

# 数字图象处理

## 综合作业一

2019010485 自 91 刘祖炎\*

2021 年 10 月 31 日

## 1 算法原理及代码分析

为使步骤更清晰，下依照作业要求中提示的算法步骤分别进行分析。

### 1.1 步骤一：频域谱计算

分块频域谱计算代码如下：

```
1 [height, width] = size(img);
2 segNum_h = ceil(height / segLength);
3 segNum_w = ceil(width / segLength);
4 for index_h = 1:segNum_h
5     for index_w = 1:segNum_w
6         % Seg Image into Parts
7         wStart = max(1, (index_w-1) * segLength + 1 - (DFTLength / 4));
8         wEnd = min(width, (index_w-1) * segLength - (DFTLength / 4) + DFTLength);
9         hStart = max(1, (index_h-1) * segLength + 1 - (DFTLength / 4));
10        hEnd = min(height, (index_h-1) * segLength - (DFTLength / 4) + DFTLength);
11        segImg = img(hStart:hEnd, wStart:wEnd);
12        segImg = adapthisteq(segImg);
13        %Apply DFT to Segmented Image
14        DFTImg = abs(fftshift(fft2(segImg)));
15    end
16 end
```

频域谱计算方式与小作业 3 中相同。如代码所示， $segLength$  代表分割的图像块大小， $DFTLength$  代表用于进行 DFT 运算的图像块大小。由于在图像边缘处，无法取得图像块的四周边缘图像，故在此处进行范围限定（代码第 11 至 14 行）。通过计算取得的更大图像块与图像本身尺寸的大小关系，限定范围在图片数组内，不会发生越界。

获取图像上下左右四角的坐标后，即可通过矩阵运算取出对应的图像小块，利用自适应直方图均衡对图像进行滤波处理，此后利用工具箱中  $fft2$ 、 $fftshift$ 、 $abs$  进行 DFT 运算得频域谱  $DFTImg$ 。

\*liuzuyan19@mails.tsinghua.edu.cn

## 1.2 步骤二：前景分割与方向估计

方向估计的代码如下：

```
1 DFTImg_sort = sort(DFTImg(:));
2 % Get positions of the points
3 [posX, posY] = find(DFTImg == DFTImg_sort(end - 1));
4 %Calculate distance and period
5 distance = sqrt((posX(1) - posX(2))^2 + (posY(1) - posY(2))^2);
6 Period(index_h, index_w) = 1 / distance;
7 %Calculate direction
8 angle = atand((posY(1)-posY(2))/(posX(1)-posX(2)));
9 Angle(index_h, index_w) = angle;
```

方向估计的方式与小作业 3 中基本相同。其中，代码中 *DFTImg* 为步骤一计算所得频域谱，利用 *sort* 函数对频域谱的函数值进行排序，并利用 *find* 函数找到频域谱中最大的两个值的位置，记录其坐标为  $(x_1, y_1), (x_2, y_2)$ 。

方向与周期的估计公式为：

$$\begin{aligned} P_i &= 1 / \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \\ D_i &= \arctan\left(\frac{y_1 - y_2}{x_1 - x_2}\right) \end{aligned} \quad (1)$$

其中，方向角度取值范围  $D_i \in [-90^\circ, 90^\circ]$ 。

前景分割即判断图像块是否属于指纹区域，在本次作业中，由于不同图像的采集方式不同，因此相比纯指纹图像，存在不同类型的干扰。对图像正确进行前景分割是此次作业生成正确结果的关键。

最终前景图利用数组 *ROI* 表示。若 *ROI* 像素值为 1，则表示该处存在指纹；若 *ROI* 像素值为 0，则表示该处不存在指纹。

在此次作业中，笔者通过大量尝试，利用以下几种方式对图像进行前景分割，判断图像块是否属于指纹区域：

- 方法一：频域图像特征

方法一实现代码如下所示：

```
1 if DFTImg_sort(end-1) < fingerParam % No Fingerprint
2     continue
3 end
4 distance = sqrt((posX(1) - posX(2))^2 + (posY(1) - posY(2))^2);
5 Period(index_h, index_w) = 1 / distance;
6 if distance == 2
7     continue
8 end
9 ROI(index_h, index_w) = 1;
```

该方法在小作业 3 中已经采用，并取得了较好的效果，因此在本次作业中继续利用此方法。

一方面，对求得的小块幅度谱，计算其最大值。若最大值小于预先设定的阈值，则表示该小块中幅度频率变化较小，从而很可能不存在指纹。另一方面，通过对幅度谱中最大值点的位置进行比较判断，去除幅度谱峰值相邻的区域，经验证可知，该区域周期值不合理，将影响后续对指纹方向的判断。

- **方法二：空域与频域方差特征**

方法二实现代码如下所示：

```

1 function [ROI_var_t, ROI_var_f] =
2     varFilter(img, segLength, DFTLength, minVar, maxVar)
3     [height, width] = size(img);
4     segNum_h = ceil(height / segLength);
5     segNum_w = ceil(width / segLength);
6     var_t = zeros(segNum_h, segNum_w);
7     var_f = zeros(segNum_h, segNum_w);
8     for index_h = 1:segNum_h
9         for index_w = 1:segNum_w
10            wStart = max(1, (index_w-1) * segLength + 1 - (DFTLength / 4));
11            wEnd = min(width, (index_w-1) * segLength -
12                        (DFTLength / 4) + DFTLength);
13            hStart = max(1, (index_h-1) * segLength + 1 - (DFTLength / 4));
14            hEnd = min(height, (index_h-1) * segLength -
15                        (DFTLength / 4) + DFTLength);
16            segImg = img(hStart:hEnd, wStart:wEnd);
17            var_t(index_h, index_w) = var(segImg(:));
18            segImg_f = abs(fftshift(fft2(segImg)));
19            var_f(index_h, index_w) = var(segImg_f(:));
20        end
21    end
22    var_t = periodFilter(var_t, 5, 1);
23    var_f = periodFilter(var_f, 5, 1);
24    var_t = Normalize(var_t);
25    var_f = Normalize(var_f);
26    var_f = var_f < maxVar;
27    var_t = var_t > minVar;
28    ROI_var_t = var_t;
29    ROI_var_f = var_f;
30 end

