

机器视觉技术课程大作业 Part1

一、实验名称：针尖刺入识别

二、实验目的

利用 matlab 编写传统算法代码，实现斑马鱼针尖刺入识别。

三、实验原理

通过观察数据集图像组和简单图像处理后的结果，得到了数种可能的检测思路。以下简单描述，其中第一种方法为本次作业成功实现的方法。

- 1、差分图像灰度值：使用对齐后的图像进行差分，在阈值化后，统计灰度值，根据阈值判断。
- 2、差分图像针尖识别：同 1，利用直线检测对刺入成功图像明显的细长针体进行识别。
- 3、差分图像直线识别：同 1，利用直线检测对刺入失败图像刺入处堆积形成的众多直线进行识别。
- 4、图像轮廓完整性：仅利用刺入后图像，通过膨胀腐蚀、阈值化等操作，放大刺入失败图像刺入堆积处。刺入成功的图像会是完整的肾状体，而刺入失败的会凹进去一部分。
- 5、特征点匹配数量：通过特征检测得到特征点，统计刺入前后图像特征点匹配的数量。由于刺入失败带来更多的形变，其匹配数量会少于刺入成功的。

实验过程中，对以上方法都进行了尝试，最终效果较好、完成度较高的是第一种。

四、实验步骤

- 1、数据图像组输入：设定了输入格式，详见 txt 文件。在带入判断函数后，首先将图像转为灰度图。
- 2、surf 特征检测：在获得 surf 特征点时，对特征阈值、特征点数量进行了调参，主要尝试增多特征点以加强下一步骤中的匹配程度，但结果改变较小，默认参数的结果已达到要求。
- 3、特征点匹配并对齐图像：通过匹配特征点，获得刺入前后图像的变换方式，并由此将刺入后的图像向刺入前的图像对齐。这里对变换方式这一参数进行了调试，`affine`（仿射变换）、`similarity`（相似变换）、`rigid`（刚性变换）中，第三者的效果最好，其他两者都会出少数扭曲状况。
- 4、填充背景：对对齐后的图像变黑的背景进行填充。这里遍历刺入后的图像，选择其中像素值为 0 的像素点，用刺入前图像对应位置的像素点的像素值进行填充。这里可能会填充非背景点，但影响不大。
- 5、差分：用刺入前图像减去对齐后的刺入后图像，获得差分图像。
- 6、阈值化：对差分图像进行阈值化，以突出刺入带来的变化。这里并没有选用已有的阈值化函数，而是通过对差分图像像素值的观察，自行选择了合适的阈值。
- 7、判断：统计像素值为 255 的像素点个数，并根据阈值判断是否刺入成功。刺入失败带来的挤压，会在差分图像中表现出来，这造成成功与失败间像素值分布的较大差距。这个阈值是在反复调试中获得的，与阈值化的阈值是相匹配的。
- 8、输出：纵向输出各测试图像组的结果，其中 1 为刺入成功，0 为刺入失败。

五、代码

main.m

```

clc;clear all;close all;

parent = 'testimg';
parentPath = dir(parent);
len = size(parentPath);
result = zeros(len-2);
for i = 1:len-2
    childPath = fullfile(parent, parentPath(i+2).name);
    img1 = imread(fullfile(childPath, '1'), 'jpg');
    img2 = imread(fullfile(childPath, '2'), 'jpg');
    result(i) = test(img1, img2);
end
disp(result')

test.m
% 定义函数 test, 参数为图像 img1、img2
function result=test(img1,img2)

% 转为灰度图
img1 = rgb2gray(img1);
img2 = rgb2gray(img2);

% surf 特征检测, 匹配并对齐图像
points1 = detectSURFFeatures(img1);
points2 = detectSURFFeatures(img2);
[features1, validPoints1] = extractFeatures(img1, points1);
[features2, validPoints2] = extractFeatures(img2, points2);
indexPairs = matchFeatures(features1, features2);
matchedPoints1 = validPoints1(indexPairs(:, 1));
matchedPoints2 = validPoints2(indexPairs(:, 2));
[tform, inlierIdx] = estimateGeometricTransform2D(matchedPoints2,
matchedPoints1, 'rigid');
%{
% 匹配图像
inliers1 = matchedPoints1(inlierIdx, :);
inliers2 = matchedPoints2(inlierIdx, :);
figure
showMatchedFeatures(img1, img2, inliers1, inliers2, 'montage');
%}
img2_new = imwarp(img2, tform, 'OutputView', imref2d(size(img1)));
%{
% 对齐后的图像
figure
imshow(img2_new)

