

タイトル	稜線方位分布による指紋のコード化
著者	高井, 信勝; TAKAI, Nobukatsu
引用	北海学園大学工学部研究報告(39): 171-183
発行日	2012-02-14

稜線方位分布による指紋のコード化

高井 信勝*

Coding of the Fingerprint by Ridgeline Direction Distribution

Nobukatsu TAKAI*

Abstract

In this research, the interval and direction of the ridgelines which constitute a fingerprint pattern are statistically analyzed by image processing by MATLAB. As a result, the macroscopic feature of a fingerprint pattern is detected as direction distribution of ridgelines. Furthermore, the feature code for identifying a fingerprint is defined as direction distribution of the ridgeline obtained by dividing into the domain of 5x5. Fingerprint authentication and collation of two fingerprints are performed by evaluating the difference of those feature codes.

1. はじめに

指紋は、指先の皮膚が盛り上がり形成される稜線とその間の谷線が形作る紋様である。これが「万人不同」、「終生不変」ということから個人を同定するひとつの手段として古くから数多くの研究がなされている^{1,2)}。特に、最近のインターネット社会では、ネット回線を通して個人を同定するときに、個々人の生体的な特徴を利用するバイオメトリックスの代表的な手段として指紋認証技術は益々重要性を増している。

このような状況の中で星野幸夫らのNECグループ²⁾は、指紋の紋様に存在する特徴点（稜線の端点と分岐点）を検出し、それらの間の距離と方向、および二つの特徴点の間をよぎる稜線数から指紋を同定（あるいは照合）する手法を開発した。一方、これより以前から指紋の巨視的な紋様の特徴を指紋照合に用いることがなされているが、コンピュータによる照合の自動化がむずかしく必ずしも確立した技術には至っていない。

指紋の稜線の紋様は、微細にみるとそれぞれで異なっているが、おおまかな形状は似ている

* 北海学園大学工学部電子情報工学科

* Department of Electronics and Information Engineering, Faculty of Engineering, Hokkai-Gakuen University



図1 代表的な指紋の紋様. 左から渦状紋, 蹄状紋 (左流れ), 蹄状紋 (右流れ), 弓状紋.

ものが多く, 大別すると渦状紋, 蹄状紋 (左流れ), 蹄状紋 (右流れ) と弓状紋に分類される. これらの代表的な紋様を図1に示す. 渦状紋は, 文字通り稜線が渦を巻くような紋様であり, 蹄状紋は渦の下側の稜線が開いて片側に流れる紋様で右に流れる型と左に流れる型がある. 弓状紋は中央部の稜線の流れがほぼ左右対称に流れて底部の稜線は水平に走る紋様であって, 丁度弓と弦からなる形状のものである. ただし, すべての指紋がこのどれかのタイプに属するわけではなく, これらの中間的な紋様が数多く存在することが知られている.

指紋をこのような稜線紋様のタイプで分類するだけでは, 多くの人の指紋を識別する特徴として利用することはできない. しかし, 稜線紋様は「万人不同」と言われるので, その紋様をもっと詳細に解析し, それを特徴付けるパラメータを見いだすなら, 個人の同定に指紋の紋様を用いることが可能である.

本研究は, このような観点から, MATLABによる画像処理技術を駆使して, 指紋紋様を構成する稜線を解析した. ここでは, 指紋紋様の巨視的な特徴を, 稜線の方位分布として検出し, それによって指紋を識別する特徴コードを提案する. さらに, それを指紋認証・照合に用いる方法を併せて説明する. 以下では, 2節で指紋の稜線間隔および稜線方位を解析するアルゴリズムを詳説し, 3節で稜線方位分布の解析例を示す. この結果に基づいて, 4節では, 指紋特徴コードを導入し, それを指紋識別に利用する方法を説明する.

2. 稜線間隔と稜線方位の解析アルゴリズム

最初に, アルゴリズムの説明を分かりやすくするために, 具体的な指紋画像を図2に示す. ここには, 処理される指紋の原画像 (左) と処理過程にある細線化画像 (右) の一例が示されている. なお, 後述されるように, 指紋画像は最初に2値化処理され, 2値画像として1を白, 0を黒 (背景) で表示されている. このとき, 現実の稜線と谷線の区別に関係せず, 白の部分を稜線として解析する.