```

由于图像方差能够较好地量化反应图像的变化特征，故此处利用图像方差作为条件之一进行前景分割。首先将图像分为与此前相同的小块 *segImg*，并通过 *fft2* 函数计算频域谱 *segImg\_f*。对求得的空域谱域频域谱，分别计算其方差，得到全图的分块方差值。

对所得空域与频域方差，进行滤波与归一化处理后与预先设定的阈值进行比较，去除不满足条件的点，即可得到前景分割区域。

对指纹图 2 而言，原图及频域方差图如图1、2所示。可以看到，频域方差图较好地反应出部分指纹

区域。



图 1: 原图

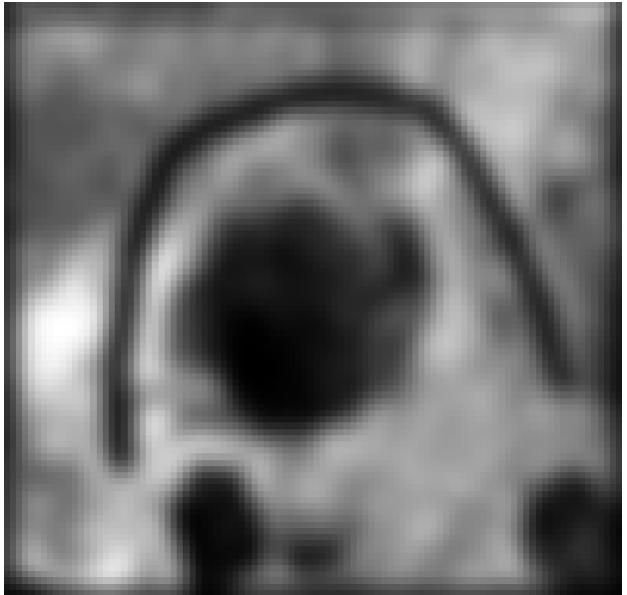


图 2: 频域方差图

对指纹图 3 而言，原图、频域方差图、时域方差图分别如图3、4、5所示。可以看出，频域方差图、时域方差图各反应了一定的指纹信息。



图 3: 原图



图 4: 频域方差图

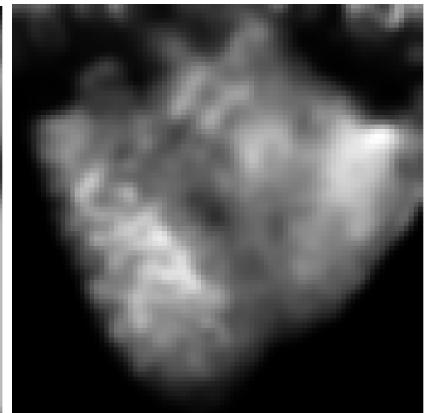


图 5: 时域方差图

- **方法三：前景分割图处理**

方法三实现代码如下所示：

```
1 ROI = ROI_var_f .* ROI .* ROI_var_t;  
2 ROI = bwareaopen(ROI, deleteArea, 8);  
3 s = strel('disk', 1);  
4 ROI = imdilate(ROI, s);
```

对方法一、方法二所得的分割矩阵，首先对其取交集，得到完整的分割矩阵。

$$M = M_1 * M_f * M_t \quad (2)$$

由于该矩阵存在一定的干扰，故需要对其进行处理。笔者在此处利用 `bwareaopen` 函数去除面积过小的干扰点，保留面积最大的区域，其中，去除面积阈值 `deleteArea` 为预先设定的参数。此后，对矩阵进行形态学处理，利用 `imdilate` 函数进行形态学膨胀，填充白色区域中的干扰点，得到连续的前景图。在此处，取卷积核 `s` 类型为 `disk`，半径为 1。

需要注意的是，由于不同图像的干扰类型不同，需要有选择性地选取上述方法进行前景分割以取得更好的效果。

### 1.3 步骤三：滤波处理

由于噪声干扰，某些块的方向和频率可能是错误的。此处利用空域平滑滤波的方法，对原图、方向图、频率图分别进行平滑。

- 原图像增强滤波

由于此题样例中部分图像指纹区域不明显，且干扰较大，故需要利用上课所提到的巴特沃斯滤波器对图像进行增强。巴特沃斯滤波器代码由老师提供，如下所示。

```
1 function result = Bfilter(img)
2     f = img;
3     [M, N] = size(img);
4     P = max(2 * [M N]);
5     D0=140;
6     W=100;
7     n=2;
8     [DX, DY] = meshgrid(1:P, 1:P);
9     D = sqrt((DX-P/2-1).^2+(DY-P/2-1).^2);
10    H = 1./(1+((D.^2-D0.^2)./(D.*W+eps)).^(2*n));
11    F = fftshift(fft2(f,P,P));
12    G = F.*H;
13    g = real(ifft2(fftshift(G)));
14    g = g(1:M,1:N);
15    result = g;
16 end
```

