

2013 高教社杯全国大学生数学建模竞赛

承 诺 书

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

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

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

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

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

我们参赛选择的题号是(从 A/B/C/D 中选择一项填写): B

我们的参赛报名号为(如果赛区设置报名号的话): 5060

所属学校(请填写完整的全名): 广东财经大学

参赛队员(打印并签名): 1. 陈佳润

2. 谢天旺

3. 王行志

指导教师或指导教师组负责人(打印并签名): 胡桂武

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

日期: 2013 年 09 月 15 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）：

碎纸复原模型与算法

摘要

本文围绕碎纸片拼接问题，建立了**碎纸距离模型**、**复原 TSP 模型**，并设计了一维碎纸复原算法、二维碎纸复原算法、三维碎纸复原算法等算法，利用 MATLAB 实现对问题的求解。

针对问题一，设计了一维碎纸复原算法（见 4.1.7），首先提取出附件 1 和附件 2 碎纸图片的像素矩阵，并对其进行二值化处理，然后利用 MATLAB 提取碎纸图片的文字特征，通过字符大小、行距等文字特征构造识别序列，并且利用识别序列的吻合度建立了碎纸距离模型（见 4.1.4），进而将碎纸复原问题转化为复原 TSP 问题（见 4.1.5），并用模拟退火法进行求解，得到了正确的复原图形及序列（详见附录一）。

针对问题二，设计了二维碎纸复原算法（见 4.2.5），首先对附件 3 和附件 4 的碎纸图片进行标准化，并对标准化有误的图片进行修正，然后提取标准化后图片的层次特征，利用层次特征对图片进行初步分类，对于机器不能分类的图片，通过编制 GUI 程序提高了人工判别效率，得到相应 11 类行特征相同的碎纸，进而将问题转化为 11 个一维碎纸复原问题并进行求解，得到了正确的复原图形及序列（详见附录二）。

针对问题三，设计了三维碎纸复原算法（见 4.3.2），首先对附件 5 的 a 面与 b 面的图片进行整合，得到 416 张三维碎纸图片，同样对图片进行标准化、提取层次特征、分类等操作，将问题维度降为一维并进行求解，得到附件 5 正反面的正确复原图片及序列（详见附录三）。

考虑到算法的量化评价问题，本文在模型改进处提出最小干预度算法，即通过计算机识别顺序与复原顺序的序列逆序数，实现了最小人工干预次数对算法优劣进行刻画。

关键词：碎纸复原算法、TSP、模拟退火法、分类降维、GUI 设计

一、问题重述

破碎文件的拼接在多项领域中有着重要的应用。传统的人工拼接复原，准确率较高，但效率很低，不能在短时间内处理大数量碎片。随着计算机技术的发展，提高拼接复原效率的碎纸片自动拼接技术被试图开发。请讨论以下问题：

1. 对于给定的来自同一页印刷文字文件的碎纸机破碎纸片（仅纵切），建立碎纸片拼接复原模型和算法，并针对附件 1、附件 2 给出的中、英文各一页文件的碎片数据进行拼接复原。
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。
3. 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。建立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。

二、问题分析

本题是研究碎纸复原问题，问题一至三，分别研究一至三维三种不同情况的碎纸复原，本文对问题的解答也是沿着这样的思维路线，即通过降维，使高维度问题转化为低维度问题。

问题一是一个一维碎纸复原问题，首先从图片中提取出像素矩阵，并对其进行二值化处理，然后根据附件碎纸均为大小、形状一致矩形选择提取其文字特征得到基于文字特征的识别序列，由识别序列定义碎纸片之间距离，将寻找最佳吻合碎纸片组合转化为寻找最短路径的 TSP 问题，并编制一维碎纸复原算法进行求解。

问题二是一个二维碎纸复原问题，碎纸经过横切纵切之后碎片间的文字特征减少，继续沿用第一问模型求解计算复杂度呈指数级递增，计算精度也将下降，考虑分析碎纸片文字行间距、大小等特点提出对应准则模型对碎纸片进行识别分类，运用问题一模型与算法先对每一类内碎纸片进行拼接复原，而后再对得到部分拼接复原图进行拼接复原，分类、复原过程适当进行人工干预提高复原精度。

问题三是一个三维碎纸复原问题，需要在问题二的基础上考虑碎纸片的正反面问题。通过问题二的分类模型进行修改，对碎纸片进行分类，通过分类对问题三进行降维处理，然后按问题二方法解决问题，复原过程中适当进行人工干预提高复原精度。

此外，本题是个实际问题，现实中需要考虑的因素远远多于题目本身，如何使模型更加贴近实际，并提供有效的拼接信息是本文面临的一大困难，通过查阅大量资料与文献，并进行适当的、合理的假设，对碎纸复原方式进行研究。

三、符号说明与模型假设

3.1 符号说明

符号	说明	符号	说明
q_{ij}	二值化前的像素值	R_i	碎片 A 的右识别序列
P_{ij}	二值化后的像素值	C	字符的宽度
$d_{a \rightarrow b}$	碎片 A 到碎片 B 的距离	D	TSP 路径的总距离
L_i	碎片 B 的左识别序列	n	识别序列长度

3.2 模型假设

3.2.1 假设文字方向是水平的

3.2.2 假设正反面打印的页边距等格式相同

3.3 假设说明

假设 3.2.1 是为了保证问题一中识别序列的有效性，与实际中大多数情况相符。

假设 3.2.2 是为了简化问题三，当页边距等打印格式相同时，正反面的层次特征会相同，可以进行降维，与实际中大多数情况是相符的

四、模型的建立与求解

4.1 一维碎纸复原模型

4.1.1 图像的预处理

根据计算机图形学的相关知识，划分空白位置和字符位置前，需对图片像素的处理，一般是通过灰度化，将图片像素值定位为 $[0, 255]$ 区间内，再通过设定阈值，区分空白位置和字体，最后进行二值化。但对于非彩色的图片，仅需区分空白与非空白两种情况即可。

