

2008年度 卒業論文

基準指紋との参照ベクトルに基づく
生体認証方式

電子情報学部 情報メディア学科

河野 瞬 畔上 洋平

目次

1	はじめに	1
2	要素技術	2
2.1	NIST Fingerprint Image Software 2	2
2.2	Reference Vector	2
2.2.1	隆線ベクトルの作成	2
2.2.2	参照ベクトルの作成	4
3	研究に用いたプログラム（提案システム）	5
3.1	使用機材	5
3.2	システム概要	5
3.3	プログラム概要	6
4	実験	9
4.1	実験方法	9
4.1.1	単一生体情報での本人拒否率と他人受入率の算出	9
4.1.2	参照ベクトルでの本人拒否率と他人受入率の算出	9
4.2	評価	10
4.3	考察	10
5	指紋認証の実験～指が奏でる音楽会～	13
5.1	実験目的	13
5.2	使用した技術	15
5.3	実験環境	16
5.4	実験内容	16
5.4.1	実験方法	16
5.4.2	音色を変化させない Midi 再生のアルゴリズム	17
5.4.3	音色を変化させる Midi 再生のアルゴリズム	17
5.4.4	Midi 再生の例	17
5.5	実験結果	18
5.5.1	音色が同じ場合	18
5.5.2	音色が異なる場合	18

5.6	評価	19
5.7	考察	20
6	Bozorth3 を基にしたオリジナル指紋認証方式	21
6.1	概要	21
6.2	要素技術	21
6.2.1	マニューシャ	21
6.2.2	マニューシャマッチング	21
6.3	プログラム概要・実装	22
6.3.1	bozorth_Az.java	22
6.3.2	Readxyt メソッド	23
6.3.3	MkComparison メソッド	23
6.3.4	Sort メソッド	23
6.3.5	MkCompatibility メソッド	24
6.3.6	IANGLE180 メソッド	24
6.3.7	CheckTable メソッド	24
6.3.8	Line_Search メソッド	25
6.4	考察	26
6.5	結論	26
7	オリジナル指紋認証方式を利用したメールクライアントサーバ	28
7.1	概要	28
7.2	認証方式	29
7.2.1	SASL	29
7.2.2	メールの送受信方法	29
7.2.3	SASL を用いた POP3 認証メカニズム	31
7.2.4	認証メカニズムの命名	33
7.2.5	文字コード	33
7.2.6	防御システム	34
7.2.7	SASL を用いた SMTP 認証メカニズム	34
7.3	実行画面	34
7.4	動作環境	34
7.4.1	SASL を用いた POP3 認証メカニズム	34

7.4.2	操作方法	35
7.4.3	Yahoo メールサーバからメールを受信する方法 . . .	35
7.4.4	TERESA (noisy) メールサーバから生体認証でメールを受信する方法	35
7.5	考察	35
8	結論	40
9	付録：指紋認証入退室プログラム	41
9.1	概要	41
9.2	認証方式	41
9.3	実行画面	41
9.4	実行方法	41
9.5	考察	42

1 はじめに

現在、指紋認証技術には、リモートサーバ上で認証することから生じる生体情報のリスクが指摘されている。これに対して、生体情報直接使わないゼロ知識証明を用いた認証の試みがある。ここで認証時に、指紋情報を比べる際の代表的なマッチング方式として、マニューシャマッチング方式が挙げられる。これは指紋から得られる特徴点の情報を元に照合を行う方式である。指紋をスキャンする際に得られる特徴点の情報量は、スキャンの度に変動するが、このマッチング方式は特徴量が変化しても高い照合精度を誇っている。しかし、このマッチング方式を適応する場合、ゼロ知識証明への応用においてスキャンの度に特徴量が変動するという曖昧さは大きな問題点となっていた。

そこで本研究では、当初「安全なリモート指紋認証方式の研究」というテーマの元に、指紋から得られる情報を使った様々な応用方法や、安全なリモート指紋認証の方法を考えてきた。その中で指紋の中心部から得られる隆線方向の値をひとつのベクトルとし、 n 個の基準データとの類似度から成る新たな n 次元のベクトルとして認証を行う参照ベクトル方式を提案する。本提案方式は生体情報の次元が安定し、また参照ベクトルを生成する際に比較する基準データも人工的な指紋データを用いる事ができるためゼロ知識証明に適している。

提案方式の認証精度を調べるため、javaを用いて試験実装を行った。また参照ベクトルを生成する際の照合値の算出方法として既存のマニューシャマッチング方式でのスコアと、コサイン尺度による類似度の2通りで精度を求め、それぞれのFAR(他人受入率)とFRR(本人拒否率)の相関表グラフを作成した。

2 要素技術

2.1 NIST Fingerprint Image Software 2

NIST Fingerprint Image Software2(NFIS2)^[1]の mindtct パッケージを用いて指紋の特徴点や隆線の向きなど多数のデータを抽出することができる．今回は特徴点の座標や信頼度が書き出された min ファイルと xyt ファイル，指紋の隆線を 16 方向に量子化した値を書き出した dm ファイル，また隆線方向が急激に変化している部分を示す hcm ファイルの 4 種を主に使用した．

また同梱されているマニューシャマッチングアルゴリズムが実装された bozorth3 を用いて，マニューシャマッチングによるスコアを得た．これは mindtct により得られる特徴点情報が記述された xyt ファイルを元にマッチングを行い，類似度のスコアを出力する．

2.2 Reference Vector

指紋の中心部を任意の領域で抽出し，抽出部分の隆線方向の値をその指紋の隆線ベクトルとする．次に作成した隆線ベクトルを，用意してある n 個の基準データと類似度比較を行い，その照合値で新たな n 次元のベクトルを作る．このベクトルを参照ベクトルと定義する．この参照ベクトルは，基準データとの類似度の値を元に作成されたベクトルであり，入力指紋の情報は一切含まれていない．また，次元数は基準データの数を変動させることで任意に設定することができる．認証時には，この参照ベクトルの比較を行うことにより，指紋情報を秘匿した認証が行えるのではないかと予測している．

尚，本研究では 21×21 の領域で中心部を抽出し，441 次元の隆線ベクトルを作成し，比較に用いた．また比較用の基準データとして 45 個のデータを用意し研究を行った．

2.2.1 隆線ベクトルの作成

1. 中心点の算出

mindtct により得られる，hcm ファイルを元に中心点の算出を行う．

指紋の元画像を図1, hcm ファイルの一部の例を図2 に示す. 図1 の hcm ファイルの一部は, おおよそ元画像の白枠部分を示している. hcm ファイルは, 隆線方向が急激に変化している部分を1, それ以外の場所は0として示されていて, この情報を元に指紋の中心点を算出する. 算出方法は単純に1が現れるx座標とy座標を平均した座標である.

2. dm 値の抽出

mindtct により得られる, dm ファイルと 2.2.1 で算出した中心点の座標を元に任意の領域で dm ファイルから隆線方向の値を抽出する. 以後, この隆線方向の値を d m 値と呼ぶ. d m ファイル内に示されている d m 値の一部と, 対応する隆線方向の値を3と4に示す. d m 値は0~15の16段階で示された値で, 隆線角度が11.25°傾く度に d m 値は1増える. この d m 値を中心部のみ抽出する. 抽出した値を1行目から順に並べ, 隆線ベクトルを作成する. 図3の d m 値を例にすると1行目の値から, 一番下の行の値まで順に

$$A = (4, 4, 5, 5, 5, 6, 6, \dots, 13, 13, 13, 13, 13) \quad (1)$$

という隆線ベクトルが作成される.



図 1: 指紋の元画像

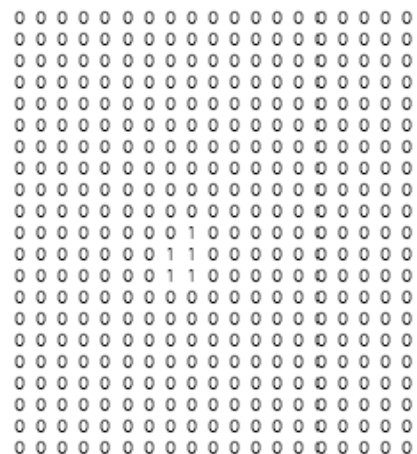


図 2: hcm ファイルの一部

```

4 4 5 5 5 6 6 7 8 8 9 10 10 11 11 12 12 12 13 1
4 4 4 5 5 5 6 7 7 8 9 10 10 11 11 12 12 13 13 1
4 4 4 4 5 5 6 6 7 8 9 10 11 11 11 12 12 13 13 1
3 3 4 4 4 5 5 6 7 8 9 10 11 11 12 12 13 13 13 1
3 3 3 4 4 5 5 6 7 8 9 10 11 12 12 13 13 13 13 1
3 3 3 3 4 4 5 5 6 7 8 10 11 12 12 13 13 13 13 1
3 3 3 3 3 4 4 5 5 7 8 9 11 12 12 13 13 13 13 1
3 3 3 3 3 3 4 4 5 6 8 9 11 12 12 13 13 13 13 1
2 2 2 2 3 3 3 4 4 5 7 9 11 12 13 13 13 13 13 1
2 2 2 2 2 2 3 3 4 5 7 9 11 12 13 13 13 13 13 1
2 2 2 2 2 2 2 3 3 4 6 12 12 13 13 13 13 13 13 1
2 2 2 2 2 2 2 2 2 2 0 13 13 13 13 13 13 13 13 1
2 2 2 2 2 2 2 2 1 1 15 14 13 13 13 13 13 13 13 1
2 2 2 2 2 2 2 1 1 0 15 14 13 13 13 13 13 13 13 1
2 2 2 2 2 2 1 1 0 0 15 14 13 13 13 13 13 13 13 1
2 2 2 2 2 1 1 0 0 15 15 14 14 13 13 13 13 13 13 1
2 2 2 1 1 0 0 15 15 15 14 14 14 13 13 13 13 13 13 1
2 2 1 1 0 0 15 15 15 14 14 14 13 13 13 13 13 13 1
1 1 1 0 0 0 15 15 15 14 14 14 13 13 13 13 13 13 1
1 1 0 0 0 15 15 14 14 14 14 13 13 13 13 13 13 13 1

