

点字自動読み取りシステム

情報処理学科 柴田 進吉

要 旨：点字で書かれた文書を読みとり、かな、英数字に変換する簡便なシステムを作成した。点字文書はイメージスキャナでイメージとして読みとり、その中から文字情報を抽出する。イメージスキャナの制御、文字抽出はパソコンを使用する。数式などで多い点字の多義性による誤訳に対処するため、かな優先モード、英数字優先モードを用意して解読を支援している。

キーワード：点字、点字墨字変換、視覚障害教育、障害者支援

はじめに

パーソナルコンピュータ用に音声読み上げ、点字表示のソフト、ハードウェアが開発され、全盲者にもコンピュータの操作、利用が可能になり、電子ファイルによる情報交換ができるようになった。これは晴眼者との間で文字による相互のコミュニケーションが容易になったということで画期的なことであり、本学視覚部でも講義に資料として電子ファイル（フロッピーディスク）を渡したり、答案をワープロ、エディタなどで作成し、電子ファイルで提出という形態が多くなっている。勿論、これで従来使用してきた点字が不要になるということはない。コンピュータが使用できないときはいうに及ばず、テキスト作成の速度が問題になる答案作成などの場合、特に現在のコンピュータでは入力ができない場合がある。数学記号、特殊記号などを扱う場合などは、点字盤、点字タイプライタによる点字テキストの作成が有利である。一方、日常的に点字を使用する機会が多くない晴眼者にとって点字テキストの正確な速読は不得意なことが多い。学生が社会で活躍する場合も同様な状況がある。また、点字には2級英語という短縮表現があり、解読に難しさを加えている。気軽に使える点字テキストの自動読み取りシステムはこのような場合の補助手段として有効であろう。これが本システム開発のひとつの動機である。また、このシステムは原稿に電子ファイルを持たない点字テキストの電子ファイル化、点字プリンタの誤り検出などにも使用できる。

システムの概要

ハードウェア

イメージスキャナ IS300-M（システムクオリティ）がパラレルインターフェイス（セントロニクス規格）でパーソナルコンピュータ PC9801に接続されている。

ソフトウェア

当初イメージスキャナのテストにBASICを用いた関係ですべてBASICである。高速化を図るため機械語にコンパイルして使用する。プログラムの大きさは機械語にコンパイルして26KBである。

点字の大きさと配列

点字は1列3点が2列並んだ、6つの点で1マスを構成している。向かって左上から下に1、2、3の点、次の列の上から4、5、6の点と呼んでいる。

普及している仲村製の点字盤では、点の直径は約1mm、高さは0.5mm程度で、縦方向の点間隔は約2.5mm、列間隔は約2mm、マス間隔は約5mmである。1行の文字数は32程度で、行間隔は15mmである。マスの大きさはどの印字器も大体同じであるが、行間隔は10mm程度のももある。無駄を省いて点字のありそうなところだけ読み取り走査し高速化するために、テキストがどの印字器で作られたかを情報として入力し、利用している。

これまでの経験では、点字盤と点筆を用いた、手による印字は点間隔、行間隔は殆ど一定しているが、点字プリンタには、打点位置や、紙送りが一定でないものがあり、マス間隔、行間隔が不整で、読み取りの走査領域を余裕をもって広くする必要なものがあった。

処理の流れ

1. 点字テキストをイメージとして読み取る。
2. イメージの中から点字成分を抽出する。
3. 点字を解読し、かな・英数字・記号列に変換する。
4. 画面に結果を表示、またはプリンタに印刷する。

各処理の概要

1. 点字テキストの読み取り

凹凸で表現された点字を読み取るには、凹凸を機械的に検出する方法、光学的に検出する方法などが考えられる。物理的、化学的に汚れたテキストの読み取りを考えると機械的方法が有利であるが、実現の容易さを考慮して、今回は手元にあったイメージスキャナを利用して読み取ることにした。イメージスキャナは画素毎に濃度を3桁の数字で表したものをデータとして出力する。