为了使图片能够清晰地描述空白位置与字符位置，首先利用数学软件 MATLAB 对图像进行预处理。首先将图片导入 MATLAB，得到对应的像素矩阵，由计算机图形学相关知识可知，空白位置的像素值为 255，为了使像素矩阵特征更为明显，需将其二值化，即

$$P_{ij} = \begin{cases} 1, q_{ij} = 255 \\ 255, \text{其它} \end{cases}$$

式中， q_{ij} 为二值化前的像素值

P_{ij} 为二值化后的像素值

通过对每张图片进行二值化，得到一系列的像素矩阵，可用于图像的特征提取。

4.1.2 碎纸特征的提取

一般来说，碎纸特征的提取大致分为两类，一类是提取碎纸的轮廓，通过外形特征进行拼接，如文献^{[1][2]}，另一类是通过提取碎纸上的文字特征，基于文字特征进行拼接，如文献^{[3][4]}，根据题意，本文探讨的碎纸片外形皆为长方形，故属于第二类。

一维碎纸，是指切割时，仅进行横或纵的切割，形状如图 4-1 所示。

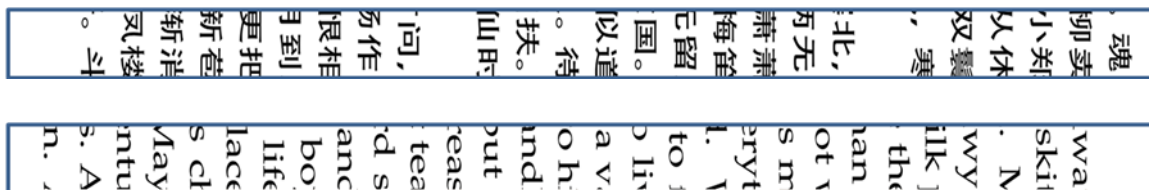


图 4-1 一维碎纸示意图

通过观察图 4-1 可以看出，碎纸可以提取出的特征有字符宽度、字距、上空、下空和行距等，见图 4-2。

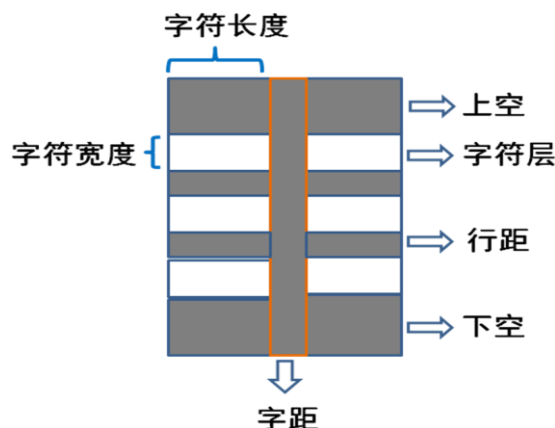


图 4-2 字符的特征

综上所述，文字特征的提取方式如下：

提取步骤：

步骤一：对图片进行二值化，文字为白色，空白为黑色

步骤二：找出图片所有行距、上空、下空，并标记其为灰色

步骤三：找出图片中所有的字距，并标记其为灰色

步骤四：通过行距、上空、下空、字距等特征计算出字符宽度。

根据题意，本文通过导入碎纸片图像的像素，并编写 MATLAB 程序对文字特征进行提取（代码见附录四 getLandR.m）。

4.1.3 基于文字特征的认识序列

通过分析汉字与英文字母被分割的情况可以发现，对于每个被切割开的字符，如图 4-3 所示，记字符的宽度为 C ，被分割后的两部分宽度分别为 $C-R$ 和 R ，而对于没被切割的字符，依旧保留宽度 C 。

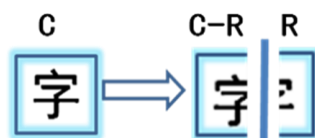


图 4-3 字符分割示意图

基于汉字分割的定义，可构造基于文字特征的认识序列，如图 4-4 所示。

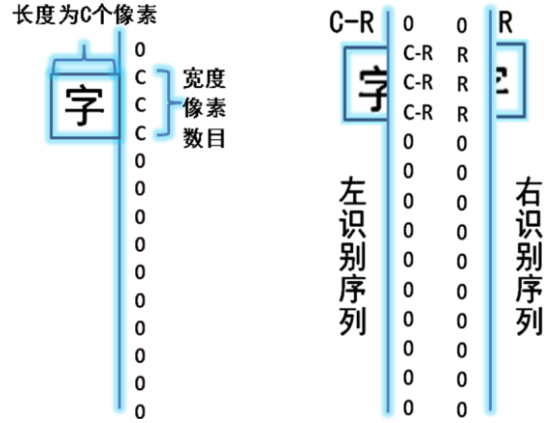


图 4-4 识别序列示意图

对于图 4-4 中的识别序列，其中无字符位置数值为 0，其它数字节点代表对应的字符长度（完整的为 C ，不完整的为 $C-R$ 或 R ）。

4.1.4 碎纸距离的定义

基于 4.1.3 识别序列的定义，本文定义碎纸 A 到碎纸 B 的距离模型为

$$\begin{cases} d_{A \rightarrow B} = \sum_{i=1}^n (L_i + R_i - CX_i)^2 \\ X_i = 0 \text{ 或 } 1 \end{cases} \quad (4.1.4)$$

式中， $d_{a \rightarrow b}$ 为碎纸 A 到碎纸 B 的距离

L_i 为碎纸 B 的左识别序列

R_i 为碎纸 A 的右识别序列

C 为字符的宽度

X_i 为表示序列 L 或 R 的第 i 位是否都有字符，是则为 1

n 为序列长度

由距离的定义公式可以看出，两个识别序列的吻合程度越大，识别序列的距离越小，理想状况下，当两个识别序列完全吻合时，距离为 0（实际情况中，由于二值化，损失了部分精度，所以不会为 0）。

4.1.5 复原 TSP 问题

TSP 问题是数图论中最著名的问题之一，即“已给一个 n 个点的完全图，每条边都有一个长度，求总长度最短的经过每个顶点正好一次的封闭回路”。

若将每个碎纸看成一个点，由 4.1.4 距离的定义可知，点与点之间存在距离，可以看出，两张碎纸如果吻合度低，那对应的距离也大，所以寻找吻合率最高的组合方式，实质就是寻找总距离最小的路径，也就是寻找一条最佳 TSP 路径。

