

2013 高教社杯全国大学生数学建模竞赛

承 诺 书

我们仔细阅读了中国大学生数学建模竞赛的竞赛规则。

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

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

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

我们参赛选择的题号是（从 A/B/C/D 中选择一项填写）：_____

我们的参赛报名号为（如果赛区设置报名号的话）：_____

所属学校（请填写完整的全名）：_____ 中国人民大学

参赛队员（打印并签名）：1. _____ 孙嘉轩

2. _____ 张如琪

3. _____ 卢山

指导教师或指导教师组负责人（打印并签名）：_____ 高金伍

日期：_____年____月____日

赛区评阅编号（由赛区组委会评阅前进行编号）：



2009 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）：

规则边缘碎纸拼接的数学模型与算法实现

摘要

本文针对问题一到问题三中不同的碎纸片所包含的信息量的大小, 不断完善的算法和程序, 对全体碎纸片进行定量分析并推导出碎纸片之间的关系。

对于问题一中的两个附件, 我们首先把所有的碎片数字化为灰度矩阵, 提取出各个碎片灰度矩阵的最左和最右侧一个像素宽度的灰度向量, 利用余弦定理计算出与一个特定碎片最左侧灰度向量在 n 维欧式空间夹角最小的某一右侧灰度向量, 记录该右侧灰度向量所属的碎片与被比对碎片, 重复这一过程, 即可得到所有碎片的应有顺序。

对于问题二中的两个附件, 观察到碎片变得更加细小, 每个碎片中所包含的信息更少, 直接利用第一问的做法不够准确。因此, 我们采用蚁群优化算法对第一问的程序进行扩充和优化, 在程序把小碎片拼成大块片段后和程序无法确定正确的序列时使用人工干预, 最终得到正确的顺序矩阵和拼图。

对于问题三中的双面附件, 虽然提高了复杂性, 但是同时也提供了更多的像素信息以进行程序处理, 在以第二问的程序为主体的基础上, 我们将某碎片 a 面左侧 (右侧) 的灰度矩阵与该碎片 b 面右侧 (左侧) 的灰度矩阵上下相连, 构造出一个长度为 360 像素的灰度向量, 以该灰度矩阵为基础进行配对和记录。与通过余弦值加权平均进行配对的结果进行比较, 必要时加入人工干预。最终得到正确的顺序矩阵和拼图。

关键词

余弦定理 n 维欧式空间 灰度矩阵 蚁群优化算法

一、问题重述

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

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

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

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

二、问题分析

问题 1、为了达到系统自动拼接文档碎片的目的是，首先应该对这些文档碎片照片进行数字化模拟。[1]具体在此问题中，首先我们要把每一个碎片图片中的不同的点根据不同的灰度赋予 0-255 中不同的值，从而得到与碎片图片相对应的灰度矩阵。根据常识，也就是我们凭借肉眼手动拼接碎片所依靠的判断标准，我们知道把一张印有大量文字的纸从中间任何一个地方切开以后，在这个切口两端距离切口非常近的地方应该是相似的。也就是说在距离切口足够近的地方，切口两端灰度数字组成的列向量应该是非常相似的，即同一断口左侧和右侧的像素列向量的余弦值[2]应接近于 1。根据这一原理，我们可以依次计算出每个碎片的左侧与其他碎片右侧的相似程度，其两向量之间余弦值最大的两个向量所归属的碎片应为相邻的碎片。[3]

问题 2、与问题 1 类似，加入蚁群优化算法[4]模式以解决每一碎片中信息量过少的问题。

问题 3、在问题 2 程序的基础上，利用双面碎片的信息优化配对方法，并对两种配对方法的结果进行比较得到最终结论。

三、符号说明

$leftcolumn_i$: 第 i 张图片最左侧像素的灰度向量

$rightcolumn_i$: 第 i 张图片最右侧像素的灰度向量

x_{ik} : 第 i 张图片最左侧像素的灰度向量中第 k 个元素的灰度值

y_{ik} : 第 i 张图片最右侧像素的灰度向量中第 k 个元素的灰度值

θ_{ij} : $leftcolumn_i$ 和 $rightcolumn_j$ 在 n 维欧式空间中的夹角

四、模型假设

- (1) 所有碎片的像素矩阵都是可知的。
- (2) 所有碎片均应被用于拼接成完整图形。
- (3) 所有碎片均为水平或垂直（可以两者均包含）切割。

五、模型的建立与求解

5.1 模型一

5.1.1 模型一的建立

模型一建立的依据是 n 维欧式空间中两个向量夹角余弦值的知识。

根据每一个图像碎片的照片得到该图片对应的灰度矩阵。取第 i 个图片对应的灰度矩阵最左端的一列记为 $leftcolumn_i$ ，最右端一列记为 $rightcolumn_i$

$$leftcolumn_i = \begin{bmatrix} x_{i1} \\ x_{i2} \\ x_{i3} \\ x_{i4} \\ \vdots \\ \vdots \\ \vdots \\ x_{in-1} \\ x_{in} \end{bmatrix} \qquad rightcolumn_i = \begin{bmatrix} y_{i1} \\ y_{i2} \\ y_{i3} \\ y_{i4} \\ \vdots \\ \vdots \\ \vdots \\ y_{in-1} \\ y_{in} \end{bmatrix}$$

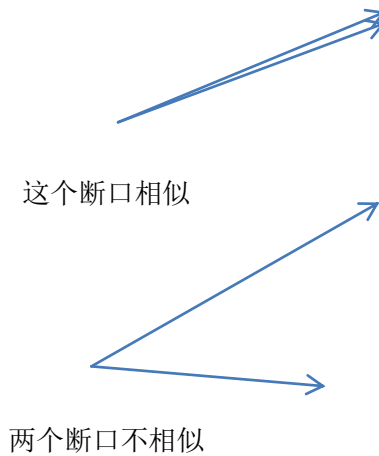
其中， x_{in} 表示第 i 个图片最左边一列第 n 个像素点的灰度。

记 $rightcolumn_i$ 和 $leftcolumn_j$ 之间的夹角为 θ_{ij}

$$\begin{aligned} \text{则目标函数为 } \max(\text{COS } \theta_{ij} &= \frac{\langle \text{rightcolumn}_i^T * \text{leftcolumn}_j \rangle}{|\text{rightcolumn}_i^T| |\text{leftcolumn}_j|}) \\ &= \max(\text{COS } \theta_{ij} = \frac{\sum_k^n x_{ik} y_{jk}}{\sqrt{\sum_k^n x_{ik}^2 * \sum_k^n y_{jk}^2}}) \end{aligned}$$

上面已经解释，该算法计算 rightcolumn_i 与 leftcolumn_j ($i \neq j$)的夹角余弦值，记录得到最大值时所对应的 j ，这就是第 i 个碎片右边应该对应的图片的编号。

