MCM2004 A

By Guan Li, Zhou Enlu, Chen Zhimin @ Zhejiang Unveirstiy

Problem： *It is a commonplace belief that the thumbprint of every human who has ever lived is different. Develop and analyze a model that will allow you to assess the probability that this is true. Compare the odds (that you found in this problem) of misidentification by fingerprint evidence against the odds of misidentification by DNA evidence.*

Summary

We approach this problem with three models: model of fingerprint uniqueness, model of fingerprint identification and model of DNA identification.

Model of fingerprint uniqueness assesses the probability in theory that fingerprints are unique. We establish a relationship between this probability (P) and the probability (P_2) that two arbitrary fingerprints are matched. Then we derive an equation to assess P_2 by quantifying the information of fingerprints using minutiae features. After estimating the parameters in the equation, we get a numerical result: The probability that "Thumbprint of every human who has ever lived is different" is about $1-1.13\times10^{-49}$.

While P_2 sets the lower bound on probability of misidentification, we further develop the second model to examine the odds of misidentification in applications. Misidentification has two types: Falsely match two different fingerprints (FM) and falsely not match two impressions of the same finger (FNM). The model reveals the relationship between the two odds and finally gives a strategy to balance them in different situations. The strategy also can help us to set parameters in a specific fingerprint identification case.

In the model of DNA identification, we also examine the odds of FM and FNM based on the principle of a current DNA identification technique. Then from the two aspects, we compare DNA identification with fingerprint identification and reach to the conclusion: The odds of fingerprint misidentification are greater than DNA.

Table of Contents

Are Fingerprints Unique?

Assumptions

- **Finger types, namely thumb, index finger, middle finger, ring finger and little finger, do not determine or affect the uniqueness of prints on them.** So we analysis the uniqueness of fingerprints as a whole instead of just thumbprints.

- **Once a fingerprint is formed, its characteristics do not change with time.** This assumption ensures that one fingerprint only has one real pattern. Actually, it is said that even before a fetus is born, his fingerprints are completely determined and will never change throughout his life. The validity of this assumption is based on the anatomy and morphogenesis of friction ridge skin [1].

Definitions and Terms

About Fingerprint

- **Fingerprint:** The real pattern on a finger;
- **Impression:** An image obtained by impressing a finger against paper or sensors are only exterior mapping of the fingerprint, and may distort and lose information.

About Fingerprint Minutiae [2]

- **Ridge:** The lines that flow in various patterns across fingerprints;
- **Valley:** The space between ridges;
- **Ridge ending:** A feature where a ridge terminates;
- **Ridge bifurcation:** A feature where a single ridge splits from a single path to two paths at a Y-junction.

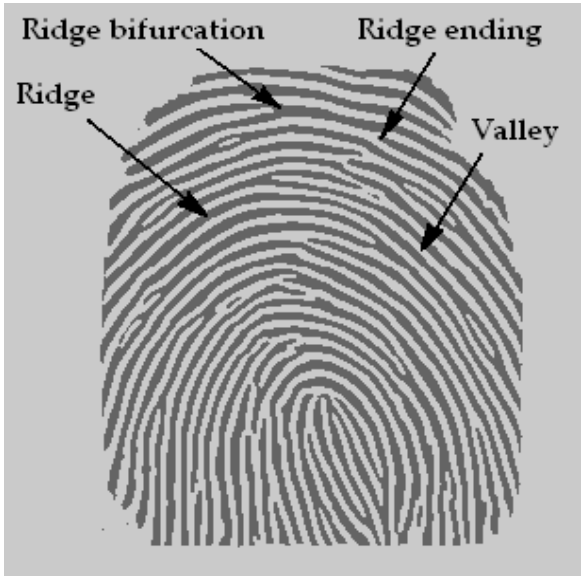

Figure 1. Illustration of ridges, valleys, ridge endings, ridge bifurcations

About Fingerprint Identification

- **False Non Match Rate (FNMR):** The probability that different impressions of the same finger are not matched.
- **False Match Rate (FMR):** The probability that impressions of different fingers are falsely matched.

Symbols

P	The probability that "the thumbprint of every human who has ever lived is different" is true.
P_2	The probability that two arbitrary fingerprint are matched.
N	The total number of persons who have ever lived.
r	Side length of the grids added on the fingerprint.
M	Total number of grids in the effective area of a fingerprint.
m	Total number of minutiae in the first fingerprint.
n	Total number of minutiae in the second fingerprint.
k	Number of minutiae that falls in the correspondent grids of two fingerprints.
p_θ	Probability that the two minutiae are matched in direction.
p_t	Probability that the two minutiae matched in type.
p	Product of p_θ and p_t.
q	Number of minutiae that are matched between two fingerprints.

Problem Analysis

We divide the problem into three subtasks:

1. **Estimation of the theoretical probability that fingerprints are unique.**

   The theoretical probability of fingerprint uniqueness can be obtained from the probability that the fingerprints of two different fingers are identical. To accomplish this, we must first find out what features characterize a fingerprint and then determine how to match these features. If the features are sufficiently similar, the two fingerprints can be identified as the same.

2. **Estimation of the practical probability of fingerprint misidentification.**

   The above theoretical probability sets an upper bound on the practical probability of fingerprints identification. Take into consideration of the general problems encountered in practical identification systems, such as image quality, different impressions of the same finger, etc. We then estimate the practical odds of misidentification by fingerprint evidence.

3. **Estimation of the probability of DNA misidentification.**

   To derive the odds of misidentification by DNA, we should first find out the principle of current technologies and estimate the probability based on the principle. The results are to be compared with the results of the second subtask.

Model of Fingerprint Uniqueness

A fingerprint can be defined as unique if there are no other fingerprints matched with it. Suppose P_2 is the probability that two arbitrary fingerprints are matched, N is the total number of persons who have ever lived.

Suppose there are $S(S \geq N)$ fingerprint types in all. Then the possibility that N fingerprints are mutually different is

$$P = \frac{N!\binom{S}{N}}{S^N}.$$

We rewrite this expression as:

$$P = \frac{N!\binom{S}{N}}{S^N} = \frac{S(S-1)(S-2)\cdots(S-N+1)}{SS\cdots S} = 1(1-\frac{1}{S})(1-\frac{2}{S})\cdots(1-\frac{N-1}{S})$$

The probability that two arbitrary fingerprints are of the same type is $\frac{1}{S}$, that means:

$$P_2 = \frac{1}{S}$$

Substituting that into the above equation, we get:

$$P = 1(1-P_2)(1-2P_2)\cdots(1-(N-1)P_2)$$

If $S \leq N$, there must be two fingerprints are of the same type. So the relationship between P and P_2 is expressed as:

$$P = \max\{(1-P_2)(1-2P_2)\cdots(1-(N-1)P_2), 0\} \qquad (1)$$

Therefore, in order to assess the probability that all fingerprints are different, we only need to calculate the probability that arbitrary two fingerprints are matched. To calculate P_2, we first extract the features that characterize a fingerprint and then match these features between any two fingerprints. The degree of feature matching determines the probability that two fingerprints are matched.

Feature Extraction

The features in a fingerprint are generally categorized into minutia features (such as ridge ending and bifurcation) and global patterns (such as the flow of the ridges) [2]. We mainly consider the minutia features because of two reasons: (a) Minutiae are independent with each other, while lines or patterns are highly related. So, minutiae features are more proper for calculating probability. (b) Most of the current identification technologies are based on the minutiae features. Our model will have the value for evaluating a large number of existing technologies.

Among all the minutiae types, we only consider the two most basic types: ridge endings and ridge bifurcations. Other common minutiae types, such as dots, empty cells, islands, bridges, enclosures, etc. can be considered as the combination of the two [4]. In addition, all the other minutiae types rarely occur.

We assume ridge endings and bifurcations are uniformly distributed in a fingerprint and they are not very close to each other. This assumption approximates the slightly over-dispersed uniform distribution found by Stony [5][3].