通过巴特沃斯滤波器可突出原图的指纹特征，使最终指纹图像更清晰。以指纹图 2 为例，经巴特沃斯滤波器处理前后的图像分别如图6、7所示。



图 6: 原图



图 7: 处理后

- 方向图滤波

对方向图的平滑滤波采取如下方式：首先将方向图乘以 2，计算其正弦图和余弦图，分别对正弦图与余弦图进行平滑后，调用 `atan2` 函数，再除以 2，即可得到平滑后的方向。其代码如下所示。

```
1 function Angle = angleFilter(Angle)
2     angleCos = cos(Angle * 2 * pi / 180);
3     angleSin = sin(Angle * 2 * pi / 180);
4     filter = fspecial('gaussian', [5, 5], 1);
5     angleCos = imfilter(angleCos, filter, 'replicate', 'same');
6     angleSin = imfilter(angleSin, filter, 'replicate', 'same');
7     angleTan = atan2(angleSin, angleCos);
8     Angle = angleTan / 2 / pi * 180;
9 end
```

其代码与思路相同，其中，平滑滤波器利用高斯滤波器实现。

- 频率图滤波

频率图滤波直接利用高斯滤波器即可实现。其代码如下所示。

```
1 function Period = periodFilter(Period, param1, param2)
2     filter = fspecial('gaussian', [param1, param1], param2);
3     Period = imfilter(Period, filter, 'replicate', 'same');
4 end
```

频率图滤波本质上为一个高斯滤波器，故本题中在其他需要进行图像平滑的部分也调用了该函数（例如 Gabor 滤波器的处理结果等）。值得一提的是，滤波后，需要对图像进行归一化处理。

## 1.4 步骤四：图像增强

图像增强利用限波通过的 Gabor 滤波器实现。其代码如下所示。

```

1 function img_gabor = gaborFilter(img, ROI, Angle, Period, segLength, DFTLength, gaborK)
2 [height, width] = size(img);
3 img_gabor = zeros(width, height);
4 segNum_h = ceil(height / segLength);
5 segNum_w = ceil(width / segLength);
6 for index_h = 1:segNum_h
7     for index_w = 1:segNum_w
8         if ROI(index_h, index_w) == 0
9             continue
10        end
11        % Seg Image into Parts
12        wStart = max(1, (index_w - 1) * segLength + 1 - (DFTLength / 4));
13        wEnd = min(width, (index_w - 1) * segLength - (DFTLength / 4) + DFTLength);
14        hStart = max(1, (index_h - 1) * segLength + 1 - (DFTLength / 4));
15        hEnd = min(height, (index_h - 1) * segLength - (DFTLength / 4) + DFTLength);
16        segImg = img(hStart:hEnd, wStart:wEnd);
17        segAngle = 90 + Angle(index_h, index_w);
18        segPeriod = max(2, gaborK * Period(index_h, index_w));
19        [rho, phi] = imgaborfilt(segImg, segPeriod, segAngle);
20        segGabor = rho .* cos(phi);
21        img_gabor(hStart:hEnd, wStart:wEnd) = segGabor;
22    end
23 end
24 end

```

首先，利用与步骤一相同的方式进行图像分割，对每一个分割出的小块  $segImg$ ，从频率图、方向图中获取其  $segAngle$ 、 $segPeirod$ ，并将其作为参数传入  $Gabor$  滤波器中 ( $Gabor$  滤波器利用工具箱函数  $imgaborfilt$  实现)。对返回的值  $(\rho, \phi)$ ，利用如下公式计算最终的滤波后的图像值：

$$\Theta = \rho * \cos \phi \quad (3)$$

其中，由于  $Gabor$  滤波器要求周期值  $Period > 2$ ，故需要进行限定以满足要求。此外，对不同类型的图像，需要对周期值进行线性增益  $gaborK$  以调整最终所得图像的指纹谱线宽度。

```

1 img = Normalize(img);
2 img = periodFilter(img, 5, 1);
3 img = Normalize(img);
4 img = imbinarize(img, 0.5);

```

完成  $Gabor$  滤波后，需要对所得图像进行归一化、滤波处理，并最终利用  $imbinarize$  函数，设定阈值为 0.5 对所得增强后图像进行二值化，得到满足要求的指纹图像。

## 2 结果分析与参数选取

下以不同图片为顺序说明程序运行结果以及对应采用的方法、选取的参数。

### 2.1 指纹图 1

指纹图 1 处理参数如下所示：

表 1：指纹图 1 参数设置

参数	数值
分割图长度 $segLength$	8
变换块长度 $DFTLength$	32
Gabor 滤波增益 $GaborK$	24
指纹阈值 $fingerParam$	20
方向滤波器尺寸 $angleFilter$	(5, 1)
周期滤波器尺寸 $periodFilter$	(5, 1)

值得一提的是，由于指纹图 1 较为清晰，故没有采用方差频域滤波、前景分割图处理、原图增强滤波的方式，取前景分割图只需采用与小作业 3 相同的方式即可取得较好的效果，详细内容可参见代码 `code1.m`。

