

複雑な複数漢字文字を対象とした印刷耐性を有する アルティメイトリンク

山寺慧司¹ 新見道治²

概要: 我々の研究室では現実世界の意味的なまとまりとサイバー空間での情報を対応づけるアルティメイトリンク技術を提唱している。本稿では、紙面に印刷された意味的なまとまりを持った複雑な複数漢字文字を対象として、URLを埋め込むアルティメイトリンク技術の改良について報告する。既存の紙面上のアルティメイトリンクでは、処理対象の文字は、簡単な字形の文字、例えばひらがなやカタカナであり、複雑な字形を対象としていなかった。そこで、本稿では複雑な複数漢字文字に対応したアルティメイトリンクを実現するための手法を提案する。具体的には文字切りだし方法および芯線抽出方法に改良を加えた。提案手法を実装したプログラムを用いて実験を行い、結果や改善点を検討した。

キーワード: アルティメイトリンク, 情報ハイディング, 複雑な漢字文字列, 意味, サイバー空間

Ultimate Link for complex multiple kanji letters with robust for print and scan

Keiji YAMADERA¹ and Michiharu NIIMI²

Abstract: In our laboratory, we are proposing an Ultimate Link technology that maps semantic coherence in the real world to information in cyberspace. In this paper, we report on the improvement of the Ultimate Link method, which embeds URL in multiple semantically coherent letters printed on paper. In the existing Ultimate Link on paper, the target characters are simple characters, such as Hiragana and/or Katakana, and complex characters are not targeted. In this paper, we proposed a method to realize Ultimate Link for multiple letters including complex kanji. Specially, we have improved the character segmentation method and the line extraction method. Experiments were conducted using a program that implements the proposed method, and the results and improvements were discussed.

Keywords: Ultimate link, Information hiding, Complex Kanji String, Meaning, Cyber world

1. はじめに

現代社会においては、多くの人々がスマートフォンを所有し、そのカメラの用途は人物や背景の撮影だけにとどまらず、QRコード[1]の読み取りなどによるインターネットに接続するための手段の1つになっている。しかしQRコードは文字情報とは別に紙面に印刷する必要があり、印刷物のレイアウトを崩してしまうことになる。QRコードの他にも、紙面情報からコンピュータ世界内の情報を読みだすスクリーンコード[2]と呼ばれる技術が存在する。スクリーンコードは、目に見えない微細なパターンを紙面に印刷しそれを読み取る技術であるが、印刷に特殊なカラー印刷技術を用いるため、一般ユーザーが指定した情報を市販されているコンピュータ機器によって印刷することを想定していない。そこで、印刷物のレイアウトを崩さず市販されているコンピュータ機器、低品質な紙を用いて簡単に「紙面

上の文字情報」にコンピュータ空間内での「デジタルコンテンツ」をリンクすることができれば非常に便利である。

上述した紙面上の情報ハイディングの問題を解決するために「アルティメイトリンク」と呼ばれる技術を利用して印刷された文字列にURLを埋め込み、抽出を行うことでWebページを表示するシステムの提案を行っている[3]。文献[3]では、簡単な字形のカタカナ複数文字の文字列への情報の埋め込み、抽出処理は成功しているが、複雑な字形の漢字やひらがなを含む複数文字の文字列への情報の埋め込み、抽出処理は失敗している。

そこで、本研究では[3]の手法を改良することで、複雑な字形文字を複数含む場合でも、頑健に埋め込み抽出できるアルティメイトリンクを提案する。具体的には、一つの文字の切り出し方法および芯線の抽出方法に関して改良を加える。

以下本稿では、2.にて[3]の手法について説明し、3.にて改

1 九州工業大学情報工学部知能情報工学科
Department of artificial intelligence,
School of Computer Science and Systems Engineering,
Kyushu Institute of Technology
2 九州工業大学大学院情報工学研究院
Faculty of Computer Science and Systems Engineering,
Kyushu Institute of Technology

良手法について解説する。4.にて実験を行い、最後に、5.でまとめる。

2. アルティメイトリンク[3]

