

Generating Summarized Preview for Education Resource based on Exploring and Comparing GUIs

Chao Ma

School of Data and
Computer Science,
Sun Yat-sen University,
Guangzhou, China
Research Institute of Sun
Yat-sen University in
Shenzhen, Shenzhen, China
qingyuanluofeng@163.com

Xiangping Chen*

Institute of Advanced
Technology,
Sun Yat-sen University,
Guangzhou, China
Research Institute of Sun
Yat-sen University in
Shenzhen, Shenzhen, China
chenxp8@mail.sysu.edu.cn

Yongsheng Rao

School of Data and
Computer Science,
Sun Yat-sen University,
School of Computer Science
and Educational Software,
Guangzhou University,
Guangzhou, China
rysheng@163.com

Mouguang Lin

Institute of Advanced
Technology,
National Engineering
Research Center of Digital
Life,
Sun Yat-sen University,
Guangzhou, China
garnettlam@126.com

Abstract—Compared to traditional education resource, Interactive Education Resource (IER) includes dynamic content to interact with users in order to enhance comprehension. The contents of IERs are relatively complex because of its dynamic feature. Exploring IERs and understanding its content during resource search become time-consuming. In this paper, we propose a preview generation approach based on automated software testing and image processing. Our approach first analyzes IERs to collect interactive elements and the event list. Based on the interaction related information, our approach simulates user actions and record snapshots step by step. Our approach compares snapshots to generate an image including the basic content, interaction region and dynamic contents. The image is used as the overall preview and the animation of snapshots collected during simulation is used as detail preview of the IER. In the experiment, we evaluate the completeness, correctness, and comprehension of the preview. The experiment result shows that the preview can be used to provide most information in the resource correctly and easy to understand. Our approach can save time for users in understanding the IERs.

Keywords- *preview generation; interactive education resource; dynamic content collage*

I. INTRODUCTION

E-learning entails the use of electronic technologies in learning and teaching. With the increasing popularity of e-learning, techniques and tools for developing interactive education resources are developed to enhance inspiration, such as Beam[1] and 3D tangible model[2]. Because this kind of education resources can enhance user interaction to improve understanding of knowledge, a wide variety of interactive educational resources are developed.

With the increasing number of interactive educational resources, searching resources become time consuming. Firstly, an interactive educational resource usually depends on a specific tool. Some of the tools for browsing interactive educational resources do not have an online version. Users are required to download the resource and its browsing tool, install the tool to open the resource, explore the resource to find out whether the resource is valuable. Secondly, the cost to understand the interactive content is relatively high. Text

information in most interactive resources is easy to understand because it usually does not involve long text. However, locating interactive elements and triggering user action are time-consuming for most readers. For the users who are unfamiliar with the way to interact with this kind of education resource, the learning cost increases.

Current research on resource preview provides approaches to save time in understanding resource with complex content, such as web page[3,4,7] and interaction[5,6]. It can avoid downloading and installing unnecessary tool and resources and save time in understanding content. Image[7], text[8], video[9,10], audio[11,12] are used to provide summarized preview and proven to be useful in practice.

However, there is no existing preview generation approaches for interactive educational resource. Considering the importance of interaction, understanding a resource includes understanding how to interact with this resource and knowing what dynamic content will appear during interaction.

In this paper, we propose an approach for generating summarized preview for interactive education resource. The summarized preview is an image which collages the basic content, interaction region and dynamic contents of the resource. In order to collect content of the resource, an interactive education resource opened by its exploring tool is viewed as a knowledge specific application. Our approach generates event list by analyzing interactive elements in the resource. Based on the event set, our approach traverses the resource and collect snapshots by triggering operations related to interactive elements. Our approach uses Heatmap algorithm to render interaction region. And then, dynamic contents are extracted by comparing snapshots and clustering changes in different snapshots. The first snapshot rendered in the interaction region and image collaged with dynamic content are used as the overall preview. The animation of snapshots collected during simulation is used as detail preview of the IER.

In case study, we randomly select 20 interactive education resource of the type Super Sketchpad[21] to valid our approach. 15 Students from Sun Yat-sen University are invited to evaluate preview effect according to its completeness,

correctness, and comprehension. The average scores of the preview completeness, correctness and comprehension ratings are 4.42, 4.53 and 4.28 (max is 5). The experiment result shows that the preview is valuable for users. The summarized preview can save 54.89% time for users on average. Users can understand 69.04% of content using the preview provided by our approach.

