

数字图像处理大作业——指纹脊线增强

陈昭熹 2017011552

2019 年 11 月 16 日

1 运行说明

1.1 文件结构

Foreground_and_LocalFreq.m 前景分割与局部频率图

LocalOrientation.m 局部方向图

notch_filter.m 理想陷波滤波器

Gabor_Filter.m Gabor 滤波器

ShowOrientation.m 方向图可视化函数

normalization.m 归一化函数

seperated_into_blocks.m 图片分割函数，将原图分割为若干个 32x32 的邻域

remove_dotted_bg.m 去除斑点背景函数（仅对图片 3.bmp 使用）

preproc_3.m 预处理（仅对图片 3.bmp 使用）

1.2 使用方法

entry_3.m 处理 3.bmp 的入口文件

entry_232.m 处理 23_2.bmp 的入口文件

entry_778.m 处理 77_8.bmp 的入口文件

2 算法流程与实现

2.1 Outline

Algorithm 1: 指纹增强算法

Input: I - Image contains fingerprint
Output: EI - Enhanced fingerprint Image

```

1 if  $I$  not empty then
2   preproc(I)
3   blocks  $\leftarrow$  seperate(I)
4   maskI  $\leftarrow$  Foreground(blocks)
5   F  $\leftarrow$  LocalFreq(blocks)
6   O  $\leftarrow$  LocalOrientation(blocks)
7   EI  $\leftarrow$  NotchFilter(F,O,maskI,blocks)
8 end

```

值得注意的是，算法中处理了三张图——前景图、频率图和方向图，其长宽均为原图的 1/8。对于这三张图的处理均基于一个先验知识——在 32x32 的小邻域内指纹的特征与正弦波类似，因此其频谱上应当有关于频谱中心对称的两个极大值，且这两个极值点不应离中心点过近（与直流分量区别开），也不应与中心点过远（与高频噪声区别开）。在这样的原理指引下派生出三张图的处理算法，后文就不再对此原理进行赘述。

2.2 预处理

由于三张测试图指纹可恢复情况不同，因此对于某些干扰较多的情况需要进行一定的预处理以获得更好的恢复效果。本文仅对图 3.bmp 进行一定的预处理，以去除背景的散点噪声以及指纹下的黑色背景，尽可能去除其他无关纹样的干扰。

对于图像 3.bmp，首先注意到其右半部分有规律的点状背景，这是会对后续方向图计算产生影响的。通过傅里叶变换找到该图频谱一个周期内存在 8 个明显的分量，通过 matlab GUI 获得这几个分量坐标后使用陷波阻断滤波器将其滤除，再还原后发现图片的点状背景被成功移除。之后对该使用过高提升滤波来增强边缘，同时去除噪声。利用去噪锐化的图片进行粗分割，计算每一个不重叠的 160x160 邻域内的均值，并以这个均值作为阈值进行二值化得到预处理之后的图像。

对于图像 77_8.bmp，注意到针对本文实现的理想陷波通过滤波需要将原图尺寸拉伸到 700x700 效果较好。

2.3 前景背景分割

基于前文讲述的先验知识作为原理，频率图提取时对于每一个小邻域首先计算 FFT，去除其中的最大值（该值必为直流分量）后，继续向下查找极值。若找到两个对称的极大值，且满足相应的阈值条件则判断为是指纹图像，将 mask 对应区域置为真值。

若找到的两个对称点为 p_1, p_2 ，对于 32x32 的邻域其傅里叶变换的中心位于坐标 (17,17)，则相应的阈值条件定义如下：

$$\text{偏心距离: } centralgap = \frac{p_1 + p_2}{2} - (17, 17)$$

$$\text{单臂欧式距离: } arm = \|p_1 - (17, 17)\|$$

$$\text{峰值功率: } mag = Intensity(p_1)$$

这些阈值条件均作为函数的参数实现，可以针对不同场景进行调节来达到更好的效果。

算法流程如下：

Algorithm 2: 前景背景分割

Input: blocks - blocks of original Image

Output: maskI - logical mask in where true stand for fingerprint region

```

1 for eachblock in blocks do
2   maskI ← false
3   region ← |FFT(eachblock)|
4   Max{region} ← 0
5   sort(region) in descending
6   while k < 10 do
7     if top-2 points in region is symmetrical about centre then
8       if points are not too far or too close to the centre then
9         maskI ← true
10      end
11    end
12    k ← k + 1
13  end
14 end
15 LowPassFilter(maskI)

