最短経路の収束を利用した文字切り出し方式の提案

田中　瑛一†

† 富士ゼロックス株式会社　〒 259–0157 神奈川県足柄上群中井町境 430 グリーンテクなかい
E-mail: †eiichi.tanaka@fujixerox.co.jp

あらまし　文字切り出しとは，文字認識システムにおいて，単文字どうしの境界を検出する処理である．文字切り出しの先行方式の大部分は投影分布と連結成分に基づく．しかし，投影分布に基づく方式は，単文字どうしの入り込みの切り出しが困難であるという欠点を持つ．また，連結成分に基づく方式は，単文字どうしの接触の切り出しが困難であるという欠点を持つ．また，連結成分の形状に基づく方式は，前処理である 2 値化によるノイズとストロークの太さ，および，文字の飾りの変動に対して不安定であるという欠点を持つ．そこで本稿は，前述の問題を解決するため，最短経路のみで構成される文字切り出し方式を提案する．これまで，最短経路を利用する文字切り出しの方式が提案されているが，これらは投影分布と連結成分に基づいて最短経路の始点を検出する．これに対して提案方式は，異なる始点を持つ複数の最短経路が同一の終点を持つこと利用して始点を検出する．最短経路が，パタンどうしの境界に集中する性質を持つことから，提案方式は，接触と入り込みを切り出す．評価の結果，先行方式に対して提案方式が，*Recall* において，日本語において 3.21%優位，英語において 3.94%優位であった．

キーワード　文字認識システム, 文字切り出し, 過分離, 最短経路, 最短経路の収束

1. はじめに

文字切り出しとは，文字認識システムにおいて，単文字どうしの境界の位置と形状を検出する処理である．

文字切り出しにおける代表的な課題として，単文字どうしの接触と入り込みの境界の検出が挙げられる [1]．これまで，多くの文字切り出し方式が提案されており [1], [2]，その大部分は，投影分布と連結成分に基づく．しかし，投影分布は，接触の境界を検出するが，入り込みの境界を検出しない．また，連結成分は，入り込みの境界を検出するが，接触の境界を検出しない．また，投影分布と連結成分の組み合わせによっても，投影分布が明確に特徴を示さない接触の境界の検出が困難である．

そこで，一定の文字間隔と文字幅を想定する方式がある．例えば，文献 [3] に示される方式は，文字間隔と文字幅の和が一定であると想定する．また，文献 [4] に示される方式は，文字幅が一定であり，文字列の高さと等しいと想定する．しかし，これらの方式は，プロポーショナルフォントと手書きの文字列のように，前述の想定が成り立たない対象において，境界の検出が困難である．

また，連結成分の形状に基づいて接触の境界を検出する方式がある．例えば，文献 [5] に示される方式は，連結成分を細線化し，連結成分の交点部分を検出する．また，文献 [6] に示される方式は，輪郭追跡によって，連結成分の凹部分を検出する．また，文献 [7] に示される方式は，背景領域の細線化によって，連結成分の凹部分を検出する．これらは，検出された特徴部分が接触であるとして，連結成分を分離する．しかし，これらの方式は，前処理である 2 値化と，文字の飾りの変動に対して不安定であるという欠点を持つ．これは，連結成分が前処理として入力画像の 2 値化を想定するためである．なお，2 値化にはスキャンの条件も含まれるものとする．文献 [5], [7] における細線化と，文献 [6] における輪郭追跡においては，所望の結果を得るために，2 値化によるノイズとストロークの太さの傾向を予め把握する必要がある．また，セリフフォントや手書き文字における，文字の飾りについても同様のことがいえる．

そこで本稿は，最短経路のみで構成される文字切り出し方式を提案する．提案方式は，接触と入り込みの境界を検出し，文字間隔と文字幅に非依存であり，多値画像に対応可能であるため前処理に 2 値化を想定しない．

最短経路は，経路上の画素の濃度値の和が最小となるように，曲線の経路を形成する．これは，接触と入り込みの境界を検出する目的から，有効な性質といえる．これまで，最短経路によって境界の形状を決定する方式 [8], [9] が提案されている．しかし，これらの方式は，境界の形状を最短経路によって決定するが，最短経路の始点を投影分布と連結成分に基づいて検出する．このため，投影分布と連結成分の欠点を継承している．これに対して提案方式は，最短経路の収束を利用して，始点を検出する．収束とは，複数の異なる始点を持つ最短経路が，同一の終点を持つことを指す．

