Detecting Reused Elements in Presentation Slides

Jie Zhang, Chuan Xiao, 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 browsing existing slides and reusing materials from them. In this paper, we investigate the problem of reused element detection in presentation slides. We develop respective techniques to identify both textual and visual elements that have been reused across multiple presentation files. Experiments are performed to evaluate the effectiveness of the proposed methods.**

***Index Terms*—Slide element reuse, presentation slide management, slide browsing.**

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. A common approach to create new presentation slides is to browse a collection of older versions and assemble new slides by copying appropriate materials from them. Detecting reused materials in presentation slides benefits many presentation-related applications; e.g., assisting composers in tracking changes in multiple versions, understanding existing presentation slides, and assembling existing slides to make new ones [2], [3], etc. Although the method to detect reused slides [1] and the method to compare different versions of a presentation file [2] have been proposed, they are either based on slide-to-slide 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 reused element cannot be identified by these methods.

In this paper, we investigate the problem of detecting reused materials in presentation slides from the perspective of individual elements. We develop different methods to detect both 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 that slide composers make minor modifications after reusing elements, similarities are taken into account to tolerate nuances between different versions. Likewise, we adopt the bag-of-words model [4] to find reused visual elements such as images, charts, and diagrams, and utilize similarities to handle the case that visual elements are transformed after being reused. The techniques to tackle the efficiency challenge are also introduced. The experimental evaluation on real presentation slide data shows that 90.5% presentation files have reuse relationship via textual elements and 17.0% files have reuse relationship via visual elements. The effectiveness of our methods on detecting reused elements is also demonstrated through experiments.

The rest of this paper is organized as follows. Sections II gives the overview of the framework and Section III proposes the methods to detect reused textual and visual elements. Section IV reports experiment results. Section V reviews related work. Section VI concludes this paper.

II. Framework Overview

A. Framework

Fig. 1 shows the overview the framework of our reused element detection method. We extract textual elements and visual elements from the database slides specified by users. Textual elements include main text (including titles) and tables. Visual elements include images, charts, and diagrams. Reused elements are then detected and the slides in which these elements appear are marked.

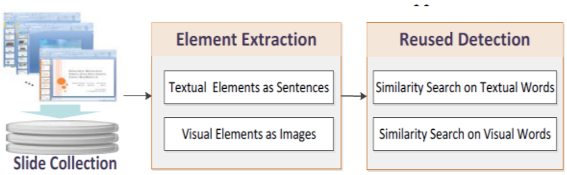

Slide Collection
Element Extraction
Textual Elements as Sentences
Visual Elements as Images
Reused Detection
Similarity Search on Textual Words
Similarity Search on Visual Words

Fig. 1. An overview of reused element detection framework.

III. Approach

A. Detecting Reused Textual Elements

We first introduce the method to detect reused element in main text of presentation slides, and then discuss the case of tables.

When the text in a slide is reused, composers may copy one or more sentences from one slide to another, but overall the

Manuscript received July 10, 2014; revised May 29, 2015.

Jie Zhang, Chuan Xiao, and Yoshiharu Ishikawa are with Graduate School of Information Science, Nagoya University, Nagoya, Japan (e-mail: jiezhang198467@gmail.com, chuanx@nagoya-u.ac.jp, ishikawa@is.nagoya-u.ac.jp).

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

DOI: 10.7763/IJKE.2015.V1.27

texts in both slides differ significantly. For this reason, we choose to detect reused textual elements 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 and delete a few words), we tokenize each sentence 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:

$$sim(x,y)=\frac{x\cap y}{x\cup y}, \qquad (1)$$

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.

We retrieve the pairs of sentences whose similarity values by Eq.1 are no smaller than a threshold t, and construct a sentence reuse graph 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 we call the sentences in the same connected component 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, we connect the pairs of sentences that satisfy the similarity constraint, 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.