```

2.4 局部频率图提取

原理前文已提及，不做赘述。选取出指纹区域的正弦分量极值后，使用两个极值点之间的欧式距离作为当前点的频率值。

算法流程如下：

Algorithm 3: 局部频率图提取

Input: blocks - blocks of original Image

Output: F - LocalFrequency Map

```

1 for eachblock in blocks do
2   region ← |FFT(eachblock)|
3   Max{region} ← 0
4   while num of peak ≠ 2 do
5     | peak ← find Max{region}
6   end
7   calculate Euclidean distance in peak
8   F ← Euclidean distance
9 end
10 LowPassFilter(F)

```

2.5 局部方向图提取

原理与前类似，不再赘述。实现过程中通过计算找到的两个极值点连线与水平轴的夹角来定义该邻域的方向图方向。

算法流程如下：

Algorithm 4: 局部方向图提取

Input: blocks - blocks of original Image

Output: O - LocalOrientation Map

```

1 for eachblock in blocks do
2   O ← π/2
3   region ← |FFT(eachblock)|
4   Max{region} ← 0
5   while num of peak ≠ 2 do
6     | peak ← find Max{region}
7   end
8   calculate angle towards horizon in peak
9   F ← angle
10 end
11 LowPassFilter(O)

```

2.6 陷波滤波器滤波

按照要求，本文使用陷波滤波器进行滤波。主要的想法是通过设计陷波通过滤波器，让找到的正弦分量通过滤波器，而其余的均滤除，从而达到增强指纹图像的效果。本文实现了两种方法，分别是理想陷波滤波器和 Gabor 滤波器，效果各有千秋。

2.6.1 理想陷波滤波器

理想陷波滤波器思想较为朴素，利用得到的频率图和方向图来计算出每一个 32×32 邻域的频谱中正弦分量的两个极值点坐标位置（即对应着需要增强的指纹图样），**将以这两个坐标位置为中心的 3×3 的邻域置为 1**，作为陷波通过的频率范围，其余的位置均为 0。然后将这个滤波器应用到 32×32 邻域的频谱上进行陷波滤波，从而能够通过傅里叶反变换还原出增强的指纹图像。

值得注意的是，为了获得更好的恢复效果，在恢复最后一步使用了一个小技巧——**基于重叠的均值平滑**。既然使用了 32×32 的邻域那么就不要浪费这些信息，滤波后变换回空域就直接将 32×32 的图线性加到恢复后的图像上，这样一来就可以利用上相邻邻域之间的信息，让指纹图像非常平滑。由于我们的邻域逻辑上是以 8×8 为中心，因此 32 的大邻域在滑动过程中会有重叠，如果直接线性叠加的话，重叠部分会因为多加了若干次导致灰度爆炸，呈现出白色。为了解决这个问题，只需要维护一个 `overlap` 变量计算图像中每个位置的重叠，每一次进行线性叠加时，对应位置就将 `overlap+1`，这样一来，处理之后将图像除以 `overlap` 矩阵就可以还原出极度平滑的指纹图像。其背后的原理就是一个均值滤波，利用了大邻域的重叠信息。

算法流程如下：

Algorithm 5: 理想陷波滤波器

Input: 原图 I, 频率图 F, 方向图 O, 前景图 maskI

Output: 增强指纹图 EI

```

1 for eachblock in blocks do
2   region ← FFT(eachblock)
3   Max{|region|} ← 0
4   region ← region * notch_LowPassFilter(F, O, maskI)
5   EI ← EI + iFFT(region)
6   average smoothen EI based on overlap
7 end
  
```

2.6.2 Gabor 滤波器

这里闹了个笑话。笔者原本并不知道 Gabor 滤波器属于陷波滤波器 (notch filter)，于是就并没有将论文 [1] 中使用 Gabor 滤波器的过程复现，后经过老师指点再经过频域分析发现 Gabor 滤波器符合陷波滤波器特点，因此又将这种方法予以实现，正好两种方法做一对比。与上一小节中的方法类似，还是取 32x32 的邻域进行 FFT 变换，通过得到的频率图和方向图直接调用 imgaborfilt 来完成滤波过程，由于前面频率图的定义十分巧妙，可以直接作为输入参数使用到 imgaborfilt 函数中，而方向图只需要将弧度转化为角度即可。

