

コンピュータ理工学特別研究報告書

題目

文書画像を読みやすく拡大表示するための
文字列再配置に関する研究

学生証番号 144335

氏名 加藤 直哉

提出日 平成 29 年 1 月 31 日

指導教員 蚊野 浩

京都産業大学
コンピュータ理工学部

要約

拡大読書器は、カメラなどで撮影した画像・映像をモニターに拡大表示することで、低視力や弱視者の読み書きを支援する装置である。拡大表示された文書を読み進める場合、表示領域を移動させる必要がある。その際、文書や拡大率の大きさによっては、改行位置で上下左右に大きく動かす必要がある。これを繰り返していると、酔ってしまうことや、読んでいた箇所がわからなくなることがある。

本研究では、拡大読書器におけるこのような不具合を軽減するために、入力した文書画像の文字列を再配置して、拡大表示する手法を提案する。ここでの文字列の再配置とは、表示するディスプレイの大きさと拡大倍率に合わせて、文書画像において文字列を改行する位置を変えることである。

本研究で開発した文字列再配置手法は、以下の手順で処理を進める。

- (1) 文書画像を入力する。
- (2) 2値化処理を行う。
- (3) 2値化画像を縦方向に投影したデータを用いて、文書画像を列ごとに分解する。
- (4) 列ごとに分解した画像を横方向に投影したデータを用いて、文字に分解する。
- (5) 文字を再配置した文書画像を生成する。
- (6) 再配置した文書画像をディスプレイに拡大表示する。

ここでの画像の投影は、縦方向または横方向に画素値を積算することでヒストグラムを得る手法である。縦書き文書の場合、まず縦方向に画像を投影する。印刷文書は一方方向に整列した文字列が、適当な間隔を空けて均等に並んでいる。従って、文字列の並びが投影方向と一致していれば、文字列の部分では積算値が大きく、文字列と文字列の間では積算値が小さくなる。このようにして得る投影データを使って、文書画像を列ごとに分解する。同様の処理を横方向にも行う。積算値が大きい箇所は文字がある部分を表している。積算値が小さい箇所は文字と文字の間の空白を表している。このヒストグラムから、空白の中心を区切りとして文字に分解する。

提案手法によって、① 文字列がほぼ完全に垂直方向に並んでおり、② 画像全体の明るさ変化（シェーディング）が少ない、という2点が当てはまる画像について、文字列再配置による再構成画像を作成できた。①に当てはまらない画像は、文書領域を囲む四角形を垂直な長方形に変換することで補正した。②に当てはまらない画像は、画像を4つの領域にわけてそれぞれで2値化することで対応した。

目次

1 章 序論	．．． 1
2 章 文書画像の拡大表示に関する従来技術	．．． 2
2.1 拡大読書器	．．． 2
2.2 活字文書 OCR	．．． 2
2.3 従来研究	．．． 4
3 章 文書画像の文字列再配置手法	．．． 6
3.1 提案手法の概要	．．． 6
3.2 文書画像の 2 値化	．．． 7
3.3 文書領域を列ごとの画像に分解する手法	．．． 8
3.4 段落列画像を生成する方法	．．． 9
3.5 分解した文字を再配置する手法	．．． 10
4 章 提案手法の検証	．．． 11
4.1 開発したプログラムの概要	．．． 11
4.2 開発したプログラムの動作検証	．．． 11
4.3 開発したプログラムでは正しく動作しない場合	．．． 15
4.4 考察	．．． 21
5 章 結論	．．． 23
参考文献	．．． 24
謝辞	．．． 24
付録	．．． 25

1 章 序論

平成18年の厚生労働省の調査によると、視覚障害をもつ人はおよそ31万人であると推計されている[1].視覚に関する障害は、「盲」と「弱視(ロービジョン)」に大きくわけられる。「盲」に分類されるのは、「視覚的な情報を全く得られない、あるいはほとんど得られない人」とされており、読み書きには点字を用いることが多い。「弱視(ロービジョン)」とは、「視力や視野などの視機能低下が原因で、読み書きや移動等の生活機能に継続的に困難を伴う状態のことで、視力がおおむね0.3未満または視力以外の視機能障害が高度な場合」とされている。また、これらの人はルーペや拡大読書器などを用いて印刷物やパソコンの画面を拡大して読んでいる[2].

拡大読書器は、カメラなどで撮影した画像・映像をモニターに拡大表示ことで、低視力や弱視者の読み書きを支援する装置である。書類や手紙等を自身で読むことを容易にする。据え置き型や携帯型がある。厚生労働省が定める、日常生活用具給付制度の対象として定められている。日常生活用具給付制度とは障害者等の日常生活がより円滑に行われるための用具を給付または貸与する制度である[3].

一般的な拡大読書器は、雑誌や書籍など拡大表示したい物を撮影機器で入力し、ディスプレイ画面に拡大表示する。拡大表示された文書を読み進める場合、拡大領域を移動させる必要がある。その際、対象物の大きさや拡大率の大きさによっては、文書の改行位置で上下または左右に大きく動かす必要がある。これを繰り返していると、酔ってしまうことや、読んでいた箇所がわからなくなることがある。

本研究では、拡大読書器におけるこのような不具合を軽減するために、入力した文書画像の文字列を再配置して拡大表示する手法を提案する。ここでの文字列の再配置とは、表示するディスプレイの大きさと拡大倍率に合わせて、文書画像において文字列を改行する位置を変えることである。

本研究でプログラムを作成するにあたり、マイクロソフト社が開発した製品である「Microsoft Visual Studio 2010」を使用した。これはアプリケーションなどの開発支援ソフトであり、その中の機能の一つである「Visual C++」を使ってプログラムを開発した。

本論文は以下のように構成される。2章では一般的な拡大読書器の機能や、活字文書OCRについて述べる。3章では、文書画像の文字列再配置について、提案手法を述べる。4章では、提案手法を適用した結果と考察を述べる。最後に5章で結論を述べる。

2章 文書画像の拡大表示に関する従来技術

この章では従来技術として、一般的な拡大読書器の機能や用途と、関連技術である活字文書のOCR技術、および関連する従来研究について述べる。

2.1 拡大読書器

拡大読書器はカメラなどで撮影した画像・映像をモニターに拡大表示し、低視力者や弱視者の読み書きを支援する装置である。市販されている拡大読書器は、主に据置型(図1(a))と携帯型(図1(b))に分けられる。



図1(a) 据置型拡大読書器



図1(b) 携帯型拡大読書器

(引用元 <http://www.neitz.co.jp/?p=151/>)

(引用元 <http://www.times.ne.jp/visual/vision/compact7hd/>)

