

2007年度 卒業論文

共通マニューシャ抽出アルゴリズム
による指紋認証システム
“CME”

電子情報学部 情報メディア学科

青木 良樹 小西 亨

目次

1	はじめに	1
2	要素技術	2
2.1	NIST Fingerprint Image Software 2	2
2.2	M(i)-tuple	2
2.2.1	M(i)-tuple の作成	3
2.2.2	共通マニユーシャ候補の抽出	4
2.3	Common Minutiae Extractor	5
2.3.1	共通マニユーシャの抽出	5
3	提案システム“CME”の実装	7
3.1	使用機材	7
3.2	システム概要	7
3.3	プログラム概要	7
4	実験	10
4.1	実験方法	10
4.1.1	本人拒否率 (FRR), 他人受入率 (FAR) の算出	10
4.2	評価	11
4.3	考察	11
5	結論	16
6	夏期休暇における課題	
	一番正しい特徴点をとれるのは誰か?	17
6.1	概要	17
6.2	実験方法	17
6.3	実行画面	18
6.4	実験結果	18
6.5	考察	19

1 はじめに

近年、キャッシュカードやクレジットカードの偽造、フィッシング詐欺によるパスワードの盗聴が増加している。銀行などではパスワードによる認証のみではなく、更に指紋認証や静脈認証などの生体情報を利用した認証を行っているものもある。このように、安全性、利便性が高い指紋認証が実用化され、普及し始めているが、普及していく中で発生すると考えられる問題がある。

既存のマニユーシャマッチング方式は、マニユーシャの絶対座標について比較を行っている。そのため、登録時と認証時のセンサが同一機種でないと、指紋画像の画質や隆起線の太さや画像のサイズにばらつきが生じ、認証できないという、センサ依存問題が発生する。

そこで本研究では、センサ依存問題を解決する手法として、Chandrasekaranらの文献^[2]において提案されている、マニユーシャ間の距離比と角度を利用したマニユーシャマッチング方式を実装し、異なるセンサで撮影した指紋画像でもマッチングを可能とする、共通マニユーシャ抽出アルゴリズム *CME* (*Common Minutiae Extractor*) を提案する。

また、提案方式の有意性を示すために Java による試験実装を行い、精度と処理性能を報告する。

2 要素技術

2.1 NIST Fingerprint Image Software 2

今回は, NIST Fingerprint Image Software2 (NFIS2)^[1]の *mindtct* パッケージを用いてマニユーシャ¹を抽出した *mindtct* は指紋画像からマニユーシャを抽出し, マニユーシャの座標や信頼度などの情報を“*min*”というテキストファイルに書き出す *mindtct* と同様に *NFIS2* が提供する機能には, 表1のようなものがある.

表 1: NFIS2 が提供する機能

機能	プログラム
指紋形状の分類	pcasys
特徴点抽出	mindtct
マッチング	bozorth3
画像処理	NFIQ

2.2 $M(i)$ -tuple

図1のようなマニユーシャを持った, 2枚の指紋画像を例に説明する. 左の画像と右の画像においてマニユーシャそれぞれの座標は異なるが, $(B, A) : (B', A')$, $(B, C) : (B', C')$ の比と, それぞれの辺が成す角度は等しい.

そこで, B についての A, C の距離比を

$$R_B(A, C) = \frac{\max(AB, BC)}{\min(AB, BC)}$$

と定める. すると, 明らかに

$$R_B(A, C) = \frac{4}{2} = R_B(C, A) = R'_B(A', C') = \frac{2}{1}$$

¹指紋の隆線における端点, 分岐点をマニユーシャと呼ぶ.