中间过程输出结果如下所示，此处展示了前景分割图（图8）、滤波前后的周期图（图9、图10）、滤波前后的 Gabor 滤波器输出图（图11、图12）作为参考。

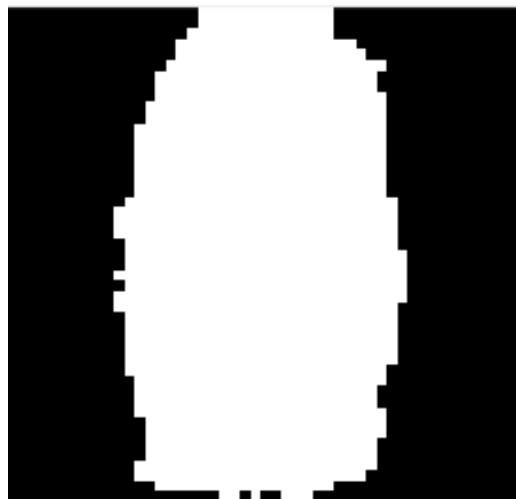


图 8：指纹图 1 前景分割图

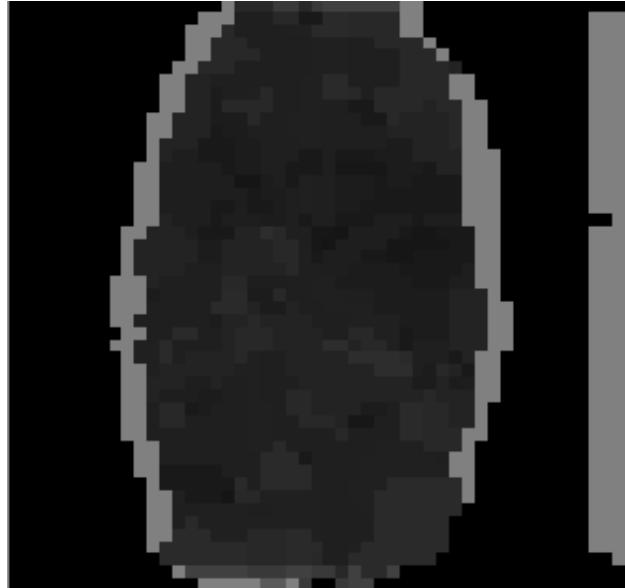


图 9: 指纹图 1 周期图

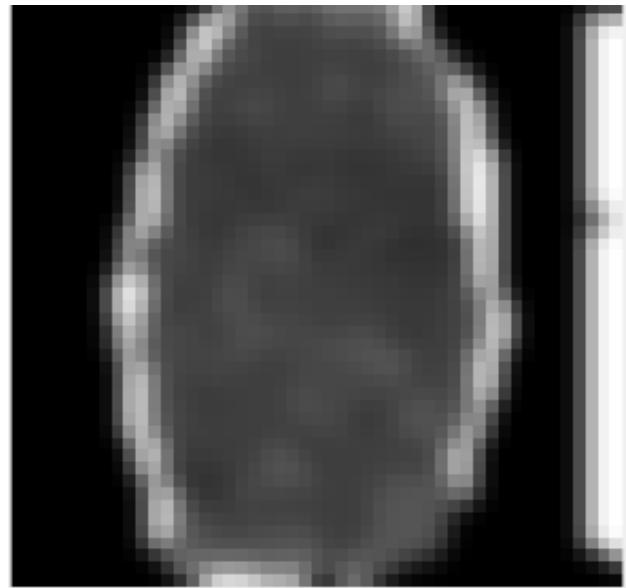


图 10: 指纹图 1 周期图 (滤波与归一化后)



图 11: 指纹图 1 增强图



图 12: 指纹图 1 增强图 (滤波与归一化后)

程序最终指纹增强结果及对应的方向图如图13、14所示。与原图对比可知，所得结果清晰度高，且前景分割、指纹增强结果准确。



图 13: 指纹图 1 输出结果

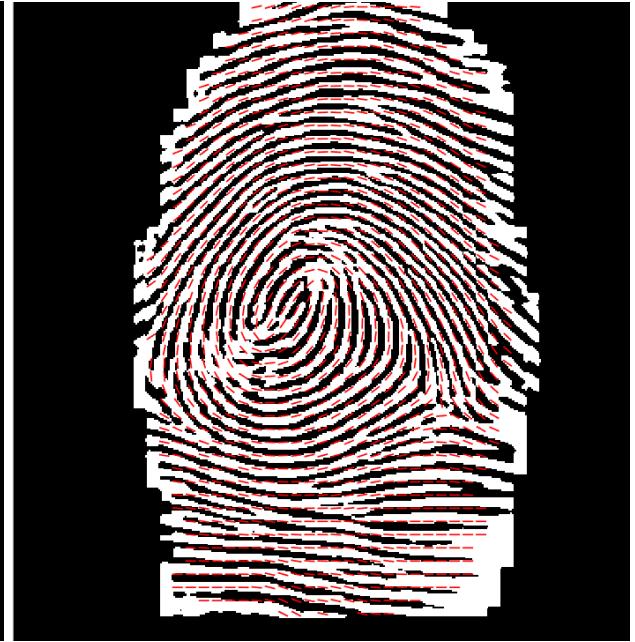


图 14: 指纹图 1 输出结果 (带方向场)

## 2.2 指纹图 2

指纹图 2 处理参数如下所示:

表 2: 指纹图 2 参数设置

参数	数值
分割图长度 $segLength$	8
变换块长度 $DFTLength$	40
Gabor 滤波增益 $GaborK$	32
指纹阈值 $fingerParam$	42
方向滤波器尺寸 $angleFilter$	(7, 3)
周期滤波器尺寸 $periodFilter$	(11, 5)
面积消除阈值 $deleteArea$	256
空域方差最小值 $varMin$	0
频域方差最大值 $varMax$	0.36

