

Managing Presentation Slides with Reused Elements

Jie Zhang, Chuan Xiao, Sheng Hu, Toyohide Watanabe, and Yoshiharu Ishikawa

Abstract—Slide presentations have become a ubiquitous tool for business and educational purposes. Instead of starting from scratch, slide composers tend to make new presentation slides by reusing materials from existing slides. Understanding how slide elements are copied from one presentation file to another and how presentation files are related to each other are difficult tasks. In this paper, it is investigated the management of multiple presentation files based on reused slide elements. Techniques are developed to detect textual and visual elements that have been reused across multiple presentation files. Interactive visualization methods are proposed to facilitate understanding the process by which these elements are reused and the relationship between the files that use them. A system with a user-friendly interface is designed, based on which experiments are performed to evaluate the effectiveness of the proposed methods.

Index Terms—Presentation slide management, slide element reuse, slide element visualization.

I. INTRODUCTION

Slide presentations are one of the most important tools for today's knowledge workers to present knowledge, exchange information, and discuss ideas. Instead of starting from scratch, slide composers tend to make new slides by reusing existing ones. An online survey shows that more than 97% people compose presentation slides by reusing existing materials [1]. One of the main reasons is to repurpose existing content for different audiences, events, formats, etc. For example, when many researchers and lecturers create new presentation slides, they reuse the lecture notes used in university courses and the reports presented in academic conferences. In business applications, people often create a summary by combining materials used in previous presentations, and modify existing slides in order to present to different audiences.

Detecting reused materials in presentation slides benefits many presentation-related applications; e.g., assisting composer in tracking changes in multiple versions, understanding existing presentation slides, and assembling existing slides to make new ones [2], [3], etc. Although the slide retrieval method for reuse [1] and the method to compare different versions of a presentation file [2] have been

proposed, they are either based on slide-to-slide comparison or file-to-file comparison. In many cases, only an individual element such as a sentence, a table, an image, or a diagram, is copied from one file to another, but overall the slides and the files differ significantly, and thus the reuse element cannot be identified by these methods.

In this paper, the problem of managing presentation slides from the perspective of individual elements is investigated. Different methods are developed to detect textual and visual elements reused in a slide repository specified by users. Textual elements are divided into sentences and further decomposed to bags of words. To detect reused sentences and consider the case when slide composers make minor modifications after reusing elements, similarities are taken into account to tolerate nuances between different versions. Likewise, the bag-of-words model is adopted to find reused visual elements, and utilize similarities to handle the case when they are transformed after reuse. The techniques to tackle the efficiency challenge are also introduced. Then two ways of visualizing reused elements are proposed. The first is to show the files that use the same element in a timeline. The second is to construct a network of presentation files connected by common elements.

Based on the proposed methods, a prototype system is designed with a user-friendly interface, which can be integrated into slide composition tools. When users are browsing a slide, they are notified with what elements in this slide have been reused across different files in the slide repository. Users can quickly find the origin of an element, as well as when and how the element is used by other files. For all the files in the repository, users may also get an overview of their relationships via common elements. They may explore any relationship to get detailed information. To demonstrate the effectiveness of our methods, experimental evaluation on real presentation slide data is conducted using the prototype system.

Our contribution can be summarized as follows: 1) an approach to presentation slide management is proposed by exploiting the notion of reused elements; 2) techniques to detect and visualize reused elements are developed in users' slide repositories; 3) a system integrated with a user-friendly interface is designed to help users browse slides through reused elements; 4) experiments on real data is conducted to evaluate the proposed methods.

The rest of this paper is organized as follows. Section II proposes the conceptual viewpoint and introduces the framework of our approach. Section III proposes the methods to detect reused elements. Section IV introduces the methods to visualize reused elements. Section V reports experiment results and our analyses. Section VI reviews related work. Section VII concludes this paper.

Manuscript received August 9, 2014; revised October 22, 2014. This work was supported in part by the Grant-in-Aid for Scientific Research (#25280039) from JSPS.

Jie Zhang, Sheng Hu, Chuan Xiao, and Yoshiharu Ishikawa are with Nagoya University, Japan (e-mail: jzhang@db.ss.is.nagoya-u.ac.jp, hu@db.ss.is.nagoya-u.ac.jp, chuanx@nagoya-u.jp, ishikawa@is.nagoya-u.ac.jp).

Toyohide Watanabe is with Nagoya Industrial Science Research Institute, Japan (e-mail: watanabe@is.nagoya-u.jp).

II. CONCEPTUAL VIEWPOINT FRAMEWORK AND APPROACHES

Fig. 1 illustrates an overview of the framework of our presentation slide management method. It is composed of two main modules: detection module and visualization module. Next the two modules are introduced respectively.