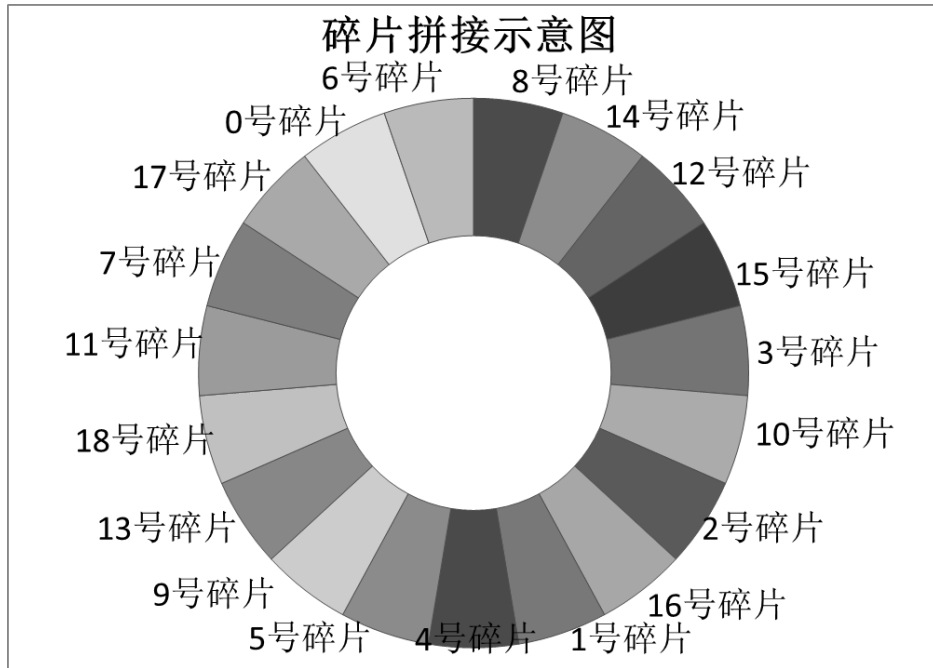
具体到问题中来说可以用几何的形式表示如下：



5.1.2 附件一的求解

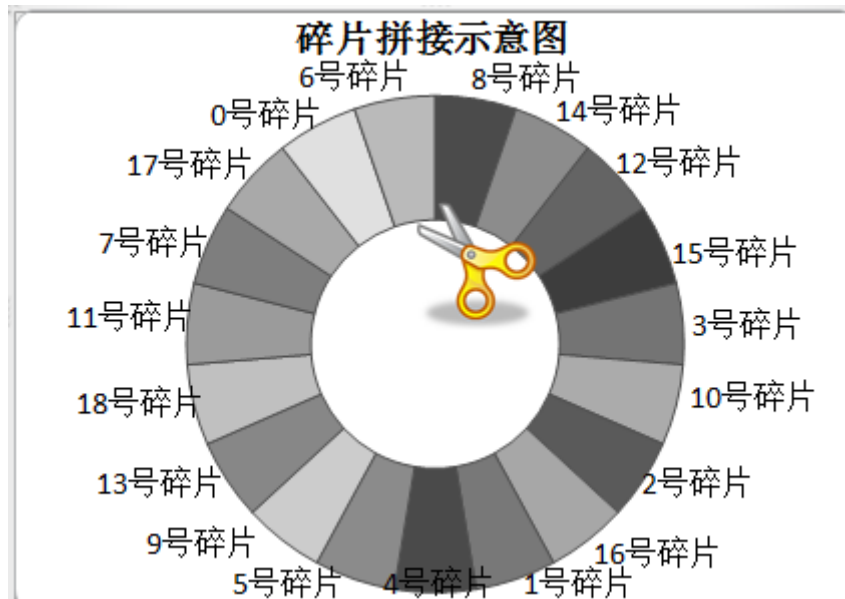
依照这种方法可以依次得到应与所有图片右端相连的图片，而且这样的关系应该是一一对应的。由于最左边与最右边的空白边的灰度矩阵完全相同，所以算法自动得出的结论应该是一个环状的图片，最左端与最右端的空白边也连在一起。

如下图所示



检测到 6 号与 8 号之间是纯白色拼接在一起的，所以应在此处展开，得到完整的文章。

如下图所示



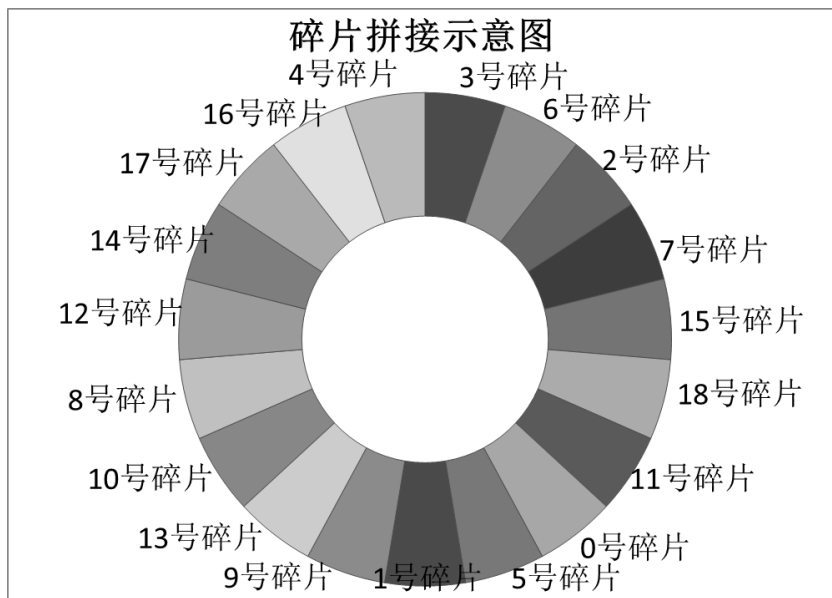
最后得到的结果应该是

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

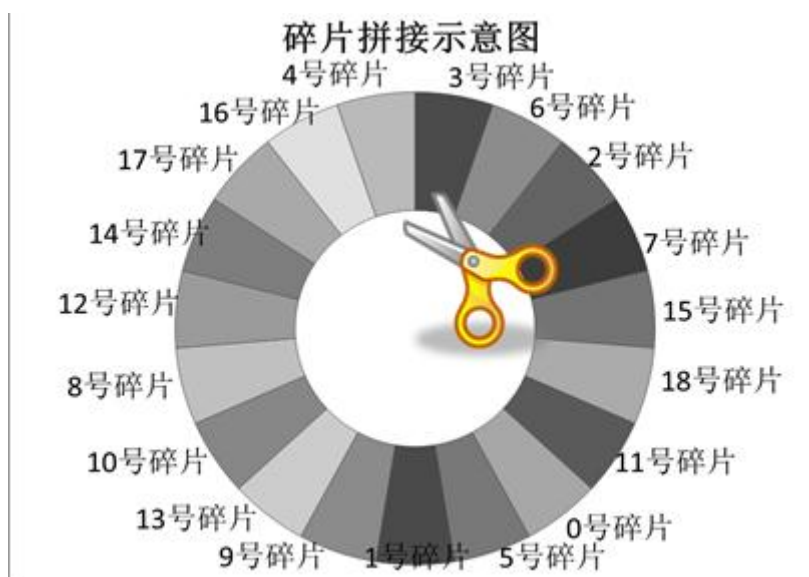
拼合后的具体图形见附录 1

5.1.2 附件二的求解

附件二是英文碎片的拼接，与附件一的拼接类似，得到的初步结论如下图所示所示：



检测到 3 号碎片与 4 号碎片之间是纯白色的拼接，所以应从此处展开。



最后得到的结果应该是

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

具体拼接后的结果见附录。

5.2 模型二

5.2.1 模型二思路的分析

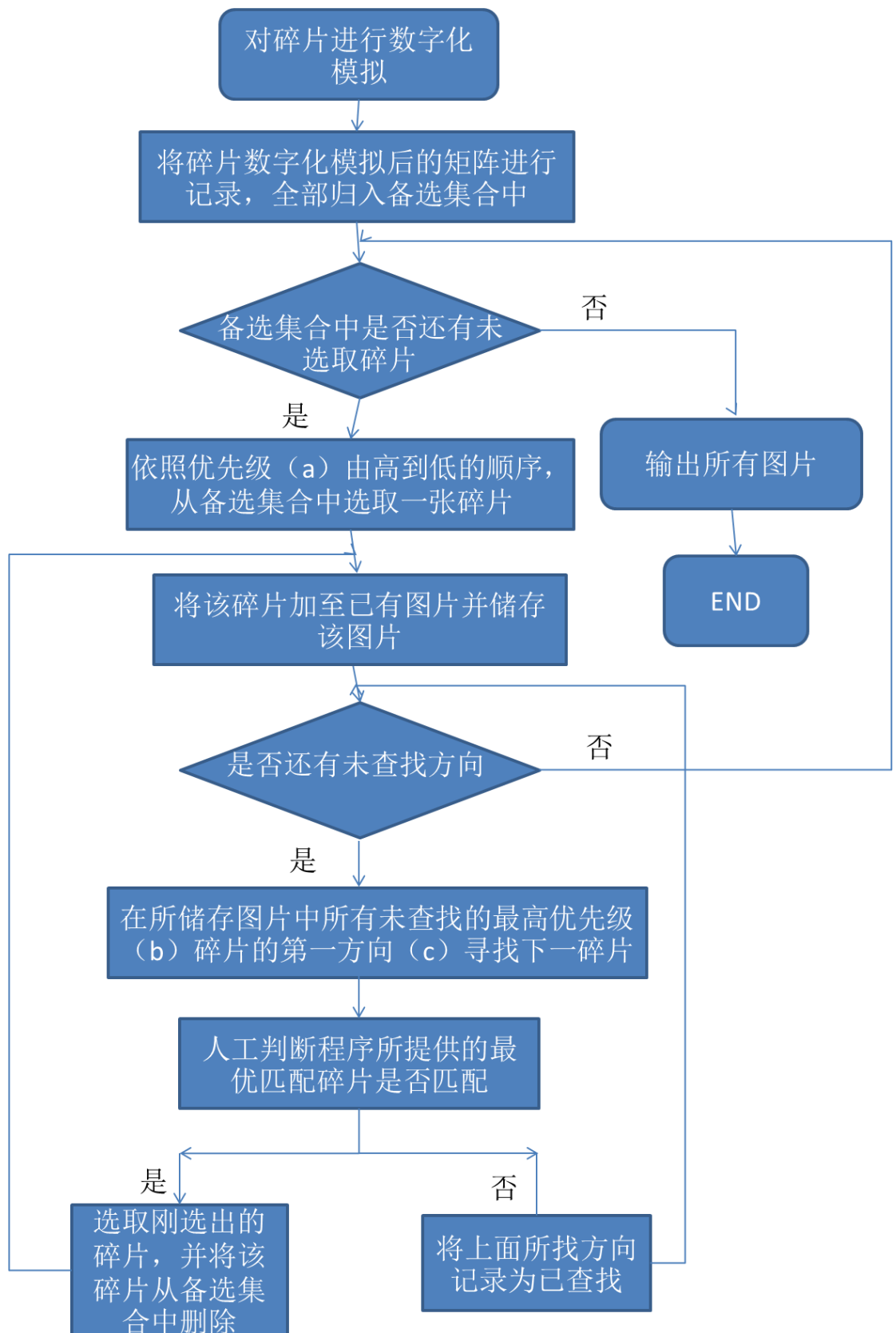
模型二比模型一复杂的地方在于切割更为细致，每一个碎片所包含的信息量远小于问题 1 中碎片所包含的信息量。为了处理这一问题，我们在进行配对拼接时要加上人工处理以便判断是否匹配的步骤。

