Extract from:

D. Maltoni, D. Maio, A.K. Jain, S. Prabhakar
Handbook of Fingerprint Recognition
Springer, New York, 2003

4.3: Minutiae-based Methods
(extract)

Makekau (1991), Grycewicz (1995, 1996), Rodolfo, Rajbenbach, and Huignard (1995), Grycewicz and Javidi (1996), Petillot, Guibert, and de Bougrenet (1996), Soifer et al. (1996), Gamble, Frye, and Grieser (1992), Wilson, Watson, and Paek (1997), Kobayashi and Toyoda (1999), and Watson, Grother, and Casasent (2000). However, these optical systems usually suffer from rotation and distortion variations and the hardware/optical components are complex and expensive; therefore, optical fingerprint matching technology has not reached satisfactory maturity yet.

4.3 Minutiae-based Methods

Minutiae matching is certainly the most well-known and widely used method for fingerprint matching, thanks to its strict analogy with the way forensic experts compare fingerprints and its acceptance as a proof of identity in the courts of law in almost all countries.

Problem formulation

Let **T** and **I** be the representation of the template and input fingerprint, respectively. Unlike in correlation-based techniques, where the fingerprint representation coincides with the fingerprint image, here the representation is a feature vector (of variable length) whose elements are the fingerprint minutiae. Each minutia may be described by a number of attributes, including its location in the fingerprint image, orientation, type (e.g., ridge termination or ridge bifurcation), a weight based on the quality of the fingerprint image in the neighborhood of the minutia, and so on. Most common minutiae matching algorithms consider each minutia as a triplet $\mathbf{m} = \{x,y,\theta\}$ that indicates the x,y minutia location coordinates and the minutia angle θ:

$$\mathbf{T} = \{\mathbf{m}_1, \mathbf{m}_2, \ldots, \mathbf{m}_m\}, \quad \mathbf{m}_i = \{x_i, y_i, \theta_i\}, \quad i = 1..m$$
$$\mathbf{I} = \{\mathbf{m}'_1, \mathbf{m}'_2, \ldots, \mathbf{m}'_n\}, \quad \mathbf{m}'_j = \{x'_j, y'_j, \theta'_j\}, \quad j = 1..n\,,$$

where m and n denote the number of minutiae in **T** and **I**, respectively.

A minutia $\mathbf{m}'_j$ in **I** and a minutia $\mathbf{m}_i$ in **T** are considered "matching," if the *spatial distance* (*sd*) between them is smaller than a given tolerance r_0 and the *direction difference* (*dd*) between them is smaller than an angular tolerance θ_0:

$$sd(\mathbf{m}'_j, \mathbf{m}_i) = \sqrt{(x'_j - x_i)^2 + (y'_j - y_i)^2} \le r_0, \quad \text{and} \tag{5}$$

$$dd(\mathbf{m}'_j, \mathbf{m}_i) = \min\left(\left|\theta'_j - \theta_i\right|, 360° - \left|\theta'_j - \theta_i\right|\right) \le \theta_0\,. \tag{6}$$

Equation (6) takes the minimum of $\left|\theta'_j - \theta_i\right|$ and $360° - \left|\theta'_j - \theta_i\right|$ because of the circularity of angles (the difference between angles of 2° and 358° is only 4°). The *tolerance boxes* (or hyper-spheres) defined by r_0 and θ_0 are necessary to compensate for the unavoidable errors made by feature extraction algorithms and to account for the small plastic distortions that cause the minutiae positions to change.

Aligning the two fingerprints is a mandatory step in order to maximize the number of matching minutiae. Correctly aligning two fingerprints certainly requires *displacement* (in x and y) and *rotation* (θ) to be recovered, and likely involves other geometrical transformations:

- *scale* has to be considered when the resolution of the two fingerprints may vary (e.g., the two fingerprint images have been taken by scanners operating at different resolutions);
- other *distortion-tolerant* geometrical transformations could be useful to match minutiae in case one or both of the fingerprints is affected by severe distortions.

