

視覚障害者のための情報支援 システムに関する研究

A Study on Information Support System
for Visually Impaired

宮本 修

2012年7月

早稲田大学大学院 先進理工学研究科
電気・情報生命専攻 学習型信号・情報処理システム研究

概要

- 研究背景

視覚障害者は、眼疾患によって視力、視野、色覚、光覚等の視機能障害が発生し、その結果として読み書き、歩行、日常生活上に能力障害が起こっている人々である。これらは社会的な不利として、就労面・経済的自立での制約に結びつくことが多い。平成18年身体障害児・者実態調査結果によると、身体障害者全体(総数3,483,000人)、及び身体障害児(総数93,100人)のうち、視覚障害者数は総数310,000人であり、視覚障害児総数は4,900人である。

視覚障害者が遭遇している困難は、主に

「移動に伴う困難」、「情報入手・発信に伴う困難」

がある。前者は道路・交通機関・公共建築物等の利用時において、また後者は日常生活において書籍や新聞などを読む際のみならず災害時や緊急時の情報提供・通信、そして避難誘導時において現れる。これらの側面に対する視覚障害者への配慮はたとえば次のようなものがある：

- ・各地の点字図書館、視覚障害者にさまざまな情報を点字、音声データで提供するネットワーク。
- ・駅における手すりや券売機に設置された点字による説明。
- ・歩道や鉄道駅などにおける突起のついた「視覚障害者誘導ブロック」、いわゆる「点字ブロック」。
- ・横断歩道や駅の階段における音声・音楽による誘導。
- ・視覚障害者用の腕時計。ふたを開けて針の位置を指で確認できる。また時間合わせを自動でしてくれる電波時計は、視覚障害者にも便がよく、音声読み上げタイプは日常生活用品として用いられる。
- ・パソコンやネットワーク関係では、マウス等による場所指定操作が困難な人々のための音声読み上げソフトによるツール。

これらを含め、視覚障害者の生活の質の向上を目指す多くの試みや整備が行われているが、改善の余地を残すものも少なくない。これらを整理すると視覚障害者が抱える課題はいくつかに類別され、たとえば

- (a) 日常生活での移動に関する必須の課題
- (b) 社会が必要とする課題

(c) 情報化時代に必須の課題

が考えられる。

これらに関して本論文の著者はいくつかの体験をもった。

一つ目は駅構内に設置されている手すり上の点字案内である。手すり上の点字案内を実際に調査する機会があったが、間違っただ点字が設置されている（打ってある）場合がとても多いことに驚かされた。2000年に鉄道の点字運賃表の間違いが相次いで発見されたにもかかわらず、その後も点字運賃表の誤りは発見され続けている。このことは、現場の点字確認作業がかなり困難であることを示していると考えられる。

二つ目は、点訳作業を実際に行うことの困難である。本論文の著者は大学で点字サークルに所属し、視覚障害者に関する多くの事実を知ることになったが、自分がまずできることの一つとして点字翻訳を学び、いくつかの点訳を試みた。点訳作業は点訳者の熟練、意志、そして忍耐を必要とする。特に一冊の書籍、大量の重要書類などの点訳作業をスムーズに行うには多大の点訳訓練が必要であり、需要に対して点訳者の数は圧倒的に少ないのが現状である。したがっていわゆるベストセラーとなった書籍は比較的速やかに点訳されるものの、一般にはまだまだ読みたい本が点訳されている率は低いと言わざるを得ず、点訳本、点訳者の増加は視覚障害者の切望するところとなっている。

コンピュータによる自動点訳の試みはいくつかある。しかしそのような自動点訳は、もとの媒体から電子化された情報に基づき、いわゆる「分かち書き」などを自動的に行おうとするものである。目的の印刷物から視覚障害者が実際に使える点訳物にする過程はまだまだといってよい。このような過程を遂行するには人間の力が必要であるが、そういったものはボランティアに依存している部分が多く、そのような人は、目が悪い人々に関心がある人間である。従って視覚障害者に関心のある人間を増やすことは自動点訳を行うことと同じくらい大切と考えられる。

三つ目は視覚障害者のためのWEBブラウザである。本論文の著者はいくつかの視覚障害者のための特別なWEBブラウザに触れる機会があったが、これからの情報化社会で活躍するためにも改善の余地があると常々感じていた。

● 本研究の目的

上述の問題を解決する手法は、大学で身に着けた情報技術の知見を通して構築するのが自然ではないかと思ひ、上述の (a), (b), (c) から

- (i) 点字デジタルカメラ画像の自動誤り訂正システム
- (ii) コーパスへの検索による点訳の分かち書き誤り修正システム
- (iii) 視覚障害者のための特殊 WEB ブラウザ

をとりあげ、視覚障害者の生活の質向上にささやかではあっても貢献したいと考えた。本論文の目的は「視覚障害者のための情報支援システム」を、上記3つの側面から取り上げてアルゴリズムを構築したうえ、それを実装してその性能を検証することである。

- 本論文の構成

第1章は、先行研究の概観をふくめた本論文の導入であり、以後、各章の内容は次のような構成になっている：

第2章では、「点字デジタルカメラ画像の自動誤り訂正システム」について述べている。デジタルカメラで撮影したステンレス製の点字を認識する場合、既存のイメージスキャナを使った点字認識のアルゴリズムをそのまま適用できないので、提案するシステムは、以下の方法でこれらの問題を解決した。

- (i) 点の存在の検出については、ストロボを発光させて撮影し、ストロボの光が点に反射したパターンを認識した。
- (ii) 点の位置の検出については、動的計画法を用いたマッチングを行うことにより、レンズの収差を吸収した。わい曲収差のあるサンプル 5102 マスに対して検証実験を行ったところ変動吸収なしの場合認識率 81.24 % であったのに対し、提案システムでは 98.73 % であった。

第3章では「コーパスへの検索による点訳の分かち書き誤り修正システム」について述べている。提案システムは、点訳された文から、その分かち書きの誤りを判別して修正するものである。提案システムで修正を行う直接の対象は、点訳ファイルであり、点字の文字コードによって記述されている。提案システムは、修正対象のある一部分を取り出し、それが大量の点訳ファイル（「点訳コーパス」と呼ぶ）の中に何件含まれているかを調べる。この含まれた件数を「ヒット件数」と呼び、その値から分かち書きの誤りを判別する。検証実験を行い、修正前の平均誤り率が 11.52% であったのに対して提案システムによる修正後の平均誤り率は 2.12% となった。このシステムにより点訳初心者の訓練に用いる事ができると考えられる。

第4章では「視覚障害者のための特殊 WEB ブラウザ」について述べてい

る。本論文の筆者が調べたいいくつかの WEB ブラウザでは、ウェブサイトのデザインやレイアウトが少し変更されただけでも、特殊 WEB ブラウザは機能しなくなる可能性が高く、また目的とするサイト以外のサイトへのアクセス能力は、全くないかきわめて限られている。プログラミング言語を用いれば、視覚障害者用の特殊 WEB ブラウザを作成するのはさほど難しくない。しかしながら、ウェブサイトの小さなデザイン変更をすべて把握し、WEB ブラウザ・プログラムの修正に反映しなければならない。したがって視覚障害者用の表計算プログラムのようなより有効な WEB ブラウザが必要であると考えられる。そのような考え方に基づいて視覚障害者用の特殊 WEB ブラウザを簡単に構築するためのツールキットを提案し、検証実験として視覚障害者が書籍通販サイトの商品を注文するために使用できる特殊 WEB ブラウザを開発している。

第 5 章「結論」では本論文の成果をまとめ、これからの方向性について述べている。

目次

第 1 章	序論	1
1.1	研究背景	1
1.2	本研究の目的	4
1.3	本論文の構成	5
1.4	先行研究	5
第 2 章	点字デジタルカメラ画像の自動誤り訂正システム	8
2.1	はじめに	8
2.2	提案システム	12
2.3	実験	20
2.4	まとめ	23
第 3 章	コーパスへの検索による点訳の分かち書き誤り修正システム	30
3.1	はじめに	30
3.2	問題提起	31
3.3	点字翻訳の問題点	32
3.4	提案システム	33
3.5	実験	42
3.6	結果と考察	46
3.7	おわりに	49
第 4 章	視覚障害者のための特殊 WEB ブラウザ	51
4.1	汎用 WEB ブラウザと特殊 WEB ブラウザ	51
4.2	提案システム	55

vi 目次

4.3	実験	59
4.4	おわりに	60
第 5 章	結論	61
5.1	本研究で得られた成果	61
5.2	今後の課題と結言	62
	謝辞	64
	参考文献	65
	著者の業績	68

第1章

序論

1.1 研究背景

視覚障害者は、眼疾患によって視力、視野、色覚、光覚等の視機能障害が発生し、その結果として読み書き、歩行、日常生活上に能力障害が起こっている人々である。これらは就労面・経済的自立での制約・不利に結びつくことが多い。平成18年身体障害児・者実態調査結果 [1] によると、身体障害者全体（総数 3,483,000 人）、及び身体障害児（総数 93,100 人）のうち視覚障害者数は総数 310,000 人であり、視覚障害児総数は 4,900 人である。視覚障害者が遭遇している困難に

- 移動に伴う困難
- 情報入手・発信に伴う困難

がある。前者は道路・交通機関・公共建築物等の利用時において、また後者は日常生活においてのみならず災害時や緊急時の情報提供・通信、そして避難誘導時において現れる。これらの側面に対する視覚障害者への配慮は比較的多く、たとえば次のようなものがある [3][4]：

- 各地の点字図書館、視覚障害者にさまざまな情報を点字、音声データで提供するネットワーク。
- 鉄道駅における手すりや券売機における点字による説明。
- 横断歩道や駅の階段での音声・音楽による誘導。
- 移動歩道や鉄道駅などでの突起のついた「視覚障害者誘導ブロック」、いわゆる

2 第1章 序論

る「点字ブロック」.

- 視覚障害者用の腕時計. ふたを開けて針の位置を指で確認できる. また時間合わせを自動でしてくれる電波時計は, 視覚障害者にも便がよく, 音声読み上げタイプは日常生活用品として用いられる.
- 洗濯機などの家電製品にはスイッチ部分に点字を刻印してあるものがある.
- パソコンやネットワーク関係では, マウス等による場所指定操作が困難な人々のためのスクリーンリーダー (音声読み上げソフトによる案内).
- 携帯電話は画面読み上げ機能がついたものがある.
- 消耗品の容器には, 例えばリンスやヘアコンディショナーと区別するために刻み模様を入れたりして視覚障害者への配慮がされたものがある.
- 牛乳パックでは上部の張り合わせの部分丸く切り取ってあるものがある.
- アルコール飲料の缶の上部などに点字を刻印してあるものがある.
- 工業規格日本工業規格 (JIS) JIS X8341 「高齢者・障害者等配慮設計指針-情報通信における機器, ソフトウェア及びサービス」では, ハードウェア・ソフトウェアやウェブコンテンツについての配慮設計指針が示されている.
- 紙幣日本銀行券 (紙幣) の下端右側または左右に, 指触りで金種を識別するためのくぼみが漉き込まれている. 日本の紙幣は, 異なる額面の紙幣は, 互いに大きさが異なるよう (額面の大きい金種の大きさが大きくなるよう) 設計されている. 多くの国では額面が大きいほど紙幣の大きさも大きくなるが, 一部の国の紙幣 (アメリカ合衆国ドル紙幣など) は, 金種が異なっても大きさが同一であり, 視覚障害者にとって使いづらいとされる.

これらを含め, 視覚障害者の生活の質の向上を目指す多くの試みや整備が行われているが, 改善の余地を残すものも少なくない. これらを自分なりに整理すると視覚障害者が抱える問題はいくつかに類別され, たとえば

- 日常生活での移動に関する必須の課題
- 社会が必要とする課題
- 情報化時代に必須の課題

が考えられる. これらの課題は各々重要であるが, 点字に関連する課題が多い. 点字は視覚障害者の重要な情報収集・発信方法となっているが, 障害等級が1・2級 (点字が必要な障害のレベル) の中で実際に点字を活用することが可能な人々は, 全体の

10 % を僅かに超えるにとどまっている [2]. 視覚障害者が情報を入手する方法は, (1) テレビが 72.4 %, (2) 家族・友人が 58.5 %, (3) ラジオが 55.5 % となっており, 特別なアクセス方法を必要としない情報ソースに依存する傾向が顕著である. これに対し, 点字の習得 (全盲) や文字の拡大 (弱視) を前提とする図書, 新聞, 雑誌と答えた人々は 25.9 % にとどまっている. 現状では, 点字の技能を持つ視覚障害者と持たない視覚障害者の情報格差は大きい.

点字に関して本論文の著者はいくつかの体験をもった. たとえば, 駅構内に設置されている手すり上の点字案内を実際に調査する機会があったが, 間違った点字が設置されている (打ってある) 場合がとても多いことに驚かされた. 2000 年に鉄道の点字運賃表の間違いが相次いで発見されたにもかかわらず, その後も点字運賃表の誤りは発見され続けている. このことは, 現場の点字確認作業がかなり困難であることを示していると考えられる.

本論文の著者は大学で点字サークルに所属し, 視覚障害者に関する多くの事実を知ることになったが, 自分がまずできることの一つとして点字翻訳を学び, いくつかの点訳を試みた. 点訳作業は点訳者の熟練, 意志, そして忍耐を必要とする. 特に一冊の書籍, 大量の重要書類などの点訳作業をスムーズに行うには多大の点訳訓練が必要であり, 需要に対して点訳者の数は圧倒的に少ないのが現状である. したがっていわゆるベストセラーとなった書籍は比較的速やかに点訳されるものの, 一般にはまだまだ読みたい本が点訳されている率は低いと言わざるを得ず, 点訳本, 点訳者の増加は視覚障害者の切望するところとなっている.

コンピュータによる自動点訳の試みはいくつかある. そのような自動点訳は, もとの媒体から電子化された情報に基づき, いわゆる「分かち書き」などを自動的に行おうとするものであり, 勿論有意義なものではある. しかし, 高性能自動点訳システムがあれば視覚障害者の点訳問題が解決するわけではない. そのような点訳システムを使って点訳をする人間が必要だからである. 少なくとも現状では, 点訳はボランティアに依存している部分が多く, そのような人は, 目が悪い人々に関心がある人間である. 従って視覚障害者に関心のある人間を増やすことは自動点訳を行うことと同じくらい大切と考えられる.

