

卒業論文概要書

Summary of Bachelor's Thesis

Date of submission: 02 / 01 / 2017

学科名 Department	情報理工	氏名 Name	合田大祐	指導 教員 Advisor	渡辺 裕 印 Seal
研究指導 Research guidance	オーディオビジュアル情報処理研究	学籍番号 Student ID number	1W120203-8 CD		
研究題目 Title	デジタルカメラを用いた AKAZE 特徴マッチングによる指紋認証 Fingerprint Authentication by AKAZE Feature Matching Using Digital Camera				

1. まえがき

近年、生体認証は様々な発展を遂げている[1][2]。生体認証の代表的な例には以下のものがある。主に認証の主流とされている指紋認証、手の中の血管の流れを認識する静脈認証、元々データベースに登録されている顔の特徴点を抽出し判別を行う顔認証、撮影した虹彩(アイリス)の画像からあらかじめ登録した特徴データで判別する虹彩認証、人間の声をデジタル化して判別を行う音声認証などである[3]。その中でも指紋認証は、最も精度が高く、かつコスト面、安全面に優れることから数多くの場面で使われている。特に近年注目されている指紋認証方法は、デジタルカメラを用いた非接触型指紋認証である。本研究では非接触型指紋認証を目的とする。

2. デジタルカメラを用いた指紋認証

本研究では、デジタルカメラを用いた指紋認証を目的とし[4]、カメラ画像を指紋画像データとして認証対象とする方式を提案する。まず、入力画像をグレースケールに変換し[5]、さらに勾配フィルタをかける[6]。この処理により、デジタルカメラを用いた指紋認証の際に発生する低コントラスト問題を回避し、指紋の凹凸を検出することが可能となる。さらにマッチング手法として、回転や拡大、輝度変化に対応できる Accelerated KAZE (AKAZE) を用いる[7]。

3. 実験結果評価

実験では、まず提案手法であるデジタルカメラを用いた指紋認証画像に対してAKAZEマッチングを用いて認証を行うことが可能かどうか調査した。その結果勾配フィルタを適用した場合には[5]、指紋線の歪が生じた。その結果、歪が特徴量と捉えられてしまい、認識精度の低下を招いた。また、直接AKAZEを認証に用いた場合には、一部の指紋認証結果から正解率が96%~100%が得られた。

4. まとめ

本論文では、デジタルカメラを用いた指紋認証として、カメラ画像を指紋画像データとして扱い、AKAZE マッチングのアルゴリズムを用いた指紋認証を提案し、実験により手法の評価を行った。評価実験により、AKAZE マッチングが高い精度で指紋認証が可能であること、勾配フィルタの適用は認証精度を低下させることが分かった。

参考文献

- [1]管ほか, “ここまで来たバイオメトリクスによる本人認証システム”, IPAJ Magazine, Vol.40, No.11, pp.1-5, Nov. 1999
- [2]酒井ほか, “バイオメトリック認証技術について”, 電気設備学会誌, Vol.30, No.10, Oct. 2014
- [3]内田薫, “指紋にとる個人認証の最前線”, 映像情報メディア学会誌 Vol.55, No.2, pp.176-179, Feb. 2001
- [4]B.Y.Hiew, Andrew B.J.Teoh and David C.L. Ngo, “Automatic Digital Camera Based Fingerprint Image Preprocessing”, IEEE Computer Graphics, Imaging and Visualization, No.6, pp.182-189, Jul. 2006
- [5]寺主一成, “グレースケールの色濃度値による解析”, SEN-IGAKKAISHI(報文), Vol.36, No.6, pp.104-109, Jun. 1980

[6]宮岡伸一郎,“画像の勾配空間フィルタリング“,映像情報メディア学会誌, Vol.57, No.10, pp.1284-1291, Oct. 2003

[7]<http://www.robosafe.com/personal/pablo.alcantarilla/publications>, 2017年1月7日アクセス

2016 年度 卒業論文

デジタルカメラを用いた
AKAZE 特徴マッチングによる指紋認証

Fingerprint authentication by
AKAZE feature matching using digital camera

指導教員 渡辺 裕 教授

早稲田大学 基幹理工学部

情報理工学科

1W120203-8

合田 大祐

目次

第1章	序論	1
1.1	研究の背景	1
1.2	本研究の目的	1
1.3	関連研究	1
1.4	論文の構成	2
第2章	指紋認証	3
2.1	はじめに	3
2.2	指紋認証	3
2.2.1	従来の指紋認証	3
2.2.2	従来の指紋認証の問題点	3
2.3	デジタルカメラを用いた指紋認証（従来研究）	4
2.3.1	デジタルカメラを用いた指紋認証の課題	8
2.4	SIFT マッチング	8
2.4.1	そのほかのキーポイント検出タイプ	11
2.4.1.1	SURF	11
2.4.1.2	FAST	12
2.4.1.3	ORB	12
2.4.1.4	KAZE, AKAZE	12
2.4.1.4.1	非線形拡散フィルタリング	13
2.4.1.4.2	非線形スケールスペース	13
2.4.1.4.3	特徴記述	14
2.5	むすび	14
第3章	提案手法	15
3.1	はじめに	15
3.2	提案手法の概要	15
3.3	画像フィルタの適用	15
3.3.1	グレースケールフィルタの適用	16
3.3.2	勾配フィルタの適用	16
3.4	KAZE, AKAZE 特徴マッチング	18
3.5	むすび	19
第4章	提案手法の評価実験と結果, 考察	20
4.1	はじめに	20
4.2	実験の概要	20
4.2.1	実験条件	20

4.2.1.1	実験環境-----	20
4.2.1.2	実験使用機材と環境-----	20
4.2.1.3	画像データ-----	21
4.2.1.3.1	原画-----	21
4.2.1.3.2	グレースケールをかけた画像-----	23
4.2.1.3.3	勾配フィルタをかけた画像-----	24
4.2.1.3.4	データセット画像-----	26
4.3	グレースケールフィルタを用いた AKAZE 特徴マッチングによる指紋認証-	28
4.3.1	グレースケールフィルタを用いた AKAZE 特徴マッチングによる指紋認 証の結果-----	28
4.3.2	グレースケールフィルタを用いた AKAZE 特徴マッチングによる指紋認 証の考察-----	29
4.4	勾配フィルタを用いた AKAZE 特徴マッチングによる指紋認証-----	29
4.4.1	勾配フィルタを用いた AKAZE 特徴マッチングによる指紋認証の結果-	29
4.4.2	勾配フィルタを用いた AKAZE 特徴マッチングによる指紋認証の考察-	30
第5章	結論と今後の課題	32
5.1	結論-----	32
5.2	今後の課題-----	32
謝辞		34
参考文献		35
図一覧		37
表一覧		39

第1章 序論

1.1 研究の背景

近年、生体認証は様々な発展を遂げている[1]。生体認証の代表的な例には以下のものがある。主に認証の主流とされている指紋認証，手の中の血管の流れを認識する静脈認証，元々データベースに登録されている顔の特徴点を抽出し判別を行う顔認証，撮影した虹彩(アイリス)の画像からあらかじめ登録した特徴データで判別する虹彩認証，人間の声をデジタル化して判別を行う音声認証などである。その中でも指紋認証は，最も精度が高く，かつコスト面，安全面に優れることから数多くの場面で使われている。特に近年注目されている指紋認証方法は，デジタルカメラを用いた非接触型指紋認証である。

デジタルカメラを用いた指紋認証は従来の指紋認証とは異なり指紋を読み取る仕組みが異なる。従来の指紋認証ではボタンに指紋センサーが組み込まれており，指紋センサー内側には何万もの電極が埋め込まれている。指紋の表面には凹凸があるため，センサーの保護板に直接触れた凸部の下にある電極側に多くの電荷が蓄積される仕組みとなっている。理由は指に電気を通す電解質を含んでいるためである。この電荷が蓄積されている電極の位置をマッピングし，指紋の画像データが出来上がる。従って，デジタルカメラを用いて指紋認証を行う手法とは仕組みが大きく異なる[2]。

本論文で研究対象とする，デジタルカメラを用いた指紋認証では，カメラで撮影した画像を指紋のデータとして扱う構造をとる。さらに **Scale Invariant Feature Transform (SIFT)** 特徴マッチング等を組み合わせて新たな認証手法を提案し，実験により手法の評価を行う。

1.2 本研究の目的

デジタルカメラによる撮影画像を指紋認証の精度が向上すれば，潔癖症などの人々でも指紋認証を受けることが可能になる。これにより多くの人々がより容易に指紋認証を受けることができるようになると考えられる。

近年，認証手法として，大局的な類似度を算出できるテンプレートマッチングと局所的類似度を算出できる **SIFT** 特徴点マッチングが注目を浴びている。そこで本研究では，デジタルカメラの撮影画像を対象とし，**SIFT** アルゴリズムを改良し精度が向上した **Accelerated KAZE (AKAZE)** 特徴マッチングを用いた指紋認証手法による精度向上を図ることを目的とする。

1.3 関連研究

デジタルカメラを用いた指紋認証では，取得画像のセグメント化，強調，及びコアポイント検出のために，光学センサーまたは静電容量センサーによって取得された指紋画像の前処理を行う[1]。従来技術であるタッチレスの指紋認識の問題点は，指紋画像，デフォーカス，モーションブラーの尾根と谷のコントラストが低いことである。本研究では，ディジ

タルカメラを指紋採取装置として選択する。従来研究では、指紋画像は、短時間フーリエ変換(STFT)分析を用いて強調される部分を判定して重ね合わせることによってコアポイントの位置を特定し判断させる。しかし、現時点では正確なコアポイント検出は95.44%であり、誤ったコアポイント検出は4.56%であることが示されている[3]。これにより実用化にはより一層の改善が必要とされている。一方で、デジタルカメラを用いた画像から識別する手法の研究はあまり進められていないのが現状である。

1.4 本論文の構成

本論文の構成は以下のとおりである。

第1章は本章であり、本研究の背景や目的について述べる。

第2章では、本論文で用いる主要な認識技術である従来の指紋認証の技術や問題点 SIFT 特徴マッチング及び AKAZE, KAZE についての説明や用語の定義について述べる。

第3章では、本論文で提案する手法について述べる。

第4章では、実験概要、結果及び考察について述べる。

第5章では、本論文の結論と今後の課題について述べる。

第2章 指紋認証

2.1 はじめに

本章では、本論文で用いる指紋認証の手法や問題点、SIFT 特徴マッチングの手法や用語の定義及び説明する。

2.2 指紋認証とは

指紋認証とは序論でも述べた通り生体認証のうちの一つである[1]。指紋認証には様々な種類や方式が採用されている。例として光の反射によって認証を行う光学式、放電することにより指紋の凹凸を様々な手法で判断する静電容量方式と電解強度測定方式、圧力や温度で指紋の凹凸の溝を判定していく感圧式と感熱式が存在している。他に指紋の認証方式、センサーのタイプ、指紋情報の保管場所など様々な手法が存在しており、以上を組み合わせることによって高い認証精度と素早い認証速度を有する。指紋自体の形、模様なども様々な種類が存在する[2]。

