

顔認識技術を用いた点字の認識

Reading Braille using Face Detection

大和 優介† 藤澤義範† 伊藤 祥一†

Yusuke Ohwa Yoshinori Fujisawa Shoichi Ito

1. はじめに

現在、世界的に使われている点字は 1825 年にフランス人のルイ・ブライユによって発明されたものである。主流となっている 6 点式点字は、6 つの点の凸によって 1 文字を表している。日本においては 1890 年にブライユによる点字を基に石川倉次が日本式点字を発明している。

点字は、視覚に障害を持つ人が情報を得るための文字で、近年公共の場での普及が進んでいる。しかし、すべての視覚障害者が点字を読めるわけではない。厚生労働省の調査[1]によると視覚障害者の点字の識字率は、12.7%とかなり低いことがわかっている。原因としては、後天的に視覚に障害を持った人が点字を習得することが難しいことが挙げられる。

つまり、点字を習得せずともテレビやラジオなどを使った音声による情報の取得が可能となっており点字の習得は必須ではないとも考えられる。しかし、外出することを考えると常に音声による案内が期待できず、結果として外出しなくなってしまうおそれがある。

総務省の調査[2]によると、表 1 に示すように、「電車・バス・タクシーなどの乗り物の利用が不便」や「道路や駅などの公共の場所の利用が不便」といった回答が多い。しかし、このような公共の場には点字が普及しており、点字を読むことができれば障害の負担が軽減し、積極的に外出できるようになると考える。

表 1: 外出するうえで、または外出しようとするうえで困ること(複数回答) (文献 2 より抜粋)

困ることや不満に思うことの内容	割合(%)
電車・バス・タクシーなどの乗り物の利用が不便	22.1
道路や駅などの公共の場所の利用が不便	13.6
利用する建物の設備(階段、トイレ、エレベーターなど)が不便	15.0
車などに身の危険を感じる	17.9
介助者がいない	1.4
事前に行き先を家族などに告げなければならぬ	2.1

点字は紙やプラスチック、金属などさまざまな素材への刻印が可能であり、公共の場での案内役だけでなく、点字書籍や触地図としても利用されている。

これまで点字を読めない人の代わりに機械的に点字を認識して読上げる研究[3]や撮影した画像から画像処理によって点字を認識して読上げる研究[4]が行われている。

本研究では、撮影した画像の点の認識に従来の方式とは大きく異なる顔認識技術を用いる。顔認識技術の精度はかなり高く、これを点の認識に応用することで点字の認識率を飛躍的に向上させることを目的とする。

2. 点字の認識手法

本研究では、次の 1~4 の手順で点字を認識する。

1. 任意の環境下で点字を撮影する
2. 撮影した画像を 2 値化処理する
3. 2 値化された画像から点を認識する
4. 点を探索して点字として情報を得る

本章では上記のそれぞれの手順について説明する。

2.1. 入力画像の撮影

点字は、指先の腹を使って読取るため、日本人の体型にあったサイズが図 1 のように JIS 規格[5]で定められており 6 点を 1 セルと呼び、清音の 1 文字を表す。

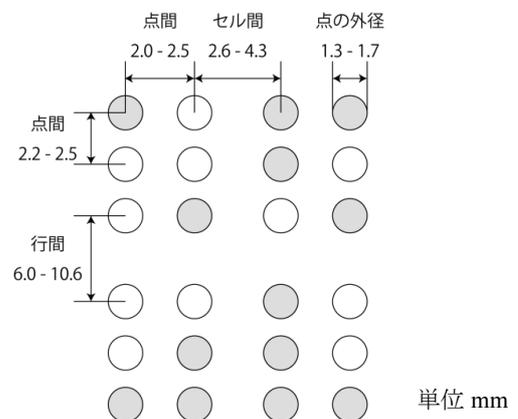


図 1: JIS 規格による点字のサイズ

点字の 1 セルを構成する 6 点のうち凸の箇所をセンサによって検出して点字として認識する。点字が刻印されている素材はさまざまであり、電気や磁気などの特性を利用したセンサを用いることはできない。また、点字が刻印されている素材の色や形状も状況により異なっている。

