

小型感圧センサを用いた点字の読み取り手法 Braille Detection Method with a Micro Pressure Sensor

伊藤祥一*
Shoichi Ito

1 はじめに

一般に点字は視覚障害者が使う文字として理解されているが、視覚障害者の 8 割以上は事故や病気による中途失明者が占めており、点字を習得している視覚障害者は全体の 1 割程度である [1]。一方で社会的に点字の必要性が認知されつつあり、公共施設や家電製品をはじめとして点字の並記が広がっている。点字を読めない人に代わって小さなデバイスがこれらの情報を読んでくれるれば一人で出歩くことも楽になり、プライバシーにかかわる情報を晴眼者に読んでもらう必要もなくなる。そのためには常に身につけておけるウェアラブルなデバイスであることが望ましく、我々は指先に装着した小型感圧センサで点字をなぞると日本語音声で読み上げてくれるウェアラブルデバイスの研究に取り組んでいる [2]。

本稿では、このデバイスを実現するための要素技術の一つとして、感圧センサで点字をなぞったときの一連の圧力データから元の点字の並びを再構成する方法について提案する。

2 圧力データ

本研究では感圧センサとしてイナバゴム株式会社製圧力分布システム YP-USP01A を用いる。0.2mm 四方の感圧素子 60×60 個が並んでおり、各素子で 2^{16} 段階の圧力を取得することができる。以下では i 行 j 列目の圧力データを $a_{i,j}$ のように表す ($0 \leq i, j \leq 59$)。感圧センサで連続的に N 個のフレームを取得した場合は $a_{i,j}^{(k)}$ のようにフレーム番号を表す添字 k を括弧つきで上につける ($0 \leq k \leq N-1$)。

図 1 は「かいだん」を表す $\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot\cdot$ という点字を左から右に、3 文字目の濁点符が入りきるあたりまで実機でなぞった様子である。実際は点字 5 文字分で 175 個のフレームから構成されている。本稿ではこの 175 フレームからなる圧力データをサンプルとして実際の動作を説明する。

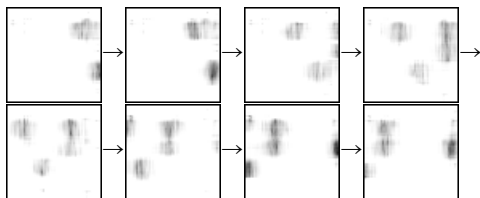


図 1 「かいだん」をなぞった際の圧力データ (15 フレームごと)

3 行座標の取得

3.1 行和の取得

まず、すべての圧力データから 1 行ごとのデータの総和 (行和) b_i を式 (1) により求める。

$$b_i = \sum_{k=0}^{N-1} \sum_{j=0}^{59} a_{i,j}^{(k)} \quad (i = 0, 1, \dots, 59) \quad (1)$$

点字には行が 3 つあるが、圧力値を行ごとに足し上げることで、点字の行に対応する部分は必然的に大きな値になる。

サンプルデータを式 (1) で処理した行和を図 2 に示す。図 2 は縦軸に上から行番号 $i = 0, 1, 2, \dots$ をとり、横軸に対応する b_i の値をプロットしたものである。圧力データの単なる和なので山の高さにはゆらぎがある。

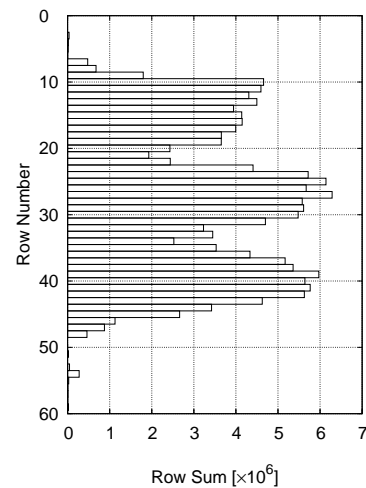


図 2 式 (1) による行和

3.2 行和の補正

図 2 を見ると、点字の凸点のある行を示していると思われる山を 3 つ確認することができる。このように人の目で見れば行座標を推測することは可能であるが、機械的に処理することはそのままでは困難である。そこで後段で機械的な処理をしやすくするため、求めた行和の補正処理を行う。

センサの感圧素子 1 個の間隔を距離 1 としたとき、距離 α だけ離れた場所の重みを返す関数 $w(\alpha)$ には式 (2) を用いる。これは JIS T0923 [3] で定められている点字の凸点の形状を楕円の半円に近似して得たものである。

$$w(\alpha) = 0.4 \sqrt{1 - \left(\frac{4}{15}\alpha\right)^2}, \quad |\alpha| \leq \frac{15}{4} \quad (2)$$

i 行目に注目したとき、1 行下 ($i+1$ 行目) の行和における重みは $\alpha = 1$ 、2 行上 ($i-2$ 行目) の行和における重みは $\alpha = -2$ を式 (2) に代入することで求める。 i 行目の行和 b_i を補正した値を c_i とし、注目している行の前後 3 行までを補正として取り込むと、 c_i は式 (3) となる。ただしセンサの上端 3 行と下端 3 行を除いて考え、 $i = 3, 4, \dots, 56$ である。