这里也使用了上一小节提到的基于重叠的均值平滑。

注意到与上面一小节不同的是，在滤波之后使用了归一化函数将滤波后的空域图均值设置为 0.5，方差设置为 1，这是为了平衡原图灰度，避免 Gabor 滤波的方块效应（如果采用二值化策略，在恢复不好的地方就会出现明显的断层，这一点也在论文中有所提及）。

算法流程如下：

Algorithm 6: Gabor 滤波器

Input: 原图 I, 频率图 F, 方向图 O, 前景图 maskI

Output: 增强指纹图 EI

```

1 for eachblock in blocks do
2   region ← FFT(eachblock)
3   Max{|region|} ← 0
4   region ← Gabor_Filter(region, F, O * 180/π)
5   EI ← EI + iFFT(region)
6   average smoothen EI based on overlap
7 end
  
```

3 结果及过程

值得注意的是，在实际应用过程中，指纹匹配往往是寻找匹配关键点，如脊线分叉和脊线终止这些特征。因此在本文实现过程中，参数调整的原则是让增强后的指纹图像中原有的这些特征点得以正确的增强，并且不会经由算法拟合出原图没有的特征点。但对于脊线的连续性和宽度均匀性则放在次一级的位置进行考虑。尽管如此还是会有不尽如人意的地方，比如在对 23_2.bmp 增强中极个别脊线会出现轻微错位移动。3.bmp 增强中在指纹顶部的部分特征没有还原出来，原图的背景直线影响过大，覆盖了指纹特征，即使肉眼也很难辨认。事实上，经过我查阅文献发现，学界常用的指纹增强算法原理有三：基于直方图、基于 FFT 频域滤波、基于 Gabor 滤波，本文使用的陷波通过滤波器属于第二种基于 FFT 的频域滤波，而这三种方法对于同一张图像所产生的误差和错误拟合可以通过文献 [2] 中的一份数据图看出，该图如下所示：

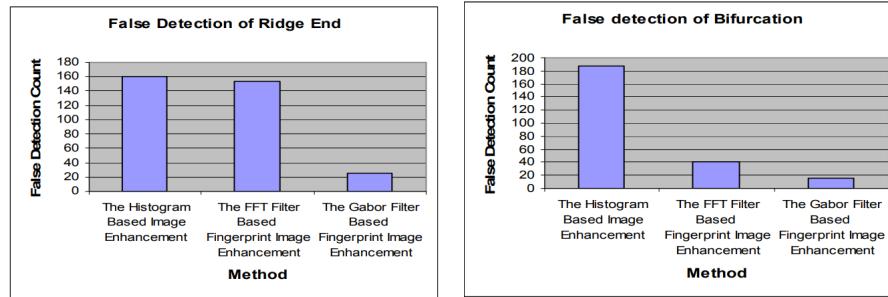


图 1: 三种方法脊线终止错误和脊线分叉错误

由图可以看出基于 FFT 的方法，如本文方法一样是具有局限性的，想要获得更好的增强效果和更少的错误可以使用 Gabor 滤波的方法。

3.1 77_8.bmp



图 2: 77_8.bmp 增强结果: 理想陷波 (左) vs Gabor(右)

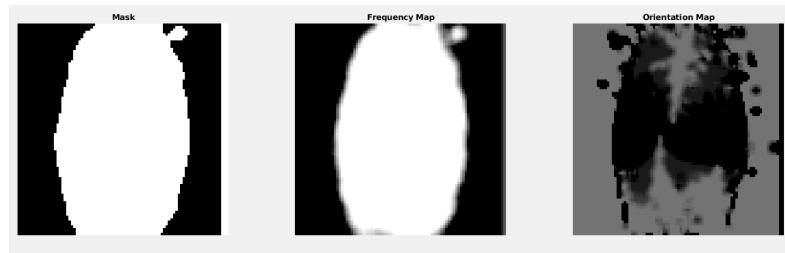


图 3: 77_8.bmp 前景、频率图、方向图



图 4: 方向图空域显示：平滑前 vs 平滑后

3.2 23_2.bmp



图 5: 23_2.bmp 增强结果: 理想陷波 (左) vs Gabor(右)