図3が指紋の稜線方位を解析するアルゴリズムのフローチャートである. 以下, この流れ図に沿って説明する.



図2 原画像（左）とその細線化画像（右）．細線化画像中の白い枠はひとつの解析領域．原画像サイズ364×256．解析領域サイズ60×50．

2.1 前処理

まず、解析する指紋のグレースケール画像をMATLABに読み込む．MATLABは、ほとんど全てのデータ形式の画像を読み込むことができる．ただし、最近のカメラやスキャナーで撮られた指紋画像は、カラー画像であるが、ここでの解析ではカラー情報は使われない．そのためカラー画像の場合には、それをグレースケール画像に変換したものを読み込む．

さらに、稜線の平均間隔や方位を求めるときには、稜線情報は2値データでよく、また、線幅情報も必要ない．そのため稜線解析の前処理として、画像の2値化およびその結果の細線化処理³⁾を行う．また、必要に応じて、平滑化やメディアンフィルタリングのような雑音低減のフィルタリング処理をおこなう．図2（右）がこのような前処理を終えた画像の一例である．

2.2 解析領域の設定

指紋の紋様を見ると稜線間隔はどこでも似たような値であることがわかる．しかし、自明なことであるが、稜線の方位は場所ごとで異なっている．というより、空間の領域ごとに異なる稜線の集合が個々の指紋の紋様を形作っている．したがって、稜線の方位を求めるときには、図2の右図に示すような限られたサイズからなる解析領域を設定する必要がある．

このとき、領域サイズをどのように決めるかは、重要な事柄であるが、最良の決定法は明瞭ではない．ただ、言えることは、設定した領域内で稜線方位がほぼ揃っている程度のサイズであることが望ましい．この条件は、サイズが十分小さければ満足されることになるが、それは指紋の紋様の特徴としては冗長性が多すぎるものになる．本稿では364行256列のサイズからなる指紋画像を扱い、5×5の25領域を解析領域として用いた．このとき一つの解析領域サイズは、60行50列になっている．

