

Research Article

Picture Preview Generation for Interactive Educational Resources

Jianxiong Wang ¹, Yongsheng Rao ¹, Xiaohong Shi ², and Xiangping Chen ³

¹Institute of Computing Science and Technology, Guangzhou University, Guangzhou 510006, China

²School of Mathematics and Information Science, Guangzhou University, Guangzhou 510006, China

³Guangdong Key Laboratory for Big Data Analysis and Simulation of Public Opinion, School of Communication and Design, Sun Yat-sen University, Guangzhou 510006, China

Correspondence should be addressed to Yongsheng Rao; rysheng@gzhu.edu.cn

Received 8 January 2021; Revised 16 April 2021; Accepted 30 April 2021; Published 13 May 2021

Academic Editor: Wei Wang

Copyright © 2021 Jianxiong Wang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

With the development of network technology, many online educational resource platforms have emerged, and the number of resources on these platforms is increasing dramatically. Compared with the traditional resources, the interactive educational resources (IERS) have hidden interactive information and dynamic content. Only when the users perform interactive operations correctly can they acquire the knowledge conveyed by the resources, which makes it a challenge to help the users understand the general content of resources and find the resources that they are interested in. Therefore, in this paper, we propose a method for generating picture previews of IERS. Firstly, we analyze the file that stores interactive resource information, obtain the interactive elements' information, extract the dependencies, and generate an operation sequence. Secondly, we execute the operation sequence using an automated software tool to mimic the interaction process between the user and the resource and save the screenshots in the process to generate a picture preview of an IER. We apply this method to a dynamic geometry platform, NetPad, and conduct experiments. Experimental results demonstrate that the method can effectively extract interactive semantic information and simulate the interactive process, and the generated picture previews are of great help to the users to understand the content of IERS.

1. Introduction

Recently, there is a consensus among the public that high-quality knowledge should be openly and easily available, and with the development of network technology, the emergence of many online education platforms makes this hope come true. These platforms can foster large-scale co-construction and sharing of educational resources [1]. However, with the rapid increase of resource categories and quantities, it has become an immense challenge to efficiently browse and understand the general content of resources for finding resources.

The resource preview, as a promising solution to this problem, has become an area of intense investigation in recent years. Existing methods have been able to preview the important content of traditional resources, such as the previews of images [2–6], documents [7–12], and video resources [13–21]. But for interactive educational

resources (IERS), there is still a lack of previews of dynamic content and interactive processes in the resources.

Understanding of any computing environment requires familiarity with its underlying technologies [22]; educational resources are the foundation of online education platforms, which include traditional educational resources and IERS; different from the former, IERS contain not only common elements such as texts, graphics, audios, and videos but also interactive elements with more intricate semantic information. Generally, these elements include buttons, sliders, and draggable objects. Interactive elements can increase learners' interest in learning, reduce cognitive loads, and enhance their understanding of knowledge. Specifically, users can interact with these elements through several interactive operations so that resources can dynamically and visually present the corresponding knowledge and the process of knowledge formation.

To obtain a good preview of IERs, the complex process of acquiring the general content of IERs should be well considered. Firstly, since IERs usually require specialized subject software tools or webpages to browse, the users need to download the resources and corresponding tools, open the resources on the local computers for browsing, and then determine whether the resources are needed. Thus, this process is time-consuming. Secondly, after opening the resource, the users need to interact with interactive elements manually. However, IERs have hidden dynamic content and interactive processes, which are difficult to extract from these elements. In specific, there are certain operating rules among interactive elements, that is, different operating sequences of interactive elements will produce different knowledge presentation effects. Only by operating the interactive elements in the appropriate order can the hidden interactive information be triggered, the dynamic content can be displayed, and the knowledge conveyed by the resource can be acquired. On the one hand, this will bring interactive doubts and obstacles to the users in understanding the resource content and affect the acquisition and the use of resources. This kind of learning-related operations will greatly increase the time and brain costs, especially for those users who are unfamiliar with the operations. On the other hand, it also brings a great challenge to the extraction of the dynamic content of resources and increases the difficulty of resource previews.

This paper proposes a method for generating picture previews of IERs. In specific, this method first analyzes the file that stores resource information, obtains interactive elements' information, extracts dependencies, generates an operation sequence, then uses an automated software tool to perform the operation sequence to mimic the user's interaction with the resource, and saves the screenshots in the process to generate picture previews of IERs.

The novelties and main contributions of this paper are reflected as follows.

