隣接グラフを用いた欧文文書画像からの文字列抽出

岩田 基[†] 黄瀬浩一[†] 松本啓之亮[†]

本論文では，異なる傾きの文字列を含んだ欧文文書画像から文字列を抽出する手法の1つとして，隣接グラフを用いた手法を提案する．隣接グラフとは，連結成分の隣接関係を表したグラフである．本手法では，連結成分の隣接関係の抽出に一般図形ボロノイ図を用いているため，入力文書画像のレイアウトや傾きに依存したパラメータを用いることなく隣接グラフを生成することができる．さらに，欧文文書の文字列は隣接グラフのパスで表現できるため，文字列抽出を，文字列として適切な隣接グラフのパスを求める問題に置き換えることができる．そのようなパスを得るため，本手法では，まず，連結成分の大きさ，連結成分間の距離などを基準に文字列の一部として確度の高いパス（シード）を求め，次に，隣接グラフの枝を用いてシードを繰り返し延長するという戦略をとる．シードの延長の際には，近接性，直線性を考慮しつつ接続すべき枝を局所的に選択するため，効率的な処理が実現できる．文書画像50サンプルを用いて実験を行った結果，文字列抽出率はおよそ90%，処理時間は平均6.82 secであり，精度の面では課題が残るものの，効率的には優れていることが分かった．

Text-line Extraction from English Document Images Using the Neighbor Graph

MOTOI IWATA,[†] KOICHI KISE[†] and KEINOSUKE MATSUMOTO[†]

This paper presents a method of text-line extraction which is applicable to documents including various orientations of text-lines. The method is characterized by the use of the neighbor graph, i.e., the representation of adjacency among connected components. The use of the area Voronoi diagram enables us to generate the neighbor graph independently of orientations of text-lines as well as layout of documents. Using the representation, the task of text-line extraction is considered to be the extraction of paths which corresponds to text-lines from the neighbor graph. In order to obtain such paths, the method first extracts parts of text-lines called seeds and then extends them iteratively based on the linearity and proximity between a seed and an edge in the neighbor graph. The extension is based only on local examination of edges so as to reduce the computational cost of the method. As the experimental results for 50 document images, we obtained 90% of text-line extraction rate with 6.82 sec processing time for a page. Although there exist some limitations in accuracy, we have confirmed that the method is flexible and efficient.

1. まえがき

近年，印刷文書をテキストデータとしてコンピュータ内に格納しようとする動きが高まっている．しかし，既存の印刷文書の数は膨大であるため，人手で入力を行うのは現実的ではない．このため，印刷文書を文書画像として取り込み，解析することによりテキストデータに変換する処理が望まれている．多くの場合，文書画像としては2値画像を考えればよいため，以下では2値文書画像の解析に焦点を絞る．

テキストデータへの変換のためには，まず，取り込んだ文書画像に対してレイアウト解析を施す必要がある．レイアウト解析とは，文書画像の構成要素の領域を切り出し，それらに，テキスト領域，図・表領域などの属性を与える処理である．文字列抽出は，レイアウト解析の一処理であり，文書画像中の文字列を表す領域を抽出する処理である．多くの文書は文字列を主体として構成されているため，文字列抽出はレイアウト解析の中心課題の1つであり，従来から様々な文字列抽出法が提案されている．

文字列抽出法は，対象とする文書画像に対してどのような仮定を設けるかによって分類できる．すべての文字列が互いに平行であると仮定して文字列を抽出する手法は，数多く提案されており，高い精度，効率を示している[1),2)]．しかし，近年，レイアウトの多様化

† 大阪府立大学工学部
Department of Computer and Systems Sciences, College of Engineering, Osaka Prefecture University

が進み，様々な傾きの文字列が1つの文書に含まれるような場合が増えている．このような対象に対する文字列抽出手法もさかんに研究されているが[3),4)]，前者に比べて十分な精度，効率が得られているとはいえない．

文字列は通常，隣接する連結成分から構成されると考えられる．そこで後者のように，文字列のレイアウトに対して，より制約の少ない文書画像から文字列を抽出するためには，連結成分の隣接関係をいかに一般性を損なわずに抽出・表現するか（表現法の確立），ならびに，その表現に基づいていかに処理を施すか（処理法の確立）の2点がより重要となってくる．

従来の表現法には，連結成分間の距離が閾値以下のものを隣接しているとする手法[4)]や，k-NN（nearest neighbor）に基づいて隣接関係を規定する手法[3)]などがある．しかし，距離の閾値や適切な k の値は，対象文書のレイアウトに依存するため，必ずしも十分に一般性を得ているとはいえない．また，処理法としては，弛緩法[4)]やハフ変換[5)]，焼きなまし法[6)]などの様々な手法が利用されている．これらの手法は，精度的に満足のいくものも多いが，一般にはかなりの処理時間が必要なため，効率の面では問題を残している．

そこで本論文では，欧文文書画像を対象とし，傾きやレイアウトによらず安定的かつ効率的に文字列を抽出する手法を提案する．対象を欧文文書画像とする理由は，欧文文書画像では連結成分をほぼ文字と見なせるため，基本的には文字抽出を考える必要がなく，文字列抽出に焦点を絞ることができるからである．本手法の表現面での特徴は，連結成分の隣接関係を一般図形ボロノイ図[7)]に基づいて抽出するとともに，隣接グラフと呼ぶ形式で表現する点にある．一般図形ボロノイ図を用いることにより，文書のレイアウトや文字列の傾きに依存したパラメータを用いることなく，連結成分の隣接関係を抽出できる．また，隣接グラフの導入によって，文字列抽出を，文字列として適切な隣接グラフのパスを求める問題に置き換えることができる．処理面では，局所的な基準を用いた繰返し処理により隣接グラフの枝を選択するという，処理時間を重視した手法を用い，そのうえでどの程度の精度が得られるかを吟味する．