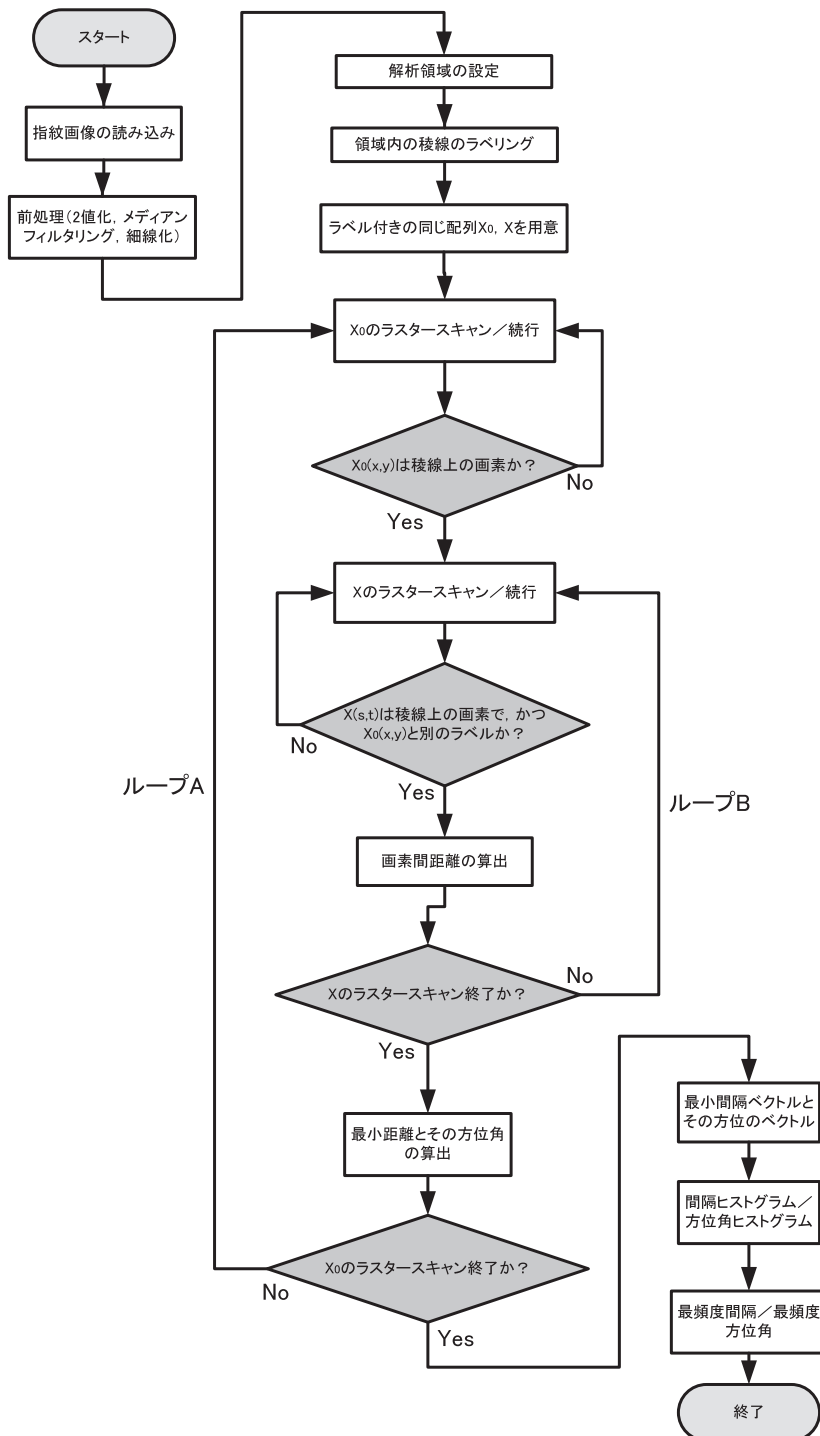


図3 稜線間隔と稜線方位を求めるアルゴリズムのフローチャート

指紋の解析は、このように指紋画像全体をブロックに分解し、各領域ごとに行う。そして指紋紋様の特徴は、4節で述べるように全解析領域の稜線を総合した特徴として把握する。なお、図3のフローチャートは一つの領域を解析するもので、複数の領域を解析するときには、解析領域を変更して、同じ処理を繰り返す。

2.3 分離した稜線のラベリング処理⁴⁾

解析領域が設定されると、そこには限られた本数の細線化された稜線が存在する。このとき、個々の稜線は連結成分、すなわち一続きの画素の集合と見なすことができる。この連結成分は、単に稜線から得られるものだけでなく、指紋画像に含まれる雑音によるものもあり、連結成分の数は統計的なゆらぎを持っている。

しかしながら、以下では雑音の影響は稜線のゆらぎとして考え、特に混乱が生じない限り、連結成分を単に稜線と呼ぶことにする。

いずれにしても、稜線の間隔とその方位の解析は、細線化パターンの一部の稜線を含む画像を解析することになる。この解析の基礎は、ある稜線上の画素から他の稜線上の画素までの距離を調べることであるので、各稜線を識別する必要がある。そのために、領域内の独立した稜線に識別番号をつけ、これによって稜線を識別する必要がある。これはラベリング処理により達成される。

2.4 解析可能条件と解析不能なときの処理

後述するように、設定した解析領域において、最終的には稜線間隔とその方位が統計的な判定を用いて決められる。そのとき、それらが一定の確度を持って得られるためには解析領域内にある程度以上の稜線画素といくつかの識別番号の異なる稜線が存在する必要がある。たとえば、極端な場合であるが、領域内に稜線を構成する画素が数個の場合や、識別番号の異なる稜線が存在しない場合には稜線間隔（距離）も方位も決められない。本法では、この状況を把握するために、次の2条件を用いた。

- ① 解析領域内の稜線の画素数（値1の画素数）が解析領域の対角線の長さの1.5倍以上であること
- ② 解析領域内の稜線の個数（これはラベリングした稜線のラベル数、すなわち、識別番号である）が3以上であること

この2条件は、解析領域を設定し、そこに含まれる稜線群をラベリングした段階で判明する。そうして、両方の条件が満たされたときに解析を行う。どちらかの条件が満たされないときには、解析不可能と判定し、稜線間隔0、方位角0と結果を与える。

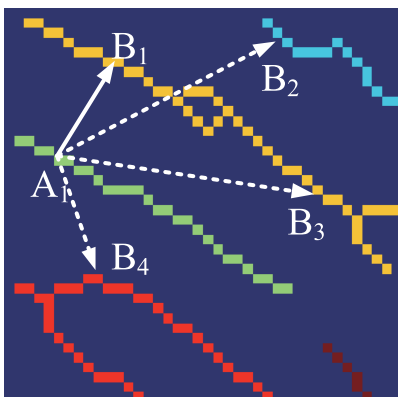


図4 画素間距離を求めるアルゴリズムを説明するための図. 解析領域内の細線化された稜線群はラベル番号に応じて異なる色で示されている.

