数字图像处理三级项目

的算法设计与实现

学号： 姓名： 李 .

学号： 姓名： 张 .

学号： 姓名： 王 .

学号： 姓名： 张紫 .

指 导 教 师：

2021年10月

指纹图像的锐化处理

李\*\*，张\*\*，王\*，张紫\*

（燕山大学 理学院）

**摘要**

指纹是人类手指末端指腹上由凹凸的皮肤所形成的纹路，是人类进化过程中自然形成的，目前尚未发现有不同的人拥有相同的指纹，所以每个人的指纹都是独一无二的。近30年来，自动指纹识别技术取得了较大的成功，但是由于各种噪声仍然存在，比如伤疤、汗渍、污迹、扭曲等等，直接影响到指纹识别的效率。

所以指纹图像的识别也变得非常具有意义，但是在采集图像的过程中，通过传感器等方式获得的指纹往往是模糊的，识别率较低，因此，实际过程中对指纹提取之前，要对原始指纹图像进行四个步骤的处理，分别是平滑处理，锐化处理，二值化处理与细化处理，这四个处理方法主要是去除图像中的噪音，使指纹图像变成一幅清晰的点线图，为了解决锐化纵向的缺陷，我们对图像进行了增强处理，更加清晰地辨认指纹。

**前言**

指纹识别具有广阔的发展背景，随着科学技术的不断发展，指纹识别应用于我们的日常生活中，例如手机指纹解锁，指纹门禁，指纹考勤，通过指纹逮捕犯罪嫌疑人等，在越来越多的领域展现出来，所以显得指纹尤为重要。

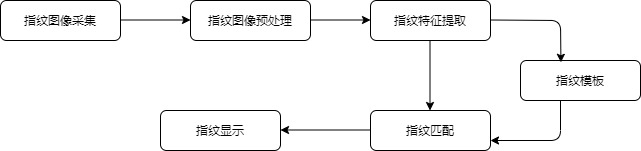
但指纹识别的核心技术仍存在许多尚未解决的问题，尤其对残缺、污损指纹图像进行识别的适应性方面不能令人满意，所以要通过读取初始指纹图像，设计程序，实现指纹图像的锐化增强，使指纹的纹理更加清晰，便于识别。其基本操作为：读取指纹图像，指纹图像的灰度处理，指纹图像平滑处理 ，指纹图像的腐蚀处理，指纹图像的锐化处理，指纹图像的二值化，指纹图像纹理的细化处理，指纹图像的增强等。再经过一系列图像处理过后，我们所得到的指纹图像边缘比原始图像更加清晰，轮廓也更加明了，可以清晰地辨认出一个人的指纹，可以为今后的指纹识别发展提供一些帮助，应用到实际生活中。

**研究报告正文**

**项目内容介绍：**

算法设计及问题分析与解决思路：

指纹识别系统大致包括以下的流程：

****

1.指纹图像的读取

将通过传感器或者别的方式得到的指纹读取到matlab中，如bmp,jpg等格式的图片文件。

通过matlab实现:

I=imread('文件路径+图像路径.jpg');

代码中I = imread('.\zhiwen\101\_1.tif');体现出来。

2.指纹图像灰度化处理

数字图像可分为灰度图像和彩色图像。通过灰度化处理和伪彩色处理，可以使伪 彩色图像与灰度图像相互转化；灰度化就是使彩色的 R，G，B分量值相等的过程。

通过matlab实现:

l=rbg2gray (1);

3.图像的平滑处理

但在原始指纹图像采集的时候，由于具体的拍摄环境和成像方法等因素的影响，指纹图像中会出现或多或少的由于干扰而含有的噪声，而图像的噪声也来自于多方面，其中具体包括了来自系统外部的干扰（如电磁波或经电源窜进系统内部的外部噪声）以及来自系统内部的干扰（如摄像机热噪声、电器机械运动而产生的抖动噪声噪声等内部噪声），而这会造成待识别指纹图像的不规范性。因此，在对指纹图像进行特征提取之前，首先对原始指纹图像进行平滑处理。

中值滤波作为一种非线性处理技术，在实际运算中不需要获取图像的统计特性，而且在一定情况下，中值滤波可以克服线性滤波器造成的图像细节模糊，故我们采用中值滤波对原图像进行平滑处理，同时采用维纳滤波作为对照平滑处理。

中值滤波的matlab实现:

I1=medfilt2(I,[3,3]);

figure,imshow(I1);

title("中值滤波");