以下では，まず，本手法で用いる隣接グラフについて説明し，次に，本手法における文字列抽出の方針を述べる．そして，本手法の処理の流れについて述べた後，文書画像50サンプルを用いて行った実験の結果から，本手法の有効性を検討する．

2. 隣接関係の表現

2.1 一般図形ボロノイ図

本手法では，連結成分間の隣接関係の抽出に，一般図形ボロノイ図を用いる．通常のボロノイ図（点ボロノイ図）が点を生成元とする領域分割図であるのに対して，一般図形ボロノイ図は任意形状の図形を生成元とする領域分割図である．

いま，生成元の図形の集合を $F=\{f_1,\ldots,f_n\}$，2点 p，q 間の距離を $d(p,q)$，点 p と図形 f_i の距離を，

$$d(p,f_i)=\min_{q\in f_i} d(p,q), \tag{1}$$

とするとき，一般図形ボロノイ図 $\mathcal{V}(F)$ は以下のように定義される．

$$\mathcal{V}(F)=\{\mathcal{V}(f_1),\ldots,\mathcal{V}(f_n)\}. \tag{2}$$

$$\mathcal{V}(f_i)=\{p|d(p,f_i)\le d(p,f_j),\forall j\neq i\}. \tag{3}$$

ここで，領域 $\mathcal{V}(f_i)$ の境界を表す辺をボロノイ辺という．

一般図形ボロノイ図では，通常，ボロノイ辺が複雑な曲線となるため，簡単には求めることができない．ところが，点ボロノイ図を用いると，以下のように近似的に構成することができる[7)]．

(1) 生成元となる図形の境界点をサンプリングレート R でサンプリングし，それらのサンプル点を生成元（母点）とする点ボロノイ図を構成する．

(2) ボロノイ辺のうち，同一の図形に属する母点から生成されたものを除去する．

なお，近似的に構成された一般図形ボロノイ図では，ボロノイ辺は折れ線となる．

さて，本手法では，上記の生成元 f_i として連結成分を考え，点間の距離 d としてユークリッド距離を用いて近似的に一般図形ボロノイ図を構成する．たとえば，**図1**(a)に対しては，図1(b)のような図となる[8)]．

以上からも分かるように，文書画像に対する一般図形ボロノイ図は，文書画像の傾きや文字列の傾きに依存せず，また文書のレイアウトに依存したパラメータを設定することなく構成可能である．

2.2 隣接グラフ

本手法では，ボロノイ辺をはさんだ連結成分どうしは隣接していると考え，連結成分間の隣接関係を定義する．隣接グラフ $G=\langle V,E\rangle$ とは，連結成分間の隣接関係を記述したグラフであり，節点 $v_i(\in V)$ が連結成分 c_i に対応し，枝 e_{ij} が連結成分 c_i，c_j 間の隣接関係を表す．図1(c)に隣接グラフを，図1(d)に求めるべき文字列を示す．この図からも分かるように，求

図1 一般図形ボロノイ図と隣接グラフ
Fig. 1 An area Voronoi diagram and a neighbor graph.

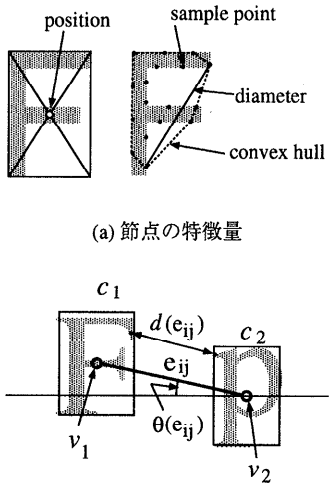


図2 特徴量
Fig. 2 Features.

めるべき文字列は通常，隣接グラフのパスで表される．

2.3 特 徴 量

隣接グラフから文字列を表すパスを選び出すために，**図2** に示すように，特徴量として，節点には位置，面積と直径，枝には距離と傾きを付与する．以下では，画像の左上を原点とし，右方向を x 軸の正方向，下方向を y 軸の正方向とする座標系を用いる．

位置 連結成分 c_i を囲み，x，y 軸に平行な辺を持つ最小の矩形を考え，その中心の座標 (x_i, y_i) を隣接グラフの節点 v_i の位置（position）とする．

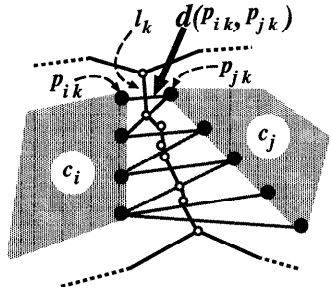


図3 ボロノイ辺と点間の距離
Fig. 3 Voronoi edges and the distance between points.