アルティメイトリンクは、現実世界における文字のある意味的なまとまりに対して、サイバー空間での付随/補足情報をリンクさせる技術である。具体的には、文字に対してバイナリデータを重畳し、抽出する技術である。処理は、2 値画像に対する埋め込み処理と、埋め込みを行った画像を印刷したものをスマートフォンなどのカメラで撮影したものから抽出する2つに大きく分けられる。以降では既存の印刷物に対するアルティメイトリンクを実現するための処理内容について説明する。

2.1 埋め込み処理

埋め込み処理は、以下の流れで行う。

- (1) 文字画像から埋め込み対象の文字列を文字認識する。
- (2) 文字列を1文字ずつに分割する。
- (3) 分割した文字を細線化する。
- (4) 細線化画像から擬似筆順の計算を行う。
- (5) 擬似筆順からグリッド点を求める。
- (6) 埋め込み情報に基づいてグリッド点に色のついた円を描画する。
- (7) 原画像の文字と円を描画した文字を置き換え、射影マーカーを描画する。

以下、各手順について具体的に説明する。

(1) 文字認識

情報を埋め込みたい複数文字の画像（以下、原画像）を用意する。原画像から光学文字認識エンジンを用いて文字認識をする。

(2) 文字列の分割

原画像から埋め込み対象の文字列を1文字ずつに分割し、切り出しを行う。文字の分割は横書きの場合を想定し、画像内の1文字ずつの幅と高さを得ることによって行う。幅は、画像内で黒色の画素をもつ列が連続する所を抜き出すことによって取得する。このとき、「リ」や「ハ」などのように1文字の中で空間が空いているものも存在する。これらの文字は多くの場合、1文字内で連続する画素の幅が小さいものが2つあるため、その幅を外れ値として検出することができる。そのため、1文字内で空間のある文字については隣り合う小さな幅を持つものを1つと見なすことで1文字分の幅を取得する。高さの取得についても同様に、画像内で黒色の画素をもつ行が連続する所を抜き出すことで取得する。これらによって取得した幅と高さにマージン

をもたせて切り抜くことで文字列の分割が行える。

(3) 文字の細線化

文字の芯線の上に円を描画するために細線化を行う。細線化により線幅が1の2値画像を得る。細線化画像の例を図1に示す。

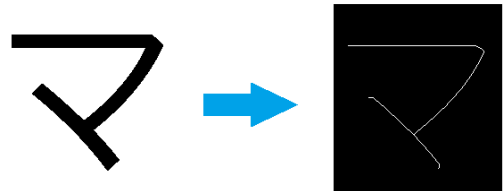


図1 細線化画像の例

Figure 1 Example of thinned image

(4) 擬似筆順の計算

擬似筆順とは、文字を細線化した画像から前景画素をたどる順番を計算したものである。この順番は輪郭線追跡法を応用したものであり、以下の手順で計算される。

手順1: ラスター走査によって前景画素の中で最初に

見つかった画素を注目画素とする。

手順2: 注目画素と隣り合う画素を反時計回りに探しだし、見つかった画素を次の注目画素とする。

手順3: 手順2を繰り返す、隣り合う画素が無くなった所で手順1に戻る。

手順4: 前景画素がすべて無くなるまで手順1から手順3を繰り返す。

上記の手順によって擬似筆順が計算される。求めた擬似筆順は、文字ごとに保存しておく。

(5) グリッド点の計算

半径 r の円を円と円の間隔を d として擬似筆順に沿って描画していく。このとき、字形や走査順によって円が重なる場合がある。このようなときにはその筆順は利用せず、描画する円が重ならないように位置を調整しながら描画可能な擬似筆順を探す。これらによって探し出された擬似筆順上の点をグリッド点と呼ぶ。グリッド点を描画した例を図2に示す。

