

OpenCV を利用した指紋照合

山中裕美 (指導教員: 金子 晃)

1 はじめに

指先の表皮模様である指紋は「万人不同」、「終生不変」という特徴を持つ。紋認証技術は犯罪捜査のみならず、ネットワーク社会における本人の確認方式として有用な為、様々な研究開発が行なわれている。

指紋認証には画像処理技術が必要不可欠である。OpenCV は、Intel によって開発され、Willow Garage 社が開発を引き継いでいる画像処理向けのオープンソースライブラリである。これを用いて画像処理を手軽に行なえる。

本研究では OpenCV を用いて指紋認証における画像処理の実装を試みる。プログラミング言語は C++ を使用する。

サンプル画像として NIST の Special Database が提供している指紋画像を利用した。

2 指紋認証のための画像処理の方針

- 採取した指紋画像が粗いため、様々な画像処理の試行により前処理を行う。
- 二値化、細線化を行う。
- マニューシャを抽出する。
- 照合を行う。

3 前処理

指紋画像には小さなノイズが含まれている。このまま細線化すると綺麗な流線にならないため、前処理を行いきる限りノイズを少なくする。

平滑化、雑音除去のために OpenCV の cvSmooth() の次の3種のフィルタを使用した。いずれもカーネル幅が3の時を説明する。

<移動平均フィルタ>

移動平均フィルタにはすべての濃淡変動を一様に平滑化し、雑音低減やぼかし効果がある。右のように注目画素の 3 × 3 近傍の平均値を輝度とする。

$\frac{1}{9}$	$\frac{1}{9}$	$\frac{1}{9}$
$\frac{1}{9}$	$\frac{1}{9}$	$\frac{1}{9}$
$\frac{1}{9}$	$\frac{1}{9}$	$\frac{1}{9}$

<メディアンフィルタ>

メディアンフィルタには微小な濃淡変動のみを選択的に平滑化しエッジや線の保存をする効果がある。注目画素の 3 × 3 近傍を小さい順にならべる。

[10,15,18,19,20,20,23,24]

その中間値を輝度とする。

18	20	24
10	15	19
16	20	23

	19	

<ガウシアンフィルタ>

ガウシアンフィルタには高周波成分を除去する効果がある。

注目画素に近いほど、平均値を計算するときの重み

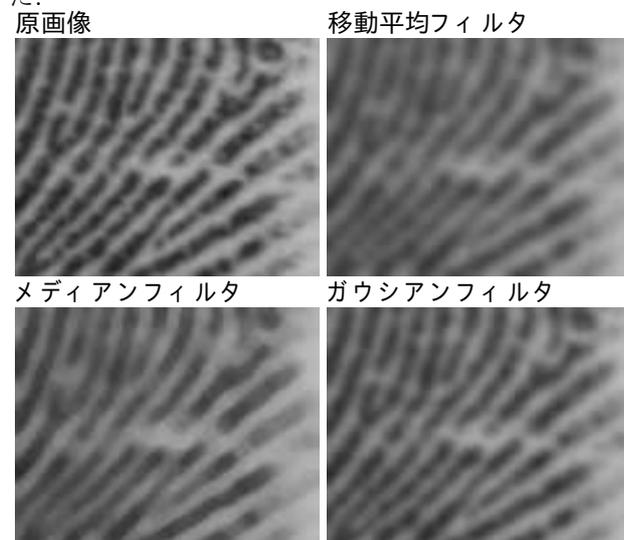
を大きくし、遠くなるほど重みを小さくなるように、次のガウス分布の関数を用いて重みのレートを決定する。

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

$\frac{1}{16}$	$\frac{2}{16}$	$\frac{1}{16}$
$\frac{2}{16}$	$\frac{4}{16}$	$\frac{2}{16}$
$\frac{1}{16}$	$\frac{2}{16}$	$\frac{1}{16}$

3.1 結果

指紋画像は小さいため、拡大して使用した。それぞれのフィルタをかけたときの結果画像は次のようになった。



4 二値化・細線化

4.1 二値化

指紋画像は指の圧力の差により画像の濃度に偏りがある。閾値を自分で与える一般的な二値化 cvThreshold() では、黒っぽい部分がすべて黒くなってしまい、白っぽい部分は見えなくなってしまう。これを解消するために、適応的二値化を行なった。適応的二値化 cvAdaptiveThreshold() は周辺画素の輝度値の平均を閾値として二値化を行う。閾値はその画素ごとに自動的に決定される。

また、フィルタにかけたものをそれぞれを二値化してみると前処理なしではギザギザしていたり、流線の中にノイズが残った。



4.2 細線化

細線化は二値化画像から線幅1の中心線を抽出する処理である。線幅を1にすることで後の処理を簡単にする。ここでは細線化画像を見やすくするために白黒反転させている。

前の三種のフィルタ処理の後それぞれ二値化を行なうと、メディアンは線の途切れが少くなり、ガウシアンは細かいノイズが最も消去されていた。今回使用したNISTの指紋画像の前処理にはメディアンフィルタとガウシアンフィルタを採用することにした。

