NISTIR 6534

NIST Special Database 27
Fingerprint Minutiae From Latent and Matching Tenprint Images

Michael D. Garris
R. Michael McCabe

U.S. DEPARTMENT OF COMMERCE
Technology Administration
Information Technology Laboratory
Information Access and User
Interfaces Division
National Institute of Standards
and Technology
Gaithersburg, MD 20899

June 2000

U.S. DEPARTMENT OF COMMERCE
William M. Daley, Secretary

TECHNOLOGY ADMINISTRATION
Dr. Cheryl L. Shavers, Under Secretary
of Commerce for Technology

NATIONAL INSTITUTE OF STANDARDS
AND TECHNOLOGY
Raymond G. Kammer, Director

ACKNOWLEDGEMENTS

We would like to acknowledge the Federal Bureau of Investigation who provided funding and resources in conjunction with NIST to support the development of this fingerprint minutiae and image database.

TABLE OF CONTENTS

LIST OF TABLES AND FIGURES

NIST Special Database 27
Fingerprint Minutiae from Latent and Matching Tenprint Images

Michael D. Garris (mgarris@nist.gov) & R. Michael McCabe (mccabe@nist.gov)

ABSTRACT

The National Institute of Standards and Technology in conjunction with the Federal Bureau of Investigation has developed a new database of grayscale fingerprint images and corresponding minutiae data. The database contains latent fingerprints from crime scenes and their matching rolled fingerprint mates. In all there are 258 latent cases. Each case includes the latent image, the matching tenprint image, and four sets of minutiae that have been validated by a professional team of latent examiners. One set of minutiae contains all minutiae points on the latent fingerprint; the second set contains all minutiae points on the tenprint mate; the other two sets contain the minutiae points in common between the latent fingerprint and tenprint mate. In all there are 27,426 minutiae recorded across the set of tenprints with 5460 minutiae in common with their matching latent fingerprint. All data files are formatted according to the ANSI/NIST-ITL 1-2000 standard using Type-1, 9, 13, & 14 records. Software utilities are provided to read, write, and manipulate these files. The database can be used to develop and test new fingerprint algorithms, test commercial and research AFIS systems, train latent examiners, and promote the ANSI/NIST file format standard.

Keywords: ANSI/NIST, database, FBI, fingerprint, grayscale, IAFIS, image, latent, minutiae, tenprint

1. INTRODUCTION

The Federal Bureau of Investigation (FBI) has been utilizing computer technology to help capture, store, and search fingerprint records since the late 70's. In the early 90's, they began developing a system to enable electronic filing of fingerprint records and images by law enforcement agencies and to handle electronic transactions between these agencies. This new system is called the Integrated Automated Fingerprint Identification System (IAFIS) and it is currently in operation in Clarksburg, WV.

The National Institute of Standards and Technology (NIST) has had a long standing relationship with the FBI. Researchers at NIST began work on the first version of the FBI's AFIS system back in the late 60's. Over the years, NIST has conducted fingerprint research, developed fingerprint technology and standards, developed methods for measuring the quality and performance of fingerprint scanners and imaging systems, and has produced a large repository of databases of FBI fingerprint images.[1]-[27]

IAFIS has been primarily designed to process fingerprints that have been captured at a booking station of a jail or that are being submitted for a civilian background check. These types of fingerprints are typically taken by inking and rolling the fingertip onto a paper fingerprint card or captured from the surface of a live scan device. Traditionally these fingerprints have been referred to as *tenprints*, as all ten fingers are typically captured.

Over the years, the FBI has accumulated more than 40,000,000 fingerprint cards on file, and they handle up to 60,000 fingerprint-related requests a day. In addition to these, there is a smaller population of fingerprints also important to the FBI. These are fingerprints captured at crime scenes that can be used as evidence in solving criminal cases. Unlike tenprints, which have been captured in a relatively controlled environment for the expressed purpose of identification, crime scene fingerprints are by nature incidentally left behind. They are often invisible to the eye without some type of chemical processing or dusting. It is for this reason that they have been traditionally called *latent* fingerprints.

As one would expect, the composition and quality of latent fingerprints are significantly different from tenprints. Typically, only a portion of the finger is present in the latent, the surface on which the latent was imprinted is unpredictable, and the clarity of friction skin details are often blurred or occluded. All this leads to fingerprints of significantly lesser quality than typical tenprints. Figure 1 shows a "good" quality latent on the left and its matching tenprint on the right.

Figure 1. Latent fingerprint (left) with matching tenprint (right).

NIST has recently begun efforts to examine how well tenprint technology applies to latent fingerprints, and how AFIS technology might be adapted and improved to process latents reliably. Studies in the lab show the current state of the art lacking with respect to processing latent fingerprint images.

As part of the IAFIS contracting process, the FBI collected a sample of latent fingerprint images to be used in a Basic Demonstration Model (BDM). This collection of data gave prospective vendors the opportunity to work with real latent fingerprint images. The FBI provided the latent fingerprint data collected for the BDM to NIST for the purpose of preparing it for public distribution. The result is *NIST Special Database 27*: "Minutiae from Latents and Matching Tenprint Images."

This database contains 258 cases. Each case consists of a latent image and its matching tenprint image (referred to as the *mate*). The scanning resolutions and sizes of these grayscale images are discussed in Section 2.1. In addition to images, this database contains characteristic features (called *minutiae*) that have been recorded from each fingerprint. Each minutia is represented by its location and orientation in the image as described in Section 2.3. AFIS systems use these minutiae points rather than the entire image to search their databases for possible identifications. In total, there are 5460 minutiae recorded that are in common between the latent fingerprints and their tenprint mates.

The minutiae identified on each image in this database were validated by at least two professional latent examiners at the FBI. Therefore, the reported minutiae should be considered reliable and useful as ground truth for measuring performance. There is potentially broad application for this collection of fingerprint data and images. The database can be used by researchers to develop and test new algorithms specifically designed to process latent fingerprints; it can be used to test commercial and research AFIS systems; and it can be used to train latent examiners to properly encode latent fingerprints prior to search.

The database is approximately 330 Mbytes in size. The images in the database are not compressed, so they represent the bulk of the database's size. The minutiae attributes (location and orientation) are encoded as ASCII text and take up significantly less storage.

All the data files in this distribution are formatted according to ANSI/NIST-ITL 1-2000 "Data Format for the Interchange of Fingerprint, Facial, Scar Mark & Tattoo (SMT) Information" standard.[28] This file format has been available to law enforcement agencies in the U.S. since 1986 for the electronic exchange of fingerprint images and related data.[9] Today, this standard supports other types of images as well, including palmprints, mugshots, scars, and tattoos. This standard has been adopted by all major law enforcement agencies in the U.S., including the FBI, and has strong support and use internationally. For the purposes of abbreviation, this standard will be referred to in this documentation as the "ANSI/NIST" standard.

Due to its special formatting , this database promotes the use of the ANSI/NIST standard. Each latent and tenprint mate image is distributed in a separate ANSI/NIST file using Type-13 and Type-14 records respectively. The minutiae recorded for each fingerprint are distributed in a separate ANSI/NIST file using Type-9 records. Software utilities, written in 'C', are provided with this distribution that read, write, and manipulate data in the ANSI/NIST file format.

Section 2 of this documentation describes the content of this database in greater detail. Section 3 discusses the format and content of the distributed data files. Section 4 documents the hierarchical organization of the distribution. Finally, Section 5 provides installation and invocation instructions for the software utilities distributed with this database.

NIST Special Database 27: "Minutiae from Latents and Matching Tenprint Images" is distributed by the Standard Reference Data Group at NIST. To get pricing information and/or to order the database on CD-ROM, please contact:

Standard Reference Data Program (srdata@nist.gov)
Bldg. 820, Mail Stop 2310
National Institute of Standards and Technology
100 Bureau Drive
Gaithersburg, MD 20899-2310
(301)975-2008 (voice)
(301)926-0416 (FAX)

2. CONTENT

As described in the introduction, this database contains a collection of fingerprint images and their marked minutiae. All images and data have been stored according to the ANSI/NIST standard. This section discusses the general content of these files.

2.1 Fingerprint Images

There are 258 latent fingerprint cases in this distribution with each case containing an image of a latent and its matching tenprint mate. Each image is (800×768) pixels in size and has been scanned at 19.69 pixels per millimeter (ppmm) (500 pixels per inch (ppi)), quantized to 256 levels of gray, and stored in an uncompressed format. As can be seen in Figure 1, there are significant differences in image composition and quality between a latent image and its mate.

Each latent fingerprint image is stored in its own ANSI/NIST file as a Type-13 tagged-field image record; each tenprint mate image is stored in its own ANSI/NIST file as a Type-14 tagged field image record; and the minutiae sets for each image are stored in their own ANSI/NIST file as a Type-9 tagged-field record.[28]

2.2 Fingerprint Attributes

A set of attributes are recorded for each fingerprint image in the database. These attributes include an overall quality, the finger position, a pattern classification, the location of any cores and deltas, and the minutiae points themselves. In this section we will discuss these attributes in turn, leaving the discussion of minutiae attributes to its own subsequent section.

