4L－7

上下境界線分に着目した文書画像からの黒画素塊検出方式

天野富夫
日本アイ・ビー・エム株式会社 東京基礎研究所

1　はじめに

文書画像の構造解析は既存印刷文書をコンピュータに自動入力するための重要な技術であるが、これらの処理においては黒画素の連結成分を追跡して得られた外接矩形の情報が多く用いられている。本稿では黒画素塊の上下境界線分に着目して黒画素塊に対応する矩形を検出する方式を提案する。従来はPC上で黒画素の追跡処理を実用的な速度で行うため、文書画像に閾値以下の短い白ランを黒で置き換えるぼかし処理を適用した後で追跡処理が行われていた。しかしこの手法では異なるカテゴリーに属する黒画素塊(例えば文字と表の枠)がぼかし処理によって連結され一つの外接矩形として検出されてしまうと、以後の解析処理に致命的な悪影響をあたえることが多かった。本方式では上下境界線分にはさまれた部分矩形を中間結果として連結性チェックの対象とすることにより、それらの高さや位置関係あるいは適当な候補領域に対して認識等を行った結果から過剰連結の影響を避けることが可能である。

2　黒画素塊矩形検出アルゴリズム

黒画素塊矩形の検出は1)上下境界線分に着目した部分矩形領域の検出、2)部分矩形領域の統合・分離による黒画素塊外接矩形の決定、の2つのステップの組み合わせで行われる。ステップ1の部分矩形検出処理の概要を図1に示す。文書画像はラスター走査され、各ライン中の白ランの長さが閾値S以下の場合は左右の黒ランを連結したランデータが計算される。 これらのランデータを一本上のラインのデータと比較することにより上下の境界線分のランデータを得ることができる。 白ランの下の黒ランは上境界、逆に白ランの上の黒ランは下境界とみなされる。 上境界線分と下境界線分は必ず対になっているので上境界線分を記録しておけば下境界線分が検出されたときに上境界線分の中からX座標が重なる線分を見つけ上下にはさまれた矩形領域の座標情報を計算することができる。下境界線分と対応がついた部分は上境界線分データから削除されるが、上境界線分が下境界と部分的に重なっている場合は、重なり部分のみが削除されるよう上境界線分の分割やランデータの変更が行われる。ぼかし処理によって線分などの非文字領域と近接する文字列が連結された場合、連結黒画素の追跡の結果としてはただ1個の矩形が検出されるのみであるが、ステップ1の手続きに従えば連結された黒画素塊を何か所かで縦割りにした矩形の集合が得られる。

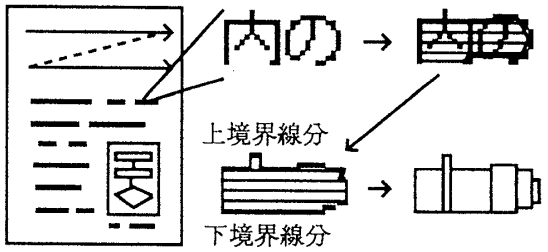

図 1: 境界線分に着目した矩形検出方式

ステップ2では過剰連結の可能性がある部分の分離や隣接する矩形領域の統合が行われる。分離・統合の方針としてはアプリケーション(自動入力の対象となる情報や処理対象文書の種別)に応じてさまざまなバリエーションが考えられるが、論文等を対象ととした文書入力システム[1]及び図面中の部品番号入力システム[2]の二つの適用例について述べる。

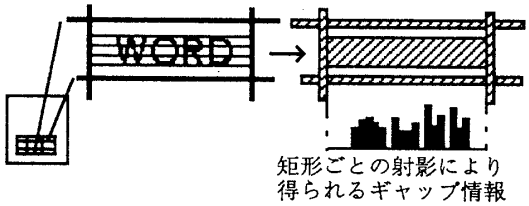

図 2: 文字要素と非文字要素の矩形の分離

文書入力システムのレイアウト解析処理にとっては、ぼかし処理による文字列と非文字列の連結を切り離しつつ本文領域については同一文字行に属する黒画素塊ができるだけ統合された矩形群として与えられることが望ましい。ここでは、矩形領域の高さと縦横比から文字と非文字列を連結している可能性のある矩形を検出し、元画像の該当領域から抽出した黒画素の縦方向射影情

A blob detection method by using horizontal top/bottom boundaries
Tomio Amano
IBM Research, Tokyo Research Laboratory

報(白画素のギャップ情報)に基づいて矩形集合を分離、グループ化しその外接矩形座標を出力している(図2)。

論文ページに限らず多くの文書画像解析処理においては黒画素塊矩形の集合を文字列、線分、絵領域等に分類する必要が生じるが、矩形領域の高さや縦横比の値は強力で抽出が容易な特徴として広く用いられているものである。これらの情報だけで文字か非文字かの判定を完全に行うことはできないが、文字・非文字間の連結の可能性をチェックすることは十分可能である。