面積 連結成分 c_i について，一般図形ボロノイ図を近似生成するときに用いたサンプル点の凸包を考え，その面積を隣接グラフの節点 v_i に付与する面積 $a(v_i)$ とする．

直径 上と同様の凸包を考え，その直径（凸包の頂点間の最長距離）を隣接グラフの節点 v_i に付与する直径（diameter）$D(v_i)$ とする．

距離 2つの連結成分 c_i, c_j を分割するボロノイ辺を $\{l_1, \ldots, l_m\}$ とする．ここで，l_k はボロノイ辺を構成する線分である．一般に線分 l_k は，c_i の境界上のサンプル点 p_{ik} と c_j の境界上のサンプル点 p_{jk} から生成されている．本手法では，**図3** に示すように，隣接グラフの枝 e_{ij} に付与する距離 $d(e_{ij})$ を，これらの点間のユークリッド距離 $d(p_{ik}, p_{jk})$ を用いて次のように定義する．

$$d(e_{ij}) = \min_{1 \le k \le m} d(p_{ik}, p_{jk}). \tag{4}$$

傾き 枝 c_{ij} の x 軸に対する角度を枝 e_{ij} の傾き $\theta(e_{ij})$ とする．

3. 文字列抽出の方針

本手法では，以下のような一般的な欧文文字列の特徴を仮定している．

A1 文字列は，互いに隣接する一連の連結成分によって構成されている．より具体的には，文字列は隣接グラフのパス $(v_1, e_{12}, v_2, \ldots, e_{n-1,n}, v_n)$ により表すことができる．ここで，$i \neq j$ ならば $v_i \neq v_j$ である．

A2 同一文字列に属する連結成分は，異なる文字列に属する連結成分に比べて近接していることが多い．

A3 同一文字列に属する連結成分は同程度の大きさを持つ．

A4 文字列は直線状に並ぶ連結成分によって構成されている．

文字列抽出は，これらの特徴を用いて文字列として適

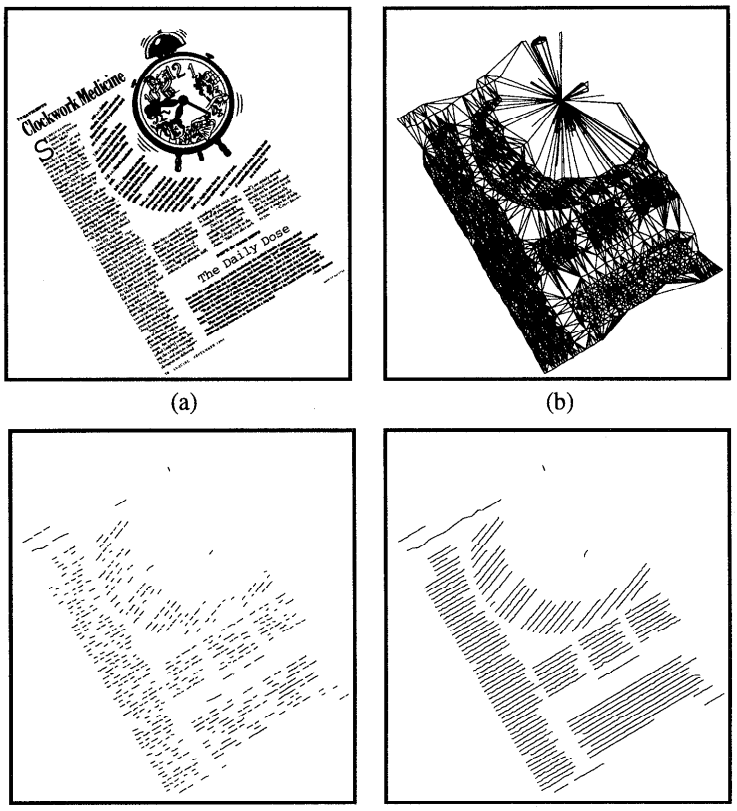


図4 処理過程．(a) 原画像 (300 dpi)，(b) 隣接グラフ，(c) シード，(d) 文字列抽出結果
Fig. 4 Results of each step. (a) input image (300 dpi), (b) neighbor graph, (c) seeds, (d) extracted text-lines.

切な連結成分の組合せを求める問題である．

以上の特徴を用いて，簡便に文字列を求める手法として，本手法では，まず，**A2**，**A3** の特徴の観点から文字列の一部である可能性の高いパスを求める．このパスをシードと呼ぶ．シードは，分岐やループを含まないので，シードから文字列の傾きを推定することができる．次に，推定した傾きの方向に，**A2**～**A4** を考慮しながらシードを延長することにより，最終的に文字列を得る．

シードを延長するとき，(i) 初期の段階ではシードに含まれる要素が少ないため，シードから求められる傾きの信頼性が低いこと，(ii) 一般に，延長する対象は一意に定まらない（シードの端には隣接グラフの枝が，多数接続している）ことが問題となる．前者については，シード延長を何回か繰り返し，条件を段階的に緩めていくことによって対処する．後者については，直線性を指標として，複数の候補を選び，その中から最も適切なものを選択することにより対処する．

4. 処理のながれ

本手法は，隣接グラフ生成，シード生成，シード延長の3ステップの処理から構成される．以下では，**図4** を用いて処理の流れを示す．なお，図4(a) は文字列の傾きが複数存在するページであり，全体が約 30° 傾いている．解像度は 300 dpi である．

4.1 隣接グラフ生成

まず，入力2値文書画像に対して一般図形ボロノイ図を生成する．このとき，凸包の面積が T_n 以下の連結成分はノイズとして除去する．次に，一般図形ボロノイ図に基づいて隣接グラフを生成する．その結果，図4(b) の隣接グラフが得られる．

4.2 シード生成

シード生成は，シード候補生成，シード候補選択の2処理から構成される．

4.2.1 シード候補生成

隣接グラフの枝の中には，文字列として明らかに不適切な枝も含まれている．そこで，まず，枝単位で不

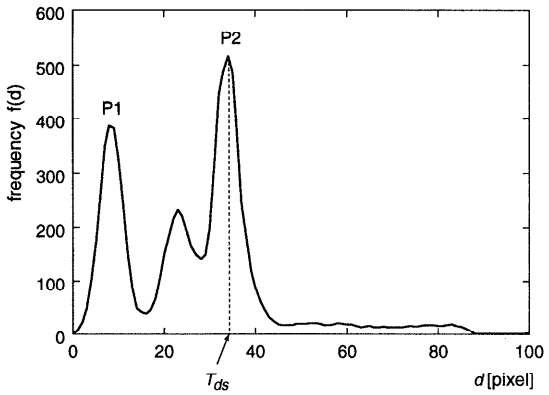


図 5 距離の頻度分布
Fig. 5 Frequency distribution of the distance d.