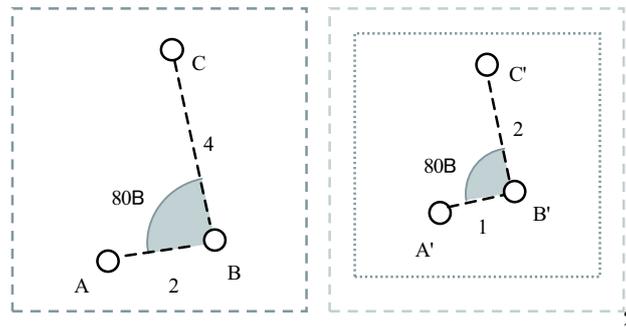


図 1: 縮小前と縮小後の画像モデル

である．距離比と角度が決められた閾値内にある B と B' を共通マニューシャ候補と呼ぶ．指紋画像から，すべての距離比，角度を算出して比較を行えば，画像サイズに依存することなくマニューシャマッチングが可能である．

2.2.1 $M(i)$ -tuple の作成

マニューシャ i の位置と角度の相対的な関係を表す特徴ベクトル “ $M(i)$ -tuple” を次の様に定める．

i 番目のマニューシャを基準とし，ユークリッド距離の 10 近傍のマニューシャ間のユークリッド距離の比 $r_i(j, k)$ と角度 $\theta_i(j, k)$ を i の tuple の要素とする．10 個のマニューシャの中の異なる組 (j, k) について，角度 $\theta_i(j, k)$ は $\angle jik$ の大きさ，距離比 $r_i(j, k)$ は次の (1) 式の様にする．

$$r_i(j, k) = \frac{\max(d(i, j), d(i, k))}{\min(d(i, j), d(i, k))} \quad (1)$$

ただし，ここで $d(i, j)$ は ij 間のユークリッド距離である． $i = 2$ の 4 近傍の例を図 2 に示す．この例における距離比は，

$$r_2(0, 4) = \frac{\max(0.93, 0.18)}{\min(0.93, 0.18)} = 1.14$$

である．こうして， N 個のマニューシャのそれぞれについて， N 個の tuple を作成する．

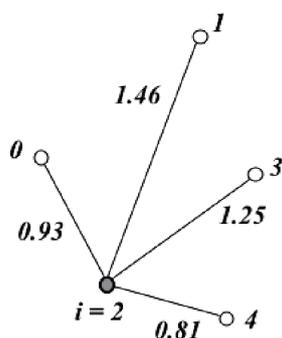


図 2: マニューシャ2の4近傍とユークリッド距離

近傍 10 点との比較を行うと, ${}_{10}C_2 = 21$ 個の距離比と角度があり, 計 45 次元のベクトルを $M(i)$ -tuple とする. 図 2 における $i = 2$ 番目のマニューシャを例として作成した tuple を表 2 に示す.

表 2: $M(2)$ -tuple の例

j, k	$r_i(j, k)$	$\theta_i(j, k)$
0, 1	1.56	50
0, 3	1.34	80
0, 4	1.14	150
1, 3	1.16	30
1, 4	1.80	90
3, 4	1.54	45

2.2.2 共通マニューシャ候補の抽出

登録画像をベースイメージ (BM), 入力画像をインプットイメージ (IM) とする. BM と IM についての tuple を各々 M_i と M'_i とする. M_i, M'_i を総当たりで比較し, 次の手順でマニューシャ間の共通マニューシャ候補を抽出する.

2つの tuple を

$$M(i) = (r_1, \dots, r_n, \theta_1, \dots, \theta_n) \quad (2)$$

$$M'(j) = (r'_1, \dots, r'_n, \theta'_1, \dots, \theta'_n) \quad (3)$$

とする時，類似度 $S_{i,j}$ を

$$S_{i,j} = \sum_{\ell=1}^n s_{i,j}(\ell) \quad (4)$$

とする．ここで，

$$s_{i,j}(\ell) = \left| \{(r'_k, \theta'_k) \mid |r_\ell - r'_k| < \lambda_r, |\theta_\ell - \theta'_k| < \lambda_\theta\} \right|$$

と定める．ただし， λ_r は距離比の閾値， λ_θ は角度についての閾値とする． n は tuple の次元であり，10 近傍のとき $n = 45$ である． $S_{i,j}$ は，すなわち，閾値未滿で一致する tuple の要素の数であり，最大 n^2 をとる．