The remainder of the paper is organized as follows: Section 2 gives a general overview of our solution, which is further detailed in the Section 3. Section 4 presents the evaluation of our approach; Section 5 discusses some related works before Section 6 concludes our work.

II. APPROACH OVERVIEW

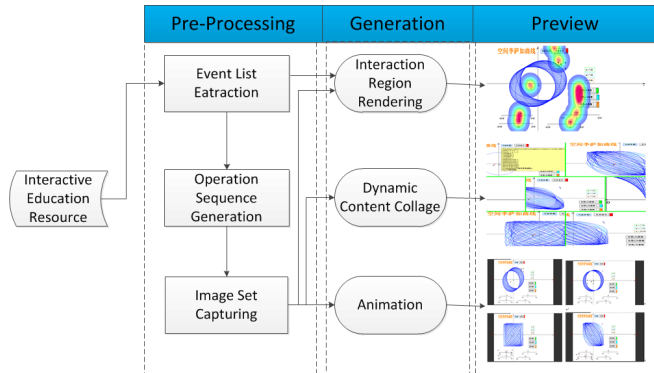


Figure 1. Approach Overview

Figure 1 shows the overview of our approach for generating summarized preview of IERs. In order to collect content of the resource, an interactive education resource opened by its exploring tool is viewed as a knowledge specific application. Inspired by GUI centric testing techniques, our approach collects event list related to the interactive element. And then, operations are extracted according to event types and operated in the GUI. Snapshots are recorded during the simulation.

The summarized preview is an image which collages the basic content, interaction region and dynamic contents of the resource. We take the first snapshot collected in the pre-processing stage as the basic content of the resource. In rendering interaction region, Heatmap algorithm is used with Gaussian model for color weight computation. And then, dynamic contents are extracted by comparing snapshots and clustering changes in different snapshots. The first snapshot rendered in the interaction region and image collaged with dynamic content are used as the overall preview. Considering that some details of the resource are missing in the summarized preview, the animation of snapshots collected during simulation is used as supplement.

III. GENERATING SUMMARIZED PREVIEW

A. Pre-processing

Interactive education resource are developed to facilitate understanding of knowledge during learning activities. The interaction between these resource and users are relatively simpler compared with the interaction types supported in applications.

- Education resource may include one or more pages. When there are more than one page, pages are usually arranged in sequence.
- The interaction types are decided by the resource definition language and exploration tool. The number of event types are usually less than the number of event types supported by programming languages.

Based on these two assumptions, we model the interaction between interactive resource and users. An interactive resource is $IR=(P, R)$, P is a set of pages, R is the sequence relationship between pages. A page can be abstracted as $P=(E, DR)$. E is a set of elements in the page, and DR is the dependency relationship between two elements. $DR=\{ \langle e_i, e_j \rangle | e_i \in E \wedge e_j \in E \}$. An element is abstracted as $e=\langle id, eType, data, actionType, scope, reactionNum \rangle$. id is the identification of an element in a page. $data$ is the data associated with the element. $eType$ is used to define the interactive and visible feature of elements. There are three types: invisible, visible&interactive, visible&static. $actionType$ is the type of action to input through computer peripherals, such as keyboard input, mouse click etc. $scope$ of an action defines the positions in the interface in which the action can be executed. $reactionNum$ is the number of reaction according to user action.

In the pre-processing stage, the information for describing an interactive resource is extracted based on the API provided by resource exploring tools. In this stage, the implementation is highly related to kind of resource. Two kinds of information are required: the events related to interactive elements and the relationships between elements. For the former information, the extraction is relatively simpler because information of interactive elements are usually directly recorded in the resource file. The implementation of extraction can be done by analyzing source file according to the file format.

The extracting relationships between interactive elements are more complex because (1) relationships are usually not recorded explicitly. (2) the classification of relation types may differ in different resource exploring tools. The mismatch requires implementing adapters in practice. It is worth noticing that our approach can generate preview even when there is a lack of relationship information. Dependency relationships are used to enhance the quality of operation sequence.