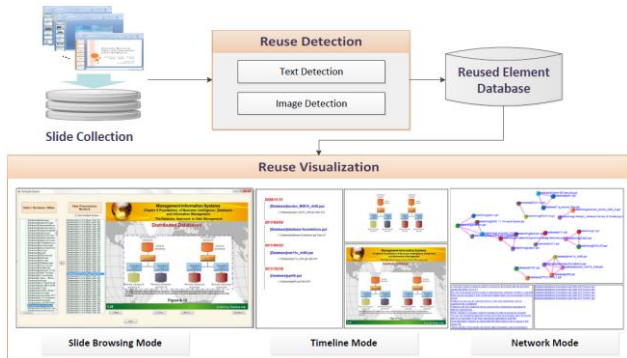


Fig. 1. An overview of slide management framework.

A. Detection Module

Users may specify the location where the repositories of slides are stored in their computers. A database of presentation files is then built by offline and stored on disk to serve future browsing. Textual and visual elements are extracted from the database slides. Textual elements include main text (including titles) and tables. Visual elements include images, charts, and diagrams. Reused elements are detected in this module and the slides in which these elements appear are marked. The detailed reused element detection method will be presented in Section III.

B. Visualization Module

This module is further divided into two submodules: timeline submodule and network submodule. In the timeline submodule, reused elements are notified to users when they are browsing a slide in the repository. Users may click an element, and all the presentation files that use the element will be shown in a timeline. Users may follow the links in the timeline to browse these slides. In the network submodule, users are shown with a presentation network, i.e., a graph connecting presentation files in the repository through reused elements. Users may quickly get an overview of the relationships amongst the presentation files. E.g., the multiple versions of a presentation file form a clique because any two of them share large numbers of common elements. A summary will be shown as a hub because it is assembled by copying elements from multiple files. The details of the two submodules will be introduced in Section IV. In addition, the users are also provided with the interface to browse slides, as they have seen in prevalent presentation composition tools such as Microsoft PowerPoint and Apple Keynote.

A prototype system is designed according to the above framework. The user interface of the system will be shown in Section V-A.

III. DETECTING REUSED ELEMENTS

In this section, the detection methods are introduced for textual element reuse and visual element reuse, respectively.

A. Detecting Reused Textual Elements

The method to detect reused element in main text of presentation slides is first introduced, and then the case of tables is discussed.

When the text in a slide is reused, composers may copy one or more sentences from one slide to another, but overall the texts in both slides differ significantly. For this reason, reuse textual elements is chosen to detect on sentence level; i.e., divide the text in each slide into sentences and then identify the sentences that have been used by multiple presentation files. In addition, considering that composers may make modifications to the reused sentence (e.g., change the order of words, insert additional words into it, and delete a few words from it), each sentence is tokenized into a bag of words with white space and punctuations, and then adopt the idea of similarity search to find reused sentences in the presence of modifications. The Jaccard coefficient is used to capture the similarity between two sentences:

$$\text{sim}(x, y) = \frac{|x \cap y|}{|x \cup y|} \quad (1)$$

where x and y are two sentences represented in bag of words, and $|x|$ denotes the cardinality of a bag x .

Example 1: Considering two sentences: "the telephone was invented by Alexander Bell in 1876" and "in 1876, Alexander Graham Bell invented the telephone". After tokenization, the two sentences become {1876, Alexander, Bell, by, in, invented, telephone, the, was} and {1876, Alexander, Bell, Graham, in, invented, telephone, the}. The similarity between them is $7/10 = 0.7$.

The pairs of sentences whose similarity values by Equation 1 are no smaller than a threshold t are retrieved, and a *sentence reuse graph* is constructed as follows: 1) Each vertex denotes a sentence in the database. 2) Two vertices are connected by an edge if the similarity between the two sentences is no smaller than t . The connected components of this graph can be computed using either a breadth-first search or a depth-first search. Since the Jaccard coefficient is a metric, sentences in the same connected component bear high similarity to each other. Thus the sentences in the same connected component are called a *reused sentence group*, and they are regarded as originate from the same sentence.

Example 2: Fig. 2 shows an example of five sentences depicted in a graph, each vertex (ellipse) denoting a sentence. Assuming $t = 0.5$, the pairs of sentences that satisfy the similarity constraint are connected, and show the similarity values next to the edges. Since there are two connected components, two groups of reused sentences are obtained from this graph.

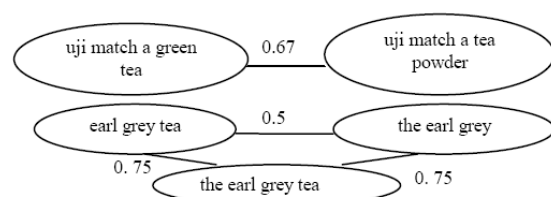


Fig. 2. Example of reused sentences.