筆者が調べたいいくつかの WEB ブラウザでは, ウェブサイトのデザインやレイアウトが少し変更されただけでも, 特殊 WEB ブラウザは機能しなくなる可能性が高く, また目的とするサイト以外のサイトへのアクセス能力は, 全くないかきわめて限られている. プログラミング言語を用いれば, 視覚障害者用の特殊 WEB ブラウザを作成

4 第1章 序論

するのはさほど難しくない。しかしながら、ウェブサイトの小さなデザイン変更をすべて把握し、WEB ブラウザ・プログラムの修正に反映しなければならない。したがって視覚障害者用の表計算プログラムのようなより有効な WEB ブラウザが必要であると考えられる。本論文の著者はいくつかの視覚障害者のための特別な WEB ブラウザに触れる機会があったが、これからの情報化社会で活躍するためにも改善の余地があると常々感じていた。

自らの立ち位置で上述のような問題を解決する手法は、大学で身に着けた情報技術の知見を通して構築するのが自然ではないかと思い、上述の3つのカテゴリから

1. 点字デジタルカメラ画像の自動誤り訂正システム
2. コーパスへの検索による点訳の分かち書き誤り修正システム
3. 視覚障害者のための特殊 WEB ブラウザ

をとりあげ視覚障害者の生活の質向上にささやかではあっても貢献したいと考えるに至った。本論文は、「視覚障害者のための情報支援システム」を、上述の3つの側面から取り組み、その成果を学術雑誌、国内外の学会などで発表した論文を踏まえて博士論文としてまとめたものである。

1.2 本研究の目的

この研究では視覚障害者のための情報支援システムとして上記 (1), (2), (3) の3つを取り上げた。本研究の目的は、これら各々に対するアルゴリズムを構築したうえ、それを実装してシステムの性能を検証することである。具体的には次の3つの課題である：

1. 点字デジタルカメラ画像の自動誤り訂正システム

デジタルカメラによって撮影された点字を認識し、普通の文字（墨字）で表示するためのプロトタイプシステムを提案し、それを評価する。今までに実用化された点字認識システムは、スキャナを使ったものに限られており、駅などに設置されている大量の手すり点字プレートを、スキャナでは読み込めないため、この確認作業はすべて人手に頼っている。提案システムはこの確認作業を補助することを目的としている。カメラの収差を「点字文字ピッチの局所的な伸縮」としてとらえ、動的計画法によるマッチングを行うことによって点字の

正確な位置を推定する。

2. コーパスへの検索による点訳の分かち書き誤り修正システム

点訳における分かち書きの修正を支援するシステムを提案する。提案システムでは、修正対象とする点訳ファイルのある一部分を取り出し、その一部分が大量の点訳ファイル(点訳コーパスと呼ぶ)の中に何件ヒットするかを調べる。そして、そのヒット件数から、分かち書きの誤りを判別し、修正する。

3. 視覚障害者のための特殊 WEB ブラウザの構築

たとえば新しいソフトウェアの使い方を学ぶ場合、目の見えるユーザは、マニュアルを参照しなくても、メニューバー、アイコン等から選択することで、そのソフトウェアを簡単に使用できる。より複雑な操作の場合でも、目の見えるユーザは、マニュアルを参照しながら選択することができる。一方、多くの場合、視覚障害者は、マニュアルを読んで完全に暗記するまではそのソフトウェアを使用できない。使いやすいソフトウェアを構築できれば、視覚障害者の情報リテラシーの向上にきわめて有効であろう。そのような視点から特殊 WEB ブラウザの構築を試みる。

1.3 本論文の構成

第2章では「点字デジタルカメラ画像の自動誤り訂正アルゴリズムとその実装」、第3章では「コーパスへの検索による点訳の分かち書き誤り修正システム」、そして第4章では「視覚障害者のための特殊 WEB ブラウザの構築」について論じる。第5章は全体のまとめとこれからの方向性を示している。

1.4 先行研究

ここでは本論文で取り上げる課題に関する先行研究を概観する。

1.4.1 点字デジタルカメラ画像の自動誤り訂正アルゴリズムとその実装に関する先行研究

これまで、点字読み取りシステムについて、さまざまなシステムが提案されている。読み取り専用ハードウェアを使うもの [5][6][7]、デジタルカメラを使うもの [8][9]、イ

メージスキャナを使うもの [10][11]，フィルム状圧力センサを使うもの [19]，デジタルビデオカメラを使うもの [15]，携帯電話内蔵カメラを使うもの [18] が提案されている．この他に 1993 年よりスキャナによる点字読み取りソフトが市販されている．いずれも，点字の手すりプレートを簡単に読めるものではない．[19] はセンサ部分で一定速度で点字をなぞる必要がある．[15][18] はデジタルカメラによって認識するもので，大変に興味深い．しかし両方とも数文字程度の点字での実験がなされているが，それより長い点字が読めるかどうかには言及されていない．一方，点字ではなく，墨字（普通の文字）の文字認識の分野において，スキャナではなく，カメラによって認識を行う研究が盛んである．それらはカメラで撮影した画像をスキャナで取り込んだ画像のように変換する技術を元に行っている．本研究においても，デジタルカメラによる文字認識の技術を参考にしている [14][16][17]．

1.4.2 コーパスへの検索による点訳の分かち書き誤り修正システムに関する先行研究

本論文は，点訳初心者訓練用分かち書き自動修正システムの提案であり，本論文の著者が知る範囲では同一目的の先行研究はないと思われる．一方関連する研究テーマとして自動点訳があり，さまざまな提案がなされている．以下にその一部をあげる．「エクストラ」[20] は市販ソフトである．石川准によって開発された．「Ibuki-ten」，「Ibuki-tenC」[21] は形態素解析エンジン Ibuki を使用した自動点訳エンジンである．「お点ちゃん」[22] は勝沼貞幸によるフリーソフトである．また，文献 [23] も，直接本論文とは関係しないが，デジカメで撮影した点字の誤り自動検出手法を提案しており，情報技術を使って，視覚障害者を支援している，という意味で共通のスタンスに立つものである．

1.4.3 視覚障害者のための特殊 WEB ブラウザの構築に関する先行研究

特殊 WEB ブラウザとは，ある特定のウェブサイトのみの特化して設計され，当該ウェブサイト用の非常に快適なユーザ・インターフェースを提供する WEB ブラウザである．しかしながら，汎用 WEB ブラウザとは異なり [38]，特殊 WEB ブラウザでは，その設計目的となった特定サイト以外のサイトにはアクセスできない．

日本にも視覚障害者用の特殊 WEB ブラウザの例がいくつかあり,「NAIIV net」(点字図書館の団体が管理する,点字書籍をダウンロードできるサイト)[32],ニュースサイト [34][33][35],インターネットラジオ局 [36],電車時刻表 [37] にアクセスするためのものなどがある.これらの WEB ブラウザの中には,販売されているものも無料で入手できるものもある.

目の見える人用の特殊 WEB ブラウザもあるとはいえ(例えば,YouTube にアクセスできる WEB ブラウザがある),特殊 WEB ブラウザは視覚障害者にとってはるかにより有効である.

第2章

点字デジタルカメラ画像の自動誤り訂正システム

2.1 はじめに

2.1.1 動機

この課題を開始するきっかけとなったのは、「ステンレス手すりに打たれた点字を認識するシステムが製作できないか」という鉄道会社からの依頼である。視覚障害者が使用する文字である点字の認識については、すでにスキャナを使って紙に打たれた点字を認識するシステムは提案されている。しかし、スキャナでは手すりを読み取ることは難しい。システムの検討にあたり、実際に駅において調査を行ったところ、間違っただ点字が設置されている（打ってある）場合がとても多いことに驚かされた。例えば、2000年に鉄道の点字運賃表の間違いが相次いで発見されたにもかかわらず、その後も点字運賃表の誤りは発見され続けている（表 2.1）。このことは、現場の点字確認作業がかなり困難であることを示していると考えられる。

手すりに設置された（打たれた）点字も、間違えているものが見つまっている。また、最初に設置されたときは正しくても、駅舎の改良工事などで位置関係が変化し、点字はそのままであるために間違えているケースもある。図 2.1 は間違えている手すりの例である。上の点字プレートは正しいが、下の点字プレートは上下逆に設置されている。東京駅の新幹線中央乗り換え口付近に以前存在したが、指摘により、現在は修正されている。また、図 2.2 は JR 錦糸町駅の点字手すりプレートである。「下り

表 2.1. 発表された点字運賃表の誤り

時期	鉄道会社 (略称)
2000 年	JR 6 社, 東武, 小田急, 都営, 横浜市交通局, 名鉄, 近鉄, 南海, 大阪市交通局, 北大阪急行
2004 年	JR 北海道, 名古屋市交通局
2007 年	西武
2008 年	JR 九州

専用階段」と書かれてあるが、現在は改良工事により、「下り専用」ではなくなっている。



図 2.1. 不適切な手すり点字の例 1

これらの問題の本質は、

「点字を人が確認する作業の困難さ」

にあると考えられる。50 音表と見比べながら読めば、まったく読めないわけではないであろうが、膨大な時間を費やすことになる。ここでの目的は、このような困難の改善をめざして、デジタルカメラによって撮影された点字を自動的に認識し、普通の文字(墨字)で表示するためのプロトタイプシステムを提案し、評価することである。



図 2.2. 不適切な手すり点字の例 2

デジタルカメラで撮影したステンレス製の点字を認識する場合、以下の2点のため、イメージスキャナを使った点字認識のアルゴリズムをそのまま適用できない。

- 1点目は、点の検出問題である

スキャナを使った従来手法の点字認識システムにおいては、点字の点に対し、斜め上から光をあて、立体の陰影を検出する。具体的には、図 2.3 のような点の画像に対し、明るい部分の下方に暗い部分が存在することを利用し、凸の点の存在を認識する。

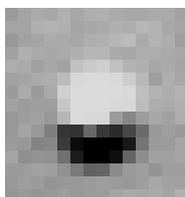


図 2.3. スキャナを使った従来の点字認識システムにより採取した点字画像

これに対し、デジタルカメラによって採取したステンレス手すりの点の画像を図 2.4 に示す。光線条件により、陰影のパターンは一定しないため、この画像に対して、スキャナで用いられた方法の適用は難しい。

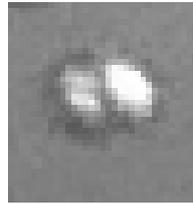


図 2.4. デジタルカメラにより採取した点字画像

- 2 点目は，点の位置検出の問題である．

スキャナを使った点字認識システムにおいては，前述の方法によって点の存在を検出し，2 値化した後，図 2.5 のように， x 軸， y 軸上への射影を使って点字の位置を検出している．これに対し，デジタルカメラを使って採取したステ

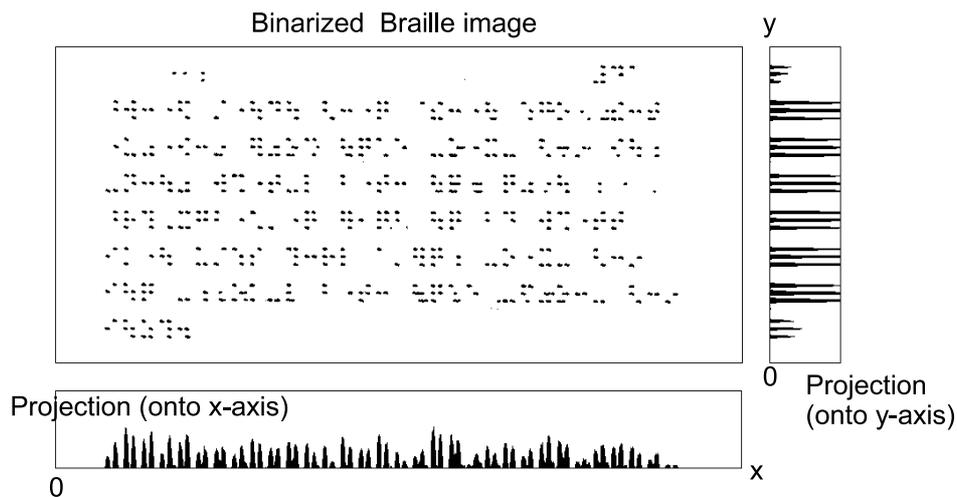


図 2.5. スキャナを使った従来の点字認識システムにおける，点の位置の検出

ンレス手すりの点字の画像を図 2.6 に示す．点字の量が少ないこと，レンズの収差があることなどから，点の位置検出についてもスキャナで用いられた方法は適用できない．



図 2.6. デジタルカメラで撮影した点字の画像

提案するシステムは、次のような方法でこれらの問題を解決した。点の存在の検出については、ストロボを発光させて撮影し、ストロボの光が点に反射したパターンを認識した。また、点の位置検出については、動的計画法を用いたマッチングを行うことにより、レンズの収差を吸収した。現実の問題と取り組む場合、考え方としては比較的単純であっても、現場から得られるデータには多くの困難が内包されていることが多い。本研究も例外ではなく、対象としての”手すりに設置された点字”であることによる特殊な困難がいくつか付随しており、本研究はそのような困難に挑戦しようとするものである。

なお、この研究はステンレス製のプレートを主な対象としている。手すりに点字を設置する手段として、ステンレス製点字プレートのほかに、樹脂、アルミニウムや亜鉛で作られた点字プレートやシールも存在し、それぞれイニシャルコスト、設置のしやすさ、耐久性などに一長一短があるが、それらについては、別稿にゆずる。

2.1.2 対象とする点字

わが国で一般的に使われている6点式点字を対象とする。他言語の点字においても、一般的に使われているものは、ほとんどが6点式である。6点式点字とは、図2.7で示される、6つの凸となっている点により、1文字を表すものである。6つの点によって点字1文字を構成すると、63通りの文字が表現される。一般的な点字では漢字は表すことができないので、カナで読み下す。カナだけで続けて書かれたものは読みにくいので、必要な場所で空白を入れ、読みやすくする。清音は点字1文字で1音を表すが、濁音、半濁音、拗音など点字2文字で1音を構成するものもある。「1」と「A」と「あ」は点字では同じ文字によって表されるが、数符の後に続く場合は「1」、外字符の後に続く場合は「A」、それ以外は「あ」をそれぞれ表す。

2.2 提案システム

本システムでは、図2.8の流れで認識を行っている。

2.2.1 点字部分の切り出し

点字手すりプレートの認識については、システムの利用者が目で見、大まかな回転の補正と切り出し範囲の指定をマウスのクリックとドラッグにより行っている。こ

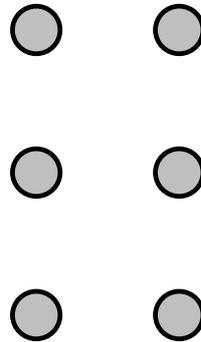


図 2.7. 6 点式点字の構成

これは鉄道会社の社員が使うことを想定しているためである。

2.2.2 点の存在する可能性を表す指標の計算

カメラによる画像採取に付随する困難を説明するため、まずスキャナによる先行研究でしばしば用いられている方法を説明する。従来の点字認識システムのうち、スキャナを用いるものでは、斜めから光を当てた画像をスキャンし、明るい部分、暗い部分、平均的な部分の3つに分類し、明るい部分の下に暗い部分がある位置を検出することにより、2値化を行うことが一般的である。一方、本システムで用いるカメラによる認識では、光源の位置は必ずしも斜め上からとは限らないので、別な方法が必要である。これまでに提案されたシステムのうち、[15][18]では、フィルタ処理の後、最終的に輝度によって2値化しているが、この方法の場合認識できる点字の材質がかなり限られてしまう。ステンレスの点の認識では、カメラでストロボを発光させて点字を撮影すると、ストロボの光が反射して、点の中心部の輝度が高くなる。しかし、点の中心から離れた部分は、輝度は比較的低い。点の中心から離れた部分は、直接反射しないからである。ここで問題となるのは、ストロボを発光させると、点の中心だけでなく、面全体も反射し、画面の中心が白く“とんで”しまうことである。白く“とんだ”部分のコントラストは低下し、点が検出しにくくなる。これについては白くなった部分を検出し、しきい値を変更することにより対応している。また、詳細は後述するが、マッチングにより画面全体を見て点字が存在する可能性を表す指標の総和が最大となる位置を計算しているので、画面の一部が白く“とんで”しまったり、空白があっても比較的正確に点を検出できる。具体的な手順は以下のとおりである。

画像上任意のピクセル $A = (x, y)$ を端点とし、水平からの角が θ となる半直線上

のピクセルのうち， A から順に半直線上をたどり，外側の隣り合うピクセルとの輝度の差分が初めて正となる点を $B(\theta)$ とする．ただし， $B(\theta)$ があまりにも A に近い場合は，これを無視し，輝度の差分が負となってから次に初めて正となる点を $B(\theta)$ として採用する．画像上における $A, B(\theta)$ の位置関係の一例を図 2.9 に，直線 $AB(\theta)$ 上の輝度分布の一例を図 2.10 に示す

また， $e(A; \theta)$ をピクセル A とピクセル $B(\theta)$ の輝度の差， $s(A; \theta)$ をピクセル A とピクセル $B(\theta)$ の間の距離とし，

$$E(A) = \int_{-\pi}^{\pi} e(A; \theta) s(A; \theta) d\theta \quad (2.1)$$

を定義する．これは，ピクセル A を中心とした（点字の）点が存在する可能性を表す指標と考えられる．実際の実装場面では，(2.1) を離散化する．

図 2.6 の入力画像に対して上記 $E(A)$ を適当なしきい値処理して 2 値化した後プロットしたものが図 2.11 である．

2.2.3 射影ひずみの補正

一般の文字（墨字）へのデジタルカメラによる文字認識においては，射影によるひずみの補正が課題のひとつとなっている．[14] は紙と背景との境界線を求め，その境界線が長方形となるように射影変換をおこなっている．[16] は文字列全体に対し，平行四辺形を長方形にする変換と，等脚台形を長方形にする変換をおこなっている．これらを参考にし，提案手法は，以下のように補正をおこなう．