维纳滤波的matlab实现:

I1=wiener2(I,[3,3]);

figure,imshow(I1);

title("维纳滤波");

直方图均衡matlab实现:

[I1,T]=histeq(I1,256);

figure,imshow(I1);

title("直方图均衡");

处理得到图像如下：（后采用直方图均衡法进一步去噪）



图（1）原始图像 图（2）中值滤波图像



图（3） 维纳滤波图像 图（4） 直方图均衡图像

放大图片后可以看到，原图像在经过中值滤波后，其边缘比经过维纳滤波后的图像更为清晰。

4.图像的锐化处理

图像平滑处理会使得图像局部细节模糊，故接下来我们对已经平滑处理过的图像进行锐化处理，提升图像的边缘清晰度。在锐化处理之前，先对得到的平滑处理后的图像进行腐蚀处理，使得指纹的纹路更细，以便于后序操作。

(1)指纹图像的腐蚀处理

指纹图像腐蚀运算后使得指纹纹路更细， 便于后续的处理，但指纹图像显得模糊，但是接下来我们将对指纹图像进行锐化，使得图像边缘更加清晰。

灰度腐蚀的 matlab 实现程序：

s=ones(3,3);

I2=imerode(I1,s);

(2)指纹图像的sobel处理

图像锐化即对图像进行处理，使图像的边缘变得鲜明。目的是为了突出图像 的边缘信息，加强轮廓特征，以便于人眼的观察和机器的识别对比。

这里我们采用的是Sobel 算子对图像增强。

Matlab实现程序如下:

H=fspecial('sobel');

Ns=filter2(H,I1);

(3)指纹图像的二值化

对指纹图像进行二值化，其基本要求就是二值化后的图像能真实的再现原指纹，具体要求为：

1. 脊线中不出现空白
2. 二值化后的脊线基本保持原来的指纹特征。
3. 指纹的纹线不应有太多的间断和相连。
4. 指纹纹线间的间距应大致相同。

二值化MATLAB实现：

I2=imbinarize(I1);

Ns2=filter2(H,I1);

figure,imshow(Ns2);

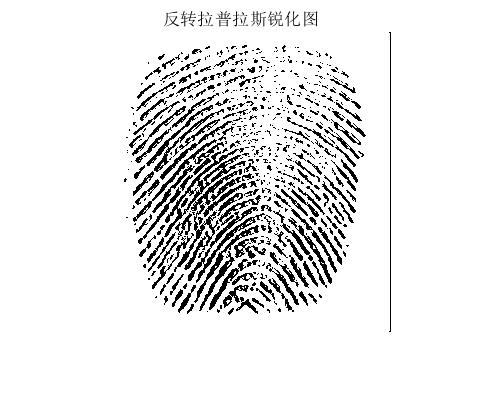
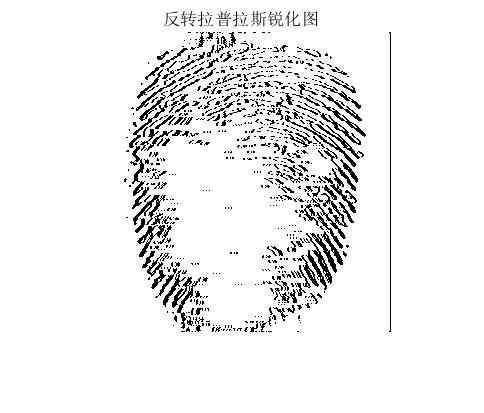
在锐化处理中，我们分别采用sobel算子与拉普拉斯算子对指纹图像进行处理，从而得到不同的效果图（同样采用在锐化前是否进行对指纹图像进行二值化操作作为对比变量）。

下图为未进行二值化操作的sobel锐化效果图与进行了二值化操作的sobel锐化效果图（均采用以白色为底，指纹为黑色进行显示）

图（5） 未二值化sobel锐化图像 图（6） 二值化sobel锐化图像

同样的下图为未进行二值化操作的拉普拉斯锐化效果图与进行了二值化操作的拉普拉斯锐化效果图（均采用以白色为底，指纹为黑色进行显示）

图（7）未二值化拉普拉斯锐化图像 图（8）二值化拉普拉斯锐化图像

可以看到，由于sobel算子在纵向上的锐化效果由于拉普拉斯算子的这一特性，在对指纹效果图的锐化处理中，用sobel算子得到的锐化图像的边缘纹理清晰度与轮廓特征明显优于使用拉普拉斯算子锐化得到的图像，并且处理后图像的像素点对比度更明显。