```

```
%}
```

% 填补移动后剩下的背景，并差分，**back** 统计黑色像素数量（包括移动后剩下的背景和原来就是黑色的像素）

```
[row,col] = size(img1);
back = 0;
for i=1:row
    for j=1:col
        if img2_new(i,j)==0
            img2_new(i,j)=img1(i,j);
            back = back + 1;
        end
    end
end
img_12 = img1 -img2_new;
%{
% 差分图像
figure
imshow(img_12)
%}
```

% 阈值化，**counts** 统计白色像素数量

```
counts = 0;
for i=1:row
    for j=1:col
        if img_12(i,j)>70
            img_12(i,j)=255;
            counts = counts + 1;
        else
            img_12(i,j)=0;
        end
    end
end
%{
% 阈值化图像
figure
imshow(img_12)
%}
```

```
if counts>5000
    result = 0;
else
    result = 1;
end
```

end

六、实验结果展示

1、整体数据集测试结果

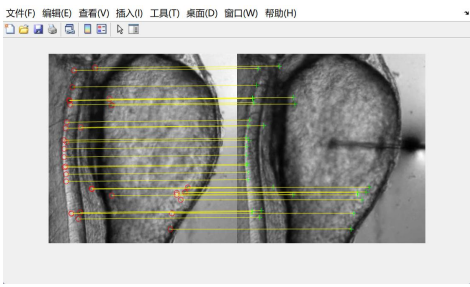
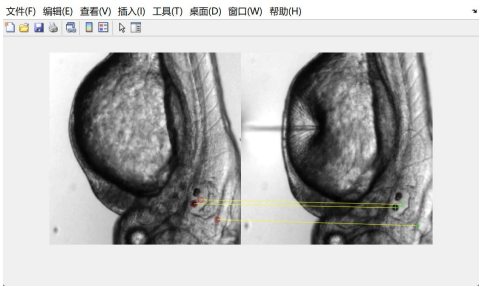
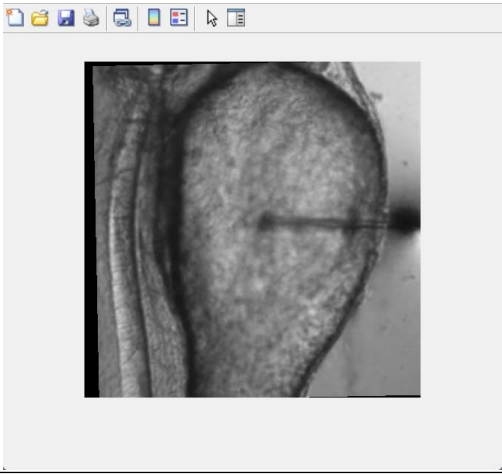
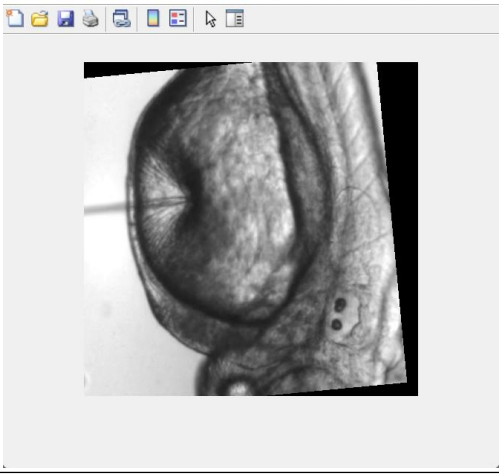
将数据集的成功与失败样例合并，成功在前，失败在后，得到 `testing` 文件夹所示的测试样例。在运行后，得到了如下结果：