```

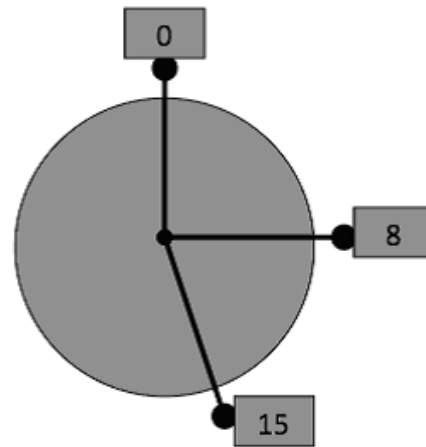


図 3: dm ファイルの一部

図 4: dm 値の隆線方向

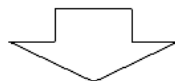
2.2.2 参照ベクトルの作成

入力データ b と、登録されている n 個の基準データ $a_1 \sim a_n$ との照合値で b 次元の参照ベクトル x を作成する。この時の入力データ b と基準データとは、コサイン尺度での類似度比較の場合は隆線ベクトル、既存のマニューシャマッチングでのスコアの場合は、mindtct により得られる xyt ファイルを指す。コサイン尺度の類似度比較については「2.3 参照ベクトルの比較」で説明する。

参照ベクトルを作り出すことにより、単一の生体情報の特徴ではなく、統計的な特徴として扱うことができる。

参照ベクトルの作成例を 5 に示す

	a_1	a_2		a_n
b	0.912	0.836		0.834



$$x = (0.912, 0.836 \dots \dots, 0.834)$$

図 5: 参照ベクトルの作成例

3 研究に用いたプログラム（提案システム）

3.1 使用機材

今回のシステムを実装するにあたって，表 1 のような機能を持った指紋リーダーを利用した．

表 1: 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

参照ベクトルを作り出す前過程として，d mファイルやh c mファイルの解析に用いたプログラムのシステムを紹介する．まず中心点算出の前段階として，h c mファイル内の値を元に，実際に画像のどのあたりの隆線方向が急激に変化しているかどうか，画像を出力して確認できるプログラムを作成した．1 が書かれている部分を白，それ以外の部分は黒として画像を出力するシステムである．d mファイルに関して，隆線方向の値を実際に描画し，隆線の流れが正しく書かれているかどうか検証するプログラムを作成した．これは 0～15 の値を実際に直線で表し，画像を出力するシステムである．

また隆線ベクトルでのコサイン尺度の類似度で，入力者の判別が行えるかどうか検証するプログラムを作成した．まず登録者の名前とその人の隆線ベクトルをテンプレートに登録する．次に U.are.U から指紋を入力し，そこから入力指紋の隆線ベクトルを作成する．最後に作成された隆線ベクトルとテンプレートの全隆線ベクトルとのコサイン類似度を求め，一番似ている隆線ベクトルの登録者の名前を画面に出力する．

3.3 プログラム概要

- LowFlowMap.java
h c mファイルの情報を入力とし，1の値が現れる場所は白，それ以外の場所は黒として，画像を出力する．尚，出力される画像はbmp形式である．h c mファイルについては，図2に示した通りである．
LowFlowMap.javaは，当初 mindtctにより得られる l f mファイルの画像化用プログラムとして開発したものであり，プログラム名はその名残である．h c mファイルも l f mファイルも中身は全て0と1の値で構成されているため応用が利いたため，そのまま利用した．出力された画像ファイルを ■ 図4に示す．
- DirectionMap.java
d mファイルの情報を入力とし，それぞれの値が示す向きを線で描画し，画像をbmp形式で出力する．d mファイルについては図3で示した通りである．元画像と出力された画像を図6と図7に示す．
- CenterPoint.java
h c mファイルを入力とし，1の値が現れるx座標とy座標をそれぞれ加算していき，その平均をとった座標を返す．
- DirAndCenter.java
LowFlowMap.javaとDirectionMap.javaとCenterPoint.javaの3つのプログラムを統合し，更に隆線ベクトルに用いる中心部の領域を抽出する．抽出した部分は赤枠でマーキングされ画像として出力される．h c mファイル，d mファイル，抽出する領域の範囲を入力とする．元画像と中心部がマーキングされた隆線画像を図8と図9に示す．
- LastCenter.java
DirAndCenter.javaに改良を加え，二つの拡張子なしのファイル名を入力とし，ファイル名に対応する各d mファイルとh c mファイルからそれぞれ隆線ベクトルを作成する．更に作成された隆線ベクトル同士のコサイン類似度を求め，その値を出力する．

- Biometrics_FinalVersion.java

GUI で上記のプログラムを全て統合し，指紋の入力と同時に，中心点の算出，及び 21×21 の中心部の領域を抽出し隆線ベクトルを作り出す．さらに画面には，抽出した部分がマーキングされた入力指紋の隆線画像と，元画像が表示される．検証ボタンを押すと，あらかじめ登録されている他の隆線ベクトルとコサイン尺度での比較を行い，最も類似した隆線ベクトルの登録者の名前を出力する．Biometrics_FinalVersion.java の実行画面を図 10 に示す．



図 6: 指紋の元画像

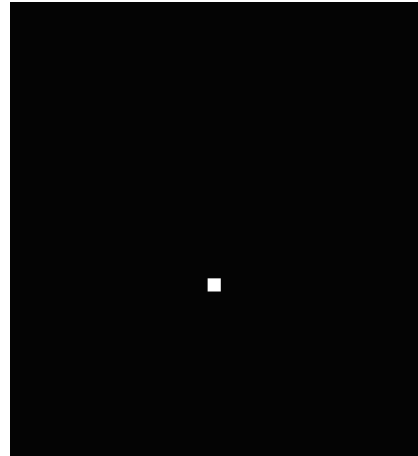


図 7: hcm ファイルの画像化



図 8: 指紋の元画像

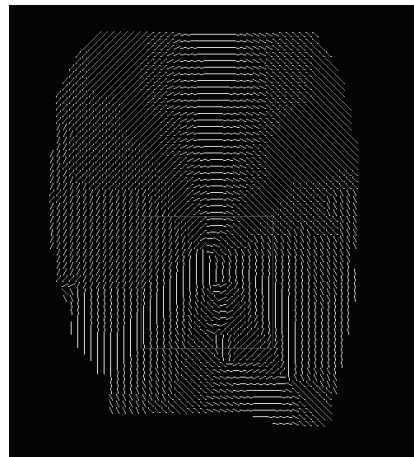


図 9: 中心をマーキングされた隆線画像

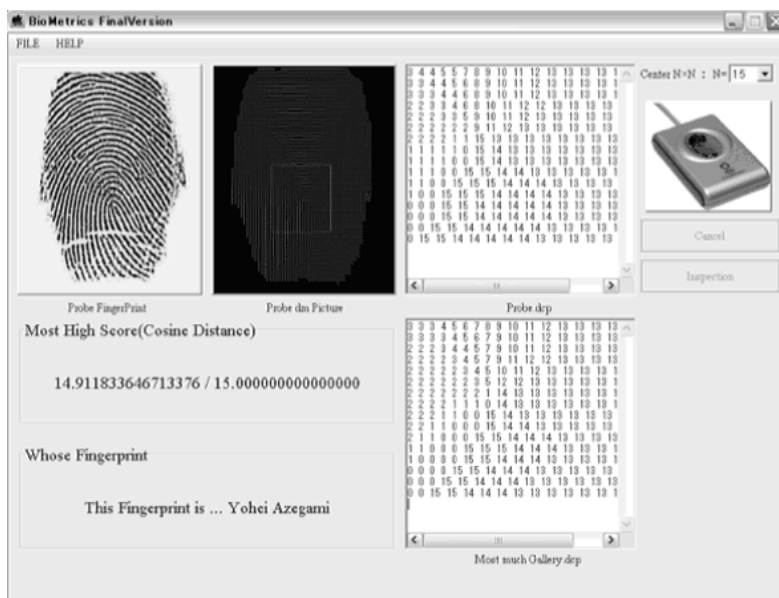


図 10: 参照ベクトルの作成例

4 実験

参照ベクトル方式の実用化において，単一の生体情報での認証精度と，参照ベクトルでの認証精度の比較を行い，精度の変動を考察する．また，その結果から参照ベクトル方式を適用する際の最適な閾値 τ を決定する必要がある．

4.1 実験方法

4.1.1 単一生体情報での本人拒否率と他人受入率の算出

FRR(本人拒否率): 被験者 4 人から指紋 100 個を取得し，各々の隆線ベクトルと xyt ファイルを得る．次に最初の入力指紋の隆線ベクトルと xyt ファイルをテンプレートとし，残り 99 個について，隆線ベクトルの場合はコサイン尺度での類似度比較を，x y t ファイルの場合は既存のマニユーシャマッチング方式でのスコアを計算する．最後に各比較値から 4 人分の平均 FRR を求める．

FAR(他人受入率): FRR の被験者とは違う，他人 36 人分の指紋をそれぞれ取得する．FRR 同様，隆線ベクトルと xyt ファイルを各々得て，他人同士の比較を行い FAR を求める．

4.1.2 参照ベクトルでの本人拒否率と他人受入率の算出

本人拒否率 (FRR) : 4.1.1 の実験と同じ 4 人の 100 個の指紋データを用い，各々の隆線ベクトルと xyt ファイルを得る．次に 100 個の入力指紋の隆線ベクトルと xyt ファイルを，本人以外の 45 個の基準データと，隆線ベクトルではコサイン類似度，xyt ファイルではマニユーシャマッチング方式でスコアを算出する．算出された基準データとのスコアからなる $n = 45$ 次元の参照ベクトル 100 個について，最初の参照ベクトルと，残り 99 個の参照ベクトルについて，コサイン尺度での類似度比較を行い，各比較値から 4 人分の平均 FRR を求める．

他人受入率 (FAR) : 4.1.1 の他人 36 人分の指紋データを用い，各々の隆線ベクトルと xyt ファイルを得る．FRR 同様に 45 個の基準データとの比

較を行い，参照ベクトルを 36 個作成する．最初の参照ベクトルと，残り 35 個の参照ベクトルについてコサイン尺度で比較を行い FRR を求める．

4.1.1 の隆線ベクトル方式とマニューシャマッチング方式で得た FRR と FAR を，それらを参照ベクトル化したときの FRR と FAR でそれぞれ相関グラフを作成する．

また各認証方式から Equal Error Rate(ERR) を求める．

4.2 評価

実験を行った結果，隆線ベクトル方式(ridge)とその参照ベクトル(ridge reference)を比べた相関グラフは図 11 のようになった．またマニューシャマッチング方式の方を比べた相関グラフは図 12 のようになった．隆線ベクトル方式について，予想では参照ベクトル化することによって精度が下がると考えていたが，他人同士と本人同士の比較値に大きな差が現れ，精度が 2.5 % 向上している．