以上步骤完成了指纹原始图像的预处理，指纹图像的特征提取是指纹识别的关键，而指纹匹配通常基于细节点匹配。指纹特征提取是从细化后的指纹图中得到细节特征点（即端点和分叉点），此特征点含有大量的伪特征，既耗时又影响匹配精度。以下采用边缘去伪和距离去伪，使得特征点去伪前后减小了近1/3，然后提取可靠特征点信息，以便实现指纹匹配。

指纹图像的细化

细化又称骨骼化，是指在不影响图像的拓扑关系的情况下，将图像中宽度大于一个像素的线条转变为只有一个像素宽度的图像的处理过程。

细化的MATLAB实现：

[a,b]=size(N11);

for i=2:a-1

for j=2:b-1

if N11(i,j)==0

if (N11(i-1,j)==0&&N11(i,j+1)==0)||(N11(i-1,j)==0&&N11(i,j-1)==0)||(N11(i+1,j)==0&&N11(i,j-1)==0)||(N11(i+1,j)==0&&N11(i,j+1)==0)

N11(i,j)=1;

else

N11(i,j)=0;

end

end

end

end

下图是细化处理后得到的指纹图像：



图（9） 细化处理后得到的指纹图像

在由上面拉普拉斯算子得到的锐化图像中，由于纵向锐化效果的缺陷，在纵向上存在一定数量的不连续残缺点，故我们考虑在将原图与经过拉普拉斯算子锐化后的两幅图像进行配准后再对图像进行锐化增强，得到拉普拉斯算子锐化图像的强化图像如下：



图（10）拉普拉斯改进图像

对该图像进行同样的细化处理，得到下图：



图（11） 拉普拉斯改进图像细化后图像

与上面的拉普拉斯算子得到的锐化图像对比，其边缘的离散点连续性与指纹图像的特征可读取性得到了明显提高。

**结语**：

由以上sobel算子和拉普拉斯算子的图像对比可以看出，两种算子锐化的效果各有各的特点，sobel 算子在纵向上的锐化效果比拉普拉斯算子的效果好，而拉普拉斯算子在横向方面的效果更优。再通过未二值化和二值化的对比来看，sobel 算子二值化后的图像，指纹纹理更加清晰，相反拉普拉斯算子的二值化后的图像出现脊线空白的现象。将指纹图像细化后，发现指纹纹理有部分仍有空白，纵向上存在一定数量的不连续残缺点，不清晰，为了解决这一问题，我们将对图像进行锐化增强，再对其细化处理，通过改进化的图像看来，指纹图像得到明显增强且纹理清晰。

**程序源代码**：

clear all;

close all;

I = imread('.\zhiwen\101\_1.tif');%103\_4

figure,imshow(I);

title("原始图像");

%I = rgb2gray(I);

%I = double(I)./255;

[m,n]=size(I);

%维纳滤波

%I1=wiener2(I,[3,3]);

%figure,imshow(I1);

%title("维纳滤波");

%中值滤波

I1=medfilt2(I,[3,3]);

figure,imshow(I1);

title("中值滤波");

%直方图均衡

[I1,T]=histeq(I1,256);

figure,imshow(I1);

title("直方图均衡");

% garbo滤波

%二值化

%I1=imbinarize(I1);

%腐蚀

%s=ones(3,3);

%I2=imerode(I1,s);

% 图像锐化

%sobel算子

H=fspecial('sobel');

Ns=filter2(H,I1);

figure,imshow(Ns);

title("未二值化进行sobel锐化")

Nsf=imcomplement(Ns);

figure,imshow(Nsf);

title("反转sobel锐化")

%二值化

I2=imbinarize(I1);

Ns2=filter2(H,I1);

figure,imshow(Ns2);

title("二值化进行sobel锐化")

Ns2f=imcomplement(Ns2);

figure,imshow(Ns2f);

title("反转sobel锐化")

%拉普拉斯算子

h=[0 1 0,-4 1 0,0 1 0];

N1=convn(I1,h,'same');

figure,imshow(N1);

title("未二值化进行拉普拉斯锐化")

N11=imcomplement(N1);

figure,imshow(N11);

title("反转拉普拉斯锐化图")

N22=convn(I2,h,'same');

figure,imshow(N22);

title("二值化进行拉普拉斯锐化")