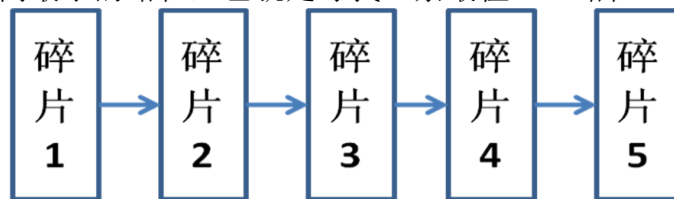


图 4-5 复原 TSP 问题示意图

将碎纸复原抽象成复原 TSP 问题，即

$$\begin{aligned} \min D &= \sum_{i=1}^{n-1} d_{i \rightarrow i+1} \\ S.T. \begin{cases} d_{A \rightarrow B} = \sum_{i=1}^n (L_i + R_i - CX_i)^2 \\ X_i = 0 \text{ 或 } 1 \end{cases} \end{aligned} \quad (4.1.5)$$

式中， D 为 TSP 路径的总距离

$d_{i \rightarrow i+1}$ 为碎纸 i 到碎纸 $i+1$ 的距离

通过求解复原 TSP 问题，可以得到每个点的访问顺序，即碎纸片的拼接序列，最后利用 MATLAB 图像拼接，得到复原了的纸片。

4.1.6 模拟退火法

模拟退火法是一种通用概率算法，用来在一个大的搜寻空间内寻找问题的最优解，具有能有效地解决 NP 难问题、避免陷入局部最优、对初值没有强依赖性等特点，在各领域已获得了广泛地应用。

算法步骤^[5]：

步骤一：令当前温度 $T = T_0$ ，即开始退火的初始温度，随机生成一个初始解 x_0 ，并计算相应的目标函数值 $E(x_0)$ 。

步骤二：令 T 等于冷却进度表的下一个值 T_i 。

步骤三：根据当前解 x_i 进行扰动，产生一个新解 x_j ，计算相应的目标函数值 $E(x_j)$ ，得到 $\Delta E = E(x_j) - E(x_i)$ 。

步骤四：若 $\Delta E > 0$ ，则新解 x_j 被接受，作为新的解；若 $\Delta E < 0$ ，则新解按概率 $e^{\frac{-\Delta E}{T_i}}$ 接受（即 Metropolis 准则）， T_i 为当前温度。

步骤五：在温度 T_i 下，重复 L_k 次扰动和接受过程（ L_k 为 Markov 链的长度），即执行步骤三和步骤四。

步骤六：判断温度 T 是否等于 T_f ，是则终止算法；否则转到步骤二继续执行。

利用模拟退火法求解求解 TSP 问题，将一个序列当成一个解，每个解对应一个目标函数值，通过二次交换、三次交换等扰动方式不断产生新解，从而搜索解空间，寻找最优的调度序列。

4.1.7 一维碎纸复原算法

综上所述，通过一维碎纸复原算法，可以实现一维碎纸的自动复原，相应的算法步骤如下：

算法步骤：

步骤一：提取出碎纸图像的像素矩阵，并进行二值化处理

步骤二：提取出二值化处理后碎纸图像的文字特征

步骤三：利用步骤二的文字特征构造识别序列，并转化为 TSP 问题

步骤四：利用模拟退火法求解 TSP 问题

4.1.8 模型求解

利用 Matlab 对图片像素进行分析，得到汉字大小 40×40（单位：像素），考虑到英文字符的特殊性，受文献^[3]的启示，将其视为 30×30（单位：像素）；通过模拟退火法求解 TSP 问题，取初始温度为 97，停止温度为 3，衰减系数为 0.9，马尔可夫链长度为 10000，利用 Matlab 进行运算，得到结果如表 4-1 所示.

表 4-1 附件 1、附件 2 拼接序列

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
附件 1	9	15	13	16	4	11	3	17	2	5	6	10	14	19	12	8	18	1	7
附件 2	4	7	3	8	16	19	12	1	6	2	10	14	11	9	13	15	18	17	5

利用表 4-1 的序列，可将碎纸片拼接处完整的纸片，且无需人工干预，复原图像见附表 1。

4.2 二维碎纸复原模型

对于二维碎纸，本文的定义是切割纸片时，进行横和纵的切割，形状如图 4-6 所示.

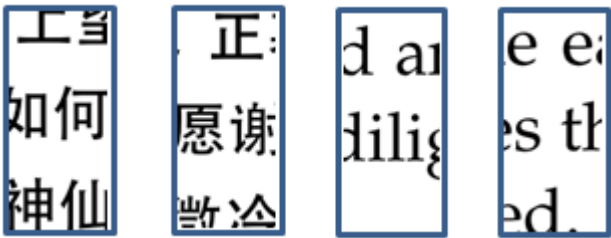


图 4-6 二维碎纸片示意图

与一维碎纸相比，二维的碎纸经过横切和纵切，可以视为是一维碎纸再切割形成的，而再切割会降低碎纸的特征值，如附件 3、附件 4，一维碎纸片被切割为 11 分，导致特征值变为原有的十一分之一，且 TSP 问题的节点由 19 个变为 208 个，算法复杂度大大增加，模拟退火法难以解决，需要通过降维减低算法复杂度，以便求解。

4.2.1 碎纸标准化与层次特征提取

碎纸片的标准化，是指将字符层与空白分离，即

$$f_i = \begin{cases} \text{灰色, } i \text{ 为空白层} \\ \text{黑色, } i \text{ 为字符层} \end{cases}$$

式中， f_i 为图片像素矩阵的第 i 行

具体做法如图 4-7 所示

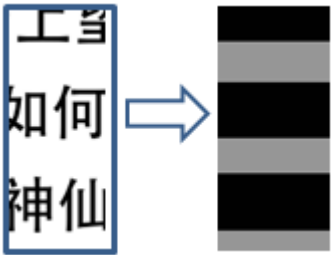


图 4-7 碎纸片的标准化

考虑到本文对碎纸进行标准化的方式，对某些字母不能有效地分层，如图 4-8 所示.