A key issue of reused textual element detection is how to

find the pairs of sentences that satisfy the constraint. A straightforward method is to compute the similarity value for every pair of sentences. If Equation 1 is computed by hashing the words in two bags, its time complexity is $O(W)$, where W is the number of words in a sentence. Let S denote the number of sentences in the database. The time complexity of comparing all pairs of sentences is $O(S^2W)$. It is too expensive for practical use because the value of S can be large; e.g., there are 35932 sentences in the 200 presentation files used in our experiment. Since the problem is exactly the set similarity join problem [4] and has been studied by the database research community, the *ppjoin* algorithm is employed [5] to efficiently find the pairs of sentences that satisfy the constraint. Its basic idea is to sort the words in each bag according to a global order and exploit the threshold t . If a pair of sentences satisfy the similarity constraint, they must share at least one word in their first p words, where $p = \lfloor \max(l_x, l_y) (1-t) \rfloor + 1$, and l_x and l_y denote the numbers of words in x and y , respectively [5]. Thus we only need to compute Equation 1 for the pairs that meet this condition, which can be found by indexing the first p words of each sentence. E.g., consider two sentences tokenized into bags of words and sorted in alphabetical order: $\{ a, b, d, e, f \}$ and $\{ c, d, e, f \}$, and a t of 0.5. Then the two bags must share at least one word in the first $\lfloor \max(5, 4) \cdot (1 - 0.5) \rfloor + 1 = 3$ words. The similarity value between the two bags is $4/6 = 0.67$, and indeed they share a common word d in their respective first three words.

For the case of tables, they are processed separately from other text in the database, and for each table the contents in all its cells are concatenated as a sentence. Then reused tables can be identified using the above method.

B. Detecting Reused Visual Elements

The method to detect reused images in presentation slides is introduced, and then the cases of charts and diagrams are discussed.

Like textual element detection, visual element detection also needs to take modification into consideration. Although composers do not often modify images with graphics editing software when copying images from one slide to another, they may transform images (e.g., by scaling and rotating) with presentation composition tools, and this will make the images bit-wise different from the original version. To address this issue, the bag-of-words model [6], a prevalent approach in computer vision, is employed to find reused images. The bag-of-words model represents images as bags of elementary image patches called visual words, as shown in Fig. 3 [7]. A dictionary of visual words called visual vocabulary is created first, and then an image can be described using the words that occur in it. To build a vocabulary of visual words, interest regions in the images are detected with Hessian-affine detector [8], which provides good performance [9] and is widely used in visual word-based studies because of its insensitiveness to affine transformations such as scaling, reflection, rotation, etc. These regions are described in 128-dimension SIFT descriptors and then clustered by a hierarchical k-means algorithm [10], each cluster representing a visual word. Then each image is represented in a bag of visual words.

Like detecting reused sentences, the Jaccard coefficient

(Equation 1) is used to measure the similarity between two bags of visual words. This similarity measure has been adopted for near-duplicate image detection [11], based on the intuition that similar images share most of their visual words.

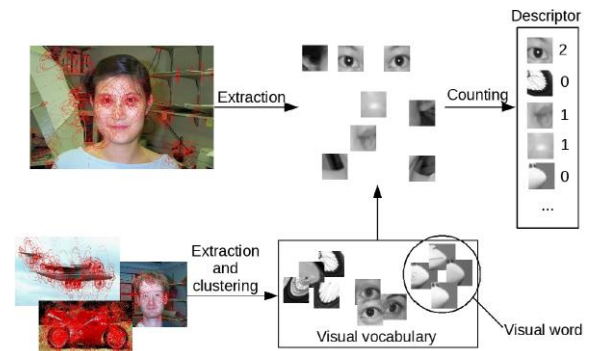


Fig. 3. Bag-of-words model for image retrieval [7].

Similar to sentence reuse graph, an image reuse graph is constructed as follows: 1) Each vertex denotes an image in the database. 2) Two vertices are connected by an edge if the similarity between the images represented in bags of visual words is no smaller than a threshold t .

The image in the same connected component are called a *reused image group*, and they are regarded as originate from the same image.

The efficiency issue also exists for images. Therefore the *ppjoin* algorithm is also used to efficiently find image pairs that satisfy the similarity constraint. The only difference from sentence reuse detection is that the algorithm is run on visual words instead of textual words.

For other types of visual elements, charts are converted to images and processed in the same way. For diagrams, since they consist of individual shapes such as rectangles, circles, and arrows, the topmost, leftmost, rightmost, and bottommost shapes are found in each slide, and convert the screenshot within this area into an image. Then the method to detect reused images can be applied.

Short sentences should be removed because they tend to provide a large number of false positives but almost no meaningful results for reuse detection. Small-size images should also be removed because they are usually simple graphics such as a single-color patch or a logo but not meaningful resources for reuse. To strike a balance between precision and recall, perform reuse detection are performed on sentences containing at least 5 words and images no smaller than 1KB.

IV. VISUALIZING REUSED ELEMENTS

In this section, the method to visualize the reused elements is presented.

A. Timeline of Reused Elements

In order to help users understand how an element (a reused sentence group or a reused image group) is reused, for each reused element, in a timeline the following information are shown: 1) the content of the element, 2) when it is used, and 3) in which files and slides it is used.

The timeline of an image is taken as an example, and its interface is shown in Fig. 4. The content of this image is

displayed on the top right corner. On the left side, the slides that contain this image are listed and grouped by presentation files. The files are sorted by increasing order of last saved time, i.e., the time when the composition of presentation slides is finished. This information can be retrieved from file properties and is shown on the left side as well. Users may click any slide listed on the left, and the content of the slide will be shown on the bottom right corner. The timeline of a reused sentence group is presented in the same way. The only difference is that the sentences in a group may differ due to composers' modifications. Thus the version in the origin file is only shown, i.e., the file with the smallest last saved time, on the top right corner.