In fact, there are two different encodings of the fingerprint attributes distributed with this database. The majority of the fingerprint attributes in this database are represented in the ANSI/NIST Type-9 record format. In practice, this record is divided up into standard and user-defined blocks of fields. The first encoding is therefore consistent with values specified for the standard block of Fields 5 through 12, while the second encoding is consistent with the FBI/IAFIS block of Fields 13 through 23. The second block of fields is defined by the FBI's "Electronic Fingerprint Transmission Specification" (EFTS) Version 7.[29] These FBI-defined fields are required to send fingerprint transactions and receive corresponding results from the FBI/IAFIS system. This is described in more detail in Sections 2.3 and 3.4.

2.2.1 Latent Image Quality

An overall quality rating has been subjectively assigned to every latent image in the database by experienced FBI latent fingerprint examiners. Quality has been separated into three general categories of *good*, *bad*, and *ugly*. Figure 2 displays examples of "bad" and "ugly" quality latent images. These can be compared to the quality of the "good" latent shown in Figure 1. The tenprint mates are assigned the same quality as their latent counterparts even though their quality is typically much better. The quality rating is encoded in the name of the data files as described in Section 4.2. Table 1 lists the distribution of latent quality across the 258 cases.

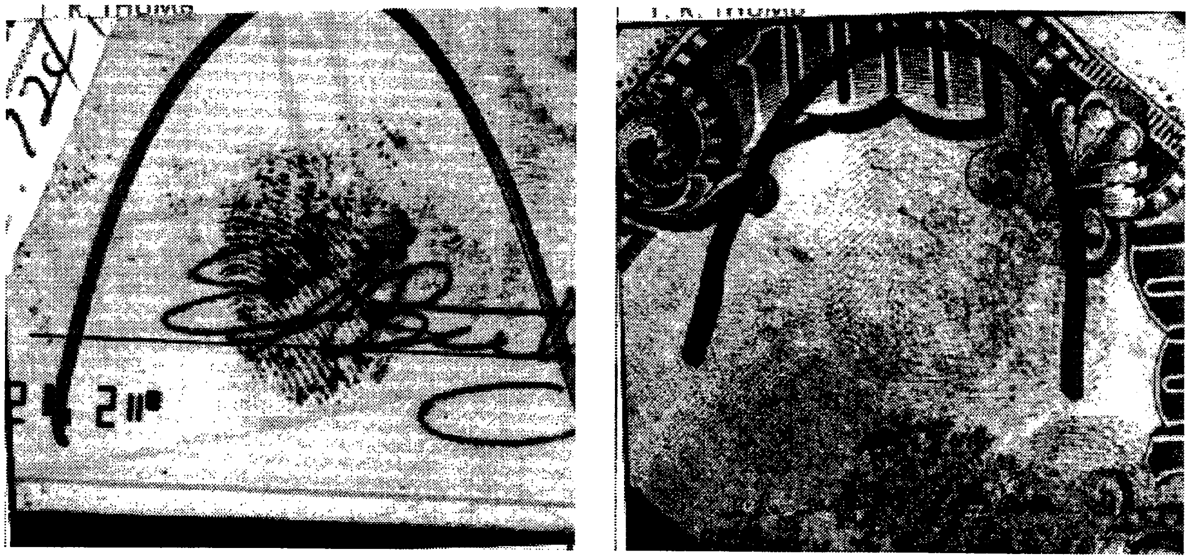

Figure 2. Example of a "bad" quality latent (left) and an "ugly" quality latent (right).

Table 1. Number of good, bad, and ugly latent cases.

Latent Quality	# of Cases
Good	88
Bad	85
Ugly	85

2.2.2 Finger Position

This attribute records the physical finger (if known) used to create the corresponding fingerprint. Finger position is recorded according to the codes listed in the second column of Table 2 as specified by the ANSI/NIST standard. For the Type-9 minutiae records in this database, finger position is recorded in Fields 6 and 14. For Type-13 and 14 image records, finger position is recorded in Field 13. Note that all latent minutiae records have been assigned a finger position of '0' because this information is typically "unknown" at a crime scene. The tenprint mate minutiae records along with all image records in this database have been assigned a physical finger position. The distribution of finger positions across the 258 latent cases is listed in the third column of Table 2.

Table 2. Finger position codes and distribution in database.

Finger Position	Finger Code	Count
Right Thumb	1	36
Right Index Finger	2	24
Right Middle Finger	3	28
Right Ring Finger	4	18
Right Little Finger	5	6
Left Thumb	6	59
Left Index Finger	7	29
Left Middle Finger	8	33
Left Ring Finger	9	21
Left Little Finger	10	4
Unknown	0	(assigned to latents)

2.2.3 Pattern Classification

The configuration of ridges within a fingerprint may be classified into three general groups of patterns. They include the category of *arch*, *loop*, and *whorl*. Examples of these pattern groups can be seen in Figure 3. In layman's terms, an arch has ridges that start at the left edge of the fingerprint and flow relatively continuously and smoothly across the image and out the right side of the image without any looping back or circular patterns. A loop has ridges of high curvature that form a loop with ridges that both enter and exit the loop from either the left or right side of the fingerprint. A whorl also has high-curvature ridges forming a more circular, concentric, or spiral pattern. For a formal definition of these groups and details on their classification, please refer to the FBI's book, "The Science of Fingerprints."[30]

These groups may be further divided into subgroups based on smaller differences existing between the patterns within each of these categories. For the purposes of this database, arches have been broken down into *plain arches* and *tented arches*, while loops have been broken down into *left slant loops* and *right slant loops.* Tented arches differ from plain arches in that the central ridges in a tented arch rise sharply and typically end in a vertical ridge or form an acute angle. The direction of a loop is determined by the side of the fingerprint image the ridges enter and leave the loop. For example, the left slant loop in Figure 3 has ridges that enter and leave the loop on the left side of the print; whereas the right slant loop has ridges entering and leaving on the right side.

For the Type-9 minutiae records in this database, pattern classification is recorded in Fields 7 and 17. Table 3 lists the pattern classifications associated with the fingerprints in the database. A separate set of codes is listed for the standard Field 7 versus the FBI/IAFIS Field 17. Note that a pattern classification of "unknown" is assigned to all latent minutiae records in the database.

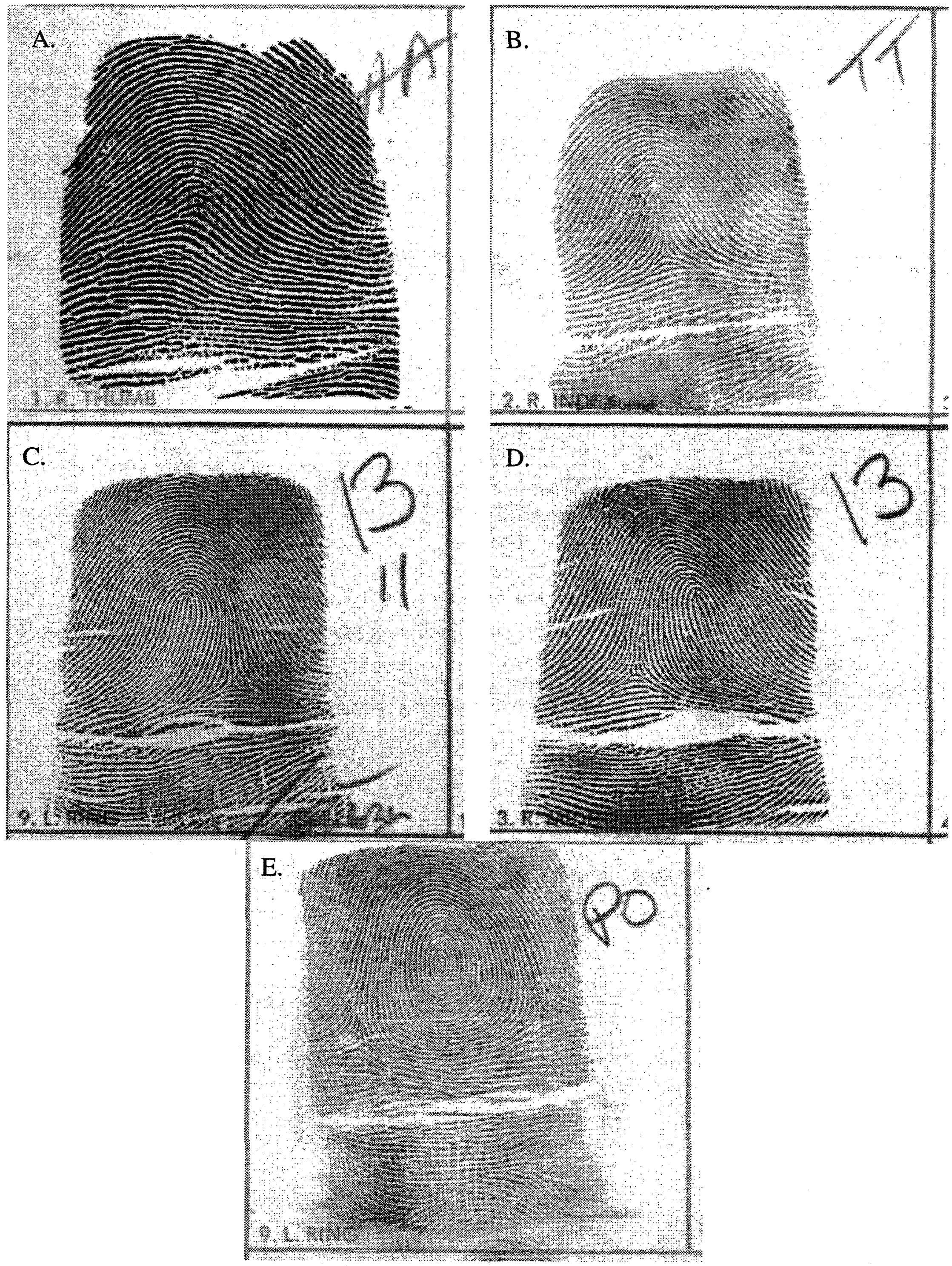

Figure 3. Pattern Class: A. plain arch, B. tented arch, C. left slant loop, D. right slant loop, and E. whorl.