模型二的数学基础与模型一类似，都是基于上文提到的假设，根据 n 维欧式空间中的向量夹角的余弦定理来进行判断。不同的是，问题一中碎片空白边较少和碎片信息量较大都比较便于处理而问题二中的附件则更为复杂一些。此外，碎片含有白色边的部分是最难进行拼接的，因为有白色边可以与任何其他的白色边进行配对，难以确定该白色边的具体位置，因此在本问题中我们碰到白色边采取的策略是跳过进行下一步，希望通过其他方向确定该碎片的具体位置。

通过上述分析我们了解了问题二与问题一的区别及其难点所在，在问题二中我们采用蚁群优化算法[5]来从某一个碎片出发，逐步得到较大片段的排列，从这个排列中我们可以得到更多的信息进而增加下一步排列时的准确性，并从这些较大的片段出发逐步得到整体的排序。

5.2.1 模型二算法的建立

把上述思想表述为算法的形式即如下图所示。



在这个算法中所用到的相关概念如下。

- 1、信息量：根据数字化模拟得到的所有碎片的灰度矩阵，可以得到这些矩阵

最左、最右、最上、最下四个方向上的边界向量，每一个碎片的四个边界向量中的灰度元素不全为 255（即边界不为纯白）的个数记为该碎片的信息量，信息量的取值范围是 0, 1, 2, 3, 4.

2、优先级 (a)：把含有 i 个信息量的碎片全部归入第 i 组，优先级顺序：第 4 组 > 第 3 组 > 第 2 组 > 第 1 组 > 第 0 组。在相同的一组中，碎片编号越大的图片优先级越高。

3、优先级 (b)：在现有碎片片段中，水平方向延伸的时候上部碎片的优先级大于下部碎片，垂直方向延伸时右边碎片大于左边碎片。

4、第一方向：在本算法中，优先寻找优先级 (b) 最大的碎片的右侧，找不到或已存在时寻找该碎片的左侧，以此方式继续寻找该碎片的上方和下方。

5.2.3 附件 3 的求解

根据上面算法所阐述的方法，我们应用 matlab 得到附件 3 的结果与拼接完成的图片见附录 1。

5.2.3 附件 4 的求解

采用与附件 3 相同的方法，应用 matlab 得到附件 4 的拼接结果与拼接完成的图片见附录 1。

5.2.4 人工干预

在本程序运行过程中，当出现切后断口为空白时需要进行人工拼接。因为此时空白断口可能与其他任何地方的空白断口相结合，出现多种拼接方法共存的问题。

此时人工干预的方法是空过这一次拼接，寻找其他方向继续进行延伸，当所有方向都延伸过时，选择其他碎片进行下一步的拼接和组合。

5.3 模型三

5.3.1 模型三思路的分析

模型三与模型二相比，区别为增加为正反面而且正反面与 a, b 的对应关系是未知的。这个区别的难度在于增加了应比对的碎片的个数，而优势在于增加了可以利用的信息量，加以办法进行利用可以提高系统自动判断的成功率。

5.3.1 模型三算法的建立

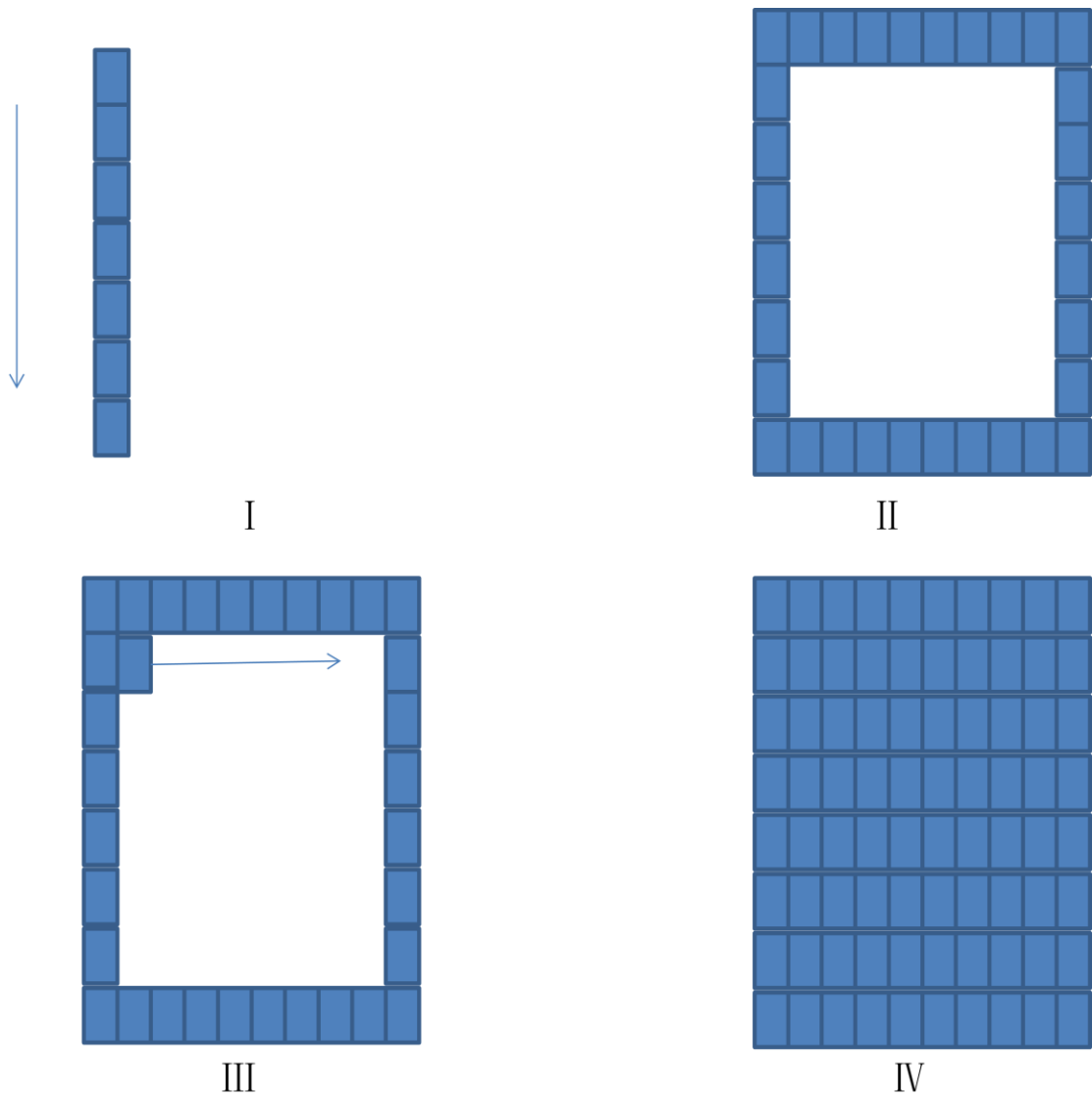
首先找出所有 $a(b)$ 面左边缘空白和 $b(a)$ 面右边缘空白的碎片，再依据 $a(b)$ 面左边缘和 $b(a)$ 面右边缘空白宽度，我们可以在碎片中确定所有位于左边界和右边界的碎片，再依据上边缘空白宽度，选取其中位于某面左上角的碎片，用算法一找到与其最匹配的下相邻碎片，人工判断其是否正确，若不正确则选取次优解碎片，直至找到匹配碎片。以此类推，完成左边界一系列的拼接。(图 I)

同样的，我们可以完成最右一列、最上一行、最下一行的拼接。(图 II)

依据第二行第一个碎片，和第二列第一个碎片的正反两面共四个边界信息，来查找出最匹配的第二行第二个碎片。以第二行第二个碎片为起点，用算法一依次查找碎片的右相邻(每次查找均有四个边界信息可用)，将第二行填满。(图 III)

同理，依次填补后面几行，将整张图片复原。(图 IV)

在图中间进行向右延伸的时候，为了利用正反两面带来的信息，我们将某碎片 a 面左侧(右侧)的灰度矩阵与该碎片 b 面右侧(左侧)的灰度矩阵上下相连，构造出一个长度为 360 像素的灰度向量，将此灰度向量与依照此模式形成的所有长度为 360 像素的灰度向量相比较，得到最为接近的碎片并记录。将此对配对碎片断口处的两组相对应向量余弦值计算出来并相加，看按照此方法得到的排序与上面方法得到的排序是否一致，若一致，则直接输出，若不一致，则进行人工判断。