The operation sequence S is a set of *event* which are triggered in the user interface. *event* is defined as $event = \langle eid, actionType, position \rangle$. An *event* is an execution of action on an interactive element. We generate operation sequence S which triggers events of all the interactive elements at least once and meet the constraints of element dependency. Based on the operation sequence, an execution script can be generated. In our implementation, we generate script AutoHotkey, which is a free, open-source macro-creation and automation software for Windows that allows users to automate repetitive tasks[13]. Based on the execution script, AutoHotkey executes the operation sequence according the *actionType*, *position* and *reactionNum*. When an operation is executed, the snapshot is recorded and saved.

Considering animation, storyboard can be used to show teaching example visually which can strengthen the comprehension of computing theory[14], we choose animation as one of the preview methods for IER. Animation can show the change of IER when each operation is executed. We write an automated playing program to play the image set of snapshots setting internal time as 1 second. Figure 2 shows the animation preview of an IER which is made by the tool Super Sketchpad and used to introduce the math knowledge about Lissajous-Figur. This example is also used in the following section.

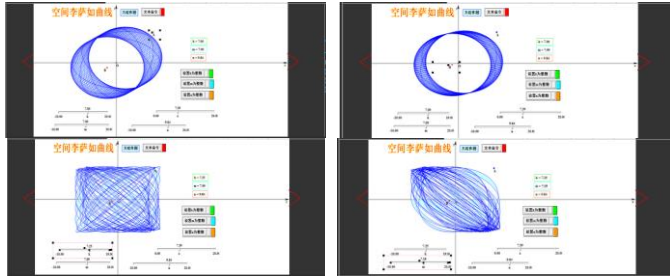


Figure 2. Animation Preview During 4 Seconds

B. Preview Generation

The generation of preview includes rendering interaction region in the basic content and collaging dynamic contents.

1) Interaction Region Rendering

Rendering interaction region is used to indicate users how to interact with the resource and avoid false interaction. Firstly, whether an element is interactive or not is not decided by its visual appearance. Interaction region can be used to distinguish interactive elements. Secondly, small size interactive elements will not be missed with explicitly rendering.

Heatmap[15] can render objects with cold color and heat color to offer high attention region for users. It is suitable to highlight interaction region in IER. We render interaction region based on Blignaut's heatmaps algorithm. The set of event *Events* and first snapshot in the set of snapshots collected in the pre-processing stage are used as input. For each event in the event set, the algorithm uses its position as the center point and create a circle with pre-defined radius as interaction region. For each pixel in the interaction region, its color weight is calculated based on Gaussian model. And then, the algorithm computes new RGB for each pixel according to the relation between RGB component composite linear model and weight. The color of points in the interaction region are changed considering both its original color and new RGB.

Algorithm: $\text{RenderInteractionRegion}(Events, P_0, R, W)$

Input: *Events* is event set, P_0 is the first snapshot, R is radius of circle for interaction region, W is weight of color.

Output: P_0 , an image rendered interaction region.

Start

initialize each value of *weightMatrix* as 0

for(*event* in *Events*) {

$centerPoint = event.position$

$Points = \{p | distance(p, centerPoint) \leq R\}$

$weightMatrix[centerPoint] = W$

for(each *p* in *Points*) {

$d = distance(p, centerPoint)$

$coe = e^{-d^2 / (2 \times (0.34 \times R)^2)}$

$weightMatrix[p] = weightMatrix[p] + W * coe$ }

for(each *point* in P_0) {

$w = weightMatrix[point]$

$R = (0 \leq w \leq 50 ? 0 : (50 < w \leq 75 ? 51 * w / 5 - 510 : 255))$

$G = (0 \leq w \leq 25 ? 51 * w / 5 : (25 < w \leq 75 ? 255 : -51 * w / 5 + 1020))$

$B = (0 \leq w \leq 25 ? 255 : (25 < w \leq 50 ? -51 * w / 5 + 510 : 0))$

$temp.R = R, temp.G = G, temp.B = B$

$point.RGB = (point.RGB + temp.RGB) * 0.5$

return P_0

Figure 3 shows the basic content rendered in the interaction region of the IER example using this algorithm. We set parameter R as 82 and W as 100.