マニューシャマッチング方式については参照ベクトル化することにより精度は 30 % 下がった．

また 4.1.1，4.1.2 の実験から各認証方式の ERR を求めたところ表 2 のようになった．

4.3 考察

今回の実験で，隆線ベクトル方式とその参照ベクトルを用いた方式での FRR と FAR はそれぞれ，閾値 $\tau = 0.8412$ において隆線ベクトル方式では $FRR = 1\%$ ， $FAR = 11\%$ ．参照ベクトル方式では，閾値 $\tau = 0.9825$ において $FRR = 0\%$ ， $FAR = 0\%$ という結果が得られた．またマニューシャマッチング方式においても，閾値 $\tau = 26$ において $FRR = 0\%$ ， $FAR = 0\%$ ．マニューシャマッチング参照ベクトル方式では，閾値 $\tau = 0.91$ において $FRR = 19.19\%$ ， $FAR = 36.11\%$ という結果が得られた．いずれにしても，参照ベクトル化することにより精度は下がることがわかり，精度が十分と言える結果ではない．しかしながら，参照ベクトルの次元数を増やしたり，基準データの中身を改変することにより精度はより向上させることができるのではないかと考えられる．

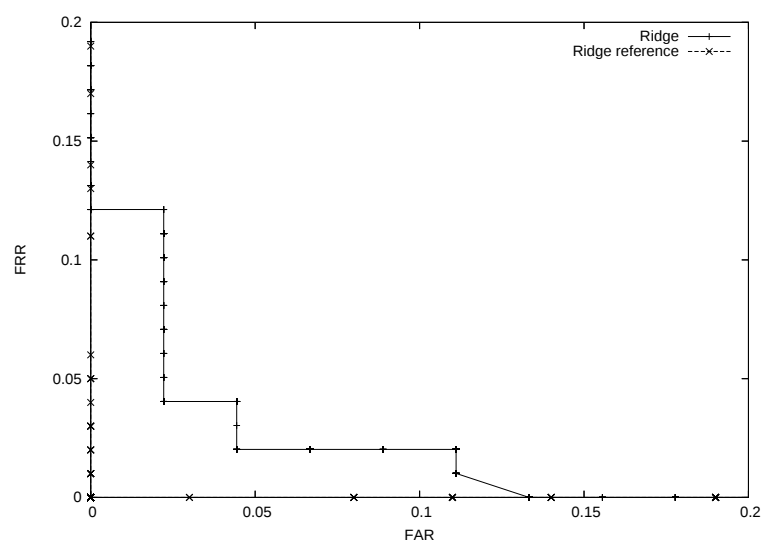


図 11: 隆線ベクトルとその参照ベクトルの認証精度

表 2: 各認証方式の EER

認証方式	EER
隆線ベクトル	4.4 %
隆線参照ベクトル	0 %
マニューシャ	0 %
マニューシャ参照ベクトル	30.0 %

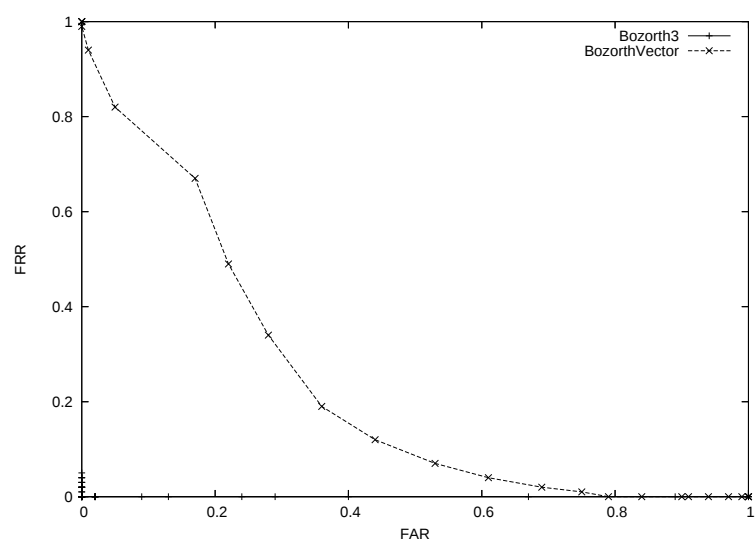


図 12: マニューシャマッチング方式とその参照ベクトルの認証精度

5 指紋認証の実験～指が奏でる音楽会～

5.1 実験目的

夏季休暇における課題として、以下の実験を行った。指紋から生成される XYT ファイルという生体情報の記載されたファイルは、一見何も規則性がないように思える。しかし、指紋のマニューシャの X 座標を X 軸に Y 座標を Y 軸にして同じ人の 2 回取得した指紋を折れ線グラフにして、比較してみると図 13 や図 14 のように似た線形が見て取れる。

更に図 15 や図 16 のように図 13 と図 14 の 10 区間の移動平均をそれぞれ求めることで、その特徴を顕著に現すことができる。これらの特徴を活かし、音楽に変換することで、同じ人の指紋を同じような音楽を奏で、本人と他人を識別できるようにしようということが目的である。また、これから到来するユビキタス社会の根本を支えることになる生体情報を身近なものに変換し、興味を持ってもらうことを目的とする。

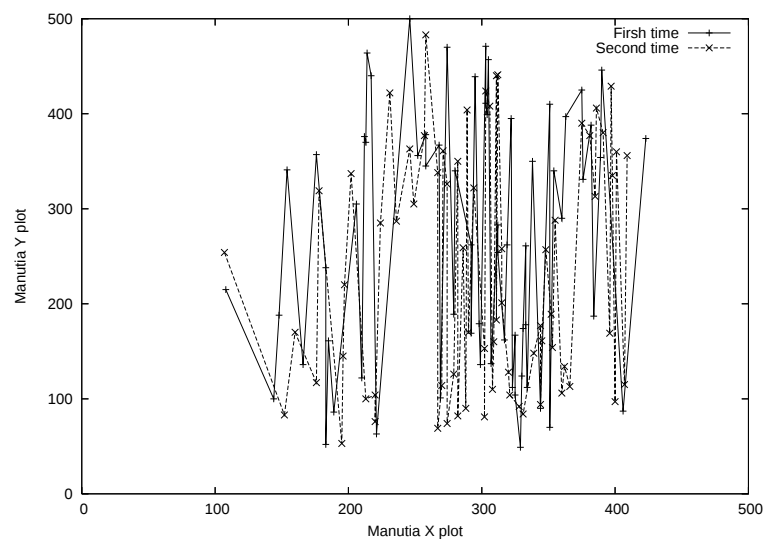


図 13: 畔上のマニューシャの折れ線グラフ

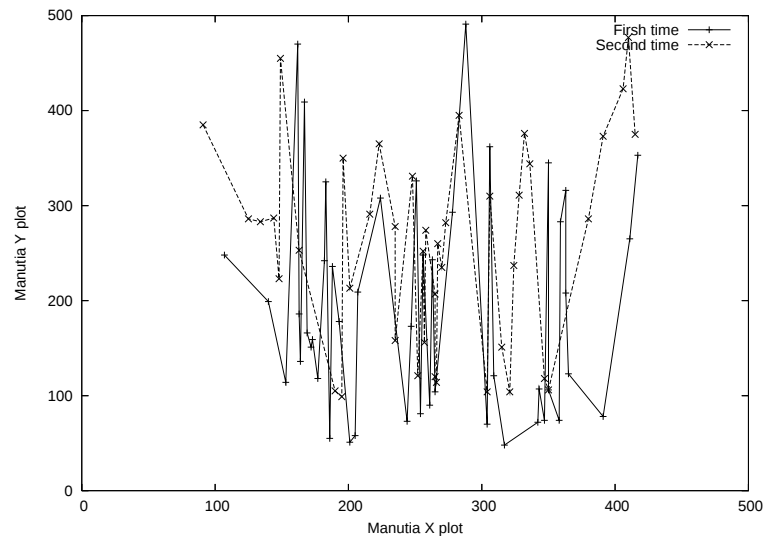


図 14: 菊池先生のマニューシャの折れ線グラフ

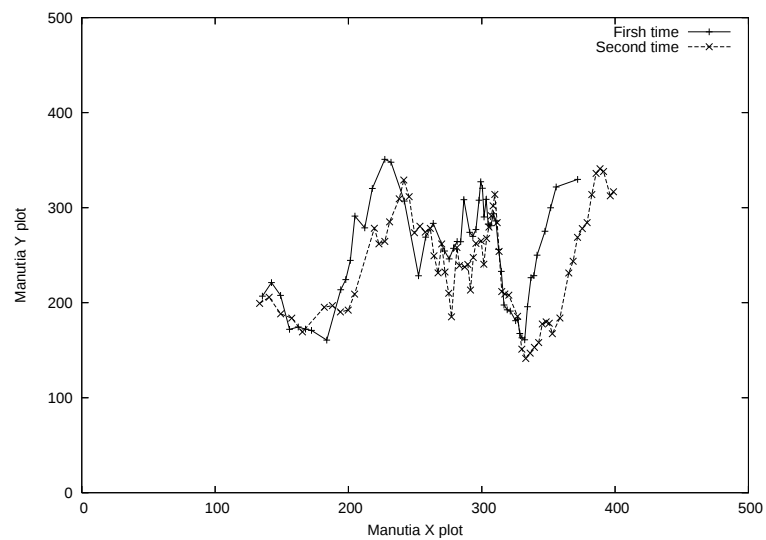


図 15: 図 13 の移動平均

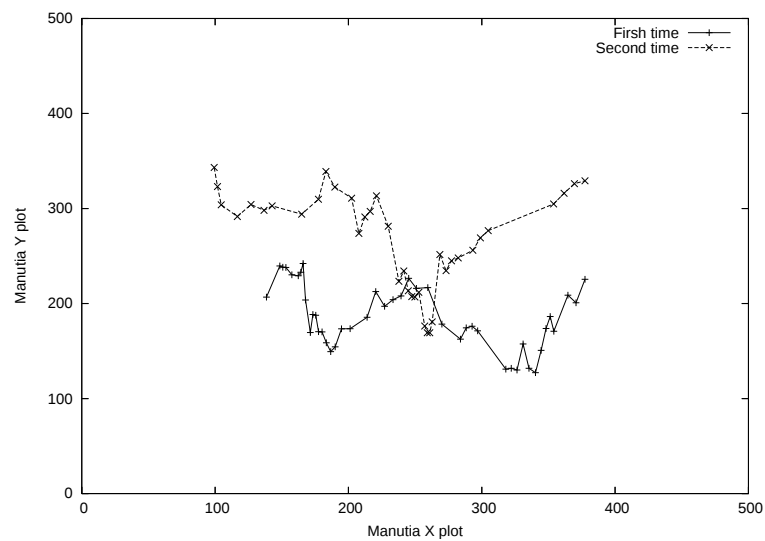


図 16: 図 14 の移動平均

5.2 使用した技術

- NFIS2(NSIT Fingerprint Image Software 2)内のパッケージ MINDTCT , Bozorth3
- 指紋画像から，指紋の特徴点を抽出するパッケージ．特徴点の座標等の情報を記録した「.xyt」という拡張子のテキストファイル(XYT ファイル)を出力する．
- XYT ファイルに記されている情報で，今回使用した情報の説明を以下に示す．

