预处理，抹掉每个字母的“长比划”，得到空白行较为规整的碎片，方便提取空白行特征信息。再定义四个行特征因子：

θ_1 为从碎片顶端像素开始向下连续白像素的个数。

θ_2 为从碎片底端像素开始向上连续白像素的个数。

η_1 为从碎片顶端像素开始向下连续黑像素的个数。

η_2 为从碎片底端像素开始向上连续黑像素的个数。

利用聚类分析分类，每类中的绝大部分碎片同属一行，人工将错误碎片调整，得到各行碎片，建立优化模型得到复原图，复原图完整度为54.55%，经过23次的人工干预得到最终的完整复原图。

在文章最后还提出了基于中英文字符不同特点的复原优化模型。

【关键字】碎纸片的拼接、特征因子、灰度矩阵、二值化矩阵、最大类间方差法、最优化、哈密尔顿路径、聚类分析

I 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。我们需要讨论以下问题：

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，要求写出干预方式及干预的时间节点。

2. 对于碎纸机既纵切又横切的情形，要求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，要求写出干预方式及干预的时间节点。

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。要求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。

II 符号说明

II.1 符号说明

问题一

- A_j 第j个碎片对应的二值化矩阵
- x_1^j, x_2^j A_j 的首、尾列的向量
- E 用于存放依次匹配复原的图片的集合
- F 用于存放碎片的集合
- $dist(x_1^i, x_2^j)$ 两点 x_1^i, x_2^j 间的绝对值距离
- $x_1^i(m)$ 向量 x_1^i 的第m个元素
- $x_2^i(m)$ 向量 x_2^i 的第m个元素
- b 用于记录碎片相邻顺序的19列行向量
- $b(m)$ b的第m列元素
- k 计数变量

问题二

- c_j 用于记录 A_j 空白行位置的列向量，即空白行特征因子
- $c_j(m)$ c_j 第m个元素

- l_1^i 灰度矩阵 A_i 的向左最小边距
- l_2^i 灰度矩阵 A_i 的向右最小边距
- ε 边界误差
- α 像素值的比例因子
- β 像素值的偏离因子
- DH 阈值
- TH 原图片的文字间距

问题三

- ω 每个空白行出现“长比划”的个数
- SH 阈值
- A 碎片二值矩阵
- $A(k, j)$ 二值矩阵中第k行, j列元素
- θ_1 从碎片顶端像素开始向下连续白像素的个数
- θ_2 从碎片底端像素开始向上连续白像素的个数
- η_1 从碎片顶端像素开始向下连续黑像素的个数
- η_2 从碎片底端像素开始向上连续黑像素的个数

模型优化

- x_i 残缺字母与第i个字母的模块矩阵中的0元素(黑像素)匹配上的对数
- x_i 残缺字母与第p个字母配对的最大比率

III 模型的建立与求解

III.1 问题一

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表格形式表达。

III.1.1 问题一的分析

为了叙述以及编程的方便，在分析过程中给予附件图片新编号，附件的图片编号去除无效0位后统一加上1作为新编号，如bmp000的新编号为bmp1。

因为颜色在计算机中用数值表示，故将附件一、二中的每张bmp图片用一个矩阵来表示。又图片的颜色只有黑白灰三种，故建立的矩阵即为灰度矩阵。只要对每张图的灰度矩阵做处理比较即可。

问题一为一维复原问题，仅考虑碎片左右(x轴方向)端的特征即可。

首先找出文章的最左端，由于原文页边距一定大于相邻文字的左右间距，故最左端文字与图片左边缘的距离最大的那张碎片即为文章的最左端。

找出最左端的碎片，就能以此从左往右拼出全文。

不妨设碎片为1(左)、2(右)相邻。则1右边缘的所有残缺文字就可以与2左边缘的所有残缺文字组成一列完整的文字，若用灰度矩阵描述此现象，如下。

左	右
255	255
255	255
255	255
255	255
255	255
255	255
255	255
255	250
76	195
0	195
0	195
240	195
255	195
97	195
0	195
0	195
254	195
255	237
182	255
0	255
0	255
203	255
255	255
149	255
0	255
0	255
235	255
255	255
137	255
0	255
13	255
255	255
255	225
37	83
0	83
63	83
255	243
255	255
191	255
0	255
0	255
193	255

将汉字“掌”劈开，得到两张图分别存放“掌”的左右两半。上图左列向量为左半“掌”最后一列像素的灰度值；上图右列向量为右半“掌”最前一列像素的灰度值。可由上述图例发现两张图相邻边缘像素列的灰度值存在高一一匹配度，即绝大部分行的灰度值相等。

根据以上思想建立最优化模型，寻找与被匹配图存在最多一一对应的图片，即可判定为被匹配图的相邻图片。

考虑到由于每个字边缘像素的灰度多种，会使相邻图的一一匹配对数下降，故将灰度矩阵二值化为只有0(黑色像素)和255(白色像素)的矩阵。我们所说的图像的二值化，就是将图像上的像素点的灰度值设置为0(黑色像素)或255(白色像素)，也就是将本题中的灰色像素点转化为白色或者黑色像素点。

最常用的方法就是设定一个全局的阈值，用阈值将图像的数据分成两部分：大于阈值的像素群和小于阈值的像素群。将大于阈值的像素群设置为白色，小于阈值的像素群设置为黑色

根据阈值选取的不同，二值化的算法分为固定阈值和自适应阈值，比较常用的二值化方法有双峰法，P参数法，迭代法和最大类间方差法

最大类间方差法是一种自适应的阈值确定的方法，它是按图像的灰度特性，将图像分成背景和目标两个部分。背景和目标之间的类间方差越大，说明构成图像的两部分的差别越大，当部分目标错分为背景或部分背景错分为目标都会导致两部分差别变小。因此，使类间方差最大的分割意味着错分概率越小。

本题采用的阈值为0.5373的最大类间方差法

在此之后，我们引入绝对值距离来描述匹配程度，具体模型如下。

III.1.2 问题一模型建立

特征因子的构建

碎片bmp1,bmp2,bmp3,...,bmp19对应的二值矩阵分别记为 A_1, A_2, \dots, A_{19} 。特征因子 x_1^j, x_2^j ($j=1,2,\dots,19$)分别为 A_j 的左、右端列向量，可看做1980维空间内的点，则共有 19×2 个列向量。

绝对值距离的定义

在 A_i (左), A_j (右)($i,j=1,2,\dots,19$,且 $i \neq j$)的两点 x_1^i, x_2^j 间定义绝对值距离:

$$dist(x_1^i, x_2^j) = \sum_{m=1}^{1980} |x_2^j(m) - x_1^i(m)|$$

$x_1^i(m)$ 表示向量 x_1^i 的第m个元素， $x_2^j(m)$ 表示向量 x_2^j 的第m个元素。

利用两点间距离的大小来判断相 A_i (左), A_j (右)边缘的匹配程度，距离越大代表匹配程度越低，距离越小代表匹配程度越高。

问题一复原模型以及算法

step1: 算出19张碎片左端的留白距离，得到距离最大的图片即为原图最左端的碎片，编号为left，其二值矩阵为 A_{left} 。

step2: 建立两个用于存放各图片二值矩阵的集合E,F。 $E = A_{left}, F = \{A_1, A_2, \dots, A_{19}\} \setminus A_{left}$ ，建立19列的顺序行向量b，并将其初始化0向量，用于存放重新排列顺序（即复原）后的碎片对应的二值矩阵标号。b(1)=left. k为计数变量，表示按照左右相邻顺序已排好k个二值矩阵（碎纸机图），初始化k=1。

step3: 利用最优化的方法为第k张图向右匹配第k+1张图。

$$\begin{cases} min = dist(x_2^{b(k)}, x_1^j) \\ A_j \in F \end{cases}$$

得最优解 A_{ss} ,

$$b(k+1) = ss;$$

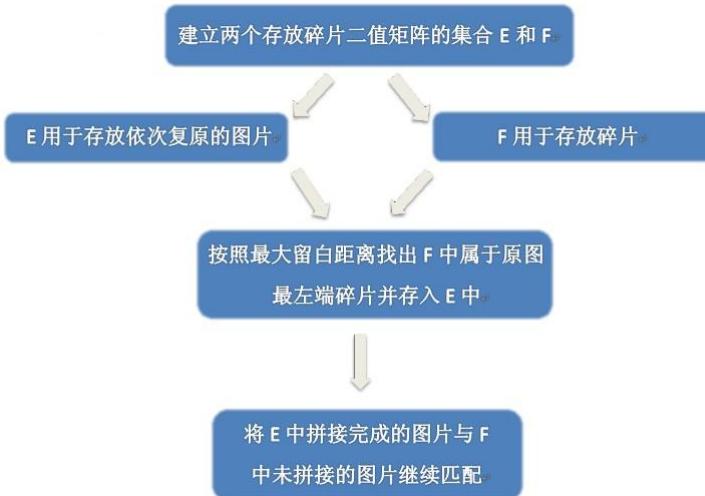
$$E = E \cup \{A_{ss}\};$$

$$F = F \setminus A_{ss};$$

$$k=k+1;$$

若 $k=19$, 终止; 否则重新开始step3.

