

特異点に基づくロバストな指紋分類アルゴリズム

A Robust Fingerprint Classification Algorithm Based on Singular Points

伊藤 康一[†] 森田 歩 早坂 昭裕 青木 孝文
東北大学大学院情報科学研究科
〒 980-8579 仙台市青葉区荒巻字青葉 05
Tel: 022-217-7169, Fax: 022-263-9308
E-mail: ito@aoki.ecei.tohoku.ac.jp
[†] 日本学術振興会特別研究員

Koichi ITO[†], Ayumi MORITA, Akihiro HAYASAKA, and Takafumi AOKI
Graduate School of Information Sciences, Tohoku University,
Aoba-yama 05, Sendai 980-8579, Japan
Tel: +81-22-217-7169, Fax: +81-22-263-9308
E-mail: ito@aoki.ecei.tohoku.ac.jp
[†] Research Fellow of the Japan Society for the Promotion of Science

あらまし 本稿では、撮像状態の悪い指紋画像を指紋の特異点に基づいて Arch, Left Loop, Right Loop, Whorl の 4 種類に分類するアルゴリズムを提案する。大規模な指紋データベースに対して照合を行う場合に、登録されているすべての指紋画像と照合すると膨大な時間が必要となる。そのため、データベースに指紋画像を登録する時に、指紋の種類によってあらかじめ分類する必要がある。本稿では、撮像状態の悪い指紋画像も分類することができるように、指紋全体の特徴だけではなく、部分的な特徴からも指紋を分類できるアルゴリズムを提案し、FVC2002 で使用された指紋データベースを使って評価した結果について述べる。

1 まえがき

ユビキタス情報化社会において、セキュリティは、最も重要なキーワードの 1 つであり、より高い信頼性を持っていて、かつ利便性に優れたシステムが必要とされている。現在までによく用いられている認証方法は、パスワード、PIN (Personal Identification Number)、鍵などの記憶や所持を必要とするものである。そのため、「忘れる」、「盗まれる」、「なくす」という危険性がある。これらの問題に対し、記憶や所持が不要で、さらに時間がたっても変化することのない究極的な認証方法とし

て、個人の身体的・行動的特徴をそのまま認証情報として利用するバイオメトリクスが注目されている。個人認証システムにおいて利用されるバイオメトリクスには、指紋・網膜・虹彩・筆跡などがある。これらの中で、指紋を用いた認証システムは、その利便性より最も研究開発・実用化されている [1]。

近年、ネットワークを介した大規模な指紋データベースを使って個人認証を行うシステムが開発されている。このような大規模なデータベースで効率的に照合するためには、登録時にあらかじめ指紋を分類する必要がある。現在までに、さまざまな指紋分類アルゴリズムが研究・開発されているが、ほとんどの場合は、指紋の特異点 (loop と delta) の位置関係を用いたアルゴリズムである [2], [3]。これに対し、本稿では、撮像状態が悪い指紋画像でも部分的な特徴を利用することで高精度に分類することができるアルゴリズムを提案する。提案するアルゴリズムは、FVC (Fingerprint Verification Competition) 2002 [4] で使用された撮像状態の悪い指紋データベースを使って評価した。

2 指紋分類アルゴリズム

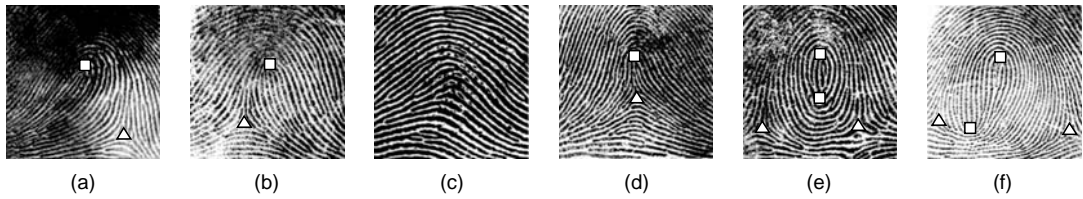


図 1: 指紋の種類: (a) Left Loop, (b) Right Loop, (c) Arch, (d) Tented Arch, (e) Whorl, (f) Whorl (Twin Loop) (□ は loop, △ は delta である)

