

Minutia Cylinder-Code: A New Representation and Matching Technique for Fingerprint Recognition

Raffaele Cappelli, *Member, IEEE*, Matteo Ferrara, and Davide Maltoni, *Member, IEEE*

Abstract—In this paper, we introduce the Minutia Cylinder-Code (MCC): a novel representation based on 3D data structures (called cylinders), built from minutiae distances and angles. The cylinders can be created starting from a subset of the mandatory features (minutiae position and direction) defined by standards like ISO/IEC 19794-2 (2005). Thanks to the cylinder invariance, fixed-length, and bit-oriented coding, some simple but very effective metrics can be defined to compute local similarities and to consolidate them into a global score. Extensive experiments over FVC2006 databases prove the superiority of MCC with respect to three well-known techniques and demonstrate the feasibility of obtaining a very effective (and interoperable) fingerprint recognition implementation for light architectures.

Index Terms—Bit-oriented, cylinder-code, fingerprint, ISO/IEC 19794-2, local minutiae matching.

1 INTRODUCTION

FINGERPRINT recognition is an intriguing pattern recognition problem that has been studied for more than 40 years. Although very effective solutions are currently available, fingerprint recognition cannot be considered a fully solved problem, and the design of accurate, interoperable, and computationally light algorithms is still an open issue [1].

Most fingerprint matching algorithms are based on minutiae (i.e., ridge ending and bifurcations). For a long time, minutiae matching was treated as a 2D point pattern matching problem aimed at determining the global (rigid) alignment leading to an optimal spatial (and directional) minutiae pairing. This formulation of the problem can be solved by searching the space of possible transformations: Hough transform is a common solution [2], [3]. Unfortunately, most of the global minutiae matching algorithms are computationally demanding and lack robustness with respect to nonlinear fingerprint distortion.

In the last decade, these weaknesses were addressed by introducing *local minutiae matching* techniques. Local minutiae structures are characterized by attributes that are invariant with respect to global transformations (e.g., translation, rotation, etc.) and therefore are suitable for matching without any a priori global alignment. Matching fingerprints based only on local minutiae arrangements relaxes global spatial relationships, which are highly distinctive, and therefore reduces the amount of information available for discriminating fingerprints. However, the benefits of both

local and global matching can be preserved by implementing hybrid strategies that perform a local structure matching followed by a consolidation stage. The local structure matching allows us to quickly and robustly determine pairs of minutiae that match locally (i.e., whose neighboring features are compatible); the consolidation is aimed at verifying if and to what extent local matches hold at global level. It is worth noting that the consolidation step is not mandatory and a score can be directly derived from the local structure matching. The local matching itself can also lead to an early rejection in the case of very different fingerprints.

Local minutiae matching algorithms evolved through three generations of methods: 1) the earlier approaches, whose local structures were typically formed by counting the number of minutiae falling inside some regions and no global consolidation was performed [4], [5], 2) the approaches by Jiang and Yau [6] and Ratha et al. [7], who first effectively encoded the relationships between a minutia and its neighboring minutiae in terms of invariant distances and angles, and proposed global consolidation, and 3) the numerous variants and evolutions of Jiang and Yau [6] and Ratha et al. [7] methods, which typically extend the feature set by taking into account: local orientation field, local frequency, ridge shape, etc., see [8], [9], [10], [11], [12], [13], [14], [15], [16], [17], [18], [19], [20], [21], [22], [23], [24], [25], [26], [27], [28], [29], [30], [31], [32], [33]. The reader may refer to [1] for an exhaustive review and classification of the literature on local minutiae matching.

Local minutiae structures can be classified into *nearest neighbor-based* and *fixed radius-based*. In the former family (well represented by Jiang and Yau's algorithm [6]), the neighbors of the central minutia are defined as its K spatially closest minutiae. This leads to fixed-length descriptors that can usually be matched very efficiently. In the latter (well represented by Ratha et al.'s algorithm [7]), the neighbors are defined as all the minutiae that are closer than a given radius R to the central minutia. The descriptor length is

• The authors are with the Biometric System Laboratory—DEIS—University of Bologna, via Sacchi 3, 47521 Cesena (FC), Italy.

E-mail: {raffaele.cappelli, matteo.ferrara, davide.maltoni}@unibo.it.

Manuscript received 16 Mar. 2009; accepted 1 Nov. 2009; published online 25 Feb. 2010.

Recommended for acceptance by N.K. Ratha.

For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number TPAMI-2009-03-0169.

Digital Object Identifier no. 10.1109/TPAMI.2010.52.

variable and depends on the local minutiae density; this can lead to a more complex local matching, but, in principle, is more tolerant against missing and spurious minutiae. Two drawbacks of [7] are: 1) the absolute encoding of radial angles (whose corresponding relative encoding is denoted as d_R in Fig. 7) that requires a sophisticated local matching and 2) the missing directional difference between the central minutia and the neighboring ones (denoted as d_θ in Fig. 7). Furthermore, the approach in [7], like most fixed-radius ones, can lead to border errors: In particular, minutiae close to the local-region border in one of the two fingerprints can be mismatched because local distortion or location inaccuracy may cause the same minutiae to move out of the local region in the other fingerprint. The technique proposed by Feng in [12] does not suffer from the above drawbacks and can be considered a state-of-the-art fixed-radius local matching algorithm. In particular, the border problem is dealt with by considering minutiae not close to the border as *matchable* and minutiae near the border as *should-be-matchable*.

This paper introduces a novel minutiae-only local representation aimed at combining the advantages of both neighbor-based and fixed-radius structures, without suffering from their respective drawbacks.

The rest of this paper is organized as follows: Section 2 introduces the main motivations of this work and summarizes the advantages of the new technique. Section 3 defines the minutiae local structures and discusses how to measure the similarity between them. Section 4 proposes four simple approaches to consolidate local similarities into a global score. In Section 5, a large number of experiments are reported to compare the new approach with three “minutiae-only” implementations of the well-known approaches described in [6], [7], [12]. Finally, Section 6 draws some concluding remarks.

2 MOTIVATIONS AND CONTRIBUTIONS

The main motivations that induced us to design a new local minutiae matching technique are:

- *Need for accurate and interoperable minutiae-only algorithms.* Most of the fingerprint matching algorithms recently proposed exploit several extra features besides minutiae; in [34], some statistics about the features used by FVC2004 participants are reported. Researchers have shown that combining features (at least partially independent) is a very effective way to improve accuracy. On the other hand, unlike minutiae features, there is still no convergence on standards that precisely define and encode these extra features (one of the first attempts is CDEFFS (2008) [35], but it is still at an early stage). The worldwide large-scale deployment of fingerprint systems demands a new generation of accurate and highly interoperable algorithms and, for this reason, we believe that minutiae-only matching algorithms compliant to ISO/IEC 19794-2 (2005) [36] (or the very similar ANSI/INCITS 378 (2004) [37]) will play a central role in coming years. Furthermore, minutiae-only templates also allow us to compress into a few hundred bytes the salient fingerprint information, thus enabling their storage on inexpensive smart cards.

- *Portability on light architectures.* One effective way to secure biometric applications against external attacks is to confine the computation inside a closed system, that is, a secure hardware platform such as a smart card or a system-on-a-chip. Unfortunately, the computational power of these low-cost secure platforms is a hundred or thousand times lower than that of a modern PC [1] and resource demanding algorithms cannot be executed on board. Algorithms designers then concentrated on the development of simplified optimized versions, often based on local minutiae matching techniques and precomputed information. However, recent MINEX II results [38] have shown that the best existing match-on-card algorithms cannot compete with the corresponding PC implementations and further research efforts are necessary. Analogous conclusions were drawn in [34] concerning the performance drop of the light category with respect to the open category in FVC2004.
- *Suitability for template protection techniques.* Template protection is currently receiving much attention because of the great benefits it can provide (e.g., nonreversibility, diversity, and revocability) [39], [1]. Unfortunately, designing effective template protection techniques (e.g., fuzzy vault [40], [41], [42], [43], [44]) without incurring a relevant accuracy drop is very challenging and seems to require alignment free, fixed-length, and noise-tolerant feature coding. As of now, no fully satisfactory solution has been proposed.

The local minutiae representation introduced in this paper is based on 3D data structures (called cylinders), built from invariant distances and angles in a neighborhood of each minutia. Cylinders can be created starting from a subset of mandatory features in standards like ISO/IEC 19794-2 (2005). In particular, we use only the position and direction of the minutiae, but not the minutiae type and the minutiae quality: In fact, minutiae type is not a robust feature and the definition of minutiae quality is not semantically clear in the standards (and could lead to interoperability problems). Thanks to the cylinder invariance, fixed-length, and bit-oriented coding, some simple but effective metrics can be defined to compute cylinder similarity. Four global-scoring techniques are then proposed to combine local similarities into a unique global score denoting the overall similarity between two fingerprints. The main advantages of the new method, called *Minutia Cylinder-Code* (MCC), are:

- MCC is a fixed-radius approach and therefore it tolerates missing and spurious minutiae better than nearest neighbor-based approaches.
- Unlike traditional fixed-radius techniques, MCC relies on a fixed-length invariant coding for each minutia and this makes the computation of local structure similarities very simple.
- Border problems are gracefully managed without extra burden in the coding and matching stages.
- Local distortion and small feature extraction errors are tolerated thanks to the adoption of smoothed functions (i.e., error tolerant) in the coding stage.
- MCC effectively deals with noisy fingerprint regions where minutiae extraction algorithms tend to place

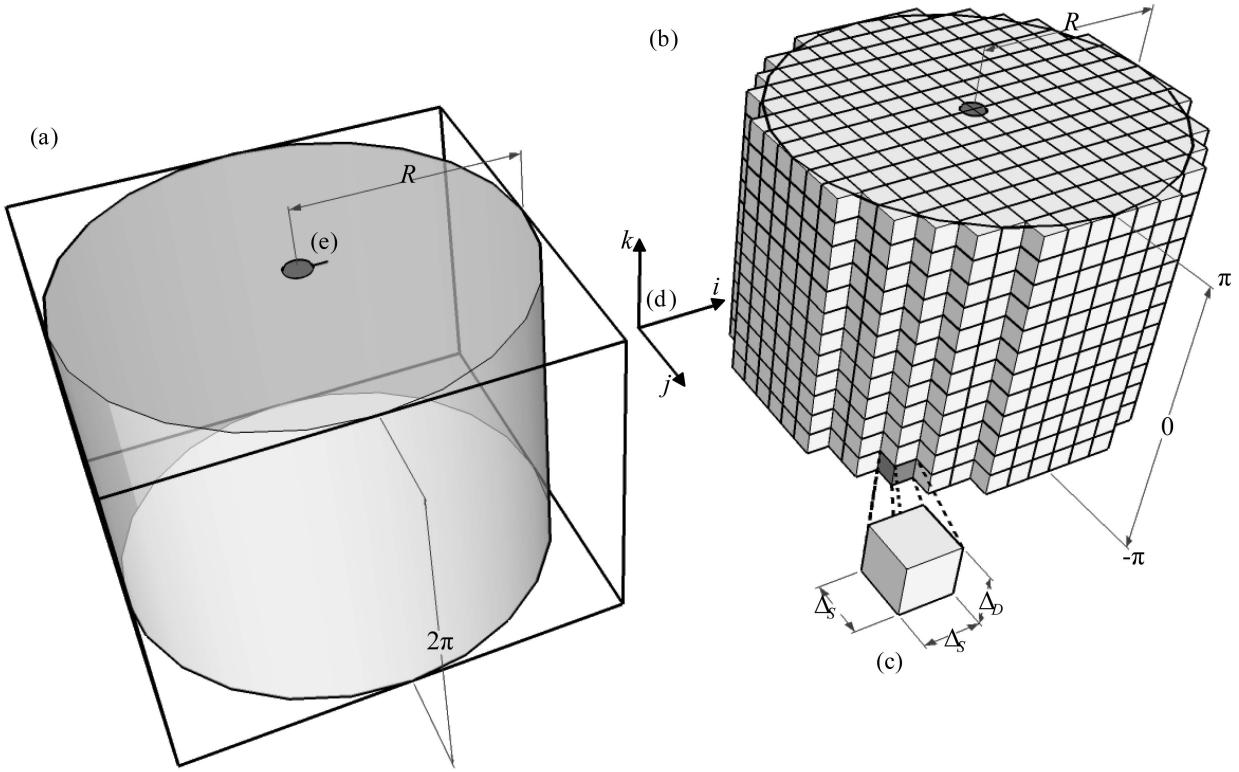


Fig. 1. A graphical representation of the local structure associated to a given minutia: (a) the cylinder with the enclosing cuboid, (b) the discretization of the cuboid into cells (c) of size $\Delta_S \times \Delta_S \times \Delta_D$: Only cells whose center is within the cylinder are shown. Note that the cuboid is rotated so that (d) axis i is aligned to (e) the direction of the corresponding minutia.

