

电 子 科 技 大 学 实 验 报 告

课程名称： 数学实验

实验地点：

指导教师：

评 分：

完成实验学生信息：

	姓名	学号	选课 序号	贡献百分比 (%)	备注 (主要工作)

1. 学生人数按照任课教师要求限定；
2. 对于“评价、改进、总结和体会”都要认真填写，和其他内容是评价实验成绩的重要参考。

图片拼接的变换矩阵

摘要

在生活中常常会遇到需要将数张有重叠部分的图像拼成一幅无缝高分辨率图像的问题，而解决这个问题的其中一种较为良好的方式是采用图像拼接技术。图像拼接技术主要分为两个部分，其一是图像配准，其二是图像融合。本文主要针对后一项问题进行了分析和解答。

关键词：回归分析、噪声消除，图像变换、图像拼接

目 录

摘要.....	2
1. 问题重述	4
2. 问题分析	4
2.1. 图像拼接的分析	4
2.2. 求解变换矩阵的分析	4
2.3. 变换矩阵的计算效果的分析	4
3. 模型假设	5
4. 定义与符号说明	5
4.1. 名词解释	5
4.2. 符号说明	5
5. 模型的建立与求解	6
5.1. 参考回归平面的建模与噪声数据的去除	6
5.1.1. 建模前准备	6
5.1.2. 模型建立	6
5.1.3. 模型求解及检验	7
5.2. 图像拼接的建模与求解	9
5.2.1. 建模前准备	10
5.2.2. 模型建立	10
5.2.3. 模型求解及检验	10
6. 模型的评价与推广	10
6.1. 模型优点	10
6.2. 模型缺点	11
6.3. 模型改进方向	11
6.4. 模型推广	11
7. 心得体会与总结	11
8. 对本实验问题的设计提出改进意见	12
参考文献.....	12
附件.....	12
附件.....	13
附件一：排序后的相对噪声值.....	13
附件二：实验代码.....	17
附件三：最终拼接后的图片.....	19

1. 问题重述

1. 已知两张图片以及配对的 SURF 特征点，需要设计方法计算出第 2 张图片变换到第 1 张图片上的“变换矩阵”。
2. 通过图像的拼接来观察变换矩阵的计算效果

2. 问题分析

每一个特征点通过相对应的横坐标和纵坐标描述，

2.1. 图像拼接的分析

要实现图像的拼接，关键在于将一张图片按照一定的旋转，平移，伸缩等操作变换后与另一种图片拼接起来。本题背景已给出若干对 SURF 特征点，通过对局部若干点的变换从而推出全局的变换矩阵。考虑一阶线性变换，我们可以利用矩阵来描述变换，问题转化为求解变换矩阵的系数。

2.2. 求解变换矩阵的分析

由于特征点提取与匹配算法的限制，部分特征点对可能不符合要求，需要去除这些噪声。剩余的多组特征点对作为自变量需要考虑其对于因变量变换矩阵系数的影响，一种统计方法是使用回归分析。每一对特征点作为一个数据用三维坐标描述，通过 matlab 的多元线性拟合函数 regress 拟合出相应的平面。平面的参数即为我们所求的变换矩阵参数。

2.3. 变换矩阵的计算效果的分析

得到了变换矩阵后，可以将第 2 张图片上的点变换到第 1 张图片上，再利用 matlab 中的绘图函数将图片显示出来，观察图片的拼接效果，即可评估变换矩阵的计算效果。

3. 模型假设

1. 特征点对中噪声数据占少数。
2. 将 srow-col-2.jpg 当作待拼接图片。
3. 图片变换是一阶线性操作。

4. 定义与符号说明

4.1. 名词解释

噪声数据：不符合匹配要求的 SURF 特征点。

参考回归平面：选定 3 个非噪声数据构造的线性回归平面。

目标回归平面：对非噪声数据构造的线性回归平面。

相对噪声值：消除噪声数据过程中，每对数据相对于参考回归平面的残差。

噪声阈值：对于相对噪声值超过噪声阈值的点可以视作噪声数据。

4.2. 符号说明

符号	说明	单位
x_1, y_1	第 1 张图片变换前的横，纵坐标	像素
x_2, y_2	第 2 张图片变换前的横，纵坐标	像素
x'	第 2 张图片变换后的横坐标	像素
y'	第 2 张图片变换后的纵坐标	像素
$t_{11} \ t_{12}$	横坐标变换系数	/
$t_{21} \ t_{22}$	纵坐标变换系数	/