点字の凹凸をスキャナで走査してみると、スキャナの光学系により、凸面よりも凹面を走査するほうが明瞭に読み取れる。

紙面に汚れ、点字以外の凹凸があると雑音になる。この雑音は後でイメージ・点字成分抽出の段階で黒レベルの濃さ、点字があると予想される部分のみ探す、などの方法で除去する。

走査時間を短縮するため、行間の白紙の部分は除き、点字列があると予想される場所を走査する。このためテキスト作成に使用した点字盤、点字タイプライタの種類から、点間隔、マス間隔、行長、行間隔を情報として利用する。

点字は通常の文字と違って点の有無の検出でよいので、スキャナは300ドット/インチで走査するが、メモリの節約のため180ドット/インチに圧縮して読みだし、更に、点字成分抽出のとき走査領域を狭め高速化するため、そのデータを紙面の縦方向に1/2に圧縮している。この時点で点字の1点は4×3ドット程度である。

現在用いているイメージスキャナはA4版まで一括して読み込めるが、小規模のシステムでも使えるようにメモリ容量を考慮して、1行分を読み、処理をし、次の行を読み込むようにしている。読み込み速度は4秒/行である。

2. イメージデータから点字成分の抽出

紙面の水平方向をx軸、縦方向をy軸とし、走査を始める原点を左上隅(XORG, YORG)とする。点字の1と2の点間距離をYGAP, 1と3の点間距離をXGAP, 4の点と隣のマスの1の点の距離をXSTEP 0, マス間隔をXSTEP 1, 行間隔をYSTEPとする。これらの値は点字器毎に異なる値を設定する。

a) 行頭の1または4の点の検出

テキストの最初の行を見つけるため、(XORG, YORG)から出発し、1行分のイメージデータ領域の上左隅からX方向に雑音成分以上の濃度の点があるか否かを調べて行く。x方向に連続した3点の濃度の合計が雑音レベルの4倍以上あれば点字の点と判断している。紙面が傾いていたり、プリンタの印字ずれがあると、行

頭ではなく後部の点を誤認するので、Y方向に走査をずらして再走査し、X座標の小さい点が検出されれば、それを最初の点字の1または4の点の座標(X1, Y1)とする。座標の決定は最初に発見した黒点を原点とし、X方向に5点、Y方向に4点の長方形の領域を走査し、X方向、Y方向の座標の重心を求める。

b) 2または5の点の座標の検出

座標(XORG, Y1 + YGAP)からX方向に走査し、点の重心座標(X2, Y2)を求める。

c) 3または6の点の座標の検出

座標(XORG, Y1 + 2 * YGAP)からX方向に走査し、点の重心座標(X3, Y3)を求める。

d) 文頭の点字の1の点の座標を求める。

(X1, Y1), (X2, Y2), (X3, Y3)の3点のX座標、Y座標のなかから最小のものを選び、仮の原点(X0, Y0)とする。仮の原点が1の点か、4の点かを識別するため、X座標が、X0 + XGAPまたはX0 + XSTEP 1 * n (n = 1 ~ 31)の場所をY方向に走査し、点があれば(X0, Y0)を文頭の1の点の座標とする。点を検出されなければ、X0 + XSTEP 0またはX0 + XSTEP 1 * nの場所をY方向に走査し、点があれば(X0 - XGAP, Y0)を文頭の点字の1の点の座標とする。

e) 点字列の読み取り

文頭の原点から順に、ひとマスの中の1の点から6の点の予想位置での黒点の有無を調べ、黒点があれば“1”, 無ければ“0”とし、6の点を最上位とする6桁の2進数に換算する。続いて隣の点字を次々とX座標の最後まで読み取っていく。読み取った数値換算の点字列は配列にしまわれる。空白は数値0となる。

点字を印字するときに印字位置がずれたり、スキャナに紙を置く位置がずれたりすることがある。これに対処するため一定文字数毎に点の重心座標を再測定し、予測座標に補正を加える。現在は7文字毎に再測定している。

3. 点字列から仮名・英数字・記号への翻訳

点字の解釈は日本点字委員会の日本点字表記法、点字数学記号解説に示されている文法を用いている。この他に情報処理点字、北米コンピュータ点字など一部に文法の異なるものがある。これらについては順次、辞書を整備して対応する。2級英語も今後の課題である。