2.2.1 従来の指紋認証

従来の指紋認証は精度が高く非常に利便性にも優れている。指紋認証の技術は非常に歴史が長いですが、一部の時代では、低い認証精度と低コスト重視の製品が出現した。そのため一時的に安全性や利便性が疑われるような印象を持たれた。しかし、現在では他の生体認証より着実に技術進歩と改善が進み、認証率はほぼ 100%に達しており、万が一に他人と間違っ て認識される可能性は 0.0002%以下とされている[4]。

2.2.2 従来の指紋認証の問題点

従来の指紋認証では、衛生面やメンテナンス面で問題があることが浮き彫りとなっている[5]。衛生面においてはカメラの部分に直接指を接触させるという行為に問題がある。特に潔癖症の人々にとっては知らない人が触れたものに触れたくないということが懸念される。最近のニュースのアンケート結果では、「女性の半分以上が『危ないからつり革をつかみたいができれば触りたくない』と回答し全体の 3 割近くは『絶対につかまない』と出ており[6]、深刻な問題となってきている。

デジタルカメラを用いる指紋認証、すなわち非接触型の指紋認識は、衛生的な観点及び、メンテナンス性の観点で、注目されている。指紋認証では、非対称性のセンサノイズを用いてデジタルカメラの識別をする。その場合、なるべく高解像度デジタルカメラ（またはタッチレス）で撮影を行う。その理由は、画像の歪みが低減できるためである。低い他人受入率（FAR）及び本人拒否率（FRR）を有する認証システムを構築するためには、歪みのない指紋画像の取得が望ましい。このため、ある程度遠隔から撮影されても、指紋情報を窃取されないための盗撮防止手法が提案されている[7]。

2.3 デジタルカメラを用いた指紋認証（従来研究）[3]

従来研究のデジタルカメラを用いた指紋認証の場合，小さい箱などの空間からテーブルランプ等の光源を決められた位置から照射し指が鮮明に映るようにする．画像のデータを容易に確認するため，単色で箱の空間を囲み指だけを正確に描写する．図 2.1 に実際に実験で用いられた環境を表す．箱のサイズは高さ 40cm 縦 25cm の横 19cm である．

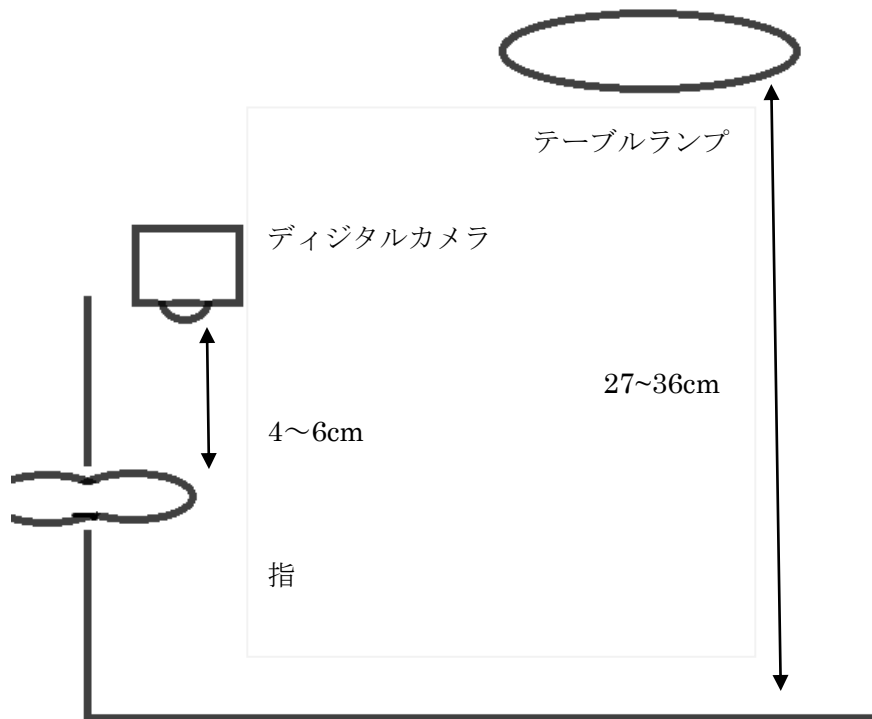


図 2.1 実行環境

捕捉された RGB フォーマットの指紋画像は，まずグレースケールに変換し，その後正規化処理を行う．平行して，指紋領域と知覚される領域を決定する．指紋バイナリマスクを作成するため，指紋候補領域に対して，形状保存を目的とした適応閾値処理を実行する．次に，正規化された画像に対し，指紋バイナリマスクを乗算する．得られた画像に対して，STFT に基づく画像特徴点抽出を行う．

次に画像取得に使用したカメラデバイスについて述べる．デジタルカメラは，8メガピクセルの有効解像度と 7 倍光学ズーム Canon "L" シリーズレンズを備えた Canon Power Shot Pro1 である．L シリーズレンズのマクロ機能により，高解像度の画像をキャプチャできるものを使用した．一方，1.2 インチに近い焦点を可能にする「スーパーマクロモード」は，明瞭な前景パターン及び背景パターンのぼけを可能にしている．テーブルランプから来

る光源は白色である。この静的光源は、得られたさ指紋画像の尾根及び谷の透明度を改善するのに十分である。画像取得プロセスの間、動きぼやけを減少させるために指は装置の壁の穴を通して挿入する形態となる。

RGB 画像から変換されたグレースケール画像に対して、不均一な照明の問題を低減するために正規化を行う。輝度正規化は重要であり、この設定を行わなければ、認識性能を低下させるため、この手順は非常に重要である。画像が正規化されていれば、STFT 分析を実行し、指紋画像を細分化してより細かく検討することが可能となる。グレースケール画像が STFT 分析で使用される場合、検出されたコア点と同様に、セグメント化され、強調された画像の認識結果は不十分であることが知られている。そのため、グレースケール画像に対して、正規化アルゴリズムを適用する。このアルゴリズムでは、以下のように、画像の平均及び分散を計算する。

$$g(x, y) = \frac{f(x, y) - m_f(x, y)}{\sigma_f(x, y)} \quad (2.1)$$

$f(x, y)$: 原画

$m_f(x, y)$: 原画の平均推定

$\sigma_f(x, y)$: 標準偏差の推定

これによって照度の差を減少させる。 $f(x, y)$ の正規化は、式(2.1)に従う。平均及び標準偏差の推定は、空間平滑化によって実行される。

望ましくないノイズの多い背景を除去し、入力データのサイズを縮小するために、指紋セグメンテーションが必要となる。まず、最も可能性の高い指紋領域、適応型閾値処理、形態学的処理を決定することによって指紋を分割する。考えられる指紋領域の決定は、画像を RGB 色空間から YCbCr 色空間に変換して実行する。分析において色差成分のみが使用される場合、肌色領域のセグメンテーションは強くなる。従って、YCbCr 色空間の CbCr 平面（色差成分）を選択して、可能性のある指紋領域を決定することによって、輝度成分の変動が除去される。YCbCr の Y に含まれる輝度成分は使用しない。次に、これらの画像を低域フィルタリングしてノイズを除去する。Cb と Cr の平均と共分散は、

$$\text{平均: } m = E(x), \quad x = [C_b C_r]^T \quad (2.2)$$

$$\text{共分散: } C = E[(x - m)(x - m)^T] \quad (2.3)$$

となる。指紋領域は以下のように表される。

$$P(x|\text{skin}) = \exp[-0.5(x - m)^T C^{-1}(x - m)] \quad (2.4)$$

可能性のある指紋領域が決定された後、画像は適応閾値処理によって二値画像に変換さ

れる。可能性のある指紋領域決定に続く適応閾値処理は、指紋領域をほとんど正確に検出するが、背景が皮膚の色に非常に似ている場合、失敗とみなされるため、セグメンテーション結果を危険にさらす。

この問題を解決するために次の手法を実行する。まず、すべてのピクセルに接続されたブロブには同じラベルを割り当てる。次により小さいサイズの接続されたブロブが選択されて削除された場合、最大のブロブのみが保持され、マスクとして動作させる。バイナリマスクに正規化画像を乗算し、細分化された正規化画像が切り取られる[5]。

指紋画像は、STFT 分析を用いて強調される。まず、画像が STFT 分析中に重なり合うウィンドウに分割される。オーバーラップウィンドウは、稜線の連続性を保持するために使用し、他のブロック処理画像操作に共通する「ブロック」効果を除去する。画像は、この小さなウィンドウ内で静止していると仮定され、およそ表面波としてモデル化することができる。隆起頻度、尾根の確率的推定も可能となる。この小領域のフーリエスペクトルを解析した後、方位とエネルギーマップが得られる。フーリエスペクトルは極座標形式 $F(r, \theta)$ で表される。このスペクトルは、確率密度関数 $p(r, \theta)$ と周辺濃度関数 $p(r)$, $p(\theta)$ で表される。方位は、確率密度関数 $p(r)$ を持つランダム変数 $p(\theta)$ と仮定される。方向の期待値を平滑化された画像 (x, y) 座標はベクトル平均化によって得られる。また、リッジ周波数は $p(\theta)$ のように確率密度関数 $p(r)$ を持つ確率変数であると仮定される。リッジ周波数の期待値は次のように求められる。

$$p(r, \theta) = \frac{|F(r, \theta)|^2}{\iint_{\theta} |F(r, \theta)|^2} \quad (2.5)$$

$$p(r) = \int_{\theta} p(r, \theta) d\theta, \quad p(\theta) = \int p(r, \theta) dr \quad (2.6)$$

方位 θ は、確率密度関数を有するランダム変数 $p(\theta)$ と仮定する。方程式の期待値を以下のように定義することが可能である。

$$E\{r\} = \frac{1}{2} \tan^{-1} \left\{ \frac{\int_{\theta} p(\theta) \sin(2\theta) d\theta}{\int_{\theta} p(\theta) \cos(2\theta) d\theta} \right\} \quad (2.7)$$

