

プレゼンテーションスライド情報の構造化

羽山徹彩[†] 難波英嗣^{††} 國藤進[†]

[†] 北陸先端科学技術大学院大学
石川県能美市旭台 1-50
^{††} 広島市立大学大学院
広島市安佐南区大塚東 3-4-1

E-mail: [†]{t-hayama,kuni}@jaist.ac.jp, ^{††}nanba@hiroshima-cu.ac.jp

あらまし 近年の電子化プレゼンテーションの普及により、講義や会議などの多くの場で電子的なプレゼンテーションスライドが利用され、蓄積されてきた。蓄積されたスライドデータは膨大かつ重要な情報資源となりつつあり、その高い利活用性が求められている。その有用なアプローチの一つとして、人間の理解を促すための視覚的な構造情報を利用することがあげられる。しかしながら、そのような構造情報は、スライド情報において明確に定義されていないため、計算機で直接的に扱うことができない。そこで、本研究では計算機が構造情報を扱えるようにするために、スライド情報を機能的な纏まりに組織化し、それら纏まりを構造化することを目的とする。提案手法ではスライド情報に含まれる不正確なオブジェクト配置や多彩なレイアウト構造に対して、スライド上のオブジェクトを距離的關係情報と機能的關係情報を用いた組織化と、視覚的情報と属性情報によりトップダウン的に木構造へ組み上げる構造化を行う。評価では、人手で作成した正解データをもとに標準的な構造化手法と比較し、提案手法の有効性を確認した。
キーワード 情報抽出、プレゼンテーションスライド、視覚的レイアウト、Web データ

Structuring Presentation Slide Information

Tessai HAYAMA[†], Hidetsugu NANBA^{††}, and Susumu KUNIFUJI[†]

[†] Japan Advanced Institute of Science and Technology
1-50, Asahidai, Nomi-shi, Ishikawa
^{††} Hiroshima City University
3-4-1, Ozukahigashi, Asaminamiku, Hiroshima

E-mail: [†]{t-hayama,kuni}@jaist.ac.jp, ^{††}nanba@hiroshima-cu.ac.jp

Abstract The widespread uses of electronic presentation are promoting the uses of slides in presentation in numerous scenarios, such as lectures and meetings. In recent years, the accumulate data stored in a slide is increasing as one of the most important information resource. Therefore, it is necessary to develop a practical usage method for the reutilization of the data on the slides. One of the useful approaches could be the visual structure information within a slide, because the visual structure information is one of the most valuable, easy to understand method for human beings. However, since the visual structure information is not defined explicitly in the slide data itself, it is difficult for computers to comprehend the structure information directly. In this paper, we propose a method of organazing and structuring to extract the structure information from within the information of the slides. The proposed method is composed of two steps, organizing primitive objects within the slide of the units as an attribute by both of object information with position and one with an attribute, and structuring the units as a hierarchy tree based on a top-down approach by visual information and attribute information. Our experiment result shows that the proposed method can extract the structure information from the slide information.

Key words Information Extraction, Presentation Slide, Visual Layout, Web Data

1. はじめに

本論文ではプレゼンテーションで使用されたスライドの利活用性を高めるために、スライドに含まれる情報を自動的に構造化する手法について述べる。

近年の電子化プレゼンテーションの普及により、講義や会議などの多くの場において電子的なプレゼンテーションスライドが利用されるようになった。利用されたスライドは遠隔講義資料や Web コンテンツとして逐次的に蓄積され、その結果としてスライドデータは膨大かつ重要な情報資源となりつつある。そのようなスライド情報に対して、情報アクセスやデータ加工を容易にする高い利活用性が求められている。

現在のスライドデータはそのまま保存され、その中のテキストに含まれる単純なキーワードによって検索されることが一般的であり、レイアウトや視覚的情報などの人が見て理解しやすくなるための有意な構造情報を排した管理がなされている。このような構造情報を保持したデータ管理が行えれば、より高度な情報処理が可能になるが、構造情報は明確に定義されていないため計算機で直接的に扱うことができない。また、人手により構造情報を付与することは膨大なコストがかかるため、計算機による自動的な構造情報の抽出が望まれている。