To quantify the minutiae features, we grid the fingerprint with numbered small squares of side length r. r is small enough so that each square contains at most one minutia. The direction of the minutia is defined as the direction of the tangent at this point. Furthermore, the minutia is either a ridge ending or a bifurcation, totally two types. Thus, each minutia is determined by a set of the **attributes**:

{*grid number, direction, type*}

Matching Process

Since a minutia is determined by its attributes, matching the minutia is to match its three attributes. Noting that the three attributes are independent, we can separately calculate the possibility of matching one attribute.

After casting grids on two fingerprints, namely, A and B, we only consider the "effective area", as shown in Figure 2.

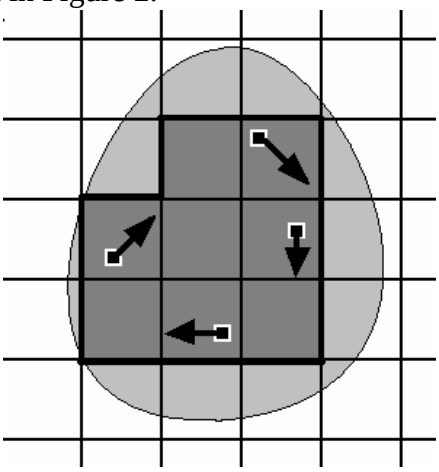

Figure 2. Grids on a fingerprint. We only consider the grids fully covered with fingerprint, denoted by deep gray squares. All deep gray grids form the "effective area". There are four minutiae in the effective area, and their directions are denoted with an arrow respectively.

The effective area should contain only integral grids and contain the same number and configuration of grids in the two fingerprints. Suppose that the total number of grids is M, fingerprint A has m minutiae and fingerprint B has n minutiae in effective area. M is a function of r, but we simply use M to denote $M(r)$.

(1) Matching in grid number

If one minutia of A and one minutia of B are in the same grid, they are said to be matched in grid number. Consider the probability that exactly k $(k \le \min(m,n))$ minutiae are matched in grid number. The occurrence that k minutiae of B are in the m minutiae-grids of A is $\binom{m}{k}$. The occurrence that the other $n-k$ minutiae of B are in the $M-m$ non-minutiae-grids of A is $\binom{M-m}{n-k}$. The occurrence that all the n minutiae of B are distributed in the M grids is $\binom{M}{n}$. Thus, the probability that exactly k minutiae of A and B are matched in grid number is given by:

$$P(M,m,n,k) = \frac{\binom{m}{k}\binom{M-m}{n-k}}{\binom{M}{n}} \qquad (2)$$

(2) Matching in direction

If one minutia of A and one minutia of B are in the same angle range, they are said to be matched in direction. We divide the space of 360° into eight angle ranges, so the probability that two minutiae matched in direction is given by:

$$P_\theta = \frac{1}{8}$$

(3) Matching in type

We assume the occurrences of ridge endings and ridge bifurcations are the same. This assumption is based on the observation of the complementary image of a fingerprint. We discover that endings in the original fingerprint become bifurcations in the complementary image, and likewise, bifurcations in original fingerprint become endings in the complementary image, as shown in Figure 3. The complementary image appears to be a fingerprint too. Therefore we estimate the probability of type matching is:

$$P_t = \frac{1}{2}$$

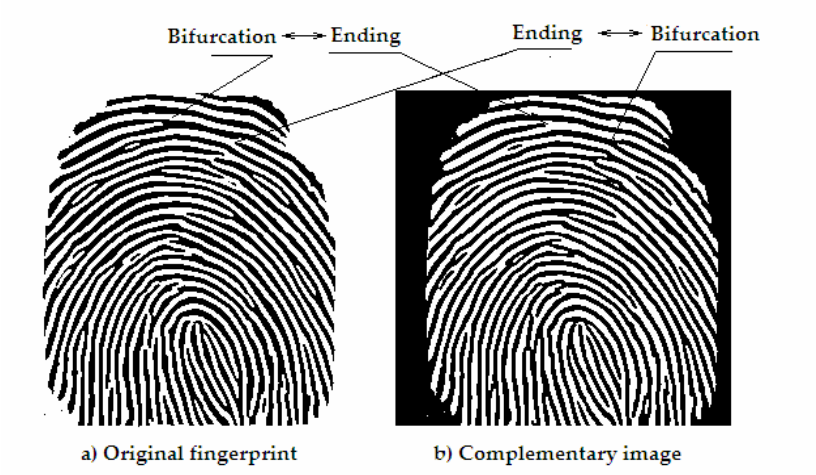

Figure 3. Original fingerprint and its complementary image, which looks also like a fingerprint.

(4) Matching of minutiae

We have assumed that the three attributes are independent. So the probability that two minutiae are matched in both direction and type is the product of the two separate probabilities, as below:

$$p = P_\alpha \times P_t = \frac{1}{8} \times \frac{1}{2} = \frac{1}{16}$$

Equation (2) gives the probability that k minutiae are matched in grid number. Among these minutiae, suppose there are exactly $q\,(q \le k)$ minutiae matched in both direction and type. The probability that q minutiae match and $k-q$ minutiae fail to match is

$$\underbrace{p \times p \cdots p}_{q} \times \underbrace{(1-p)\cdots(1-p)}_{k-q} = p^q (1-p)^{k-q}$$

There are $\binom{k}{q}$ ways to choose the q minutiae from the k minutiae, so the probability that q minutiae among k minutiae are matched:

$$\binom{k}{q} p^k (1-p)^{k-q}$$

Varying k from q to $\min(m,n)$, we add the probabilities that exactly q minutiae are matched among the k minutiae matched in grid number. Thus we get the probability that exactly q minutiae in two fingerprints are matched, as below:

$$P_1(M,m,n,q) = \sum_{k=q}^{\min(m,n)} \left(\frac{\binom{m}{k}\binom{M-m}{n-k}}{\binom{M}{n}} \times \binom{k}{q} p^q (1-p)^{k-q} \right)$$

Further, the probability that at least q minutiae in two fingerprints are matched is the sum of probabilities that $q, q+1, \cdots, \min(m,n)$ minutiae are matched. It is expressed as:

$$P_2(M,m,n,q) = \sum_{j=q}^{\min(m,n)} \sum_{k=j}^{\min(m,n)} \left(\frac{\binom{m}{k}\binom{M-m}{n-k}}{\binom{M}{n}} \times \binom{k}{j} p^j (1-p)^{k-j} \right) \qquad (3)$$

Calculation of P

In the equation (3), parameters M, m, n, q are to be determined.

For a complete match, we hope q to be as large as possible. Because more minutiae are matched, more likely two fingerprints are identical. On the other hand, q cannot exceed the total number of minutiae in the fingerprint. So we set q to $\min(m,n)$.

The total number of minutiae in a fingerprint conforms to normal distribution with its mean at 36 [3]. So we set m and n both to 36. Consequently, $\min(m,n)$ and q both equal to 36.

The value of M is dependent on the size of the grids. As stated above, each grid should be small enough to ensure at most one minutia in it. The smaller the grid, the more precise minutiae position is. However, when grids are too small, minutiae become indistinguishable in them. So the grid size has a lower limit. For example, to identify a ridge bifurcation in one grid, the grid should contain at least two ridges and two valleys. A fingerprint usually has about 20 ridges and valleys respectively. As the result, a rough estimation of M is $\frac{20 \times 20}{2 \times 2} = 100$.

Substituting the values into equation (3), we get the probability that two arbitrary fingerprints are matched:

$$P_2 = \frac{\binom{m}{q}\binom{M-m}{n-q}}{\binom{M}{n}} \times p^q = \frac{\binom{36}{36}\binom{100-36}{36-36}}{\binom{100}{36}} \times \left(\frac{1}{16}\right)^{36} \approx 2.27 \times 10^{-71}$$

The expression of $(1-P_2)(1-2P_2)\cdots(1-(N-1)P_2)$ can be expanded as:

$$1 - \left(\sum_{1 \le i_1 \le N-1} i_1\right) P_2 + \left(\sum_{1 \le i_1 < i_2 \le N-1} i_1 i_2\right) P_2^2 - \cdots + (-1)^{N-1} \left(\sum_{1 \le i_1 < i_2 < \cdots < i_{N-1} \le N-1} i_1 i_2 \cdots i_{N-1}\right) P_2^{N-1}$$

All the terms after the second term are exponentially smaller than the second term, so they can be neglected. Equation (1) is reduced to:

$$P \approx \max\{1 - \frac{(N-1)N}{2} P_2, 0\}$$

The total number of persons who ever lived is on the order of 10^{11} [6]. Substituting the values into the above expression, we get $1 - 1.13 \times 10^{-49}$.

