

一、基于局部结构和细节特征点的指纹编码

1. 细节特征点的筛选

考虑到编码的字节限制，故有必要对细节特征点进行筛选。另外，由于皮肤的弹性，越靠近边缘的指纹更容易受到非线性扭曲的影响，因此本文仅选取临近中心点的部分细节特征点。

本文设定阈值 R ，并认为细节特征点与中心点 $P_0(x_0, y_0)$ 的距离小于此值时才纳入编码。进而通过遍历所有细节特征点，结合上述的特征计算方法，可以得到相应的特征集合 S ，可表示如下：

$$S = \{(x_i, y_i, \theta_{fi}, I_i, n_i) | \sqrt{(x_i - x_0)^2 + (y_i - y_0)^2} \leq R, I_i \in \{a, b, c\}, i = 1, 2, \dots, N\}$$

其中 I 表示细节特征点的种类，若 $I = a$ ，表示该特征点为中心点， $I = b$ 则表示该特征点为端点， $I = c$ 则说明该特征点为分叉点， n 为中心点与特征点连线穿过的脊线数， θ_f 为特征点所在区域的块方向场角度。

需要注意的是，为保证满足字节限制的要求，若在半径为 R 的领域内存在过多的特征点，本文仅选取距离中心点最近的20个特征点。

2. 构建极坐标系

将中心点 $P_0(x_c, y_c)$ 作为极点，其所在方向场的角度为 θ_c ，沿此角度作射线 OP_0 作为极轴，构建具有旋转平移不变性的极坐标系。

3. 坐标系转换

为将上述的特征集合映射至构建的极坐标系中，需计算出特征点坐标与中心点的坐标以及两点所在方向场的方向差。

$$\begin{cases} \rho = \sqrt{(x - x_0)^2 + (y - y_0)^2} \\ \theta = \arctan \frac{y - y_0}{x - x_0} - \theta_c \\ \Delta\theta = \theta_f - \theta_{0f} \end{cases}$$

故特征集合修改为：

$$S = \{(\rho_i, \theta_i, \Delta\theta_i, I_i, n_i) | \rho_i \leq R, I_i \in \{a, b, c\}, i = 1, 2, \dots, N\}$$

其中 θ, n, I 的含义与上文相同。

4. 特征编码

本文采取特征向量的方式进行编码，即使得一个特征点的各特征作为一列内的元素。同一个指纹中的 N 个特征点便对应应有 N 列向量，最后整合存储。

匹配：

本文采取分层次的特征点相似度计算方法。首先将两个指纹的中心点的坐标对齐，并通过旋转使得两者的方向（即所在处的块方向场角度）相同，进而判断 P_{assi1} 和 P_{assi2} 所在之处的方向场角度是否相同，若两者差距大于 $\pi/3$ 则可以认为两者来源不同，停止计算，否则继续计算。

接下来将细节点匹配问题转化为指派问题，并以Ravi提出的指纹细节点匹配分数作为目标函数[这里要插入文献]进行计算。最后根据得分的高低判断两个指纹的来源是否相同。

具体流程如图所示：

（插图）

1.基于指派问题分析指纹异同

接下来进行细节点的匹配，考虑到理想的完全匹配状态下，两个指纹上的细节特征点应一一对应。但实际上，指纹采集质量或是指纹来源不同都有可能使得指纹的细节特征对应关系被破坏。

若在计算相似程度时，将两个指纹上的细节特征点错配、重复配，则会导致相似程度的计算出现较大误差。为此，本文通过将指纹的细节匹配问题转化为指派问题进行分析，使得在最佳匹配关系的基础上判断指纹相似程度。故首先需要计算任意两个指纹之间的相似性，并汇总为相似性矩阵。

两指纹之间的相似性通过如下方式定义：

Step1:将各特征点在上述相对极坐标系中的坐标 (θ_i, ρ_i) ，穿过脊线的条数 n_i ，特征点与中心点的方向场角度差 $\Delta\theta_i$ ，特征点的种类 I_i 都进行归一化。以特征点与中心点穿过脊线的条数为例：

$$n'_k = \frac{n_k - \min(n_i)}{\max(n_i) - \min(n_i)}$$

Step2:运用欧氏距离计算任意两个指纹之间的差异性。

$$d_{ij} = \sqrt{(\theta_i - \theta_j)^2 + (\Delta\theta_i - \Delta\theta_j)^2 + (\rho_i - \rho_j)^2 + (I_i - I_j)^2 + (n_i - n_j)^2}$$

Step3:构建相似性矩阵

（加一个矩阵展示一下）

进而构建指派问题的整数规划模型：

本文选取Ravi提出的匹配分数计算方法作为目标函数：

$$\max Score = \frac{M_{MO}}{\max(N_1, N_2)}$$

其中 M_{MO} 为匹配成功的点的个数， N_1 录入细节点的总数目， N_2 模板细节点的总数目。这种匹配分数的计算方法在一定程度上惩罚了不完整指纹图像匹配以及重叠部分较少的情况。

根据特征点的一一对应原则建立相应的约束：

(约束1)

实际匹配过程中允许存在一定误差，故两点之间相差距离 d_{ij} 超过阈值 L 时才认定匹配失败：

(约束2)

通过0-1变量的约束得到成功匹配的特征点总数：**(约束3)**

综上，将模型汇总可得到：

（约束和目标函数加起来写在一起）