图 4-8 特殊字符

如图 4-9，一些字符会使分层数增加，但是会导致提取层次特征有误，故应该对这些特殊字符进行修正，即令其图片中间的空白行也变为黑色。

对于标准化后的图片，其层次特征明显，可提取出图片的层次特征，本文通过提取文字层的层次特征对图片进行分类，提取的步骤如下（程序见附录四）：

提取步骤：

步骤一：对图片进行标准化；

步骤二：对标准化后的图片进行修正

步骤三：遍历标准化后的像素矩阵，记录图片每个文字层的起始位置和结束位置；

例如图 4-8 的图片，提取的层次特征如图 4-9 所示。

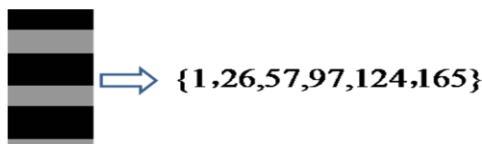


图 4-9 层次特征提取

4.2.3 通过层次特征分类

通过 4.2.2 的层次特征提取，每张图片都得到一组对应的层次特征，可利用这些特征对二维碎纸进行分类，以降低问题的维数，分类需按照一定的步骤：

分类步骤：

步骤一：对比每张图片的特征值，若有 3-4 个特征值相同，那么两张图片归为一类；

步骤二：对于无法分类的图片，记为一个集合；

步骤三：利用 GUI 程序，对于无法分类的图片集合，人为地逐个与已知类别进行对比，从而进行分类。

为了更好地对无法分类的碎纸进行分类，本文通过 MATLAB 的 GUI 编程，实现了步骤三对比程序的设计（代码见附录四 SingleMatch.m），GUI 界面如图 4-10 所示。

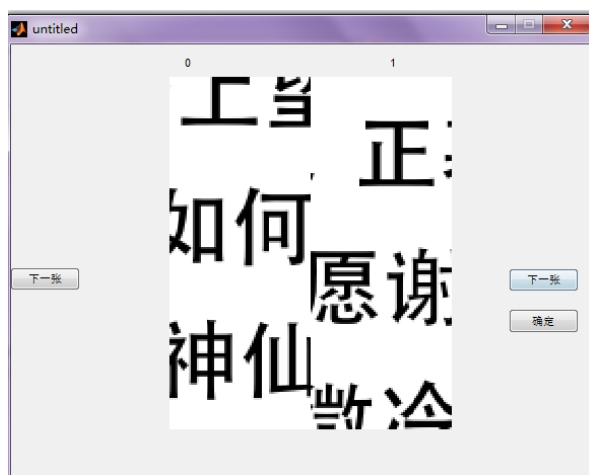


图 4-10 GUI 程序

通过计算机初始分类和人为精细分类，可对二维碎纸进行分类，分类后转化为多个一维碎纸复原问题。

4.2.5 二维碎纸复原算法

根据 4.2.4 设定的分类方法，可以将二维碎纸片复原问题转化为多个一维碎纸片复原问题，但是由于文字特征只有原本的一小部分，且空白区域占碎纸片大部分比例，导致一维碎纸复原模型复原程度不高，需要对结果进行人工干预，如图 4-11 所示。

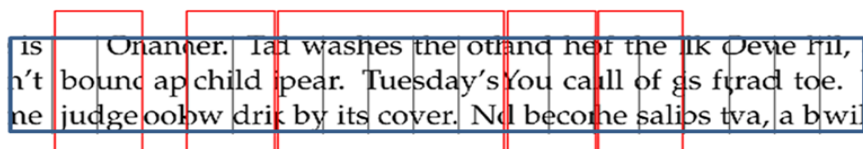


图 4-11 转化后的一维碎纸复原情况

图 4-11 中，红圈区域代表拼接成功的碎纸，由图可看出，一维碎纸复原模型能将大部分碎纸拼接，但仍需人工调整。

综上所述，二维碎纸复原步骤为：

复原步骤：

步骤一：对二维碎纸进行标准化；

步骤二：提取二维碎纸的层次特征；

步骤三：利用层次特征，进行机器划分和小部分人工干预，实现分类；

步骤四：构造多个一维碎纸问题，并进行求解；

步骤五：根据步骤四的结果，进行适当的人工校正。

4.2.6 模型求解

利用 4.2.5 的碎纸复原算法分别对附件 3、附件 4 的二维碎纸进行复原，分别转化为 11 个一维碎纸复原问题，并利用 4.1 的碎纸复原模型进行复原，得到结果见附录二。

表 4-2 附件 3 结果顺序表

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
1	049	054	065	143	186	002	057	192	178	118	190	095	011	022	129	028	091	188	141
2	061	019	078	067	069	099	162	096	131	079	063	116	163	072	006	177	020	052	036
3	168	100	076	062	142	030	041	023	147	191	050	179	120	086	195	026	001	087	018
4	038	148	046	161	024	035	081	189	122	103	130	193	088	167	025	008	009	105	074
5	071	156	083	132	200	017	080	033	202	198	015	133	170	205	085	152	165	027	060
6	014	128	003	159	082	199	135	012	073	160	203	169	134	039	031	051	107	115	176
7	094	034	084	183	090	047	121	042	124	144	077	112	149	097	136	164	127	058	043
8	125	013	182	109	197	016	184	110	187	066	106	150	021	173	157	181	204	139	145
9	029	064	111	201	005	092	180	048	037	075	055	044	206	010	104	098	172	171	059
10	007	208	138	158	126	068	175	045	174	000	137	053	056	093	153	070	166	032	196
11	089	146	102	154	114	040	151	207	155	140	185	108	117	004	101	113	194	119	123

表 4-3 附件 4 结果顺序表

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
1	191	075	011	154	190	184	002	104	180	064	106	004	149	032	204	065	039	067	147
2	201	148	170	196	198	094	113	164	078	103	091	080	101	026	100	006	017	028	146
3	086	051	107	029	040	158	186	098	024	117	150	005	059	058	092	030	037	046	127
4	019	194	093	141	088	121	126	105	155	114	176	182	151	022	057	202	071	165	082
5	159	139	001	129	63	138	153	053	038	123	120	175	085	050	160	187	097	203	031
6	020	041	108	116	136	073	036	207	135	015	076	043	199	045	173	079	161	179	143
7	208	021	007	049	061	119	033	142	168	062	169	054	192	133	118	189	162	197	112
8	070	084	060	014	068	174	137	195	008	047	172	156	096	023	099	122	090	185	109
9	132	181	095	069	167	163	166	188	111	144	206	003	130	034	013	110	025	027	178
10	171	042	066	205	010	157	074	145	083	134	055	018	056	035	016	009	183	152	044
11	081	077	128	200	131	052	125	140	193	087	089	048	072	012	177	124	000	102	115

其中，附件三为汉字碎纸，较为方正，分类效果好，基本上无须人工干预，且拼接效果较好，需要人工校正次数较少；附件四为英文碎纸，分类效果一般，需要一定次数的人工干预，拼接效果良好，需要人工校正的次数较少。

