

# 碎纸片拼接复原的研究

## 摘要

本文针对碎纸片拼接复原的三个问题进行建模及求解。首先对碎纸片图形矩阵进行预处理，运用直方图谷点门限法对矩阵二值化；再确定碎纸片间的匹配度矩阵；最后通过全局匹配，实现碎纸片的拼接复原。

对于问题一，我们首先计算 19 幅碎纸片的匹配度矩阵，由边界处对应黑像素点数确定，建立 0-1 整数规划模型对 19 幅碎纸片进行拼接分配，从而使得其拼接后图形的匹配度总和最大，然后运用匈牙利算法通过 Matlab 编程对该模型进行求解。附件 1、2 的具体答案详见正文。

对于问题二的中文碎纸片处理，通过基于边界匹配的拼接算法，对碎纸片矩阵进行局部匹配，建立匹配度矩阵还需要考虑字符是否行对齐。然后对碎纸片进行全局匹配，确定原图左边界的 11 幅碎纸片，从左往右进行横向匹配。横向匹配根据匹配度矩阵生成匹配度最高的一整行碎纸片拼接图。当我们发现该行拼接图拼接错误时，进行人工干预，输入拼接错误的图的位置序号，通过程序将该处两幅图的匹配度改为 0。然后再根据修改后的匹配度矩阵横向匹配重新生成拼接图。得到正确的横向匹配拼接图后，再通过纵向匹配，模仿横向匹配的操作，完成原图的复原。上述算法通过 Matlab 编程进行求解，在附件 3 的复原过程中，我们一共人工干预了 18 次（人工干预一次是指将错误位置的图调整至正确为止）。附件 3 具体答案详见正文。

对于问题二的英文碎纸片处理，重复问题二的中文碎纸片处理步骤。在全局匹配过程中，我们加入自动字符识别来辅助匹配度进行边界匹配，以减少人工干预次数。自动字符识别基于模板匹配，运用二值图像间匹配度算法，基于实际情况考虑，采用二值图像间的Hausdorff距离来表示碎纸片间的匹配度。在附件 4 的复原过程中，我们一共人工干预了 21 次。附件 4 具体答案详见正文。

对于问题三，确定附件 5 中原图左边界的 22 幅碎纸片（正反面），根据正反面字符平齐的特点，建立 0-1 整数规划模型分配对齐的左边界碎纸片，通过 Lingo 编程求解。再利用问题二的英文碎纸片处理方法，对其进行横向复原，根据配对的左边界碎纸片以及纵向匹配度矩阵，完成附件 5 的纵向复原。对附件 5 求解，我们一共人工干预了 42 次。附件 5 具体答案详见正文。

**关键词：** 0-1 整数规划；全局匹配；字符识别；人工干预；

## 一、 问题重述

### 1.1. 背景

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。

但是，如何建立模型来获取准确率和拼接复原效率都较高的拼接方法，是本题的关键。

### 1.2. 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、 符号说明

$A_i$	碎纸片图形的矩阵
$A_{ijk}$	该附录中第 $k$ 幅碎纸片的图形矩阵的第 $i$ 行第 $j$ 列的元素
$b_i$	$A_i$ 右边界的黑像素点数
$b_j$	$A_j$ 左边界的黑像素点数
$b_{ij}$	$A_i$ 右边界与 $A_j$ 左边界相匹配的黑像素点数
$cov_{ij}$	$A_i$ 右边界与 $A_j$ 左边界的匹配度

## 三、 模型假设

1. 附件中所有的碎纸片图中，被切割的两幅图的切割边界矩阵中的黑点数近似。

2. 如果两个碎纸片能够准确拼接，则处于拼接处的英文字符能够被自动识别为有效的英文字符。
3. 每个待拼接的破碎文件有其统一的排版格式，尤其是字体和行间距。

## 四、 问题分析

### 4.1. 对问题一的分析

针对问题一，即如何建立模型与算法将纵切碎纸片复原，纵切碎纸片的左右两侧与之相连的碎纸片有关于黑像素点的对应关系，故生成其匹配度矩阵，再通过 0-1 规划找出其最优分配。

### 4.2. 对问题二的分析

针对问题二，即如何建立模型与算法将横纵切碎纸片复原，横纵切碎纸片左右两侧与之相连的碎纸片有对应关系，故也可生成其匹配度矩阵，匹配度矩阵还需考虑字符是否行对应。由于是横纵切，可以通过横向匹配得到 11 幅横向复原图片，再通过纵向匹配复原原图。

### 4.3. 对问题三的分析

针对问题三，即如何建立模型与算法将双面横纵切碎纸片复原，根据碎纸片正反应字符平齐的特点，可以进行左边界图片正反面匹配，再通过问题二的求解方法，将图复原。

## 五、 模型的建立与求解

### 5.1. 模型准备

#### 5.1.1. 预处理

将图的矩阵读入，分别为  $A_0, A_1, \dots, A_n$ 。由于矩阵中存在灰点（即灰度值不是 0 也不是 255 的点），而灰点的存在会对矩阵的处理产生一定的影响，因此对矩阵进行二值化处理，利用直方图谷点门限法<sup>[1]</sup>进行处理。

##### 5.1.1.1. 直方图谷点门限法

当图像的灰度直方图分布为双峰形式时，取谷点作为门限值，就能很容易的分割出背景和物体。其表达式为：

$$g(x, y) = \begin{cases} 0, & f(x, y) < TH \\ 1, & f(x, y) \geq TH \end{cases} \quad (1)$$

式中 TH 为谷点门限值。具体计算过程为：计算整个原灰度像素的灰度均值  $m$  和均方差  $s$ ，取两者之差作为阈值（即亮度值）： $\mu = m - s$ 。若图像中某像素的灰度值大于  $\mu$ ，则该像素点为 255（即白点），否则像素点为 0（即黑点）。

### 5.1.2. 碎纸片的匹配度的确定

根据预处理后得到的碎纸片矩阵，其匹配度矩阵为 $cov_{n*n}$ ， $A_i$ 与 $A_j$ 的匹配度由 $A_i$ 右边界的黑色点数 $b_i$ ， $A_j$ 左边界的黑色点数 $b_j$ 以及 $A_i$ 与 $A_j$ 相对应的黑色点数 $b_{ij}$ 确定，公式如下：

$$cov_{ij} = \frac{b_{ij}}{b_i + b_j} \quad (2)$$

## 5.2. 问题一：纵切碎纸片复原

### 5.2.1. 解题思路

建立 19 幅图的匹配度矩阵，对这 19 幅图拼接分配的 0-1 整数规划模型，使得其拼接后图形的匹配度最大，并通过匈牙利算法求解。

### 5.2.2. 模型建立

我们建立 19 幅图拼接分配的 0-1 整数规划模型。

设模型的决策变量 $x_{ij}$ 是 0-1 变量，即：

$$x_{ij} = \begin{cases} 1, & \text{当图 } i-1 \text{ 右侧与图 } j-1 \text{ 左侧相连接时,} \\ 0, & \text{当图 } i-1 \text{ 右侧与图 } j-1 \text{ 左侧不连接时,} \end{cases} \quad (i, j = 1, \dots, 19). \quad (3)$$

$$\max \sum_{i=1}^{19} \sum_{j=1}^{19} cov_{ij} x_{ij} \quad (4)$$

$$\text{s.t.} \begin{cases} \sum_{j=0}^{19} x_{ij} = 1, i = 1, \dots, 19, \\ \sum_{i=0}^{19} x_{ij} = 1, j = 1, \dots, 19, \\ x_{ij} = 0 \text{ 或 } 1 (i, j = 1, \dots, 19). \end{cases} \quad (5)$$

### 5.2.3. 模型求解

我们先通过 Matlab 得到 19 幅图的匹配度矩阵（代码见附录 2），再根据 19 幅图拼接分配的 0-1 整数规划模型，通过匈牙利算法求解。

#### 5.2.3.1. 匈牙利算法

**指派问题的最优性：**如果效益矩阵 $C = (c_{ij})_{n*n}$ 的第  $i$  行、第  $j$  列中的每个元素分别减去一个常数  $a$ 、 $b$  变为矩阵 $D = (d_{ij})_{n*n}$ 的则以新的矩阵  $D$  为效益矩阵和新的目标函数与原效益矩阵  $C$  和原目标函数求得的最优解相同，最优值只差一个常数。

根据指派问题的最优性，将求最优解的问题转化为求效益矩阵的最大 1 元素组的问题。即目标函数换为：

$$\max Z = \sum_{i=1}^n \sum_{j=1}^n c_{ij} x_{ij} \quad (6)$$

此令  $M = \max_{i,j}(c_{ij})$  和  $f_{ij} = M - c_{ij} \geq 0$ ，则效率矩阵变为  $F = (f_{ij})_{n \times n}$ 。于是

考虑目标函数为  $\max Z = \sum_{i=1}^n \sum_{j=1}^n f_{ij} x_{ij}$  的问题。

匈牙利算法步骤：

**Step1:** 对效益矩阵进行变换，使每行每列都出现有 0 元素：

- (1) 从效益矩阵  $C$  中每一行减去该行的最小元素；
- (2) 在所得矩阵中每一列减去该列的最小元素，所得矩阵记为  $D = (d_{ij})_{n \times n}$ ；

**Step2:** 将矩阵  $D$  中 0 元素置为 1 元素，非零元素置为 0 元素，记此矩阵为  $E$ ；

**Step3:** 确定独立 1 元素组：

- (1) 在矩阵  $E$  含有 1 元素的各行中选择 1 元素最少的行，比较该行中各 1 元素所在的列中 1 元素的个数，选择 1 元素的个数最少的一列的那个 1 元素；
- (2) 将所选的 1 元素所在的行和列清 0；
- (3) 重复 **Step2** 和 **Step3**，直到没有 1 元素为止，即得到一个独立 1 元素组；

**Step4:** 判断是否为最大独立 1 元素组：

- (1) 如果所得独立 1 元素组是原效益矩阵的最大独立 1 元素组（即 1 元素的个数等于矩阵的阶数），则已得到最优解，停止计算；
- (2) 如果所得独立 1 元素组还不是原效益矩阵的最大独立 1 元素组，那么利用寻找可扩路的方法对其进行扩张，进行下一步；

**Step5:** 利用寻找可扩路方法确定最大独立 1 元素组：

- (1) 做最少的直线覆盖矩阵  $D$  的所有 0 元素；
- (2) 在没有被直线覆盖的部分找出最小元素，在没有被直线覆盖的各行减去此最小元素，在没被直线覆盖的各列加上此最小元素，得到一个新的矩阵，返回 **Step2**。

通过 **Matlab** 编程利用匈牙利算法<sup>[2]</sup>（代码见附录 2），并根据匹配度矩阵可确定复原图的最左侧和最右侧的图，求解得：

- 1) 附件 1 碎纸片复原序号（从左到右排列），如下表 1

08	14	12	15	03	10	02	16	01	04	15	09	13	18	11	07	17	00	06
----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----

表 1

（附件 1 的复原图片见附录 1）

- 2) 附件 2 碎纸片复原序号（从左到右排列），如下表 2

03	06	02	07	15	18	11	00	05	01	09	13	10	08	12	14	17	16	04
----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----

表 2

（附件 2 的复原图片见附录 1）

附件 1、2 的求解过程均不需要人工干预。

### 5.3. 问题二：纵横切碎纸片复原

#### 5.3.1. 解题思路

通过对碎纸片的分析，建立基于碎纸片图的边界拼接模型，利用相应的预处理、局部匹配、自动字符识别和全局匹配算法求解。

#### 5.3.2. 模型建立

建立基于碎纸片图的边界匹配拼接的模型。模型分为 3 部分：预处理、局部匹配模型、全局匹配模型。

##### 5.3.2.1. 预处理

首先，根据 5.1 模型准备中预处理的方法，对读入碎纸片图形进行二值化的直方图谷点门限法对灰度进行处理。

其次，根据附件 4 中的英文碎纸片拼接图，英文的书写是按照笔画的下端必须紧贴四线格的基准线（即图 1 实线），如图 1：

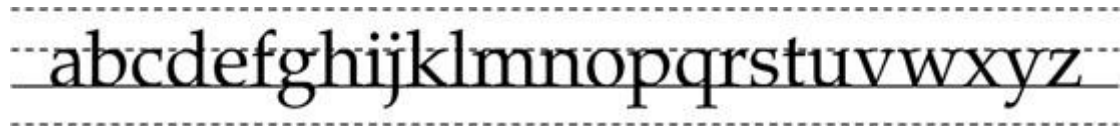


图 1

根据此特性，有利于我们进行局部匹配。所以对每幅碎纸片取英文字符的基准线。

工	1
当	26
如	57
何	97
神	125
仙	165
	180

图 2

对于附件 3 中的中文碎纸片，考虑到中文有很好的行平齐性质，对每幅碎纸片取字符占据的行像素矩阵，并进行两两比较，判断是否平齐。例如在图 2 中，我们可以确定其行像素矩阵为{1,...,26,57,...97,125,...165}。