numerous spurious minutiae (close to each other); this is made possible by the saturation effect produced by a limiting function.

- The bit-oriented coding (one of the possible implementations of MCC) makes cylinder matching extremely simple and fast, reducing it to a sequence of bit-wise operations (e.g., AND, XOR) that can be efficiently implemented even on very simple CPUs.

3 THE LOCAL STRUCTURES

MCC representation associates a local structure to each minutia. This structure encodes spatial and directional relationships between the minutia and its (fixed-radius) neighborhood and can be conveniently represented as a cylinder whose base and height are related to the spatial and directional information, respectively (Fig. 1).

Let $T = \{m_1, m_2, \dots, m_n\}$ be an ISO/IEC 19794-2 minutiae template [36]: Each minutia m is a triplet $m = \{x_m, y_m, \theta_m\}$, where x_m and y_m are the minutia location, θ_m is the minutia direction (in the range $[0, 2\pi]$). In the following, Section 3.1 describes how the local structure of a given minutia m is built, Section 3.2 discusses the creation of a whole cylinder-set from T , and Section 3.3 introduces a similarity measure between cylinders; finally, Section 3.4 focuses on a bit-oriented efficient implementation.

3.1 The Cylinder of a Given Minutia

The local structure associated to a given minutia $m = \{x_m, y_m, \theta_m\}$ is represented by a cylinder with radius R and height 2π whose base is centered on the minutia location (x_m, y_m) , see Fig. 1a.

The cylinder is enclosed inside a cuboid whose base is aligned according to the minutia direction θ_m ; the cuboid is discretized into $N_C = N_S \times N_S \times N_D$ cells. Each cell is a small cuboid with $\Delta_S \times \Delta_S$ base and Δ_D height, where $\Delta_S = \frac{2R}{N_S}$ and $\Delta_D = \frac{2\pi}{N_D}$ (Fig. 1b).

Each cell can be uniquely identified by three indices (i, j, k) that denote its position in the cuboid enclosing the cylinder, with $i, j \in I_S = \{n \in \mathbb{N}, 1 \leq n \leq N_S\}$ and $k \in I_D = \{n \in \mathbb{N}, 1 \leq n \leq N_D\}$.

Let

$$d\varphi_k = -\pi + \left(k - \frac{1}{2}\right) \cdot \Delta_D \quad (1)$$

be the angle associated to all cells at height k in the cylinder and let

$$p_{i,j}^m = \begin{bmatrix} x_m \\ y_m \end{bmatrix} + \Delta_S \cdot \begin{bmatrix} \cos(\theta_m) & \sin(\theta_m) \\ -\sin(\theta_m) & \cos(\theta_m) \end{bmatrix} \cdot \begin{bmatrix} i - \frac{N_S+1}{2} \\ j - \frac{N_S+1}{2} \end{bmatrix} \quad (2)$$

be the two-dimensional point corresponding to the center of the cells with indices i, j (projected onto the cylinder's base), expressed in the spatial coordinates of the minutiae template; since these points are projected onto the base, index k is not needed.

For each cell (i, j, k) , a numerical value $C_m(i, j, k)$ is calculated by accumulating contributions from each minutia m_t belonging to the neighborhood $N_{p_{i,j}^m}$ of $p_{i,j}^m$:

$$N_{p_{i,j}^m} = \{m_t \in T; m_t \neq m, d_S(m_t, p_{i,j}^m) \leq 3\sigma\}, \quad (3)$$

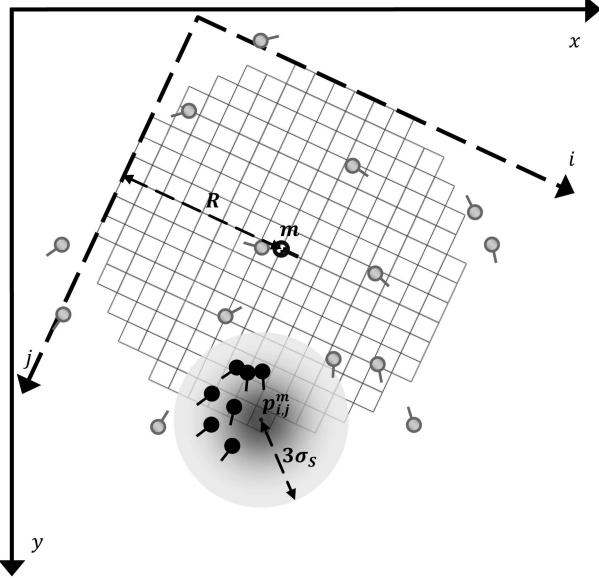


Fig. 2. Section of a cylinder associated to a minutia m . All of the minutiae involved in the construction of the cylinder are shown. Note that they do not necessarily lie inside the cylinder base since an offset of $3\sigma_S$ is allowed. $G_S(t)$ values in the neighborhood of a given cell (with center $p_{i,j}^m$) are highlighted (darker areas represent higher values). The black minutiae are those within neighborhood $N_{p_{i,j}^m}$.

where $3\sigma_S$ is the radius of the neighborhood (see Fig. 2) and $d_S(m, p)$ is the euclidean distance between minutia m and point p .

Function $C_m : I_S \times I_S \times I_D \rightarrow V$ is defined as follows:

$$C_m(i, j, k) = \begin{cases} \Psi \left(\sum_{m_t \in N_{p_{i,j}^m}} (C_m^S(m_t, p_{i,j}^m) \cdot C_m^D(m_t, d\varphi_k)) \right) & \text{if } \xi_m(p_{i,j}^m) = \text{valid} \\ \text{invalid} & \text{otherwise,} \end{cases} \quad (4)$$

where:

- $V = [0, 1] \cup \{\text{invalid}\}$ is the function codomain.
- The two terms $C_m^S(m_t, p_{i,j}^m)$ and $C_m^D(m_t, d\varphi_k)$ are the spatial and directional contribution of minutia m_t , respectively (they will be described in the following paragraphs):
-

$$\xi_m(p_{i,j}^m) = \begin{cases} \text{valid} & \text{if } d_S(m, p_{i,j}^m) \leq R \text{ and} \\ & p_{i,j}^m \in \text{ConvHull}(T, \Omega) \\ \text{invalid} & \text{otherwise,} \end{cases}$$

where $\text{ConvHull}(T, \Omega)$ is the convex hull [45] of the minutiae in T , enlarged by adding an offset of Ω pixels (see Fig. 5a). Intuitively, a cell is considered as *valid* if and only if its center $p_{i,j}^m$ is contained in the intersection of the cylinder's base with the convex hull determined by all the minutiae in T (see Fig. 5b): This condition is important to avoid considering portions of the cylinder that probably lie outside the fingerprint area and hence cannot contain relevant information.

- $\Psi(v) = Z(v, \mu_\Psi, \tau_\Psi)$ is a sigmoid function, controlled by two parameters (μ_Ψ and τ_Ψ), that limits the contribution of dense minutiae clusters (typical of

noisy regions), and ensures that the final value is in the range $[0, 1]$; the sigmoid function is defined as:

$$Z(v, \mu, \tau) = \frac{1}{1 + e^{-\tau(v - \mu)}}. \quad (5)$$

Basically, the value $C_m(i, j, k)$ of a valid cell represents the likelihood of finding minutiae near $p_{i,j}^m$ with a directional difference, with respect to m , close to $d\varphi_k$. This likelihood is obtained by summing the contributions of all the minutiae in neighborhood $N_{p_{i,j}^m}$. The contribution of each minutia m_t is defined as the product of C_m^S and C_m^D .

$C_m^S(m_t, p_{i,j}^m)$ is the spatial contribution that minutia m_t gives to cell (i, j, k) ; it is defined as a function of the euclidean distance between m_t and $p_{i,j}^m$:

$$C_m^S(m_t, p_{i,j}^m) = G_S(d_S(m_t, p_{i,j}^m)), \quad (6)$$

where

$$G_S(t) = \frac{1}{\sigma_S \sqrt{2\pi}} e^{-\frac{t^2}{2\sigma_S^2}} \quad (7)$$

is the Gaussian function with zero mean and σ_S standard deviation.

Fig. 2 graphically shows the values of $G_S(t)$ in the neighborhood of a given cell (darker areas represent higher values). It is worth noting that minutiae involved in the computation of $C_m(i, j, k)$ do not necessarily lie inside the base of the cylinder centered in m with radius R ; in fact, minutiae lying in the circular ring of radius $[R, R + 3\sigma_S]$ still contribute to $C_m(i, j, k)$ and this allows us to avoid the tedious border effect.

$C_m^D(m_t, d\varphi_k)$ is the directional contribution of m_t ; it is defined as a function of: 1) $d\varphi_k$ and 2) the directional difference between θ_m and θ_{m_t} . Intuitively, the contribution is high when 1) and 2) are close to each other.

$$C_m^D(m_t, d\varphi_k) = G_D(d\phi(d\varphi_k, d\theta(m, m_t))), \quad (8)$$

where $d\phi(\theta_1, \theta_2)$ is the difference between two angles θ_1, θ_2 :

$$d\phi(\theta_1, \theta_2) = \begin{cases} \theta_1 - \theta_2 & \text{if } -\pi \leq \theta_1 - \theta_2 < \pi \\ 2\pi + \theta_1 - \theta_2 & \text{if } \theta_1 - \theta_2 < -\pi \\ -2\pi + \theta_1 - \theta_2 & \text{if } \theta_1 - \theta_2 \geq \pi, \end{cases} \quad (9)$$

and $d\theta(m_1, m_2)$ is the directional difference between two minutiae:

$$d\theta(m_1, m_2) = d\phi(\theta_{m_1}, \theta_{m_2}). \quad (10)$$

$G_D(\alpha)$ is the area under a Gaussian (with zero mean and standard deviation σ_D) in the interval $[\alpha - \frac{\Delta_D}{2}, \alpha + \frac{\Delta_D}{2}]$:

$$G_D(\alpha) = \frac{1}{\sigma_D \sqrt{2\pi}} \int_{\alpha - \frac{\Delta_D}{2}}^{\alpha + \frac{\Delta_D}{2}} e^{-\frac{t^2}{2\sigma_D^2}} dt. \quad (11)$$

Fig. 3 shows the local structure associated to a given minutia m in a simplified case where there is only one minutia that contributes to cell values $C_m(i, j, k)$. Fig. 4 shows the cylinder associated to a minutia with five minutiae in its neighborhood.