この式によって平滑化された画像は、ベクトル平均化によって得られる。隆起頻度は、確率密度関数 $p(r)$ を有する確率変数であるという形で仮定する。リッジ周波数の期待値は、次のように求める。

$$E\{r\} = \int p(r)rdr \quad (2.8)$$

周波数マップを滑らかにするため、 3×3 ガウシアンマスクを適用する。これにエネルギーマップを組み合わせる。エネルギーマップは細かい指紋セグメンテーションを得るための領域マスクとして使用される。領域マスクは最適閾値処理技術を用いて得られることが可能である。エネルギーマップは以下のような計算式から算出される。

$$E(x, y) = \log \left\{ \iint_{\theta} |F(r, \theta)|^2 \right\} \quad (2.9)$$

エネルギーマップによって算出された画像は、コヒーレンス画像を計算する際に使用され、ブロック境界での隆起流の不連続性、特に、コアに近く高い曲率の領域及び複数のデルタを有する。次にフーリエ領域では、 16×16 の重なり合ったブロックに分割された画像をフィルタリングするために、周波数及び方向選択フィルタを使用する。フィルタの式を以下のような形式で表す。

$$H_r(r) = \sqrt{\frac{(rr_{BW})^{2n}}{(rr_{BW})^{2n} + (r^2 - r_{BW}^2)^{2n}}} \quad (2.10)$$

$$H_{\phi}(\phi) = \begin{cases} \frac{\cos^2 \pi(\phi - \phi_c)}{2\phi_{BW}} & \text{if } |\phi - \phi_c| \leq \phi_{BW} \\ 0 & \text{otherwise} \end{cases} \quad (2.11)$$

r_{BW} : ラジアル帯域幅 ϕ_{BW} : 角度帯域幅 ϕ_c : 平均方向

指紋画像の最大曲率の点として、指紋の中心点を定義する。画像が強調された後、方位マップは、STFT 分析を用いて再び得られる。向きを持つブロックは、'1'とラベル付けされ、それ以外は '0'とマークされる。誤ったコアポイント検出をもたらす可能性のある偽点またはクラスタ領域を低減するために、モルフォロジー演算が実行される[8]。このプロセスの目的は、バイナリイメージ内のオブジェクトを滑らかにし、塗りつぶし、または除去することである。半径が3ピクセルの円盤状の構造要素が使用される。このステップでは、小さな領域が削除され、よりコンパクトなクラスタが作成される。これは、検索処理を高速化するのに役立つ。パスは '1'のブロックでのみトレースされ、傾きが0から遭遇する範囲にない場合にはマークされる。最後に、マーク数が最も多いブロックで負のy方向の傾きが計算し、コアポイントの位置を特定する。これによりデジタルカメラから取得した画像の判定をする。

2.3.1 デジタルカメラによる指紋認証の課題

デジタルカメラによる指紋認証は現在の時点では高い精度を誇っているが実用化できるほどの高精度ではなく問題も発生している。セグメント化，強調，及びコアポイント検出のために，光学センサーまたは静電容量センサーによって取得された指紋画像を前処理する従来の技術の完成度はまだ不十分である。非接触型指紋認証の問題点は，指紋画像そのものの取得や，デフォーカス，モーションブラーの尾根と谷のコントラストが低いことである。

2.4 SIFT マッチング[9]

SIFT マッチングとはスケールとキーポイントの検出をし，特徴量という画像内の輪郭，線の傾斜などを値としたものを算出して比較画像を結びつけるものである。一般的な使用用途として「異なる画像間でのマッチング」や「パノラマ画像の自動作成」，「一般物体認識」などが存在する。

SIFT 特徴マッチングのアルゴリズムに関して説明する。初めに特徴点と呼ばれる候補点を特定の位置から順を追って探索をする。探索が終了した後，その探索した候補点から絞り込み，特徴点を限定する。次に特徴の記述を行う。まず各特徴点の勾配を検出する。その後，各特徴点の勾配方向のヒストグラムの計算をする。一般的な SIFT による全体の流れのフローチャートを図 2.2 に示す。

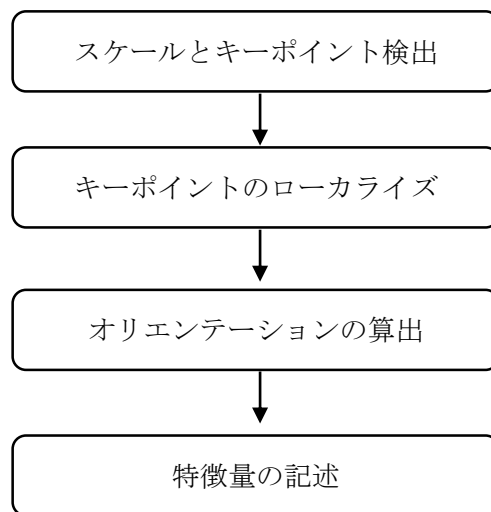


図 2.2 SIFT 特徴マッチングのフローチャート

SIFT におけるキーポイント検出の必要性は重要である。キーポイント検出では Difference of Gaussian (DoG) フィルタを用いることによってスケールスペースという画像による成分を平滑化し，処理を行うことにより 3 次元空間で画像を解析する。極値探索と呼ばれる偏導関数を用いずに極小値の探索し，キーポイントの位置とスケールを決定する。

平滑化とは画像にガウシアンフィルタを用いてぼかすことである。ガウシアンフィルタ

では注目要素の周りの倍率は離散化を用いて決定する．ガウシアンフィルタを真上から表す，実質的な適用範囲は円形になる．その円形をガウス窓と表現している．スケール値が大きくなるとガウス窓が大きくなる．その結果，計算量も莫大となる．図 2.3 にガウス窓によるガウシアンフィルタとスケール変化によるガウス窓の変化を表す．

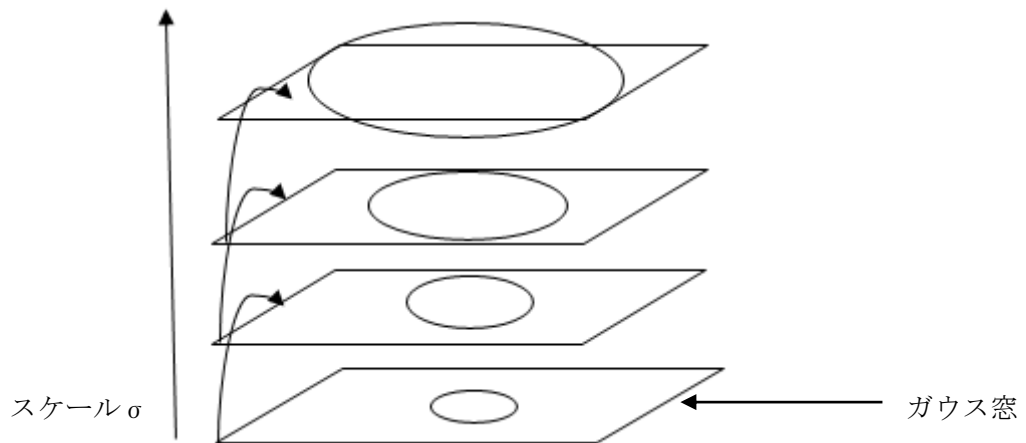


図 2.3 スケール変化によるガウス窓

スケールをいかにして追加するかは平滑化という部分が重要となる．Laplacian-of-Gaussian(LoG)によるスケール探索においてはスケール σ を変化させた LoG オペレータを適用する．LoG オペレータの計算式は以下に示す．

$$\text{LoG オペレータ: LoG} = f(\sigma) = -\frac{x^2+y^2-2\sigma^2}{2\pi\sigma^6} \exp\left(-\frac{x^2+y^2}{2\sigma^2}\right) \quad (2.12)$$

σ : スケール

x, y : 注目画素からの距離

LoG によるスケール探索を行う場合，特徴点の検出やスケール探索による計算コストが高くなる．LoG は，平滑化の量がわずかに異なる 2つのガウシアンフィルタ画像の差分として近似できる．しかし，スケールスペースの生成には入力画像とそのダウンサンプル結果に対し，多くの畳み込み演算を行う必要があるため計算量が多くなる．

LoG フィルタの問題点を解決するためには特徴点の検出とスケール探索を同時に行うことにより計算コストを抑えられ，なおかつ LoG と同じ効果を得ることができる DoG フィルタが存在する．

SIFT における特徴量とは，異なるスケールの平衡化画像の差分画像である DoG 画像の極致を指す，現画像を $I(u, v)$ ，ガウシアンカーネルを $G(x, y, \sigma)$ と定義する．その結果によって得られる平滑化画像を $L(u, v, \sigma)$ は以下の式で定義される．

$$D(u, v, \sigma) = L(u, v, k\sigma) - L(u, v, \sigma) \quad (2.13)$$

$$L(u, v, \sigma) = G(x, y, \sigma) \times I(u, v) \quad (2.14)$$

$$G = (x, y, \sigma) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right) \quad (2.15)$$

原理としては **Scale-normalized Laplacian of Gaussian(LoG)**の最大値が得られるスケールを、その領域の基準スケールとする場合がある。だが、先ほども計算式で述べた通り **LoG**の計算量は膨大である。そのため、**DoG**による近似と画像のダウンサンプリングを行う。これにより、大幅に計算量を削減する。**SIFT**では画像のマッチングを行うために各画像で抽出した特徴点同士で、総当たり比較を行う。比較はユークリッド距離を用いて行う。以下にユークリッド距離の計算式を表す[10]。

$$d(v^{k_{I_1}}, v^{k_{I_2}}) = \sqrt{\sum_{i=1}^{128} (v^{k_{I_1}} - v^{k_{I_2}})^2} \quad (2.16)$$

k : キーポイント

v^k : キーポイント特徴量

この計算式では d が最も小さい点同士を対応点として比較を行う。場合によっては制約をつけることによりマッチングを行う。

キーポイントのローカライズを行う際にキーポイントに向かない点がいくつか存在する。エッジ上の点や **DoG** 出力の小さい点である。2つの点はノイズの影響を受けやすい。そのためキーポイント候補点のサブピクセルでの **DoG** 出力値を再計算することによりコントラストによるしきい値の処理を行う。

オリエンテーションの算出を行う際にはキーポイントのオリエンテーションとして向きに最適化を行うことにより回転の不変な特徴量を算出する。キーポイントが検出された平滑化画像 $L(u, v)$ の勾配速度 $m(u, v)$ と勾配方向 $\theta(u, v)$ を算出する場合以下の式で表せる。

$$m(u, v) = \sqrt{(L(u+1, v) - L(u-1, v))^2 + (L(u, v+1) - L(u, v-1))^2} \quad (2.17)$$

$$\theta(u, v) = \tan^{-1} \frac{L(u, v+1) - L(u, v-1)}{L(u+1, v) - L(u-1, v)} \quad (2.18)$$

