印刷文書画像の領域分割と保存

2H－2

木下 義夫　今井 正和　烏野 武
奈良先端科学技術大学院大学 情報科学研究科

1　はじめに

電子化された文書は、遠隔操作で本の閲覧ができる、省スペースである、本文の検索が容易である、などの長所をもつ[1]。そこで、現時点で大多数を占める紙メディアの出版物を電子化するシステムが望まれる。電子化の際の条件として、(1) 利用者に対して情報量を落さない、(2) 論文集のような整った様式の文書だけでなく縦書き・横書きの混在を含め、雑誌の紙面まであらゆる書式に対応する、(3) 大量の文書を処理するために人間の介入が不要でなければならない、などがある。

これらの条件ををふまえて、文書画像を効率良く保存するシステムを構成したので報告する。また、実験結果によりこの手法の有効性を示す。

2　処理の流れ

本稿では、400dpi、モノクロ2値で読み込むスキャナに対象をしぼった。以下に処理の流れを説明する。

2.1　サブサンプリング・3値化

400dpiのオリジナル画像を100dpiの解像度に落とす。単純に間引きをしたのでは、小さな点が拾われなかったり、網かけの部分が真っ白になるといった情報落ちがおきる。その解決策として、元の画像の4×4ドット毎に黒画素の数をカウントし、それが4個以上なら黒、1〜3個なら赤、0個なら白を処理用画像の対応するドットの色とした（図1）。

2.2　連結領域の抽出

黒・赤画素領域を画像の右上から順に抽出する。このとき、連結領域を外接矩形で切り取って抽出するのではなく、その領域の輪郭そのままの形で抽出する。これにより、入り組んだレイアウトの紙面でも適切に

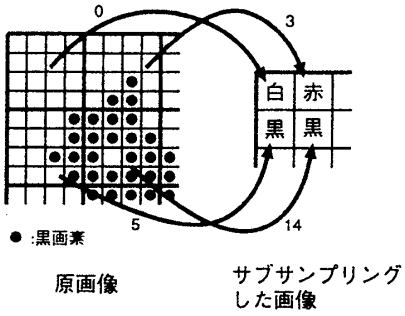


図1: サブサンプリング・3値化

領域抽出が行なえる。そして抽出した各領域が、ノイズ領域、文字（列）候補領域、セパレータ領域（水平または垂直方向の直線）、画像領域のうちのどれであるかを判断する。以下に判別のアルゴリズムを示す。

2.2.1　ノイズ領域

赤画素を1ポイント、黒画素を3ポイントとして領域内の合計ポイントを計算し、4ポイント以下の領域はノイズとみなし、除去する。この方法により、スキャナで読み込む際の細かなノイズはほとんど除去できる。逆に文字中の点などが間違って除去されることはない。これ以降は、赤・黒の画素をまとめて黒画素と呼ぶ。

2.2.2　セパレータ領域

領域の縦／横比が極端に大きい、または小さいものは、セパレータ（直線）領域と判断する。

2.2.3　画像領域 (1)

外接矩形の高さ、幅共に30ドット以上あり、外接矩形の面積がある値以上の領域、つまり文字（列）にしては大き過ぎる領域は、画像領域と判断する。さらに、画像領域と判断されると、その内部に大きな空白領域があるかを探し、あればその内部も同様の処理（連結画素の抽出）を再帰的に行なう。

Segmentation and Preservation of Printed Document Images
Yoshio KINOSHITA, Masakazu IMAI and Takeshi UNO
Nara Institute of Science and Technology
8916-5 Takayama, Ikoma, Nara 630-01, Japan

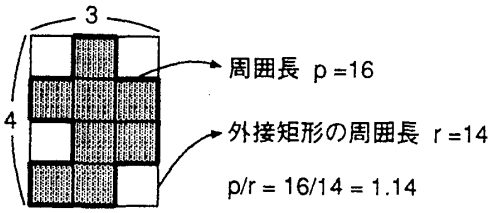


図 2: 画像領域（2）

2.2.4 画像領域 (2)

前の段階では、外接矩形の高さと幅のみで判断しているので、太い直線や、見出しの白抜き文字、少し傾いている直線などは画像領域だと判断できていない。そこで、連結黒画素領域の周囲長を p、領域の外接矩形の周囲長を r として、p/rを計算する（図 2）。長方形や、円など、凹の部分がない領域では p/r は最小で、1.0 になる。領域の周囲に凹凸が多くなるほどその値は大きくなる。そこで、p/rがある閾値以下ならば領域の周囲の凹凸が少ないので文字領域ではない（画像領域）と判断する。この判断基準を用いると、画像に対して新たな操作をすることなく、既知の数値で簡単に求まる値を使うので処理速度を落とさずにすむ。

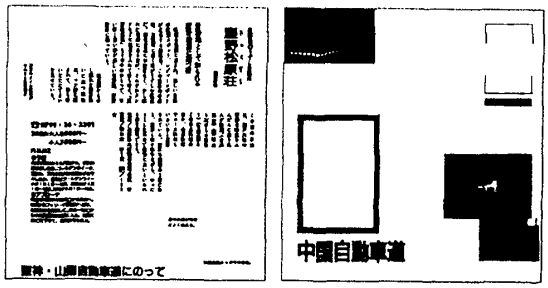
2.2.5 文字（列）候補領域

これまでのいずれの領域であるとも判定されなかった領域を文字（列）候補とする。

2.3 文字列抽出・文字認識

文字（列）候補領域に対して水平方向、垂直方向ともに 10 ドット程度の拡大処理をして、文字列のブロックを得る。水平・垂直方向に 10 ドットの拡大処理とは、すべての黒画素について、その左右 5 ドット以内をすべて黒画素とし、さらにその画像のすべての黒画素について上下 5 ドット以内をすべて黒画素とする処理である。ただし、セパレータ及び画像領域を越えては拡大させない。そして、連結した各ブロックに対して、縦・横方向の黒画素数のヒストグラムをとり、ひとつの文字列ごとに分割する。これを OCR(Optical Character Reader) のルーチンに渡し、コード化する。OCR で読めなかった文字列候補領域は画像領域とし、画像のまま保存する。人による後処理はしない。

原画像



抽出された文字列領域と画像領域

図 3: 実験結果

3 実験

インプリメンテーションは領域の種類判別の部分までできている。A4 サイズの論文集や雑誌について実験を行なった。実行例を図 3に示す。セパレータ、文字列、画像領域がおおむね正しく認識されていることが分かる。処理時間は、紙面中の連結領域の個数などにより大幅に異なるが、1 ～ 2 分／頁程度だった。

4 むすび

文書画像の領域分割に際して、サブサンプリングの時に 3 値化をする、領域をそのままの形で抽出する、などの手法を用いることにより、書式の定まっていない雑誌などの紙面でも正確かつ高速に処理でき、効率良く文書画像を保存できることを示した。

今後は、グレイスケールの画像にも対応できるようにするとともに、文字列領域をより正確に抽出できるようにしたい。

参考文献

[1] Story, G. A. O'Gorman, L., Fox, D., Schaper, L. L. and Jagadish, H.: The RightPages Image-Based Electoronic Library for Alerting and Browsing, *COMPUTER*, Vol. 25, No. 9, pp. 17-27 (1992).