According to our first assumption on Page 1, the probability of a thumbprint equals that of a fingerprint. Thus we arrive at the conclusion: **the probability that "Thumbprint of every human who has ever lived is unique" is about** $1-1.13\times10^{-49}$**.**

Model of Fingerprint Identification

From the model of fingerprint uniqueness, we find that theoretically, it is almost true that every person's fingerprint is unique. That is the theoretical basis for fingerprint identification in many application areas. While the probability sets a limit on the performance of practical fingerprint identification, we still want to investigate into the practical misidentification probability. We consider two most typical applications: a) Criminal fingerprint matching. A fingerprint of unknown ownership is matched against a database of known fingerprints. b) Authentication fingerprint matching. Access is granted only if the applicant's fingerprint matches the stored fingerprint.

The main difference between the two applications is that criminal matching puts more weight on corresponding two fingerprints, while authentication emphasizes more on distinguishing two fingerprints. In other words, criminal matching should not escape any possible match between two fingerprints. Authentication system should ensure its safety by excluding any possible different fingerprint; even at the price of inconvenience (e.g. Sometimes a person has to impress the finger several times in order to have one match.)

To quantitatively analyze the different requirements of the two applications, we introduce two measures:

- **FMR:** False Match Rate. The probability that impressions of different fingers are falsely matched.
- **FNMR:** False Non Match Rate. The probability that different impressions of the same finger are not matched.

FMR and FNMR are the two aspects of misidentification.

To tolerate error in measurements, we give the criterion that two fingerprints are matched:

- **Criterion of matching:** We set the parameter q as the threshold. If more than q minutiae are matched, then the two fingerprints are judged as the same. Otherwise, they are judged as different.

Expression of FMR and FNMR

Equation (3) not only enables us to calculate the probability that every fingerprint is unique, but also reveals how several factors (such as grid size, number of minutiae) influence the probability that two different fingerprints are falsely matched. According to the criterion of matching, FMR is the probability that at least q minutiae are matched. So its expression can be derived from equation (3) with small changes:

$$FMR(M,m,n,q)=\sum_{j=q}^{\min(m,n)}\sum_{k=j}^{\min(m,n)}\left(\frac{\binom{m}{k}\binom{M-m}{n-k}}{\binom{M}{n}}\times\binom{k}{j}p^{j}(1-p)^{k-j}\right) \qquad (4)$$

Now consider how to express FNMR. Suppose that correspondent minutiae refer to the same minutiae in two impressions of the same finger. Using fingerprint database [7], we can get the distribution of distance along x and y axes between the correspondent minutiae (detailed in Appendix I). The distribution of distance along x axis is fitted to the normal distribution, with mean at $\mu_x = 0$ and variance of $\sigma_x = 2.748$. The distribution along y axis is fitted to the normal distribution, with mean at $\mu_y = 0$ and variance of $\sigma_y = 2.8756$. The distance is in the unit of image pixels. Figure 4 shows the fitting of distribution.

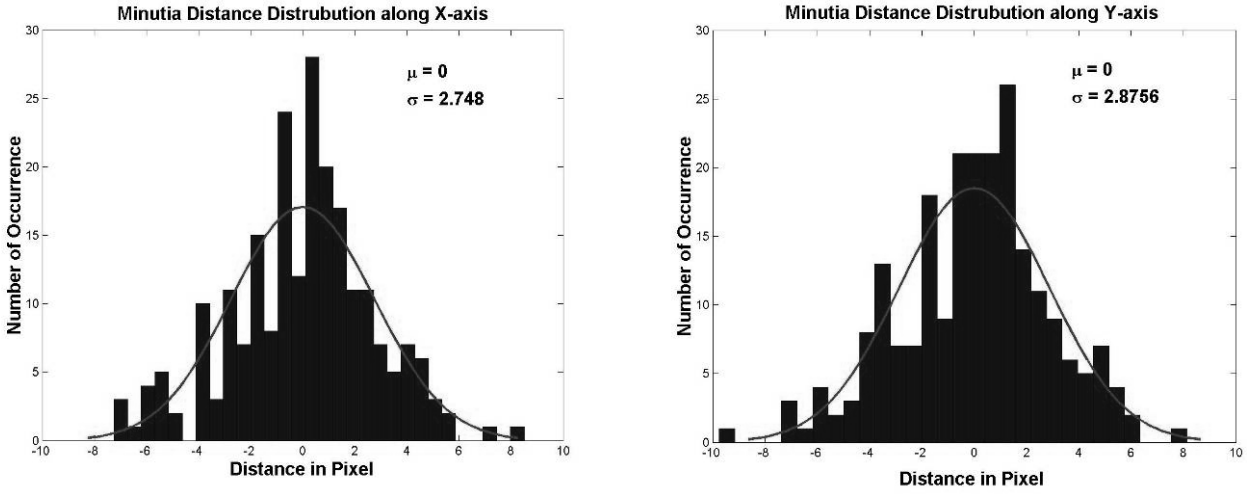

Figure 4. Minutiae displacement (between to impressions) distribution along x and y axis direction respectively. They are then both fitted with normal distribution. The mean value and variance are written on the up-right corner.

Thus, if the side length of grid is given, the probability that correspondent minutiae are matched is determined. Suppose p_x is the probability that x distance of correspondent minutiae is within the range of r, p_y is the probability that y distance of correspondent minutiae is within the same range. Because the two distributions are independent, the probability that correspondent minutiae fall in the same grid is $p_x(r)p_y(r)$. The difference in angles of correspondent minutiae conform to the normal distribution with mean at $\mu_\theta = 0^\circ$ and variance of $\sigma_\theta = 10^\circ$. By calculation we obtain the probability that correspondent minutiae fall in the same angle range is $p_\theta = 0.975$. Therefore, the probability that correspondent minutiae are matched:

$$p_m(r) = p_x(r)p_y(r)p_\theta$$

The probability that at least q pairs of correspondent minutiae are matched:

$$\sum_{i=q}^{\min(m,n)}\left(\binom{\min(m,n)}{i}p_m{}^{i}(1-p_m)^{\min(m,n)-i}\right)$$

$FNMR$ is the probability that less than q pairs of correspondent minutiae are matched. So it is expressed as below:

$$FNMR(r,m,n,q) = 1 - \sum_{i=q}^{\min(m,n)} \left(\binom{\min(m,n)}{i} p_m{}^i (1-p_m)^{\min(m,n)-i} \right) \qquad (5)$$

Parameter Analysis

Expressions of FMR and $FNMR$ have four parameters, namely M, m, n and q. Since M is a function with respect to $r/2\sigma$, where σ is obtained from the distance distribution, FMR and $FNMR$ are ultimately determined by r, m, n, q. Parameters m and n conform to a normal distribution with their mean values at 36 [3]. So we fix m =36, n =36, and analyze the change of FMR and $FNMR$ with respect to q and r.

(1) Change of *q*

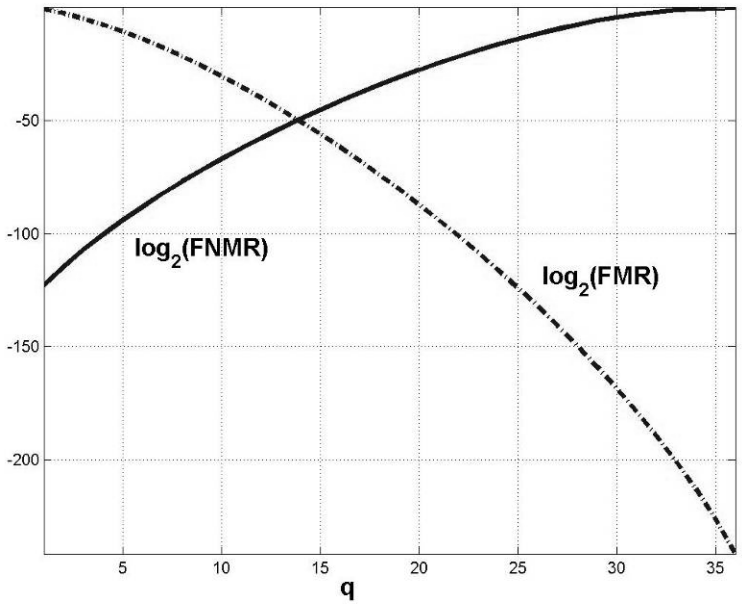

Figure 5. Fix r and examine the change of *FMR* and *FNMR* with respect to *q*. Plotted logarithmically. *FMR* decreases and *FNMR* increases as q increases.