拡大読書器の基本構成は、モニター、カメラ、コントローラー、資料台、照明からなる。拡大読書器には拡大機能だけではなく、白黒反転機能、コントラスト強調機能、XYテーブル、カメラのオートフォーカス機能、読む箇所を強調するためのライン機能、マスク機能、赤外フィルター、遠近の切り替えなどの機能が備わっている[4]。XYテーブルは2段構造になっている資料台のことで、上のテーブルは左右に、下のテーブルは上下に動く。XYテーブルの移動方向を固定することも可能で、片方だけを固定することで、段落ごとに読み進めやすくなる。その他の機能は、見やすさに関する機能であり、使用者の視力の状態に合わせて見やすさを調整できる。

価格は数万円から30万円以上するものもある。機能や拡大率などは機種によって違うので、用途に合わせて選ぶ必要がある。

2.2 活字文書OCR

OCR(Optical Character Recognition)は文書画像の文字を認識する技術である。新聞や雑誌などの印刷文字や手書き文字をイメージスキャナやデジタルカメラで読みとり、文字コードに変換する技術である[5]。

印刷や手書きの文書をデジタル化することで様々な用途に使用可能である。視覚障害を持つ人を支援する方法として、デジタル化したデータを読み上げソフトで音声化する

ることや、文字コードを画像化して拡大表示するなどが考えられる。しかし、文字の誤認識が起きた場合、間違っただけの情報を提供することになる。

市販されている OCR ソフトに、パナソニックソリューションテクノロジー株式会社が販売している「読取革命」がある[6]。このソフトは文書画像や PDF ファイルの文字を認識し、Word, Excel, PowerPoint などのファイルに変換することができる。同社の Web サイトに公開されている Ver. 15 の体験版を用いて、文字認識の性能を調査した。調査用の画像(図 2)として、岸見一郎、古賀史健著「嫌われる勇気-自己啓発アドラーの教え」(ダイヤモンド社, 2013)の p. 25 をデジタルカメラで撮影したものを使用した。元画像は幅 939 画素, 高さ 1550 画素で、文字数は 600 字である。

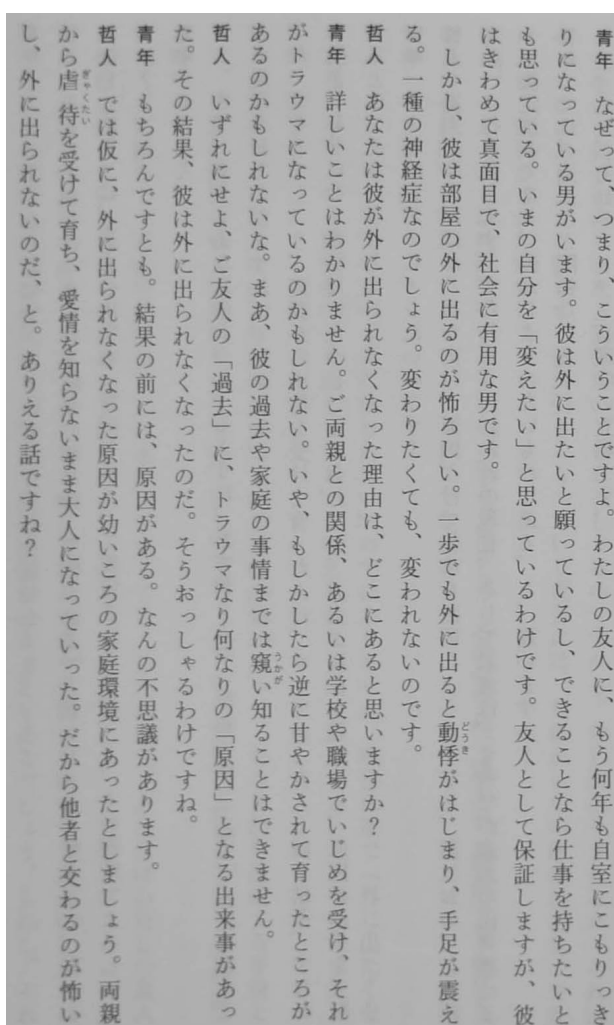


図 2 OCR のテストに使用した画像

表 1 は画像の解像度の違いによる認識率の違いである。図 2 の元画像を読み取らせた場合、誤字は 1%であった。解像度を 75%に落とした場合、認識率は元画像と同じであった。

解像度を 50%に落とした場合, 誤字の割合が 15%になった. この結果から, 市販の OCR ソフトによる印刷文書の認識率は, 元画像の解像度が十分であれば, 99%程度であることがわかった. 図 2 の文書の場合, 99%の認識率であれば元画像の内容を完全に理解することができる.

表 1 画像の解像度の違いによる文字認識率の変化

画像の解像度 (%)	100	75	50
誤字の数(字)	6	6	90
誤字の割合 (%)	1	1	15

2.3 従来研究

文献[7]は拡大読書器が備えるべき画像処理機能としてマスク処理, 白黒反転処理, コントラスト改善処理が重要であると述べている. マスク処理とは, 不要な部分を黒で埋めてしまい, 羞明を防ぐことを目的としている(図 3. 1, 3. 2). 羞明とは, 強い光を受けた時に, 目の痛みなどが生じることで, マスク処理はまぶしさを過剰に感じてしまう人のための機能である.

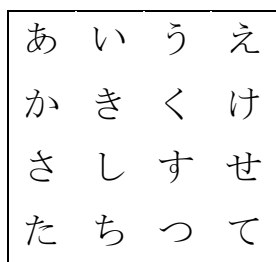


図 3(a) マスク処理をする元画像

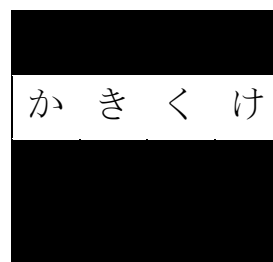


図 3(b) マスク処理をした画像

背景が白に近く, 文字が黒に近い文書画像の場合, まぶしさを過剰に感じる人には見づらい場合がある. 白黒反転処理はこれを軽減するために, 背景を黒にし, 文字を白に反転させることで羞明を防ぐ処理である.

コントラスト改善処理は, 諧調が低下した画像に 2 値化を用いることで, 文字と背景の区別がつきやすいようにする処理である. 2 値化を行うと, 境界領域において, 照明変動の影響などで輝度値が変動するために, チラツキが発生することがある. これを改善するために, 輝度強調処理を加えると, チラツキが起こらないとされている.

文献[8]では、デジタルズームによる細かい拡大率の調整や、白黒2値化画像に対するコントラスト調整機能の必要性が述べられている。

今回調査した範囲では、本研究と類似の研究はなかった。

3章 文書画像の文字列再配置手法

3.1 提案手法の概要

拡大読書器は, 文書画像をカメラなどで撮影した画像を拡大表示する. ここで, 拡大画像の縦横の画素数がディスプレイの画素数よりも大きいと, 読み進むにつれ, 縦横の両方向にスクロールしなければ読み通すことができないため, 使いにくい.