Table 3. Pattern classifications and codes.

Pattern Class	Standard Codes	FBI/IAFIS Codes
Arch (type not designated)		AU
Plain Arch	PA	
Tented Arch	TA	
Right Slant Loop	RS	RS
Left Slant Loop	LS	LS
Whorl (type not designated)	WN	WU
Unknown or Unclassifiable	UN	UC

Because the configuration of ridges within a fingerprint is naturally determined by genetics, there can be ambiguity that makes it difficult to absolutely categorize some fingerprints into arbitrary groups. Therefore, this database has up to three possible pattern classifications associated with each tenprint mate in the database. Table 4 lists the distribution of pattern classifications across the 258 cases. The classes are listed according to the standard-defined codes in Table 3.

Table 4. Distribution of pattern classifications.

Pattern Classes			Counts
Class 1	Class 2	Class 3	
PA			10
TA			2
TA	RS		1
RS			54
RS	TA		6
RS	WN		1
LS			80
LS	TA		8
WN			66
WN	RS		3
WN	RS	LS	27

2.2.4 Core(s) & Delta(s)

There are two types of focal points that are used to help classify the fingerprints as loops and/or whorls. They are the *core* and *delta*. A core is essentially the center of the fingerprint, and more specifically is the center of a high-curvature region of ridges. A delta is a region in a fingerprint where ridges form a triangular configuration. There can be multiple cores and deltas existing within a fingerprint, and their count and position help categorize the pattern of a fingerprint. For a formal definition of core and delta, and a discussion of the rules on how they are used to determine the pattern class of a fingerprint, please refer to Reference [30]. The fingerprint shown in Figure 4 has its core and delta labeled.

Core attributes are stored in the Type-9 record in Fields 8 & 21. Delta attributes are stored in the Type-9 record in Fields 9 & 22. There can be up to two cores and/or two deltas stored for each fingerprint in the database. Cores and deltas have been labeled for both latent and tenprint mate images. If no core exists in the fingerprint image, then the core attribute fields are omitted from the record. The same is true for deltas.

There are approximately 270 cores and 120 deltas recorded from the latent fingerprints of the database, and approximately 340 cores and 335 deltas recorded from the tenprint mates in the database.

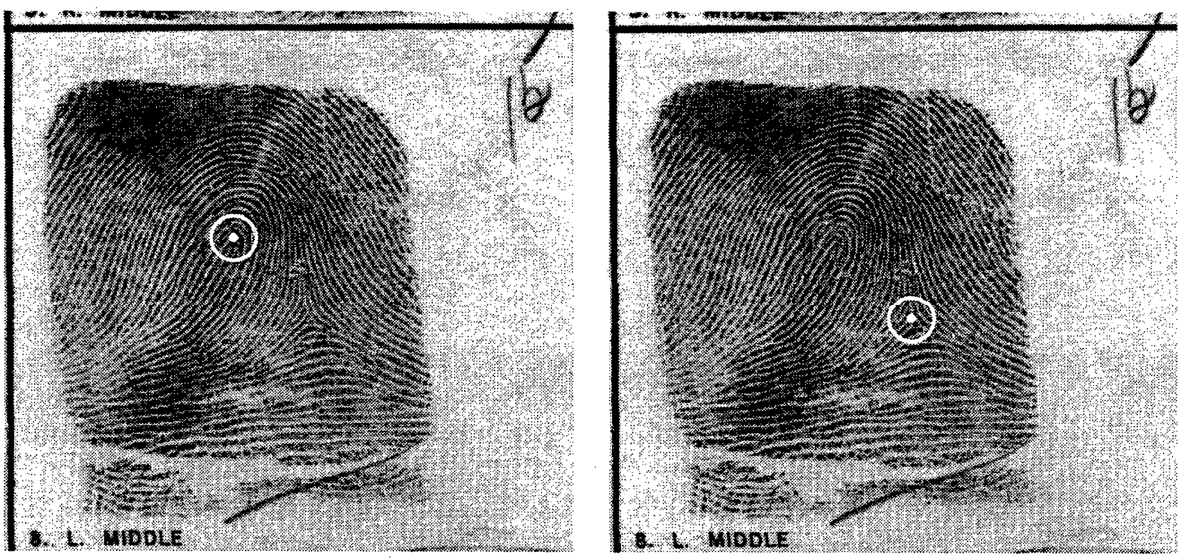

Figure 4. Example of a core (left) and delta (right).

2.3 Minutiae Attributes

The majority of the fingerprint attributes in this database are related to the minutiae points on each of the images in the distribution. This section describes these attributes in more detail.

The images in this database are all (800×768) pixels in size. Rather than search on all 614,400 pixels to determine if two fingerprints match each other, minutiae features are recorded and used for searching. These features include points in a finger's friction skin where ridges end (called a *ridge ending*) or split (called a ridge *bifurcation*). Typically, there are on the order of 100 to 200 minutiae on a tenprint, while a latent may only have a dozen. In order to search and match fingerprints, the coordinate location and the orientation of the ridge at each minutia point are recorded. Figure 5 shows an example of the two types of minutiae. The minutiae are marked in the right image, and the tails on the markers point in the direction of the minutia's orientation.

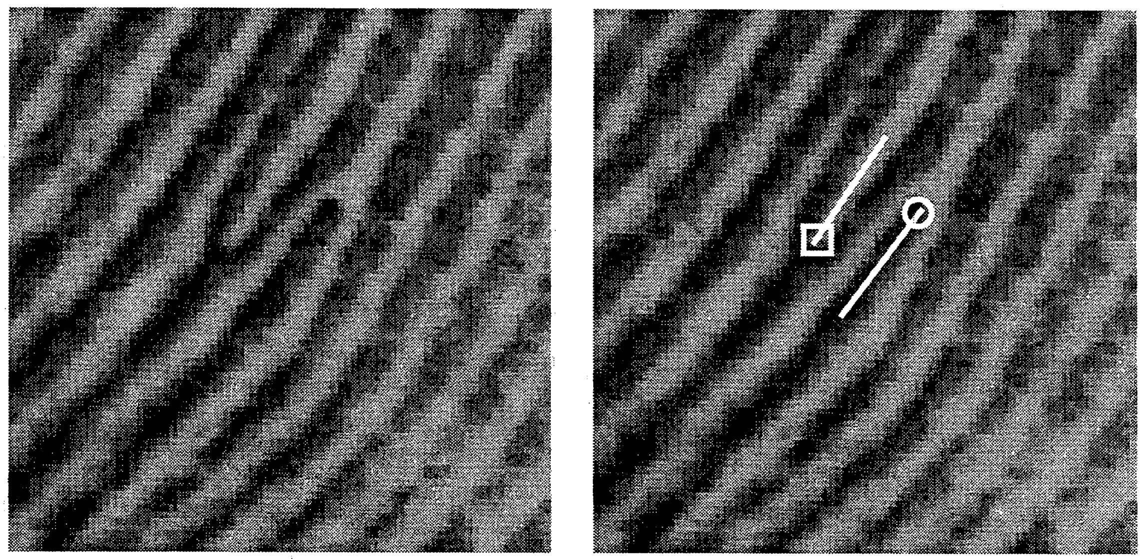

Figure 5. Minutiae: bifurcation (square marker) and ridge ending (circle marker).

For each image in the database, whether a latent or its tenprint mate, there is an *ideal* set of minutiae and a *matched* set of minutiae. The ideal set contains all those minutiae on the respective image. In the case of the latent, all the points visible in the latent image are recorded. In the case of the tenprint mate, all points visible in the tenprint image are recorded. The matched set contains only those minutiae that are in common between the latent and tenprint mate images. Unlike the ideal set, there is a one-to-one correspondence in the minutiae between the latent and the mate in the matched set. These different sets of minutiae can be used for different applications and experiments.

Each of these sets of minutiae is stored as a separate Type-9 record in its own ANSI/NIST file. Table 5 lists the number of minutiae points recorded in each of the database's minutiae sets. Notice that the counts for the matched sets are identical. Although the minutiae count for the ideal latent set should be at least as great as the matched latent set, additional minutiae were identified in the matched latent set when the tenprint and latents were visually compared.

Table 5. Number of minutiae recorded in each set.