* 長野工業高等専門学校 電子情報工学科 Department of Electronics and Computer Science, National Institute of Technology, Nagano College

$$c_i = \sum_{\alpha=-3}^3 w(\alpha)b_{i+\alpha} \quad (i = 3, 4, \dots, 56) \quad (3)$$

式 (3) で計算された補正後の行和を図 3 に示す. 図 3 は図 2 と比較して山の形が滑らかになり, 山の頂点の位置を推測しやすい形になっている.

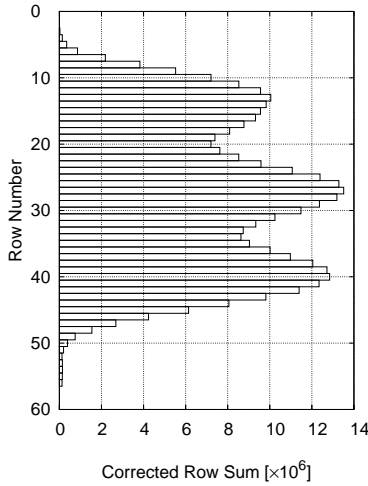


図 3 式 (3) による補正後の行和



図 4 同じ場所に乗り続けるゴーストノイズ

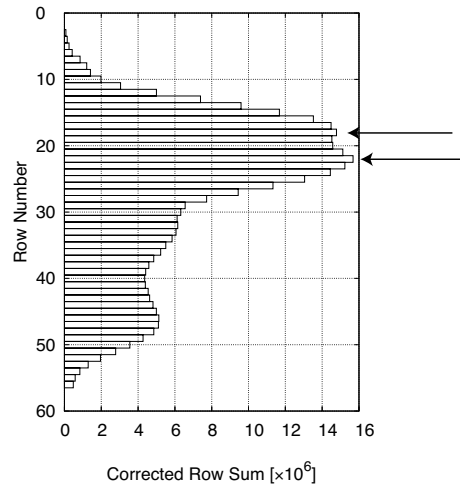


図 5 補正後の行和で近い位置に山が 2 つできてしまった例

3.3 行座標の取得

点字には縦に 3 つの行がある. 感圧センサの縦 60 個の素子のうち何行目がこの 3 つの行のそれぞれに対応するか, 補正された行和の値から推測する. 点字の i 行目の行座標を R_i ($i = 1, 2, 3$) とする. R_i の取り得る範囲は式 (4) とする.

$$0 \leq R_1 < R_2 < R_3 \leq 59 \quad (4)$$

手順 1

まず補正後の行和 c_i について, 前後の何行よりも値が大きいかを数える. すなわち, 特定の i について,

$$c_i > c_{i-\beta} \quad (0 \leq i - \beta \leq 59) \quad (5)$$

$$c_i > c_{i+\beta} \quad (0 \leq i + \beta \leq 59) \quad (6)$$

が同時に満たされる最大の正整数 β を探す.

手順 2

手順 1 で求めた β のうち, 値が一定以上の行 (ここでは 3 とした) の上位 3 つを残す. サンプルデータではこの条件で 27 行目 ($\beta = 27$), 40 行目 ($\beta = 11$), 13 行目 ($\beta = 10$) の 3 行が残った. もし β が一定以上となる行が存在しなければ, 行座標を推測できないとしてエラーとする.

値が一定 (ここでは 3) 以上の行のみを残した理由は感圧センサの特性を考慮したためである. 実際の圧力データを与えたとき, ほとんどのフレームの同じ場所に図 4 左上の○で囲んだ部分のような影 (ゴースト) が乗り続ける場合があった. この場合, 図 5 の 18 行目と 22 行目 (矢印の位置) のように, 近い位置に 2 つの山のピークができてしまう. ゴーストによる山は手順 1 で数えた整数 β が小さいという特徴があるため, ある程度より β が大きい行のみを残すことでゴーストでできる小さな山のピークが除外される.

手順 3

手順 2 で残った行数が 3 の場合, 行番号の昇順に R_1, R_2, R_3 として行座標の取得を終了する. 手順 2 の例であれば, $R_1 = 13, R_2 = 27, R_3 = 40$ となる. 縦 60 個の感圧素子のほぼ中央に点字が置かれたとすれば, この結果は妥当である.