BM の N_B 個のマニューシャと IM の N_I 個のマニューシャの全ての組み合わせについて，類似度 $S_{i,j}$ についての閾値 $T_1 = 3$ 以上となる共通マニューシャ候補集合を

$$C = \{(i, j) \mid S_{i,j} \geq T_1\}$$

と定める．

2.3 Common Minutiae Extractor

2.3.1 共通マニューシャの抽出

絞り込んだ全ての共通マニューシャ候補において，同様の手法でより厳密な共通マニューシャの組を抽出する．

Step 1 : 全ての共通マニューシャ候補 $(i, j) \in C$ について 4 式により類似度を求める．ただし，共通マニューシャ候補数 $m = |C|$ ，とする．

Step 2 : こうして定められた $m \times m$ の類似度を，BM の y 座標について $S_{i,j}$ をソートし直す．その行列を

$$D = \begin{pmatrix} S_{1,1} & \cdots & S_{1,m} \\ \vdots & \ddots & \vdots \\ S_{m,1} & \cdots & S_{m,m} \end{pmatrix}$$

とおく．インデックス集合 $K_1 = \{1, \dots, m\}$ ， $J_1 = \phi$ とする．

Step 3 : $k = 1, \dots, m$ について, $j = 1, \dots, m$ で $S_{k,j}$ の最大値をとるものを j_k とおく .

j_k について, $S_{k,j_k} > T_2$ ならば,

$$J_{k+1} = J_k \cup \{j_k\}$$

$$K_{k+1} = \{\ell \in K_k \mid \ell > j_k\}$$

と更新し, *Step 3* を繰り返す . ここで, T_2 は類似度 S_{k,j_k} についての閾値であり, 実験では $T_2 = 15$ とした . 条件を満たさなければ, $K_{k+1} = K_k$, $J_{k+1} = J_k$ とする .

Step 4 : *IM* のスコアを, $S_{IM} = |J_m|$ とする . $S_{IM} \geq T_3$ であったならば, *IM* は *BM* とマッチしているとする . T_3 はマッチングの厳密さを決める閾値である .

3 提案システム“CME”の実装

3.1 使用機材

今回のシステムを実装するにあたって，表3のような機能を持った指紋リーダーを利用した．

表 3: U.are.U 4000(Digital Persona 社)

インターフェース	USB(1.0/1.1/2.0)
ケーブル長	175cm
解像認識エリア	14.6mm × 18.1mm
解像データ	512dpi 8bit グレースケール
価格	19,000 円
Web サイト	http://www.digitalpersona.com/

3.2 システム概要

ベースイメージ (BM) とインプットイメージ (IM) の指紋画像を U.are.U により撮影し，それぞれの画像からマニューシャを抽出する．次に，それらのマニューシャをもとに $M(i)$ -tuple と $M'(i)$ -tuple をそれぞれ作成する． $M(i)$ -tuple と $M'(i)$ -tuple を比較し，偽マニューシャなどのノイズ除去を行い，純度の高いマニューシャを抽出する．そのマニューシャを入力として， $M(i)$ -tuple と同様に $C(i)$ -tuple と $C'(i)$ -tuple を作成し，比較した結果の画像を画面に表示する．

3.3 プログラム概要

- Minutia.java
min ファイルの情報を入力とし，マニューシャ情報を読み込み，各マニューシャのユークリッド距離を計算する．
min ファイルの一部を表4に示す．min ファイルの内容を全て掲載すると膨大な量になってしまうため，割愛する．min ファイルは1

行目に画像サイズ，3行目に抽出したマニューシャ数が記されており，5行目以降はマニューシャの情報になっている．マニューシャの情報は“:”で区切られており，左から，マニューシャナンバー，x座標，y座標，角度，信頼度の順に並ぶ．今回は，マニューシャナンバー，x座標，y座標，信頼度の4つの要素を利用した．

- MiTuple.java

ユークリッド距離を入力とし，中心となるマニューシャ i の10近傍の距離比 $r_i(j,k)$ ，角度 $\theta_i(j,k)$ を算出し，M(i)-tupleを記録したファイルを出力する．

tupleファイルの一部を表5に示す．tupleファイルは1行目に比の数，2行目に抽出したtuple数が記されており，7行目以降はtupleの情報になっている．tupleの情報は最初に， i 番目のマニューシャの座標と，minファイルにおいてのマニューシャナンバーが記録され，次の行からそのtuple情報が記録される．tuple情報は左から，インデックス，距離比 r ，角度 θ であり，PointA，PointBは，図1におけるAとCである，x座標や，y座標が記録される．