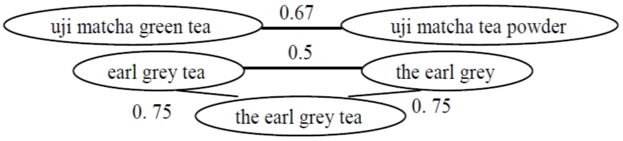


Fig. 2. Example of reused sentences.

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 we compute Eq.1 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 [5] and has been studied by the database research community, we employ the ppjoin algorithm [6], a state-of-the-art method to this problem, to efficiently find the pairs of sentences that satisfy the constraint.

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

B. Detecting Reused Visual Elements

We introduce the method to detect reused images in presentation slides, and then discuss the cases of charts and diagrams.

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 [4], 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. 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, we detect interest regions in the images with Hessian-affine detector [7], which provides good performance [8] 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 [9], each cluster representing a visual word. Then each image is represented in a bag of visual words.

Like detecting reused sentences, the Jaccard coefficient (Eq. (1)) is used to measure the similarity between two bags of visual words. This similarity measure has been adopted for near-duplicate image detection [10], based on the intuition that similar images share most of their visual words.

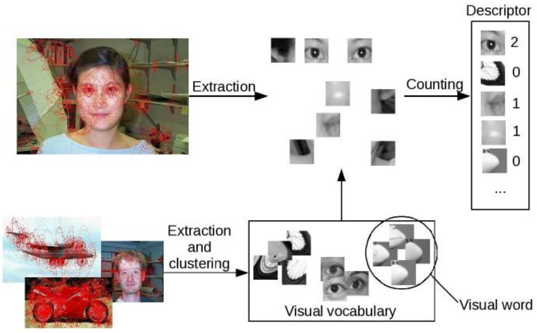


Fig. 3. Bag-of-words model for image retrieval [11].

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. We call the image in the same connected component a *reused image group* as originate from the same image.

The efficiency issue also exists for images. Therefore we also use the ppjoin algorithm 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, we find the topmost, leftmost, rightmost, and bottommost shapes in each slide, and convert the screenshot within this area into an image. Then the above reused image detection method can be applied.

We need to remove short sentences because they 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, we perform reuse detection on sentences containing at least 5 words and images whose sizes are no smaller than 1KB.

IV. Experiments

In this section, we report the experiment results and our analyses.

A. 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 8GB of RAM.

TABLE I: Dataset Statistics

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

The percentage of files having reuse relationship is shown in Table II. It can be observed that most files have reused textual elements and some reused visual elements.

TABLE II: Coverage of Relationship

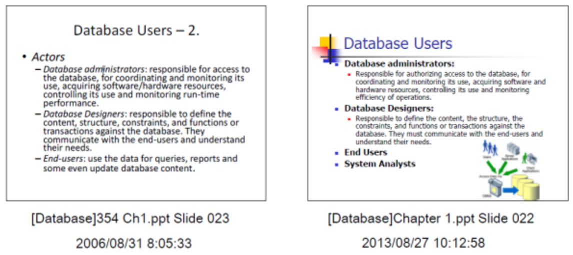
Type	Percentage
Textual	90.5%
Visual	17.0%
Textual and Visual	15.0%
None	7.5%

B. Example Detection Results

We show some example reuse detection results.

The example result of reused textual element detection is shown in Fig. 4. 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. The two paragraphs on the left slide are copied to the right slide and slightly modified.

The example result of reused visual element detection is shown in Fig. 5. An image is copied from left to right and then scaled. The difference between the two slides is that the original version contains more text on the bottom.



Database Users – 2.

- *Actors*
  - *Database administrators*: responsible for access to the database, for coordinating and monitoring its use, acquiring software/hardware resources, controlling its use and monitoring run-time performance.
  - *Database Designers*: responsible to define the content, structure, constraints, and functions or transactions against the database. They communicate with the end-users and understand their needs.
  - *End-users*: use the data for queries, reports and some even update database content.