2.1 指紋の種類

指紋は、隆線の流れと特異点によりいくつかの種類に分類することができる。現在、一般的に用いられている指紋の種類は、1900年にHenryが提案したものである [5]。Henryは、指紋の特異点に基づいて図1に示す5つの種類 (Left Loop, Right Loop, Arch, Tented Arch, Whorl) に分類した。ここで、Whorlは、WhorlとTwin Loopの2つに分類されることもある。本稿では、Left Loop, Right Loop, Arch, Whorlの4種類に分類するアルゴリズムについて述べる。

2.2 分類アルゴリズム

提案するアルゴリズムは、(1) 前処理, (2) 特異点の抽出, (3) 分類の3つの操作で構成される。

(1) 前処理

撮像状態の悪い指紋画像にも対応するために、以下に示す手順で前処理を行う。

1. 指紋の抽出 (図 2(b))
2. ブロックの平均値で画像を2値化 (図 2(c))
3. 画像の強調と平滑化 (図 2(d))
4. 角度マップの生成 (図 2(e))
5. 角度マップを利用した2値化 (図 2(f))
6. 角度マップの生成

以下では、前処理で行われる各処理について説明するが、それぞれの処理で必要となるパラメータ (ブロックの大きさなど) は、指紋画像を採取するセンサの特性に依存する。本稿で示しているパラメータの具体的な値は、FVC2002の指紋画像データベース DB1 を使用する場合に用いる値である。

指紋画像に含まれる余分な情報を取り除き、指紋のみを抽出するマスクの作成について述べる。まず、それぞれの 3×3 のブロックに対して画素値の分散を求める。指紋の部分では、隆線と溝の間の画素値の差が大きいため、分散は大きな値となる。一方、指紋以外の部分に

残っているノイズなどは、画素値に差があまりないため分散は小さな値となる。求めた分散から指紋のみを抽出するマスクを作成するために、分散の平均値の $1/4$ で分散を2値化する。2値化することで、指紋の隆線のみが浮き出するため、モルフォロジカルフィルタを利用して指紋を抽出するマスクを作成する。ここで、利用したモルフォロジカルフィルタは、dilation (膨張) と erosion (縮退) である。このように作成したマスクを図 2 (b) に示す。

撮像状態の悪い指紋画像は、隆線がつぶれていたりかすれていたりするため、あらかじめ隆線を強調する必要がある。そこで、ブロック単位の平均値を求め、それを閾値としてブロックの中心画素を2値化する。本稿では、ブロックサイズを 11×11 とした。この処理により、つぶれていた隆線も図 2(c) のように強調することができる。

次に、かすれや傷などでとぎれてしまった隆線をつなげるために画像を強調する。現在までに提案されている指紋画像の強調には、隆線の角度や局所的な周波数を利用し、それぞれの画素に適切なフィルタリングをする手法 [6]–[8] などがある。しかし、これらは計算量が多いため大規模なデータベースには向いていない。そこで、本稿では、指紋画像の周波数成分を利用した画像強調を使った [9]。この手法は、画素ごとではなくブロックごとに処理を行うことができ、さらに離散フーリエ変換と逆離散フーリエ変換の2つの操作で実現できるため、画素ごとに計算が必要となる手法よりも高速に強調処理を行うことができる。例えば、 $N_1 \times N_2$ の指紋画像 $f(n_1, n_2)$ を考える ($0 \leq n_1 \leq N_1 - 1, 0 \leq n_2 \leq N_2 - 1$)。 $f(n_1, n_2)$ を2次元離散時間フーリエ変換すると次式となる。

$$F(k_1, k_2) = \sum_{n_1=0}^{N_1-1} \sum_{n_2=0}^{N_2-1} f(n_1, n_2) W_{N_1}^{n_1 k_1} W_{N_2}^{n_2 k_2} \quad (1)$$

ここで、 $W_{N_i} = e^{-j \frac{2\pi}{N_i}}$ ($i = 1, 2$), $0 \leq k_1 \leq N_1 - 1, 0 \leq$