表 3: XYT ファイルで使した情報

X	1 枚 特徴点の X 座標
Y	1 枚 特徴点の Y 座標
T	特徴点の角度
REL	特徴点の信頼度 (0 ~ 100 で表される)

5.3 実験環境

以下のような環境で実験を行った．

表 4: 実験環境

被験者数	菊池研 15 名
XYT ファイル	15×10 個
マニユーシャ抽出	NFIS2(MINDTCT)
指紋リーダー	Digital Persona U.are.U4000
音再生プログラム	指紋奏で～る

5.4 実験内容

5.4.1 実験方法

1. 菊池研のメンバー 15 名のうち複数名の利き手人指し指の指紋を取得する
2. 指紋から生成される xyt ファイルを元に midi で音楽を奏でる
3. midi の作成の仕方は，xyt ファイルの X の値の差を音の長さにし，Y の値を音階のド～シに割り当てている．
4. 複数名のうちの 1 名の指紋の音楽を奏でて，誰の指紋かを音だけで当てる

表 5: XYT ファイルと Midi の関係

X	音の長さ
Y	7 つの音階に当てはめる
T	未使用
REL	Bozoth3 使用時に 40 以上を使用

5.4.2 音色を変化させないMidi再生のアルゴリズム

1. 指を入力する
2. 得られた指紋画像を Mindtct し、指紋のデータである XYT ファイルを生成する。
3. XYT ファイルのデータを元に midi 生成しながら再生する。生成する方法は、Y の値の最大が 550 であり、音階を 7 段階に分割したいので、 $550/7=79$ が音階 1 段階当たりの bozorth3 の値の量になる。その音階と bozorth3 の値の関係を表 6 に示す。

表 6: 音階と bozorth3 の値の関係

音階	Bozorth3 の値の範囲
シ	480 ~ 550
ラ	400 ~ 479
ソ	320 ~ 399
ファ	240 ~ 319
ミ	160 ~ 239
レ	80 ~ 159
ド	0 ~ 79

5.4.3 音色を変化させる Midi 再生のアルゴリズム

上記の 2 と 3 の間に次の処理を追加する。Mindtct して生成された XYT ファイルを bozorth3 にて判別し、予め登録されている人であれば、その人用に登録されている音色を決定する。登録されていない人である場合は、音色 1 に決定される。

5.4.4 Midi 再生の例

例えば、指紋から得られたマニューシャの XYT ファイルが表 7 であるとする。表 7 のマニューシャ A の X は 30 であり、マニューシャ B は 60 で

ある．そのため， $60-30=30$ がマニューシャA の音の長さになる．Y は 60 であり，表 6 の 0～79 であるので，音階はドと決まる．これを A～D まで行ったものを表 8 に示す．

表 7: XYT ファイルの例

マニューシャ	X	Y
A	30	60
B	60	80
C	70	490
D	90	320

表 8: 表 7 を音に変換した場合

マニューシャ	音の長さ	音階
A	$60 - 30 = 30$	ド
B	$70 - 60 = 10$	レ
C	$90 - 70 = 20$	シ
D	0	ソ

5.5 実験結果

5.5.1 音色が同じ場合

音色が同じ場合，人数毎に誰の指紋であることを判別した実験結果を表 9 に示す．判別に成功した場合を 1 とし，失敗した場合を 0 と定める．

5.5.2 音色が異なる場合

合宿の発表では，畔上，鈴木，石川の 3 名の音を河野が 1 度聞き，もう一度 3 名の内誰か一人の指紋の音を演奏し，誰の音であることを判別することに成功した．

表 9: 音色が同じ場合の実験結果

人数（人）	1 回	2 回目	3 回目	合計
2	1	0	1	2
3	0	1	0	1
4	1	0	1	1
5	0	0	1	1
6	0	0	0	0

音色が異なる場合，人数毎に誰の指紋であるかを判別した実験結果を表 10 に示す．

表 10: 音色が異なる場合の実験結果

人数（人）	1 回	2 回目	3 回目	合計
2	1	1	1	3
3	1	1	1	3
4	1	0	1	3
5	1	0	1	2
6	1	1	1	3

5.6 評価

実験 1 と実験 2 の結果を比較したグラフを図 17 に示す．図 17 から分かるように，音色が同じ場合に比べ，断然音色が異なるほうが成功し易い．表 9 の結果で成功した原因は，指紋を入力してもらった木野君の 1 音当たりの音の長さが長いという特徴が存在したためである．そのため，似ている音で判別したとは言えない．また，表 10 の人数が 5 人のときの 2 回目で失敗した理由は，松谷さんと水沼くんの音色が似ているため，どちらの音が判別できなかった．

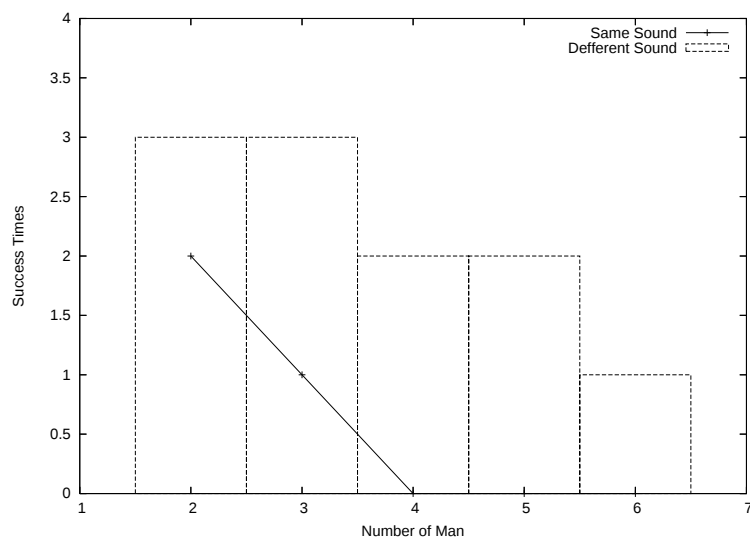


図 17: 音を奏でる人数と成功回数の推移

5.7 考察

今回の実験 1 の通り，人が異なる場合は異なる音色を奏でるという前提であれば，複数名のうち 1 名を当てることは可能である．しかし，それ以上の人数になると，当てる人の記憶力次第になる．また実験 2 から分かるように，音色が同じだと当てるのは非常に難しい．マニューシャの X 座標と Y 座標が似ているという関係性を活かした今回の実験であったが，単に音に変換しただけでは，似た音を奏でても人間がそれを似た音であると理解することは難しいことが分かった．しかし，bozorth3 の値を決めているネットワークの深さを利用して音を鳴らせることができれば，良い結果が生まれる可能性はある．

またこの少ない人数での認証を行う実験を基に，参照ベクトル方式が誕生した．

6 Bozorth3 を基にしたオリジナル指紋認証方式

6.1 概要

指紋認証の際に用いられる代表的なマッチング方式として、マニューシャマッチング方式が挙げられる。NIST Fingerprint Image Software2(NFIS2)^[1]のひとつに、マニューシャマッチングアルゴリズムを実装した bozorth3 というソフトがある。マニューシャマッチングの理解を深めるため、この bozorth3 について解析を行い、実際に java を用いてアルゴリズムの試験実装を行った。

6.2 要素技術

6.2.1 マニューシャ

指紋の隆線には、端点や分岐点が存在する。これらの点を総称してマニューシャと呼ぶ。マニューシャの例を図 19 に示す。NIST Fingerprint Image Software2(NFIS2)^[1] の、mindtct パッケージを用いることで、指紋画像からマニューシャを抽出し、その x,y 座標と隆線方向の角度、マニューシャの信頼度を表す値の 4 つが書かれた xyt ファイルを得ることができる。xyt ファイルの一部を図 18 に示す。

6.2.2 マニューシャマッチング

bozorth3 のマニューシャマッチング方式は大きく分けて 4 つのステップで成り立っている。

Step1 信頼性のあるマニューシャの抽出
マニューシャの信頼度は 0 ~ 100 の段階で示されていて、100 に近づくほど信頼度は高いと言える。xyt ファイルからマニューシャ情報を読み込む際に、信頼度の閾値を設け、入力側と比較側、それぞれ信頼度が閾値以上のマニューシャだけを読み込む。尚、bozorth3 では閾値を 40 以上として読み込んでいる。一般にマニューシャの数は平均 80 個程度あると考えられている。

Step2 Comparison Table 入力側マニューシャと比較側マニューシャそれぞれで、ComparisonTable(入力側) と ComparisonTable(比較側) を作成する。ComparisonTable はそれ

ぞれのマニューシャ情報を元に、全通りのペアを作成して、2点間の距離 d_{kj} と角度 θ 、また各マニューシャと2点間の線分がなす角 b_1, b_2 を求め記憶させたデータベースを意味する。ComparisonTable の作成時の d_{kj} , θ , 及び b_1, b_2 の求め方を図 21、実際の ComparisonTable の一部を図 20 に示す。Step3 Compatibility Table 作成された入力側マニューシャの ComparisonTable と、比較側マニューシャの ComparisonTable について、2点間の距離、および2点のマニューシャがなす角の3つのデータを比較する。比較した3つの値が設定された閾値以内であった場合、各マニューシャのインデックス番号と2点間の角度の差を保存する。この保存されたデータベースが CompatibilityTable である。CompatibilityTable の一部を図 22 に示す。Step4 対応するマニューシャペアを繋ぐ CompatibilityTable に保存されている関連付けされたマニューシャのインデックス情報を頼りに、対応するマニューシャ間を繋ぎ、繋がった数をスコアとして出力する。出力されたスコアが入力側データと比較側データの類似度となる。マニューシャ間を繋ぐ例を図 23 に示す。

このマッチング方式は、各マニューシャのペア間の距離と角度を元に比較を行っているため、比較する際、スキャンした各指紋画像の角度が異なっていたとしてもマッチングを行うことができる。

6.3 プログラム概要・実装

bozorth3 のアルゴリズムに習って、Java を用いて実装を行った。bozorth3 は C 言語で組まれていたが、ソースの解読が困難だったためブラックボックスとして出力情報を参考にアルゴリズムに沿って実装を行った。