そこで、筆者らは条件を整えた状態で点字を撮影し、その画像を使って顔認識技術による点字の読取りを行うことと本手法の一連の手順の確立を行うこととした。

本研究で対象とする点字は、白地の点字用の用紙に点字プリンタで打ち出されたものとし、そこに加えて黒色の墨字による印刷があるものとした。点字書籍などでは点字の刻印に加えて墨字による印刷がある場合が多いので、このような状況を想定して条件を整えた。

カメラを使いこの点字を撮影すると、白地に白の凸があるだけなので、点字をはっきりと撮影することができなかった。そこで、一定の方向から光をあてて、点字の凸の部分が作る影を撮影することとした。

撮影環境としては、図 2 に示すように天井の白色蛍光灯の直下に垂直になるように点字が刻印された用紙を立てる。その用紙のからおよそ 30cm 離れた場所から用紙と水平になるようにカメラを置いて撮影する。

† 長野工業高等専門学校 電子情報工学科

実際に撮影された画像の一部を拡大したものを図 3 に示す。図 3 からわかるように点字の凸の部分に対して一定方向に影ができています。このように撮影された画像を使い、点字の認識処理を行う。

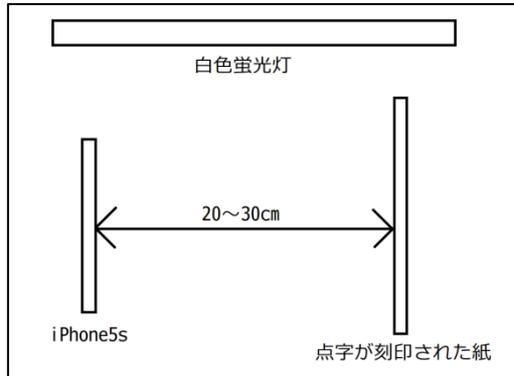


図 2：点字の撮影環境

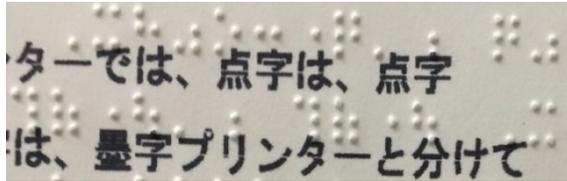


図 3：撮影された点字の一部

2.2. 画像の 2 値化

今回の撮影環境において、画像は幅 2448×高 3264 ピクセルのフルカラー画像となる。

この画像から 2 値化処理により墨字を除去し、点の影のみを強調する。画像の 2 値化処理では、しきい値を決定しそのしきい値を基準に白と黒を決定するのが一般的である。ただし、しきい値の決定の仕方によっては、点の影のみを強調することができない場合もあるため独自の方法を検討している。

著者らは、入力画像のヒストグラムを使って 2 値化処理のしきい値を計算することとした。

まず、撮影された図 3 の画像をグレースケールに変換して図 4 のようなヒストグラムを作成する。

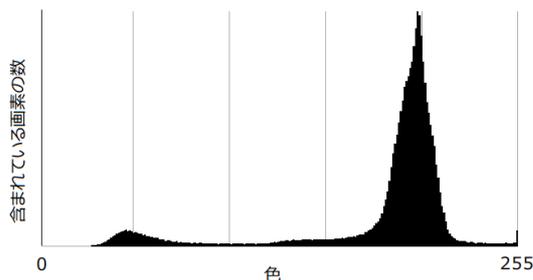


図 4：画像のヒストグラム

図 4 のヒストグラムにおいて 0 から 255 までの色の値で、画素数が一番多い色を調べる。その時の色の値を max とし、下記の式でしきい値を計算する。

$$\text{しきい値} = \frac{(255 + \max)}{2}$$

計算により求めたしきい値を使って、図 3 の画像を 2 値化処理し白黒反転した結果を図 5 に示す。

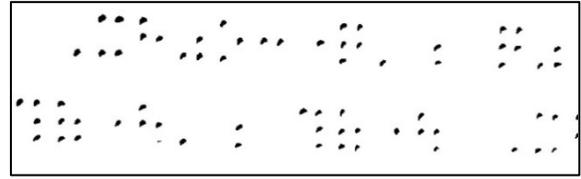


図 5：2 値化された画像

図 3 と図 5 を比べてわかるように墨字は除去されており、点の凸も強調されていることがわかる。

このような撮影環境としきい値の計算方法を使って、438 個の点について処理を行った。用紙にすると A4 で 3 枚程度である。元々の用紙に刻印された点字と 2 値化処理によって得られた結果を 1 点 1 点比較した結果、20 個の点が 2 値化後に正しく認識されていないことがわかった。