Minutiae Set	Minutiae Count
Ideal Latent	5303
Ideal Tenprint	27426
Matched Latent	5460
Matched Tenprint	5460

In general, the minutiae points on every fingerprint in the database have been verified by at least two FBI latent examiners. The ideal minutiae on the tenprint mate were initially detected by an automatic AFIS system, and then manually validated and edited as deemed necessary by the examiners. The same option was available for generating the ideal minutiae on the latent image, but the performance of the AFIS on latent fingerprints was poor enough that the majority of latent images were entirely labeled by hand. The matched pairs of minutiae were also produced manually.

A minutia point in the database is described by its coordinate location within the fingerprint image, its ridge orientation, the type of minutiae, and a possible quality. As already mentioned, the ANSI/NIST Type-9 record is divided up into standard and user-defined blocks of fields. In all cases, Fields 1 through 4 are mandatory. Fields 5 through 12 provide a standardized means of reporting and representing minutiae attributes associated with a given image. Subsequent fields have been registered in blocks to specific AFIS vendor/user groups. For example, Fields 13 through 23 have been assigned to the FBI.[29] These blocks enable the encoding and submission of vendor-proprietary attributes that help ensure the best quality search results possible.

The minutiae points in this database have been recorded in the Type-9 records in both Fields 12 and 23. Essentially, the minutiae attributes are redundantly recorded with more detailed attributes (such as neighboring ridge counts) recorded in the FBI/IAFIS block of fields. A description of how these attributes are formatted in Fields 12 and 23 is provided in Section 3.4. The more general minutiae attributes are described below.

2.3.1 (X, Y) Location

The location of each minutiae is represented by a coordinate location within the fingerprint's image. Different AFIS systems represent this location differently. The ANSI/NIST standard and the FBI EFTS specify units of distance in terms of 0.01 millimeters (mm) from an origin. The ANSI/NIST standard specifies the origin in the bottom left of the image, while the FBI/IAFIS system specifies the origin in the top left of the image. This results in identical X-coordinates and a direct linear inversion of the Y-coordinates.

Given that all images in the database are (800x768) pixels in size and the images were scanned at 19.69 ppmm, the dimensions of each image is (40.63x39.00) mm which in standard units of 0.01 mm is

$$4063 \times 3900 \quad = \quad ((800/19.69)*100) \times ((768/19.69)*100).$$

Thus the pixel coordinate (200, 192) will be represented in standard units at

$$(1016, 2924) \quad = \quad ((200/19.69)*100)\,,\,(3900-((192/19.69)*100)-1)$$

whereas in FBI/IAFIS units the pixel coordinate is represented at

$$(1016, 975) \quad = \quad ((200/19.69)*100)\,,\,((192/19.69)*100).$$

2.3.2 Orientation

The orientation of the minutiae is represented in degrees, with zero degrees pointing horizontal and to the right, and increasing degrees proceeding counter-clockwise. The orientation of a ridge ending is determined by measuring the angle between the horizontal axis and the line starting at the minutia point and running through the middle of the ridge. The orientation of a bifurcation is determined by measuring the angle between the horizontal axis and the line starting at the minutia point and running through the middle of the intervening valley between the bifurcating ridges.

The minutiae plotted in Figure 6 illustrate the line to which the angle of orientation is measured. Each minutia symbol is comprised of a circle or square marking the location of the minutia point, and the line or tail proceeding from the circle or square is projected either along the ridge ending's ridge, or the bifurcation's valley. The angle of orientation as specified by the FBI's EFTS Version 7 is exactly 180 degrees out of phase with the angle of orientation as specified by the ANSI/NIST standard. These angles are illustrated in Figure 6 where angle 'A' represents standard minutiae orientation and angle 'B' represents the FBI/IAFIS orientation. The location and orientation of a minutia are combined and stored as a single field value in the Type-9 record as described in Section 3.4.

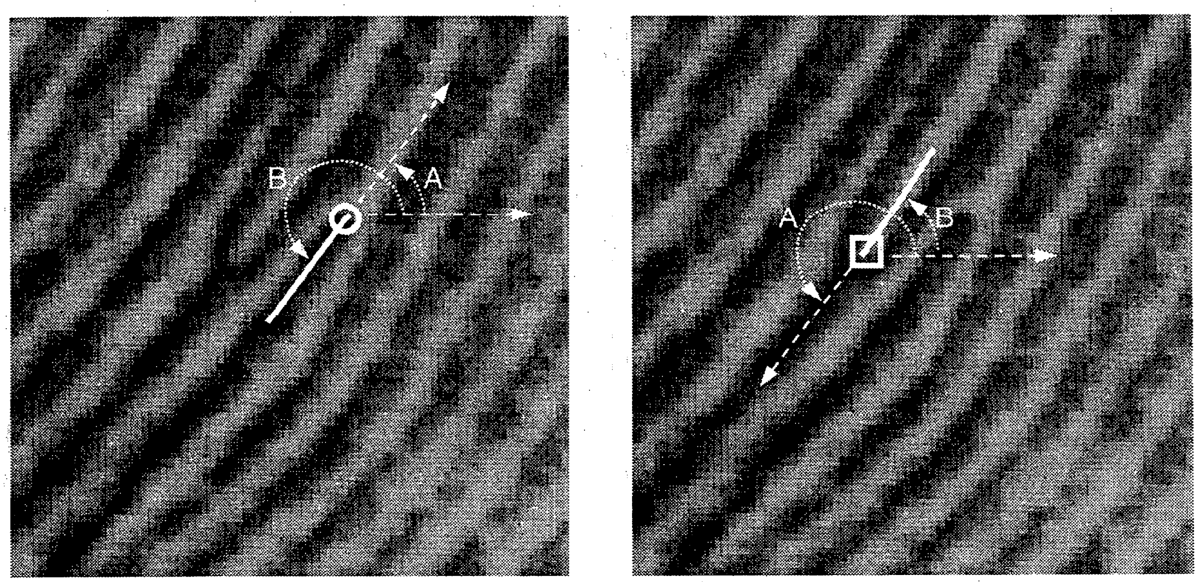

Figure 6. Minutiae orientation: A. standard angle, B. FBI/IAFIS angle.

2.3.3 Type

As shown in Figure 6, there are two general types of fingerprint minutiae, ridge endings and bifurcations. In this database, minutiae types have been assigned to only the ideal latent minutiae files. The other three sets of minutiae, the ideal tenprint, matched latent, and the tenprint minutiae files have minutiae type recorded as "undetermined". Again, the codes used for representing minutiae type are slightly different between the ANSI/NIST standard and the FBI/IAFIS as seen in Table 6.

Table 6. Minutiae type codes.

Minutiae Type	Standard Codes	FBI/IAFIS Codes
Ridge Ending	A	A
Ridge Bifurcation	B	B
Undetermined	D	C

2.3.4 Quality

The ANSI/NIST standard provides for a quality value to be recorded with each minutiae point reported in a Type-9 record. The standard permits values ranging from 0 to 63 to be recorded with a value of '2' representing the highest quality possible and quality successively decreasing to where '63' represent the lowest quality possible. Values '0' and '1' have special meaning. A value of '0' indicates the minutiae points were manually encoded; the value '1' indicates that the minutiae were automatically encoded; and in either case the quality is undetermined.

In this database, the minutiae in the matched latent and tenprint sets have been assigned a default quality value of '0' because they were manually encoded. The ideal tenprint set of minutiae has been assigned a default value of '1', because these points were derived using an automatic AFIS system. Only the ideal latent set of minutiae have been assigned measured quality values.

The quality values recorded in the ideal latent set were assigned subsequent to the placement of the minutiae on the images, and a significant period of time separated these two tasks. When the quality values were added, FBI latent examiners were asked to manually rate the quality of each minutia point and assign one of three quality levels: good, medium, or poor. Quality was to be rated based on the condition of the image in the location in which the minutia was positioned and based on how clearly identifiable the type of the minutia was in the image.

Table 7 lists the distribution of qualities across the minutiae points in the ideal latent set. Good quality has been assigned the value '6'; medium quality '11', and poor quality '16'. Notice that there are also 80 minutiae in the set with "undetermined" quality. These 80 minutiae belong to three latent cases (case numbers 047, 213, and 220) which apparently were skipped when the minutiae qualities were added to the data.

Table 7. Distribution of minutiae quality within the ideal latent set.

Minutiae Quality	Quality Value	Counts
Good	6	3429
Medium	11	1304
Poor	16	490
Undetermined	0	80

3. FORMAT

In the previous section, the general content of the database was discussed. This section describes the format in which the data has been encoded.

3.1 Overview of ANSI/NIST Standard

In 1986, NIST in cooperation with the American National Standards Institute (ANSI), the FBI, and other law enforcement agencies drafted and adopted a standard specifying a format for the electronic interchange of fingerprint images and related data including minutiae attributes.[9] Over the years, the standard has evolved to include other types of images and attributes related to mugshots, scars, and tattoos. The standard is currently under revision, and the proposed ANSI/NIST-ITL 1-2000 standard should be adopted by the summer of 2000.[28]

The standard has been updated to include tagged field records for representing images. Specifically, tagged field image records Type-13, 14, 15, and 16 have been added. Prior to this version, fingerprint images were encoded entirely within binary records (Type-3, 4, 5, 6, and 7). In support of the updated standard, the files in this database have been formatted according to these new record types. Each latent image is stored in a Type-13 record within an ANSI/NIST file. Each tenprint mate image is stored in a Type-14 record within an ANSI/NIST file.

