

凸点抽出を用いた点字画像の認識

高野 源太[†] 伊藤 祥一[‡]

長野工業高等専門学校 電子情報工学科^{†‡}

1 はじめに

点字とは、指先の触覚を利用して読み取ることができる文字であり、視覚障害者が情報を得るための手段の一つである。バリアフリーが進んだ現在、点字は様々な場所に使われているが、点字が読めない晴眼者が点字を設置する作業を行い、「上下反対で設置してしまう」、「単語の途中で改行してしまう」等の原因で間違った点字が設置されてしまう事例がある¹⁾。このような間違いを防ぐために、晴眼者向けの iOS 用点字翻訳アプリケーションの開発が行われている²⁾。このアプリケーションでは、スマートフォンで撮影された点字画像を二値化し、そこから点字を読み取って翻訳を行う。この二値化処理には、画像に光源による白飛びがあった場合や文字が重なっていた場合などで点字の凸点をうまく認識できないという課題があった。本研究では、先行研究の点字の認識手法を改良し、点字画像の状態にかかわらず、なるべく高い精度で点字画像から点字の凸点を認識できるようにすることを目的とする。

2 点字認識の手法

点字画像を取得し、点字をデータ化して翻訳に至るまでの流れを図 1 に示す。本研究では、図 1 の項目のうち「画像処理」と「二値化・データ化」の部分の研究している。「画像処理」は光による白飛びや文字が重なっている点字画像からそれらを削除し誤認識しにくい画像を作成すること、「二値化・データ化」は処理した画像から点字の凸点のみを抽出することを指しており、最終的に既に先行研究で開発されている「翻訳処理」に対応できるようなデータを出力するところまでを行っている。なお、先行研究と同様に AndroidStudio 上で Java および OpenCV を用いて実装を行った。

3 研究内容

3.1 凸点認識を行いやすくするための画像の前処理

点字画像をスマートフォンなどで撮影した時、光源の位置や設置場所によっては白飛び(光の反射)や文字の重なりによって点字がうまく読み取れない場合がある。そのような場合でも点字の凸点を抽出できるように、白飛びや文字を埋めるという作業を行う。手法としては、図 2 に示すような、カメラで撮影した白飛びがある点字画像を、OpenCV の単純なしきい値処理を行う threshold 関数を用いて白飛び・文字部分を黒色で埋めた画像(図 3)と白飛びしている部分・文字が重なっている部分を抜き出し、色を調整した画像(図 4)をクロマキー処理で合成することで白飛びをなくすという処理を行う。この処理の結果を図 5 に示す。図 5 より白飛びの部分が埋まり、反射が抑えら

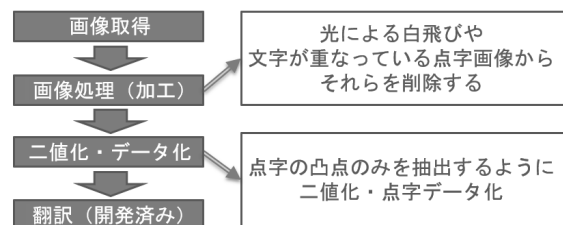


図 1 点字データ化までの処理の流れ



図 2 白飛びしている点字画像

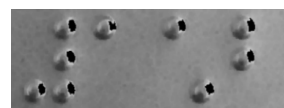


図 3 図 2 の白飛び部分を塗りつぶした画像

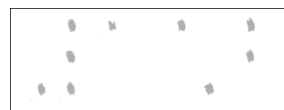


図 4 図 2 の白飛び部分のみを抽出し補正した画像

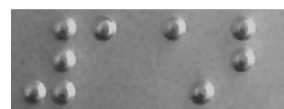


図 5 白飛びの処理結果

れていることわかる。

3.2 点字部分の二値化

図 6 に示す文字のある画像に対して同様の処理を行った場合の結果を図 7 に示す。

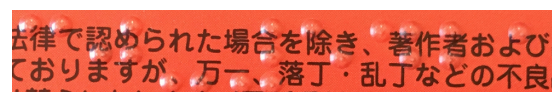


図 6 文字が重なっている点字画像



図 7 文字の処理結果

図 7 では文字の部分がある程度消去されていることが分かる

Braille Image Recognition using Braille Dot Extraction

[†] Genta Takano: National Institute of Technology, Nagano College

[‡] Shoichi Ito: National Institute of Technology, Nagano College

が、点字部分が若干潰れてしまい視認でもわかりにくくなっている。このような、点字部分がぼけてしまっている画像から点字を認識し、凸点部分を二値化画像として抽出する方法として、Radial Snakes³⁾ という方法を参考にした凸点の輪郭を抽出する方法（凸点抽出法）による凸点の抽出を試行している。具体的な工程を以下に示す。