Firstly, we fix r and examine the change of FMR and *FNMR* with respect to q. Figure 5 shows the logarithm values of FMR and *FNMR* when $r/2\sigma = 2.1$. According to the criterion of matching, if more than *q* minutiae are matched, then the two fingerprints are judged as the same. So as q increases, more minutiae need to be matched in order to declare two impressions are from the same finger. The probability of mismatching two impressions from different fingers decreases. That explains why the value of FMR decreases as q increases. If two impressions are from the same finger, not all pairs of correspondent minutiae can be matched due to elastic,

rotational and translational variances of the fingerprint and extra noise in images. So as q increases, the probability that they fail to have enough minutiae matched increases. That explains why *FNMR* increases as q increases.

(2) Change of *r*

Next, we fix q and examine the change of *FMR* and *FNMR* with respect to $r/2\sigma$. Figure 6 shows the logarithm values of *FMR* and *FNMR* when q=12. r is the side length of grids in unit of pixel. According to Appendix I, the distance of correspondent minutiae conforms to normal distribution with mean at 0 and variance of 2.81. For a given *r*, we can calculate the probability that correspondent minutiae fall in the same grid, according to

$$p(r) = 2\int_0^{r/2\sigma} \frac{1}{\sqrt{2\pi}} e^{-u^2/2} du \,.$$

If r increases, the grid enlarges and $p(r)$ increases accordingly. So it is more likely to match two impressions of the same finger. That explains why *FNMR* decreases as *r* increases. However, larger grid size also leads to higher probability that minutiae of different fingerprints fall in the same grid and thus the probability of mismatching the two fingerprints. That explains why *FMR* increases as *r* increases.

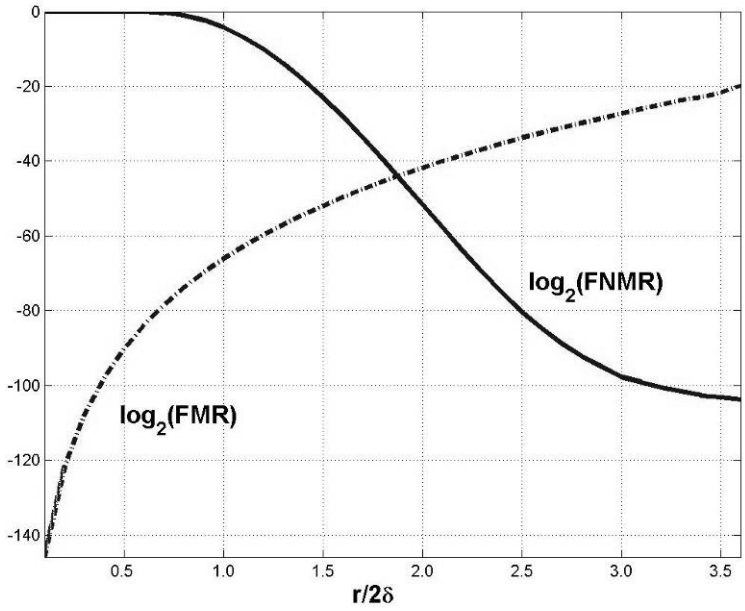

Figure 6. Fix q and examine the change of *FMR* and *FNMR* with respect to *r*. Plotted logarithmically. *FNMR* decreases and *FMR* increases as *r* increases.

(3) Simultaneous change of *q* and *r*

Figure 7 shows the change of *FMR* and *FNMR* with respect to the simultaneous change of *q* and *r*. We can clearly see the curve determined by $FNMR = FMR$.

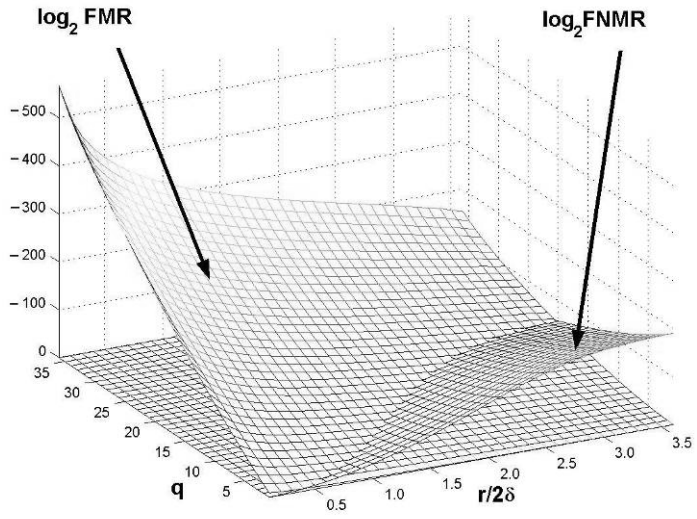

Figure 7. *FMR* and *FNMR* with respect to the simultaneous change of *q* and *r*. We can see a "valley" between two curved faces which denoted the points where *FNMR=FMR.*

Misidentification Control Strategy

Misidentification includes False-Match (FM) and False-Non-Match (FNM). From parameter analysis, we can see *FMR* and *FNMR* always change in opposite direction. So when seeking the minimum odds of misidentification, we should consider how to balance *FMR* and *FNMR*. Usually we put equal weight on *FMR* and *FNMR*, so we set the optimal function as:

$$\min \ FMR + FNMR$$
$$s.b. \ FMR = FNMR$$

However, putting equal weight on *FMR* and *FNMR* is not always desirable. In some situations, we might favor one over the other. The criminal identification case has a strict limit on *FNMR*, representing failure to match the criminal fingerprint with the one in database. It is much tolerant to FMR, because suspects can be excluded by other factors. In contrast, the authentication system has a strict limit on FMR, representing falsely matching a different fingerprint with the stored one.

Considering the different situations, we write a program in VC++ (Appendix II) to quickly solve for *q* and *r* to meet the different requirements on *FMR* and *FNMR*. It works like this:

First, we input the initialization parameters, such as image size, variance σ_r;

Second, we input the optimal function of *FMR* and *FNMR*;

Third, the program solves the optimal function and output *q*, *r*, *FMR* and *FNMR*.

The result of *q* tells us how to set the matching criterion, *r* for the grid size on the image. *FMR* and *FNMR* predict the odds of misidentification in this situation.

In consideration of the image quality, we can change the parameters *m* and *n* in the program. Smaller value is used for images that contain only part of the fingerprints or for images that are not clear, because less minutiae can be found on

them.

We apply this strategy to four typical situations and list the results here. Table 1 shows the results of putting equal weight on *FMR* and *FNMR*. Note that *FMR* and *FNMR* are not strictly equal, that is because the search step in program is discrete. Table 2 shows the results of criminal identification situation, where *FNMR* should be very low. Here we set $FMR = 10^8 FNMR$. Table 3 shows the results of authentication situation, where *FMR* should be very low. Here we set $10^8 FMR = FNMR$. Table 4 shows the results of a typical court situation, where $q = 12$ according to "the 12-point guideline" [3]. 12-point guideline means that a match consisting of 12 minutiae points is considered as sufficient evidence in many courts of law.