[Database]354 Ch1.ppt Slide 023

2006/08/31 8:05:33

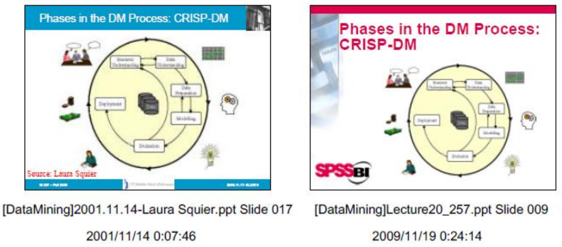
Database Users

- **Database administrators:**
  - Responsible for authorizing access to the database, for coordinating and monitoring its use, acquiring software and hardware resources, controlling its use and monitoring efficiency of operations.
- **Database Designers:**
  - Responsible to define the content, the structure, the constraints, and functions or transactions against the database. They must communicate with the end-users and understand their needs.
- **End Users**
- **System Analysts**

[Database]Chapter 1.ppt Slide 022

2013/08/27 10:12:58

Fig. 4. Example result of reused textual element detection.



Phases in the DM Process: CRISP-DM

Source: Laura Squier

[DataMining]2001.11.14-Laura Squier.ppt Slide 017

2001/11/14 0:07:46

Phases in the DM Process: CRISP-DM

[DataMining]Lecture20_257.ppt Slide 009

2009/11/19 0:24:14

Fig. 5. Example result of reused visual element detection.

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.

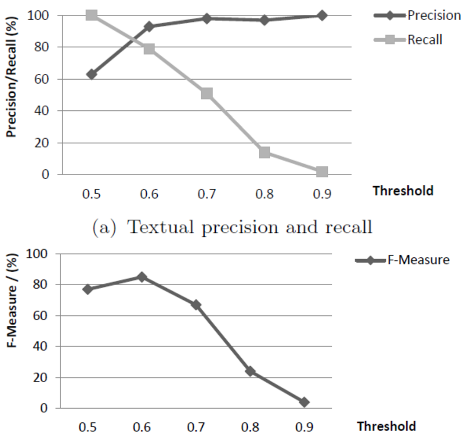
C. Effectiveness of Reuse Detection

We study the effectiveness of reuse detection by varying the similarity threshold *t*. We measure the precision – the percentage of reused results amid the retrieved ones, and the recall – the percentage of retrieved results amid the reused ones, formally defined by the following equations, where R_l 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.

$$\mathrm{Pr}ecision = \frac{|R_l|\cap|R_t|}{|R_t|}, \mathrm{Re}call = \frac{|R_l|\cap|R_t|}{|R_l|}$$

For reused textual element detection, we vary the threshold *t* from 0.5 to 0.9. Fig. 6(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. 6(b) shows the F_1 score of reused textual element detection with varying thresholds. The general trend is that the F_1 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 F_1 when *t* = 0:6, we set t as 0.6 for the default setting of reused textual element detection.



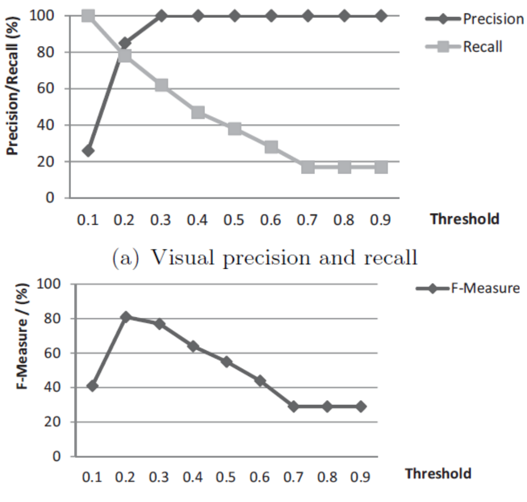
(a) Textual precision and recall

(b) Textual F_1

Fig. 6. Experiment results of textual reuse detection.

For reused visual element detection, we vary the threshold *t* from 0.1 to 0.9, and plot the precision and recall in Fig. 7(a). Similar trends can be observed as we have 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. 7(b) 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.