3.2 Creation of a Cylinder-Set

The cylinder-set obtained from an ISO/IEC 19794-2 minutiae template T is defined as:

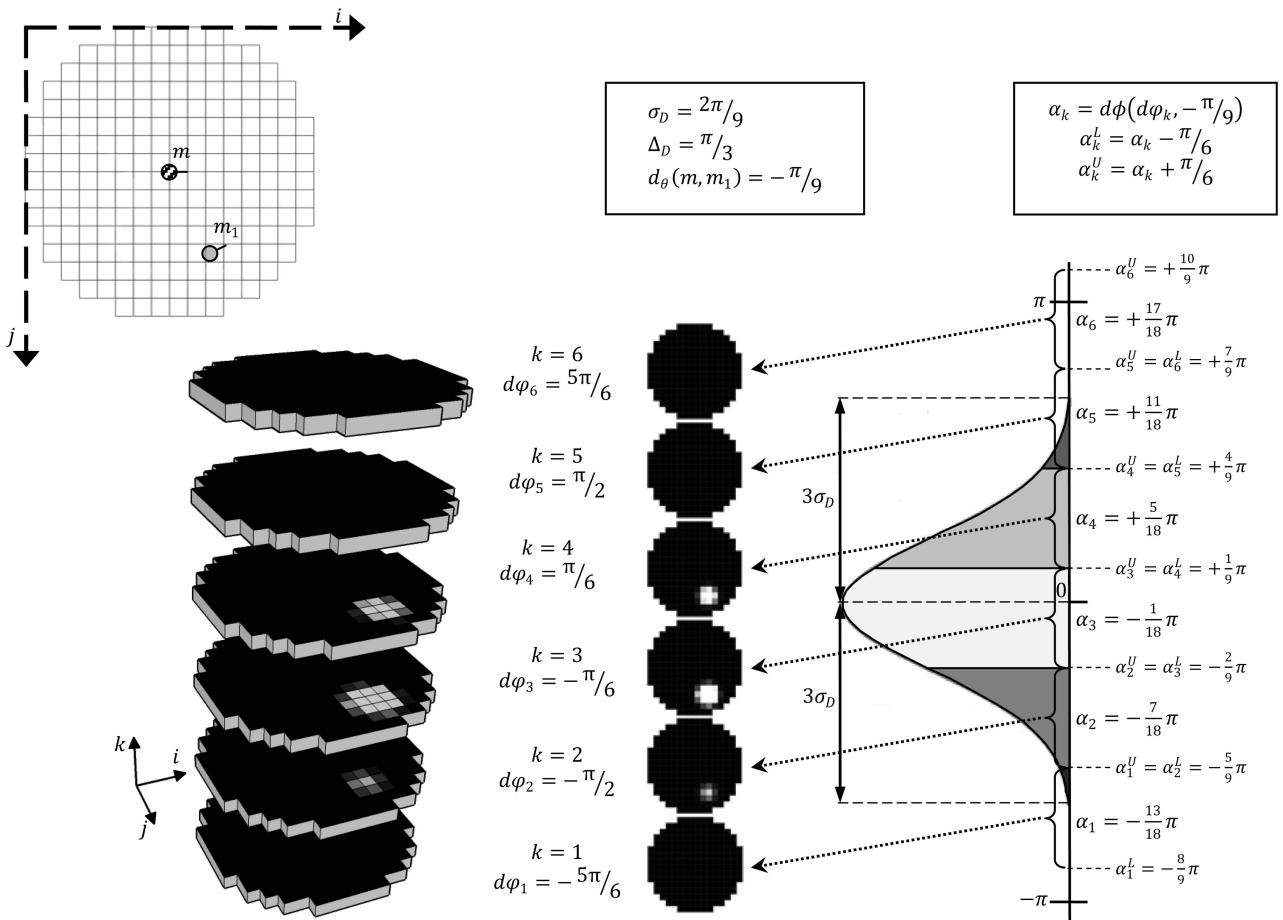


Fig. 3. A simplified case where only one minutia (m_1) contributes to the cylinder associated to minutia m . Different $C_m(i, j, k)$ values are represented by different gray levels (the lighter, the greater). The N_D areas (six in this example) under the Gaussian curve are graphically highlighted and the relevant values in (8) and (11) are numerically exemplified for each k : In particular, $\alpha_k = d\phi(d\varphi_k, d_\theta(m, m_1))$ is the input value of function G_D in (8), while α_k^L and α_k^U are the lower and upper limits of the integral in (11), respectively. In practice, minutia m_1 contributes to more cylinder sections with different weights, according to its directional difference with m . Note that nonzero cell values are not perfectly symmetric with respect to the cell containing m_1 : This is because m_1 does not exactly lie in the center of the cell.

$$CS = \{C_m | C_m \text{ is not invalid}, m \in T\}, \quad (12)$$

where C_m is the cylinder associated to minutia m , containing values $C_m(i, j, k)$. A cylinder C_m is considered *invalid* in the following cases:

- there are less than min_{VC} valid cells in the cylinder;
- there are less than min_M minutiae that contribute to the cylinder (i.e., there are less than min_M minutiae m_t such that $d_S(m_t, m) \leq R + 3\sigma_S$, with $m_t \neq m$).

Fig. 5 shows a minutia template and three valid cylinders from the corresponding cylinder-set.

3.3 The Similarity between Two Cylinders

Each cylinder is a local data structure:

- invariant for translation and rotation since 1) it only encodes distances and directional differences between minutiae (see (6) and (8)), and 2) its base is rotated according to the corresponding minutia direction, see (2);
- robust against skin distortion (which is small at a local level) and against small feature extraction errors, thanks to the smoothed nature of the functions

defining the contribution of each minutia (see (7) and (11)), and to the limiting function Ψ in (4);

- with a fixed-length given by the number of cells N_C .

For the above reasons, the similarity between two cylinders can be simply defined using a vector correlation measure, as described in the following paragraphs.

Given a cylinder C_m , let $lin : I_S \times I_S \times I_D \rightarrow \mathbb{N}$ be a function that linearizes the cylinder cell indices:

$$lin(i, j, k) = (k - 1) \cdot (N_S)^2 + (j - 1) \cdot N_S + i \quad (13)$$

and let $\mathbf{c}_m \in V^{N_C}$ be the vector derived from C_m (V is the codomain of (4)), according to (13):

$$\mathbf{c}_m[lin(i, j, k)] = C_m(i, j, k). \quad (14)$$

Given two minutiae a and b , let \mathbf{c}_a and \mathbf{c}_b be the vectors derived from cylinders C_a and C_b : Two corresponding elements $\mathbf{c}_a[t]$ and $\mathbf{c}_b[t]$ are considered as *matchable* if and only if $\mathbf{c}_a[t] \neq invalid \wedge \mathbf{c}_b[t] \neq invalid$. Let $\mathbf{c}_{a|b}, \mathbf{c}_{b|a} \in [0, 1]^{N_C}$ be the two vectors derived from \mathbf{c}_a and \mathbf{c}_b considering *matchable* elements only:

$$\mathbf{c}_{a|b}[t] = \begin{cases} \mathbf{c}_a[t] & \text{if } \mathbf{c}_a[t] \text{ and } \mathbf{c}_b[t] \text{ are matchable} \\ 0 & \text{otherwise,} \end{cases} \quad (15)$$

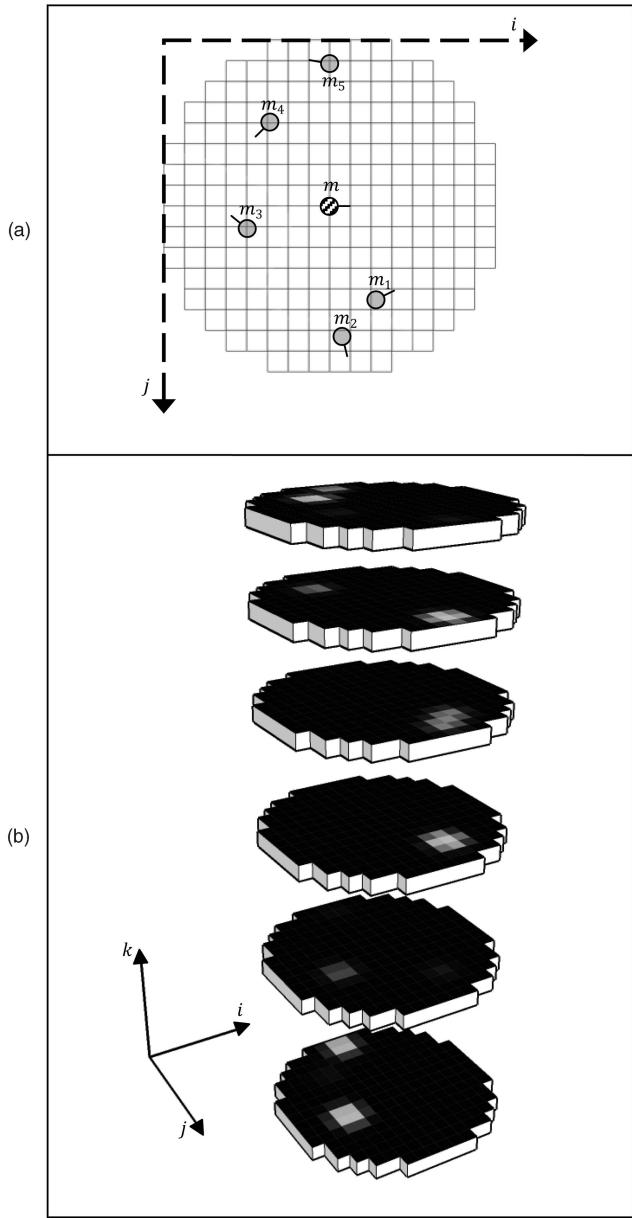


Fig. 4. A graphical representation of a cylinder: (a) the minutiae involved and (b) the cell values: lighter areas represent higher values.

$$\mathbf{c}_{b|a}[t] = \begin{cases} \mathbf{c}_b[t] & \text{if } \mathbf{c}_b[t] \text{ and } \mathbf{c}_a[t] \text{ are matchable} \\ 0 & \text{otherwise.} \end{cases} \quad (16)$$

In practice, *matchable* elements correspond to the intersection of the valid cells of the two cylinders.