式(2.17)と式(2.18)を用いて勾配情報から勾配方向ヒストグラムを作成する。全方向を 36 方向に離散化させることにより、キーポイントの持つスケールに対応した領域が確定し、勾配を算出することができる。次にガウス窓と勾配強度から重みをヒストグラムに加算し、勾配方向ヒストグラムを作成する。

特徴記述では、オリエンテーションの割り当てでは勾配方向ヒストグラムからキーポイントの代表的なオリエンテーションを算出する。特徴量の記述をする際にキーポイントが持つオリエンテーションに合わせて回転させ、回転に対する不変性を得る。特徴ベクトルを算出する手順は、まず周辺領域を一边 4 ブロックの計 16 ブロックに分割する。次にブロックごとに 8 方向の 45 度ずつの勾配方向のヒストグラムを作成する。これにより、ヒストグラムの各方向が特徴ベクトルになる。結果的に周辺領域の 16 ブロックと勾配方向の 8 方向を加算し、128 次元と表すことができる。

SIFT ではこれらの組み合わせを行い、上記の計算式を用いる。従来の手法では SIFT を高次元に拡張し SIFT アルゴリズムから派生し、より高い性能を示す N-SFIT などの改良されたマッチング手法も多数存在している。

2.4.1 その他のキーポイント検出タイプについて

SIFT は複数存在している DoG 画像からキーポイントの検出によりマッチングを実行する。そのため、同時にキーポイントを中心とした特徴量記述を行う範囲を表すスケールも検討する必要がある。DoG 画像の作成は計算コストが高いという問題があり、さらに複数の DoG 画像を一時的に保管するメモリ量が多いという問題がある。そのため、SIFT アルゴリズムが発表されて以降、様々なアプローチによって特徴量アルゴリズムが進化し続けている[10]。以下で説明をする。キーポイント検出タイプの時代による変遷を図 2.6 に表す。

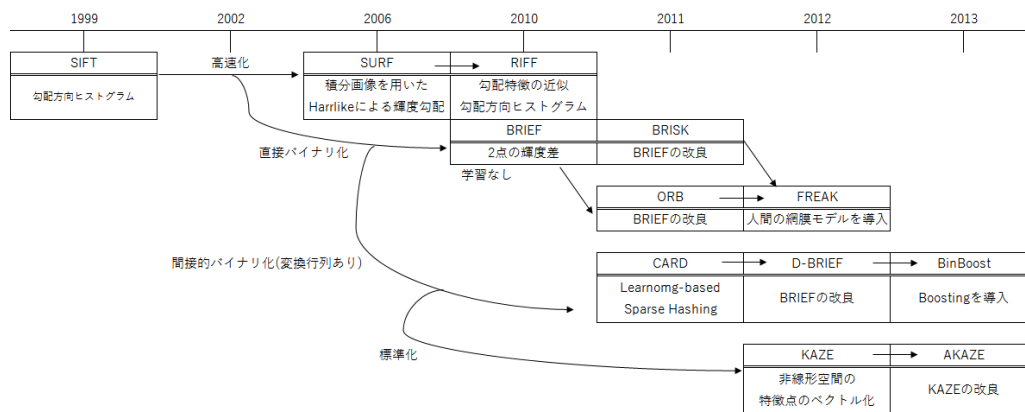


図 2.4 SIFT マッチングの変遷

2.4.1.1 SURF について

SURF では積分画像を駆使する。Box フィルタを用いて高速化実現により性能の向上を図っている。SIFT はキーポイント抽出とスケール検出によってガウス関数と異なる DoG 処理を何回も行う必要があるため処理に負担がかかる。原因としてこの処理には SIFT アルゴリズムの計算で非常に大きな負荷がかけられているためである。

このアルゴリズムでは、キーポイント抽出とスケール検出の両方を Hessian-Laplace 検

出に Box フィルタを組み合わせることで近似して求めている[9]。これにより積分画像の高速化が行われる。

2.4.1.2 FAST

類似したアルゴリズムでコーナーに特化したコーナーだけのキーポイントを対象とし、省メモリ化を図った SIFT 派生版も存在している。FAST では、決定木により高速かつ効率的なキーポイント検出の実行に成功している。多くのコーナー検出手法ではコーナーらしさのレスポンス値の局所を最大にすることで両端のコーナー点を得ることができる。しかし、決定木による検出アルゴリズムではレスポンスの値を出力できない。そこで画素値と連続する円弧上の画素値の差を比較し、大きくなれば高くなると定義する。FAST は検出する部位が限定されているが、検出速度が非常に高いという利点がある。2013 年にはこのアルゴリズムに改良を加えた Cascaded FAST が開発されている[10]。

2.4.1.3 ORB

ORB とはアルゴリズムにおいて特徴点、特徴量を抽出するアルゴリズムであり、移動、回転ズームのすべてにおいて頑健である。

SIFT においてアルゴリズムが移動と回転に加えてズームに対するスケール変化にも対応できるようになった。しかし現在の SIFT では計算量の膨大さやそこから生じる処理速度の遅さから SURF が提案されてきた。しかし、どちらも特許で守られているため使用する際には特許料を支払いが生じる場合がある[9]。そのためフリーでも使用することができ、なおかつ移動と回転、スケール変化に対して対応が可能で計算速度も速くフリーで使用することが可能なアルゴリズム KAZE が 2011 年に提案された。

2.4.1.4 KAZE, AKAZE マッチング[11]

ECCV2012 において発表された KAZE と、高速化を加えて改良を行った Accelerated KAZE (AKAZE)が存在する。KAZE や AKAZE のベースとなっている技術は SIFT や SURF である。ガウシアンフィルタによるスケールスペースはガウシアンフィルタが等方的であるため、オブジェクトのエッジをぼかした処理となるため、局所的な特徴を容易に検出できない[12]。そのため AKAZE や KAZE では、非線形で非局所的なスケールスペースを用いることでこれらの問題を解決している。

さらに AKAZE では特徴記述子を定義し Modified-Local Difference Binary(M-LDB)という独自の記述子を使用する[13]。また、ピラミッド構造の計算を高速化するために独自の工夫を組み入れ、ロバスト性の向上と速度の高速性を図っている。ロバスト性についてはスケールや回転、高度、アフィンのすべてにおいて SIFT, FAST, SURF や KAZE, AKAZE が最も性能が良いとされており、正確なトラッキングが可能であることが期待されている。

2.4.1.4.1 非線形拡散フィルタリング

KAZE, AKAZE では非線形空間において、輝度の拡散に偏微分方程式を用いて以下の式で表している。L を輝度、(x, y, z) を画像座標としている。

$$\frac{\partial L}{\partial t} = \text{div}((x, y, z) \cdot \nabla L) \quad (2.20)$$

さらに式(2.20)と AOS 手法を用いて、ベクトル行列で表した方程式の離散化をする。A_l は各次元における画像伝導率を行列で表している。

$$\frac{L^{i+1} - L^i}{\tau} = \sum_{l=1}^m A_l(L^i) L^{i+1} \quad (2.21)$$

$$L^{i+1} = \left(I - \tau \sum_{l=1}^m A_l(L^i) \right)^{-1} L^i \quad (2.22)$$

式(2.21)と式(2.22)を用いることにより、ガウスの消去法と線型方程式を組み合わせることで高速化を図る。

2.4.1.4.2 非線形スケールスペース

KAZE, AKAZE で特徴点を抽出する記述子は離散化スケールスペースではオクターブ O とサブスケール S によって連続で整理される[14]。オクターブ O とサブレベル S, スケール σ は以下の式のように表す。σ₀ はベーススケールレベル、N は処理時の画像枚数を表す。

$$\begin{aligned} \sigma_0(o, s) &= \sigma_0 2^{\frac{o+s}{S}} & (2.23) \\ o &\in [0, 1, \dots, O-1] \\ s &\in [0, 1, \dots, S-1] \\ i &\in [0, 1, \dots, N] \end{aligned}$$

式(2.23)はピクセル単位で表されている。拡散フィルタは時間軸として定義される。よって以下の式で時間軸に表す。

$$t_i = \frac{1}{2}, (i = 0, 1, \dots, N) \quad (2.24)$$

非線形スケールスペースで各フィルタ画像と t_i は一致しない。しかし、KAZE, AKAZE で

はガウス空間で拡散関数を 1 と等しくすることにより一致する。

実際の処理ではまず元画像のヒストグラムからコントラストパラメータ k と進行時間 t_i を計算する。次に標準偏差 σ_0 のガウス関数を用いて畳み込みを行い、AOS 手法を用いて非線形空間を形成する。

2.4.1.4.3 特徴記述

KAZE, AKAZE を用いて特徴点を検索するために、複数のスケールレベルにおいて、ヘッセル行列に存在する特徴点とその空間を特徴付けた際に発生する反応を以下の微分演算子の式を用いて抽出する。

$$L_{Hessian} = \sigma^2(L_{xx}L_{yy} - L_{xy}^2) \quad (2.25)$$

異なるスケールレベルでの反応を空間の最大値とその位置を用いて解析する。特徴点の位置は式(2.23)によりサブピクセルレベルで推測する。

特徴量記述をする際に、記述子を構築するために各特徴点での方向を算出する必要がある。各特徴点の半径までの周囲を探索し、ガウス関数によって重み付けする。導関数の反応がベクトル空間内の点として表記される[15]。

記述子を構築する際に M-LDB の記述子に非線形空間のフレームワークを追加し、構築する。特徴点をスケールで記述するため、大きさは正方形のグリッド上で算出する。グリッドをさらに副領域に分割し、各領域での導関数の反応を副領域の中心で計算したガウス関数によって重み付けし、記述子ベクトルに合算する[13]。副領域の記述子ベクトルは、特徴点を中心としたガウス関数によって重み付けされる。特徴点の方向検出では、各領域の記述子ベクトルも方向に基づいて回転するためコントラストに左右されない。

2.5 むすび

本章では、本論文で用いる主要な認識技術である従来の指紋認証の技術や問題点 SIFT 特徴マッチング及び AKAZE, KAZE について説明すると共に、用語の定義について述べた。

まず、SIFT における基本的な概念や用語について説明した。次に、基本構造や、各特徴量アルゴリズムの性能について説明した。次章では、本章で説明した手法や特徴量を用いた提案手法についての説明を行う。

第3章 提案手法

3.1 はじめに

本章では、デジタルカメラを用いた指紋認証を目的として勾配フィルタを用いた輪郭形成を行い、キーポイントを検出し、AKAZEを用いた認証手法を提案する。以下、処理の詳細について説明する。

3.2 提案手法の概要

提案手法は、デジタルカメラを用いた指紋認証において SIFT 特徴マッチングのアルゴリズムを用いた AKAZE のシステムと勾配フィルタを組み合わせた手法である。