6、模型的优缺点和适用性分析

6.1 模型的准确性分析

根据上述算法得到的结果和最后加上人工干预后拼得结果的比较，我们可以得到应用上述算法所得结果的准确性。

对于模型一来说，由于整个碎片中信息较多，我们在对于附件 1 和附件 2 的处理上都一次得到了正确的结论。横向的正确率为 100%。

对于模型二来说，整个文章碎片化的程度较高，在某一区域内可用的信息并不足够，在得到正确的拼接方式以后，我们计算附件 3 和 4 用这种算法拼接的成功率。

对于附件 3，分别计算四个方向上的成功率和横断面纵断面的成功率如下：

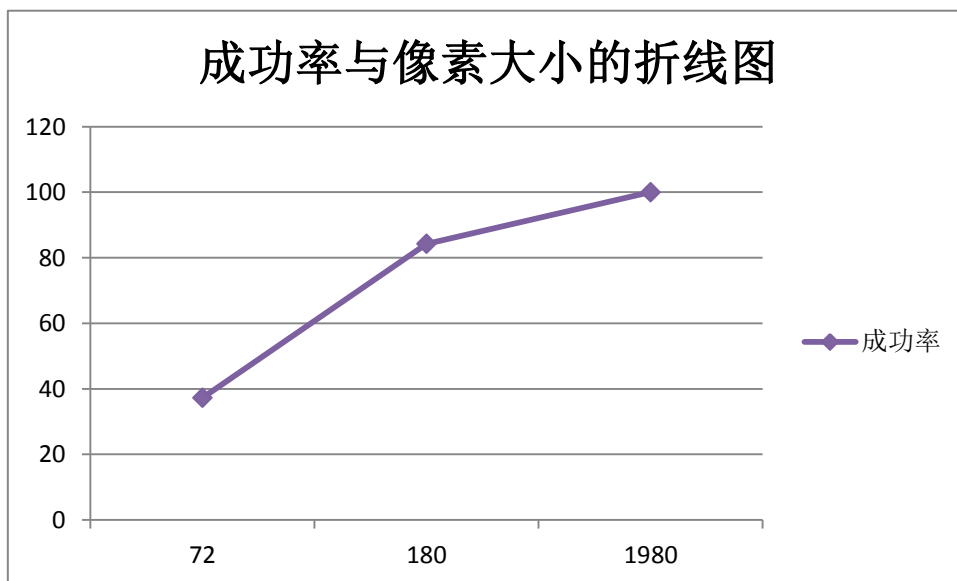
从左到右成功率	0.853535354
从右到左成功率	0.883838384
从上到下成功率	0.431578947
从下到上成功率	0.405263158
横向 180 像素成功率	0.868686869
纵向 72 像素成功率	0.418421053

相似的，对于附件 4，分别计算这几个方向上的成功率和横断面纵断面的成功率如下：

从左到右成功率	0.833333333
从右到左成功率	0.797979798
从上到下成功率	0.226315789
从下到上成功率	0.426315789
横向 180 像素的成功率	0.815656566
纵向 72 像素的成功率	0.326315789

直观上我们可以看出，横向推断的成功率较高而纵向推断的成功率较低，符合我们的直观判断。

为了更加准确地判断出像素与成功率之间关系，我们利用上面得出的数据画出图形来表述这种相关关系。



我们可以在上述图表中得到如下结论：

- 1、 像素越高，利用余弦定理拼接的成功率就越高。
- 2、 随着像素的增加，刚开始成功率增长较快，后来成功率增长越来越慢。当像素大于一定的数据时，成功率可以达到 95%以上，拼接方法成功率较高。

6.2 模型的适用性分析

为了使用上述几个模型，应尽量提高碎片图片的分辨率以提高系统分析的成功率。

7、参考文献

- [1] 吴军. 数学之美. 北京: 人民邮电出版社,2012
- [2] Wikipedia. Laws of cosines, http://en.wikipedia.org/wiki/Law_of_cosines, 2013 年 9 月 16 日
- [3] 王轩. 碎片拼接. 杭州: 浙江大学, 2012
- [4] 杨剑峰. 蚁群算法及其应用研究. 杭州: 浙江大学, 2007
- [5] 温文波, 杜维. 蚁群算法概述. 石油化工自动化, 2002, 1(19): 19-22.

附录 1、

第一部分：表格形式

(1) 问题 1 碎片拼接 1×19 表格

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

问题 2 碎片拼接 1×19 表格

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

(2) 问题 3 碎片拼接 11×19 表格

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
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	73
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
71	156	83	132	200	17	80	33	202	198	15	133	170	205	85	152	165	27	60
89	146	102	154	114	40	151	207	155	140	185	108	117	101	4	113	194	119	123

问题 4 碎片拼接 11×19 表格

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

(3) 问题 5 碎片拼接 2×11×19 表格

正面

136a	47b	20b	164a	81a	189a	29b	18a	108b	66b	110b	174a	183a	150b
5b	152b	147b	60a	59b	14b	79b	144b	120a	22b	124a	192b	25a	44b
143a	200a	86a	187a	131a	56a	138b	45b	137a	61a	94a	98b	121b	38b
83b	39a	97b	175b	72a	93b	132a	87b	198a	181a	34b	156b	206a	173a
90b	203a	162a	2b	139a	70a	41b	170a	151a	1a	166a	115a	65a	191b
13b	24b	57b	142b	208b	64a	102a	17a	12b	28a	154a	197b	158b	58b

35b	159b	73a	193a	163b	130b	21a	202b	53a	177a	16a	19a	92a	190a
172b	122b	182a	40b	127b	188b	68a	8a	117a	167b	75a	63a	67b	46b
105b	204a	141b	135a	27b	80a	0a	185b	176b	126a	74a	32b	69b	4b
9a	145b	82a	205b	15a	101b	118a	129a	62b	52b	71a	33a	119b	160a
54a	196a	112b	103b	55a	100a	106a	91b	49a	26a	113b	134b	104b	6b

155b	140b	125b	111a	78a
178b	76a	36b	10a	89b
30b	42a	84a	153b	186a
194a	169a	161b	11a	199a
37a	180b	149a	107b	88a
207b	116a	179a	184a	114b
50b	201b	31b	171a	146b
168b	157b	128b	195b	165a
77b	148a	85a	7a	3a
95b	51a	48b	133b	23a
123b	109b	96a	43b	99b

背面

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

81b	164b	20a	47a	136b
59a	60b	147a	152a	5a
131b	187b	86b	200b	143b
72b	175a	97a	39b	83a
139b	2a	162b	203b	90a
208a	142a	57a	24a	13a
163a	193b	73b	159a	35a
127a	40a	182b	122a	172a
27a	135b	141a	204b	105a
15b	205a	82b	145a	9b
55b	103a	112a	196b	54b

第二部分：图片形式

(1) 附件 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。村里村北响辘车。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。汤发云腴酳白，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 2

fair of face.

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

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

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

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

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

(2) 附件 3

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。

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。

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。

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 late, exactly is the earliest time. Less is more.

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

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

(3) 附件 5
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 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.

背面

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

附录 2

附录 2

源程序列表

序号	文件名	对应附件编号	作用
1	mcm2013b_p1.m	1, 2	得出第一问正确排序
2	mcm2013b_p2.m	3, 4	存储第二问碎片信息 末尾注释中含有了人工 干预阶段四个方向的子 程序
3	mcm2013b_p2stats.m	3, 4	统计横向纵向正确率
4	veccos.m	1 ~ 5	计算夹角余弦值函数
5	mcm2013b_p3.m	5	存储第三问碎片信息， 在人工干预条件下得出 正确排序

序号 1 源程序

```
function [idx_sequence] = mcm2013b_p1()
% 本程序适用于/cumcm2013problems/B 中的 1, 2 文件夹碎片文件

% 程序调用查看方法
% 1.修改 directory 变量为测试用机碎片文件所在文件夹的绝对路径
% 2.调用 mcm2013b_p1(), 命令行显示从左到右正确的碎片排序
% 3.MATLAB 弹出生成原图片的缩微图(清晰度较低, 可放大查看)
% 4.查看 directory 文件夹中生成的 result.bmp 图片

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%% 请修改文件夹路径, 请将‘附件 1 (2)’文件夹的‘附件’两字去掉, 保证绝对路径
全英文
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
directory = '/Users/oh_baizhima/Desktop/MCM/cumcm2013problems/B/2/';
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```