下面用算法流程图直观地表达该算法步骤



III.1.3 问题一模型求解

利用Matlab编写程序求解得到:

问题1附件1图片拼接序号.

8	14	12	15	3	10	2	16	1	4	5	9	13	18	11	7	17	0	6
---	----	----	----	---	----	---	----	---	---	---	---	----	----	----	---	----	---	---

问题1附件2图片拼接序号.

3	6	2	7	15	18	11	0	5	1	9	13	10	8	12	14	17	16	4
---	---	---	---	----	----	----	---	---	---	---	----	----	---	----	----	----	----	---

III.2 问题二

对于碎纸机既纵切又横切的情形, 请设计碎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。

III.2.1 问题二的分析

问题二属于二维复原问题, 不仅考虑碎片左右(x轴方向)端特征, 还需考虑上下(y轴方向)端特征。使问题更加复杂化的主要原因有:

1. 碎片个数增多增加问题的需考虑因素。

2. 文章首行文字与页面顶端(低端)的距离不会明显大于行距, 故难以定位处于文章顶端和底端的碎片。

3.部分碎片中的文字与页面上下左右中的某些边没有相交，导致某些图片无法依靠问题一中仅由图片边界的灰度值得出匹配。

4.段落间距较大和段首有空字对全篇的排序造成较大影响。

考虑到问题一中的特征因子难以满足本问题的要求，故增设两个特征因子——空白行和向左（向右）最小边距（具体定义见模型建立）。

先找出复原图最左端的11个头碎片，根据空白行特征因子为头碎片找到同行的剩余碎片，复原出只有横切的11条碎片，最后利用上下端的特征将11条碎片依上至下按顺序复原出最终图。

III.2.2 问题二模型建立

本问用每张图的灰度矩阵处理以下问题。

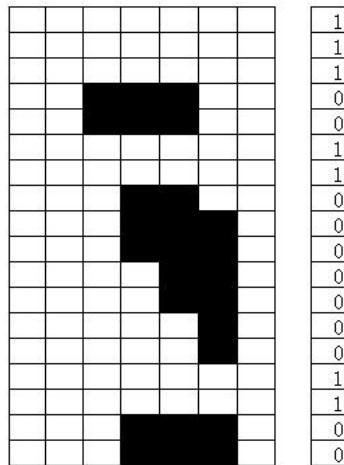
空白行特征因子定义

对于碎片的灰度矩阵 $A_j(j = 1, 2, \dots, 19)$,构造1980行的列向量 $c_j, c_j(m)$ 代表列向量第m行元素。

$$c_j(m) = \begin{cases} 1, & A_j \text{ 的第 } m \text{ 行所有像素值之和为 } 72 \\ 0, & \text{否则} \end{cases}$$

c_j 用以记录 A_j 的空白行位置。

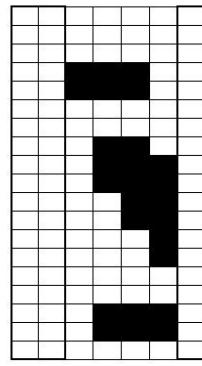
下图以18行的像素矩阵解释说明上述定义。



故最右边的列向量即为空白行特征因子 c_j 。

向左（向右）最小边距特征因子定义

对于灰度矩阵 A_i 的向左（向右）最小边距分别记为 l_1^i, l_2^i 。向左（向右）最小边距为碎片最左端向右（最右端向左）数的空白像素列个数,如下图。



如上图碎片最左端向右数的空白像素列个数2，碎片最右端向左的空白像素列个数1。故 $l_1^i = 2, l_2^i = 1$ 。

边界误差定义^[1]

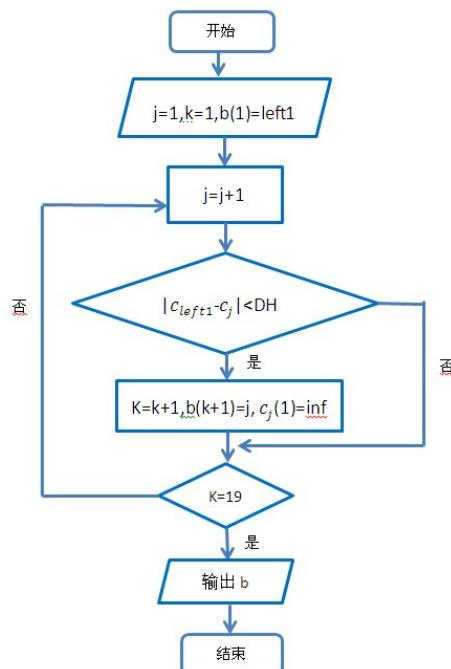
A_i (左), A_j (右)的边界误差用以描述两个灰度矩阵边缘的匹配程度，作用类似于问题一中的绝对值距离。

图片 A_i (左), A_j (右)间的边界误差定义 ε 为： $\varepsilon = \frac{1}{180} \sum_{m=1}^{180} (x_2^i(m) - \alpha x_1^j(m) - \beta)^2$
其中 α 为像素值的比例因子， β 为像素值的偏离因子。这里 $\alpha=1$, $\beta=0$ 。

头碎片寻找同行剩余碎片的模型和算法

为头碎片 A_{left1} 寻找同行剩余碎片。若碎片 A_j 与 A_{left1} 同行，在理论上有 $c_{left1} = c_j$ 。但由于本问题利用未二值化的灰度矩阵分析，故需要考虑字符边缘白噪声造成的误差。故给出一个阈值 DH ，若有 $|c_{left1} - c_j| < DH$ ，就认为 A_j 与 A_{left1} 同行。

程序框图如下：



复原同行碎片的最优化模型

现为灰度矩阵 A_s 寻找右相邻碎片。

目标为在可供匹配的所有灰度矩阵中找寻与灰度矩阵 A_s 边缘匹配程度最大者，即边缘误差最小者。

为了严格筛选，增加约束条件，以图例说明含义。

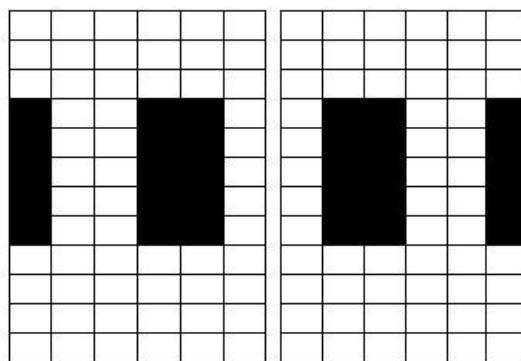
为便于说明假设碎片为 12×6 像素，且只有黑白两色。一个 5×2 的像素矩阵代表一个完整字符，设 Th 为原图片的文字间距(单位：像素).在以下说明过程中设 $Th=2$ 像素。

图1,图2为左右相邻的两张碎片。则图1，图2被碎纸机劈开时，劈开位置可分为两类分别对应情况一、情况二。

若为情况一，则图1的向右最小边距 $l_2^1 = 1$,图2的向左最小边距 $l_1^2 = 1$ 则有

$$l_2^1 + l_1^2 = Th$$

情况一：从两字符间留白部分劈开



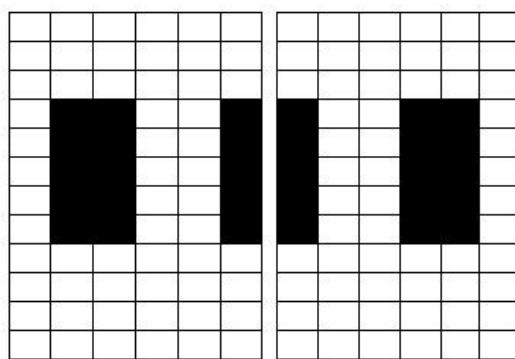
1

2

若为情况二，则图1的向右最小边距 $l_2^1 = 0$,图2的向左最小边距 $l_1^2 = 0$ 则有

$$l_2^1 + l_1^2 < Th$$

情况二：从某字符中间劈开



1

2

可知若 A_s (左)与 A_j (右)相邻，则定有

$$l_2^s + l_1^j \leq Th$$

观察附件三、四的碎片，发现部分碎片的文字不与碎片左端或右端相交，对于此类图片无法根据边缘匹配程度来选择相邻图片，而这些难以配对的图片常常干扰其他图片的匹配过程。经过试用比较，可利用上述不等式排除此类图片的干扰。

$$\begin{aligned} \min \varepsilon = & \frac{1}{180} \sum_{m=1}^{180} (x_2^s(m) - \alpha x_1^j(m) - \beta)^2 \\ & \left\{ \begin{array}{l} l_2^s + l_1^j \leq Th \\ A_j \in \bar{F} \end{array} \right. \end{aligned}$$