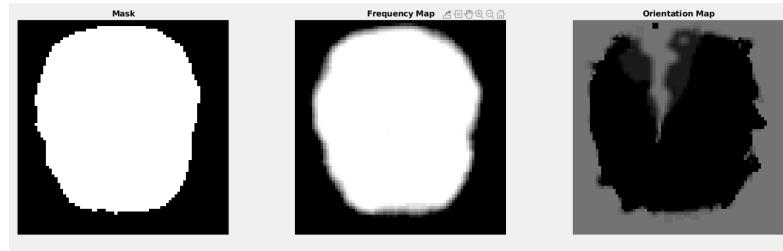


图 6: 23_2.bmp 前景、频率图、方向图



图 7: 方向图空域显示：平滑前 vs 平滑后

3.3 3.bmp



图 8: 3.bmp 去除点状背景

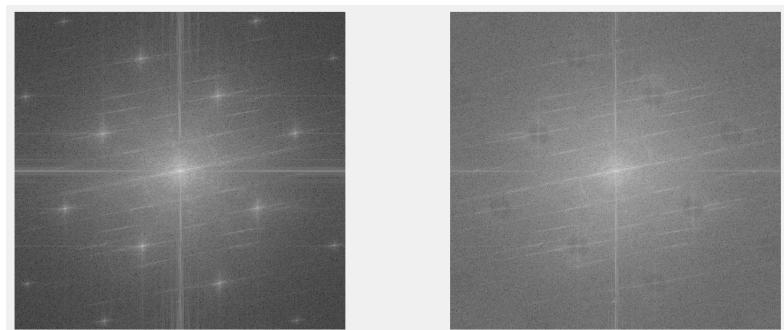


图 9: 去除点状背景过程——频谱变化

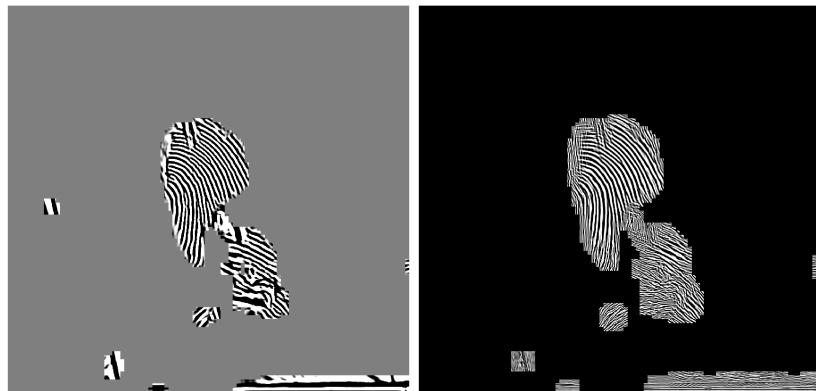


图 10: 3.bmp 增强结果: 理想陷波 (左) vs Gabor(右)

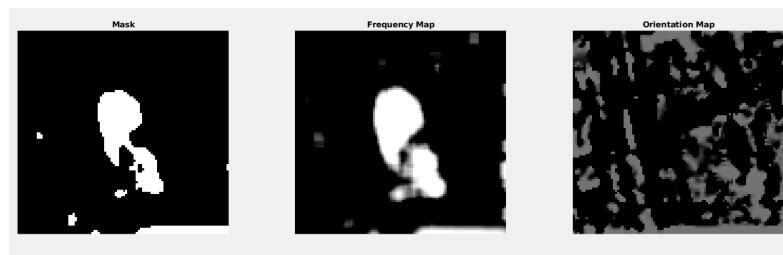


图 11: 3.bmp 前景、频率图、方向图



图 12: 方向图空域显示: 平滑前 vs 平滑后

参考文献

- [1] Lin Hong, Yifei Wan, and Anil K. Jain. "Fingerprint image enhancement: Algorithm and performance evaluation." IEEE Transactions on Pattern Analysis and Machine Intelligence 20.8 (1998): 777-789.
- [2] M.Rajinikannan, D.Ashok Kumar, and R.Muthuraj. May 2010. "Estimating the Impact of Fingerprint Enhancement Algorithms for Better Minutia Detection", Internationl Journal of Computer Applications(0975-8887), vol. 2, pp. 36-42