Table 1. Solution to minimum misidentification odds (equal weight on *FMR* and *FNMR*)

m, *n*	*q*	$r/(2\sigma)$	*FMR*	*FNMR*
m=*n*=12	6	3.0	1.68731e-009	1.53871e-008
13	7	3.4	3.22168e-010	1.25545e-008
14	7	3.0	2.04412e-010	1.7035e-009
15	7	3.0	7.06331e-010	8.51772e-011
16	8	3.0	2.93487e-011	1.89432e-010
18	9	3.0	4.8954e-012	2.11467e-011
20	10	3.0	9.3428e-013	2.36858e-012
22	11	3.0	2.0157e-013	2.66071e-013
24	12	3.0	4.86863e-014	2.99647e-014
26	13	3.0	1.30597e-014	3.38216e-015
30	15	2.9	3.85808e-016	1.38786e-016
36	19	2.9	4.52071e-019	8.00181e-018

Table 2. Solution to minimum misidentification odds, in criminal identification case

m, *n*	*q*	$r/(2\sigma)$	*FMR*	*FNMR*
m=*n*=12	4	3.4	2.78919e-005	1.17912e-012
13	4	3.4	5.69265e-005	4.00849e-014
14	5	3.4	3.54993e-006	1.36961e-013
15	6	3.4	7.77937e-006	4.8836e-015
16	6	3.2	2.73834e-007	2.86099e-014
18	7	3.2	4.35237e-008	3.28227e-015
20	8	3.2	7.99549e-009	3.69919e-016
22	9	3.2	1.67653e-009	4.11772e-017
24	10	3.1	2.07813e-010	8.35089e-018
26	11	3.1	5.1344e-011	9.52729e-019
30	13	3.0	1.46146e-012	2.46124e-020
36	16	3.0	5.25319e-014	3.93459e-023

Table 3. Solution to minimum identification odds, in authentication case

m, n	*q*	$r/(2\sigma)$	*FMR*	*FNMR*
m=*n*=12	8	2.9	2.52811e-014	2.33442e-005
13	9	3.2	4.33412e-015	1.75046e-005
14	9	2.8	2.40067e-015	1.75046e-005
15	10	3.1	6.19901e-016	2.32607e-006
16	10	2.9	6.19901e-016	2.32607e-006
18	11	2.9	1.93931e-016	3.96552e-008
20	12	2.9	4.6523e-017	4.7696e-009
22	14	3.1	9.91297e-019	4.9631e-009
24	15	3.0	1.1052e-019	8.02963e-010
26	16	3.0	4.41303e-020	9.2172e-011
30	18	2.9	1.98119e-021	3.08658e-012
36	21	2.8	4.19217e-023	1.99454e-014

Table 4. Solution of a typical court of law case (q = 12)

m, n	$r/(2\sigma)$	*FMR*	*FNMR*
m=*n*=12	2.5	6.93759e-029	0.452982
13	3.5	1.07247e-022	0.0429298
14	3.3	1.0158e-021	0.00569996
15	3.1	4.57148e-021	0.000736756
16	3.2	1.64615e-019	5.541e-005
18	3.5	2.03323e-016	1.87467e-007
20	3.4	3.2579e-015	7.44159e-010
22	3.6	4.37813e-013	1.53933e-012
24	3.0	4.86863e-014	2.99647e-014
26	2.7	3.70944e-014	3.24482e-015
30	2.3	4.42041e-014	1.08063e-015
36	1.9	7.65678e-014	1.89136e-014

Model of DNA Identification

Introduction to DNA Identification

According to the present STR (Short Tandem Repeat) technique[8], given two DNA strings, we can locate a certain number of loci. We then check if a certain allele-pair at one locus in one DNA string is exactly the same as the allele-pair appears at the same locus. If all allele-pairs at all n loci are checked to be the same, we announce that the DNA strings tested are from the same person; otherwise they are from different persons.

FNMR and FMR Calculation

Suppose we check A ($A \geq 1$) loci on the two strings.

At locus i ($1 \leq i \leq A$), we have B_i kinds of gene, and their probabilities of occurrence are $P_g(i, j)$ ($1 \leq j \leq B_i$, $\sum_{j=1}^{B_i} P_g(i, j) = 1$). These probabilities are usually obtained from a large database.

The allele-pair patterns appearing at locus i have two different types:

(1) **homozygotes:** The genes in the pair are of the same kind, namely $G_j \bullet G_j$ and the probability of the occurrence is $P_g{}^2(i, j)$;

(2) **heterozygotes:** The genes in the pair are different, namely $G_j \bullet G_k$ ($j \neq k$) and the probability of the occurrence is $2P_g(i, j)P_g(i,k)$.

Now we are able to calculate *FNMR* and *FMR*.

***FNMR*:** When the two strings compared are from the same person, the DNA method will announce them as matching, as far as the gene can always be recognized, because the DNA genes are discrete and mutual exclusive. Theoretically, recognition can always be successful. Therefore, we have *FNMR*=0;

***FMR*:** When the two test strings are different, the False Match Rate can be calculated from probabilities of gene occurrences. Since the probability of different gene's occurrence at a certain locus might vary, we can only get the expectation value of *FMR*.

$$E(FMR) = \prod_{i=1}^{A}\left(\sum_{j=1}^{B_i} P_g{}^4(i, j) + 4 \cdot \sum_{1 \leq j < k \leq B_i} P_g{}^2(i, j) P_g{}^2(i, k)\right)$$

$$= \prod_{i=1}^{A}\left(2 \cdot \left(\sum_{j=1}^{B_i} P_g{}^4(i, j)\right)^2 - \sum_{j=1}^{B_i} P_g{}^4(i, j)\right)$$

For a test DNA string, if all the loci are occupied by the most common genes, the risk that it matches with a second string is very high. We can calculate $FMR_{\max}$ in this way.

We let

$$P_g(i, j_0) = \max_{1 \leq j \leq B_i}\{P_g(i, j)\}$$

$$P_g(i, j_i) = \max_{\substack{1 \leq j \leq B_i \\ j \neq j_0}}\{P_g(i, j)\},$$

And assume

$$P_{g\max}(i) = \max\{P_g{}^2(i, j_0), 2P_g(i, j_0)P_g(i, j_1)\}$$

So

$$FMR_{\max} = \prod_{i=1}^{A} P_{g\_\max}(i)$$

If the second string is from a criminal, the innocent owner of the first DNA string has the most chance to be treated unfairly in court. This shows that $FMR_{\max}$ is a very important factor evaluating a certain DNA identification method.

Numerical Results

In order to calculate $E(FMR)$ and $FMR_{\max}$, we obtain the probability of gene occurring at a certain locus $P_g(i,j)$ $(1 \le j \le B_i, \sum_{j=1}^{B_i} P_g(i,j) = 1)$ from a real DNA database[9]. In this database, 2500 irrelevant blood samples are collected throughout Southern China, and the probabilities of all genes at 13 loci are recorded.

Interpol approach always checks 7 loci on two test strings, and US CODIS always 13[10]. We write a program (Appendix II) in VC++ to obtain the result. The input to the program is the probability of genes and the number (*A*) of loci being tested. The output is as below:

Interpol (number of loci $A = 7$):
$$E(FMR) = 4.6373 \times 10^{-9}$$
$$FMR_{\max} = 6.021825 \times 10^{-7}$$

US CODIS (number of loci $A = 13$):
$$E(FMR) = 3.914421 \times 10^{-14}$$
$$FMR_{\max} = 9.925334 \times 10^{-11}$$

Fingerprint vs. DNA

Since *FMR* and *FNMR* both affect the odds of misidentification, in order to get a small misidentification rate, we have to decrease both *FMR* and *FNMR*. As shown in Table 4, we match at least 12 pairs of minutiae (*q*=12), which is the sufficient evidence in many courts of law. By looking up the table, we finds m, n have to be at least 24. Otherwise either *FMR* or *FNMR* gets a larger value, thus results to larger odds of misidentification.

At this time, we get

$$FMR \approx 5.0 \times 10^{-14}$$
$$FNMR \approx 3.0 \times 10^{-14}.$$

From the DNA model, with the US CODIS(13 loci) standard, we have

$$E(FMR) = 3.914421 \times 10^{-14}$$

The above results show that *FMR*s of fingerprint and DNA methods are on the same order (10^{-14}), which means, these two methods have almost the same power to distinguish different persons.