5. 模型的建立与求解

5.1. 参考回归平面的建模与噪声数据的去除

5.1.1. 建模前准备

假设第 2 张图片变换前的坐标为 (x, y) ，变换后的坐标为 (x', y') 而变换前后的坐标满足下列方程组：

$$x' = t_{11}x + t_{12}y + p_x.$$

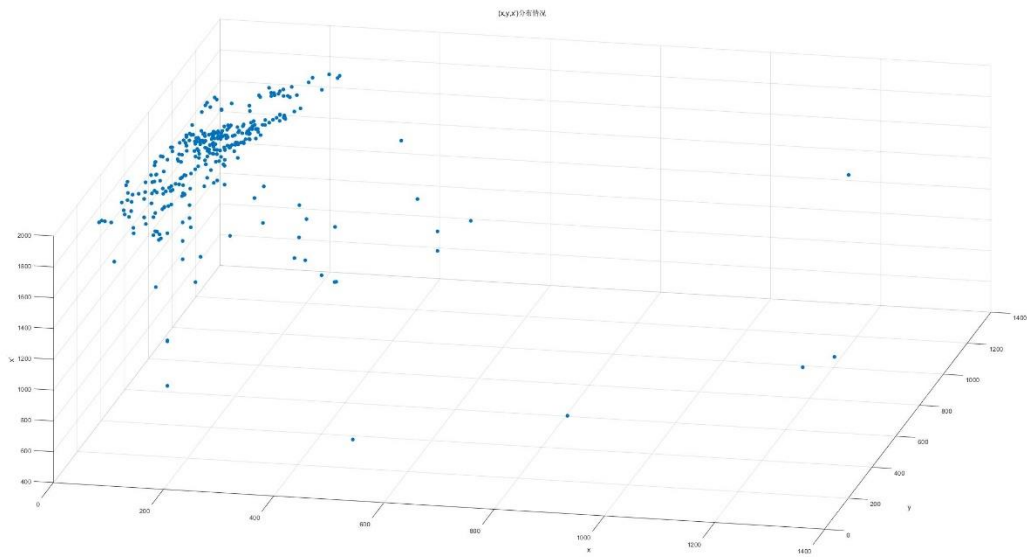
$$y' = t_{21}x + t_{22}y + p_y.$$

可以将上述方程组转化为矩阵乘法的形式

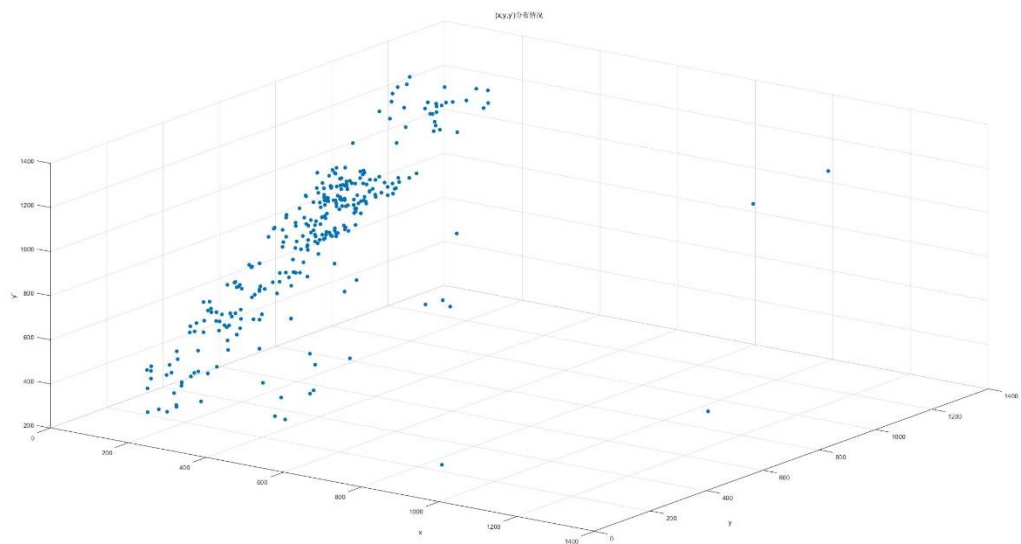
$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} t_{11} & t_{12} & p_x \\ t_{21} & t_{22} & p_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