N222=imcomplement(N22);

figure,imshow(N222);

title("反转拉普拉斯锐化图")

% 细化

[a,b]=size(N11);

for i=2:a-1

for j=2:b-1

if N11(i,j)==0

if (N11(i-1,j)==0&&N11(i,j+1)==0)||(N11(i-1,j)==0&&N11(i,j-1)==0)||(N11(i+1,j)==0&&N11(i,j-1)==0)||(N11(i+1,j)==0&&N11(i,j+1)==0)

N11(i,j)=1;

else

N11(i,j)=0;

end

end

end

end

figure,imshow(N11);

title("细化图")

对拉普拉斯算子锐化算法的改进：

clear all;

close all;

I = imread('.\zhiwen\103\_4.tif');

figure,imshow(I);

title("原始图像")

I1=medfilt2(I,[3,3]);

[I1,T]=histeq(I1,256);

%拉普拉斯未二值化的锐化

h=[0 1 0,-4 1 0,0 1 0];

N1=convn(I1,h,'same');

N11=imcomplement(N1);

% 增强

temp = N1-double(I1);

tp1=fft2(I1);

tp2=fft2(temp);

out=dftregistration(tp1,tp2,1);%配准

[width,height]=size(temp);

Newtemp=zeros(width,height);

for i=1:width

for j=1:height

source\_x=i+1;

source\_y=j+1;

if(source\_x>width||source\_y>height)

Newtemp(i,j)=0;

else if(((source\_x/double(uint16(source\_x)))==1.0)&&((source\_y/double(uint16(source\_y))==1.0)))

Newtemp(i,j)=temp(int16(source\_x),int16(source\_y));

end

end

end

end

tp3=imadd(uint8(I1),uint8(Newtemp),'uint16');

figure,imshow(tp3,[]);

% 细化

tp3=medfilt2(tp3,[3,3]);

h=[0 1 0,-4 1 0,0 1 0];

tp3=convn(tp3,h,'same');

tp3=imcomplement(tp3);

[a,b]=size(tp3);

for i=2:a-1

for j=2:b-1

if tp3(i,j)==0

if (tp3(i-1,j)==0&&tp3(i,j+1)==0)||(tp3(i-1,j)==0&&tp3(i,j-1)==0)||(tp3(i+1,j)==0&&tp3(i,j-1)==0)||(tp3(i+1,j)==0&&tp3(i,j+1)==0)

tp3(i,j)=1;

else

tp3(i,j)=0;

end

end

end

end

figure,imshow(tp3,[]);

**结论：**

1. 张子龙：通过此次数字图像处理三级项目，我对matlab的使用比之前更加熟练，对图像处理的基本原理有了进一步深刻的认识，通过对比参照的图像处理方法，让我对更多matlab内置的关于图像处理相关函数的处理能力及效果有了清晰的认识，通过此次项目的开展，拓展了我对应用软件处理实际问题的能力。

主要工作：报告算法原理部分的书写以及算法设计的参与

1. 李华龙：通过此次数字图像处理的项目，我参阅了许多相关资料，并了解了指纹图像处理的前期工作，对图像的前期处理有了更深的认识。同时，我还粗略探索了一下指纹特征提取，虽然没有成功，但在试错的过程中也学到了很多新东西，学会了文献中图像增强的方法。通过此次项目的实践，开拓了我对matlab图像处理应用的认识。
2. 王雪：在这次三级项目中，我的主要任务是查找资料，提供了本次报告的撰写方向，然后撰写实验报告正文部分，以及修整论文，通过这次三级项目，我对这学期学到的知识进行了巩固和实践，但也同时发现自己的不足，对知识掌握还不够深刻，对编写程序方面仍有很大缺陷，总是不能够完整地联系在一起，还需要更加努力地学习课程知识，巩固自己的专业内容。
3. 张紫琪：本次三级项目我的主要工作是完成三级项目报告。通过一学期的课程学习，虽说还没有完全掌握数字图像处理技术，但也收获了不少，对于数字图像方面的知识有了深入的了解，更加了解了数字图像处理的本质，对于一些耳熟能详的数字图像相关术语有了明确的认识，也了解了图像处理技术中一些常用处理技术的实质，并是对MATLAB很好的实践与复习。

**参考文献**：

1. 周霞.指纹图像锐化算法的分析与研究
2. 基于MATLAB的指纹图像增强方法
3. 基于MATLAB指纹图像细节特征提取