本研究では, これを改善するために文書画像の文字の並びを維持した状態で, 文字単位で画像を再構成し, 文書の内容を変えることなく画像の縦横比を変更する方法を提案する. このように変換した拡大画像を表示することで, 一つの方向にスクロールするだけで文書を読み通すことが可能になる. なお, 本論文では縦書き文書を用いて研究内容を説明するが, 基本的な考え方は横書き文書でも同様である.

図4に提案手法の処理の流れを示す. まず, 文書画像を入力し, 2値化処理を行う. 次に, 2値化画像を縦方向に投影したデータを用いて, 文書画像を列ごとに分解する. 列に分解した画像を横方向に投影処理し, 文字に分解する. 最後に, 文字を再配置することで文書画像を再構成し, ディスプレイに拡大表示する.

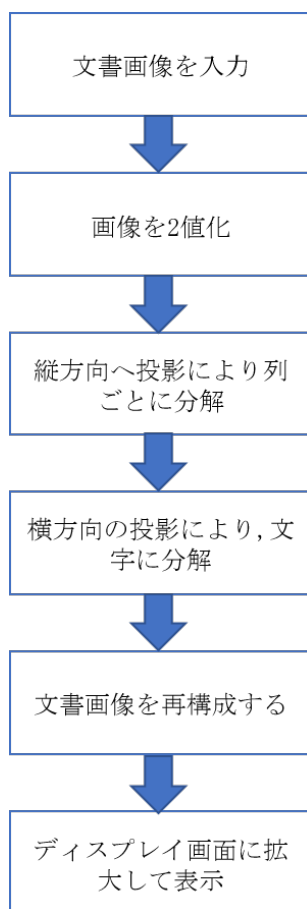


図4 提案手法の処理の流れ

3.2 文書画像の2値化

文書画像を文字領域と背景に2値化する。図2の画像を、OpenCVが提供する2値化関数 `threshold()` を使い、引数 `cv::THRESH_BINARY|cv::THRESH_OTSU` で2値化した例を図5に示す。この場合、大津のアルゴリズムを用いて最適な閾値を計算し、その閾値に従って2値化処理を行っている。

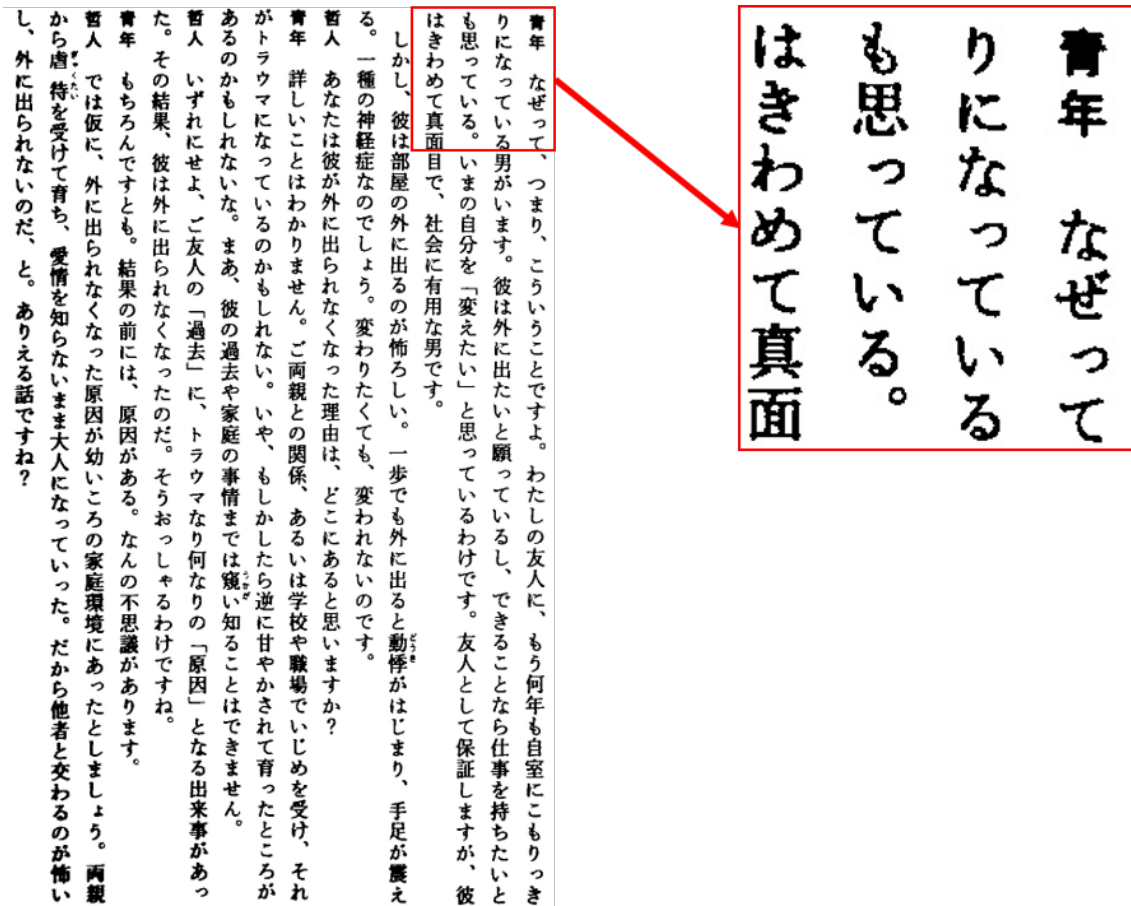


図5 大津の方法で2値化した画像

図5の画像は文字領域と背景を正しく2値化できている。しかし、表示するための画像として利用するには、図5右上の「青」の字など、微妙な濃淡が潰れているという問題がある。これを改善するために、単純な2値化ではなく、以下で説明するように、文字部分の濃淡を残す処理を加えた。

図2の文書画像のヒストグラムを図6に示す。赤線が2値化の閾値である。赤線の右側は背景の画素値である。これらの値は白(255)になるように変換する。そして、左側の画素値を式(1)で処理する。式(1)の計算結果が0未満の場合は0に、255を上回った場合は255に画素値を制限する。この処理により、文字の潰れが改善され、元画像の文字領域の濃淡変化が保たれた画像を生成することができた。その結果画像の一部を図7に示す。図7の画像は厳密に言えば濃淡画像である。しかし、背景領域と文字領域を区別