5.1.2. 模型建立

首先对待处理数据进行分析：



横轴为 x_2 , 纵轴为 y_2 , 竖轴为 x_1



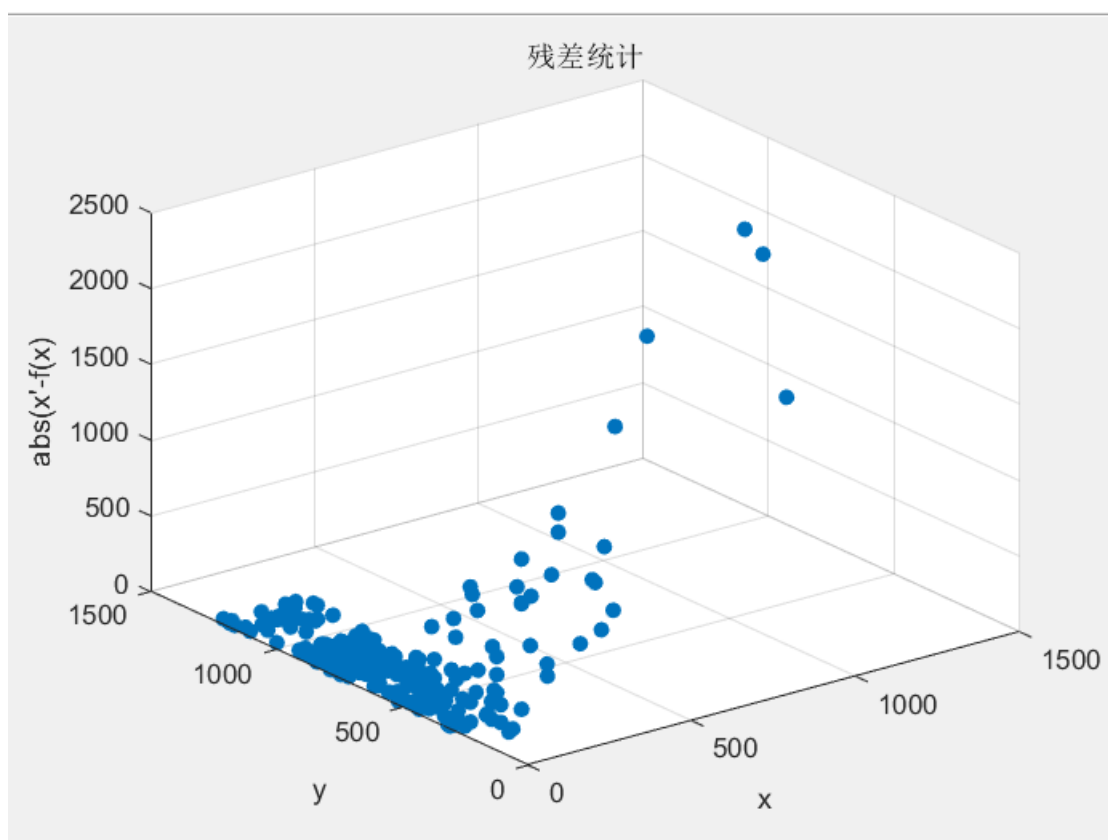
横轴为 x_2 , 纵轴为 y_2 , 竖轴为 y_1

将每组数据用三维坐标 (x_2, y_2, x_1) (x_2, y_2, y_1) 表示出来, 我们发现, 绝大部分坐标都集中于一个特定的平面, 我们可以将这个平面视作我们要求的回归平面, 而将其他点是为噪声数据。