2.5 稜線間隔とその方位角の検出

図4は、ラベリングされた稜線パターンから稜線間隔を求めるための説明図である. ここで、細線化された線幅1の稜線がラベルごとに色分けされて示されている.

稜線間隔は、図4の点 A_1 のようにあるラベルの稜線上の画素からラベル番号の異なる他の稜線上の画素 (B_1, B_2, B_3, \dots) まで距離のうちで最短のものと考えてよい. そこで、解析する細線化されラベリングされた、図4のような、二つの同じ配列 X_0, X を用意する. まず、配列 X_0 をラスタースキャンして稜線上の点 (m_0, n_0) が見いだされたときに配列 X のラスタースキャンを開始し、点 (m_0, n_0) が属する連結成分 (この図では、緑色の連結成分) とは別の連結成分に属するすべての点 (m, n) までの距離 (絶対値距離) を求める. ここで、 (m_0, n_0) および (m, n) は画素の配列番号である. このようにして、配列 X のラスタースキャンが一通り終わると (図3のループBが終了すると)、点 (m, n) と同数の距離が得られるが、そのうちの最小の距離が点 (m_0, n_0) から隣の稜線までの距離 d であるとみなす. つまり、

$$d = \min(\sqrt{(m - m_0)^2 + (n - n_0)^2}) \quad (1)$$

が得られる. 次に、このようにして得られた最短距離 (稜線間隔) に対する方位角 θ を

$$\theta = \tan^{-1}\left(\frac{n - n_0}{m - m_0}\right) \quad (2)$$

により求める. ここで、 (m_0, n_0) および (m, n) は式(1)の d を与える配列位置の組である. また、 θ は水平軸から測った稜線の方位角であるが、これによって定義される方位角は、 m が行番号、 n が列番号であるので、点 (m_0, n_0) で稜線に接する直線が水平軸に対してなす角度である. したがって、その点で間隔距離がなす角、つまり稜線に垂直な直線がなす角とは補角の関係にある.

以上のようにして、配列 X_0 上の稜線上のすべての画素ごとに一組の (d, θ) が得られる。したがって、配列 X_0 のラスタースキャンが終わると（図3のループAを終えると）、解析領域内の稜線の画素数に等しい距離と方位角が得られる。すなわち、 d と θ はそれぞれ稜線間の最小距離とその方位角の集合として得られる。

2.6 平均稜線間隔の算出

解析領域内の稜線は平行であることはなく、通常は少なからず湾曲している。あるいはまた種々の雑音加わることが普通であるので、上で得られた稜線間隔の集合では、稜線間隔はバラツキを持って得られる。これを具体的に示す一例がヒストグラム分布の図5である。この場合には稜線間隔は7（画素間距離の7倍）の周りに分布している。そして、この平均稜線間隔 \bar{d} は次式により算出される。