- Candidature.java

2つのM(i)-upleを入力とし，各M(i)-tuple同士の距離比，角度を総当たりで比較してスコアテーブルを作成する．スコアテーブル作成後， $s_{i,j}$ の大きいM(i)-tupleの対を探し出し，そのマニューシャの座標を記録したBM，IMファイルを出力する．

BM，IMファイルの一部を表5に示す．BM，IMファイルは1行目から共通マニューシャ候補の情報が記され，左からx座標，y座標，tuple番号となる．

- DrawLightning.java

BM，IMファイルを入力とし，C(i)-tupleの生成，及び共通マニューシャの抽出を行う．

- MainFrame.java

GUIで2枚の指紋画像から抽出した共通マニューシャを表示する．指紋を撮影すると同時にmindtctを実行し，CMEに必要なminファ

イルを出力する。撮影された2枚の指紋画像をそれぞれBM, IMとしてCMEによるマニューシャ抽出を行う。共通マニューシャの抽出は, Candidature オブジェクトを生成することによって実現している。

次に, ユーザーインターフェースについて説明する。登録ボタンを押すと指紋入力待ち状態となり, 2枚の指紋が撮影可能になる。指紋を2枚撮影すると登録ボタンは取消ボタンとなり, 取消ボタンを押すと現在行っている認証を取り消して最初の状態に戻る。認証ボタンを押すと, CMEによる共通マニューシャ抽出を実行し, 対応した共通マニューシャを視覚的に分かりやすくするため, 共通マニューシャ同士を線で結び, 最終的に抽出された共通マニューシャを表示する。ここで, 図中の赤点は共通マニューシャであり, 黄点が共通マニューシャである。

MainFrame.java 実行画面を図3に示す。

表 4: min ファイルの一部

```
Image (w,h) 500 550

80 Minutiae Detected

0 : 60, 307 : 2 : 0.127 ...
1 : 68, 238 : 4 : 0.139 ...
2 : 68, 280 : 19 : 0.168 ...
3 : 74, 205 : 4 : 0.145 ...
4 : 91, 156 : 4 : 0.142 ...
5 : 97, 273 : 3 : 0.963 ...
:
```

表 5: tuple ファイルの一部

```
Number of ratio : 45
Number of tuple : 43
File name : Tuple/01.tuple

Format - S.No - -Rati - -Degree

M(0)-tuple [87,449] Minutiae[7]
1-1.0-2.0-//PointA-15##27##192##483-//PointB-16##28##193##419
2-1.17-6.0-//PointA-15##27##192##483-//PointB-13##24##179##430
3-1.2-14.0-//PointA-15##27##192##483-//PointB-14##25##179##442
4-1.58-43.0-//PointA-15##27##192##483-//PointB-5##14##121##388
:
```

4 実験

共通マニューシャ抽出アルゴリズム“CME”を実装するにあたり，最適な T_3 の値を見つける必要があるため，下記に示すような方法で実験を行い，最適な閾値 T_3 を決定する．

4.1 実験方法

4.1.1 本人拒否率 (FRR)，他人受入率 (FAR) の算出

FRR(本人拒否率): 被験者 1 人の指紋を 551 枚収集する．そのうち 1 枚の指紋画像を BM とし，残りの 550 枚の IM を照合する．

FAR(他人受入率): 被験者 11 人の指紋を 1 人につき 50 枚収集する．FRR と同様の BM に対して，被験者それぞれの指紋 50 枚を照合する．

閾値: $\lambda_r = 0.01$ ， $\lambda_\theta = 3.5$ を用いた．

今回は各グラフにおいて FAR と FRR が交わるときの T_3 を最適な閾値とし，2 つの実験を行った．

実験 1: 同じサイズの指紋画像における評価
実験 2: 異なるサイズの指紋画像における評価

表 6: BM , IM ファイルの一部

284 244 32
359 336 42
314 265 37
323 361 39
278 93 30
121 388 5
⋮

$T_3 = 1, 2, \dots, 30$ と変化させていき FRR , FAR を算出し , 最適な T_3 を決定する . なお , 実験 2 で使用する異なるサイズの指紋画像は BM に対し 70 % の大きさの IM を使用した .