文書画像解析のアプリケーションには図面中に点在する部品番号等の文字列を読み取りたいといった要求も多い。この場合もぼかし処理によって文字列が引出し線とつながってしまったケース等を救済するため文字列の高さに関する情報を用いることができるが、文字列の候補をいったん認識してみてその結果から入力対象となっている文字列を同定することも可能である。部品番号は桁数や使われている文字種に一定の制約(例えば数字4桁の後に英大文字1桁がつく等)があり、文字列候補領域から文字の切り出しと認識を行ってみてこの種の制約と合致するか否かによって必要な情報のみを入力することができる。図面内文字読み取りの例を図3に示す。

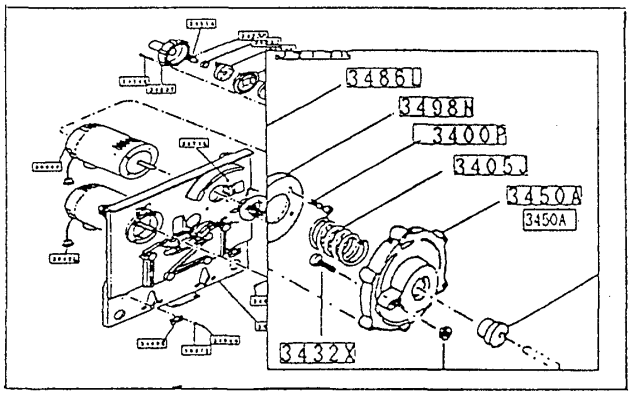

図 3: 図面内文字読み取りの実行例

3 評価実験

四種類の横書き文書(論文、欧文論文、マニュアル、雑誌)各5枚(16 dots/ mm でスキャンした3072x4480画素のデータ)にぼかしの閾値 S を 32(2mm) から 80(5mm) まで 16(1mm) ずつ変化させて検出誤り数を測定した(表1)。全体で約1000行の本文領域については全ての文字列黒画素塊を非文字列要素と分離して検出することができた。図表領域については文書Dで誤りが多く発生しているが、これは文書Dでは表の一部がハーフトーンで網掛けされていたためである。ハーフトーン部分を除いた図表領域中の文字列774個に対する検出率はS=32で99.6%、S=80で93.3%であった。この場合の誤りの原因は閾値 S を大きくしたとき枠の中の文字が枠ごと塗りつぶされてしまうためであった。また処理に要した時間は同じ画像にぼかし無しで連結黒画素の追跡処理を行った場合の1/4から1/12であった。

図面内文字列については実際に認識までを行うシステムを作成し評価を行った。五種類の版元の異なる図面合計89枚(10 dots/ mm でスキャンした1920x2688画素のデータ)を処理した結果92.5%の部品番号を読み取ることができた(表2)。100以上の部品番号を含む複雑な図面をPC(80386 CPU, 20-MHz clock)で処理するのに要した時間は文字列候補領域の検出に20秒、文字認識と文字種に関する制約を利用した結果のチェックに40秒(文字認識自体のスピードは30文字/秒)であった。

表 1: ぼかしの閾値と誤り数の関係

文書	矩形数*	S=32	S=48	S=64	S=80
A	42408	0	0	0	0
B	25797	2	18	46	46
C	4213	1	1	1	1
D	115940	53	56	54	57
合計	188358	56	75	101	104

*黒画素の連結を追跡して得られる外接矩形の数

表 2: 部品番号読み取り結果

版元	A	B	C	D	E	合計
枚数	20	20	9	20	20	89
部品番号	735	802	220	712	621	3090
検出もれ	12	7	2	4	3	28(0.9%)
欠け/過剰	26	18	10	37	25	116(3.8%)
認識誤り	14	26	2	7	38	87(2.8%)
合計	52	51	14	48	66	231(7.5%)

4 まとめ

文書画像内の黒画素成分の上下境界線分に着目して黒画素塊に対応する矩形を検出する方式を提案した。本方式は連結する黒画素の追跡を行う方式に比べて高速でありまた従来のぼかし処理による高速化の副作用をおさえることに成功している。結果として得られる矩形は、近傍画素との連結性に基づく黒画素塊外接矩形と完全に一致するわけではないが、レイアウト解析や文字列検出等の文書画像解析処理においては同様に扱えるものである。

参考文献

[1] Amano T. et al. : "DRS: A Workstation-Based Document Recognition System for Text Entry", IEEE Computer, 25, 7, pp.67-71 (1992).

[2] Amano T. et al. : "A Character-String Detection Algorithm Using Horizontal Boundaries and Its Application to a Part-Number Entry System", Proc. SPIE, Vol. 1,452, pp.330-339 (1991).