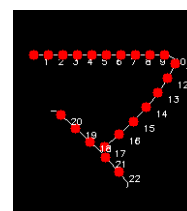


図2 グリッド点描画の例

Figure 2 Example of grid point drawing

(6) 色のついた円の描画

求めたグリッド点に対して、埋め込み情報に対応して色づけした円を描画する。円を描画する位置は、ランダムシードを固定しグリッド点の中から埋め込む情報の長さによってランダムで選ぶ。このとき、1つの円で2ビット(4通り)を表すようにし、射影マーカー用にも色が必要なため全部で5色設定する。色の設定はHSV色空間で考えると $S = 255$, $V = 255$ として、 H の値を図3のように変化させることで5色設定する。図3の設定色は上から順に、射影マーカー、埋め込み情報 00, 01, 10, 11 である。



図3 グリッド点/射影マーカーへの設定色
Figure 3 colors for grid points/projection marker

(7) 原画像への置き換えと射影マーカーの描画

色のついた円を描画した各文字を原画像の元の位置へ置き換える。さらに、抽出の際に画像を正規化するために射影マーカーを4点描画する。

情報を埋め込んだ文字画像の例を図4に示す。

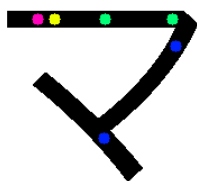


図4 埋め込み例
Figure 4 Example of embedding

情報を埋め込んだ文字は図4のようになり、文字上に埋め込み情報によって複数の色の円が描画される。

2.2 抽出処理

抽出処理は以下の流れで行う。

- (1) 射影マーカーを検出し、射影変換する。
- (2) 射影変換後の撮影画像に対象の文字列を文字認識する。
- (3) 変換後の画像で文字列の分割を行う。
- (4) 分割した文字画像から文字上の円の検出を行う。
- (5) 保存してある擬似筆順から参照グリッド点を求め、検出した円の中心と最も近い参照グリッド点を

求めることで最近円を得る。

- (6) 最近円の色に対応する埋め込んだビット列を順番に並べる。

以下、各手順について具体的に説明する。

(1) 射影変換

撮影画像を埋め込み時と同様の位置、サイズに変換するために射影変換を行う。変換を行うための基準として射影マーカーの検出する。HSV色空間であらかじめ設定した射影マーカー用の色でマスク画像を作成する。作成したマスク画像に膨張処理を行い、膨張マスク画像を得る。この膨張マスク画像からハフ変換によって円を検出する。

(2) 文字認識

射影変換後の画像から埋め込み処理時と同様の方法で文字認識する。

(3) 文字列の分割

射影変換後の画像で文字列の分割を行う。分割方法は埋め込み処理と同様の方法で行う。

(4) 文字上の円の検出

1文字ずつ切り出した画像から色のついた円の検出を行う。検出方法はHSV色空間であらかじめ設定した色ごとに射影マーカーの検出と同様の方法で行う。

(5) 最近円の検出

保存してある各文字の擬似筆順情報からグリッド点を求め、埋め込みの際に決めたランダムシードと同じものを使用して検出された円の数に応じてグリッド点からランダムに点を選ぶ。これを参照グリッド点とする。(4)で検出された円の中心座標と最も近い距離にある参照グリッド点を求める。

(6) ビット列の抽出

(5)で求めた最近円から、その色に対応するビット列を順番に並べることでビット列の抽出を行う。

2.3 追従実験

2.1, 2.2で示した既存手法を用いて文献[3]での漢字のみ、漢字とひらがな混合の複数文字の画像に対して追従実験を行った。

2.3.1 漢字のみ複数文字の場合

実験に用いた画像は以下のものである。埋め込む情報は便宜的に「abcdefghijklmn」の14文字である。

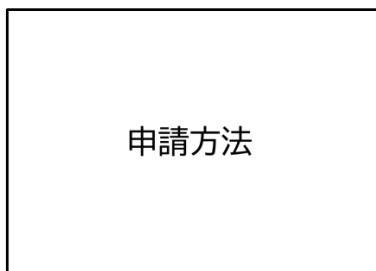


図 5 埋め込み対象画像 (漢字)

Figure 5 Image to embed (kanji)