$$\bar{d} = \frac{\sum_i d_i h(d_i)}{\sum_{i=1} h(d_i)} \quad (3)$$

ここで、 $h(d_i)$ は距離 d_i のヒストグラム分布であり、図5においては $d_i = 0, 1, 2, \dots, 30$ と取られている。

式(3)から得られる平均の稜線間隔は、ひとつの指紋画像では解析領域を変えてもほぼ一定の値であることが知られる。

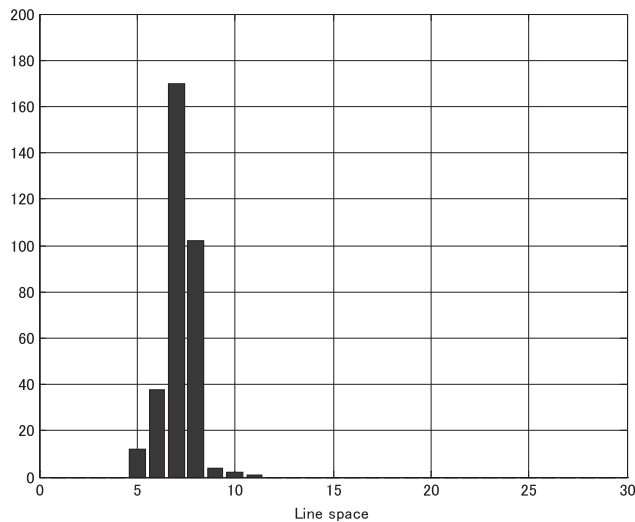


図5 稜線間隔のヒストグラムの一例。

2.7 稜線の平均方位

一方、次に述べる稜線の平均方位は領域ごとに異なる値を取り、この方位分布が指紋の紋様を形成すると考えてよい。したがって、この方位分布は重要である。

稜線上の各画素における稜線の方位は、式(2)によって得られる角度 θ の集合である。この値は、解析領域内で稜線がすべて平行であることはないから、当然のこととして、稜線上の各画素において異なるのが自然であって、ここでも平均の方位角を求めることになる。つまり、平均の方位角 $\bar{\theta}$ は、離散的な方位角 θ_i を横軸にとったヒストグラム $h(\theta_i)$ に対して

$$\bar{\theta} = \frac{\sum_i \theta_i h(\theta_i)}{\sum_i h(\theta_i)} \quad (4)$$

により得られる。このとき注意しなければならないことは、方位角の符号の問題である。すなわち、式(2)で得られる方位角 θ は

$$-\pi/2 \leq \theta \leq \pi/2 \quad (5)$$

の範囲の値として得られるが、 $\theta = \pm\pi/2$ はいずれも稜線が水平軸に対して垂直、つまり画素の列の並び方向の方位である。そのため、同じ垂直な稜線の状態が $\theta = +\pi/2$ と得られたり、 $\theta = -\pi/2$ と得られたりする問題である。

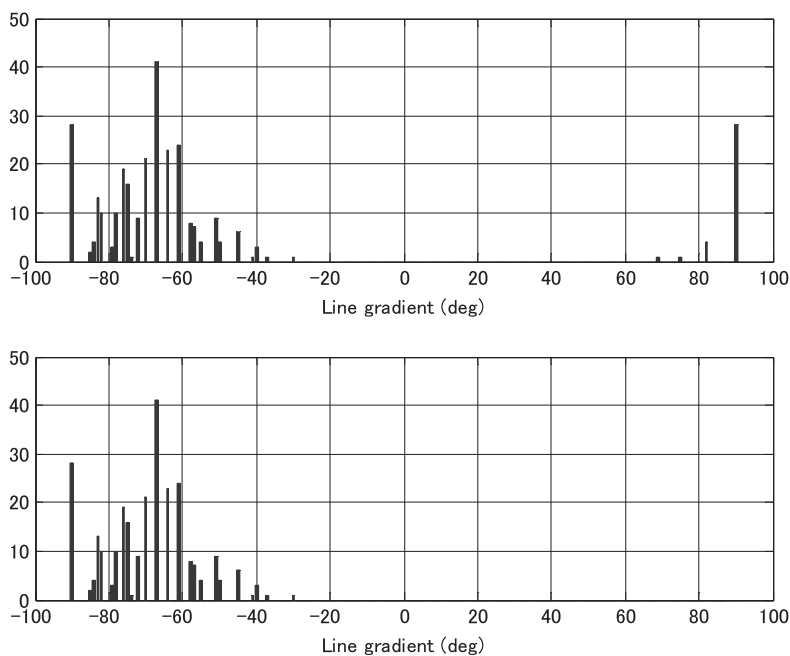


図6 稜線の方位角ヒストグラム。(上) 方位角符号の判断以前、(下) 角度符号の判断後。

図6（上図）は、このような符号のゆらぎによる影響が現れた方位角ヒストグラムの具体的な一例である。ここで横軸は方位角度で -90° から 90° （ $-\pi/2$ から $\pi/2$ ）であって、この間にヒストグラムが 1° ごとにカウントされる。その結果、この図では、負の領域に現れる方位角ヒストグラムのほかに正の領域、特に、 $\theta = +90^\circ$ にもヒストグラムが存在する。これは本来 $\theta = -90^\circ$ とカウントされるものが算出データのゆらぎ、具体的には、式（2）における n_0 と n の大小のゆらぎによってプラスにカウントされたものと考えられる。