(a) Visual precision and recall

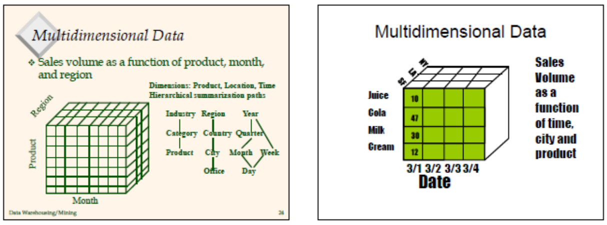
(b) Visual F_1

Fig. 7. Experiment results of visual reuse detection.

D. Error Analysis

For textual element reuse detection, an example of false positive is shown in Fig. 8. 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.

The false positive of visual element reuse detection is shown in Fig. 9. 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.


Multidimensional Data

- Sales volume as a function of product, month, and region

Dimensions: Product, Location, Time
Hierarchical summarization paths

Multidimensional Data

Sales Volume as a function of time, city and product

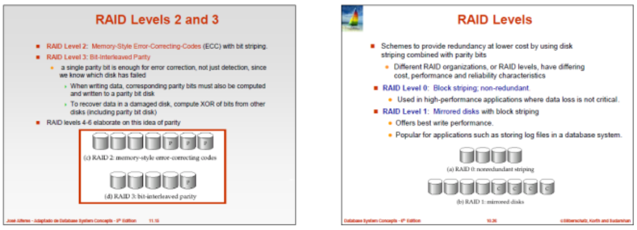
[DataMining]Chpt2.ppt Slide 024

2004/02/05 7:39:50

[DataMining]Analysis technologies - day3 slides.ppt Slide 007

2004/06/26 0:46:44

Fig. 8. False positive of reused textual element detection.


RAID Levels 2 and 3

- RAID Level 2: Memory-Style Error-Correcting-Codes (ECC) with bit striping.
- RAID Level 3: Bit-Interleaved Parity
  - a single parity bit is enough for error correction, not just detection, since we know which disk has failed
    - When writing data, corresponding parity bits must also be computed and written to a parity bit disk
    - To recover data in a damaged disk, compute XOR of bits from other disks (including parity bit disk)
- RAID levels 4-6 elaborate on this idea of parity

P
P
P
(c) RAID 2: memory-style error-correcting codes
P
(d) RAID 3: bit-interleaved parity
José Alferes - Adaptado de Database System Concepts - 5th Edition
11.15

RAID Levels

- Schemes to provide redundancy at lower cost by using disk striping combined with parity bits
  - Different RAID organizations, or RAID levels, have differing cost, performance and reliability characteristics
- RAID Level 0: Block striping; non-redundant.
  - Used in high-performance applications where data loss is not critical.
- RAID Level 1: Mirrored disks with block striping
  - Offers best write performance.
  - Popular for applications such as storing log files in a database system.

(a) RAID 0: nonredundant striping
C
C
C
C
(b) RAID 1: mirrored disks
Database System Concepts - 6th Edition
10.26
©Silberschatz, Korth and Sudarshan

[Database]ch11(1).ppt Slide 014

2007/03/07 9:18:07

[Database]ch10.ppt Slide 025

2012/03/15 0:32:19

Fig. 9. False positive of reused visual element detection.

V. 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 [12] 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 [13] was developed using machine learning techniques. In [14], text recognition techniques were employed to support image and video search using keywords. An XML based system was developed [15] by extracting textual features to compute a fuzzy relevance score for each database slide. In [16], 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 [17]. The problem of processing diagram queries was also investigated [18].

Another body of work focuses on presentation slide composition. Outline Wizard [13] is a presentation composition method on the basis of outline matching. Topic clustering [19] and hierarchical organization [20] were also employed to develop composition methods. There are also a few literatures on generating slides from academic papers [21], discourse structures [22], or textbook chapters [23].

VI. 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, and Nature Science of SZU (#201437).

References

[1] M. Sharmin, Bergman, J. Lu, and R. B. Konuru, "On slide-based contextual cues for presentation reuse," in *Proc. International Conference on Intelligent User Interfaces*, 2012, pp. 129–138.