Th 为原图片的文字间距(单位：像素), \bar{F} 为可供匹配的碎片灰度矩阵全体构成的集合。

复原同行碎片的图论模型

现为灰度矩阵 A_1 寻找右相邻碎片。

将灰度矩阵 A_j 的左右列向量 x_1^j, x_2^j 都视为顶点。可按照如下图的建立规则得到非负有向图如下：

1. 若 $i=1$ 且 $j \neq 1$ ，先在 x_1^1, x_2^1 连边，其方向为 $x_1^1 \rightarrow x_2^1$ ，权为 0. 再对 x_2^s, x_1^j ($j=2, \dots, 19$) 连边，其方向为 $x_2^s \rightarrow x_1^j$ ，权为两顶点的边界误差。
 2. 若 $i, j \neq 1$ 且 $i \neq j$ ，在顶点 x_2^j, x_1^i ($i, j=2, \dots, 19$) 连边，其方向为 $x_2^j \rightarrow x_1^i$ ，权为两顶点的边界误差。
 3. 若 $i, j \neq 1$ 且 $i=j$ ，在顶点 x_1^i, x_2^i ($i=2, \dots, 19$) 连边，其方向为 $x_1^i \rightarrow x_2^i$ ，权为 0.
- 则求问题转化为在非负有向图中寻找一条权值最小的哈密尔顿路径的问题。

问题二复原步骤

step1: 计算每张碎片的向左最小距离，找出其中距离最大的11张碎片即为原图最左边的头碎片。其图片序号记为 $left1, left2, \dots, left11$ 。

step2: 根据头碎片寻找同行剩余碎片的模型和算法对11个头碎片分别找出各自的同行碎片。

step3: 对每行碎片利用复原同行碎片的最优化模型复原，其中令 s 依次为 $left1, left2, \dots, left11$, F 为各头碎片同行的灰度矩阵集合。

step4: 类比问题一的作法对11张横切行碎片根据上下端特征复原。

其中，在step2中也可以利用复原同行碎片的图论模型复原行碎片，将同行碎片重新编号，使其编号分别等于该图在b中所处的列号。

人工干预方式与时间节点

在step3中，对每行碎片利用复原同行碎片的最优化模型复原时，当两图片间的 ε 值虽较其余的小，但图片实际上是不连续的，这时候就需要人工干预进行校正，以获得较好的图像

还原结果。此外还可能发生多组合的 ε 值相同且最优，这时候也需要人工干预获得连续的相邻图片组合。针对附件3的初次复原图，找出图片不连续的行进行人工干预，共有四行的图片不完全连续，人工干预方式为：从属于该行的不连续的图片中找出与不连续处左端相连续的图，经过人工统计的干预总次数为11次。针对附件4的初次复原图，找出图片不连续的行进行人工干预，共有六行的图片不完全连续，人工干预方式也为：从属于该行的不连续的图片中找出与不连续处左端相连续的图，经过人工统计的干预总次数为25次。

III.2.3 问题二模型求解

由Matlab程序求解的：

问题2附件3图片拼接序号.

7	208	138	158	126	68	175	45	174	0	137	53	56	93	153	70	166	32	196
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
38	148	46	161	24	35	81	189	122	103	130	193	88	167	25	8	9	105	74
49	54	65	143	186	2	57	192	178	118	190	95	129	28	91	188	141	11	22
61	19	78	67	69	99	162	96	131	79	163	72	6	177	20	52	36	116	63
71	156	80	33	202	198	15	133	170	205	85	152	165	27	60	205	165	60	207
89	146	102	154	114	40	151	207	155	140	185	108	117	4	101	113	194	119	123
94	34	84	183	90	47	121	42	124	144	77	112	149	97	136	164	127	58	136
125	13	182	109	197	16	184	110	187	66	106	150	21	173	157	181	204	139	145
168	100	76	62	142	30	41	23	147	191	50	179	120	86	195	26	1	86	195

问题2附件4图片拼接序号.

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

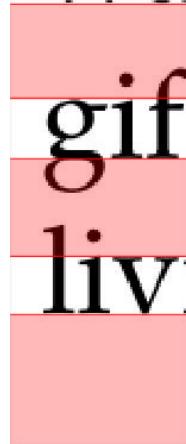
III.3 问题三

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。

III.3.1 问题三的分析

本间考虑点如下：

1. 碎片内容为英文字母，由于英文字符本身长比划较多(例如，字母d,y,l,h)，而这些较长比划使行与行间的空白部分支离破碎，碎片图像的空白行特征不明显，如图。



在上图中浅红色区域代表行间的空白部分，由图可发现空白部分被字母g、i、f、l分割，难以得到空白的准确位置，继而难以用空白位置描述两碎片的匹配程度。

2. 共418张图片，且涉及正反面，若按照问题一、二的方法两块碎片间比较匹配程度，处理时间就且匹配结果准确度将大大下降。

3. 附件五中碎片的相似程度较大，沿用问题一二的特征因子已经不能解决本文。

应对方案如下：

1. 针对英文字符的较长比划多的特点，对碎片图像进行预处理，“抹去”长比划，使得行间空白更为完整。

2. 针对碎片差异较为明显的行特征新设行特征因子组，使得行特征因子组能够“高效”反应碎片信息。利用聚类分析的方法将419张碎片聚为11类，同类的大部分碎片处于同行。

III.3.2 问题三模型建立

英文碎片图像预处理方法

对拥有较长比划的字母全体做出本文的界定，有以下字母：

b、d、f、g、h、i、j、k、l、p、q、t、y以及所有大写字母。

对附件五前20张碎片图中出现“长比划”的现象进行统计，结果如下。

$$\omega = \frac{\text{本碎片空白行中“长比划”的出现总个数}}{\text{空白行数}}$$

ω 为碎片每个空白行出现的“长比划”的个数。

前20张碎片中每个空白行出现的“长比划”的个数

碎片编号	1	2	3	4	5	6	7	8	9	10
ω	0.50	2.00	3.00	1.00	1.00	0.33	0.75	0.33	1.00	1.00
碎片编号	11	12	13	14	15	16	17	18	19	20
ω	0.67	2	0.33	1.00	1.25	0.45	1.00	1.67	1.25	0.75

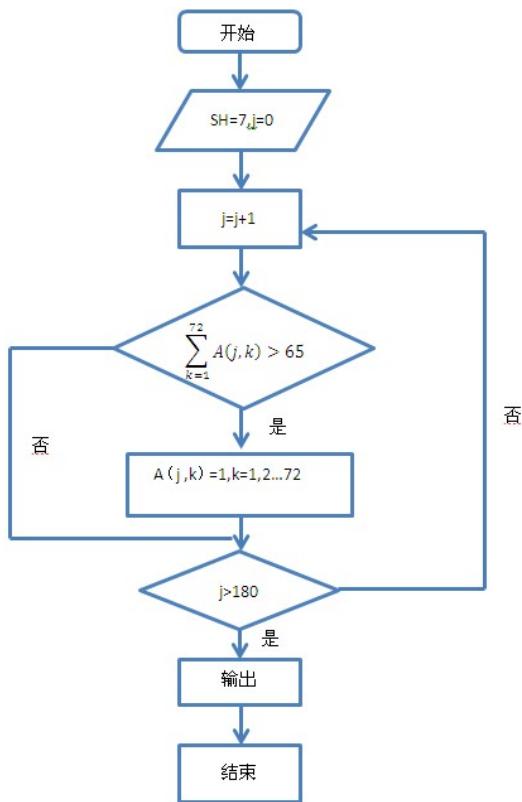
碎片图像中每个空白行出现“长比划”个数的范围为0.33~2.00。

为了让接下来的判断更为容易，二值化碎片的灰度矩阵，得到只有黑色和白色的碎片图。

抹去“长比划”基本思想：给定阈值SH,对碎片分别计算每行的黑像素总数。若总数小于给定阈值SH, 即认为本像素行处于空白行中，且行中的黑像素全是“长比划”的一部分，继而重新设定该行的每个像素都为白像素，即抹去该长比划。

根据统计每个“长比划”在空白行中的平均宽度为5像素。若阈值设定太小，则不能抹掉大部分的“长比划”；若阈值设定太大，则会抹掉不是“长比划”的那些比划。为了较大程度的兼顾二者，根据上述统计数据，在反复调试后得到一个效果比较理想的阈值 $SH=5 \times 1.4 = 7$ （像素）。