認識率としては、およそ 95.6%であった。認識できなかった点の共通点としては、墨字と重なっていることが挙げられる。ヒストグラムからしきい値を計算する際に、墨字の除去を意識した設定を行ったためにこのような結果になったと考えられる。この他にも点字プリンターで刻印した点の凸の高さが低く影が短い場合や、逆に影が長い場合にも正しく認識できていないことがわかった。また、撮影の際の用紙の歪みなどで影の色が薄い場合にも正しく認識できない。

2.3. 点字検出方法

前節までで点字の画像から点の凸の影のみを強調することができた。本節では、強調された影の特徴を顔認識技術によって認識して点がどのように分布しているかを探索する方法について説明する。点の探索によって 1 セルを検出できれば点字として認識ができたこととなる。

点の認識には OpenCV[6]に搭載されている顔認識ライブラリを使用した。OpenCV の顔認識では、検出したい物体の特徴サンプルを多数用意して検出器を作成すれば、顔以外の物体を認識できることがわかっている[7]。

そこで、点字の点の凸の部分を作る影を特徴サンプルとして 637 個用意して本研究の検出器を作成した。サンプルの数と検出率との関係について、今回は計測していないが、これ以上のサンプルを用意しても検出率に変化が見られなかった。検出対象となる点の特徴が異なれば検出器を再度構築する必要があるため、サンプル数と検出率の変化についてはこれからの研究で明確にする必要がある。

図 6 は 2 値化された画像から 1 行のみを抽出したものである。検出器を使って図 6 から点を探検する。



図 6：2 値化された一列の点字

図 7 は検出結果である。作成した検出器を使って顔認識技術によって検出された点を四角形で囲んだ様子である。

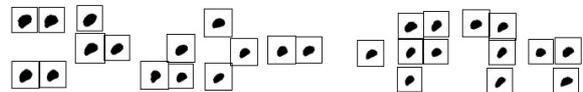


図 7：点字を検出した結果

OpenCV のライブラリにより検出された点を囲う四角形の左上の座標および幅、高さを得ることができる。しかし、それらは個々の点の情報であり、点字としての認識ではない。そこで、検出された点の座標データから 1 セルごとの点字としてのまとまりを認識する必要がある。

図 7 のような点の検出結果に対して縦方向と横方向に点の並びを走査し、結果として図 11 のような格子線を引いた図を生成する。これを左上の点から順に 6 個ずつ読取ることで点字として認識を行う。

縦方向と横方向に線を引く手法は同様であるため、縦方向に線を引く方法についてのみ説明する。

検出された点の四角形について縦方向に走査する。点字 1 行について処理しているため、最大で 3 点が見つかるはずである。2 値化処理された画像内に見つかったすべての点の四角形の幅の平均値を求めて、これを `average` とする。縦方向に見つかった点のそれぞれの x 座標において、次の式を満たす場合は同一の列の点として考える。

$$|x_2 - x_1| < \text{average} \times 0.3$$

ここで、 x_1, x_2 は、2 つの点の x 座標であり、列に 1 点しかない場合は、その点の x 座標に対して線が引かれることになる。点が無い場合は、線は引かれない。

同一の列と判断された点に対して、図 8 のように x 座標の平均を計算してその座標に対して縦方向に線を引くことになる。



図 8：縦方向に 1 本線を引いた様子

`average` の重みについては、仮に同一列に極小の点が発見された場合に、それを含めて x 座標の平均を求めるときに大幅に線の x 座標がずれる可能性がある。そこで、あえて重みをつけて極小の点を取り除くような処理を行っている。

