**研究历史**

Jan Evangelista Purkinje (1787–1869)，1823年，一位捷克生理学家兼布列斯劳大学解剖学教授发表了一篇论文，讨论了9种指纹模式，但他没有提到使用指尖的可能性去确认人的身份。

几年后，德国解剖学家乔治·冯·梅斯纳(Georg von Meissner，1829-1905)研究了摩擦脊，

五年后，1858年，威廉·詹姆斯·赫歇尔爵士(Sir William James Herschel)开始在印度进行指纹识别。

Juan Vucetich，阿根廷一名警察局长创造了记录个人指纹的第一种方法。

在美国，Dr. Henry P. De Forrest于1902年在纽约公务员系统使用指纹。

随着指纹检测的出现，许多犯罪分子为了避免留下指纹，不得不戴上手套，从而使犯罪调查更加困难。

**项目概况**

指纹识别系统是在生物识别应用下使用的，提高了用户的安全性。

该系统对指纹数据进行处理，并采集指纹的识别特征。

接下来，将此信息与先前存储的各种指纹信息进行比较。

在进行比较之后，系统确定输入图像是否与数据库中已经存在的指纹数据相匹配。

**与已有的算法进行比较**

**实现FR系统**

**实验结果**

**方法：**

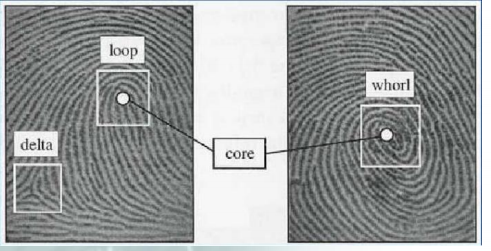
指纹图案由一系列山脊和山谷组成。

在指纹图像中，脊以暗线的形式出现，而山谷是脊之间的光区。

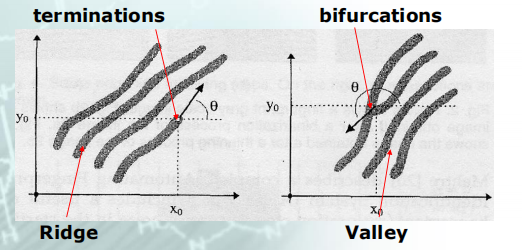
指纹图像将有一个或多个区域，其中脊线有一个独特的形状。这些形状的特点通常是高曲率区域或频繁的脊端，并被称为奇异区域。

这些奇异区域的三种基本类型是loop环、delta三角区和whorl螺旋区。

这些形状通常由高曲率的区域或经常的脊末端来表征，并且被称为奇异区域。



**基于细节的方法：**

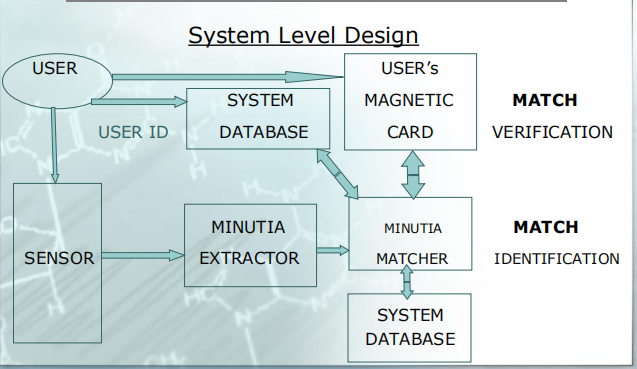
  
Terminations：终端

Bifurcations：分支

Ridge：山脊

Valley：山谷

**系统级设计：**



**程序：**

二值化：

它是将灰度图像转换成黑白图像的过程。

在灰度图像中，像素可以接受256个不同的灰度值，而每个像素被指定为黑白图像。



细化：

用块滤波方法进行的细化方法试图保留每个脊的最外层像素。

这些步骤将形态学处理应用于图像，以减少脊。两个形态过程是膨胀和侵蚀。



后处理步骤Post-Processing Steps：

删除假细节：False Minutia Remove:



**摘要：**

我们的团队项目是研究并实现一种基于细节点匹配的指纹识别系统，该系统经常用于各种指纹算法和技术中。该方法主要涉及从样本指纹图像中提取细节点，然后根据两个指纹之间的细节点对数进行指纹匹配。