The similarity between the two cylinders is defined as:

$$\gamma(a, b) = \begin{cases} 1 - \frac{\|\mathbf{c}_{a|b} - \mathbf{c}_{b|a}\|}{\|\mathbf{c}_{a|b}\| + \|\mathbf{c}_{b|a}\|} & \text{if } C_a \text{ and } C_b \text{ are matchable} \\ 0 & \text{otherwise,} \end{cases} \quad (17)$$

where two cylinders are *matchable* if the following conditions are met:

1. the directional difference between the two minutiae is not greater than δ_θ ($d_\theta(a, b) \leq \delta_\theta$);
2. at least min_{ME} corresponding elements in the two vectors \mathbf{c}_a and \mathbf{c}_b are *matchable*;

$$3. \|\mathbf{c}_{a|b}\| + \|\mathbf{c}_{b|a}\| \neq 0.$$

The first condition helps to reduce the number of *matchable* cylinders by assuming a maximum possible rotation between the two fingerprints; the second condition avoids comparing cylinders with a too small valid intersection; the third condition excludes the case where a sufficiently large valid intersection of two valid cylinders does not contain any information.

Note that $\gamma(a, b)$ is always in the range $[0, 1]$: Zero means no similarity and one denotes maximum similarity. In the following, depending on the context, we will refer to $\gamma(a, b)$ as cylinder similarity, local structure similarity, or minutiae similarity; because of the 1:1 relationship between minutiae and cylinders, this notation flexibility does not lead to ambiguities.

3.4 Bit-Based Implementation

The characteristics of the local structures and similarity measure introduced in the previous sections make MCC well suited for a bit-based implementation. To this purpose, $\Psi(v)$ in (4) may be changed from a sigmoid to a unit step function:

$$\Psi_{Bit}(v) = \begin{cases} 1 & \text{if } v \geq \mu_\Psi \\ 0 & \text{otherwise,} \end{cases} \quad (18)$$

thus constraining the codomain of $C_m(i, j, k)$ to the binary values 0, 1, and *invalid*. In such an implementation, a given cylinder C_m can be stored as two bit-vectors $\mathbf{c}_m, \hat{\mathbf{c}}_m \in \{0, 1\}^{N_C}$, the former storing the cell values and the latter denoting the cell validities (see also (13)):

$$\begin{aligned} \mathbf{c}_m[\text{lin}(i, j, k)] &= \begin{cases} 1 & \text{if } C_m(i, j, k) = 1 \\ 0 & \text{otherwise,} \end{cases} \\ \hat{\mathbf{c}}_m[\text{lin}(i, j, k)] &= \begin{cases} 1 & \text{if } C_m(i, j, k) \neq \text{invalid} \\ 0 & \text{otherwise.} \end{cases} \end{aligned} \quad (19)$$

In practice, vector $\hat{\mathbf{c}}_m$ can be used as a bit-mask to select the valid bits in \mathbf{c}_m ; in this way, the two vectors defined in (15) and (16) can be calculated as follows:

$$\mathbf{c}_{a|b} = \mathbf{c}_a \text{ AND } \hat{\mathbf{c}}_b, \mathbf{c}_{b|a} = \mathbf{c}_b \text{ AND } \hat{\mathbf{c}}_a, \quad (20)$$

where AND denotes the *bitwise-and* between two bit-vectors and $\hat{\mathbf{c}}_{ab} = \hat{\mathbf{c}}_a \text{ AND } \hat{\mathbf{c}}_b$ is the intersection of the two masks. Finally, the similarity between the two cylinders can be computed as:

$$\gamma_{Bit}(a, b) = \begin{cases} 1 - \frac{\|\mathbf{c}_{a|b} \text{ XOR } \mathbf{c}_{b|a}\|}{\|\mathbf{c}_{a|b}\| + \|\mathbf{c}_{b|a}\|} & \text{if } C_a \text{ and } C_b \text{ are matchable} \\ 0 & \text{otherwise,} \end{cases} \quad (21)$$

where XOR denotes the *bitwise-exclusive-or* between two bit-vectors. Note that the norm of a bit-vector can be simply computed by calculating the square root of the number of bits with value one. Fig. 6 shows an example of cylinder obtained using the bit-based implementation.

Table 1 compares the number of floating-point and integer operations involved in the computation of the similarity between two cylinders for a normal and bit-based implementation, respectively. Note that the bit-based implementation requires only five floating-point operations and a very small number of integer and bitwise operations. Hence, (20) and (21) can be implemented very efficiently, even on light

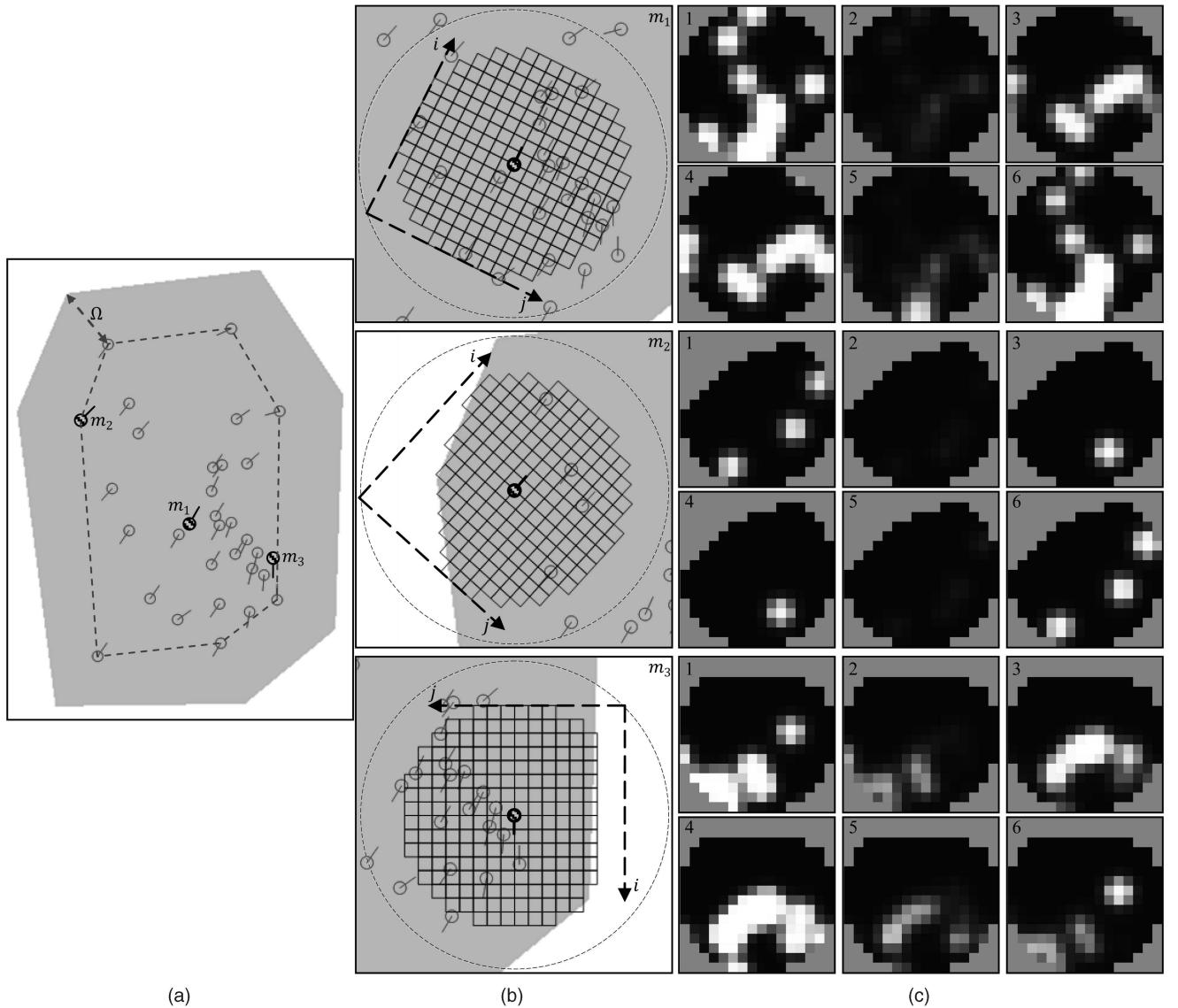


Fig. 5. (a) A minutiae template with the corresponding convex hull. For each of the three minutiae highlighted in (a), column (b) shows the base of the corresponding cylinder (only valid cells are drawn); minutiae within the dashed circles are those that contribute to the cylinder cell values. Column (c) shows the cell values of the three cylinders for each value of $k \in \{1, \dots, 6\}$ (lighter elements represent higher values); note that the cylinder sections in (c) are rotated according to the direction of the corresponding minutia.

architectures (e.g., smart cards) where floating-point operations are absent or very slow because they have to be replaced by surrogates (fixed-point arithmetic or software emulation).

4 GLOBAL SCORE AND CONSOLIDATION

In the previous section, a measure of local similarity between cylinders was proposed. In order to compare two minutiae templates (i.e., two fingerprints), a single value (*global score*) denoting their overall similarity has to be obtained from the local similarities. In the following, four simple techniques, inspired by ideas already proposed in the literature, are introduced to combine local similarities into a global score. While we are aware that more accurate techniques may be designed for global matching, here we intentionally focus on simple approaches that can be efficiently implemented even on light architectures.

The first two may be classified as “pure local techniques” since they only combine local similarities; the other two

implement a consolidation step to obtain a score that reflects to what extent the local relationships hold at global level. In the experimental evaluation where MCC is compared to three well-known local algorithms, these four techniques are applied both to MCC and to the other ones.

Given two ISO/IEC 19794-2 minutiae templates $A = \{a_1, a_2, \dots, a_{n_A}\}$ and $B = \{b_1, b_2, \dots, b_{n_B}\}$, let:

- $\gamma(a, b)$ be the local similarity between minutia $a \in A$ and $b \in B$, with $\gamma : A \times B \rightarrow [0, 1]$;
- $\Gamma \in [0, 1]^{n_A \times n_B}$ be a matrix containing all of the local similarities, with $\Gamma[r, c] = \gamma(a_r, b_c)$.

4.1 Local Similarity Sort (LSS)

This technique sorts all of the local similarities and selects the top n_P ; let P be the set of selected n_P minutiae-index pairs:

$$P = \{(r_t, c_t)\}, t = 1, \dots, n_P, 1 \leq r_t \leq n_A, 1 \leq c_t \leq n_B; \quad (22)$$

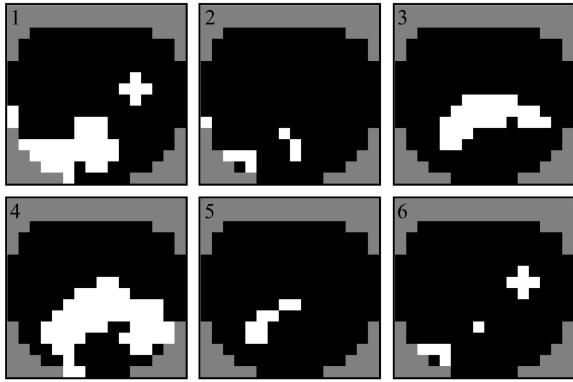


Fig. 6. The cell values of the cylinder associated to minutia m_3 in Fig. 5 using the bit-based implementation (black = 0, white = 1, gray = invalid).

the global score is calculated as the average of the corresponding local similarities:

$$S(A, B) = \frac{\sum_{(r,c) \in P} \Gamma[r, c]}{n_P}. \quad (23)$$

The value of n_P is not an overall constant since it partially depends on the number of minutiae in the two templates:

$$n_P = \min_{n_P} + \lfloor (Z(\min\{n_A, n_B\}, \mu_P, \tau_P)) \cdot (max_{n_p} - min_{n_p}) \rfloor, \quad (24)$$

where μ_P , τ_P , \min_{n_p} , and \max_{n_p} are parameters, Z is the sigmoid function defined in (5), and $\lfloor \cdot \rfloor$ denotes the rounding operator.