以下では，まず，2. 章にて本稿が想定する文字認識システムと文字切り出しの性能要件を説明する．次に，3. 章にて本稿における最短経路の定義を説明する．次に，4. 章にて提案方式を説明する．次に，5. 章にて提案方式の文字切り出し方式としての性能の評価結果を示す．最後に，6. 章にて本稿をまとめる．

2. 過分離指向の文字切り出し

本稿では，文献 [10], [11] に示されるような，過分離指向の文字切り出しを行う文字認識システムを想定する．

以下，過分離指向の文字認識システムの処理の流れを説明する．図 1 に，説明のための模式図を示す．

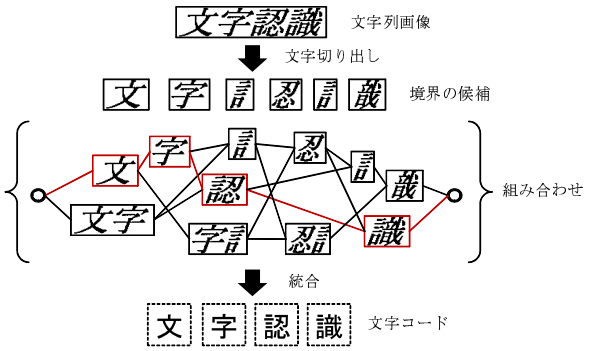


図 1　過分離指向である文字認識システム

いま，文字列画像の認識を考える．まず，文字切り出しが境界の候補を検出する．このとき，文字切り出しが過分離指向であるため，誤った境界が検出される場合がある．図 1 に示す例では，過分離指向のために「認」と「識」の偏と旁を分離している．

このため，境界の候補の組み合わせを多重に仮説し，仮説の中から，最適な組み合わせを選択する必要がある．これを本稿では統合処理とよぶ．統合処理では，まず，境界の組み合わせによって生成される領域に対して，文字識別を行い，文字コードとその識別スコアを与える．また，文字コードに基づいて，自然言語処理によってスコアを与える．最後に，前述のスコアに基づいて，最適な境界の組み合わせを決定する．これにより，誤った境界が棄却され，文字列画像が正しく認識される．すなわち，文字切り出しの *Precision* は，統合処理によって修正されるため，必ずしも文字認識システムの認識精度に直結しない．ただし，*Precision* の不足によって，統合処理が修正可能な範囲をこえる誤りが発生すること，また，境界の組み合わせの数が増加し，誤った境界の組み合わせを出力する割合が高まること，さらには，統合処理の計算負荷が増加することがある．

しかし，文字切り出しが正しい境界を検出していない場合，文字列画像は正しく認識されない．これは，一般に，文字認識システムの文字識別は単文字の識別を想定するためである．すなわち，文字切り出しの *Recall* は，文字認識システムの認識精度のボトルネックとなる．

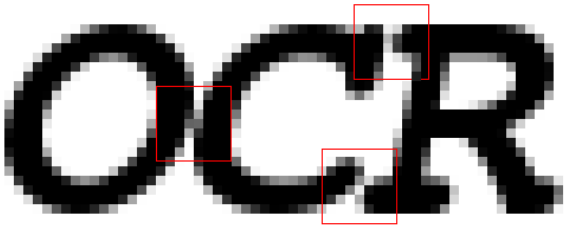
以上のことから，過分離指向の文字切り出しの性能要件は，統合処理が修正可能な範囲の *Precision* において，より高い *Recall* を示すことといえる．

3. 最 短 経 路

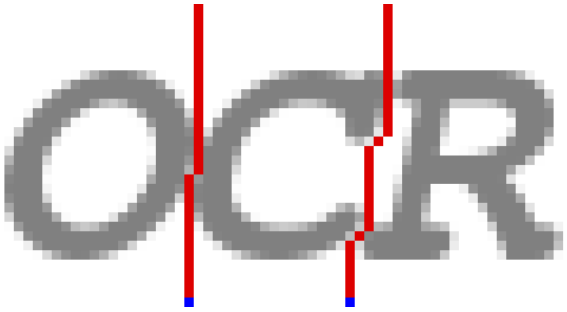
本稿における最短経路を定義する．本章で説明する最短経路は，先行方式 [8], [9] と同様のものである．