このような処理をすべての縦の列に行い、線を引いた様子が図 9 である。

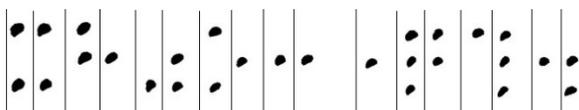


図 9：縦方向に全て線を引いた様子

図 9 から縦方向に対してそれぞれの点がグループに分割されていることが見て取れる。しかし、中央付近の線の間隔が他に比べて大きくなっていることや右端の点の右に線が引かれていないことも見て取れる。

これは、すべての列に点が存在するときには問題は無いが列に点が存在しないときにはこのような状態になる。そこで、列の右側に点が存在しなくとも正しく列を分割する方法を検討する必要がある。

著者らは分割された列の幅の平均値を求めてその値の 1.4 倍を次の線の x 座標とした。線の有無については、次の線の x 座標までの距離が平均の 1.4 倍以上のとき、線が存在しないとした。

理論的には平均値を適応すればよいと考えられるが 2 値化などの処理後の点が規格より若干大きくなることを考慮してこのような値としている。

このような一連の処理を施して縦方向に全ての線を引いたものが図 10 である。ただし、予測して引いた線はあえて強調してある。



図 10：予測線を引いた様子

以上が縦方向に線を引く手順となる。

同様の手順を y 座標に適応することで横方向の線を引くことができる。縦と横のすべてに線を引いた様子を図 11 に示す。

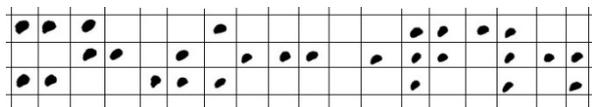


図 11：すべての点を分割した様子

縦線と横線で囲まれた領域を 1 マスとすると、図 11 に対して、左上のマスから下に向かってマス内の点の有無を調べ、6 マスごとに 1 セルとして 1 つのデータにすることができる。ただし、マスに対する点の有無についてはマスの面積のうち 10% 以上を黒が占めているか否かで判断する。点があると判断されれば 1 とし、点がなければ 0 とする。これにより、図 11 からは、(101101), (110010), (001011), (101010), (010010), (000010), (111110), (100111), (010011) という 9 文字分の点字データを取得することができる。

この先の処理としては、機械的に 1 セルを単純に 1 文字に置き換えて発話すればよいことになる。ただし、点字独特の文法などもあるため、ある程度複雑になるはなるが自然言語処理のような複雑さはない。

3. 評価

本章では、2 値化処理による点の検出と顔認識技術を用いた点の認識と点字の検出についての評価について述べる。

3.1. 2 値化処理

図 12 は、2 値化処理後に点の検出結果が 100% であった画像である。図 13 は、検出結果が 90% であった画像である。どちらも同じ画像であるが、光のあたり方による影のでき方が異なっている。また、全体の色も異なっている。同一の環境下で撮影したつもりであるが、用紙の歪みや外光の影響によりでる違いであると考えられる。

図 12 については、左上から右下に向かって徐々に暗くなっているように見える。2 値化処理した結果を見ても左上の点ほど大きく、影が強調されていることがわかる。逆に右下は点が小さくあまり強調されていない。

図 13 は、図 12 に比べて全体的に暗くなっている。2 値化処理のしきい値次第ではこのような画像であっても点の検出は可能な場合もあるが、画像によってしきい値の計算が異なるのは実用的ではない。

今回、筆者らが整えた撮影条件において全てを正確に実現することができれば、しきい値の計算方法も含めて 100%の精度で撮影画像から点を検出することができることがわかった。

