

H-031

携帯情報端末のための回転拡散ネットによる虹彩認証 Iris Recognition by a Rotation Spreading Neural Network for Personal Digital Assistant

國宗 高志†
Takashi Kunimune

高野 博史†
Hironobu Takano

中村 清実†
Kiyomi Nakamura

1. はじめに

近年、携帯情報端末における高度なセキュリティの必要性が高まっている。高度なセキュリティの一つとして生体認証があり、生体認証の中でも虹彩認証は認証精度が高いことが知られている。従来の虹彩認証アルゴリズム[1]では、虹彩模様の変位量を認識できないため、あらゆる回転方位でマッチングを行う必要がある。しかし、回転拡散ネットを用いた虹彩認証法[2]は、虹彩模様の回転方位を認識できるため回転変位量を高速に補正できる。したがって、カメラを搭載した携帯情報端末を手で持って個人認証を行う状況では、認証時に撮像して得られた虹彩画像には登録虹彩画像に対して回転変位が生じるため、本認証法が有効であると考えられる。そこで、本研究では、回転拡散ネットを用いた虹彩認証システムを携帯情報端末に実装するために、アルゴリズムの最適化を行うことで高速化を図った。さらに、最適化により生じる形状認識精度への影響について調査した。

2. 回転拡散ネットによる虹彩認証

回転拡散ネットによる虹彩認証法は、取得した画像の回転変位量をベクトル合成法により認識して回転補正し、ユークリッド距離により認証を行う。ベクトル合成法を用いた回転拡散ネットによる虹彩認証システムの概要を図1に示す。

本虹彩認証法は、回転方位の補正を行い、形状を認識する。回転方位の認識および形状の認識は、拡散パターンを用いて行われる。まず、虹彩画像に極座標変換を施す。そして、極座標変換画像にガウス関数で表わされる拡散ウェイトを掛け合わせることで複数方位の拡散パターンを生成し、それを登録拡散パターン V_{Li} ($i=1,2,\dots,6$) とする。認証時は、各方位の登録拡散パターン V_{Li} と認証拡散パターン V_R を掛け合わせて合成することで回転方位を認識する。次に、認識方位 ϕ だけ極座標変換の開始位置を移動することで認証虹彩画像の回転方位を補正する。回転方位補正後に再度生成される認証拡散パターン V_R と登録拡散パターン V_{Li} のユークリッド距離を計算し、閾値処理により形状認識をして個人の判別を行う。

3. 実行時間短縮のためのアルゴリズム最適化

携帯情報端末における本虹彩認証アルゴリズムの実行時間を調査するため、携帯情報端末開発プラットフォームである Sophia Systems 社の Sandgate III-P へ本認証法を実装した。そして、あらかじめ虹彩画像取得用のカメラで撮影された虹彩画像を用いて、Sandgate III-P 上で

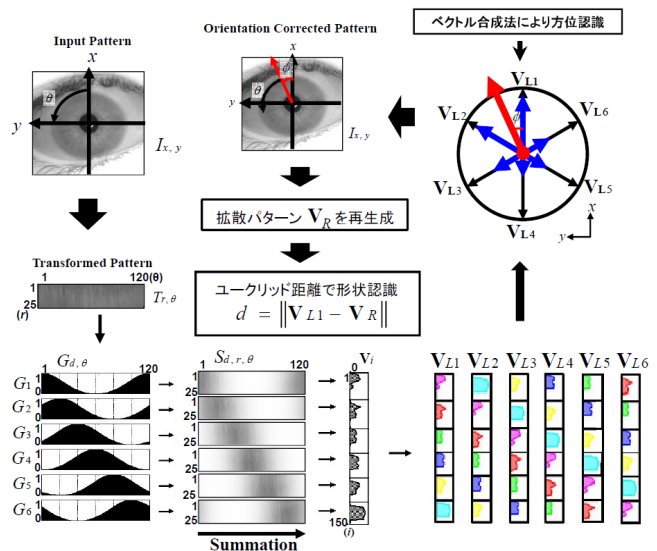


図1 回転拡散ネットを用いた虹彩認証法

登録および認証に要する処理時間を計測した。

Sandgate III-P の CPU 動作周波数は 624MHz であり、近年の PC と比べて処理能力が劣るため、実行時間が長くなる。Sandgate III-P における本虹彩認証アルゴリズムの実行時間は、画像の読み込みに 0.045 秒、極座標変換に 1.099 秒、拡散パターン生成に 0.055 秒を要し、特に極座標変換に時間を要することが分かった。そこで、極座標変換の際に用いる正弦関数のテーブル化を行い、極座標変換の際の線形補間を 1 点のみで行うことで極座標変換を高速化した。さらに、極座標変換画像を回転方位 (θ) 方向にシフトすることで方位補正を行い、アルゴリズムの最適化を図った。図2に、従来の本虹彩認証法のフローチャートとアルゴリズム最適化を行った本虹彩認証法のフローチャートを示す。また、以下に実行時間を短縮するためのアルゴリズム最適化法について詳しく述べる。

3.1 正弦関数のテーブル化

図2-①に示すように、極座標変換は正弦関数と余弦関数を用いて計算をしているため、多くの処理時間を要している。そこで、正弦関数をあらかじめテーブル化しておくことで実行時間の短縮を図った。本虹彩認証アルゴリズムで必要となる角度数は、1 度ごとに 0 度から 359 度の 360 個である。したがって、360 個の正弦関数値のルックアップテーブルを作製し、計算時にはその値を用いる。余弦関数については正弦関数のテーブル値を用いて対応する。この処理により、極座標変換に要する時間が 1.099 秒から 0.322 秒と大幅に短縮された。