我们的实现主要包括图像增强、图像分割、特征(细节)提取和细节匹配。它最终产生了一个百分比的分数，这表明是否有两个FINGING指纹是否匹配。该项目用MATLAB编写。

**介绍：**

生物识别是指使用独特的生理(例如。指纹，掌纹，虹膜，面部)和行为(如。(步态、特征)特征，称为识别个体的生物特征识别器

指纹识别是用于个人身份识别的最古老、最可靠的生物特征之一。指纹识别已经使用了100多年，并且已经从乏味的手工指纹匹配中走了很长一段路。古代手工匹配指纹的程序非常繁琐，耗时，需要熟练的人员。

手指皮肤是由摩擦脊和汗孔组成的，这些毛孔都是沿着这些纹路形成的。摩擦脊是在胎儿生命中产生的，只有一般形状是遗传定义的。区别人的身体特征的本质是由于人类固有的个体遗传多样性以及影响胚胎发育的随机过程。摩擦脊在一个人的成年生活中保持不变。他们可以重建自己，即使在受伤的情况下，只要伤害不太严重。

指纹是最成熟的生物识别技术之一，在世界各地的法庭上被认为是合法的证据。最近，越来越多的民用和商业应用正在使用或积极考虑使用指纹识别，因为廉价和紧凑的固体统计数据。与其他生物识别技术相比，e扫描器及其优越且经证明的匹配性能。

与指纹识别系统有关的一些重要术语解释如下：

1.指纹采集：如何获取指纹图像，以及如何以适当的机器可读的格式表示指纹图像。

2.指纹鉴定：确定两个指纹是否来自同一个手指

3.指纹识别：在数据库中搜索查询指纹

4.指纹分类：将给定的指纹按其几何特征分配给预先指定的类别之一

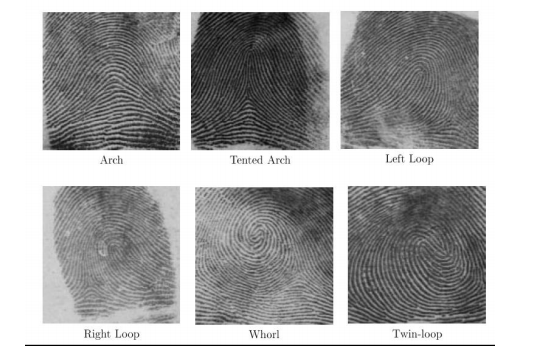
在指纹识别和指纹验证系统的情况下，我们的任务将分为两个阶段：

1. 离线相位：先由特征提取模块采集和处理被验证人指纹的几幅指纹图像；提取的特征作为模板存储在数据库中，供以后使用。

2．在线阶段：被验证的个人给出他/她的身份(在验证系统的情况下)，并将他/她的手指放在无墨指纹扫描仪上，从捕获的指纹图像中提取细节点。

然后将这些细节输入到匹配模块，该模块将它们与数据库中的自己的模板(如果是验证系统)或针对数据库中的所有用户(在身份识别系统的情况下)

因此，在指纹识别系统中，我们需要将输入的指纹与数据库中每个其他用户的存储模板进行匹配。为了减少这种计算和搜索开销，必须有一种指纹5分类系统，这将有助于我们严格限制数据库的大小，我们需要提取细节特征并匹配传入的指纹样本。因此，指纹可以基于宏观水平的几何结构被分类为六个类别中的一个。



因此，在此基础上对指纹进行分类，可以使待检索的数据库更小。此外，如果使用多个手指的图像，我们将有更多的类别，而要搜索的数据库的大小将进一步缩小。

（2）指纹匹配技术

大量的指纹匹配方法可以粗略地分为三大类。

基于相关的匹配：对两个指纹图像进行叠加，并针对不同的对准(例如，各种位移和旋转)计算相应像素之间的相关性。

基于细节点的匹配：这是目前最流行、应用最广泛的指纹比对技术，是指纹检验人员进行指纹比对的基础。从这两个指纹中提取细节点，并将其存储在二维平面上的点集中。基于细节点的匹配实质上是寻找模板和输入细节集之间的对齐，从而得到最大数量的细节对。

基于模式(或基于图像)的匹配：基于模式的算法比较了以前存储的模板和候选指纹之间的基本指纹模式(ARCH、WILL和循环)。这要求图像以相同的方向对齐。为此，该算法在指纹图像中找到一个中心点，并以此为中心。在基于模式的算法中，模板包含对齐指纹图像中模式的类型、大小和方向。将候选指纹图像与模板进行图形比较，以确定它们匹配的程度。