由于指纹图 2 噪声较大, 故利用巴特沃斯滤波器进行滤波, 滤波效果如图6、7所示。为取得更好的前景分割效果, 笔者利用频域方差与峰值特征进行交叉限定, 并利用形态学膨胀运算对前景图进行处理, 取得了较好的效果。

中间过程输出结果如下所示, 此处展示了前景分割图的产生过程 (图15、图16、图17、图18)、滤波前后的周期图 (图19、图20)、滤波前后的 Gabor 滤波器输出图 (图21、图22) 作为参考。



图 15: 指纹图 2 前景分割图 (频域峰值特征)



图 16: 指纹图 2 前景分割图 (频域方差特征)



图 17: 指纹图 2 前景分割图 (交集)

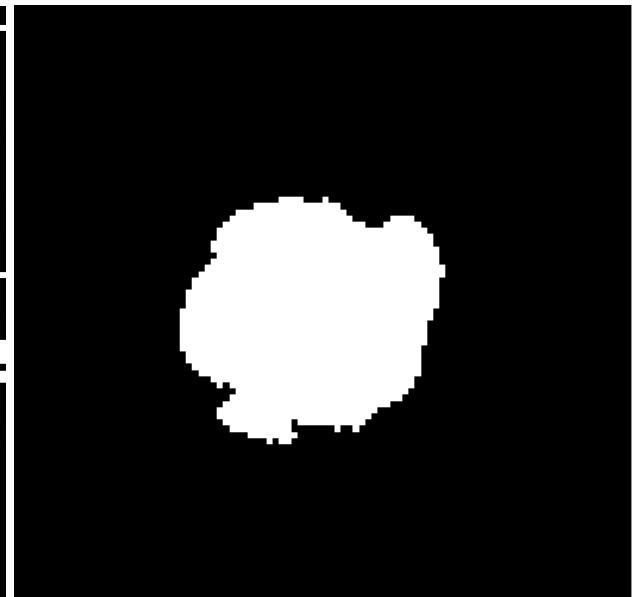


图 18: 指纹图 2 前景分割图 (形态学处理后)



图 19: 指纹图 2 周期图



图 20: 指纹图 2 周期图 (滤波与归一化后)



图 21: 指纹图 2 增强图



图 22: 指纹图 2 增强图 (滤波与归一化后)

程序最终指纹增强结果及对应的方向图如图23、24所示。与原图对比可知，算法准确地分割出了前景区域，且指纹增强结果准确，鲁棒性强。



图 23: 指纹图 2 输出结果

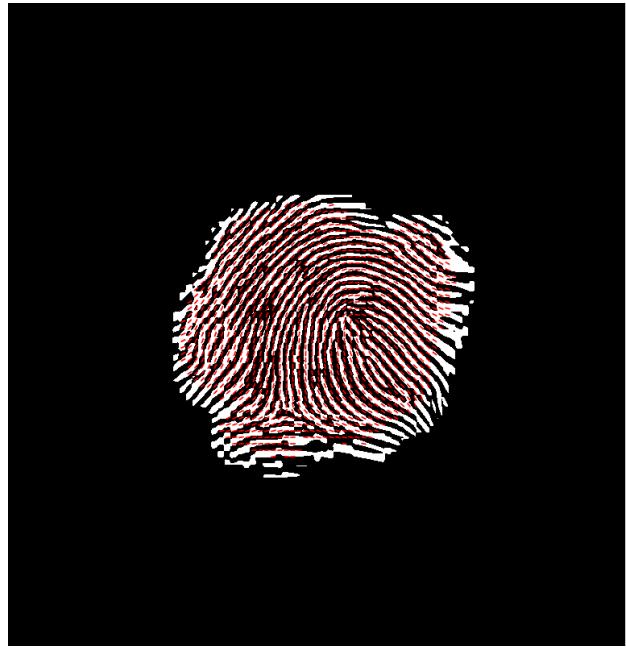


图 24: 指纹图 2 输出结果 (带方向场)

### 2.3 指纹图 3

指纹图 3 处理参数如下所示:

表 3: 指纹图 3 参数设置

参数	数值
分割图长度 $segLength$	24
变换块长度 $DFTLength$	64
Gabor 滤波增益 $GaborK$	28
指纹阈值 $fingerParam$	42
方向滤波器尺寸 $angleFilter$	(5, 1)
周期滤波器尺寸 $periodFilter$	(7, 3)
面积消除阈值 $deleteArea$	256
空域方差最小值 $varMin$	0.15
频域方差最大值 $varMax$	0.8

为取得更好的前景分割效果, 笔者利用时域方差、频域方差与峰值特征共同限定前景图范围, 取得了较为准确的结果。

中间过程输出结果如下所示, 此处展示了前景分割图的产生过程 (图25、图26、图27、图28、图29)、滤波前后的周期图 (图30、图31)、滤波前后的 Gabor 滤波器输出图 (图32、图33) 作为参考。