手順 4

手順 2 で残った行数が 2 の場合, 残り 1 つを以下の手順で推測する.

1. 手順 2 で取得した 2 つの行座標のうち, 小さい方を A , 大きい方を B として, $\delta \leftarrow B - A$ を計算する.
2. $\delta \geq 20$ ならば, A は 1 行目の座標, B は 3 行目の座標であると考え, A と B の平均値を 2 行目の座標とする. すなわち, $R_1 \leftarrow A, R_2 \leftarrow (A + B)/2, R_3 \leftarrow B$ とする.
3. $\delta < 20$ ならば, A と B は隣接する 2 行であると考え, この場合, 上の 2 行なのか下の 2 行なのかを決めなければならない.
 - (a) $A - \delta < 0$ ならば, A が 1 行目, B が 2 行目であると考え, 3 行目は 1-2 行目と同じだけ空いていると考え, $R_1 \leftarrow A, R_2 \leftarrow B, R_3 \leftarrow B + \delta$ とする.
 - (b) $B + \delta \geq 60$ ならば, A が 2 行目, B が 3 行目であると考え, 1 行目は 2-3 行目と同じだけ空いていると考え, $R_1 \leftarrow A - \delta, R_2 \leftarrow A, R_3 \leftarrow B$ とする.
 - (c) 上記 2 つに当てはまらない場合, $D \leftarrow A - \delta, E \leftarrow B + \delta$ とし, 補正後の行和 c_i を比較する.
 - i. $c_D > c_E$ ならば, この 2 行は下の 2 行であると考え, $R_1 \leftarrow A - \delta, R_2 \leftarrow A, R_3 \leftarrow B$ とする.
 - ii. $c_D < c_E$ ならば, この 2 行は上の 2 行であると考え, $R_1 \leftarrow A, R_2 \leftarrow B, R_3 \leftarrow B + \delta$ とする.

手順5

手順2で残った行数が1の場合、これをAとする。残り2つを推測するため、隣り合う行との間隔δをここでは12と決め打ちして以下のように処理する。

1. $D \leftarrow A + \delta$ と $E \leftarrow A + 2\delta$ について、 $3 \leq D, E \leq 56$ ならば $s_1 \leftarrow c_D + c_E$ とする。
2. $D \leftarrow A - \delta$ と $E \leftarrow A + \delta$ について、 $3 \leq D, E \leq 56$ ならば $s_2 \leftarrow c_D + c_E$ とする。
3. $D \leftarrow A - 2\delta$ と $E \leftarrow A - \delta$ について、 $3 \leq D, E \leq 56$ ならば $s_3 \leftarrow c_D + c_E$ とする。
4. $S \leftarrow \max(s_1, s_2, s_3)$ とする。
5. $S = s_1$ ならば A は 1 行目の座標であると考え、 $R_1 \leftarrow A, R_2 \leftarrow A + \delta, R_3 \leftarrow A + 2\delta$ とする。
6. $S = s_2$ ならば A は 2 行目の座標であると考え、 $R_1 \leftarrow A - \delta, R_2 \leftarrow A, R_3 \leftarrow A + \delta$ とする。
7. $S = s_3$ ならば A は 3 行目の座標であると考え、 $R_1 \leftarrow A - 2\delta, R_2 \leftarrow A - \delta, R_3 \leftarrow A$ とする。

ここまでで R_1, R_2, R_3 が決まった。

4 列座標の取得

行和の取得と補正の処理を列についても行い、その後で列座標を取得する。

4.1 列和の取得

k フレーム目の圧力データからそのフレームの列ごとの総和 $d_j^{(k)}$ を式 (7) により求める。

$$d_j^{(k)} = \sum_{i=0}^{59} a_{i,j}^{(k)} \quad (j = 0, 1, \dots, 59) \quad (7)$$

行のときは式 (1) で N 個すべてのフレームについて足し上げを行ったが、列についてはフレームごとに点の場所が異なる可能性があるため個別のフレームについて処理を行う必要がある。例として図1の右上1フレーム分の圧力データから式 (7) で計算された列和を図6に示す。

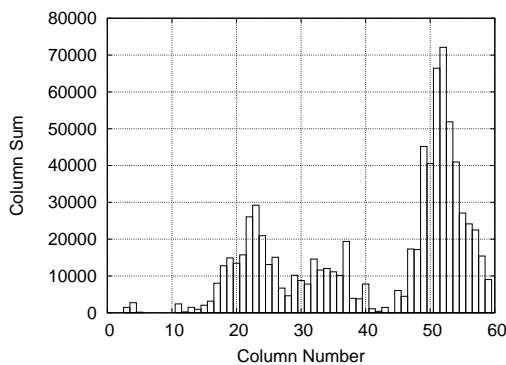


図6 式(7)による列和

4.2 列和の補正

3.2章で行った行和の補正と同様に、求めた列和 $d_j^{(k)}$ に対して周囲の列を考慮した重み付けを行う。

重み付けには式 (2) を用い、注目している列から α マス左の列の重みを $w(-\alpha)$ 、注目している列から α マス右の列の重みを $w(\alpha)$ とする。k フレーム目の j 列目の列和 $d_j^{(k)}$ を補正し