これまでドキュメントの利活用性を高めるために、様々な対象をもとにした構造化手法が研究されてきた。従来研究の多くは、Web ページの HTML タグ [1], [8], [9], 枠線を持つ帳簿票 [2], [5], [7], および社報 [6] など比較的整った文書形式を対象として、有効な成果が得られてきた。しかしながら、スライド情報を対象とした場合には、不正確に配置された構成要素や多様なレイアウト構造が含まれるため、従来手法をそのまま適用し構造化することが困難である。

そこで、本研究では計算機がスライドに含まれる構造情報を扱えるようにするために、スライド情報を対象とした構造化手法を提案する。提案手法は、まずスライド情報を機能的な纏まりに組織化し、次にそれら纏まりを構造化することを目的とする。スライド情報に対し情報の纏まりやそれらの関係性がメタデータとして新たに付与されることで、計算機が一度にアクセスすべき情報領域やその情報に付随する関連情報を扱うことができ、その結果として情報検索やオーサリングツールなどへの高度なスライド情報処理が実現可能となる。

2. アプローチ

2.1 スライド情報とその構造

スライドに含まれる情報は、「テキスト」、「写真」、「グラフ」および「基本図形」などのプリミティブなオブジェクトから構成され、それらオブジェクトが、「タイトル」、「本文」、「図」、「表」および「装飾」といったスライド情報を伝えるための機能的な纏まり(ユニット)を成している。図1の例では、(A)、(C)および(F)は、すべてテキストタイプのプリミティブなオブジェクトであるが、それぞれ「タイトル」、「本文」、および「図」と異なる

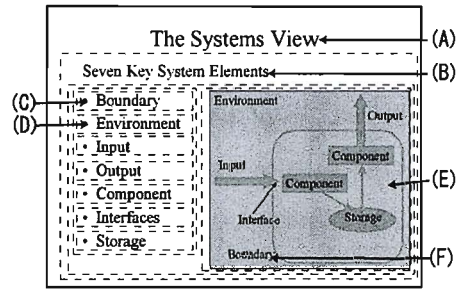


図1 スライドに含まれる情報をもつ構造の例
(点線に囲まれたオブジェクトはユニットの階層関係を表している。)

属性をもったユニット^(注1)として認識することができる。その際、(F)は(E)などの複数のオブジェクトとともに、1つの図属性をもつユニットを成している。そのように、たとえ同じタイプのオブジェクトであっても、それぞれが異なる属性であったり、複数のオブジェクトが纏まりを成すことで属性を持つことができたりすることがある。

スライド情報の構造は、そのようなユニットに対し、レイアウトや装飾などの視覚的情報を手掛かりとして関係付けられることで、階層的な木構造として表現することができる。図1の例において、(A)がタイトル属性のユニットとして認識できた場合にはタイトルがスライド全体の内容を表示しているため、そのユニットをルートオブジェクトとして割り当てることができる。また、(B)と(C)は字下げによって、親子ノードとして関係付けることができ、(C)と(D)は同レベルの箇条書き項目のため、兄弟ノードとして関係付けることができる。さらに、囲み線(領域)にいくつかのユニットが内包されていた場合には、それらユニットの集合を部分木として表現することができる。

2.2 構造化

前節で述べたように、スライドに含まれる情報の構造はプリミティブなオブジェクトに対し、以下の処理手続きを行うことで、獲得することができる。

- Step 1: プリミティブなオブジェクトを機能的な纏まりとするユニットに組織化する。
- Step 2: ユニットの階層木として構造化する。

Step 1 の処理では、[4]と同様にオブジェクト間の距離の近さや重複といったオブジェクトの距離的な関係情報を利用することができる。しかしながら、それらオブジェクトはスライド作成者が作成する際に、マウス操作などで自由配置されているため、不適切な重複や位置のずれが生じることもあり、このような場合には距離的な関係情報だけで適切なグループへ割り当ることが難しい。このような問題に対し、組織化の際にはオブジェクトの距離的な関係情報だけでなく、機能的な関係情報も利用することができる。例えば、テキストタイプのオブジェクトと基本図形タイプのオブジェクトが重複している場合には、

(注1): 本論文の「ユニット」は、1つ或いは複数のオブジェクトから構成された属性を持つことができる構成単位のことを意味する。

そのテキストタイプのオブジェクトが箇条書き項目の1つとする機能的な関係があるなら、そのオブジェクトを本文属性とするユニットととして適切に割り当てることができる可能性がある。