した画像であるので、2 値的な画像ということはできる。本論文で 2 値化画像とよぶときは、図 7 のような画像を指す。

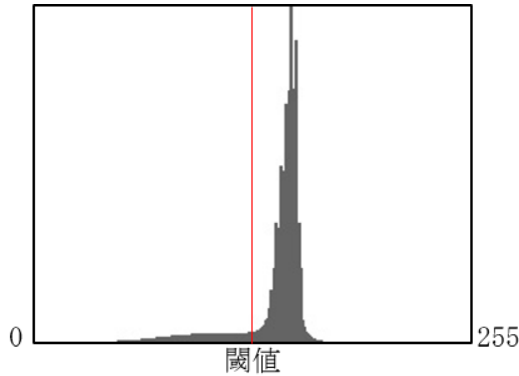


図 6 図 2 の文書画像のヒストグラム

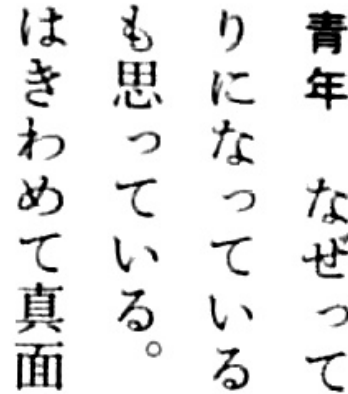


図 7 図 2 を式(1)で処理した画像

$$pv = pv \times 5 - 380 \quad (0 \leq pv \leq 255) \dots (1)$$

3.3 文書画像を列ごとに分解する方法

次に文書画像を文字列に分解する。これを行うために、3.2 で得た 2 値化画像を投影する。ここでの画像の投影は、縦方向または横方向に画素値を積算することでヒストグラムを得る手法である。

印刷文書は一方方向に整列した文字列が、適当な間隔を空けて均等に並んでいる。従って、文字列の並びが投影方向と一致していれば、文字列の部分では積算値が大きく、文字列と文字列の間では積算値が小さくなる。このようにして得る投影データ（ヒストグラム）をもとに、文書画像を文字列に分解する。

縦書き文書の場合、縦方向に画像を投影する。図 8 は、図 4 を 2 値化した画像（図 7）を、縦方向に投影したヒストグラムである。縦軸は、255 から画素値を引いた値の積算値であり、横軸は積算した列の水平方向の位置である。黒い柱のようにになっている部分は、文字列が存在することを表す。柱の間は文字列が存在しないことを表している。この投影データをもとに、文書画像を文字列ごとに分解する。

図 8 に、文字が検出されている付近に、少しか黒画素が検出されている部分がある。これは漢字などにつけられたルビに対応した部分である。縦書き文書画像の場合、文字の右側にルビがつけられる場合があるので、それを考慮して文字列を分解した。文字列の分解は文字列間の中心点を基準として行う。ルビのみを抽出することを防ぐために、ルビによって積算値が低くなった値を文字列であると判断しないように画素値の積算値の閾値を設定した。

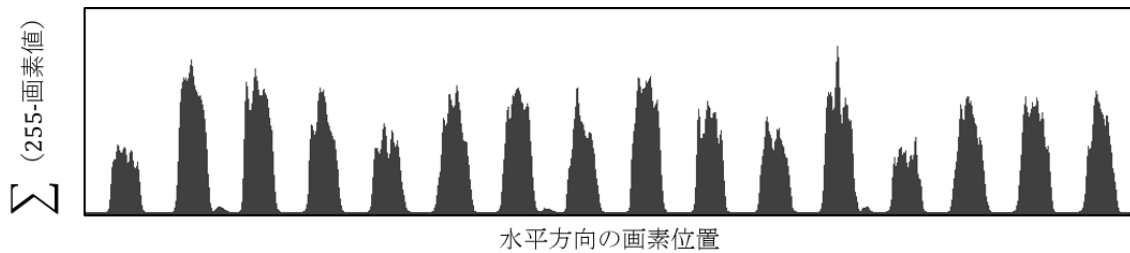


図8 縦書き文書画像を縦方向に投影したヒストグラム

3.4 段落列画像を生成する方法

列ごとに分けた画像から文字の位置を抽出するために、横方向への投影を行う。図10は、図9の画像を横方向へ投影したヒストグラムである。積算値が大きい箇所は文字がある部分を表している。積算値が小さい箇所は文字と文字の間の空白を表している。このヒストグラムから、空白の中心を文字と文字を区切る位置として抽出する。

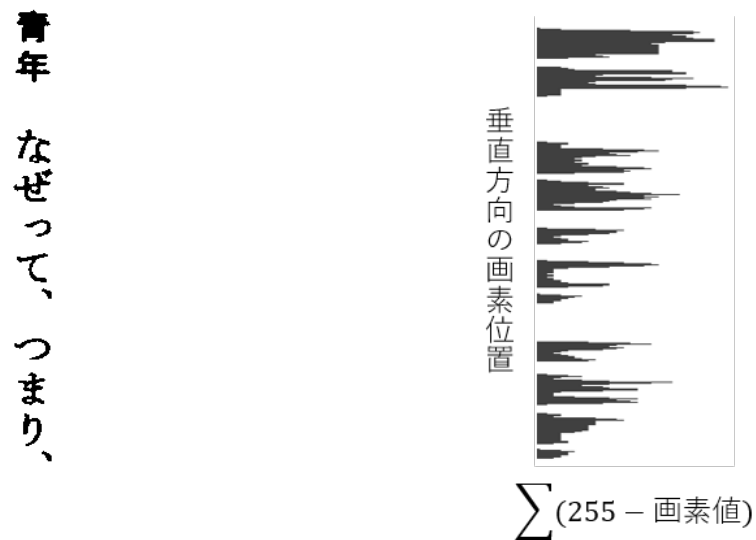


図9 1列に分解された文書画像

図10 図9を横方向へ投影したヒストグラム

次に、横方向の投影データから、その列が次の列につながっているのか、段落の終端であるのかを判断する。これを行うために文字列の終端位置を、元画像の下端と比較する。これらの位置に一定以上の差があれば、段落の終端であると判断する。そうでなければ、段落が続いていると判断する。この手法の問題は、文字列が一番下の位置で段落が終わった場合、段落の終端位置と認識できないことである。

列ごとの画像から、その上と下に存在する空白部分を取り除き、空白のない列画像を作る。空白のない列画像を、段落を単位として縦方向に接続することで、段落内の文字が一行に並んだ段落列画像を生成する。

3.5 分解した文字を再配置する手法

文書画像を構成する文字の位置を再配置する方法を説明する。再配置とは、文書画像を拡大表示した時、一行の長さがディスプレイの縦画素数以下になるように、改行位置を変えることである。ここでは、縦書き文書画像の再配置を行う場合について説明する。

基本的な考え方は、縦に長い段落列画像を適切な位置で改行し、それらを横に並べた画像を作ることである。以下では、一つの段落列画像を、縦画素数 v のディスプレイに、拡大倍率 m で表示するための再構成画像の作り方を説明する。

$c(k)$ を k 番目の文字の区切りとなる画素位置とする。一番目の改行位置を、式(2)を満たす $c(i)$ とする。このようにすれば、最初の列の部分画像を m 倍に拡大表示したとき、ディスプレイの縦画素数に収まる。

$$\begin{cases} c(i) \leq v/m \\ c(i+1) > v/m \end{cases} \dots (2)$$

二番目以降の改行位置は、式(3)を満たす $c(j)$ とする。

$$\begin{cases} c(j) - c(i) \leq v/m \\ c(j+1) - c(i) > v/m \end{cases} \dots (3)$$