いま，対象の画像について，それぞれの画素を頂点とし，画素間の移動を辺とするグラフを考える．このグラフにおいて，辺で結ばれる頂点のセットを経路とする．最短経路とは，ある始点を持つ複数の経路のうち，経路上の画素の濃度値の和が最小の経路とする．

図 2 に，最短経路の例を示す．図 2(a) は入力画像である．赤枠部分において，接触と入り込みがみられる．図 2(b) は図 2(a) に対して．最短経路を解析した結果である．図 2(b) において，青点が始点を表す．また，赤線が最短経路を表す．最短経路が，接触と入り込みの境界を正しく検出している．



(a) 入力画像



(b) 最短経路

図 2　最短経路による文字切り出しの例

以下，最短経路の一般的な算出法を説明する．説明のための模式図を図 3 に示す．また，そのアルゴリズムを処理別に，Algorithm1,Algorithm2 に示す．

なお，各値について，それぞれ，以下の通りである．

$img(x,y)$	：	位置 (x,y) における，濃度値
$Width$	：	画像幅（文字列の方向）
$Height$	：	画像高さ（経路の進行方向）
$f(x,y)$	：	経路の和情報
$\phi(x,y)$	：	経路の移動情報
$path(y)$	：	経路 $path$ の，高さ y における位置
s	：	始点

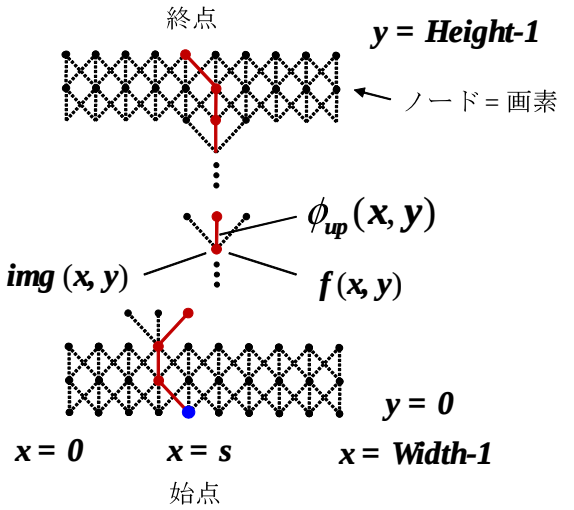


図 3　最短経路とその解析（上向き）

Algorithm 1 最短経路解析

for all x **such that** $0 \leqq x \leqq Width - 1$ **do**
  $f(x, Height - 1) \leftarrow img(x, Height - 1)$
end for
for $y \leftarrow Height - 1$ **to** 0 **do**
  for all x **such that** $0 \leqq x \leqq Width - 1$ **do**
    $f(x, y - 1) \leftarrow f(x + i, y) + img(x, y - 1)$
    $\phi_{up}(x, y - 1) \leftarrow i$
    where ... $i = \underset{-1 \leqq i \leqq +1}{arg\ min}\, f(x + i, y)$
  end for
  $y \leftarrow y - 1$
end for
return ϕ_{up}

Algorithm 2 最短経路作成

$path(0) \leftarrow s$
for $y \leftarrow 0$ **to** $Height - 2$ **do**
  $path(y + 1) \leftarrow path(y) + \phi_{up}(path(y))$
  $y \leftarrow y + 1$
end for
return $path$

まず，Algorithm1 によって，経路の移動情報である ϕ を算出する．Algorithm1 は，画像上の最短経路を，後方帰納的に求める一般的なアルゴリズムのひとつである．y 層目において，経路上の画素の濃度値の和である $f(x + i, y)$ が最小である i を進行方向として $\phi(x, y - 1)$ に保持する．進行方向 i が決定に従って $f(x + i, y)$ に $img(x + i, y - 1)$ を加算する．最短経路の進行方向と逆の方向に画像を走査するよう，以上の処理を繰り返す．

次に，Algorithm2 によって，ϕ を参照して，始点 s を持つ最短経路 $path$ を算出する．Algorithm2 は，$Height - 1$ 回の加算である．なお，図 3 において，実線で結ばれる経路が最短経路を表す．

