

模式识别

第十章 模式识别应用-指纹识别

郭园方

北京航空航天大学计算机学院

内容

- * 指纹识别简介
- * 指纹图像处理
- * 指纹特征匹配
- * 结论与展望

引言

- * 传统身份鉴别方法
 - * 钥匙、证件、ATM卡、密码.....
- * 存在的问题
 - * 遗忘、丢失、被伪造.....
- * 基于生物和行为特征进行身份鉴定

引言

- * 指纹的优点:

- * **稳定性**: 从胎儿六个月到尸体腐烂, 指纹的类型、结构和统计特征等在总体上是稳定的。
- * **独特性**: 每个人的指纹都是独一无二的。研究表明: 指纹最多可用来区分 10^{96} 个人。

背景和意义

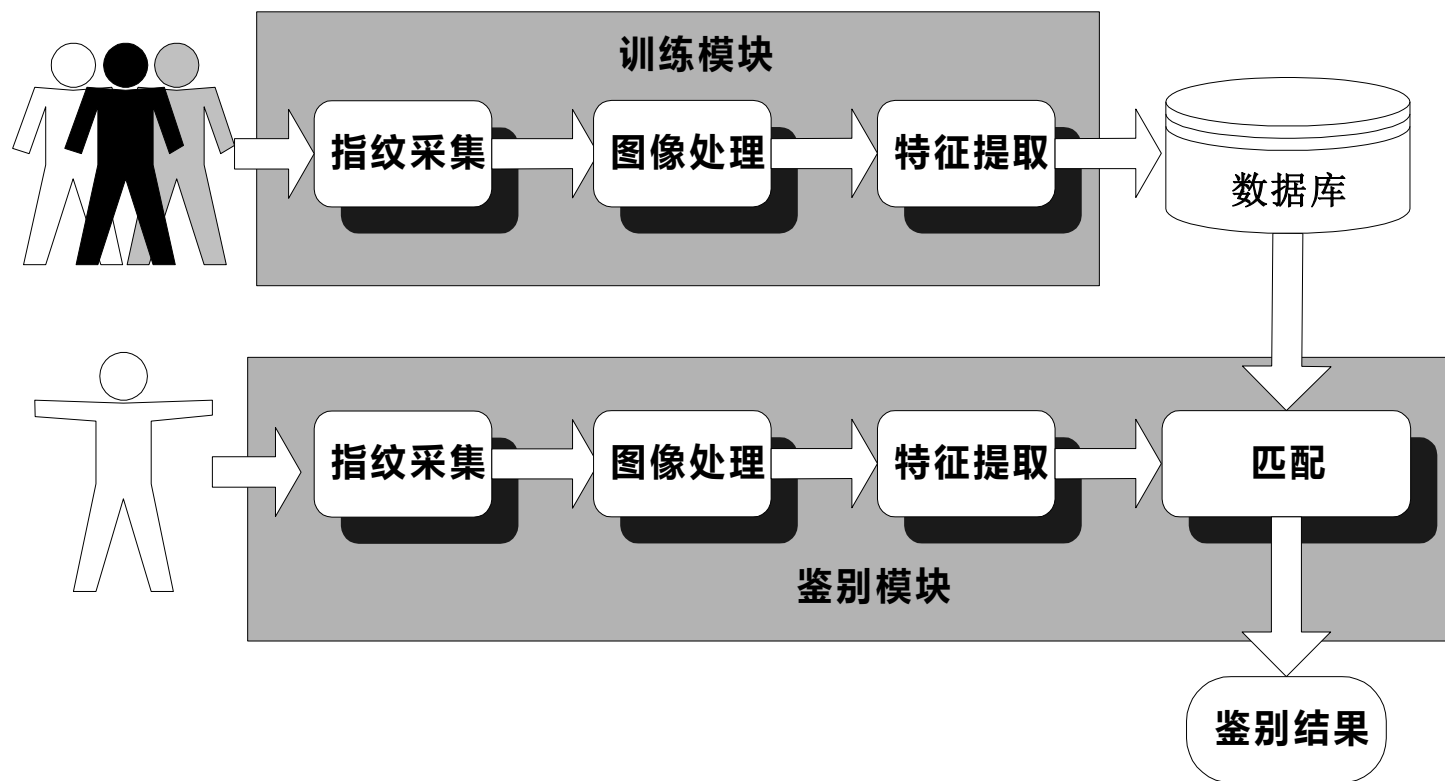
- * 指纹的优点

- * 在刑侦领域有多年应用基础
- * 最为人们所接受
- * 独特性和普遍性得到广泛认可

- * 面临的问题

- * 性能方面仍有待提高
- * 5%的人指纹无法提取有效特征

系统框图



指纹身份鉴别中的关键问题

- * 数据采集
- * 描述原始输入数据
- * 定义不同模式之间的相似度
- * 匹配算法
- * 对系统进行评估和测试

数据采集

- * 线传感器
- * 面传感器

典型特征提取

- * **特征选取**：Herry特征、Galton特征、奇异点（中心点和三角点）、小孔、纹理特征；
- * **特征提取**：图像处理、纹理分析、直接在灰度图上提取特征；
- * **现状**：最常用的特征是端点和分叉点；最常用的特征提取方法是图像处理；

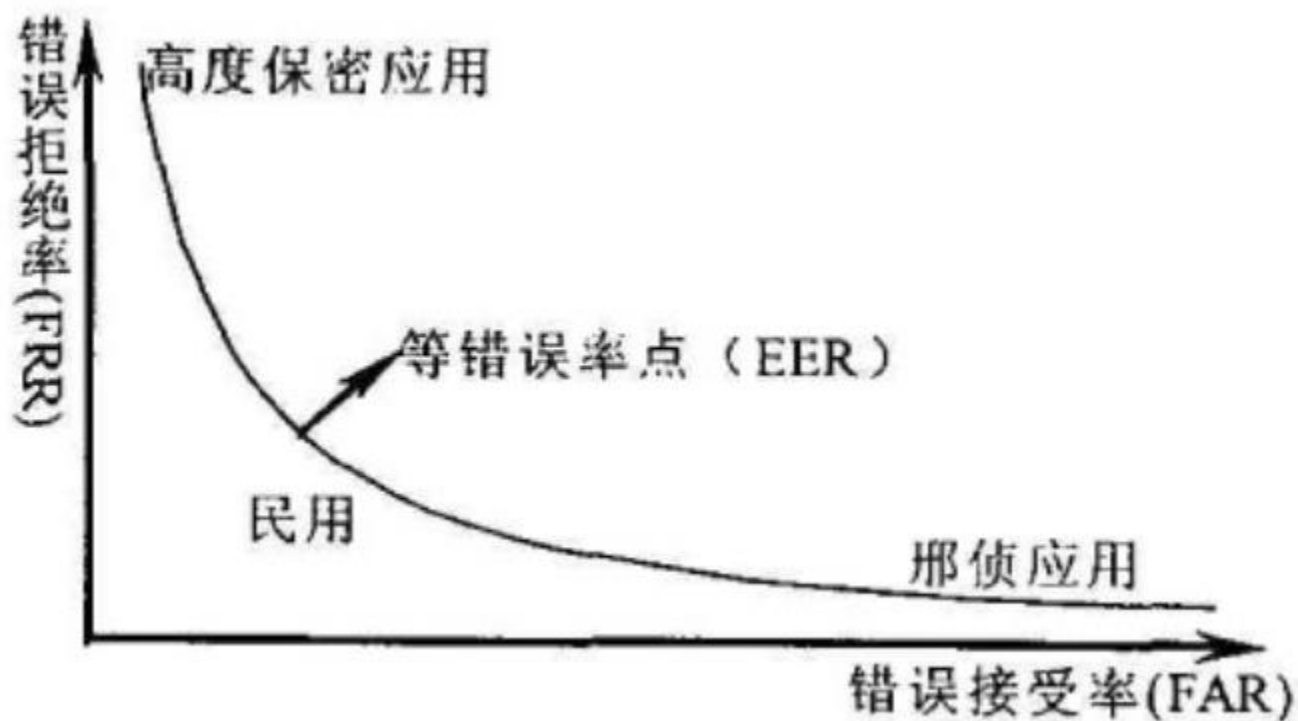
匹配算法

- * 匹配算法和提取的特征强相关
- * 现状：对于点模式，最重要的是寻找对应点。

系统性能评价

- * 对鉴定系统：FAR、FRR和ROC曲线等
 - * FAR
 - * FRR
 - * ROC曲线
- * 对识别系统：真正的指纹拥有者在结果中的概率
- * 对两种系统都适用：图像拒绝率

系统性能评价



指纹分类

- * 句法方法
- * 基于特征提取的方法：奇异点。
- * 神经网络的方法
- * 现状：神经网络方法和特征提取的方法比较流行。

指纹识别的两种方式

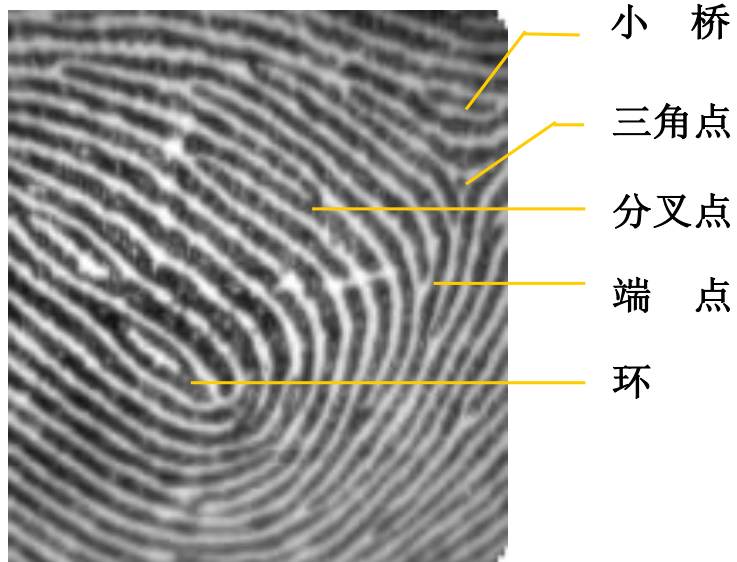
- * 验证(一对一)

- * 通过把采集到的指纹与已登记的指纹进行一对一的比对,来确认身份
- * 例如: 银行\电子商务

- * 辨识(一对多)