た値を $e_j^{(k)}$ とする。注目している列の左右3列までを補正として取り込むと、 $e_j^{(k)}$ は式 (8) となる。ただし $j = 3, 4, \dots, 56$ である。

$$e_j^{(k)} = \sum_{\alpha=-3}^3 w(\alpha) d_{j+\alpha}^{(k)} \quad (j = 3, 4, \dots, 56) \quad (8)$$

図1の右上の1フレーム分の圧力データから式 (8) で計算された補正後の列和を図7に示す。

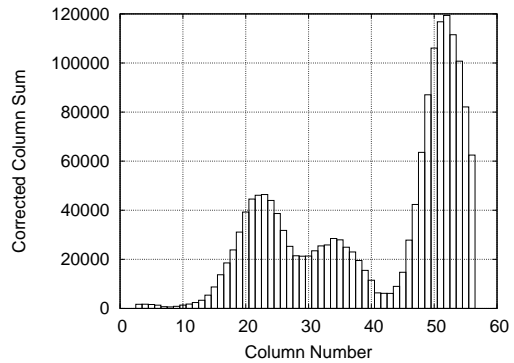


図7 式(8)による補正後の列和

4.3 列座標の取得

補正後の列和から列座標を取得する。k 番目のフレームの60列について式 (8) により列和の補正值 $e_j^{(k)}$ が得られたとする。この $e_j^{(k)}$ ($j = 3, 4, \dots, 56$) について、前後の何列よりも値が大きいかを数える。すなわち、特定の j, k について、

$$e_j^{(k)} > e_{j-\beta}^{(k)} \quad (0 \leq j - \beta \leq 59) \quad (9)$$

$$e_j^{(k)} > e_{j+\beta}^{(k)} \quad (0 \leq j + \beta \leq 59) \quad (10)$$

が同時に満たされる最大の正整数 β を探す。

$\beta \geq 4$ の列番号 j を列座標として取得する。図7に示された補正後の列和から列座標を取得すると23列目 ($\beta = 20$)、34列目 ($\beta = 7$)、52列目 ($\beta = 4$)、の3つが列座標として求まる。これらの値を図7と比較すると山が局所的に頂点を持つような列座標をうまく取得できている。

5 点情報の取得

3章で取得した行座標 R_1, R_2, R_3 は N 個のフレームすべてで共通である。4章で取得した列座標は N 個のフレーム1つ1つで異なる。これらを図1の1フレームごとに重ね合わせてプロットしたものが図8である。ここから、交点に凸点が存在するかどうかを判断する。

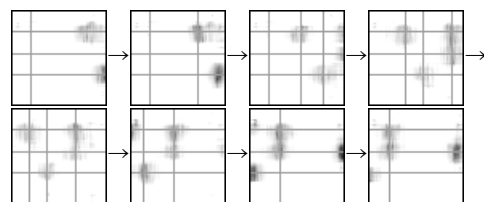


図8 取得された行座標と列座標 (図1に対応)

5.1 補正值の計算

図8の交点が本当に凸点かどうかを判断するためにここでまず補正值を計算する。式(2)から、注目している交点の周囲3マスずつを補正として取り込めばよい。k番目のフレームで交点の行座標がi, 列座標がjだったとき、この交点での補正值 $f_{i,j}^{(k)}$ は式(11)により求める。

$$f_{i,j}^{(k)} = \sum_{\beta=-3}^3 \sum_{\gamma=-3}^3 w(\sqrt{\beta^2 + \gamma^2}) a_{i+\beta, j+\gamma}^{(k)} \quad (11)$$

5.2 補正值の比較

取得した補正值 $f_{i,j}^{(k)}$ を元に点字の凸点かどうかの判断をする。感圧素子に点字を指で押しつけた際の強弱は一定ではないので、しきい値を固定して使うのではなく、フレームごとの強弱を凸点の有無の判定に反映する必要がある。具体的には以下の手順を実行する。

1. 式(11)で計算された補正值 $f_{i,j}^{(k)}$ は、1つの列(=1つのj)について最大で3つ(3行分)の補正值がある。1行目の値を g_1 , 2行目の値を g_2 , 3行目の値を g_3 とする。
2. $G \leftarrow \max(g_1, g_2, g_3)$ とする。
3. $G < 6000$ ならばその列に凸点はないと確定する。
4. $2g_1 > G$ ならば1行目に凸点があると判断する。
5. $3g_2 > G$ ならば2行目に凸点があると判断する。
6. $2g_3 > G$ ならば3行目に凸点があると判断する。