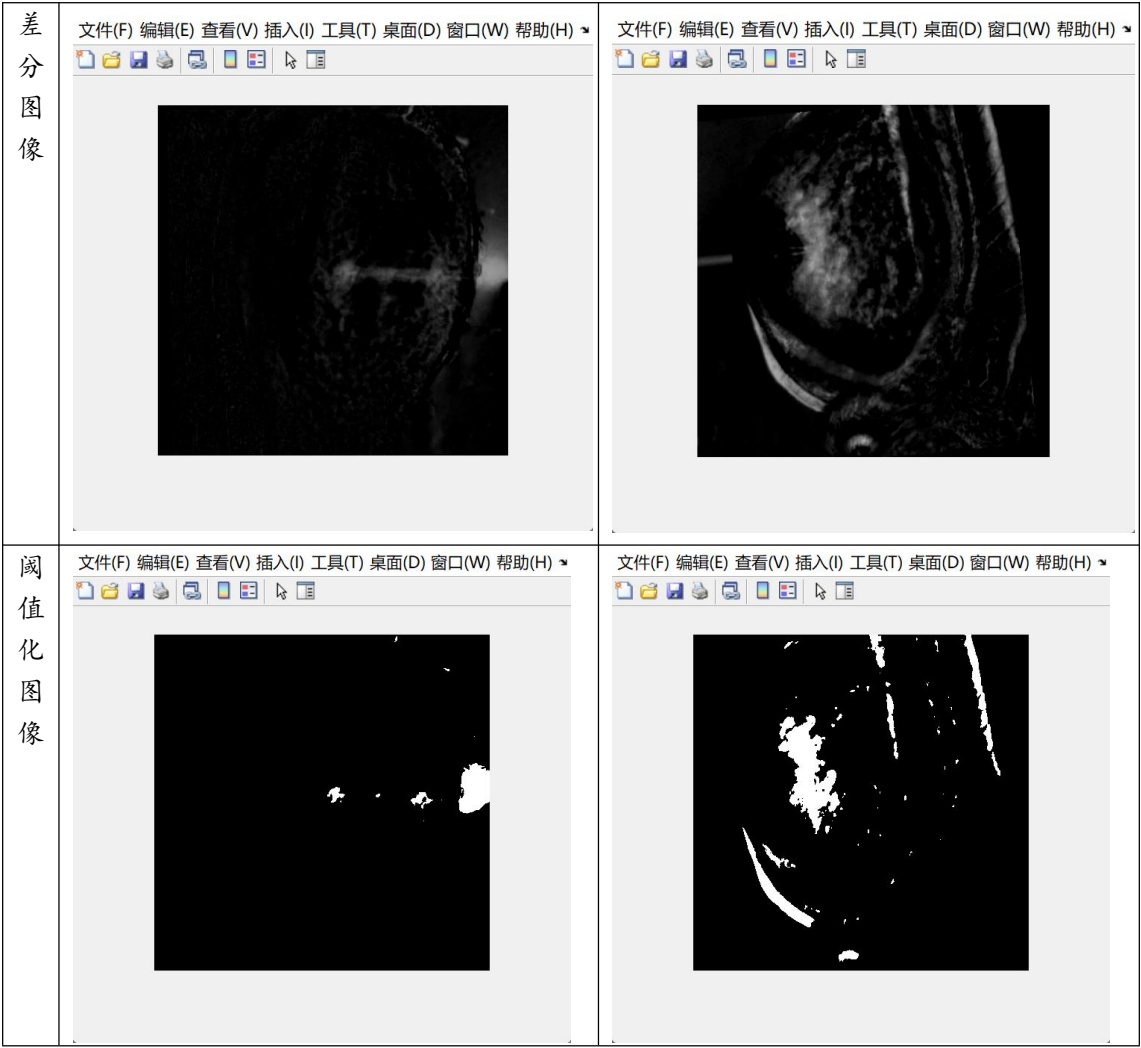
```
0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

可以发现，除第一个测试图像组外，其他图像组都得到了正确的结果。而第一个测试组失败的原因，在于刺入前后图像的亮度有差别。

2、数据图像组的处理过程展示

以下以第 2 组和第 21 组图像组的图像处理过程为例

处 理	第 2 组图像	第 21 组图像
特 征 点 匹 配 图 像		
对 齐 后 的 图 像		



通过对比可以发现，刺入失败的图像组的匹配特征点要远远少于刺入成功的，而由于挤压，在阈值化后，刺入失败的差分图像的像素值为 255 的像素点要远多于刺入成功的。除去个例，本识别方法在题设背景下的正确率可以得到保证。

七、分析

本代码突出了刺入前后图像的灰度值变化,然而这需要保证前后的图像是在相同状况下进行的拍照。利用特征匹配对齐图像的方法，可以减小空间、形状变化带来的影响，然而如果曝光度等因素不同，还是会对识别结果造成较大的影响。