事前処理で入力画像をグレースケール画像に抽出し、色情報を限定する。指の指紋を抽出するために指紋の凹凸を識別しやすくするために勾配フィルタをかける。

これにより、指紋画像の尾根と谷のコントラストが低いことによる認証精度の低減を解決させ、認証精度を向上させることを目標としている。次に AKAZE マッチングを用いて認証を行う。その後、キーポイントを検出、比較を行い、結果を出力する。提案手法の概要を図 3.1 に示す。

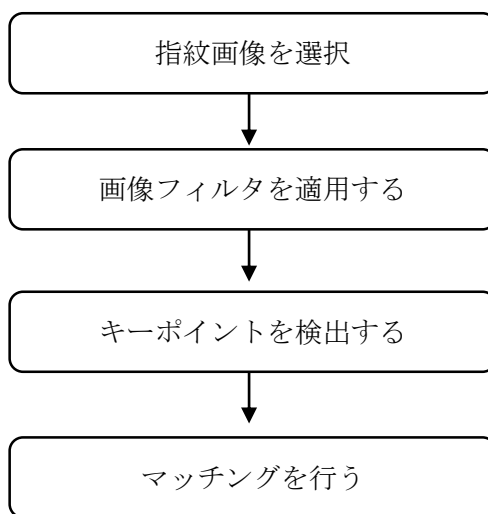


図 3.1 デジタルカメラを用いた指紋認証のフローチャート

3.3 画像フィルタの適用

抽出した画像に対して、以下のように処理を実行する。

1. 指の原画像を抽出する。
2. 抽出した画像からグレースケール画像を形成する。
3. グレースケール画像にした写真に勾配フィルタをかける。

画像の処理については以下について詳しく述べる。

3.3.1 グレースケールフィルタの適用

抽出した画像からグレースケール画像を適用させるため、グレースケールフィルタを適用する。グレースケールとは、画像などに使われる色の種類と範囲を表す手法の一つで、白と黒と中間の色の何段階かの灰色を用いる。明るい色から暗い色まで灰色をいくつ用意するかにより表現の精彩さが異なる。コンピュータで扱う際は灰色の階調の細かさをビット数で表す[16]。

1ビットの場合は白と黒のみで中間の灰色が無い状態で、グレースケール2値（モノクロ2値）とも呼ばれる。通常は8ビットや16ビットが用いる。これにより、元画像の色彩情報量を白と黒のみに限定することによって認証をスムーズに行わせることを目的としている。さらにグレースケール画像によって、勾配フィルタを容易にかけることができる。

3.3.2 勾配フィルタの適用

抽出したグレースケール画像から勾配フィルタをかけることにより、平面画像から指の指紋の凹凸を表示させることが可能となる。ここで勾配フィルタについて説明する。勾配フィルタは空間1次微分を計算し、輪郭を検出するフィルタである[17]。画像の勾配は垂直導関数を要素とする2次元ベクトルであり、以下の式を用いる。

$$\nabla f = \begin{pmatrix} f_x \\ f_y \end{pmatrix} \quad (3.1)$$

$$|\nabla f| = \sqrt{f_x^2 + f_y^2} \quad (3.2)$$

x, y 平面は水平面とし、画像の明るさを地面の高さとする。これにより勾配は (x, y) の座標で斜面が最も急な方向の傾斜を表すことができる。これにより、画像の濃淡が急激に変化するエッジ部分を検出することができ、光を当てることによって発生する指紋の凹凸によるコントラストも検出できる[17]。

勾配フィルタでは右、左、上、下、上、左下、左上、右上、右下の8方向を行列の計算を用いて選択できる。今回は輪郭を正確に検知するため、光源の向きに合わせて上から下に向かう1方向の検出を選択し、画像の抽出を行った。

実際の例をあげる。図3.2を原画、図3.3にグレースケール画像、図3.4に勾配フィルタをかけた画像例を示す。

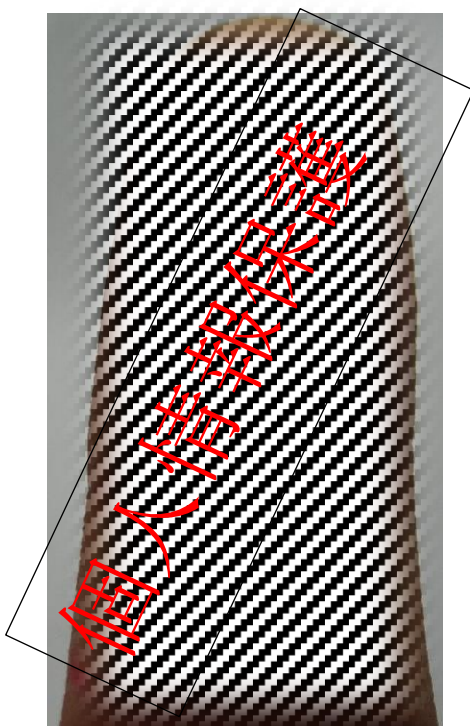


図 3.2 原画の指紋画像の例

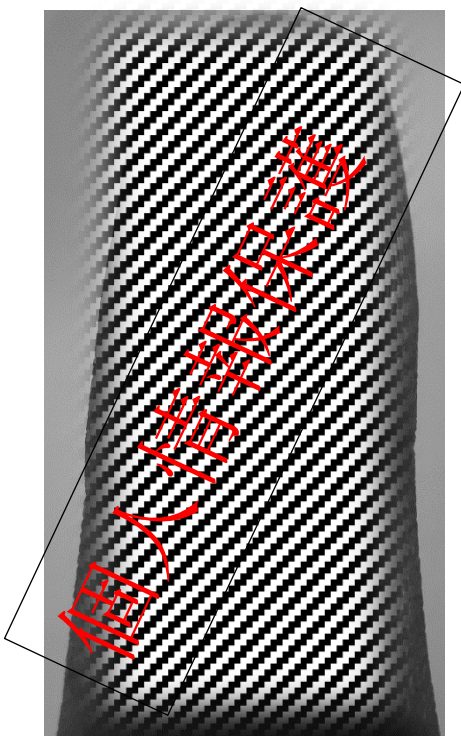


図 3.3 グレースケールフィルタをかけた指紋画像の例

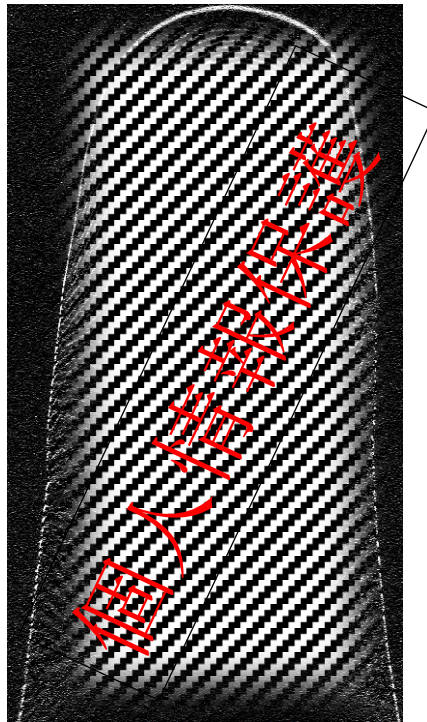


図 3.4 勾配フィルタをかけた指紋画像の例

この図より勾配フィルタでは平面画像から凹凸の判別が可能だと分かる．よって本論文で研究対象とする実験でもこのフィルタを用いる．

3.4 KAZE, AKAZE マッチング

第 2 章で説明した SIFT マッチング手法のアルゴリズムより性能面などを考慮して AKAZE マッチングを使用する．提案手法の識別は以下の処理を行うことにより実行する．

1. AKAZE マッチングを用いてキーポイントを検出する
2. 対象 1, 2 のキーポイント検出が終わりしだい識別を行う
3. 識別の終わった結果画像とキーポイント検出量（総数）を算出する．

まず AKAZE マッチングによるキーポイントの検出を行うこの時に使用するのは先ほどの勾配フィルタをかけた画像である．フィルタをかけて指紋の凹凸のコントラストが鮮明に出ている写真に AKAZE マッチングを用いてキーポイント量を算出する．実際の指紋認証では特徴数は約 20~40 個が適切である[18]．よって,その結果に準拠しキーポイントを算出する．

勾配フィルタをかけた画像から，キーポイントをすべて表示させるのではなく認証の優先度の高い個数 50 個を検出する．AKAZE マッチングのキーポイント検出はアルゴリズムの中で最も精密かつ検出量が多いからである[9]．

さらに AKAZE は, KAZE を高速に改良したアルゴリズムである. KAZE, AKAZE は, SIFT や SURF と同様に, 拡大縮小・回転・照明変化に強いロバストな特徴量抽出アルゴリズムである. KAZE, AKAZE は輝度変化が少ない部分でも特徴量を抽出できる[14].

各対象物のキーポイントの検出が終わり次第, 実際のキーポイント同士をマッチングさせる. マッチングが終了次第, 処理結果画像とキーポイント検出量を算出する. これによって指紋が本人であるかを判別する.

3.5 むすび

本章ではデジタルカメラを用いた指紋認証を目的として勾配フィルタを用いた輪郭形成を行い, キーポイントを検出し, AKAZE を用いた認証手法を提案した.

第 4 章では, 本章で説明した提案手法の評価実験を行う.

第4章 提案手法の実験と結果, 及び考察

4.1 はじめに

本章では, デジタルカメラで撮影した指紋画像を用いて第 3 章で説明した提案手法の実験を行う.

4.2 実験の概要

提案手法の実験として, 二つの実験を行う. 一つ目の実験では, 指紋画像を正面から撮影し, グレースケールフィルタをかける. あらかじめ登録してあるデータセットと照合を行い正しい画像を選択できるかを判別する. 二つ目の実験では, 一つ目の実験と同様の形式で勾配フィルタを用いて照合を行い, 正しい指紋画像を選択出来ているかを判別する.

4.2.1 実験の条件

実験で行う環境や条件に付いてを以下に示す

4.2.1.1 実験環境

実験に使用した環境を以下の表 4.1 に示す. また, AKAZE マッチング及び勾配フィルタのプログラムは OpenCV3 を用いて使用した

表 4.1 実験環境

OS	Windows 10 Pro
CPU	Intel(R) Xeon(R) CPU E3-1240 v3 @ 3.40GHz
GPU	NVIDIA Quadro K600

4.2.1.2 実験使用機材と環境

実験によって使用する機材を表 4.2 に示す. なお実験環境は従来手法と同じ形式で実験を行った[3].