Step 2 の処理では、一般的なドキュメント構造化手法として、レイアウトテンプレート照合による方法と視覚的な情報に含まれる規則性に基づいた方法がある。前者はスライドの多様なレイアウトテンプレートを網羅的に準備する必要性から現実的に採用することが難しい。そのため、本研究では後者の視覚的な情報に含まれる規則性に基づいた方法を探る。しかしながら、ユニットを関係付ける規則性を検出するためには、レイアウトや囲みなどの視覚的な情報を手描かりとして用いることが有効であるものの、その情報を利用するだけでは不十分である。例えば、図属性をもつユニットがスライドの大領域を占めていた場合、多くのオブジェクトを1枚のスライドへ詰め込むために、視覚的なレイアウトの規則性が損なわれるかもしれない。そのような問題を補うための1つの方法として、ユニットの属性情報を利用することがあげられる。予めレイアウトの規則性に沿わないオブジェクトの属性やその並びを考慮することで、視覚的な情報を持つ規則性を保持できる可能性がある。

3. 提案手法

本節では、スライド情報を構造化するための提案手法について述べる。提案手法は、1枚のスライドに含まれるプリミティブなオブジェクトに対する、以下の処理手続きから構成される。

- (1) 機能的関係を考慮したスライド情報の組織化
- (2) トップダウン的アプローチに基づいた構造化

3.1 機能的関係を考慮したスライド情報の組織化

スライドに含まれるオブジェクトを組織化するために、まず機能的な関係により各オブジェクトの属性を決定し、次に距離的な関係によりオブジェクトのユニットを検出する。また、属性の決定では、より明確であるオブジェクトから順に行っていくとともに、それと機能的な関係があるオブジェクトの属性決定にも影響を与えていく。

その処理手順を以下に示す。

1) オブジェクトの属性ごとに得点を付ける。

各属性の尤もらしさを評価項目とした表1の得点表を用いて、各オブジェクトの属性ごとに得点を付ける。その際、他のオブジェクトと関係し、評価される項目(表1の下線項目)では、属性ごとにその関係オブジェクトを機能的関係としてリスト化を行う。

図2の例では、Object(a)の属性[タイトル, 本文, 図, 表]に対して、[5, 2, 0, 0]と得点付けされる。また、Object(d)の本文属性および図属性の機能的関係リストには、Object(c), および(e)と(f)が含まれる。

2) オブジェクトの属性を特定する。

オブジェクトの属性を特定するために、最も属性が特定しやすいオブジェクトの属性を決定していく。その際、そのオブジェクトの属性と機能的関係をもつオブジェクトの属性の得点に対

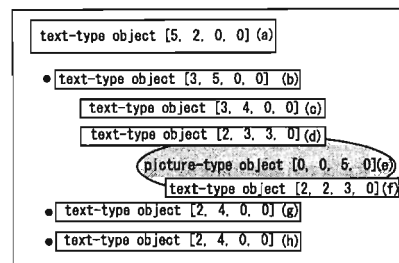


図2 各属性を得点付けるスライドの例
(“[]”に含まれる数字は、それぞれタイトル, 本文, 図, および表の属性得点を示している。)

しても変更を行う。

その処理手順を以下、2.1 から 2.3 に示す。

2.1) 属性が未決定であるオブジェクトの中から最も属性が特定しやすい属性をもつオブジェクトを選出する。

まず、属性が未決定な各オブジェクトに対し、4つの属性得点から最高得点のもつ属性を候補属性 (*attri_cand*) として設定する。そして、種類が異なる候補属性をもつオブジェクト間でも属性が特定しやすさを比較できるように、式(1)と(2)を用いて、オブジェクトごとに候補属性の尤もらしい程度 (*Li_Attri*) を算出する。

$$Ev_{(attri)} = \begin{cases} Attri_Val_{(attri)} \\ (if\ attri_cand == attri) \\ MaxScore_{(attri)} - Attri_Val_{(attri)} \\ (otherwise) \end{cases} \quad (1)$$

$$Li_Attri = Ev_{('title')} * Ev_{('body-text')} * Ev_{('figure')} * Ev_{('table')}. \quad (2)$$

