



上海工程技术大学

实 验 报 告

姓名 班级 学号 实验日期

课程名称 数字内容安全

指导教师

成绩

实验名称：数字图像复制粘贴检测

一、实验目的

掌握数字图像复制粘贴篡改检测的原理，用 MATLAB 语言编程、运行。

二、实验内容

使用 MATLAB 编程实现数字图像复制粘贴检测

三、实验代码及运行结果

sift.m 文件代码如下：

本代码主要实现：读取一幅图像，并返回它的 SIFT 关键点信息

```
2 function [imageRGB, descriptors, locs] = sift(imageFile)
3 %     imageFile: 图片文件名 %...%
8 -     image = imread(imageFile);
9 -     imageRGB = image;
10 - %如果原图像是彩色图像的话，把彩色图像转化为灰度图像
11 -     if size(image,3)
12 -         image = rgb2gray(image);
13 -     end
14 - %获取图像的尺寸
15 -     [rows, cols] = size(image);
16 - % 转化成PGM文件,为了能使"keypoints"可以执行
17 -     f = fopen('tmp.pgm', 'w');
18 -     if f == -1
19 -         error('不能读取tmp.pgm文件。');
20 -     end
21 -     fprintf(f, 'P5\n%d\n%d\n255\n', cols, rows);
22 -     fwrite(f, image, 'uint8');
23 -     fclose(f);
24 - % 调用"keypoints"执行
25 -     if isunix
26 -         command = '!./sift ';
27 -     else
28 -         command = '!siftWin32 ';
```



上海工程技术大学

实验报告

姓名 班级 学号 实验日期

课程名称 数字内容安全

指导教师

成绩

```
29 — end
30 — command = [command ' <tmp.pgm >tmp.key'];
31 — eval(command);
32 — % 打开tmp.key并检查其头文件
33 — g = fopen('tmp.key', 'r');
34 — if g == -1
35 —     error('不能打开tmp.key文件。');
36 — end
37 — [header, count] = fscanf(g, '%d %d', [1 2]);
38 — if count ~= 2
39 —     error('无效的“keypoint”文件。');
40 — end
41 — num = header(1);
42 — len = header(2);
43 — if len ~= 128
44 —     error('关键点描绘子长度不对，应该是128!');
45 — end
46 — % 使用已知的图像尺寸创建两幅输出图像
47 — locs = double(zeros(num, 4));
48 — descriptors = double(zeros(num, 128));
49 — % 分析“tmp.key”
50 — for i = 1:num
51 —     [vector, count] = fscanf(g, '%f %f %f %f', [1 4]); %行数、列数、尺度半径、方向
52 —     if count ~= 4
53 —         error('无效的“keypoint”文件格式');
54 —     end
55 —     locs(i, :) = vector(1, :);
56 —
57 —     [descrip, count] = fscanf(g, '%d', [1 len]);
58 —     if (count ~= 128)
59 —         error('无效的“keypoint”文件值');
60 —     end
61 —     % 单位化每个输入向量
62 —     descrip = descrip / sqrt(sum(descrip.^2));
63 —     descriptors(i, :) = descrip(1, :);
64 — end
65 — %关闭文件
66 — fclose(g);
67 — delete tmp.key
68 — delete tmp.pgm
69 —
```

copyMoveDetection.m 文件代码如下:



上海工程技术大学

实验报告

姓名 班级 学号 实验日期

课程名称 数字内容安全

指导教师

成绩

本代码主要实现：检测同一副数字图像中复制粘贴的区域

```
1 function copyMoveDetection(imageName, threshold)
2     [im, des, loc] = sift(imageName);
3     if nargin<2
4         threshold= 0.4;
5     end
6     match = ones(1, size(des,1));
7     for i = 1 : size(des,1)
8         distance_I_other = distanceVectorMatrix(des(i,:), des);
9         [vals, indx] = sort(distance_I_other);
10        if (vals(2) < threshold * vals(3))
11            match(i) = indx(2);
12        else
13            match(i) = 0;
14        end
15    end
16    figure('Position', [100 100 size(im,2) size(im,1)]);
17    colormap('gray');
18    imagesc(im);
19    hold on;
20    for i = 1: size(des,1)
21        if (match(i) > 0)
22            line([loc(i,2) loc(match(i),2)], [loc(i,1) loc(match(i),1)], 'linestyle', 'dashed');
23        end
24    end
```

distanceVectorMatrix.m 文件代码如下：

本代码的主要功能：返回行向量 vector 与矩阵 matrix 中每一行向量的欧几里得距离

```
1 function result = distanceVectorMatrix(vector, matrix)
2     lengthVector = length(vector);
3     [rowMatrix, colMatrix] = size(matrix);
4     if(lengthVector ~= colMatrix)
5         error('函数输入参数尺寸不正确，请检查后重新输入!');
6     end
7     colVec = ones(rowMatrix, 1);
8     matTemp = (colVec*vector-matrix).^2;
9     result = sqrt(sum(matTemp, 2));
```