```

% 数据初始化
colvec_left = zeros(1980,19); % 存储所有图片最左边一列向量的矩阵
colvec_right = zeros(1980,19); % 存储所有图片最右边一列向量的矩阵
data_matrix = zeros(1980,72,19); % 存储所有图片信息的矩阵
% 读取文件，存储左右边栏的向量信息
for i=0:18
    filename = '0';
    if i<10
        filename = strcat(filename,'0'); %如果小于编号 10 添加字符 0
    end
    filename = strcat(filename,int2str(i));
    filename = strcat(filename, '.bmp');

    path = strcat(directory,filename); % 补充完整文件路径
    [A,M] = imread(path,'bmp');
    data_matrix(:,:,i+1) = A;
    colvec_left(:,i+1) = A(:,1);
    colvec_right(:,i+1) = A(:,size(A,2));
end
% 构造一个 19*2 的矩阵，右边一列从 0 到 18，左边一列分别填入与这些序号对应图片左端
相连图片的序号
idx_matrix = zeros(19,2);
for i=0:18
    idx_matrix(i+1,2) = i;
end
for i=0:18
    % 当前图片的左边与其他哪个图片的右边相连（最匹配）
    left_col = colvec_left(:,i+1);
    left_maxcos = -1;
    left_maxcos_idx = -1;
    value = 0;
    for j=0:18 % find the greatest value from other
        value = veccos(left_col,colvec_right(:,j+1)); %余弦定理，计算两个向量的夹角余弦
        if i == j
            continue
        end
        if value > left_maxcos
            left_maxcos = value;
            left_maxcos_idx = j;
        end
    end
    idx_matrix(i+1,1) = left_maxcos_idx;
end

```

```

end

white = 255*ones(1980,1);
%找出最左端的图片序号
leftmost = 0;
leftmost_idx = -1; % 记录最左端的图片序号，初始化为-1
for i=0:18
    left_col = colvec_left(:,i+1);
    value = veccos(white,left_col);
    if value > leftmost
        leftmost_idx = i;
        leftmost = value;
    end
end
idx_matrix(leftmost_idx+1,1) = -10000; %标记最左端
%找出右栏中的边界图片序号
rightmost = 0;
rightmost_idx = -1; % 记录最右端的图片序号，初始化为-1
for i=0:18
    right_col = colvec_right(:,i+1);
    value = veccos(white,right_col);
    if value > rightmost
        rightmost_idx = i;
        rightmost = value;
    end
end

idx_sequence = -1*ones(1,19); % 记录图片标号的左右顺序
idx_sequence(1) = leftmost_idx;
current_idx = idx_sequence(1);
for i=2:18
    % 找到当前图片右边的图片序号,减 1 是因为矩阵行数从 1 开始，图片序号从 0 开始
    next_idx = find(idx_matrix(:,1) == current_idx) - 1;
    idx_sequence(i) = next_idx;
    current_idx = next_idx;
end
idx_sequence(19) = rightmost_idx;

% 生成图片过程

% bmp_matrix 存储生成图片的像素矩阵，按顺序向右拼接而成
% 加 1 是因为图片序号从 0 开始，矩阵行数从 1 开始
bmp_matrix = data_matrix(:,:,idx_sequence(1)+1);

```

```

for i=1:18
    bmp_matrix = [bmp_matrix data_matrix(:, :, idx_sequence(i+1)+1)];
end
% 存储生成图片在 directory 变量代表的路径中，名称为 result.bmp
% 碎片 1 的图片存放在/Users/oh_baizhima/Desktop/MCM/cumcm2013problems/B/1/
% 碎片 2 的图片存放在/Users/oh_baizhima/Desktop/MCM/cumcm2013problems/B/2/
bmp_out = strcat(directory,'result.bmp');

imwrite(bmp_matrix,bmp_out);
% MATLAB 会提示图片过大不适合展示，并呈现出比较模糊的缩略图，可以点击放大查看
imshow(bmp_out)

```

序号 2 源程序

```

function [A,colvec_left,colvec_right,rowvec_top,rowvec_bot] = mcm2013b_p2()
%% B 题第二问程序
% 本程序适用于/cumcm2013problems/B 中的 3, 4 文件夹碎片文件

% 程序调用查看方法
% 1.修改 directory 变量为测试用机碎片文件所在文件夹的绝对路径
% 2.调用 mcm2013b_p2(), 命令行显示从左到右正确的碎片排序
% 3.MATLAB 弹出生成原图片的缩微图（清晰度较低，可放大查看）
% 4.查看 directory 文件夹中生成的 result.bmp 图片

%% 清除工作环境的历史数据
clear all; clc;

%% 修改路径信息

%%%%%%%%%%
%%%%%%%%%%
%%% 请修改文件夹路径，请将‘附件 3（4）’文件夹的‘附件’两字去掉，保证绝对路径
全英文
%%%%%%%%%%
%%%%%%%%%%
directory = '/Users/oh_baizhima/Desktop/MCM/cumcm2013problems/B/3/';
%%%%%%%%%%
%%%%%%%%%%

```

```

%% 数据初始化
colvec_left = zeros(180,209); % 存储所有图片最左边一列向量的矩阵
colvec_right = zeros(180,209); % 存储所有图片最右边一列向量的矩阵
rowvec_top = zeros(209,72); % 存储所有图片最上边一行向量的矩阵
rowvec_bot = zeros(209,72); % 存储所有图片最下边一行向量的矩阵
data_matrix = zeros(180,72,209); % 存储所有图片信息的矩阵
white_left_idx = []; % 存储所有左边栏为全白色的图片序号
white_right_idx = []; % 存储所有右边栏为全白色的图片序号
info_volume = zeros(209,1); %存储每个图片信息量的值（信息量：不全白边的数量）
info_0=[]; %存储所有信息量为 0 的图片序号
info_1=[]; %存储所有信息量为 1 的图片序号
info_2=[]; %存储所有信息量为 2 的图片序号
info_3=[]; %存储所有信息量为 3 的图片序号
info_4=[]; %存储所有信息量为 4 的图片序号

% 定义白边栏向量
whitecol = 255*ones(180,1);
white_left_count = 1;
whiterow = 255*ones(1,72);
white_right_count = 1;
%% 读取文件阶段，并分别存储每个图片的左右边栏
for i=0:208
    filename = '';
    if i<100
        filename = strcat(filename,'0'); %如果小于编号 100 添加字符 0
    end
    if i<10
        filename = strcat(filename,'0'); %如果小于编号 10 再添加字符 0
    end
    filename = strcat(filename,int2str(i));
    filename = strcat(filename,'.bmp');
    path = strcat(directory,filename); %补充完整文件路径
    [A,M] = imread(path,'bmp');
    data_matrix(:,:,i+1) = A; % 存储图像灰度信息
    colvec_left(:,i+1) = A(:,1); % 存储图像左边栏列向量
    colvec_right(:,i+1) = A(:,size(A,2)); % 存储图像右边栏列向量
    rowvec_top(i+1,:) = A(1,:); % 存储图像顶部行向量
    rowvec_bot(i+1,:) = A(size(A,1),:); % 存储图像底部行向量
    current_info_volume = 4; % 初始化信息量为 4

    if A(:,1) == whitecol
        white_left_idx = [white_left_idx i]; % 记录左边栏全白的图片序号
        white_left_count = white_left_count + 1;
    end
end

```

```

        current_info_volume = current_info_volume - 1;%左边栏全白，减1个信息量
    end
    if A(:,size(A,2)) == whitecol
        white_right_idx =[white_right_idx i]; % 记录右边栏全白的图片序号
        white_right_count = white_right_count + 1;
        current_info_volume = current_info_volume - 1;%右边栏全白，减1个信息量
    end
    if A(1,:) == whiterow
        current_info_volume = current_info_volume - 1;%顶部全白，减1个信息量
    end
    if A(size(A,1),:) == whiterow
        current_info_volume = current_info_volume - 1;%底部全白，减1个信息量
    end

    info_volume(i+1,1) = current_info_volume;%信息量的值存储在信息量矩阵
end

%% 按照信息量大小将图片序号分别存入 info_0 至 info_4 的向量中
% 因为第 i 列存放编号为 i-1 的图片，所以做一个-1 的操作，存放编号数
info_0 = find(info_volume(:,1) == 0);
info_0 = info_0 - ones(size(info_0,1),1);

info_1 = find(info_volume(:,1) == 1);
info_1 = info_1 - ones(size(info_1,1),1);

info_2 = find(info_volume(:,1) == 2);
info_2 = info_2 - ones(size(info_2,1),1);

info_3 = find(info_volume(:,1) == 3);
info_3 = info_3 - ones(size(info_3,1),1);

info_4 = find(info_volume(:,1) == 4);
info_4 = info_4 - ones(size(info_4,1),1);
%% 分别计算两组边栏的各自的白色边缘宽度
% 左边栏
left_white_width_data = [];
for i=0:size(white_left_idx,2)-1
    check_idx = white_left_idx(i+1);
    check_bmp_matrix = data_matrix(:, :, check_idx+1);
    white_width = 0;
    %从左到右检查全白得列数
    while check_bmp_matrix(:, white_width+1) == whitecol
        white_width = white_width + 1;
    end
end

```

```

end
% 存放左白边宽度
left_white_width_data = [left_white_width_data; check_idx white_width];
end
% 右边栏
right_white_width_data = [];
for i=0:size(white_right_idx,2)-1
    check_idx = white_right_idx(i+1);
    check_bmp_matrix = data_matrix(:, :, check_idx+1);
    white_width = 0;
    %从右至左检查全白得列数
    %check_col = size(check_bmp_matrix,2)-white_width
    while check_bmp_matrix(:,size(check_bmp_matrix,2)-white_width) == whitecol
        white_width = white_width + 1;
    end
    %存放右白边宽度
    right_white_width_data=[right_white_width_data; check_idx white_width];
end