However, the DNA method has *FNMR*=0, which is absolutely smaller than $FNMR \approx 3.0 \times 10^{-14}$ of the fingerprint method. This means that the fingerprint method has much less power in matching different profiles of the same person.

So we arrive at the conclusion that **the odds of fingerprint misidentification are greater than the odds of DNA method.**

Moreover, from the aspect of real application, due to the limitation of image sensoring, processing, and so on, it is not easy to make great improvement to the current fingerprint method, in terms of decreasing *FMR*. On the contrary, we can drastically decrease the misidentification rate by simply adding a locus for comparison in DNA approach. Therefore, **judging from the feasibility of model accuracy improving, DNA method is also better than fingerprint method.**

Model Validation

Sensitivity Analysis

In the fingerprint uniqueness model, we calculate the probability that two different fingerprints are matched by the following equation:

$$P_2 = \frac{\binom{m}{m}\binom{M-m}{0}}{\binom{M}{m}} \times p^m = \frac{p^m}{\binom{M}{m}}$$

After simplifying the equation, P_2 is determined only by m and M. Now we analyze how the error in parameter estimation influences the result. Alter M and m by 5%, recalculate the probability, and we get $P_2' \approx 8.2 \times 10^{-71}$. Compared with the result $P_2 \approx 2.27 \times 10^{-71}$, they are very close. So the model of fingerprint uniqueness is robust to give an estimation of the probability.

Validation of Fingerprint Identification Model

All our fingerprint considerations are based on Minutiae Matching. There are other approaches to identify fingerprints. FVC2002 - the 2nd International Competition for Fingerprint Verification Algorithms allows all kinds of approaches implemented [11]. An evaluation of each algorithm with respect to *FMR* and *FNMR* is depicted in Figure 8. Totally fifteen algorithms are tested against the same fingerprint database. The general tendency in the figure shows: if we want to get a low *FMR*, we have to trade off with high *FNMR*, and vice versa.

Our model can explain the internal mutual restriction between *FMR* and *FNMR*. And this is consistent with the phenomenon shown in the figure. It infers that although our model was developed from Minutia Matching, the conclusions drawn from it are valid to other approaches. Thus our conclusions are universal.

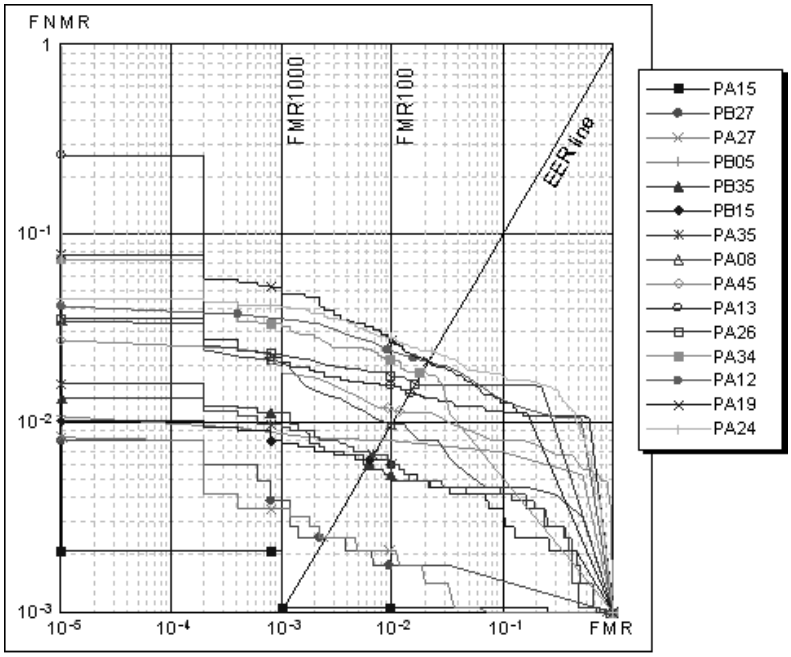

Figure 8. Fifteen unknown algorithms tested against FCV2002 fingerprint database. All algorithms are evaluated with *FMR* and *FNMR*, and each algorithm is represented with a curved line. From the general tendency of the lines, we may conclude *FMR* and *FNMR* are a sort of inversely related in this fingerprint identification problem. This diagram is obtained from the website of FCV2002 [11].

Strengths and Weaknesses

Strengths

- Our fingerprint uniqueness model is directly derived from the idea of "Minutiae Matching" and is independent of specific techniques. So when the conclusion on the probability is drawn, we are free of complicated conditional probabilities going along with specific techniques, such as by sensors or image processing. Therefore the conclusion reflects the essence of the argument "the thumbprint of every human who has ever lived is different".

- The fingerprint identification model is on the other hand, firmly related with real applications. We even provide a fast-checking table guiding readers to design an effective identification approach best fitting a specific application.

Weaknesses

- Our models are based on the assumption that all minutiae are normally distributed on the effective area of a finger. However, there are some voices supporting minutiae to be clustered in some certain areas. And this might result in errors.

- Our calculation of the DNA model depends on the probabilities of gene occurrence, thus determined by the database we choose. It is possible that with a different race, we may have different final misidentification rate. However, the following statement is always true no matter how we choose the database: “the more loci we check, the less misidentification will occur.”

- We separately analyze fingerprint and DNA identifications. However, there might be some relation between them, like fingerprint patterns are partially determined by DNA. If the relation is found, we can compare them on a more substantial level.

Further Discussion

General Analysis of Person Identification Problem

While a real person identification process might involve more than one method, for example, a thumbprint combined with a Date of Birth, it is safe that we first analyze a single-method case without losing generality. It is easy to combine different methods later on, only if none of the two methods combined are related, in terms of probability.

A Single-Method Case

Assume the total population is H.

Any Identification Method M always involves two aspects:

(1) **Sensitivity**: to distinguish different persons, and

(2) **Robustness**: to announce the same person as identical.

Consequently, we will have two kinds of false:

(1) **False Match**: cannot differ from two different persons (failure in Sensitivity);

(2) **False Not Match**: judge the same person as different (failure in Robustness).

We define $R \in [0,1]$ a Robustness variable shows our subjective matching requirement in a certain application. For example, $R = 0$ means there is no need to care if the same person is judged identical with this application, and $R = 1$ means that in this application the method should never miss announcing the same person as identical.

Thus we have a set of evaluation rules to measure if a certain method is “effective” in a certain application.

$$\begin{cases} FMR \le \dfrac{1}{H} \\ 1 - FNMR \ge R \\ 0 \le FMR, FNMR, R \le 1 \end{cases},$$

The second inequality is directly derived from the definition of R, thus easy to perceive. It is easier for us to explain the first inequality from the perspective of Information Theory. We can re-write the first inequality as following:

$$-\log FMR \geq -\log \frac{1}{H},$$

The amount of information $-\log FMR$, that we need to differentiate two arbitrary elements should be equal to or larger than the actual information contained in H elements, i.e. $-\log \frac{1}{H}$.

A Multiple-Method Case

Suppose there are N different methods used sequentially in the identification process. For the consideration of ease probability calculation, we assume that any of the two methods used here are irrelevant. Thus the formulae are modified into below:

$$\begin{cases} \prod_{i=1}^{N} FMR_i \leq \frac{1}{H} \\ \prod_{i=1}^{N} (1 - FNMR_i) \geq R \\ 0 \leq FMR_i, FNMR_i, R \leq 1 \\ i \in [1, N] \end{cases}$$

Example Analysis

Consider some extreme methods:

A. One method can be defined as that **any two persons to be compared are different** (here the two persons can be the same one), thus $FMR = 0$, and $FNMR = 1$.
B. The other extreme is that we take **any two persons to be compared are the same**, thus $FMR = 1$, and $FNMR = 0$.
C. The best method occurs in the **"Social Security Number (SSN)" or "Internet IP"** distribution: the number can single out a person from a set of people. Both FMR, and $FNMR$ are 0.
D. Take Fingerprint Criminal Identification as an Multi-Method case, as discussed before, in this case, *FMR* can be large, but we want to get the minimized *FNMR*. As the result, after matching a fingerprint obtained from the murder spot with the fingerprint database, police may get six suspects from AFIS. Then a different kind of identification method is applied, such as footprint mark, and finally the murderer's identity is clear. So **this means we sometimes do not have to pursue the identification quality of a single method, if we combine proper methods together, we may have very good identification result.**

References

[1] http://www.xs4all.nl/~dacty/schedule.htm

[2] *An Overview of Fingerprint Verification Technologies*, Information Security Technical Report, Vol.3, No.1, 1998, P.21-32.