ここで $c(i)$ は直前の列で改行した位置である。このようにすれば、2番目以降の列についても、 m 倍に拡大したときに、ディスプレイの縦画素数に収まる。

このように決めた改行位置で段落列画像を分解し、それらを横に並べたものを拡大表示のための再構成画像とよぶ。再構成画像を m 倍に拡大しても、その縦画素数はディスプレイの縦画素数 v を超えない。

4章 提案手法の検証

4.1 開発したプログラムの概要

開発したプログラムの処理の流れを図 11 に示す。

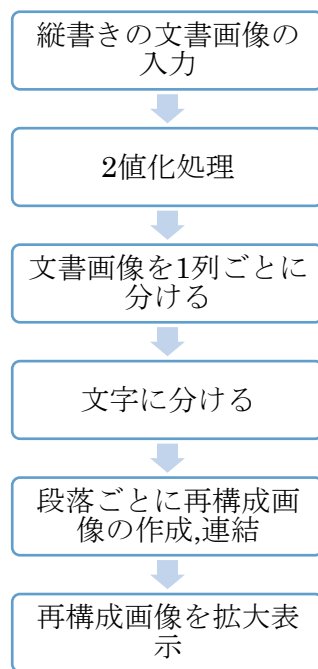


図 11 開発したプログラムの流れ

1 枚の縦書き文書画像を入力し, 3 章で述べた処理を行うことで, 段落ごとに再構成文書画像を生成する. 再構成文書画像を指定倍率で拡大し, 画像表示ウィンドウに表示する. 表示ウィンドウは, 「↑」, 「↓」, 「←」, 「→」キーで画面操作する. 「↑」, 「↓」キーで拡大・縮小率を設定する. 「←」, 「→」キーで文書の横スクロールを行う.

4.2 開発したプログラムの動作検証

図 2 の画像を用いて, 開発したプログラムの動作を検証した. iPhone 6s で表示することを想定して, ディスプレイの画素数は縦 667 画素, 横 375 画素として動作させた.

図 12 (a)~(d)に倍率 2~5 倍の再構成画像 (拡大する前の画像) の一部を示す. これらの図を比較すると, 次のことがわかる. ① 拡大倍率によって文章を改行する位置が異なっている. ② 図 12 (c), (d)において, 文字「こ」が分断されている. ほかに「!」, 「?」, 「え」, 「会」などの文字は分断されてしまう. 横方向への投影を行うと, 字そのものに空白部分がある場合, 分断されてしまうためである.

青年 なぜって、つ
まり、こういうこと
ですよ。わたしの友
人に、もう何年も自
室にこもりっきりに
なっている男がいま
す。彼は外に出たい
と願っているし、で
きることなら仕事を
持ちたいとも思っ
ている。いまの自分を

(a) 2倍で拡大表示する場合

青年 なぜっ
て、つまり、
こういうこ
とですよ。わ
たしの友人に
、もう何年も
自室にこもり
っきりになっ
ている男がい
ます。彼は外
に出たいと願
っているし、

(b) 3倍で拡大表示する場合

青年 な
ぜって、
つまり、
こうい
うこと
ですよ。
わたし
の友人
に、も
う何
年も自
室にこ
もりに
つきり

(c) 4倍で拡大表示する場合

青年
なぜっ
て、つ
まり、
こうい
うこと
ですよ。
わたし
の友人
に、も
う何年

(d) 5倍で拡大表示する場合

図 12 図 2 の画像に対する再構成文書画像の例

次に, 図 2 の画像の画素数を縦横 50%にしたものを元画像として, それから 4 倍と 6 倍用の再構成画像を生成した. それらの画像を図 13 (a), (b)に示す. この結果から, 開発したプログラムは, 図 2 の画像の解像度が 50%になっても正しく処理ができた.

青年 なぜって、つ
まり、こういうこと
ですよ。わたしの友
人に、もう何年も自
室にこもりつきり
になっている男がい
ます。彼は外に出た
いと願っているし、
できることなら仕
事を持ちたいとも
思っている。いまの
自分を「変えたい」

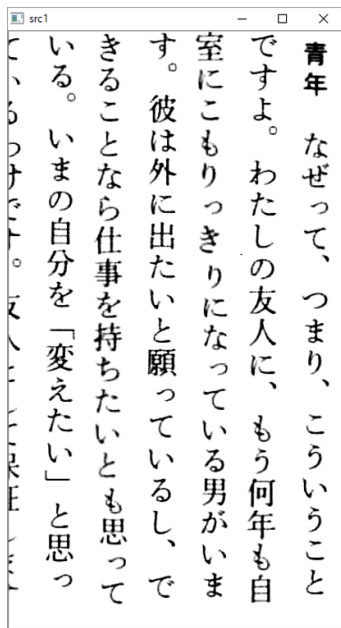
(a) 4倍で拡大表示する場合

青年 なぜつ
て、つまり、
こういうこ
とですよ。わ
たしの友人に
、もう何年も
自室にこもり
つきりになっ
ている男がい
ます。彼は外
に出たいと願
っているし、

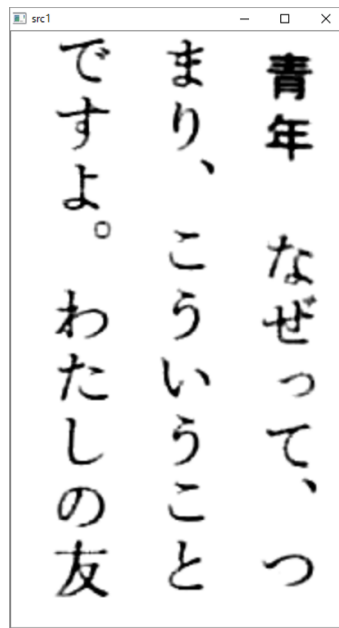
(b) 6倍で拡大表示する場合

図 13 図 2 の画像の解像度を 50%にし, 生成した再構成文書画像の例

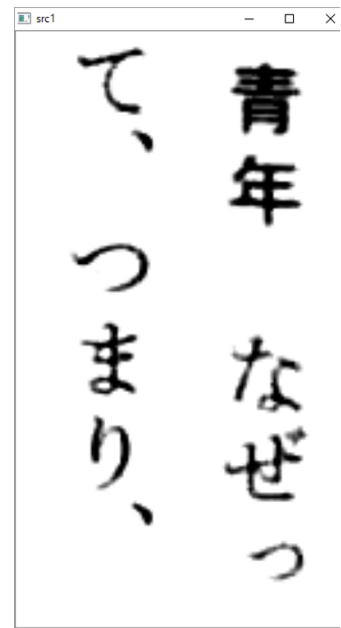
図 14 に, 図 12.1 と図 12.2 の再構成画像と倍率 1 倍の再構成画像を, iPhone6S の画面に拡大表示した設定の画像を示す.