問題は、このような不連続な方位角の 180° の飛びが加わったヒストグラムをそのまま用いて、式（4）から平均の方位角を求めると誤った結果になることである。なぜなら、解析データのわずかなゆらぎによって、本来の向きとは反対の方位角になるからである。本法では、この問題を回避するために、稜線方位角の正領域と負領域のヒストグラムの総計を比較し、カウント数の多い方の領域だけをヒストグラムとして採用し、少ない方の領域は放棄した（図6（下図））。このように、多数決的に、最初に稜線方位角の符号を確定し、それに基づいて平均の方位角を式（4）により算出した。

なお、この背景には、本法が限られた解析領域内で稜線の方位角は統計的に一定の符号を持っていることを前提としている状況がある。したがって、そこにおいて正負の方位角が混在する状態は、解析不能あるいは解析の対象外と扱った。

3. 稜線方位マップ

図3の解析フローチャートはひとつの解析領域のものであるが、その領域で最頻度距離（つまり稜線間隔）と最頻度方位角（稜線の方位角）が求まったあとで、解析領域を変えて繰り返し実行すると、指紋画像全体に渡る稜線の方位角マップが得られる。

画像サイズが364行256行からなる指紋画像を 5×5 の解析領域（領域サイズ60行50列）に分割し、各領域で稜線方位を求めた結果のいくつかを図7に示す。この図でa, bが渦状紋、c, dが蹄状紋（右流れ）、e, fが蹄状紋（左流れ）、g, hが弓状紋に分類される指紋である。いずれも左は原画像、右が得られた稜線方位マップで、そこでは指紋の細線化パターンの上に解析領域を描き、その中央部分に得られ稜線方位を矢印で示してある。ただし、解析不能と判断された領域には矢印は示されていない。また、稜線方位は、稜線に沿って矢印によって与えられ、矢印の向きは、稜線方位が 0° のときに水平方向の右向き、プラスのときは右上向き、マイナスのときは右下向きである。

図7の稜線方位マップにみられるように、どの指紋の場合も各領域で稜線方位が首尾よく得られている。渦状紋では渦の中心を取り巻くように稜線方位が得られ、蹄状紋では下方の稜線の流れに従った稜線方位が得られている。また、弓状紋では隆起した部分の稜線方位が得られていて、これらは人の目に映る指紋と同様な紋様として見て取れる。

