

携帯型カメラを用いた視覚障害者向け点字認識システム

Braille Recognition System with a Portable Camera for Visually Handicapped People

美原 義行[†] 杉本 晃宏^{††} 柴山 悦哉[†] 高橋 伸^{†††}

Yoshiyuki MIHARA Akihiro SUGIMOTO Etsuya SHIBAYAMA Shin TAKAHASHI

[†] 東京工業大学 ^{††} 国立情報学研究所 ^{†††} 筑波大学

Tokyo Institute of Technology National Institute of Informatics University of Tsukuba

mihara1@is.titech.ac.jp

本研究では、視覚に障害をもちながらも点字を読むことができない人を対象に、頭部に装着したカメラからの入力画像から点字を認識して、音声化するシステムを提案する。本システムを装着したユーザは点字を探し出し、読んで欲しい点字を指差しによって指定する。システムは、ユーザが点字を探している状態と点字を指定している状態とを識別する。そして、点字を指定している状態であると認識すると、指の動きから画像内の点字領域を切り出し、その領域内にある点字を認識し、その内容を音声化する。

1 はじめに

全国で視覚に障害を持つ人¹は約 31 万人いると言われている [1]。そのうち点字を読めない視覚障害者は、全体の約 9 割にも上る [2]。この原因として中途失明してしまうケースが近年増えていること、そして年をとってからでは点字の習得が難しいことが挙げられる。高齢化社会に伴い、今後ますます後天的な視覚障害者が増え、点字を読むことができない視覚障害者が増えることが予想される。一方、現状では、駅の切符売り場など、視覚障害者向けの情報提供には点字を用いたものが多い。そのため、実世界の点字を認識して音読するシステムの普及が期待されている。

現在、点字を認識するシステムでは O.B.R (Optical Braille Recognition) [3][4] が普及している。これは点字文書をスキャナで読み取ることにより、その内容を音声化してユーザに伝える。一方、視覚障害者がエレベータで目的の階に行ったり、切符を購入するためには、点字を自ら探し、探し当てた点字をシステムに認識させ、その点字の内容を知り、目的のボタンを探し出す必要がある。従来の O.B.R は、この様にユーザが持ち歩いて、点字を認識させる目的には向かない。このような実世界の点字認識システムによって、ユーザが一人で切符を買えたり、エ

レベータで目的の階に行けるようになることが期待できる。

実世界の点字読み取りシステムとしては、触覚センサを用いたものが既に提案されている [5]。このシステムは本研究と同じ動機の下、点字が読めない視覚障害者のために開発されている。触覚センサの利点はセンサが小さく、携帯性に優れていることが挙げられる。そして、これによりセンサを点字の近くまでもっていくことが可能であり、小さな点字を正確に認識することが期待できる。一方でセンサを指に装着しなければならないため、視覚障害者が手探りで点字を探す作業の邪魔になる可能性がある。

本研究では、ユーザが携帯型カメラを装着し、ウェアラブルコンピュータによって入力画像から点字を認識し、その結果を音声によってユーザに伝えるシステムを提案する (図 1)。提案システムでは、ユーザが探し当てた点字を撮影するため、またユーザの点字を探す作業を邪魔しないように、カメラを頭部に装着する。それにより、視覚障害者でも自分の指先の方向に頭を向けられるので、手探りで見つけた点字の方向に頭部 (カメラ) を向けることができ、目的の点字を撮影することができる。

2 携帯型カメラを用いた点字認識システム

本システムを装着したユーザは、まず、点字標識の前に立ち、以下の動作を行う。

¹ 身体障害者福祉法 (昭和 24 年法律 283 号) 別表第 5 号 (第 7 条関係) に規定される視覚障害者等級が 6 級以上の者 (一眼の視力が 0.02 以上のもの)