我们可以先初步预估一个参考回归平面, 将离参考回归平面较远的点去除。

5. 1. 3. 模型求解及检验

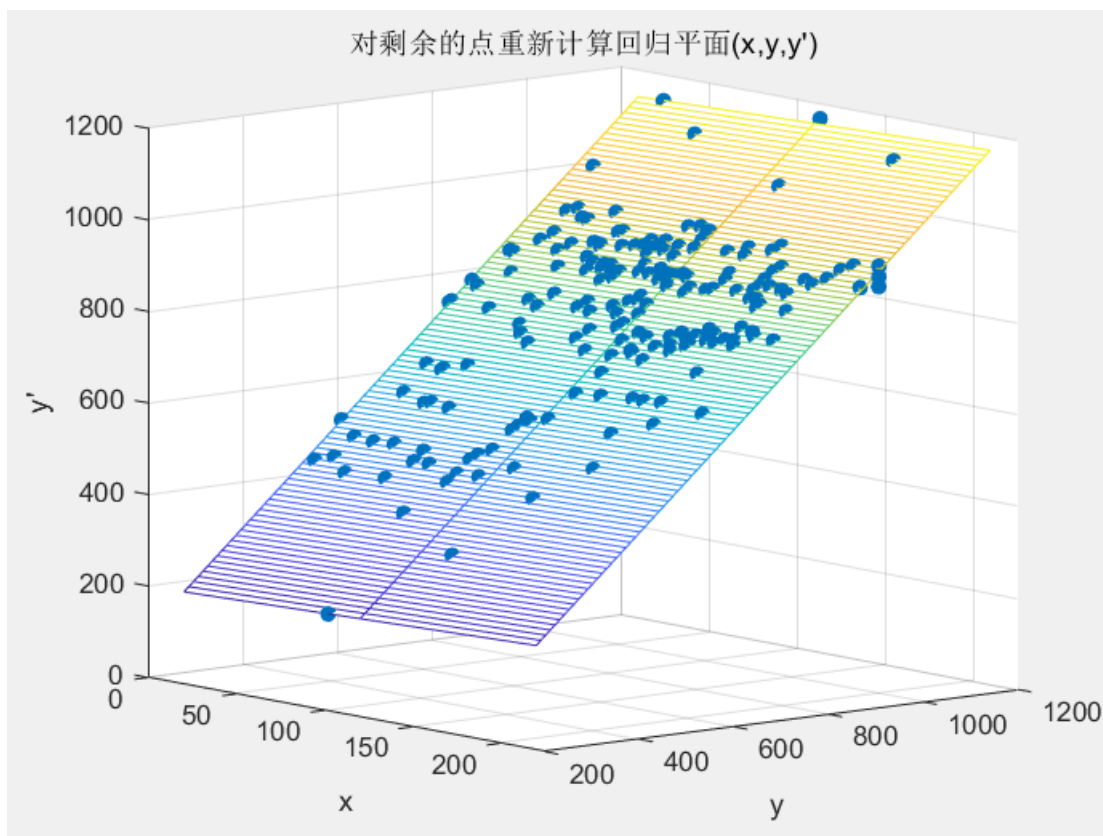
首先由上述分析以及 (x_2, y_2, x_1) (x_2, y_2, y_1) 坐标图，我们可以得到 3 组非噪声数据点，分别是原始数据的第 154, 215, 237 组。调用线性回归函数得到参考回归平面，设相对噪声值为数据点与回归平面的残差，求解出残差分布的图像如下。