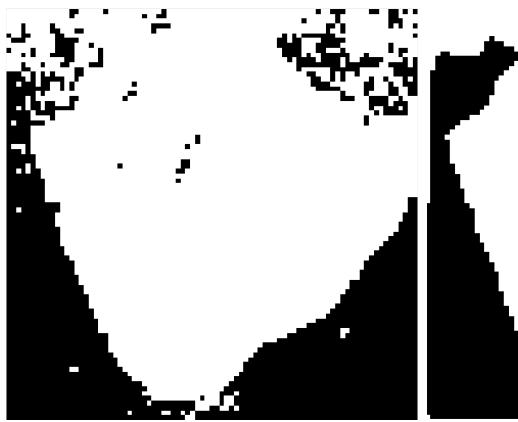


图 25: 指纹图 3 前景分割图 (频域峰值特征)



图 26: 指纹图 3 前景分割图 (频域方差特征)



图 27: 指纹图 3 前景分割图 (空间方差特征)



图 28: 指纹图 3 前景分割图 (交集)



图 29: 指纹图 3 前景分割图 (形态学处理后)

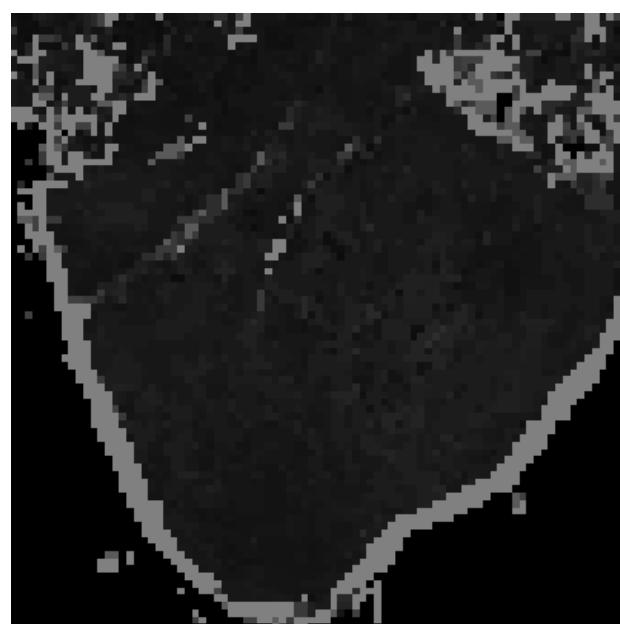


图 30: 指纹图 3 周期图

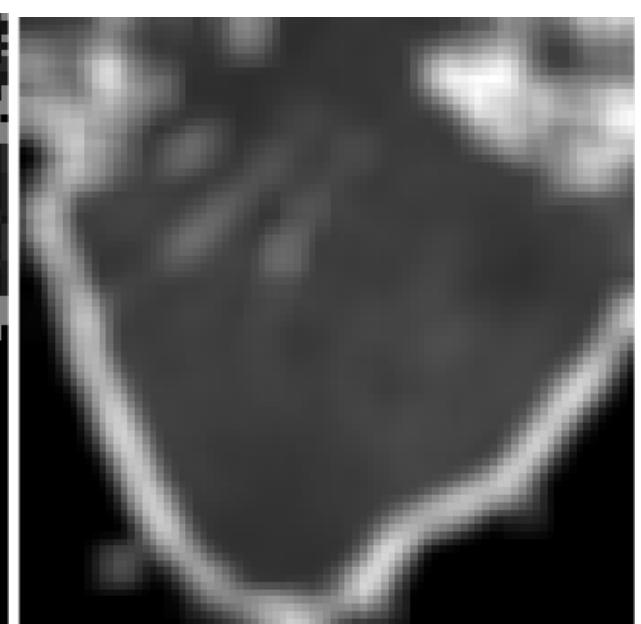


图 31: 指纹图 3 周期图 (滤波与归一化后)

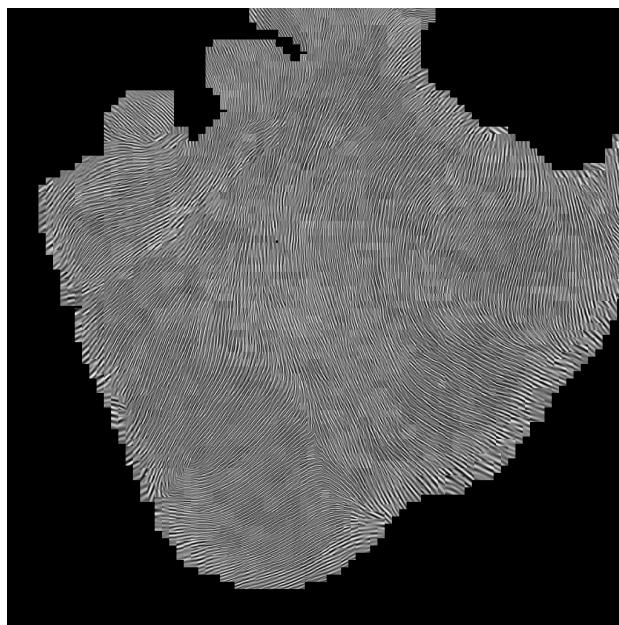


图 32: 指纹图 3 增强图

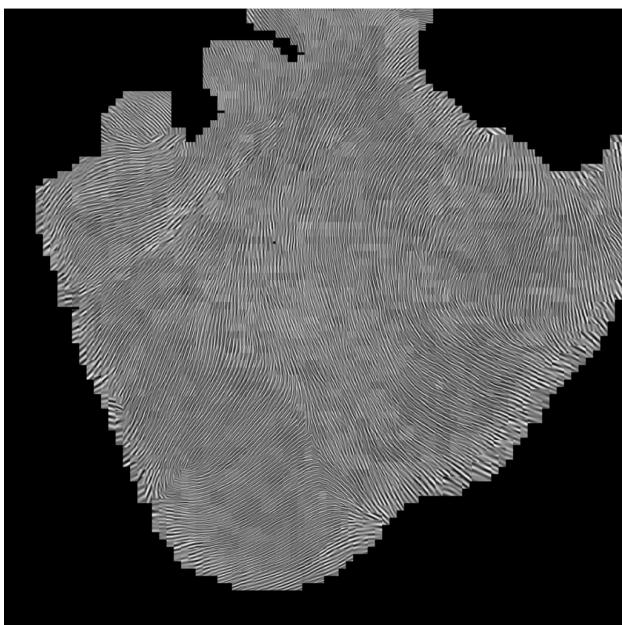


图 33: 指纹图 3 增强图 (滤波与归一化后)