なお，以上の説明は，上向きの最短経路を求める場合である．画像を分離する目的から，始点と終点を画像上の向かい合う辺の上に持つものとする．グラフは，画素を頂点，画素間の移動を辺として，画像の高さ方向に層を持ち，1 層の移動において，幅方向に ±1px の移動範囲を持つ．また，入力画像は多値（2 値を含む）である．

4.　最短経路の収束による文字切り出し

最短経路は，パタン（本稿では，文字）を成す画素を避けるよう，曲線の経路を形成する．これは，入り込みの境界を検出する目的から，有効な性質といえる．また，パタンを成す画素を避けられない場合，その濃度値の和が最小となるように経路を形成する．これは，接触の境界を検出する目的から，有効な性質といえる．さらに，多値画像に対して算出可能であるため，2 値化の前処理を想定しないという利点を持つ．

しかし，最短経路の性質が有効に活用されるか否かは，その始点 s に大きく依存する．最短経路を利用する先行方式 [8], [9] は，投影分布と連結成分に基づいて最短経路の始点を検出する．このため，先行方式は，投影分布と連結成分から検出されない位置にある境界を検出しない，という欠点があった．

この問題を解決するため，本稿は，最短経路の収束を利用して，最短経路の始点を検出する方式を提案する．これにより提案方式は最短経路のみで構成され，文字間隔と文字幅に非依存であり，多値画像に対応可能という利点を持つ．以下，提案方式を説明する．

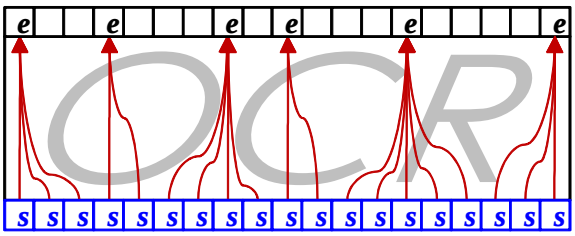


図 4　最短経路の収束

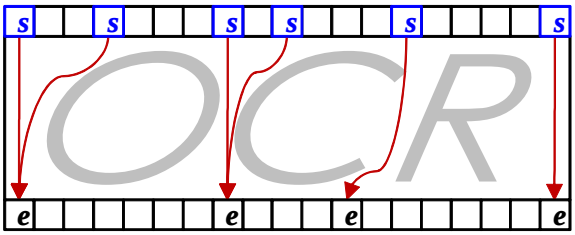


図 5　往復による，最短経路の検出

提案方式の基本アイデアは，異なる始点を持つ複数の最短経路が，同一の終点を持つことの利用である．

まず，画像の一辺のすべての画素を始点として，最短経路を求める．このとき，異なる始点を持つ複数の最短

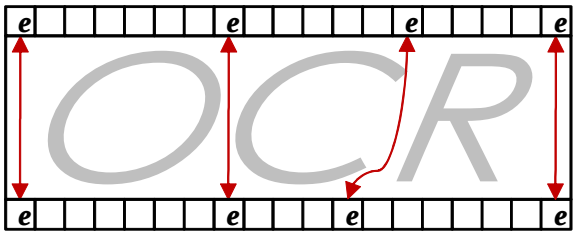


図 6　完全に収束した最短経路

経路が合流し，同一の終点を持つことがある．本稿ではこのことを，最短経路の収束とよぶ．なお，合流しない単一の最短経路は，既に収束しているとみなす．この様子を図 4 に示す．図 4 において，矢印が最短経路を表す．また，s は始点，e は終点を表す．

次に，前述の最短経路の収束により得られた終点を始点として，逆方向に最短経路を求める．このことを本稿では，往復とよぶ．この様子を図 5 に示す．

最短経路には，合流がありうるが，分流がない．よって，往復回数に対して，検出される始点の総数は単調減少となる．最後に，往復に対して始点の総数が不変となる状態となる．本稿ではこのことを，完全な収束とよぶ．この様子を図 6 に示す．