残差分布图（以 (x_2, y_2, x_1) 为例）

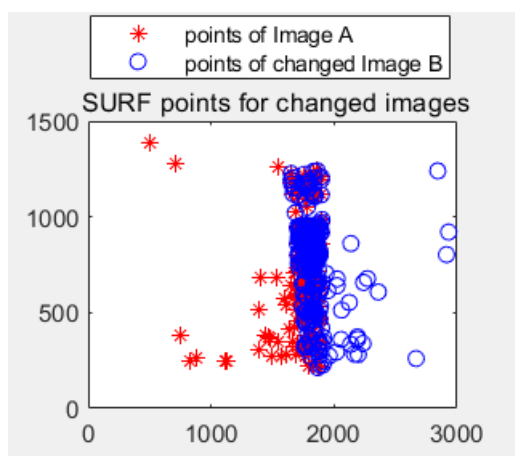
将相对噪声值排序，得到以下数据（见附件一）：

实验中发现将噪声阈值设为 2 可以实现较好的图像拼接效果，去掉相对噪声值大于噪声阈值的点，再对剩余的点进行一次回归平面的计算，计算后得出的回归平面如下图所示。



目标回归平面与剩余点的拟合程度（以 (x_2, y_2, y_1) 为例）

由图可见，非噪声点基本分布在回归平面上，也就意味着剩余的点基本上是两幅图像可以一一配对的点，因此完成了对变换矩阵的求解。



将特征点按照变换矩阵变换后的分布情况

5. 2. 图像拼接的建模与求解

5.2.1. 建模前准备

有了计算效果较好得变换矩阵后，还需进一步观察图像的拼接效果，以此来最终评价变换矩阵的计算效果。首先创建一张 1050*3000 的 uint8 数组储存新图片的像素。

5.2.2. 模型建立

参照图片 srow-col-1 中的像素直接写入新图片，待拼接 srow-col-2 的像素经过变换矩阵变换后写入新图片。利用 matlab 中的 imshow 函数观察图像的拼接效果。

5.2.3. 模型求解及检验

使用 for 循环遍历图片的每一个像素按照上述步骤写入到新图，并观察点分布的位置情况，



图像的拼接效果如图所示，可见，最终图像能基本呈现出无缝拼接效果，也就从侧面体现出变换矩阵的计算效果较好。

6. 模型的评价与推广

6.1. 模型优点

1. 拼接效果良好，符合预期。

2. 去除噪声数据的过程简单高效。
3. 一阶线性变换可以处理大部分图片操作。
4. 求解参考回归平面的过程保留了大部分原始数据，且拟合效果良好。
5. 利用图像直观分析数据特点。

6.2. 模型缺点

1. 选取参照点的过程中依赖人工容易出错。
2. 图片变换过程出现了像素损失。
3. 噪声阈值的选取缺乏理论依据。
4. 待拼接图片变换过程较慢。
5. 一阶线性变换对一些复杂操作不适用，需要更高级算法。

6.3. 模型改进方向

1. 改用随机选取参考点，并判断该组参考点是否可行。
2. 改用更高效的算法转换拼接图片。
3. 设计模型量化拼接效果。

6.4. 模型推广

本模型通过研究特征点从而得出整体图片的变换矩阵，该基于线性回归方法的模型同样适用一类通过研究部分性质从而得出整体性质的问题，如天气，市场分析等问题。

7. 心得体会与总结

通过本次实验，我们学会如何通过已有数据和问题来分析其特征，初次遇到此类问题时对庞大数据的特点一无所知，所幸通过已有的数学知识才得以分析问题本质，这让我们深深体会到了数理基础的重要性：基础的大学数学甚至中学数学涉及的内容却能在实际生活中有广泛的应用。另外，充分利用 matlab 强大的功能也是本次实验成功元素之一：通过绘图函数直观展现数据特点、通过回归分析函数快速拟合平面、通过图像处理功能转换拼接图片。最后，本次实验结束也给我们留下更进一步的思考：是否存在更高效快速的方法来处理图片，对于非线性变换的问题又该如何处理。这些都是值得我们下一步研究的问题。

本次实验我们总结了几组方法：

1. 利用数据特点去除噪声点。
2. 利用回归分析处理一类函数多元线性关系问题。

3. 利用像素数组处理新图片。

8. 对本实验问题的设计提出改进意见

本次实验题目深入浅出，涉及知识面广，任务量足。希望在题目中增加 SURF 特征点相关背景，了解其特点及可能存在的问题，方便后续实验进行。

参考文献

- [1]范钦慧，工程力学，北京：清华大学出版社，2005.8，73-74
- [2]叶其孝，沈永欢，数学手册，北京：科学出版社，2006.1 516-518
- [3]陈杰，MATLAB 宝典，北京：电子工业出版社，2010.3，340-370
- [4]姜启源，谢金星，叶俊，数学模型，北京：高等教育出版社，2011.1, 85-115
- [5]隋秀凇，高安邦，实用机床设计手册，北京：机械工业出版社，2010.4，附录
- [6]黄雍检，陶冶，钱祖平，最优化方法—MATLAB 应用，北京：人民邮电出版社，2010.11