In any case, tolerating a higher number of transformations results in additional degrees of freedom to the minutiae matcher: when a matcher is designed, this issue needs to be carefully evaluated, as each degree of freedom results in a huge number of new possible alignments which significantly increases the chance of incorrectly matching two fingerprints from different fingers.

Let *map*(.) be the function that maps a minutia $\mathbf{m}'_j$ (from **I**) into $\mathbf{m}''_j$ according to a given geometrical transformation; for example, by considering a displacement of [Δx, Δy] and a counterclockwise rotation θ around the origin[1]:

$$map_{\Delta x,\Delta y,\theta}\left(\mathbf{m}'_j = \{x'_j, y'_j, \theta'_j\}\right) = \mathbf{m}''_j = \{x''_j, y''_j, \theta'_j + \theta\}, \quad \text{where}$$

$$\begin{bmatrix} x''_j \\ y''_j \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x'_j \\ y'_j \end{bmatrix} + \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix}.$$

Let *mm*(.) be an indicator function that returns 1 in the case where the minutiae $\mathbf{m}''_j$ and $\mathbf{m}_i$ match according to Equations (5) and (6):

$$mm\left(\mathbf{m}''_j, \mathbf{m}_i\right) = \begin{cases} 1 & sd\left(\mathbf{m}''_j, \mathbf{m}_i\right) \le r_0 \quad \text{and} \quad dd\left(\mathbf{m}''_j, \mathbf{m}_i\right) \le \theta_0 \\ 0 & \text{otherwise.} \end{cases}$$

Then, the matching problem can be formulated as

$$\underset{\Delta x,\Delta y,\theta,P}{\text{maximize}} \sum_{i=1}^{m} mm\left(map_{\Delta x,\Delta y,\theta}\left(\mathbf{m}'_{P(i)}\right), \mathbf{m}_i\right), \tag{7}$$

where $P(i)$ is an unknown function that determines the *pairing* between **I** and **T** minutiae; in particular, each minutia has either exactly one mate in the other fingerprint or has no mate at all:

1 The origin is usually selected as the minutiae centroid (i.e., the average point); before the matching step, minutiae coordinates are adjusted by subtracting the centroid coordinates.

1. $P(i) = j$ indicates that the mate of the $\mathbf{m}_i$ in **T** is the minutia $\mathbf{m}'_j$ in **I**;
2. $P(i)$ = null indicates that minutia $\mathbf{m}_i$ in **T** has no mate in **I**;
3. a minutia $\mathbf{m}'_j$ in **I**, such that $\forall\ i = 1..m,\ P(i) \neq j$ has no mate in **T**;
4. $\forall\ i = 1..m,\ k = 1..m,\ i \neq k \Rightarrow P(i) \neq P(k)$ or $P(i) = P(k)$ = null (this requires that each minutia in **I** is associated with a maximum of one minutia in **T**).

Note that, in general, $P(i) = j$ does not necessarily mean that minutiae $\mathbf{m}'_j$ and $\mathbf{m}_i$ match in the sense of Equations (5) and (6) but only that they are the most likely pair under the current transformation.

Expression (7) requires that the number of minutiae mates be maximized, independently of how strict these mates are; in other words, if two minutiae comply with Equations (5) and (6), then their contribution to expression (7) is made independently of their spatial distance and of their direction difference. Alternatives to expression (7) may be introduced where the residual (i.e., the spatial distance and the direction difference between minutiae) for the optimal alignment is also taken into account.