6.3.1 bozorth_Az.java

入力側 xyt ファイルと、比較側 xyt ファイルを入力とし、各マニューシャ情報を配列 P 、及び配列 G にそれぞれ保存し、マニューシャマッチングを行う。マッチング後のスコアを結果として出力する。

89	275	247	16
93	252	247	17
96	342	236	17
96	218	247	18
103	161	247	16
125	448	225	18
131	69	259	15
134	251	247	96
135	483	225	16
139	154	247	98
139	50	270	14
142	265	56	98
142	134	67	92
144	418	225	94
144	169	67	97
158	307	56	95

図 18: xyt ファイルの一部

6.3.2 Readxyt メソッド

入力された xyt ファイルを，用意した二次元各配列 ProbeSt[] と GallerySt[] に保存する．保存する際に，信頼値の情報を元に判定を行い，信頼あるマニューシャのみを保存する．尚，本プログラム中では bozorth3 に習い信頼値 40 以上で設定した．

6.3.3 MkComparison メソッド

保存された各配列情報を元にそれぞれマニューシャペアを作り，マニューシャ間の距離，角度，マニューシャ間との線分までの角度を計算し，新たに用意した二次元配列，Pm[] と Gn[] にそれぞれ情報を保存する．

6.3.4 Sort メソッド

ペア間の情報が保存された，二次元配列の各 ComparisonTable，Pm[][] と Gn[][] のデータを，保存されているデータのひとつであるマニューシャ間の距離が短い順にソートを行う．尚，ソートにはバブルソートを使用



図 19: マニューシャの例

しているため，ソート手法の改善を行えばプログラムの処理速度はより向上すると思われる．

6.3.5 MkCompatibility メソッド

閾値 Δd , $\Delta b1$, $\Delta b2$ を設け，ソートされた各 ComparisonTable の情報を元にペア情報の比較を行っていく． Δd はマニューシャ間の距離の閾値であり， $\Delta b1$ はマニューシャペアのうち的一方のマニューシャが2点間の線分となす角度の閾， $\Delta b2$ は残りのもう一方についての閾値である．これら3つの閾値の条件をクリアした入力側 ComparisonTable のペア情報と，比較側 ComparisonTable のペア情報が，新たに用意された CompatibilityTable である二次元配列 Comp_Table[][] 内に保存される．

6.3.6 IANGLE180 メソッド

マニューシャのペア間の比較の際，角度に 360 度の値が現れた場合，0 度として扱うために用いる変換用のメソッドである．

6.3.7 CheckTable メソッド

作成された CompatibilityTable の情報を使ってマニューシャ間を繋ぐ処理を行う際に，繋がるかどうかのチェックを行うメソッドである．繋がっ

た場合 true を返し，また boolean 型の一次元配列 ScanLine[] で繋がった CompatibilityTable のインデックス番号と同じ配列番号部分を true としチェックしていく．このメソッドは次に説明する Line_Search メソッド内で繰り返し呼ばれるメソッドである．

6.3.8 Line_Search メソッド

Compatibility 内全ての情報についてインデックス順に CheckTable メソッドを呼び出し比較を行い，繋がった場合スコアを加算していく．CheckTable メソッドを繰り返し呼び出すことにより，繋がりが無くなるまで Compatibility 内のデータをサーチすることができる．また本プログラムでは繋がりが 300 以上あった場合，本人同士と推定し処理を中断することで若干の高速化を図っている．

これらの処理を行い，出力されるスコアを元に本人であるか否かを判定する．尚，本プログラムは，bozorth3 とは異なるスコアを示すプログラムになっているが，本人であると推定できるスコアとして 40 以上を設定している．

-----Comparison Table-----					
d_kj	b1	b2	k	j	$\Delta \theta$
97	206	386	1	2	93
16	-7	4	1	3	-120
117	207	207	1	4	94
167	20	222	1	5	-93
83	210	210	1	6	97
61	0	11	1	7	-113
186	14	36	1	8	-99
173	201	213	1	9	100
74	233	233	1	10	120
230	12	45	1	11	-101
151	209	221	1	12	108
97	26	206	2	1	-87
111	21	32	2	3	-92
20	212	212	2	4	99
264	22	224	2	5	-91
16	5	5	2	6	-108
154	16	27	2	7	-97
282	18	40	2	8	-95
77	210	222	2	9	109
46	-21	-21	2	10	-134
325	16	49	2	11	-97
63	233	245	2	12	132

図 20: ComparisonTable の一部

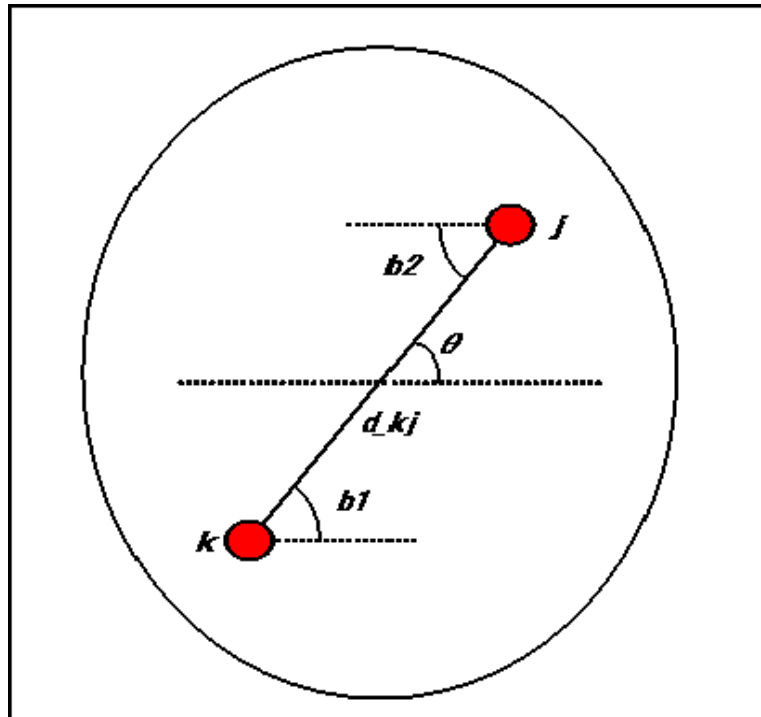


図 21: マニューシャペア間で求める各値の例

6.4 考察

本プログラムは精度実験を行っていないため、明確な精度を示すことができないが、試験的に何度か認証を行った結果、bozorth3 には遠く及ばない精度となってしまうていた。また本プログラムはbozorth3 と違い、指紋画像の回転に弱いことがわかった。原因としてマニューシャ間がなす角度の値を使用していないことにあると考えられる。しかしながら、本人か他人であるか判別した際、比較指紋によって差はあったが、判別成功率は高かったのではないかなと言える。

6.5 結論

マニューシャマッチングの理解を深めるために、bozorth3 の解析を行い、Java によって試験実装を試みた。その結果、bozorth3 には及ばない精度となってしまった。原因として回転角に対応できなかったことと、閾値

の設定の難しさが挙げられる．しかしながらアルゴリズムに沿った認証を行い，本人，他人を識別することには成功することはでき，マニューシャマッチングの理解を深めることはできたと思う．今後の課題として，マニューシャ間の比較の際の最適な閾値の設定や，マニューシャ間の角度を利用して回転角にも対応できるようなプログラムを追加することが課題である．

-----CompatibilityTable-----					
No	$\Delta \theta$	P(m)k	P(m)j	G(n)k	G(n)j

1:	6	3	52	5	51
2:	0	6	15	6	15
3:	13	6	43	4	46
4:	0	6	45	6	45
5:	0	6	60	6	60
6:	39	7	27	17	64
7:	29	8	27	2	17
8:	0	8	58	5	56
9:	8	10	63	6	60
10:	171	11	19	18	3
11:	7	12	43	1	24
12:	110	12	46	33	21
13:	15	14	37	19	56
14:	0	15	6	15	6
15:	0	15	40	15	40

図 22: Compatibility Table データの一部

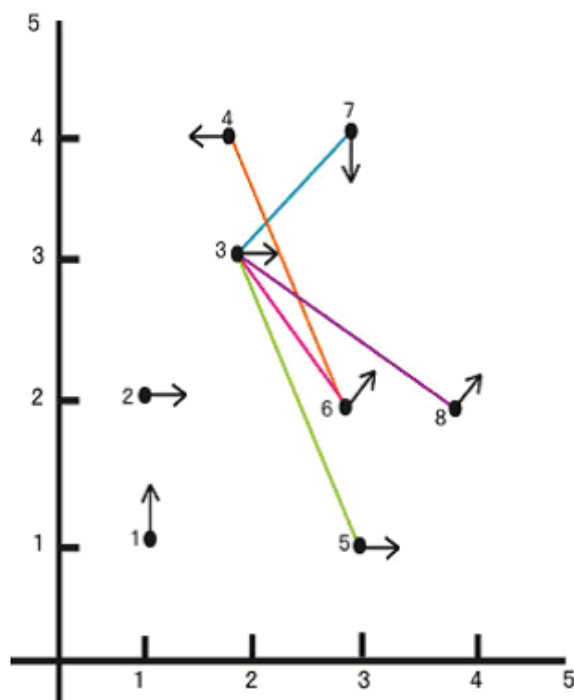


図 23: マニューシャ間を繋ぐ例

7 オリジナル指紋認証方式を利用したメールクライアントサーバ

7.1 概要

現在，日常生活でメールを活用している人は数多くいる．利用されている多くのメールクライアントとメールサーバの認証には，ID とパスワードを用いた認証が使われている．しかし，メールの受信や送信の度に ID とパスワードを入力する時間的な損失や利便性の欠如は否めない．また，ID 及びパスワードを紛失した場合の損害は膨大なものとなる．そこで我々は，利便性が高くまた紛失することの無い生体情報を用いた認証，所謂，生体認証に着目した．そして生体認証の中でも，とりわけ利用率の高い指紋認証に焦点を当てた．

代表的な既存の指紋のマッチング方式には，マニューシャマッチング方式が利用される．この方式は，指紋から得られるマニューシャの座標

や向きを利用してマッチングを行う方式である．我々は既存のマニユーシャマッチング方式の仕組みの理解と精度の操作を行うために，オリジナルのマニユーシャマッチング方式の開発を行った．