点字の文字列は必ず横書きで，点字文字列を構成する点字 1 文字は点が縦に 3 点，横に 2 点の格子状に並んでいる．そのため，一行の点字文字列全体をみると，すべての点は横向きの 3 本の直線のいずれかの上に存在する（図 2.12 a）．点字画像から，この横向きの 3 本の直線を検出する．検出した線より，まず中央の線が水平になるように画像の回転をおこなう．次に，上の線と下の線の 2 本が水平となるように射影変換をおこなう（図 2.12 b）．3 本の横向きの直線を検出には，ある直線上にあるピクセルの $E(A)$ の総和を求める．その総和が大きい順番に 3 本を選ぶ．ただし，2 番目以降の線を選ぶ際には，すでに選んだ直線の近傍であるかどうか判定し，近傍であれば，これを棄却する．

これらの補正を行った後の点字はそれぞれ傾斜している．この傾斜を補正するため，

点字文字列全体について，点字文字列を内包する台形を，高さの等しい長方形にする変換を行う。(図 2.12 c,d)

点字 1 文字毎ではなく，点字文字列全体をまとめて補正する理由は，点字は 1 文字が 1 点から構成されるものも存在するため，そのような点字は単独では全く補正できないからである。

台形を高さの等しい長方形にする変換については，変換後の x 軸上への射影に注目して，変換を行っている。具体的には，あるパラメータで台形を長方形にする射影変換を行った後， $E(A)$ の x 軸上への射影を求め，射影の 2 乗の総和をとり，総和が最大となるパラメータを採用し，変形する。パラメータは台形の左上端の水平移動量 $dx1$ と，右上端の水平移動量 $dx2$ の 2 つである。点字文字列を内包する台形は等脚台形とは限らないため，必ずしも $dx1 = dx2$ ではない。

2.2.4 点字の大きさ，ピッチの解析

検出すべき点字文字ピッチを W ，点字 1 文字中の x 方向の点ピッチを w ，点字 1 文字中の y 方向の点ピッチを h とする (図 2.13)。従来のスキャナを使う方法の点字認識では，点字の大きさを検出するために，射影の自己相関を解析している。具体的には，前処理後の点字 1 ページ全体の画像から， x 軸， y 軸上への射影を求め，それをもとにして，もっともよくあてはまる，点字文字ピッチ，点字の行ピッチ，点字 1 文字の中の点と点の縦のピッチ，点と点の横のピッチを求めている。点字 1 ページの点字文字数，行数の典型的な値は横 32 文字，縦 17 行である。

それに比べ，本システムにおいては，認識対象となる点字が 1 行のみ，と対象となる点字の量が少ない。また，点字の行が糸巻型，または樽型の収差によってカーブしている場合があることなどから， w, h については射影を用いた方法ではうまくピッチが求められないことが多い。そのため，以下のような方法を用いている。

- step 1

x 方向 $0 - X$ ピクセル， y 方向 $0 - Y$ ピクセルからなる対象画像が与えられた時，点字 1 文字中の x ，及び y 方向の点ピッチ w, h は次の関数の最大化で計算する：

$$F(w, h) = \sum_{x=0}^{X-w} \sum_{y=0}^{Y-2h} f(x, y; w, h) \quad (2.2)$$

$$\begin{aligned} f(x, y; w, h) = & c(x, y) \times \{c(x+w, y) \\ & + c(x, y+h) + c(x+w, y+h) \\ & + c(x, y+2h) + c(x+w, y+2h)\} \end{aligned} \quad (2.3)$$

ここに $c(x, y)$ は点 (x, y) における輝度値である。関数 $F(w, h)$ は、ある点と、その点から x 方向に w 、 y 方向に h だけ離れた点の輝度を掛け、それらの一画面分の総和をとっている。更に、 y 方向に $2h$ 離れた点の輝度も考慮している。点字文字列中のある縦の3点を見たとき、 $2h$ を考慮せず、 h だけを考慮すると、上と下の点のみが存在し、中央の点が存在しない場合、評価関数 F の値が0になってしまうからである。 $F(w, h)$ の計算には、画面上のすべてのピクセルについて f を計算している。ただし、画面下端から $2h$ 、または画面右端から w の範囲においては、 f は定義できないため、計算を行っていない。式(2.3)には、収差は考慮されていないが、以下の理由で収差は大きな問題とはならないと考えられる。 $c(x, y)$ の値が高いピクセルは、点の中心をピークとして、ある程度の幅をもって分布している。この幅が存在することにより、 W や w に対応する画面上の点と点の距離が収差によって多少の変動があったとしても、 $f(x, y; w, h)$ の値は、急激には変化しない。そのため、点字の大きさ、点字文字ピッチの解析で大きな誤差は発生しないと考えられる。一方、小さな誤差は後述の処理(2.5 点候補位置の推定)によって吸収することができる。ここで水平方向の W, w について述べたが、垂直方向の h についても同様である。

- step 2

点字文字ピッチ W の計算。任意の点 $A = (x, y)$ の x 軸上への射影を $P(x)$ とし

$$R(W) = \sum_{x=0}^{X-W} P(x) \times P(x+W) \quad (2.4)$$

を最大化する W を図2.13の W とする。ここでは経験的に、 W と w の比はある範囲に限られることを使っている。図2.14最上段が $E(A)$ の例、そし

て2番目が x 軸上への射影 $P(x)$ である。点字文字ピッチである W と点字1文字内の点のピッチである w とで計算方法を変えている理由は、ちょうど点字1文字分ずれる W のほうがより評価関数の最大値が明確に出やすいためである。

- step 3

step 1 で計算した w が実は図 2.13 の $(W - w)$ になっている可能性は否定できないであろう。step 2 で計算した W をもとに w を更新する。具体的には、 $w \leq W/2$ の条件を加え、step 1 に戻り再度計算する。 $w \leq W/2$ の条件を加えることにより、 w と $(W - w)$ を混同する可能性が相当下がる。step 1 再計算後は step 2 以後の再処理は行わない。

2.2.5 点候補位置の推定

それぞれの点の、より正確な位置を推定する。図 2.14 最下段は、 x 軸上への射影 $P(x)$ が理想的な場合である。ここでいう理想的とは、射影のピークが等間隔に並んでいることをいう。画像から得られた射影が、このような等間隔であれば点の位置の推定は簡単である。しかし、現実の場面では、カメラで撮影した画像には、収差が存在するため、点字文字ピッチはすべて同じではない。この収差による点字文字ピッチの変動を吸収するため、1次元の動的計画法を用い、理想的な射影と実際の射影とのマッチングを行うことにより、正確な位置を推定する。カメラの収差を取り除くためには、あらかじめ格子などの参照物体を撮影し、カメラのパラメータを決定する、カメラキャリブレーションを行う方法もあるが、操作が複雑になることがある。

N 個の点字から構成される点字文字列について、最適なそれぞれの点字の x 方向の位置を、 x 軸上への射影から求める問題について考える。

点字のピッチが、局所的に伸びたとしても、それを吸収するため、点字の位置にある射影の総和が最大になるようにする。点字文字ピッチを W 、点字内の点と点のピッチを w とする。

画像端から点字文字列の1文字目開始位置までの余白量を Q とする。最初に i 番目の点字の x 方向の開始位置が0となるように、画像全体を x の負方向に Q だけ平行移動させる。もし点字のピッチがすべて等しければ、 i 番目の点字の開始位置は単純に $W \times (i - 1)$ で表される。しかし、実データ上での点字文字列のピッチは等しく

ないことが多い．これを吸収するため，パラメータ U_i を導入し， i 番目の点字の開始位置を $W \times (i - 1) + U_i$ で表す． $i - 1$ 番目と i 番目の点字の点字文字ピッチは $W + U_i - U_{i-1}$ で表され，次のような制限を設ける．

$$0 \leq U_i - U_{i-1} \leq 2$$

これは，点字文字ピッチに，ある決まった値を基準に 0 ピクセルから 2 ピクセルまでの伸びに対応する．

U_i が負となる場合，つまり点字文字ピッチが局所的に縮むことも本来考えられるが，計算を簡単にするために，縮む場合について計算しない代わりに，点字文字ピッチが 1 ピクセル伸びる場合と 2 ピクセル伸びる場合について計算し，もともとの点字文字ピッチである W の値を， W そのままと， W から 1 減じたものの 2 通り計算している．

まず $P(x)$ が x 軸上への射影であることを思い出す．ひとつの点字は 2 つの列から成り立っていることに注意すれば， i 番目の点字が存在する位置の可能性を表す指標として，

$$\begin{aligned} d_i(U_i) &= P(W \times (i - 1) + U_i) \\ &\quad + P(W \times (i - 1) + U_i + w) \end{aligned} \quad (2.5)$$

が考えられる．したがって N 個の点字文字列全体が存在する位置の可能性を示す指標として

$$F = \sum_{i=1}^N d_i(U_i) \quad (2.6)$$

を考え，これを最大化する．

$$\underset{(U_1, \dots, U_N)}{\text{maximize}} F = \sum_{i=1}^N d_i(U_i) \quad (2.7)$$

$$\text{subject to } 0 \leq U_i - U_{i-1} \leq 2 \quad (2.8)$$

これは動的計画法 (Dynamic Programming[31]) により解くことができる 初期条

件を

$$g_1(0) = d_1(0) \quad (2.9)$$

とすると Dynamic Programming 漸化式は

$$g_i(j) = \max \begin{pmatrix} g_{i-1}(j-2) \\ g_{i-1}(j-1) \\ g_{i-1}(j) \end{pmatrix} + d_i(j) \quad (2.10)$$

for all $i = 1, \dots, N, j = 0, \dots, 2i$

である．最適化された i 番目の点字の右方向へのずれを $BestU_i$ と定義すると，終了状態での $BestU_N$ は

$$BestU_N = \operatorname{argmax}_{(0 \leq j \leq 2N)} g_N(j) \quad (2.11)$$

$\operatorname{argmax}_{(0 \leq j \leq 2N)} g_N(j)$ は $g_N(j)$ が max となる j

である．この計算の例を図 2.15 に示す．太線で示されたのは，それぞれの i に対する， $BestU_i$ である．

最初に余白量 Q だけ x の負方向に平行移動させているので，ここで求めた射影の総和の最大値 $g_N(BestU_N)$ は Q の関数である． $g_N(BestU_N)$ を最大化するような Q を求め，その Q を最適な余白量 $BestQ$ とする．余白量を $BestQ$ とした条件下でのバックトレースにより $BestU_i$ を確定させる．

求めた $BestU_i$ と $BestQ$ より，余白量の平行移動を行う前の画像における， i 番目の点字の x 方向の開始位置は $BestQ + W \times (i - 1) + BestU_i$ と表される．

点字の y 方向の開始位置は， y 軸上への射影と，点ピッチ h により容易に求められる．(図 2.16)

2.2.6 点検出

上述の方法で計算した点 $A = (x, y)$ に対して式 (2.1) で定義された $E(A)$ の値が，あるしきい値以上であれば，点が存在すると判定する．このしきい値は，画像が白く“とんで”いる場合でも正しく検出できるように，点 A 周辺の輝度の平均により，変化させている．

2.2.7 墨字への翻訳

最終的には日本点字表記法 [26] の規則に従い，検出した点から点字でない普通の文字 (墨字) への翻訳をおこない，結果を表示する．正しく検出されたかどうかを使用

者が目視で確認するため，入力画像と，検出した点字の点もあわせて表示する．

2.3 実験

2.3.1 実験1 わい曲収差のある画像に対する認識

点字手すりプレート画像データの採取

Panasonic 製のデジタルカメラである DMC-FZ5 の内蔵ズームレンズを最も広角側にしたとき，同じ長さの物体を撮影しても，レンズのわい曲収差により，画面の中心付近に写ったもののほうが，左右の端に写ったものよりも 5% 長い．このときの焦点距離 (35mm 換算) は 36mm である．

この状態のカメラで撮影された点字を認識する実験を行った．あわせて変動吸収を行わない設定で認識をおこない，変動吸収の有効性を確認した．カメラの位置は，点字の鉛直上方より，手前，または奥の方向に 30 度以内の傾斜で，左右には傾斜していない．

サンプルは JR 千葉駅にある，すべての点字手すりプレート 104 枚をデジタルカメラで撮影した．手すりの材質はすべてステンレスであった．

デジタルカメラの機種，スペック，撮影条件は以下のとおりである．

Panasonic 製	DMC-FZ5
カメラ有効画素数	500 万画素
焦点距離 (35mm 換算)	36mm
撮影距離	約 0.35m

手すりプレート 104 枚のうち，点字が 2 行にわたっているものもあり，合計 196 行に対応する画像を得た．点字文字列中の空白も含めた点字文字数をマス数と定義すると，それぞれの行は最小 14 マス，最大 40 マスの点字で構成されていた．

撮影にあたって，カメラ内蔵のストロボを強制発光させた．日光が当たっている手すりの撮影については，内側を黒く塗った箱で覆い，外光の影響を受けないようにした．しかし，完全に覆ってしまうと，カメラの自動焦点機能が動作しにくくなるので，多少の光が入るようにした．地下にある階段など，薄暗い場所においては，同じ理由で箱を使わずに，そのまま撮影した．点字を箱で覆って撮影している場面を図 2.17 に示す．

点字手すりプレートの認識

上述の 196 行のサンプルについて認識をおこなった結果を，表 2.2 に示す．なお，認識時間は Intel Core2 Quad 2.4GHz を使用したパソコンにおいて，すべて 30 秒以内である．比較のため，動的計画法による点字文字ピッチの変動吸収を行わない認識実験もあわせて行った．

表 2.2. 認識結果 (わい曲収差のあるサンプル)

	全マス数	正しく認識したマス数	認識率 %
変動吸収あり	5102	5037	98.73 %
変動吸収なし	5102	4145	81.24 %

ここで，認識率は以下のように定義している．

$$\text{認識率} = \frac{\text{正しく認識したマス数}}{\text{全マス数}}$$

通常の文字認識においては，空白を他の文字と間違える可能性は低いと考えられるので，計算には含まないことが多い．

一方，点字は語と語の間に空白を空ける，分かち書きと呼ばれる書き方である．その空白が空白として認識されていることも計算に入れる必要があるため上述の認識率で評価した．

また，点字 1 マスは 6 点で構成されているが，そのうちの 1 点でも間違っている場合，1 マスの間違いとした．

点字文字ピッチの変動吸収を行わなかった場合の，点字位置の誤認識例を図 2.18 に示す．変動吸収をおこなった場合は，すべて正しく認識した．変動吸収を行わなかった場合，1 マス目から 10 マス目までは正しく認識したが，11 マス目以降は誤って認識した．これは，レンズのわい曲収差により，サンプルの点字文字列 (図 2.18 a) では，右側に近くなるほど，点字文字ピッチが少しずつ長くなっている．変動吸収を行わなかった場合の誤認識は，この点字文字ピッチの変動を吸収できなかったことが原因であると考えられる．

2.3.2 実験2 カメラ位置が傾斜した場合の認識

点字手すりプレート画像データの採取

点字の鉛直上方より，左方向に 30 度傾斜した位置から撮影し，認識をおこなった。手前，または奥の方向の傾斜については，実験 1 と同じく 30 度以内である。左方向への傾斜と，手前，または奥の方向の傾斜については，複合して起こりうる。比較のため，左方向へ傾斜していないカメラ位置からも撮影し，あわせて認識をおこなった。

Panasonic 製	DMC-FZ38
カメラ有効画素数	1210 万画素
記録画素数	500 万画素
焦点距離 (35mm 換算)	45mm
撮影距離	約 0.4m

これ以外の撮影場所，条件，サンプルの数は実験 1 と同じである。

左方向への傾斜角 0 度，30 度のそれぞれのカメラ位置から撮影された点字手すりプレートのサンプルと，その認識結果の一例を図 2.19 に示す。認識結果のひらがなは，点字を翻訳したものである。

点字手すりプレートの認識

実験 1 と同じく，196 行のサンプルについて認識を行った結果を，表 2.3 に示す。認識に要した時間は，実験 1 と同じくすべて 30 秒以内である。

表 2.3. 認識結果 (カメラ位置が左に傾斜した場合の認識)