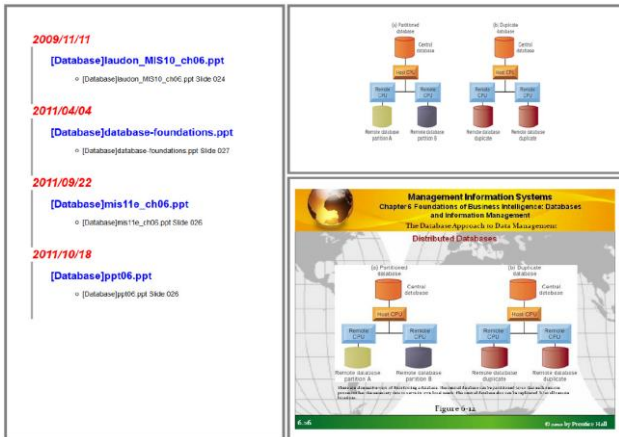


Fig. 4. User interface of timeline.

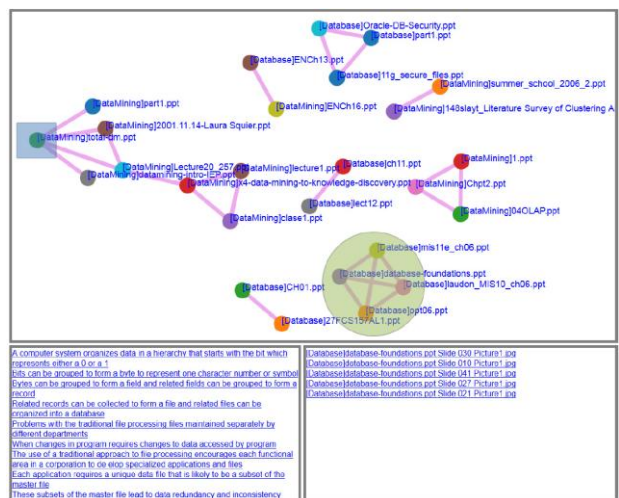


Fig. 5. User interface of presentation network.

B. Presentation Network Based on Reused Elements

Another method of visualization is to show an overview of the relationship between presentation files in terms of reused elements. To this end, a presentation network is constructed as follows: 1) each vertex denotes a presentation file; 2) two vertices are connected by an edge if the two files contain sentences (images) in the same reused sentence (image) group.

Fig. 5 shows the interface of a presentation network. The network is displayed on the top, consisting of 26 files. The file names are given next to the vertices. Users may click any edge in the network, and the common elements between the two files will be shown on the bottom, with sentences on the left and images on the right. Then users may click a common

element to view its timeline. As can be seen from the figure, the vertices in the circled area constitute a clique. They are multiple versions of the same file. In addition, the vertex in the rectangle area is a hub connected to many files. Its contents indicate it is a summary.

V. EXPERIMENTS

A presentation slide management system is designed integrated with the methods proposed in previous sections. In this section, the user interface of the system is introduced and then the results and analyses of our experiment conducted on the system are reported.

A. Prototype System

Our prototype system is implemented in C#, except that the visualization of the presentation network was implemented in HTML with Data-Driven Documents JavaScript Library (D3.js) [12]. The user interface is shown in Fig. 6.

TABLE I: DATASET STATISTICS

Attribute	Number
Files	200
Slides	10327
Sentences (≥ 5 words)	35932
Images (≥ 1 KB)	2282
Average number of words in a sentence	10.1
Average number of visual words in an image	480.6

For the first-time use of this system, the user needs to click the "Select Database Slides" button on the top left corner, and choose the folders that contain the database slide files. Then the database slides will be scanned to detect reused elements. On the left side there are two columns. The left column lists file names in the database. Users may click any file, and the slide numbers in this file will be listed in the right column. The main window is on the right side, and it can switch between three modes: slide, timeline, and network. Users may click any slide in the slide number column, and its content will be shown in the main window. The "Prev" and "Next" buttons help users browse slides, and the main window will show the previous slide and the next slide when they are clicked, respectively.

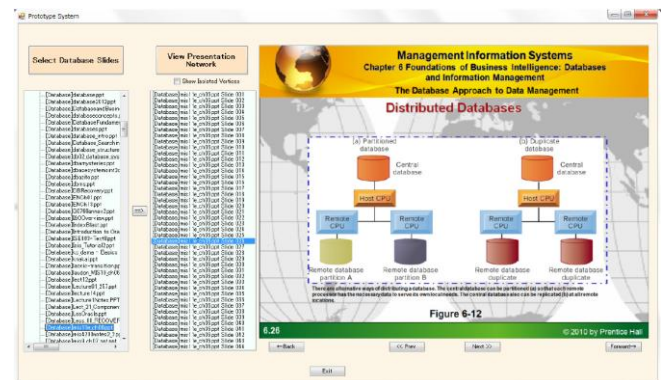


Fig. 6. User interface of prototype system.