オリジナルのマニユーシャマッチング方式を用いて指紋認証を行い，メールを受信することができるメールクライアントとメールサーバを Java によって実装を行った．

7.2 認証方式

現在メールのプロトコルには，SMTP，POP3，IMAP が利用されている．我々の研究ではメールの受信をすることが主たる目的であるので，メールの受信に利用される POP3 を使用することにした．しかし，POP3 のままでは，生体認証を行うことはできないので，SASL の要求する範囲に於いて，POP3 に生体認証を行えるようにした POP3++ の開発を行った．

7.2.1 SASL

コネクションベースのプロトコルにユーザ認証機能や暗号化機能を提供するレイヤのことである．SMTP 認証（SMTP-AUTH）や APOP などで使われており，RFC 2222 で規定されている．SASL によってクライアントはサーバから認証を受けることができ，オプションでセキュリティ機構を利用することもできる．SASL は，多くの認証機構が提供されている．主なものには，ユーザ ID やパスワードを平文のまま送信する PLAIN や LOGIN，チャレンジ&レスポンス方式で認証を行う CRAMMD5，DIGEST-MD5，ケルベロス（Kerberos）v4，GSSAPI，S/Key などがある．SASL の認証機構を表 11 に示す．

7.2.2 メールを送受信する方法

メールの送受信の方法には，表 12 に示すように，平文を使用した送信と受信，暗号化した送信と受信に分けることができる．また，メールの送受信の方法を図式化したものを図 24 に示す．また，以下に各々の方式の説明を示す．

表 11: SASL の認証機構

平文使用	チャレンジ&レスポンス使用
PLAIN	CRAM-MD5
LOGIN	DIGEST-MD5
	Kerberos v4
	GSSAPI
	S/Key

- SMTP
インターネットやイントラネットで電子メールを送信するためのプロトコル。サーバ間でメールのやり取りをしたり、クライアントがサーバにメールを送信する際に用いられる。
- POP before SMTP
POP の認証を行ってから、メールの送信を行う方法である。
- SMTP-AUTH
SMTP に SASL の認証を加えたもの
- SMTP over SSL
SMTP を SSL 上で通信する方法
- POP3
電子メールを保存しているサーバからメールを受信するためのプロトコル。パスワードは平文でサーバに送られる
- APOP
POP に使われるパスワードを暗号化する認証方法。使い捨てパスワード（ワンタイム・パスワード）方式を使用
- POP over SSL
POP のパスワードを暗号化するプロトコル。APOP もあるが、APOP が暗号化するのはパスワードのみ。POP over SSL はメール本文も暗号化する。

- S/MIME

S/MIME は MIME を拡張したもので、電子メールの暗号化とデジタル署名を行う。公開鍵暗号方式には RSA を用いている点など、同様の暗号化方式である PGP とよく似ているが、S/MIME では、認証局がデジタル署名して発行した公開鍵証明書が必要である。

- PGP

「Web-of-trust (信用の輪)」の理念に基づき、別のユーザが自らの秘密鍵で署名できる

表 12: メールを送受信の方法

	平文使用	チャレンジ&レスポンス使用
送信	SMTP POP before SMTP	SMTP-AUTH SMTP over SSL S/MIME PGP
受信	POP3	APOP POP over SSL S/MIME PGP

7.2.3 SASL を用いた POP3 認証メカニズム

SASL を用いた場合、POP3 のサーバが AUTHORIZATION の状態でなくてはならない。AUTHORIZATION とは、認証中のことであり、メールボックスの操作をすることができない状態のことを指す。他にもメールボックスの操作が可能な TRANSACTION 状態や通信を終了するための UPDATE 状態が存在する。

サーバ側の SASL メカニズムによるリストの表示の仕方を以下に示す

1. “SASL” の文字列の後にスペースを用い、メカニズム名を記述する
2. SASL 接続を用いて認証を行えるようにする

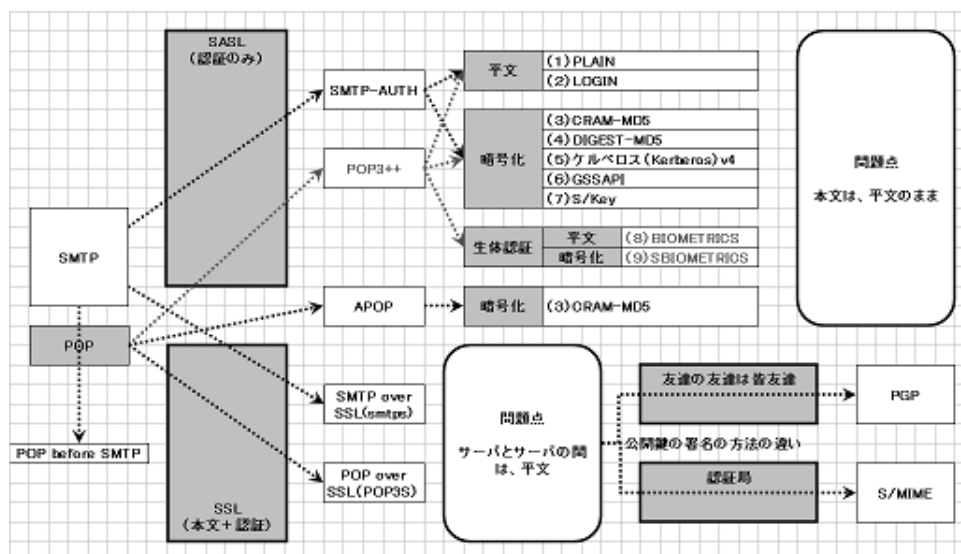


図 24: メールの送受信の方法を図式化

これらを最初の接続からのプロトコル例を表 13 に示す。S はサーバを C はクライアントを表す。もし、AUTH コマンドがサポート外の場合、サーバはクライアントへ“-ERR”を送る。クライアントが認証をキャンセルしたい時は、“*”を送る。“*”が送られてきたサーバは、“-ERR”を送る

SASL によると AUTH コマンドによって、サーバ側の返答を変化させなければならない。まとめたものを図 25 に示す

1. クライアントが STLS を選択した場合

表 13: 最初の接続からのプロトコル例

```

S : +OK pop.example.com BlurdyBlurp POP3 server ready
C : CAPA
S : +OK List of capabilities follows
S : SASL PLAIN DIGEST-MD5 GSSAPI ANONYMOUS
S : STLS
S : IMPLEMENTATION Blurdy Blurp POP3 server
S : .

```

STLS のコネクションを張った中で、更にもう一度認証のメカニズムを問う。その際は、STLS の表示をしないこと。ちなみに STLS は SSL/TLS のことである。TLS(Transport Layer Security) とは SSL3.0 をもとに若干の改良が加えられたもので、暗号化して送受信するプロトコルの 1 つ。

2. クライアントが PLAIN を選択した場合

PLAIN を選択した場合、2 つの可能性が考えられる。1 つ目は、PLAIN+「userid userid password」をクライアントが送信する場合。そして 2 つ目が、PLAIN のみを送信する場合である

3. クライアントが LOGIN を選択した場合

クライアントは AUTH LOGIN が送る。サーバ側は、「Username:」を Base64 に変換したものを送る。クライアントからは、UserID を Base64 に変換したものを送る。サーバ側は、「Password:」を Base64 に変換したものを送る。クライアントはパスワードを Base64 化した文字列を送る。サーバ側は、もしクライアントの ID とパスワードが一致していることを確認したらメールボックスをロックする

4. クライアントが DIGEST-MD5 を選択した場合

サーバ側は、「+」スペースの後に初期チャレンジを送る必要がある

7.2.4 認証メカニズムの命名

認証メカニズムを命名するには、ASCII 文字のうち、A~Z の大文字、0~9 の数字、ハイフン、アンダースコアを用いることができる。我々の研究では、生体情報を平文、若しくは暗号文を用いる事で認証を行う。そこで、平文で認証を行うメカニズムの名前を「BIOMETRICS」、暗号文で認証を行うメカニズムを「SBIOMETRICS」と名付ける。

7.2.5 文字コード

ID を送信する場合、必ず Unicode にエンコードしなければならないが、我々が作成するプログラムは、Unicode で文字列を扱う Java 言語で作成したので、特に変換する必要はない

7.2.6 防御システム

以下の攻撃に対して防御できるべきである

1. ハイジャック攻撃

成りすまし的一种でクライアントとサーバの正規セッションに割り込んで、セッションを奪い取る行為

2. クライアントが PLAIN を選択した場合

以前の通信内容を入手して、同じ内容を送信することによって、「なりすまし」などを行う攻撃

7.2.7 SASL を用いた SMTP 認証メカニズム

Telnet を用いて smtp サーバと接続する場合には、

open サーバ名 ポート番号

と打ち込む必要がある。以下は、実験で実際に使用したコマンドである。

Open smtp.mail.yahoo.co.jp 587

図 26 に実際にキーボードで SASL による認証が施されているサーバで認証し、メールを送ってみた様子を示す。また、図 27 に図 26 のサーバとクライアント間のプロトコルのやり取りを図式化したものを示す。

7.3 実行画面

図 28 に java で作成したメールクライアントソフト“Alicia”を示す。

7.4 動作環境

7.4.1 SASL を用いた POP3 認証メカニズム

表 14 にメールクライアントの動作環境を示す。

表 14: 動作環境

容量	200M 以上
デバイス	U.are.U
OS	WindowsXP

7.4.2 操作方法

1. FingerPrintMailer.class を実行
2. メールクライアントの画面が表示
図 29 に表示された画面の解説をする

7.4.3 Yahoo メールサーバからメールを受信する方法

各種設定ボタンを押し、「POP3 サーバ名」と「POP3 ポート番号」を以下のように変更し、設定ボタンを押し、後はメール受信ボタン押すことで、メールを受信することができる。

7.4.4 TERESA (noisy) メールサーバから生体認証でメールを受信する方法

SSH で noisy にアクセスし、FingerPrintServer を起動する。FingerPrint-Client 側で、「POP3 サーバ名」と「POP3 ポート番号」を noisy.cs.dm.u-tokai.ac.jp と 7777 に変更する。その後、生体認証でメール受信ボタンを押し、U.are.U に自分の指紋を置くことで、メールを受信することができる。

7.5 考察

ID とパスワードを使用せずに、指紋をデバイスに置くだけでメールを受信することに成功した。このことにより、ID やパスワードの紛失をすることがなく、またキーボードを使用することを必要としない。したがって、指紋の読み取り装置さえ備えてあれば、パソコンに不慣れな人であってもメールを見ることが可能になった。