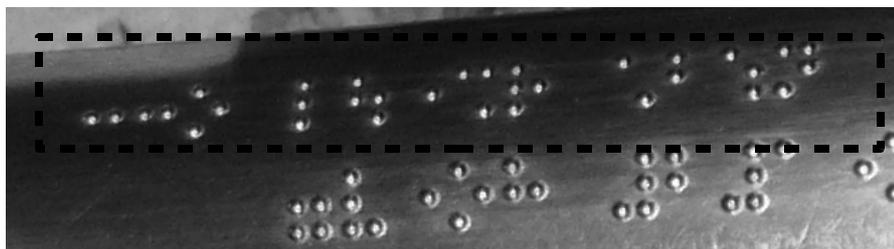
左方向への			
傾斜角	全マス数	正しく認識したマス数	認識率 %
0 度	5102	5070	99.37 %
30 度	5102	5051	99.00 %

左方向への傾斜角 0 度の場合は，撮影した画像の中心付近がストロボの光の反射により白く“とんで”しまうが，傾斜角 30 度では，白く“とんで”しまう部分は撮影した

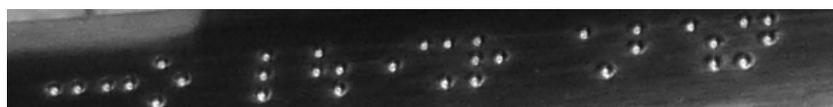
画像の中には存在しない。(図 2.19 a,b) 傾斜角 0 度と、傾斜角 30 度の認識結果の間には、射影ひずみの影響の他に、これによる影響があると考えられる。

2.4 まとめ

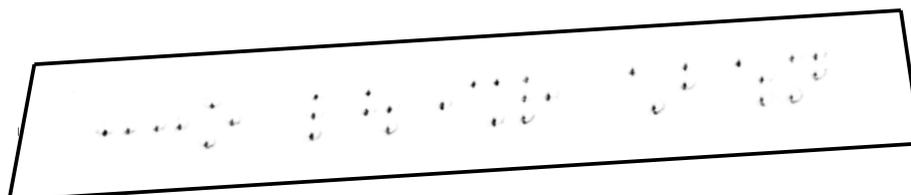
デジタルカメラによる点字手すりの認識システムを提案し、評価を行った。本システムでは、JR 東日本をはじめ多くの駅で使用されている、比較的長い点字文字数のステンレス製の点字手すりを高精度で認識することを目指した。これにより、従来、人の目に頼っていた点字手すりプレートの確認作業を高速かつ簡単に行える可能性を示せたと考える。今後は、鉄道会社や実際に使用される方の意見も多く取り入れ、認識率だけでなく、操作性にも十分配慮した使いやすいシステムになるように、さらなる改良を加えていきたいと思っている。



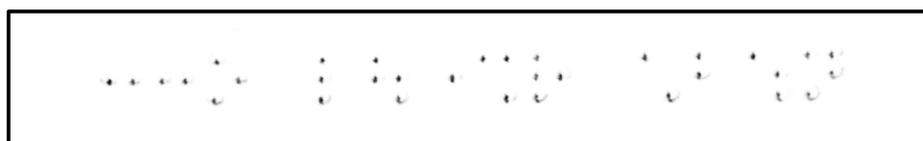
↓ (2.1) 点字部分の切り出し



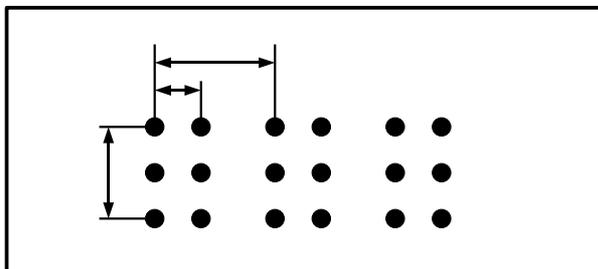
↓ (2.2) 点の存在する可能性を表す指標の計算



↓ (2.3) 射影ひずみの補正



(2.4) ピッチの検出



(2.5) 点候補位置の推定

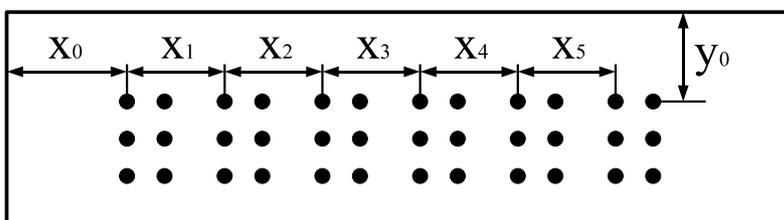


図 2.8. 認識処理の流れ

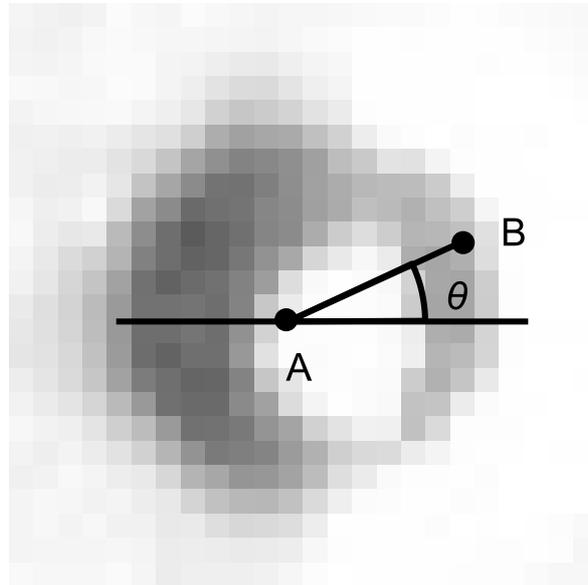


図 2.9. 点の存在する可能性を表す指標の計算

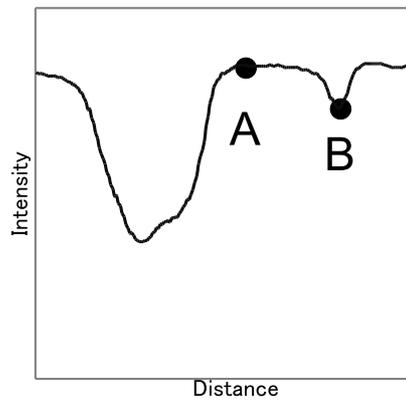


図 2.10. 直線 AB 上の輝度分布の一例

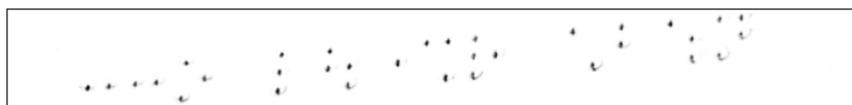


図 2.11. 2 値化後の $E(A)$ の例 .

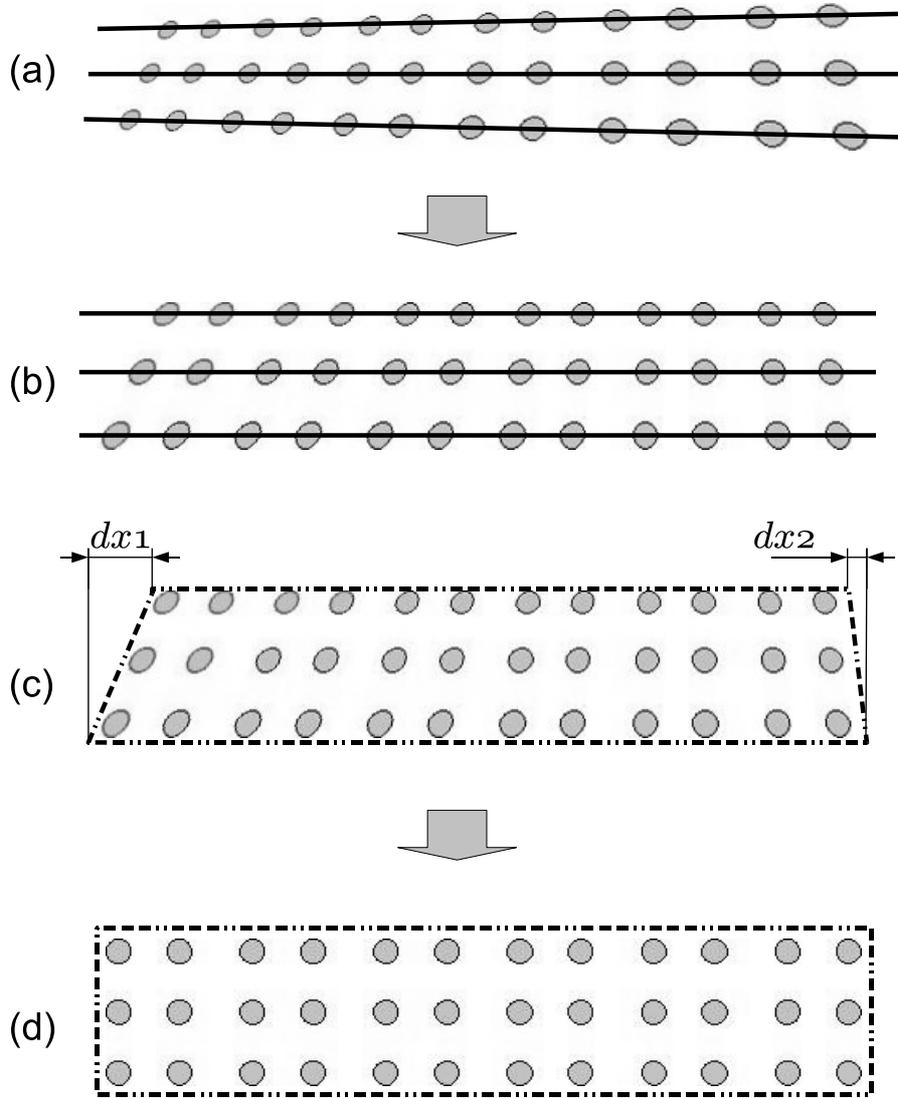


図 2.12. 射影ひずみの補正の説明図

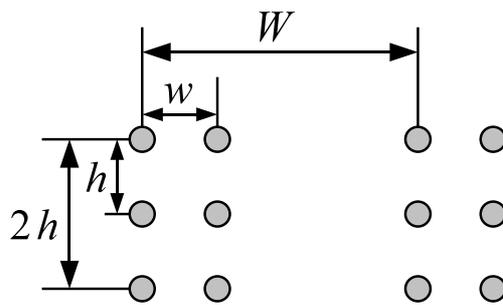


図 2.13. 位置関係

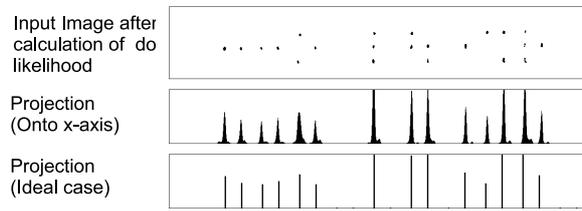


図 2.14. 入力画像と x 軸上への射影の例

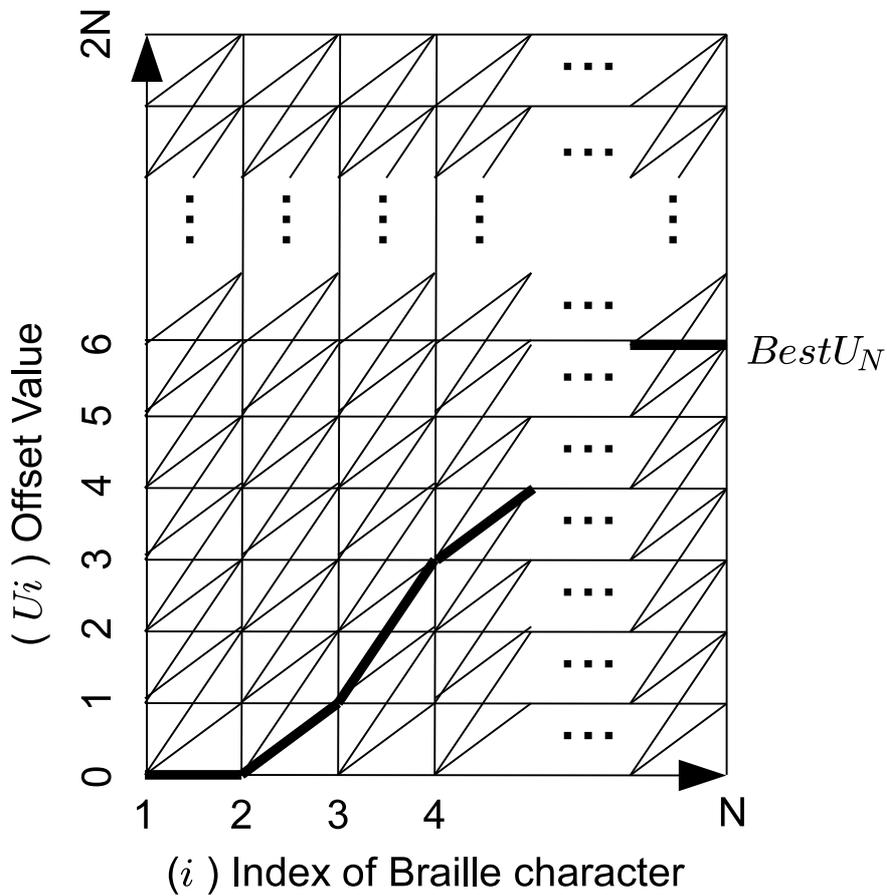


図 2.15. 動的計画法による $BestU_i$ の計算

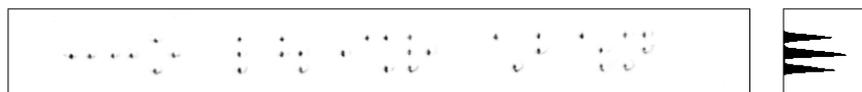


図 2.16. 2 値化された $E(A)$ と, y 軸上への射影

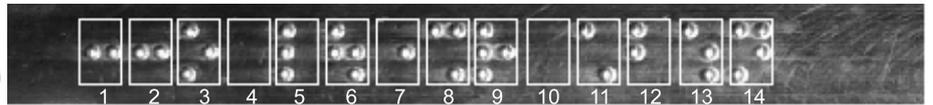


図 2.17. 点字を箱で覆って撮影している場面

(a) Input image



(b) Results
(Proposed algorithm)



(c) Results (Without
matching process)

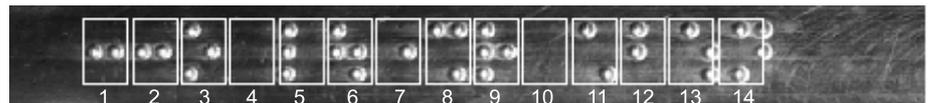


図 2.18. 点字文字ピッチの変動吸収を行わなかった場合の、点字位置の誤認識例

(a) 左方向への傾斜角0度から撮影したサンプル
 Sampled image for 0 degree slanted.



(b) 左方向への傾斜角30度から撮影したサンプル
 Sampled image for 30 degree left slanted.



(c) 上図サンプルのそれぞれ下段点字の認識結果
 (左方向への傾斜角0度と30度で同じ結果)
 Results for lower Braille line of upper samples.
 (Both 0 and 30 degree slanted is the same.)



さくら なりた なりた_くーこー、 とーきょー よこはま くりはま

図 2.19. 左に傾斜した位置からの認識実験の点字サンプルと認識結果 .