情報を埋め込む際は、バイナリデータとして埋め込むため Ascii コード (7 ビット) に変換する。図 5 の画像に対して情報の埋め込み処理を行った後、撮影画像から情報の抽出処理を行ったところ、文字列の分割の際に「請」と「法」という字が以下のように誤って分割され、情報の抽出が正しく行われなかった。

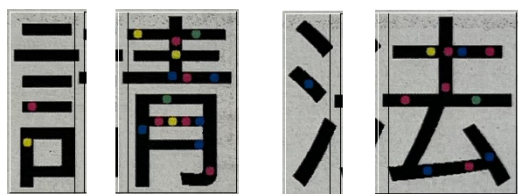


図 6 文字分割失敗例

Figure 6 Example of character segmentation failure

2.3.2 漢字とひらがな混合の場合

実験に用いた画像は以下のものである。埋め込む情報は漢字の時と同様に「abcdefghijklmn」の 14 文字である。

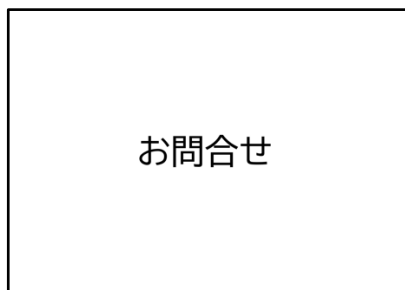


図 7 埋め込み対象画像 (漢字とひらがな)

Figure 7 Image to embed (kanji and hiragana)

図 7 の画像に対して情報の埋め込み処理を行った後、撮影画像から情報の抽出処理を行ったところ、図 8 に示すように「合」の字に青色の円が 4 つ描画されているにもかかわらず、検出された 3 つとなっている。このため情報の抽出が正しく行われなかった。

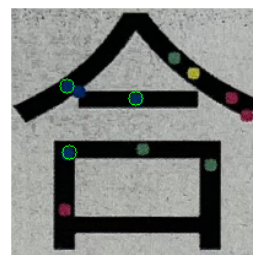


図 8 検出失敗例

Figure 8 Example of detection failure

2.4 問題点の考察

2.3.1, 2.3.2 の実験結果から漢字やひらがなを含む場合において情報の抽出が正しく行われなかった理由を考察する。図 5 の撮影画像から情報の抽出処理を行う際の文字分割が図 6 のように失敗した原因は、外れ値の検出がうまく行われなかったためである。外れ値の検出は四分位範囲を用いた手法により行っているが、その手法は図 5 のような文字数が少ない場合には、外れ値を外れ値として検出することができないことがあるためである。また、図 7 の撮影画像から情報抽出を行う際の文字上の円の検出が失敗した原因は、漢字はカタカナやひらがなに比べて形が複雑なものがあるため、描画する円の距離が近くなりすぎる場合があるためである。

3. 問題点解決のための改良手法

2.4 で述べた問題点を解決するため、既存手法の改良を行う。

3.1 新たな文字分割手法の提案

文字分割の手法を、文字認識の結果を利用した手法に変更する。例えば文字認識の結果、 n 文字認識したとすると、このとき文字列の横幅が X であるとする、 (X/n) が 1 文字分の横幅の長さであるとして、文字列の分割を行う。

ここで、2.1 の(2)で述べた黒色の画素を持つ列が連続している所を抜き出して 1 文字分の横幅の長さを取得し、取得した横幅の長さに対して外れ値検出を行い、その結果を最終的な文字列の分割結果とする手法を「既存の文字分割手法」と呼ぶ。これに対して、文字認識の結果を利用する手法を「新たな文字分割手法」と呼ぶ。

新たな文字分割手法は、以下の流れで行う。

- (1) 既存の文字分割手法により、文字列全体の横幅の長さを求める。
- (2) 文字列全体の横幅の長さを X とし、文字認識の結果 n 文字認識したとすると、1 文字分の横幅の長さ $char_length = X/n$ を得る。