As shown in Fig. 6, reused elements are highlighted in the main window when users are browsing slide contents. It will be switched to the timeline mode (Fig. 4) when users click a reused element. Users may follow the links in the timeline and the main window can switch back to slide mode.

To view the presentation network, users need to click the “View Presentation Network” button on the top left. Then the main window will be switched to the network mode (Fig. 5). By default, vertices with no edge connected in the network are hidden. These vertices will appear if the check box “Show Isolated Vertices” on the top left corner is ticked. Users may click vertices in the network to view slides, or edges to view common elements. They may also navigate through different views using the “Back” and “Forward” buttons in the main window, like browsing Web pages.

B. Experiment Setup

Our dataset consists of lecture notes of database courses and data mining courses in universities in USA. Table I provides the statistics about the dataset. The experiments are run on a PC with a 3.40 GHz CPU and 8 GB of RAM.

C. Evaluating Reuse Detection

Coverage of Reuse Relationship: the percentage of files having reuse relationship is first tested and the results are shown in Table II. It can be observed that most files in the dataset have reused textual elements or their textual elements have been reused by other files. A small percentage of files have reused visual elements or visual elements have been reused by other files, and most of these files have reused/been reused textual elements as well. The percentage of the files that do not have any reuse relationship is very small. The results demonstrate the reuse relationship in the dataset.

Example Detection Results: some example reuse detection results are shown.

The example result of reused textual element detection is shown in Fig. 7. The slide contents are displayed on the top, while the context information – file names, slide numbers, and last saved times – is given on the bottom. It can be seen that the two paragraphs on the left slide are copied to the right slide and slightly modified.

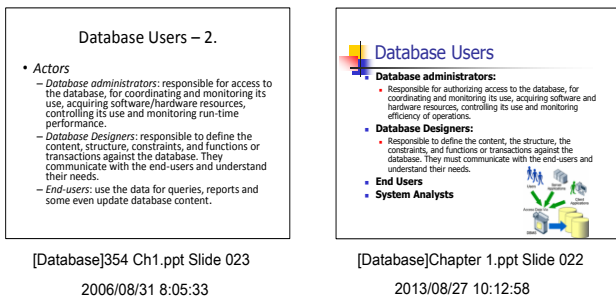


Fig. 7. Example result of text reuse detection.

TABLE II: COVERAGE OF RELATIONSHIP

Type	Percentage
Textual	90.5%
Visual	17.0%
Textual and Visual	15.0%
None	7.5%

The example result of reused visual element detection is shown in Fig. 8. It can be seen that an image is copied from left to right and then scaled. Another difference is that the slide on the original version contains more text on the bottom.

In this example, the text similarity is 36%, image similarity is 100%, and attribute similarity is 11%. The overall similarity is only 49%, and hence cannot be identified by the

method in [1]. The mean square error of screenshots is 6835, the edit distance between the texts in the two slides is 61, and the slide IDs are different. Thus the method in [2] cannot detect the reuse in this pair either.

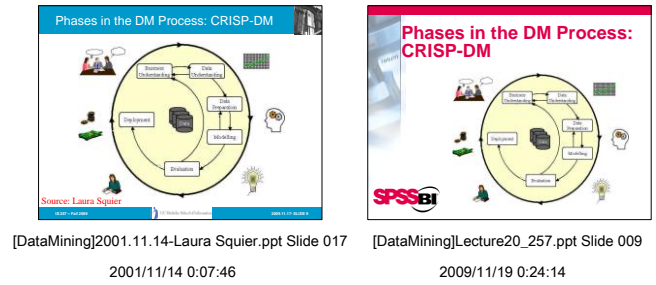


Fig. 8. Example result of image reuse detection.

Quality of Reuse Detection Results: the quality of reuse detection is studied by varying the similarity threshold t . The precision – the percentage of reused results amid the retrieved ones, and the recall – the percentage of retrieved results amid the reused ones are measured, formally defined by the following equations, where R_t denotes the set of true reused sentence/image groups in the dataset, and R'_t denotes the set of reused sentence/image groups identified by our method.

$$Precision = \frac{|R_t \cap R'_t|}{|R'_t|}, Recall = \frac{|R_t \cap R'_t|}{|R_t|}$$

For reused textual element detection, the threshold t is varied from 0.5 to 0.9. Fig. 9(a) shows the precision and recall. The precision increases with t and reaches 100% when t is 0.9. The recall decreases with t and drops to only 2% when t is 0.9. The reason is that when t increases, the similarity constraint becomes stricter, and thus fewer pairs of sentences satisfy the constraint. False positives are reduced, and this results in the increase of precision. On the other hand, this causes that the method misses true results, and consequently decreases the recall. The overall best quality is achieved when $t = 0.6$.