ここで、*attri*, *Attri_Val*_(*attri*), および *MaxScore*_(*attri*) は、ある属性名、その属性に付けられた得点、およびその属性の最大得点(注2)を示している。図2の例では、Object(b), (g) および (e) の候補属性の尤もらしい程度は、375, 300, および 15625 が算出され、その結果、object(e) が他の二つの object よりも属性が特定しやすいといえる。

最後に、属性の尤もらしい程度が最大であるオブジェクトを選出し、その候補属性を属性として決定する。その際に、尤度の値が最大となるオブジェクトが複数あった場合には、そのなかで最も左上位置にあるオブジェクトを選択し、その属性を決定する。

2.2) 機能的関係があるオブジェクトの属性決定へ影響を与える。

2.1) において、あるオブジェクトの属性が決定されたが、それ以外の属性に対して機能的関係にあるオブジェクトはそのオブジェクトと無関係であるため、その関係を解消するように属

(注2) : 表1の得点表においてはすべての最大得点が5である。

表 1 属性の尤もらしさに基づいた得点項目

タイトル属性のための評価項目		本文属性のための評価項目	
・フォントの大きさ > $Threshold(fontsize1)$	+1	・箇条書きシンボル有	+1
・トップからの距離 > $Threshold(y_axis_position)$	+1	・同じ左位置で同じフォントのテキストタイプのオブジェクトがある	+1
・スライド上のオブジェクトの最上位に位置する	+1	・左上/右下の位置にテキストタイプのオブジェクトがある	+1
・スライドに含まれる中で最大のフォントサイズ	+1	・フォントサイズ > $Threshold(fontsize2)$	+1
・文字数 > $Threshold(number_of_characters)$	+1	・文字数 > $Threshold(number_of_characters)$	+1
図属性のための評価項目		表属性のための評価項目	
・グラフ/図タイプのオブジェクト	5	・表に含まれるセルの半数以上にデータが含まれている	5
・完全にグラフ/図タイプのオブジェクトに重複している	4	・表に含まれるセルの半数以下にデータが含まれている	4
・部分的にグラフ/図タイプのオブジェクトに重複している	4	・完全に表のセル領域に重複している	4
・間接的に図表タイプのオブジェクトと重複する	3	・部分的に表のセル領域に重複している	3
・グラフ/図の重複したグループの中でトップ/ボトムに位置するテキストタイプのオブジェクト	-1	・間接的に表のセル領域と重複している	3
・テキストを含まない基本図形	4		
・文字数 < $Threshold(number_of_characters)$	+1		

$Threshold(fontsize1)$ および $Threshold(fontsize2)$ はそれぞれ文字サイズのパラメータを、 $Threshold(y_axis_position)$ および $Threshold(number_of_characters)$ は上からの位置と文字数のパラメータを示しており、さらに下線部は他のオブジェクトとの関係によって評価される項目を表している。

性得点を改める必要がある。その手順としては、2.1) で各オブジェクトの属性ごとに作成した機能的関係リストを利用して、以下の規則に基づき他のオブジェクトの属性得点を変更する。

- 属性決定されたオブジェクトがタイトル属性とされたなら、属性未決定のオブジェクトのタイトル属性の得点を 0 に改定し、そのオブジェクトのタイトル属性以外の機能的関係リストに含まれるオブジェクトの属性得点を 1 減点する。
- 属性決定されたオブジェクトがタイトル以外の属性とされたなら、そのオブジェクトの決定属性以外の機能的関係リストに含まれるオブジェクトの属性得点を 1 減点する。

図 2 の例において、オブジェクト (d) が最初にタイトル属性として決定された場合、すべてオブジェクトのタイトル属性の得点は 0 に設定される。また、オブジェクト (d) が本文属性として決定された場合、オブジェクト (d) の図属性の関連リストに含まれている属性未決定オブジェクト (e) と (f) の図属性は 2 および 1 に再設定される。

2.3) すべてのオブジェクトの属性が決定されるまで、2.1) と 2.2) の処理を繰り返す。

3) 距離関係情報に基づきオブジェクトを組織化する。

すべてのオブジェクトの属性が特定された後、図/表属性とするオブジェクトは図/表属性の機能的関係リストに含まれ、図/表属性を持つオブジェクトと纏められ、それを 1 つのユニットとする。