点字から仮名・記号への変換は、1対1の対応をしているものと、そうでないものに分れる。後者は解読の対象となる文字の前後の文字を調べて決めることになる。例えば、2の点のみの点字は促音符(ツ)であるが、文字列が0, 2, 2, 2, 0と続くと点線(・・・)になる。点字は通常何も指定しないときはかな文字を意味するが、外文字符がつくと英字、数字符号がつくと数字を

意味することになっている。これら3つのモードはそれぞれ1つの点字の意味が前後関係で決まるものがあるので、それぞれについて辞書を用意してある。また、“1ねんめ”と書くときは、1の前に数字符号がつくが、“ね”の前にはかな符号などというものはつかないので、“ね”は数字ではないということを判別しなければならない。

4. 表示, 印刷

数式, 特殊記号はプリンタによって表現できない記号もあるので, 簡単なプリンタを想定して, 文字で表現するようにしている。例えば, \int は integral, ベクトル記号は vector などである。豊富なフォントを内蔵しているプリンタ用に辞書を変更するのは, 文字の代わりに記号に対応した数値を出力すればよいので簡単である。

課題と対策

1. 点字テキストの電子ファイル化

光学的読み取りの問題点は先にも触れたように原紙の汚れ, 折れ曲がりなどによる変形で点字の点が無いところが黒と読み取られる場合である。今回は教室で書かれた点字テキストやプリンタで印字された新しいものを対象に考えたので汚れの問題は殆どないが, 薄い点字用紙に点密度の大きい文字, 例えば1の点から6の点まである“メ”のような文字を多く印字すると, 紙が歪み, イメージスキャナガラスの面に密着せず, 反射率が下がり, 雑音となる。これは紙面をテンプレートや, スポンジのような柔軟なもので抑えることでガラス面へ密着させる必要がある。物理的, 化学的に汚れた紙面は回復が困難であるが, 例えば, 点字の凸部のみを黒に着色すれば, 信号対雑音比のよい読み取りができる。

着色発泡インキで作られた点字テキストの読み取りは点字盤, 点字プリンタに比べてはるかに明瞭に読み取れる。

2. 点字から, かな・英数字・記号への変換

点字の文法には機械的な解釈が難しいものがある。通常

の文章は殆ど問題がないが, 数式を含むものは誤変換の可能性もある。例えば, 1%と1pとは点字の表記は同じであり, 分脈を加味して解釈しなければならない。また“3xyz”はxの前に外文字符号をつけなくてよいので, “3ふむま”とも読める。また, “15/s”は“15夜の”とも読める。点字の多義性に対処するには知的な辞書が必要である。現在は, かなモード, 英字モードの2つのモードを指定できるようになっていて, 英字モードを指定すると外文字符号がついていなくても英字として読む。文章として正しくないものは, その行を指定して, かなと英字モードの組み合わせで解読することになっている。

3. 処理の高速化

現在, パソコンにPC9821Ap2を用いると, スキャナからの読み込みに4秒/行, 点字の抽出, 解釈に1秒/行を要し, 1ページ(約800字)の処理に約2.5分を要する。実用システムには更に高速化することが望ましい。高速化のためには, イメージスキャナ, インターフェイス, パソコンを性能の良いものにする, および処理ソフトウェアの改善が必要である。ハードウェアは主に経済的な問題であるが, ソフトウェアは処理アルゴリズム, 使用言語を検討する必要がある。処理アルゴリズムについては全画面, あるいは複数行の一括読み込み, 無駄な領域の読み飛ばしなどを予定している。使用言語はBASICをコンパイルしているが, C言語に書き換えることでも改善の可能性はある。

4. 両面印字のテキストへの対応

使用したイメージスキャナでは点字の凹面の方が凸面より明瞭に読みとれるが, 凸面でも読みとりは可能である。両面印字テキストの読みとりは, 行間隔のバラツキが少なく, 両面の行が重ならなければ十分可能である。1枚の紙面の情報量を多くするために両面印字の点字テキストが普及している, これに対処することも必要である。