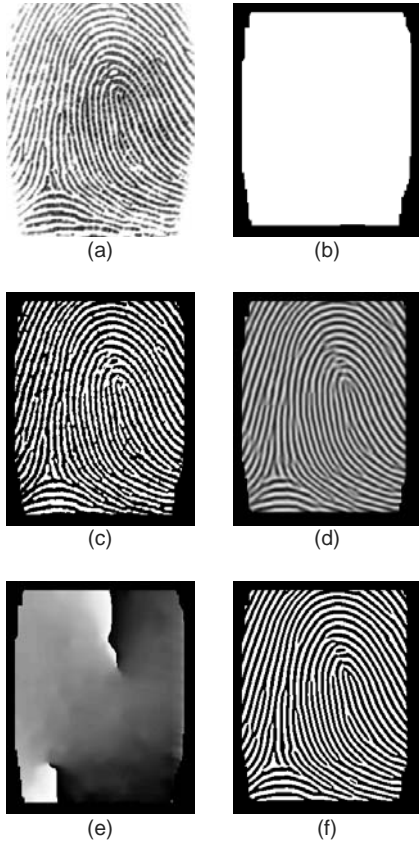


図 2: 前処理で得られる画像: (a) 原画像, (b) 指紋のみを抽出するマスク, (c) ブロックの平均値で 2 値化した画像, (d) 強調と平滑化を行った後の画像, (e) 指紋の角度マップ, (f) 角度マップを利用して 2 値化した画像

$k_2 \leq N_2 - 1$ である。 $F(k_1, k_2)$ の振幅成分を $A(k_1, k_2)$, 位相成分を $\Theta(k_1, k_2)$ とすると, 強調後の周波数成分は次式で表される。

$$F'(k_1, k_2) = A^k(k_1, k_2)e^{j\Theta(k_1, k_2)} \quad (2)$$

ここで, k ($2 \leq k \leq 3$) はパラメータである。提案するアルゴリズムでは, $k = 2.4$ とした。 $F'(k_1, k_2)$ を 2 次元離散フーリエ逆変換して得られる画像が強調画像となる。実際には, 振幅成分を k 乗することでエネルギーが増大するため, ある程度のブロック (例えば 16×16) で処理を行い, 画素値を 256 階調に正規化する必要がある。ブロック単位で強調処理を行うので, 平滑化フィルタを使ってブロック間をなめらかにする (図 2(d))。

続いて, 強調した画像から角度マップを作成する。強調後の指紋画像に 3×3 の Sobel フィルタをかけ, 縦方向と横方向の勾配 (Gradient) を求める。縦方向の勾配を $G_x(n_1, n_2)$, 横方向の勾配を $G_y(n_1, n_2)$ とすると画素 (n_1, n_2) における隆線の角度 $\theta(n_1, n_2)$ は次式で与えられる。

$$\theta(n_1, n_2) = \frac{1}{2} \tan^{-1} \left(\frac{2G_{xy}}{G_{xx} - G_{yy}} \right) + \frac{\pi}{2} \quad (3)$$

ここで, G_{xy}, G_{xx}, G_{yy} は次式で表される。

$$G_{xy} = \sum_{m_1=n_1-8}^{n_1+8} \sum_{m_2=n_2-8}^{n_2+8} G_x(m_1, m_2)G_y(m_1, m_2)$$

$$G_{xx} = \sum_{m_1=n_1-8}^{n_1+8} \sum_{m_2=n_2-8}^{n_2+8} G_x(m_1, m_2)^2$$

$$G_{yy} = \sum_{m_1=n_1-8}^{n_1+8} \sum_{m_2=n_2-8}^{n_2+8} G_y(m_1, m_2)^2$$

図 2(e) は, 得られた角度マップ $\theta(n_1, n_2)$ ($0 \leq \theta(n_1, n_2) \leq \pi$) である。

強調後の指紋はある程度鮮明になっているが, まだ隆線が不鮮明な部分がある。そこで, 角度マップを利用して指紋画像を 2 値化する。角度マップを利用した 2 値化とは, 紋の流れに沿って画像を 2 値化する手法である。まず, 注目する画素を中心とした 9 行 7 列のマスクを作り, 行の方向が注目している画素の角度と一致するように回転させる。その後, 行方向に画素値の和をとり, 中央の行の和がそのマスク内の行方向の和の平均値より高ければマスクの中心画素を 1 と, 低ければ 0 とする。以上の操作を全画素に対して行うことにより, 指紋画像が 2 値化される。この方法では, 画素を隆線の流れに沿って, 周囲の画素も考慮しながら 2 値化するので, 隆線の途切れなどを修復することができる (図 2(f))。その後, 2 値化した画像から再び角度マップを作成する。