第 3 章

コーパスへの検索による点訳の分かち書き誤り修正システム

3.1 はじめに

点訳作業は、点訳者の熟練、意志、そして忍耐を必要とする。特に一冊の書籍、大量の重要書類などの点訳作業をスムーズに行うには多くの点訳訓練が必要であり、需要に対して点訳者の数は圧倒的に少ないのが現状である。したがっていわゆるベストセラーとなった書籍は比較的速やかに点訳されるものの、一般にはまだまだ読みたい本が点訳されている率は低いと言わざるを得ず、点訳本、点訳者の増加は視覚障害者の切望するところとなっている。

本論文の著者が大学の点訳サークルに入って、最初に点訳を始めたとき、点字用紙一枚を仕上げるのに 13 枚打ち直した。一枚を打つのに点字タイプライタでおよそ 1 時間程度かかり、さらに原稿を声で読み上げながら、先輩にチェックしてもらうのにも相当な時間がかかった。厚さ 1cm の文庫本一冊は点字用紙にすると、およそ 300 枚くらいである。1989 年の当時は、まだ点字プリンタは普及しておらず、点字を打つ道具は点字タイプライタが一般的であった。点訳とは、気の遠くなるような作業だと思った。現在はパソコンと点字プリンタが普及し、肉体的には楽になったものの、本稿の主題である点字の「分かち書き」を習得する困難さは現在でも変わらない。

点訳作業は自動点訳できてしまえば終わり、ではない。仮に自動点訳装置が 100% 正確に点訳可能であったとしても、実際にこの文書を視覚障害者の読めるものにしよう、と思う人がいないと点訳は遂行されない。

すなわち，印刷物と視覚障害者の間を結ぶ人間が必要であり，その行為は点訳に興味を持って初めてできることではないかと考えられる．

本論文の著者は長年の経験を通じ，点訳を行いたいと思う人を育てるには，点訳初心者を効率良く次のステップに進めるように補助するシステムが有効であると考えるに至った．ここでの目的は，点訳初心者が苦手とすることが多い「分かち書き」に対する自動修正支援システムを提案し，点訳初心者の点訳スキル向上に貢献しようとする事である．実データを用いて，提案手法の有効性を検証する．

3.2 問題提起

一般にシステムは何らかの原理に基づいてアルゴリズムを構築し，自動的に行うものであるが，ひとつの原理のみで問題全体を統合的に解決することは困難であることが多い．現実の問題には各々固有の困難があり，それらを踏まえての定式化と解決策が必要となるからである．提案システムも例外ではなく，ここで考察する問題には日本語点訳に固有の問題があるので，それについて述べる．

3.2.1 日本語点字と分かち書き

世界中で一般的に用いられている 6 点式点字では，1 文字が 6 つの点の有無によって構成されているため，63 文字しか表現することができない．日本語点字では，漢字は表現できず，カナだけしか存在しない．カナだけでマス空けをせずに文章を書くと，単語の切れ目が明確でない部分が頻出して非常に分かりにくいことが多いだけでなく，複数の解釈可能性が発生しうる．例えば「かねをくれたのむ」は

- 「金をくれ，頼む」
- 「金をくれた，飲む」

の 2 通りの解釈が可能なので，適当な場所でマスを空ける．点字の実際の配列を図 3.1 に示す．以下，点字のマス空けは“ / ”で表し，マスを空けることを「分かち書き」を行う，と表現する．

「かねを / くれ / たのむ」

と分かち書きされれば，読みやすく，誤解される可能性も減少する．ある文字列の内

部で分かち書きを行わないことを「続ける」と表現し、分かち書きを行うことを「切る」と表現する。このように日本語の点訳には「分かち書き」という固有の側面が付随する。ひとつの単語が文字の集まりとして初めから定義されている、例えば西欧系言語には存在しない側面である。また、一般的に点字表記法では、ウ列長音は“ー”(長音符)で表記し、格助詞の“は”と“へ”は発音の通りに“わ”と“え”と表記する。これらも問題を複雑にする要因の一つである。

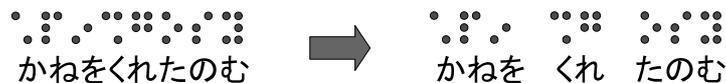


図 3.1. 点字の分かち書きの例

3.3 点字翻訳の問題点

3.3.1 点訳の分かち書きの困難さについて

一冊の本の点訳は数人から数十人で分担して行われることが多い。各々の点訳者がそれぞれ勝手に分かち書きをすると、読む人にとっては混乱の原因となるので、点訳規則の通りに点訳しなくてはならない。点訳作業の大きな困難がここにある。

3.3.2 分かち書きの具体例

日本語点字の「分かち書き」の意味とその困難さをより具体的に説明するため、非課税、非常識などの「非」について考える。これは後に来る要素に依存して、「切る」場合と、「続ける」場合の両方が存在する。点訳のてびき [30] には、

- 「原則として、「接頭語・接尾語・造語要素と自立語との間は続けて書く。」と記載されている。例として、「非合法」、「非人情」が挙げられている。しかし、備考1として、
- 「接頭語・接尾語・造語要素であっても、意味の理解を助ける場合には、発音上の切れ目を考慮して区切って書いてよい」と記載されている。例として、「非人道的」が挙げられている。点字は表音文字

であるから，話し言葉の切れ目に準じるという理解を行うことは可能である．しかし，「非公式学校」や「非営利団体」のような例も存在する．この単語の分かち書きは困難で，「非公式」という一つの単語としてとらえるか，「非」「公式」という二つの単語としてとらえるかというのは，解釈する人に依存すると言わざるをえない．それぞれの熟語の例を図 3.2 に示す．ここでは「非」を例にとって紹介したが，「純」「新」など，他にも接頭語的要素は，多数存在している．また，接頭語的要素ではないが，「みぎはんしん」は途中で分かち書きをしないが，「みぎ / はんぶん」は途中で分かち書きをする，など点字初心者にとって，規則の不可解さについて考えさせられるような例は多数存在する．

「非」のすぐ後ろを続ける熟語
非常口 非課税 非通知

「非」のすぐ後ろを切る熟語
非人道的 非効率的 非上場企業

「非」のすぐ後ろを切るか続けるかの判断が
非常に困難な熟語
非公式学校 非営利団体

図 3.2. 接頭語「非」の後に続く熟語の例

3.4 提案システム

図 3.3 は提案システムのコンセプトを模式的に示したものである．

提案システムは，点訳された文から，その分かち書きの誤りを判別し，修正するものである．提案システムで修正を行う直接の対象は，点訳ファイルであり，点字の文字コードによって記述されている．すでに述べたとおり，本論文の説明においては，点字をかなで表現しているが，実際の提案システム内部では点字の文字コードで検索，比較を行っている．修正の範囲は，基本的には対象となる点訳ファイルのすべての語である．提案システムは，修正対象のある一部分を取り出し，それが大量の点訳ファイル（「点訳コーパス」と呼ぶ）の中に何件含まれているかを調べる．この含まれた件数を「ヒット件数」と呼ぶことにする．そして，そのヒット件数から，分かち

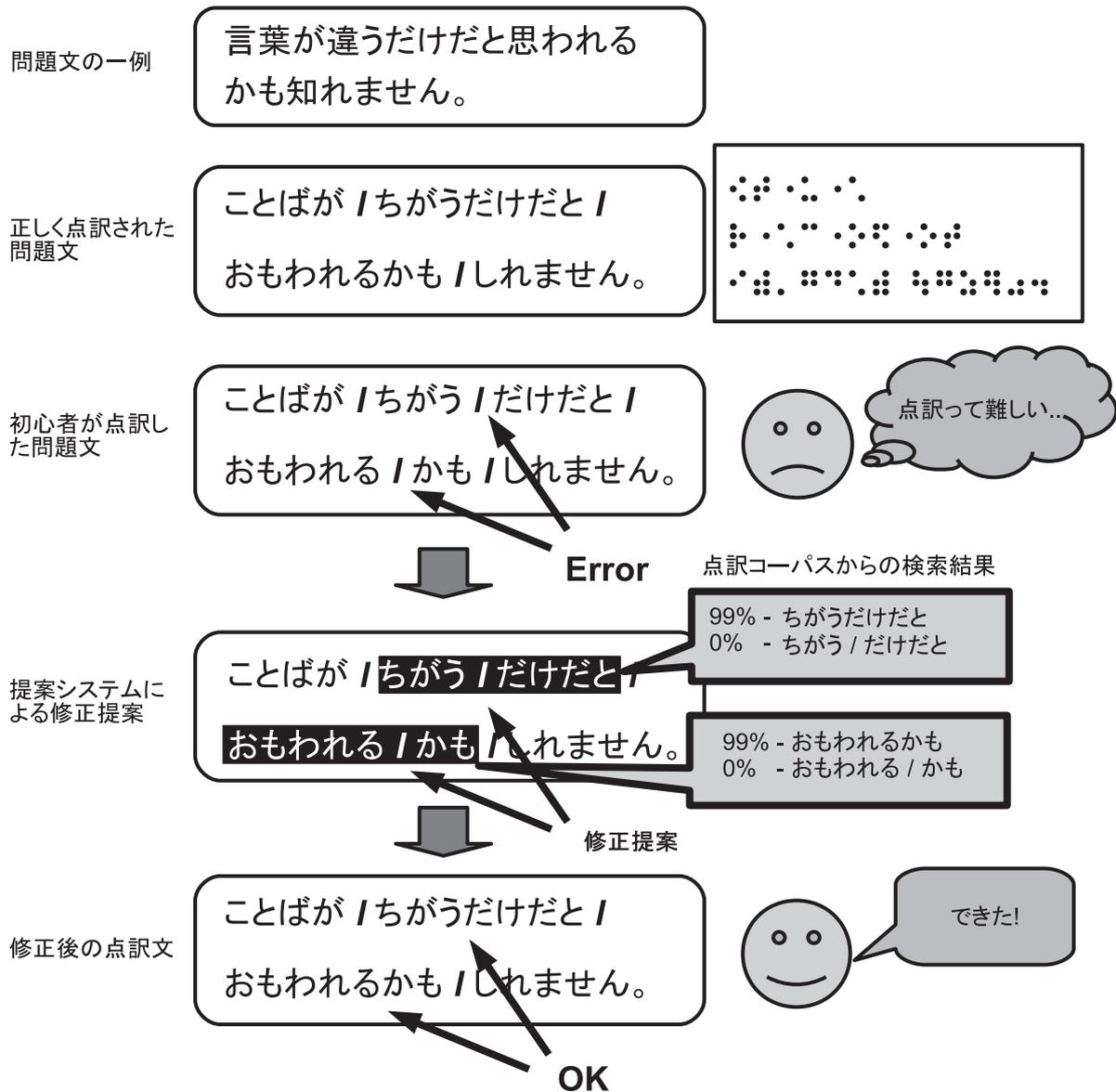


図 3.3. 提案システムのコンセプト模式図

書きの誤りを判別する。

3.4.1 コーパス検索

提案システムでは、大量の点訳ファイル（点訳コーパス）から検索を高速に行うため、あらかじめ対象点訳ファイルから点字文字列を抽出して、索引ファイルにまとめている [24]。索引のキーには、「点字の分かち書き」で区切られた、点字部分文字列

をコードに変換したものをを用いた。これを「キ一点字文字列」と呼ぶ。検索を高速にするため、原文上で「キ一点字文字列」に後続する 14 文字の点字文字列も索引ファイルに含めた。また、原文上で「キ一点字文字列」に先行する点字文字列も 8 文字を索引ファイルに含めている。この先行する点字文字列は点訳文字列の修正には機能しないが、ユーザが用例を検討する際の補助として機能させることができる。例えば、「北アメリカ」は点字表記辞典 [25] によると「きたあめりか」と続けるが、点訳コーパスで「きたあめりか」を検索すると、「きた / あめりか」と切られている用例もわずかながら見つかる。原文上で先行する点字文字列も索引ファイルに含めているために、これが「ひがしから / きた / あめりか / かんたいわ」の表示も可能であり、その場合ユーザは「きた」が形容詞の「北」ではなく、動詞の「来た」であると容易に理解できる。

3.4.2 分かち書きの修正アルゴリズム

分かち書きの誤りは次の 2 種類ある。

1. 本来であればマス空けをせずに続けるべき場所をマス空けしてしまった誤り。
2. 本来であればマス空けをするべき場所でマス空けをしなかった誤り。

まず、(1) について考える。修正前の点字文字列の、すべてのマス空けについて、マスを空ける場合と、空けない場合の両方について上述のヒット件数を求める。そのヒット件数の比が $1 : \alpha$ 以上であれば、修正すべきものと判断し、マス空けを削除する。

次に、(2) について考える。修正前の点字文字列の、あるマス空けによって囲まれた部分点字文字列について、マス空けをしなかった場合と、部分点字文字列中のある位置でマス空けをした場合とのヒット件数の比を求める。その比が $1 : \alpha$ 以上になるようなマス空けが存在すれば、修正すべきものと判断し、マス空けを追加する。ここで、 $1 : \alpha$ は割合のしきい値である。しきい値の決め方は後述の 3.5.4 項、3.6.1 項において考察を行う。例として、点字文字列の「とってかえず」、「らんど / まーく」について、分かち書き修正を試みた場合の途中経過を図 3.4 に示す。

例1.

対象点字文字列: 「とってかえす」

コーパスに対する検索結果:

「とってかえす」:	37件	-(1)
「とって / かえす」:	672件	-(2)

→ (1) : (2) = 1 : 18.16 > しきい値 α であれば
「とって / かえす」と修正

例2.

対象点字文字列: 「らんど / まーく」

コーパスに対する検索結果

「らんど / まーく」:	53件	-(1)
「らんどまーく」:	909件	-(2)

→ (1) : (2) = 1 : 17.15 > しきい値 α であれば
「らんどまーく」と修正

図 3.4. 分かち書きの修正アルゴリズム適用例

3.4.3 「読み」からの自動点訳

ある点字文字列 B がまったくコーパスにヒットしない場合も当然考えられる。その場合、かな漢字変換の文節数最少法 [27] に類似の手法を用いて分かち書き対応する。文節数最少法では、解析結果の文節数を最少にする解析を採用するが、本システムでは、それぞれがコーパスにヒットするように解析結果をいくつか分割し、その分割数が最少になる分割を採用する。最少分割問題を定義するため次を考える：

- 入力点字文字列のうち、 i 番目の点字から j 番目の点字までの部分点字文字列を $B(i, j)$ とする。
- したがって点字 L 文字から構成される入力点字文字列は $B(1, L)$ となる。
- 以下、部分点字文字列を単語と表現し、複数の単語の集合を「単語列」と呼ぶ。
- 単語を w とするとき、 w をコーパスから検索したときのヒット件数を $hit(w)$ と書く。

- 単語列 w_1, w_2, \dots に含まれる単語の数を $|w_1, w_2, \dots|$ とする .

この時, 対象とする最少分割問題は次のように定義される :

点字の単語分割問題

入力文字列 $B(1, L)$ の単語分割は ,

$$B(1, L) = w_1, w_2, \dots$$

$$w_{best_1}, w_{best_2}, \dots = \arg \min_{w_1, w_2, \dots} |w_1, w_2, \dots|$$

ただし w_1, w_2, \dots は全て $hit(w_k) > 0$

となるような単語列 $w_{best_1}, w_{best_2}, \dots$ を求める問題と定義される .

これは複数の可能な経路をもつ逐次データの最適な経路を求める Viterbi アルゴリズム [28] によって計算することができる . 概略を下記に示す .

1. $N(x)$ は 1 文字目から x 文字目までの最少分割数 (全ての分割された部分点文字列が $hit(w) > 0$) とする . もし分割できなければ $N(x) = \infty$ とする .
2. $W(x)$ は 1 文字目から x 文字目までの最少分割数を与える単語列のうち , 末尾の単語とする .
3. $I(x)$ は 1 文字目から x 文字目までの最少分割数を与える単語列のうち , 末尾の単語の開始位置とする .
4. $S(x)$ は w_{best_x} の開始位置を保持する配列とする .