4.2 評価

実験を行った結果 , 実験 1 では $T_3 = 11$ のときに $FRR = 0.13$, $FAR = 0.12$ を得た . 実験 2 では $T_3 = 7$ のときに $FRR = 0.29$, $FAR = 0.27$ を得た . 以上のことから , IM のサイズによって最適な閾値が異なることがわかる . 実験 1 の認証精度を図 4 , 実験 2 の認証精度を図 5 に示す .

図 6 にサイズの異なる 2 つの指紋画像について提案方式を適用した結果を示す . 異なるサイズの指紋画像でも共通マニューシャを抽出できていることがわかる .

4.3 考察

本実験で得られた最適な閾値 T_3 を , 実験 1 において $T_3 = 11$ の時に $FRR = 0.13$, $FAR = 0.12$ が得られた . また , 実験 2 では $T_3 = 7$ の時に $FRR = 0.29$, $FAR = 0.27$ が得られた . これは , 単純計算で 10 回に 1 回以上 , あるいは 2 回以上の誤りがあると考えられ , 認証精度としては十分な結果ではない .



図 3: MainFrame.java 実行画面

しかし、提案システムによって異なるサイズの指紋画像において、共通マニューシャを抽出できることが図 5、図 6 から明らかになった。指紋画像のサイズによって最適な T_3 が異なるため、それよって動的に T_3 の値を設定する必要があると言える。また、今回は $M(i)$ -tuple 作成時の近傍数や、 r や θ 、 T_1 、 T_2 などの各閾値を一定にして行ったが、それらは検証を繰り返していく中で、ヒューリスティックに定めたものに過ぎず、最適な値とは限らない。従って、 T_3 以外の閾値においても最適な値を求める必要がある。

特に、本人、他人間での共通マニューシャ候補数の変動が極めて大きいことが分かった。今回の実験では 550 枚の IM 画像を使用した。その中で共通マニューシャ候補数が 30 個以上取れる指紋が 57% の割合で存在する。一方、BM を他人の指紋により認証した場合は、共通マニューシャ候補数が 30 個以上とれる指紋は 14% と明らかに低い。つまり、共通マニューシャ候補を抽出した時点で IM 画像が本人のものか、他人のものかをある程度判断できるということがいえる。本人の指紋で認証した場合