In addition to facilitating the exchange of images, a key aspect of the ANSI/NIST standard is that it provides a textual encoding of fingerprint minutiae attributes. For transmission and processing purposes, a fingerprint can be represented by a few thousand bytes of minutiae data as opposed to several hundred thousand bytes of pixel data. This represents a significant reduction in required bandwidth and storage to the FBI and other law enforcement agencies. Minutiae attributes are stored in the ANSI/NIST Type-9 record.

3.1.1 Hierarchical Structures: File, Record, Field, Subfield, and Information Item

The ANSI/NIST standard organizes data within several defined hierarchical structures. Data is stored at the highest level within *files*. An ANSI/NIST file is comprised of one or more *records*; a record is comprised of *fields*; a field is comprised of *subfields*; and subfields are comprised of *information items*. The simplest and most common type of field contains a single value stored as an information item. However, when multiple attributes are to be associated within a field, they may be represented as a sequence of information items. When it is desired to repeat multiple instances of the same ordered list of values within a field, a sequence of information items may be assigned to a subfield, and then instances of the subfield may be repeated within the field. This last example is analogous to rows in a table. The information items are equivalent to the cells in a table, the subgroups are equivalent to the table's rows, and the table itself is the field.

3.1.2 Separator Characters

The ANSI/NIST records used in this distribution are all tagged field records, and they are entirely ASCII encoded (with the exception of the image data field as described in Section 3.3). A set of non-printable ASCII characters, referred to as separator characters, is used to delimit the data into the standard's hierarchical structures. Table 8 lists the one-byte hexadecimal values of the separator characters and associates them with the particular structure they delimit.

Table 8. ANSI/NIST separator characters.

Separator Character	Hexadecimal Value	Structure Delimited
FS	0x1C	Record
GS	0x1D	Field
RS	0x1E	Subfield
US	0x1F	Information Item

The *FS* character is used to signal the end of a record, the *GS* character is used to signal the end of a field, the *RS* character is used to signal the end of a subfield, and a *US* character is used to signal the end of an information item. If a lower-level structure ends simultaneously with a higher-level structure, then the lower-level separator character is omitted in place of the higher-level separator character. For example, a *US* character does not follow the last information item in a field, instead the *GS* character of the field will terminate the information item. If the field is the last in the record, then the *GS* character will be omitted and the field will be terminated by the record's *FS* character. Given this use of the separator characters, an effective parsing strategy is to trap for instances of separator characters and let the value of the separator dictate to which level (record, field, subfield, or information item) the *next* data value in the file is to be associated.

3.1.3 Field Tags

The Type-1, 9, 13, & 14 records contain tagged fields in which each field begins with an identifier string (or tag). The format of the tag is "*R.F*:", where *R* is the number of the record type, and *F* is the number of the field. The data for the field immediately follows the ':'. Not only does the standard allocate and define record types, but it also allocates and defines the contents of specific fields, assigning each field a specific field number which is represented by *F* in the tag. For example "1.003:", is the tag associated with record Type-1, Field 3, and this field is defined to contain the table of contents for the remainder of the ANSI/NIST file. For a complete list of fields and their definitions, please refer to the ANSI/NIST standard.[28]

3.2 Format of Record Content Tables

In this document, Tables 9 through 12 have been carefully constructed to give the reader a snapshot of which fields are populated with what values within each record of every ANSI/NIST file in the database. This section briefly documents the format of these tables.

Each complete row of the tables represents a field that is assigned a value within the given record in the database. The first column in the table lists the tag used to identify the particular field. The second column lists the field's name. Hierarchically, a field is comprised of one or more subfields. The remaining 5 columns in the table are used to represent the configuration of subfields present in the field. The third column, labeled "Subfield_i", lists the index of the current subfield in the field. Subfields are comprised of one or more information items, and columns four through seven represent the configuration of information items present in the current subfield. The fourth column, labeled "Item_i", lists the index of the current information item in the subfield; column five lists the information item's name; and column six provides information regarding the item's value and/or format.

Values listed in column six between quotation marks in non-italics font are the actual values contained in the associated information item. A value listed in italics specifies either the format of the information item or it points to a reference that documents the information item's value in more detail. If the italics value is of the form *[ref: ...]*, then the value points to a reference. The reference "Table 5" for example points to a specific table in *this* document; the reference "ANSI/NIST" points to the ANSI/NIST-ITL 1-2000 document; "EFTS V7" points to the FBI's IAFIS documentation; and a reference of "Field 9.010:" refers to the value of the specified field. A reference to an external document is used for those fields deemed less relevant to the utility of this database and whose values vary from field to field.

These tables specify several different formats. *XXXXYYYY* represents an (X, Y) coordinate pair with each coordinate zero-padded to four digits to the left and then concatenated into a single integer. For example, 12950964, represents the coordinate (1295, 964). *TTT* represents an orientation angle in degrees that is zero-padded to 3 digits to the left. *XXXXYYYYTTT* then represents a concatenation of a (X, Y) coordinate pair with an orientation angle. *N* simply represents an integer value, whereas *QIIIFPT.EEE* represents a file name in the database as described in Section 4.2.

The last column in these tables, labeled "Sep", documents the use of the standard separator characters as they delimit information items and represent changes in the data hierarchy. It should be noted that the separator characters are listed as if *every* information item in the tables is included in the file and *none* that are optional have been omitted.

3.3 Image File Format: Type-1 then Type 13 or 14 Records

The images in the database are stored in ANSI/NIST tagged field image files. The latent images are stored in files containing a Type-1 and Type-13 record, while the tenprint mate images are stored in files containing a Type-1 and Type-14 record. The formats of the Type-13 and 14 records are the same with the distinction only being that Type-13 records are reserved for latent fingerprints and Type-14 records are reserved for tenprints.

According to the standard, *every* ANSI/NIST file starts with a Type-1 record. Table 9 documents the contents of the Type-1 record stored in each latent image file within the database. The item values are prototypical for *all* Type-1 records in the distribution. The only variation will be in the value stored in the "Record Type" information item of Field 1.003.

As can be seen in Table 10, the Type-13 record contains a sequence of predefined tagged fields representing various attributes of the image including its pixel dimensions, scan resolution, and compression technique used. The last field in a tagged field image record is always assigned number "999", and it contains the image's binary pixel data. All images in this distribution are uncompressed, so the pixels in this field are in raster scan order, left-to-right, top-to-bottom. All tagged field image records are terminated with an *FS* character which also terminates the image data field. The same fields in Table 10 are used for the Type-14 record, only the field references of '13' in the table are replaced with '14'.

Table 9. Type-1 record content.

Field Tag	Field Name	Field Format				
		Subfield$_i$	Subfield Format			
			Item$_i$	Item Name	Item Value	Sep**
1.001:	Logical Record Length	1	1	Logical Record Length	*[ref: ANSI/NIST]*	*GS*
1.002:	Version Number	1	1	Version Number	"0300"	*GS*
1.003:	File Content	1	1	Type-1 Record Indicator	"1"	*US*
			2	Remaining Records	"1"	*RS*
		2	1	Record Type	"13"	*US*
			2	IDC	"00"	*GS*
1.004:	Type of Transaction	1	1	Type of Transaction	"NISTDATA"	*GS*
1.005:	Date	1	1	Date	"19990707"	*GS*
1.006:	Priority	1	1	Priority	"1"	*GS*
1.007:	Destination Agency ID	1	1	Destination Agency ID	"DAI000000"	*GS*
1.008:	Originating Agency ID	1	1	Originating Agency ID	"MDNISTVIP"	*GS*
1.009:	Transaction Control Number	1	1	Transaction Control Number	*QIIIFPT.EEE*	*GS*
1.011:	Native Scanning Resolution	1	1	Native Scanning Resolution	"19.69"	*GS*
1.012:	Nominal Transmitting Resolution	1	1	Nominal Transmitting Resolution	"19.69"	*GS*

Table 10. Type-13 record content.

Field Tag	Field Name	Field Format				
		Subfield$_i$	Subfield Format			
			Item$_i$	Item Name	Item Value	Sep
13.001:	Logical Record Length	1	1	Logical Record Length	*[ref: ANSI/NIST]*	*GS*
13.002:	Image Designation Character	1	1	Image Designation Character	*[ref: ANSI/NIST]*	*GS*
13.003:	Impression Type	1	1	Impression Type	*[ref: ANSI/NIST]*	*GS*
13.004:	Source Agency / ORI	1	1	Source Agency / ORI	"DCFBIAFIS"	*GS*
13.005:	Latent Capture Date	1	1	Latent Capture Date	"19990707"	*GS*
13.006:	Horizontal Line Length	1	1	Horizontal Line Length	"800"	*GS*
13.007:	Vertical Line Length	1	1	Vertical Line Length	"768"	*GS*
13.008:	Scale Units	1	1	Scale Units	"2"	*GS*
13.009:	Horizontal Pixel Scale	1	1	Horizontal Pixel Scale	"197"	*GS*
13.010:	Vertical Pixel Scale	1	1	Vertical Pixel Scale	"197"	*GS*
13.011:	Compression Algorithm	1	1	Compression Algorithm	"NONE"	*GS*
13.012:	Bits per Pixel	1	1	Bits per Pixel	"8"	*GS*
13.013:	Finger / Palm Position	1	1	Finger / Palm Position	*[ref: Table 2]*	*GS*
13.999:	Image Data	1	1	Image Data	*binary pixel data*	*FS*

3.4 Minutiae File Format: Type-1 then Type-9 Record

There are four sets of minutiae distributed for each of the 258 latent cases in the database. There is the pair of *ideal* latent and tenprint mate minutiae and the pair of *matched* latent and tenprint mate minutiae. Each of the minutiae sets is stored in a separate ANSI/NIST file, and each file is comprised of a Type-1 record followed by a Type-9 minutiae record. The Type-1 record is nearly identical to that shown in Table 9 except the "Record Type" information item in Field 1.003 is set to '9'.