1. 初期中心点を決め八方向に線を引く。
2. 線上において変化の大きいところを特徴点として点を打つ。
3. 特徴点のうち円形に近くなるものを候補点とする。
4. 中心点を再度設定し 2. に戻る。（中心点が同じ箇所へ停止するまで繰り返す。）

図 8 の画像に対して凸点抽出法を適応した時のイメージを図 9 に示す。図 9 の各画像の左上についている番号は輪郭抽出法の各工程の番号の処理に対応しており、各画像はその処理を行ったときの様子である。



図 8 凸点抽出を行う元画像

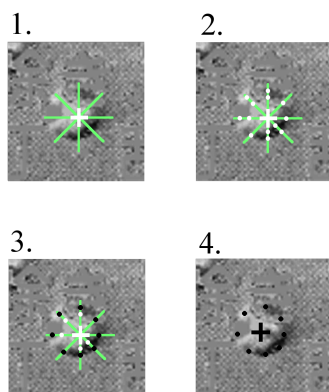


図 9 凸点抽出法のイメージ

また、抽出精度を高めるため凸点抽出法に「探索距離は画像の縦幅の 15% 以内とする」、「特徴点が 8 方向中 3 方向以上で見つからなかった場合探索を中断しその中心点を破棄する」という制約をつけている。

4 処理結果

図 10 の点字画像に対して前処理を行い、凸点抽出によって点字を認識した結果を図 11 に、同様に図 12 の画像について点字認識を行った結果を図 13 に示す。なお凸点抽出の工程は、処理の軽量化のため中心点の位置が停止するまでではなく 5 回繰り返すことで処理を完了している。



図 10 “おくじょー”とかかれた点字画像

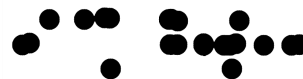


図 11 図 10 の画像について点字認識を行った結果



図 12 “Battleships”とかかれた点字画像



図 13 図 12 の画像について点字認識を行った結果

5 評価と今後の展望

図 11 より図 10 の点字画像から凸点の位置を抽出できていることがわかる。しかし、図 10 と図 11 を比べると若干位置がずれていること、二つの点がくっついてほとんど同じ箇所に描画されている様子が確認できる。また、図 13 の画像は光源による極端な反射はないが、画像の左側に光源があるため左側のコントラストが低くなっており凸点の検知ができなくなっていることが確認できる。

本研究の凸点抽出法を用いた点字の検出について、現状の状態では、先行研究の課題であった白飛び、文字の消去はある程度行うことができたが、画像内で光源の影響による微妙な明るさの違いがあると点字が認識されないという場合があった。また、各々の点について個別に凸点抽出の判定を行うため同じ高さにある点でも認識結果としての出力では若干位置がずれていることや同じ点に複数の点があると判定されてしまうことがあり先行研究の翻訳機能に接続することが難しくなっている。以上のことから本研究の改善すべき点として、凸点抽出処理を行う前に画像のコントラストを一定のものに統一すること、凸点抽出について同じ高さにある点の位置をもとにずれを補正することなどの改善策が挙げられる。この手法で正しく点字を認識できるようになれば、光源の影響などで不要な箇所が重なっていても点字の検出が行えるようになる。

参考文献

[1] 点字案内 目立つミス：読売新聞 1999 年 8 月 6 日（金）発行
 [2] Shoichi Ito and Yoshinori Fujisawa: Tender – Smartphone Application for Braille Reading, 7th ACIS International Conference on Applied Computing & Information Technology (ACIT2019) proceedings, pp.69-74, 2019.
 [3] 不明瞭な凸形状輪郭の抽出法：高橋正信・中野雅行, 情報科学技術フォーラム一般講演論文集, 2002(3), pp.113-114. 2002.