##### 5.3.2.2. 局部匹配模型

通过建立局部匹配模型，来量化任意两幅碎纸片边界匹配的相关程度。根据 5.1 模型准备中匹配度矩阵生成，得到匹配度  $cov_{ij}$ ：

$$cov_{ij} = \frac{b_{ij}}{b_i + b_j} \quad (7)$$

根据预处理中的取字符基准线，若  $A_i, A_j$  的基准线不在平齐的基准线上，则

$a_{ij} = 0$ ，反之， $a_{ij} = 1$ 。

$$cov_{ij} = \begin{cases} cov_{ij} = 0, & \text{若 } a_{ij} = 0, \\ cov_{ij} \text{ 不变}, & \text{若 } a_{ij} = 1. \end{cases} \quad (8)$$

### 5.3.2.3. 全局匹配模型

根据局部匹配模型的匹配度 $cov_{ij}$ 来确定任意两幅碎纸片拼接的优先级，匹配度越高，两幅碎纸片拼接的优先级越高。

横向匹配：搜索原图左边界的 11 幅碎纸片图，以这 11 幅图为左边界，从左向右进行横向匹配，根据匹配度 $cov_{ij}$ 确定的优先级进行匹配，使得横向生成的整行复原图的匹配度最高。

纵向匹配：由横向匹配的 11 幅碎纸片拼接图，确定纵向的匹配度，进行纵向匹配，得到原图。

### 5.3.3. 模型求解

对该模型利用基于边界匹配的拼接算法来进行求解，并建立自动字符识别来辅助边界匹配，减少人工干预次数，算法流程如下：

#### 5.3.3.1. 预处理

Step1: Matlab 读入附录中的碎纸片图片 $A_1, \dots, A_{209}$ 存为矩阵；

Step2: 用直方图谷点门限法对矩阵进行二值化处理：计算任意矩阵 $A_i$ 的灰度均值 $m$ 和均方差 $s$ ，取两者之差作为阈值： $\mu = m - s$ 。若图像中某像素的灰度值大于 $\mu$ ，则该像素点为 255，否则像素点为 0；

Step3: 取每幅碎纸片的基准线：对任意矩阵 $A_i$ 操作，计算 $A_i$ 每行的黑像素点数 $d_i$ ， $d_{i,i+1}$ 代表 $A_i$ 矩阵中第  $i$  行黑像素点数减去第  $i+1$  行黑像素点数， $d_{i,i+1} = d_i - d_{i+1}$ ，找到 $d_{i,i+1}$ 最大的第  $i$  行，以此为 $A_i$ 中的一条基准线。

#### 5.3.3.2. 局部匹配

Step1: 重复 5.1 模型准备中矩阵 $A_i, A_j$ 匹配度计算：

$$cov_{ij} = \frac{b_{ij}}{b_i + b_j} \quad (9)$$

Step2: 根据预处理中的 Step3，可得任意 $A_i$ 的基准线  $i$ ，通过与 $A_j$ 的基准线  $j$  进行对比，若  $i-j=64*k$  ( $k=0,1,2$ )，则 $a_{ij}=1$ ，反之 $a_{ij}=0$ 。对 $cov_{ij}$ 进行如下处理：

$$cov_{ij} = \begin{cases} cov_{ij} = 0, & \text{如果 } a_{ij} = 0, \\ cov_{ij} \text{ 不变}, & \text{如果 } a_{ij} = 1. \end{cases} \quad (10)$$

其中 64 是通过统计得出附件的英文的碎纸片的行间距。

### 5.3.3.3. 自动字符识别

字符识别是模式的重要应用领域，经过预处理之后，文字识别的一般过程：特征抽取、分类识别、结果识别。

特征抽取：选择稳定、有代表性的特征往往是一个识别系统成功的关键。特征提取方法需要满足的三个条件：

- 1) 提取的特征相互独立，互不相关；
- 2) 特征提取能减小类间距离，增大类间距离；
- 3) 特征向量维数尽可能小。

根据特征生成的方式，将特征分为三类：局部特征、全局特征、结构特征。由于英文字符较简单，我们可以不考虑字符的局部特征或结构特征，而通过全局变换得到全局特征。

分类识别：分类方法，对于英文文字匹配分类，大体上可分为模板匹配，统计决策，句法结构等。在这里我们使用的是模板匹配的方法。

#### 5.3.3.3.1. 模板匹配

模板匹配(Template Match)原理是从待识别的图像或图像区域中提取的若干特征量与模板相应的特征量逐个进行比较，计算它们之间规格化的互相关量，期中互相关量最大的一个就表示其间相似程度最高，可将图像归于相应的类。模板匹配需要事先建立标准模板库，模板库中的模板通常是二值化后的数字模板。一种简单有效的匹配方法是将标准化后的数字图像与字符模板逐个匹配，求出相似度。

本题中使用的部分英文字符模板（附录中包含全部模板）：



#### 5.3.3.3.2. 二值图像间匹配度的算法

为了使字符识别的适应范围更加广泛，考虑到实际情况中可能会出现噪声、图像的缺损、污迹等，故采用通过计算二值图像间的Hausdorff距离，来表示图像间的匹配程度。

**Hausdorff 距离：**Hausdorff是一种描述两组点集之间相似程度的一种度量，常用于检测二值图像间的匹配程度。

假设有两组点集合  $A = \{a_1, \dots, a_p\}$ ,  $B = \{b_1, \dots, b_q\}$ ，则这两个点集之间的Hausdorff距离定义为： $H(A, B) = \max(h(A, B), h(B, A))$ 。  
其中， $h(A, B)$ ,  $h(B, A)$ 为单向的Hausdorff距离， $h(A, B)$ 表示点集A中所有点到点集B的最小距离的最大值

$$\begin{cases} h(A, B) = \max(a \in A) \min(b \in B) ||a - b|| \\ h(B, A) = \max(b \in B) \min(a \in A) ||b - a|| \end{cases} \quad (11)$$

$||$ 是点集A和点集B间的距离范数，常用的是Euclidean距离。如上面所示， $H(A, B)$ 是Hausdorff距离的最基本形式，称为双向Hausdorff距离， $h(A, B)$ 是从集合A到集合B的单向Hausdorff距离，同理， $h(B, A)$ 为从集合B到集合A的单向Hausdorff距离。 $h(A, B)$ 表示对于集合A中的每一个点 $a_i$ ，求出它到集合B的最小



距离，并进行排序，所取出最大值，同理， $h(B,A)$ 表示对于集合B中的每一个点 $b_j$ ，求出它到集合A的最小距离，并从 $j$ 个最小距离中所取出的最大值。单向Hausdorff距离 $h(A,B)$ 和 $h(B,A)$ 中的较大者即为 $H(A,B)$ 。

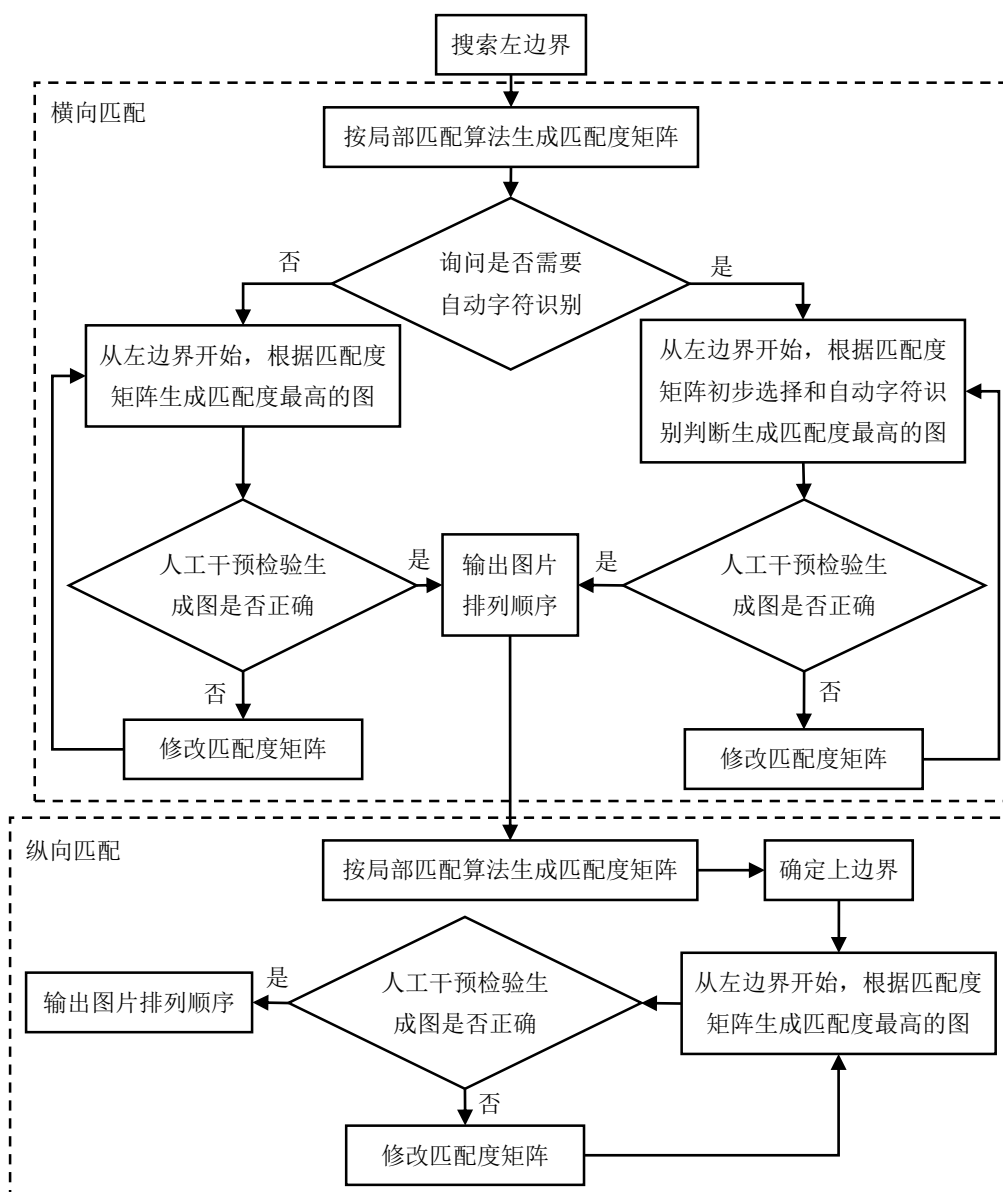
测试：

$$\begin{array}{lcl} \text{E} + \text{I} \longrightarrow \text{E} & \min(\text{Hausdorff} - \text{distance}) = 0 \\ \text{E} + \text{V} \longrightarrow \text{EV} & \min(\text{Hausdorff} - \text{distance}) = 180 \end{array}$$

$\min(\text{Hausdorff} - \text{distance})$ 为拼接图像与各模板Hausdorff距离的最小值。Hausdorff距离越小，说明图像越匹配。

我们将自动字符识别加入到全局匹配流程中，让其辅助匹配度进行边界匹配，以减少人工干预的次数。

#### 5.3.3.4. 全局匹配



**Step1:** 搜索原图左边界的 11 幅碎纸片图：判断所有矩阵 $A_{i1k}$ 是否都为 255，筛选出左边界为白色的矩阵，再通过计算 $A_i$ 中黑点纵坐标到矩阵第一列的最小距离 $d_i$ ，选出前 11 位 $d_i$ 最大的矩阵 $A_i$ 为左边界的 11 幅碎纸片图；