2行目の凸点判定(上記5)のみ条件が異なる。これは、実機での圧力データの値が比較的小さかったため、凸点と判断する基準をすこし緩く設定しているためである。

ここまでの処理で、凸点が存在する列の列座標と、その列における凸点の有無が取得できる。図9は図8の交点情報をもとに取得された凸点の場所をプロットしたものである。

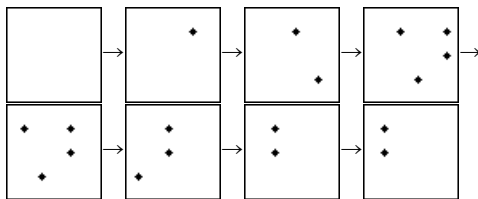


図9 取得された凸点の位置(図8に対応)

6 凸点の位置情報の統合

次に図9の個々のフレームの凸点の位置を1つの大きなフレームに統合する。

6.1 列座標のシフト

5章では連続するN個のフレームの1つ1つで凸点の座標を抽出した。このうち凸点が1つもないフレームを除外して、凸点情報を持つフレームがM個になったとする。除外したフレームの間を詰めて0から順に番号を振り直し、個々のフレームを

$$0 \leq k \leq M-1 \quad (12)$$

で区別する。k番目のフレームに含まれる凸点の列数を $p^{(k)}$ で表す。k番目のフレーム中のi番目の列座標を $h_i^{(k)}$ で表す。ここで $h_i^{(k)}$ は列座標を保持しているので

$$0 \leq h_i^{(k)} \leq 59 \quad (13)$$

である。iは取得された順に $0, 1, \dots, p^{(k)} - 1$ である。

M個のフレームのそれぞれは左から右へ点字文字列をなぞる際の移動量を考慮していない単発の圧力データの集合である。5章で得られた凸点の情報を隣接するフレーム間で比較することで移動量を推測し、M個あるフレームを1個の大きなフレームに統合する。具体的には以下の手順を実行する。

1. オフセット $\Delta \leftarrow 0$ とする。
2. $k = 1, 2, \dots, M-1$ について手順3~7の処理を行う。
3. $i = 0, 1, \dots, p^{(k)} - 1$ について手順4~5の処理を行う。
4. $j = 0, 1, \dots, p^{(k-1)} - 1$ について、 $|h_i^{(k)} - h_j^{(k-1)}|$ の最小値 δ を求める。
5. $\delta \leq 5$ ならば、k-1番目のフレームで $h_i^{(k-1)}$ の列に存在していた点が移動してk番目のフレームのどこかに映り込んでいると判断する。このときの δ の値を σ_i として保存する。 $\delta > 5$ ならば $\sigma_i \leftarrow 0$ とする。
6. 手順5で求めた σ_i のうち、0でないものの平均値 ϵ を求め、 $\Delta \leftarrow \Delta + \epsilon$ とする。 ϵ はすべての列座標の組み合わせでk-1番目のフレームからk番目のフレームになったときに平均的にどの程度の移動量があるかということの意味している。
7. Δ は $k=0$ のフレームからの凸点の位置の横方向のずれを蓄積しているものなので、k個目のフレームの列座標すべてに Δ を加算する。

$$h_i^{(k)} \leftarrow h_i^{(k)} + \Delta \quad (i = 0, 1, \dots, p^{(k)} - 1) \quad (14)$$

ここまでで、 $k = 0, 1, \dots, M-1$ 番目のフレームについて、これまでは1つ1つのフレームの中での列座標を保持していたものを、 $k=0$ のフレームから見てどれくらいの距離にあるかという座標に焼き直したものが得られている。図10に例を示す。図10左列は、図9の左上のフレームから15, 30, 45, 60フレーム後を取りだして縦に並べたものである。どのフレームも左端から $0, 1, \dots, 59$ という列座標が振られている。先の手順で先頭フレームからのオフセット Δ が後続のフレームに足されていくので、図10右列のように、各フレームの左端の列座標が14, 26, 38のようにずれていく。