- * 把采集到的指纹同指纹数据库中的指纹逐一比对,从中找出相匹配的指纹
- * 例如: 疑犯查找

指纹的特征



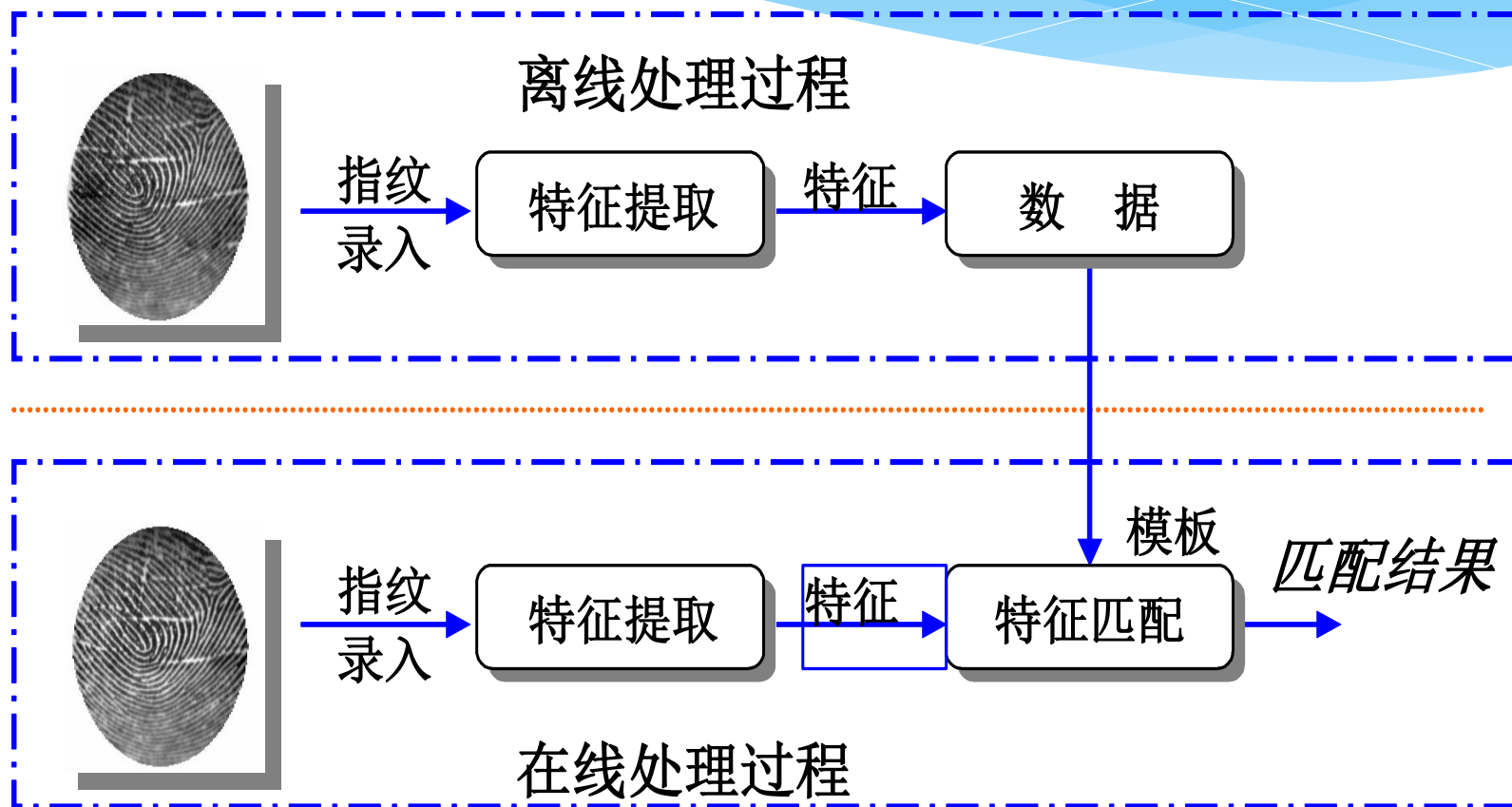
- * 指纹识别所利用的主要是指纹的细节特征点：小桥、环、分叉点、三角点、端点等。

指纹的常用特征



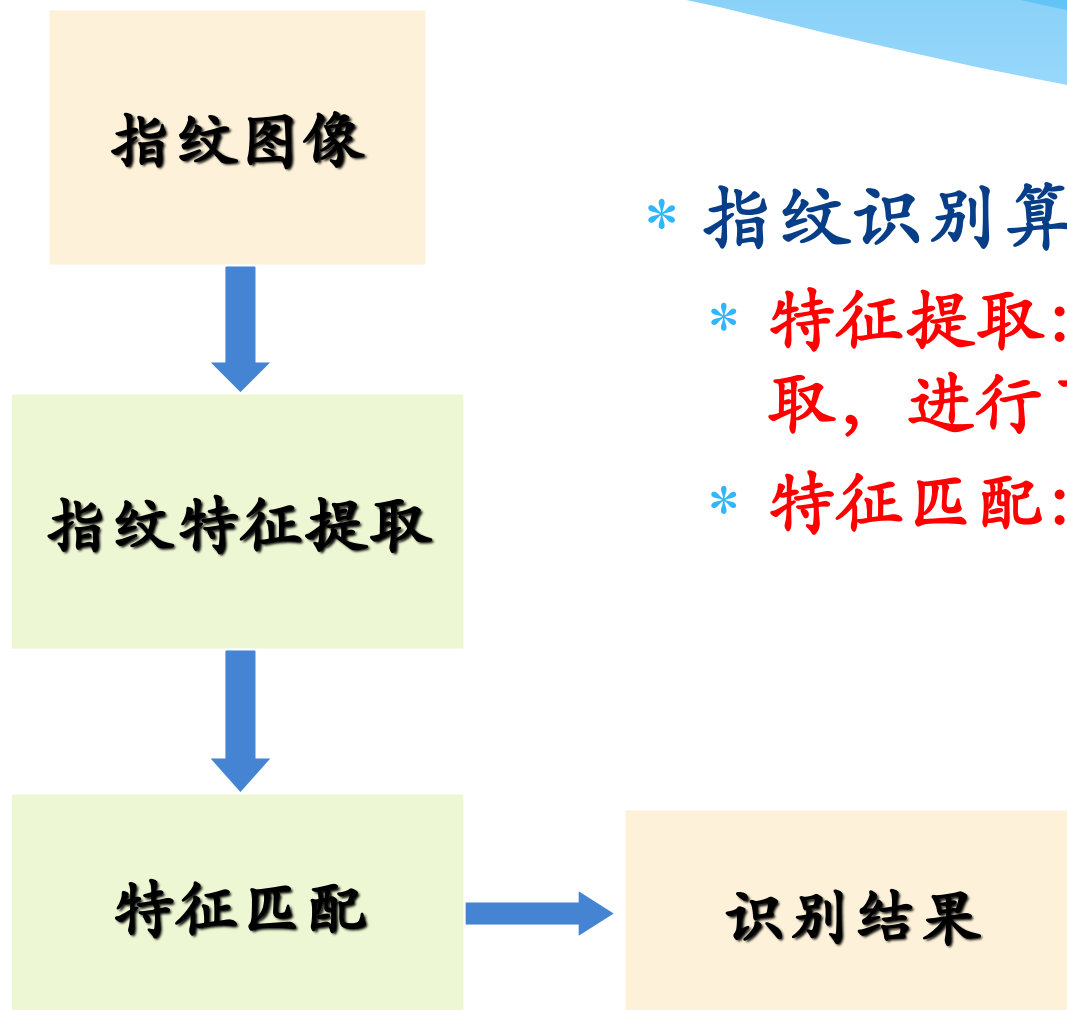
- * 自动指纹识别系统使用了其中两种主要的特征，即分叉点和端点，统称为细节特征 (minutiae)