提案方式は，以上のように収束によって得られた始点を持つ最短経路を，境界の候補とする．

検出される最短経路の総数は，文字列の上部と下部の余白の大きさに対して，単調減少である．これは，余白が大きいほど，最短経路の移動範囲が広まり，合流の機会が増加するためである．このとき，文字列の下部の余白は上向きの最短経路のみに影響する．反対に，文字列の上部の余白は下向きの最短経路のみに影響する，この様子を図 7 に示す．図 7 において，下辺の青線が始点のセットを表す．また，赤領域が最短経路を表す．

文字列画像の余白を推定することで，7(a) に示すような，最短経路の過剰な回り込みによる誤りは抑止される．過分離指向の文字認識システムにおいては，図 7(b) から図 7(c) の設定が望ましい．このためには，文字列の下部の余白を $blank_{bottom}$ として，Algorithm1 において，$0 \leqq y \leqq blank_{bottom}$ の区間について $\phi_{up}(x, y) \leftarrow 0$ とすればよい．

(a) 余白 ← 大　　(b) 余白 ← 中　　(c) 余白 ← 小

図 7　余白（文字列の下部）による収束の調節

図 8 に，提案方式の構成を示す．$blank_{\{bottom,top\}}$ は文字列の上部と下部の余白である．$turn$ は往復回数である．S は始点のセットである．P は経路のセットである．

まず，Algorithm1 によって上向きと下向きの経路の移動情報 $\phi_{\{up,down\}}$ を作成する．次に，図 4, 5, 6 に示すように，最短経路の収束によって始点のセット S を作成する．最後に，Algorithm2 によって S に含まれる始点について経路を作成し，経路のセット P を出力する．

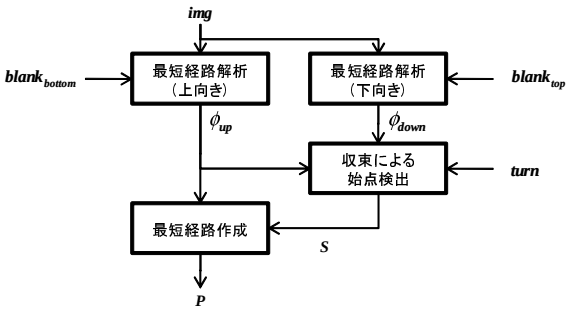
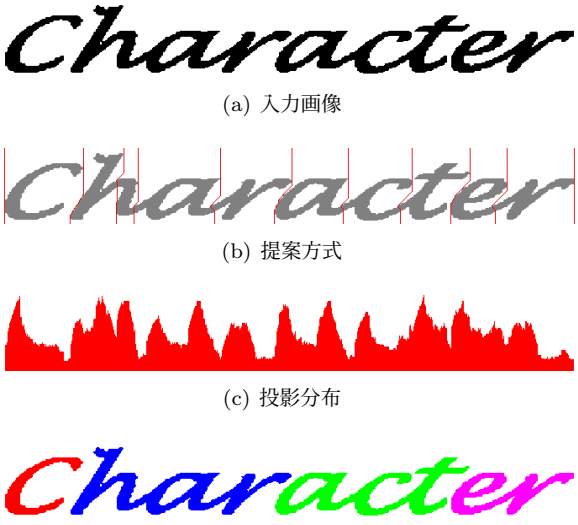


図 8　提案方式：最短経路の収束による文字切り出し

図 9 に，提案方式の処理例を示す．図 9(b) が，図 9(a) に対する，提案方式の出力例である．検出された最短経路を赤線で示す．また，図 9(c) に投影分布を，図 9(d) に連結成分を示す．図 9(c) に示す例では，「$cter$」の境界の検出が困難といえる．図 9(d) に示す例では，「har」，「act」および「er」に境界の検出が困難といえる．

これに対して，提案方式が，接触と入り込み文字の境界を正しく検出することが分かる．このように，収束を利用して始点を検出することで，最短経路がパタンどうしの境界に集中する性質が活用される．



(a) 入力画像

(b) 提案方式

(c) 投影分布

(d) 連結成分

図 9　処　理　例

なお，提案方式は，$f(x-1, y) = f(x, y) = f(x+1, y)$ であるとき，$i \leftarrow 0$ とする．また，$f(x-1, y) = f(x+1, y) < f(x, y)$ であるとき，上向きの経路においては $i \leftarrow 1$，下向きの経路においては $i \leftarrow -1$ とする．また，完全な収束において，上向きと下向きの経路の始点と終点は一致