1倍表示



2倍表示



3倍表示

図 14 倍率 1 倍, 2 倍, 3 倍の再構成画像を拡大表示した画像

4.3 開発したプログラムが正しく動作しない場合

開発したプログラムは図2の画像に対して、正しく動作した。図2の画像は、①文字列がほぼ完全に垂直方向に並んでおり、②画像全体の明るさ変化（シェーディング）が少ない、という好条件をそなえた画像である。ユーザが撮影する文書画像は、一般に、もっと条件が悪い画像になる。そのような画像に対する動作を検証した。

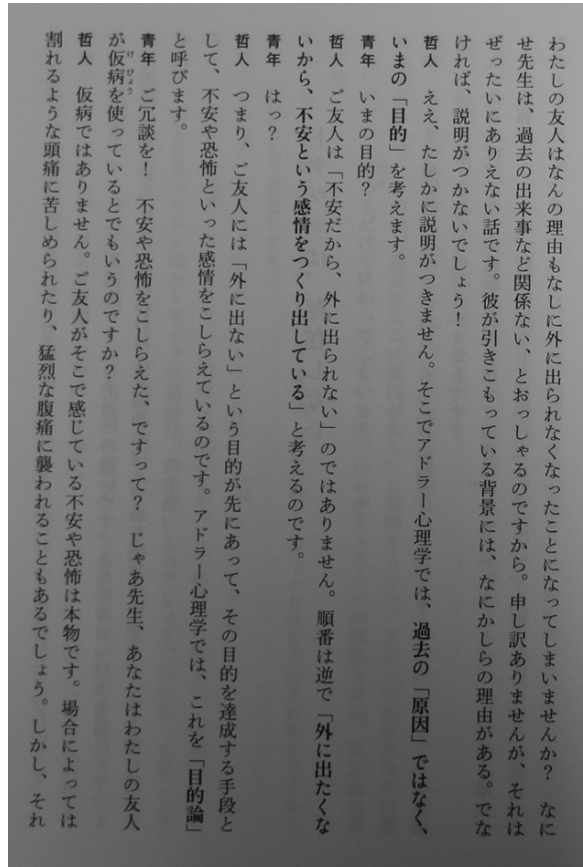


図15 少し斜めに傾いた文書画像

図15の文書画像は、図2と同じ書籍の、別のページ(p27)を撮影した文書画像である。この文書画像をプログラムに入力したが、再構成画像を作ることができなかった。この画像で問題となっているのは、以下の2点である。

1. 文字列が傾いているため、投影データから文字列を分解できない。
2. 画像の上部分と下部分で背景の明るさが異なり、画像全体を一つの閾値で2値化できない。

1の問題を解決する方法ために、文字列の傾きを補正する方法を検討した。まずエッジ抽出を行う。図15をエッジ抽出した画像が図16である。図16から、エッジ点の凸包を求

め,そこから凸包を近似する四角形を推定した(図 17). この四角形を平面射影変換により垂直な長方形に変換すると,文字列が垂直になった文書画像(図 18)を得ることができた.

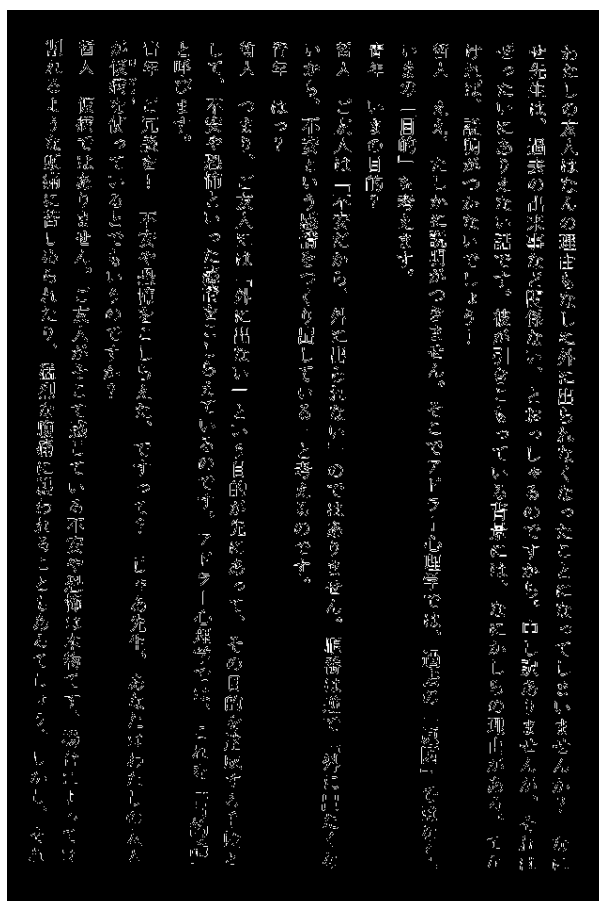


図 16 図 15 のエッジ画像

おたしの友人はなんの理由もなく外に出られなくなったことになってしまいましたか？ なにせ先生は、過去の出来事など関係ない、とおっしゃるのですから。申し訳ありませんが、それはぜったいありえない話です。彼が引きこもっている背景には、なにかしらの理由がある。でなければ、説明がつかないでしょう！

哲人 ええ、たしかに説明がつかえません。そこでアドラー心理学では、過去の「原因」ではなく、いまの「目的」を考えます。

青年 いまの目的？

哲人 ご友人は「不安だから、外に出られない」ではありません。甲斐は逆で「外に出たくないから、不安という感情をつくり出している」と考えるのです。

青年 はっ？

哲人 つまり、ご友人には「外に出ない」という目的が先にあって、その目的を達成する手段として、不安や恐怖といった感情をこしらえているのです。アドラー心理学では、これを「目的論」と呼びます。

青年 ご冗談を！ 不安や恐怖をこしらえた、ですって？

哲人 仮病ではありません。ご友人がそこで感じている不安や恐怖は本物です。場合によっては割れるような頭痛に苦しめられたり、猛烈な腹痛に襲われることもあるでしょう。しかし、それ

図 17 図 16 のエッジ画像の凸包から四角形を推定した画像

わたしの友人はなんの理由もなしに外に出られなくなったことになってしまいましたか？ なにせ先生は、過去の出来事など関係ない、とおっしゃるのですから。申し訳ありませんが、それはぜったいにありえない話です。彼が引きこもっている背景には、なにかしらの理由がある。でなければ、説明がつかないでしょう！