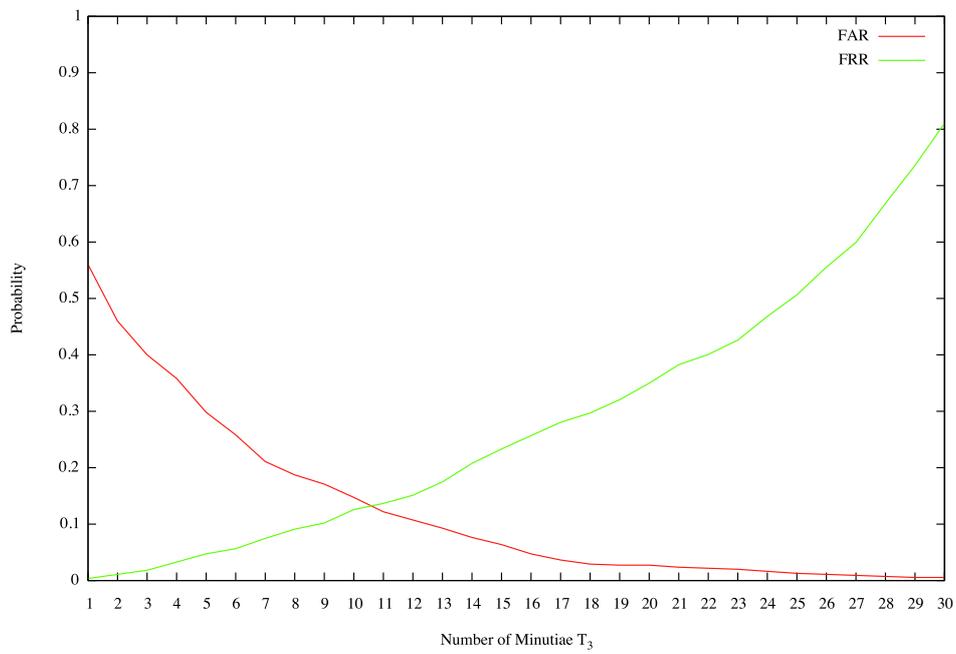


図 4: 同サイズの指紋画像を使用した認証精度

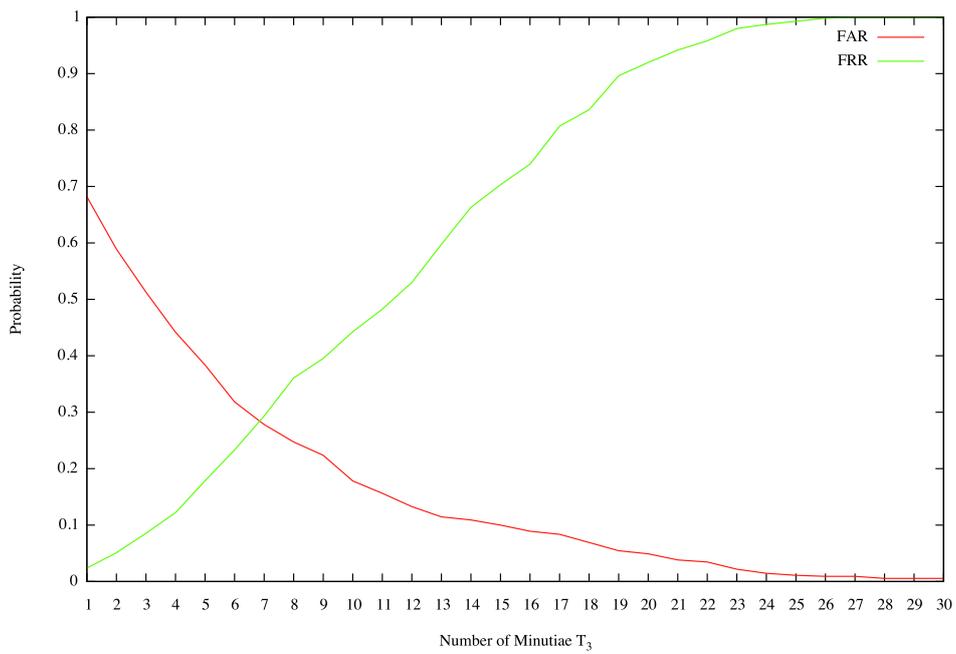


図 5: 異なるサイズの指紋画像を使用した認証精度



図 6: 異なるサイズでのマッチング実行例 (左:BM , 右:IM)

と，他人の指紋で認証した場合の共通マニューシャ数のヒストグラムを図 7 に示す．

このことから，共通マニューシャ候補数に応じて，共通マニューシャを抽出する際の閾値を動的に変化させた方が良いと考えられる．例えば，共通マニューシャ候補数が 30 個未満の場合，他人の可能性が高いため，共通マニューシャ抽出の際に $\lambda_r, \lambda_\theta$ を低く見積もる．閾値を厳しくすることにより，他人の指紋であれば共通マニューシャは更に抽出しにくくなり，本人の指紋であれば閾値が厳しくなっても共通マニューシャはある程度抽出される．