するが，途中の経路は必ずしも一致しない．このとき，図 8 の構成においては，上向きの経路を出力する．

5. 評価実験

5.1 評価用テストチャート

評価に用いたチャートは，活字の文字列をプリントし，さらにスキャンして得られたビットマップ画像である．ひとつのチャートには，ひとつの水平な文字列がある．また，言語は日本語と英語の 2 種がある．また，それぞれの言語について，文章内容が 5 種ある．

さらに，画像処理の観点から，スキャン色深さ，スキャン解像度，コピー劣化，文字サイズ，文字形状，フォント種，フォント効果をパラメータとして，その値が異なる組み合わせを 6 種作成した．評価チャートのパラメータの組み合せを表 1 に示す．なお，フォント種は，いずれもプロポーショナルフォントである．以上のパラメータの組み合わせにより，チャートである文字列画像の総数は 60（= 言語 2 種 × 文書 5 種 × 組み合わせ 6 種）となった．なお，接触と入り込みに対する性能をみる目的から，文字間隔を狭めに設定した．

なお，日本語のセットにおいて，検出すべき境界の数は 624 である．また，英語のセットにおいて，検出すべき境界の数は 956 である．

チャートの例を図 10 に示す．それぞれ，10(a) は，日本語の，組み合わせ *No.*2 のチャートである．10(b) は，英語の，組み合わせ *No.*4 のチャートである．

5.2 評価結果

文字切り出し方式の評価指標として *Recall* と *Precision* の 2 つを示す．

Recall とは，真の境界のうち，評価対象の文字切り出しが，どれだけの真の境界を検出したか，を示す割合である．*Precision* とは，評価対象の文字切り出しが検出した境界のうち，どれだけが真の境界か，を示す割合である．それぞれ，式 1 と式 2 のように算出する．

$$Recall = \frac{\text{検出された真の境界数}}{\text{真の境界数}} \quad (1)$$

$$Precision = \frac{\text{検出された真の境界数}}{\text{検出された境界数}} \quad (2)$$

比較対象の先行方式は, [8] を採用した．これは，提案方式と同様に，先行方式 [8] が文字間隔と文字幅に非依存であり，多値画像に対応可能であるためである．先行方式 [8] は，最短経路を利用する．ただし，その始点を投影分布と連結成分に基づいて検出する．また，連結成分を画像の等高線特徴 [12] に基づいて求めることで，多値画像に対応する．

Recall の比較に先立って，両方式の性能調節の平等を図るため，*Precision* が同等となるよう，両方式を調節した．調節用のチャートのセットは，評価用のチャートのセットに対して，さらに，文字間隔の値を変化させたセットを用いた．

表 2　評価結果

評価指標	*Recall*		*Precision*	
チャートの言語	日	英	日	英
提案方式	94.55 %	93.23 %	61.71 %	77.43 %
先行方式 [8]	91.34 %	89.29 %	49.09 %	62.78 %

表 2 に評価結果を示す．評価の結果，*Recall* と *Precision* について，ともに提案方式が優位であった．

特に *Recall* において，提案方式が優位であったことの主要因は，接触に対する境界の検出である．具体例を図 11,12,13 に示す．それぞれ，図 11 は，入力画像，図 12 は，先行方式 [8] による処理結果，図 13 は，提案方式による処理結果である．

図 11,12,13 の例では，接触と入り込みのため，先行方式 [8] が正しく検出しない境界がある．一方，提案方式はこれを正しく検出している．これは，最短経路の収束が，投影分布と連結成分に対して，境界の位置をより正しく検出したためである．

(a)　(b)

図 11　入力画像（接触のある文字列）

(a)　(b)

図 12　処理例：先行方式 [8]

(a)　(b)

図 13　処理例：提案方式

提案方式が，境界を正しく検出しない誤りの例を図 14 に示す．図 14(a) における,「06」,「文字」と「認識」に誤りがみられる．この誤りは，接触のため，真の境界が最短経路では求まらないために生じている．また，図 14(b) における,「ピュ」の半濁点と「が」の濁点に誤りがみられる．この誤りは，入り込みのため，始点が正しく検出されないために生じている．なお，これらは先行方式 [8] においても同様の誤りがみられる．

