文書画像のレイアウトモデル作成方式

3J−4

山下晶夫, 豊川和治
日本アイ・ビー・エム 東京基礎研究所

1 はじめに

文書画像のレイアウト解析を行うためには、1ページに内包されるオブジェクトの階層関係を記述した木構造モデルが用いられることが多い。我々は、オブジェクトに関する記述を極力抑えてモデルの定義を容易にした木構造モデルによるレイアウト理解方式の開発を行っているが[1]、このようなモデルをユーザが定義する場合を考えると、階層構造並びに解析に必要なパラメータを正しく記述することは一般には困難である。

本稿では、複雑なモデルの階層構造を視覚的な表示、操作を行うことでユーザーにとってわかりやすくすることと、機械が画像解析して木の作成をある程度までサポートしたり、予め文書画像固有のメタ知識を定めて詳細なパラメータ設定を簡単にすることにより、ユーザーのモデル作成負担を軽減する方式について述べる。

この方式によれば1ページの解析結果から同種のページに対応できるような汎用的なモデルを簡単に作成することができる。

2 レイアウトモデル

一般に文書画像のレイアウトを記述するのには、水平、垂直方向に並んだ矩形の階層構造を用いる例が多い[2]。我々のモデルもこのような矩形の階層構造を基本としている。論文フロントページのレイアウトモデルの定義を表1に示す。

表1: レイアウトモデルの例

Lv	Name	Man	Dir	Element	Min	Max	LS	RS	TS	BS
0	Paper	Yes	Ver	Dummy	1	1	W	W	W	W
1	Header	Yes	Ver	String	1	1	W	W	W	W
1	Title	Yes	Ver	String	1	3	W	W	W	
1	Author	Yes	Ver	String	1	3	W	W		
1	Affil.	Yes	Ver	String	1	3	W	W		W
1	Abstract	Yes	Ver	String	1	10	W	W	W	W
1	Body	Yes	Hor	Dummy	1	3	W	W	W	W
2	Column	Yes	Ver	Dummy	1	1	W	W	W	W
3	M-Block	Yes	Ver	Dummy	1	10	W	W	W	B/W
4	Block	Yes	Ver	String	1	N	W	W	W	B/W
3	Footnote	No	Ver	String	1	6	W	W	B	W
1	Page no.	Yes	Ver	String	1	1	W	W	W	W

表の各行は、フロントページのオブジェクトの一つに対応している。　Lv は木の深さ、Man は、そのノードの表わすオブジェクトがレイアウト中に必ず存在するか (Yes) 否か (No) を表わす。Dir は、子ノードの配置が横並びか (Hor) 縦並びか (Ver) を表わす。また、Element は、そのノードがレイアウト木の非ターミナルノードか (Dummy) ターミナルノードか (String) を表わし、Min,Max は、非ターミナルノードについては子ノードの最小、最大個数を、ターミナルノードについてはオブジェクトが包含する文字列の最小、最大個数を表わす。Separ は、そのノードの構成要素の左右上下 (LS,RS,TS,BS) に明らかに他のオブジェクトとの境となるセパレータが必ず存在するか否かを示す。W が白いセパレータ (広い行間など) を、B が黒いセパレータ (罫線など) を、B/W が B 又は W が存在することを表している。

本方式の目的は、このようなレイアウトモデルを実際の画像解析結果を対話的に修正していくことにより作成していくことにある。

3 モデルの半自動作成

モデルの作成プロセスは文書画像を解析してレイアウト木を作成する部分と、レイアウト木に修正を加え、モデル用のパラメータを木のノードに対話的に設定していくことで汎用的なモデルを作成していく部分に大きく分割される。以下それぞれのプロセスについて詳しく述べる。

3.1 画像解析

連結黒画素領域の追跡や黒ランレングスの組み合わせ法により[3]、文書画像から文字列、罫線、その他の黒画素領域をすべて矩形で表現して抽出することが可能である。本方式では全ての文字列 (或は文字列候補) が矩形領域として抽出されることを前提としている。文字列、縦横罫線等の矩形を基に、白画素領域を囲む矩形集合を求め、一定長さ、幅以上の矩形を縦或は横のセパレータとして登録する。　一定長さ以上の黒罫線も罫線セパレータとして登録する。縦横のセパレータを基にして領域の切り分けを行う。分割に先立って図形領域を対象から除く。図形領域を除いた画像全体について、幾つかの領域に分割する縦のセパレータが見付かれば、これを用いて画像を分割する。次に各領域についてそれらを幾つかの領域に再分割できる横のセパレータがあれば、さらに小さな領域に分割する。このように縦横のセパレータを交互に使いながら再帰的に分割を繰り返し画像全体を木構造を成す領域群に分ける。

A Method for Making a Layout Model of Document Images
Akio YAMASHITA, Kazuharu TOYOKAWA
IBM Research, Tokyo Research Laboratory

