Minutia Template Exchange Format

Ruud M. Bolle, Nalini K. Ratha, Andrew Senior and Sharath Pankanti
IBM Thomas J. Watson Research Center
Yorktown Heights, NY 10598
{bolle,ratha, aws, sharat}@watson.ibm.com

Abstract

Fingerprints have been used for authentication and identification purposes for centuries. Most of the fingerprint identification/authentication systems are based on "minutia features" though there are other image correlation-based systems. In the minutia-based systems, vendors use many "private" features to improve accuracy. This becomes an obstacle for easy exchange of templates to support inter-operability of fingerprint templates. In this paper, we propose a minutia template exchange format keeping inter-operability in mind.

1 Background and Motivation

A common minutia template is an effort to describe the publicly "known" elements of a minutia template in an open format for minutia-based fingerprint authentication and identification systems. Even though vendors are allowed to use additional private information for improving the accuracy of their matchers, it is expected that the common minutia template will at least ensure a minimum level of performance for the user. Moreover, a common minutia template will result in customer confidence for using fingerprints in corporate-wide security applications where the overall solution is not expected to be dependent on a specific vendor. For example, consider the use of fingerprints for authentication in automatic teller machines (ATMs). An ATM card issued by a bank could hold one vendor's template where as an ATM from another bank could use a second vendor's matcher. To provide service at the ATMs, it is highly desirable that they support open exchange of the templates.

Efforts have been made in the past by ANSI [1] and presently by NIST [2] to arrive at a standard for minutia format. Here, we propose a similar common minutia format amenable for easy exchange. The vendors will ensure quality of the minutia extraction by verifying compliance with a database that will be provided. The paper is organized as follows. Section 2 provides useful definitions and conventions used. File formats are specified in section 3. Compliance procedures are briefly presented in section 4 and conclusions are described in section 5.

2 Definitions

The following definitions are used in this proposal:
1. The minutia coordinate system is shown in Fig. 1.
2. A reasonable definition of a ridge ending and the associated angle are as shown in Fig. 2.
3. A reasonable definition of a ridge bifurcation and the associated angle are as shown in Fig. 3.
4. Metric units will be used for representing distance wherever applicable.
5. For each minutia, the location (x,y), ridge angle (θ) and the minutia type (T) will be made available in a suitable format.
6. The minutia type can be one of the following options: ridge bifurcation, ridge ending, and unknown.
7. The distances when expressed in world coordinates will be in multiples of 1/1000 of a centimeter.
8. The angle will be quantized to a power of 2 (i.e., 2^N) levels.
All numbers are assumed to be unsigned and hence positive.

3 File Format

The proposed common minutia template consists of the following three components: (i) a mandatory header section; (ii) a mandatory public section and (iii) an optional private section. The following sections describe the components in detail.

3.1 Header

The header section contains descriptive information as listed in Table 1. It starts with a four byte long "magic number". The first three bytes of the magic number are "CMT", an abbreviation for the "Common Minutia Template". The fourth byte represents the mode of

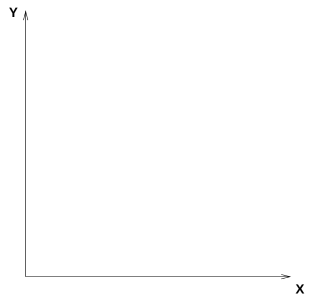

Figure 1. Minutiae coordinate system.

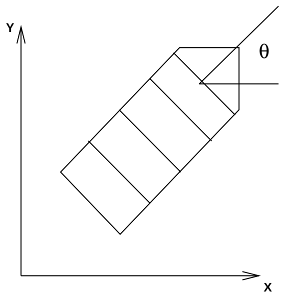

Figure 2. Ridge ending and associated angle.

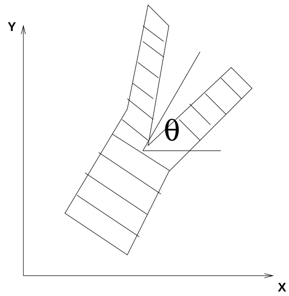

Figure 3. Ridge bifurcation and associated angle.