(3) (2)で求めた $char_length$ を1文字分の横幅の長さとして、文字列の再分割を行う。

以下、各手順について具体的に説明する。

(1) 文字列全体の横幅の長さの取得

既存の文字分割手法により分割された文字列において、 i 文字目に分割された文字の始点の x 座標を sp_w_i 、終点の x 座標を ep_w_i とし、 i 文字目の横幅の長さを $char_w_i$ とすると、 $char_w_i$ は以下の式で求められる。

$$char_w_i = ep_w_i - sp_w_i \quad (1)$$

既存の文字分割手法によって N 文字に文字分割されたとすると、文字列全体の横幅の長さ X は以下の式で求められる。

$$X = \sum_{i=1}^N char_w_i \quad (2)$$

(2) 1文字分の横幅の長さの取得

文字認識の結果、 $n(n < N)$ 文字認識したとする。式(2)で求めた文字列全体の横幅の長さ X より、1文字分の横幅の長さ $char_length$ は、以下の式で求められる。

$$char_length = X/n \quad (3)$$

(3) 文字列の再分割

式(3)で求めた $char_length$ を1文字分の横幅の長さとして文字列の再分割を行う。新たな文字分割手法において、 i 文字目に分割された文字の始点の x 座標を sp_wr_i 、終点の x 座標を ep_wr_i とすると、文字列の再分割は以下の手順で行われる。

手順1: 1文字目の始点の x 座標 (sp_w_1) を sp_wr_1 とする。

手順2: 新たな1文字目の終点の x 座標 (ep_wr_1) は以下の式で求まる。

$$ep_wr_1 = sp_wr_1 + char_length \quad (4)$$

手順3: 2文字目以降の分割を行っていくが、以下のような場合分けを行いながら分割を行う。

手順3-1: 新たな $(i-1)(i \geq 2)$ 文字目の終点の x 座標 (ep_wr_{i-1})が既存の文字分割手法での i 文字目の始点の x 座標 (sp_w_i) よりも大きい場合は、新たな i 文字目の始点の x 座標 (sp_wr_i)を既存の文字分割手法での $(i+1)$ 文字目の始点の x 座標 (sp_wr_{i+1})に変更し、式(4)と同様の計算を行い、 i 文字目の終点の x 座標 (ep_wr_i)を求める。

手順3-2: 手順3-1以外の場合は、 i 文字目の始点の x 座標 (sp_wr_i)を sp_w_i とし、式(4)と同様の計算を行い、 i 文字目の終点の x 座標 (ep_wr_i)を求める。

手順4: 手順3を新たな文字分割手法による始点の x 座標の数が文字認識で取得した文字数になるまで繰り返す。

(1) から (3) までの処理を行うことにより、新たに文字列が再分割され正しく文字列の分割を行うことができる。

3.2 新たな文字分割手法の処理の様子

3.1で述べた新たな文字分割手法による文字分割の様子を具体例を用いて説明する。2.3.1で述べたように文字分割が正しく行われなかった「申請方法」を例として説明する。既存の文字分割手法による文字列の分割結果は、図9のようになっている。

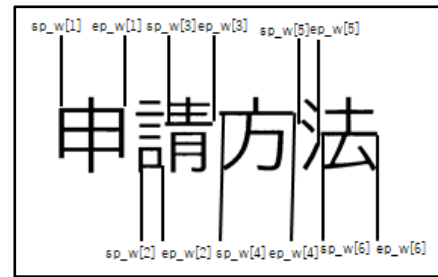


図9 既存の文字分割手法による分割結果

Figure 9 Segmentation results from existing character segmentation methods

まず、1文字目は手順1, 2を行う。図9は以下のように更新される。

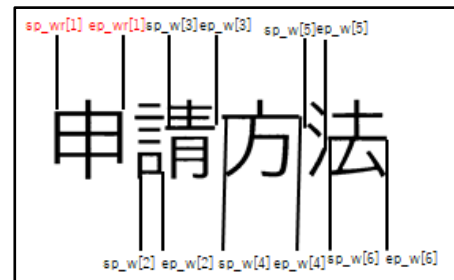


図10 新たな文字分割の様子 (1)

Figure 10 New character segmentation (1)

新たな1文字目の終点 (ep_wr_1)と既存の文字分割手法での2文字目の始点 (sp_w_2)を比べると、既存の文字分割手法での2文字目の始点のほうが大きいので、手順3-2を行う。図10は以下のように更新される。