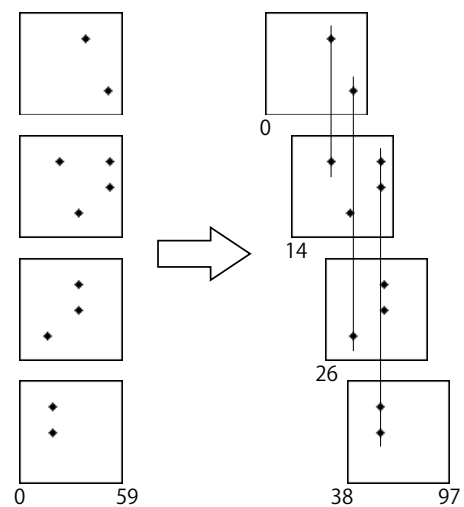


図10 図9に対して列座標のシフトを行った結果

6.2 凸点情報の統合

次に凸点情報の統合を行い、図 10 ではバラバラになっているフレームを 1 枚にまとめる。隣接する 2 個のフレームを見て同じ凸点と見なしうるものを残し、時間軸上で後のフレームに出現した凸点を追加する。これで 2 個のフレームの情報が 1 個に統合されたので、この処理を繰り返し、最終的に 1 個の統合された点情報を得る。具体的には以下の手順を実行する。

1. $k = 1, 3, 5, \dots$ について手順 2~5 の処理を行う。
2. $i = 0, 1, \dots, p^{(k)} - 1$ について手順 3~5 の処理を行う。
3. $j = 0, 1, \dots, p^{(k-1)} - 1$ について、 $|h_i^{(k)} - h_j^{(k-1)}|$ の最小値 δ を求める。最小となるときに列座標 $T \leftarrow h_i^{(k)}$ とそのときのインデックス $l \leftarrow j$ を保存しておく。
4. $\delta \leq 7$ ならば、 $h_l^{(k-1)} \leftarrow h_l^{(k-1)} + h_i^{(k)}$ とする。
5. $\delta > 7$ ならば、その列は $k-1$ 番目のフレームから k 番目のフレームになったときに新しく出現した列であるとみなし、 $k-1$ 番目のフレームに新たな列として列座標 T を追加する。
6. $k = 1, 3, 5, \dots$ のフレームがそれぞれ $k-1 = 0, 2, 4, \dots$ のフレームに統合されたので、 $k = 1, 3, 5, \dots$ のフレームをすべて除外する。フレーム数が $M/2$ 個になる。除外したフレームの間を詰めて先頭からフレーム番号を $0, 1, 2, \dots, M/2 - 1$ と振り直す。
7. フレーム数が 1 になるまで手順 1~6 を繰り返す。
8. 1 つに統合された点情報の各列座標および点のパターンについて、足し合わせた回数が 5 以上のものの平均をとり、小数点以下を四捨五入して凸点の有無を決める。足し合わせた回数が 5 未満のものは突発的なノイズであるとみなし、無視する。

1 つ前のフレームから列がどれだけ移動したかという情報をもとに列座標の補正を行い、補正後の列座標同士で値の近いものを重ね合わせ、最後に重ね合わせた回数で平均を取ることによって、各列座標とその列における点の有無を多数決の要領で決める。1 枚に統合された点の配置を、左端の点の列座標を 0 として、1 つ前の列座標からの横方向の距離と、その列にある 3 つの凸点の有無を 1/0 で表すと実行結果 1 の結果が得られた。

実行結果 1 統合された点の列座標とパターン

1	0.000000	100
2	11.644703	001
3	18.100000	110
4	41.141319	010
5	17.065738	101
6	12.193760	010
7	17.253457	001
8	12.729060	011

これを図示すると図 11 となる。図 11 は「かいだん」という点字文字列 $\begin{matrix} \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & & \cdot & \cdot & \cdot \\ \cdot & & \cdot & \cdot & \cdot \end{matrix}$ を表しており、 N 個の圧力データの集まりから、1 つの点字文字列を構成する凸点の集まりを 1 枚の絵として統合することができている。

7 点字の取得

最後に、統合後の点情報を基に空列なども考慮した最終的な点字の並びを取得する。図 11 の時点で手元にあるのは凸点

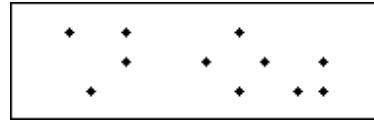


図 11 統合された凸点の配置(「かいだん」)

表 1 列間距離の実測値

図 12 上の変数名	値
d_1	11.8174
d_2	16.8653
d_3	29.2796
d_4	40.8077

のある座標のみである。点字では凸点がない空列も文字の一部として重要な意味がある。点字文字列を正確に日本語に翻訳するためには、凸点の座標データから空列が間にいくつ挟まっているかを推測しながら、図 11 の凸点の集合を $\cdot, \cdot, \cdot, \cdot, \cdot, \cdot, \cdot, \cdot, \cdot$ の文字単位に分離しなければならない。

点字は物理的な設計が決まっているので、感圧センサで点字を取り込んだときの列間距離をあるていど決め打ちする。図 12 のように列間距離をパラメトライズし、実際に測定されたデータから平均値を求め、表 1 の値を使うこととした。