各領域に分けられた文字列群から、行、各行毎の文字高さ、ベースラインを求め、これらを基に行ピッチ或は文字サイズが変化する行を推定する。これらの行の上にある行間を横のサブセパレータとして登録する。また縦方向に規則正しくならんだ二つの文字列群を分離する一定の大きさ以上の白画素矩形 (セパレータよりは小さい) が見付かった場合にはこれを縦のサブセパレータして登録しておく。サブセパレータはセパレータにはなりえなかった構成要素の区切りを補完するという性格を持つ。

領域を縦、或は横に区切るサブセパレータを使ってさらに細分化をおこなう。図1の左に実際に画像が分割された結果を、右に木を模式的に表示したものを示す。右の木の各ノードの上下に示したシンボルはセパレータ、或はサブセパレータを表している。木のターミナルノードはいわゆる行になるが、右の木ではターミナルノードの一段上のノードまでしか示していない。一つのターミナルノード (一行分) しか内包していなくても親ノードとなる矩形が必ず存在する。

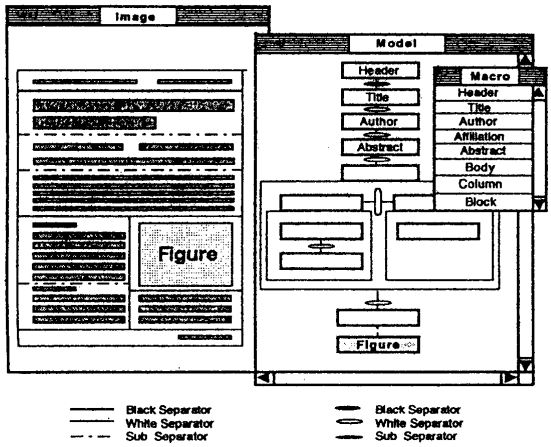

図 1: モデル作成の例

3.2 木の修正とパラメータの設定

画像解析はどの文書画像に対しても同じように行われるので、必ずしても個々のオブジェクトを正しく分割してはいない。このような場合には、例えば、矩形の挿入、削除、グルーピングといったグラフィカルな修正操作により正しい分割に直す。 処理の対象は画像そのものではなく、文字列抽出の結果得られた矩形集合なので、カットアンドペーストのような画像に対する処理ではなく、グラフィックエディターに類似した処理が可能である。

解析の結果求められた木の各ノードはブランクのままなので、モデルにするためには領域名を初めとする各種パラメータをセットする。実画像の解析結果から、上下左右のセパレータの有無、矩形の数等はわかるが、モデルとして登録する場合には、画像ごとにばらつくもの (矩形の数) と比較的安定した特徴 (罫線の有無) を考慮しなければならない。そこでデフォルトパラメータセットとして図1中に示したようなマクロパラメータセットを定義しておいてユーザーに選択させ、実際の解析結果と照らし合わせて変更を加えたものを各ノードにセットする。 マクロパラメータには図中の例に示すように代表的な文書のオブジェクトについてモデル中の各パラメータのデフォルト値がセットされている。

この中で階層構造を示す項目 (Lv 、Dir 、Element) 及びセパレータに関する項目は、解析結果の木構造から親子関係、順序、並び方向等を判断し新しくセットする。領域に内包される矩形の数 (Min 、Max) は解析結果とマクロパラメータ中の値から推定したものをセットする。 このようにして模式的な木のノードにマクロパラメータを対話的にセットしていく。勿論マクロを使わずに各パラメータをセットすることも可能である。

このモデルでは、子ノードの繰り返しを定義できる所に特徴がある。このことによりモデルは汎用性を増し、多くの文書画像をカバーできるようになる。解析の結果得られたレイアウト木はページに対応した一つの木にすぎないが、パラメータセット時に子ノードの繰り返しの最小、最大数を設定し冗長性を持たせることで、同種の他のページに対応できるレイアウトモデルを作成することができる。

4 おわりに

本稿では、階層構造、領域の繰り返し、解析のためのパラメータといったユーザーにとってわかりにくいモデル上での指定を実際の画像解析結果をもとにグラフィカルな設定をしていくことで容易に行う方式について述べた。作成したレイアウトモデルの汎用性については、パラメータにどの程度冗長性を持たせればページ毎の変動にモデルが対処出来るかどうかを今後実験的に確かめていく予定である。

参考文献

[1] A.Yamashita et al.: "A Model-Based Layout Understanding Method for the Document Recognition System (DRS)", Proc. First ICDAR, pp.131-138 (1991).

[2] G.Nagy and S.C Seth: "Hierarchical Representation of Optically Scanned Documents", Proc. 7th Int. Conf. on Pattern Recognition, pp.347-349 (1984).

[3] T.Amano et al.: "A Character String Extraction Algorithm Using Horizontal Boundaries",Proc.SPIE/SPSE, vol 1452-23 (1991).