点字の単語分割問題を解くアルゴリズム

(1) 初期化

$$N(0) = 0$$

for $i = 1$ to L do, $N(i) = \infty$

(2) 繰り返し: $x = 1 \sim L$

$$N(x) = \min f(i, x)$$

$$W(x) = B(I(x), x)$$

$$I(x) = \arg \min_i f(i, x)$$

$$f(i, x) = \begin{cases} N(i-1) + 1 & \text{if } \text{hit}(B(i, x)) > 0 \\ \infty & \text{if } \text{hit}(B(i, x)) = 0 \end{cases}$$

(3) 終了

$$M = N(L)$$

$$w_{best_M} = W(L)$$

$$S(M) = I(L)$$

(4) 繰り返し: $k = M - 1, M - 2 \dots 1$ (バックトラック)

$$w_{best_k} = W(S(k+1) - 1)$$

$$S(k) = I(S(k+1) - 1)$$

(5) $w_{best_1}, w_{best_2}, \dots, w_{best_M}$ が複数ある場合

単語列それぞれのコーパスに対するヒット件数 $\text{hit}(w_{best_k})$ は 1 以上で、なおかつ分割数 M が最少になるように分割するが、そのような分割方法が複数ある場合、ヒット件数の総積

$$\prod_{k=1}^M \text{hit}(w_{best_k})$$

が最大となるような分割を採用する。

このアルゴリズムでは、例えば「常識に欠けます」という文を点字で読み下すと「じょーしきにかけます」となる。この点字文字列の分かち書きを判定する過程を図3.5に示す。この点字文字列はそのままではコーパスにはヒットしない。それぞれがコーパスにヒットするように最少分割すると、

- 「じょーし / きにかけます」
と、
- 「じょーしきに / かけます」
の二通りに解釈できる。

前者は「じょーし」のヒット件数が8372件、「きにかけます」のヒット件数が9件、前者のヒット件数の積は 7.5×10^4 、後者は「じょーしきに」のヒット件数が4048件、「かけます」のヒット件数が3192件、後者のヒット件数の積は約 1.3×10^7 であり、後者のヒット件数の積のほうが大きいため、後者の「じょーしきに / かけます」を採用する。

3.4.4 点字の仮名遣い，連濁

日本語点字に固有の問題のひとつにウ列長音の扱いがある。例えば「腹の内が」は「はらの / うちが」だが、誤って分かち書きを行わなかった場合、点訳の仮名遣いのルールにより、ウ列長音は“ー”(長音符)を使って表記するため、「はらのーちが」となってしまう。このような点字文字列を検索する場合、長音を「う」に直してから、コーパスを検索している。またさらにもう1つの問題として、連濁の扱いがある。連濁とは語と語が接続するときに、後の語の最初の音が清音から濁音に変化することを言う。例を挙げると、「かぶしき」と「かいしゃ」が接続した場合に、「かいしゃ」の最初の音が清音から濁音に変化して「かぶしきがいしゃ」となる現象である。連濁をおこす可能性があると考えられる語は、これについても考慮して、点訳コーパスを検索している。「たまり醤油」は「たまり / しょーゆ」と「たまりじょーゆ」の両方について、コーパスを検索している。

分かち書きされていない点字文字列
「じょーしきにかけます」の分かち書き

1. 部分点字文字列を点訳コーパスより検索

じょーしきにかけます	(0件)
じょー	(25614件)
じょーし	(8339件)
じょーしき	(4393件)
じょーしきに	(4045件)
うし	(5311件)
しき	(34279件)
しきに	(5429件)
きに	(440182件)
きにかけます	(9件)
かけ	(68824件)
かけます	(3193件)
ます	(9494件)

2. 件数>0で、最少分割数となる解を求める

じょーし / きに / かけ / ます	(4分割)	×
じょーし / きに / かけます	(3分割)	×
じょーしきに / かけ / ます	(3分割)	×
じょーし / きにかけます	(2分割)	○ - (1)
じょーしきに / かけます	(2分割)	○ - (2)

3. 分割数が同じ場合はヒット件数の総積の最大となる解を採用する

(1) じょーし (8339件) / きにかけます (9件)
 $8339 \times 9 = 75051$

(2) じょーしきに (4045件) / かけます (3193件)
 $4045 \times 3193 = 12915685$

(1)<(2) よって
 (2)の「じょーしきに / かけます」を採用

図 3.5. 「読み」からの自動点訳のアルゴリズム適用例

3.4.5 “副作用” とその低減について

「もう学校には行かない」は点訳では、「もー / がっこーにわ / いかない」とマス
 空けされる。「もーがっこー」をコーパスから検索すると、途中で分かち書きをする

「もー / がっこー」のヒット件数が 56 件で全体の 2%，分かち書きをしないで一続きで書く「もーがっこー」のヒット件数は 2869 件で全体の 98% である．3.4.2 項で示した通り，この件数の比がしきい値 $1 : \alpha$ 以上であれば，提案システムは誤って修正を行ってしまう．つまり，「もう学校には行かない」という文を提案システムにかけると，「盲学校には行かない」という意味に変換されてしまう．「もう学校には」など，ある程度“副作用”が発生する単語を予測し，修正から除外することも考えられるが，完全に副作用をなくすことは難しい．修正後，修正箇所を人が見てチェックする必要がある．

“副作用”を低減するためには，これを予測する必要がある．これには，正しいとわかっている点訳ファイルに対し，あえて修正をおこない，どの単語が「誤って修正されてしまった」かを記録することにより，自動的に予測することができると考えられる．サピエ図書館に登録されている 500 タイトルの点訳ファイルについて，“副作用”を検出するため，あえて誤り修正をおこなった結果，以下のような“副作用”が検出された．ヒット件数の割合のしきい値は $1 : 10$ としている．

- いた美和 [正:いた / みわ 誤:いたみわ (痛みわ)]
- いたメモの [正:いた / めもの 誤:いためもの (炒め物)]
- この蓑 [正:この / みの 誤:このみの (好みの)]
- (手紙を) したため， [正:したため 誤:した / ため ((動作を) したため，)]
- 美奈さんわ [正:みな / さんわ 誤:みなさんわ (皆さんわ)]
- 蒸すこと [正:むす / こと 誤:むすこと (息子と)]

3.4.6 数字の取り扱い

数字はすべて，同じものとして取り扱う．分かち書きの規則上，数の大きさが分かち書きに影響をあたえることはないからである．例えば，「第 1 次」，「第 2 次」...「第 n 次」などはすべて同じ分かち書きである．索引ファイルの作成時に，数字は便宜上，すべて 1 に置換し，登録してある．参照時にも，数字は便宜上 1 に置換し，参照する．

3.5 実験

3.5.1 コーパスに使用した点訳ファイル

点字図書館などの団体である，全国視覚障害者情報提供施設協会が運営している「ないーぶネット」(現在は「サピエ図書館」に移行)に登録されていた点訳ファイルのうち，一部の34217タイトルの点訳ファイルをコーパスの作成に使用した．2001年に新たな点字表記法 [26](以下新表記法とする)が制定されているが，予備実験では，2001年版より前の点字表記法(以下旧表記法とする)に基づく点訳ファイルも混在しており，これが悪影響を与えることが分かった．そのため，あらかじめ新表記法に基づく点訳ファイルのみを選んで使用した．点字表記法が新表記法に基づくかどうかの判断は，以下のように行った．

1. 新表記法では，名詞 + サ変動詞「する」(「掃除する」「勉強する」等)の扱いが変わり，旧表記法では原則として前の語に続け，新表記法では前の語との間はマスを空ける．これを判別に利用した．
2. 但し，旧表記法であっても，名詞 + サ変動詞ではない，単独の「する」は前を空ける．また，新表記法においても，サ変動詞ではない「キャスル」や「ござりまする」などは当然前と続けることになる．具体的には，あるタイトルの点訳ファイルについて，全ての「する」を数え，「する」の前が空いていない場合の割合が3%以下であれば，そのタイトルは新表記法に基づいているとみなす．ただし，「対する」「制する」「属する」などは，新表記法に基づいていても，前の語と続けるので，これらは予め除外した．
3. 上述のしきい値3%については，以下の考察から設定した．
 - (a) 本論文の著者が点訳ファイルの内容を確認し，旧表記法に基づくものであると判断した20タイトルについて，「する」の前が空いている割合は，平均64.80%(最少31.75%，最大88.46%)であった．
 - (b) 一方，同じく新表記法に基づくものであると判断した点訳ファイル20タイトルについて，「する」の前が空いている割合は，平均0.43%(最少0%，最大3.51%)であった．
 - (c) また，すべての点訳ファイル(コーパスに入れなかったものも含め，42514タイトル)について，それぞれのタイトル毎の「する」の前が空いている

割合を算出した。

- (d) これらの割合で分類したヒストグラムを図 3.6 に示す。横軸はタイトル数，縦軸は「する」の前が空いている割合である。これらを検討し，割合が 3% 以下である 34217 タイトルをコーパスに含めることとした。

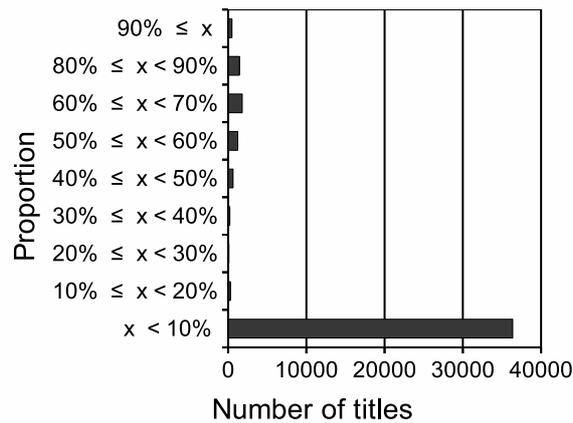


図 3.6. すべての点訳タイトルを，それぞれのタイトル毎の「する」の前をマス空ける割合 x で分類したヒストグラム

コーパスに使用した点訳ファイルの分類別の内訳を表 3.1 に示す。最終的なコーパスのファイルサイズは約 32GB である。

3.5.2 評価用データセット

実験に用いた原文，修正前点訳ファイルと，正解点訳ファイルは次のようにして取得，作成した。点字図書館などの情報提供施設の団体により運営されている「サピエ図書館」で公開されている点訳ファイルの，いくつかのタイトルについて，その原本書籍を購入し，裁断した後，イメージスキャナと OCR ソフトにより漢字かな交じりテキストに変換した。テキスト化した文書は複数人が目で見えて校正を行った。これらにより 3400 文，146061 文字の原文を得た。修正前点訳ファイルは評価用データセットの原文を各種の自動点訳ソフトで点訳したものを使用した。正解点訳には公開されている点訳ファイルを用いた。このようにして作った修正前点訳ファイルと正解点訳を照らし合わせると，原文には以下に述べる誤りの定義を用いると 429 個の分かち書きの誤りがあった。

表 3.1. コーパスに使用した点訳ファイルの分類別タイトル数

分類	タイトル数	割合 (%)
0 総記	2159	6.31
1 哲学	1671	4.88
2 歴史	1555	4.54
3 社会科学	3342	9.77
4 自然科学	1900	5.55
5 技術	1523	4.45
6 産業	524	1.53
7 芸術	1536	4.49
8 言語	1560	4.56
9 文学	15727	45.96
記載なし, 不明	2720	7.95
合計	34217	100.00

3.5.3 誤りの定義

分かち書きの誤り

以下の通り3種類に分類する.

1. 「切りすぎ」: 正解は続けているにもかかわらず, 対象が切れている
2. 「切り忘れ」: 正解点訳では切れているにもかかわらず, 対象点訳では切れていない
3. 「切り位置誤り」: 「切りすぎ」と「切り忘れ」が隣接している場合, 例えば「回転待ち行列」が正解点訳は「かいてんまち / ぎょーれつ」であるが, 対象点訳では「かいてん / まちぎょーれつ」となっているような場合, 単に「切りすぎ」1個と「切り忘れ」1個とはせず, 「切り位置誤り」1個とカウントした。「切りすぎ」「切り忘れ」「切り位置誤り」は合計して「分かち書きの誤り」とした.

読みの誤り

正解点訳の単語を単位とし，それに対応する対象点訳が 1 文字でも誤っている場合，1 個の読みの誤りとした．

読みが誤っている部分については，分かち書きも誤っている可能性は高い．読みと分かち書きが両方誤っている場合も，「読みの誤り」に分類した．

分かち書き誤り率

上述の考察を踏まえ，分かち書き誤り率を以下のように定義した．

$$\text{分かち書き誤り率} = \frac{\text{分かち書き誤り数}}{\text{正解点訳の分かち書き数}}$$

3.5.4 実験 1 ヒット件数の割合のしきい値の検討

3.4.2 項において説明した，アルゴリズムにおけるヒット件数の割合の，修正を行うか否かのしきい値を検討するため，この割合のしきい値 $1 : \alpha$ を

1. $1 : 2$ とした場合，
2. $1 : 5$ とした場合，
3. $1 : 10$ とした場合，
4. $1 : 20$ とした場合，

の 4 通りで，それぞれ提案システムによる修正を行った．

3.5.5 実験 2 点訳コーパスの信頼性についての検討

点訳コーパスは全国の点訳者が点訳したものを採録している．採録された点訳は厳しい基準で作成されているが，点訳コーパスの検索結果が 100% 正しい点訳であることが保証されているわけではない．そこで点訳コーパスの信頼性について検討するため，まず点字表記辞典 [25] に掲載されている語から無作為に 1023 語を選んだ．それらの語の，分かち書きをすべて消去した文字列を，前述の点訳コーパスによって検索し，点字表記辞典 [25] の通りに出力されるかを確認した．

3.5.6 実験3 点訳初心者による点訳の修正

ここでの主目的の実験設定を述べる。点訳初心者の訓練には通常1時間程度の講習を10数回繰り返し、その後少しずつ点訳をしてもらう方法がよく用いられる。このような過程で、提案システムを用いることにより、誤りをある程度自分で修正することができ、点訳スキル向上に寄与できるのではないかと考えている。本研究では点訳未経験者に次のような設定で点訳をしてもらい、その点訳を提案システムで修正し、どのレベルの点訳が可能であるかを検討した。

- 被験者は実験以前に点訳の経験はない。人数は10名、年齢は21歳から67歳であった。
- 問題文、正解点訳は評価用データセットの一部を使用した。
- 被験者10名は、それぞれ異なった問題文を用いた。問題文の文数はそれぞれ15文で、問題文の文字数(漢字かな交じり文)は15文の合計で500文字から715文字、平均592文字であった。
- 点訳についての講習を30分間だけ行い、その後点訳をさせた。
- 点訳後、提案システムを用いて修正を行い、修正前と修正後の分かち書きの誤り率を算出した。
- 今回の実験においては仮名遣いの誤りについては考慮しないものとした。つまり、仮名遣いの誤りについては、最初の点訳後、本論文の著者が修正した。

3.6 結果と考察

3.6.1 実験1 ヒット件数の割合のしきい値の検討

3.5.2で述べたように対象データセットは3400文、146061文字からなり、修正前点訳ファイルと正解点訳を照らし合わせると、修正前点訳ファイルには429個の分かち書きの誤りがあった。これに基づきヒット件数の割合のしきい値を検討するための実験1の結果を表3.2に示す。修正時間はCore i7 3.4GHzのコンピュータにおいて、それぞれ882秒であった。以下の実験では、修正時間はこのコンピュータにおける所要時間を表記する。

割合のしきい値が1:10の場合が最良であり、最も誤りが少なくなったので、以降

表 3.2. 提案分ち誤り修正アルゴリズム性能評価実験．ヒット率 $1 : \alpha$ しきい値により性能が変わる．

ヒット件数の割合のしきい値	分ち誤り数修正前 (個)	分ち誤り数修正後 (個)
1 : 2	429	222
1 : 5	429	216
1 : 10	429	212
1 : 20	429	222

の実験においては、「割合のしきい値を $1 : 10$ とする。」

$1 : 10$ の場合に比較して、 $1 : 2$ の場合は、誤りが 10 個増加した。この内訳を見ると、正しい修正により誤りが 13 個減少したものの、誤った修正 (副作用) により誤りが 23 個増加し、差し引きで誤りが 10 個増加したものである。このときの正しい修正の例と誤った修正の例を以下に示す。

・正しい修正の例

誤:じみんとーないでの (自民党内での) 5 例 正:じみん / とーないでの 17 例

・誤った修正の例

正:じゆーのみを (事由のみを) 6 例 誤:じゆーの / みを (自由の実を) 16 例
 正:いわた / しに (岩田氏に) 4 例 誤:いわたしに (磐田市に) 11 例

3.6.2 実験 2 点訳コーパスの信頼性についての検討

結果

点字表記辞典 [25] に掲載されている語から無作為に 1023 語を選び、点訳コーパスから検索した。点訳コーパスの検索結果のうち、点字表記辞典の分かち書きを正解とし、それ以外を不正解とした。結果を表 3.3 に示す。

表 3.3. 点字表記辞典掲載語の点訳コーパスへの検索結果を，不正解数と正解数で分類したもの

不正解数と正解数による分類	単語数 (語)	割合 (%)
不正解数 $\times 10 \leq$ 正解数	913	89.25
不正解数 \leq 正解数 $<$ 不正解数 $\times 10$	105	10.26
不正解数 $>$ 正解数	5	0.49
計	1023	100.00