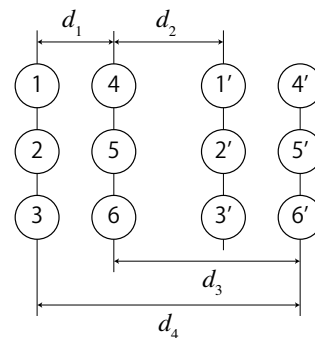


図 12 点字の取得処理に必要となる 4 つの列間距離

点情報を統合した結果、凸点のある列数 (= 実行結果 1 の行数) が L 列になったとする。凸点の列間距離を調べ、適切に空列を挿入する操作を繰り返す。 x_j には、途中に空列を挟んでいる場合も含め、 $j-1$ 列目と j 列目が図 12 の d_1 から d_4 のどのパターンにもっとも近いかを 1~4 の数値で保持する。 y_j は j 列目が点字 1 文字の中の 1 列目か 2 列目をか 1 か 2 で保持する。たとえば $y_{j-1} = y_j = 1$ ならば、隣接する $j-1$ 列目と j 列目の凸点はいずれも点字 1 文字の中の 1 列目という意味なので、間に空列が 1 つはさまるべきということがわかる。凸点の集合から点字 1 文字ごとに切り出す手順は具体的には以下の通りである。なおここでは便宜上、点字 1 文字の中での 1 列目を A 列、2 列目を B 列と書く。

1. 2, 3, ..., L 列目における前の列との列間距離 (実行結果 1 の左列の 2 行目以降の値) を $v_2, v_3, \dots, v_{L-1}, v_L$ とする。
2. $j = 2, 3, \dots, L$ について v_j と表 1 の d_1, d_2, d_3, d_4 との差の絶対値を計算する。差の絶対値が最小となる d_i の i を $x_j \leftarrow i$ として保存する。
3. $j = 2, 3, \dots, L$ について以下を行う。

- (a) $j > 2$ かつ $x_{j-1} = x_j = 1$ の場合, $v_j \leq v_{j-1}$ ならば j 列目は B 列であるとして 2 列前まで番号付けして $y_j \leftarrow 2, y_{j-1} \leftarrow 1, y_{j-2} \leftarrow 2$ とする.
 - (b) $j > 2$ かつ $x_{j-1} = x_j = 1$ ではない場合, j 列目は B 列であるとして 1 列前まで番号付けして $y_j \leftarrow 2, y_{j-1} \leftarrow 1$ とする.
 - (c) $x_j = 2$ かつ $v_j \leq v_{j-1}$ ならば j 列目は A 列であるとして 1 列前まで番号付けして $y_j \leftarrow 1, y_{j-1} \leftarrow 2$ とする.
4. $j = L, L-1, \dots, 1$ について最初に y_j が 1 または 2 となる j を探し, $z \leftarrow j+1$ とする.
 5. $z \leq L$ ならば, $j = z, z+1, \dots, L$ について以下を行う.
 - (a) $x_j = 3$ ならば, A 列か B 列かはわからないが, 図 12 から間に 1 列空列があることがわかるので, $y_j \leftarrow y_{j-1}$ とする.
 - (b) $x_j = 4$ かつ $y_{j-1} = 1$ ならば, $j-1$ 列目は A 列でそこから 2 列はさんで j 列目になっているということなので $y_j \leftarrow 2$ とする.
 - (c) $x_j = 4$ かつ $y_{j-1} = 2$ ならば, $j-1$ 列目は B 列でそこから 2 列はさんで j 列目になっているということなので $y_j \leftarrow 1$ とする.
 6. $j = L-1, L-2, \dots, 1$ 列目のまだ列番号が振られていない列について以下を行う.
 - (a) $x_{j+1} = 3$ ならば, $y_j \leftarrow y_{j+1}$ とする.
 - (b) $x_{j+1} = 4$ かつ $y_{j+1} = 1$ ならば, $j+1$ 列目は A 列でそこから 2 列はさんで j 列目になっているということなので $y_j \leftarrow 2$ とする.
 - (c) $x_{j+1} = 4$ かつ $y_{j+1} = 2$ ならば, $j+1$ 列目は B 列でそこから 2 列はさんで j 列目になっているということなので $y_j \leftarrow 1$ とする.
 7. $j = 2, 3, \dots, L$ について以下を行う.
 - (a) $y_j = y_{j-1}$ ならば, j 列目と $j-1$ 列目の間に空列を 1 つ挿入する.
 - (b) もし $x_j \geq 3$ かつ $y_j \neq y_{j-1}$ ならば, j 列目と $j-1$ 列目の間に空列を 2 つ挿入する.
 8. $y_1 = 2$ ならば, 現在先頭にある列が B 列であるので先頭に空列を 1 つ挿入する.
 9. $y_L = 1$ ならば, 現在末尾にある列が A 列であるので最後尾に空列を 1 つ挿入する.