哲人 ええ、たしかに説明がつきません。そこでアドラー心理学では、過去の「原因」ではなく、いまの「目的」を考えます。

青年 いまの目的？

哲人 ご友人は「不安だから、外に出られない」ではありません。順番は逆で「外に出たくないから、不安という感情をつくり出している」と考えるのです。

青年 はっ？

哲人 つまり、ご友人には「外に出ない」という目的が先にあって、その目的を達成する手段として、不安や恐怖といった感情をこしらえているのです。アドラー心理学では、これを「目的論」と呼びます。

青年 ご冗談を！ 不安や恐怖をこしらえた、ですって？ じゃあ先生、あなたはわたしの友人が仮病を使っているともいいますか？

哲人 仮病ではありません。ご友人がそこで感じている不安や恐怖は本物です。場合によっては割れるような頭痛に苦しめられたり、猛烈な腹痛に襲われることもあるでしょう。しかし、それ

図 18 図 15 の文字列の傾きを修正した画像

次に、2 の問題を検討する。図 18 を、OpenCV が提供する 2 値化関数 `threshold()` を使い、引数 `cv::THRESH_BINARY|cv::THRESH_OTSU` で 2 値化したものが図 19 である。この関数は画像全体を一つの閾値で処理するため、全体的な濃淡に変化がある場合、正しく 2 値化できない。この画像の場合、画像の下部分が黒く欠損する結果になった。

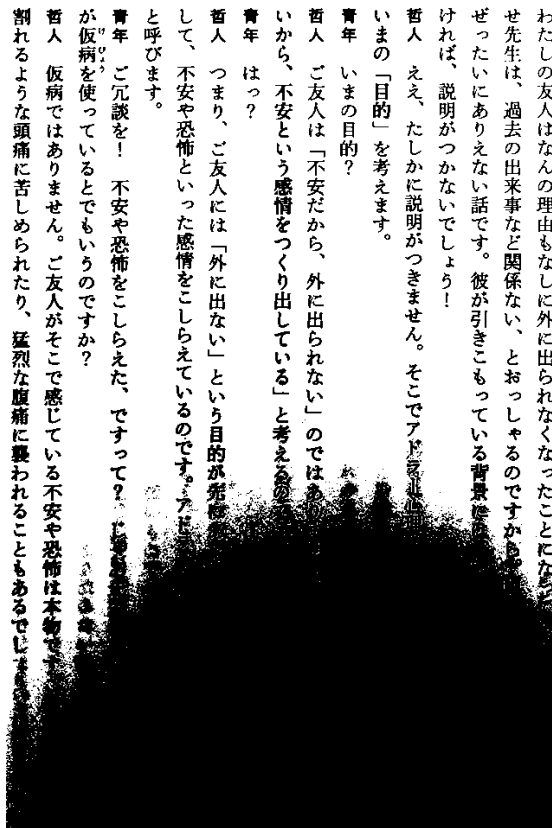


図 19 図 18 を 2 値化した画像

これを解決するために、全体を 4 領域に分けて画像を処理した。図 18 を単純に、領域 1 (左上)、領域 2 (右上)、領域 3 (左下)、領域 4 (右下) に領域分割した。4 つの領域のそれぞれのヒストグラムを図 20 に示す。それぞれの領域に分けた画像を、図 19 と同じように OpenCV が提供する 2 値化関数 `threshold()` を使い、

引数 `cv::THRESH_BINARY | cv::THRESH_OTSU` で 2 値化したものが図 21 である。やや軽減されたが、図 19 と同じく画像の下部分である領域 3、領域 4 が黒く欠損してしまう結果となった。そこで、それぞれの領域で、2 値化の閾値を次のように設定した。

領域 1: 画像の下ほど、小さくする。

領域 2: 領域 1 と同じ処理。

領域 3: 領域 1, 2 より小さくし、右側ほど小さく、かつ下側ほど小さくする。

領域 4: 領域 1, 2 より小さくし、左側ほど小さく、かつ下側ほど小さくする。

これらの設定を図 18 の画像に適用した結果が図 22 である。この画像を再構成処理した結果、図 23 のような画像が生成できた。

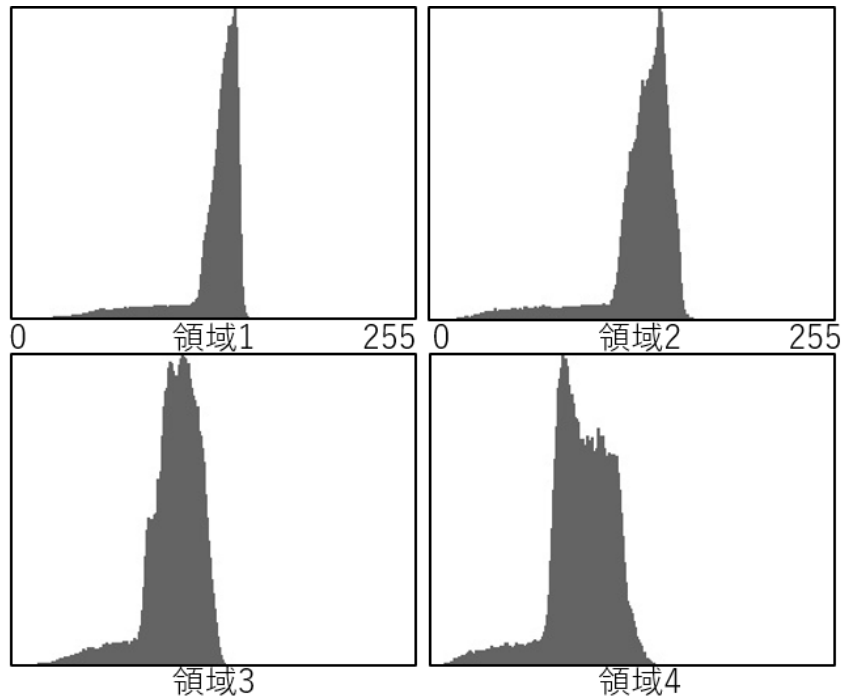


図 20 図 18 を 4 領域に分けた, それぞれのヒストグラム

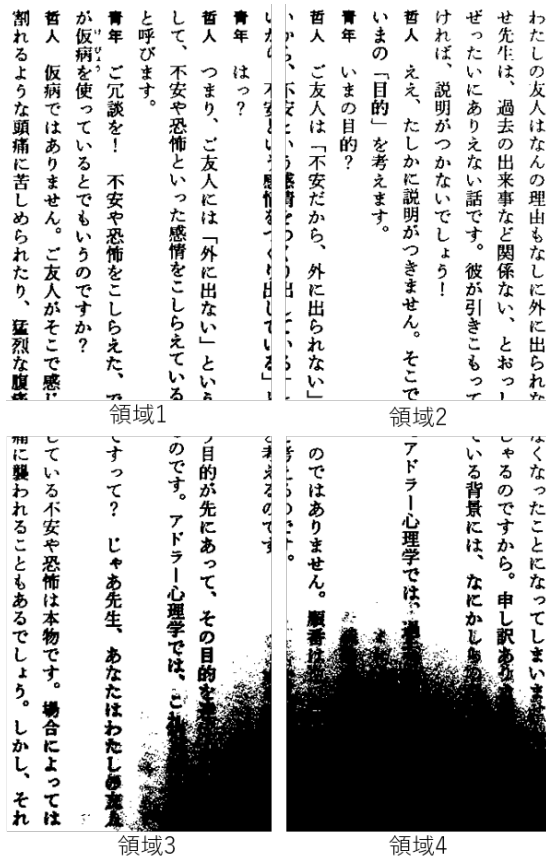


図 21 図 18 を 4 領域に分け, それぞれを 2 値化した画像

わたしの友人はなんの理由もなしに外に出られなくなったことになってしまいましたか？ なにせ先生は、過去の出来事など関係ない、とおっしゃるので。申し訳ありませんが、それはぜったいにありえない話です。彼が引きこもっている背景には、なにかしらの理由がある。でなければ、説明がつかないでしょう！