考察

今回点字表記辞典から選んだ語のうち，89.25% は，不正解数と正解数の割合がしきい値の 1:10 の範囲に入った．また，正解数が不正解数を下回ったものは選んだ語の 0.49% であった．不正解と正解の割合がしきい値の 1:10 の範囲に入らないものは合計で 10.75% であった．点字表記辞典は，分かち書きの難解な単語のみを収録しており，そのために不正解数が多かったものと考えられる．したがって，利用者は提案システムの結果に誤りも有りうる，という前提のもとに使う必要があることはもとよりである．

3.6.3 実験 3 点訳初心者による点訳の修正

結果

提案システムを使用しなかった場合の平均誤り率 (修正前平均誤り率とする) と，提案システムを使用して修正した場合の平均誤り率 (修正後平均誤り率とする) を図 3.7 に示す．

点訳の時間には特に制限を設けなかったが，17 分から 35 分の間であった．このうち，提案システムが修正を行う時間は，Core i7 3.4GHz のコンピュータにおいて 15 文につき 3 秒以内であった．

考察

修正前の平均誤り率が 11.52% であったのに対して修正後 2.12% となったので，平均誤り率は 81.60% 減少したことになる．このシステムにより点訳初心者の訓練に

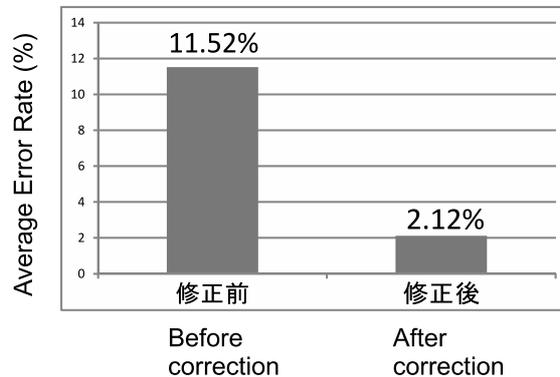


図 3.7. 点訳初心者による点訳の提案システム使用前と使用後の平均誤り率

用いる事ができると考えられる。修正できなかった誤りには、以下の傾向がみられた。

- ある語に、付属語が続いた場合で、両者が続いた単語が点訳コーパスに含まれていない場合。この場合、付属語は前の語に続けるべきであるが、コーパスに含まれていないため、修正できなかった。

例:べた記事ばかり (自立語「べた記事」 + 付属語「ばかり」)

内観意見としては、「提案システムでは誤りがだんだん少なくなるので楽しかった」「また機会があればやってみたい」「そうか、こういうふうに分けるべきだったか」「点訳は大変だということが分かった」などがあつた。

3.7 おわりに

提案システムによる修正を行うことにより、点訳にさほど熟練していない人が点訳した場合に、より正確な分かち書きを行うことができることを示した。提案システムを点訳初心者の訓練に用いることにより、少しでも多くの人々に点訳に対する興味と問題意識を向上させてもらい、ひいては間接的にはあるが視覚障害者の Quality of Life 向上に貢献する可能性を示していると考えられる。

“かたこと”でもある言語を話せる人がいたほうがよい場合があるのと同様、点字に少しでも興味を持ち、“かたこと”ができる人物がいることは視覚障害者にとって意味があると考えられる。例えば、飲食店における点字メニューや、会議の簡単な議題など、特に短い文章においては、分かち書きがさほど正確でなくても、場合によっては有用である。そのような場面にも提案システムは利用可能と考えている。

50 第3章 コーパスへの検索による点訳の分かち書き誤り修正システム

高度の点訳には、分かち書きだけではなく、難しい熟語の読みの正確さや、場合によっては図書の内容についての知識、また点訳基準についての知識などが必要であり、提案システムを使ったからといって、熟練点訳者と同じ点訳がすぐにできるわけではないことは勿論である。

第 4 章

視覚障害者のための特殊 WEB ブラウザ

4.1 汎用 WEB ブラウザと特殊 WEB ブラウザ

4.1.1 特殊 WEB ブラウザとはなにか

特殊 WEB ブラウザとは、ある特定のウェブサイトのみの特化して設計され、当該ウェブサイト用の快適なユーザ・インターフェースを提供する WEB ブラウザである。しかしながら、汎用 WEB ブラウザとは異なり、特殊 WEB ブラウザでは、その設計目的となった特定サイト以外のサイトにはアクセスできない。

日本にも視覚障害者用の特殊 WEB ブラウザの例がいくつかあり、「NAIIV net」（点字図書館の団体が管理する、点字書籍をダウンロードできるサイト）[32]、ニュースサイト [33][34][35]、インターネットラジオ局 [36]、電車時刻表 [37] にアクセスするためのものなどがある。これらの WEB ブラウザの中には、販売されているものも無料で入手できるものもある。

目の見える人用の特殊 WEB ブラウザもあるとはいえ（例えば、YouTube にアクセスできる WEB ブラウザがある）、視覚障害者にとって特殊 WEB ブラウザは有効なものである。

新しいソフトウェアの使い方を学ぶ場合を考えよう。目の見えるユーザは、マニュアルを参照しなくても、例えばメニューバー、アイコン等から選択することで、そのソフトウェアを簡単に使用できる。より複雑な操作の場合でも、目の見えるユーザは、マ

マニュアルを参照しながら選択することができる。よって、目の見えるユーザにとって、特殊 WEB ブラウザと汎用 WEB ブラウザの違いはほとんどないといってよい。一方、多くの場合、視覚障害者は、マニュアルを読んで完全に暗記するまでは、そのソフトウェアを使用できない。この理由により、特殊 WEB ブラウザは目の見える人よりも視覚障害者にとってはるかに重要である。

視覚障害者はスクリーンリーダー・ソフトウェアを用いればパソコンを使用できるが、このソフトウェアは少なくともはじめは使うのが非常に難しい。使いやすいソフトウェアを採用できれば、視覚障害者の情報リテラシーの向上に有効であろう。

4.1.2 汎用 WEB ブラウザと特殊 WEB ブラウザとの比較

視覚障害者用の特殊 WEB ブラウザの有効性を調べるため、あるタスクについて特殊 WEB ブラウザと汎用 WEB ブラウザとの違いを検討した。具体的には、特殊 WEB ブラウザと汎用 WEB ブラウザの両方をそれぞれ用いて、「NAIIV net」から点字書籍をダウンロードして閲覧し、それに要したキー操作の回数を調べた。汎用 WEB ブラウザで要したキー操作が 155 回だったのに対し、特殊 WEB ブラウザで要したキー操作は 36 回だった。図 4.1 にスクリーンショットの一つを示す。

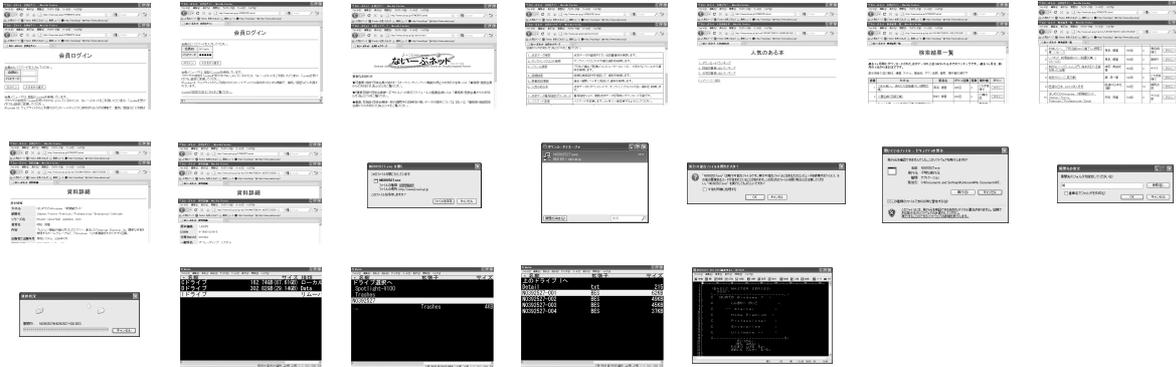
視覚障害者にとってどのような WEB ブラウザが最良であるかは場合による。この問題は、ユーザのスキル・レベル、サイト更新の頻度、魅力的なコンテンツがあるか等、様々な要因に影響される。日本でもっともよく使われている特殊 WEB ブラウザは、おそらく NAIIV net のアクセス用に設計されたものである。

4.1.3 特殊 WEB ブラウザの長所と問題点

特殊 WEB ブラウザの長所

- 特殊 WEB ブラウザは一般に操作が簡単である。視覚障害者用に設計された使いやすいインターフェースを提供でき、上下移動のカーソル・キーとエンター・キーのみを用いて操作できる。
- サポートが明快である。視覚障害者は、自身でソフトウェアをインストールし、

Screenshots for downloading and reading one Braille file from library with general browser



Screenshots for downloading and reading one Braille file from library with special browser



図 4.1. 汎用 WEB ブラウザと特殊 WEB ブラウザで書籍をダウンロードし閲覧した場合のスクリーンショット

操作し、使い方を学ぶことができる。

特殊 WEB ブラウザの問題点

- ウェブサイトのデザインやレイアウトが少し変更されただけでも、特殊 WEB ブラウザは機能しなくなる可能性が高い。
- 目的とするサイト以外のサイトへのアクセス能力は、全くないかきわめて限られている。

4.1.4 パラメータ設定

ある従来の WEB ブラウザ [38] では、正規表現 (RegEx) を用いて記述される、「パラメータ・ファイル」と呼ばれるスクリプトを明記している。このスクリプトの目的は、ユーザをウェブサイトの更新に適応させることである。しかしながら、このスクリプトを作成するのは多くの困難が付随する。リスト 1 は、スクリプト・ファイルの

例である。

List. 1 従来の WEB ブラウザの「パラメータ・ファイル」の例

```
AllCategory=<a href="/([a-z0-9-_]+)/update/[^.]+\..html)">
([^<]+)</a>(\((([0-9]{2}/[0-9]{2} [0-9]{2}: [0-9]{2})\))</li>){0,1}
Category9=
<a href="(\./editorial([0-9]{4}[0-9]{2}[0-9]{2}){0,1}\..html)">([<]+)</a>
Category10=
<a href="(\./column([0-9]{4}[0-9]{2}[0-9]{2}){0,1}\..html)">([<]+)</a>
```

4.1.5 表計算プログラムとの類似

計算表を作成する場合を考えよう。BASIC や C などのコンピュータ言語を用いれば、表計算用のプログラムを作成するのはさほど難しくない。しかしながら、新たな項目が挿入された場合やそれ以外の小さな変更があった場合、そのプログラムを更新しなければならない。一方、プログラミング言語ではなく表計算プログラムを用いれば、この更新は比較的簡単になる。しかし、簡単に使用できる表計算プログラムを作成するのは難しい。

視覚障害者用の特殊 WEB ブラウザについても同じことが言える。プログラミング言語を用いれば、視覚障害者用の特殊 WEB ブラウザを作成するのはさほど難しくない。とはいえ、ウェブサイトの小さなデザイン変更をすべて把握し、WEB ブラウザ・プログラムの修正に反映しなければならない。ゆえに、視覚障害者用の表計算プログラムのようなより有効な WEB ブラウザが必要であると考えられる。

4.1.6 コメント

筆者は、2001年に視覚障害者用の特殊 WEB ブラウザを開発した。開発作業はさほど苦労しなかったが、その保守には多大の時間と労力を費やさざるを得なかった。ウェブサイトのデザインが急に変更されれば、WEB ブラウザをすぐに更新しなければならず、そのような変更はほとんどのウェブサイトでごく普通に発生していたからである。

4.2 提案システム

視覚障害者用の特殊 WEB ブラウザを簡単に構築するためのツールキットを提案する。現在入手できる特殊 WEB ブラウザでは、目的とするウェブサイトのデザインが変更されれば、開発者はソフトウェアを更新しなければならない。一方、提案する WEB ブラウザでは、現在のウェブページを以前のウェブページと照合することにより更新情報を入手できる。本ツールキットには2つの特徴がある。特殊 WEB ブラウザを簡単に作成できることと、本ツールキットを用いて作成した WEB ブラウザではレンダリング・エラーがめったに起きないことである。図 4.2 に従来のアプローチと提案手法を用いた場合のスクリーンショットの一例を示す。

4.2.1 特殊 WEB ブラウザの構築方法

閲覧（コンピュータからユーザへのインターフェース）

汎用 WEB ブラウザは、どの情報が重要かを認識することはできない。基本的に、汎用 WEB ブラウザはウェブページ上のすべての項目を読み上げる（音声合成によるテキスト読み上げ）。一方、特殊 WEB ブラウザは必要な情報だけを読み上げる。広告は読み上げるべきではない。また、項目を読み上げる順番は関連度や重要度に基づいてソートするべきである。

よって、WEB ブラウザを構築するにあたって読み上げるべきデータを簡単に指定できるようにすることは重要である。提案する WEB ブラウザでは、ユーザが読み上げる項目を DOM(Document Object Model) ツリーから指定する。そしてユーザが指定した項目に対応するボックスがウェブページに表示される。これにより簡単に項目を指定できる。

図 4.3 に表からリストボックスへの変換例を示す。

選択（ユーザからコンピュータへのインターフェース）

提案する WEB ブラウザは、リンク、プッシュボタン、ラジオボタン、リストボックス、テキストボックス等を扱うことができる。

- Conventional Way 1.

Generic browser

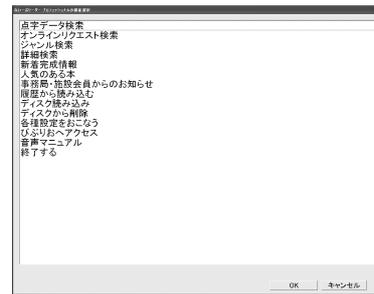
➡ Difficult for beginner



- Conventional Way 2.

Special browser

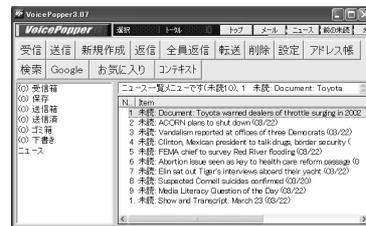
➡ Update is needed if site design is changed.



- Conventional Way 3.

Special browser

➡ Script is hard to write



Example of part of script.

```
[NewsURL_RegExp]
AllCategory=<li><a href="/([0-9]{4}/[A-Z]+/([a-z]+)/{0,1}([0-9]{2}/[0-9]{2})/[^/]+/index.html)">([<]+)</a>
```

- Proposed way

Special browser



図 4.2. 従来のアプローチと提案システムによる WEB ブラウザ構築方法を示すスクリーンショット

取引日時	取引番号	お支払金額(円)	お預り金額(円)	摘要	残高(円)
2010年01月29日 09:29:15	00001	4,650		振込 オオシマ ヒロユキ	3,349
2010年01月29日 09:29:15	00001	52		振込手数料	3,297
2010年02月19日 02:46:36	00001		20,000	セブンATM	23,297

Table with general-purpose browser



Document Object Model of the table

Document Object Model of the table

Selected item of Document Object Model is indicated by a box in another window.