The most valuable data in this distribution is contained in the Type-9 records of the database. It is here the minutiae attributes that have been manually scrutinized and verified by a team of professional FBI latent examiners and validated by NIST are stored. Of all the record types specified in the ANSI/NIST standard, the Type-9 remains the most controversial and its contents are still in the process of being augmented and refined. This is *not* to say that the Type-9 record is unusable. On the contrary, standard fields for minutiae and their representations have been in place since the original version of the standard in 1986.[9]

The issue surrounding these standard fields is that they represent only the common minutiae attributes used by various AFIS systems. There are a number of different vendors that have developed and are providing AFIS technology to the law enforcement community at large. Each of these systems operates on its own proprietary set of algorithms, and therefore exploit different sets of attributes computed from the same fingerprint image. It is argued that these proprietary features are required for optimum AFIS performance. To help ensure the best performance possible, the original fields of the Type-9 record have been expanded to include additional blocks of fields registered to specific AFIS vendors. These fields are strictly vendor-defined and may be used to encode vendor-proprietary features. For example, the FBI has been allocated Fields 13 through 23 for use with their IAFIS system.[29]

In light of these developments of the Type-9 record, both the ANSI/NIST standard fields and the FBI/IAFIS fields of the Type-9 records have been populated with values in this distribution. The standard fields are listed in Table 11 and the FBI/IAFIS block of fields is listed in Table 12.

Fields 9.001 through 9.004 are mandatory for any Type-9 record regardless of what subsequent fields are populated in the record. Field 9.004 has been set to 'U' in all Type-9 records in the database to signal the fact that the FBI/IAFIS user-defined fields are present in the record. The standard fields in Table 11 include attributes such as finger position, pattern classification, and the location of cores and deltas. The standard minutiae attributes include location, orientation, and type.

Comparing the standard fields in Table 11 to the FBI/IAFIS fields in Table 12, one can see that much of the attribute data is redundantly represented between the two sets of fields with more attributes and more details stored with each minutia in the FBI/IAFIS fields. For a complete description of these details, please refer to the FBI's EFTS Version 7.[29] Perhaps the most significant addition to the fields in Table 12 is the neighboring ridge count data stored with each minutia point in Field 9.023.

Table 11. Type-9 record standard fields.

Field Tag	Field Name	Field Format				
		Subfield$_i$	Subfield Format			
			Item$_i$	Item Name	Item Value	Sep**
9.001:	Logical Record Length	1	1	Record Length	*[ref: ANSI/NIST]*	*GS*
9.002:	Image Designation Character	1	1	Image Designation Character	*[ref: ANSI/NIST]*	*GS*
9.003:	Impression Type	1	1	Impression Type	*[ref: ANSI/NIST]*	*GS*
9.004:	Minutiae Format	1	1	Minutiae Format	"U"	*GS*
9.005:	Originating Fingerprint Reading System	1	1	System Name	"AFIS/FBI"	*US*
			2	Method Indicator	*[ref: ANSI/NIST]*	*GS*
9.006:	Finger Position	1	1	Finger Position	*[ref: Table 2]*	*GS*
9.007:	Fingerprint Pattern Classification(s)	1	1	Table Type	"T"	*US*
			2	Pattern Class	*[ref: Table 3]*	*RS*
		2*	1	Pattern Class	*[ref: Table 3]*	*RS*
		3*	1	Pattern Class	*[ref: Table 3]*	*GS*
9.008:*	Core(s) Position	1	1	X, Y Location	*XXXXYYYY*	*RS*
		2*	1	X, Y Location	*XXXXYYYY*	*GS*
9.009:*	Delta(s) Position	1	1	X, Y Location	*XXXXYYYY*	*RS*
		2*	1	X, Y Location	*XXXXYYYY*	*GS*
9.010:	Number of Minutiae	1	1	Number of Minutiae	*N*	*GS*
9.011:	Minutiae Ridge Count Indicator	1	1	Minutiae Ridge Count Indicator	"0"	*GS*
9.012:	Minutiae and Ridge Count Data	1	1	Index Number	"001"	*US*
			2	X, Y, and Theta Values	*XXXXYYYYTTT*	*US*
			3	Quality Measure	*[ref: Table 7]*	*US*
			4	Minutia Type Designation	*[ref: Table 6]*	*RS*
		⋮				
		N	1	Index Number	*[ref: Field 9.010:]*	*US*
			2	X, Y, and Theta Values	*XXXXYYYYTTTT*	*US*
			3	Quality Measure	*[ref: Table 7]*	*US*
			4	Minutia Type Designation	*[ref: Table 6]*	*GS*

* Field or subfield is omitted in some ANSI/NIST files in the database.

** Separator characters are listed as if all information items in the table are included in the file.

Table 12. Type-9 record FBI/IAFIS fields.

Field Tag	Field Name	Field Format				
		Subfield$_i$	Subfield Format			
			Item$_i$	Item Name	Item Value	Sep**
9.014:	Finger Number	1	1	Finger Number	*[ref: Table 2]*	*GS*
9.015:	Number of Minutiae	1	1	Number of Minutiae	*N*	*GS*
9.016:	Fingerprint Characterization Process	1	1	Equipment	"AFIS/FBI"	*US*
			2	Version Identifier	"A1"	*US*
			3	Method	"NMV"	*GS*
9.017:	AFIS/FBI Pattern Classification	1	1	Pattern Class	*[ref: Table 3]*	*US*
			2	Ridge Count 1	*[ref: EFTS V7]*	*US*
			3	Ridge Count 2	*[ref: EFTS V7]*	*RS*
		2*	1	Pattern Class	*[ref: Table 3]*	*US*
			2	Ridge Count 1	*[ref: EFTS V7]*	*US*
			3	Ridge Count 2	*[ref: EFTS V7]*	*RS*
		3*	1	Pattern Class	*[ref: Table 3]*	*US*
			2	Ridge Count 1	*[ref: EFTS V7]*	*US*
			3	Ridge Count 2	*[ref: EFTS V7]*	*GS*
9.020:	Orientation Uncertainty	1	1	Orientation Uncertainty	"0"	*GS*
9.021:*	Core Attributes	1	1	X, Y Location	*XXXXYYYY*	*US*
			2	Direction in Degrees	*TTT*	*US*
			3	Position Uncertainty	"0000"	*RS*
		2*	1	X, Y Location	*XXXXYYYY*	*US*
			2	Direction in Degrees	*TTT*	*US*
			3	Position Uncertainty	"0000"	*GS*
9.022:*	Delta Attributes	1	1	X, Y Location	*XXXXYYYY*	*US*
			2	Upward Flow	*TTT*	*US*
			3	Leftward Flow	*TTT*	*US*
			4	Rightward Flow	*TTT*	*US*
			5	Position Uncertainty	"0000"	*RS*
		2*	1	X, Y Location	*XXXXYYYY*	*US*
			2	Upward Flow	*TTT*	*US*
			3	Leftward Flow	*TTT*	*US*
			4	Rightward Flow	*TTT*	*US*
			5	Position Uncertainty	"0000"	*GS*

9.023:	Minutiae Attributes	1	1	Minutiae Index	"001"	*US*
			2	Location / Direction	*XXXXYYYYTTT****	*US*
			3	Quality Measure	*[ref: Table 7]*	*US*
			4	Minutia Type	*[ref: Table 6]*	*US*
			5	Index / Ridge Count 1	*[ref: EFTS V7]*	*US*
			6	Index / Ridge Count 2	*[ref: EFTS V7]*	*US*
			7	Index / Ridge Count 3	*[ref: EFTS V7]*	*US*
			8	Index / Ridge Count 4	*[ref: EFTS V7]*	*US*
			9	Index / Ridge Count 5	*[ref: EFTS V7]*	*US*
			10	Index / Ridge Count 6	*[ref: EFTS V7]*	*US*
			11	Index / Ridge Count 7	*[ref: EFTS V7]*	*US*
			12	Index / Ridge Count 8	*[ref: EFTS V7]*	*US*
			13	Octant Residuals	*[ref: EFTS V7]*	*RS*
		. . .				
		N	1	Minutiae Index	*[ref: Field 9.015:]*	*US*
			2	Location / Direction	*XXXXYYYYTTT****	*US*
			3	Quality Measure	*[ref: Table 7]*	*US*
			4	Minutia Type	*[ref: Table 6]*	*US*
			5	Index / Ridge Count 1	*[ref: EFTS V7]*	*US*
			6	Index / Ridge Count 2	*[ref: EFTS V7]*	*US*
			7	Index / Ridge Count 3	*[ref: EFTS V7]*	*US*
			8	Index / Ridge Count 4	*[ref: EFTS V7]*	*US*
			9	Index / Ridge Count 5	*[ref: EFTS V7]*	*US*
			10	Index / Ridge Count 6	*[ref: EFTS V7]*	*US*
			11	Index / Ridge Count 7	*[ref: EFTS V7]*	*US*
			12	Index / Ridge Count 8	*[ref: EFTS V7]*	*US*
			13	Octant Residuals	*[ref: EFTS V7]*	*FS*

* Field or subfield is omitted in some ANSI/NIST files in the database.