表 4.2 実験機材

型番	DSC-HX200V
画素数(有効画素)	1890 万画素(1820 万画素)
撮影素子	1/2.3 型 CMOS (裏面照射型)
焦点距離	27~810mm
光学ズーム	30 倍

4.2.1.3 画像データ

評価, 実験を用いる指紋画像は手の指（左右 10 本）の指紋を撮影し, 切り出した画像を使用する. 指紋画像はそれぞれ別の指紋である. 認証をする際に同じ仕様で撮影し勾配フィルタを加えた画像を 10 枚使用する. 画像の内訳を表 4.3 に示す. また各指紋画像とフィルタごとの指紋画像を図 4.1-4.35 にそれぞれ示す.

表 4.3 実験環境

指紋画像	画像数
原画	10
グレースケール画像	10
勾配フィルタ	10
認証用(グレースケール画像)	10
認証用(勾配フィルタ)	10

4.2.1.3.1 原画像

原画の指紋画像を図 4.1-4.10 に示す.



図 4.1 原画の指紋画像(左人差し指)

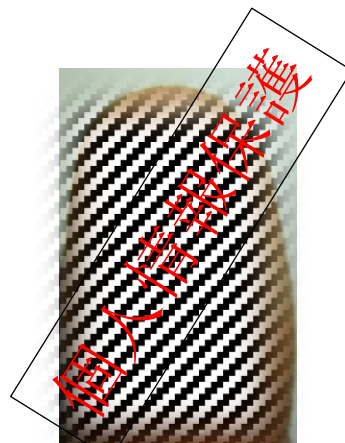


図 4.2 原画の指紋画像(右人差し指)

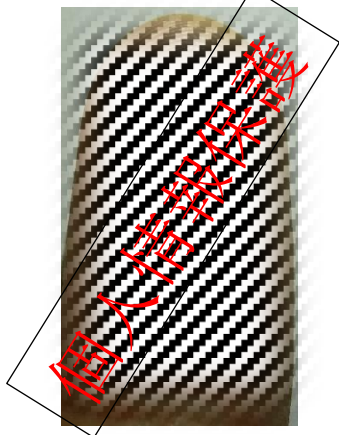


図 4.3 原画の指紋画像(左小指)

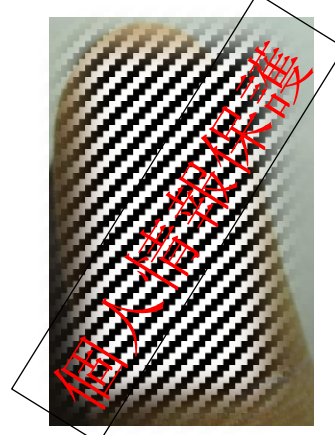


図 4.4 原画の指紋画像(右小指)

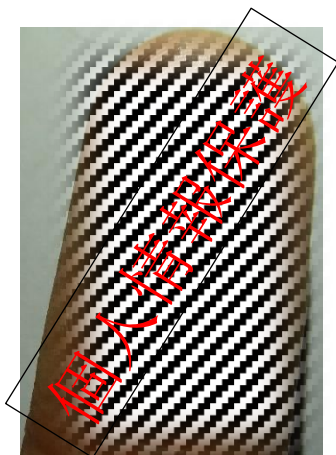


図 4.5 原画の指紋画像(左薬指)

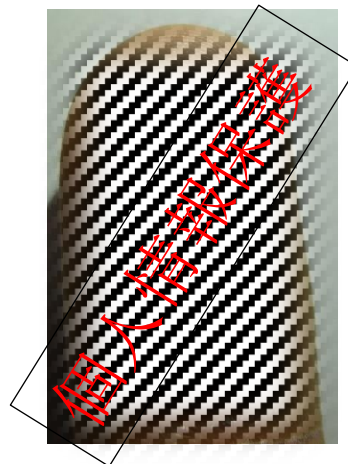


図 4.6 原画の指紋画像(右薬指)



図 4.7 原画の指紋画像(左中指)



図 4.8 原画の指紋画像(右中指)

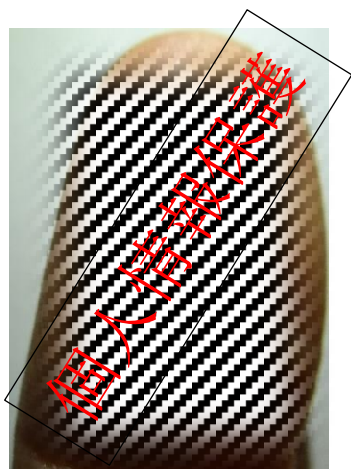


図 4.9 原画の指紋画像(左親指)

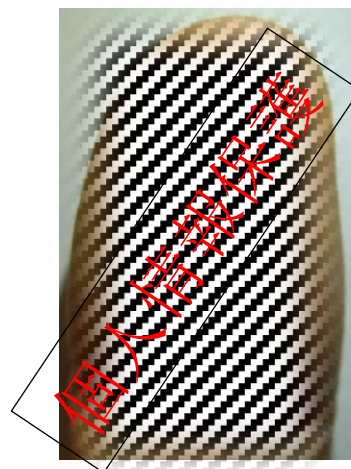


図 4.10 原画の指紋画像(右親指)

4.2.1.3.2 グレースケール画像

原画像にグレースケールフィルタをかけた画像を図 4.11-4.20 に示す。

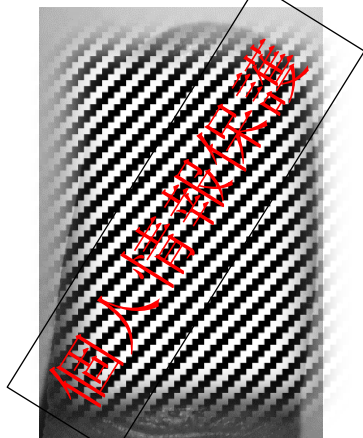


図 4.11 グレースケールの指紋画像(左人差し指)
(右人差し指)

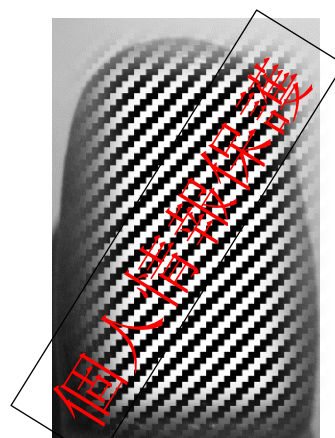


図 4.12 グレースケールの指紋画像

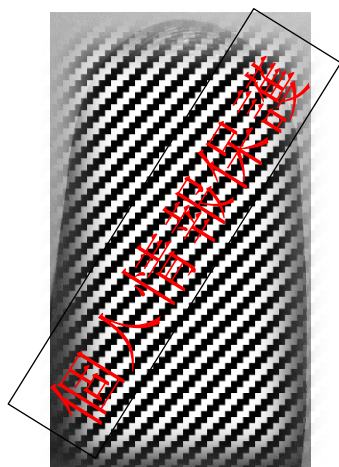


図 4.13 グレースケールの指紋画像(左小指)
(右小指)



図 4.14 グレースケールの指紋画像

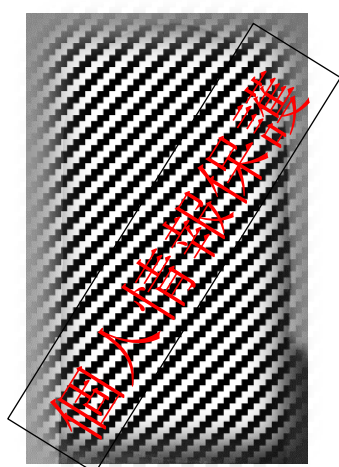


図 4.15 グレースケールの指紋画像(左薬指)

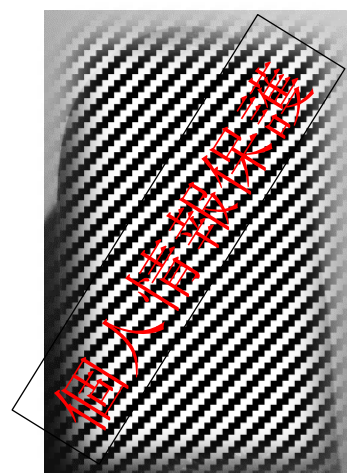


図 4.16 グレースケールの指紋画像

(右薬指)

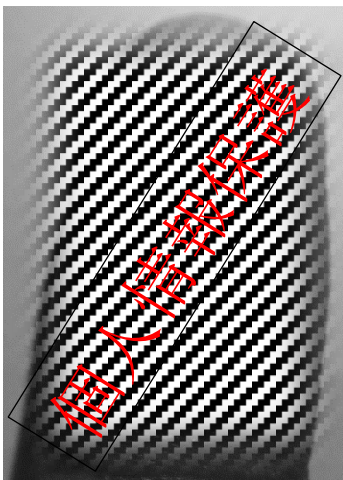


図 4.17 グレースケールの指紋画像(左中指)
(右中指)

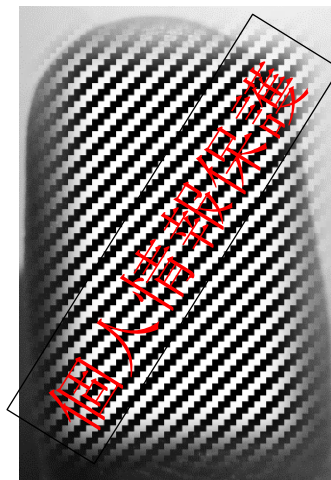


図 4.18 グレースケールの指紋画像

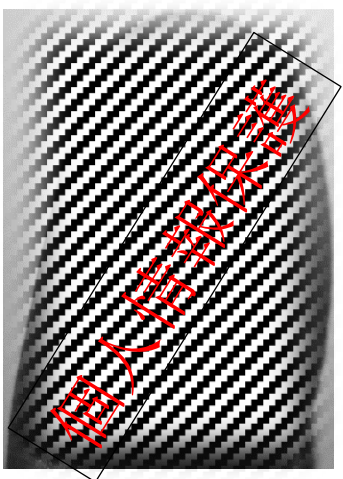


図 4.19 グレースケールの指紋画像(左中指)
(右中指)

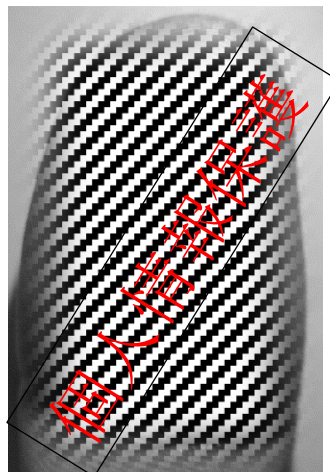


図 4.20 グレースケールの指紋画像

4.2.1.3.3 勾配フィルタをかけた画像

グレースケール画像に勾配フィルタをかけた指紋画像を図 4.21-4.30 に示す.

