Structure Extraction from Presentation Slide Information

Tessai Hayama[1], Hidetsugu Nanba[2], and Susumu Kunifuji[1]

[1] Graduate School of Knowledge Science
Japan Advanced Institute of Science and Technology
1-1, Nomishi, Ishikawa, Japan
{t-hayama,kuni}@jaist.ac.jp
[2] Faculty of Information Sciences
Hiroshima City University
3-4-1, Ozukahigashi, Asaminamiku, Hiroshima, Japan
nanba@its.hiroshima-cu.ac.jp

Abstract. Electronic presentations are used in numerous scenarios, such as lectures and meetings. In recent years, the widespread use of electronic presentations means that presentation slide data is increasing as one of industry's most important information resources. Therefore, it is necessary to develop a practical usage method for the reutilisation of the data on slides. An approach to achieve this is to focus on visual structure information within a slide, because visual structure information is one of the most valuable, easy to understand methods for humans. However, since visual structure information is not explicitly defined in the slide data itself, computers have difficulty comprehending structure information directly. In this paper, we propose a method of extracting structure information from slide information. The proposed method is composed of two steps: organising objects within the slide as units, such as title, body text, figure and table, and structuring the units as a hierarchy tree based on a top-down approach.

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

1 Introduction

The widespread use of electronic presentations is increasing the number of slides that businesses accumulate. Since used slides are often stored and reused as e-Learning or Web content, the data stored on slides is rapidly becoming one of industry's most important information resources. Therefore, it is necessary to develop a practical usage method for the reutilisation of the data stored on slides. One approach to preparing a slide data reutilisation system is to use visual structure information within a slide. Most systems currently handling slide data convert the slide data into simplified text data, and then users access the data by using a linear sequence of words. Although a slide's visual structure information, such as visual form and layout, is valuable to easily understanding the context of the data, current slide data reutilisation systems always ignore such information. If a system could handle the text data with its structural information, then the system would be able to facilitate the intelligent processing of slide data. Structure information representing the relationships among the data entities is not explicitly defined in the slide however. Therefore, it is not easy to manually add the definition to all existing slide data. Thus, we need to develop technology to automatically extract structure information from the slide data.

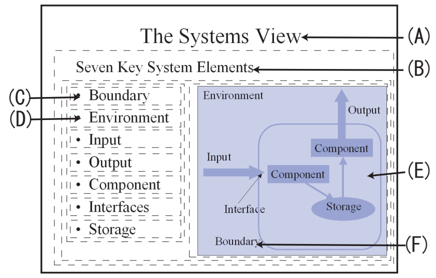
Several methods for extracting structure information from documents have been proposed[8, 7, 1]. Rosenfeld et al.[6] and Zhai et al.[9] suggested a structure extraction method for PDF and Web documents using probabilistic approaches, such as the machine-learning and tree-graph-matching algorithms, respectively. These approaches need to prepare a large amount of annotated data, and the models made from the data are dependent on the data. Although it is useful to adapt target data containing a few types of information structure, it is difficult to adapt target data containing various types of information structure, such as on slides. Nanno et al.[5] proposed a method of extracting structure information from Web pages using the repetition of elements within the Web page. However, the method is inapplicable for slide structure extraction because slide data does not include an explicit regular element, such as an HTML tag. Ishihara et al.[3] proposed an extraction method based on close distances among the objects within a slide to analyze structure information focusing on diagrams within a slide. Because the objects within the slide can be created freely and then manually allocated on each slide, slides sometimes include objects in incorrectly overlapped positions. Thus, Ishihara et al.'s method cannot analyze the structure information appropriately. Although the previous methods can effectively extract structure information from a document with formal formatting, they cannot extract structure information from the information on a slide that has a varied layout structure and incorrectly placed objects.

In this paper, we propose a method of extracting structure information from the information on slides. The proposed method is composed of two steps: organising primitive objects within the slide as units, such as title, body text, figure and table, and structuring the units as a hierarchy tree based on a top-down approach. Knowledge of slide structure is useful in various applications. For example, the knowledge of structure represents the visual structure that the current slide readers cannot utilise, so that it allows blind users to understand presentation documents more easily. In addition, a system that presents slides on hand-held devices with small display screens can use the structure for segmentation and assign a human-like layout to the segmented slide information.