在我们的项目中，我们实现了一种基于细节点的匹配技术。这种方法已经得到了深入的研究，也是目前现有指纹识别产品的骨干。

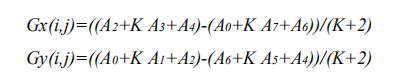
**理论综述与实施细节**

**1.1脊向**

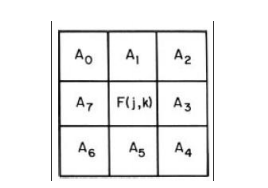
脊向是在整个图像中获取脊的角度的过程。在WxW块体的块状基础上，计算了脊向.W一般等于16。

脊向的第一步是计算图像每个像素的梯度。

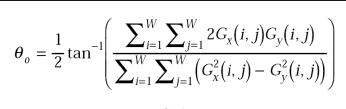
用Sobel算子计算了Gx(i，j)和Gy(I，j)的梯度。Sobel算子如下：



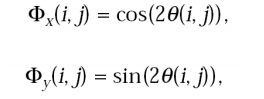
其中K=2表示Sobel算子。像素AI的含义如下：



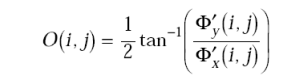
脊向的下一步是使用所有这些像素的梯度值来计算W\*W块的脊角，如下所示：



由于噪声的存在，山脊和山谷结构的破坏，细节等。在输入图像中，估计的局部脊方向q(i，j)可能并不总是正确的。由于在不出现奇异点的局部邻域中，局部脊方向变化缓慢，因此可以使用低通滤波器来修正不正确的局部脊方向。为了执行低通滤波，需要将定向图像转换成连续矢量场，定义如下：



该连续矢量场通过一个低通高斯型滤波器，得到改进的定向图像。最后，利用以下公式从滤波的向量场中计算出局部方向：



**1.2图像增强**

劣质形象的含义：

可能会产生大量虚假的细节，

很大一部分真正的细节可能会被忽略。

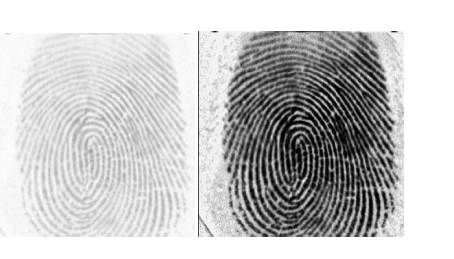
它们的定位(位置和方向)可能会出现很大的误差。

增强算法提高了指纹图像中脊和谷结构的清晰度。以下步骤构成增强算法：

1. 正常化：输入图像被归一化，使得脊和谷容易被区分。
2. 局部方向估计：从归一化指纹图像中计算出脊线的方向，在每个块中求出所需的大小。
3. [计]频率估算：利用归一化图像和定向图像，计算图像中每个块的频率。
4. 利用一组调整局部脊线方向和频率的Gabor滤波器对图像进行滤波，将脊线和谷线清晰地分离出来，从而降低了伪细节的概率。

**1.3归一化**

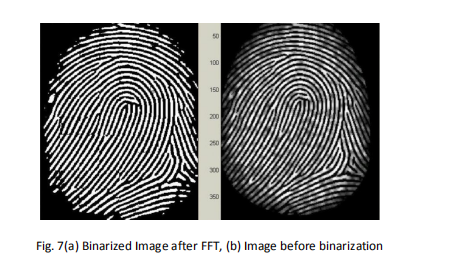
对于归一化，我们进行了简单的直方图均衡化，通过变换指纹图像中的值来增强图像的对比度。