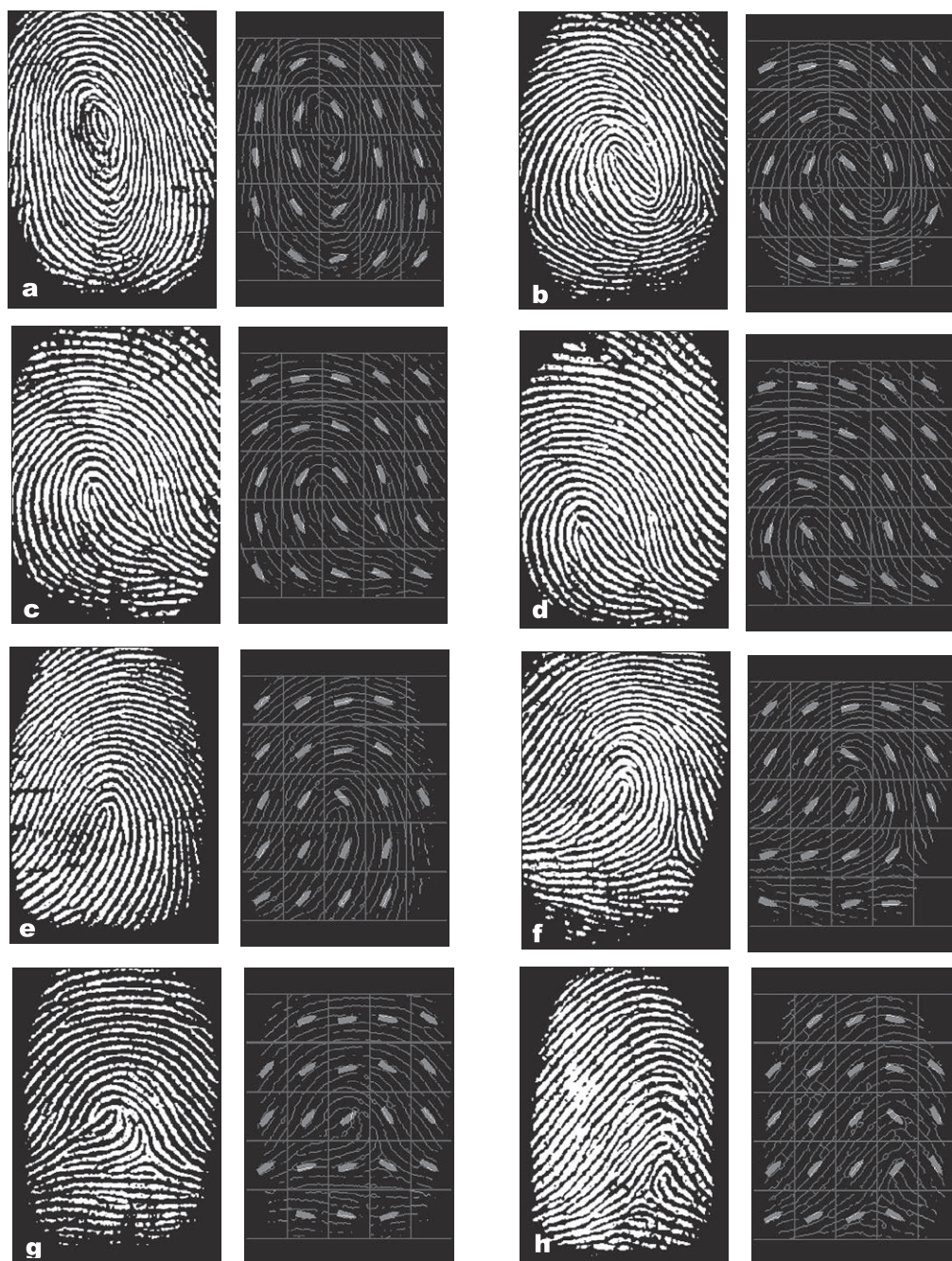


図7 指紋の紋様（左）とその稜線の方位マップ（右）。
 (a, b) 渦状紋, (c, d) 蹄状紋（右流れ）, (e, f) 蹄状紋（右流れ）, (g, h) 弓状紋

この中で、特に強調されることは、稜線に欠陥があって途中で途切れていたり、あるいは雑音加わって指紋の構造が崩れている部分でも、かなりよく稜線方位が得られていることである。このことは、本手法が雑音などによる欠陥をもった指紋に対しても適用できることを意味している。

4. 特徴コードとそれによる指紋照合法

図7に示された稜線方位マップは、正に指紋の紋様を反映している。したがって、それをそのまま指紋特徴コードとして利用できる。ここで、「指紋特徴コード」とはおのおのの指紋を特徴付ける数値パラメータで、ここで提案するものは稜線方位角を要素とする 5×5 の行列（多元数の数値パラメータ）が指紋特徴コードである。ただし、行列要素は、方位角そのものではなくその取り得る最大値つまり $\pi/2$ で規格化した値としている。したがって、行列要素は $[-1, +1]$ の範囲の値で与えられる。

図8に、二つの指紋とそれぞれの指紋特徴コードを画像化して示したパターンを示してある。このような画像化した指紋特徴コードパターンでは、稜線方位角のプラスの最大値が白、最小値が黒で表示される。したがって、コードパターンごとに最大値や最小値が異なるので、パターンの明暗とコード値は1対1に対応しない。しかし、二つのコード値が近い値のときは両者のパターン模様はほぼ一致し、完全に同じときは、言うまでもなく両者の模様は完全に一