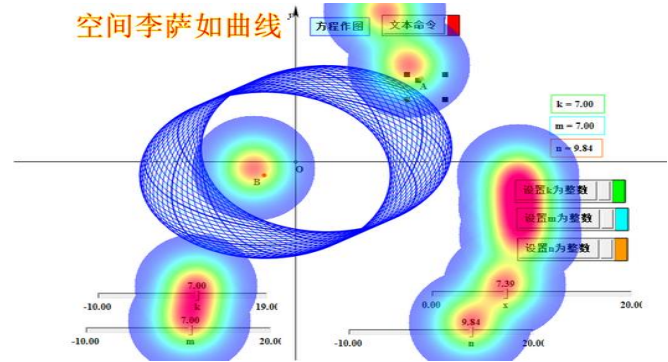


Figure 3. Interaction Region Rendering

2) Dynamic Content Collage

When users interact with IER, the dynamic contents will appear. These dynamic contents can be found out by comparing snapshots. The number of dynamic contents may increase with the complexity of the IER. The dynamic contents may be distributed in different positions. As a result, if we want to offer an effective preview, these dynamic contents are required to be merged into one image.

Collage is a technique to put many small images into one image. It can be used to merge dynamic content of education resource. The first step is selecting representative image. Different from existing collage approach[16] which takes face as important feature and selects important regions by ROI, our approach collects dynamic contents as important regions. The dynamic contents can be found out by comparing snapshots captured during execution. In most cases, a dynamic content

can be considered more important if its area is larger. As a result, our algorithm try to find out the top k dynamic contents with largest area.

Algorithm: GetDynamicContents($P_0, ChangedSnapshotSet, k$)

Input: P_0 is the first snapshot, $ChangedSnapshotSet$ is the set of snapshot except P_0 , k is the number of candidate areas for collage.

Output: $DynamicContents$: the set of the top k dynamic contents with largest area.

computeDifferentArea($snapshot, P_0$) return a rectangle containing dynamic content between $snapshot$ and P_0 .

Start

```

for(each snapshot in ChangedSnapshotSet){
    area=computeDifferentArea(snapshot, P0)
    if(DynamicContents.size < k){
        DynamicContents=DynamicContents ∪ {area}
    }
    else{
        minArea={a|a ∈ DynamicContents ∧
                b ∈ DynamicContents ∧ a<=b}
        if(area > minArea){
            DynamicContents=DynamicContents ∪ {area}-{minArea}
        }
    }
}
return DynamicContents

```

When collaging the set of dynamic contents, the main problem is that the area of dynamic contents are different. For a dynamic content whose area is smaller than cut area, we append it into result regions directly. Otherwise, we use k-Means algorithm to cluster changed areas and select a dynamic content with larger area as main change region. Because the clustering speed is slow when the area of changed region is large, we separate a change region as a set of lines and use midpoint to represent a line during clustering. The midpoints are recovered to lines in changed region when clustering is finished.

The input of the algorithm is the set of dynamic contents $DynamicContents$ generated using the algorithm SelectDynamicContents. In the segment of dynamic content, the content is viewed as a set of horizontal lines. The algorithm is similar when the content is viewed as a set of vertical lines.

Algorithm: CollageDynamicContent($DynamicContents, cutWidth, cutHeight, clusterNum$)

Input: $DynamicContents$ is a set of dynamic contents, $cutWidth$ and $cutHeight$ are used to set the cut area, $clusterNum$ is number of clusters.

Output: An image merged dynamic content.