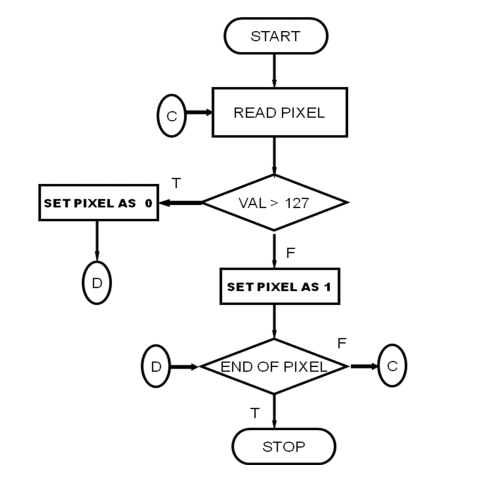
**1.4图像二值化**

图像二值化是一种将8位灰度图像转换为1位图像的过程。操作结束后，指纹中的脊被突出显示为黑色，而皱纹则是白色的。

采用局部自适应二值化方法对指纹图像进行二值化。在该方法中，图像被划分为16×16像素的块。如果像素值大于像素所属的当前块的平均强度值，则像素值设置为1(图3.4)。

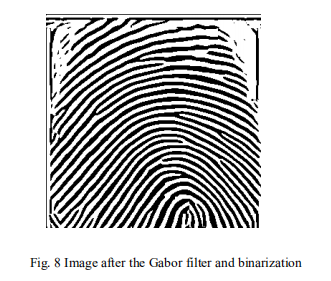


二值化流程图

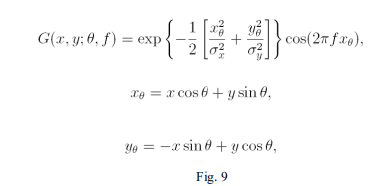


1.5过滤

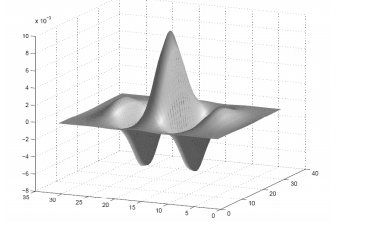
山脊和山谷的正弦波在局部恒定方向上变化缓慢。因此，调谐到相应频率和取向的带通滤波器可以有效地去除不希望的噪声并保留真正的脊部和谷地结构。Gabor滤波器具有频率选择性和方位选择性，在空间和频域都具有最佳的联合分辨率。因此，采用Gabor滤波器作为带通滤波器，消除噪声，保持真实的岭/谷结构是合适的。



Gabor滤波器的结构如下：



在我们的实现中，Gabor滤波器所需的频率f为常数，角度输入由方向图提供。这是方向图有用的地方。频率值为1/7，结果满意。在空间域中，Gabor过滤器如下所示：



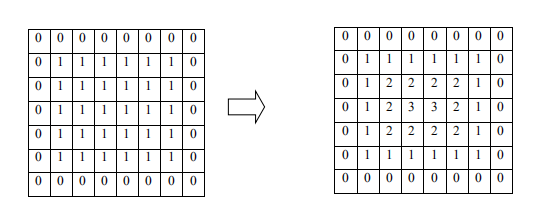
为了得到上述增强图像中所示的输出图像，需要对图像进行二值化。这是通过使用内置的MATLAB函数im2bw来完成的，方法是为阈值提供一个0.3的值。这个值是从试验和错误中找到的。

**1.6细化**

第二类顺序细化算法是并行的。在并行细化算法中，单个像素删除的判定是基于前一次迭代的结果。与顺序算法一样，并行细化通常考虑当前像素周围的3\*3邻域。基于邻域中的像素应用一组删除规则。完全并行算法很难保持连通性，因此它们经常被分解成子迭代，其中只考虑15删除像素的子集。非迭代细化方法不是基于对单个像素的检查。一些流行的基于非像素的方法包括中轴变换、距离变换和由直线跟随确定中心线。

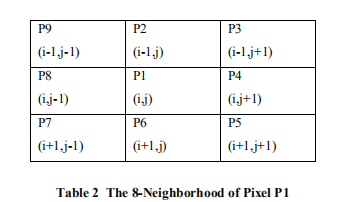
在下面的方法中，确定图像中黑色空间的中点，然后将其连接起来形成骨架。这是快速计算，但往往产生噪音骨骼。有人猜测，人类自然会以类似的方式进行减薄。

另一种确定中心线的方法是跟踪物体的轮廓。通过同时跟踪物体两侧的轮廓，可以计算出一条连续的中心线。图像的骨架是由这些相连的中心线形成的。利用距离变换计算像素强度。在图中，在最大像素强度下，圆心处的暗线会增加。注意，还有其他计算中轴变换的方法。