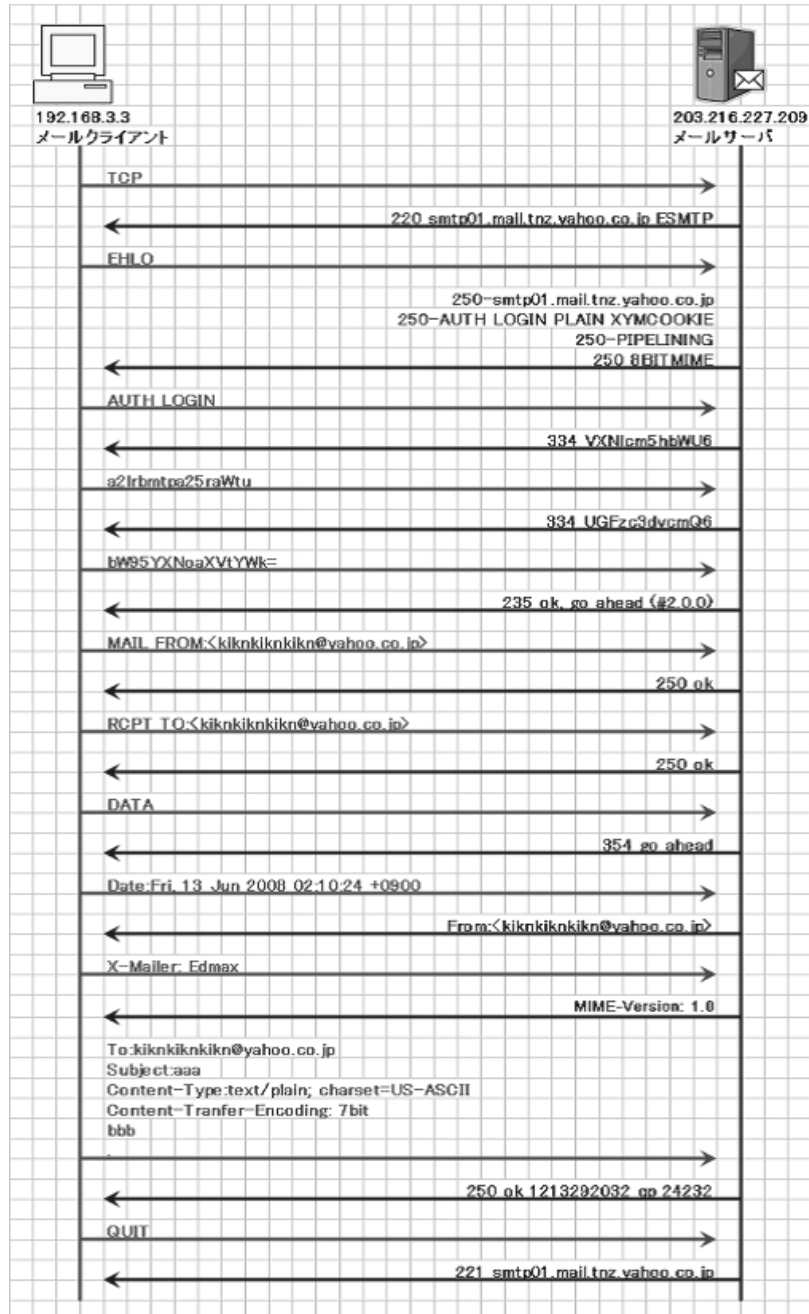


図 25: メールの送受信の方法を図式化

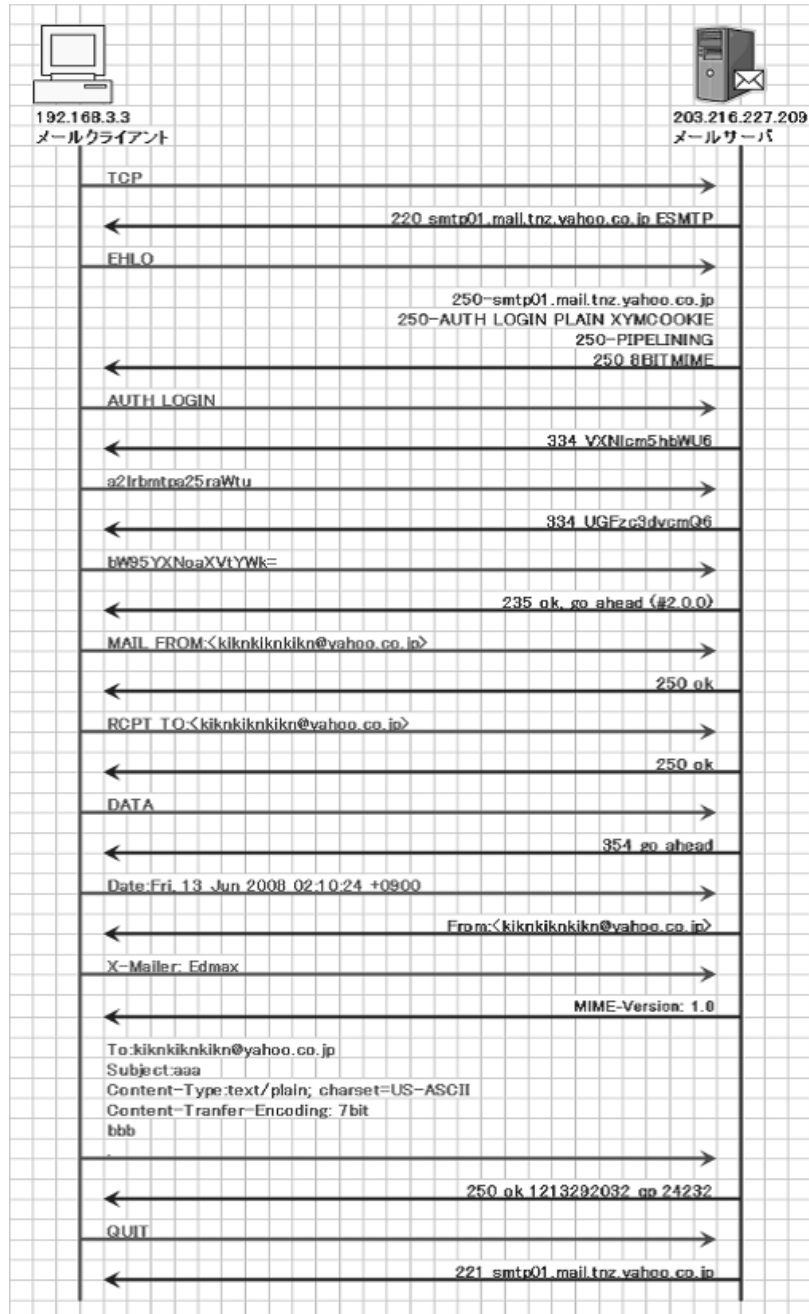


図 27: 図 26 を図式化したもの



図 28: Alicia

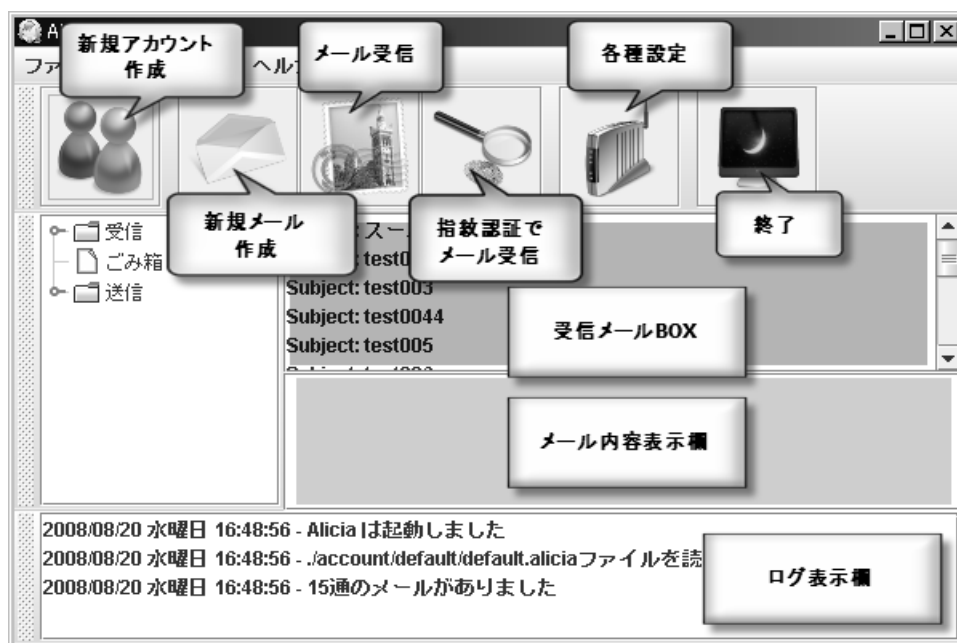


図 29: メールクライアントの画面の解説

8 結論

指紋から音楽を作成し，限られた人数での認証を基に，この方式を考え出された．

ゼロ知識証明への適応のため，単一の生体情報を用いるのではなく，基準データから得られる類似度を基にした統計的情報を用い，次元の安定化と生体情報の秘匿が行える参照ベクトル方式を提案した．また参照ベクトル方式と単一の生体情報での FRR と FAR を算出した結果，実験より，隆線ベクトル方式では閾値 $\tau = 0.8412$ で，FRR = 1 %，FAR = 11 %．参照ベクトル化した場合では閾値 $\tau = 0.9825$ で，FRR = 0 %，FAR = 0 %を得ることができた．マニユーシャマッチング方式においても，単一の生体情報の場合，閾値 $\tau = 26$ で，FRR = 0 %，FAR = 0 %．参照ベクトル化した場合，閾値 $\tau = 0.91$ で FRR = 19.19 %，FAR = 36.11 %となり，どちらの方式においても参照ベクトルにすることにより精度が下がることがわかった．

今後の課題として，十分な精度が得られる最適な基準データを人工的に作り上げることが挙げられる．また，この方式を使ったメールの受信及び送信を行うこともできるだろう．

9 付録：指紋認証入退室プログラム

9.1 概要

我々の研究室で使用されていた，入室管理を担う静脈認証が頻繁に動かないという事態が発生したため，それに代わる生体認証を用いて入室管理を行うアプリケーション開発をした．また，指紋読み込みデバイスが頻繁に落ち，アプリケーションが止まってしまうという深刻な事態に直面していたため，その原因と耐久性を知るために，このアプリケーションの開発を行った．

9.2 認証方式

Bozorth3 のマニユーシャマッチング方式を利用する．研究室のメンバーから取得した指紋をテンプレートとして，サーバに登録しておき，入室する際に入力された指紋とのスコアが最も高い人マッチングされる．