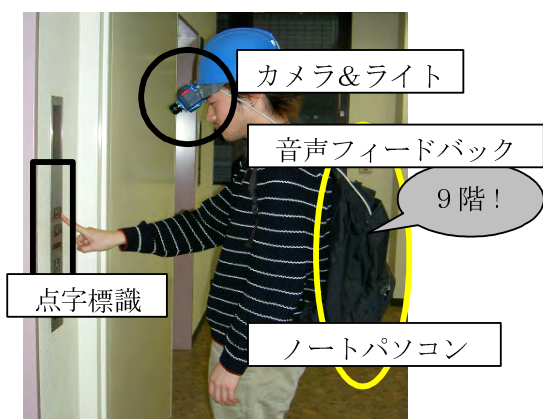


図 1: システム概観

1. 点字標識を手探りで探す .
2. 点字を探し当てたら , システムに知らせる .

システムが , カメラの前にある点字をすべて認識してしまうと , ユーザはその点字がどこにあるかが分からず , 混乱してしまう . そこで , システムは , カメラに写ったすべての点字を読み上げるのではなく , ユーザが指定した点字のみを読み上げる . エレベータや切符の自動販売機で各ボタンの意味をユーザが知るためには , このような選択的な読み上げが必要になる .

これを実現するために , システムはユーザが点字を探している状態と , 探し当てた点字を指定している状態を識別する必要がある . そのために , 本研究では以下のようなジェスチャーを導入した .

- 手探りの状態 :
両手 (片手) を開いて点字を探す (図 2) .
- 点字を探し出した状態 :
点字の下に指をずらし , 点字を指差す (図 3) .
- 点字の範囲を指定している状態 :
点字を指差しながら , 指を左右に動かす (図 4) .

これらは自然なジェスチャーであり , ユーザにとって覚えやすく , 負担が少ない .

ユーザは , 点字を探している間 , および点字を指差し , 範囲を指定している間 , システムが自分の状態を正確に判断しているかに関して不安を感じる . そこで本研究では , システムが判断したユーザの状態を音で表現することにした . 具体的には , ユーザに音の高低でフィードバックする .



図 2: 手探りの状態



図 3: 探し出した状態

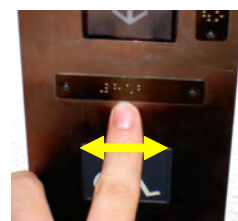


図 4: 点字の範囲を指定

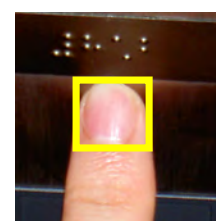


図 5: ブロックの例

3 実装

3.1 システムの処理の流れ

本システムの主な処理を図 6 に示す . まず , カメラからの入力画像を解析して , システムはユーザが点字を探している状態であると判断したとき , また , 指を一本も検出できなかったときは , そのまま指先のトラッキングを続ける . 点字を指差している状態であると判断したときは , フレームの間の指の動きを見て , 指先の領域を切り出す . 次に , 切り出した領域内の点字を認識して , その内容をユーザに音声でフィードバックする .

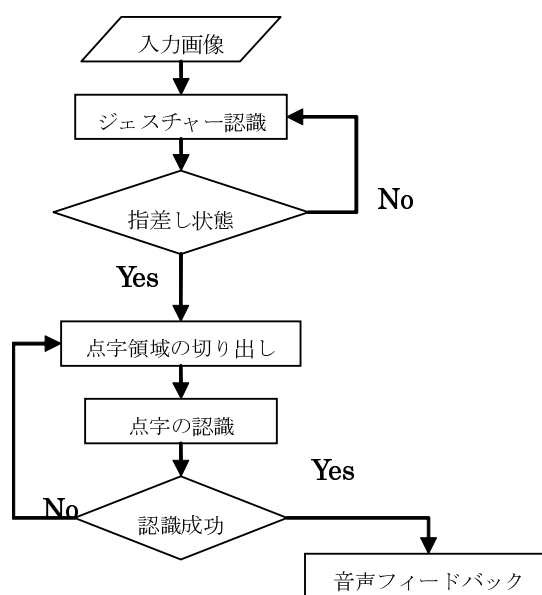


図 6: 処理の流れ

3.2 ジェスチャー認識

ジェスチャー認識においては、フレーム内に指があるかどうかを判定する。そのために、ユーザの指先(爪)のカラーモデルをあらかじめ取得しておく。本研究では、指先のカラーモデルは以下の変換式で求められる rg 平面を用いて生成した。