2010/01/29 09:29:15	No. 00001	Withdrawal: 4650	Note: 振込 オオシマ ヒ	
2010/01/29 09:29:15	No. 00001	Withdrawal: 52	Note: 振込手数料	Balanc
2010/02/19 02:46:36	No. 00001	Deposit: 20000	Note: セブンATM	Balanc

もどる
終了する
ファイルへ出力

2010/02/19 02:46:36	No. 00001	Deposit: 20000	Note: セブンATM	Balance: 23297
------------------------	-----------	----------------	--------------	----------------

OK キャンセル

Table with proposed browser

図 4.3. 提案の WEB ブラウザ(構築した WEB ブラウザ)のスクリーンショットの例

図 4.4 は、汎用 WEB ブラウザで表示されたそれらの項目の例を示している。汎用 WEB ブラウザではウェブページ上に多数の選択可能な項目がある場合、ユーザは 1 つを選択して使用しなければならない。通常、ユーザはそれらの選択可能な項目の位置を記憶しなければならない。特殊 WEB ブラウザでは、ユーザは選択可能な項目を次々に使用することができる。

図 4.5 は、汎用 WEB ブラウザと特殊 WEB ブラウザで表示された、いくつかの選

択可能な項目を含むウェブページの例を示している。IDとパスワード用のテキストボックスには、通常は毎回同じ情報が入力される。よって、IDとパスワードはプログラムに保存してもよい。もちろん、ユーザは毎回それらを入力してもよい。



図 4.4. 選択用ユーザ・インターフェースのスクリーンショット

General-purpose browser

照会期間: 年 月

Special browser

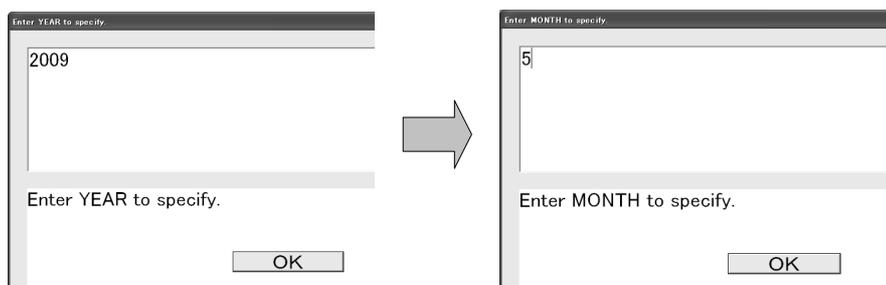


図 4.5. 選択用ユーザ・インターフェースのスクリーンショット

4.2.2 照合アルゴリズム

ソフトウェアはウェブサイトの以前のバージョンを保存しておき、ウェブサイトの現在のバージョンを取得して以前のページと現在のページを照合する。この照合は動的計画法により行われる。

4.3 実験

視覚障害者が Amazon の商品を注文するのに使用できる特殊 WEB ブラウザを開発した。この特殊 WEB ブラウザは基本的に次のように使用される。

- ユーザはバーコード・スキャナを用いて商品のバーコードをスキャンする。
- ソフトウェアはその商品の詳細情報を取得し、その情報をユーザに対して読み上げる。
- ユーザがエンター・キーを押して「購入」を選択すると、ソフトウェアは Amazon にその商品を注文する。
- ソフトウェアをインストールする際は、あらかじめ 1-Click 注文の設定をする必要がある。
- このシナリオでは、ユーザは、モバイル・インターネット接続したポータブル・コンピュータを用いて、スーパーマーケットの商品の名称を知ることができる。
- ユーザは、商品に添付されたバーコード自体ではなく、商品棚のバーコードをスキャンすることになる。
- したがって、その WEB ブラウザでは Amazon 以外のバーコードデータベースも選択できる。

60 第4章 視覚障害者のための特殊 WEB ブラウザ



図 4.6. 提案システムを利用した場合のスクリーンショット

4.4 おわりに

本章では視覚障害者用の特殊 WEB ブラウザを構築するためのツールキットを提案し、その有効性を示した。その結果から、本ツールキットは視覚障害者の生活の質の向上に貢献すると考えられる。

第5章

結論

5.1 本研究で得られた成果

第2章「点字デジタルカメラ画像の自動誤り訂正システム」ではデジタルカメラで撮影したステンレス製の点字を認識する問題に取り組んだ。まず、イメージスキャナを使った点字認識のアルゴリズムは、次の理由で、そのまま適用できないことを指摘した：

- スキャナを使った従来手法の点字認識システムにおいては、点字の点に対し、斜め上から光をあて、立体の陰影を検出する。具体的には、点の画像に対し、明るい部分の下方に暗い部分が存在することを利用し、凸の点の存在を認識する。これに対し、デジタルカメラによって採取したステンレス手すりの点の画像は、光線条件により陰影のパターンは一定しないため、この画像に対して、スキャナで用いられる方法の適用は難しい。
- スキャナを使った点字認識システムにおいては、前述の方法によって点の存在を検出し、 x 軸、 y 軸上への射影を使って点字の位置を検出している。これに対し、デジタルカメラを使って採取したステンレス手すりの点字の画像では、点字の量が少ないこと、レンズの収差があることなどから、点の位置の検出についてもスキャナで用いられた方法は適用できない。

これらを踏まえ、提案システムは以下の方法でこれらの問題を解決した。

- 点の存在の検出については、ストロボを発光させて撮影し、ストロボの光が点

に反射したパターンを認識した。

- また、点の位置の検出については、動的計画法を用いたマッチングを行うことにより、レンズの収差を吸収した。

わい曲収差のあるサンプル 5102 マスに対して検証実験を行ったところ変動吸収なしの場合認識率 81.24 % であったのにたいし、提案システムでは 98.73 % であった。

第3章 「コーパスへの検索による点訳の分かち書き誤り修正システム」で提案してシステムは、点訳された文から、その分かち書きの誤りを判別し、修正するものである。提案システムで修正を行う直接の対象は点訳ファイルであり、点字の文字コードによって記述されている。修正の範囲は、基本的には対象となる点訳ファイルのすべての語である。提案システムは、修正対象のある一部分を取り出し、それが大量の点訳ファイル（「点訳コーパス」と呼ぶ）の中に何件含まれているかを調べる。この含まれた件数を「ヒット件数」と呼んだ。そして、そのヒット件数から、分かち書きの誤りを判別する方法である。検証実験を行い、修正前の平均誤り率が 11.52 % であったのに対して提案システムによる修正後の平均誤り率は 2.12 % となった。平均誤り率は 81.60 % 減少したことになる。このシステムにより点訳初心者の訓練に用いる事ができると考えられる。

第4章 「視覚障害者のための特殊 WEB ブラウザ」では視覚障害者用の特殊 WEB ブラウザを簡単に構築するためのツールキットを提案し、検証実験として視覚障害者が書籍通信販売会社の商品を注文するのに使用できる特殊 WEB ブラウザを開発した。

5.2 今後の課題と結言

これから手がけたい課題は次のようなものである：

- コーパスによる誤り検出システムのしきい値自動調整
コーパスによる誤り検出システムのしきい値は、予備的な評価に基づき単一のしきい値 1:10 を採用している。このしきい値はシステム全体の性能を大きく左右し、提案手法には副作用も付随することは 3.4.5 で述べた。これを改善する方法の一つとして単一ではなく段階的にいくつかのしきい値を用いるようにすることが考えられる。例えば、ある種の系列の組み合わせにおいては、1:10 というしきい値では誤りが生じているので、系列をいくつかのクラスタ（部分

集合) に分割し, クラスタ毎に異なる最適なしきい値を割り当てるような方法が考えている. これは副査村田教授から示唆いただいた.

- 視覚障害者のための楽器演奏学習支援ツール

視覚障害者にとって楽器演奏は困難であることが多い. 著者はリコーダー演奏を視覚障害者が比較的容易に取り組める装置を考案してプロトタイプを実装した. これは指につけたボタンが楽譜に従って使用者の指を押すことにより, どの指をどのような順番で押していくかを指示するものである. 何人かのボランティアに試してもらったところ, 楽しみながら, 練習をしてもらえた. いずれは研究論文としてまとめ, また本格的な実装も試みたい考えている.

- 視覚障害者のための歩行ナビゲータ

GPS を用いた歩行ナビゲータはすでに実装されている [39]. これは装置を首からかけ, GPS 信号をもとに歩行者に指示を与えながら歩くものである. 本論文の著者はこれとは別の視点から, 歩行しながら Web Cam で周りの景色を撮影してもらい, 機械学習的手法から一般物体認識アルゴリズムを構築し, そのうえで, たとえば到達目標への道筋を歩行者に与えようとするものである. 一般物体認識をオンラインで実装するには, おそらく逐次 Monte Carlo など比較的高度な実装手法が必要とされると予想され, まだアルゴリズム構築には至っていないが, 興味ある問題と考えている.

本論文で述べた諸システム構築は多くの時間とエネルギーそして根気を必要とする. 華々しい側面はないが, 本稿のようなシステム構築は, 社会での見えない部分を支えていくのではないかと考えている. このような研究を引き続き地道に続け, ささやかではあるが, 社会に貢献していきたいと考えている.

謝辞

本論文審査委員を務めていただき、貴重なコメントのみならずミスプリ・レイアウトを含め詳細なご指摘をいただきました松本隆(早稲田大学)、内田健康(早稲田大学)、村田昇(早稲田大学)、渡辺亮(早稲田大学)、井上真郷(早稲田大学)、石川准(静岡県立大学)の先生方に感謝します。また、石川准先生には学外審査員を務めていただいたばかりでなく、ひとかたならぬご協力をいただきました。多くのディスカッションと、貴重なコメントをいただいた赤澤史嗣氏に感謝の意を表します。

参考文献

- [1] <http://www.mhlw.go.jp/toukei/saikin/hw/shintai/06/index.html>
- [2] 柏倉 秀克, ”視覚障害者問題の特質と支援上の諸課題”, 桜花学園大学人文学部 研究紀要 第10号 2008
- [3] <https://www.sapie.or.jp/>
- [4] <http://ja.wikipedia.org/wiki/>
- [5] Yasuhiko Ogawa, Yoshinobu Kikuchi and Nobuyuki Ohtake ”Braille reader Computers for Handicapped” Persons 4th International Conference, ICCHP'94 September 1994 Proceedings, pp. 582-589, 1994
- [6] 小山善文, 田島敏久, 古賀広昭, ”凸点・凹点が混在する点字の認識法と劣化点字に対する読み取り制度の評価” 電子情報通信学会論文誌 D-II Vol.J79-D-II No.5 pp.727-736 March 1996
- [7] I.Dias, ”A portable device for optically recognizing Braille-Part II: Software development”, 7th Australian & Newzealand Intelligent Information Systems Conference, pp.18-21, November 2001
- [8] T.W. Hentzschel and P.Blenkhorn, ”An Optical Reading Systems for Embossed Braille Characters using a Twin Shadows Approach”, Journal of Microcomputer Applications, pp.341-345, June1995.
- [9] C.Ng, V.Ng, and Y.Lau. ”Regular feature extraction for recognition of Braille.” 3rd International Conference on Computational Intelligence and Multimedia Applications, 1999. ICCIMA '99. Proceedings, pp. 302-306, September 1999.
- [10] Jan Mennens, Luc van Tichelen, Guido Francois, and Jan J.Engelen ”Optical Recognition of Braille Writing Using Standard Equipment” IEEE Transac-

- tions of rehabilitation engineering, Vol.2 No.4, December 1994.
- [11] Wong, L., W.Abdulla, and S.Hussmann, "A Software Algorithm Prototype for Optical Recognition of Embossed Braille", the 17th conference of the International Conference in Pattern Recognition, August 2004.
- [12] 美原義行, 杉本晃宏, 柴山悦哉, 高橋伸, "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" 日本ソフトウェア科学会第 21 回大会, December 2004
- [13] 吉野和芳, 張善俊 "カメラ付き携帯電話を用いた点字リーダ", 第 23 回八工学会カンファレンス 2008, pp.115-116, August 2008.
- [14] P. Clark and M. Mermehdi, "Recognising text in real scenes," International Journal of Document Analysis and Recognition, vol. 4, pp. 243-257, Aug. 2002
- [15] 美原義行, 杉本晃宏, 柴山悦哉, 高橋伸, "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" 日本ソフトウェア科学会第 21 回大会, December 2004
- [16] 萩原敏浩, 渡部広一, 河岡司, "ひずみを有する活字文字列における形状補正手法", 信学論 (D-II), vol. J86-D-II, no. 2, pp. 262-271, 2003
- [17] 黄瀬浩一, 大町真一郎, 内田誠一, 岩村雅一, "カメラを用いた文字認識・文書画像解析の現状と課題", 信学技報, PRMU2004-246, March 2005
- [18] 吉野和芳, 張善俊 "カメラ付き携帯電話を用いた点字リーダ", 第 23 回八工学会カンファレンス 2008, pp.115-116, August 2008.
- [19] 田中真美, 宮田薫, 西沢達夫, 長南征二, "点字読み取り用センサシステムの開発", 福祉工学シンポジウム講演論文集, no.2, pp. 187-188, Nov. 2002
- [20] 石川准, "自動点訳ソフト「エクストラ」", <http://www.extra.co.jp/> .
- [21] 兵藤安昭, 横平貫志, 早川哲史, 村上裕, 池田尚志, "誤り箇所指摘機能をもたせた点字翻訳編集システム ibuki-ten", 電子情報通信学会論文誌. D-I, vol.84, no.7, pp.1102- 1111, 2001 .
- [22] 勝沼貞幸, "パソコン点訳ソフト「お点ちゃん」", <http://www17.plala.or.jp/otengan/> .
- [23] 宮本修, "デジタルカメラによる手すりの点字認識", 電子情報通信学会論文誌. D, vol.93, no.10, pp.2281- 2291, 2010 .
- [24] 北研二, 津田和彦, 獅々堀正幹, 情報検索アルゴリズム, 共立出版, 2002 .
- [25] 『点字表記辞典改定新版』編集委員会, 点字表記辞典改定新版, 視覚障害者支援総合センター, 2002 .
- [26] 日本点字委員会, 日本点字表記法, 2001 年版

- [27] 吉村賢治, 日高達, 吉田将, “文節数最小法を用いたべた書き日本語文の形態素解析,” 情報処理学会論文誌, vol.24, no.1, pp.40-46, 1983 .
<http://ci.nii.ac.jp/naid/110002723740/>
- [28] A. Viterbi, “Error bounds for convolutional codes and an asymptotically optimum decoding algorithm,” IEEE Transactions on Information Theory, vol.13, pp.260-269, 1967.
- [29] 日本点字委員会, 日本点字表記法 2001 年版, 大活字, 2001 .
- [30] 点訳のてびき, 全国視覚障害者情報提供施設協会第 3 版 (2002).
- [31] R. Bellman, “Dynamic Programming”, Dover Publications, 2003
- [32] Enable Software Inc., “Naiiv Reader”
- [33] Kochi System Development, “MyNews”, <http://www.aok-net.com/>
- [34] Nori Katsuragi, “Voice Popper”, <http://homepage2.nifty.com/oss/>
- [35] Yoshiaki Miyazaki, “MM news”, <http://www.am-corp2.com/>
- [36] Yoshiaki Miyazaki, “MM net radio”, <http://www.am-corp2.com/>
- [37] Kochi System Development, “MyRoute”, <http://www.aok-net.com/>
- [38] C. Asakawa and T. Itoh, “User interface of a home page reader”, Proceedings of the 3rd International ACM SIGCAPH Conference on Assistive Technologies (ASSETS' 98), pp. 149-156, 1998.
- [39] “GPS による視覚障害者歩行支援システムの開発”, 電子情報通信学会 信学技報 2005 年 1 月

著者の業績

論文

宮本修

“ デジタルカメラによる手すりの点字認識, 電子情報通信学会論文誌, Vol. J93-D, No.10, Oct. 2010.

M. Kobayashi, S. Masaki, O. Miyamoto, Y. Nakagawa, Y. Komiya and T. Matsumoto

“ RAV (Reparameterized Angle Variations) Algorithm for On-line Handwriting Recognition ”, International Journal of Document Analysis and Recognition, Volume 3 Issue 3, pp.181-191, 2001.

国際会議 (査読有)

O. Miyamoto

“ Simple browser construction toolkit ”, ICCHP International Conference on Computers Helping People with Special Needs, Part 1, LNCS 6179, pp. 384-391 Jul., 2010.

国内発表

宮本修, 長岡英司, 大武信之

“ 光学的読み取り装置による点字認識 ”, 電子情報通信学会技術研究報告. ET, 教育工学 108(470), pp. 89-92, 2009.

宮本修

“ 視覚障害者向けブラウザをめぐる状況 ”, 情報バリアフリー懇談会第 2 回, 旧郵政省, 2000.

宮本修, 大武信之, 長岡英司, 松本隆

“ 視覚障害者向け GPS ナビゲーションシステムにおけるカメラと画像検索サイトを用いた歩行支援 ”, 電子情報通信学会 2009 年ソサイエティ大会 A-17-7, 2009.

宮本修, 中川洋一, 松本隆

“ On-line 文字認識アルゴリズム Reparametrized Angle Variations を高速に実行するハードウェアボードについて ”, 電子情報通信学会技術研究報告, vol.94, No.548, pp.49-56, 1995.