%% 生成完整图片过程

% bmp_matrix 存储生成图片的像素矩阵，按顺序向右拼接而成
% 加 1 是因为图片序号从 0 开始，矩阵行数从 1 开始
canvas_matrix = 255*ones(11*180,19*72);% 初始化全白的合成图片大小的矩阵
A = xlsread(strcat(directory,'photo3.xlsx'));
for row=1:11
    for col=1:19
        photo_idx = A(row,col);
        if photo_idx >= 0
            r_start = 180*(row-1)+1;
            r_end = 180*row;
            c_start = 72*(col-1)+1;
            c_end = 72*col;
            canvas_matrix(r_start:r_end,c_start:c_end) = ...
                data_matrix(:, :, photo_idx+1);
        end
    end
end
bmp_out = strcat(directory,'result.bmp');
imwrite(canvas_matrix,bmp_out);

%{

```

```

%% 人工干预阶段的四种爬行方法
%% 左推右

current_bmp_idx = 148; %填入当前图片序号
current_right_col = colvec_right(:,current_bmp_idx + 1);
match_idx = -1;
match_val = -1;
cos_value_matrix = zeros(209,2);
for i=0:208
    cos_value_matrix(i+1,1) = i;
end

for i=0:208
    %a = current_right_col';
    %b = colvec_left(:,i+1)';
    value = veccos(current_right_col,colvec_left(:,i+1));
    cos_value_matrix(i+1,2) = value;
    if value > match_val && i~= current_bmp_idx
        match_val = value;
        match_idx = i;
    end
end
match_idx %输出右边图片序号

```

```

%% 右推左

current_bmp_idx = 183; %填入当前图片序号
current_left_col = colvec_left(:,current_bmp_idx + 1);
match_idx = -1;
match_val = -1;
cos_value_matrix = zeros(209,2);
for i=0:208
    cos_value_matrix(i+1,1) = i;
end

for i=0:208
    %a = current_right_col';
    %b = colvec_left(:,i+1)';
    value = veccos(current_left_col,colvec_right(:,i+1));
    cos_value_matrix(i+1,2) = value;
    if value > match_val
        match_val = value;
    end
end

```

```

        match_idx = i;
    end
end
match_idx %输出左边图片序号

%% 上推下
current_bmp_idx = 74; %填入当前图片序号
current_bot_row = rowvec_bot(current_bmp_idx + 1,:);
match_idx = -1;
match_val = -1;
cos_value_matrix = zeros(209,2);
for i=0:208
    cos_value_matrix(i+1,1) = i;
end

for i=0:208
    value = veccos(current_bot_row,rowvec_top(i+1,:));
    cos_value_matrix(i+1,2) = value;
    if value > match_val
        match_val = value;
        match_idx = i;
    end
end
match_idx %输出下边图片序号

%% 下推上
current_bmp_idx = 163; %填入当前图片序号
current_top_row = rowvec_top(current_bmp_idx + 1,:);
match_idx = -1;
match_val = -1;
cos_value_matrix = zeros(209,2);
for i=0:208
    cos_value_matrix(i+1,1) = i;
end

for i=0:208
    value = veccos(current_top_row,rowvec_bot(i+1,:));
    cos_value_matrix(i+1,2) = value;
    if value > match_val
        match_val = value;
        match_idx = i;
    end
end
match_idx %输出上边图片序号

```



```
%}
```

序号 3 源程序

```
function C = mcm2013b_p2stats()
clear all;clc;

%% 导出文件目录
directory = '/Users/oh_baizhima/Desktop/MCM/cumcm2013problems/B/3/';
path = strcat(directory,'3stats.xlsx');

[A,colvec_left,colvec_right,rowvec_top,rowvec_bot] = mcm2013b_p2();
fprintf('图片信息读取完成,请稍后...\n');
%% 生成统计左右、上下方向正确与否的 0\1 矩阵

% 生成有空隙的大矩阵 C，大小为 11*2+10,19*2+18
% 行数 11*2 是存储左右两个方向，+10 给上下方向留出空间
% 列数 19*2+18，列数 19*2 是存储上下两个方向结果，+18 给左右方向留出空间
% 为了视觉效果，之前 11*19 序号矩阵的每个序号在大矩阵 C 中占据 2*2 的位置，
% 并且值全部填满序号值
%
% 矩阵片段示意图：
%      ...   ...   ...
% ... 33 33 -1 68 68 ... 此处-1 所占单元格记录左推右，正确填写 300，错误填写-300
% ... 33 33 -1 68 68     此处-1 所占单元格记录右推左，正确填写 400，错误填写-400
% ... -1 -1 -1 -1 -1 ... 此处第 1 个-1 所占单元格记录上推下是否正确，正确填写 500，错误
填写-500
% ... 72 72 -1 55 55
% ... 72 72 -1 55 55 ...
% ... -1 -1 -1 -1 -1 ... 此处第 2 个-1 所占单元格记录下推上是否正确，正确填写 600，错误
填写-600
%      ...   ...   ...

C = -1*ones(11*2+10,19*2+18);
for row = 1:11
    for col = 1:19
        C(3*row-2,3*col-2) = A(row,col); %左上
        C(3*row-2,3*col-1) = A(row,col); %右上
        C(3*row-1,3*col-2) = A(row,col); %左下
        C(3*row-1,3*col-1) = A(row,col); %右下
```

```

end
end

for row = 1:11
    fprintf('左右递推关系已经存储%d 行,共 11 行,请稍后...\n',row);
    for col_between = 1:18
        left_idx = A(row,col_between);
        right_idx = A(row,col_between+1);
        % 统计左推右
        current_bmp_idx = left_idx;
        current_right_col = colvec_right(:,current_bmp_idx + 1);
        match_idx = -1;
        match_val = -1;
        cos_value_matrix = zeros(209,2);
        for i=0:208
            cos_value_matrix(i+1,1) = i;
        end

        for i=0:208
            value = veccos(current_right_col,colvec_left(:,i+1));
            cos_value_matrix(i+1,2) = value;
            if value > match_val && i~= current_bmp_idx
                match_val = value;
                match_idx = i;
            end
        end
    end
    %
    if match_idx == right_idx
        C(3*row-2,3*col_between) = 300;
    else
        C(3*row-2,3*col_between) = -300;
    end
end

% 统计右推左
current_bmp_idx = right_idx;
current_left_col = colvec_left(:,current_bmp_idx + 1);
match_idx = -1;
match_val = -1;
cos_value_matrix = zeros(209,2);
for i=0:208
    cos_value_matrix(i+1,1) = i;
end
for i=0:208
    value = veccos(current_left_col,colvec_right(:,i+1));

```

```

        cos_value_matrix(i+1,2) = value;
        if value > match_val
            match_val = value;
            match_idx = i;
        end
    end
end
%

if match_idx == left_idx
    C(3*row-1,3*col_between) = 400;
else
    C(3*row-1,3*col_between) = -400;
end
end
end

for col=1:19
    fprintf('上下递推关系已经存储%d 列,共 19 列,请稍后...\n',col);
    for row_between = 1:10
        up_idx = A(row_between,col);
        down_idx = A(row_between+1,col);
        % 统计上到下
        current_bmp_idx = up_idx;
        current_bot_row = rowvec_bot(current_bmp_idx + 1,:);
        match_idx = -1;
        match_val = -1;
        cos_value_matrix = zeros(209,2);
        for i=0:208
            cos_value_matrix(i+1,1) = i;
        end
        for i=0:208
            value = veccos(current_bot_row,rowvec_top(i+1,:));
            cos_value_matrix(i+1,2) = value;
            if value > match_val
                match_val = value;
                match_idx = i;
            end
        end
    end

    if match_idx == down_idx
        C(3*row_between,3*col-2) = 500;
    else
        C(3*row_between,3*col-2) = -500;
    end
end
end

```

```

end
% 统计下到上

current_bmp_idx = down_idx; %填入当前图片序号
current_top_row = rowvec_top(current_bmp_idx + 1,:);
match_idx = -1;
match_val = -1;
cos_value_matrix = zeros(209,2);
for i=0:208
    cos_value_matrix(i+1,1) = i;
end
for i=0:208
    value = veccos(current_top_row,rowvec_bot(i+1,:));
    cos_value_matrix(i+1,2) = value;
    if value > match_val
        match_val = value;
        match_idx = i;
    end
end

if match_idx == up_idx
    C(3*row_between,3*col-1) = 600;
else
    C(3*row_between,3*col-1) = -600;
end
end

end

end

xlswrite(path,C);

```

序号 4 源程序

```

function [value] = veccos(vec1,vec2)
% 计算两个向量夹角的余弦值  $\cos(\text{vec1},\text{vec2}) = \text{vec1} \cdot \text{vec2} / (|\text{vec1}| \cdot |\text{vec2}|)$ 
value = sum(vec1.*vec2)/(sqrt(sum(vec1.*vec1))*sqrt(sum(vec2.*vec2)));

```

序号 5 源程序