適切であると判断できるものを除去しておく．具体的には，**A3** に基づき，隣接グラフの枝の中から，

$$\frac{\min(a(v_i), a(v_j))}{\max(a(v_i), a(v_j))} \leq T_a, \tag{5}$$

$$\frac{\min(D(v_i), D(v_j))}{\max(D(v_i), D(v_j))} \leq T_D, \tag{6}$$

を満たす枝 e_{ij} を除去する．また，それによって接続する枝のなくなった節点も除去する．

次に，隣接グラフの枝のうち，$d(e_{ij})$ が T_{ds} 以下である枝を，$d(e_{ij})$ の小さい順に選び，両端点 v_i，v_j がシード候補に含まれるかどうかによって，以下の 3 つのいずれかの処理を行う．

(1) v_i，v_j のいずれもシード候補に含まれない場合，e_{ij} と v_i，v_j を新たにシード候補とする．
(2) v_i のみがシード候補に含まれる場合，e_{ij} と v_j をそのシード候補に加える．
(3) v_i，v_j のいずれもシード候補に含まれる場合，次の枝について処理を行う．

以上の処理によって，T_{ds} 以下の距離の枝に接続している節点はすべてシード候補のうちのいずれかに含まれる．

なお，適切な T_{ds} の値は文書のレイアウトに依存するため，自動設定が望ましい．本手法では，**図 5** に示すように，距離 $d(e_{ij})$ の頻度分布に基づいて T_{ds} を設定する．図 5 にも示されているように，頻度分布には，明確なピークが存在する．そのうち，最も大きな距離を持つピーク P2 は，本文文字列の行間を表すものである．本手法では，T_{ds} の値をこの距離に設定する．

4.2.2 シード候補選択

次に，シード候補からシードを選択する．具体的には，シード候補 s_k のうち，

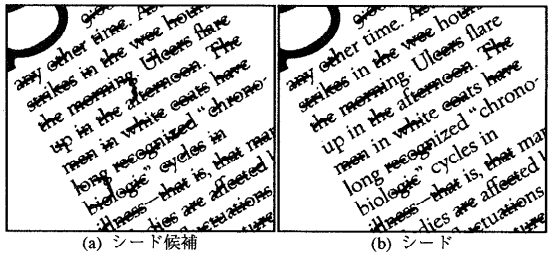


図 6 シード候補選択
Fig. 6 Selection of seed candidates.

$$V_\theta(s_k) \leq T_{v\theta}, \tag{7}$$

$$V_d(s_k) \leq T_{vd}, \tag{8}$$

を満たし，複数の枝を含むパスをシードとする．ここで，$V_\theta(s_k)$ はシード候補内の枝の傾きの分散，$V_d(s_k)$ はシード候補内の枝の距離の分散，$T_{v\theta}$，T_{vd} はしきい値である．式 (7)，(8) は，枝の傾きや距離のばらつきが大きい連結成分の並びはシードとして不適であることを意味している．また，1 本の枝からなるものは，文字列の方向性を定めるうえで不適であるためシードとはしない．**図 6** (b) に，図 6 (a) のシード候補から選ばれたシードを示す．図 4 (c) に，こうして得られたシードを示す．

4.2.3 シードの特徴量

シードには，以下の特徴量を付与しておく．これらの特徴量は，次の，シード延長で用いる．

距離 シード s に含まれる枝の距離の平均をシード s の距離 $d(s)$ とする．

傾き シード s の両端にある節点を結ぶ直線の傾きをシード s の傾き $\theta(s)$ とする．

4.3 シード延長

最後に，シードを延長し，文字列を抽出する．シード延長の処理は N 回繰り返し，現在の繰り返し回数 n ($\leq N$) を用いて延長判定条件を緩めていく．

本手法では，毎回の繰返しにおいて，各シードを両端から可能な限り延長する．手順の概略を**図 7** に示す．ここで，“Select_Edge” は，ループ回数 n に応じてシード s の v_i 側から延長する枝 e_{ij} を選ぶ処理であり，“Merge_Edge” は，そのような枝が存在した場合に，シード s の v_i 側に枝 e_{ij} と節点 v_j を接続する処理である．このとき，枝が他のシードとの間に引かれたものである場合，その枝をシードに加えることは 2 つのシードを統合することを意味するため，相手側のシードからも延長判定を行い，信頼性を向上させる．

処理の詳細は以下のとおりである．

まず，“Select_Edge” において，シード s の一端 v_i に接続する枝 e_{ij} のうち，延長しても分岐，ループが

```
for n = 1 to N do
 foreach seed s do
  if the seed s exists in S(the set of seeds) then
   do
    let v1 be a vertex at an end of s
    let v2 be a vertex at the other end of s
    s_old ← s
    e1x ← Select_Edge(s,v1,n)
    e2y ← Select_Edge(s,v2,n)
    s ← Merge_Edge(s,v1,e1x)
    s ← Merge_Edge(s,v2,e2y)
    if s_old ≠ s then
     Delete all seeds which share edges of s from S
     Add s in S
     Recalculate features of s
    end if
   while s_old ≠ s
  end if
 end
end
```

図7 シード延長
Fig. 7 Extension of seeds.