Solving the minutiae matching problem (expression (7)) is trivial when the correct alignment $(\Delta x, \Delta y, \theta)$ is known; in fact, the pairing (i.e., the function P) can be determined by setting for each $i = 1..m$:

- $P(i) = j$ if $\mathbf{m}''_j = map_{\Delta x,\Delta y,\theta}(\mathbf{m}'_j)$ is closest to $\mathbf{m}_i$ among the minutiae $\{\ \mathbf{m}''_k = map_{\Delta x,\Delta y,\theta}(\mathbf{m}'_k)\ |\ k = 1..n,\ \ mm(\mathbf{m}''_k, \mathbf{m}_i) = 1\ \}$;
- $P(i)$ = null if $\forall\ k = 1..n,\ mm(map_{\Delta x,\Delta y,\theta}(\mathbf{m}'_k), \mathbf{m}_i) = 0$.

To comply with constraint 4 above, each minutia $\mathbf{m}''_j$ already mated has to be marked, to avoid mating it twice or more. Figure 4.4 shows an example of minutiae pairing given a fingerprint alignment.

To achieve the optimum pairing (according to Equation (7)), a slightly more complicated scheme should be adopted: in fact, in the case when a minutia of **I** falls within the tolerance hyper-sphere of more than one minutia of **T**, the optimum assignment is that which maximizes the number of mates (refer to Figure 4.5 for a simple example).

The maximization in (7) can be easily solved if the function P (minutiae correspondence) is known; in this case, the unknown alignment $(\Delta x, \Delta y, \theta)$ can be determined in the least square sense (Umeyana (1991) and Chang et al. (1997)). Unfortunately, in practice, neither the alignment parameters nor the correspondence function P are known and, therefore, solving the matching problem is very hard. A brute force approach, that is, evaluating all the possible solutions (correspondences and alignments) is prohibitive as the number of possible solutions is exponential in the number of minutiae (the function P is more than a permutation due to the possible null values). A few brute force approaches have also been proposed in the literature;

for example, Huvanandana, Kim, and Hwang (2000) proposed coarsely quantizing the minutiae locations and performing an exhaustive search to find the optimum alignment.

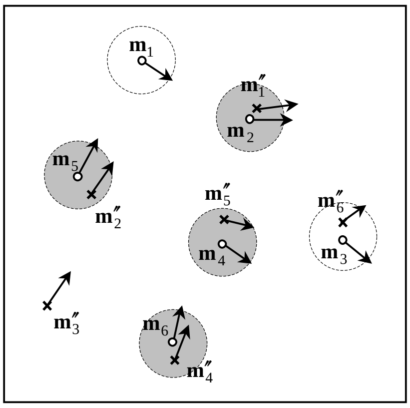

Figure 4.4. Minutiae of I mapped into T coordinates for a given alignment. Minutiae of T are denoted by **o**s, whereas I minutiae are denoted by ×s. Note that I minutiae are referred to as $\mathbf{m}''$, because what is shown in the figure is their mapping into T coordinates. Pairing is performed according to the minimum distance. The dashed circles indicate the maximum spatial distance. The gray circles denote successfully mated minutiae; minutia $\mathbf{m}_1$ of T and minutia $\mathbf{m}''_3$ of I have no mates, minutiae $\mathbf{m}_3$ and $\mathbf{m}''_6$ cannot be mated due to their large direction difference.

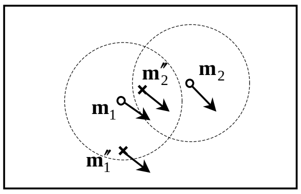

Figure 4.5. In this example, if $\mathbf{m}_1$ were mated with $\mathbf{m}''_2$ (the closest minutia), $\mathbf{m}_2$ would remain unmated; however, pairing $\mathbf{m}_1$ with $\mathbf{m}''_1$, allows $\mathbf{m}_2$ to be mated with $\mathbf{m}''_2$, thus maximizing Equation (7).

In the pattern recognition literature the minutiae matching problem has been generally addressed as a *point pattern matching* problem. Even though a small difference exists due to the presence of a direction associated with each minutia point, the two problems may be approached analogously. Because of its central role in many pattern recognition and computer vision tasks (e.g., object matching, remote sensing, camera calibration, motion estimation),