上述思想的具体操作步骤见如下程序框图：



其中A代表碎片的二值矩阵， $A(k,j)$ 代表二值矩阵中第k行，j列元素。

碎片预处理前后图片比较如下：



由图看出右端图片字母d的“长比划”被抹去。

行特征因子组定义

对于碎片定义行特征因子 $\theta_1, \theta_2, \eta_1, \eta_2$

θ_1 为从碎片顶端像素开始向下连续白像素的个数。

θ_2 为从碎片底端像素开始向上连续白像素的个数。

η_1 为从碎片顶端像素开始向下连续黑像素的个数。

η_2 为从碎片底端像素开始向上连续黑像素的个数。

利用各碎片指标值的方差来描述各碎片此行特征的差异度，差异度越大，则利用该指标区分度越大，则指标的选取较优。

通过计算指标值 $\theta_1, \theta_2, \eta_1, \eta_2$ 的方差分别为：583.9479、500.0132、71.7089、29.0768。可见 θ_1, θ_2 指标选取相当好， η_1, η_2 指标选取较优。

观察发现，每组碎片的a、b图的行特征都相同，可知文档正反面的页边距完全相同，所以若按行特征分类，任意组的a、b图都将分入同一行。

可根据四个指标利用聚类分析的方法将 209×2 张图分为11类，因为指标都描述的是行特征，故可认为同类的绝大部分碎片属于同行，有模型求解得到的结果可验证这一点。对聚类后的同行碎片进行人工干预，得到11类碎片，每类中有19个碎片，且每类中的19个碎片都属于同一行。故接下来只要先对同行碎片复原，再复原全文。

同行碎片复原模型与算法

根据观察图片可知，位于行首及行尾的图片位于左端的全空白部分宽度较大，可作为与其他图片相区别的重要特征。通过matlab编程调整限制条件左端全白列部分宽度及右端全白部分宽度，可以求出左端全空白部分18的图片个数有13个，右端全空白部分18的图片个数有11个，比较这两组数，取其公共部分（共11个）作为某一面的行首，得到11个行首编号为left1, left2, ..., left11。

设立匹配指标函数，当两张图片是相邻的情况下，前一张正面的最后一列得与后一张正面的第一列的匹配值要比其它组合小，前一张反面的第一列得与后一张反面的最后一列的匹配值要比其它组合小。考虑到图片为灰度图，做如下匹配指标函数模型：

$$W = \frac{w_1 + w_2 + w_3}{180}$$

其中，

$$w_1 = \sum_{i=1}^{180} (a_i - b(i))^2$$

。 a_i, b_i 分别为前一张图灰度矩阵的最后一列的第*i*行数据和后一张图灰度矩阵的第一列的第*i*行数据。

$$w_2 = \sum_{i=1}^{180} (c_i - d(i))^2$$

。 c_i, d_i 分别为前一张图的反面灰度矩阵的第一列的第*i*行数据和后一张图的反面灰度矩阵

的第一列的第i行数据。

$$w_3 = (\theta_1^1 - \theta_1^2)^2$$

表示两矩阵特征因子 θ_1 相似程度。 θ_1^1 、 θ_1^2 分别表示前一张图和后一张图的特征因子 θ_1 值。

先以left1所在行为例给出同行复原模型与算法。为方便操作，可先将本行的图片重新编号令碎片left1的新编号为1，剩下碎片为编号2,...,19。

step1：建立集合F用于存放未定顺序的碎片。

建立19列的顺序行向量b,并将其初始化0向量，用于存放正面重新排列顺序(即复原)后的碎片编号。b(1)=1.

k为计数变量，k=1。

step2:为b(k)碎片的正面匹配右相邻的碎片正面图。求解下述最优化模型。设QI代表前一张碎片，HO代表后一张碎片;qi代表前一张碎片编号，ho代表后一张碎片编号。

$$\begin{aligned} &\min W \\ &\begin{cases} qi = b(k) \\ HO \in F \end{cases} \end{aligned}$$

得到碎片b(k)最优匹配HO*,则

F=F\HO*;

k=k+1;

b(k)=ho*;

若此时k=19，运算结束，否则重新开始step2。

11个行都复原了之后，同理问题二方法对11行进行排序，得到复原图。

人工干预方式与时间节点

在step2中，当两图片间的W值虽较其余的小，但图片实际上是不连续的，这时候就需要人工干预进行校正，以获得较好的图像还原结果。此外还可能发生多组合的W值相同且最优，这时候也需要人工干预获得连续的相邻图片组合。在用matlab求出每行的排序后，将行排序得到初步的图形，观察图形不连续的部分，之后就针对这些部分用人工干预的手段重新排序求得连续地图片组合。未经过人工干预之前得到的图片完整度为54.55%，经过23次的人工干预可得到原图。

III.3.3 问题三模型求解

以下给出最短距离法聚类结果，由于大部分类的碎片个数均为30~60个，故每类只给出其中8个碎片编号，若某类中总碎片个数小于等于8个，则给出全部碎片编号。完全结果数据见附录。

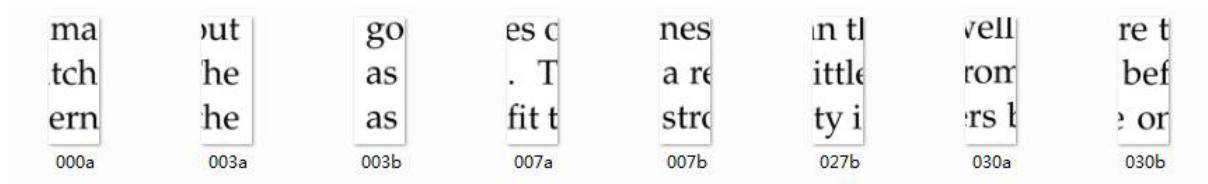
数据格式说明

表中编号为Matlab中记录的图片编号，未还原为附件5中的编号格式。编号为i的碎片即为附件中按照000a → 000b → 001a → 001b顺序排列的第i张图。

最短距离法聚类结果

类别								
1	1	7	8	15	16	56	61	62
2	11	22	29	30	33	39	43	52
3	9	92	124	170	276			
4	250							
5	3	4	5	6	10	17	18	19
6	325							
7	23	31	47	48	67	70	80	98
8	2	55	162	211	254	272	283	353
9	13	14	27	28	49	53	54	87
10	12							
11	273							

聚类后，同类中的大部分碎片属于同行。例如，下图为表中展示的类别1的8个碎片，明显属于同一行。



但是按照最小聚类法得到的分类方案，每类所拥有的碎片个数极不均衡，1~11类中的碎片数分别为61、51、5、1、61、1、32、10、46、1、1。故考虑用不同的聚类法对碎片聚类。

本问利用另外两种聚类方法——最长距离法和内平方距离法。聚类结果数据见附录。

比较上述聚类方法，应用内平方距离法后，每类所拥有的碎片个数均衡度是三种方法中最好的。故采用内平方距离法的结果数据进行人工干预，得到11类同行但未排序的碎片，数据如下表。

数据格式说明

1. 下表中每一行代表属于同行的但未排序的碎片编号。

2. 下表中编号i代表附件5中ia、ib正反两张图。

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

最终复原图的碎片排序为：

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

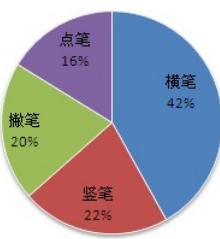
IV 模型优化

IV.1 基于汉字的拼接复原模型^[11]

IV.1.1 模型分析

根据文献【8】，汉字从书写形式看，是一种平面型方块体，汉字的所有笔画都有秩序地分布在平面的方框中，这是汉字从外观上最明显的特点，这也就是我们常说的汉字独特的“方块型”结构。通常认为，汉字的基本笔画是横，竖，撇，点，折。对汉字的各种笔画的出现频率进行统计，张性初等人在1965年《心理学报》“汉字的各种笔画的使用频率的估计”中统计结果为：横笔占31%，竖笔占16%，撇笔占15%，点笔占12%（如下图），而张静贤在2004年《汉字教程》中统计结果为：横笔占27.68%，竖笔占17.60%，撇笔占15.95%，点笔占13.62%（如下图）。通过比较我们可以发现汉字在横笔的出现的频率最高。同时根据”GB130001字符集汉字字序（笔画序）规范“中的相关统计，目前使用的汉字总共有20902个，平均每个字12.8画，其中12画的汉字最多，共有1957个，而在”现代汉语常用字表“中，常用汉字为3500个，平均每个字9.7画，其中9画的汉字最多，一共415个。基于上述的数据，可以推断出”GB130001字符集“中，平均每个汉字有3.54画的横笔，而常用汉字中，平均每个汉字有2.68画的横笔，可以说横笔在整个汉字结构中出现频次最高，占用重要的地位。

1965年汉字笔画使用频率



2004年《汉字教程》统计结果



根据上述统计，结合实际的碎片的情况，在每张碎片中都会出现相当多的完整或不完整的汉字结构，而依照统计规律，每个完整或不完整的汉字结构中，都可能出现多个”横笔“的结构，利用其突出的特点，建立汉字内部不同结构之间的联系，匹配不同碎片的汉字结构，进而拼接整张图像，以达到恢复整张图片的目的。