Fig. 9(b) shows the F1 score of reused textual element detection with varying thresholds. The general trend is that the F1 score decreases when the threshold is rising. This is because the recall drops with increasing t and it changes more rapidly than the precision. Since our method achieves best F1 when $t = 0.6$, we set t as 0.6 for the default setting of reused textual element detection.

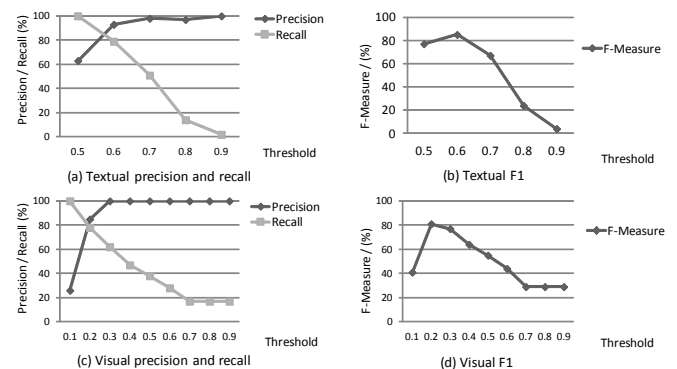


Fig. 9. Experiment results of reuse detection.

For reused visual element detection, the threshold t is varied from 0.1 to 0.9, and plot the precision and recall in Fig.

9(c), with exact duplicates removed. Similar trends can be observed as seen in the evaluation of textual element detection. The difference is that the precision in visual element detection changes more significantly. E.g., both trends are as high as 100%, but as low as 82% and 26%, respectively. This is because when t is low, the bag-of-words model retrieves similar images such as apples in different colors, but obviously they have no reuse relationship. The overall best quality is achieved when $t = 0.2$.

Fig. 9(d) shows the F1 score of reused visual element detection with varying thresholds. The general trend is that it first increases with t , peaks when $t = 0.2$, and drops as t keeps increasing. Therefore 0.2 is set as the default setting of the threshold of reused visual element detection.

D. Evaluating Reuse Visualization

In order to evaluate the reuse visualization method as well as the overall efficacy of our system, an experiment was conducted by recruiting volunteers who deal with presentation slides on a daily basis and performing a trial of our system on the presentation files made by the volunteers, followed by a questionnaire and an interview.

Table III summarizes the users' opinions on our system. The scores are given on a 5-point Likert scale. As for the overall opinions, four out of five users gave 5 points to our system. The users were impressed with the design and the implementation of the system, commenting that this system would be helpful to the users who intend to track reused elements among quite a few slides.

For the element timeline, the users considered the representations intuitive and easy to understand. One of them liked the apparent and impressive images shown by the system. For the presentation network, the users found the interface engaging and were impressed by the innovation of this idea. One user commented that the network shows the correlation between slides clearly and information can be briefly obtained from the network.

The users also assessed the usability of the system. They reported that the system helps understand the slides and it shows changes in slides intuitively through the element timeline. When being asked about the usefulness for presentation slide composition, they used words such as "somehow useful", "relatively high", "extremely useful", etc.

After the trial of the system, users were eager to use it in the future. They also pointed out the limitations of the system, which be discussed later in this section.

E. Error Analysis

TABLE III: OVERAGE OF RELATIONSHIP

Metric	Score
Overall rating	4.8
Identification of reused elements from slides	4.8
Interpretability of element timeline	4.8
Interpretability of presentation network	4.6
Usefulness for understanding presentation slides	4.6
Usefulness for tracking changes in slides	4.6
Usefulness for presentation slide composition	4.4
Interest in future use	4.6

For textual element reuse detection, an example of false positive is shown in Fig. 10. The sentences detected are "Sales

volume as a function of product, month, and region" and "Sales Volume as a function of time, city and product". The similarity between them is 0.692 but from the slides they are not reused sentences.

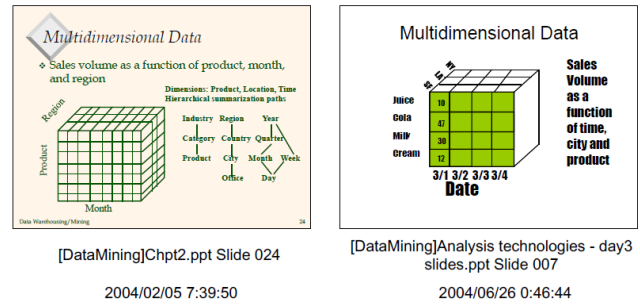


Fig. 10. False positive of reused textual element detection.

The false positive of visual element reuse detection is shown in Fig. 11. Both slides contain an image on the bottom the similarity between the visual words of the two images is 0.278. They are similar but not reused images. Both errors are due to the decreased precision of the bag-of-words model under low thresholds. A possible remedy to the above errors is to refine the results identified by similarity search with the more sophisticated machine learning techniques.