(2) 特異点 (Singular Point) の抽出

指紋を分類するときに使う特異点 (Singular Point) の抽出法について述べる。指紋には, 隆線の方向が急激に変化する特異点が存在する。このような特異点には, loop と delta の 2 つがある。提案するアルゴリズムでは, 特異点を抽出するために Poincaré Index を用いた手法を使った [2]。まず, $[0, \pi]$ の範囲で求まっている角度マップを $0, \pi/4, \pi/2, 3\pi/4$ の 4 方向に量子化する。図 3 のように, 画素 (i, j) を中心として角度マップから 3×3 のブロックを取り出し, 左上の画素から時計回りに $\theta_0 \dots \theta_7$ とする。ここで, 隣接画素ではなく, 4 画素おきの画素を取り出して 3×3 のブロックとした。取り出

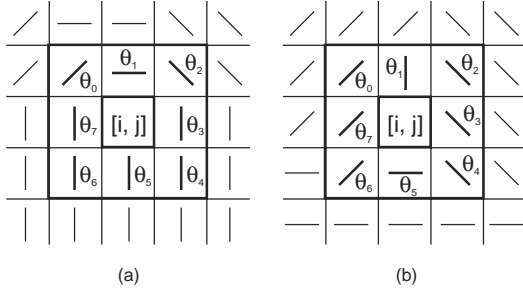


図 3: 特異点の抽出: (a) loop ($P(i, j) = \pi$), (b) delta ($P(i, j) = -\pi$)

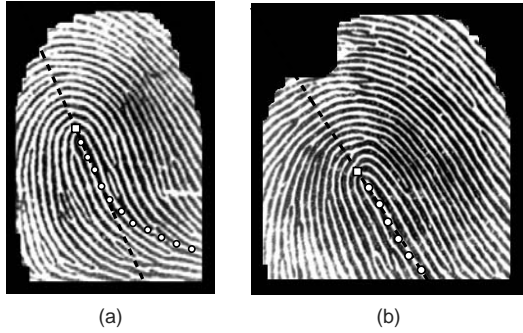


図 4: **LL** と **RL** の判定: (a) loop から隆線をたどり **RL** に分類された指紋, (b) **LL** と **RL** のどちらにも判断できず, **L** に分類された指紋 (破線は loop の角度に対して引いた直線であり, \circ は 20 画素ごとに隆線をたどった軌跡である)

したブロックに対して次式で表される Poincaré Index $P(i, j)$ を求める.

$$P(i, j) = \sum_{k=0}^7 (\theta_{(k+1) \bmod 8} - \theta_k) \quad (4)$$

ただし,

$$\left| \theta_{(k+1) \bmod 8} - \theta_k \right| \leq \frac{\pi}{2}$$

である. 図 3 のように, $P(i, j) = \pi$ の場合は loop, $P(i, j) = -\pi$ の場合は delta となる. これら以外の場合はすべて $P(i, j) = 0$ であり, ブロックの中心が隆線であることを示す.

(3) 分類



(a) (b)

図 5: **A** の判定: (a) **A** に分類された指紋, (b) **A** に分類されない指紋 (\circ は **A** を判定するための基準点である)

指紋の分類は, Poincaré Index で求めた loop と delta の位置関係で判断する. 本稿で提案するアルゴリズムでは, Arch (**A**), Left Loop (**LL**), Right Loop (**RL**), Whorl (**W**) の 4 種類に分類する. それぞれの場合の loop と delta の位置関係は, 図 1 に示した通りである. 撮像状態が悪い場合は, 指紋画像から loop と delta が同時に検出されないため, 特異点の位置関係だけではなく, 部分的な情報を使って分類しなければならない. 撮像状態の悪い指紋画像も分類できるように以下の点を考慮してアルゴリズムを作成した.