IV.1.2 模型建立

根据二值化矩阵沿着碎片的左右边缘向内进行搜索，准确记录碎片图像中各个汉字边缘处“横笔”的位置，根据碎片图像中左右边列处的汉字“横笔”位置，以两个图片之间的笔画匹配总数为依据，笔画匹配总数最大的两个图片就是相邻的图片

step1：建立 5×3 的二值化矩阵M：

1	1	1
0	0	0
0	0	0
0	0	0
1	1	1

沿着各个碎片的左右边缘向内进行搜索，向内搜索的范围是三个像素点，判断汉字的像素点是否能够满足矩阵M，若满足条件，认为该汉字结构是一个“横笔”，则将其直接延伸至图像的最左边或者最右边，否则，汉字的结构保持不变；搜索整个碎片，并且准确记录碎片图像中各个汉字边缘处“横笔”的位置。

step2：对比两个碎片图像中左右边列处的汉字“横笔”位置，若其中有“横笔”位置完全一致，则认为有一个汉字结构匹配上，并且记录两个碎片之间的总的笔画的匹配数；以当前碎片图像为基准，与其他碎片图像继续重复上述比较过程，最终，以两个碎片之间的笔画匹配总数为依据，笔画匹配总数最大的两个碎片就是相邻的碎片。同时，考虑到每个碎片图像中汉字个数，以及文章段落之间的结构特点等因素，设置了笔画匹配总数下限值，当至少有五个“横笔”位置一致时，认为两个碎片图像可能是邻近的，可以进行拼接处理。

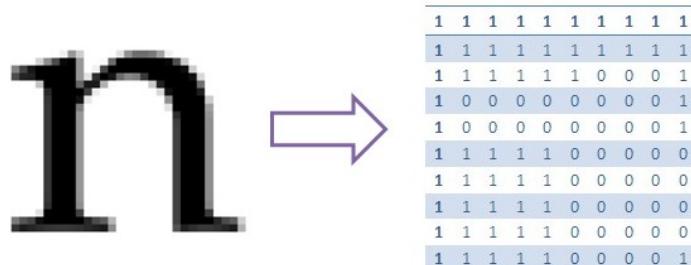
step3：为了避免出现多张碎片图像与同一张碎片图像相匹配的情况，根据最大相关性的原则，即“匹配上”横笔“总数最多的碎片是相邻图像，此时将两张碎片图像拼接在一起。

step4：通过以上步骤可以实现对中文碎片的拼接复原。

IV.2 基于字母的拼接复原模型

IV.2.1 模型分析

参考文献【7】，考虑到英文字母的特性，按照大写字母和小写字母，根据碎片图像的像素，可以做出52个字母的二值化矩阵模板，我们以小写字母n为例



得到的n的模板是关于 26×30 的二值化矩阵，由于矩阵信息量大，我们只截取了其中的一部分，完整的矩阵见附录

对边缘进行52个字母模型的匹配处理，根据最大匹配原则，可以推测出边缘的字母原型，我们取180b碎片进行上述做法的论证

ne t
ste
eeks

按照上述原则，可以得到该碎片左边缘的补全字母为m和e，右边缘的补全字母为t和s，下边缘的补全字母为e,e,k,s,按照残缺字母像素所在的模板矩阵预测出相邻碎片的边缘像素的位置。

IV.2.2 模型建立

step1:设52个字母中第i个字母的模块矩阵中元素为0的个数为I，碎片中边缘残缺字母二值化矩阵的0元素与52个字母的模块矩阵的0元素(黑像素)进行匹配，令 x_i 表示残缺字母与第i个字母的模块矩阵中的0元素(黑像素)匹配上的对数

目标函数：

$$x_p = \max \frac{x_i}{I}, i = 1, 2, \dots, 52$$

x_p 表示残缺字母与第p个字母配对的最大比率，即认为该残缺字母为p字母

step2:按照上述原则找出每个碎片的四条边残缺字母的原字母，其中若某个边界为空，则跳过此条边

step3:统计碎片j左右两边缘的补全字母的内容以及位置，将碎片j左边的补全字母的原形式与碎片i右边的补全字母的原形式按照所属模板矩阵进行匹配，先按照两边缘补全字母的内容进行匹配，若出现多个字母匹配的情况，再按照字母的位置进行配对。再结合问题二以及问题三中的方法，最终可得到，按行求解得到的最优匹配结果

step4:每行的最优匹配结果得到之后，按照上下边缘的字母的配对情况，最终得到完整的匹配图像

IV.2.3 模型评价

本模型是针对英文单词由52个大写字母与小写字母独立构成的特征而建立边缘残缺字母补全之后再进行匹配的模型，这是区别于汉字字符的不同之处，再结合问题二以及问题三中的匹配方式，可让模型的匹配效率得以提高。

参考文献

- [1] 王冬平,王清贤,罗军勇,李炳龙, BMP图像碎片重组中的候选权重方法, 计算机应用, 27(12): 3062-3065, 2007.
- [2] 罗智中, 基于文字特征的文档碎纸片半自动拼接, 计算机工程与应用, 48(5): 207-210, 2012.

- [3] 黄永安,李文成,高小科, Matlab 7.0/Simulink 6.0 应用实例仿真与高校算法开发, 北京: 清华大学出版社, 2008年6月.
- [4] 王沫然, Matlab6.0与科学计算, 北京: 电子工业出版社,2001年9月.
- [5] 罗智中, 基于文字特征的文档碎纸片半自动拼接, 计算机工程与应用, 48(5):207-P210,2012.
- [6] 李卓,邱慧娟, 基于相关系数的快速图像匹配研究, 北京理工大学学报, 27(11):998-1000,2007.
- [7] 豆丁网, 基于BP神经网络的字母识别系统: [论文] <http://www.docin.com/p-467961357.html> , 访问时间(2013年9月14日).
- [8] 邢楠,张婧等, 基于文字特征的碎纸机破碎文档的恢复方法: [发明专利] <http://pan.baidu.com/share/link?shareid=1119097902&uk=152125572&app=zd> , 访问时间(2013年9月14日).

V 附录

本队用以参赛软件为Matlab（2010b）。

V.1 图表数据结果

附件一复原图

城上层楼叠巘。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。为谁流睇不归家。
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，情人道、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。临水纵横回晚鞞。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。
簟纹如水玉肌凉。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。

附件二复原图

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-

附件三复原图

便邮。温香熟美。醉慢云鬟垂两耳。多谢春工。红是玉红。一颗樱 不是花桃樊素口。不爱黄金，只爱人长久。学画鸦儿 眉尖已作伤春皱。犹未就。清泪斑斑，挥断柔肠寸。嗔人问。背灯尽残妆。春事阑珊芳草 盈弔偷歇。客里风光，又过清明节。小院黄昏，落红处处闻啼鴂。岁云暮，别。人忆须早计，要褐裘。故乡归去千里，佳处 我醉歌时君和，醉倒须君 留。辄迟扶我，惟酒可忘忧。一任刘玄德，相对卧高楼。记取西湖西畔，正取西湖西处，空翠烟霏。算诗人相得，如我与君稀。约他年、东还海道，愿趁东还海志莫相违。西州路，不应回首，为我沾衣。料峭春风吹酒醒。微冷。风 吻洒 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。

九贪忙何处追游。三分春色一分愁。雨翻榆荚阵，风 分愁榆荚风 且 转柳花坐间。爱君才器两俱全。异乡风景却依然。团扇只 乡风然。只 相如 堪题往行人。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。别酒劝君君一酌，清润潘郎，又是何郎婿。记取钗头新利市。莫将分付东邻子。西塞山边邻子 鶯飞。散花洲外片帆微。桃花流水鳜鱼肥。主人瞋小。欲向东风先醉。欲 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。此事如何到得侬。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

附件四复原图

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

附件五复原图

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 now 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

问题三聚类分析结果

最长距离法

1	2	3	4	5	6	7	8	9	10	11
11	1	3	4	10	9	273	23	35	2	13
22	7	6	5	26	17		31	37	12	14
29	8	36	19	32	18		47	38	55	27
30	15	75	20	34	21		48	41	162	28
33	16	84	24	40	45		67	42	211	49
39	56	131	25	50	46		70	44	254	53
43	61	141	68	57	51		80	58	272	54
52	62	182	69	63	73		98	59	283	87
64	65	216	76	101	74		103	60	353	88
81	66	229	79	102	92		106	71	371	99
82	77	231	83	104	93		126	72	410	100
89	78	234	97	105	94		143	95		109
90	85	279	125	108	124		165	96		110
107	86	299	132	113	138		167	117		111
119	91	303	142	116	147		192	129		112
120	114	310	144	118	153		195	133		115
121	123	341	145	186	160		204	134		130
122	137	360	146	191	170		237	148		183
127	139	362	166	203	179		240	157		184
128	140	384	168	205	185		259	158		193
135	149	407	175	212	235		268	163		194
136	150	415	176	230	241		314	164		199
151	155		177	233	242		321	217		200
152	156		178	238	250		324	218		201
154	161		181	239	256		339	221		202
159	169		187	267	257		347	222		206
180	171		188	293	258		351	223		207
236	172		196	318	276		363	224		208
245	173		215	323	290		389	251		209
246	174		232	325	295		399	252		210
249	189		260	335	296		412	261		213