基于指纹的身份鉴别系统



离线过程提取用户的指纹样本的特征，在线过程鉴定用户的身份

指纹识别的算法流程



* 指纹识别算法的关键问题

- * 特征提取：结合方向图和纹路提取，进行了细节特征提取
- * 特征匹配：基于点模式匹配

指纹匹配关键问题一：特征提取

- * 特征提取的准确性直接影响系统的性能
- * 问：如何准确地在灰度图上提取指纹分叉点和端点？
- * 答：利用多种图像处理方法。

特征提取的算法流程

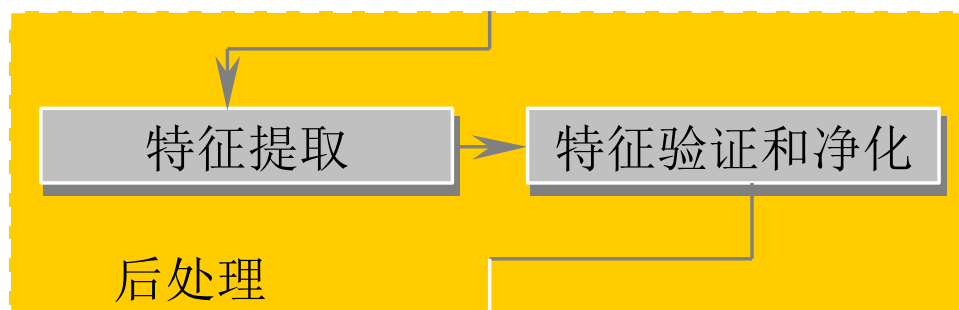
灰度图像



二值化图像

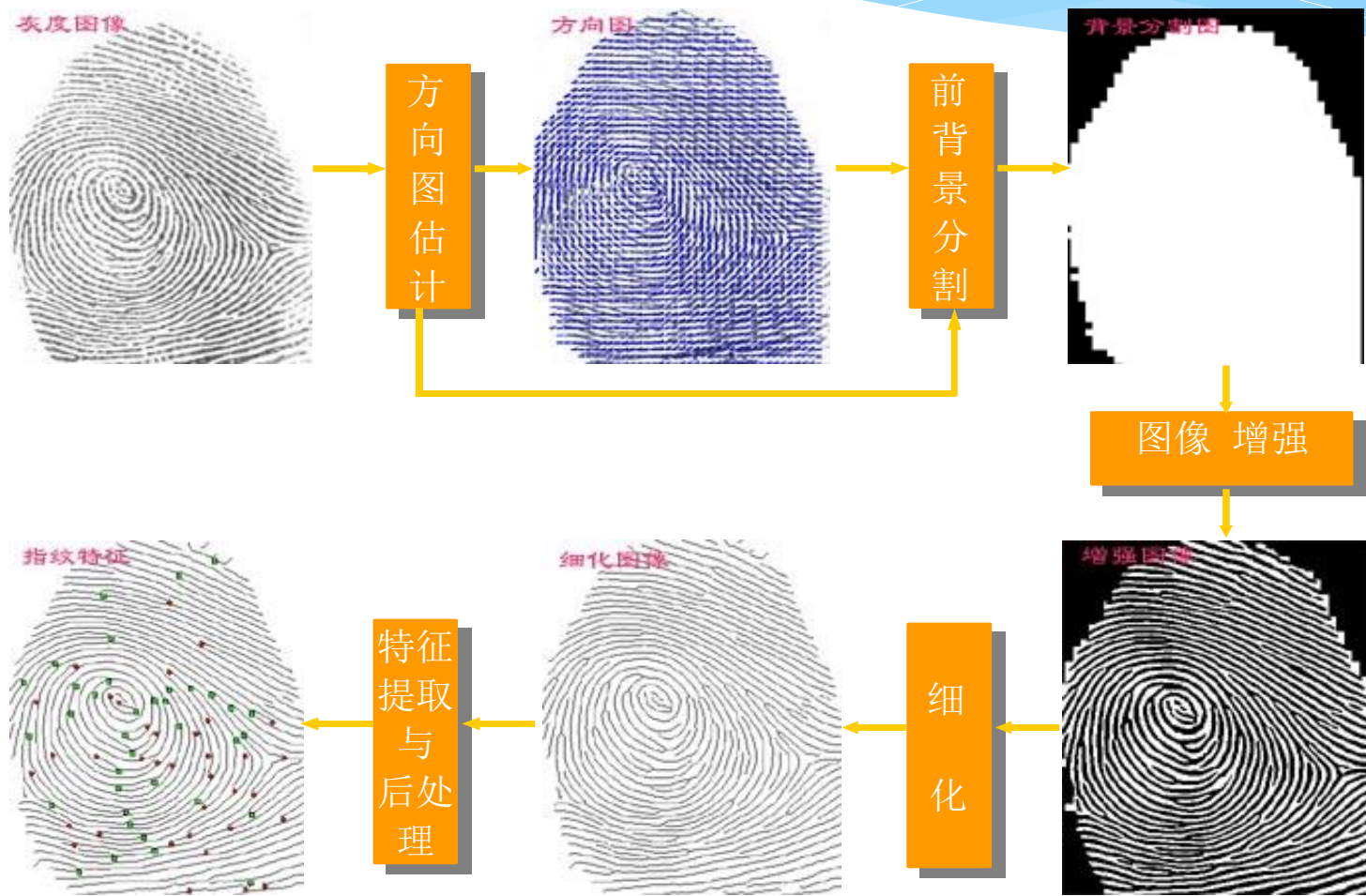
细化

细化图像



指纹特征

指纹识别框图



指纹匹配关键问题二：特征匹配

- * 指纹特征的特点：

- * 尽管一些匹配方法中使用了结构和统计信息，但是指纹特征归根结底仍然是点模式。

- * 点模式匹配的关键问题：

- * 寻找对应点

- * 确定模式之间的相似程度

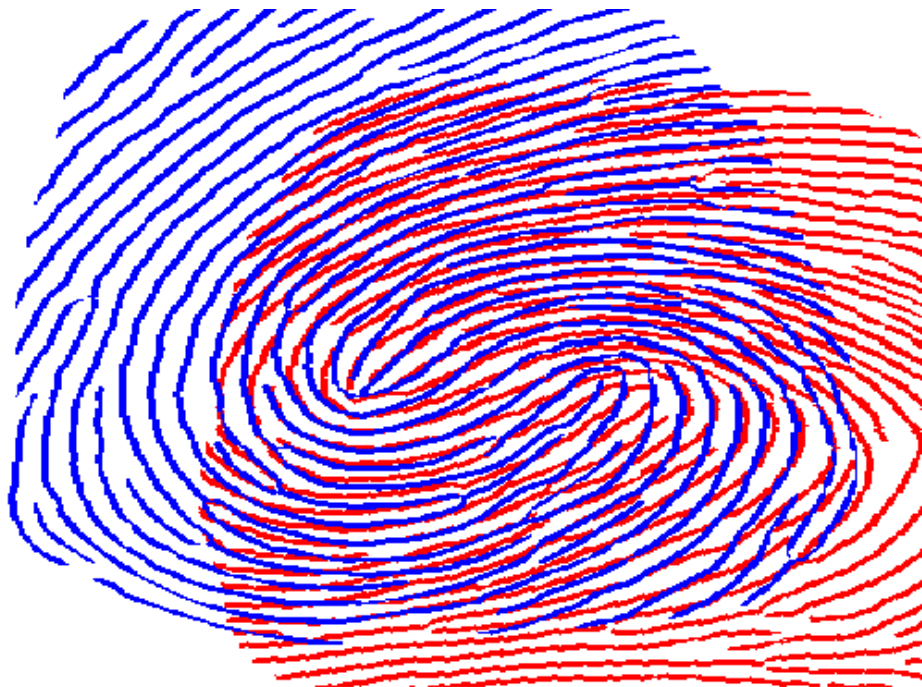
特征匹配

- * 寻找对应点

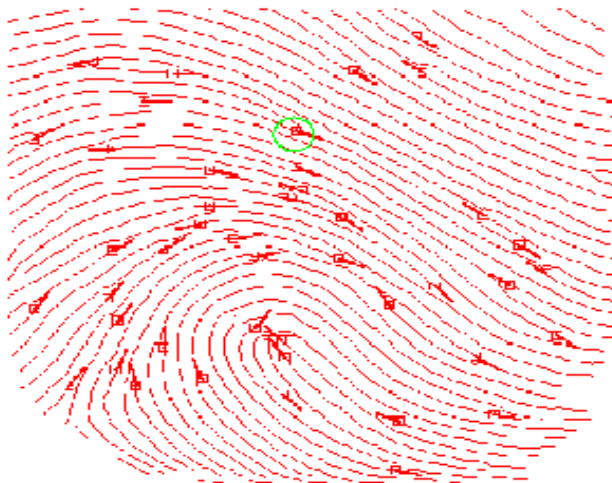
- * 线索：可以利用指纹特征点得的结构和统计信息，如特征点所在纹路的采样或者和周围特征点的关系信息。

特征匹配示意图

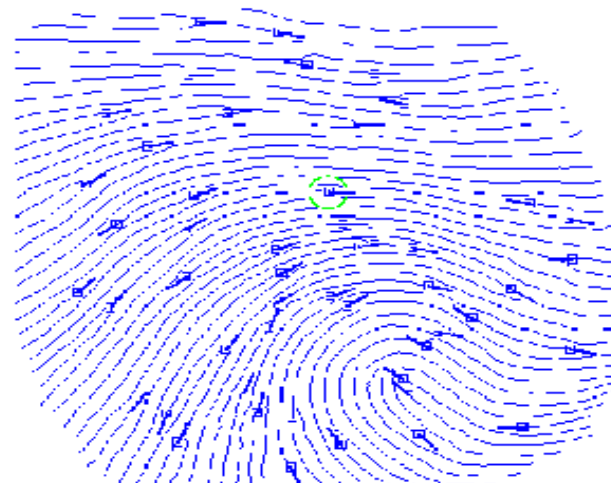
- * 根据对应点参数对两个指纹特征（库中的参考模板和待验证模板）进行旋转和平移操作



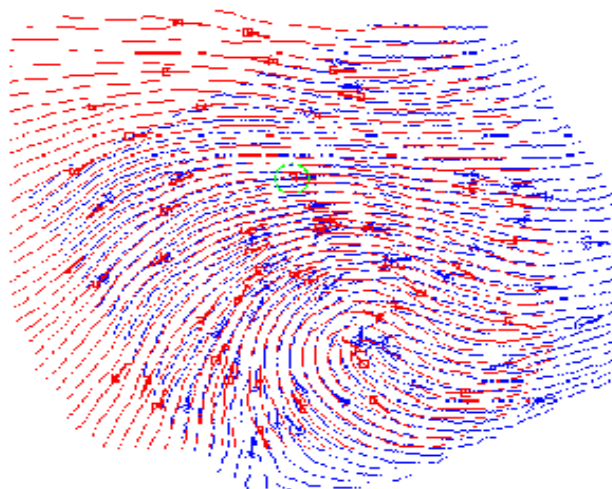
指纹识别的基本流程



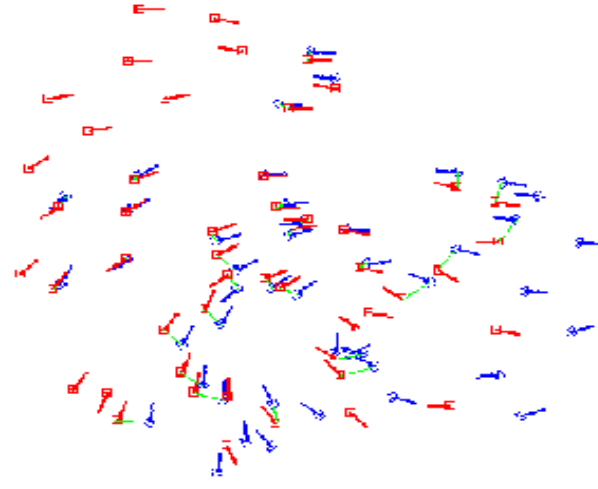
(a)



(b)



(c)



(d)

指纹图像处理

- * 纹路宽度估计
- * 三种常用指纹图像增强滤波器的比较
- * 有效的指纹后处理方法
- * 一种融合的奇异区域提取方法

纹路宽度估计

- * 现有方法

- * 频域分析方法

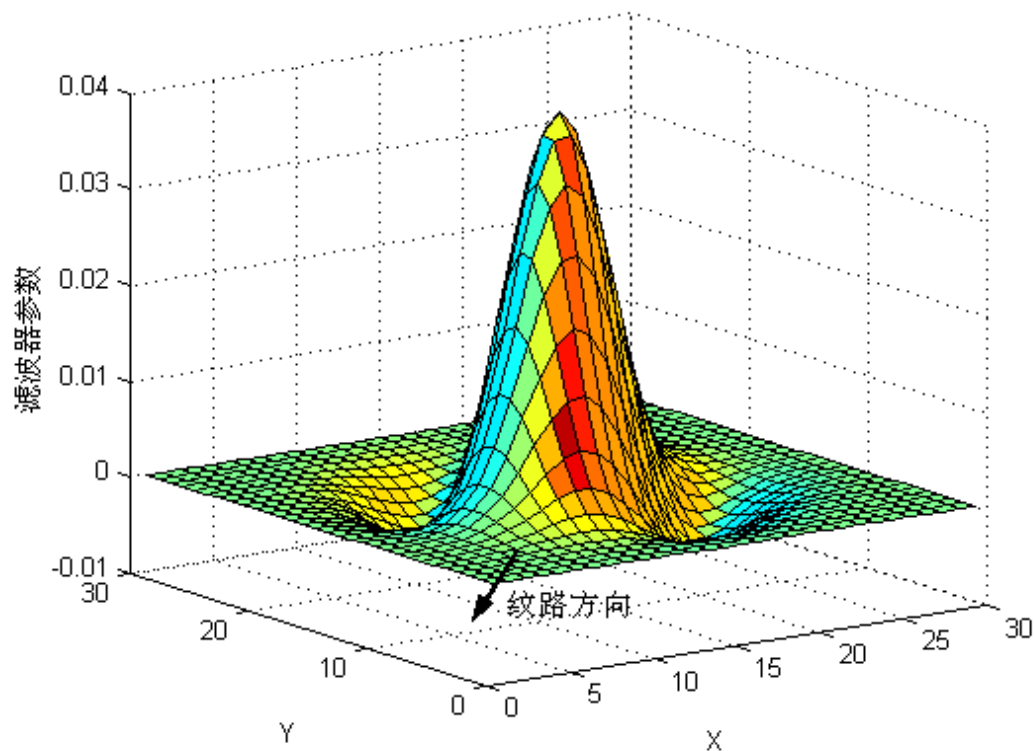
- * 空域分析方法

纹路宽度估计

- * 不易受到纹路噪声的影响
- * 在对投影进行分析的时候考虑到了各种可能的情况，从而保证了算法在规则发生错误时候仍然能够得到正确的结果
- * 通过纹路宽度估计，使得我们可以根据图像本身的特性进行图像处理，从而达到自适应图像分析的目的

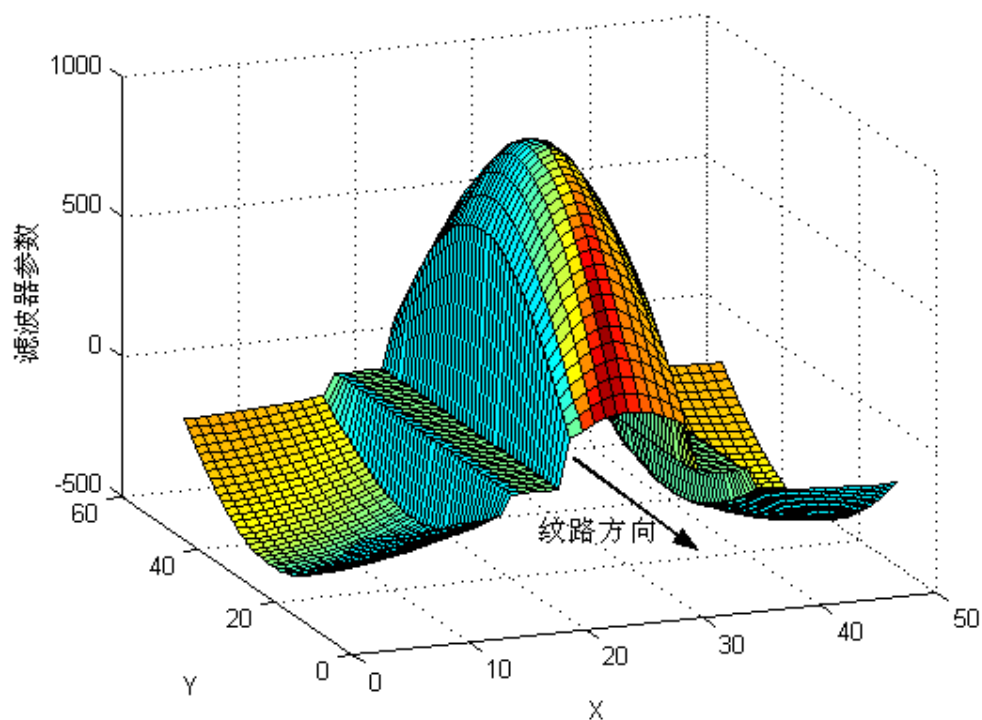
指纹图像增强方法比较

* Gabor滤波器



指纹图像增强方法比较

* 方向自适应滤波器 (O'Gorman)