表 1　評価チャートのパラメータ組み合わせ

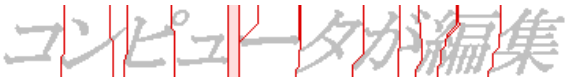
	スキャン色深さ [値]	スキャン解像度 [dpi]	コピー劣化	文字サイズ [pt]	文字形状	フォント種	フォント効果
*No.*1	2	200	かすれ	12	縦長	ゴシック体	なし
*No.*2	2	300	つぶれ	20	縦長	明朝体	斜体
*No.*3	2	400	なし	20	標準	行書体	なし
*No.*4	256	200	つぶれ	16	標準	ゴシック体	太字
*No.*5	256	300	かすれ	16	横長	行書体	斜体
*No.*6	256	400	なし	12	横長	明朝体	太字

(a) 日本語，組み合わせ *No.*2

(b) 英語，組み合わせ *No.*4

図 10　評価用チャートの例

(a) 接触による，検出の誤り



(b) 入り込みによる，検出の誤り

図 14　提案方式による切り出しの失敗例

6. ま と め

本稿では，最短経路の収束を利用した文字切り出し方式を提案した．提案方式は，異なる始点を持つ最短経路が，同一の終点を持つこと利用して，最短経路の始点を検出する．このため，提案方式は，文字間隔と文字幅に非依存であり，多値画像に対応可能という利点を持つ．

評価の結果，提案方式が，投影分布と連結成分に基づく先行方式 [8] に対して，日本語と英語についてともに優位であった．

文　献

[1] Tanzila Saba, Ghazali Sulong, Amjad Rehman, “A Survey on Methods and Strategies on Touched Characters Segmentation,” International Journal of Research and Reviews in Computer Science, Vol.1, No.2, pp.103-114, June 2010.

[2] Richard.G. Casey, Eric. Lecolinet, “A Survey of Methods and Strategies in Character Segmentation,” IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol.17, pp.690-706, 1996.

[3] 秋山照雄, 内藤誠一郎, 増田功, “非接触文字優先切出しによる印刷物からの文字切出し法,” 電子情報通信学会論文誌, Vol.J67-D, No.10, pp.1194-1201, Oct 1984.

[4] 中嶋正臣, 米倉雄司, “平滑化周辺分布と判別分析を用いた手書き文字切出し方式,” 電子情報通信学会論文誌 D-II, Vol.J78-D-II, No.7, pp.1039-1046, Jul 1995.

[5] 諏訪美佐子, “グラフ理論の手法を利用した自由手書き文字切出し,” 電子情報通信学会技術研究報告,PRMU2000-87, Vol.100, pp.9-16, Oct 2000.

[6] A.Ventzislav “Using Critical Points in Contours for Segmentation of Touching Characters,” Proceedings of the 5th International Conference on Computer Systems and Technologies, 2004.

[7] 中山英久, 藤原勇太, 加藤寧, “背景領域細線化を用いた手書き文字切出しの改良手法,” Forum on Information Technology 2009, pp.365-370, Sep 2009.

[8] Seong-Whan Lee, Dong-June Lee, Hee-Seon Park, “A New Methodology for Gray-Scale Character Segmentation and Recognition,” IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol.18, No.10, Oct 1996.

[9] Jia Tse, Dean Curtis, Christopher Jones, Evangelos Yfantis, “An OCR-Independent Character Segmentation Using Shortest-Path in Grayscale Document Images,” 6th International Conference on Machine Learning and Applications, ICMLA 2007, pp.142-147, Dec 2007.

[10] 村瀬洋, 若林徹, 梅田三千雄, “言語情報を利用した手書き文字列からの文字切り出しと認識，” 電子情報通信学会論文誌 D, Vol.J69-D, No.9, pp.1292-1301, 1986.

[11] 嶺竜治, 古賀昌史, 佐匂裕, “N-gram 言語統計量を併用した島駆動型文字列認識方式，” 電子情報通信学会論文誌 D, Vol.J89-D, No.5, pp.1011-1018, May 2006.

[12] Seong-Whan Lee, Y-J Kim, “Direct Extraction of Toporographic Features for Gray Scale Character Recognition,” IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol.17, No.7, pp.724-729, July 1995.