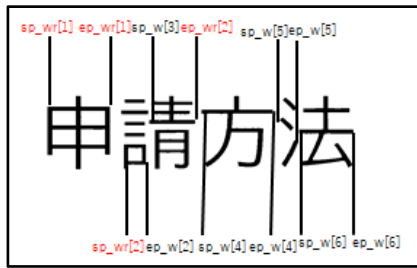


図 11 新たな文字分割の様子 (2)

Figure 11 New character segmentation (2)

新たな 2 文字目の終点 (ep_{wr_2}) と既存の文字分割手法での 3 文字目の始点 (sp_{w_3}) を比べると、新たな 2 文字目の終点のほうが大きいので、手順 3-1 を行う。図 11 は以下のように更新される。

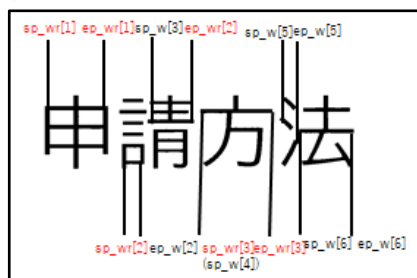


図 12 新たな文字分割の様子 (3)

Figure 12 New character segmentation (3)

新たな 3 文字目の終点 (ep_{wr_3}) と既存の文字分割手法での 4 文字目の始点 (sp_{w_4}) を比べると、新たな 3 文字目の終点のほうが大きいので、手順 3-1 を行う。図 12 は以下のように更新される。

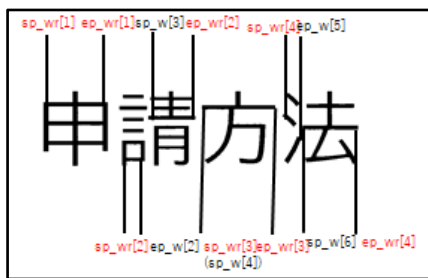


図 13 新たな文字分割の様子 (4)

Figure 13 New character segmentation (4)

最終的な文字分割の結果は、以下ようになる。

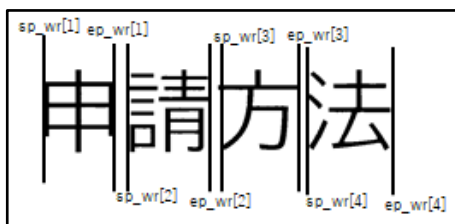


図 14 新たな文字分割手法による分割結果

Figure 14 Segmentation results using the newer one

図 14 を見て分かる通り、正しく文字列の分割が行われていることが確認できる。

3.3 文字の細線化処理前での文字の収縮処理

情報の埋め込み処理の際の文字の細線化画像生成前に、色のついた円を描画する際に円と円の距離が近くなりすぎるのを防ぐため、文字の収縮処理を行う。以下に、文字の収縮処理を行った例を示す。

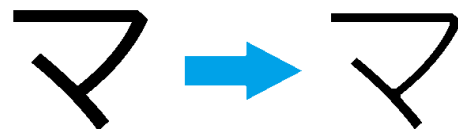


図 15 収縮処理例

Figure 15 Example of erosion

4. 改良手法による実験

3 節で述べた改良手法を用いて漢字やひらがなを含んだ字形が複雑な文字画像に対して情報の埋め込み、抽出の実験を行った。

4.1 漢字のみ複数文字の実験

2.3.1 と同様に「申請方法」の文字画像を用いて実験を行う。埋め込む情報も同様に「abcdefghijklmn」の 14 文字である。



図 16 埋め込み対象画像 (漢字)

Figure 16 Image for embedding (kanji)

図 16 の画像に情報の埋め込み処理を行った後、撮影画像から情報の抽出を行う。撮影画像を射影変換し、文字認識を行った後に新たな文字分割手法により文字列の分割を行った画像を以下に示す。