附件

附件清单

附件一：排序后的相对噪声值

附件二：实验代码

附件三：最终拼接后的图片

附件

附件一：排序后的相对噪声值

1 . 0↵	45 . 0.448338849282209	89 . 0.83805715939843
2 . 0↵	46 . 0.454692225870758	90 . 0.84824425363103
3 . 0.00000000000045474↵	47 . 0.457655927304359	91 . 0.85578429968359
4 . 0.00904394427016086↵	48 . 0.468318028071053	92 . 0.858536116697678
5 . 0.0175032520071454↵	49 . 0.480872281788152	93 . 0.882206673887140
6 . 0.0184517321865769↵	50 . 0.495999057207428	94 . 0.892670443239013
7 . 0.0240171451405331↵	51 . 0.507108839341072	95 . 0.895316488758454
8 . 0.0266908007981783↵	52 . 0.532804474589284	96 . 0.929807917792459
9 . 0.0640895876254035↵	53 . 0.538740874006408	97 . 0.951114513523635
10 . 0.0993196068588986↵	54 . 0.539549771274551	98 . 0.956868979707451
11 . 0.108916221840900↵	55 . 0.546387475723350	99 . 0.967582467809280
12 . 0.112044215449487↵	56 . 0.547860912756278	100 . 0.969484161918217
13 . 0.127108056824000↵	57 . 0.558511911052165	101 . 0.974739136967855
14 . 0.155239207099612↵	58 . 0.564691120183852	102 . 0.977564185906431
15 . 0.160749449107016↵	59 . 0.598663588659065	103 . 0.978893020952000
16 . 0.219341769599623↵	60 . 0.621078818460319	104 . 0.997276141211387
17 . 0.237591997498384↵	61 . 0.635628633993520	105 . 1.000723542821118
18 . 0.237842006220944↵	62 . 0.643951795611656	106 . 1.004139987114299
19 . 0.240090908977891↵	63 . 0.647231131297758	107 . 1.027347037457968
20 . 0.247096825431754↵	64 . 0.656449494035769	108 . 1.033565621414259
21 . 0.249962925728823↵	65 . 0.662351801772047	109 . 1.036602205015709
22 . 0.256548729548285↵	66 . 0.663871440950516	110 . 1.056357972371648
23 . 0.266493672825618↵	67 . 0.671122364922212	111 . 1.060321140087138
24 . 0.279661023212384↵	68 . 0.673681023159134	112 . 1.071228369461109
25 . 0.285198067185547↵	69 . 0.676197144342268	113 . 1.105037257578268
26 . 0.311052837650550↵	70 . 0.676909152090502	114 . 1.106146365146299
27 . 0.324268830927622↵	71 . 0.679834108165096	115 . 1.107337603344609
28 . 0.334469313883801↵	72 . 0.696529795971173	116 . 1.111105792197588
29 . 0.345338937793258↵	73 . 0.701345290601012	117 . 1.113565165502688
30 . 0.350519149822958↵	74 . 0.710484278888998	118 . 1.119847430825088
31 . 0.353481431111959↵	75 . 0.712746150682051	119 . 1.120182506494768
32 . 0.357210891245359↵	76 . 0.714847089172736	120 . 1.123436726624509
33 . 0.360560005335628↵	77 . 0.716261074571548	121 . 1.127221410998579
34 . 0.367904234394473↵	78 . 0.724930727132914	122 . 1.129958944222159
35 . 0.383554148989788↵	79 . 0.729622832407131	123 . 1.142364006698249
36 . 0.386384726590222↵	80 . 0.735803961416195	124 . 1.144614641685629
37 . 0.387860312691828↵	81 . 0.747803199277314	125 . 1.162029158824009
38 . 0.409375094016468↵	82 . 0.764700666404451	126 . 1.163111692619179
39 . 0.409396623948624↵	83 . 0.773761156376850	127 . 1.171951119503099
40 . 0.417817547031063↵	84 . 0.776183330358208	128 . 1.174653422886659
41 . 0.419321085933007↵	85 . 0.785651843550113	129 . 1.181514471928049
42 . 0.426013071770058↵	86 . 0.792716619461544	130 . 1.184389515818119
43 . 0.439313689197661↵	87 . 0.803200139548608	131 . 1.186998687351899
44 . 0.445834612677572↵	88 . 0.824261473660044	132 . 1.190977389805549