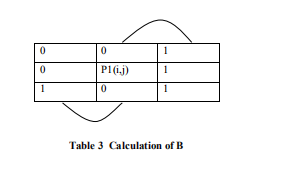


为了解释所使用的细化算法，我们将一步地解释它，以便更好地理解它。

该算法的伪代码使用以下变量。I=原始二进制图像，其中黑色像素为“0”，白色像素为“1”。 图像中的对象由连接的白色像素组成。J和K是在算法的每一次迭代中使用的临时图像。J是(n-1)第四次迭代输出，K是当前的，或者说是第n次，迭代输出p(I)是正在考虑的当前像素。



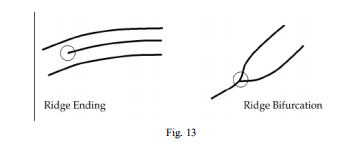
算法中使用的另外两个变量是A和B。像素P1的A被定义为：和(P2…)(P9)。像素P1的B是从P9返回到它自己的顺时针循环中0到1跃迁数的计数。例如，假设我们的像素如下图所示。b将e因为有两个0到1个顺时针顺序的跃迁。



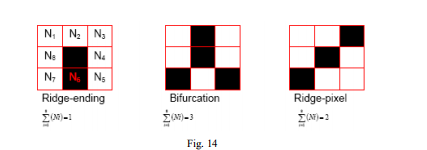
该算法分2次迭代运行.在每个子迭代中，应用不同的规则来决定是否删除像素。

**2）细节提取**

我们的指纹识别和验证的实现是基于细节点的拓扑结构匹配。我们只考虑两种细节：脊端和分叉。如下图所示：



从指纹图像的完美细化的山脊图中提取细节点是一项琐碎的任务。我们所需要做的就是计算脊像素的数目，细化图像上的每个脊像素都是被以下规则包围并依赖于这些规则，我们可以将细节点分配给这些像素：



然而，由于噪声、图像采集、皮肤变形等方面的限制，获得的指纹图像并不理想。因此，如果我们进行模拟，就会出现许多虚假的细节。Ly遵循上述的细节检测方法。为了解决这个问题，已经提出了各种启发式算法，并且我们已经实施了以下规则以去除大部分寄生细节，R在稀薄的图像中从噪音中排出：

如果几个细节在一个小区域内形成一个簇，那么除离簇中心最近的一个外，全部移除。

如果两个细节点足够近，彼此面对，但两者之间没有山脊，那就把它们都移除。

除了指纹图像中的噪声外，细化后的图像可能并不理想。如果是这样的话，细辛提取可能不会产生正确的结果。

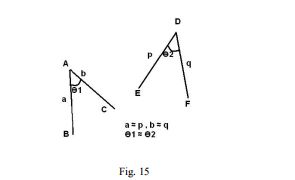
**3）细节匹配**

细节点匹配是在提取细节点之后的一步，在这里，我们将从两幅样本指纹图像中提取的细节点进行匹配，并检验它们是否来自同一个鳍。不管是不是。

然而，在两幅图像上使用蛮力和匹配细节之前，需要执行的一个关键步骤是图像的对齐。对齐是必要的，这样我们才能正确地匹配图像。我们还需要注意因手指的塑性变形而引起的细节定位上的差异。目前流行的细节提取算法要么包括细节的使用，要么包括细节的使用。或使用Hough变换。这些方法和大多数其他方法都很难实现，需要实现几个复杂的函数。

因此，我们决定在局部和全局细节特征计算技术的启发下，实现一种细节点匹配算法。

我们的算法将所有的细节点划分为三组细节点。对于每一组细节点，我们都存储了其中一个细节点与其他细节点之间的距离以及形成的角度。在这两段距离之间。

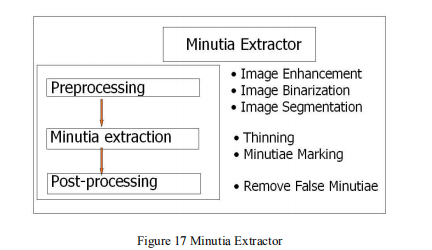


因此，当我们在图像中有n个细节点时，我们就有n(n-1)(n-1)/6细节点三重态。因此，我们需要为传入的图像和存储的映像计算和存储这些三胞胎。

然后，用蛮力法对包含细节三重奏的两个表进行匹配，并计算匹配三重奏的个数。

1. **算法级设计**

为了实现细节提取器，研究人员广泛采用了三阶段提取法.它们是预处理、细节提取和后处理阶段(图17)。



**1.向前看**

虽然我们的实现能够成功地识别两幅指纹图像是否属于同一个手指，但它绝不是完美的。如结果所示，有时会产生错误的结果。此外，该算法的计算时间仍然太长，无法实现无缝实时应用.