**Step2:** 横向匹配见流程图，其中修改匹配度矩阵是通过程序将该图匹配度最高的碎纸片图匹配值改为 0；（不需要对中文碎纸片进行模板匹配）

**Step3:** 纵向匹配见流程图，修改匹配度矩阵如 Step2。

**Step4:** 复原出原图。

### 5.3.3.5. 附件 3 求解

附件 3 的碎纸片图为中文碎纸片图，故不需要进行全局匹配的自动字符识别的操作。

通过 Matlab 编程实现基于边界匹配的拼接算法（代码见附录 3）。其中人工干预的方式是当横向匹配生成拼接的 19 幅图并发现拼接有错误后，输入拼接错误图的位置。程序会将该拼接错误图的匹配度改为 0，重新生成新图。

人工干预示例如下图 3、4：

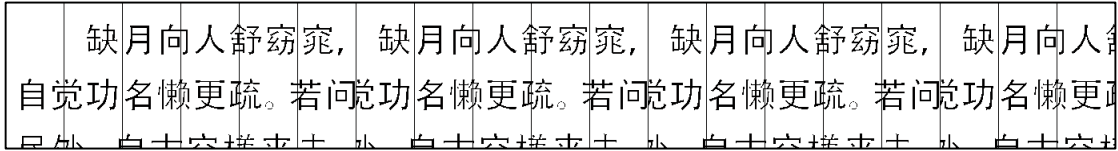


图 3

对于图 2，我们可以明显地分辨出，第 6 幅图之后出现错误（其中每一根黑线代表一幅图的右边界，便于区分）。所以在程序中输入 6，图像便会变成下图 4：

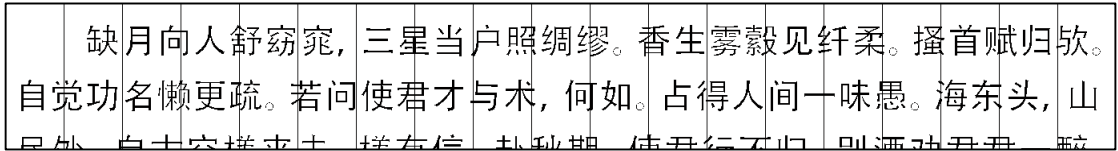
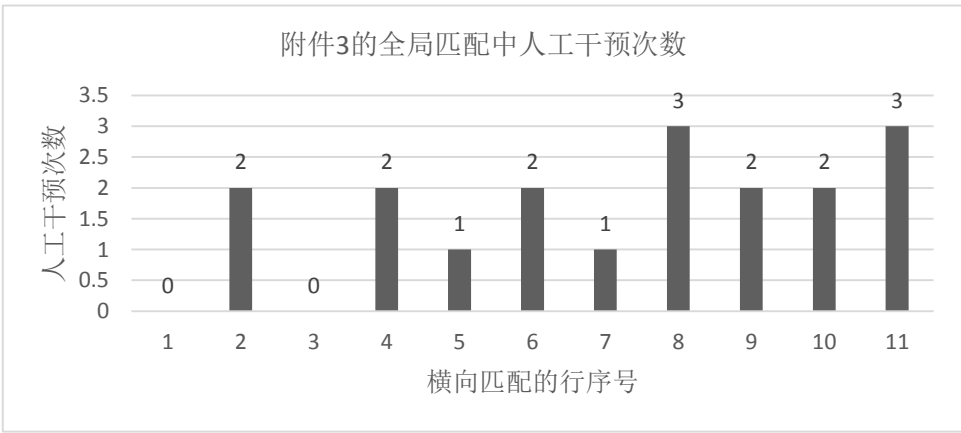


图 4

由此可见，通过人工干预，将该横向图调整正确。在附件 3 的求解过程中，一共人工干预次数为 18 次，下表 3 表示横向匹配过程中的每行人工干预次数（人工干预一次指将错误位置的图调整至正确）。



附件 3 碎纸片复原序号，如下表 3：

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

表 3

（附件 3 的复原图片见附录 1）

5.3.3.6. 附件 4 求解

附件 4 的碎纸片图为英语碎纸片图，通过 Matlab 编程实现基于边界匹配的拼接算法（代码见附录 3），其中人工干预示例如下图 5、6：

One	hand	washes	the	other.	Talk	of	th	he	is	torashes	the
bound	to	appear.	Tuesday's	child	is	full	c	can't	Ror.	Tuesda	
judge	a	book	by	its	cover.	Now	drips	the	ecome	for	its
cover.											

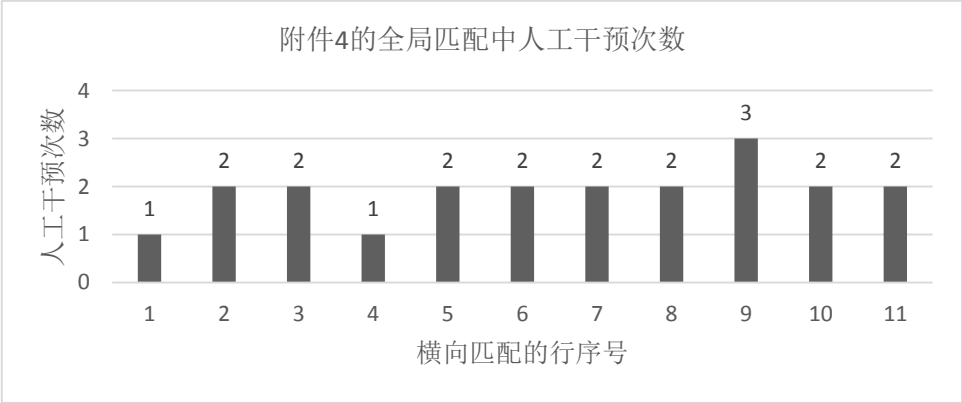
图 5

对于图 4，我们可以明显地分辨出，第 13 幅图之后出现错误（其中每一根黑线代表一幅图的右边界）。所以在程序中输入 13，图像便会变成下图 6：

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

图 6:

由此可见，通过人工干预，将该横向图调整正确。在附件 3 的求解过程中，一共人工干预次数为 21 次，下表 3 表示横向匹配过程中的每行人工干预次数（人工干预一次指将错误位置的图调整至正确）。



附件 4 碎纸片复原序号，如下表 4：

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

表 4

（附件 4 的复原图片见附录 1）

#### 5.4. 问题三：双面横纵切碎纸片复原

##### 5.4.1. 解题思路

根据附件 5 碎纸片图形分析，正反面图形的基准线是平齐，故找出正反面的左边界碎纸片图形，通过建立 0-1 整数规划模型确定左边界碎纸片图形正反面的匹配，再通过问题二的求解方法求解。

##### 5.4.2. 模型建立

通过全局分配 Step1 得到正反面各 11 个左边界碎纸片 $A_i(i = 1 \dots 22)$ ，预处理 Step3 得到 $A_i$ 的基准线 $d_i$ ，确定任意矩阵 $A_i, A_j$ 的基准线匹配程度 $c_{ij}$ ：

$$c_{ij} = \begin{cases} \min_{i,j} |(d_i - d_j) - 64 * k|, (k = 0,1,2) \\ 32, \text{if}(i = j) \end{cases} \quad (i, j = 1, \dots, 22) \quad (12)$$

通过基准线匹配程度最小来对 22 个左边界碎纸片进行正反面分配，建立 0-1 整数规划模型，得

设模型的决策变量 $x_{ij}$ 是 0-1 变量，即

$$x_{ij} = \begin{cases} 1, \text{当左边界图 } i \text{ 与左边界图 } j \text{ 正反面配对} \\ 0, \text{当左边界图 } i \text{ 与左边界图 } j \text{ 不能正反面配对} \end{cases} \quad (i, j = 1, \dots, 22). \quad (13)$$

$$\min \sum_{i=1}^{22} \sum_{j=1}^{22} c_{ij} x_{ij} \quad (14)$$

$$\text{s. t. } \begin{cases} \sum_{j=0}^{22} x_{ij} = 1 \quad (i = 1, \dots, 22) \\ \sum_{i=0}^{22} x_{ij} = 1 \quad (j = 1, \dots, 22) \\ x_{ij} = x_{ji} \quad (i, j = 1, \dots, 22) \\ x_{ij} = 0 \text{ or } 1 \quad (i, j = 1, \dots, 19) \end{cases} \quad (15)$$

5. 4. 3. 模型求解

5. 4. 3. 1. 算法流程

- Step1: 5.3.3.1.中预处理;
- Step2: 用 Lingo 求解基准线匹配程度 0-1 整数规划模型;
- Step3: 5.3.3.2.中局部匹配处理;
- Step4: 5.3.3.4.中全局匹配处理, 得到复原图。

5. 4. 3. 2. 附件 5 求解

根据 0-1 整数规划模型, 通过 Lingo 编程求解 (代码见附录 4), 求解得, 表格同一列表示配对, 表 5:

172b	146a	143a	078b	009a	099a	105b	005b	083b	114a	090b
165b	035b	186b	136b	023b	054a	003b	089a	199b	013b	088b

表 5: 左边界匹配序号

再通过 5.2 中问题二的基于边界匹配的算法, 通过 Matlab 求解 (代码见附录 4)。

人工干预示例如下图 7、8:

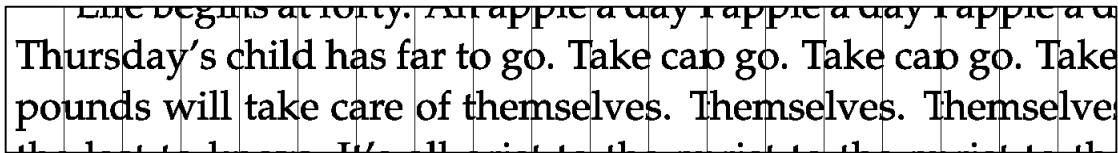


图 7

对于图 6, 我们可以明显地分辨出, 第 12 幅图之后出现错误 (其中每一根黑线代表一幅图的右边界, 便于区分)。所以在程序中输入 12, 图像便会变成下图 8:

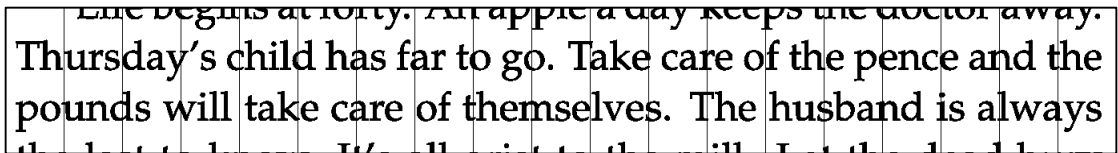
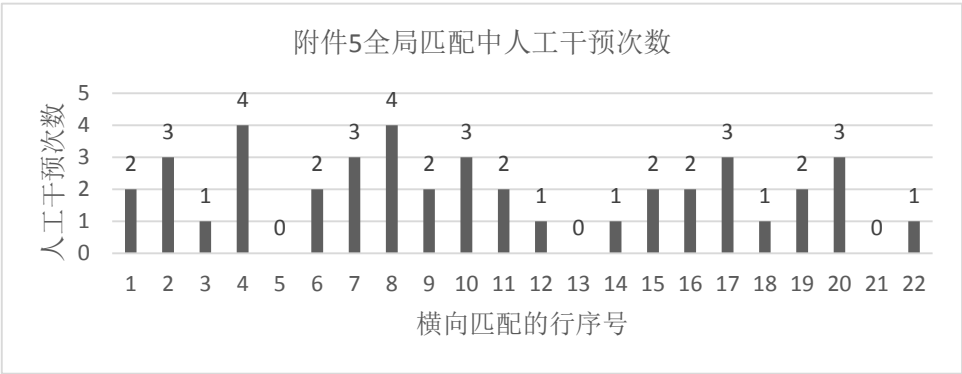


图 8

由此可见, 通过人工干预, 将该横向图调整正确。在附件 5 的求解过程中, 一共人工干预次数为 42 次, 下表 4 表示横向匹配过程中的每行人工干预次数 (人工干预一次指将错误位置的图调整正确为止)。