† 富山県立大学大学院 工学研究科
Graduate School of Engineering, Toyama Prefectural University

3.2 方位補正のための極座標変換の省略

本虹彩認証法では認証の際、ベクトル合成法を用いて回転方位を認識した後、図 2-②に示すように、極座標変換の開始位置を認識方位だけ移動させることで回転方位ずれの補正を行っている。そのため、認証時は極座標変換 2 回分の時間を要する。そこで、2 回目の方位補正のための極座標変換を行わずに、1 回目の極座標変換画像を認識方位分 θ 方向にシフトすることで方位補正とし、実行時間を短縮した(図 2-②')。この回転補正を施した極座標変換画像を用いて拡散パターンを生成し、登録拡散パターンとのユークリッド距離を求めて認証する。

3.3 線形補間を行う点数の変更

本虹彩認証法では極座標変換の際、図 2-③のように、極座標の任意の 1 点を求める際に、近傍 9 点の画素値を線形補間により求め、足し合わせている。これは、極座標変換によって生じる画素値の誤差を軽減するためである。しかし、極座標画像の 1 画素を求めるために 9 点の演算を行うため、処理に多くの時間を費やす。そこで、図 2-③' に示すように、線形補間を 1 点のみで行うことで、処理時間の短縮を行った。

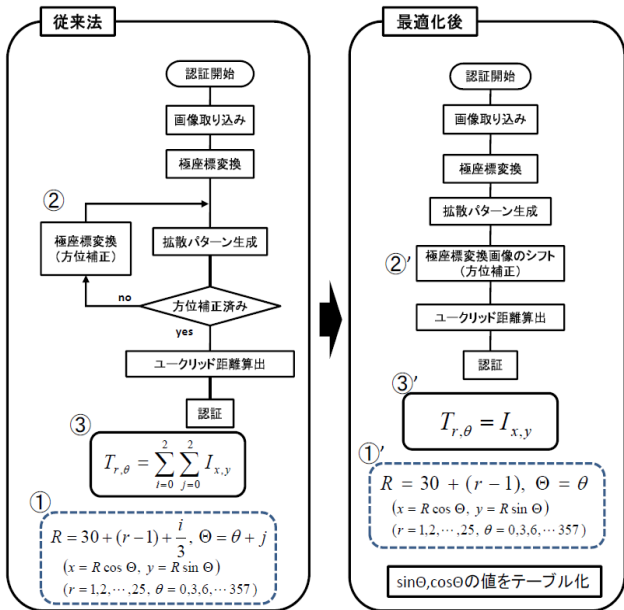


図 2 アルゴリズムの最適化

4. アルゴリズム最適化による影響

前節で述べた実行時間を短縮するためのアルゴリズム最適化法を組み合わせ、高速化したことによる Sandgate III-P 上での実行時間と認証精度への影響を調査した。

4.1 実行時間の変化

表 1 に、従来の回転拡散ネットによる虹彩認証法と高速化した本虹彩認証法の Sandgate III-P における実行時間を示す。表 1 より、従来の方法では登録に 1.353 秒、認証に 2.6 秒と長い時間を要したが、高速化により登録時間が 0.23 秒、認証時間が 0.367 秒と大幅に短縮され、

携帯情報端末を用いても利用者に負担がかけられない時間で認証することが可能となった。

表 1 各処理の実行時間の比較

	従来方法[sec]	高速化[sec]
極座標変換	1.099	0.035
拡散パターン生成	0.055	0.055
登録	1.353	0.230
認証	2.600	0.367

4.2 認証精度への影響

アルゴリズムを最適化したことによって生じる認証精度への影響を実験により調査した。実験は、22 人分の登録画像および認証画像の 2 種類の虹彩画像を用いて、従来の方法と高速化した方法について認識精度を求めた。他人受入率と本人拒否率が同率となるポイントにおける誤り率を EER (等誤り率) といい、認証精度の評価基準とした。従来の回転拡散ネットを用いた虹彩認証法での EER は、ユークリッド距離が 0.32 の時 0.73% となった。これに対して、アルゴリズムの最適化により高速化を行った方法での EER は、ユークリッド距離が 0.32 の時 0.64% となった。従来方法と高速化方法の EER の差は僅かであり、アルゴリズムの最適化による認証精度への影響はないと考えられる。

5. おわりに

本研究では、ベクトル合成法を用いた回転拡散ネットによる虹彩認証アルゴリズムを Sandgate III-P に実装し、高速化を行った。高速化により、利用者に負担がかけられない程度の時間で登録および認証が可能となった。また、高速化による認証精度の低下は見られず、本研究における高速化アルゴリズムが有用であることを示した。

従来の虹彩認証システムは、角膜反射光によるノイズをなくすために近赤外光領域を用いて虹彩模様の撮影を行っており、可視光領域で得られた画像では認識が困難である。しかし、携帯情報端末に搭載されるカメラは可視光で撮影することを目的としている。そこで、今後の課題として、可視光領域で撮影された虹彩模様を用いて認識を行う方法を検討する必要がある。

参考文献

[1] John Daugman, "How Iris Recognition Works", IEEE Transactions on Circuits and Systems for Video Technology, Vol.14, No.1, pp.21-30, 2004.
 [2] H. Takano, H. Kobayashi and K. Nakamura, "Rotation invariant iris recognition method adaptive to ambient lighting variation", IEICE Trans. Inf. & Syst., Vol.E90-D, No.6, pp.955-962, 2007.