生じない枝について，図8に示すようなシード s の傾きと枝 e_{ij} の傾きとの誤差，

$$\theta_e(s,e_{ij}) = |\theta(s) - \theta(e_{ij})|, \quad (9)$$

を求める．次に，$\theta_e(s,e_{ij})$ が小さい順に K 本を上限として選ぶ．こうして選んだ枝に対し，$\theta_e(s,e_{ij})$ が小さい順に以下の延長判定を行う．

$$J(s,e_{ij}) = \frac{\theta_e(s,e_{ij})}{\frac{n}{N}C_\theta} + \frac{d_e(s,e_{ij})}{C_d} \leq 1. \quad (10)$$

$$d_e(s,e_{ij}) = (d(s) - d(e_{ij}))^2. \quad (11)$$

図9に示すように，C_θ, C_d は式(10)を満たす (θ_e, d_e) の領域を規定する値である．

式(10)を最初に満たした枝を e_{ij}，枝が接続する節点のうち，シード s に含まれない方を v_j としたとき，"Merge_Edge" において以下の処理を行う．

(1) v_j が s 以外のシードに含まれていない場合には，e_{ij} と v_j をシード s の要素に加える．

(2) v_j が s 以外のシード s' に含まれている場合には，図10のように s からは適当でも s' からは不適当な枝を除外するため，以下の判定を付加的に行う．

- v_j に接続しているすべての枝 e について $\theta_e(s',e)$ を求め，$\theta_e(s',e)$ の小さい方から K 番目以内に e_{ij} が入っている．
- $J(s',e_{ij}) \leq 1$

の2つが満たされていれば，e_{ij} を s の要素に加え，s と s' を統合する．

N 回の延長処理が終了した後，枝の数が M 本以下であるシードは文字列ではないと判断する．これは，少数の連結成分からなるシードは，文字列ではなくノイズであると見なす方が適切であるという考えに基づいている．最後に，こうして得られた延長後のシードを文字列として出力する．

出力結果を図4(d)に示す．シードが生成されていない文字列や文字間隔が広い文字列，また，カラム間隔が狭い部分では誤抽出が見受けられるが，大多数の文字列は異なる傾きを持つにもかかわらず，正しく抽出されている．

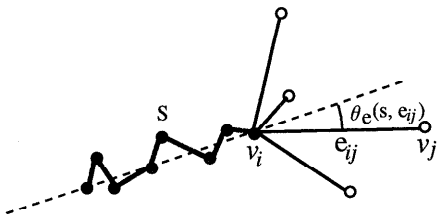


図8 シードの傾きと枝の傾きとの誤差
Fig. 8 Angular error between a seed and an edge.

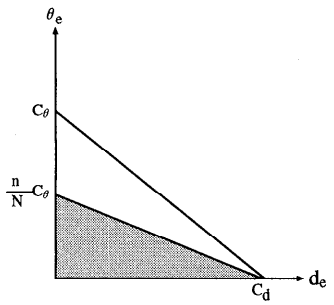


図9 シードの延長判定条件
Fig. 9 Criterion of seed extension.

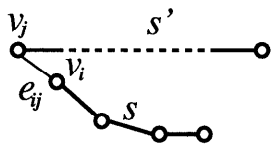


図10 枝の付加的な判定の必要性．枝 e は s から見て適切であっても，s' からは不適当となる．
Fig. 10 Need of an additional test with a different seed s'. The edge e is appropriate for s but not for s'.

5. 実　　験

5.1 実験条件

本手法をC言語により実現し，PC（Pentium II 300 MHz, 256 MB実メモリ，OS:RedHat Linux 4.2）を用いて実験を行った．実験には，2つのデータセットUW1, Nonrectを用いた．UW1はワシントン大学の文書画像データベースIに含まれる25枚の画像からなり，互いに平行な文字列のみを持つ．レイアウトとし

表1 実験に用いたパラメータ
Table 1 Values of parameters.

ステップ	隣接グラフ生成		シードの生成				シードの延長				
パラメータ	R	T_n	T_a	T_D	$T_{v\theta}$	T_{vd}	N	$K_{\max}$	M	C_d	C_θ
値	1/7	64	1/40	1/10	400	50	10	2	2	1600	50

ては，すべて矩形レイアウト，すなわちブロックが矩形で表現できるものであり，22枚が1カラム，2枚が2カラム，1枚が3カラムである．一方，Nonrectは，種々の欧文文書25枚をスキャナで取り込み，作成したものである．レイアウトは非矩形レイアウト，すなわちブロックが任意形状をとるものであり，そのうち6枚は互いに平行でない文字列を含む．また，カラム数は，3枚が1カラム，5枚が2カラム，13枚が3カラム，4枚が4カラムである．文書画像の傾きに対するロバスト性を評価するために，Nonrectの文書画像は，いずれも反時計方向に10°傾かせている．文書画像の解像度と平均の大きさは，300 dpi，2685 × 3379である．

実験には，**表1**に示すパラメータを用いた．パラメータの設定には，UW1，Nonrectとは異なる文書画像50枚を用いた．内訳は，ワシントン大学の文書画像データベースの画像25枚，Nonrectと同様に雑誌から取り込んだ文書画像25枚である．各パラメータの値は，これらのサンプルに対して適当な結果が得られる値の範囲の中央値とした．