程序最终指纹增强结果及对应的方向图如图34、35所示。与原图对比可知，算法准确地分割出了前景区域，且在掌纹的较大干扰中成功提取出其中的纹路特征，取得了较好的效果。

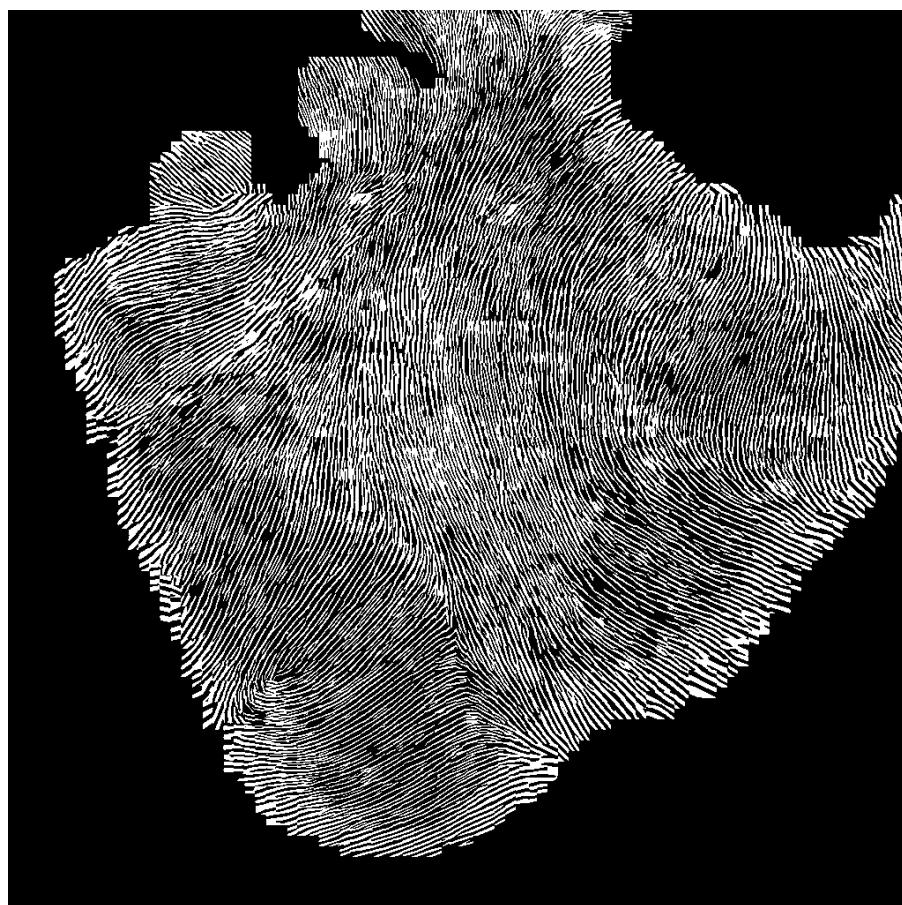


图 34: 指纹图 3 输出结果

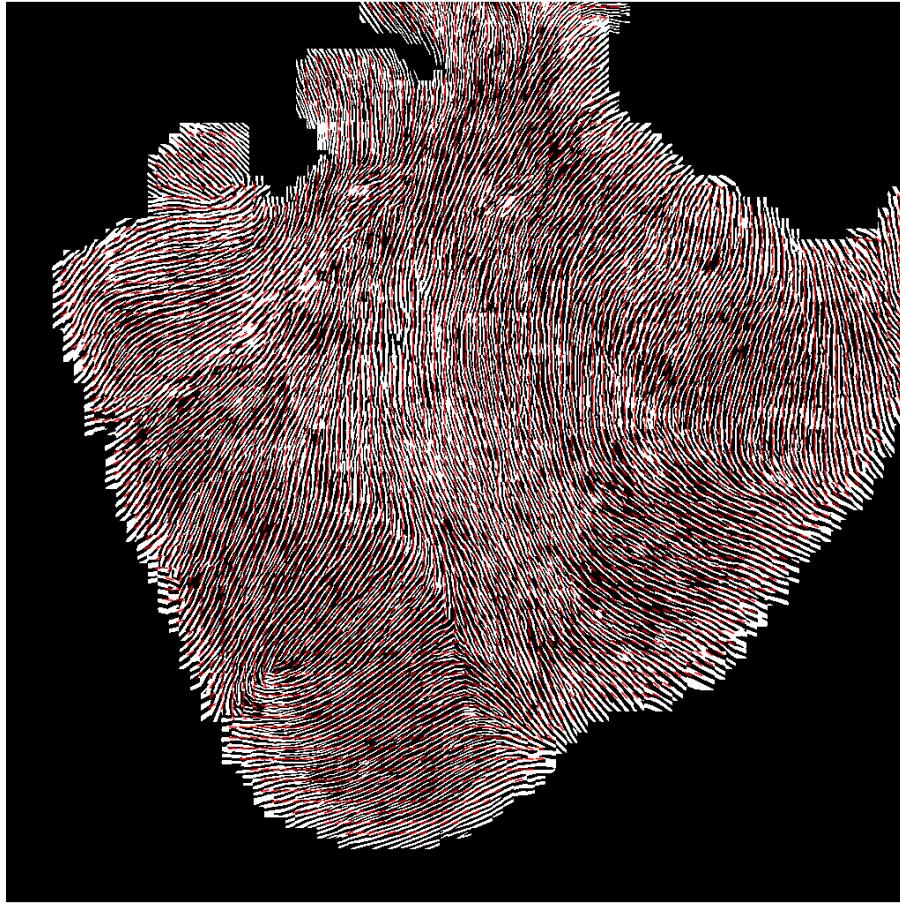


图 35: 指纹图 3 输出结果 (带方向场)

### 3 遇到的困难与解决方案

本次指纹增强大作业代码量不大，且原理也不复杂，但由于所给的三个样例图像各有差异，且部分图像存在的干扰极大，仅使用作业描述中的步骤无法得到较好的效果，因此，笔者花费了大量的时间设计更多算法，最终取得了较好的效果。概括而言，主要遇到的困难有以下几点。

- 图像去噪

由于此次作业提供的各指纹图像各有特点，例如，指纹图 2 具有噪声和污渍，且边缘部分有指纹采集器本身的黑色圆环；指纹图 3 实质上为掌纹，纹路细密且间断多。因此，如何对图像进行去噪及增强处理成为此次实验成功的关键之一。

除作业要求中提到的方法外，在图像滤波方面，笔者利用课上所学知识，应用巴特沃斯滤波器对具有噪声的图像进行特征提取，通过可视化结果可以发现，噪声被较好地消除。比较可知，若不采用该方法进行滤波，最终得到的增强结果将存在较多的噪声。此外，笔者通过调节 Gabor 滤波器的增益参数对最终图像的纹路强度进行调整，最终使得指纹纹路均匀美观。

- 前景分割

利用此前的方法无法对指纹图 2、指纹图 3 较好地进行前景分割。笔者通过较多时间的思考与探索，采用如下几个方法解决前景分割问题。

由于具有指纹的区域频率较高，故频谱图的峰值显然可作为判断依据之一。此外，刻画变化程度的另一个指标即为图像的方差，且由于方差的采样范围为整个图像，而峰值的采样范围仅为图像中的一个点，故方差相比于峰值具有更高的稳定性。实验结果也证明了这一点，在具有较大噪声干扰的图像中，峰值图同样具有较多噪声，而方差图则基本为完整的区域。笔者通过限定空域、频域的方差数值范围，成功提取了具有指纹的图块特征，在未进行特殊处理的前提下，达到了较好的前景分割效果。