4.2 Local Similarity Assignment (LSA)

The Hungarian algorithm [46] is used to solve the linear assignment problem on matrix Γ , that is, to find the set of n_P pairs $P = \{(r_i, c_i)\}$ that maximizes $S(A, B)$ in (23) without considering the same minutia more than once (note that this is not guaranteed by LSS). The values of n_P and the global score are calculated as in (24) and (23), respectively.

4.3 Local Similarity Sort with Relaxation (LSS-R)

This technique is inspired by the relaxation approach initially proposed in [47] and recently applied to triangular minutiae structures in [33]. The basic idea is to iteratively modify the local similarities based on the compatibility among minutiae relationships. In particular, given a pair of minutiae (a, b) , if the global relationships among a and some other minutiae in A are compatible with the global relationships among b and the corresponding minutiae in B , then the local similarity between a and b is strengthened, otherwise it is weakened.

As a preliminary step, n_R pairs (r_t, c_t) are selected using the LSS technique, with $n_R = \min\{n_A, n_B\}$ (usually $n_R \gg n_P$).

Let $\lambda_t^0 = \Gamma[r_t, c_t]$ be the initial similarity of pair t ; the similarity at iteration i of the relaxation procedure is

$$\lambda_t^i = w_R \cdot \lambda_t^{i-1} + (1 - w_R) \cdot \left(\sum_{\substack{k=1 \\ k \neq t}}^{n_R} \rho(t, k) \cdot \lambda_k^{i-1} \right) / (n_R - 1), \quad (25)$$

TABLE 1
Number of Operations Required to Compute
the Similarity between Two Cylinders

	<i>Normal implementation</i>	<i>Bit-based implementation</i>		
as a function of N_C	for $N_C = 1536^\dagger$	as a function of N_C and rs^\ddagger	for $N_C = 1536$, $rs = 32$	
Square root extraction (float)	3	3	3	3
Multiplications and divisions (float)	$3 \cdot N_C + 1$	4609	1	1
Sums and subtractions (float)	$4 \cdot N_C - 1$	6143	1	1
Comparisons (i.e., checking if a value is invalid) (float)	$2 \cdot N_C$	3072	0	0
Sums (integer)	0	0	$\frac{3 \cdot N_C}{rs} - 2$	142
Counting number of 1's in a register	0	0	$\frac{3 \cdot N_C}{rs}$	144
Bitwise AND	0	0	$\frac{3 \cdot N_C}{rs}$	144
Bitwise XOR	0	0	$\frac{N_C}{rs}$	48

$\dagger N_C = 1536$ corresponds to $N_S = 16$ and $N_D = 6$, which are the default values in our implementation (see Table 2).

\ddagger Number of bits in the CPU registers.

where $w_R \in [0, 1]$ is a weighting factor and

$$\begin{aligned} \rho(t, k) &= \prod_{i=1}^3 Z(d_i, \mu_i^\rho, \tau_i^\rho), \\ d_1 &= |d_S(a_{r_t}, a_{r_k}) - d_S(b_{c_t}, b_{c_k})|, \\ d_2 &= |d\phi(d_\theta(a_{r_t}, a_{r_k}), d_\theta(b_{c_t}, b_{c_k}))|, \\ d_3 &= |d\phi(d_R(a_{r_t}, a_{r_k}), d_R(b_{c_t}, b_{c_k}))|. \end{aligned} \quad (26)$$

$\rho(t, k)$ is a measure of the compatibility between two pairs of minutiae: minutiae (a_{r_t}, a_{r_k}) of template A and minutiae (b_{c_t}, b_{c_k}) of template B . The compatibility value is based on the similarity between three features that are invariant for rotation and translation (see Fig. 7); it is calculated as the product of three terms, d_1 , d_2 , and d_3 , which are normalized by means of sigmoid functions (5) with specific parameters. d_1 denotes the similarity between the minutiae spatial distances, d_2 compares the directional differences, and d_3 compares the radial angles. The radial angle is defined as the angle subtended by the edge connecting the two minutiae and the direction of the first one (Fig. 7):

$$d_R(m_1, m_2) = d\phi(\theta_{m_1}, \text{atan2}(-y_{m_2} + y_{m_1}, x_{m_2} - x_{m_1})). \quad (27)$$

n_{rel} iterations of the relaxation procedure are executed on all the n_R pairs; then, similarly to [33], the efficiency of pair t is calculated as:

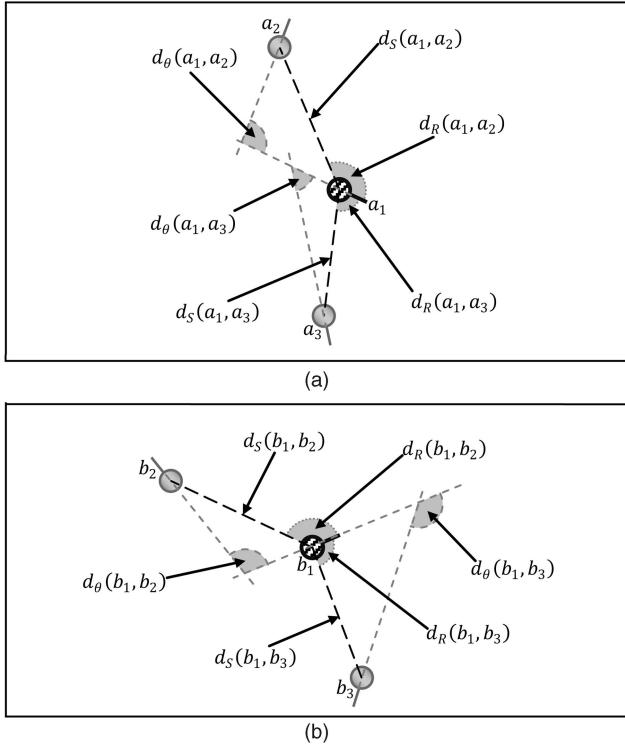


Fig. 7. An example of the global relationships considered in the relaxation procedure. The similarity λ_t^i between minutiae a_1 and b_1 is modified according to: 1) the compatibility between the global relationships $a_1 \leftrightarrow a_2$ and $b_1 \leftrightarrow b_2$ ($\rho(1, 2)$), 2) the compatibility between $a_1 \leftrightarrow a_3$ and $b_1 \leftrightarrow b_3$ ($\rho(1, 3)$). The three invariant features used to calculate $\rho(t, k)$ are graphically highlighted: 1) the spatial distances (dashed black lines), 2) the directional differences (gray angles with dashed border), and 3) the radial angles (gray angles with dotted border).

$$\varepsilon_t = \frac{\lambda_t^{n_{rel}}}{\lambda_t^0}. \quad (28)$$

Intuitively, a high efficiency is achieved for the pairs of minutiae whose similarity is substantially strengthened because of high compatibility with other pairs, whereas pairs of local structures that initially obtained a high similarity by chance will be penalized by the relaxation process and their final efficiency will be quite low.

To determine the global score, the n_P pairs with the largest efficiency are selected from the n_R pairs (the value of n_P is calculated as in (24)). The global score is computed as in (23), but using the relaxed similarity values $\lambda_t^{n_{rel}}$ instead of the values in matrix Γ .

4.4 Local Similarity Assignment with Relaxation (LSA-R)

This technique is identical to the previous one (LSS-R), except that, in the preliminary step, the n_R pairs (r_t, c_t) are selected with the LSA technique. The computation of the final score is identical to LSS-R as well: It is a simple average of the relaxed similarities $\lambda_t^{n_{rel}}$ of the n_P pairs with the largest efficiency.

5 EXPERIMENTAL EVALUATION

In this section, in order to evaluate accuracy and efficiency of MCC, experiments aimed at comparing it with minutiae-only

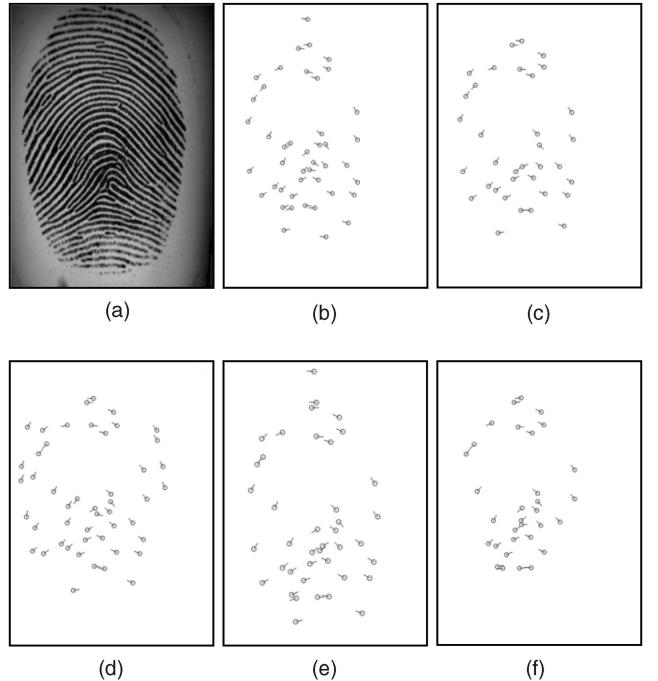


Fig. 8. (a) A fingerprint from FVC2006 DB2 and the corresponding ISO templates obtained by the five minutiae extractors (b)-(f).

implementations of three well-known local minutiae matching methods [6], [7], [12] are reported.

5.1 Benchmark Data Sets

In the first battery of experiments, all of the algorithms have been extensively evaluated on five data sets ($DS2a$, $DS2b$, $DS2c$, $DS2d$, $DS2e$) of ISO/IEC 19794-2 templates, derived from the fingerprint images in FVC2006 [48] DB2. These data sets have been obtained using five ISO-compliant minutiae extractors (identified in the following by the letters a , b , c , d , e) provided by five of the best-performing FVC2006 participants. Fig. 8 shows a fingerprint from FVC2006 DB2 with the five corresponding ISO templates. The choice of using FVC2006 DB2 as the principal data set is motivated by the fact that it was acquired with a large-area optical sensor of medium-high quality, which is well-suited for the algorithms evaluated since it allows a sufficiently large number of minutiae to be extracted. However, the same tests have also been performed on the other three FVC2006 databases; hence, in the following, the results are reported on a total of 20 data sets: $DS[1-4][a-e]$ (the number denotes the corresponding FVC2006 database and the letter the minutiae extractor). Each data set contains 1680 ISO/IEC 19794-2 templates, obtained from the 1,680 fingerprints in the corresponding FVC2006 database (140 fingers and 12 impressions per finger, see [48]).

Fig. 9 shows a sample fingerprint from each FVC2006 database; note that DB1 was acquired with a small area-scanner at 250 dpi, which is not well-suited for minutiae extraction and matching: This explains why error rates on the corresponding data sets $DS1[a-e]$ are high, not only for MCC, but also for the other minutiae-only algorithms it is compared against; see Section 5.4.