5.2 実験結果と考察

実験結果を**表2**に示す．表2において，“文字列数”と“文字列抽出率”はそれぞれ，各データセットにおける文字列数の総和と正しく抽出された文字列の割合である．この実験では，ハイフンやドット，カンマ，ピリオドのように，ごく小さい連結成分が文字列に加わっていなくても正解であるとしている．

表2において，UW1とNonrectで文字列抽出率に差がほとんどないことから，本手法は，レイアウトや傾きに依存しない処理が可能であることが分かる．一方，全体としては，およそ90%程度の文字列抽出率にとどまっており，改善の余地があるといえる．

処理誤りは，大きく，過分割，過統合，無抽出の3種類に分類される．過分割とは1つの文字列を複数に分割した場合，過統合とは逆に複数の文字列を統合した場合，無抽出とは1つの文字列をまったく抽出できなかった場合を指す．各々の割合を表2に，その典型的な原因を以下に示す．

過分割 **図11**(a)に示すように，複数の文字が1つの連結成分となっている場合には，直径が大きくなりすぎて延長判定の対象にならず，過分割が生じた．また，図11(b)のように，同一文字列中の連結成分間の距離が大きい場合には，延長判定の式(10)を満たさないため，過分割となった．

過統合 シード生成の段階で誤っている場合（図11(c)）に過統合が多く生じた．

無抽出 文字列中にシードが含まれない場合に無抽出となった．

これらの誤りは，本手法のような局所的な判断に基づく処理の限界を示すものである．すなわち，これらの誤りを訂正し，より精度を向上させるためには，テキストブロックの情報など，より大域的な情報を用いる必要があろう．

表2 実験結果
Table 2 Experimental results.

	文字列数	文字列抽出率	処理誤り		
			過分割	過統合	無抽出
UW1	1111	89.1%	7.84%	0.63%	2.43%
Nonrect	2975	89.9%	4.00%	4.73%	1.37%
計	4086	89.7%	5.03%	3.61%	1.66%

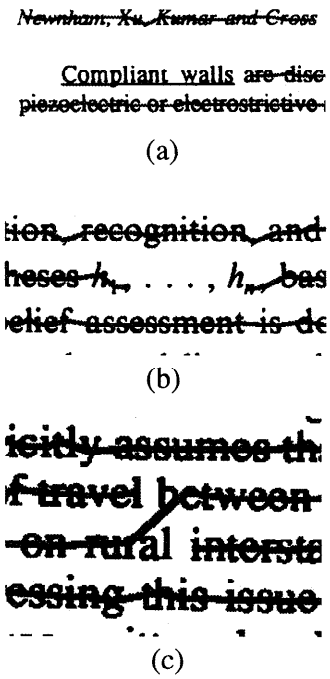


図11 処理誤りの例
Fig. 11 Examples of errors.

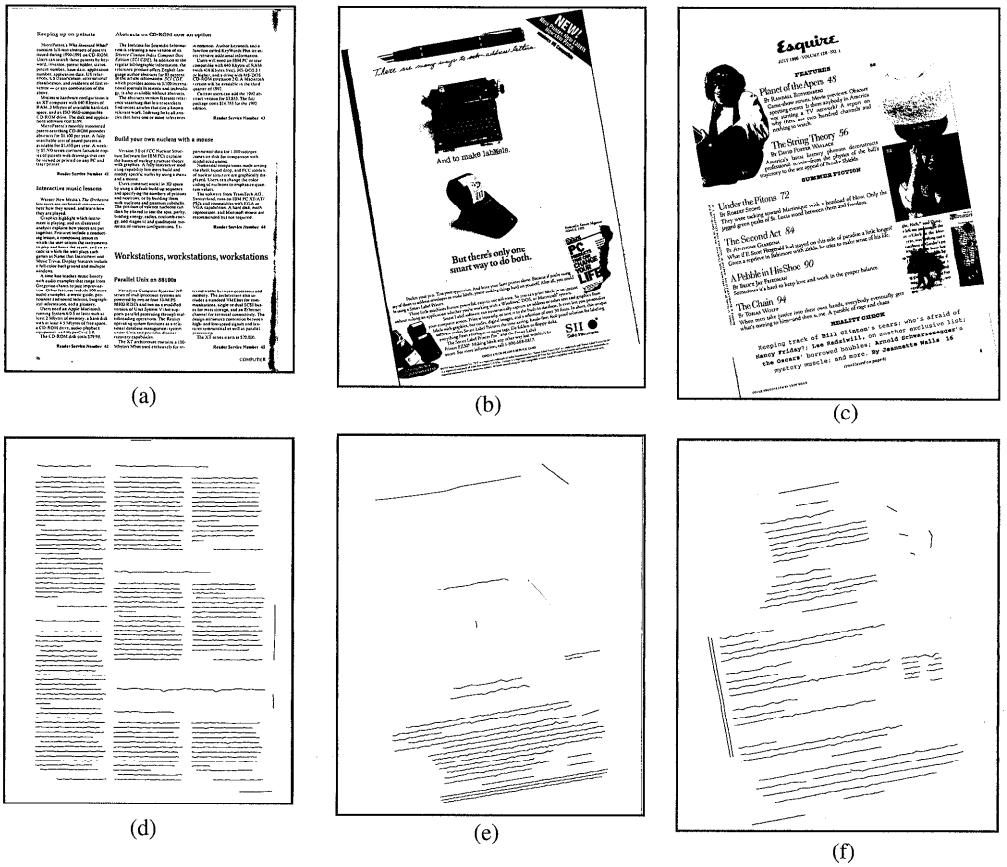


図 12 処理例
Fig. 12 Examples of results.