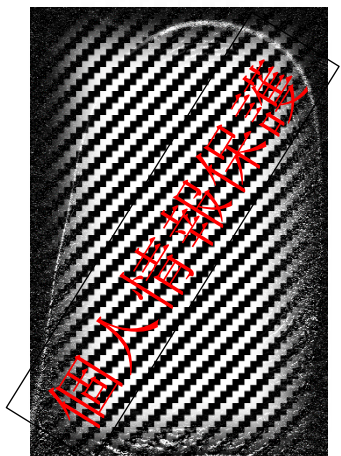


図 4.21 勾配フィルタの指紋画像(左人差し指)
(右人差し指)



図 4.22 勾配フィルタの指紋画像

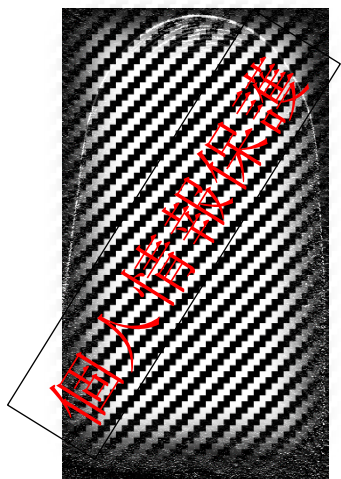


図 4.23 勾配フィルタの指紋画像(左小指)
(右小指)

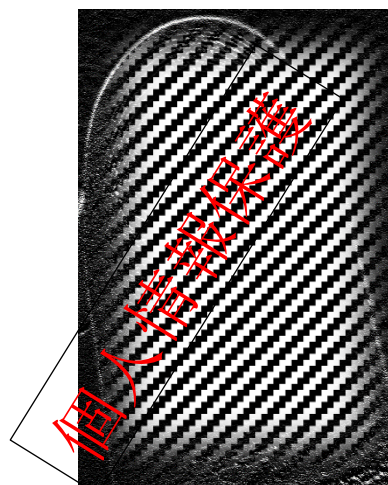


図 4.24 勾配フィルタの指紋画像

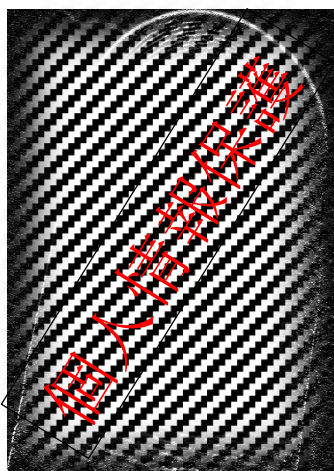


図 4.25 勾配フィルタの指紋画像(左薬指)
(右薬指)



図 4.26 勾配フィルタの指紋画像

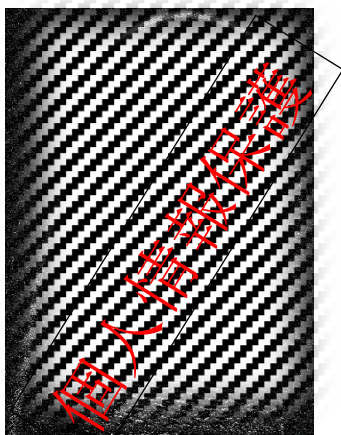


図 4.27 勾配フィルタの指紋画像(左中指)
(右中指)

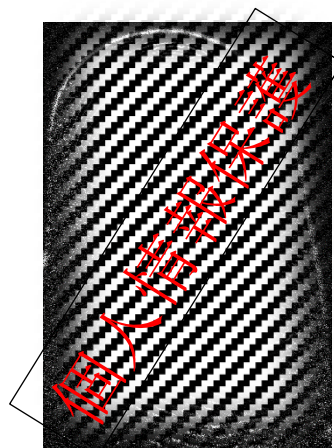


図 4.28 勾配フィルタの指紋画像



図 4.29 勾配フィルタの指紋画像(左中指)
(右中指)

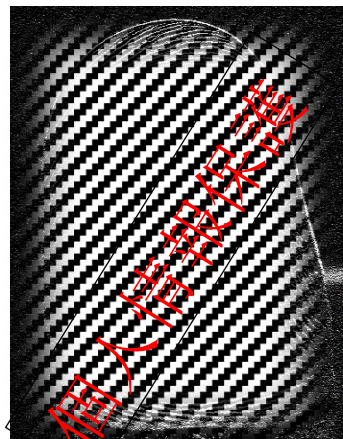


図 4.30 勾配フィルタの指紋画像

4.2.1.3.4 データセット画像

図 4.11~4.20 と図 4.21~4.30 までの指紋データを一つの画像にまとめ認証用として用いる。今回の実験では画像のサイズを統一せずにそれぞれカメラから 4~6cm 離れた位置からの指のサイズで組み合わせた。グレースケール画像と勾配フィルタをそれぞれかけた認証用の画像セットを図 4.36~4.37 に表す。



図 4.36 認証用の画像セット (グレースケール画像)



図 4.36 認証用の画像セット (勾配フィルタ画像)

データセットの指紋の配置は左上から右に順番に人差し指，小指，薬指，中指，親指となっている．一部の指紋が回転しているが，SIFT の性能面において回転や拡大，縮小に強い

ことから誤差の範囲内であると考えられる[9].

4.3 グレースケールフィルタを用いた AKAZE 特徴マッチングによる指紋認証

実験1では, 各指紋のデータセットをデジタルカメラの撮影により作成する. 本実験ではまず指を10本用意する. 次に指定の指を撮影し, データセットにグレースケール画像のフィルタをかけて照合できるかを判定する. さらに, その時点の総キーポイント数と指紋が正解の指紋と結びついているかの正解, 不正解数を計測する.

4.3.1 グレースケールフィルタを用いた AKAZE 特徴マッチングによる指紋認証の結果

提案手法の AKAZE 特徴マッチングを用いてデータセットと再撮影した指紋を用いて判別を行った. 正しい指を選択した点を正解点, 別の指を選択した点を不正解点とし, 結果を表4.4に示す. 実際の認証例を図4.37に示す左側に任意の指紋を表示し, キーポイントマッチングで優先度が高い50個の点をもとにそれぞれの点に繋いでいる.

ここで, 正解率は以下の式より算出する.

$$\text{正解率(\%)} = \frac{\text{正解点数}}{50} \times 100(\%) \quad (4.1)$$

表 4.4 グレースケールフィルタを用いた指紋認証によるマッチング結果

指	正解点数	不正解点数	総特徴点数	正解率(%)
左人差し指	50	0	915	100
右人差し指	48	2	721	96
左小指	40	10	100	80
右小指	27	23	156	54
左薬指	49	1	609	98
右薬指	49	1	503	98
左中指	50	0	188	100
右中指	38	12	1313	76
左親指	49	1	639	98
右親指	50	0	559	100



図 4.37 グレースケールフィルタを用いた認証例

指の回転や撮影位置などによって総キーポイント検出数や正解点が変わってくるが、半数以上が正解の指紋に繋げることができた。

4.3.2 グレースケールフィルタを用いた AKAZE 特徴マッチングによる指紋認証の考察

提案手法の正解率を比較したところ、10 本中 7 本の指が 96~100%の正解率を得ることが示された。しかし、撮影の方式によっては指自身が回転してしまい指紋のズレなどが生じた。そのため、正解点の取得数が低く正解率も低かった。データセット以外の指紋を選択したところ、キーポイントの検出が分散した。

グレースケールによって輝度を白から黒の多階調のグレーで表現することにより、容易に認証できた。AKAZE 特徴マッチングの特徴量を細かく分割された非線形空間にある特徴点の導関数を求め、重み付けした各データをベクトル化し算出する特性により局所的な特徴を検出できた。

4.4 勾配フィルタを用いた AKAZE 特徴マッチングによる指紋認証

実験 2 では勾配フィルタをかけた画像で実験 1 と同様の実験を行い、実験 1 と同じ数値を計測する。なお使用するデータセットの画像は実験 1 と同じ画像を使用する。

4.4.1 勾配フィルタを用いた AKAZE 特徴マッチングによる指紋認証の結果

提案手法の AKAZE 特徴マッチングを用いてデータセットと再撮影した認証用指紋を用いて識別を行った。正しい指を選択した点を正解点、別の指を選択した点を不正解点として結果を表 4.4 に示す。実際の認証例を図 4.38 に示す左側に任意の指紋を表示しそこからキーポイントマッチングで優先度が高い 50 個の点をもとにそれぞれの点に繋いでいる。

表 4.5 勾配フィルタを用いた指紋認証によるマッチング結果

指	正解指数	不正解指数	総特徴点数	正解率(%)
左人差し指	3	47	65310	6
右人差し指	1	49	64833	2
左小指	1	49	65548	2
右小指	0	50	63403	0
左薬指	1	49	65548	2
右薬指	3	47	64833	6
左中指	1	49	65548	2
右中指	5	45	64403	10
左親指	5	45	63164	10
右親指	4	46	62926	8

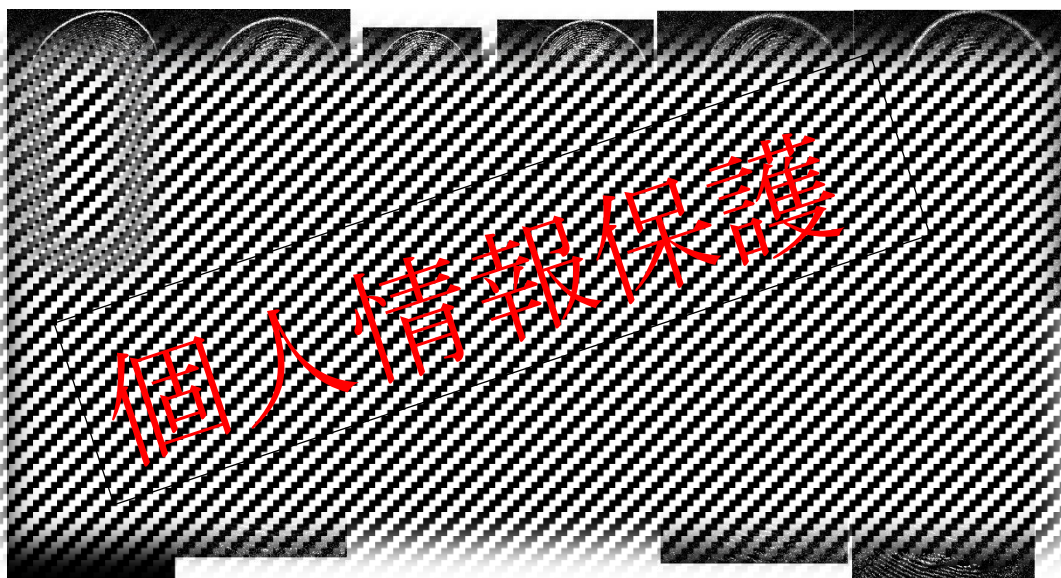


図 4.38 勾配フィルタを用いた認証例

総キーポイント検出数が実験 1 よりもとても多く、認証時間も 10 分以上かかった。また、多くのキーポイントが正解の指に繋げることができなかった。

4.4.2 勾配フィルタを用いた AKAZE 特徴マッチングによる指紋認証の考察

提案手法の正解率を比較したところ、ほとんどの指が正解率は平均的に 0~10%であった。グレースケールの画像とは違い勾配フィルタでは領域の境界において、画素の輝度値の変