4.3 三维碎纸复原模型

对于三维碎纸，本文的定义是切割纸片时，不仅进行横和纵的切割，还需区分正面与反面，其形状如图 4-12 所示。

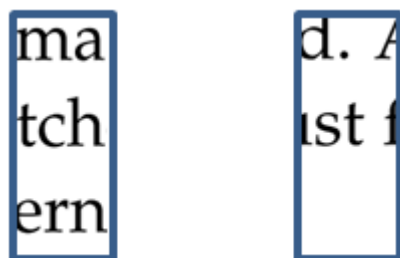


图 4-12 三维碎纸片示意图

与二维碎纸相比，三维碎纸需要区分正反面，算法复杂度更高。

4.3.1 三维模型的降维

假设 3.2.2 假设了正反面打印的页边距等格式相同，基于此假设，正反两面的层次特征是相同的，所以进行分类时，可以分为一类，所以问题三可转换为问题二，只是图片数量增加了一倍。

同样的，利用 4.2.3 的分类方式，可以将问题转化为多个一维碎纸复原问题，并利用一维碎纸复原算法进行复原。

而关于 GUI 程序，也需要进行适当修改，使之能同时对比多张图片（代码见附录四 MultMatch.m），如图 4-13 所示。

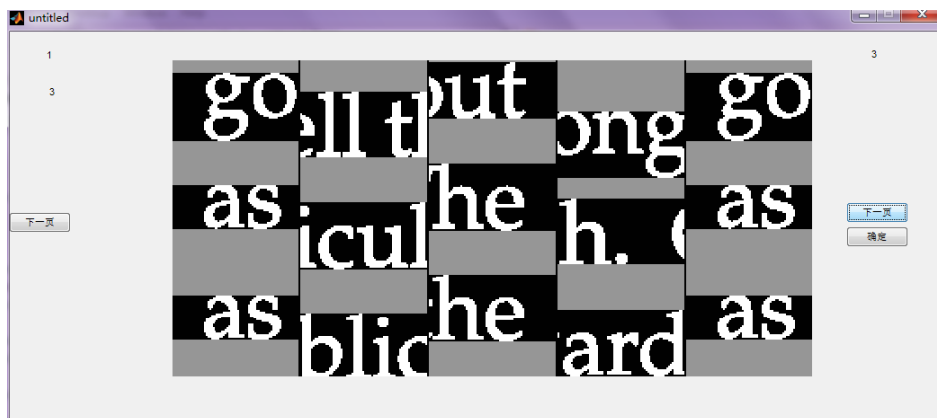


图 4-13 多图片对比 GUI 程序

4.3.2 三维碎纸复原算法

根据 4.2.4 设定的分类方法，可以将二维碎纸复原问题转化为多个一维碎纸复原问题，但是由于文字特征只有原本的一小部分，且空白区域占碎纸大部分比例，导致一维碎纸复原模型复原程度不高，需要对结果进行人工干预，如图 4-14 所示。

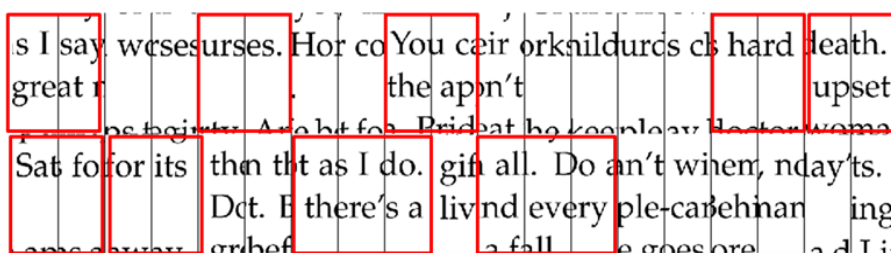


图 4-14 转化后的一维碎纸复原情况

图 4-14 中，红圈区域代表拼接成功的碎纸，由图可看出，一维碎纸复原模型能将大部分碎纸拼接，但仍需人工调整。

综上所述，三维碎纸复原步骤为：

复原步骤：

步骤一：对三维碎纸进行标准化；

步骤二：提取三维碎纸两面的层次特征，并进行合并，转化成二维碎纸复原问题；

步骤三：利用层次特征，进行机器划分和小部分人工干预，实现分类；

步骤四：构造多个一维碎纸问题，并进行求解；

步骤五：根据步骤四的结果，进行适当的人工校正。

4.3.3 模型求解

根据 4.3.2 的算法，对附件 5 的三维碎纸进行复原，得到结果如表 4-4、表 4-5 所示。

表 4-4 附件 5 正面结果图

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	55a	100a	106a	091b	049a	026a	113b	134b	104b	006b	123b	109b	096a	043b	099b

表 4-5 附件 5 反面结果图

078b	111b	125a	140a	155a	150a	183b	174b	110a	066a	108a	18b	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	39b	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	24a	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	08b	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

通过三维碎纸复原模型，能比较有效地实现三维碎纸的复原，附件 5 的复原图片见附录三，由于增加了算法复杂度，所以分类和人工校正的次数相应增加。

五、模型改进与推广

5.1 模型优点

本文采用的碎纸复原模型，综合利用了计算机高速计算能力以及人的文字图像识别和理解能力，拼接效率比纯人工高，拼接准确性也好于纯计算机拼接法。

一维碎纸复原模型与传统的灰值分析相比，考虑了字符大小这一特征，识别速度快，准确度高；

二维碎纸复原模型避免了模拟退火法解的长度过长而导致的算法复杂度快速增加，以分类降维，从而简化问题；在人工干预部分使用 GUI 程序帮助人工识别，比纯粹的人工干预更为高效；

5.2 模型缺点

二维碎纸复原模型和三维碎纸复原模型都需要进行人工干预，虽然提高了准确程度，但是降低了复原的效率；

三维碎纸模型基于正反面打印的页边距等格式相同的假设，所以当正反面打印格式不同时，三维碎纸模型便不适用，所以该模型具有局限性。

5.3 模型改进

5.3.1 适用彩色图片的改进

在 4.1 中，考虑到图片只有黑白两种情况，所以并未指定阈值，而对于彩色图片的处理，需根据具体情况制定阈值，以便更好地二值化，若需提取更加明显的图片，还需进行去噪等处理。

5.3.2 最小干预度算法

在二维碎纸复原算法和三维碎纸复原算法中都需要适用人工干预，人工干预的次数代表了算法的好坏，本文在此提出一个量化人工干预程度的方式，如图 5-1 所示



图 5-1 正确序列与机器识别顺序

由图 5-1 所示，记机器识别得到的顺序为 $T = \{2, 1, 4, 5, 3\}$ ，而正确的顺序式 $G = \{1, 2, 3, 4, 5\}$ ，此处定义序列逆序数为最小人工干预次数，即为得到正确序列，需要做的变换次数，可以算出它们的序列逆序数为 2。