上海工程技术大学

实验报告

姓名 班级 学号 实验日期

课程名称 数字内容安全 指导教师 成绩

运行结果:

输入:

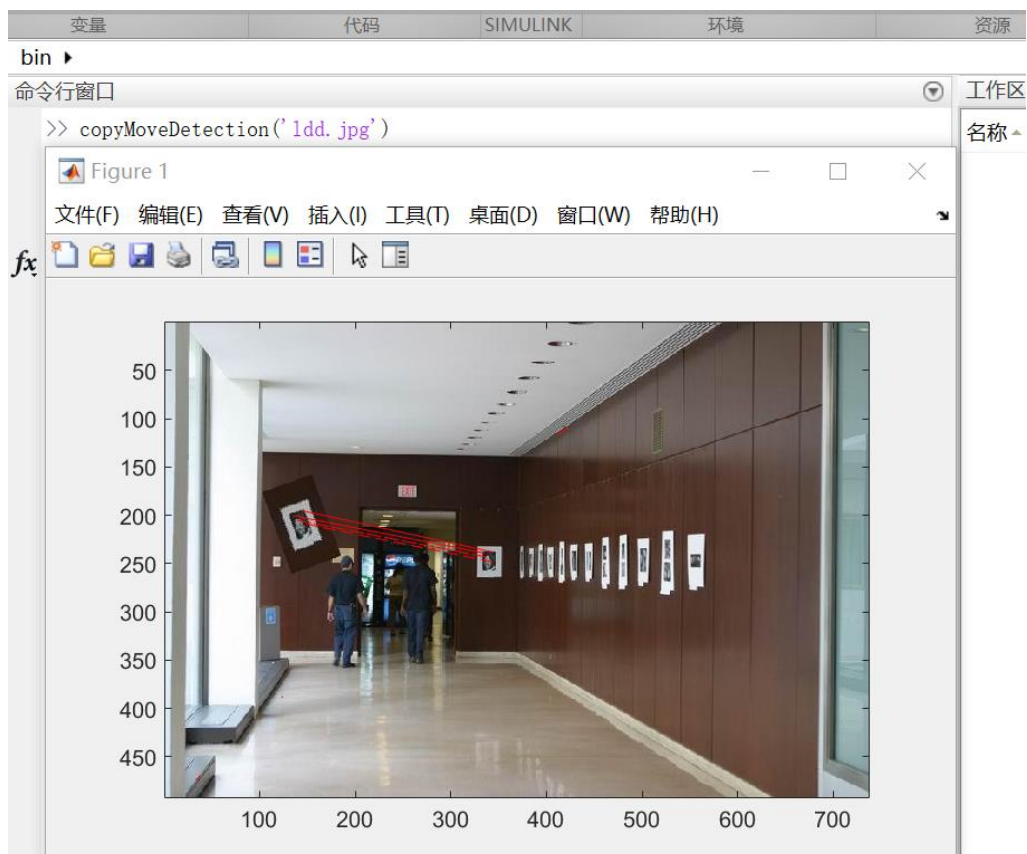
try

JPG 图片文件)



宽度: 737 高度: 492

输出:





上海工程技术大学

实 验 报 告

姓名 班级 学号 实验日期

课程名称 数字内容安全

指导教师

成绩

四、实验总结/心得

【1】、通过本次实验，我掌握了数字图像复制粘贴篡改检测的原理，其主要过程是：

1、提取 SIFT 关键点和特征描述子

提取 SIFT 的关键点： $P = (P_1, P_2, \dots, P_n)$ 及其对应的特征描述子： $F = (F_1, F_2, \dots, F_n)$

2、关键点匹配

根据各特征描述子之间的欧氏距离获取相互匹配的关节点。

(1)、计算描述子 F_1 与其它 $(n-1)$ 个描述子之间的距离（一般为欧氏距离），最

后得到： $D = (D_1, D_2, \dots, D_{(n-1)})$

(2)、将向量 D 按从小到大的顺序排列，之后得到：

$D' = (D'_1, D'_2, \dots, D'_{(n-1)})$

如果 $\frac{D'_1}{D'_2} < T$ ，那么可以得到与描述子 F_1 匹配的关键点 P_j （此时的 D'_1 对应描述子

F_1 与描述子 F_j 之间的距离），并将关键点 P_1 加入到点集 A 中，将关键点 P_j 加入到点集 B 中；否则：进行步骤（3）。

(3)、对于描述子 $F_2 \sim F_n$ ，重复步骤（1）、（2）。

3、将点集 A 与点集 B 中对应的点用线连起来。

【2】程序运行时用到了 siftWin32.exe，需将其放在 MATLAB 工程文件中的 bin 目录下的 win32 中。否则会报错：'siftWin32' 不是内部或外部命令，也不是可运行的程序