```

function [Y1,Y2] = mcm2013b_p3()
%% B 题第三问程序
% 本程序适用于/cumcm2013problems/B 中的 5 文件夹碎片文件

```

```

% 程序调用查看方法
% 1.修改 directory 变量为测试用机碎片文件所在文件夹的绝对路径
% 2.调用 mcm2013b_p3(), 命令行显示从左到右正确的碎片排序
% 3.MATLAB 弹出生成原图片的缩微图（清晰度较低，可放大查看）
% 4.查看 directory 文件夹中生成的 result.bmp 图片

%% 清除工作环境的历史数据
clear all; clc;

%% 修改路径信息

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%
%% 请修改文件夹路径，请将‘附件 5’文件夹的‘附件’两字去掉，保证绝对路径全英文
%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%
directory = '/Users/oh_baizhima/Desktop/MCM/cumcm2013problems/B/5/';
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%
%% 数据初始化
colvec_left = zeros(180,209*2); % 存储所有图片最左边一列向量的矩阵
colvec_right = zeros(180,209*2); % 存储所有图片最右边一列向量的矩阵
rowvec_top = zeros(209*2,72); % 存储所有图片最上边一行向量的矩阵
rowvec_bot = zeros(209*2,72); % 存储所有图片最下边一行向量的矩阵
data_matrix = zeros(180,72,209*2); % 存储所有图片信息的矩阵
white_left_idx = []; % 存储所有左边栏为全白色的图片序号
white_right_idx = []; % 存储所有右边栏为全白色的图片序号
white_top_idx=[];
white_bot_idx=[];
info_volume = zeros(209*2,1); %存储每个图片信息量的值（信息量：不全白边的数量）
leftmost_col_idx = []; % 存储最左边栏序号
revised_leftmost_col_idx = []; %正确的 22 个边栏碎片编号

info_0=[]; %存储所有信息量为 0 的图片序号
info_1=[]; %存储所有信息量为 1 的图片序号
info_2=[]; %存储所有信息量为 2 的图片序号
info_3=[]; %存储所有信息量为 3 的图片序号
info_4=[]; %存储所有信息量为 4 的图片序号
info_5=[]; %存储所有信息量为 5 的图片序号
info_6=[]; %存储所有信息量为 6 的图片序号
info_7=[]; %存储所有信息量为 7 的图片序号

```

```

info_8=[]; %存储所有信息量为 8 的图片序号

% 定义白边栏向量
whitecol = 255*ones(180,1);
%white_left_count = 1; %左边宽度计量
whiterow = 255*ones(1,72);
%white_right_count = 1;

%% 读取文件阶段，并分别存储每个图片正反面的左右边栏
for i=0:208
    filename = "";
    if i<100
        filename = strcat(filename,'0'); %如果小于编号 100 添加字符 0
    end
    if i<10
        filename = strcat(filename,'0'); %如果小于编号 10 再添加字符 0
    end
    filename = strcat(filename,int2str(i));
    for side=0:1
        if side == 0
            filename_side = strcat(filename,'a');
            filename_side = strcat(filename_side,'.bmp');
        else
            filename_side = strcat(filename,'b');
            filename_side = strcat(filename_side,'.bmp');
        end

        path = strcat(directory,filename_side); %补充完整文件路径
        [A,M] = imread(path,'bmp');
        num_count = 2*i+side+1;
        data_matrix(:, :, num_count) = A; % 存储图像灰度信息
        colvec_left(:, num_count) = A(:,1); % 存储图像左边栏列向量
        colvec_right(:, num_count) = A(:,size(A,2)); % 存储图像右边栏列向量
        rowvec_top(num_count, :) = A(1,:); % 存储图像顶部行向量
        rowvec_bot(num_count, :) = A(size(A,1),:); % 存储图像底部行向量

    end
end

%% 计算全部图片信息量
for i=0:208
    current_info_volume = 8; % 初始化信息量为 8
    % 计算 A 面信息量

```

```

A = data_matrix(:, :, 2*i+1);
if A(:,1) == whitecol
    white_left_idx = [white_left_idx; i 1]; % 记录左边栏全白的图片序号
    %white_left_count = white_left_count + 1;
    current_info_volume = current_info_volume - 1;%左边栏全白，减 1 个信息量
end
if A(:,size(A,2)) == whitecol
    white_right_idx = [white_right_idx; i 1]; % 记录右边栏全白的图片序号
    %white_right_count = white_right_count + 1;
    current_info_volume = current_info_volume - 1;%右边栏全白，减 1 个信息量
end
if A(1,:) == whiterow
    white_top_idx = [white_top_idx; i 1];
    current_info_volume = current_info_volume - 1;%顶部全白，减 1 个信息量
end
if A(size(A,1),:) == whiterow
    white_bot_idx = [white_bot_idx; i 1];
    current_info_volume = current_info_volume - 1;%底部全白，减 1 个信息量
end
% 计算 B 面信息量
A = data_matrix(:, :, 2*i+2);
if A(:,1) == whitecol
    white_left_idx = [white_left_idx; i 2]; % 记录左边栏全白的图片序号
    %white_left_count = white_left_count + 1;
    current_info_volume = current_info_volume - 1;%左边栏全白，减 1 个信息量
end
if A(:,size(A,2)) == whitecol
    white_right_idx = [white_right_idx; i 2]; % 记录右边栏全白的图片序号
    %white_right_count = white_right_count + 1;
    current_info_volume = current_info_volume - 1;%右边栏全白，减 1 个信息量
end
if A(1,:) == whiterow
    white_top_idx = [white_top_idx; i 2];
    current_info_volume = current_info_volume - 1;%顶部全白，减 1 个信息量
end
if A(size(A,1),:) == whiterow
    white_bot_idx = [white_bot_idx; i 2];
    current_info_volume = current_info_volume - 1;%底部全白，减 1 个信息量
end
info_volume(i+1,1) = current_info_volume; %信息量的值存储在信息量矩阵第二列

end

%% 按照信息量大小将图片序号分别存入 info_0 至 info_8 的向量中

```

```

% 因为第 i 列存放编号为 i-1 的图片，所以做一个-1 的操作，存放编号数
info_0 = find(info_volume(:,1) == 0);
info_0 = info_0 - ones(size(info_0,1),1);

info_1 = find(info_volume(:,1) == 1);
info_1 = info_1 - ones(size(info_1,1),1);

info_2 = find(info_volume(:,1) == 2);
info_2 = info_2 - ones(size(info_2,1),1);

info_3 = find(info_volume(:,1) == 3);
info_3 = info_3 - ones(size(info_3,1),1);

info_4 = find(info_volume(:,1) == 4);
info_4 = info_4 - ones(size(info_4,1),1);

info_5 = find(info_volume(:,1) == 5);
info_5 = info_5 - ones(size(info_5,1),1);

info_6 = find(info_volume(:,1) == 6);
info_6 = info_6 - ones(size(info_6,1),1);

info_7 = find(info_volume(:,1) == 7);
info_7 = info_7 - ones(size(info_7,1),1);

info_8 = find(info_volume(:,1) == 8);
info_8 = info_8 - ones(size(info_8,1),1);

%% 寻找最左(右)边一列的碎片
% 特征：假设图片 M 的 a 面左边栏为纯白，且 M 的 b 面右边栏为纯白，则判定这是某一面
最左边一列
for i=1:size(white_left_idx,1)
    current_idx = white_left_idx(i,1);
    current_side = white_left_idx(i,2); % 1 代表 A 面， 2 代表 B 面
    opposite_side = 3 - current_side;
    if find(white_right_idx(:,1)) > 0 % 首先找到有
        %candidate_row_num = find(white_right_idx(:,1));
        %if white_right_idx(candidate_row_num,2) == opposite_side %其次确定是反面
            leftmost_col_idx = [leftmost_col_idx; current_idx];
        %end
    end
end

%% 计算最左(右)边一列的碎片的白色边栏宽度

```



```

% 首先将存放候选最左(右)列序号的向量向右扩充两列，分别存放两面的边宽值
leftmost_col_idx = [leftmost_col_idx zeros(size(leftmost_col_idx,1),4)];
for i=1:size(leftmost_col_idx,1)
    % 先找到当前序号的哪面右和哪面左是白边
    current_idx = leftmost_col_idx(i,1);
    current_side_bmp = data_matrix(:,current_idx*2+1);
    opposite_side_bmp = data_matrix(:,current_idx*2+2);
    % 两面左边往右
    current_white_width = 0;
    opposite_white_width = 0;
    while current_side_bmp(:,current_white_width+1) == whitecol
        current_white_width = current_white_width + 1;
    end
    while opposite_side_bmp(:,opposite_white_width+1) == whitecol
        opposite_white_width = opposite_white_width + 1;
    end
    leftmost_col_idx(i,2) = current_white_width; %A 面左
    leftmost_col_idx(i,5) = opposite_white_width; %B 面左