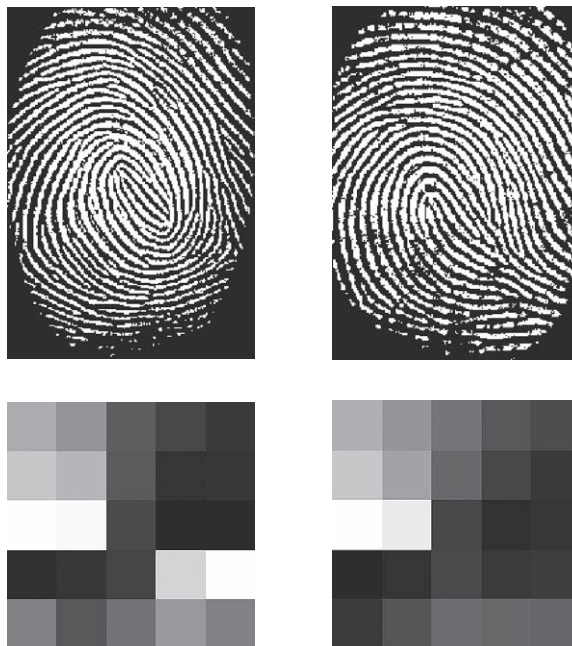


図8 二つの指紋（上）と画像化したそれらの指紋特徴コード（下）

致する．このように，指紋特徴コードをパターン化して表示すると，指紋の同一性を視覚的にとらえることができる．

指紋特徴コードを数値的に評価して指紋識別や照合を行うことができる．その一つの方法は，二つの特徴コードの差を取り，その差のrms値（二乗平均平方根）を求めて実行できる．この値は，差の平均値がゼロと考えられるので標準偏差値（Standard deviation：std）に等しい．たとえば，**図8**の場合は，全く異なる二つ指紋の特徴コードであり，両者の差のrms値は0.3261と得られる．この値は，平均値がゼロで分布の幅が ± 1 の一様分布の場合のrms値である $1/\sqrt{3} = 0.5774$ に匹敵する値である．一方，**図9**の指紋は**図8**（左）の指紋が 10° だけ回転したときの指紋とその特徴コードパターンである．このときの回転前後の特徴コードから得られた差のrms値は0.0569であった．この値は，異なる指紋の場合（**図8**の場合）のほぼ $1/10$ である．このように異なる指紋と同一指紋で回転が加わったときでは，二つの指紋の特徴コードの差のrms値に大きな違いが生じる．

そこで，5組の全く異なる二つの指紋について特徴コードの差のrms値を調べると，0.2～0.6の範囲の値であった．一方，一つの指紋を 0° から 10° の範囲で回転させたときの回転前後の5通りの組み合わせで調べると，特徴コードの差のrms値は0.0～0.13の範囲の値として得られた．

このように， 5×5 サイズの行列で与えられる稜線方位の特徴コードが，指紋のわずかな回



図9 **図8**（左）の指紋を 10° 回転したときの指紋とその特徴コードパターン

転によって大きな変化を受けず、これによって指紋の回転に強い指紋認識が可能であることが確かめられた。

5. おわりに

本稿では、MATLABによる画像処理技術を駆使して、指紋紋様を構成する稜線の間隔およびその方位を統計的な手段を用いて解析するアルゴリズムを提案した。そして、このアルゴリズムを適用して、指紋紋様の巨視的な特徴を稜線の方位分布として検出した。さらに、 5×5 の領域に分割して得られる稜線の方位分布を、指紋を識別するための指紋特徴コードとして採用し、それを指紋認証・照合に用いることができることを示した。

指紋認証・照合では、指紋特徴コードの差のrms値を求め、この値が小さいときに同一指紋と認定できる。そのとき、同一指紋で一方がわずかに回転したときでも、特徴コードの差のrms値はほとんど影響を受けず、同一認定ができることが明らかにされた。しかし、指紋の位置ずれや変形などが指紋の認定に与える影響は今後の課題として残されている。

本研究は、戦略的研究基盤形成支援事業「電磁・光センシングを主体とする生体関連情報の先進的計測・処理技術の開発と応用」の一環として行った。

参考文献

- 1) D. Maltoni, D. Mario, A. K. Jain and S. Prabhakar : *Handbook of Fringerprint Recognition* (Springer-Verlag New York, Inc., 2003).
- 2) 画像電子学会編 星野幸夫監修：指紋認証技術（電機大学出版会，2005）。
- 3) 高井信勝：「2値画像の細線化における連結数の役割の再検討」，工学部研究報告，第38号，pp. 155-172（北海学園大学，2011）。
- 4) 高井信勝：MATLAB入門 [増補版]，第11章，p. 168（工学社，2002）。