$$r = \frac{R}{R+G+B}, g = \frac{G}{R+G+B}$$

rg 平面を用いると、照明の変化に対してもロバストに肌色を認識できることが知られている [6]。

まず、指先と同程度の大きさのブロックを考える(図5)。点字とカメラの距離はほぼ一定で、カメラに写る指の大きさもほぼ一定であることから、ブロックの大きさは固定した。指先のブロックのすべての画素について RGB 値を rg 平面に射影してそのブロックのカラーヒストグラムを作成し、カラーモデルとして保持しておく。次に、入力画像を走査して、フレーム内に事前に作っておいたモデルと類似するブロックがあるかどうか探索する。そして、フレーム内にこのようなブロックが一つだけ検出されたとき指差し状態であると判断し、類似するブロックが多数見つかった場合は手探り状態であると判断する。

3.3 点字領域の切り出し

点字を指差している状態であると判断したとき、システムは指の動きを見て ROI (Region of Interest) を切り出す。このとき切り出す ROI の幅は、指が一本であると判定されてから、その指が動かされた範囲とする。また、カメラとの点字の距離はほぼ一定なので、切り出す ROI の高さを固定した。

3.4 点字の認識

点字は触って読むものであり、見るために作られていない。そのため、点字は目立たないように背景色と色分けされていないことが多く、点字を構成する個々の点は小さい。このため、ノイズの影響で、点でないところを点と判断してしまう可能性がある。さらに、頭部に装着したカメラで斜め上方から点字を撮影することになり、画像上では点と点が重なって観測されてしまう可能性がある。

このような問題を解決するために、点字からの反射光を抽出するような画像処理を施した。また、認識にはテンプレートマッチングを用いた。さらに、点

字にライトを当てることにより、点字抽出の精度を高める工夫を行った。

点字に反射した光を抽出するために、まず Sobel のエッジ抽出を行い、閾値を用いて二値化する(図8)。ライトを当てているため、どの点もほぼ同様の光り方をする。また、二値画像において点を表す領域の大きさや形状は、ほとんど同じになる。そこで、点と見なされた領域のみに注目して点字を認識する。

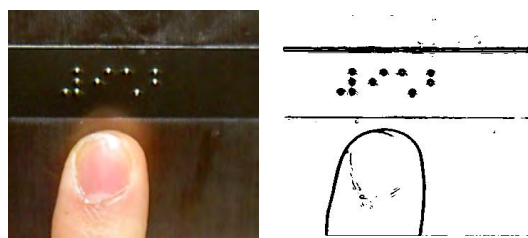


図 7: 切り出した領域 図 8: その二値画像

テンプレートマッチングではサンプルの点字列を数多く生成し、その点字列を二値画像に当てはめ、最も確からしいサンプルを採用する。二値画像において、点字は必ずこのサンプルの点列上にくるはずであり、この点列上にはない点はノイズであると判断することができる。また、点同士が重なってしまった領域についても、サンプルの点列を見ることにより点であると判断する。

点字は縦 3 点、横 2 列の 6 点から一マスが構成されている。そして、点の間隔、点の大きさが標準化されているという特徴をもっている。この特徴からサンプルの点字列を簡単に表現できる(図9)。