指纹图像增强方法比较

* Fourier滤波器 (Watson)

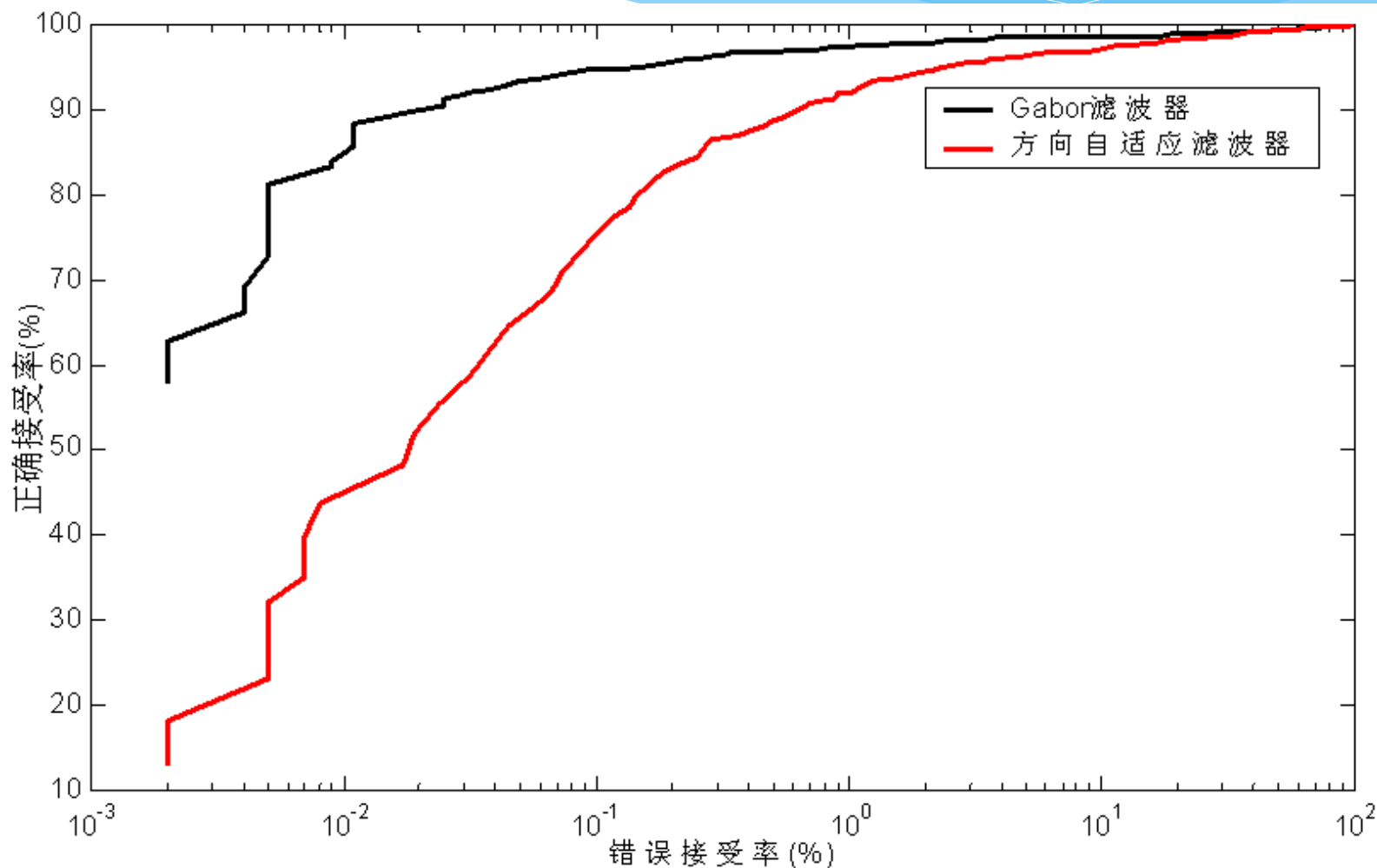
$$X_{jk} + jY_{jk} \longrightarrow \left| X_{jk} + jY_{jk} \right|^{\alpha} (X_{jk} + jY_{jk})$$

- * 基本思想：既然指纹纹路在局部区域内具有很强的方向性，这种规律性在频域有所反映，因此对指纹图像局部区域的Fourier变换，在加强其能量较大的频段的同时抑制能量较小的部分，就可以达到增强指纹规律性的目的。从计算公式上看，该方法能够自适应地增强指纹的规律性，并且规律性越强，增强的幅度越大。

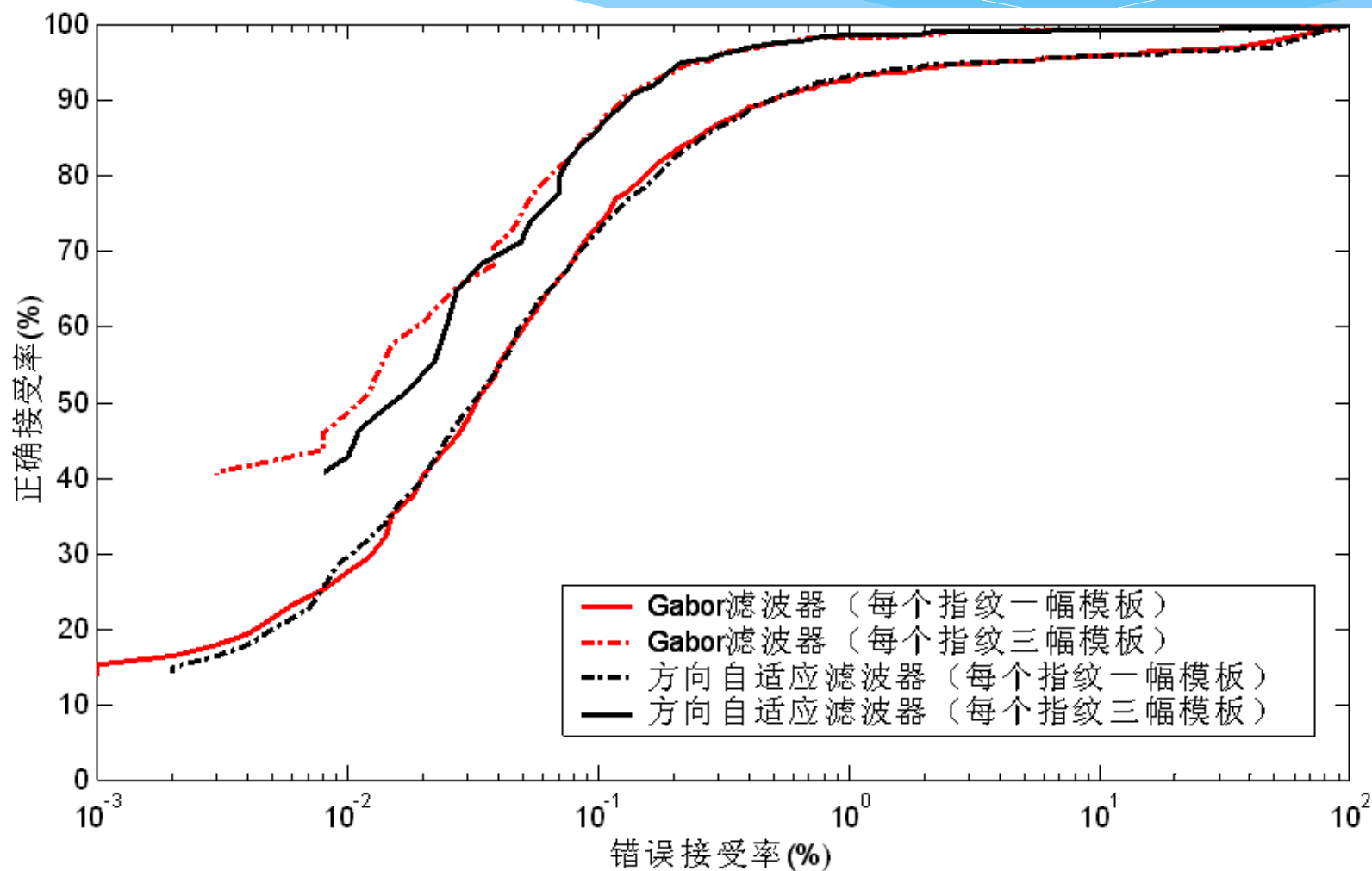




指纹图像增强方法比较



指纹图像增强方法比较



指纹图像增强方法比较

- * 三种滤波器的平滑能力按Gabor、方向自适应滤波器和Fourier滤波器的顺序递减；而捕捉纹路细节的能力也相应地按相反的顺序排列
- * 方向自适应滤波器捕捉细节的能力对系统性能带来的贡献仍然小于其引入的噪声对性能的影响。因此，Gabor滤波器用于指纹图像增强是比较可靠的选择
- * 上述结论也仅适用于通过光学传感技术获得的图像

指纹特征提取

- * 一种有效的指纹后处理方法

$$CN = 0.5 \sum_{i=1}^{i=8} |P_i - P_{i+1}|$$

- * CN为1，表示该点为端点；
- * CN为2表明该点是正常的纹路点；
- * CN为3表明该点是一个分叉点。

P_4	P_3	P_2
P_5	P	P_1
P_6	P_7	P_8

指纹特征提取

- * 细化图像中的**虚假特征**表现为：

- * 纹路的错误断裂和错误粘连

- * 产生的**原因**：

- * 按压的轻重、手指的湿度、皮肤疾病、录入仪的特性、预处理和滤波的影响等等。

- * **解决方法**：

- * 后处理：对细化图上提取出的细节特征进行分析处理，以便鲁棒地提取出所有真实的特征。

指纹特征提取

- * 大部分方法通过分析特征点周围的结构或者统计特征来确定虚假特征点：
 - * Ratha等首先用形态学方法去出毛刺，然后根据3条规则去除虚假特征；
 - * Farina等根据启发式规则去除各种虚假特征，如纹路断裂、小桥和毛刺等，并使用特征点周围的纹路拓扑信息来验证每个纹路的有效性；
 - * Xiao等使用统计数据描述细节结构特征，如邻域中与当前特征相对和相连的特征数目与类型等。

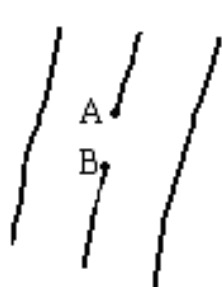
指纹特征提取

* 其它方法包括:

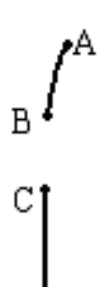
- * Bhanu在特征点邻域内检查方差和均值，并和全图的均值方差作比较，以验证特征正确性；
- * Jiang在后处理过程中加入了特征的置信度信息；
- * O'Gorman和Wilson用逻辑判断或者权值函数计算特征的统计特征以去除虚假细节；
- * Hong定义了一个目标函数来描述相对的细节特征，并用对目标函数取阈值方法来去除虚假细节；
- * Prabhakar用监督的学习向量量化器（Learning Vector Quantizer）对特征和非特征进行学习以验证特征的有效性。

指纹特征提取

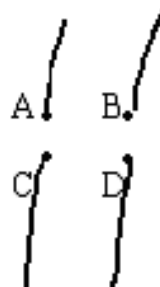
* 基本思想：通过定位虚假特征结构来定位虚假特征。



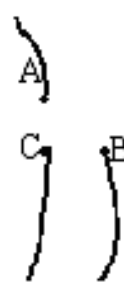
(a)



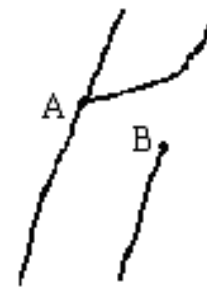
(b)



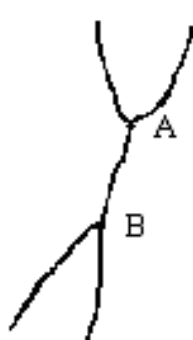
(c)



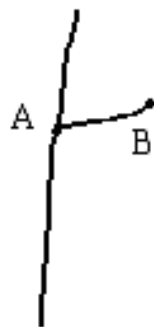
(d)



(e)



(f)



(g)



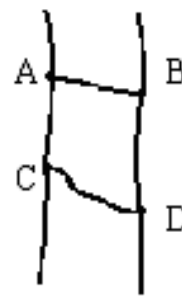
(h)



(i)



(j)



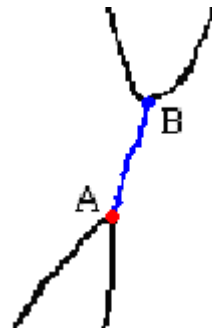
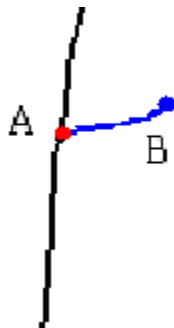
(k)

指纹后处理

- * 一个好的后处理算法应满足：
 - * 应对所有的特征点进行处理，以区分真实特征和虚假特征
 - * 为了降低计算复杂性，每步应只处理一种虚假特征
 - * 前面的处理不应破坏后续处理所需的信息
 - * 如果前面的处理中引入了新的虚假特征，应在后续处理中将其去除

指纹后处理

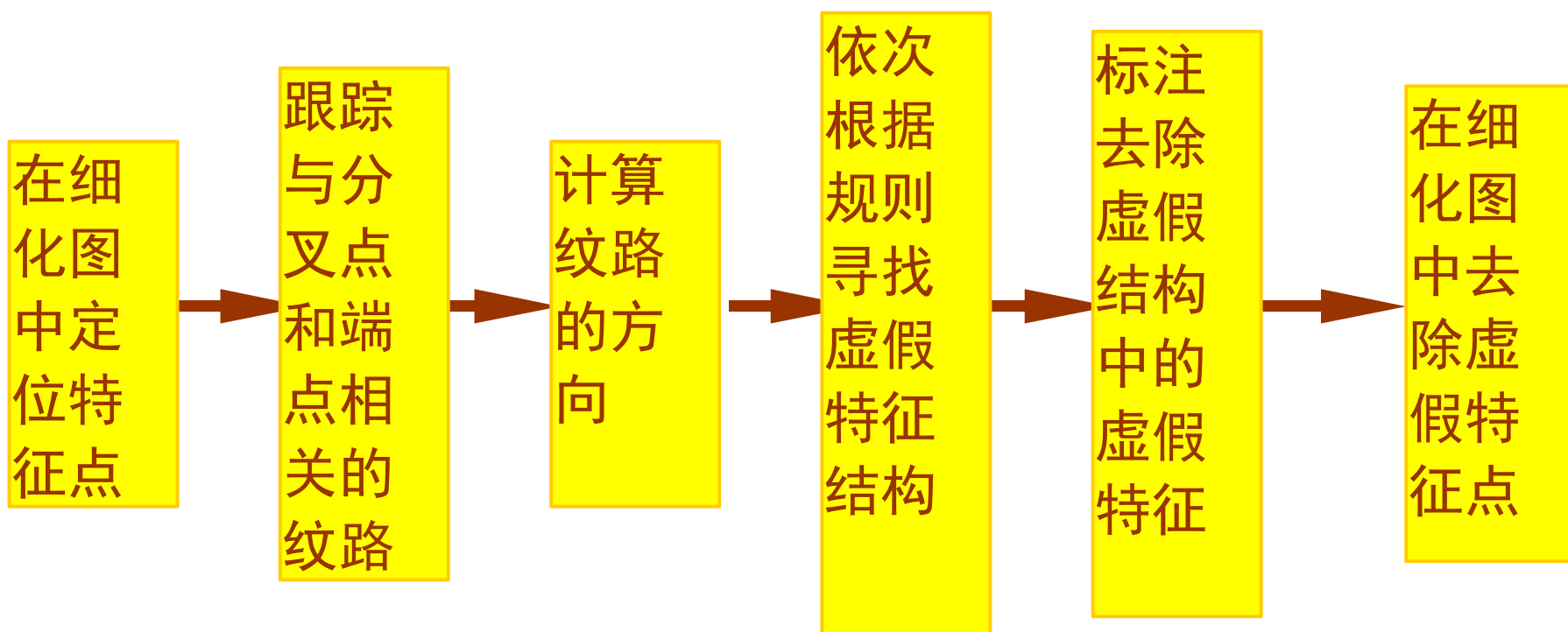
- * 在细化图上**定位每个特征点**（分叉点和端点）
- * 跟踪和每个分叉点相关的三条分叉纹路；跟踪和每个端点相关的纹路。跟踪结束的条件为：
 - * 跟踪点数超过某个预先设定的值
 - * 遇到另外一个特征点



指纹后处理 - 基本假设

- * 假设1: 指纹基本上是由相互平行的纹线组成的, 作为正常纹路中的异常情况, 特征在少数区域出现。因此, 一副指纹图像上真实特征的数据是有限的, 一般不多于100个。
- * 假设2: 平行的纹路之间不会出现特征。因此, 两个特征之间的距离不应大于纹路的间距。

指纹后处理 - 算法描述



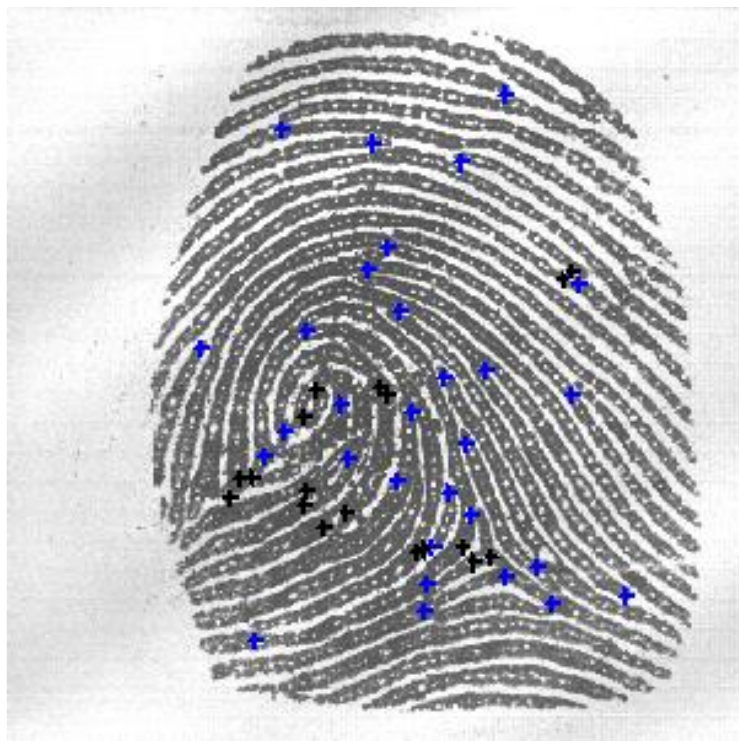
指纹后处理 - 实验条件

- * 图像大小：300*300
- * 录入仪：Veridicom公司
- * 计算机主频：Pentium III 450MHz

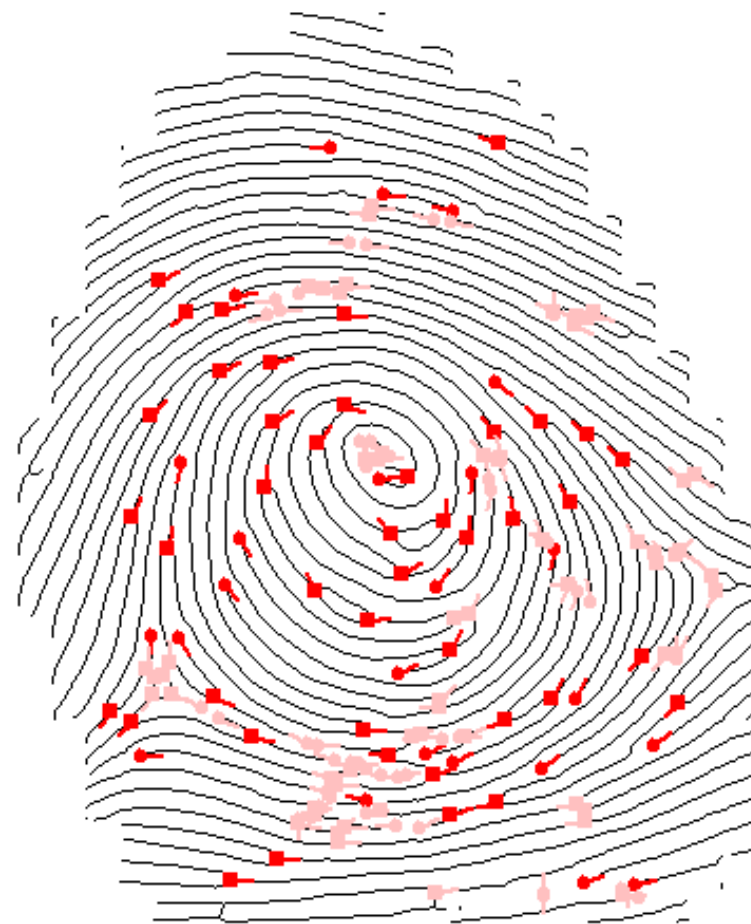
指纹后处理 - 实验结果



指纹后处理 - 实验结果



指纹后处理 - 实验结果



指纹后处理 - 实验结果

编号	指纹类型	处理前特征数	处理后特征数	剩余的虚假特征数	未检测出的真实特征数	虚假特征的去除比例	真实特征的检测比例
1	Whorl	87	45	3	1	93%	98%
2	Central Pocket	46	41	0	2	100%	95%
3	Whorl	70	42	1	1	97%	98%
4	Whorl	56	36	3	1	87%	97%
5	Right Loop	41	28	1	1	93%	96%
6	Whorl	99	47	6	3	90%	93%
7	Left Loop	62	37	0	2	100%	95%
8	Whorl	69	45	3	2	92%	95%
合计		530	169	17	13	96%	92%