In all of the data sets, minutiae coordinates are encoded at 500 dpi.

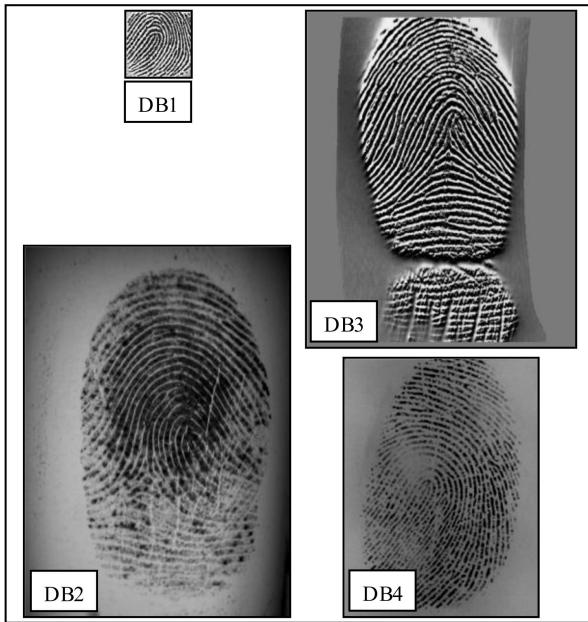


Fig. 9. A fingerprint from each FVC2006 database, at the same scale factor.

5.2 Algorithms Evaluated

Three versions of MCC and three minutiae-only implementations of well-known algorithms have been compared on the 20 data sets:

- MCC16—MCC with $N_S = 16$ (see Table 2);
- MCC16b—MCC with $N_S = 16$ and bit-based implementation (see Section 3.4);
- MCC8b—MCC with $N_S = 8$ (see Table 2) and bit-based implementation;
- *Jiang*—the local matching phase of the approach proposed in [6];
- *Ratha*—the local matching phase of the approach proposed in [7];
- *Feng*—the local matching phase of the approach proposed in [12].

Except for parameter N_S , all three versions of MCC use the same parameter values (Table 2); these values have been initially calibrated on *DB2d* since d is the most accurate of the five minutiae extractors, and then maintained steady for all the 19 remaining data sets. As to the other three algorithms, the parameter values specified in the original papers have been used; for parameters whose values were not given in the original papers, optimal values have been determined on *DB2d*. The algorithms have been implemented as described in the corresponding papers, except for a few minor changes:

- In *Jiang* and *Ratha*, the contribution of ridge-count information has been neglected since this information (not mandatory in the ISO/IEC 19794-2 template format) is not provided by any of the five extractors used in the experiments and this work focuses on algorithms using only the mandatory information in the ISO/IEC 19794-2 format.

TABLE 2
Parameter Values

Parameter(s)	Description	Value
R	Cylinder radius (in pixel)	70
N_S	Number of cells along the cylinder diameter	16 [MCC16(b)] 8 [MCC8b]
N_D	Number of cylinder sections	6
σ_S	Standard deviation in (7)	$\frac{28}{3}$
σ_D	Standard deviation in (11)	$\frac{2}{9}\pi$
μ_Ψ, τ_Ψ	Sigmoid parameters for function Ψ	$\frac{1}{100}, 400$
Ω	Offset applied to enlarge the convex hull (in pixel)	50
min_{VC}	Minimum number of valid cells for a cylinder to be valid	75% of the max. number of valid cells in a cylinder
min_M	Minimum number of minutiae for a cylinder to be valid	2
min_{ME}	Minimum number of matching elements in two matchable cylinders	60% of the max. number of matching elements
δ_θ	Maximum global rotation allowed between two templates	$\frac{\pi}{2}$
μ_P, τ_P	Sigmoid parameters in (24)	$20, \frac{2}{5}$
min_{n_p}, max_{n_p}	Minimum and maximum number of minutiae in (24)	4, 12
w_R	Weight parameter in (25)	$\frac{1}{2}$
μ_1^p, τ_1^p	Sigmoid parameters for d_1 (26)	$5, -\frac{8}{5}$
μ_2^p, τ_2^p	Sigmoid parameters for d_2 (26)	$\frac{\pi}{12}, -30$
μ_3^p, τ_3^p	Sigmoid parameters for d_3 (26)	$\frac{\pi}{12}, -30$
n_{rel}	Number of relaxation iterations for LSS-R and LSA-R	5

- in *Feng*, a minimum number of minutiae (three) has been required for a minutiae neighborhood to be valid (according to our experiments, without this correction, its accuracy markedly drops); furthermore, since information on the fingerprint pattern area (required in the original algorithm, see [12]) is not available in ISO/IEC 19794-2 templates, the fingerprint pattern area is approximated with the minutiae convex hull which is also used in MCC (see Section 3.1).

Both MCC and the other algorithms have been implemented in C#.

Each of the six algorithms has been combined with each of the global-scoring techniques described in Section 4: LSS, LSA, LSS-R, and LSA-R, thus obtaining a total of 24 matching approaches to be tested.

5.3 Test Protocol

For each data set, the FVC2006 testing protocol has been adopted:

TABLE 3
Accuracy of the Algorithms on the Five Data Sets Obtained from FVC2006 DB2 (Percentage Values)

	DS2a		DS2b		DS2c		DS2d		DS2e		
	EER	FMR ₁₀₀₀	EER	FMR ₁₀₀₀	EER	FMR ₁₀₀₀	EER	FMR ₁₀₀₀	EER	FMR ₁₀₀₀	
LSS	MCC16	2.07	5.35	1.44	3.34	6.62	21.23	0.46	1.02	2.69	7.70
	MCC16b	2.24	6.67	1.69	4.44	6.76	24.20	0.55	1.62	2.78	7.77
	MCC8b	2.28	7.12	1.73	5.23	7.54	26.43	0.59	1.92	2.88	8.34
	Jiang	5.37	16.50	6.50	13.82	16.48	38.33	3.23	7.72	8.82	19.69
	Ratha	9.11	34.72	11.68	39.73	18.68	51.28	7.78	32.20	10.93	37.33
LSA	Feng	3.52	7.36	4.58	11.52	11.09	23.81	2.51	5.17	5.33	12.2
	MCC16	1.97	4.61	1.14	2.67	5.87	15.44	0.33	0.69	2.31	5.78
	MCC16b	2.07	5.70	1.35	3.46	6.18	15.95	0.44	1.07	2.36	6.35
	MCC8b	2.07	5.99	1.47	3.81	7.03	21.37	0.45	1.12	2.57	6.09
	Jiang	5.11	15.57	6.75	13.92	17.27	36.85	3.20	6.97	9.08	21.23
LSS-R	Ratha	8.06	26.99	10.41	33.02	17.56	44.63	6.87	24.63	9.88	30.36
	Feng	3.42	6.83	4.36	10.44	11.09	22.38	2.17	4.45	5.18	11.02
	MCC16	1.41	2.52	0.64	1.20	3.19	7.15	0.21	0.24	1.17	2.15
	MCC16b	1.41	2.60	0.64	1.23	3.33	7.60	0.22	0.27	1.19	2.23
	MCC8b	1.46	3.05	0.67	1.18	3.82	7.99	0.20	0.28	1.37	2.62
LSA-R	Jiang	3.66	7.91	3.60	5.89	11.48	22.13	1.22	2.04	5.47	9.67
	Ratha	2.34	3.76	0.96	1.72	6.82	9.36	0.41	0.46	2.16	3.44
	Feng	3.27	5.76	4.35	9.25	11.11	22.44	2.03	3.66	5.39	11.02
	MCC16	1.23	1.98	0.48	0.73	2.98	5.91	0.15	0.18	1.04	2.04
	MCC16b	1.21	1.97	0.47	0.90	3.06	6.17	0.17	0.18	1.08	2.07
	MCC8b	1.23	2.14	0.59	0.89	3.66	7.11	0.18	0.25	1.28	2.41
	Jiang	4.06	7.98	3.54	6.40	11.00	20.83	1.22	2.02	5.12	9.56
	Ratha	2.91	5.10	1.12	1.93	8.03	10.94	0.49	0.58	2.78	4.42
	Feng	3.01	5.44	4.19	8.67	11.12	21.02	1.78	3.17	5.25	9.72

- Each template is compared against the remaining ones of the same finger to obtain the False Non Match Rate (FNMR). If template T_1 is compared against T_2 , the symmetric comparison (T_2 against T_1) is not executed to avoid correlation in the matching scores. The total number of genuine tests is: $\frac{12 \times 11}{2} \times 140 = 9,240$.
- The first template of each finger is compared against the first template of the remaining fingers in the data set, to determine the False Match Rate (FMR). If template T_1 is compared to T_2 , the symmetric comparison (T_2 against T_1) is not executed, to avoid correlation in the scores. The total number of impostor tests is: $\frac{140 \times 139}{2} = 9,730$.

In case of failure to process or match templates, the corresponding matching scores are set to zero.

For each algorithm and for each data set, the following performance indicators are considered:

- Equal-Error-Rate (EER) [49];
- FMR₁₀₀₀ (the lowest FNMR for FMR $\leq 0.1\%$) [34];
- average matching time, subdivided into:
 - T_{cs} : average time to create the local structures from an ISO/IEC 19794-2 template;
 - T_{ls} : average time to compute all the local similarities between the local structures obtained from two templates (i.e., to fill matrix Γ);
 - T_{gs} : average time to calculate the global score from the local similarities (i.e., from Γ);

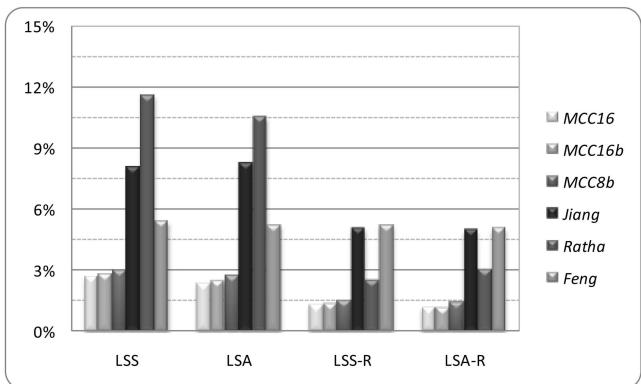


Fig. 10. Average EER over the five data sets DS2[a-e], for each of the four global-scoring techniques.

- Average memory size of the local structures created from a template, expressed in bytes.

5.4 Results: Accuracy

Table 3 reports the EER and FMR₁₀₀₀ of all of the algorithms, combined with the four global-scoring techniques, on DS2[a-e]. For each global-scoring technique, the best result on each data set is highlighted in bold; the overall best EER and FMR₁₀₀₀ are underlined. The graphs in Figs. 10 and 11 report, for each global-scoring technique, the average EER and FMR₁₀₀₀ over the five data sets, respectively; Fig. 12 reports the DET graph on DS2d, using the LSA-R technique.