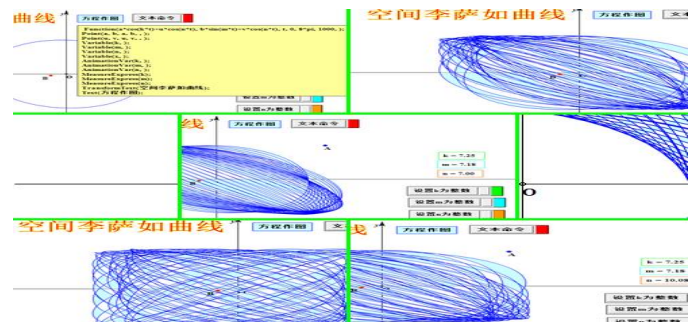
Start

```

for(each content in DynamicContents) {
    if(content.area <= cutWidth × cutHeight)
        Regions.append(content)
    else{
        for(each row in content) {
            lineList= lineList ∪ getLineList(row)
        }
        MidPointSet= {line.midPoint| line ∈ lineList }
        clusterSet = k-Means(MidPointSet, clusterNum)
        maxArea =-1
        for(each cluster in clusterSet){
            lineSet={line| line ∈ lineList ∧ line.midPoint ∈ cluster }
            maxX = max({line.x| line ∈ lineSet })
            minX = min({line.x| line ∈ lineSet })
            maxY = max({line.y| line ∈ lineSet })
            minY = min({line.y| line ∈ lineSet })
            area=(maxX-minX) × (maxY-minY)
            if(area > maxArea){
                maxArea=area
                tempRegion=Rectangle(minX, maxX, MinY, maxY)
            }
        }
        Regions.append(tempRegion)
    }
}
collage Regions into one image as ResultImage
return ResultImage

```

Figure 4 shows the collage of dynamic contents of the example IER “Lissajous-Figur”. We use lines to separate each region. The collage result shows that new elements can be easily found out in the preview. However, there may exist similar contents if a dynamic element moves or the details of an element change according to different operation.



IV. RELATED WORK

Research works on resource preview are developed to save time in interpreting complex contents. Approaches are developed to provide webpage preview, video preview, audio preview and so on. Our work is different that we focus on providing preview for interactive education resources. An interactive education resource opened by its exploring tool is viewed as a knowledge specific application. A lot of techniques such as scaling, pre-fetch and precomputation, collage, key information extraction, resource visualization are developed in the preview generation approaches. Our approach also employs techniques used in these works .

In order to generate preview for web page, Yoo et al[3] use thumbnail to offer link preview. They adjust thumbnails' size and transparency according to the distance between cursor and link. Aula et al [4] compare visual web page preview with textual web page preview on predicting web page's benefit and find that adding title and url around thumbnail can decrease underestimated prejudice. Wittenburg et al[7] regard outer links as potential content user's looking for and organize images of outer links as image streaming to generate image preview.

In order to generate preview for complex interaction, Rekimoto[6] designs "PreSense", which is a touch interactive press key device. Users can slide key and know what will happen if users press this key. Drewes et al [5] use translucent bitmap to enhance tool tips, they override the function "GetMessageString" in the framework under windows platform to implement that cursor moves on window menu widget will show translucent dialog.

In order to generate preview for video, Barletta et al [9] extract video, audio, text key information from video, offer 2~3 minutes main key frames and second key frames under 20 seconds for users. Craggs et al [10] propose a approach to record user's tags for video frame when users are watching videos. A server uses meta-data and tag to return a video animation which made up of key frames. Zhao[17] offers visualization for open education resource video. Their approach extracts key frames from video and uses collage technique to organize important content into one image. Komlodi et al[18] use dynamic and static key frames to offer the preview of video.

Audio preview is also used in the context that users can not look at the message directly. Parente et al [11] develop AEI software to offer preview of audio files for blind users. Their approach finds document hyperlink and extracts content to read for blinders. Shirazi et al [12] extract SMS content, analyze the content to get SMS type and play the tune corresponding SMS type for users.

A lot of techniques are developed in the preview generation approaches. Scaling[3,4,18], pre-fetch and precomputation [5,6,7,11,12], key information extraction[9,10,17] and collage[16,20] are widely used to generate preview. Ahn et al[8] use icon's brightness and darkness degree to analyze the relativity which multiple documents to the same theme. Khan et al [19] use CSG to extract concept of course and use concept node to form course map. Luo et al[20] propose a way to collage important video content in one image.

V. CASE STUDY

A. Case Study Setting

We use our approach to generate preview of interactive education resource for Super Sketchpad, which is a software for teaching mathematics by providing geometry drawing and reasoning during creation of education resource. The preview functionality is implemented in the resource search engine and providing service to students learning courses related to Super Sketchpad in Guangzhou University (in the Website: <http://www.appoperation.com/search/>). The search engine now contains about 1,000 interactive education resources of the type Super Sketchpad. We randomly select 20 interactive education resource which have at least 5 events to valid our approach.