4) 装飾属性をオブジェクトに割り当てる。

すべてのオブジェクトの属性を特定した後、以下の 2 種類のオブジェクトに対し、装飾属性へ変更する。

- ・ 本文属性のオブジェクトを内包する基本図形タイプのオブジェクト
- ・ 単体で図属性をもつ矢印タイプのオブジェクト

3.2 トップダウン的アプローチに基づいた構造化

スライドに含まれる視覚的な構造の規則性を検出することで、ユニットを木構造として構造化する。本研究では、トップダウン的な領域分割によって構造化を行う。つまり、ユニットが含まれるブロック領域を段階的に分割し、その結果として各分割段階の前後のブロックで親子ノード関係とする構造を得ることができる。

本構造化の手順を以下に示す。

0) 前処理

ユニット間の重複する領域は視覚的レイアウトの規則性の検出を阻害するため、まずユニット間の境界を以下の規則に従って、生成する。

- ・ 本文属性と図/表属性のユニットの領域が重複していた場合、図表属性のユニット領域を重複がなくなるまで縦横に削る。
- ・ 本文属性のユニット間の領域が重複していた場合、各ユニットの領域を中間位置まで削る。

1) 初期設定

初期ブロックとルートノードをタイトル属性のユニットに基づき設定する。タイトル属性のユニットが含まれている場合には、そのユニットをルートノードに割り当て、初期ブロックをタイトル属性のユニット以下に位置するすべてのユニットを含む領域とする。タイトル属性のユニットが含まれない場合には、ルートノードにユニットを割り当てず、初期ブロックをユニット全体が含まれる領域とする。

2) 縦方向へのブロック分割

ブロックを縦断する空領域が含まれている場合には、それぞれの空領域によってそのブロックを分割する。ブロックに空領域が含まれていない場合には、この処理を実行しない。

3) 横方向へのブロック分割

横方向へのブロック分割は、以下の 3 つの段階からなり、順

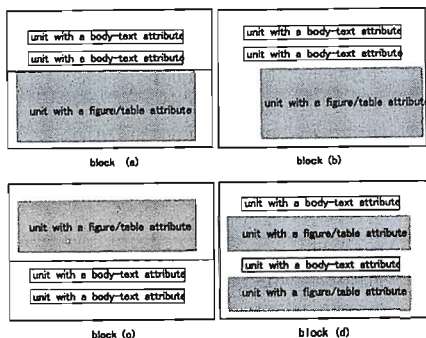


図3 ブロックに含まれるユニットの属性の並びとその分割点の例

に適用される。

- 横断する空領域による分割
- 属性系列による分割
- 箇条書き項目による分割

横断する空領域による分割 ブロックに横断するある程度幅のある空領域が検出された場合には、それぞれの領域で分割し、次の処理 4)へ進む。

属性系列による分割 ブロックに含まれるユニットの属性の並びを調べ、その属性列が本文属性と図表属性との関係が以下の (i), (ii) のヒューリスティックな規則に適合した場合には、各分割規則を適用し、次の処理 4)へ進む。

(i) ブロックの最上位置にあるユニットの属性が本文属性であり、そのユニット位置より左となる図属性のユニットがブロックに含まれている場合には、そのブロックをその図属性の上位置で 2 つに分割する。ただし、最上位置にある本文属性のユニットが箇条書き項目の 1 つであり、その図属性が箇条書き項目の間に位置する場合は適用しない。

(ii) ブロックの最上位置にあるユニットの属性が図表属性である場合には、そのブロックをそのユニットの下位置で 2 つに分割する。

図 3 の例において、規則 (i) により、block(a) は 2 つに分割されるが、block(b) と (d) は分割されない。また、規則 (ii) により、block(c) は分割される。

箇条書き項目による分割 ブロックの最上位置のユニットが本文属性の箇条書き項目である場合には、そのブロックを各箇条書き項目の上位置で分割し、次の処理 4)へ進む。

以上において、どの横方向へのブロック分割処理が適用されなかった場合には、ブロックの最上位置にある本文属性のユニットと、それ以下のユニットとの 2 つのブロックへ分割する。

4) すべてのブロックが高々 1 つのユニットが含まれるようになるまで、2) と 3) の分割処理を繰り返す。

4. システムの概要

提案手法の一連の処理を実現するためのシステムを Microsoft Visual Studio C# によって開発した。開発システムは、Microsoft PowerPoint 形式のスライドファイルを入力として、