2 Definition and Problems for Structure Extraction

2.1 Slide Information and Its Structure

Slide information is composed of one or more primitive objects, such as texts, pictures, lines and basic diagrams. Each object is recognised as a functional attribute, such as title, body text, figure, table and decoration. For example, as



The Systems View

Seven Key System Elements

- Boundary
- Environment
- Input
- Output
- Component
- Interfaces
- Storage

Fig. 1. An example of slide information and its structure. (Object(s) boxed by a dashed line represent a unit able to function as an attribute.)

shown in Fig. 1, (A), (C) and (F) are primitive text-type objects and have the functional attributes of title, body text and figure, respectively. In such a case, (F) is recognised with (E) as a unit of figure attribute. Thus, even if the objects are similar in type, as in the example, they may function either as different attributes or as a group of two or more objects, but not as a single object.

The structure of information within a slide can be represented as a hierarchy tree of the units in which the object(s) can function as an attribute: a set of similar units. To detect the relationships between the units, cues such as the slide's visual form and layout can be used. For example, as shown in Fig. 1, object (A) with a title attribute can be assigned to the root node. Objects ((B) and (C)), related by indent, and objects ((C) and (D)), itemised in same level, can be assigned to the parent-child nodes and sibling nodes, respectively. In addition, a text box can be represented by a partial tree composing the objects within the box. Thus, the visual cues within a slide provide the relationships between the units to detect the tree structure.

2.2 Extracting Structure Information from Slide Information

As described in the previous section, structure information within a slide is built on primitive objects by the following steps: (1) Organising primitive objects within a slide into units that can function as an attribute and (2) structuring the units as a hierarchy tree.

In step 1, to detect the units that are able to have an attribute, the close distance and the overlap between the objects can be used as described in [3]. However, since the objects are freely created and then manually allocated on the slide, incorrect object placement is inevitable and thus may fail to lead to the detection of the separations between the units. To detect the separations between the units, not only the distance relationships but also the functional relationships between the objects can be used. For example, if text- and diagram-type objects overlap, the text-type object can be properly identified as a body-text attribute using a bullet point list. Therefore, inappropriate organisation of the slide can be eliminated.

In step 2, the structuring of the visual layout is often used as an approach based on the regularity of the unit relationships as well as the matching of template layouts. Since it is difficult to prepare a large collection of layout templates as in the latter approach, we will apply the former approach to structure the units. Although it is useful to use the visual cues, such as indents and bullet point lists, to detect the regularity of the relationship of the units, it is inappropriate to use them alone. For instance, if a figure object is allocated in a large area of the slide, the regularity with the visual layout might be disturbed. To compensate the regular disturbance, the system can also use attribute information with the units. Even if the figure object disturbs the layout regularity, we can maintain layout regularity using the regularity of units of the object.

3 Proposed Method

We propose a method of extracting structure information from the information within a slide. The method consists of the following stages. First, organising primitive objects within a slide into units using function relationships among the objects. Second, structuring the units based on a top-down approach.

3.1 Organising Information in a Slide into Units Using Functional Relationships

To identify an uncertain attribute within an object, we assumed that the attribute could be determined by the functional relationships between that object and the other objects on the slide with more certain attributes. Therefore, this proposed method assigns the likely attribute within each object, and then determines the attribute of the object in order of the object with the most obvious attribute, which also affects the determination of the attribute(s) of the functionally related object(s). The organisation can be achieved with the following procedures.

1) Assigning the likelihood of each attribute within each object: The score of each attribute within each object is assigned using a score sheet, as shown in Table 1. The score sheet is made based on the type, position and size of an object with the distinction of each attribute. The points within the sheet are scored according to the following rules: an object with the properties that indicate the likelihood of each attribute is given points for each attribute. If the type of the object influences the scoring for each attribute to a greater degree than the object's position and size, an object functionally related as the likelihood of each attribute is given points for each attribute.

The score is calculated by the number of items that are matched with the object. For example, as shown in Fig. 2, object (a) has attribute scores of title, body text, figure and table as 5, 2, 0 and 0, respectively. In addition to the assignment, the functional relationship of each attribute within each object is listed if the scoring of each attribute is used as a relation to the other object(s), such as the items underlined in the score sheet.