[3] *On the Individuality of Fingerprints*, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 24, No. 8, August, 2002.

[4] *Summarization of Fingerprint Recognition Methods*, Journal of East China Shipbuilding Institute (Natural Science Edition), Vol. 17 No.3, Jun. 2003.

[5] D.A. Stoney, "*Distribution of Epidermal Ridge Minutiae*", Am. J. Physical Anthropology, vol. 77, pp. 367-376, 1988.

[6] http://www.prb.org/Content/ContentGroups/PTarticle/0ct-Dec02/How_Many_People_Have_Ever_Lived_on_Earth_.htm

[7] Fingdb.zip, Databases and Software, Biometric Systems Lab. http://bias.csr.unibo.it/research/biolab/bio_tree.html

[8] http://www.cstl.nist.gov/biotech/strbase/

[9] http://www.37c.com.cn/literature/analecta/data/fyxzz/200001/001.html Du Zhichun, el al. *Polymorphism of 13 STR loci for establishment of Chinese criminal DNA database* DF795.2

[10] Interpol handbook on DNA data exchange and practice Interpol press release 6 November 2001

[11] Performance Evaluation, FVC2000, Fingerprint Verification Competition, http://bias.csr.unibo.it/fvc2000/perfeval.asp

Appendix I

Distribution Fitting

Side length of grid r

In an actual fingerprint image, the value of M is related with r - the side length of the grids. r should be so large that the same minutia point in different images can fall in the same grid, meanwhile r should be small enough to contain at most one distinct minutia. So the value of r should be determined with great caution.

Our approach is to fit the distribution of position distance of the same minutia in different impressions along x and y axes separately, form which we can choose a proper lower bound of r. We used a fingerprint database [5] that contains 168 images (21fingers – 8 images per finger) of good quality. The images are 256*256 gray-scale PC TIF files.

Pre-processing of data-images

Consider the 8 images of the same finger. Any two are of the same scale (since they are obtained from the same finger and should be through the same process), but may have rotations and translations.

Suppose we set image A as a reference, and an image B should be aligned against A. (x, y) is coordinates of an arbitrary point in A, (x', y') is the coordinates of the corresponding point in the aligned image B. The transformation is give by:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & x_0 \\ \sin\theta & \cos\theta & y_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

where θ is the rotational angle, and x_0 and y_0 are the translation distance in x and y axis respectively.

When $\theta \approx 0^\circ$, which is true in our database images, we approximate $\cos\theta \approx 1$, $\sin\theta \approx 0$. Thus, the above equation reduces to:

$$\begin{cases} x' \approx x + x_0 \\ y' \approx y + y_0 \end{cases}$$

which means we can neglect the rotational difference in two images.

Assume there is no translation displacement, the distance between the two points of the same minutia should be around 0, and the best case is that all the distances are 0 (a case that image A compared with image A itself). We simply eliminated translation displacement with a subtraction of “mean value of all distances” from every distance value.

Distance Distribution Calculation

First, we arbitrarily pick up one set of the impressions and manually locate 6 correspondent minutiae in all impressions. See Figure 9.

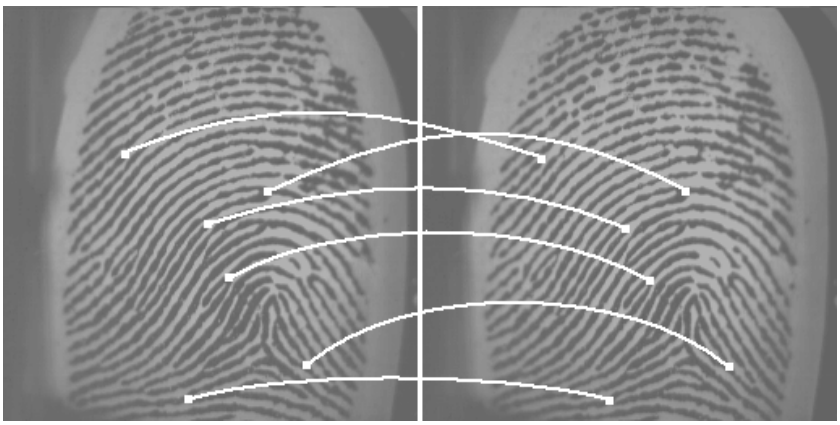

Figure 9. Manually picked 6 mimutiae, this fingerprint image is obtained from [7].

Suppose the minutiae are ordered with label j, and then all the minutiae in the i th impression are denoted by:

$$\{(x_{i,1}, y_{i,1}), (x_{i,2}, y_{i,2}), \cdots, (x_{i,j}, y_{i,j}), \cdots\} \quad (i \in [1,8],\ j \in [1,6])$$

Second, we get the difference between every pair in any two different images from the 8 images of the same finger along x and y axes. The distances of the 1st and 2nd images are listed in the first row of the table as below, and the distance values are adjusted by subtracting the mean value of 15.3333. Here a minus sign before a certain value just indicates the direction of calculation:

	1st point	**2nd point**	**3rd point**	**4th point**	**5th point**	**6th point**
Directly Calculated Distance	13	16	20	23	16	4
Adjusted Distance	-2.3333	0.66667	4.6667	7.6667	0.66667	-11.333

Finally, all minutiae distance distribution in the unit of pixel are depicted in Figure 4. With the Matlab 'normfit' function, we get $\delta_x = 2.748$, $\delta_y = 2.8756$ thus we set $\delta = (\delta_x + \delta_y)/2 = 2.8118$.