「前処理」、「組織化」および「構造化」の一連の手続きを行った後、プリミティブなオブジェクトの情報に対しメタデータとして機能的な纏まりの情報とそれらの構造情報を付与した XML 形式のデータファイルを出力する。

前処理では独自に開発したフィルタプログラムにより、スライドファイルからプリミティブなオブジェクトを、位置、色、テキストのフォントとサイズなどの物性情報とともに抽出することを行う。さらに、テキストタイプのオブジェクトに対しては、箇条書き項目や複数行に渡る一文の判定を行った。箇条書き項目判定では、テキストの先頭（左側）付近に小領域の基本図形がある場合、その図形を箇条書きシンボルと見なし、そのテキストとともに箇条書き項目とした。複数行に渡る一文判定では現在のシステムが日本語に対応しているため、上下位置に隣接するとともにフォントとそのサイズが同じテキストに対し、係り受け関係を用いた一文判定手法を適用している。

5. 評価実験

5.1 方法と準備

提案手法の有効性を確認するために、以下の 2 点について検証することを目的とする。

- 距離的關係情報だけでなく、機能的關係情報も利用することで、スライドに含まれるプリミティブなオブジェクトを効果的に組織化できるかどうか。
- 視覚的情報と属性情報を利用することで、トップダウンのアプローチにもとづきユニットが構造化できるかどうか。

組織化と構造化では、以下の式で表される *Precision*, *Recall*, および *F-measure* によって、提案手法と標準的な手法との比較実験を行った。本実験で用いた標準的な手法は、提案手法の距離的關係情報だけを用いた組織化を行う。具体的には、フォントサイズが最大のテキストタイプのオブジェクトをタイトル属性とし、図タイプのオブジェクトとある程度の距離範囲内があるオブジェクトを図属性のユニットとして組織化を行う。また、オブジェクトの属性決定が機能的關係するオブジェクトへ相互的に影響を与える提案手法の機能 (3.1 節の手順 2.2) の有効性も確認するために、その機能の有無についても比較した。

$$Recall = \frac{Matched_CorrectData}{Total_CorrectData} \quad (3)$$

$$Precision = \frac{Matched_CorrectData}{Total_DetectedData} \quad (4)$$

$$F-measure = \frac{2 * Recall * Precision}{Recall + Precision} \quad (5)$$

ここで、*Matched_CorrectData*, *Total_CorrectData*, および *Total_DetectedData* は、正解データに適合した数、正解データの総数、および検出されたデータの総数を示す。

実験データは、Web 上から自動収集された日本語で書かれた平均 25.1 ページの 10 組のスライドを収集し、用いた。その正解データは、独自に開発した編集用ツールによって人手で纏まりとその属性、およびそれらの構造情報を付与することによって、作成された。

表 2 組織化処理における属性ごとの精度

属性名 (オブジェクト数)		タイトル (246)	本文 (893)	図 (199)	表 (9)	装飾 (228)
提案手法	Recall	0.98	0.87	0.80	1.00	0.89
	Precision	0.99	0.84	0.74	1.00	0.91
	F-measure	0.99	0.85	0.77	1.00	0.90
提案手法 (<i>Func_Rel</i> を利用しない場合)	Recall	0.98	0.85	0.75	1.00	0.87
	Precision	0.99	0.85	0.69	1.00	0.81
	F-measure	0.99	0.85	0.72	1.00	0.84
標準的な方法	Recall	0.87	0.73	0.51	1.00	0.85
	Precision	0.99	0.84	0.53	0.90	0.91
	F-measure	0.93	0.78	0.52	0.95	0.88

Func_Rel はオブジェクト同士の機能的な関係情報を表している。

表 3 構造化処理における各スライドの正解率ごとのスライド数の割合

各スライドの正解率の範囲	1.00	0.99-0.80	0.79-0.60	0.59-0.00	N/A
提案手法	0.68	0.02	0.10	0.16	0.04
提案手法 (<i>Func_Rel</i> を利用しない場合)	0.62	0.03	0.10	0.21	0.04
標準的な手法	0.53	0.03	0.05	0.35	0.04