指纹后处理 - 实验结果

编号	指纹类型	处理前特征数	处理后特征数	剩余虚假特征数	未检测出的真实特征数	虚假特征的去除比例	真实特征的检测比例
合计		1654	576	44	54	96.1%	90.8%

* 计算效率上:

* Pentium III 450 MHz, 128M内存的PC机上运行
时间不超过10ms

奇异点检测

- * 奇异点检测在指纹身份鉴别中有很多应用：如奇异点的位置关系可以用于指纹的分类，奇异点也可以作为两个指纹的对应点。
- * 奇异点的检测方法中最重要的是Poincaré Index的方法。其他还有基于邻域的剪枝方法、基于模板匹配的方法和神经网络的方法。

奇异点检测

- * 融合局部方向方差和Poincaré index的奇异区域检测方法
- * 根据中心点的定义，中心点是指纹中纹路呈U型分布的平行纹路中最内侧的最高（低）点，也就是曲率最大的点。通过对细化图像跟踪实现中心点的精确定位

奇异点检测



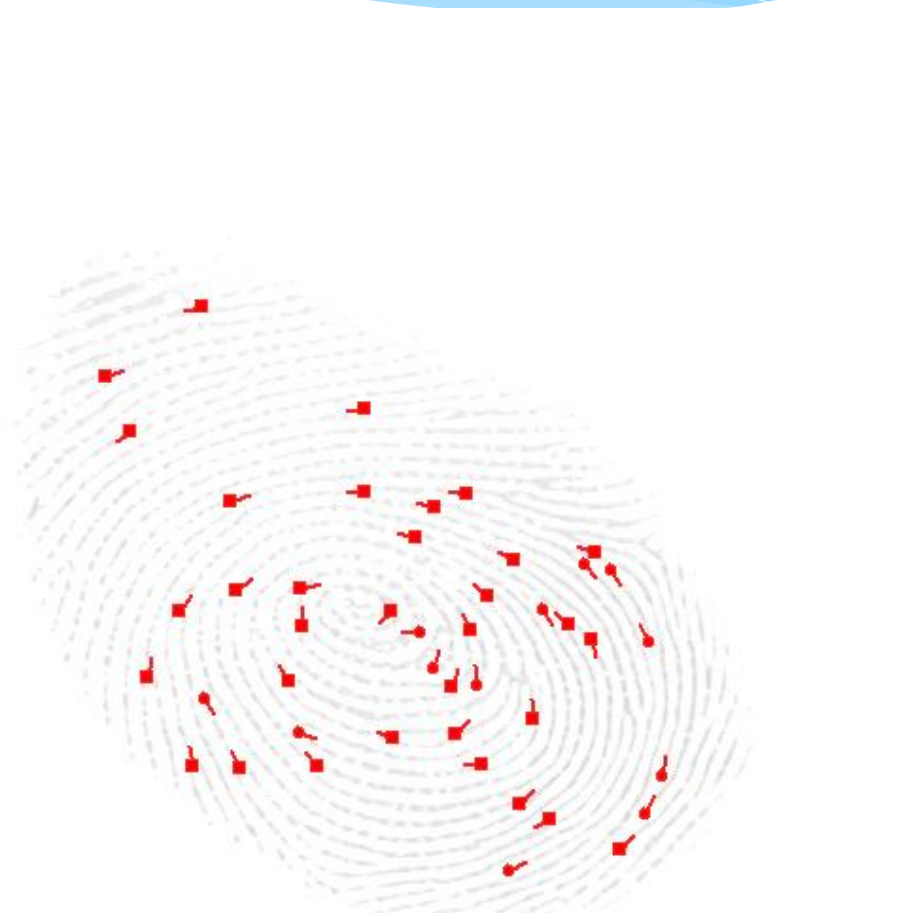
奇异点检测

数据库	局部方差法		Poincaré Index		融合方法	
	虚警率	漏检率	虚警率	漏检率	虚警率	漏检率
NIST	11.6%	1.2%	12.7%	3.7%	4.3%	4.3%
NLPR	16.7%	1.9%	23.8%	3.8%	1.9%	4.1%

奇异点检测



指纹特征匹配 - 基于误差扩散的指纹匹配



基于误差扩散的指纹匹配

* 难点

- * 形变：旋转、平移、放缩以及非线性形变
- * 参考模板和输入模板的对应点未知
- * 特征提取算法可能会引入度量误差；图像处理也会受到噪声的影响

基于误差扩散的指纹匹配

- * 研究现状

- * 大部分算法采用两步匹配：

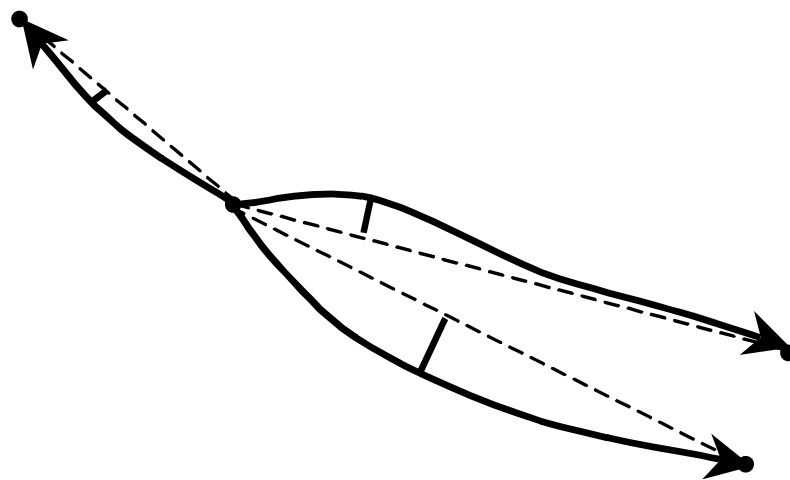
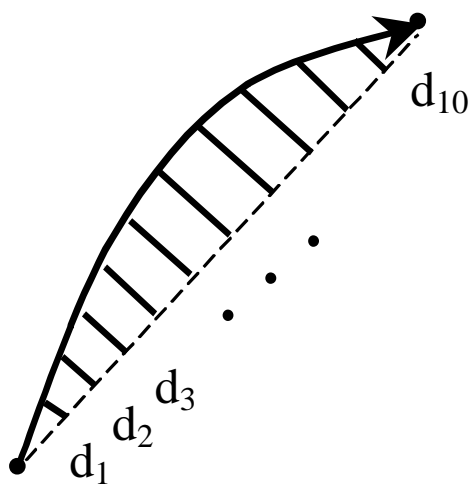
- * 寻找对应点：纹路信息、方向图、特征点与周围特征的关系等

- * 匹配：串匹配、图匹配、树匹配、三角匹配等

基于误差扩散的指纹匹配

- * 特征点的相似度量
- * 修正的Hough变换
- * 公共区域估计
- * 基于误差扩散的匹配

基于误差扩散的指纹匹配



- 特征点
- 纹路跟踪终止点

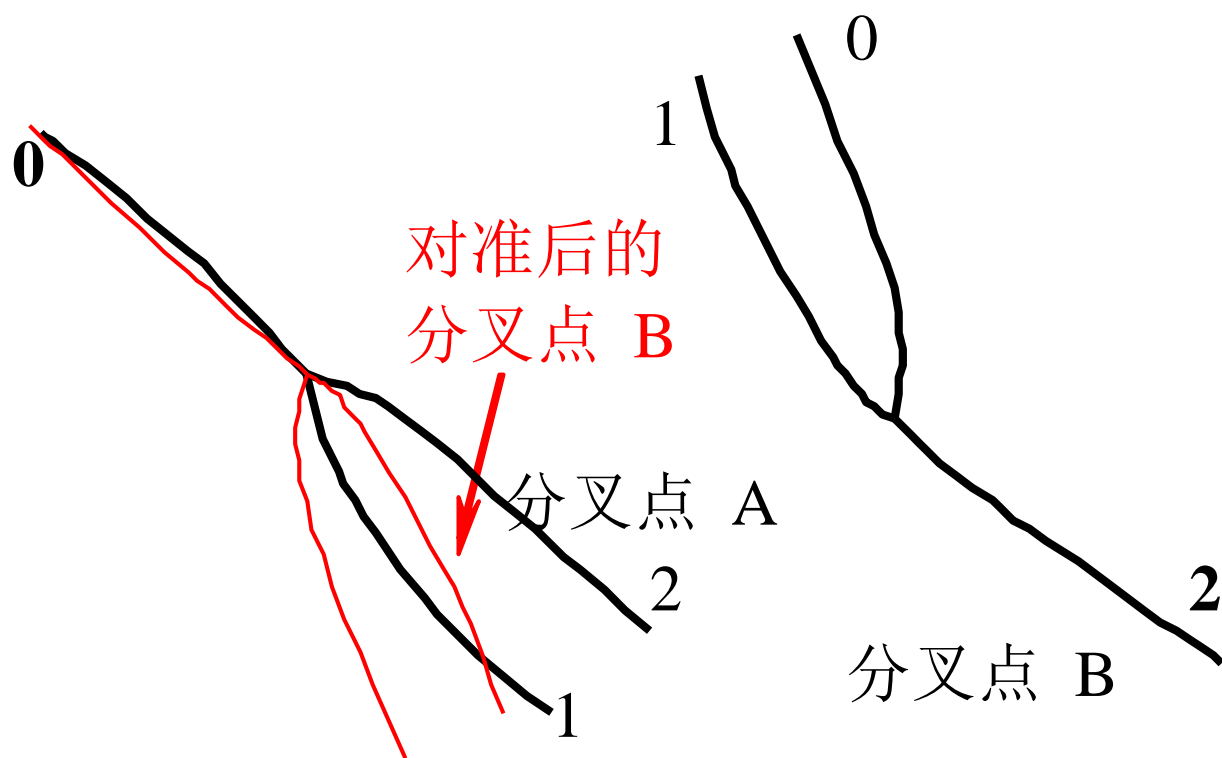
基于误差扩散的指纹匹配

* 对于端点对:
$$S = \frac{\sum_{i=0}^L (d^i - D^i)^2}{L}$$

* 对于分叉点对:
$$S = \frac{\sum_{j=0}^2 \frac{\sum_{i=0}^{L_j} (d_j^i - D_{(k+j)\%3}^i)^2}{L_j} + \Delta\theta}{4}$$