为了使我们的实现效率更高，在图像增强步骤和细节点匹配算法方面还有改进的余地。图像增强通常是最关键的一步，因为只要图像质量很高，细化和细节提取就无法取得任何有用的结果。我们已经在高质量的在线图像上测试了算法。在实际应用程序中，这种情况可能不是每次都是这样的。如果手指上的指纹太干或太湿，总是有可能的。此外，一个人有可能打伤了他/她的手指，而且由于割伤/烧伤等原因，可能发生了严重的改变。

要使我们的应用程序适合实时操作，一个非常重要的步骤是实现一个能够更好地匹配图像的细节匹配算法。效率高而且时间短得多。Jain等人提出的有效算法利用了每一个细节的岭信息。建立了脊的一维表示法，得到了每一个细节点。相应的脊也被存储。利用所有这些脊的信息，找出山脊之间的最佳匹配，并利用这些脊之间的角度和距离对图像进行旋转和平移。只有在这个旋转步骤之后才能进行细节匹配。

该算法可能的另一个改进是实现某些启发式，以消除虚假的细节。当查看每个像素的邻域计数时，可以包含更多选项。此外，Jain等人提出的某些规则也可以用来尽量减少虚假的细节。

1. **结论**

我的项目结合了许多方法来构建一个细节提取器和一个细节匹配器。多种方法的结合来源于对研究论文的广泛研究。还有一些新的变化，比如使用形态学操作进行分割，特殊考虑三重分支计数的细节标记，通过将分支分解成三个终结点来进行细节统一，在我的项目中，在统一的XY坐标系中，经过两步变换后进行了定位，这在我提到的其他文献中没有得到报道。

在此基础上，建立了一个贯穿指纹识别各个阶段的MATLAB编码程序。了解指纹识别的过程是很有帮助的。并演示了指纹识别的问题。

上述实施是为了了解指纹识别是如何被用作识别人类身份的一种生物特征的。它包括了从细节到外部的所有阶段。从指纹到细节点的匹配，产生匹配分数。在加工的中间阶段使用各种标准技术。

与其他形式的生物识别技术相比，验证率相对较低，这表明所使用的算法不太可靠，容易受到尺度和弹性效应的影响。c变形。发现了各种新的技术和算法，取得了较好的效果。

指纹识别的一个主要挑战是对指纹图像质量差的预处理，这也增加了指纹识别率低的问题。

**参考文献：**

[1] The Handbook of Fingerprint Recognition. Davide Maltoni, Dario Maio, Anil K Jain, Salil Prabhakar; Springer; 2003

[2] A. Jain and L. Hong, On-line Fingerprint Verification, Proc. 13th ICPR, Vienna, pp. 596-600, 1996.

[3] T. Jea and V. Govindaraju, "A Minutia-Based Partial Fingerprint Recognition System", Pattern Recognition 2005

[4] A. K. Jain, S. Prabhakar, and S. Pankanti, "Twin Test: On Discriminability of Fingerprints, " Proc. 3rd International Conference on Audio- and Video-Based Person Authentication, pp. 211-216, Sweden, June 6-8

[5] Raymond Thai, ‘Fingerprint Image Enhancement and Minutiae-Extraction,” Thesis submitted to School of Computer Science and Software Engineering, University of Western Australia

[6] Kovacs-Vajna, Z.M., "A fingerprint verification system based on triangular matching and dynamic time warping", IEEE Trans. Patt. Anal. Machine Intell., vol. 22, no. 11, pp. 1266 -1276, Nov.2000.

[7] Ratha, N.K.; Bolle, R.M.; Pandit, V.D.; Vaish, V., “Robust fingerprint authentication using local structural similarity”, *Applications of Computer Vision*, 2000, Fifth IEEE Workshop on. , 2000 Page(s): 29 –34

[8] AK Jain, A. Ross, and S. Prabhakar, Fingerprint Matching Using Minutiae and Texture Features , Proc. of International Conference on Image Processing, 200