133 .	1.19323746348164	177 .	1.62964316495822	221 .	2.64873354277097
134 .	1.20179108740126	178 .	1.69951107246015	222 .	2.67216905147325
135 .	1.20409155480320	179 .	1.77345359227093	223 .	2.68622174100210
136 .	1.21400671411516	180 .	1.77658317681107	224 .	2.71164071577027
137 .	1.21681794477036	181 .	1.77711057049055	225 .	2.72498750648037
138 .	1.21875226203997	182 .	1.77893921398186	226 .	2.78614482444073
139 .	1.22109623418169	183 .	1.78344635919893	227 .	2.80220611151458
140 .	1.22644629363458	184 .	1.80815226361301	228 .	2.84093610568470
141 .	1.23473473294894	185 .	1.85042449825141	229 .	2.84343830779153
142 .	1.24035396020804	186 .	1.85800243830681	230 .	2.84362481169683
143 .	1.24431896342912	187 .	1.86620528668618	231 .	2.85028681364224
144 .	1.25216646454851	188 .	1.86911044824387	232 .	2.86003891734435
145 .	1.25225559245246	189 .	1.91793839999355	233 .	2.90884233363317
146 .	1.25736109467675	190 .	1.91795687999479	234 .	2.98519944507666
147 .	1.26051781742740	191 .	1.95182838801225	235 .	3.00114640061952
148 .	1.26150428481537	192 .	1.96821683101666	236 .	3.03459931289149
149 .	1.26196162544261	193 .	1.96905039543617	237 .	3.14603636990955
150 .	1.26297703693467	194 .	1.97093744693893	238 .	3.24726628396093
151 .	1.30418326286986	195 .	1.98076708124609	239 .	3.36799384588016
152 .	1.30879609831027	196 .	1.98158109073074	240 .	3.39101851862210
153 .	1.32068038487228	197 .	2.07303297674366	241 .	3.48559496228586
154 .	1.32074005216941	198 .	2.10828911012413	242 .	3.54214232788809
155 .	1.32443483449720	199 .	2.15755081856400	243 .	3.79562180318567
156 .	1.36893742604434	200 .	2.20014609486611	244 .	3.88744200622091
157 .	1.37878366307200	201 .	2.22340642415134	245 .	4.01053169453121
158 .	1.37953580408475	202 .	2.24762048126172	246 .	4.02240013516257
159 .	1.38634326133047	203 .	2.25942526234644	247 .	4.11151456912808
160 .	1.39238240198210	204 .	2.29306888700899	248 .	4.15058823001414
161 .	1.40115293532858	205 .	2.30530404749902	249 .	4.43828765712419
162 .	1.40358228840023	206 .	2.33148739184844	250 .	4.47586662702065
163 .	1.44017136436764	207 .	2.34141131320826	251 .	5.30371794477037
164 .	1.46725862825542	208 .	2.35764887715732	252 .	5.54313297674366
165 .	1.51030933277661	209 .	2.36462632476218	253 .	6.77203862891656
166 .	1.53051604201050	210 .	2.38870269821177	254 .	100.851694793781
167 .	1.53347926299080	211 .	2.39792993779315	255 .	116.978613508781
168 .	1.53917038536133	212 .	2.41228820132483	256 .	117.682162553061
169 .	1.54158622228124	213 .	2.42216856491064	257 .	120.525887475723
170 .	1.54455538261050	214 .	2.43211275993758	258 .	120.551165621414
171 .	1.57889575511808	215 .	2.46554962057098	259 .	120.560457146699
172 .	1.59706443238247	216 .	2.48714212719051	260 .	120.812928355964
173 .	1.60045201264393	217 .	2.57124495534686	261 .	120.950253381354
174 .	1.60066886874984	218 .	2.58592514238535	262 .	121.225287657124
175 .	1.61591895445576	219 .	2.60452310103051	263 .	121.308323101031
176 .	1.62248982221604	220 .	2.62556903886230	264 .	121.738030902176