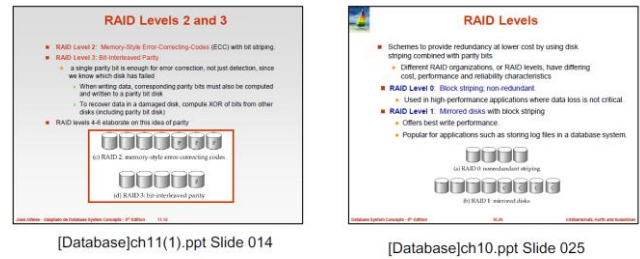


Fig. 11. False positive of reused visual element detection.

VI. RELATED WORK

Prevalent presentation composition tools such as Microsoft PowerPoint and Open Office Impress mainly focus on providing tools for creating and presenting slides, but they do not provide any way of seeing an overview of the differences between multiple versions. For this reason, a presentation slide management system was developed to visually compare between different versions of presentation files [2]. The system compares pixel-level image differences between slides and differences between the texts on each slide. It also provides an interactive visualization tool for users to examine differences between presentations. The difference between our work and this study is that our method focuses on exhibiting how individual elements are used in different slides, while their work focuses on presenting users differences between slides.

The notion of presentation slide reuse was first proposed in [3]. An online survey was conducted to study how often users start composing presentation slides from existing ones and what types of materials are often reused. A system was developed based on the survey [1]. Users can select a slide as the query and the system recommends relevant slides stored on users' machines. However, this method can only process whole slide queries but cannot deal with individual elements

such as a sentence or an image in a slide.

Our reused element detection method is related to the problem of presentation slide retrieval. Unlike our sentence level retrieval, many approaches focus on processing keyword queries. UPRISE [13] is a search engine developed to handle keyword queries based on the notion of impression of keywords in slides. To find images for a textual query, a system called SLIDIR [14] was developed using machine learning techniques. In [15], text recognition techniques were employed to support image and video search using keywords. An XML based system was developed [16] by extracting textual features to compute a fuzzy relevance score for each database slide. In [17], a slide retrieval and browsing method was proposed based on mining relationships between slides and generating snippets. Besides keyword retrieval, retrieving graphical elements has also been studied. E.g., the indexing and retrieval method in which slides are captured as images was proposed in [18]. The problem of processing diagram queries was also investigated [19].

Another body of work focuses on presentation slide composition. Outline Wizard [14] is a presentation composition method on the basis of outline matching. Topic clustering [20] and hierarchical organization [21] were also employed to develop composition methods. There are also a few literatures on generating slides from academic papers [22], discourse structures [23], or textbook chapters [24].

VII. CONCLUSION AND FUTURE WORK

In this paper, we proposed an approach to managing presentation slides by exploiting reused slide elements. We developed different techniques to find textual and visual elements reused in database slides. We devised interactive visualization tools to help users understand how these elements are reused and how the presentation files are related to each other. On the basis of the proposed techniques, a prototype system with a user-friendly interface is designed. Experiments were conducted on top of the system and demonstrated the effectiveness of the proposed methods.

Our future work is to explore the composition methods by reusing existing materials. Users may input keywords or use examples to describe what kind of materials they want. Then our method retrieves relevant elements from the database and automatically generates presentation slides.

ACKNOWLEDGMENT

This research was partly supported by the Grant-in-Aid for Scientific Research (#25280039) from JSPS.