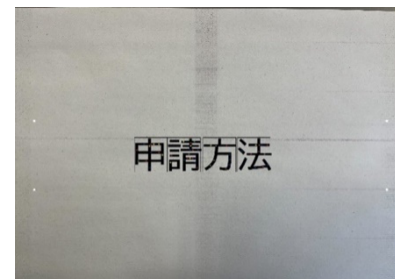


図 17 文字分割結果 (漢字)

Figure 17 Character segmentation result (kanji)

図 17 の短形領域の切り取り結果は図 18 の通りである。

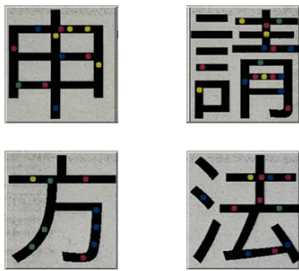


図 18 短形領域の切り取り画像
Figure 18 Cut out image of short-form area

図 18 を見て分かる通り、正しく文字列の分割が行われている。これにより、図 17 から正しく情報の抽出が行え、漢字のみの場合で情報の抽出が行えることを確認した。

4.2 漢字とひらがな混合の複数文字の実験

2.3.2 と同様に「お問合せ」の文字画像を用いて実験を行う。埋め込む情報も同様に「abcdefghijklmn」の 14 文字である。



図 19 埋め込み対象画像 (漢字とひらがな)
Figure 19 Image for embedding (kanji and hiragana)

図 19 の画像に情報の埋め込み処理を行う。文字認識、文字列の分割を行っていき、細線化処理を行う前に 3.3 で述べた文字の収縮処理を行う。収縮処理を行った画像に対して細線化処理、擬似筆順計算、グリッド点算出、色のついた円の描画を行っていく。以下に収縮処理を行わずに円を描画した「合」の画像と、収縮処理を行ってから円を描画した「合」の画像を示す。

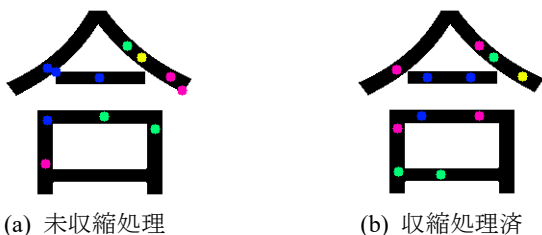


図 20 円描画画像の比較
Figure 20 Comparison of circle drawing

図 20 の(a)を見ると、情報の抽出処理で青色の円の検出に失敗する原因となった円と円の距離が近すぎるといった問題が発生している。一方、図 20 の(b)を見ると、円と円の距離が近すぎるといった問題は解決していることが分かる。

図 19 の画像に情報の埋め込み処理を行い、A4 用紙に印刷し、撮影した画像を以下に示す。

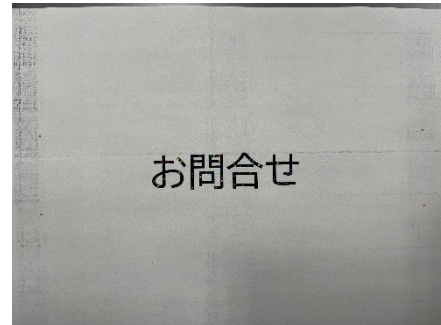


図 21 撮影画像 (漢字とひらがな)
Figure 21 Captured image (kanji and hiragana)

図 21 から正しく情報の抽出が行え、漢字とひらがな混合の場合において情報の抽出が行えることを確認した。

4.3 その他の文字画像での追加実験

漢字のみの複数文字において、画数が少ないものと画数が多いものの 2 つの場合で実験を行った。

4.3.1 画数が少ない漢字複数文字の実験

図 22 に利用した画数の少ない文字の原画像を、図 23 に撮影画像を示す。なお埋め込み情報は「abcdefghijklmn」の 14 文字である。

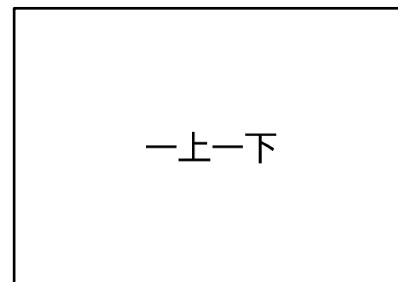


図 22 埋め込み対象画像 (画数が少ないもの)
Figure 22 Image for embedding (a few strokes)