** Separator characters are listed as if all information items in the table are included in the file.

*** Coordinates in the FBI/IAFIS fields have origin in upper-left corner of image, where as coordinates in the standard fields have origin in bottom-left corner of image. Orientations in the FBI/IAFIS fields are 180 degrees out of phase with orientations in the standard fields.

4. ORGANIZATION

This distribution contains a combination of data, software, and documentation all of which is stored in a hierarchical collection of directories stored on CD-ROM. This section describes the organization of the distribution. ANSI/NIST data files are contained under the top-level directory **data**; documentation is provided in the top-level directory **doc**; while the remaining top-level directories (**bin**, **include**, **lib** and **src**) contain and support the software utilities in this distribution.

4.1 Directory Hierarchy

The ANSI/NIST data files are contained in the top-level distribution directory **data**. Under **data** are three subdirectories (**good**, **bad**, and **ugly**) containing 88, 85, and 85 latent cases respectively. The cases have been categorized into these three directories based on the quality of the latent image. There is a subdirectory under the quality directories for each latent case in the database. The names of these subdirectories include a unique integer case identifier ranging from 1 to 300.

Within each latent case there are six ANSI/NIST formatted files. The latent and tenprint mate images are in two separate files with extension "eft". The latent image is encapsulated in a tagged image Type-13 record, and the tenprint mate is encapsulated in a tagged image Type-14 record. Each image has been scanned at 19.69 ppmm (500 ppi), quantized to 256 levels of gray, and stored in an uncompressed format.

In addition to the two image files, there are four minutiae files corresponding to the ideal minutiae on the latent, the ideal minutiae on the tenprint mate, the matched minutiae on the latent, and the matched minutiae on the mate. All four files have the extension "lff". Each set of minutiae is stored in a tagged field Type-9 record. Figure 7 illustrates the data directory hierarchy.

```
                              data
                                |
        ------------------------+-----------------------
        |                       |                      |
       good                    bad                    ugly
        |                       |                      |
    ----+----              -----+-----            -----+-----
    |       |              |         |            |         |
  g001 ... g099          b101 ... b200          u201 ... u300
  |
  |
g001l2u.eft    (latent image)
g001t2u.eft    (tenprint mate image)
g001l2i.lff    (ideal latent minutiae)
g001l2m.lff    (matched latent minutiae)
g001t2i.lff    (ideal tenprint mate minutiae)
g001t2m.lff    (matched tenprint mate minutiae)
```

Figure 7. Data directory hierarchy.

4.2 File Nomenclature

A fixed file naming convention is used for the image and minutiae files in this distribution. An example of this convention can be seen in Figure 7 with the six filenames listed under the latent case directory **g001**. The naming convention is documented in Figure 8.

```
Format:  QIIIFPT.EEE

         Q   - quality of latent case
               Values:
                     g = good
                     b = bad
                     u = ugly

         III - unique integer latent case identifier

         F   - fingerprint type
               Values:
                     l = latent
                     t = tenprint mate

         P   - finger position
               Values:
                     1 = right thumb
                     2 = right index finger
                     3 = right middle finger
                     4 = right ring finger
                     5 = right little finger
                     6 = left thumb
                     7 = left index finger
                     8 = left middle finger
                     9 = left ring finger
                     0 = left little finger

         T   - minutiae or image type
               Values:
                     i = ideal minutiae set
                     m = matched minutiae set
                     u = uncompressed image

         EEE - file extension
               Values:
                     lff = minutiae feature file
                     eft = image file
```

Figure 8. Data file naming convention.

5. UTILITIES

This distribution includes three software utilities useful for reading, writing, and manipulating the content of the ANSI/NIST data files included in the database.

The program `nist2txt` reads an ANSI/NIST file and writes its contents back out to a new file in a format that can be viewed and changed with a text editor. The program `txt2nist` reads a formatted text file (like those produced by `nist2txt`) and writes the file's contents back out in ANSI/NIST format. With these two utilities, one can view and make changes to the contents of ANSI/NIST files with the use of a simple text editor. The third program *`ansinist`* conducts batch-oriented operations on an ANSI/NIST file. These operations include printing contents to the computer screen and deleting, substituting, and inserting contents. These operations can be performed on specific information items, subfields, fields, records, or the entire file.

5.1 Installation and Compilation

The distributed source code has been written in ANSI 'C', and compilation scripts compatible with the UNIX ***make*** utility are provided. The source code and compilation scripts have been designed and tested to work with the free, publicly available Linux operating system and GNU ***gcc*** compiler and ***gmake*** utility.[31][32] The software may also be compiled to run on computers running the family of Win32 operating systems by first installing the free, publicly available Cygwin library and associated tools.[33] The porting of the software to other operating systems, compilers, or compilation environments is the responsibility of the recipient.[1]

The software can be installed and compiled by first copying the contents of the CD-ROM to a read/writable disk partition on your computer. Copying of the top-level **doc** and **data** directories are not required to successfully compile the software. The directory to which you copy is referred to as the *installation directory*. The permissions on the copied subdirectories and the compilation scripts (named "makefile.mak") should then be changed to read/writable. Once copied and permissions changed, the software can be compiled by executing the following commands in the top-level installation directory on a Linux machine:

```
% make -f makefile.mak PROJDIR=<install_dir> depend
% make -f makefile.mak PROJDIR=<install_dir> install
```

where the text `<install_dir>` is replaced by your specific installation directory path. Alternatively, on a Win32 machine with the Cygwin library and utilities installed, type the following commands:

```
% make -f makefile.mak PROJDIR=<install_dir> EXEEXT=.exe depend
% make -f makefile.mak PROJDIR=<install_dir> EXEEXT=.exe install
```

Successful compilation will produce three utilities whose executable files are stored in the top-level **bin** directory. To invoke these utilities you can specify a full path to these files, or you may add the top-level **bin** directory to your environment's execution path.

[1] Specific software products identified in this paper were used in order to adequately support the development of the database described in this document. In no case does such identification imply recommendation or endorsement by the National Institute of Standards and Technology, nor does it imply that the equipment identified is necessarily the best available for the purpose.

5.2 User's Guide

This sections describes the functionality of the utilities provided in this distribution and gives instruction on how each of the utilities is invoked.

5.2.1 nist2txt <ansi_nist in> <fmttext out>

Parses a standard compliant ANSI/NIST-ITL 1-2000 file and writes its contents to new file in a textually viewable and editable format. Binary image fields are stored to temporary files and externally referenced in the output file.

Arguments:

```
<ansi_nist in> - the name of the ANSI/NIST file to be
                 parsed.
<fmttext out>  - the name of the resulting text file.
```

Example:

```
% nist2txt g00112i.lff g00112i.fmt
```

Converts the contents of the file (a file containing the ideal minutiae of a latent) to a textually formatted and editable file.

On error:

Upon an error, the program posts a message to standard error and exits with a non-zero status.

5.2.2 txt2nist <fmttext in> <ansi_nist out>

Parses a textually formatted version of an ANSI/NIST file and writes its contents to a new file in the standard compliant format.

Arguments:

```
<fmttext in>   - the textually formatted file to be
                 converted.
<ansi_nist out>- the name of the ANSI/NIST file be
                 created.
```

Example:

```
% txt2nist g00112i.fmt g00112i.lff
```

Converts the contents of the text file (a file containing the ideal minutiae of a latent) to an ANSI/NIST formatted file.

On error:

Upon an error, the program posts a message to standard error and exits with a non-zero status.

5.2.3 ansinist <operation>

Possible Operations:

```
-print       {all|r.[f[.s[.i]]]}           <file in> [file out]
-delete      <r[.f[.s[.i]]]>               <file in> [file out]
-substitute  <r.f.s.i>    <new value>      <file in> [file out]
-substitute  <r[.f[.s]]>  <fmttext file>   <file in> [file out]
-insert      <r.f.s.i>    <new value>      <file in> [file out]
-insert      <r[.f[.s]]>  <fmttext file>   <file in> [file out]
```

Parses a standard compliant ANSI/NIST-ITL 1-2000 file, manipulates its contents, and writes the results back out. Batch operations may be conducted at the logical record, field, subfield, or information item. Possible operations include printing, deleting, substituting, or inserting data.

Operations:

-print {all|r.[f[.s[.i]]]} <file in> [file out]

Prints the contents of the specified structure (file, record, field, subfield, or information item) to either the specified output file or to standard output.