此外，笔者也发现，仅使用上述方法得到的结果会存在干扰区域与不连续的点。通过查阅资料与反复尝试，由于指纹图像中的指纹具有集中性与唯一性，因此笔者首先利用小面积连通域消除函数选取出图像中的最大区域作为指纹区域，并利用形态学膨胀的方法填充图像中不连续的点。观察结果可知，该方法同样取得了较好的效果。

- 参数选取与调试

由于此次作业存在较多的参数，且不同参数之间会产生相互影响，故笔者花费了较多时间进行参数选择与调试，具体的参数选取在上文中已经列出。

## 4 收获与心得体会

- 指纹增强算法的全流程

通过实际 coding，笔者进一步了解了指纹增强算法与自己的图像处理能力。

- 算法的优化与鲁棒性

笔者通过此次大作业认识到了算法优化与鲁棒性的作用。对不同类型的指纹图，应当采取不同的算法流程以取得更好的效果，另一方面，在设计算法时也应尽量提高算法的鲁棒性，尽量能够做到同一套算法在不同的图中均能使用，而尽量少针对特定图像进行特殊处理。笔者通过设计方差图、形态学处理等前景分割方式，实现了较好的鲁棒性，对不同的图像仅需通过调节超参数即可实现前景分割效果。

- 参数与方法的可解释性

尽管此次实验有较多参数需要调节，且部分参数调节结果与预期截然不同，笔者在调节参数与优化流程的过程中也发现了一些规律与原理。

例如，增大角度滤波器的半径可能使最终图像更加平滑，但也将降低最终指纹图像的准确性。通过调节 Gabor 滤波器的增益，可以调节最终指纹图线的宽度。通过设计方差图可判断出图像块变化的程度。通过小面积连通域去除的方法可筛选出图像中最大的部分。对分割图与变换块长度的设置应当考虑到指纹图线的宽度与整个图像的大小，兼顾准确性与效率。

这提示我们尤其在传统算法中，应当关注所采用的参数与方法的可解释性，从而提升我们的效率。

- 工程思维的应用

此次大作业较为开放，在题目流程的基础上也可以自主实现更多的算法。通过此次大作业，笔者锻炼了自己的工程能力与工程思维。

## 5 可能的改进方向

- 借用更加有效的去噪原理或采用机器学习、深度学习等方法，设计更加鲁棒且能够根据输入图像自适应变化的指纹增强算法。
- 可以通过优化逻辑，进一步提升算法效率。