このように動的に閾値を設定することにより抽出精度が向上することが考えられる．

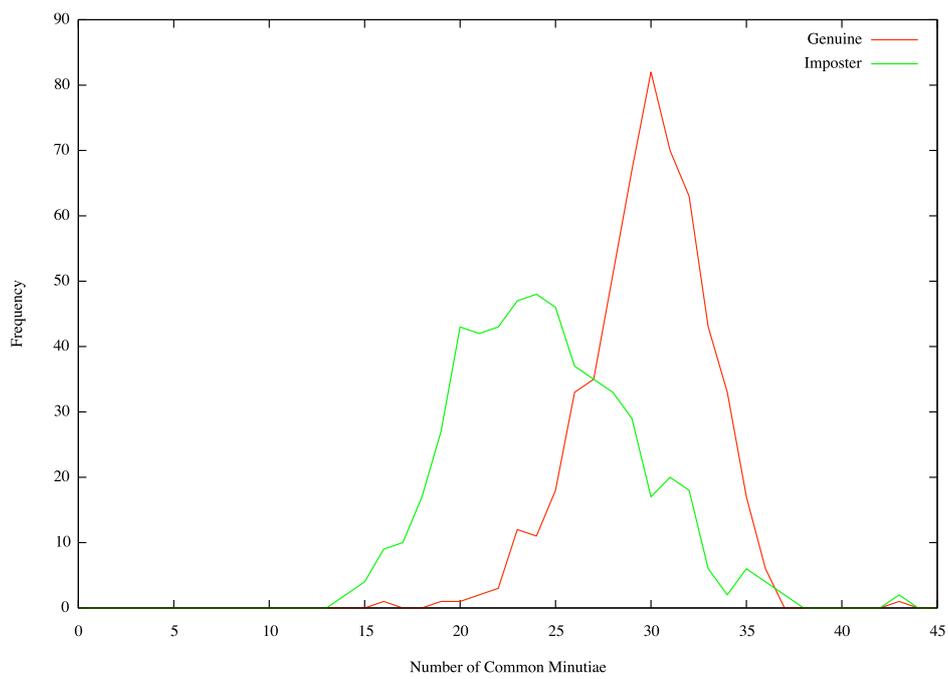


図 7: 共通マニューシャ数のヒストグラム

5 結論

マニューシャ間の距離比と角度を利用することで、画像の拡大、縮小にロバストな指紋マッチングアルゴリズムを提案した。また、この提案アルゴリズムの試験実装を行なった結果、実験1では閾値 $T_3 = 11$ の時に $FAR = 12\%$, $FRR = 13\%$ を得ることができ、実験2では閾値 $T_3 = 7$ の時に $FAR = 27\%$, $FRR = 29\%$ を得た。

今後の課題として、共通マニューシャ候補数と閾値の関係、パラメータの最適化が挙げられる。

6 夏期休暇における課題

一番正しい特徴点をとれるのは誰か？

6.1 概要

指紋のマッチングに，マニユーシャ間の距離比や角度などのマニユーシャの位置関係を利用した特徴点抽出アルゴリズムが Chandrasekaran らによって提案されている^[2]．これは，マニユーシャ間の距離比や角度を利用して $M(i)$ -tuple を作成し，その情報をもとに，マニユーシャ木を生成することで共通マニユーシャを抽出するものである．この方式を実装し，再現率，適合率を求めることで共通マニユーシャの抽出精度を評価すると同時に，その精度が最も高い人が誰なのかを見つけ出す．

6.2 実験方法

当研究室関係者 38 名の右手中指の指紋を 2 枚ずつ採取し 1 枚を BM 画像，一方を IM 画像として認証を行う．指紋画像を目視して本当に正しい共通マニユーシャを見つけ出し，文献^[2]を実装し，抽出した共通マニユーシャとどれだけ合致しているかを調査する．また，どれだけ間違っ て抽出したかを調べることで，共通マニユーシャの抽出精度を評価することができる．その指標が再現率と適合率となる．

文献^[2]では距離比と角度の許容誤差について具体的な数値が述べられていながったが，今回はこれらの数値を適当な値に設定した．

表 7: 環境，許容誤差

BM	1 枚 × 38 名
IM	1 枚 × 38 名
距離比誤差	0.2
角度誤差	0.2(Degree)

6.3 実行画面

プログラムの実行画面を図8に示す。指紋画像は元々白地だが、黒字の方が見やすいためプログラム内で色を変更している。図8において、赤い点が抽出した共通マニューシャ候補、黄色い点が抽出した共通マニューシャ、青い丸で囲われた点がプログラム実行後に目視で調べた共通マニューシャである。つまり青い丸で囲われた点は正しい共通マニューシャであり、黄色い点でなおかつ青い丸で囲われていればうまく抽出できているということになる。