得到复原图序号如下表 6、7：

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 6

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

表 7

（附件 5 的复原图片见附录 1）

## 六、 模型评价及改进

### 6.1. 模型评价

- 1) 对于问题一，我们建立了 0-1 整数规划模型分配 19 幅碎纸片的拼接，使得图形拼接的匹配度最大，再通过匈牙利算法求解，求解效率高并且拼接的正确性很高，故此模型及算法适用于解决纵切的 19 幅的拼接。
- 2) 对于问题二，我们通过基于边界匹配的拼接算法，对碎纸片矩阵进行预处理、局部匹配、全局匹配。其中局部匹配的匹配度是通过边缘点计算，但如果在碎纸片大小较小的情况下，区分度不大，需要改进。全局匹配中我们采用了自动字符识别，此方法能够辅助进行边界匹配，这是本文的一个亮点，但是该算法的效率不够高，需要改进其算法。
- 3) 对于问题三，我们根据基准线平行特点，建立 0-1 整数规划模型分配同行的左边界碎纸片，再利用问题二的处理方法，复原其中一面原图，根据分配同行的左边界碎纸片，复原原图。此方法对于处理双面的横纵切碎纸片，效率较高并且正确率也相对较高。

## 6.2. 模型改进

- 1) 对于第二题既纵切又横切的情形，中文文档的拼接复原所依赖的匹配度矩阵是根据文档碎片边缘点的匹配建立而成。这个方法简单高效，但是在碎片大小比较小的情况下，碎片间的区分度并不大。为此，我们进行了进一步研究，利用模式识别中的汉字识别来判断碎片的连接是否匹配。即判断处于连接处的图形是否能被识别为汉字，若能，则说明连接是有效的。具体算法是先建立一个标准汉字的字库，利用统计原理，得到一个匹配阈值，匹配度大于阈值就认定为匹配。然后利用统计与结构模式识别结合方法，判断样本与字库中汉字的匹配度。如果这些匹配度中最大值大于阈值，则认定样本是一个汉字。

虽然得到了匹配器，但是并不能保证匹配得一定正确。这时候可以加入人工干预，同时将干预结果反馈给匹配矩阵，增加匹配矩阵的准确度。经过，数轮迭代后，匹配的成功率能得到显著提高，也能有效降低人工干预的次数。

- 2) 同理，对于英文的识别，我们已经尝试加入了基于模板匹配的字符识别，并应用了Hausdorff算法判断样本与字库的匹配度。由于Hausdorff算法给出的只是图像的离散度，对于中心趋势并不能很好表征。因此深入下去，我们打算优化Hausdorff算法，采用广义Hausdorff算法，先求点集之间的局部最优解，选取优化结果中的距离最大值作为Hausdorff距离。

另外，如前所述，Hausdorff算法虽然准确但是效率并不高。考虑到本题的局限性，可以将图像的噪点，旋转扭曲，污渍等等不计算在内。我们可以以降低算法的准确性换取高效性，采用模式识别的其他算法，如PCA模板匹配法，基于类中心的欧式距离匹配法，马氏距离分类法等等。

## 七、参考文献

- [1] 何耕嫻. 印刷体文档图像的中文字符识别. 硕士论文. 2011. P16
- [2] 韩中庚. 数学建模方法及其应用. 高等教育出版社. 2005. P188-190
- [3] 杨启帆,谈之奕,何勇. 数学建模. 浙江大学出版社. 2006.
- [4] 杨淑莹. 模式识别与智能计算-matlab 技术实现. 电子工业出版社. 2007. P17-37, P47-60.
- [5] 贺强. 字符识别的相关方法研究. 硕士论文. 2010.
- [6] Daniel P. Huttenlocher, Gregory A. Klanderman, and William J. Rucklidge. Comparing Images Using the Hausdorff Distance. IEEE TRANSACTIONS ON PAmRN ANALYSIS AND MACHINE INTELLIGENCE, VOL. 15, NO. 9, SEPTEMBER 1993.

## 附录 1

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。村里村北响犍车。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。双鬓绿坠。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，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。临水纵横回晚鞚。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。何物与侬归去、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，飐飐。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 1 结果



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-

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。主人瞋小。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

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

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.

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

## 附录 2

% 问题一、二的解决需要 Matlab, 问题三的解决需要 Matlab 和 Lingo  
% 问题一的代码在附录 2, 问题二的代码在附录 3, 问题 3 的代码在附录 4

% 该附录为问题一的Matlab代码

% 程序运行完, 会在该程序的同一目录下生成bmp格式的结果图

```
function [] = question1()
```

% 问题一的主函数

```
graph = zeros(1980,1368);
```

```
A = zeros(1980,72,19);
```

```
for i = 0:1
```

```
    for j = 0:9
```

```
        if(i * 10 + j + 1 <= 19)
```

```
            A(:, :, i * 10 + j + 1) =
```

```
imread(['0',num2str(i),num2str(j)], 'bmp');
```

```
        end
```

```
    end
```

```
end
```

% 读取图像

```
matr = genermatr(A);
```

```
[~,ans]=fenpei(matr);
```

```
temp = 0;
```

```
temprow = 0;
```

```
tempcol = 0;
```

```
col = 0;
```

```
row = 0;
```

```
for i = 1:19
```

```
    for j = 1:19
```

```
        if(matr(i,j) ~= 0)
```

```
            temp = 1;
```

```
            break;
```

```
        end
```

```
    end
```

```
    if(temp == 0)
```

```
        temprow = i;
```

```
        break;
```

```
    else
```

```
        temp = 0;
```

```
    end
```

```
end
```

% i是最右的纵切图

```
for j = 1:19
```

```
    for i = 1:19
```

```
        if(matr(i,j) ~= 0)
```

```

        temp = 1;
        break;
    end
end
if(temp == 0)
    tempcol = j;
    break;
else
    temp = 0;
end
end
% j是最左的纵切图
col = tempcol;
graph = A(:,:,col);
while(row ~= temprow)
    for i = 1:19
        if(ans(col,i) == 1)
            row = i;
            break;
        end
    end
    graph = [graph,A(:,:,row)];
    col = row;
end
imwrite(graph, 'graph.bmp');
end

% 生成匹配度矩阵
% 输入19幅纵切图的图形矩阵
% 输出相应的匹配度矩阵
% 匹配度矩阵的第i行第j列表示：第i幅图在左，第j幅图在右的匹配度
function [matr] = genermatr(A)

matr = zeros(19,19);
temp = 0;

for i = 1:19
    for j = 1:1980
        k = 1;
        if(A(j,k,i) ~= 0 && A(j,k,i) ~= 255)
            A(j,k,i) = 0;
        end
        k = 72;
        if(A(j,k,i) ~= 0 && A(j,k,i) ~= 255)

```

```

        A(j,k,i) = 0;
    end
end
end
% 将灰点改成黑点，增加匹配度

for i = 1:19
    for j = 1:19
        if(i == j)
            matr(i,j) = 0;
        else
            for k = 1:1980
                if(A(k,72,i) == 0 && A(k,1,j) == 0)
                    temp = temp + 1;
                end
            end
            matr(i,j) = temp / 1980;
        end
        temp = 0;
    end
end
% 生成匹配度矩阵

end

% 输入效率矩阵，marix为方阵
% 若效率矩阵中有M，则用一充分大的数代替
% 输出z为最优解，ans为最优分配矩阵
function [z,ans]=fenpei(marix)

a=marix;
% 确定矩阵维数
s=length(a);
x=max(max(a));
c=zeros(s,s);
c(:,:)=x;
b=c-a;
a=b;
% 确定矩阵行最小值，进行行减
ml=min(a');
for i=1:s
    a(i,:)=a(i,:)-ml(i);
end
% 确定矩阵列最小值，进行列减

```



```

mr=min(a);
for j=1:s
    a(:,j)=a(:,j)-mr(j);
end

% start working
num=0;
while (num~=s)
    index=ones(s);
    index=a&index;
    index=~index;
    flag = zeros(s);
    ans = zeros(s);
    while (sum(sum(index)))
        for i=1:s
            t=0;
            l=0;
            for j=1:s
                if(flag(i,j)==0&&index(i,j)==1)
                    l=l+1;
                    t=j;
                end
            end
            if(l==1)
                flag(:,t)=flag(:,t)+1;
                index(:,t)=0;
                ans(i,t)=1;
            end
        end
        for j=1:s
            t=0;
            r=0;
            for i=1:s
                if(flag(i,j)==0&&index(i,j)==1)
                    r=r+1;
                    t=i;
                end
            end
            if(r==1)
                flag(t,:)=flag(t,:)+1;
                index(t,:)=0;
                ans(t,j)=1;
            end
        end
    end
end

```

```

end
num=sum(sum(ans));
if(s==num)
    break;
end
m=max(max(a));
for i=1:s
    for j=1:s
        if(flag(i,j)==0)
            if(a(i,j)<m)
                m=a(i,j);
            end
        end
    end
end
for i=1:s
    for j=1:s
        if(flag(i,j)==0)
            a(i,j)=a(i,j)-m;
        end
        if(flag(i,j)==2)
            a(i,j)=a(i,j)+m;
        end
    end
end
end
zm=ans.*b;
z=0;
z=sum(sum(zm));
end

```

## 附录 3

% 问题一、二的解决需要 Matlab，问题三的解决需要 Matlab 和 Lingo  
% 问题一的代码在附录 2，问题二的代码在附录 3，问题 3 的代码在附录 4

% 该附录为问题二附录三的Matlab代码

% 程序运行完，会在该程序的同一目录下生成bmp格式的结果图

```
function [] = question2chinese()
```

% 该程序为问题二附录三的主函数

```
A = zeros(180,72,209);
```

```
for i = 0:2
```

```
    for j = 0:9
```

```
        for k = 0:9
```

```
            if(i * 100 + j * 10 + k <= 208)
```

```
                A(:,:,i * 100 + j * 10 + k + 1) =
```

```
imread([num2str(i),num2str(j),num2str(k)], 'bmp');
```

```
            end
```

```
        end
```

```
    end
```

```
end
```

% 读取图像

```
ansgraph = total(A);
```

```
genergraph(A,ansgraph);
```

```
finansgraph = total2();
```

```
finalgraph = [];
```

```
for i = 1:11
```

```
    graph = A(:,:,finansgraph(i,1));
```

```
    for j = 2:19
```

```
        graph = [graph,A(:,:,finansgraph(i,j))];
```

```
    end
```

```
    finalgraph = [finalgraph;graph];
```

```
end
```

```
imwrite(finalgraph, 'graph.bmp');
```

```
end
```

% 纵向匹配矩阵

% 提供接口供人工匹配

% 输出纵向匹配矩阵

```
function [finansgraph] = total2()
```

```
A1 = zeros(180,1368,11);
```

```
for i = 1:11
```

```
    A1(:,:,i) = imread(['graph',num2str(i)], 'bmp');
```

```

end
col = genermatr2(A1);
finansgraph = zeros(11,1);
gr = zeros(1,1368);

inputnum = 1;
while(inputnum ~= 0)
    tempj = 1;
    col2 = 5;
    finansgraph(tempj) = col2;
    while(tempj < 11 && ~isempty(find(col(col2,:) ~= 0)))
        p1 = find(col(col2,:) == max(col(col2,:)));
        tempj = tempj + 1;
        finansgraph(tempj) = p1;
        col2 = p1;
    end
    graph = A1(:, :, 5);
    for l = 2:11
        graph = [graph; gr; A1(:, :, finansgraph(l))];
    end
    imshow(graph);
    inputnum = input('Which picture is wrong? If none, please enter
0. ');
    if(inputnum ~= 0)
        col(finansgraph(inputnum), finansgraph(inputnum+1)) = 0;
    end
end

end

function [] = genergraph(A,ansgraph);

for i = 1:11
    graph = A(:, :, ansgraph(i,1));
    for j = 2:19
        graph = [graph, A(:, :, ansgraph(i,j))];
    end
    imwrite(graph, ['graph', num2str(i), '.bmp']);
end

end

% 横向匹配矩阵
% 提供接口供人工匹配

```

```

% 输入图形矩阵
% 输出横向匹配矩阵
function [ansgraph] = total(A)

row = genermatr(A);
ansgraph = zeros(11,19);
[w,o]=findleft(A);
w = w';
o = o';
gr = zeros(180,1);
know = input('Where do you want to start?');
for i = know:11
    inputnum = 1;
    while(inputnum ~= 0)
        tempj = 1;
        row2 = w(i);
        ansgraph(i,tempj) = row2;
        while(tempj < 19 && ~isempty(find(row(row2,:) ~= 0)))
            p1 = find(row(row2,:) == max(max(row(row2,:)))));
            tempj = tempj + 1;
            ansgraph(i,tempj) = p1;
            row2 = p1;
        end
        graph = A(:, :, w(i));
        for l = 2:tempj
            graph = [graph,gr,A(:, :, ansgraph(i,l))];
        end
        imshow(graph);
        inputnum = input('Which picture is wrong? If none, please
enter 0. ');
        if(inputnum ~= 0)
            row(ansgraph(i,inputnum),ansgraph(i,inputnum+1)) = 0;
        end
    end
end
for d = 1:19
    for t = 1:209
        row(t,ansgraph(i,d)) = 0;
    end
end
end
end

% 输入所有图的图形矩阵
% 输出匹配度矩阵