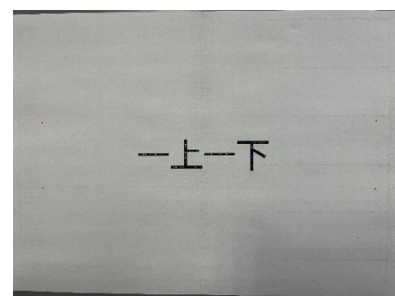


図 23 撮影画像 (画数が少ないもの)
Figure 23 Captured image (a few strokes)

図 23 から正しく情報の抽出が行え、画数が少ない漢字複数文字の場合において情報の抽出が行えることを確認した。

4.3.2 画数が多い漢字複数文字の実験

実験に用いた画像は以下のものである。埋め込む情報は「abcdefghijklmn」の 14 文字である。



図 24 埋め込み対象画像 (画数が多いもの)
Figure 24 Image for embedding (many strokes)

図 24 に情報の埋め込み処理を行った後に A4 用紙に印刷し、撮影を行った画像を以下に示す。

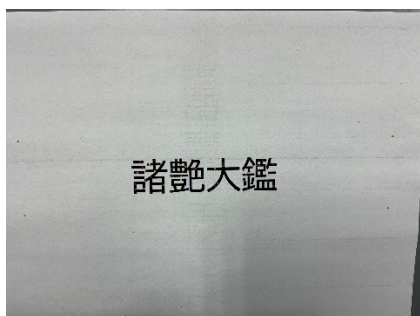


図 25 撮影画像 (画数が多いもの)
Figure 25 Captured image (many strokes)

図 25 から正しく情報の抽出は行えたが、埋め込み処理の際に収縮処理を強め、抽出処理の際の文字認識時に画像をリサイズし、文字認識エンジンのパラメータを調整することにより成功したので、画数が多い漢字複数文字の場合は、再現性が担保されていない結果となった。

5. 考察とまとめ

本稿では、漢字やひらがなを含む場合に対応する印刷耐性を有するアルティメイトリンクを提案した。新たな文字分割手法により、既存の文字分割手法では失敗していた文字列に対しても正しく文字列の分割が行えるようになった。また、情報の埋め込み処理の際の細線化前に文字の収縮処理を行うことで、円と円の距離が近くなりすぎて円の検出に失敗するという問題を解決することができた。しかし、画数が多い漢字複数文字の場合は、各種パラメータの調整に試行錯誤が必要であった。今後は画数が多い漢字複数文字の場合に再現性が担保できるようなパラメータの探索が

必要である。さらに、今回の実験では背景が白の紙面に印刷した文字に対して実験を行ったため、ポスターなどの背景に色のついた掲示物に対しても実験を行う必要がある。また、今回の実験では撮影する照明条件が蛍光灯の下のみでの単一の条件下であったため、太陽光の下や暗室など、複数の照明条件下で実験を行う必要がある。

今回の実験に用いた手法では、スマートフォンのカメラで撮影した画像を一度 PC に取り込み処理を行った。今後の実用性を考えるとスマートフォン内で情報の抽出処理のシステムが完結し、Web ページを表示できるようなシステムを構築し、実用性について評価を行う必要があると考えている。

謝辞 本研究は JSPS 科研費 JP19K12879 の助成を受けたものです。

参考文献

- [1] QR コード, <https://www.denso-wave.com/ja/system/qr/fundamental/>, 参照 Jan.31.2023
- [2] スクリーンコード, <https://www.apollo-japan.ne.jp/technology/screencode/>, 参照 Jan.31.2023
- [3] 牟田賢弘, 新見道治, 山内幸治, "意味的なまとまりを持った複数文字を対象とした印刷耐性を有するアルティメイトリンク", 信学技報, vol. 121, no. 417, EMM2021-108, pp. 86-90, 2022 年 3 月.