サンプルデータを元に取得した最終的な点字のパターンは実行結果 2 のようになった. 左から 3 行 2 列のパターンで, 1 が凸点のある部分, 0 が無い部分である. $\begin{matrix} \bullet & \bullet & \bullet & \bullet & \bullet & \bullet \\ \bullet & \bullet & \bullet & \bullet & \bullet & \bullet \\ \bullet & \bullet & \bullet & \bullet & \bullet & \bullet \end{matrix}$ に対応するパターンが正確に取得できている. あとはこれを点訳すれば, 圧力データの集合から「かいだん」という日本語文字列が得られたことになる.

実行結果 2 最終的に得られた点字パターン

1	10	10	00	10	00
2	00	10	01	01	01
3	01	00	00	10	11

8 評価

ここまでで説明したアルゴリズムの評価のため, 点字の文字列 28 個を用意し, 感圧センサでなぞった圧力データを処理した. 評価用の点字文字列はキングジム社製テブラ PRO

SR6700D で作成した. 本研究で用いている感圧センサは感圧素子部が基板に固定されているため点字を感圧素子部でなぞるのではなく, 感圧素子部に点字を押し当ててスライドさせるという方法で圧力データを取得した. その結果, 28 個の点字文字列のうち, 14 個を正確に再構成することができた. 「点字図書館」という点字の圧力データを「てんじとしょあん」としてしまふ(「か」の 6 の点欠落して $\begin{matrix} \bullet & \bullet & \bullet & \bullet & \bullet & \bullet \\ \bullet & \bullet & \bullet & \bullet & \bullet & \bullet \\ \bullet & \bullet & \bullet & \bullet & \bullet & \bullet \end{matrix} \rightarrow \begin{matrix} \bullet & \bullet & \bullet & \bullet & \bullet & \bullet \\ \bullet & \bullet & \bullet & \bullet & \bullet & \bullet \\ \bullet & \bullet & \bullet & \bullet & \bullet & \bullet \end{matrix}$ となった) など, 1 つの文字列中に 2 つ以内の点の誤りが見られたものが 5 個あった. 同様に 3 つ以上の点の誤りが見られたものが 8 個あった. 誤検出が特定の字種に偏るといった傾向はなく, 点字を感圧素子部に押し当てる際の力の入れ加減などが影響しているものと考えられる.

9 まとめ

点字文字列を感圧センサでなぞったときの一連の圧力データから点字文字列の並びを再構成するアルゴリズムについて提案した. 28 個の点字文字列について評価を行ったところ, 14 個について圧力データから点字文字列を正確に再構成することができ, 誤りが 2 点以内で再構成できたものも 5 個あった. 点字の凸点の形状に由来する圧力の分布や常に同じ場所に影が出るゴーストなど, 感圧センサで点字をなぞる用途に特有の事情も考慮しつつ, 比較的良好な精度で点字文字列の再構成ができています.

本稿で述べたアルゴリズムには今回使用した感圧センサの実機にあわせて調整したパラメータが複数含まれており, 将来的に多様な機種に本アルゴリズムを適用するためにはこれらのパラメータの自動調整の仕組みが必要である. 今回は特定の点字プリンタからの出力を評価に用いたが, 現実の点字は金属やプラスチックなど多様な素材に刻印されているため圧力分布の様子が変わることが想定される. また, 物理的なサイズも JIS と微妙にずれていることが多くある. 今後はさらに多様な点字に対応できるようにアルゴリズムの改良を進めるとともに, 点字を日本語に翻訳する際にスペルコレクタの仕組みを取り入れて, 圧力データから誤った点字文字列が再構成されてしまっても出力される日本語は正しく修正されているといった機能を取り込む [4].

本アルゴリズムと小型感圧センサを用いてウェアラブルな点字読み上げデバイスを完成させることで, 視覚障害者の生活品質の向上に役立てていきたい. なお, 本研究は JSPS 科研費 JP19K12917 の助成を受けたものである.

参考文献

- [1] 厚生労働省社会・援護局福祉部企画課: 平成 18 年身体障害児・者実態調査結果, <https://www.mhlw.go.jp/toukei/saikin/hw/shintai/06/d1/01.pdf> (2006).
- [2] 伊藤祥一, 藤澤義範: 点字読み取りデバイスの開発, 第 16 回情報科学技術フォーラム講演論文集, pp.447-448 (2017).
- [3] 日本規格協会: 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 (2009).
- [4] 下平啓太, 伊藤祥一: 圧力センサから読み込まれた点字のスペル訂正アルゴリズム, 情報処理学会第 82 回全国大会講演論文集, pp.4-395-4-396 (2020).