    % 两面右边往左
    current_white_width = 0;
    opposite_white_width = 0;
    while current_side_bmp(:,size(current_side_bmp,2)-current_white_width) == whitecol
        current_white_width = current_white_width + 1;
    end
    while opposite_side_bmp(:,size(opposite_side_bmp,2)-opposite_white_width) == whitecol
        opposite_white_width = opposite_white_width + 1;
    end
    leftmost_col_idx(i,4) = current_white_width; %A 面右
    leftmost_col_idx(i,3) = opposite_white_width; %B 面右
end
% 筛选出正确边栏碎片序号
for i=1:size(leftmost_col_idx,1)
    above10count =0;
    for col=2:5
        if leftmost_col_idx(i,col) > 10
            above10count = above10count+1;
        end
    end
    if above10count == 2
        revised_leftmost_col_idx = ...
            [revised_leftmost_col_idx;leftmost_col_idx(i,1)];
    end
end
end

```

```
revised_leftmost_col_idx = [revised_leftmost_col_idx zeros(size(revised_leftmost_col_idx,1),4)];
```

```
% 找出边栏中，选出最顶和最底的碎片，各两个，共四个
```

```
for i=1:size(revised_leftmost_col_idx,1)
```

```
    current_idx = revised_leftmost_col_idx(i,1);
```

```
    current_side_bmp = data_matrix(:, :, current_idx*2+1);
```

```
    top_white_width = 0;
```

```
    bot_white_width = 0;
```

```
    while current_side_bmp(top_white_width+1,:) == whiterow
```

```
        top_white_width = top_white_width + 1;
```

```
    end
```

```
    while current_side_bmp(size(current_side_bmp,1)-bot_white_width,:) == whiterow
```

```
        bot_white_width = bot_white_width + 1;
```

```
    end
```

```
    revised_leftmost_col_idx(i,2) = top_white_width;
```

```
    revised_leftmost_col_idx(i,4) = bot_white_width;
```

```
    %反面
```

```
    current_side_bmp = data_matrix(:, :, current_idx*2+2);
```

```
    top_white_width = 0;
```

```
    bot_white_width = 0;
```

```
    while current_side_bmp(top_white_width+1,:) == whiterow
```

```
        top_white_width = top_white_width + 1;
```

```
    end
```

```
    while current_side_bmp(size(current_side_bmp,1)-bot_white_width,:) == whiterow
```

```
        bot_white_width = bot_white_width + 1;
```

```
    end
```

```
    revised_leftmost_col_idx(i,3) = top_white_width;
```

```
    revised_leftmost_col_idx(i,5) = bot_white_width;
```

```
end
```

```
%% 根据 revised_leftmost_col_idx 得到的上下白边宽度数值，推断出左（右）下的两个碎片
```

```
% 编号分别为 54 和 99
```

```
%% 生成图片
```

```
Y1_idx = zeros(11,19);
```

```
Y2_idx = zeros(11,19);
```

```
Y2_idx_test = zeros(11,19);
```

```

Y1 = 255*ones(11*180,19*72);
Y2 = 255*ones(11*180,19*72);

Y1_idx = xlsread(strcat(directory,'side_1.xlsx'));
for i=0:size(Y1_idx,1)-1
    for j = 0:size(Y1_idx,2)-1
        Y1(180*i+1:180*i+180,72*j+1:72*j+72) = data_matrix(:,Y1_idx(i+1,j+1));
    end
end
bmp_out = strcat(directory,'result.bmp');
imwrite(Y1,bmp_out);
Y2_idx = xlsread(strcat(directory,'side_2.xlsx'));
for i=0:size(Y2_idx,1)-1
    for j = 0:size(Y2_idx,2)-1
        Y2(180*i+1:180*i+180,72*j+1:72*j+72) = data_matrix(:,Y2_idx(i+1,j+1));
    end
end
bmp_out = strcat(directory,'result2.bmp');
imwrite(Y2,bmp_out);

%% 人工干预后，得到一面的外边框系数矩阵
X = -1*ones(11,19,2);
X(1,:,1) = [136 47 20 164 81 189 29 18 108 66 110 174 183 150 155 140 125 111 78];
X(1,:,2) = [1 2 2 1 1 1 2 1 2 2 2 1 1 2 2 2 1 1];
X(11,:,1) = [54 196 112 103 55 100 106 91 49 26 113 134 104 6 123 109 96 43 99];
X(11,:,2) = [1 1 2 2 1 1 1 2 1 1 2 2 2 2 2 2 1 2 2];
X(:,1,1) = [136;5;143;83;90;13;35;172;105;9;54];
X(:,1,2) = [1;2;1;2;2;2;2;2;2;1;1];
X(:,19,1)=[78;89;186;199;88;114;146;165;3;23;99];
X(:,19,2)=[1;2;1;1;1;2;2;1;1;2];

for k=2:10
    current_bmp_idx = X(k,1,1);
    current_side = X(k,1,2);
    for j=2:18
        match_idx = -1;
        match_val = -1;
        match_side = 0; % 1 是 a, 2 是 b
        a_side_right_col = colvec_right(:,current_bmp_idx*2 + current_side);
        b_side_left_col = colvec_left(:, current_bmp_idx*2 + (3 - current_side));
        long_col = [a_side_right_col;b_side_left_col];
        %cos_value_matrix = zeros(209,3); %第一列图片序号，第二列 cos 值
        %for i=0:208

```

```

        %cos_value_matrix(i+1,1) = i;
    %end
    for i=0:208
        test_col = [colvec_left(:,i*2+side); colvec_right(:,i*2+side)];
        value = veccos(long_col,test_col);
        %cos_value_matrix(i+1,2) = value;
        if value > match_val && i ~= current_bmp_idx
            match_val = value;
            match_idx = i;
            match_side = 1;
        end
        test_col = [colvec_right(:,i*2+side);colvec_left(:,i*2+side)];
        value = veccos(long_col,test_col);
        %cos_value_matrix(i+1,3) = value;
        if value > match_val && i ~= current_bmp_idx
            match_val = value;
            match_idx = i;
            match_side = 2;
        end
    end
    end
    X(k,j,1) = match_idx;
    X(k,j,2) = match_side;
    current_bmp_idx = match_idx;
    current_side = match_side;
end
end

%{
%% 新版左推右

current_bmp_idx = 35; %填入当前图片序号
current_side = 1; % 填入当前图片正反面
match_idx = -1;
match_val = -1;
match_side = 0; % 1 是 a, 2 是 b
%if current_side == 2
a_side_right_col = colvec_right(:,current_bmp_idx*2 + current_side);
b_side_left_col = colvec_left(:, current_bmp_idx*2 + (3 - current_side));

long_col = [a_side_right_col;b_side_left_col];
cos_value_matrix = zeros(209,3); %第一列图片序号, 第二列 cos 值
for i=0:208
    cos_value_matrix(i+1,1) = i;

```

```

end
for i=0:208
    test_col = [colvec_left(:,i*2+side); colvec_right(:,i*2+side)];
    value = veccos(long_col,test_col);
    cos_value_matrix(i+1,2) = value;
    if value > match_val && i ~= current_bmp_idx
        match_val = value;
        match_idx = i;
        match_side = 1;
    end
    test_col = [colvec_right(:,i*2+side);colvec_left(:,i*2+side)];
    value = veccos(long_col,test_col);
    cos_value_matrix(i+1,3) = value;
    if value > match_val && i ~= current_bmp_idx
        match_val = value;
        match_idx = i;
        match_side = 2;
    end
end
end
match_idx %输出右边图片序号
match_side

%% 老版左推右
current_bmp_idx = 196; %填入当前图片序号
current_side = 1; % 填入当前图片正反面
a_side_right_col = colvec_right(:,current_bmp_idx*2 + current_side);
b_side_left_col = colvec_left(:, current_bmp_idx*2 + current_side);

match_idx = -1;
match_val = -1;
match_side = 0; % 1 是 a, 2 是 b
% 初始化 cos_value_matrix 矩阵
cos_value_matrix = zeros(418,3); %第一列图片序号, 第二列 a(1)\b(2)面, 第三列 cos 值
for i=0:417
    if mod(i,2) == 0
        cos_value_matrix(i+1,1) = i/2;
        cos_value_matrix(i+1,2) = 1;
    else
        cos_value_matrix(i+1,1) = (i-1)/2;
        cos_value_matrix(i+1,2) = 2;
    end
end

```

```

end
% 分别计算，寻找最大 cos
for i=0:417
    value = veccos(a_side_right_col,colvec_left(:,i+1));
    opposite_value = veccos(b_side_left_col,colvec_right(:,i+1));
    cos_value_matrix(i+1,3) = value + opposite_value;
    if i == 7
        value
        opposite_value
        cos_value_matrix(i+1,3)
    end
    if value > match_val && i~= 2*current_bmp_idx+1 && i~= 2*current_bmp_idx+2
        match_val = value;
        if mod(i,2) == 0
            match_idx = i/2;
            match_side = 1;
        else
            match_idx = (i-1)/2;
            match_side = 2;
        end
    end
end
end
match_idx %输出右边图片序号
match_side
%}

```