```

```

function [row] = genermatr(A)

row = zeros(209,209);
temprow = 0;
t1=0;
for i = 1:209
    for j = 1:180
        for k = 1:72
            if(A(j,k,i) < 255)
                A(j,k,i) = 0;
            end
        end
    end
end
end
% 将灰点改成黑点
for i = 1:209
    for j = 1:209
        if(i == j)
            row(i,j) = 0;
        else
            for k = 1:180
                if(A(k,72,i) == 0 && A(k,1,j) == 0)
                    temprow = temprow + 1;
                end
                if(A(k,72,i) == 0)
                    t1=t1+1;
                end
                if(A(k,1,j) == 0)
                    t1=t1+1;
                end
            end
            if(t1==0)
                row(i,j)=0;
            else
                row(i,j) = 2 * temprow / t1;
            end
            t1 = 0;
        end
    end
    temprow = 0;
end
end
for i = 1:209
    for j = 1:209
        row(i,j) = row(i,j) + rand()*0.05;
    end
end

```

```

        end
    end
    for i = 1:209
        for j = (i+1):209
            [r1,~] = find(A(:, :, i) == 0);
            [r2,~] = find(A(:, :, j) == 0);
            r1 = sort(r1);
            r2 = sort(r2);
            r1 = r1';
            r2 = r2';
            r1 = unique(r1);
            r2 = unique(r2);
            if (length(intersect(r1,r2))/min(length(r1),length(r2))) < 0.9)
                row(i,j) = 0;
                row(j,i) = 0;
            end
        end
    end
end
% 判断两幅图是否可能属于同一行
end

```

% 对于已经横向匹配结束的11幅图，进行纵向匹配度矩阵生成  
 % 输入11幅图的图形矩阵  
 % 输出匹配度矩阵

```
function [col] = genermatr2(A1)
```

```

col = zeros(11,11);
tempcol = 0;
temp = 0;
for i = 1:11
    for j = 1:1368
        if (A1(1,j,i) ~= 255)
            A1(1,j,i) = 0;
        end
        if (A1(180,j,i) ~= 255)
            A1(180,j,i) = 0;
        end
    end
end
for i = 1:11
    for j = 1:11
        if(i == j)
            col(i,j) = 0;
        else

```

```

        for h = 1:1368
            if(A1(180,h,i) == 0 && A1(1,h,j) == 0)
                tempcol = tempcol + 1;
            end
            if(A1(180,h,i) == 0)
                temp = temp + 1;
            end
            if(A1(1,h,j) == 0)
                temp = temp + 1;
            end
        end
        if(temp == 0)
            col(i,j) = 1;    %两边都无黑点，匹配度为1
        else
            col(i,j) = 2 * tempcol / temp;
        end
    end
    tempcol = 0;
    temp = 0;
end
end
for i = 1:11
    for j = 1:11
        if(i ~= j)
            col(i,j) = col(i,j) + rand()*0.01;
        end
    end
end
end
% 生成匹配度矩阵
end

```

```

% 输入图形矩阵
% 输出排在最左边的图
function [w,o]=findleft(A)
j=1;
x=[];
for i=1:209
    if(A(:,1,i)==255)
        x(j)=i;
        j=j+1;
    end
end
s=length(x);

```



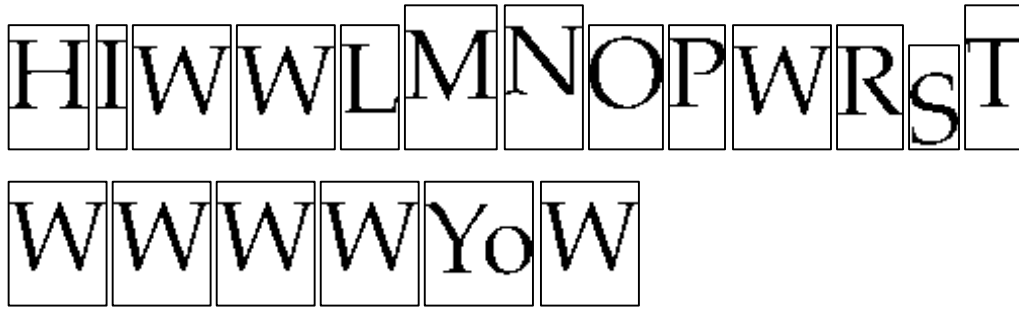
```

q=[];
p=1;
for i=1:16
    max=72;
    for j=1:180
        for z=1:72
            if (A(j,z,x(i))==0)
                if (z<max)
                    max=z;
                end
            end
        end
    end
    q(p)=max;
    p=p+1;
end
e=zeros(11,1);
h=zeros(5,1)
e=find(q>10);
h=find(q<10);
w=zeros(11,1);
for i=1:11
    w(i)=x(1,e(1,i));
end
o=zeros(5,1);
for i=1:5
    o(i)=x(1,h(1,i));
end
end
end

```

---

abcdefghiWklmnopWr  
stuvwWyWABCDEFG



```

% 模板库
% 用来进行自动字符识别
% 由于附件中许多英文字符并没出现，所以用w代替，结果不受影响
% 问题一、二的解决需要 Matlab，问题三的解决需要 Matlab 和 Lingo
% 问题一的代码在附录 2，问题二的代码在附录 3，问题 3 的代码在附录 4
% 该附录为问题二附录四的Matlab代码
% 程序运行完，会在该程序的同一目录下生成bmp格式的结果图
function [] = question2english()
% 该程序为问题二附录四的主函数
A = zeros(180,72,209);
for i = 0:2
    for j = 0:9
        for k = 0:9
            if(i * 100 + j * 10 + k <= 208)
                A(:,:,i * 100 + j * 10 + k + 1) =
imread([num2str(i),num2str(j),num2str(k)], 'bmp');
            end
        end
    end
end
end
% 读取图像
ansgraph = total();
genergraph(A,ansgraph);
col = genermatr2();
finansgraph = total2(col);
finalgraph = [];
for i = 1:11
    graph = A(:,:,finansgraph(i,1));
    for j = 2:19
        graph = [graph,A(:,:,finansgraph(i,j))];
    end
    finalgraph = [finalgraph;graph];
end
imwrite(finalgraph, 'graph.bmp');
end

```

```

% 横向匹配矩阵
% 提供接口供人工匹配
% 输出横向匹配矩阵
% 输入匹配值矩阵
function [ansgraph] = total()

A = zeros(180,72,209);
for i = 0:2
    for j = 0:9
        for k = 0:9
            if(i * 100 + j * 10 + k <= 208)
                A(:,:,i * 100 + j * 10 + k + 1) =
imread([num2str(i),num2str(j),num2str(k)], 'bmp');
            end
        end
    end
end
end
% 读取图像
row = genermatr();
ansgraph = zeros(11,19);
[w,o]=findleft(A);
w = w';
know = 0;
know = input('Do you need to use OCR function?(1/0)');
if(know == 0)
    gr = zeros(180,1);
    tit = input('Where do you want to start?');
    for i = tit:11
        inputnum = 1;
        while(inputnum ~= 0)
            tempj = 1;
            row2 = w(i);
            ansgraph(i,tempj) = row2;
            while(tempj < 19 && ~isempty(find(row(row2,:) ~= 0)))
                p1 = find(row(row2,:) == max(row(row2,:)));
                tempj = tempj + 1;
                ansgraph(i,tempj) = p1;
                row2 = p1;
            end
            graph = A(:,:,w(i));
            for l = 2:tempj
                graph = [graph,gr,A(:,:,ansgraph(i,l))];
            end
            imshow(graph);
        end
    end
end

```

```

        inputnum = input('Which picture is wrong? If none, please
enter 0.');
```

```

        if(inputnum ~= 0)
            if(inputnum == 1000)
                inputnum = input('Which picture is wrong? If none,
please enter 0.');
```

```

                posit = input('Which picture do you want to show?');
                row(ansgraph(i,inputnum),posit) = 10;
            else
                row(ansgraph(i,inputnum),ansgraph(i,inputnum+1)) =
0;

                end
            end
        end
    for d = 1:19
        for t = 1:209
            row(t,ansgraph(i,d)) = 0;
            row(ansgraph(i,d),t) = 0;
        end
    end
    disp((ansgraph(i,:)));
end
else
    gr = zeros(180,1);
    tit = input('Where do you want to start?');
    for i = tit:11
        inputnum = 1;
        while(inputnum ~= 0)
            tempj = 1;
            row2 = w(i);
            ansgraph(i,tempj) = row2;
            while(tempj < 19 && ~isempty(find(row(row2,:) ~= 0)))
                p1 = find(row(row2,:) == max(row(row2,:)));
                if(hd (A(:, :, row2), A(:, :, p1)) == 0)
                    row(row2,p1) = 0;
                    continue;
                end
                tempj = tempj + 1;
                ansgraph(i,tempj) = p1;
                row2 = p1;
            end
            graph = A(:, :, w(i));
            for l = 2:tempj
                graph = [graph,gr,A(:, :, ansgraph(i,l))];
            end
        end
    end
end

```

```

        end
        imshow(graph);
        inputnum = input('Which picture is wrong? If none, please
enter 0. ');
        if(inputnum ~= 0)
            if(inputnum == 1000)
                inputnum = input('Which picture is wrong? If none,
please enter 0. ');
                posit = input('Which picture do you want to show?');
                row(ansgraph(i,inputnum),posit) = 10;
            else
                row(ansgraph(i,inputnum),ansgraph(i,inputnum+1)) =
0;
            end
        end
    end
    for d = 1:19
        for t = 1:209
            row(t,ansgraph(i,d)) = 0;
            row(ansgraph(i,d),t) = 0;
        end
    end
    disp((ansgraph(i,:)));
end
end

% 纵向匹配矩阵
% 提供接口供人工匹配
% 输出纵向匹配矩阵
% 输入纵向匹配值矩阵
function [finansgraph] = total2(col)

A1 = zeros(180,1368,11);
for i = 1:11
    A1(:,:,i) = imread(['graph',num2str(i)],'bmp');
end

finansgraph = zeros(11,1);
gr = zeros(1,1368);

inputnum = 1;
while(inputnum ~= 0)
    tempj = 1;

```

```

col2 = 5;
finansgraph(tempj) = col2;
while(tempj < 11 && ~isempty(find(col(col2,:) ~= 0)))
    p1 = find(col(col2,:) == max(col(col2,:)));
    tempj = tempj + 1;
    finansgraph(tempj) = p1;
    col2 = p1;
end
graph = A1(:, :, 5);
for l = 2:11
    graph = [graph; gr; A1(:, :, finansgraph(l))];
end
imshow(graph);
inputnum = input('Which picture is wrong? If none, please enter
0. ');
if(inputnum ~= 0)
    col(finansgraph(inputnum), finansgraph(inputnum+1)) = 0;
end
end

end

% 生成纵向匹配值矩阵
function [col] = genermatr2()

col = zeros(11,11);
tempcol = 0;
temp = 0;
A = zeros(180,1368,11);
for i = 1:11
    A(:, :, i) = imread(['graph', num2str(i)], 'bmp');
end
for i = 1:11
    for j = 1:1368
        if (A(1,j,i) ~= 255)
            A(1,j,i) = 0;
        end
        if (A(180,j,i) ~= 255)
            A(180,j,i) = 0;
        end
    end
end
end
for i = 1:11
    for j = 1:11

```

```

        if(i == j)
            col(i,j) = 0;
        else
            for h = 1:1368
                if(A(180,h,i) == 0 && A(1,h,j) == 0)
                    tempcol = tempcol + 1;
                end
                if(A(180,h,i) == 0)
                    temp = temp + 1;
                end
                if(A(1,h,j) == 0)
                    temp = temp + 1;
                end
            end
            if(temp == 0)
                col(i,j) = 0;
            else
                col(i,j) = 2 * tempcol / temp;
            end
        end
        tempcol = 0;
        temp = 0;
    end
end
for i = 1:11
    for j = 1:11
        if(i ~= j)
            col(i,j) = col(i,j) + rand()*0.01;
        end
    end
end
end

function [] = genergraph(A,ansgraph)

for i = 1:11
    graph = A(:, :, ansgraph(i,1));
    for j = 2:19
        graph = [graph, A(:, :, ansgraph(i,j))];
    end
    imwrite(graph, ['graph', num2str(i), '.bmp']);
end

end