Field description	Field length	Example	Comments
Magic number (signature)	4 bytes	"CMT1"	Binary format
Total length of the template	4 bytes	0x2000	Size in bytes (sum of size of header+public+private)
Offset to private component	4 bytes	0x1530	Offset in bytes
Generating vendor identification	4 bytes	"IBM "	Vendor generating the template
Feature extraction software version	4 bytes	"2.3 "	In string format
Scanner used	8 bytes	"DBI235 "	In string format
Pixel aspect ratio	2 bytes	0x1009	16/9; Y- first byte; X- second byte
Sensor resolution in X	2 bytes	0x4CE5	In pixels per meter
Ridge definition	4 bits	0x0	Black ridge = 0 White ridge = 1
Distance unit	4 bits	0x0	=0 in pixels =1 in units of 1/1000 cm (1 cm is 1000 units)
Data resolution (X and Y,θ, T) (in bits)	2 bytes	0x0F31	4 bits for each, off by one (e.g., x, y: 16 bits, θ: 4 bits and T: 2 bits)

Table 1: Header description in compact (binary) mode.

the template. There are two modes of describing the open minutia template: (a) compact (binary) and (b) descriptive (ASCII).

The compact representation is designed to give a smaller sized minutia template and would be useful for storage constrained cards such as magnetic stripe-based credit cards. The descriptive version is a full-fledged version of the template in ASCII that could be conveniently viewed/edited on any computer editor. Note that the use of ASCII mode does not require the private component to be written in ASCII. The fourth byte of the magic number is assigned the value '0' (0x30) for ASCII mode and ('1'or '2' =0x31 or 0x32) for binary mode.

Currently, in compact mode, two number representation techniques: little-endian (0x31) and big-endian (0x32), are supported. The numeric fields in binary mode should be interpreted accordingly. However, in the ASCII mode the numbers are represented as a sequence of digits separated by a field separator (0x0A=LF).

In the binary mode, the header size is fixed to 42 bytes. The public component of the record in this mode can be accessed by this offset. However, in the ASCII mode, the fields are allowed to have variable lengths and are separated by the field separator described earlier. Hence an offset from the beginning of the file to the beginning of the public component is provided. Similarly an offset to the private section is also made available. The total size of the record is also available as a field in both the modes. The total size and the offsets are in bytes. In the ASCII mode, the fields in the header section are represented by ASCII character strings. If a field is numeric, the string should be interpreted accordingly. All the fields in the header are mandatory except the comment field in ASCII mode. Note that multiple lines of comment are permissible. The public component offset should be used to reach the beginning of the public component. The following fields: scanner used, the vendor ID, software version and the magic number are directly interchangeable between the two modes, i.e., they have same values in both the modes (except for the flag specifying the binary or ASCII mode).

Field description	Field length	Example	Comments
Magic number (signature)	4 chars	“CMT0”	ASCII format
Total length of the template	4 chars	“2048”	Size in bytes (sum of size of header+public+private)
Offset to public component	4 chars	“100”	Offset in bytes
Offset to private component	4 chars	“1530”	Offset in bytes
Date of image acquisition	8 chars	“19990315”	In string format; “yyyymmdd”
Date of feature extraction	8 chars	“19990415”	Same as above
Generating vendor identification	4 chars	“IBM ”	Vendor generating the template
Feature extraction software version	4 chars	“2.3 ”	In string format;
Scanner used	8 chars	“DBI235 ”	In string format
Image size	Variable	“640x480”	In string format; rows x cols
Pixel aspect ratio	Variable	“16/9”	In string format: Y/X: 16/9
Sensor resolution in X	Variable	“196”	In string format (pixels per cm)
Ridge definition	1 char	“0”	In string format; Black ridge = 0 White ridge = 1
Distance unit	1 char	“0”	In string format 0: in pixels 1: in units of 0.01 mm (1 cm is 1000 units)
Theta unit	Variable	“16”	In string format; Number of levels in 360 degrees
Comment	Variable	“A comment string”	In ASCII string form; multi-line comments are permissible

Table 2: Header description in ASCII mode.

The ridge definition field is set to 1 if the scanner produces white ridges or the features are really marked on valleys. Note that the exact location of the ridge feature could be different depending on the convention used for the ridge and the characteristics of the ridge location and thinning algorithms.