Arguments:

all the whole file is printed. Binary image fields are stored to a temporary file name which is externally referenced in the output. This option is equivalent to running "nist2txt" on the input file.

r the record at physical position 'r' is printed.

r.f the field at the physical record position 'r' and field position 'f' is printed.

r.f.s the subfield at the physical record position 'r', field position 'f', and subfield position 's' is printed.

r.f.s.i the information item at the physical record position 'r', field position 'f', subfield position 's', and information item position 'i' is printed.

<file in> the ANSI/NIST file to be printed.

[file out] optional output file.

Operations Continued:

-delete **<r[.f[.s[.i]]]>** **<file in> [file out]**

Deletes the specified structure (record, field, subfield, or information item) from the ANSI/NIST file, writing the results either to the specified output file or to standard output.

Arguments:

r the record at physical position 'r' to be deleted.

r.f the field at the physical record position 'r' and field position 'f' to be deleted.

r.f.s the subfield at the physical record position 'r', field position 'f', and subfield position 's' to be deleted.

r.f.s.i the information item at the physical record position 'r', field position 'f', subfield position 's', and information item position 'i' to be deleted.

<file in> the ANSI/NIST file to be modified.

[file out] optional output file.

Operations Continued:

-substitute <r.f.s.i> <new value> <file in> [file out]

Substitutes the contents of the specified information item in an ANSI/NIST file with the string value provided on the command line, writing the results either to the specified output file or to standard output.

Arguments:

r.f.s.i the position indices of the information item to be substituted.

<new value> the new string value.

<file in> the ANSI/NIST file to be modified.

[file out] optional output file.

-substitute <r[.f[.s]]> <fmttext file> <file in> [file out]

Substitutes the contents of the specified structure (record, field, or subfield) in an ANSI/NIST file with the contents of a textual file consistent in format to the files produced by "nist2txt". The results are written to either the specified output file or to standard output.

Arguments:

r the record at physical position 'r' to be substituted.

r.f the field at the physical record position 'r' and field position 'f' to be substituted.

r.f.s the subfield at the physical record position 'r', field position 'f', and subfield position 's' to be substituted.

<fmttext file> file containing new contents to be substituted.

<file in> the ANSI/NIST file to be modified.

[file out] optional output file.

Operations Continued:

-insert <r.f.s.i> <new value> <file in> [file out]

Inserts an information item at the specified position into an ANSI/NIST file with the string value provided on the command line, writing the results either to the specified output file or to standard output.

Arguments:

r.f.s.i the position indices of the new information item to be inserted.

<new value> the new string value.

<file in> the ANSI/NIST file to be modified.

[file out] optional output file.

-insert <r[.f[.s]]> <fmttext file> <file in> [file out]

Inserts a structure (record, field, or subfield) into an ANSI/NIST file with the contents of a textual file consistent in format to the files produced by "nist2txt". The results are written to either the specified output file or to standard output.

Arguments:

r the record at physical position 'r' to be inserted.

r.f the field at the physical record position 'r' and field position 'f' to be inserted.

r.f.s the subfield at the physical record position 'r', field position 'f', and subfield position 's' to be inserted.

<fmttext file> file containing new contents to be inserted.

<file in> the ANSI/NIST file to be modified.

[file out] optional output file.

Examples:

```
% ansinist -print 1.3 g001l2i.lff
```

Prints the first record, third field's contents (the CNT field) to standard output.

```
% ansinist -substitute 1.4.1.1 LFFS g001l2i.lff new.lff
```

Replaces the first record, fourth field, first subfield, first information item's value (the TOT) with "LFFS" and writes the new file to "new.lff".

```
% ansinist -delete 2 g001l2i.lff new.lff
```

Deletes the entire second record (the Type-9) in the input file and writes the results to the file "new.lff".

On error:

Upon an error, the program posts a message to standard error and exits with a non-zero status.

6. REFERENCES

[1] J.H. Wegstein, "A Semi-automated Single Fingerprint Identification System", NBS Technical Note 481, April 1969.

[2] J.H. Wegstein, "Automated Fingerprint Identification", NBS Technical Note 538, August 1970.

[3] J.H. Wegstein, "Manual and Computerized Footprint Identification", NBS Technical Note 712, February 1972.

[4] R.T. Moore, "The Influence of Ink on The Quality of Fingerprint Impressions", NBS Technical Report NBSIR 74-627, December 1974.

[5] J.H. Wegstein, "The M40 Fingerprint Matcher", NBS Technical Note 878, July 1975.

[6] J.H. Wegstein, and J.F. Rafferty, "The LX39 latent Fingerprint Matcher", NBS Special Publication 500-36, August 1978.

[7] R.T. Moore, "Results of Fingerprint Image Quality Experiments", NBS Technical Report NBSIR 81-2298, June 1981.

[8] J.H. Wegstein, "An Automated Fingerprint Identification System", NBS Special Publication 500-89, February 1982.

[9] R.M. McCabe, and R.T. Moore, "Data Format for Information Interchange", American National Standard ANSI/NBS-ICST 1-1986, August 1986.

[10] R.T. Moore, "Automated Fingerprint Identification Systems - Benchmark Test of Relative Performance", American National Standard ANSI/IAI 1-1988, February 1988.

[11] R.T. Moore, "Automated Fingerprint Identification Systems – Glossary of Terms and Acronyms", American National Standard ANSI/IAI 2-1988, July 1988.

[12] R.T. Moore, R.M. McCabe, and R.A. Wilkinson, "AFRS Performance Evaluation Tests", NBS Technical Report NBSIR 88-3831, August 1988.

[13] "Minimum Image Quality Requirements for Live Scan, Electronically Produced Fingerprint Cards", Technical Report for the Federal Bureau of Investigation – Identification Division, November 1988.

[14] C. Watson, "*NIST Special Database 4*: 8-bit Gray Scale Images of Fingerprint Image Groups," CD-ROM & documentation, March 1992.

[15] C.L. Wilson, G.T. Candela, P.J. Grother, C.I. Watson, and R.A. Wilkinson, "Massively Parallel Neural Network Fingerprint Classification System," Technical Report NISTIR 4880, July 1992.

[16] R. McCabe, C. Wilson, and D. Grubb, , "Research Considerations Regarding FBI-IAFIS Tasks & Requirements", NIST Technical Report NISTIR 4892, July 1992.

[17] G.T. Candela and R. Chellappa, "Comparative Performance of Classification Methods for Fingerprints," Technical Report NISTIR 5163, April 1993.

[18] C. Watson, "*NIST Special Database 9*: 8-Bit Gray Scale Images of Mated Fingerprint Card Pairs," Vol. 1-5, CD-ROM & documentation, May 1993.

[19] C. Watson, "*NIST Special Database 10*: Supplemental Fingerprint Card Data (SFCD) for NIST Special Database 9," CD-ROM & documentation, June 1993.

[20] C. Watson, "*NIST Special Database 14*: Mated Fingerprint Card Pairs 2," CD-ROM & documentation, September 1993.

[21] R.M. McCabe, "Data Format for the Interchange of Fingerprint Information", American National Standard ANSI/NIST-CSL 1-1993, November 1993.

[22] J.L. Blue, G.T. Candela, P.J. Grother, R. Chellappa, C.L. Wilson, "Evaluation of Pattern Classifiers for Fingerprint and OCR Application," in Pattern Recognition, 27, pp. 485-501, 1994.

[23] C.L. Wilson, G.T. Candela, C.I. Watson, " Neural Network Fingerprint Classification," in Journal for Artificial Neural Networks, 1(2), 203-228, 1994.

[24] C.I. Watson, J. Candela, P. Grother, "Comparison of FFT Fingerprint Filtering Methods for Neural Network Classification," Technical Report NISTIR 5493 September 1994.

[25] G.T. Candela, P.J. Grother, C.I. Watson, R.A. Wilkinson, C.L. Wilson, "PCASYS - A Pattern-level Classification Automation System for Fingerprints," Technical Report NISTIR 5647 & CD-ROM, April 1995.

[26] R.M. McCabe, "Data Format for the Interchange of Fingerprint, Facial & SMT Information," American National Standard ANSI/NIST-ITL 1a-1997, April 1997.

[27] C. Watson, "*NIST Special Database 24*: Digital Video of Live-Scan Fingerprint Data," CD-ROM & documentation, July 1998.

[28] R.M. McCabe, "Data Format for the Interchange of Fingerprint, Facial, Scar Mark & Tattoo (SMT) Information," American National Standard ANSI/NIST-ITL 1-2000, July 2000. Available from R.M. McCabe at NIST, 100 Bureau Drive, Stop 8940, Gaithersburg, MD 20899-8940.

[29] "Electronic Fingerprint Transmission Specification," CJIS-RS-0010 (V7). Available from Criminal Justice Information Services Division, Federal Bureau of Investigation, 935 Pennsylvania Avenue, NW, Washington D.C. 20535.

[30] "The Science of Fingerprints," Rev. 12-84, U.S. Department of Justice, Federal Bureau of Investigation. Available from U.S. Government Printing Office, Washington D.C. 20402.

[31] Linux - a freely available clone of the UNIX operating system. Learn more at http://www.linux.org.

[32] GNU project - free UNIX-like utilities. Learn more at http://www.gnu.org.

[33] Cygwin tools - free GNU utility port for Win32 machines. Learn more at http://sourceware.cygnus.com/cygwin/.