262	190		265	344	305		413	274		214
289	197		266	348	315			281		219
294	198		280	355	316			282		220
306	243		291	356	345			286		225
320	244		292	359	346			301		226
328	253		300	364	357			302		227
331	255		304	382	377			309		228
332	263		313	386	404			311		247
336	264		322	390				312		248
337	271		326	396				317		269
338	275		333	403				319		270
343	277		334	405				327		285
358	278		340	406				329		393
365	284		342	411				330		394
366	287		352	416				349		417
381	288		361					350		
385	297		383					367		
387	298		397					368		
391	307		398					369		
392	308		400					370		
	354		408					379		
	372		414					380		
	373							388		
	374							395		
	375							418		
	376									
	378									
	401									
	402									
	409									

最短距离法

1	2	3	4	5	6	7	8	9	10	11
1	11	9	250	3	325	23	2	13	12	273
7	22	92		4		31	55	14		
8	29	124		5		47	162	27		
15	30	170		6		48	211	28		
16	33	276		10		67	254	49		
56	39			17		70	272	53		
61	43			18		80	283	54		
62	52			19		98	353	87		
65	64			20		103	371	88		
66	81			21		106	410	99		
77	82			24		126		100		
78	89			25		143		109		
85	90			26		165		110		
86	107			32		167		111		
91	119			34		192		112		
114	120			35		195		115		
123	121			36		204		130		
137	122			37		237		183		
139	127			38		240		184		
140	128			40		259		193		
149	135			41		268		194		
150	136			42		314		199		
155	151			44		321		200		
156	152			45		324		201		
161	154			46		339		202		
169	159			50		347		206		
171	180			51		351		207		

172	236			57		363		208		
173	245			58		389		209		
174	246			59		399		210		
189	249			60		412		213		
190	262			63		413		214		
197	289			68				219		
198	294			69				220		
243	306			71				225		
244	320			72				226		
253	328			73				227		
255	331			74				228		
263	332			75				247		
264	336			76				248		
271	337			79				269		
275	338			83				270		
277	343			84				285		
278	358			93				393		
284	365			94				394		
287	366			95				417		
288	381			96						
297	385			97						
298	387			101						
307	391			102						
308	392			104						
354				105						
372				108						
373				113						
374				116						
375				117						
376				118						
378				125						
401				129						
402				131						
409				132						

内平方距离法

1	2	3	4	5	6	7	8	9	10	11
37	4	17	26	1	10	3	2	13	9	23
38	5	18	34	7	19	6	12	14	11	31
42	25	21	40	8	20	36	55	27	22	47
44	35	45	50	15	24	75	162	28	29	48
95	41	46	57	16	32	84	211	49	30	67
133	58	51	63	56	68	131	254	53	33	70
134	59	73	101	61	69	141	272	54	39	80
148	60	74	102	62	79	182	283	87	43	98
157	71	93	108	65	97	216	353	88	52	103
163	72	94	116	66	104	229	371	99	64	106
164	76	138	118	77	105	231	410	100	81	126
217	83	153	147	78	113	234		109	82	143
218	96	160	185	85	125	279		110	89	165
221	117	179	186	86	132	299		111	90	167
222	129	235	205	91	144	303		112	92	192
223	142	241	230	114	145	310		115	107	195
224	158	242	233	123	146	341		130	119	204
251	181	256	293	137	166	360		183	120	237
261	215	257	318	139	168	362		184	121	240
274	252	258	325	140	175	384		193	122	259
282	281	290	344	149	176	407		194	124	268
301	286	295	355	150	177	415		199	127	273
311	302	296	356	155	178			200	128	314
312	304	305	359	156	187			201	135	321
327	309	315	382	161	188			202	136	324

329	317	316	396	169	191			206	151	339
330	319	345	403	171	196			207	152	347
349	333	346	404	172	203			208	154	351
350	342	357	405	173	212			209	159	363
368	361	377	406	174	232			210	170	389
379	367		416	189	238			213	180	399
380	369			190	239			214	236	412
	370			197	260			219	245	413
	388			198	265			220	246	
	395			243	266			225	249	
	408			244	267			226	250	
	418			253	280			227	262	
				255	291			228	276	
				263	292			247	289	
				264	300			248	294	
				271	313			269	306	
				275	322			270	320	
				277	323			285	328	
				278	326			393	331	
				284	334			394	332	
				287	335			417	336	
				288	340				337	
				297	348				338	
				298	352				343	
				307	364				358	
				308	383				365	
				354	386				366	
				372	390				381	
				373	397				385	
				374	398				387	
				375	400				391	
				376	411				392	
				378	414					
				401						
				402						
				409						

模型优化字母n的模型矩阵

V.2 Matlab程序代码

本队用以参赛软件为Matlab (2010b)。

复原附件一

```
clear all;clc; % 导入附件1图像矩阵  
cell{1,1}=imread('000.bmp');  
cell{1,2}=imread('001.bmp');  
cell{1,3}=imread('002.bmp');  
cell{1,4}=imread('003.bmp');  
cell{1,5}=imread('004.bmp');  
cell{1,6}=imread('005.bmp');  
cell{1,7}=imread('006.bmp');  
cell{1,8}=imread('007.bmp');  
cell{1,9}=imread('008.bmp');  
cell{1,10}=imread('009.bmp');  
cell{1,11}=imread('010.bmp');  
cell{1,12}=imread('011.bmp');  
cell{1,13}=imread('012.bmp');  
cell{1,14}=imread('013.bmp');  
cell{1,15}=imread('014.bmp');  
cell{1,16}=imread('015.bmp');
```

```

cell{1,17}=imread('016.bmp');
cell{1,18}=imread('017.bmp');
cell{1,19}=imread('018.bmp');
for i=1:19
    level=graythresh(cell{1,i});
    cell1{1,i}=im2bw(cell{1,i},level); %图像二值化处理
end
for i=1:19
    for k=1:19
        xs(i,k)=0;
    for j=1:1980
        if(cell1{1,i}(j,72)==cell1{1,k}(j,1)) %求相似度的矩阵
            xs(i,k)=1+xs(i,k);
        end
    end
    end
    end
%load 'ti1.mat';
for i=1:19
    xs(i,i)=0;
end
for i=1:19
    da(i)=max(xs(i,:));
end
wei=find(da==max(da));
for i=1:19
    k=find(xs(i,[1:19])==da(i)); %求出两两相邻的矩阵
    lian(i,1)=i; %为前者
    lian(i,2)=k; %为后者
end
lian(wei,1)=0;
tou=lian(wei,2);
xu(1)=tou;
for i=1:18
    xu(i+1)=lian(xu(i),2); %正确的图像排序序列
end
%根据求出的序列xu画图
for i=1:19
    I(:,[72*(i-1)+1:72*i])=cell1,xu(i); %将图复原
end

```

```

imwrite(I,'hanzi.jpg','quality',100);
imshow('hanzi.jpg') %输出图像
复原附件二
clear all;clc; %导入附件2图片矩阵
cell{1,1}=imread('000.bmp');
cell{1,2}=imread('001.bmp');
cell{1,3}=imread('002.bmp');
cell{1,4}=imread('003.bmp');
cell{1,5}=imread('004.bmp');
cell{1,6}=imread('005.bmp');
cell{1,7}=imread('006.bmp');
cell{1,8}=imread('007.bmp');
cell{1,9}=imread('008.bmp');
cell{1,10}=imread('009.bmp');
cell{1,11}=imread('010.bmp');
cell{1,12}=imread('011.bmp');
cell{1,13}=imread('012.bmp');
cell{1,14}=imread('013.bmp');
cell{1,15}=imread('014.bmp');
cell{1,16}=imread('015.bmp');
cell{1,17}=imread('016.bmp');
cell{1,18}=imread('017.bmp');
cell{1,19}=imread('018.bmp');
for i=1:19
level=graythresh(cell{1,i});
cell1{1,i}=im2bw(cell{1,i},level); %为方便求解先进行图像二值化处理
end
for i=1:19
for k=1:19
xs(i,k)=0;
for j=1:1980
if(cell1{1,i}(j,72)==cell1{1,k}(j,1)) %构造相似度矩阵
xs(i,k)=1+xs(i,k);
end
end
end
end
%load 'ti1.mat';
for i=1:19
xs(i,i)=0;