The aspect ratio should be specified in the simplest form, i.e., 16/9 should not be specified as 32/18. The scanner used field should be assigned a default value “unknown” when the scanner is not known as in the case of using previously scanned fingerprint images. The use of default values is to be strongly discouraged. The pixel aspect ratio is represented by a ratio of two numbers. In the binary mode, the two numbers Y and X are represented by two unsigned characters and in the ASCII mode, Y and X are represented by two integers written in ASCII string mode (e.g., “16/9”).

The sensor resolution in X-axis is expressed in terms of pixels per meter. The Y-resolution can be computed from the X-resolution and the pixel aspect ratio. The location of a minutia is expressed either in pixel units or in world coordinates as multiples of 1/1000 cm. The ridge direction at a minutia point is expressed in the units of a power of 2 fraction of 360 degrees. The theta unit in the header specifies this. In the compact mode, the number of bits for (X, Y, θ and T) must be specified. X and Y are assumed to have the same number

Field description	Field length	Example	Comment
Minutia count	1 byte	050	Unsigned char
For each minutia			
X_i	K bits	0x32	K obtained from data resolution in header
Y_i	K bits	0x45	Same as above
θ_i	M bits	0x16	M obtained from data resolution in header
Minutia type T_i	N bits	0x2	N obtained from header (Ending=0/ Bifurcation=1/ Unknown=2)

Table 3: Description of the public component in compact mode.

Field description	Field length	Example	Comment
Minutia count	Upto 3 char	50	Char string
For each minutia			
X_i	Upto 4 char	200	Char string
Y_i	Upto 4 char	241	Char string
θ_i	Upto 4 char	12	Char string
Minutia type T_i	1 char	U	Char (Ending=E/ Bifurcation=B/ Unknown=U)

Table 4: Description of the public component in ASCII mode.

of bits. For example, 16-bit integers for X (and Y) are expressed by 0xF for the field in the header. The ridge angle and minutia type field widths are specified similarly. Note that in the header the data resolution field stores the values K-1, M-1 and N-1. The image size field specifies the number of rows and columns in the image. This field is not available in compact mode.

3.2 Public component

This component describes the details of the minutiae. It contains the following information about the minutiae. Other details of minutia information, if used, should be stored in the private component.

Note that the actual values of K, M and N are obtained from the header field “data resolution”. For operational reasons, it should be ensured that a single minutia record occupies an integer multiple of bytes. The minutia type T can be assigned more bits to reach a byte boundary although only two or three values are used. There is no explicit minutia field separator in this mode.

Note that the angles have been assumed to use 2^N quantized levels. X, Y and θ are to be treated as numbers although they are represented as ASCII character strings. The components of a minutia are separated by a single space character (0x20). The field separator character (LF) is used to separate minutiae.

3.3 Private component

An optional binary coded vendor specific structure that could contain any vendor-specific information about the fingerprint such as vendor-specific minutia details and non-minutia features. It could also include singular point details such as core and delta locations. Such information can be stored in any format and can be encrypted, digitally signed or compressed as desired.

4 Compliance

For compliance testing, a database of fingerprint images will be provided along with the expected common minutiae format in ASCII and compact modes. The vendors ensure the correctness of their templates by examining these templates. For matcher compliance, a second database will be provided and the vendors will provide the common minutiae templates for the images in this database. In order to verify compliance, (i) the private component will be deleted. (ii) the public minutiae component will be matched and the error rates will be reported.

5 Conclusions

In this paper, we have set out a file format for the exchange of fingerprint feature information between different fingerprint systems. The definition of a compact but comprehensive template standard will ensure inter-operability of fingerprint systems from different vendors, encouraging open standards and growth in the industry.

While the public exchange of minutia information between feature extraction software from different vendors is not sufficient to achieve the maximum level of performance of any vendor’s software, this format will ensure a usable level of accuracy in an open environment, while retaining high accuracy, proprietary authentication with templates generated by a vendor’s own software.

References

[1] American National Standard for Information Systems, “Data format for the interchange of fingerprint information, ANSI/NIST-CSL 1-1993, Nov. 1993.

[2] F. Podio, J. Dunn, L. O’Gorman and P. Collier, “Common biometric exchange file format”, Version 4, April 1999.