Table 1. Score sheet of attribute based on the likelihood of the attributes

Items for title attribute		Items for body-text attribute	
With font size > $Threshold_{(fontsize1)}$	+1	With bullet symbol	+1
With position from the top > $Threshold_{(y\_axis\_position)}$	+1	Existing text-type object(s) with similar format and at the same left-position.	+1
With the nearest top-position in the slide	+1	Existing the other text type object on the position of upper left/lower right of it	+1
With the largest font size in the slide	+1	With font size > $Threshold_{(fontsize2)}$	+1
With number of characters > $Threshold_{(number\_of\_characters)}$	+1	With number of characters > $Threshold_{(number\_of\_characters)}$	+1
Items for figure attribute		**Items for table attribute**	
Graph/Picture object	5	With number of data more than half of cells within table	5
Complete overlapping G/P_Obj	4	With number of data less than half of cells within table	4
Partially overlapping G/P_Obj	4	Complete overlapping cell area within table	4
Overlapping G/P_Obj indirectly	3	Partially overlapping cell area within table	3
Text-type object at top/down position among a group overlapped G/P_Obj directly/indirectly	-1	Overlapping cell area within table indirectly	3
Diagram object with no text	4		
With number of characters < $Threshold_{(number\_of\_characters)}$	+1		

$Threshold_{(fontsize1)}$, $Threshold_{(fontsize2)}$, $Threshold_{(Y_axis_position)}$ and $Threshold_{(number\_of\_characters)}$ are represented parameters of font size, font size, distance from the top-position and number of characters, respectively, G/P_Obj and underlined items indicate graph/picture object and the scoring using relationships among other object(s), respectively.

(2) Identifying the attributes of the objects: By detecting an object with a maximum likelihood of an attribute, the attribute of the object is determined and then the other object(s) functionally related to this other object is affected. The process consists of the following three steps: 2.1, 2.2 and 2.3.

- **(2.1) Detecting an object with maximum likelihood of an attribute among the objects with non-determined attributes:** First, each object with a non-determined attribute is set as a candidate attribute (*attri_can didate*), which is one with the largest scores among the four attributes. The likelihood of the candidate attribute contains not only the likelihood degree of the candidate attribute but also the unlikelihood degree of the other attributes. Thus, the likelihood of the candidate attribute is defined and its value (*Li_Attri*) is given by equations (1) and (2). Here, *attri*, $Attri\_Val_{(attri)}$ and $MaxScore_{(attri)}$ indicate an attribute, its scores assigned in step 1 and the max score for each attribute[3], respectively.

[3] In the score sheet of Table 1, the max score for each attribute is 5.

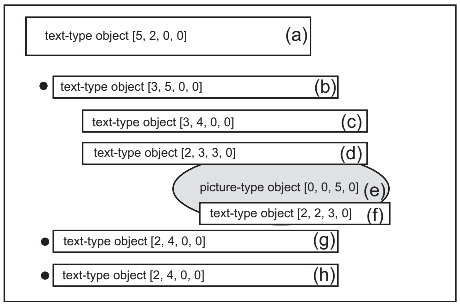

text-type object [5, 2, 0, 0] (a)
text-type object [3, 5, 0, 0] (b)
text-type object [3, 4, 0, 0] (c)
text-type object [2, 3, 3, 0] (d)
picture-type object [0, 0, 5, 0] (e)
text-type object [2, 2, 3, 0] (f)
text-type object [2, 4, 0, 0] (g)
text-type object [2, 4, 0, 0] (h)

Fig. 2. An example of a slide including attributes scores (The numbers within square brackets indicate the attribute scores of title, body text, figure and table.)

$$Ev_{(attri)} = \begin{cases} Attri\_Val_{(attri)} \text{ (if } attri\_candidate \ == \ attri) \\ MaxScore_{(attri)} \ - \ Attri\_Val_{(attri)} \text{ (otherwise)} \end{cases} \quad (1)$$

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

For example, as shown in Fig. 2, the likelihood attribute values of objects (b) and (g) are set by 375 and 300, respectively, so that (b) is more likely to be identified with the body-text attribute than (g). Finally, an object with the largest attribute value is detected and determined with the candidate attribute.

- **(2.2) Changing each attribute score with each object(s) related to the object, which is determined in step 2.1:** The rules are as follows:
  - If the object is identified as a title attribute, each object, including the relationship lists but excluding title, is set by each attribute score, which is then subtracted by 1. Also, the title attributes of all objects are set by 0.
  - If the object is identified as an attribute, excluding title, each object included in the relationships lists of the attribute is set by the attribute score, which is subtracted by 1.
- **(2.3) Repeating steps 2.1 and 2.2 until the attributes of all objects are determined:** An object with a more certain attribute is preferentially assigned the attribute and can affect the determination of an attribute within the functionally related object(s).

(3) Organising the objects into units using close distance and object overlap: After determining the attributes of all objects within the slide, the objects with figure attributes are organised based on the objects' figure relationships list.