It is worth noting that the best result is always achieved by one of the three versions of MCC and that any of the three versions is always more accurate than the other algorithms, except on DS2c with the LSS technique, where the FMR₁₀₀₀ of Feng (23.81 percent) is lower than that of MCC16b and MCC8b (24.20 percent and 26.43 percent, respectively). The overall best result is achieved by MCC16 on DS2d using the LSA-R technique (EER = 0.15 percent, FMR₁₀₀₀ = 0.18%); this result would put MCC16 in the ninth place in the ranking of the FVC2006 Open Category and in the second place in the Light Category (see [48]). Considering that FVC2006 algorithms do not rely only on ISO/IEC 19794-2 minutiae information, but typically exploit other features (e.g., orientation field, ridge density, etc.), we think that the accuracy obtained by MCC16 is definitely very

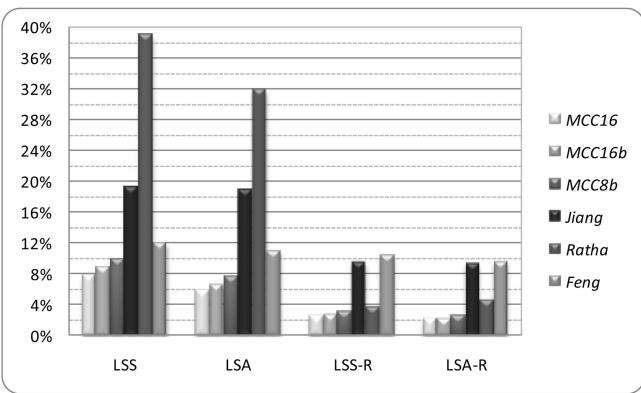


Fig. 11. Average FMR₁₀₀₀ over the five data sets DS2[a-e], for each of the four global-scoring techniques.

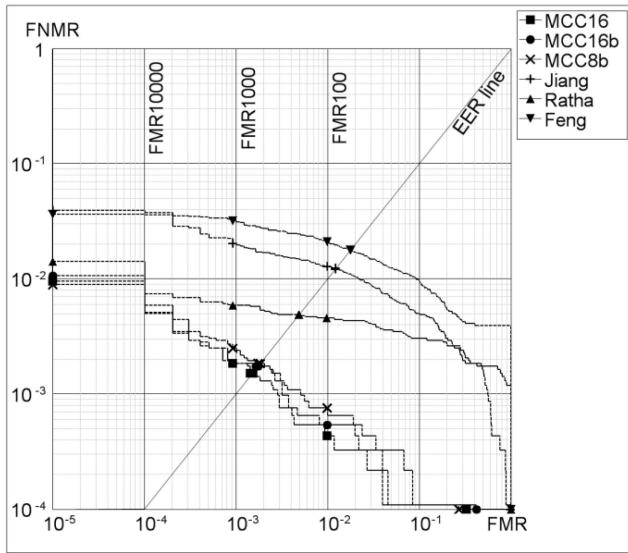


Fig. 12. DET graph of the six algorithms on *DS2d*, using LSA-R.

good. It is also worth noting that the accuracy drop of MCC bit-based implementations (with respect to the MCC normal implementation) is very limited.

Tables A.1, A.2, and A.3 in Appendix A, which can be found on the Computer Society Digital Library at <http://doi.ieeecomputersociety.org/10.1109/TPAMI.2010.52>, report the EER and FMR_{1000} of all the algorithms, combined with the four global-scoring techniques, on *DS*[1,3,4][a-e]. In each table, for each global-scoring technique, the best result on each data set is highlighted in bold and the overall best EER and FMR_{1000} are underlined. The corresponding graphs in Figs. A.1, A.3, A.5 and A.2, A.4, A.6, which can be found on the Computer Society Digital Library at <http://doi.ieeecomputersociety.org/10.1109/TPAMI.2010.52>, report the average EER and FMR_{1000} , respectively. Note that also, in each of these data sets, the most accurate results are always achieved by one of the three versions of MCC; the superiority of MCC is well evident from the graphs, which show how the average error rates are always below those of the other algorithms. As to the four proposed global scoring techniques, from the experiments it is evident that:

- as expected, the consolidation stage markedly increases the accuracy; however, MCC (and sometimes *Feng*) achieves a good accuracy even without consolidation;
- the use of the Hungarian algorithm to optimally solve the assignment algorithm, in spite of the computational overhead, leads to a small accuracy improvement and therefore its adoption is not advised when efficiency is a concern.

5.5 Results: Efficiency

Table 4 reports the average matching times measured, over all 20 data sets, on a Pentium 4 CPU (2.8 GHz): Note that four columns are reported for T_{gs} since it depends on the specific global-scoring technique used. From the table, the following observations may be made:

- The average time taken by MCC to create the local structures from an ISO/IEC 19794-2 template (T_{cs}) is

TABLE 4
Average Matching Times over All Data Sets (Milliseconds)

	T_{cs}	T_{ls}	T_{gs}			
			LSS	LSA	LSS-R	LSA-R
MCC16	21.0	21.0	0.5	4.3	2.7	4.7
MCC16b	17.3	1.2	0.5	4.3	2.8	4.7
MCC8b	4.2	0.3	0.5	4.2	2.9	4.8
Jiang	1.0	0.8	0.4	4.3	2.6	4.1
Ratha	1.0	250.7	0.5	4.3	2.8	4.4
Feng	0.2	12.3	0.5	2.4	2.8	3.1

within 4.2 ms and 21 ms. As was reasonable to expect, this time is higher than in the other algorithms, however, we believe this does not limit the applicability of MCC for the following reasons: 1) According to our experience (and keeping in mind the high margin for code optimization), even if this step were to be implemented on very light architectures, the 4.2 ms of *MCC8b* should not become more than one second; 2) match-on-card solutions would not need to perform the cylinder computation at verification time; in fact, the cylinder-set of the acquired fingerprint may be computed on the PC and the template stored inside the smart-card may already contain the cylinder-set pre-computed at enrollment stage¹; 3) in identification (one-to-many) applications, the cylinder-set needs to be precomputed only once for each template and not at each comparison.

- *MCC8b* exhibits the lowest local similarity computation time (the average T_{ls} is 0.3 ms): Note that this time refers to a C# implementation, without any assembly-language or hardware-oriented optimization that the bit-based nature of the similarity measure could allow.
- The average time taken to calculate the global score (T_{gs}) in general does not depend on the specific local matching algorithm, with a noticeable exception: The techniques based on the assignment problem (LSA and LSA-R) are definitely faster when coupled with *Feng*; this may be due to the specific distribution of local similarities produced by such an algorithm that, on average, requires fewer iterations of the Hungarian method.
- To provide a reference, we note that the average matching time over the four databases of the top 10 FVC2006 participants is much higher than MCC: 416 ms for the Open Category and 53 ms for the Light Category. However a direct comparison is not feasible since the times reported in FVC2006 correspond to “template against image” matching and therefore include one feature extraction, which is a time demanding task (see [48] and [34]).

Table 5 shows, for each algorithm, the average memory size of the local structures created from an ISO/IEC 19794-2 template: the average has been calculated over all the 20 data sets. The memory size is reported considering both raw format and compression with two general-purpose lossless

1. If the template needs to be stored in ISO format, the cylinder-set may be stored as proprietary data in the “extended data” section, see [36].

TABLE 5
Average Memory Size of the Local Structures,
over All Data Sets, Measured in Bytes

	Raw format	Compressed format (rar)		Compressed format (zip)	
	Size	Size	Ratio	Size	Ratio
MCC16	209253	103766	202%	104595	200%
MCC16b	7630	1457	524%	1642	465%
MCC8b	1913	605	316%	655	292%
Jiang	1068	608	176%	647	165%
Ratha	26543	19487	136%	20046	132%
Feng	1428	567	252%	614	233%

compression techniques: *rar* [50] and *zip* [51]. It is worth noting that:

- MCC16 requires a considerable amount of memory because it encodes cell values as floating-point data and, therefore, it is not suitable to run on resource-limited platforms. This is not the case of MCC16b and MCC8b.
- Without any compression, the local structures of MCC16b and MCC8b, although larger than those of Jiang and Feng, can be stored and managed into a typical smart card.
- The local structures of MCC16b and MCC8b can be compressed much more than the others, probably due to their bit-based composition: Once compressed, the local structure size of MCC8b is comparable to that of Jiang and Feng.
- The average template size of the top 10 FVC2006 participants is: 4,478 bytes for the Open Category (hence, higher than MCC8b) and 1,175 bytes for the Light Category (not far from MCC8b).

6 CONCLUSION

In this paper, we introduced Minutia Cylinder-Code (MCC): a novel minutiae-only representation and matching technique for fingerprint recognition. MCC relies on a robust discretization of the neighborhood of each minutia into a 3D cell-based structure named cylinder. Simple but effective techniques for the computation and consolidation of cylinder similarities are provided to determine the global similarity between two fingerprints.

In order to compare MCC with three well-known approaches, a systematic experimentation has been carried out, involving a total of 24 matching approaches (six algorithms and four global-scoring techniques) over 20 minutiae data sets extracted from FVC2006 databases, resulting in more than nine million matching attempts. Experimental results demonstrate that MCC is more accurate than well-known minutiae-only local matching techniques ([6], [7], [12]). MCC is also very fast and suitable to be simply coded in hardware due to the bit-wise nature of the matching technique; this allows its porting on inexpensive secure platforms such as a smart-card or a system-on-a-chip.

While in this paper we focused on the problem of robustly and efficiently matching two fingerprints, we

believe that the peculiarities of MCC also allow the development of new effective techniques for fingerprint indexing and template protection: These two issues are the main targets of our future research efforts.

ACKNOWLEDGMENTS

Patent pending (Cappelli, Ferrara, Maio, Maltoni: IT-BO2009A000149).