Appendix II

1. VC++ program calculating fingerprint *FNMR* and *FMR* with respect to *r* and *q*

```
#include<iostream>
#include<vector>
#include<string>
#include<cstdio>
#include<cstdlib>
#include<cmath>
using namespace std;

#define for if(0); else for
#define MIN(x, y) (((x) < (y)) ? (x) : (y))

const double coef = 1e8;
```

```
//---------------------------------
double C(int a, int b){
	if(a < b) return 0;
	if(a == b) return 1;
	if(b > a / 2) b = a - b;
	double ret = 1;
	for(int s = 1, d = a; s <= b; s++, d--)
		ret = ret * d / s;
	return ret;
}

double F(int M, int m, int n, int k){
	return C(m, k) * C(M - m, n - k) / C(M, n);
}
double bin(int n, int k, double p){
	return C(n, k) * pow(p, k) * pow(1 - p, n - k);
}
//---------------------------------

bool comp_equ(double fmr, double fnmr, double new_fmr, double new_fnmr){
	double temp = (fmr + fnmr) / (new_fmr + new_fnmr);
	if(temp >= 2 || 0.5 < temp && temp < 2 && new_fmr * new_fnmr < fmr * fnmr)
		return true;
	return false;
}

bool comp_id(double fmr, double fnmr, double new_fmr, double new_fnmr){
	return comp_equ(fmr, coef * fnmr, new_fmr, coef*new_fnmr);
}

bool comp_ver(double fmr, double fnmr, double new_fmr, double new_fnmr){
	return comp_equ(coef * fmr, fnmr, coef * new_fmr, new_fnmr);
}
double fmr[100][100][100], fnmr[100][100][100]; // form: [m][q][r/(2*dt) * 10]

/**
 * @param MN, MX	the minimum and maximum minutia number in the effective area
 * @param size	image's size
 * @param div		if the effective area have M grids, the MX can't exceed M / div
 * @param pr		odds of correspondent minutiae in different images of same finger falling in same grid.
 * @param pth		odds of correspondent minutiae in different images of same finger having same grid having the same direction and style
 * @param pl		odds of different minutiae having same direction and style
 * @param dt	standard variation of the distribution of the grid side length
 * @param overlap	effective area / image area
 *
 */

class featureMatch
{

	int MX, MN, size, div;
	double pr, pth, pl, dt, overlap;

	vector<double> p;
public:
	featureMatch(){
		for(int i = 0; i < 100; i++)
			for(int j = 0; j < 100; j++)
				for(int k = 0; k < 100; k++)
					fmr[i][j][k] = fnmr[i][j][k] = 1.0;
		size = 256;
		MX = 36;
```

```
        MN = 12;
        div = 1;
        overlap = 0.25;
        dt = 2.87;
        pl = 0.0625;
        pth = 0.975;
        double pp[] = {
                0.5000, 0.5398, 0.5793, 0.6179, 0.6554, 0.6915,
                0.7257, 0.7580, 0.7881, 0.8159, 0.8413, 0.8643,
                0.8849, 0.9032, 0.9192, 0.9332, 0.9452, 0.9554,
                0.9641, 0.9713, 0.9772, 0.9821, 0.9861, 0.9893,
                0.9918, 0.9938, 0.9953, 0.9965, 0.9974, 0.9981,
                0.9987, 0.9990, 0.9993, 0.9995, 0.9997, 0.9998,
                0.9999};
        p = vector<double>(pp, pp + 37);
}

/**
  * create all data.
  */
void init(){
        for(int i = 1; i <= 36; i++){
                int M = floor(size * size * overlap / (0.2 * i * dt) / (0.2 * i * dt));
                pr = (2*p[i] - 1) * (2*p[i] - 1);
                for(int m = MN; m <= MIN(M / div, MX); m++){
                        opti(M, m, i);
                }
        }
}
void opti(int M, int m, int no){
        double U[100];

        for(int i = 0; i <= m; i++)
                U[i] = F(M, m, m, i);

        fmr[m][m + 1][no] = 0;
        fnmr[m][m + 1][no] = 1;

        for(int i = m; i >= 0; i--){
                double fmr_temp = 0;
                for(int j = i; j <= m; j++)
                        fmr_temp += U[j] * bin(j, i, pl);
                fmr[m][i][no] = fmr[m][i+1][no] + fmr_temp;
        }

        fnmr[m][0][no] = 0;
        for(int i = 1; i <= m; i++)
                fnmr[m][i][no] = fnmr[m][i-1][no] + bin(m, i - 1, pr * pth);
}

/**
  * print the ans according the compare function f
  */
void printBest(bool (*f)(double , double , double , double )){
        double b_fmr = 1, b_fnmr = 1;

        int b_m, b_q, b_no;
        for(int no = 1; no <= 36; no++)
                for(int m = MN; m <= MX; m++)
                        for(int q = 12; q <= m; q++)
                                if(f(b_fmr, b_fnmr, fmr[m][q][no], fnmr[m][q][no])){
                                        b_fmr = fmr[m][q][no];
                                        b_fnmr = fnmr[m][q][no];
                                        b_m = m, b_q = q, b_no = no;
                                }
```

```
            print(b_m, b_q, b_no);
        }

        void printBest1(bool (*f)(double , double , double , double )){

            for(int m = 12; m <= MX; m++){
                double b_fmr = 1, b_fnmr = 1;

                int b_m, b_q, b_no;
                for(int no = 1; no <= 36; no++)
                for(int q = 12; q <= 12; q++)
                    if(f(b_fmr, b_fnmr, fmr[m][q][no], fnmr[m][q][no])){
                        b_fmr = fmr[m][q][no];
                        b_fnmr = fnmr[m][q][no];
                        b_m = m, b_q = q, b_no = no;
                    }
                print(b_m, b_q, b_no);
            }

        }
        void print(int b_m, int b_q, int b_no){
            int M = floor(size * size * overlap / (0.2 * b_no * dt) / (0.2 * b_no * dt));
            cout << M << ' ' << b_m << ' ' << b_q << ' ' << b_no << ' ' << fmr[b_m][b_q][b_no] << '
' << fnmr[b_m][b_q][b_no] << endl;
        }
        void printM(int m, FILE *fp){
            for(int j = 1; j <= m; j++){
                for(int i = 1; i <= 36; i++)
                    fprintf(fp, "%03.3lf ", -log(fmr[m][j][i])/log(2));
                fprintf(fp, "\n");
            }
            fprintf(fp, "\n**********************\n");
            for(int j = 1; j <= m; j++){
                for(int i = 1; i <= 36; i++)
                    fprintf(fp, "%03.3lf ", -log(fnmr[m][j][i])/log(2));
                fprintf(fp, "\n");
            }
        }
};

int main(){
        featureMatch a;
        a.init();
        a.printBest1(comp_equ);
        cout << "--------------------"<<endl;
        a.printBest1(comp_id);
        cout << "--------------------"<<endl;
        a.printBest1(comp_ver);
        cout << "--------------------"<<endl;
        a.printBest(comp_equ);
        a.printBest(comp_id);
        a.printBest(comp_ver);
//      FILE * fp = fopen("数据.txt", "w");
//      a.printM(36, fp);
        return 0;
}
```

2. VC++ program calculating DNA FMR

```
#include<iostream>
#include<string>
#include<vector>
#include<algorithm>
using namespace std;

#define for if(0); else for
```

```
struct GP{
	string name;
	int n;
	vector<string> vt;
	vector<double> vp;
	GP(string name, int n, vector<string> vt, vector<double> vp):name(name), n(n), vp(vp){
		valid();
	}
	void valid(){
		double sum = 0;
		for(int i = 0; i < vp.size(); i++)
			sum += vp[i];
		for(int i = 0; i < vp.size(); i++)
			vp[i] /= sum;

		cout << name << ':' << sum << endl;
	}
	double get_match_rate_evg(){
		double ret1 = 0, ret2 = 0;
		for(int i = 0; i < n; i++){
			ret1 += vp[i] * vp[i];
			ret2 += vp[i] * vp[i] * vp[i] * vp[i];
		}
		cout << "rate_evg " << 2 * ret1 * ret1 - ret2 << endl;
		return 2 * ret1 * ret1 - ret2;
	}
	double get_match_rate_max(){
		if(n == 1) return 1;
		vector<double> cp = vp;
		sort(cp.begin(), cp.end());
		if(cp[n-1] > 2 * cp[n-2]) return cp[n-1] * cp[n-1];
		else return 2 * cp[n-1] * cp[n-2];
	}
};

struct DNA{
	vector<GP> vg;
	double get_fmr_evg(vector<string> str){
		double ret = 1.0;
		for(int i = 0; i < vg.size(); i++)
			if(find(str.begin(), str.end(), vg[i].name) != str.end())
				ret *= vg[i].get_match_rate_evg();
		return ret;
	}
	double get_fmr_max(vector<string> str){
		double ret = 1.0;
		for(int i = 0; i < vg.size(); i++)
			if(find(str.begin(), str.end(), vg[i].name) != str.end())
				ret *= vg[i].get_match_rate_max();
		return ret;
	}
};

int main(){
	string name;
	int n;
	vector<string> vt;
	vector<double> vp;
	DNA a;
	while(cin>>name>>n){
		vt.clear();
		vp.clear();
		for(int i = 0; i < n; i++){
			string nm;
```

```
                double p;
                cin>>nm>>p;
                vt.push_back(nm);
                vp.push_back(p);
            }
            a.vg.push_back(GP(name, n, vt, vp));
        }

        vector<string> inter;
        inter.push_back("D3S1358");
        inter.push_back("VWA");
        inter.push_back("FGA");
        inter.push_back("D8S1179");
        inter.push_back("D21S11");
        inter.push_back("D18S51");
        inter.push_back("TH01");

        vector<string> usa = inter;
        usa.push_back("D7S820");
        usa.push_back("CSF1PO");
        usa.push_back("TPOX");
        usa.push_back("D16S539");
        usa.push_back("D13S317");
        usa.push_back("D5S818");
        printf("%.6e %.6e\n", a.get_fmr_evg(inter), a.get_fmr_max(inter));
        printf("%.6e %.6e\n", a.get_fmr_evg(usa), a.get_fmr_max(usa));
        return 0;
}
```