REFERENCES

- [1] M. Sharmin, L. Bergman, J. Lu, and R. B. Konuru, "On slide-based contextual cues for presentation reuse," in *Proc. International Conference on Intelligent User Interfaces*, pp. 129–138, 2012.
- [2] S. M. Drucker, G. Petschnigg, and M. Agrawala, "Comparing and managing multiple versions of slide presentations," in *Proc. ACM Symposium on User Interface Software and Technology*, pp. 47–56, 2006.
- [3] Y. Mejova, K. D. Schepper, L. Bergman, and J. Lu, "Reuse in the wild: an empirical and ethnographic study of organizational content reuse," in *Proc. ACM CHI Conference on Human Factors in Computing Systems*, pp. 2877–2886, 2011.
- [4] S. Chaudhuri, V. Ganti, and R. Kaushik, "A primitive operator for similarity joins in data cleaning," in *Proc. International Conference on Data Engineering*, p. 5, 2006.
- [5] C. Xiao, W. Wang, X. Lin, J. X. Yu, and G. Wang, "Efficient similarity joins for near-duplicate detection," *ACM Transactions on Database Systems*, vol. 36, no. 3, pp. 15, 2011.
- [6] J. Sivic and A. Zisserman, "Video google: A text retrieval approach to object matching in videos," in *Proc. IEEE International Conference on Computer Vision*, pp. 1470–1477, 2003.
- [7] P. Tirilly, V. Claveau, and P. Gros, "Distances and weighting schemes for bag of visual words image retrieval," *Multimedia Information Retrieval*, pp. 323–332, 2010.
- [8] K. Mikolajczyk and C. Schmid, "An affine invariant interest point detector," in *Proc. European Conference on Computer Vision*, pp. 128–142, 2002.
- [9] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. J. V. Gool, "A comparison of affine region detectors," *International Journal of Computer Vision*, vol. 65, no. 1–2, pp. 43–72, 2005.
- [10] D. Nistér and H. Stewénius, "Scalable recognition with a vocabulary tree," in *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 2161–2168, 2006.
- [11] O. Chum, J. Philbin, and A. Zisserman, "Near duplicate image detection: min-hash and tf-idf weighting," in *Proc. British Machine Vision Conference*, pp. 1–10, 2008.
- [12] Data-driven documents. [Online]. Available: <http://www.d3js.org/>
- [13] H. Yokota, T. Kobayashi, T. Muraki, and S. Naoi, "Uprise: Unified presentation slide retrieval by impression search engine," *IEICE Transactions*, vol. 87-D, no. 2, pp. 397–406, 2004.
- [14] L. Bergman, J. Lu, R. B. Konuru, J. MacNaught, and D. L. Yeh, "Outline wizard: presentation composition and search," in *Proc. International Conference on Intelligent User Interfaces*, pp. 209–218, 2010.
- [15] C.-Y. Chen, "An integrated system supporting effective indexing, browsing and retrieval of microsoft powerpoint presentation database," *ICDE Workshops*, p. 16, 2006.
- [16] A. Kushki, M. Ajmal, and K. N. Plataniotis, "Hierarchical fuzzy detection similarity combination for presentation slide retrieval," *EURASIP Journal on Advances in Signal Processing*, 2008.
- [17] Y. Wang and K. Sumiya, "A browsing method for presentation slides based on semantic relations and document structure for e-learning," *Journal of Information Processing*, vol. 20, no. 1, pp. 11–25, 2012.
- [18] A. Vinciarelli and J.-M. Odobez, "Application of information retrieval technologies to presentation slides," *IEEE Transactions on Multimedia*, vol. 8, no. 5, pp. 981–995, 2006.
- [19] S. Tanaka, T. Tezuka, A. Aoyama, F. Kimura, and A. Maeda, "Slide retrieval technique using features of figures," in *Proc. International Multi-Conference of Engineers and Computer Scientists*, vol. 1, 2013.
- [20] R. P. Spicer, Y.-R. Lin, A. Kelliher, and H. Sundaram, "Nextslideplease: Authoring and delivering agile multimedia presentations," *ACM Transactions on Multimedia Computing, Communications, and Applications*, vol. 8, no. 4, pp. 53, 2012.
- [21] B. B. Bederson and J. D. Hollan, "Pad++: A zooming graphical interface for exploring alternate interface physics," in *Proc. ACM Symposium on User Interface Software and Technology*, pp. 17–26, 1994.
- [22] M. Sravanthi, C. R. Chowdary, and P. S. Kumar, "Slidesgen: Automatic generation of presentation slides for a technical paper using summarization," presented at Florida Artificial Intelligence Research Society Conference, 2009.
- [23] K. Hanaue, Y. Ishiguro, and T. Watanabe, "Composition method of presentation slides using diagrammatic representation of discourse structure," *J. Knowledge and Web Intelligence*, vol. 3, no. 3, pp. 237–255, 2012.
- [24] Y. Wang and K. Sumiya, "A method for generating presentation slides based on expression styles using document structure," *I. J. Knowledge and Web Intelligence*, vol. 4, no. 1, pp. 93–112, 2013.



Jie Zhang was born in Henan, China in 1984. She is a Ph.D. candidate in Graduate School of Information Science, Nagoya University. She received B.E. degree from Xi'an University of post and telecommunications, China in 2006, and M.E. degree from Xi'an Jiaotong University, China in 2009. Her research interests include e-learning and text mining. She is a student member of IPSJ.



Chuan Xiao was born in Liaoning, China in 1983. He is an assistant professor in Graduate School of Information Science, Nagoya University. He received B.E. degree from Northeastern University, China in 2005, and Ph.D. degree from The University of New South Wales in 2010. His research interests include data cleaning, data integration, textual databases, and graph databases. He is a member of DBSJ.



Sheng Hu was born in Jiangshu, China in 1991. He is a master student in Graduate School of Information Science, Nagoya University. He received B.E. degree from North China Electric Power University in 2013. His main research interest is graph data management.



Toyohide Watanabe was born in Kyoto, Japan in 1949. He is a senior researcher in Nagoya Industrial Science Research Institute. He was a professor in Graduate School of Information Science, Nagoya University. He received B.S., M.S., and Ph.D. degrees from Kyoto University in 1972, 1974, and 1983, respectively. His research interests include knowledge / data engineering, computer-supported collaborative

learning, document understanding, parallel and distributed process interaction, and drawing interpretation, etc. He is a member of AAAI, AACE, ACM, ICEJ, IEEE, IPSJ, JSAI, JSISE, and JSSS.



Yoshiharu Ishikawa was born in Fukuoka, Japan in 1965. He is a professor in Graduate School of Information Science, Nagoya University. He received B.S., M.E., and Dr. Eng. degrees from University of Tsukuba in 1989, 1991, and 1995, respectively. His research interests include spatio-temporal databases, mobile databases, sensor databases, data mining, information retrieval, and web information systems. He is a member of ACM, DBSJ, IEEE, IEICE, IPSJ, and JSAI.