```

```

function [row] = genermatr()

A = zeros(180,72,209);
row = zeros(209,209);
temprow = 0;
t1=0;
for i = 0:2
    for j = 0:9
        for k = 0:9
            if(i * 100 + j * 10 + k <= 208)
                A(:, :, i * 100 + j * 10 + k + 1) =
imread([num2str(i), num2str(j), num2str(k)], 'bmp');
            end
        end
    end
end
end
% 读取图像
for i = 1:209
    for j = 1:180
        for k = 1:72
            if(A(j,k,i) < 255)
                A(j,k,i) = 0;
            end
        end
    end
end
end
% 将灰点改成黑点
for i = 1:209
    for j = 1:209
        if(i == j)
            row(i,j) = 0;
        else
            for k = 1:180
                if(A(k,72,i) == 0 && A(k,1,j) == 0)
                    temprow = temprow + 1;
                end
                if(A(k,72,i) == 0)
                    t1=t1+1;
                end
                if(A(k,1,j) == 0)
                    t1=t1+1;
                end
            end
        end
    end
end

```



```

        if(t1==0)
            row(i,j)=1;
        else
            row(i,j) = 2 * temprow / t1;
        end
        t1 = 0;
    end
    temprow = 0;
end
end
for i = 1:209
    for j = 1:209
        if(i ~= j)
            row(i,j) = row(i,j) + rand()*0.05;
        end
    end
end
end

[Q]=getblack();
for i = 1:209
    for j = (i+1):209
        if(mod((Q(i) - Q(j)),64) > 2 && 64 - mod((Q(i) - Q(j)),64) >
2)
            row(i,j) = 0;
            row(j,i) = 0;
        end
    end
end
end
end

% 输入图形矩阵
% 输出排在最左边的图
function [w,o]=findleft(A)
j=1;
x=[];
for i=1:209
    if(A(:,1,i)==255)
        x(j)=i;
        j=j+1;
    end
end
end
s=length(x);
q=[];
p=1;

```

```

for i=1:s
    max=72;
    for j=1:180
        for z=1:72
            if (A(j,z,x(i))==0)
                if (z<max)
                    max=z;
                end
            end
        end
    end
    q(p)=max;
    p=p+1;
end
e=zeros(1,11);
[b,n]=sort(q);
e=n(s-11:s);
h=zeros(1,s-11);
h=n(1:s-11);
w=zeros(11,1);
for i=1:11
    w(i)=x(1,e(1,i));
end
o=zeros(s-11,1);
for i=1:s-11
    o(i)=x(1,h(1,i));
end
end

% 输入a1,a2
% 输出两幅图是否能够匹配
function [po] = hd (a1,a2);

[h,l1] = size(a1);
[h,l2] = size(a2);

aa = [a1,a2];
co = [aa(:,(l1-59):(l1+60))];

for i = 1:h
    for j = 1:120
        if(co(i,j) < 64) % 灰度调节
            co(i,j) = 0;
        else co(i,j) = 255;
        end
    end
end

```

```

        end
    end
end

edge = zeros(h,1);
ee = 0;
for i = 1:h
    if (co(i,60) + co(i,61) == 0)
        edge(i) = 1;
        ee = ee + 1;
    end
end

```

```

if (ee > 0)
d = zeros(h,1);
for i = 1:h
    for j = 1:120
        if (co(i,j) == 0)
            d(i) = 1;
            break;
        end
    end
end
end

```

% 分行操作

```

l = 0;
maxl = 0;
up = 0;
down = 0;
for i = 1:h
    if (d(i) == 1)
        l = l + 1;
    else if (l > maxl)
        maxl = l;
        down = i - 1;
        up = down - l + 1;
    end
    l = 0;
end
end

ll = floor((up+down)/2) + 32;
if (ll>h)
    ll=ll-64;

```

```

end

% 取出边界特征最明显的一行
tl = mod(11,64);
tt = 0;
tmax = 0;
ttemp = 0;
for i = 1:h
    if (mod(abs(i-tl),64) == 0)
        if (ttemp > tmax)
            tmax = ttemp;
            ttemp = 0;
            tt = i;
        end
    else
        if (edge(i) == 1)
            ttemp = ttemp + 1;
        end
    end
end
end

down = tt;
up = tt - 64;
if (up<=0)
    up = 1;
end
oo = [co(up:down,:)];
else
    up = 0;
    down = 0;
    oo = 0;
end
if (oo == 0)
    po=1;
else
    % 取字
    left = 60;
    right = 61;
    [hh,~] = size(oo);
    col = zeros(1,120);

    for i = 1:120
        for j = 1:hh
            if (oo(j,i) == 0)

```

```

        col(1,i) = 1;
        break;
    end
end
end

while(col(1,left) == 1)
    left = left - 1;
end
while(col(1,right) == 1)
    right = right + 1;
end

ch = [oo(:,left:right)];
[ii,jj] = size(ch);
distance = hausdorff(ch);
    if (distance < 50)
        po = 1;
    else po = 0;
    end
end
end

function [currentlow] = hausdorff(i1);
currentlow=1000;
currentlength=0;
currentnum=1;
[~,length]=size(i1); % 样本长度
for num=1:52 % 对每个模板进行匹配度检测
    imageName=strcat(num2str(num),'.bmp');
    i2=imread(imageName); % 读入模板

    [~,testlength]=size(i2); %读入模板长度
    j1=edge(i1,'canny');
    jj1=edge(i2,'canny');
    [aa,cc]=size(j1);
    [aaa,ccc]=size(jj1);
%
for m1=1:aa
    for n1=1:cc
        if j1(m1,n1)>0
            j1(m1,n1)=1;
        else
            j1(m1,n1)=0;

```

```

        end
    end
end
k1=zeros(aa,cc);
k1(1,1)=j1(1,1);
for m2=2:aa-1
    for n2=2:cc-1
        z=zeros(1,5);
        z(1,1)=j1(m2-1,n2-1)+4;
        z(1,2)=j1(m2-1,n2)+3;
        z(1,3)=j1(m2-1,n2+1)+4;
        z(1,4)=j1(m2,n2-1)+3;
        z(1,5)=j1(m2,n2);
        for x=1:4
            mini=z(1,5);
            if(mini>z(1,x))
                mini=z(1,x);
            end
        end
        k1(m2,n2)=mini;
    end
end
for mm1=1:aaa
    for nn1=1:ccc
        if jj1(mm1,nn1)>0
            jj1(mm1,nn1)=1;
        else
            jj1(mm1,nn1)=0;
        end
    end
end
kk1=zeros(aaa,ccc);
kk1(1,1)=jj1(1,1);
for mm2=2:aaa-1
    for nn2=2:ccc-1
        zz=zeros(1,5);
        zz(1,1)=jj1(mm2-1,nn2-1)+4;
        zz(1,2)=jj1(mm2-1,nn2)+3;
        zz(1,3)=jj1(mm2-1,nn2+1)+4;
        zz(1,4)=jj1(mm2,nn2-1)+3;
        zz(1,5)=jj1(mm2,nn2);
        for xx=1:4
            mini2=zz(1,5);
            if(mini2>zz(1,x))

```

```

        mini2=zz(1,x);
    else
        continue
    end
end
    kk1(mm2,nn2)=mini2;
end
end
t=0;
for t1=1:aa
    for t2=1:cc
        if(k1(t1,t2)==1)
            t=t+1;
        end
    end
end
w=zeros(t,2);
p=0;
for t1=1:aa
    for t2=1:cc

        if k1(t1,t2)==1
            p=p+1;
            w(p,:)=[t1 t2];
        else
            continue
        end
    end
end
tt=0;
for tt1=1:aaa
    for tt2=1:ccc
        if(kk1(tt1,tt2)==1)
            tt=tt+1;
        end
    end
end
ww=zeros(tt,2);
pp=0;
for tt1=1:aaa
    for tt2=1:ccc

        if kk1(tt1,tt2)==1
            pp=pp+1;

```

```

        ww(pp,:)=[tt1 tt2];
    else
        continue
    end
end
end
end
b=zeros(1,t);
min_a=((w(1,1)-ww(1,1)).^2+(w(1,2)-ww(1,2)).^2).^(1/2);
for i=1:t
    for j=1:tt
        if ((w(i,1)-ww(j,1)).^2+(w(i,2)-ww(j,2)).^2)<min_a
            min_a=(w(i,1)-ww(j,1)).^2+(w(i,2)-ww(j,2)).^2;
        end
    end
    b(1,i)=min_a;
end
end % 求出向量b
max_a=b(1,1); %找出最大值
for i=1:t
    if max_a<b(1,i)
        max_a=b(1,i);
    end
end
c=zeros(1,tt);
for l=1:tt
    for k=1:t
        min_b=(ww(1,1)-w(1,1)).^2+(ww(1,2)-w(1,2)).^2;
        if ((ww(1,1)-w(k,1)).^2+(ww(1,2)-w(k,2)).^2)<min_b
            min_b=(ww(1,1)-w(k,1)).^2+(ww(1,2)-w(k,2)).^2;
        end
    end
    c(1,l)=min_b;
end
end % 求出向量c
max_b=c(1,1); % 找出最大值
for l=1:tt
    if max_b<c(1,l)
        max_b=c(1,l);
    end
end
if max_a>max_b
    max_hausdorfff=max_a;
else
    max_hausdorfff=max_b;
end
end

```



```

if (max_hausdorff<currentlow)
    currentlow=max_hausdorff;
    currentlength=testlength;
    currentnum=num;
else if (max_hausdorff==currentlow)
    if ((abs(testlength-length))<(abs(currentlength-length)))
        currentlength=testlength;
        currentnum=num;
    end
end
end
end
end

```

```

function [Q]=getblack()
A = zeros(180,72,209);
for i = 0:2
    for j = 0:9
        for k = 0:9
            if(i * 100 + j * 10 + k <= 208)
                A(:, :, i * 100 + j * 10 + k + 1) =
imread([num2str(i),num2str(j),num2str(k)], 'bmp');
            end
        end
    end
end
end
% 读取图像
for i = 1:209
    for j = 1:180
        for k = 1:72
            if(A(j,k,i) ~= 0 && A(j,k,i) ~= 255)
                A(j,k,i) = 0;
            end
        end
    end
end
end
% 将灰点改成黑点
B=zeros(180,1,209);
for i = 1:209
    for j = 1:180
        temp=0;
        for k = 1:72
            if(A(j,k,i) == 0)
                temp = temp+1;
            end
        end
    end
end

```

```

        end
    end
    B(j,1,i) = temp;
end
end
C = zeros(180,209);
for i=1:209
    C(:,i) = B(:,1,i);
end

C(76,48) = 20;
C(77,48) = 20;
C(54,65) = 20;

Q=zeros(1,209);
for i = 1:209
    max=0;
    h=0;
    for j = 1:179
        if(max < (C(j,i)-C(j+1,i)))
            max = C(j,i)-C(j+1,i);
            h=j;
        end
    end
    Q(1,i)=h;
end
end

% 生成灰度值矩阵
function [gray] = genergray()

A = zeros(180,72,209);
for i = 0:2
    for j = 0:9
        for k = 0:9
            if(i * 100 + j * 10 + k <= 208)
                A(:, :, i * 100 + j * 10 + k + 1) =
imread([num2str(i),num2str(j),num2str(k)], 'bmp');
            end
        end
    end
end
end
gray = zeros(209,254);
for i = 1:209

```

```

    for j = 1:180
        for k = 1:72
            if(A(j,k,i) ~= 0 && A(j,k,i) ~= 255)
                gray(i,A(j,k,i)) = gray(i,A(j,k,i)) + 1;
            end
        end
    end
end
end

% 输入灰度值矩阵
% 生成阈值矩阵
function [sigma] = genersigma(gray)

sigma = zeros(1,209);
miu = zeros(1,209);
for i = 1:209
    sum = 0;
    sum2 = 0;
    sum3 = 0;
    for j = 1:254
        sum = sum + gray(i,j);
        sum2 = sum2 + gray(i,j) * j;
    end
    miu(i) = sum2 / sum;
    for j = 1:254
        sum3 = sum3 + gray(i,j) * (j - miu(i)) ^ 2;
    end
    sigma(i) = miu(i) - sqrt(sum3 / sum);
end
end