15 students from Sun Yat-sen University are invited to evaluate preview effect. Each participant chooses to evaluate 10-20 preview. They are asked to download the tool of Super Sketchpad and the resources, and then use the tool to explore and interact with the resource manually. At last, they compare the preview with their understanding of the resource. For each resource, participants are asked to evaluate the completeness, correctness, and comprehension ability of the preview and give a score.

In this paper, preview completeness means resource information offered by the preview compared to the entire information in resource; preview correctness indicates the extent users are able to understand content in resource correctly without ambiguity; preview comprehension means comprehension easiness or difficulty users understand content in resource by the preview. For each preview dimension, score for rating can be from 1 to 5. Higher score means better preview effect on preview dimension. In addition, participants are asked to answer whether the preview save time, how much time is saved and how much content can be understood from the preview.

B. Experiment Result

Figure 5 shows the preview rating on three valid dimensions of 15 participants: completeness, correctness, easy comprehension. The vertical axis shows the rating scores. The higher preview rating is, the better the preview effect is. The horizontal axis shows the name of resource used in the evaluation. According the experiment result, the average score for preview completeness is 4.42, the average score for preview correctness is 4.53, the average score for preview comprehension is 4.28. It indicates that the preview include most key contents and better summarized preview effect for IER.

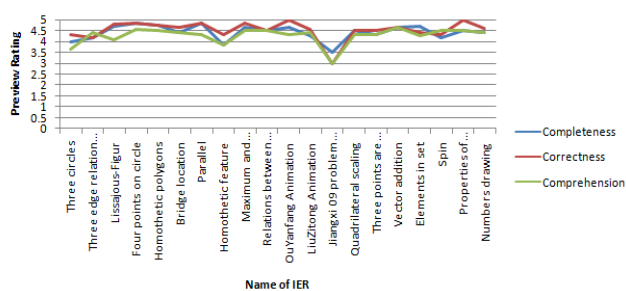


Figure 5. Rating of IER Preview

All participants agree that using the preview can save time during searching resource of Super Sketchpad. Figure 6 shows the percentage of time saved using our preview. According to the experiment result, users can save 54.89% of time on average using the preview. It saves half the time instead of user's complex manual exploration. Users have 69.04% content understanding on average. It means that users can get general content from IER and don't need to download IER.

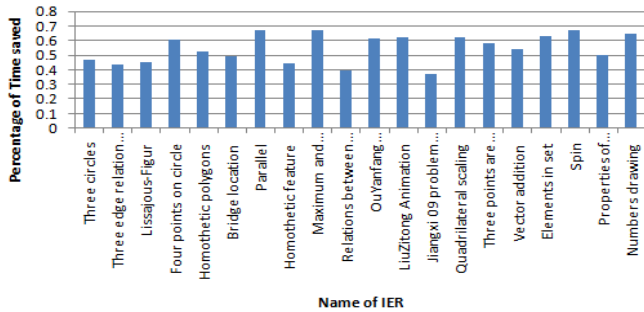


Figure 6. Percentage of Time Saved

VI. CONCLUSIONS AND FUTURE WORK

In this paper, we propose a summarized preview approach based on exploring and comparing GUIs for interactive education resource. The summarized preview is an image included basic content, interaction region and dynamic content. The experiment result shows that the preview can be used to provide most information in the resource correctly and easy to understand.

In this paper, we implement our approach for the IERs developed using Super Sketchpad. In our future work, we will apply our approach in different kinds of IERs to evaluate the extensibility and effect of our approach in generating preview for education resources.

ACKNOWLEDGMENT

This research is supported by the NSFC Guangdong Joint Fund (No. U1201252), the Science and Technology Planning Project of Guangdong Province (No. 2014B010110003), National Natural Science Foundation of China (No. 61502546, 61370186), and National High-tech R&D Program (863 Program) (NO. 2015AA015408).