化が大きい。そのため、画素値の変化に対して微分演算を行えば、エッジの検出を行うことができる。デジタル化された画像に対する計算機による処理では、微分演算の代わりに差分演算を行うことにより可能となる。しかし、指紋画像に差分演算を行うことにより、画面に含まれる雑音成分にも反応してしまうため指紋の輪郭の部分にノイズが発生してしまう [17]。そのため、ノイズの部分に AKAZE の特徴量のキーポイント検出が反応してしまい正確なマッチングが行えなかったと考えられる。

第5章 結論と今後の課題

5.1 結論

本研究では、デジタルカメラを用いた指紋認証を目的としてグレースケールフィルタや勾配フィルタを用いた輪郭形成を行い、キーポイントを検出し、AKAZEを用いた特徴量マッチングを行う手法を提案すると共に、実験により手法の評価を行った。提案手法の正解率と従来手法での認証精度を比較すると、精度認証としては従来手法では95.6%、提案手法ではグレースケールフィルタをかけた場合の認証では正解率は96%~100%となり精度が向上したと考えられる。ただし、指の回転などが発生した場合、正解率は54~90%と低下した。

一方で、勾配フィルタをかけてAKAZE特徴マッチングを用いた場合ではフィルタをかけることによって発生する画像の雑音や輪郭のひずみの発生によって正しく指紋認証が行えなかった。

今後、提案手法の指紋などに従来手法の指紋ごとに発生する模様の学習などの改良を加えることでより精度の高い手法にすることができると考える。

5.2 今後の課題と提案

今後の課題としては、以下の3点が挙げられる。

- ・各フィルタに適した認証手法の速度改良

今回の実験ではどちらも認証時間が実験1では10秒、実験2では10分以上かかった。実際に実用化がされている指紋認証の平均照合時間は0.3秒と非常に高速であるので、アルゴリズムの改良や認証手法の検討といったことが重要となる。

- ・指紋画像が十分な場合の評価実験

本研究では、評価実験の対象として指紋を10本使用したが、それによる精度はまばらであり不十分であることが分かった。

さらにフィルタごとによる指紋検出の差を評価することができなかった。そのため、各フィルタによる評価実験と評価に適した、より画像数の多いデータセットを用いる等、指紋画像数を増やして評価実験を再度行う必要がある。

- ・パターン学習による精度向上

本研究では従来の手法であるパターン学習を用いなかった。すなわち、指紋と電極版の磁気の差分から白黒画像を検出し模様によるいくつかの特徴点を利用する方法、或いは線が分かれる分岐点や三角州のような形の場所の問題、さらに袋小路のような端点などの特徴点を探し出し、その特徴点が指紋の中心点からどの位置にあるのかをデータ化する方法ではなかった。

そのためこのような指紋認証特有の特徴点などを学習し、今回の提案手法と組み合わせることによるパターン学習が必要であると考えられる。

謝辞

本研究を行うにあたって必要な環境を与えてくださり、また本論文の執筆や研究について指導してくださった渡辺裕教授と早稲田大学国際情報通信センター招聘研究員の石川孝明氏に心から感謝致します。

日頃からアドバイスをくださった研究室の皆様にも感謝致します。

参考文献

- [1]管ほか, “ここまで来たバイオメトリクスによる本人認証システム”, IPAJ Magazine, Vol.40, No.11, pp.1-5, Nov. 1999
- [2]酒井ほか, “バイオメトリック認証技術について”, 電気設備学会誌, Vol.30, No.10, Oct. 2014
- [3]B.Y.Hiew, Andrew B.J.Teoh and David C.L. Ngo, “Automatic Digital Camera Based Fingerprint Image Preprocessing”, IEEE Computer Graphics, Imaging and Visualization, No.6, pp.182-189, Jul. 2006
- [4]内田薫, “指紋にとる個人認証の最前線”, 映像情報メディア学会誌 Vol.55, No.2, pp.176-179, Feb. 2001
- [5]鳥居恵多, “非接触型指紋認証を目的とした構造線抽出法”, 映像情報メディア学会誌 Vol.59, No.7, pp.1001-1010, Jul. 2005
- [6] J-Cast ニュース, (<http://www.j-cast.com/2016/11/21283865.html?p=all>, 2017年1月4日アクセス)
- [7]大金偉夫, “ユーザーの利便性を考慮した指紋の盗撮防止手法”, Computer Security Symposium 2016, No.11-13, pp.255-362, Nov. 2016
- [8]手島ほか, “モルフォロジー演算を用いた3次元形状の類似度検索の一手法”, 映像情報メディア学会誌 Vol.57, No.10, pp.1284-1291, Oct.2003
- [9]藤吉弘亘, “Gradient ベースの特徴量抽出-SIFT と HOG-”, 情報処理学会研究報告, IPSJ SIG Technical Report, CVIM-160, No.9, pp.211-219, Sep.2007
- [10]H.J.Chien, C.C.Chuang, C.Y.Chen, R.Klette “When to Use What Feature? SIFT, SURF, ORB, or A-KAZE Features for Monocular Visual Odometry,” IEEE Image and Vision Computing New Zealand (IVCNZ), No.11, pp.2151-2205, Nov. 2016
- [11]<http://www.robosafe.com/personal/pablo.alcantarilla/publications.html>, 2017年1月7日アクセス

- [12]田邊ほか, “SIFT 特徴量とガウス混合モデルを用いた一般物体認識法の性能評価“, 2011 年電気関係学会九州支部連合大会, 08-2P-15, No.9, pp.573, Sep. 2011
- [13]Y.Liu, C.Lan, F.Yao, L.Li, “Oblique Remote Sensing Image Matching Based on Improved AKAZE Algorithm,” IEEE Information Science and Technology (ICIST), No.5, pp.448-454, May. 2016
- [14]S.Urban, M.Weinmann, “Finding A Good Feature Detector-Descriptor Combination for The 2Dkeypoint-Based Registration of TLS Point Clouds,” ISPRS Annals of the Photogrammetry, Vol.II-3/W5, No5, pp121-128, May. 2015
- [15]Pietro Perona, Jitendra Malik, “Scale-Space and Edge Detection Using Anisotropic Diffusion,” IEEE Trans PAMI, Vol.12, NO.7, pp629-639, Jul. 1999
- [16]寺主一成, “グレースケールの色濃度値による解析“, SEN-I GAKKAISHI(報文), Vol. 36, No.6, pp.104-109, Jun. 1980
- [17]宮岡伸一郎, “画像の勾配空間フィルタリング“, 映像情報メディア学会誌, Vol.57, No.10, pp.1284-1291, Oct. 2003
- [18]A.E.Abdel-Hakim, A.A.Farag, “CSIFT: A SIFT Descriptor with Color Invariant Characteristics,” IEEE Computer Society Conference, Vol.2, No.3, pp1978-1983, Apr. 2006.

図一覧

- 図 2.1 従来実験の実行環境
- 図 2.2 SIFT 特徴マッチングのフローチャート
- 図 2.3 スケール変化によるガウス窓
- 図 2.4 SIFT マッチングの変遷
- 図 3.1 デジタルカメラを用いた指紋認証のフローチャート
- 図 3.2 原画の指紋画像の例
- 図 3.3 グレースケールフィルタをかけた指紋画像の例
- 図 3.4 勾配フィルタをかけた指紋画像の例
- 図 4.1 原画の指紋画像(左人差し指)
- 図 4.2 原画の指紋画像(右人差し指)
- 図 4.3 原画の指紋画像(左小指)
- 図 4.4 原画の指紋画像(右小指)
- 図 4.5 原画の指紋画像(左薬指)
- 図 4.6 原画の指紋画像(右薬指)
- 図 4.7 原画の指紋画像(左中指)
- 図 4.8 原画の指紋画像(右中指)
- 図 4.9 原画の指紋画像(左親指)
- 図 4.10 原画の指紋画像(右親指)
- 図 4.11 グレースケールの指紋画像(左人差し指)
- 図 4.12 グレースケールの指紋画像(右人差し指)
- 図 4.13 グレースケールの指紋画像(左小指)
- 図 4.14 グレースケールの指紋画像(右小指)
- 図 4.15 グレースケールの指紋画像(左薬指)
- 図 4.16 グレースケールの指紋画像(右薬指)
- 図 4.17 グレースケールの指紋画像(左中指)
- 図 4.18 グレースケールの指紋画像(右中指)
- 図 4.19 グレースケールの指紋画像(左親指)
- 図 4.20 グレースケールの指紋画像(右親指)
- 図 4.21 勾配フィルタの指紋画像(左人差し指)
- 図 4.22 勾配フィルタの指紋画像(右人差し指)
- 図 4.23 勾配フィルタの指紋画像(左小指)
- 図 4.24 勾配フィルタの指紋画像(右小指)
- 図 4.25 勾配フィルタの指紋画像(左薬指)
- 図 4.26 勾配フィルタの指紋画像(右薬指)
- 図 4.27 勾配フィルタの指紋画像(左中指)

- 図 4.28 勾配フィルタの指紋画像(右中指)
- 図 4.29 勾配フィルタの指紋画像(左親指)
- 図 4.30 勾配フィルタの指紋画像(右親指)
- 図 4.36 認証用の画像セット (グレースケール画像)
- 図 4.36 認証用の画像セット (勾配フィルタ)
- 図 4.37 グレースケールフィルタを用いた認証例
- 図 4.38 勾配フィルタを用いた認証例

表一覧

表 4.1 実験環境

表 4.2 実験機材

表 4.3 実験環境

表 4.4 グレースケールフィルタを用いた指紋認証によるマッチング結果

表 4.5 勾配フィルタを用いた指紋認証によるマッチング結果