```

## 附录 4

% 问题一、二的解决需要 Matlab, 问题三的解决需要 Matlab 和 Lingo  
% 问题一的代码在附录 2, 问题二的代码在附录 3, 问题 3 的代码在附录 4

### 1. Matlab 代码

```
% 该程序也可以自动字符识别
% 该附录为问题三的Matlab代码和Lingo代码
% 程序运行完, 会在该程序的同一目录下生成bmp格式的结果图
function [] = question3()
% 该程序为问题三的主函数
A = zeros(180,72,418);

for i = 0:2
    for j = 0:9
        for k = 0:9
            if(i * 100 + j * 10 + k <= 208)
                A(:, :, (i * 100 + j * 10 + k + 1) * 2 - 1) =
imread([num2str(i), num2str(j), num2str(k), 'a'], 'bmp');
                A(:, :, (i * 100 + j * 10 + k + 1) * 2) =
imread([num2str(i), num2str(j), num2str(k), 'b'], 'bmp');
            end
        end
    end
end
% 读取图像
ansgraph = total(row);
genergraph(A,ansgraph);
col = genermatr2();
finansgraph = total2(col);
finalgraph = [];
for i = 1:11
    graph = A(:, :, finansgraph(i,1));
    for j = 2:19
        graph = [graph, A(:, :, finansgraph(i,j))];
    end
    finalgraph = [finalgraph; graph];
end
imwrite(finalgraph, 'graph(1).bmp');
for i = 12:22
    graph = A(:, :, finansgraph(i,1));
    for j = 2:19
        graph = [graph, A(:, :, finansgraph(i,j))];
    end
end
```

```

        finalgraph = [finalgraph;graph];
end
imwrite(finalgraph, 'graph(2).bmp');
end

function [finansgraph] = total2(col)

A1 = zeros(180,1368,22);
for i = 1:22
    A1(:,:,i) = imread(['graph',num2str(i)], 'bmp');
end

finansgraph = zeros(22,1);
gr = zeros(1,1368);

inputnum = 1;
while(inputnum ~= 0)
    tempj = 1;
    col2 = 10;
    finansgraph(tempj) = col2;
    while(tempj < 11 && ~isempty(find(col(col2,:) ~= 0)))
        p1 = find(col(col2,:) == max(col(col2,:)));
        tempj = tempj + 1;
        finansgraph(tempj) = p1;
        col2 = p1;
    end
    tempj = 12;
    col2 = 12;
    finansgraph(tempj) = col2;
    while(tempj < 22 && ~isempty(find(col(col2,:) ~= 0)))
        p1 = find(col(col2,:) == max(col(col2,:)));
        tempj = tempj + 1;
        finansgraph(tempj) = p1;
        col2 = p1;
    end
    graph = A1(:,:,10);
    for l = 2:22
        graph = [graph;gr;A1(:,:,finansgraph(l))];
    end
    imshow(graph);
    inputnum = input('Which picture is wrong? If none, please enter
0. ');
    if(inputnum ~= 0)
        col(finansgraph(inputnum),finansgraph(inputnum+1)) = 0;

```

```

        end
    end
end

function [ansgraph] = total(row)

A = zeros(180,72,418);

for i = 0:2
    for j = 0:9
        for k = 0:9
            if(i * 100 + j * 10 + k <= 208)
                A(:, :, (i * 100 + j * 10 + k + 1) * 2 - 1) =
imread([num2str(i), num2str(j), num2str(k), 'a'], 'bmp');
                A(:, :, (i * 100 + j * 10 + k + 1) * 2) =
imread([num2str(i), num2str(j), num2str(k), 'b'], 'bmp');
            end
        end
    end
end

for i = 1:418
    for j = 1:180
        for k = 1:72
            if(A(j,k,i) ~= 0 && A(j,k,i) ~= 255)
                A(j,k,i) = 0;
            end
        end
    end
end

ansgraph = zeros(22,19);
w = findleft(A);
w = w';
w(22) = 229;
know = 0;
know = input('Do you need to use OCR function?(1/0)');
if(know == 0)
    gr = zeros(180,1);
    start = input('Where to start?');
    for i = start:22
        inputnum = 1;
        while(inputnum ~= 0)
            tempj = 1;

```

```

row2 = w(i);
ansgraph(i,tempj) = row2;
while(tempj < 19 && ~isempty(find(row(row2,:) ~= 0)))
    p1 = find(row(row2,:) == max(row(row2,:)));
    tempj = tempj + 1;
    ansgraph(i,tempj) = p1;
    row2 = p1;
end
graph = A(:, :, w(i));
for l = 2:tempj
    graph = [graph, gr, A(:, :, ansgraph(i, l))];
end
imshow(graph);
inputnum = input('Which picture is wrong? If none, please
enter 0. ');
if(inputnum ~= 0)
    if(inputnum == 1000)
        inputnum = input('Which picture is wrong? If none,
please enter 0. ');
        posit = input('Which picture do you want to show? ');
        row(ansgraph(i, inputnum), posit) = 10;
    else
        row(ansgraph(i, inputnum), ansgraph(i, inputnum+1)) = 0;
    end
end
end
end
for d = 1:19
    for t = 1:418
        row(t, ansgraph(i, d)) = 0;
        row(ansgraph(i, d), t) = 0;
    end
end
disp((ansgraph(i, :)));
end
else
gr = zeros(180,1);
start = input('Where to start? ');
for i = start:22
    inputnum = 1;
    while(inputnum ~= 0)
        tempj = 1;
        row2 = w(i);
        ansgraph(i, tempj) = row2;
        while(tempj < 19 && ~isempty(find(row(row2,:) ~= 0)))

```

```

        p1 = find(row(row2,:) == max(row(row2,:)));
        if(hd (A(:,:,row2),A(:,:,p1)) == 0)
            row(row2,p1) = 0;
            continue;
        end
        tempj = tempj + 1;
        ansgraph(i,tempj) = p1;
        row2 = p1;
    end
    graph = A(:,:,w(i));
    for l = 2:tempj
        graph = [graph,gr,A(:,:,ansgraph(i,l))];
    end
    imshow(graph);
    inputnum = input('Which picture is wrong? If none, please
enter 0. ');
    if(inputnum ~= 0)
        if(inputnum == 1000)
            inputnum = input('Which picture is wrong? If none,
please enter 0. ');
            posit = input('Which picture do you want to show? ');
            row(ansgraph(i,inputnum),posit) = 10;
        else
            row(ansgraph(i,inputnum),ansgraph(i,inputnum+1)) = 0;
        end
    end
end
end
for d = 1:19
    for t = 1:418
        row(t,ansgraph(i,d)) = 0;
        row(ansgraph(i,d),t) = 0;
    end
end
end
disp((ansgraph(i,:)));
end
end
end

function [col] = genermatr2()

col = zeros(22,22);
tempcol = 0;
temp = 0;

```



```

A = zeros(180,1368,22);
for i = 1:22
    A(:, :, i) = imread(['graph', num2str(i)], 'bmp');
end

for i = 1:22
    for j = 1:1368
        if (A(1,j,i) ~= 255)
            A(1,j,i) = 0;
        end
        if (A(180,j,i) ~= 255)
            A(180,j,i) = 0;
        end
    end
end

for i = 1:22
    for j = 1:22
        if(i == j)
            col(i,j) = 0;
        else
            for h = 1:1368
                if(A(180,h,i) == 0 && A(1,h,j) == 0)
                    tempcol = tempcol + 1;
                end
                if(A(180,h,i) == 0)
                    temp = temp + 1;
                end
                if(A(1,h,j) == 0)
                    temp = temp + 1;
                end
            end
            if(temp == 0)
                col(i,j) = 1; % 两边都无黑点，匹配度为1
            else
                col(i,j) = 2 * tempcol / temp;
            end
        end
        tempcol = 0;
        temp = 0;
    end
end

for i = 1:22

```

```

        for j = 1:22
            if(i ~= j)
                col(i,j) = col(i,j) + rand()*0.01;
            end
        end
    end
end
end

function [row] = genermatr()

row = zeros(418,418);
temprow = 0;
t1=0;
A = zeros(180,72,418);

for i = 0:2
    for j = 0:9
        for k = 0:9
            if(i * 100 + j * 10 + k <= 208)
                A(:, :, (i * 100 + j * 10 + k + 1) * 2 - 1) =
imread([num2str(i),num2str(j),num2str(k), 'a'], 'bmp');
                A(:, :, (i * 100 + j * 10 + k + 1) * 2) =
imread([num2str(i),num2str(j),num2str(k), 'b'], 'bmp');
            end
        end
    end
end
end
%读取图像

for i = 1:418
    for j = 1:180
        for k = 1:72
            if(A(j,k,i) ~= 0 && A(j,k,i) ~= 255)
                A(j,k,i) = 0;
            end
        end
    end
end
end
% 读取图像

w = findleft(A); % 22个最左
Q = getblack(); % 每幅图的下限

for i = 1:418

```

```

for j = 1:418
    if(i == j)
        row(i,j) = 0;
    else
        for k = 1:180
            if(A(k,72,i) == 0 && A(k,1,j) == 0)
                temprow = temprow + 1;
            end
            if(A(k,72,i) == 0)
                t1=t1+1;
            end
            if(A(k,1,j) == 0)
                t1=t1+1;
            end
        end
        if(t1==0)
            row(i,j)=1;
        else
            row(i,j) = 2 * temprow / t1;
        end
        t1 = 0;
    end
    temprow = 0;
end

for i = 1:418
    for j = 1:418
        if(i ~= j)
            row(i,j) = row(i,j) + rand()*0.05;
        end
    end
end

for i = 1:418
    for j = (i+1):418
        if(mod((Q(i) - Q(j)),64) > 2 && 64 - mod((Q(i) - Q(j)),64) >
2)
            row(i,j) = 0;
            row(j,i) = 0;
        end
    end
end
end
% 判断两幅图是否可能属于同一行

```

```

for i = 1:22
    for j = 1:418
        row(j,w(i)) = 0;
    end
end

end

function [w]=findup(A)
[a,b,c]=size(A);
j=1;
x=[];
for i=1:c
    if(A(1,:,i)==255)
        x(j)=i;
        j=j+1;
    end
end
s=length(x);
q=[];
p=1;
for i=1:s
    max=72;
    for j=1:b
        for z=1:a
            if(A(z,j,x(i))==0)
                if(z<max)
                    max=z;
                end
            end
        end
        end
        q(p)=max;
        p=p+1;
    end
    e=zeros(1,2);
    [d,n]=sort(q);
    e=n(s-1:s);
    w=zeros(2,1);
    for i=1:2
        w(i)=x(1,e(1,i));
    end
end
end

```

```

function [w]=findleft(A)
j=1;
x=[];
for i=1:418
    if(A(:,1,i)==255)
        x(j)=i;
        j=j+1;
    end
end
s=length(x);
q=[];
p=1;
for i=1:s
    max=72;
    for j=1:180
        for z=1:72
            if(A(j,z,x(i))==0)
                if(z<max)
                    max=z;
                end
            end
        end
    end
    q(p)=max;
    p=p+1;
end
e=zeros(1,22);
[b,n]=sort(q);
e=n(s-21:s);
w=zeros(22,1);
for i=1:22
    w(i)=x(1,e(1,i));
end
end

function [Q]=getblack()
A = zeros(180,72,418);
for i = 0:2
    for j = 0:9
        for k = 0:9
            if(i * 100 + j * 10 + k <= 208)
                A(:,:(i * 100 + j * 10 + k + 1) * 2 - 1) =
imread([num2str(i),num2str(j),num2str(k),'a'],'bmp');

```

```

        A(:, :, (i * 100 + j * 10 + k + 1) * 2) =
imread([num2str(i), num2str(j), num2str(k), 'b'], 'bmp');
    end
end
end
end
% 读取图像

for i = 1:418
    for j = 1:180
        for k = 1:72
            if(A(j,k,i) ~= 0 && A(j,k,i) ~= 255)
                A(j,k,i) = 0;
            end
        end
    end
end
% 将灰点改成黑点
B=zeros(180,1,418);
for i = 1:418
    for j = 1:180
        temp=0;
        for k = 1:72
            if(A(j,k,i) == 0)
                temp = temp+1;
            end
        end
        B(j,1,i) = temp;
    end
end
C = zeros(180,418);
for i=1:418
    C(:,i) = B(:,1,i);
end
Q=zeros(1,418);
for i = 1:418
    max=0;
    h=0;
    for j = 1:179
        if(max < (C(j,i)-C(j+1,i)))
            max = C(j,i)-C(j+1,i);
            h=j;
        end
    end
end
end