```

```

end
for i=1:19
da(i)=max(xs(i,:));
end
wei=find(da==max(da));
for i=1:19
k=find(xs(i,[1:19])==da(i)); %求出两两相邻矩阵
lian(i,1)=i; %为前者
lian(i,2)=k; %为后者
end
lian(wei,1)=0;
lian(wei,1)=0;
tou=lian(wei,2);
xu(1)=tou;
for i=1:18
xu(i+1)=lian(xu(i),2); % 求出正确排序的序列
end
%load 'xu.mat';
%根据以上图片排序序列xu复原图片
for i=1:19
I(:,[72*(i-1)+1:72*i])=cell{1,xu(i)};
end
imwrite(I,'English.jpg','quality',100);
imshow('English.jpg') %输出图片
复原附件三
%找每行的行首图片
clear all;clc;
load 't3.mat';
for i=1:209
level=graythresh(t3{i,1});
bm{i,1}=im2bw(t3{i,1},level); %图像二值化处理
end
k=0;
for i=1:209
b=0;
for j=1:9
b=b+sum(bm{i,1}(:,j));
end
if(b==180*9)
k=k+1;

```

```

left(k)=i; %每行的行首
end
end
k=0;
for i=1:209
b=0;
for j=1:9
b=b+sum(bm{i,1}{:,j+63});
end
if(b==180*9)
k=k+1;
right(k)=i; %每行的行尾
end
end
k=0;
for i=1:209
b=0;
for j=1:38
b=b+sum(bm{i,1}{j,:});
end
if(b==72*38)
k=k+1;
up(k)=i; %第一行
end
end
k=0;
for i=1:209
b=0;
for j=1:61
b=b+sum(bm{i,1}{j+119,:});
end
if(b==72*61)
k=k+1;
down(k)=i; %最后一行
end
end
%搜寻每个行首的图片所在行的其余图片
k=0;
for i=1:209
b=0;

```

```

for j=1:9
b=b+sum(bm{i,1}{:,j});
end
if(b==180*9)
k=k+1;
tou(k)=i; %每行第一个
end
end
hang=zeros(209,180);
for k=1:209
for i=1:180
if(sum(bm{k,1}{i,:})==72) %说明这行是空白的
hang(k,i)=0;
else
hang(k,i)=1;
end
end
end
for i=1:11
tu{tou(i),1}{:,1} = -1*ones(180,1);
end
for k=1:11
for i=1:209
for j=1:11
c(k,tou(j))=inf;
end
c(k,i)=sum(abs((hang(tou(k),:)-hang(i,:))));
end
end
for k=1:11
for p=1:18
xiao=min(c(k,:));
[i,j]=find(c(k,:)==xiao);
h(k,p)=j(1); %每行表示属于第k行除行首外的图片
c(k,j(1))=inf;
end
end
for i=1:18
for k=1:18
xs(i,k)=0;

```

```

for j=1:180
if(bm{h(1,i),1}(j,72)==bm{h(1,k),1}(j,1))
xs(i,k)=1+xs(i,k);
end
end
end
end
h;%每行的元素个数
%求解每行的排序,以n=50和15为例:
clear all;clc;format short
load 't3.mat';
load 'hang.mat';
bm=t3;
%求解每行的排序以行首是50的为例:
n=50
d=[3 12 23 29 193 55 58 66 92 96 119 130 142 144 179 187 189 191];
afa=1;beta=0;
for ci=1:18
for i=1:18
ca(i)=0;
for j=1:180
w=(bm{n,1}(j,72)-afa*bm{d(i),1}(j,1)-beta)^2/180;
ca(i)=w+ca(i);
end
end
k=find(ca==min(ca));
lo=length(k);
if(lo<1)
k
for i=1:lo
s(:,[1:72])=bm{n,1};
s(:,[73:144])=bm{d(k(i)),1};
imwrite(s,'lena.jpg','quality',100);
figure;
imshow('lena.jpg')
end
k=input('人工干预');
end
xu(ci)=d(k);
bm{d(k),1}(:,1)=-inf*ones(180,1);

```

```

n=d(k);
end
xu %最终序列
%求解每行的排序以行首是50的为例:
n=15
d=[129,4,160,83,200,136,13,74,161,204,170,135,40,32,52,108,116,177];
afa=1;beta=0;
for ci=1:18
for i=1:18
ca(i)=0;
for j=1:180
w=(bm{n,1}{j,72}-afa*bm{d(i),1}{j,1}-beta)^2/180;
ca(i)=w+ca(i);
end
end
k=find(ca==min(ca));
lo=length(k);
if(lo>1)
k
for i=1:lo
s(:,[1:72])=bm{n,1};
s(:,[73:144])=bm{d(k(i)),1};
imwrite(s,'lena.jpg','quality',100);
figure;
imshow('lena.jpg')
end
k=input('人工干预');
end
xu(ci)=d(k);
bm{d(k),1}{:,1}=-inf*ones(180,1);
n=d(k);
end
xu %最终序列
clear all;clc;
load 'ht.mat';
load 'BMP3.mat';
t3=BMPfile3;
bm=t3;
for i=1:11
for j=1:19

```

```

AI([(1+(i-1)*180):(180*i)],[(1+(j-1)*72):(72*j)])=t3{ht(i,j),1};
end
end
for i=1:11
bmp{i,1}=AI(1+(i-1)*180:180*i,1:72*19);
end
d=[5 6 11 4 7 2 9 10 3 1 8]; %在这里人工干预调整行的排列顺序得到最终复原图
for i=1:11
AI([(1+(i-1)*180):(180*i)],:) bmp{d(i),1};
end
imwrite(AI,'hangtu.jpg','quality',100);
imshow('hangtu.jpg')

复原附件四
clear all;clc;
load 't4.mat';
%求出误差矩阵
for i = 1:209
t4{i,1} = double(t4{i,1}); %转化为double型
end
bm=t4;
W=inf*ones(209,209); %W为误差矩阵
afa=1;beta=1;
for i=1:209
for j=1:209
W(i,j)=0;
for k=1:180
w=(bm{i,1}(k,72)-afa*bm{j,1}(k,1)-beta)^(1/180);
W(i,j)=W(i,j)+w;
end
end
end
%图像二值化处理
for i=1:209
level=graythresh(t4{i,1});
bm{i,1}=im2bw(t4{i,1},level);
end
%找出每一行的行首图片
k=0;
for i=1:209
b=0;

```

```

for j=1:9
b=b+sum(bm{i,1}{(:,j)});
end
if(b==180*9)
k=k+1;
left(k)=i; %每行第一个
end
end
k=0;
for i=1:209
b=0;
for j=1:9
b=b+sum(bm{i,1}{(:,j+63)});
end
if(b==180*9)
k=k+1;
right(k)=i; %每行最后一个
end
end
k=0;
for i=1:209
b=0;
for j=1:38
b=b+sum(bm{i,1}{(j,:)});
end
if(b==72*38)
k=k+1;
up(k)=i; %第一行
end
end
k=0;
for i=1:209
b=0;
for j=1:61
b=b+sum(bm{i,1}{(j+119,:)});
end
if(b==72*61)
k=k+1;
down(k)=i; %最后一行
end

```

```

end
%通过人工干预，找出其中不属于行首的147
left=[20,21,71,82,87,133,'147',160,172,192,202,209];
load 'W.mat';
load 't4.mat';
for i = 1:209
t4{i,1} = double(t4{i,1}); %转化为double型
end
bm=t4;
left=[20,21,71,82,87,133,'147',160,172,192,202,209];
%以21为行首的其余图片矩阵为例：
n=21;
for ci=1:18
k=find(W(n,:)==min(W(n,:)));
lo=length(k);
if(lo>1)
k
for i=1:lo
s(:,[1:72])=bm{n,1};
s(:,[73:144])=bm{k(i),1};
imwrite(s,'lena.jpg','quality',100);
figure;
imshow('lena.jpg')
end
b=input('人工干预'); %当出现多个可与当前匹配项时进行人工干预，输入b为人认为
较符合匹配的序号
if(b==0)
for i=1:lo
W(n,k)=inf;
end
close all;
else
k=k(b);
end
end
k=k;
W(n,k)=inf;
be(ci)=k;
n=k;
close all; end

```

```

be %输出以21为行首的其余图片矩阵
%将n变为71得到以71为行首的其余图片矩阵:
n=71;
for ci=1:18
k=find(W(n,:)==min(W(n,:)));
lo=length(k);
if(lo>1)
k
for i=1:lo
s(:,[1:72])=bm{n,1};
s(:,[73:144])=bm{k(i),1};
imwrite(s,'lena.jpg','quality',100);
figure;
imshow('lena.jpg')
end
b=input('人工干预'); %当出现多个可与当前匹配项时进行人工干预，输入b为人认为
较符合匹配的序号
if(b==0)
for i=1:lo
W(n,k)=inf;
end
close all;
else
k=k(b);
end
end
k=k;
W(n,k)=inf;
be(ci)=k;
n=k;
close all;
end
be %输出以71为行首的其余图片矩阵
clear all;clc;
ht=load('ht.txt');
load 't4.mat';
bm=t4;
for i=1:11
for j=1:19
AI([(1+(i-1)*180):(180*i)],[(1+(j-1)*72):(72*j)])=bm{ht(i,j),1};