図 12 に処理例を示す．図 12 (a)～(c) が入力画像で，図 12 (d)～(f) がそれぞれ図 12 (a)～(c) に対応する処理結果である．図 12 (a) の文書画像は UW1，図 12 (b)，(c) の文書画像は Nonrect のものである．ここで，図 12 (b) は，右上に異なる傾きの文字列が存在し，文字列を構成する文字の大きさやフォントが異なるものが，同一文書画像中に含まれている．図 12 (c) は，左中央に異なる傾きの文字列がある文書画像である．図 12 (d) における処理誤りはページの境目の誤抽出のみであり，他の文字列は正しく抽出された．図 12 (e) では，右上の異なる傾きの文字列も含め，ほぼすべての文字列が正しく抽出されている．図 12 (f) では，同一文字列間でも，単語間距離が大きいため，過分割が見受けられた．しかし，それ以外の文字列は良好に抽出できており，異なる傾きの文字列も正しく抽出された．このことから，本手法は上記のようなレイアウトの差異を吸収し，文字列を抽出できるといえる．

最後に，処理時間について考察する．処理の規模を知るため，実験に用いた文書画像のデータを平均したところ，連結成分数が 8565，隣接グラフの枝の数が 7660 であり，文字列として選ばれた枝の数は 2404 であった．平均処理時間とその内訳を**表 3** に示す．特徴としては，全体の処理時間の半分をラベリングなどの画像処理に費していること，シードの生成，延長にはほとんど時間がかかっていないことがあげられる．この理由は，本手法がシードの延長に，局所的な選択基準のみを用いていることと考えられ，シード生成．延長の処理がほかに比べて十分効率的であるといえる．

表 3 処理時間
Table 3 Computation time.

隣接グラフ生成	
ラベリング，境界線追跡，サンプリング	3.39 sec (49.7%)
一般図形ボロノイ図の生成	1.74 sec (25.5%)
特徴抽出	1.40 sec (20.5%)
シード生成，延長	0.28 scc (4.1%)
その他	0.01 sec (0.2%)
計	6.82 sec

6. 関連研究

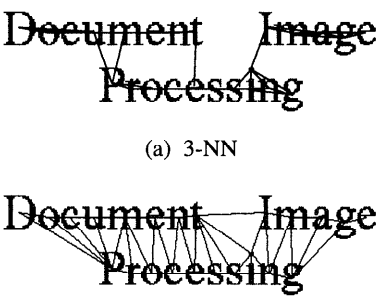
本章では，レイアウトや文字列の傾きによらない，従来の手法[3)~6)]と本手法とを比較する．以下，文書画像の表現法と，表現から文字列を抽出する処理法を区別して議論を行う．

6.1 表現法

文書画像の表現は，連結成分の隣接関係の定義の仕方によって分類できる．単純な方法としては，連結成分間の距離がしきい値以下の連結成分どうしは隣接しているとするものがある．この方法を，効果や効率の面で改善した手法[4),6)]は提案されているものの，しきい値を決めておかねばならないという問題がある．

距離のしきい値が不要な表現法として，O'Gorman は連結成分の k-NN を用いている[3)]．表現法としては，k-NN と隣接グラフとは類似しているが，あらかじめ設定する必要のあるパラメータの安定性に大きな違いがある．隣接グラフにおいて設定すべき値は R と T_n であるが，T_n については O'Gorman の方法でも類似のしきい値が必要となる．そこで，ここでは隣接グラフの R と k-NN の k の安定性について議論を行う．まず，k-NN の k は文書画像ごとに適切に決定しないと，k-NN で得られた平面グラフの中に正しい文字列が含まれない可能性がある．たとえば，**図 13** (a) では，$k = 3$ が不適切であるため，"Document" と "Image" の間の枝が抽出されていない．この枝を抽出するためには，$k \geq 7$ であることが必要である．これは，k-NN の k の適切な値が文書のレイアウトに依存していることを意味する．また，k を大きくすると，文字列に含まれない枝が多く抽出され，k-NN からの文字列抽出の処理に悪影響を及ぼすことが考えられる．それに対し，隣接グラフの R は一般図形ボロノイ図が連結成分の輪郭をどの程度反映するかを示す値なので，文書画像の解像度に依存する値だと考えられる．図 13 (b) は $R = 1/7$ の隣接グラフである．これに示されるように，"Document" と "Image" のように間が離れている単語間の枝も抽出されている．さらに，R は，ある程度細かくサンプリングするように設定すれば，値を変動させても隣接グラフの生成結果に大きな影響を与えない．以上から，隣接グラフの方がパラメータの安定性が高いと考えられる．

ところで，隣接グラフではつねに文字列を構成する枝が抽出可能とは限らない．たとえば，**図 14** のように，単語間よりも上下の行間の方が狭い場合には隣接関係を正しく抽出できない．ただし，このような場合はごくまれであると考えられる．



(a) 3-NN

(b) neighbor graph

図 13 k-NN と隣接グラフ
Fig. 13 k-NN and the neighbor graph.