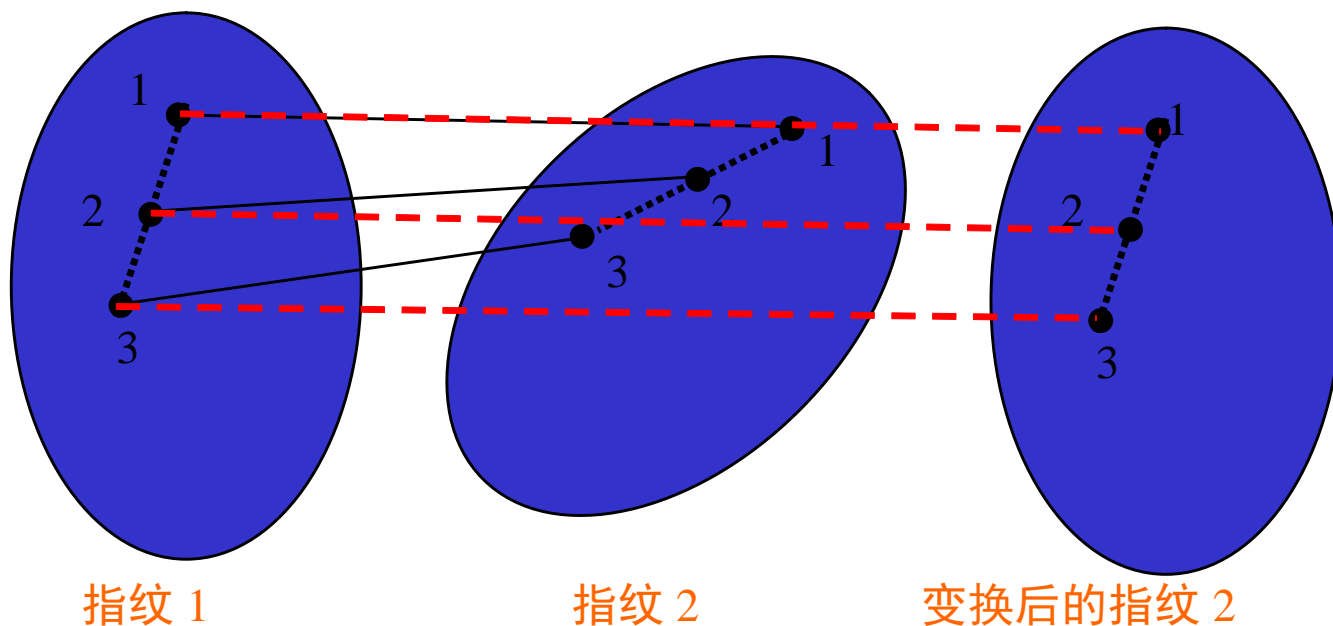
$$\Delta\theta = \min_{i=0}^{i=2} (|\theta_1 - \Theta_{(i+1)\%3} - (\theta_0 - \Theta_i)| + |\theta_2 - \Theta_{(i+2)\%3} - (\theta_0 - \Theta_i)|)$$

基于误差扩散的指纹匹配



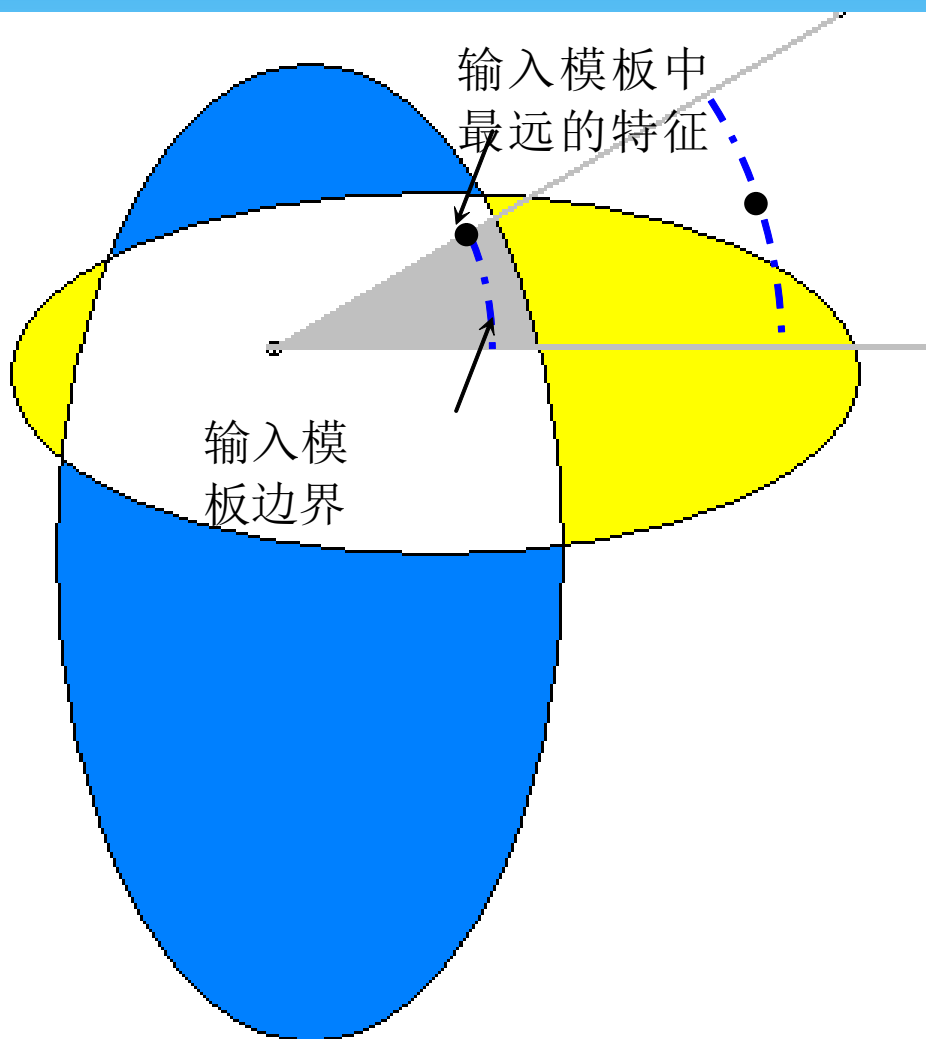
基于误差扩散的指纹匹配

- * 对应点估计 – Hough变换，以找到一对或若干对最可靠的对应点



基于误差扩散的指纹匹配

- * 估计公共区域
- * 只对比公共区域的匹配程度



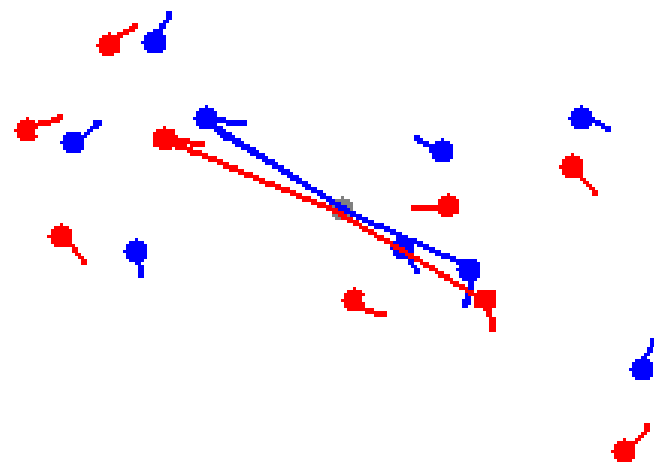
基于误差扩散的指纹匹配

- * MatchedSet初始化

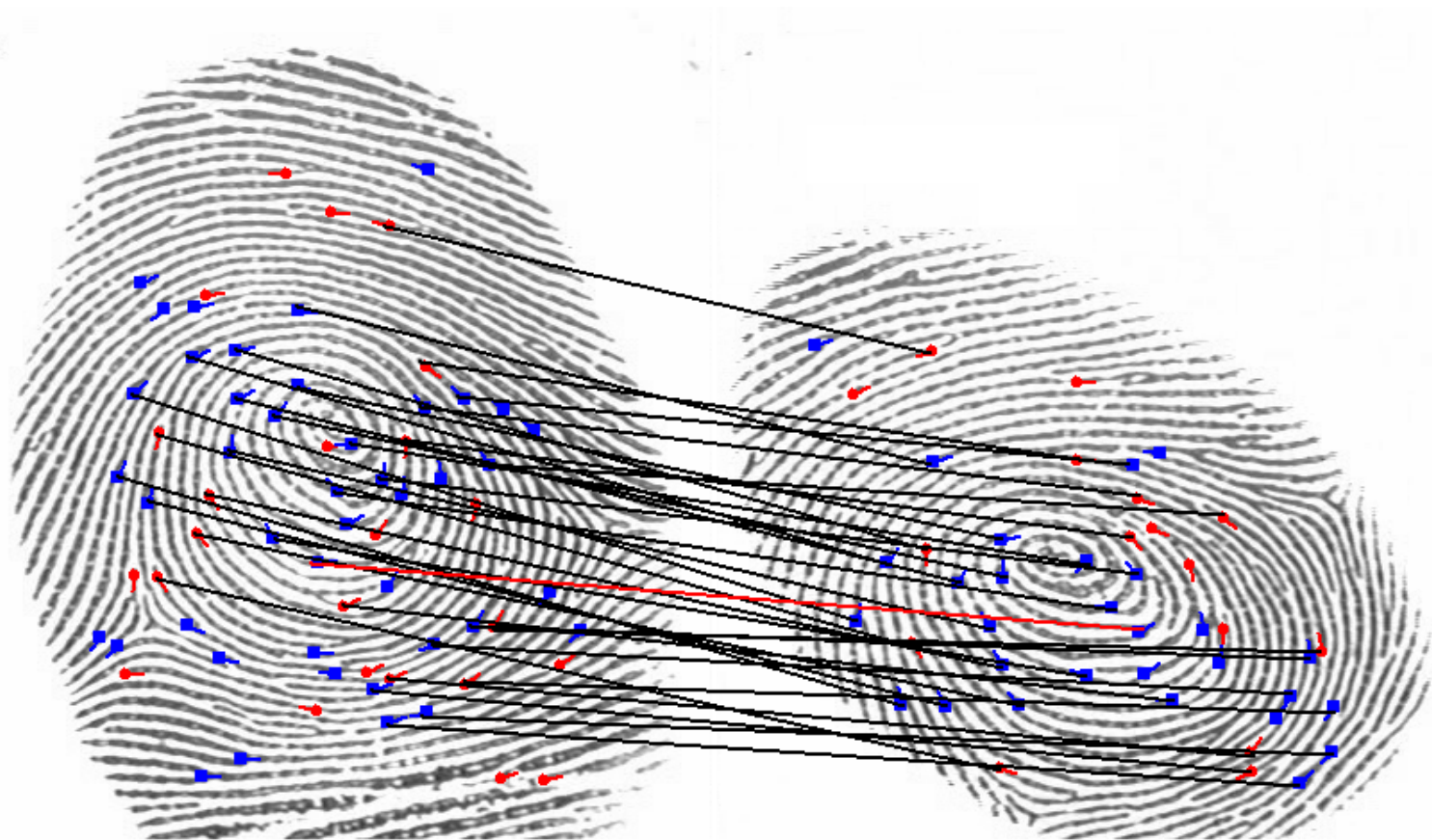
- * 对MatchedSet进行初始化可以提高算法的效率。
- * 如果不进行MatchedSet进行初始化，那么误差扩散可依据的只有一对对应点，而无任何匹配误差信息，从而使扩散失去根据