9.3 実行画面

指紋認証ログインが起動した実行画面を図 30 に示す

9.4 実行方法

FingerPrintLogin.class を実行することでログインクライアントを起動することができる．ログインサーバ側では，FingerPrintSaveLogin.class を実行しておく．クライアントが起動したら，指紋読み取り装置である U.are.U に指を置くことで，表示が退室状態であれば入室表示に変化する．また，入室状態の表示であれば，退室状態の表示に変化する．また，このプログラムには，図 3 . 1 の右下の表示によって，U.are.U の状態を表示する．U.are.U には以下の API の定数が存在する．

- DpFpEventHandler.EVT_VERIFICATION_CANCELED
- DpFpEventHandler.EVT_REGISTRATION_CANCELED

- DpFpEventHandler.EVT_TEMPLATE_ACQUISITION_CANCELED
- DpFpEventHandler.EVT_REGISTRATION_ERROR
- DpFpEventHandler.EVT_IDENTIFICATION_CANCELED
- DpFpEventHandler.EVT_TEMPLATE_EXTRACTED
- DpFpEventHandler.EVT_IDENTIFICATION_ERROR
- DpFpEventHandler.EVT_IMAGE_RECEIVED
- DpFpEventHandler.EVT_REGISTRATION_COMPLETE

9.5 考察

API 定数の中で、指紋がデバイスに置かれ指紋画像を取得する場合には、DpFpEventHandler.EVT_IMAGE_RECEIVED の状態である必要があることが分かった。また、DpFpEventHandler.EVT_REGISTRATION_COMPLETE の状態にあるときは、U.are.U に指紋を置いて、反応しないことも判明。更に、DpFpEventHandler.EVT_REGISTRATION_CANCELED、DpFpEventHandler.EVT_REGISTRATION_ERROR、DpFpEventHandler.EVT_TEMPLATE_EXTRACTED の状態のときに U.are.U がエラーを出し、アプリケーションが落ちてしまうことが判明した。

MSHARE ファイル デバッグ ヘルプ		
名義 / I D 直近の入室時刻 直近の退室時刻 本日累計入室時間 経累計入室時間 入室時間ランク : 岡地浩明 / okun : 2008/01/01 00:00:00 : 2008/00/05 05:05:18 : 0:00:00.000 : 57:37:38.611 : 4位	名義 / I D 直近の入室時刻 直近の退室時刻 本日累計入室時間 経累計入室時間 入室時間ランク : 本間智史 / soday : 2008/00/05 05:05:18 : 2008/00/07 00:20:00 : 0:00:00.000 : 57:39:18.611 : 3位	名義 / I D 直近の入室時刻 直近の退室時刻 本日累計入室時間 経累計入室時間 入室時間ランク : 木澤寛厚 / kazurathie : 2008/00/05 05:05:18 : 2008/00/07 00:20:00 : 0:00:00.000 : 57:44:40.895 : 6位
名義 / I D 直近の入室時刻 直近の退室時刻 本日累計入室時間 経累計入室時間 入室時間ランク : 熊崎彰雄 / hokoten : 2008/00/09 22:40:18 : 2008/00/09 22:46:28 : 0:00:00.000 : 57:06:20.895 : 5位	名義 / I D 直近の入室時刻 直近の退室時刻 本日累計入室時間 経累計入室時間 入室時間ランク : 陳苗 / miso : 2008/00/05 05:05:16 : 2008/00/05 05:05:16 : 0:00:00.000 : 56:59:40.895 : 7位	名義 / I D 直近の入室時刻 直近の退室時刻 本日累計入室時間 経累計入室時間 入室時間ランク : 松谷重昭 / hupin3 : 2008/00/05 05:05:16 : 2008/00/05 05:05:16 : 0:00:00.000 : 56:59:40.895 : 7位
名義 / I D 直近の入室時刻 直近の退室時刻 本日累計入室時間 経累計入室時間 入室時間ランク : 本野誠大 / ace4210 : 2008/00/05 05:05:18 : 2008/00/05 05:05:18 : 0:00:00.000 : 56:59:40.895 : 7位	名義 / I D 直近の入室時刻 直近の退室時刻 本日累計入室時間 経累計入室時間 入室時間ランク : 鈴木集人 / hokub810 : 2008/00/05 05:05:18 : 2008/00/07 00:20:00 : 0:00:00.000 : 56:59:40.895 : 7位	名義 / I D 直近の入室時刻 直近の退室時刻 本日累計入室時間 経累計入室時間 入室時間ランク : 河野崎 / claymore : 2008/00/10 05:54:54 : 2008/00/10 05:54:54 : 2008/00/10 05:54:54 : 27:40:02:16.892 : 4位
名義 / I D 直近の入室時刻 直近の退室時刻 本日累計入室時間 経累計入室時間 入室時間ランク : 窪田元 / geyama : 2008/00/05 05:05:16 : 2008/00/05 05:05:16 : 0:00:00.000 : 56:59:40.895 : 7位	名義 / I D 直近の入室時刻 直近の退室時刻 本日累計入室時間 経累計入室時間 入室時間ランク : 野上洋平 / moyoshi_yohs : 2008/00/05 05:05:16 : 2008/00/05 05:05:16 : 0:00:00.000 : 56:59:40.895 : 7位	名義 / I D 直近の入室時刻 直近の退室時刻 本日累計入室時間 経累計入室時間 入室時間ランク : 香川大介 / nagomih03 : 2008/00/05 05:05:16 : 2008/00/05 05:05:16 : 0:00:00.000 : 56:59:40.895 : 7位
名義 / I D 直近の入室時刻 直近の退室時刻 本日累計入室時間 経累計入室時間 入室時間ランク : 水沼輝 / noma-2u : 2008/00/10 05:55:15 : 2008/00/10 05:55:21 : 0:19:30.374 : 57:55:28.905 : 2位	名義 / I D 直近の入室時刻 直近の退室時刻 本日累計入室時間 経累計入室時間 入室時間ランク : 松尾峻彦 / azuma : 2008/00/09 22:56:00 : 2008/00/10 04:30:59 : 0:00:00.000 : 56:59:40.895 : 7位	名義 / I D 直近の入室時刻 直近の退室時刻 本日累計入室時間 経累計入室時間 入室時間ランク : 石川建博 / tam : 2008/00/05 05:05:18 : 2008/00/05 05:05:18 : 0:00:00.000 : 56:59:40.895 : 7位
名義 / I D 直近の入室時刻 直近の退室時刻 本日累計入室時間 経累計入室時間 入室時間ランク : 鈴木孝彰 / turbotom : 2008/00/05 05:05:16 : 2008/00/07 00:20:00 : 0:00:00.000 : 56:59:40.895 : 7位	名義 / I D 直近の入室時刻 直近の退室時刻 本日累計入室時間 経累計入室時間 入室時間ランク : 原生利直 / uriborn : 2008/00/05 05:05:16 : 2008/00/05 05:05:16 : 0:00:00.000 : 56:59:40.895 : 7位	
本日の検察当番は ・ 梅上 ・ 石川 です	ただ今 2008/12/31 00:21:00	

図 30: 実行画面

参考文献

- [1] “NIST Fingerprint Image Software 2 (NFIS2)”,
<http://fingerprint.nist.gov/NFIS/>
- [2] 菊池，河野，畔上，西垣，尾形：“直交基底指紋への参照度を特徴量とした安全な生体認証プロトコル”，暗号と情報セキュリティシンポジウム SCIS2009（発表予定），2009
- [3] 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.
- [4] “Java で MidiFile を作る”，<http://www.cwbp.com/mdpw/java/file.html>
- [5] 永井，菊池，尾形，西垣，秘密計算に適する距離の使用によるリモート生体認証，暗号と情報セキュリティシンポジウム SCIS 2008，2008．
- [6] 永井慧，菊池浩明，尾形わかは，西垣正勝，“ZeroBio 秘密ニューラルネットワーク評価を用いた指紋認証システム”，情報処理学会論文誌，Vol.48，No.7，pp.2307-2318，2007．
- [7] ”java における base64 の性能テスト“，
<http://trasis.jp/blog/lab/2008/03/javabase64.html>
- [8] ” エンコーディング - encoding（国際化/日本語化）“，
<http://javafaq.jp/S146.html>
- [9] ” メールにかけられた呪文「MIME～後編」“，
<http://www.atmarkit.co.jp/fnetwork/rensai/netpro04/netpro01.html>
- [10] ” みんなのプログラミング無料講座 Java.Swing Lesson7“，
http://www015.upp.so-net.ne.jp/heyjude/java_swing007.htm
- [11] ”JFC/Swing 入門“，http://www.ongs.co.jp/publications/articles/JAVAPRESS/Vol.14/index_print.html

- [12] ” マ ル チ ス レッド・プ ロ グ ラ ミ ン グ の 基 本 “ ,
http://www.itarchitect.jp/technology_and_programming/-/10795-4.html
- [13] ” ロ グ ア ウ ト し た 後 も コ マ ン ド を 実 行 し 続 け る “ ,
<http://itpro.nikkeibp.co.jp/article/COLUMN/20060227/230850/>
- [14] ”UNIX の 部 屋 コ マ ン ド 検 索: tail“ , <http://x68000.q-e-d.net/68user/unix/pickup?tail>
- [15] ”POP (Post Office Protocol)“ , https://www.cs.dm.u-tokai.ac.jp/internal/share/Biometrics/2008_FP/080605/pop.pdf
- [16] ” R F C 日 本 語 版 リ ス ト “ , <http://www5d.biglobe.ne.jp/stssk/rfcjlist.html>
- [17] ”Index of rfcj“ , <http://jbpe.tripod.com/rfcj/>
- [18] ”SASL (シ ン プ ル 認 証 と セ キ ュ リ テ ィ 層) “ ,
<http://www.ipa.go.jp/security/rfc/RFC2222JA.html>
- [19] ”SMTP-AUTH (SASL) “ , <http://www.club.kyutech.ac.jp/support/manual/smtpauth.html>
- [20] ” 指 紋 が キ ー と な る 金 庫 ”Indexed Fuzzy Vault“ の 開 発 “ , 2006 年 度 卒 業 論 文
- [21] ” 付 録 1 NIST 指 紋 照 合 公 開 ソ フ ト ウ ェ ア “ ,
<http://www.nmda.or.jp/nmda/bio/pdf/2006part4-1.pdf>

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，本研究を推進するにあたって，御親切なる御教示ならびに御激励を賜りました，東京工業大学イノベーションマネジメント研究科尾形わかは准教授，静岡大学創造科学技術大学院西垣正勝准教授に厚く御礼申し上げます．

また，活発な議論，技術的なご教示を頂いた ZeroBIO 研究プロジェクト研究員一同に深く感謝を申し上げます．

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．