```

```

        Q(1,i)=h;
    end
end

function [] = genergraph(A,ansgraph)

for i = 1:22
    graph = A(:, :, ansgraph(i,1));
    for j = 2:19
        graph = [graph, A(:, :, ansgraph(i,j))];
    end
    imwrite(graph, ['graph', num2str(i), '.bmp']);
end

end

function [startmatr] = gener22matr()

A = zeros(180,72,418);

for i = 0:2
    for j = 0:9
        for k = 0:9
            if(i * 100 + j * 10 + k <= 208)
                A(:, :, (i * 100 + j * 10 + k + 1) * 2 - 1) =
imread([num2str(i), num2str(j), num2str(k), 'a'], 'bmp');
                A(:, :, (i * 100 + j * 10 + k + 1) * 2) =
imread([num2str(i), num2str(j), num2str(k), 'b'], 'bmp');
            end
        end
    end
end
% 读取图像

for i = 1:418
    for j = 1:180
        for k = 1:72
            if(A(j,k,i) ~= 0 && A(j,k,i) ~= 255)
                A(j,k,i) = 0;
            end
        end
    end
end
% 将灰点改成黑点

```

```

w = [28 72 182 212 287 12 19 109 168 273 346 158 179 199 293 332 374
400 8 48 178 229]; % 22个最左
Q = getblack(); % 每幅图的下限

startmatr = zeros(22,22);

for i = 1:22
    for j = 1:22
        if(mod(Q(w(i)) - Q(w(j)),64) < 32)
            startmatr(i,j) = mod(Q(w(i)) - Q(w(j)),64);
            startmatr(j,i) = mod(Q(w(i)) - Q(w(j)),64);
        else
            startmatr(i,j) = 64 - mod(Q(w(i)) - Q(w(j)),64);
            startmatr(j,i) = 64 - mod(Q(w(i)) - Q(w(j)),64);
        end
    end
end

for i = 1:22
    startmatr(i,i) = 32;
end

end

% 输入a1,a2
% 输出两幅图是否能够匹配
function [po] = hd (a1,a2);

[h,l1] = size(a1);
[h,l2] = size(a2);

aa = [a1,a2];
co = [aa(:,(l1-59):(l1+60))];

for i = 1:h
    for j = 1:120
        if(co(i,j) < 64) % 灰度调节
            co(i,j) = 0;
        else co(i,j) = 255;
        end
    end
end
end

```



```

edge = zeros(h,1);
ee = 0;
for i = 1:h
    if (co(i,60) + co(i,61) == 0)
        edge(i) = 1;
        ee = ee + 1;
    end
end

```

```

if (ee > 0)
d = zeros(h,1);
for i = 1:h
    for j = 1:120
        if (co(i,j) == 0)
            d(i) = 1;
            break;
        end
    end
end
end

```

% 分行操作

```

l = 0;
maxl = 0;
up = 0;
down = 0;
for i = 1:h
    if (d(i) == 1)
        l = l + 1;
    else if (l > maxl)
        maxl = l;
        down = i - 1;
        up = down - l + 1;
    end
    l = 0;
end
end
l1 = floor((up+down)/2) + 32;
if (l1>h)
    l1=l1-64;
end

```

% 取出边界特征最明显的一行

```

t1 = mod(l1,64);

```

```

tt = 0;
tmax = 0;
ttemp = 0;
for i = 1:h
    if (mod(abs(i-tl),64) == 0)
        if (ttemp > tmax)
            tmax = ttemp;
            ttemp = 0;
            tt = i;
        end
    else
        if (edge(i) == 1)
            ttemp = ttemp + 1;
        end
    end
end

down = tt;
up = tt - 64;
if (up<=0)
    up = 1;
end
oo = [co(up:down,:)];
else
    up = 0;
    down = 0;
    oo = 0;
end
if (oo == 0)
    po=1;
else
    % 取字
    left = 60;
    right = 61;
    [hh,~] = size(oo);
    col = zeros(1,120);

    for i = 1:120
        for j = 1:hh
            if (oo(j,i) == 0)
                col(1,i) = 1;
                break;
            end
        end
    end
end

```

```

end

while(col(1,left) == 1)
    left = left - 1;
end
while(col(1,right) == 1)
    right = right + 1;
end

ch = [oo(:,left:right)];
[ii,jj] = size(ch);
distance = hausdorff(ch);
    if (distance < 50)
        po = 1;
    else po = 0;
    end
end
end

function [currentlow] = hausdorff(i1);
currentlow=1000;
currentlength=0;
currentnum=1;
[~,length]=size(i1); % 样本长度
for num=1:52 % 对每个模板进行匹配度检测
    imageName=strcat(num2str(num),'.bmp');
    i2=imread(imageName); % 读入模板

    [~,testlength]=size(i2); %读入模板长度
    j1=edge(i1,'canny');
    jj1=edge(i2,'canny');
    [aa,cc]=size(j1);
    [aaa,ccc]=size(jj1);
%
for m1=1:aa
    for n1=1:cc
        if j1(m1,n1)>0
            j1(m1,n1)=1;
        else
            j1(m1,n1)=0;
        end
    end
end
end
k1=zeros(aa,cc);

```

```

k1(1,1)=j1(1,1);
for m2=2:aa-1
    for n2=2:cc-1
        z=zeros(1,5);
        z(1,1)=j1(m2-1,n2-1)+4;
        z(1,2)=j1(m2-1,n2)+3;
        z(1,3)=j1(m2-1,n2+1)+4;
        z(1,4)=j1(m2,n2-1)+3;
        z(1,5)=j1(m2,n2);
        for x=1:4
            mini=z(1,5);
            if(mini>z(1,x))
                mini=z(1,x);
            end
        end
        k1(m2,n2)=mini;
    end
end
for mm1=1:aaa
    for nn1=1:ccc
        if jj1(mm1,nn1)>0
            jj1(mm1,nn1)=1;
        else
            jj1(mm1,nn1)=0;
        end
    end
end
end
kk1=zeros(aaa,ccc);
kk1(1,1)=jj1(1,1);
for mm2=2:aaa-1
    for nn2=2:ccc-1
        zz=zeros(1,5);
        zz(1,1)=jj1(mm2-1,nn2-1)+4;
        zz(1,2)=jj1(mm2-1,nn2)+3;
        zz(1,3)=jj1(mm2-1,nn2+1)+4;
        zz(1,4)=jj1(mm2,nn2-1)+3;
        zz(1,5)=jj1(mm2,nn2);
        for xx=1:4
            mini2=zz(1,5);
            if(mini2>zz(1,x))
                mini2=zz(1,x);
            else
                continue
            end
        end
    end
end

```

```

        end
        kk1(mm2,nn2)=mini2;
    end
end
t=0;
for t1=1:aa
    for t2=1:cc
        if(k1(t1,t2)==1)
            t=t+1;
        end
    end
end
w=zeros(t,2);
p=0;
for t1=1:aa
    for t2=1:cc

        if k1(t1,t2)==1
            p=p+1;
            w(p,:)=[t1 t2];
        else
            continue
        end
    end
end
tt=0;
for tt1=1:aaa
    for tt2=1:ccc
        if(kk1(tt1,tt2)==1)
            tt=tt+1;
        end
    end
end
ww=zeros(tt,2);
pp=0;
for tt1=1:aaa
    for tt2=1:ccc

        if kk1(tt1,tt2)==1
            pp=pp+1;
            ww(pp,:)=[tt1 tt2];
        else
            continue
        end
    end
end

```

```

        end
    end
    b=zeros(1,t);
    min_a=((w(1,1)-ww(1,1)).^2+(w(1,2)-ww(1,2)).^2).^(1/2);
    for i=1:t
        for j=1:tt
            if ((w(i,1)-ww(j,1)).^2+(w(i,2)-ww(j,2)).^2)<min_a
                min_a=(w(i,1)-ww(j,1)).^2+(w(i,2)-ww(j,2)).^2;
            end
        end
        b(1,i)=min_a;
    end
end % 求出向量b
max_a=b(1,1); %找出最大值
for i=1:t
    if max_a<b(1,i)
        max_a=b(1,i);
    end
end
c=zeros(1,tt);
for l=1:tt
    for k=1:t
        min_b=(ww(1,1)-w(1,1)).^2+(ww(1,2)-w(1,2)).^2;
        if ((ww(1,1)-w(k,1)).^2+(ww(1,2)-w(k,2)).^2)<min_b
            min_b=(ww(1,1)-w(k,1)).^2+(ww(1,2)-w(k,2)).^2;
        end
    end
    c(1,l)=min_b;
end
end % 求出向量c
max_b=c(1,1); % 找出最大值
for l=1:tt
    if max_b<c(1,l)
        max_b=c(1,l);
    end
end
if max_a>max_b
    max_hausdorff=max_a;
else
    max_hausdorff=max_b;
end

if (max_hausdorff<currentlow)
    currentlow=max_hausdorff;
    currentlength=testlength;
    currentnum=num;
end

```

```

else if (max_hausdorff==currentlow)
    if ((abs(testlength-length))<(abs(currentlength-length)))
        currentlength=testlength;
        currentnum=num;
    end
end
end
end
end

```

## 2. Lingo 代码

```

model:
sets:
department/1..22/;
type/1..22/;
benefit(department,type):d,x;
endsets
min=@sum(benefit(i,j):d(i,j)*x(i,j));
@for(benefit:@bin(x));
@for(department(i):
    @sum(type(j):x(i,j))=1);
@for(type(i):@sum(department(j):x(i,j))=1);
@for(benefit(i,j):x(i,j)=x(j,i));
data:
d=32   13  10  31  31  20  22  10  20  9   23  10  19  10  11  21  30  22  31
    20  11  1
13  32  23  20  20  7   29  23  31  4   10  3   6   23  2   8   17  29  18  31
    24  14
10  23  32  21  21  30  12  0   10  19  31  20  29  0   21  31  24  12  23  10
    1   9
31  20  21  32  0   13  9   21  11  24  10  23  14  21  22  12  3   9   2   11
    20  30
31  20  21  0   32  13  9   21  11  24  10  23  14  21  22  12  3   9   2   11
    20  30
20  7   30  13  13  32  22  30  24  11  3   10  1   30  9   1   10  22  11  24
    31  21
22  29  12  9   9   22  32  12  2   31  19  32  23  12  31  21  12  0   11  2
    11  21
10  23  0   21  21  30  12  32  10  19  31  20  29  0   21  31  24  12  23  10
    1   9

```

```

20 31 10 11 11 24 2 10 32 29 21 30 25 10 31 23 14 2 13 0
9 19
9 4 19 24 24 11 31 19 29 32 14 1 10 19 2 12 21 31 22 29
20 10
23 10 31 10 10 3 19 31 21 14 32 13 4 31 12 2 7 19 8 21
30 24
10 3 20 23 23 10 32 20 30 1 13 32 9 20 1 11 20 32 21 30
21 11
19 6 29 14 14 1 23 29 25 10 4 9 32 29 8 2 11 23 12 25
30 20
10 23 0 21 21 30 12 0 10 19 31 20 29 32 21 31 24 12 23 10
1 9
11 2 21 22 22 9 31 21 31 2 12 1 8 21 32 10 19 31 20 31
22 12
21 8 31 12 12 1 21 31 23 12 2 11 2 31 10 32 9 21 10 23
32 22
30 17 24 3 3 10 12 24 14 21 7 20 11 24 19 9 32 12 1 14
23 31
22 29 12 9 9 22 0 12 2 31 19 32 23 12 31 21 12 32 11 2
11 21
31 18 23 2 2 11 11 23 13 22 8 21 12 23 20 10 1 11 32 13
22 32
20 31 10 11 11 24 2 10 0 29 21 30 25 10 31 23 14 2 13 32
9 19
11 24 1 20 20 31 11 1 9 20 30 21 30 1 22 32 23 11 22 9
32 10
1 14 9 30 30 21 21 9 19 10 24 11 20 9 12 22 31 21 32 19
10 32
;
enddata
end

```