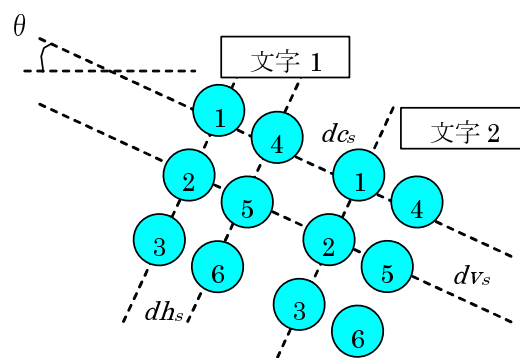


図 9: 点列のサンプル

テンプレートの状態 X_s を以下のように定義する。

$$X_s = (x_s, y_s, k_s, dh_s, dv_s, dc_s (= \alpha \cdot dh_s))$$

(x_s, y_s) は 1 番目の文字の点 1 の重心座標であり, サンプルを生成する上で基準点となるものである. 初期値は点と判断された領域の中で, x 座標が一番小さい領域における x 座標の最小値, y 座標の最小値である. k_s は $\tan \theta$ であり, θ は水平方向に対するカメラの傾きである. dh_s は点の横方向の間隔 (点 1-点 4 間), dv_s は点の縦方向の間隔 (点 1-点 2 間) である. また, dc_s は点字間隔 (点 4-点 1 間) であり, dh_s の定数倍で表すことができる. 点字の間隔が標準化されていることから点同士の縦方向, 横方向の間隔は同じ比率である. しかし, 撮影条件によってこの比率通りに写らないことがあるので, 縦方向と横方向の間隔をそれぞれ変数として持つことにした.

以上の変数に適当な値を代入していき, サンプル点列を生成する.

次にサンプル点列が確からしいかどうかを判断する. サンプルの尤度は以下の基準をもとに決定する.

1. 点とみなされた領域内にサンプルの点が含まれているかどうか, そしてその領域内に必要以上にサンプルの点が含まれていないか.
2. このサンプルの点列を二値画像に当てはめたとき, 対応する文字があるかどうか.

これらの基準を下に, サンプルごとに尤度を求め, 尤度が最大になるサンプルを採用する.

4 評価実験

4.1 実験方法

東京工業大学内エレベータの点字表示と東急大岡山駅の切符販売機の点字表示に対して, 本システムを用いて認識実験を行った. エレベータでは 6ヶ所の点字を 5 回ずつ計 30 回認識実験を行った. また切符売り場では 5ヶ所の点字を 3 回ずつ計 15 回認識実験を行った.

4.2 実験条件

エレベータは室内にあり, 外光の影響を受けない. 一方, 駅は外光の影響を受ける環境であった. 実験に用いたカメラとノートパソコンは以下の通りである.

- カメラ: ARTRAY ARTCAM-OV200
 - 解像度: 1600x1200

- レンズ画角: 30.0 度

- ノートパソコン

- CPU: Intel Pentium M 1GHz
- メモリ: 768MB

点字とカメラの距離は約 45cm であった. 画角は 30 度 (対象までの距離が 450mm では視野 240mm × 180mm) であるが, この程度の画角があれば, 視覚障害者の方でも目標の点字を写すことが可能である. なお, このとき一つの点は約 10x10 ピクセル, タテ, ヨコの間隔はどちらも約 3 ピクセルで撮影された. 実験場所において指先のカラーモデルを生成した. また, テンプレートマッチングを行うため生成した点列のサンプルの数は約 4500 とした.

4.3 実験結果

エレベータにおける実験ではすべて正しく点字を認識することができた. 認識例を図 10 に示す. ROI を切り出してから点字を認識するまでの実行平均時間は 273.5ms であり, 認識した点字を読み上げるのに約 1 秒かかった. 点字を指差してから, ROI を切り出すまでにユーザの指の動きを約 3 秒観察するので, ユーザにフィードバックするまで平均で約 4 秒かかった.

一方, 切符売り場においては, 15 回の実験中で正しく点字を認識できたのは 3 回のみであり, 正しく認識できたのは 1ヶ所のみであった. 正しく認識できなかった 12 回は切符販売機の点字が文字の上に位置していたため, 点字だけを認識することができなかったためである (図 11).