(4) Assigning a decoration attribute: After the units with an attribute are detected, a diagram-type object including unit(s) with a body-text attribute and a non-organised arrow-shape type object are reassigned as decoration attributes.

3.2 Structuring the Units Based on a Top-Down Approach

By detecting regularity in a slide's visual structure, the system builds the units as a tree structure. The structuring is based on a top-down regional dividing approach; step by step, the block region including the units is divided into more blocks so that every additional dividing step creates a hierarchical structure by defining parent-child relationships between the units.

The procedures of the structuring are as follows:

(1) Setting initial state: The initial block and a root node are set based on the unit with the title attribute. If a unit with a title attribute is included, the unit is assigned to the root node and the initial block is created to contain the region, including the units below the unit with the title attribute. Otherwise, the root node is created as a blank, and the block is created by the region that includes all the units.

(2) Dividing block(s) vertically: If block(s) have vertical blank space(s), then these block(s) are divided into more blocks according to each space. Otherwise, this step is skipped.

(3) Dividing block(s) horizontally: The step consists of four stages:

In the first stage, large horizontal blank space(s) are sought from block(s). If horizontal blank space(s) larger than the threshold is found in the block(s), then each block is divided into more blocks according to the space. Then, the system proceeds to step 4.

In the second stage, the units' attribute sequences in the block(s) are checked. If one of the sequences in each block is matched with the following heuristic rules, which are based on the attribute relationship between a body text and a figure/table, the block is divided into two blocks according to the matched rule. Then, the system goes to step 4.

(i) If 'an attribute within *TopObj*' == 'body-text' and an object with a figure attribute to the left from *TopObj* is included, then the block is divided into two blocks by the top position of the object with a figure attribute. Only a bullet point list with the *TopObj* is not included in the object with the figure attribute.

(ii) If 'an attribute within *TopObj*' == 'figure/table', then the block is divided into two blocks by the bottom position of the object at the top of the block.

where TopObj presents an object at the top of the block.

In the third stage, the unit at the top position of each block is checked. If the unit is identified as a body-text attribute and is included in a bullet point list, then the block is divided into more blocks by the top position of each of the bullet points and then goes to step 4.

In the final stage, the unit at the top position of each block is checked. If the unit is identified as a body-text attribute, then the block is divided into the unit and other units within the block.

(4) Repeating steps 2 and 3 until every block includes one unit.

4 Experimental Evaluation

4.1 Method and Preparation

The experiment we conducted mainly focused on two points: (1) whether it was effective for the organising process to apply functional relationships among primitive objects within a slide, and (2) whether it was appropriate for the structuring process to apply a top-down approach, which provides attribute information within the units. At present, no structuring methods of slide information and standardised data sets are available for evaluation. Therefore, to evaluate this proposed method, it is necessary to prepare a comparable method.

To compare the proposed method, we used two methods as follows. The first method is set as a standard method that is organised by the objects based only on information derived from the objects' distance relationships. For example, an object with a figure/table type and object(s) overlapping or closely allocated with the object are organised into a unit with a figure/table attribute. The second method is set as a proposal method without the functional relationship ('*Func_rel*'). For instance, an attribute within each object is assigned as an attribute with the highest score in step 2.1 of the organisation, and steps 2.2 and 2.3 are cancelled. In addition, we examined the accuracy of the structuring process using each unit data, which was produced in the organising process. To compare these two methods, we used recall, precision and F-measure. These performance measures are calculated with the following formulas.

$$Recall = \frac{\text{number of detected units matched with correct}}{\text{total number of correct}} \tag{3}$$

$$Precision = \frac{\text{number of detected units matched with correct}}{\text{total number of detected units}} \tag{4}$$

$$F - measure = \frac{2 * Recall * Precision}{Recall + Precision} \tag{5}$$

We created a slide data set for the evaluation. The data set included 30 slides randomly selected from a research paper database[4], which included papers and more than 10,000 slides from the Internet. The average number of pages in slides was 16.4. The structure data of these slides was created manually by manipulating a specially developed system interface. We implemented a system that could automatically generate the structure of information in a slide from the slide file. The system was developed in Microsoft Visual Studio C#, which accepts its input data from a Microsoft PowerPoint (PPT) file and outputs an XML file containing structure information.