REFERENCES

- [1] Zeina Atrash Leong, Michael S. Horn. The BEAM: a Digitally Enhanced Balance Beam for Mathematics Education. In IDC '10 Proceedings of the 9th International Conference on Interaction Design and Children, New York: ACM, 2010, 290-292
- [2] Oai Ha, Ning Fang. Development of Interactive 3D Tangible Models as Teaching Aids to Improve Students' Spatial Ability in STEM Education, In Frontiers in Education Conference, IEEE, 2013, 1302-1304
- [3] ByungIn Yoo, JongHo Lea, YeunBae Kim, The seamless browser: enhancing the speed of web browsing by zooming and preview

- thumbnails, WWW '08 Proceedings of the 17th international conference on World Wide Web, New York, ACM, 2008, 1019~1020
- [4] Aula A, Khan R M, Guan Z, et al, A comparison of visual and textual page previews in judging the helpfulness of web pages, In Proc. WWW '10 Proceedings of the 19th international conference on World wide web, New York, ACM, 2010, 51~60
- [5] Heiko Drewes, Albrecht Schmidt, WYSIWYG-Tool Tips: Enhancing Tool Tips with Translucent Preview Bitmaps, Advances in Visual Computing Volume 3804 of the series Lecture Notes in Computer Science, USA, 647~652
- [6] Jun Rekimoto, Takaaki Ishizawa, Carsten Schwesig, et al, PreSense: interaction techniques for finger sensing input devices, UIST '03 Proceedings of the 16th annual ACM symposium on User interface software and technology, New York, ACM, 2003, 203~212
- [7] Kent Wittenburg, Wissam Ali-Ahrnad, Daniel LaLiberte, et al, Rapid-Fire image Previews for information Navigation, In AVI '98 Proceedings of the working conference on Advanced visual interfaces, New York, ACM, 1998, 76~82
- [8] Jae-wook Ahn, Peter Brusilovsky, Guiding Educational Resources for iSchool Students with Topic-based Adaptive Visualization, Proceedings of the 2011 iConference, New York, ACM, 2011, 632~633
- [9] A. Barletta, M. Mayer, B. Moser, Leafing digital content, IUI '04 Proceedings of the 9th international conference on Intelligent user interfaces, New York, ACM, 2004, 217~219
- [10] Barnaby Craggs, Myles Kilgallon Scott, Jason Alexander, ThumbReels: Query-Sensitive Web Video Previews Based on Temporal, Crowdsourced, Semantic Tagging, CHI '14 Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, New York, ACM, 2014, 1217~1220
- [11] Parente P, Audio Enriched Links: Web Page Previews for Blind Users, In Proceedings of the 6th international ACM SIGACCESS conference on Computers and accessibility, New York, ACM, 2003, 2~8
- [12] Alireza Sahami Shirazi, Ari-Heikki Sarjanoja, Florian Alt, et al, Understanding the Impact of Abstracted Audio Preview of SMS, In CHI'10 Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, New York, ACM, 2010, 1735~1738
- [13] AutoHotKey .https://autohotkey.com/
- [14] Daniela Chudá Visualization in education of theoretical computer science, CompSysTech '07 Proceedings of the 2007 international conference on Computer systems and technologies, New York, ACM, 2007, Article No. 84
- [15] Pieter Blihnaut, Visual span and other parameters for the generation of heatmaps, ETRA'10 Proceedings of the 2010 Symposium on Eye-Tracking Research & Applications, New York, ACM, 2010, 125~128
- [16] Rother C, Bordeaux L, Hamadi Y, et al, Autocollage, SIGGRAPH '06 ACM SIGGRAPH 2006 Papers, New York, ACM, 2006, 847~852
- [17] Baoquan Zhao, Songhua Xu, Shujin Lin, et al, A new visual navigation system for exploring biomedical Open Educational Resource (OER) videos, Journal of the American Medical Informatics Association Advance Access, 2015, pii: ocv123. doi: 10.1093/jamia/ocv123
- [18] Anita Komlodi, Gary Marchionini, Key Frame Preview Techniques for Video Browsing, In Proc. DL'98, New York, ACM Press, 1998, 118~125
- [19] Javed I. Khan, Manas S. Hardas, Observing Knowledge Clustering for Educational, In K-CAP '07 Proceedings of the 4th international conference on Knowledge, New York, ACM, 2007, 193~194
- [20] Sheng-Jie Luo, Chun-Yu Tsai, Hsiao-Ching You, et al, Previewing video content with dynamic and interactable collage, SA '12 SIGGRAPH Asia 2012 Posters, New York, ACM, 2012, Article No. 21
- [21] Jingzhong Zhang, Xichdeng Peng. Free Software SSP for Teaching Mathematics. Symbolic Computation and Education. 2007.8. pp:115-135