通过序列逆序数，可以量化评价碎纸复原算法与模型的好坏。

5.4 模型推广

模拟退火法是一种通用概率演算法，用来在一个大的搜寻空间内找寻命题的最优解，是解决 TSP 问题的有效算法，也可以较高的效率求解最大截问题、0-1 背包问题、图着色问题、调度问题等等

碎纸复原模型实质是基于边缘像素的吻合度的，这种思想同样适用于音频、视频的拼接。

参考文献

- [1] 张欣, 卜彦龙, 朱良家, 周宗潭 物证复原系统中的碎纸轮廓提取技术研究[J], 11, 23(11): 184, 2006.
- [2] Patrick Butler, Prithwish Chakra borthy, Naren Ramakrishan, The Deshredder: A Visual Analytic Approach to Reconstructing Shredded Documents[J]. Department of Computer Science and Discovery Analytics Center, Virginia Tech, Blacksburg, VA 24061
- [3] 罗智中, 基于文字特征的文档碎纸片半自动拼接[J], 计算机工程与应用, 48(5): 207-210, 2013
- [4] Hei Wang Chan, Evan Gillespie, Delfino Leong, Design and Implementation of a Paper De-shredder. ECE 412 Term Project Report. P2-4 December 6, 2010
- [5] 卓金武, Matlab 在数学建模中的应用[M], 北京: 北京航空航天大学出版社, 2011
- [6] 司守奎, 孙玺菁, 数学建模算法与应用[M], 北京: 国防工业出版社, 2011
- [7] 史峰, 王辉, 郁磊, 胡裴, Matlab 智能算法 30 个案例分析[M], 北京: 北京航空航天大学出版社, 2011
- [8] 杨郑, 基于块匹配和特征点匹配的图象拼接算法研究[D], 成都, 西南交通大学, 2009

附录

附录一 附件 1、附件 2 复原结果

城上层楼叠嶂。城下清淮古汴。举手揖吴云，人与暮天俱远。魂断。魂断。后夜松江月满。簌簌衣巾莎枣花。村里村北响犂车。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。谁念紫损襄王，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。临水纵横回晚靥。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，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。

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-

表 1 附件 1、附件 2 复原结果

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
附件 1	9	15	13	16	4	11	3	17	2	5	6	10	14	19	12	8	18	1	7
附件 2	4	7	3	8	16	19	12	1	6	2	10	14	11	9	13	15	18	17	5

附录二 附件 3、附件 4 复原结果

便邮。温香熟美。醉慢云鬓垂两耳。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。回首向来萧瑟处。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。主人慎小。欲向东风先醉倒。已属君家。且更从容等待他。愿我已无当世望，似君须向古人求。岁寒松柏肯惊秋。

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

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

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

表 2 附件 3 结果顺序表

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
1	049	054	065	143	186	002	057	192	178	118	190	095	011	022	129	028	091	188	141
2	061	019	078	067	069	099	162	096	131	079	063	116	163	072	006	177	020	052	036
3	168	100	076	062	142	030	041	023	147	191	050	179	120	086	195	026	001	087	018
4	038	148	046	161	024	035	081	189	122	103	130	193	088	167	025	008	009	105	074
5	071	156	083	132	200	017	080	033	202	198	015	133	170	205	085	152	165	027	060
6	014	128	003	159	082	199	135	012	073	160	203	169	134	039	031	051	107	115	176
7	094	034	084	183	090	047	121	042	124	144	077	112	149	097	136	164	127	058	043
8	125	013	182	109	197	016	184	110	187	066	106	150	021	173	157	181	204	139	145
9	029	064	111	201	005	092	180	048	037	075	055	044	206	010	104	098	172	171	059
10	007	208	138	158	126	068	175	045	174	000	137	053	056	093	153	070	166	032	196
11	089	146	102	154	114	040	151	207	155	140	185	108	117	004	101	113	194	119	123

表 3 附件 4 结果顺序表

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
1	191	075	011	154	190	184	002	104	180	064	106	004	149	032	204	065	039	067	147
2	201	148	170	196	198	094	113	164	078	103	091	080	101	026	100	006	017	028	146
3	086	051	107	029	040	158	186	098	024	117	150	005	059	058	092	030	037	046	127
4	019	194	093	141	088	121	126	105	155	114	176	182	151	022	057	202	071	165	082
5	159	139	001	129	63	138	153	053	038	123	120	175	085	050	160	187	097	203	031
6	020	041	108	116	136	073	036	207	135	015	076	043	199	045	173	079	161	179	143
7	208	021	007	049	061	119	033	142	168	062	169	054	192	133	118	189	162	197	112
8	070	084	060	014	068	174	137	195	008	047	172	156	096	023	099	122	090	185	109
9	132	181	095	069	167	163	166	188	111	144	206	003	130	034	013	110	025	027	178
10	171	042	066	205	010	157	074	145	083	134	055	018	056	035	016	009	183	152	044
11	081	077	128	200	131	052	125	140	193	087	089	048	072	012	177	124	000	102	115

附录三 附件 5 正反面复原结果

正面

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 any-one 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.

反面

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

表 4 附件 5 正面结果图

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	55a	100a	106a	091b	049a	026a	113b	134b	104b	006b	123b	109b	096a	043b	099b

表 5 附件 5 反面结果图

078b	111b	125a	140a	155a	150a	183b	174b	110a	066a	108a	18b	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	39b	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	24a	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	08b	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