哲人 ええ、たしかに説明がつかえません。そこでアドラー心理学では、過去の「原因」ではなく、いまの「目的」を考えます。

青年 いまの目的？

哲人 ご友人は「不安だから、外に出られない」ではありません。順番は逆で「外に出たくないから、不安という感情をつくり出している」と考えるのです。

青年 はっ？

哲人 つまり、ご友人には「外に出ない」という目的が先にあって、その目的を達成する手段として、不安や恐怖といった感情をこしらえているのです。アドラー心理学では、これを「目的論」と呼びます。

青年 ご冗談を！ 不安や恐怖をこしらえた、ですって？ じゃあ先生、あなたはわたしの友人が仮病を使っているともいいますか？

哲人 仮病ではありません。ご友人がそこで感じている不安や恐怖は本物です。場合によっては割れるような頭痛に苦しめられたり、猛烈な腹痛に襲われることもあるでしょう。しかし、それ

図 22 図 18 を提案手法で 2 値化した画像

わたしの友人はなんの理由もなしに外に出られなくなったことになってしまいましたか？ なにせ先生は、過去の出来事など関係ない、とおっしゃるので。申し訳ありませんが、それはぜったいにありえない話です。彼が引きこもっている背景には、なにかしらの理由がある。でなければ、説明がつかないでしょう！

図 23 図 22 を 3 倍で拡大表示するための再構成文書画像

4.4 考察

本研究により、① 文字列がほぼ完全に垂直方向に並んでおり、② 画像全体の明るさ変化（シェーディング）が少ない、という 2 点の条件を整えた画像を、おおむね、正しく処理することができた。しかし、以下の点について改善の余地がある。

(1) 文字「こ」のように正しく認識できない文字がある。

(2) 段落末尾を正しく検出できない場合がある。

開発した技術を拡大読書器の機能として使用するには、これらの点を改善し、精度を高める必要がある。

またより本質的な課題として、以下のことがある。

(1) 上で指摘した2点の条件を満たすように文書画像を入力することは難しい。したがって、入力した文書画像を分析し前処理することで、投影処理を行いやすい画像に調節することが必要である。

(2) 今回の報告では、2枚の文書画像を使って動作を検証したが、異なる条件の画像を使用するなど、提案手法の更なる検証が必要である。

(3) 弱視者による被験者テストで、性能を検証する必要がある。

5章 結論

文字列が整った縦書きの文書画像を、投影処理によって列、および文字に分解することで、改行位置を変更した再構成文書画像を作成することができた。これにより、文書画像を拡大して読む場合に、縦方向への移動がなくなり、横方向の移動のみで読み進めることが可能となった。しかし、① 文字列がほぼ完全に垂直方向に並んでおり、② 画像全体の明るさ変化（シェーディング）が少ない、という2点に当てはまらない画像を使用した場合、投影処理が正しく行えず再構成画像を作成できなかった。

そこで、①に当てはまらない文字列が曲がった画像は、エッジ抽出をして凸包から得られた四角形から平面射影変換により垂直な長方形に変換した。②に当てはまらない全体の明るさの変化が大きい画像は、4つの領域にわけてそれぞれに処理を行い、投影処理ができる画像に変換した。

課題として、次のことが残っている。

- (1) 入力画像をプログラム上で分析し前処理を行い、投影処理を行いやすい画像に調節する必要がある。
- (2) 今回の報告で使用した2枚の画像以外のものを使用するなど、提案手法の更なる検証が必要である。
- (3) 提案手法によって得られた再構成画像によって、読書効率が改善したかどうかの調査、検証が必要である。

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 平成 20 年),
<http://www.mhlw.go.jp/toukei/saikin/hw/shintai/06/index.html>
- [2] 「教職員のための障害者生修学支援ガイド(平成 26 年度改訂版)」(日本学生支援機構 2015), pp.23.
- [3] 「日常用具付与事業の概要」(厚生労働省 2006),
<http://www.mhlw.go.jp/bunya/shougaihoken/yogu/seikatsu.html>
- [4] 「拡大読書器(CCTV)について」(アサクラメガネ),
<http://www.asakuramegane.co.jp/CCTV10.html>
- [5] 「OCR とは」(メディアドライブ株式会社),
<https://mediadrive.jp/technology/ocr.html>
- [6] 「日本語・英字活字カラーOCR ソフト「読取革命 Ver. 15」(パナソニックソリューションテクノロジー株式会社),
<http://www.panasonic.com/jp/company/pstc/products/yomikaku.html>
- [7] 「画像処理技術による拡大読書器の実現」, 青木恭太, 秋山仁, 四之宮佑馬, 村山慎二郎, 電子情報通信学会技術研究報告書. WIT, 福祉情報工学 106(57), pp.77-81, 2006 年.
- [8] 「画像処理を用いた拡大読書器の試作と評価」, 野本真広, 他, ロボティクス・メカトロニクス講演会講演概要集 2011, 2P1-D07, 2011 年.

謝辞

本論文を作成にあたり,丁寧な御指導を賜りました蚊野浩教授に感謝いたします.

付録 本研究で開発したプログラムの説明などを付録とする

[プログラム名]

drtest.cpp

[内容]

このプログラムは,入力した文書画像ファイルを文字列再配置によって再構成し,入力した倍率,指定したウィンドウサイズ上で拡大表示する.プログラムの流れは以下の通りである.2 値化処理し,縦方向への投影処理で文字列に分解する.文字列に横方向への投影を行い,文字の終端位置から段落ごとの画像にする.段落画像を横方向へ投影し,文字に分解する.入力された拡大倍率と表示するディスプレイの大きさから,文字列再配置を行った再構成画像を作成する.ウィンドウを表示し,拡大した再構成画像を表示する.「←」,「→」キーで画像の表示位置の変更,「↑」,「↓」キーで拡大縮小が可能.Esc キーが押されたら,プログラムを終了する.