```

```

end
end
for i=1:11
bmp{i,1}=AI(1+(i-1)*180:180*i,1:72*19);
end
d=[1 2 3 4 5 6 7 8 9 10 11]; %此为通过人工干预调整行的排序得到的正确行排序
for i=1:11
AI([(1+(i-1)*180):(180*i)],:) = bmp{d(i),1};
end
imwrite(AI,'hangtu.jpg','quality',100);
imshow('hangtu.jpg')
复原附件五
%求出每行的排序 (见Tfive.m) clear all;clc;
load 'hang1.mat';
load 'hang2.mat';
load 'tu.mat';
hang=load('hang.txt');
for i = 1:418
tu{i,1} = double(tu{i,1}); %× a? ^ ?adoubleD1
end
bm=tu;
for r=1:11
if(r==6)
continue; % μ
end
ju=hang(r,:);
n=length(ju);
W=inf*ones(n,n);
afa=1;beta=1;
for i=1:n
for j=1:n
W(i,j)=0;
w3=(hang1(i)-hang1(j))̂/180;
w4=(hang2(i)-hang2(j))̂/180;
for k=1:180
w1=(bm{ju(i),1}(k,72)-afa*bm{ju(j),1}(k,1)-beta)̂/180; %?ó2????ó
if((mod(ju(i),2)==1)&(mod(ju(j),2)==1))
w2=(bm{ju(j)+1,1}(k,72)-afa*bm{ju(i)+1+1,1}(k,1)-beta)̂/180;
elseif((mod(ju(i),2)==1)&(mod(ju(j),2)==0))
w2=(bm{ju(j)-1,1}(k,72)-afa*bm{ju(i)+1,1}(k,1)-beta)̂/180;

```

```

elseif((mod(ju(i),2)==0)&(mod(ju(j),2)==1))
w2=(bm{ju(j)+1,1}(k,72)-afa*bm{ju(i)-1,1}(k,1)-beta)̂/180;
elseif((mod(ju(i),2)==0)&(mod(ju(j),2)==0))
w2=(bm{ju(j)-1,1}(k,72)-afa*bm{ju(i)-1,1}(k,1)-beta)̂/180;
end
W(i,j)=W(i,j)+w1+w2+w3+w4;
end
end
end
n=1;
for ci=1:18
k=find(W(n,:)==min(W(n,:)));
if(k==n)
W(n,k)=inf;
k=find(W(n,:)==min(W(n,:)));
end
k=k(1);
be(r,ci)=ju(k);
W(n,k)=inf;
n=k;
for i=1:19
W(i,n)=inf;
end
end
end
%未人工干预前求得的复原图(见yuanben.m)
clear all;clc;
ht=load('chubu.txt');
load 'tu.mat';
bm=tu;
for i=1:11
for j=1:19
AI([(i-1)*180:(180*i)],[(j-1)*72:(72*j)])=bm{ht(i,j),1};
end
end
for i=1:11
bmp{i,1}=AI(1+(i-1)*180:180*i,1:72*19);
end
d=[1 2 3 4 5 6 7 8 9 10 11]; %此为通过人工干预调整行的排序得到的正确行排序
for i=1:11

```

```

AI([(1+(i-1)*180):(180*i)],:)=bmp{d(i),1};
end
figure(1)
imwrite(AI,'hangtu.jpg','quality',100); %输出图像
imshow('hangtu.jpg')
%用长方体圈出需要人工干预进行重新匹配的位置
% Create rectangle
annotation(figure(1),'rectangle',...
[0.364836017569546 0.844854070660522 0.152733528550512 0.0384024577572965],...
'FaceColor','flat');
% Create rectangle
annotation(figure(1),'rectangle',...
[0.523693997071742 0.660522273425499 0.113202049780381 0.0322580645161289],...
'FaceColor','flat');
% Create rectangle
annotation(figure(1),'rectangle',...
[0.364103953147877 0.466973886328725 0.272060029282577 0.0368663594470047],...
'FaceColor','flat');
% Create rectangle
annotation(figure(1),'rectangle',...
[0.363371888726208 0.359447004608295 0.0517086383601757 0.0291858678955454],...
'FaceColor','flat');
% Create rectangle
annotation(figure(1),'rectangle',...
[0.494411420204978 0.248847926267281 0.0626896046852123 0.032258064516129],...
'FaceColor','flat');
% Create rectangle
annotation(figure(1),'rectangle',...
[0.580062957540264 0.112135176651306 0.0553689604685211 0.0291858678955453],...
'FaceColor','flat');
title('用长方体圈出需要人工干预进行重新匹配的位置');
fprintf('完整度为百分之54.55n')
fprintf('所需人工干预次数为')
ci=ceil(1980*1368*0.5455/(180*72*5))
%输出正确图像 (fuyuan.m) clear all;clc;
load 'ht.mat';
load 'tu.mat';
bm=tu;
for i=1:11
for j=1:19

```

```

AI([(1+(i-1)*180):(180*i)],[(1+(j-1)*72):(72*j)])=bm{ht(i,j),1};
end
end
for i=1:11
bmp{i,1}=AI(1+(i-1)*180:180*i,1:72*19);
end
d=[1 2 3 4 5 6 7 8 9 10 11]; %此为通过人工干预调整行的排序得到的正确行排序
for i=1:11
AI([(1+(i-1)*180):(180*i)],:)=bmp{d(i),1};
end
imwrite(AI,'hangtu.jpg','quality',100); %输出图像
imshow('hangtu.jpg')
%求反面图片(fuyuanfan.m)
clear all;clc;
load 'ht.mat';
load 'tu.mat';
bm=tu;
for i=1:11
for j=1:19
if(mod(ht(i,j),2)==1 )
ht(i,j)=ht(i,j)+1;
else
ht(i,j)=ht(i,j)-1;
end
end
end
ht=fliplr(ht); %求反面矩阵
for i=1:11
for j=1:19
AI([(1+(i-1)*180):(180*i)],[(1+(j-1)*72):(72*j)])=bm{ht(i,j),1};
end
end
for i=1:11
bmp{i,1}=AI(1+(i-1)*180:180*i,1:72*19);
end
d=[1 2 3 4 5 6 7 8 9 10 11]; %此为通过人工干预调整行的排序得到的正确行排序
for i=1:11
AI([(1+(i-1)*180):(180*i)],:)=bmp{d(i),1};
end
imwrite(AI,'fantu.jpg','quality',100); %输出图像

```

```

imshow('fantu.jpg')
%此为功能函数，用于识别两个图像是否满足连续相邻，可共同目录下调用但不能
单独运行

function [k]=panbie1(W,bm,n)
k=find(W(n,:)==min(W(n,:)));
lo=length(k);
if(lo>1) %当出现多个较优的可行解时
k
for i=1:lo
s(:,[1:72])=bm{n,1};
s(:,[73:144])=bm{k(i),1};
imwrite(s,'lena.jpg','quality',100);
figure;
imshow('lena.jpg')
end
b=input('人工干预'); %判断是否有满足连续的邻接图
if(b==0)
for i=1:lo
W(n,k)=inf; %将这些不满足的情况剔除
end
close all;
[k]=panbie(W,bm,n); %循环调用函数
else
k=k(b);
end
end
s(:,[1:72])=bm{n,1};
s(:,[73:144])=bm{k,1};
imwrite(s,'lena.jpg','quality',100);
figure;
imshow('lena.jpg')
b=input('识别');
if(b==0)
W(n,k)=inf;
close all;
[k]=panbie(W,bm,n);
else
k=k;
W(n,k)=inf;
End

```

%此为功能函数，用于识别两个图像是否满足连续相邻，可共同目录下调用但不能单独运行

```
function [k]=shibie(W,bm,ju,n)
k=find(W(n,:)==min(W(n,:)));
lo=length(k);
if(lo>1) %当出现多个较优的可行解时
k
for i=1:lo
s(:,[1:72])=bm{ju(n),1};
s(:,[73:144])=bm{ju(k(i)),1};
imwrite(s,'lena.jpg','quality',100);
figure;
imshow('lena.jpg')
end
b=input('人工干预'); %判断是否有满足连续的邻接图
if(b==0)
for i=1:lo
W(n,k)=inf; %将这些不满足的情况剔除
end
close all;
[k]=shibie(W,bm,ju,n); %循环调用函数
else
k=k(b);
end
end
s(:,[1:72])=bm{ju(n),1};
s(:,[73:144])=bm{ju(k),1};
imwrite(s,'lena.jpg','quality',100);
figure;
imshow('lena.jpg')
b=input('识别');
if(b==0)
W(n,k)=inf;
close all;
[k]=shibie(W,bm,ju,n);
else
k=k;
W(n,k)=inf;
end
```