- 指紋全体が撮像されていない場合は, loop からの隆線の流れが loop の角度に対して左右のどちらに流れているかで判断する (図 4(a)). **LL** と **RL** のどちらにも分類できないものは Loop (**L**) とする (図 4(b)).
- A** の場合は, 指紋に特異点が存在しない場合であるが, 撮像状態が悪い指紋画像では **A** でなくても特異点が存在しない場合がある. **A** であるかどうかを調べるために, 列方向に隆線の流れが急で, 行方向に隆線の流れが緩やかな点を調べる. まず, 列方向に対して最も隆線の流れが急である列 col を次式で求める.

$$col = \left\{ j \left| \max \left\{ \sum_j |\theta(i, j) - \theta(i-1, j)| \right\} \right. \right\} \quad (5)$$

次に, col 列目に対して隆線の流れが最も緩やか



図 6: 分類することができない指紋画像: (a) delta しか存在しないため分類することができない指紋, (b) loop も delta も存在せず, **A** にも分類されなかった指紋 (Δ は delta, \circ は **A** 判定の基準点である)

な行 row を次式で求める.

$$row = \left\{ i \mid \min \left\{ \sum_{k=-2}^2 |\theta(i, col + k)| \right\} \right\} \quad (6)$$

以上より求めた (row, col) が **A** であるかを判断するための基準点となる. **A** である場合は, 基準点付近の隆線は扇型のような流れになっている (図 5(a)). 一方, **A** でない場合は, 図 5(b) のように隆線が一直線に流れている. 以上より **A** であるかどうかを判断する.

- 図 6 のようにどの種類にも分類することができない指紋画像が存在するので, Other (**O**) という種類を用意する.

以上の点を考慮して作成した指紋分類アルゴリズムを図 7 に示す.

3 実験結果

提案する指紋分類アルゴリズムを使って撮像状態の悪い指紋画像を分類した結果を示す.

今回の実験では, FVC2002 [4] で使用された指紋データベースを使った. FVC2002 では 4 種類のデータベースが用意されているが, 今回はその中で Identix 社製の光学式センサーで撮像されたデータベース (DB1) を使った. このデータベースには, 100 人からそれぞれ違うタイミングで 8 回撮影した画像 800 枚が格納されている. 今回の実験では, **A**, **LL**, **RL**, **L**, **W**, **O** の 6 種類に分類した.

procedure 指紋分類アルゴリズム

```

1. 指紋の種類を O とする;
2. if loop が存在する then
3.   loop の角度を求める;
4.   if loop の数は 1 つである then
5.     if デルタが存在する then
6.       delta が loop に対して左右どちらに位置
           するかを調べる;
7.       if delta の数が 1 つである then
8.         if loop が上向きである then
9.           種類を W とする
10.        else if delta が loop より上に
            位置する then
11.          隆線の流れを使って LL, RL, L に
            分類する
12.          else if loop と delta が近くにある then
13.            種類を A とする
14.          else
15.            delta が loop の左に位置すれば RL
            とし, 右に位置すれば LL とする
16.          end
17.        else if delta の数が 2 つである then
18.          if loop が上向きで, delta が loop の
            左右に位置する then
19.            種類を W とする
20.          end
21.        end
22.      else
23.        if loop が上向きである then
24.          種類を W とする
25.        else
26.          隆線の流れをたどり, LL, RL, L に
            分類する
27.        end
28.      end
29.    else { loop が複数存在する }
30.    if loop 間の角度差が  $\pi$  前後である then
31.      種類を W とする
32.    end
33.  end
34. else { loop が存在しない }
35.   A であるかを判定するために, 基準点を求める;
36.   求めた基準点付近の隆線の流れ調べる;
37.   if 隆線の流れが扇形である then
38.     種類を A とする
39.   end
40. end.

```

図 7: 指紋分類アルゴリズム

実験結果を表 1 に示す. 撮像状態が悪いため, 実際の種類が **LL**, または **RL** であったとしても, いくつかは **L** に分類される. **LL** と **RL** に対して **L** に分類されても正しいと判断すると, 分類を誤る割合は 9.12% である. また, **O** はどの種類に分類されるか判断できない指紋であるので, どちらか一方でも **O** であれば照合を行うとすると, 分類を誤る割合は 4.25% である.