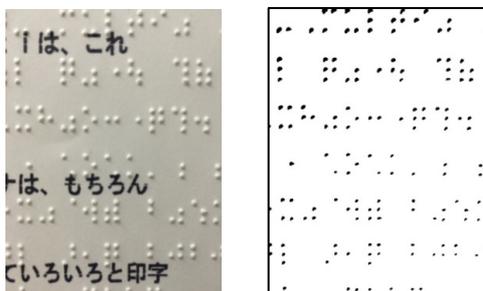


図 12：2 値化後に点を 100%検出できた画像

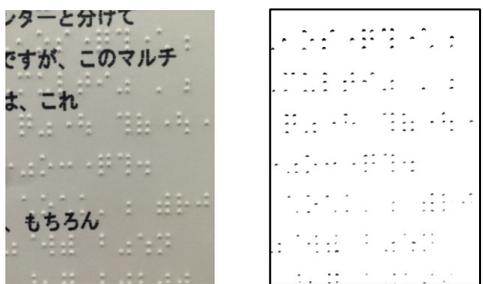


図 13：2 値化後に点を 90%検出できた画像

3.2. 点の認識

図 14 は、被写体とカメラの距離は 30cm 以上離して撮影した画像の 2 値化後の画像である。



図 14：2 値化後の画像サンプル 1

2 値化後の点の検出もできており、ノイズなども少ない。しかし、図 15 のように点の認識において分割処理が正しく行えていないことがわかる。



図 15：画像サンプル 1 の点の認識結果

この原因としては、被写体とカメラとの距離が関係していると考えられる。距離が離れただけ撮影された画像の点小さくなり、点の認識は正しく行えてはいるのだが、分割の際に隣の点との距離が近すぎて正しく分割が行えていないことがわかる。

図 16 は、画像サンプル 1 と同一の画像を著者らの条件下で撮影した画像である。



図 16：2 値化後の画像サンプル 2

図 14 と比べて点が大きく、点の間隔も広く撮影されていることがわかる。

このような撮影環境においては、図 17 のように正しく認識もできていることがわかる。

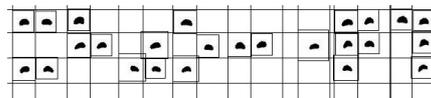


図 17：画像サンプル 2 の点の認識結果

これらの結果より、著者らが想定する条件が整っている環境で撮影した点字においては、顔認識技術を用いて点を正しく検出することができ、検出した点を点字の情報として出力できることがわかった。

4. まとめと課題

本稿で著者らは、一定の環境下で撮影した点字画像から顔認識技術を用いて点を認識し、点字の 1 セルの情報を出力できることを報告した。

撮影画像のヒストグラムから計算したしきい値において正しく 2 値化処理することができ、さらに、墨字の除去も行うことができた。また、600 点以上の特徴サンプルを使って作成した検出器を用いて 2 値化処理後の点から点を認識することに成功した。その結果、認識した点を分割し点字の 1 セルの情報として出力することに成功した。

今後の課題として、撮影環境の条件の緩和が挙げられる。今回の評価で撮影距離を変化させたとき、点の分割が困難なことがわかった。この点について早急に対応したいと考えている。合わせて、検出器による認識率の変化と特徴サンプルの関係を明らかにすると同時に分割線の予測アルゴリズム再検討が行う必要がある。

また、現在の手法では空列から始まる点字を誤認識する可能性もあるため、この点の検討が必要である。

参考文献

[1] 平成 18 年身体障害児・者実態調査結果:厚生労働省 社会・援護局 障害保健福祉部 企画課, <http://www.mhlw.go.jp/toukei/saikin/hw/shintai/06/dl/01.pdf>, 2008.

[2] 電子タグと携帯電話を活用した視覚障がい者のための公共トイレ音声案内システムに関する調査研究:電子タグと携帯電話を活用した視覚障がい者のための公共トイレ音声案内システムに関する調査研究会, <http://www.soumu.go.jp/soutsu/hokuriku/resarch/houkoku/1.1.pdf>, p.6, 2005.

[3] 田中真美, 宮田薫, 西澤達夫, 長南征二: 点字読み取り用触覚センサシステムの開発, 日本機械学会[No.02-9] Dynamics and Design Conference 2002 CD-ROM 論文集, 815, 2002.

[4] 細洞知也, 仁田周一, 武藤篤生: CCD カメラを用いた点字の抽出と認識, 1997 年電子情報通信学会総合大会, p.260, 1997.

[5] OpenCV, <http://opencv.org/>

[6] JIS T0923:2009(高齢者・障害者配慮設計指針一点字の表示原則及び点字表示方法—消費生活製品の操作部): 日本規格協会, 2009.

[7] Cascade Classifier Training — OpenCV 2.4.11.0 documentation,

http://docs.opencv.org/doc/user_guide/ug_traincascade.html