附录四 程序代码

模拟退火法代码

fitness.m

```
fitness.m
function y=fitness(a,b)
n=size(a,1);
for i=1:n
    if a(i)==0 && b(i)==0
        t(i)=0;
        continue;
    end
    t(i)=a(i)+b(i)-40;
    t(i)=t(i)*t(i);
end
y=sum(t);
end
```

getLandR.m

```
function [l r]=getLandR(p000)
n=size(p000,1);
m=size(p000,2);
for i=1:n
    for j=1:m
        if p000(i,j)==255
            p000(i,j)=1;
        else
            p000(i,j)=255;
        end
    end
end
for i=1:n
    if sum(p000(i,:))==m
        for j=1:m
            p000(i,j)=150;
        end
    end
end
for i=1:n
    if p000(i,1)==150
        temp(i)=1;
    else
        temp(i)=0;
    end
end
end
```

```

j=1;
for i=1:n-1
    if (temp(i)==1 && temp(i+1)==0) || (temp(i)==0 && temp(i+1)==1)
        record(j)=i+1;
        j=j+1;
        temp2(i)=1;
    else
        temp2(i)=0;
    end
end
%设置字距
nn=sum(temp2);
for i=1:2:nn
    for j=1:m
        if sum(p000(record(i):record(i+1)-1,j))==record(i+1)-record(i);
            for k=record(i):record(i+1)-1
                p000(k,j)=150;
            end
        end
    end
end
%获取左边数据
for i=1:n
    if p000(i,1)==150
        l(i)=0;
        continue;
    end
    for j=1:m-1
        if p000(i,j+1)==150
            l(i)=j;
            break;
        end
    end
end
l=l';
%获取右边数据
for i=1:n
    if p000(i,m)==150
        r(i)=0;
        continue;
    end
    for j=m:-1:2
        if p000(i,j-1)==150
            r(i)=m-j+1;
        end
    end
end

```

```

                break;
            end
        end
    end
end
r=r';
end

```

Main.m

```

clear,clc
p000=imread('001.bmp');
%批量导入图片
for j=1:19
    i=j-1;
    %生成图片名
    a=floor(i/100);
    b=mod(floor(i/10),10);
    c=mod(i,10);
    t=strcat(num2str(a),num2str(b));
    t=strcat(t,num2str(c));
    t=strcat(t,'.bmp');
    %导入图片
    p000=imread(t);
    %计算左右边界
    [l(:,j),r(:,j)]=getLandR(p000);
    clear p000;
end
for i=1:19
    for j=1:19
        dis(i,j)=fitness(r(:,i),l(:,j));
    end
end
%模拟退火法
x=randperm(19);%产生初始解
a = 0.98;%衰减系数
t0=97;%初始温度
tf = 3;%停止温度
t = t0;%初始化温度
Markov_length = 10000;%马尔可夫链长度
amount = size(x,2);%解的长度
sol_new = x;%产生新解
E_current = inf;%当前解的值
E_best = inf;%最佳解的值
sol_currrent = sol_new;%当前解等于新产生的解
sol_best = sol_new;%

```

```

while t >=tf
    for rr=1:Markov_length
        if(rand<0.5)
            ind1 = 0; ind2 = 0;
            while(ind1 == ind2)
                ind1 = ceil(rand.*amount);
                ind2 = ceil(rand.*amount);
            end
            tmp1 = sol_new(ind1);
            sol_new(ind1) = sol_new(ind2);
            sol_new(ind2) = tmp1;
        else
            ind1 = 0; ind2 = 0; ind3 = 0;
            while(ind1 == ind2) || (ind1 == ind3)...
                || (ind2 == ind3) || (abs(ind1-ind2) == 1)
                    ind1 = ceil(rand.*amount);
                    ind2 = ceil(rand.*amount);
                    ind3 = ceil(rand.*amount);
            end
            tmp1 = ind1; tmp2 = ind2; tmp3 = ind3;
            if(ind1<ind2)&&(ind2<ind3);
            elseif(ind1<ind3)&&(ind3<ind2)
                ind2 = tmp3; ind3 = tmp2;
            elseif(ind2<ind1)&&(ind1<ind3)
                ind1 = tmp2; ind2 = tmp1;
            elseif(ind2<ind3)&&(ind3<ind1)
                ind1 = tmp2; ind2 = tmp3; ind3 = tmp1;
            elseif(ind3<ind1)&&(ind1<ind2)
                ind1 = tmp3; ind2 = tmp1; ind3 = tmp2;
            elseif(ind3<ind2)&&(ind2<ind1)
                ind1 = tmp3; ind2 = tmp2; ind3 = tmp1;
            end
            tmp1list1 = sol_new((ind1+1):(ind2-1));
            sol_new((ind1+1):(ind1+ind3-ind2+1))=...
            sol_new((ind2):(ind3));
            sol_new((ind1+ind3-ind2+2):ind3)= ...
            tmp1list1;
        end
        summ=0;
        for i=1:amount-1
            sum=sum+dis(sol_new(i), sol_new(i+1));
        end
        E_new=summ;
        if E_new<E_current

```

```

        E_current = E_new;
        sol_current = sol_new;
        if E_new < E_best
            E_best = E_new;
            sol_best = sol_new;
        end
    else
        if rand < exp(-(E_new - E_current)./t)
            E_current = E_new;
            sol_current = sol_new;
        else
            sol_new = sol_current;
        end
    end
end
end
E_best
t = t.*a;
end
for i=1:19
    if sum(l(:, i)) == 0
        first = i;
        break;
    end
end
for i=1:19
    if sol_best(i) == first
        first = i;
        break;
    end
end
sol_best = [sol_best(first:end), sol_best(1:first-1)];
temp = [];
for j=1:19
    i = sol_best(j) - 1;
    %生成图片名
    a = floor(i/100);
    b = mod(floor(i/10), 10);
    c = mod(i, 10);
    t = strcat(num2str(a), num2str(b));
    t = strcat(t, num2str(c));
    t = strcat(t, '.bmp');
    %导入图片
    p000 = imread(t);
    %计算左右边界

```

```

    temp=[temp p000];
    clear p000;
end
imshow(temp);

```

GUI 程序代码

Init.m

```

clear,clc
global start;
global have;
global num;
have(1:209)=0;
start=1;
num=1;
global istart;
istart=1;
global result;
result(1:209)=-1;

```

SingleMatch.m

```

function varargout = untitled(varargin)
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',   gui_Singleton, ...
                  'gui_OpeningFcn', @untitled_OpeningFcn, ...
                  'gui_OutputFcn',  @untitled_OutputFcn, ...
                  'gui_LayoutFcn',  [] , ...
                  'gui_Callback',    []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end

function untitled_OpeningFcn(hObject, eventdata, handles, varargin)
handles.output = hObject;
guidata(hObject, handles);

function varargout = untitled_OutputFcn(hObject, eventdata, handles)
varargout{1} = handles.output;