265 .	121.8265347056804	
266 .	122.2946589823394	
267 .	122.4856461506824	
268 .	122.6487780774624	
269 .	122.6975388492824	
270 .	122.8987724943614	
271 .	122.9525509884234	
272 .	124.7152569133844	
273 .	125.7861388492824	
274 .	126.0467877882144	
275 .	128.6688388492824	
276 .	129.5623386289174	
277 .	220.3474121789044	
278 .	230.5554616208994	
279 .	236.1572416442104	
280 .	237.1175457792834	
281 .	239.9907619756374	
282 .	240.5784695803074	
283 .	241.7529266491574	
284 .	246.2823877882144	
285 .	250.6865439093694	
286 .	252.4270371475114	
287 .	354.5426627524234	
288 .	365.1076877882144	
289 .	369.9170867243574	
290 .	474.0692190782284	
291 .	475.1294116228354	
292 .	482.9100854454774	
293 .	487.2302669523034	
294 .	490.7186590763804	
295 .	490.9782038376584	
296 .	495.2684044196524	
297 .	589.3857637618094	
298 .	614.6173177665494	
299 .	657.2550751130434	
300 .	707.8564852697894	
301 .	716.9181805184844	
302 .	736.4902042829384	
303 .	796.9707947937824	
304 .	803.0458947937824	
305 .	862.9263042829384	
306 .	1091.652394793784	309 . 1786.005623967974
307 .	1283.647217555554	310 . 2156.257176576924
308 .	1735.390972569504	311 . 2213.915534363704

附件二：实验代码

```
function []=splic()
%拼接两张图片

[reg2 reg3] = mregress(); %获取变换矩阵
Fx=@(x,y)reg2(1) + reg2(2)*x + reg2(3)*y;
Fy=@(x,y)reg3(1) + reg3(2)*x + reg3(3)*y;

filename1 = 'srow-col-1.jpg'; %载入图片
filename2 = 'srow-col-2.jpg';
imgA =imread(filename1);
imgB =imread(filename2);

imgC=uint8(zeros(1050,3080,3)); %创建新图

for i=1:1263
    for j=1:1451
        xx=round(Fx(j,i));
        yy=round(Fy(j,i));
        if (xx>0&&yy>0)
            imgC(yy,xx,1:3)=imgB((i),(j),1:3); %将'srow-col-2.jpg'中的坐标
变换后写入
        end
    end
end

end
for i=1:1429
    for j=1:1920
        imgC(i,j,1:3)=imgA((i),(j),1:3); %写入参照照片'srow-col-1.jpg'
    end
end

end

imshow(imgC)
end

function [reg2 reg3]=mregress()
%对数据求出相应回归平面
```

```

load locA.txt % 导入第 1 个图片文件的 SURf 特征点
load locB.txt % 导入第 2 个图片文件的 SURf 特征点

y=locA([154 215 237],1); %经观察这三个点在所求回归平面上
x1=locB([154 215 237],1);
x2=locB([154 215 237],2);

X=[ones(3,1),x1,x2];

reg1=regress(y,X); %调用线性回归函数初步估计回归平面

F1=@(x,y)reg1(1) + reg1(2)*x + reg1(3)*y;

y=locA(:,1);
x1=locB(:,1);
x2=locB(:,2);

ax1=x1;
ax2=x2;
for i=1:311
    ay(i)=abs(F1(ax1(i),ax2(i))-y(i)); %计算残差
end

[ay,ind]=sort(ay); %对残差排序
ax1=ax1(ind);
ax2=ax2(ind);
y=y(ind);
%ans=[ax1,ax2,ay']

theta=2;
m=max(find(ay<theta)); %去除残差大于阈值的点
%m 为剩余点个数, 共 196 个
y=y(1:m,1);
ax1=ax1(1:m,1);
ax2=ax2(1:m,1);

X=[ones(m,1),ax1,ax2];

reg2=regress(y,X); %对剩余的点重新计算回归平面 (求出 x' 的变换矩阵系数)
%-----
y=locA(ind,2);
y=y(1:m,1);

```

```
reg3=regress(y,X); %对剩余的点重新计算回归平面（求出  $y'$  的变换矩阵系数）  
%-----  
end
```

附件三：最终拼接后的图片