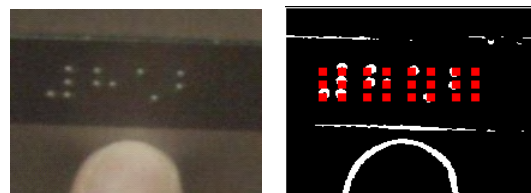


図 10: 切り出した ROI とその二値画像にサンプル点列を当てはめたもの



図 11: 点字の下に文字がある表示とその二値画像

4.4 インタビュー

実験以外に、実際に本システムを視覚障害者の方 4 名に見てもらった。ユーザが点字を指差してからシステムが点字を読むまで約 4 秒かかることについては、「問題ない」という回答をいただいた。またユーザの状態を音の高低でフィードバックすることについても「大変良い、覚えやすい」という回答をいただいた。

5 考察

点字の下に文字がない場合はすべて正しく認識できたが、点字の後ろに文字があり、点と重なってしまう場合には正しく認識できなかった。

正しく認識できなかった原因は、図 11 の二値画像のように「10」という文字と点が同じ領域になってしまい、点字領域だけを切り出すことができなかつたためである。現在の認識手法では、点の領域であるかどうかを大きさや形状のみから判断している。そのため、文字と点と同じ領域になってしまうと、その領域は点のような形状をしていないため、その領域を点の領域と認識できなくなる。このため、この部分の点を見落とすことになってしまった。

本研究ではテンプレートマッチングを用いることにより、点がないところに発生する誤検出に対応しているが、図 11 のように点があるのにないと判断してしまうエラーからは回復できない。この問題に対処するには、意味的な推測により点を補完することが有用であると考えられる。例えば、画像認識で場所、状況を把握してそこに書かれている点字を推測する手法が考えられる。

一方、正しく認識できなかった別の原因として、二値化するまでに用いた画像処理の手法が安易であったことも挙げられる。本手法では、エッジを抽出して閾値を用いて二値化しているだけなので、後ろに文字がある場合には、二値化した際に点と文字の領域を分割することができなかった。点に反射した光だけを抽出するように画像処理の手法を改善することによって、この問題に対処できると考えられる。

さらに、設備の面でも工夫することにより認識率の向上が期待できる。具体的には、シャッタースピード、絞り等を調節して光量を少なくし、フラッシュライトのような強い光を点字に当てることで、文字が写らずに点字に反射した光だけを撮影することができると考えられる。

6 まとめ

本研究では点字を読むことができない視覚障害者を対象に、頭に装着したカメラを利用して、点字を認識するシステムを開発し、その有用性を検証する実験・評価を行った。今後は、第 5 章に挙げた課題を解決するとともに、点字表示までの誘導支援も視野に入れ、機能を拡張したいと考えている。

参考文献

- [1] 厚生労働省社会・援護局障害保健福祉部: “身体障害児・者実態調査結果 (平成 13 年 6 月 1 日調査)”, 2002.
- [2] 財務省印刷局: “障害者白書 (平成 14 年度版)”, 内閣府編, 2002.
- [3] R. Ritchings, A. Antonacopoulos and D. Drakopoulos: “Analysis of Scanned Braille Documents”, *Proc. of IAPR Workshop on Document Analysis System*, pp.417-424, 1994.
- [4] X. Fernandez Hermida, A. Corbacho Rodriguez and F. Martin Rodriguez: “A Braille O.C.R. for Blind People”, *Proc. of International Conference on Signal Processing Applications and Technology*, 1996
- [5] T. Nishizawa and M. Tanaka: “The Development of a Tactile Sensor System for Reading Braille”, *Proc. of Center on Disabilities Technology and Persons with Disabilities Conference*, 2003
- [6] J. Terrillon, A. Pilpre, Y. Niwa and K. Yamamoto: “Analysis of Human Skin Color Images for a Large Set of Color Spaces and for Different Camera Systems”, *Proc. of IAPR Workshop on Machine Vision Application*, pp.20-25, 2002