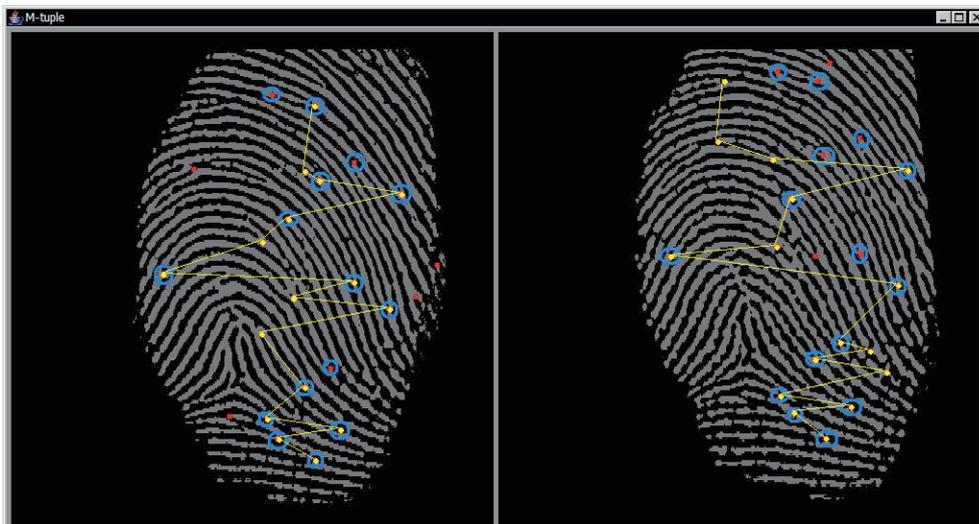


図 8: 実行画面

6.4 実験結果

再現率と適合率を求める。適合率を P 、再現率を R 、抽出した全てのマニューシャを N 、抽出した共通マニューシャを S 、目視で確認した共通マニューシャを S_m とすると、 P と R は以下の式で求められる。

$$P = \frac{S_m}{N}$$

$$R = \frac{S_m}{S}$$

表 8: スコア上位 10 名

順位	氏名	スコア
1	岡本 健太	71
2	白石 隆	68
3	後藤 裕太	67
3	小西 亨	67
5	田中 雄	60
6	甲田 博揮	59
6	磯崎 邦隆	59
8	菅 秀俊	58
8	石原 進一	58
8	内田 京祐	58

本実験で仕様した 76 枚の指紋画像全てにおいて P と R を求める。1 人 2 枚の画像があるので、1 人につき 2 つずつの P と R が得られる。それぞれ 2 つの平均値を P_a, R_a とする。最後にスコアとなる平均値 A を 5 式によって求め、それをもとに順位を決定する。以上の手順で求めた結果を表 8 に示す。

$$A = \frac{R_a P_a}{2} \quad (5)$$

6.5 考察

i 番目の共通マニューシャを抽出する時、 $i-1$ 番目のマニューシャが共通マニューシャであることが条件となっている。つまり、もし $i=0$ 番目のマニューシャ(始点)が共通マニューシャでない場合は、最初から間違っただ抽出をしていることになり精度が落ちてしまう。

今回の実験では全ての指紋において y 座標の一番小さいマニューシャが始点となっている。以上のことから始点を正確に抽出することが必要である。

参考文献

- [1] “NIST Fingerprint Image Software 2 (NFIS2)” .
<http://fingerprint.nist.gov/NFIS/>
- [2] A. Chandrasekaran, et al, “Fingerprint algorithm Based on Tree Comparison using Ratios of Relational Distances”, The 2nd Int. Conf. on Availability, Reliability and Security, pp. 273-280, 2007.

謝辞

本研究を遂行するにあたり，多くの方々より御指導，御激励を受け賜りました．特に，多大なるご指導を賜りました，東海大学情報理工学部情報メディア学科 菊池浩明教授に深甚なる感謝を申し上げます．

また，多大なる御指導を賜り，本研究を導いていただいた永井慧氏に深くお礼申し上げます．最後に，本研究に協力して下さった菊池研究室の皆さんに感謝の意を述べると共に，謝辞とさせていただきます．