5.2 結果と考察

組織化および構造化の実験結果を、それぞれ表 2 と表 3 に示す。

表 2 が示すように、提案手法は他の手法よりも、すべての属性において高い精度で組織化できることがわかった。特に図属性の検出に関しては他の属性に比べ、提案手法が良好な結果が得られていた。このことは、機能的関係情報を用いた提案手法が距離的な関係情報だけで対処できないオブジェクト間の不適切な重複に対しても、ある程度適切に組織化できるといえる。

表 3 は構造化処理を行った結果をもとに、各スライドに含まれるユニットの関係がどの程度の割合で正確に抽出できていたかを示している。その結果として、提案手法が完全に構造化できたスライド数の割合が 0.68、80% 以上の正確さで構造化されたスライド数の割合が 0.70 であり、それに対して標準的な手法が 80% 以上の正確さで構造化できたスライド数の割合が 0.56 であった。そのため、提案手法が正確に属性を特定できることに加え、それらのデータをより良く構造化することができるという。

最後に、我々は実験で得られた提案手法の結果に含まれるエラーの原因を実際に確認した。その問題の 1 つとして、オブジェクト間の関係を視覚的なレイアウトでなく、テキストの記述内容で定義されている場合があげられる。この問題に対処するためには視覚的な構造情報だけでなく、テキストの内容分析も考慮した構造化が必要となる。

6. さいごに

本論文では、膨大かつ重要な情報資源となりつつあるスライドデータの利活用性を高める技術として、スライドに含まれる情報を自動的に構造化する手法について述べた。提案手法では、スライドに含まれる不正確なオブジェクト配置や多彩なレイアウト構造に対して、オブジェクトの距離的關係情報と機能的關係情報をもとにスライドに含まれるプリミティブなオブジェクトを機能的な纏まりに組織化し、それら纏まりを視覚的情報と

属性情報にもとづきトップダウン的に木構造として構造化することを行った。評価では、人手で作成した正解データをもとに標準的な構造化手法と比較し、提案手法の有効性を確認した。

今後の課題としては、視覚的な情報だけでなく、テキストの内容を考慮することで提案するスライド情報の構造化手法を改善していくことがあげられる。また、提案手法により抽出された構造情報をスライドデータのメタデータとして付与させることで、スライドデータの検索、閲覧、および編集などを支援する応用アプリケーションや、スライド情報を分析する手法 [3] を開発していきたい。

文 献

- [1] 南野朋之, 斎藤豪, 奥村学: 繰返し構造に基づいた Web ページの構造化, 情報処理学会論文誌, Vol.45, No.9 (2004).
- [2] 成瀬 博之, 渡辺 豊英, 駱 琴, 杉江 昇: 梓野線情報を用いた帳票文書の構造認識, 電子情報通信学会論文誌 D-II, 75(8), pp.1372-1385 (1992).
- [3] Tessai Hayama, Hidetsugu Nanba, Susumu Kunifuji: Alignment between a Technical Paper and Presentation Sheets Using a Hidden Markov Model, AMT2005(Active Media Technology) (2005).
- [4] Tatsuya Ishihara, Hironobu Takagi, Takashi Itoh and Chieko Asakawa: Analyzing Visual Layout for a Non-Visual Presentation-Documents Interface, In Proc. the ACM ASSETS 2006, pp.165-172 (2006).
- [5] 駱 琴, 渡辺 豊英, 杉江 昇: 多種帳票文書の構造認識, 電子情報通信学会論文誌 D-II, J76-D-2(10), pp.2165-2176 (1993).
- [6] Binyamin Rosenfeld, Ronen Feldman, Yonatan Aumann: Structural extraction from visual layout of documents, CIKM '02: Proceedings of the eleventh international conference on Information and knowledge management(2002).
- [7] Watanabe, T., Luo, Q., Sugie, N.: Layout Recognition of Multi-Kinds of Table-Form. Documents, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol.17, No.4, pp.432-445 (1995).
- [8] Y. Yang and H. Zhang: HTML Page Analysis Based on Visual Cues, In Procs. of the Sixth International Conference on Document Analysis and Recognition, pp.859-864, pp.10-13 (2001).
- [9] Yanhong Zhai and Bing Liu: Structured Data Extraction from the Web Based on Partial Tree Alignment, IEEE Transactions on Knowledge and Data Engineering, Vol.18,