Table 2. Accuracy for each attribute results in the organising process

Attribute (Number of its unit)		Title (602)	Body text (2267)	Figure (430)	Table (12)	Decoration (393)
Proposed Method	Recall	0.98	0.98	0.90	1.00	0.90
	Precision	1.00	0.91	0.83	1.00	0.62
	F-measure	0.99	0.94	0.86	1.00	0.73
Proposed Method without using *Func_Rel*	Recall	0.97	0.94	0.87	1.00	0.86
	Precision	0.98	0.92	0.80	1.00	0.63
	F-measure	0.98	0.93	0.84	1.00	0.73
Standard Method	Recall	0.93	0.84	0.57	1.00	0.86
	Precision	0.98	0.90	0.59	1.00	0.63
	F-measure	0.95	0.87	0.58	1.00	0.73

Table 3. Ratio in pages for each correct ratio of results in the structuring process

Ratio of correct link number within a page	1.00	0.99-0.80	0.79-0.60	0.59-0.00	N/A
Proposed Method	0.75	0.05	0.07	0.10	0.04
Proposed Method without using *Func_Rel*	0.62	0.06	0.08	0.20	0.04
Standard Method	0.60	0.04	0.05	0.28	0.04

4.2 Results and Discussion

The results of the organising and structuring processes are summarised in Tables 2 and 3, respectively. Table 2 shows that the proposed method can organise slide information into units with an attribute better than other methods. In particular, this method can also detect a unit as a figure attribute. The organisation of units with a figure attribute is susceptible to the effect of the distance between the objects. Thus, the proposed method providing the functional relationship can eliminate the incorrect placement of slide objects. Table 3 shows that the proposed method is able to identify an attribute more correctly and also that the proposed method can structure the information within a slide better than any other methods. The proposed method used attribute information within units in the structuring process to compensate for the regularity of the visual layout. If units and their attributes were identified more correctly, the proposed method would be able to function more effectively. Thus, attribute information with the units is important for extracting slide structure information; therefore, our approach is useful for extracting information.

We also checked the errors caused by the proposed method in the experiment. One of the problems is that the relationships between the slide's objects are defined by text content, not by the slide's visual layout. This makes it necessary, therefore, to apply a text analysis technique to detect the relationships among the objects.

5 Conclusion and Future Work

In this paper, we proposed a technique, involving organizing and structuring processes, to extract structure information from the information within a slide. The organising process used functional relationships between the objects, and not only the information derived from the close distances between the objects, to eliminate potentially inappropriate organising. In the structuring process, attribute information within the units, as well as the visual cues on the slide, was used to detect how the regularity of the layout structure could be improved.

Although our current system still needs some modifications, our experimental result shows that the proposed method can extract structure information from slide information. In our future work, we are planning to develop slide applications using the structure data extracted by this technique and the slide information processing technique[2] that we have developed.

References

1. Anjewierden, A.: AIDAS: Incremental Logical Structure Discovery in PDF Documents, In Procs. the 6th International Conference on Document Analysis and Recognition, pp.374–378 (2001).
2. Hayama, T., Nanba, H. and Kunifuji, S.: Alignment between a Technical Paper and Presentation Sheets Using a Hidden Markov Model, In Proc. Active Media Technology 2005, pp.102–106 (2005).
3. Ishihara, T., Takagi, H., Itoh, T. and Asakawa, C.: Analyzing Visual Layout for a Non-Visual Presentation-Document Interface, In Proc. the 8th International ACM SIGACCESS Conference on Computers and Accessibility, pp.165-172 (2006).
4. Nanba, H., Abekawa, T., Okumura, M. and Saito, S.: Bilingual presri: Integration of multiple research paper databases, In Proc. the 7th RIAO Conference: Coupling approaches, coupling media and coupling languages for. information retrieval, pp.195-211 (2004).
5. Nanno, T., Saito, S. and Okumura, M.,; Structuring Web Pages Based on Repetition of. Elements, In Proc. the 2nd International Workshop on Web Document Analysis, pp.58–60 (2003).
6. Rosenfeld, B., Feldman, R. and Aumann, Y.: Structural extraction from visual layout of documents, In Procs. the 11th International Conference on Information and Knowledge Management, pp.203–210 (2002).
7. Watanabe, T., Luo, Q. and 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. Yang, Y. and Zhang, H.: HTML Page Analysis Based on Visual Cues, In Procs. the 6th International Conference on Document Analysis and Recognition, pp.859–864, pp.10–13 (2001).
9. Zhai, Y. and Liu, B.: Structured Data Extraction from the Web Based on Partial Tree Alignment, IEEE Transactions on Knowledge and Data Engineering, Vol.18, No.12, pp.1614–1628 (2006).