5 マニューシャ抽出

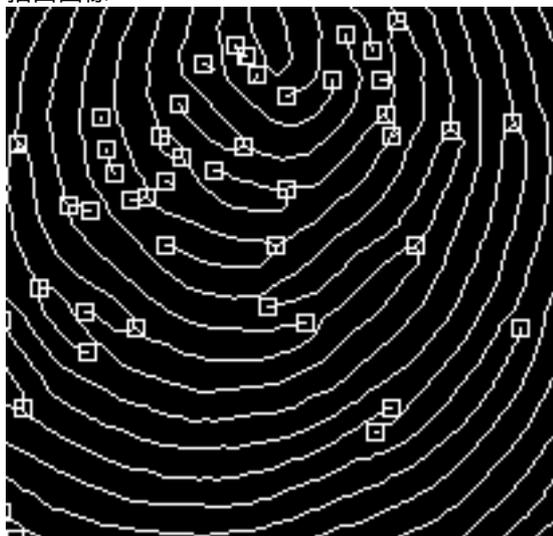
マニューシャとは指紋の中にある特徴点のことで端点と分岐点がある。一つの指紋の中にあるマニューシャの位置関係などから照合を行なう。

次の方法で方法で指紋の細線化画像からマニューシャを抽出した。

8連結の細線化画像における、端点、分岐点のパターンをテンプレートとして、テンプレートマッチングを行い、細線化画像の特徴点にマーキングする。

cvMatchTemplate()によるテンプレートマッチングはテンプレートを入力画像上でスライドさせ、類似度の高い場所を探することができる。

抽出画像



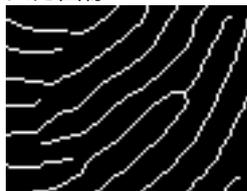
6 照合

6.1 対象領域検出

マニューシャマッチングの前に大まかな対象画像、対象領域の目星をつけなければいけない。指紋の部分画像をテンプレートマッチングで登録画像から最も類似した領域を見つける。cvTemplateMatching()はテンプレートを平行に走査する。傾いた画像の場合も検出できるように、入力画像を360度回転させて最も類似したものを返すようにした。

検出画像

入力画像



6.2 マニューシャマッチング

6.2.1 類似度測定の方法

登録画像の対象領域のマニューシャの個数をM1とする。入力画像と登録画像のマニューシャの重ね合わせを行い、重なったマニューシャの個数をM2とする。類似度をM2/M1とした。

<重ね合わせについて>

2つの比較指紋画像の特徴点のみを取り出し、一方の画像は端点をRGB(255,0,0)、分岐点をRGB(0,255,0)に配色。もう一方は逆の配色をし、二つの画像をピクセル毎に足し合わせた。2色が交わりRGB(255,255,0)になった箇所の個数が2つの画像間で重なったマニューシャの個数となる。尚、特徴点を配色した円は直径を6ピクセル分にしており、比較画像のマニューシャの小さなずれを考慮している。

<特徴点の数え方について>

特徴点の画像を細線化し、一つの円が1ピクセルの点になるよう画像変換したのち、すべての点を合わせた面積を求め、その面積を特徴点の個数とした。

6.2.2 閾値の決定

NISTには同一人物の同じ指の指紋が各2つずつ保存されている。20人分の指紋でマニューシャマッチングを行い、それぞれの類似度の結果のうち、最も値が小さいものを閾値とする。

<結果>

類似度は最大のもので、0.992424という結果が得られたが、最低のものは0.223684となった。今回の方法ではマニューシャ以外の特徴点が多いため、よい閾値を得られなかった。

6.2.3 指紋の同定

類似度がある閾値以上であれば同一人物の指紋とする。実装は今後の課題とする。

7 今後の課題

- 今回の問題点として、マニューシャではない端点、分岐点も特徴点と見なしているため、2枚の画像で無意味な点が多く、指紋の同定の閾値が低くなった。不必要な特徴点を除くことを考えなければならない。
- 今回行なった前処理の方法では、くっきりとした指紋画像でなければ綺麗な細線化画像を得られなかった。前処理を改良してより滑らかな指紋画像を得るようにしたい。

参考文献

- [1] Gary Bradski, Adrian Kaebler, (訳) 松田晃一 (2009), 『詳解 OpenCV - コンピュータビジョンライブラリを使った画像処理・認識』, オライリー・ジャパン.
- [2] 奈良先端科学技術大学院大学, OpenCV プログラミングブック制作チーム (2007), 『OpenCV プログラミングブック 第2版』, 毎日コミュニケーションズ.
- [3] 瀬戸洋一 (2002), 『サイバーセキュリティにおける生体認証技術』, 共立出版.
- [4] 「画像処理ソリューション」, <http://imaging-solution.blog107.fc2.com/blog-entry-165.html>