```



```

function pushbutton1_Callback(h0bject, eventdata, handles)
clc;
global have;
global p000;
global istart;
global e;
if istart>209
    istart=mod(istart, 209);
end
for i=istart:size(have, 2)
    if have(i)==0
        e=i;
        set(handles.text1, 'String', num2str(i-1));
        break;
    end
end
istart=i+1;
e=e-1;
a=floor(e/100);
b=mod(floor(e/10), 10);
c=mod(e, 10);
t=strcat(num2str(a), num2str(b));
t=strcat(t, num2str(c));
t=strcat(t, '. bmp');
%导入图片
p001=imread(t);
temp=[p000 p001];
imshow(temp);

```

```

function pushbutton2_Callback(h0bject, eventdata, handles)
global have;
global e;
global result;
global start;
have(e+1)=1;
result(e+1)=start+1;

```

```

function pushbutton3_Callback(h0bject, eventdata, handles)
clc;
global start;
global have;
global result;
global p000;

```

```

for i=1:size(have, 2)
    if have(i)==0
        start=i;
        have(i)=1;
        set(handles.text2, 'String', num2str(i-1));
        break;
    end
end

end

start=start-1;
result(start+1)=start+1;
a=floor(start/100);
b=mod(floor(start/10), 10);
c=mod(start, 10);
t=strcat(num2str(a), num2str(b));
t=strcat(t, num2str(c));
t=strcat(t, '. bmp');
%导入图片
p000=imread(t);
imshow(p000);

```

MultipMatch.m

```

function varargout = untitled(varargin)
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',   gui_Singleton, ...
                  'gui_OpeningFcn', @untitled_OpeningFcn, ...
                  'gui_OutputFcn',  @untitled_OutputFcn, ...
                  'gui_LayoutFcn',   [] , ...
                  'gui_Callback',    []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end
if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end

function untitled_OpeningFcn(hObject, eventdata, handles, varargin)
handles.output = hObject;
guidata(hObject, handles);

```

```

function varargout = untitled_OutputFcn(hObject, eventdata, handles)
varargout{1} = handles.output;

function pushbutton1_Callback(hObject, eventdata, handles)
clc;
global have;
global p000;
global q000;
global istart;
global e;
if istart>209
    istart=mod(istart,209);
end
for i=istart:size(have,2)
    if have(i)==0
        e=i;
        set(handles.text1,'String',num2str(i-1));
        break;
    end
end
    istart=i+1;
    e=e-1;
    a=floor(e/100);
    b=mod(floor(e/10),10);
    c=mod(e,10);
    t=strcat(num2str(a),num2str(b));
    t=strcat(t,num2str(c));
    t1=strcat(t,'a.bmp');
    %导入图片
    t2=strcat(t,'b.bmp');
    p001=imread(t1);
    q001=imread(t2);
n=size(p001,1);
m=size(q001,2);
for i=1:n
    for j=1:m
        if p001(i,j)==255
            p001(i,j)=1;
        else
            p001(i,j)=255;
        end
    end
end
end
for i=1:n

```

```

        if sum(p001(i, :))==m
            for j=1:m
                p001(i, j)=150;
            end
        end
    end
end
for i=1:n
    for j=1:m
        if q001(i, j)==255
            q001(i, j)=1;
        else
            q001(i, j)=255;
        end
    end
end
end
for i=1:n
    if sum(q001(i, :))==m
        for j=1:m
            q001(i, j)=150;
        end
    end
end
end
    line(1:180)=1;
    line=line';
    temp=[q001, line, p000, line, p001, line, q000, line, q001];
    imshow(temp);

function pushbutton2_Callback(hObject, eventdata, handles)
global have;
global e;
global result;
global start;
have(e+1)=1;
result(e+1)=start+1;
set(handles.text6, 'String', num2str(sum(have)));

function pushbutton4_Callback(hObject, eventdata, handles)
clc;
global start;
global have;
global result;
global p000;
global q000;
for i=1:size(have, 2)

```

```

        if have(i)==0
            start=i;
            have(i)=1;
            set(handles.text2,'String',num2str(i-1));
            break;
        end
    end
end
start=start-1;
result(start+1)=start+1;
a=floor(start/100);
b=mod(floor(start/10),10);
c=mod(start,10);
t=strcat(num2str(a),num2str(b));
t=strcat(t,num2str(c));
t1=strcat(t,'a.bmp');
t2=strcat(t,'b.bmp');
%导入图片
p000=imread(t1);
q000=imread(t2);
n=size(p000,1);
m=size(q000,2);
for i=1:n
    for j=1:m
        if p000(i,j)==255
            p000(i,j)=1;
        else
            p000(i,j)=255;
        end
    end
end
for i=1:n
    if sum(p000(i,:))==m
        for j=1:m
            p000(i,j)=150;
        end
    end
end
for i=1:n
    for j=1:m
        if q000(i,j)==255
            q000(i,j)=1;
        else
            q000(i,j)=255;
        end
    end
end

```

```

        end
    end
    for i=1:n
        if sum(q000(i,:))==m
            for j=1:m
                q000(i,j)=150;
            end
        end
    end
    end
    ttt=[p000,q000];
    imshow(ttt);

```

层次特征提取程序

chuli.m

```

for j=1:209
    i=j-1;
    %生成图片名
    a=floor(i/100);
    b=mod(floor(i/10),10);
    c=mod(i,10);
    t=strcat(num2str(a),num2str(b));
    t=strcat(t,num2str(c));
    t=strcat(t,'.bmp');
    %导入图片
    p000=imread(t);
    %计算左右边界
    n=size(p000,1);
    m=size(p000,2);
    for ii=1:n
        for jj=1:m
            if p000(ii,jj)==255
                p000(ii,jj)=1;
            else
                p000(ii,jj)=255;
            end
        end
    end
    end
    for ii=1:n
        if sum(p000(ii,:))==m
            for jj=1:m
                p000(ii,jj)=150;
            end
        else
            for jj=1:m

```

```

                p000(i i, j j)=1;
            end
        end
    end
    str=['a/n', num2str(i), '. bmp'];
    imwrite(p000, str, 'bmp');
    clear p000;
end

```

calc.m

```

clear, clc
map=[];
for j=1:209
    t=['n', num2str(j-1), '. bmp'];
    %导入图片
    p000=imread(t);
    k=1;
    %计算左右边界
    flag=1;
    for i=1:180
        if p000(i)==flag
            if flag==150
                map(j, k)=i-1;
            else
                map(j, k)=i;
            end
            k=k+1;
            if flag==1
                flag=150;
            else
                flag=1;
            end
        end
    end
    if flag==150
        map(j, k)=180;
    end
    clear p000;
end

```