基于误差扩散的指纹匹配

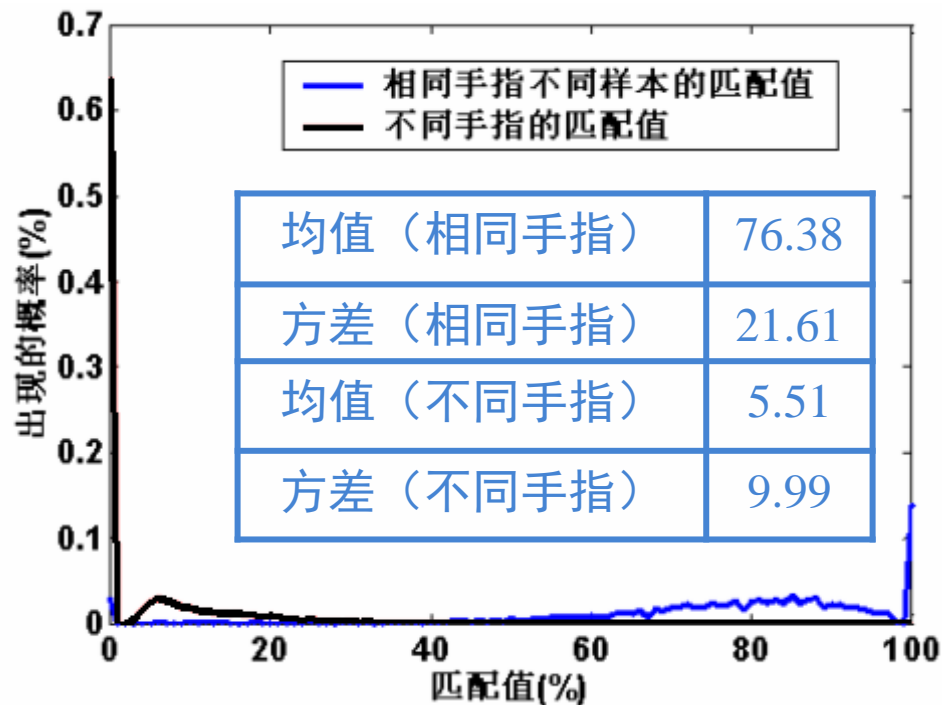
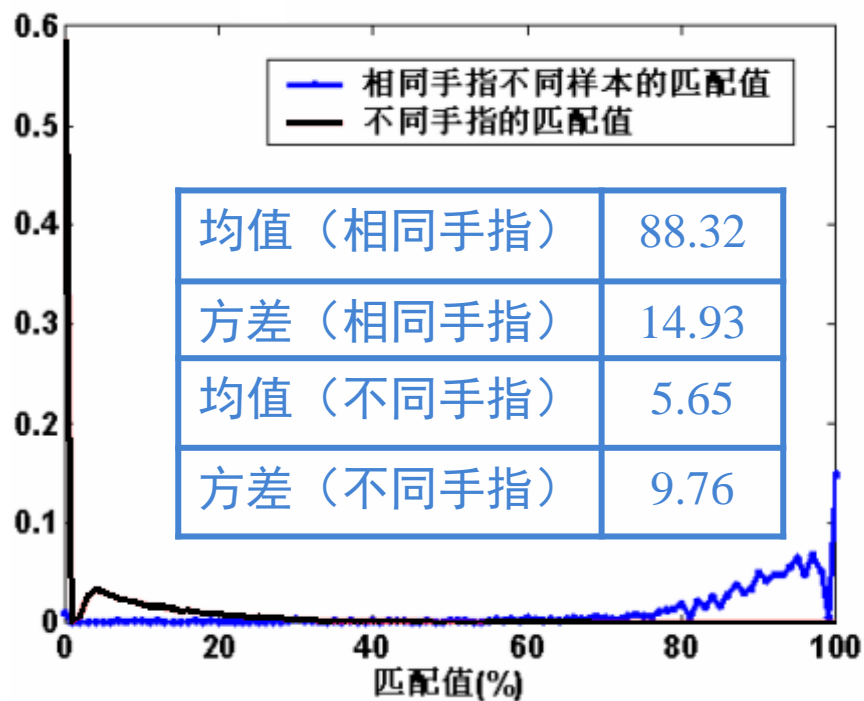
- * 对MatchedSet中的每个已匹配点寻找距离其最近的若干个待匹配点
- * 根据已匹配点的匹配误差估计待匹配点的误差
- * 当MatchedSet被遍历后，验证每一对点的有效性，删除那些被重复匹配的点
- * 继续上面的3步，直至MatchedSet不再增加为止
- * 两枚指纹的相似度定义为：
$$S = 100 \sqrt{\frac{M \times M}{n_t' N_l'}}$$



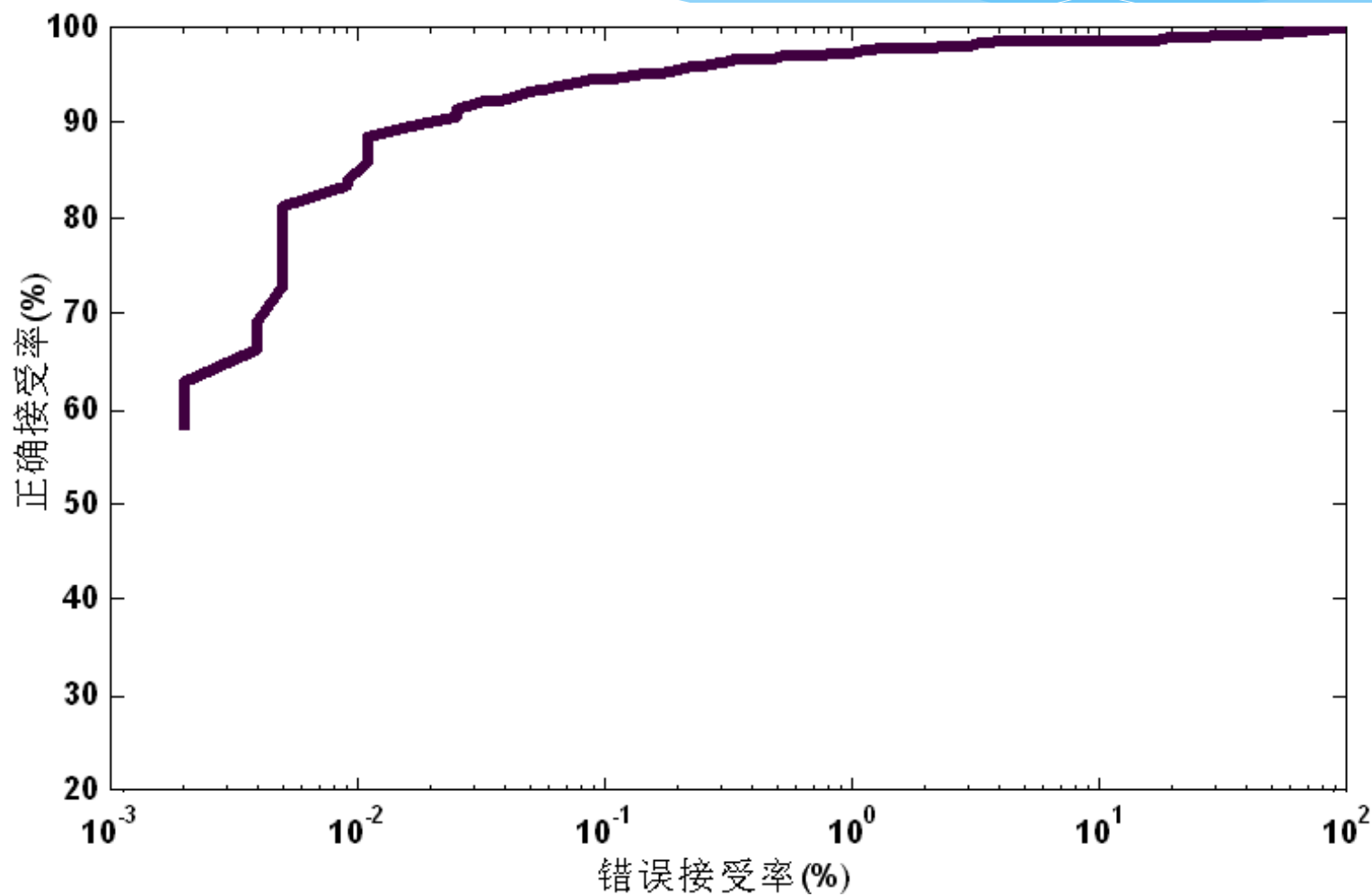
基于误差扩散的指纹匹配



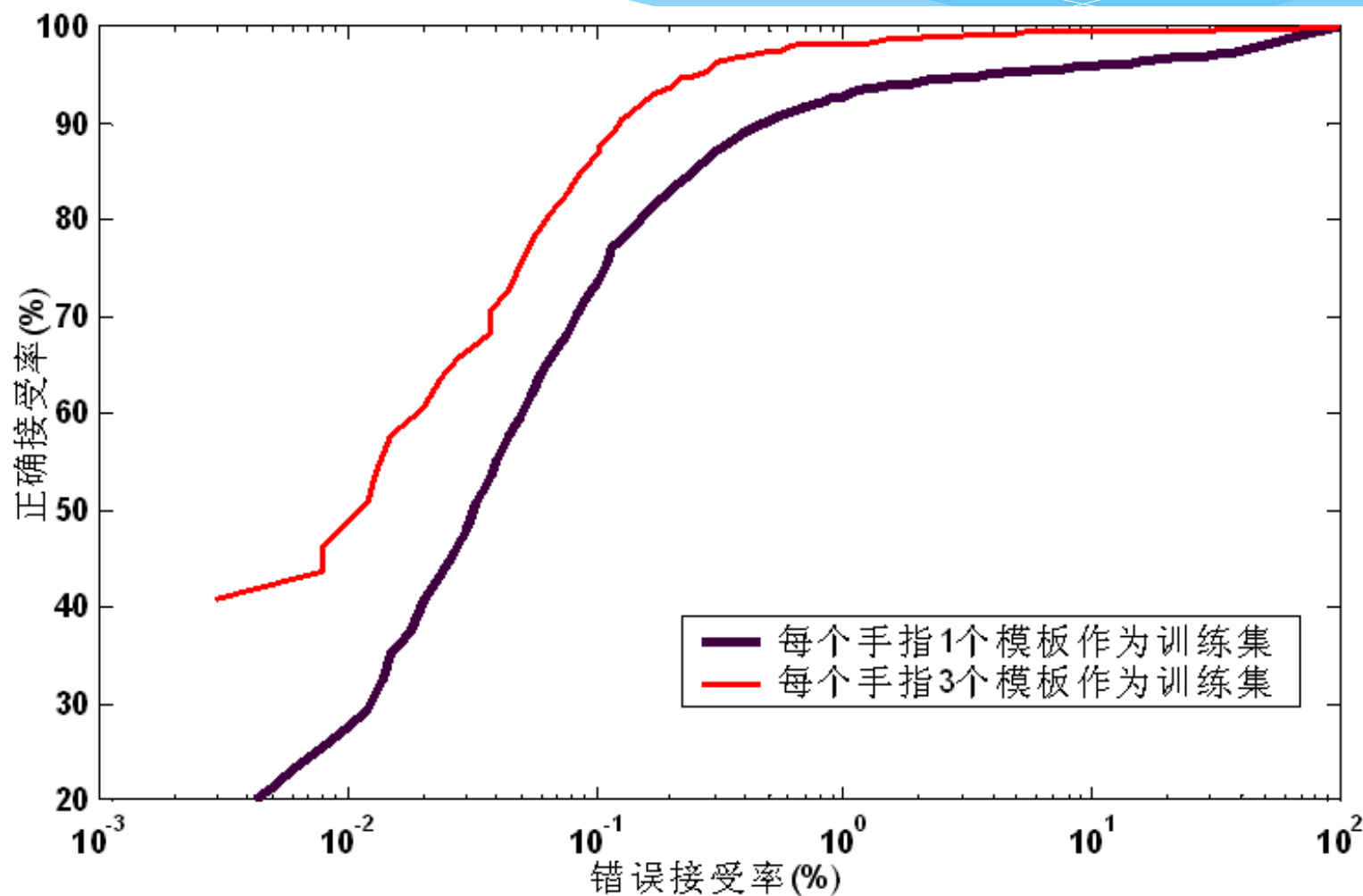
基于误差扩散的指纹匹配



基于误差扩散的指纹匹配



基于误差扩散的指纹匹配



基于误差扩散的指纹匹配

* 优点

- * 同时使用端点和分叉点寻找对应点
- * 采用修正的Hough变换方法
- * 将指纹在公共区域的相似度作为指纹的相似度
- * 采用基于误差扩散的匹配方法

谢谢