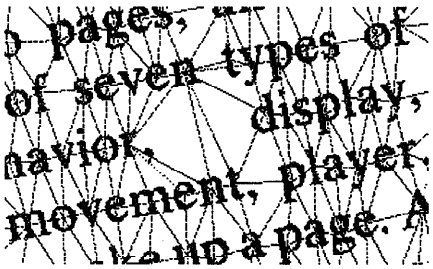


図 14 隣接グラフの限界
Fig. 14 Limit of the neighbor graph.

6.2 処理法

文書画像の表現から文字列を抽出する処理法には，k-NN に対する距離と傾きを用いたフィルタリング[3)]，弛緩法[4)]，ハフ変換[5)]，焼きなまし法[6)]など様々なものが提案されている．特に，弛緩法や焼きなまし法は，全体として適切と考えられる抽出を行うため高い精度が期待できるが，処理時間がかかるという問題がある．本手法は，局所的な判断で文字列を決定していくという，精度よりも効率を重視した処理法をとっている．そこで，ここでは本手法と同様に効率面を重視した処理法である O'Gorman の処理法と本手法との比較を行う．O'Gorman の方法では，はじめに Docstrum を用いて文書画像全体で文字列の傾きを推定するため，1 つの文書画像中に極端に傾きの異なる文字列が複数含まれていたときに対処できない．それに対し，本手法では，文字列の傾きは各シードで推定しているため，そのような場合でも処理することができる．

7. むすび

本論文では，欧文文書画像から文字列を抽出する手法として，隣接グラフを用いた手法を提案した．本手法は，連結成分間の隣接関係に着目し，距離や傾きなどの特徴量を用いて隣接グラフの枝を取捨選択し，文

字列を抽出するものである．本手法の特徴は，隣接グラフを用いることにより，あらかじめパラメータを設定することなく連結成分間の隣接関係を抽出し，文書画像や文字列の傾きに依存しない処理を可能とした点である．本論文では，隣接グラフの有効性を確認するために，処理時間を重視した手法でどの程度の精度を得ることができるかを，様々なレイアウトの文書画像50サンプルを用いて実験し，吟味した．その結果，精度面では課題が残るものの，効率的には優れていることが分かった．

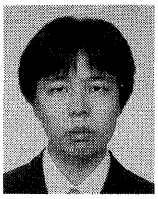
今後の課題としては，隣接グラフを用いたレイアウト解析法の考案があげられる．すなわち，文字列抽出の結果とブロック抽出[8]の結果から互いにフィードバックをかけ，精度を向上させるような手法の考案である．

謝辞 本研究は文部省科研費，電気通信普及財団の補助による．

参考文献

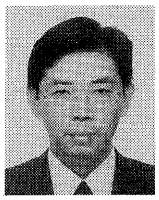
1) 秋山照雄，増田 功：周辺分布，線密度，外接矩形特徴を併用した文書画像の領域分割，電子通信学会論文誌，Vol.J69-D, No.8, pp.1187–1195 (1996).
2) Wahl, F., Wong, K. and Casey, R.: Block Segmentation and Text Extraction in Mixed Text/Image Documents, *Computer Graphics and Image Processing*, Vol.20, pp.375–390 (1982).
3) O'Gorman, L.: The Document Spectrum for Page Layout Analysis, *IEEE Trans. PAMI*, Vol.15, No.11, pp.1162–1173 (1993).
4) Hönes, F. and Lichter, J.: Layout Extraction of Mixed Mode Documents, *Machine Vision and Applications*, Vol.7, pp.237–246 (1994).
5) Fletcher, L. and Kasturi, R.: A Robust Algorithm for Text String Separation from Mixed Text/Graphics Images, *IEEE Trans. PAMI*, Vol.10, No.6, pp.910–918 (1988).
6) Gyohten, K., Sumiya, T. and Babaguchi, N.: A Multi-Agent Based Method for Extracting Characters and Character Strings, *IEICE Trans. Info. & Syst., Japan*, Vol.E97-D, No.5, pp.450–455 (1996).
7) Sugihara, K.: Approximation of Generalized Voronoi Diagrams by Ordinary Voronoi Diagrams, *CVGIP: Graphical Models and Image Processing*, Vol.55, pp.522–531 (1993).
8) Kise, K., Sato, A. and Iwata, M.: Segmentation of Page Images Using the Area Voronoi Diagram, *Computer Vision and Image Understanding*, Vol.70, No.3, pp.370–382 (1998).

(平成10年12月25日受付)
(平成11年 6 月 3 日採録)



岩田 基

昭和50年生．平成9年大阪府立大学大学院工学研究科電気・情報系専攻博士前期課程に飛び級入学，平成11年同修了．平成11年4月より大阪府立大学工学部助手，現在に至る．文書画像理解の研究に従事．電子情報通信学会学会員．



黄瀬 浩一（正会員）

昭和38年生．昭和61年大阪大学工学部通信工学科卒業．昭和63年同大学大学院通信工学専攻博士前期課程修了．平成2年大阪府立大学工学部助手．現在，同助教授．パターン認識，画像理解等の研究に従事．博士（工学）．電子情報通信学会，人工知能学会等会員．

松本啓之亮（正会員）

昭和27年生．昭和53年京都大学大学院工学研究科精密工学専攻修士課程修了．同年三菱電機（株）入社，中央研究所・産業システム研究所等に勤務ののち，平成8年大阪府立大学工学部情報工学科教授となり，現在に至る．この間，主に電力系統の運用・制御システムへの知識工学の適用に関する研究に従事し，最近は，ソフトウェアの研究にも着手．工学博士．昭和58年日本自動制御協会槇木記念賞論文賞，昭和59年電気学会論文賞受賞．電気学会，IEEE等会員．