図 8 に分類を間違える指紋画像の例を示す. 図 8(a) は, 実際は **RL** であるが, **LL** と分類される. これは, loop の下に傷があるため, loop の角度を正しく求めることができないからである. そのため, delta が loop に対して右に位置すると誤って判定されてしまう. 図 8(b) は, 実際は **W** であるが, **L** に分類される. 全体が写っていれば, 画像の下方に上向きの loop があり **W** と分類す



図 8: 分類を間違ってしまう指紋: (a) LL と判定される指紋 (実際は RL), (b) L と判定される指紋 (実際は W)

ることができるが、この画像では、下の方が切れてしまっているため、loop が 1 つしか求められず、誤って L に分類されてしまう。

今回使用したデータベースに対して、筆者らが提案している位相限定相関法を使った照合アルゴリズムで照合実験を行った [10]。分類アルゴリズムを使用しない場合は、EER (Equal Error Rate) は 7% であった。これに対し、分類アルゴリズムによって決められた種類が一致しない場合は照合スコアを 0 とし、種類が一致した場合に照合アルゴリズムで照合すると、EER は 4% であった。提案するアルゴリズムを照合前のおおざっぱな照合として利用することで、照合精度を向上することが可能である。

4 まとめ

本稿では、指紋の特異点を利用して撮像状態にロバストな指紋画像の分類アルゴリズムを提案し、FVC2002 の指紋データベースを使って性能評価を行った。実験結果より提案する指紋分類アルゴリズムは、撮像状態が悪くても高精度に指紋を分類することができることを示した。今後は、一般的に分類アルゴリズムの評価に使用されている NIST (National Institute of Standards and Technology) の指紋画像データベースを使って実験を行い、他の分類アルゴリズムとの性能比較を行う予定である。

謝辞

本研究を進めるにあたり、有益なご助言をいただいた株式会社山武ビジョンセンシング部の小林孝次氏、

表 1: 分類結果

Assigned Class	True Class			
	A	LL	RL	W
A	12	1	1	0
LL	2	227	3	4
RL	1	0	262	11
L	0	33	33	10
W	1	0	0	160
O	0	11	13	15

中島寛氏に感謝いたします。

参考文献

- [1] D. Maltoni, D. Maio, A. K. Jain, and S. Prabhakar, *Handbook of Fingerprint Recognition*, Springer, 2003.
- [2] M. Kawagoe and A. Tojo, "Fingerprint pattern classification," *Pattern Recognition*, Vol. 17, No. 3, pp. 295–303, 1984.
- [3] K. Karu and A. K. Jain, "Fingerprint classification," *Pattern Recognition*, Vol. 29, No. 3, pp. 389–404, 1996.
- [4] <http://bias.csr.unibo.it/fvc2002/>
- [5] E. Henry, *Classification and Uses of Finger Prints*, Routledge, 1900.
- [6] L. ÓGorman and J. V. Nickerson, "An approach to fingerprint filter design," *Pattern Recognition*, Vol. 22, No. 1, pp. 29–38, 1989.
- [7] B. G. Sherlock, D. M. Monro, and K. Millard, "Fingerprint enhancement by directional fourier filtering," *IEE Proc.-Vis. Image Signal Process.*, Vol. 141, No. 2, pp. 87–94, April 1994.
- [8] L. Hong, Y. Wan, and A. Jain, "Fingerprint image enhancement: algorithm and performance evaluation," *IEEE Trans. PAMI*, Vol. 20, No. 8, pp. 777–789, August 1998.
- [9] A. J. Willis and L. Myers, "A cost-effective fingerprint recognition system for use with low-quality prints and damaged fingerprints," *Pattern Recognition*, Vol. 34, No. 2, pp. 255–270, February 2001.
- [10] K. Ito, H. Nakajima, K. Kobayashi, T. Aoki, and T. Higuchi, "A fingerprint matching algorithm using phase-only correlation," *IEICE Trans. Fundamentals*, Vol. E87-A, No. 3, pp. 682–691, March 2004.