REFERENCES

- [1] D. Maltoni, D. Maio, A.K. Jain, and S. Prabhakar, *Handbook of Fingerprint Recognition*. second ed. Springer-Verlag, 2009.
- [2] N.K. Ratha, K. Karu, S. Chen, and A.K. Jain, "A Real-Time Matching System for Large Fingerprint Databases," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 18, no. 8, pp. 799-813, Aug. 1996.
- [3] S.H. Chang, F.H. Cheng, W.H. Hsu, and G.Z. Wu, "Fast Algorithm for Point Pattern Matching: Invariant to Translations, Rotations and Scale Changes," *Pattern Recognition*, vol. 30, pp. 311-320, 1997.
- [4] A.K. Hreichak and J.A. McHugh, "Automated Fingerprint Recognition Using Structural Matching," *Pattern Recognition*, vol. 23, pp. 893-904, 1990.
- [5] A.J. Willis and L. Myers, "A Cost-Effective Fingerprint Recognition System for Use with Low-Quality Prints and Damaged Fingertips," *Pattern Recognition*, vol. 34, pp. 255-270, 2001.
- [6] X. Jiang and W.Y. Yau, "Fingerprint Minutiae Matching Based on the Local and Global Structures," *Proc. Int'l Conf. Pattern Recognition*, vol. 2, pp. 6038-6041, 2000.
- [7] N.K. Ratha, V.D. Pandit, R.M. Bolle, and V. Vaish, "Robust Fingerprint Authentication Using Local Structural Similarity," *Proc. IEEE Workshop Applications of Computer Vision*, pp. 29-34, 2000.
- [8] T.Y. Jea and V. Govindaraju, "A Minutia-Based Partial Fingerprint Recognition System," *Pattern Recognition*, vol. 38, pp. 1672-1684, 2005.
- [9] S. Chikkerur, A.N. Cartwright, and V. Govindaraju, "K-plet and Coupled BFS: A Graph Based Fingerprint Representation and Matching Algorithm," *Lecture Notes in Computer Science*, vol. 3832, pp. 309-315, Springer, 2006.
- [10] D. Kwon, I.D. Yun, D.H. Kim, and S.U. Lee, "Fingerprint Matching Method Using Minutiae Clustering and Warping," *Proc. Int'l Conf. Pattern Recognition*, vol. 4, 2006.
- [11] H. Chen, J. Tian, and X. Yang, "Fingerprint Matching with Registration Pattern Inspection," *Lecture Notes in Computer Science*, pp. 327-334, Springer, 2003.
- [12] J. Feng, "Combining Minutiae Descriptors for Fingerprint Matching," *Pattern Recognition*, vol. 41, pp. 342-352, 2008.
- [13] X. Tan and B. Bhanu, "A Robust Two Step Approach for Fingerprint Identification," *Pattern Recognition Letters*, vol. 24, pp. 2127-2134, 2003.
- [14] G. Parziale and A. Niel, "A Fingerprint Matching Using Minutiae Triangulation," *Lecture Notes in Computer Science*, pp. 241-248, Springer, 2004.
- [15] X. Chen, J. Tian, J. Yang, and Y. Zhang, "An Algorithm for Distorted Fingerprint Matching Based on Local Triangle Feature Set," *IEEE Trans. Information Forensics and Security*, vol. 1, no. 2, pp. 169-177, June 2006.
- [16] D.Q. Zhao, F. Su, and A. Cai, "Fingerprint Registration Using Minutia Clusters and Centroid Structure," *Proc. Int'l Conf. Pattern Recognition*, pp. 413-416, 2006.
- [17] W. Xu, X. Chen, and J. Feng, "A Robust Fingerprint Matching Approach: Growing and Fusing of Local Structures," *Lecture Notes in Computer Science*, vol. 4642, p. 134, Springer, 2007.
- [18] X. Linag, A. Bishnu, and T. Asano, "A Robust Fingerprint Indexing Scheme Using Minutia Neighborhood Structure and Low-Order Delaunay Triangles," *IEEE Trans. Information Forensics and Security*, vol. 2, no. 4, pp. 721-733, Dec. 2007.
- [19] M. Tico and P. Kuosmanen, "Fingerprint Matching Using an Orientation-Based Minutia Descriptor," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 25, no. 8, pp. 1009-1014, Aug. 2003.
- [20] J. Qi and Y. Wang, "A Robust Fingerprint Matching Method," *Pattern Recognition*, vol. 38, pp. 1665-1671, 2005.

- [21] E. Zhu, J. Yin, and G. Zhang, "Fingerprint Matching Based on Global Alignment of Multiple Reference Minutiae," *Pattern Recognition*, vol. 38, pp. 1685-1694, 2005.
- [22] X. Wang, J. Li, and Y. Niu, "Fingerprint Matching Using Orientationcodes and PolyLines," *Pattern Recognition*, vol. 40, pp. 3164-3177, 2007.
- [23] Y. He, J. Tian, L. Li, H. Chen, and X. Yang, "Fingerprint Matching Based on Global Comprehensive Similarity," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 28, no. 6, pp. 850-862, June 2006.
- [24] X. He, J. Tian, L. Li, Y. He, and X. Yang, "Modeling and Analysis of Local Comprehensive Minutia Relation for Fingerprint Matching," *IEEE Trans. Systems, Man, and Cybernetics, Part B*, vol. 37, no. 5, pp. 1204-1211, Oct. 2007.
- [25] H. Wei, M. Guo, and Z. Ou, "Fingerprint Verification Based on Multistage Minutiae Matching," *Proc. Int'l Conf. Pattern Recognition*, pp. 1058-1061, 2006.
- [26] G.S. Ng, X. Tong, X. Tang, and D. Shi, "Adjacent Orientation Vector Based Fingerprint Minutiae Matching System," *Proc. Int'l Conf. Pattern Recognition*, pp. 528-531, 2004.
- [27] X. Tong, J. Huang, X. Tang, and D. Shi, "Fingerprint Minutiae Matching Using the Adjacent Feature Vector," *Pattern Recognition Letters*, vol. 26, pp. 1337-1345, 2005.
- [28] L. Sha, F. Zhao, and X. Tang, "Minutiae-Based Fingerprint Matching Using Subset Combination," *Proc. Int'l Conf. Pattern Recognition*, pp. 566-569, 2006.
- [29] J. Feng, Z. Ouyang, and A. Cai, "Fingerprint Matching Using Ridges," *Pattern Recognition*, vol. 39, pp. 2131-2140, 2006.
- [30] Y. Zhang, X. Yang, Q. Su, and J. Tian, "Fingerprint Recognition Based on Combined Features," *Lecture Notes in Computer Science*, vol. 4642, p. 281, Springer, 2007.
- [31] D. Lee, K. Choi, and J. Kim, "A Robust Fingerprint Matching Algorithm Using Local Alignment," *Proc. Int'l Conf. Pattern Recognition*, vol. 16, pp. 803-806, 2002.
- [32] L. Sha and X. Tang, "Orientation-Improved Minutiae for Fingerprint Matching," *Proc. Int'l Conf. Pattern Recognition*, vol. 4, pp. 432-435, 2004.
- [33] Y. Feng, J. Feng, X. Chen, and Z. Song, "A Novel Fingerprint Matching Scheme Based on Local Structure Compatibility," *Proc. Int'l Conf. Pattern Recognition*, pp. 374-377, 2006.
- [34] R. Cappelli, D. Maio, D. Maltoni, J.L. Wayman, and A.K. Jain, "Performance Evaluation of Fingerprint Verification Systems," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 28, no. 1, pp. 3-18, Jan. 2006.
- [35] "Data Format for the Interchange of Extended Fingerprint and Palmprint Features—Addendum to ANSI/NIST-ITL 1-2007," *ANSI/NIST, Working Draft 0.2*, 2008.
- [36] ISO/IEC 19794-2:2005, Information Technology—Biometric Data Interchange Formats—Part 2: Finger Minutiae Data, 2005.
- [37] "INCITS 378-2004—Finger Minutiae Format for Data Interchange," *ANSI/INCITS Standard*, 2004.
- [38] P. Grother, W. Salamon, C. Watson, M. Indovina, and P. Flanagan, "MINEX II: Performance of Fingerprint Match-on-Card Algorithms," technical report, 2007.
- [39] A.K. Jain, K. Nandakumar, and A. Nagar, "Biometric Template Security," *EURASIP J. Advances in Signal Processing*, vol. 8, 2008.
- [40] A. Jules and M. Sudan, "A Fuzzy Vault Scheme," *Proc. Int'l Symp. Information Theory*, 2002.
- [41] U. Uludag and A.K. Jain, "Fuzzy Fingerprint Vault," *Proc. Workshop Biometrics: Challenges Arising from Theory to Practice*, pp. 13-16, 2004.
- [42] U. Uludag and A.K. Jain, "Securing Fingerprint Template: Fuzzy Vault with Helper Data," *Proc. Computer Vision and Pattern Recognition Workshop*, pp. 163-171, 2006.
- [43] J. Jeffers and A. Arakala, "Minutiae-Based Structures for a Fuzzy Vault," *Proc. Biometric Consortium Conf.*, pp. 1-6, 2006.
- [44] J. Jeffers and A. Arakala, "Fingerprint Alignment for a Minutiae-Based Fuzzy Vault," *Proc. Biometrics Symp.*, pp. 1-6, 2007.
- [45] F.P. Preparata and M.I. Shamos, *Computational Geometry: An Introduction*. Springer, 1985.
- [46] H.W. Kuhn, "The Hungarian Method for the Assignment Problem," *Naval Research Logistics Quarterly*, vol. 2, pp. 83-97, 1955.
- [47] A. Rosenfeld, R.A. Hummel, and S.W. Zucker, "Scene Labeling by Relaxation Operations," *IEEE Trans. Systems, Man, and Cybernetics*, vol. 6, no. 6, pp. 420-433, June 1976.
- [48] BioLab, FVC2006 Web Site, <http://bias.csr.unibo.it/fvc2006/>, Nov. 2009.
- [49] D. Maio, D. Maltoni, R. Cappelli, J.L. Wayman, and A.K. Jain, "FVC2000: Fingerprint Verification Competition," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 24, no. 3, pp. 402-412, Mar. 2002.
- [50] Wikipedia, Rar File Format, "<http://en.wikipedia.org/wiki/Rar>" <http://en.wikipedia.org/wiki/Rar>, Nov. 2009.
- [51] Wikipedia, Zip File Format, "[http://en.wikipedia.org/wiki/ZIP_\(file_format\)](http://en.wikipedia.org/wiki/ZIP_(file_format))" [http://en.wikipedia.org/wiki/ZIP_\(file_format\)](http://en.wikipedia.org/wiki/ZIP_(file_format)), Nov. 2009.



Raffaele Cappelli received the Laurea degree cum laude in computer science from the University of Bologna in 1998, and the PhD degree in computer science and electronic engineering from DEIS, University of Bologna, in 2002. He is an associate researcher at the University of Bologna, Italy. He teaches pattern recognition in the Department of Computer Science and he is a member of the Biometric System Laboratory (University of Bologna, Cesena, Italy). His research interests include pattern recognition, image retrieval by similarity and biometric systems (fingerprint classification and recognition, synthetic fingerprint generation, fingerprint aliveness detection, fingerprint scanner quality, face recognition, and performance evaluation methodologies). He is a member of the IEEE.



Matteo Ferrara received the bachelor's degree cum laude in computer science from the University of Bologna in March 2004, the master's degree cum laude in October 2005, and the PhD degree from the Department of Electronics, Computer Science, and Systems—DEIS, University of Bologna. During his PhD studies, which he completed in 2009, he worked on biometric fingerprint recognition systems. Today, he is an assistant teacher in computer architectures and pattern recognition courses on the Faculty of Computer Science, University of Bologna, and he is a member of the Biometric System Laboratory (Cesena, Italy). His research interests include image processing, pattern recognition, biometric systems (fingerprint recognition, performance evaluation of biometric systems, fingerprint scanner quality, and face recognition), information security, cryptography, digital signature, and smart card.



Davide Maltoni is an associate professor at the University of Bologna (Department of Electronics, Informatics, and Systems—DEIS). He teaches computer architectures and pattern recognition in the Department of Computer Science, University of Bologna, Cesena, Italy. His research interests are in the area of pattern recognition and computer vision. In particular, he is active in the field of biometric systems (fingerprint recognition, face recognition, hand recognition, performance evaluation of biometric systems). He is a codirector of the Biometric Systems Laboratory (Cesena, Italy), which is internationally known for its research and publications in the field. He is an author of two books: *Biometric Systems, Technology, Design and Performance Evaluation* (Springer, 2005) and *Handbook of Fingerprint Recognition* (Springer, 2003), which received the PSP award from the Association of American Publishers. He is currently an associate editor of the journals: *Pattern Recognition* and the *IEEE Transactions on Information Forensic and Security*. He is a member of the IEEE.

▷ For more information on this or any other computing topic, please visit our Digital Library at www.computer.org/publications/dlib.