[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*, 2006, pp. 47–56.

[3] Y. Mejova, K. 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*, 2011, pp. 2877–2886.

[4] J. Sivic and A. Zisserman, "Video google: A text retrieval approach to object matching in videos," in *Proc. IEEE International Conference on Computer Vision*, 2003, pp. 1470–1477.

[5] S. Chaudhuri, V. Ganti, and R. Kaushik, "Aprimitive operator for similarity joins in data cleaning," in *Proc. International Conference on Data Engineering*, 2006, p. 5.

[6] C. Xiao, W. Wang, X.-M. Lin, J. X. Yu, and G.-R. Wang, "Efficient similarity joins for near-duplicate detection," *ACM Transactions on Database Systems*, vol. 36, no. 3, p. 15, 2011.

[7] K. Mikolajczyk and C. Schmid, "An affine invariant interest point detector," in *Proc. European Conference on Computer Vision*, 2002, pp. 128–142.

[8] 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.

[9] 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.

[10] 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.

[11] 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.

[12] H. Yokota, T. Kobayashi, T. Muraki, and S. Naoi, "Uprise: Unified presentation slide retrieval by impression search engine," *IEICE Transactions*, vol. 87-D(2), pp. 397–406, 2004.

[13] L.e Bergman, J. Lu, R. B. Konuru, J. Naught, and D. L. Yeh, "Outline wizard: Presentation composition and search," in *Proc. International Conference on Intelligent User Interfaces*, pp. 209–218, 2010.

[14] C.-Y. Chen, "An integrated system supporting effective indexing, browsing and retrieval of microsoft powerpoint presentation database," *ICDE Workshops*, pp. 16, 2006.

[15] A. Kushki, M. Ajmal, and K. N. Plataniotis, "Hierarchical fuzzy feature similarity combination for presentation slide retrieval," *Journal on Advances in Signal Processing*, 2008.

[16] Y.-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.

[17] 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.

[18] 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.

[19] 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, p. 53, 2012.

[20] 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.

[21] M. Sravanthi, C. R. Chowdary, and P. S. Kumar, "Slidesgen: Automatic generation of presentation slides for a technical paper using summarization," in *Proc. Florida Artificial Intelligence Research Society Conference*, 2009.

[22] K. Hanaue, Y. Ishiguro, and T. Watanabe, "Composition method of presentation slides using diagrammatic representation of discourse structure," *I. J. Knowledge and Web Intelligence*, vol. 3, no. 3, pp. 237–255, 2012.

[23] 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 is a post-doctoral researcher in College of Computer Science and Software Engineering, Shenzhen University. She reveived the B.E. degree from Xi'an University of Post and Telecommunications, China in 2006, the M.E. degree from Xi'an Jiaotong University, China in 2009, and the Ph.D. degree in Nagoya University, Japan in 2015. Her research interests include data mining, textual and visual data processing.

Chuan Xiao is an assistant professor in Graduate School of Information Science, Nagoya University. He received the B.E. degree from Northeasten University, China in 2005, and the Ph.D. degree from the Universtiy 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.

Toyohide Watanabe is a senior researcher in Nagoya Industrial Science Research Institute. He was a professor in Graduate School of Information Science, Nagoya University. He received the B.S. M.S. and Ph.D. degrees from Kyoto University, Japan in 1972, 1974, and 1983, respectively. His research interests include knowledge, data engineering, computer-supported collaborative learning, document understanding, etc. He is a member of AAAI, AACE, ACM, ICEJ, IEEE, IPSJ, JSAI, JSISE, and JSSS.

Yoshiharu Ishikawa is a professor in Graduate School of Information Science, Nagoya University. He received the B.S., M.E., and Dr. Eng. degrees from University of Tsukuba, Japan in 1989, 1991, and 1995, respectively. His research interests include spatio-temporal databases, information retrieval, and Web information systems. He is a member of ACM, DBSJ, IEEE, IEICE, IPSJ, and JSA.