- (1) It proposes a method for generating picture previews for IERs; this method can accurately extract the information of interactive elements, analyze the appropriate interaction sequence, and then mimic the interactive process between a human and the resource automatically.
- (2) Extensive experiments are conducted on a dataset of dynamic geometric interactive resources, which verify that the method can represent the general content of IERs and enable time-saved browsing for users. Compared with manual interaction, the proposed method saves users 87.3% of the time on average and helps users understand 88.6% of the content of resources.
- (3) Picture previews can be applied to other related work of interactive resources, such as automatic classification of resources and retrieval of resources by pictures [23].

The rest of this paper is organized as follows. The second section will review the previous work related to resource previews. The third section of this paper will elaborate on details of the proposed method, including how to extract dependencies of interactive elements and generate an operation sequence, how to use automated software tools to achieve interaction with resources, and how to obtain pictures during the interaction. The fourth section will apply the proposed method to the dynamic geometry platform, Netpad. The fifth section will introduce a series of experiments. Finally, the final section will summarize the paper.

2. Related Work

The resource preview aims to help users to understand the general content of resources without downloading files, saving time for users effectively. The predecessors have already done some work in traditional resource previews, for example, the previews of images, documents, and video resources.

To achieve image resource previews, researchers often extract the original image as a thumbnail. Kopf et al. [2] proposed a content-adaptive image reduction method. The kernel function consists of two Gaussian kernels defined in space and color, respectively; then, the kernel function is optimized iteratively to make it coincide with the image content. Cengiz et al. [3] proposed a perceptual-based method to reduce the size of images, which evaluates the quality of images based on human perception and retains the image details of perception. The speed of this method is faster than Kopf's algorithm. Weber et al. [4] proposed an algorithm based on a convolution filter. They believed that pixels with larger color differences from neighboring pixels contribute more to the image detail. Gastal et al. [5] proposed an image reduction method that can properly retain high-frequency features. To ensure the high precision of the fingerprint identification system, Alsmirat et al. [6] accurately determined the optimal fingerprint image compression ratio.

The online previews of document resources allow users to view files on the browser without downloading resources. Companies such as Baidu online disk used file transcoding to convert Office files to the SWF format [7]. In addition to emphasizing the presentation of the previews, the document previews also need to be based on extracting the central content of the documents. To summarize the content of the documents, researchers have proposed many methods, such as evolutionary algorithm [8], enhanced genetic algorithm [9], integer linear programming [10], deep learning [11], and matrix factorization [12]. The traditional machine learning algorithms and deep learning methods are applied to the field of document content summary.

As the duration of video resources is generally longer, users need to spend more time viewing videos in full. In order to enable users to quickly and roughly understand the content of a video, some video preview methods have

emerged. Key frame extraction is a prerequisite for video previews. Ramos et al. [13] combined the optical flow method and the colony algorithm of artificial bacteria colony to extract key frames of videos of dense crowds. For unstructured user videos, Wang et al. [14] proposed a two-layer group sparse representation method to analyze the spatio-temporal correlation of video data. Using this method to extract key frames of user video is different from the traditional clustering-based method, which relies on the signal reconstruction perspective to select key frames. Chen et al. [15] considered that different users may be interested in different segments in the same video and proposed a key frame extraction method based on user interest. Zhao et al. [16] used the golden section method in the mathematical calculation to extract key frames. After obtaining the interactive video with the resource, Ma et al. [17] used a classification algorithm to obtain key frames and the Heatmap algorithm to render the interactive area. In the field of video search, video thumbnails provide a preview of the video content. Song et al. [18] considered the aesthetic value of thumbnails and combined two important features, high relevance of video content and superior aesthetic quality. In view of the fact that traditional video thumbnails are generated only once based on the visual characteristics of the video, without considering the users' search intent, some researchers took query statements into consideration [19, 20] and combined video contexts to divide video clips. A small video related to the query sentence is used as a result. It is only a certain segment in the video and cannot summarize the overall content of the video, so it lacks overall relevance. Yuan et al. [21] proposed a new graphical convolutional video thumbnail pointer (GTP). GTP uses graph convolutional networks to model the sentence-video semantic interaction and the internal relationship of the video combined with the sentence information. Finally, the frames in the multiple video clips that are highly related to the query sentence are selected, and these frames are combined as a thumbnail of the video.

From the related work done by the predecessors in the field of resource previews, we can summarize a general pattern. First of all, we must extract the central content of the resources and then display it in a form that is convenient for users to accept. We transfer this pattern to the previews of IERs. Owing to hidden dynamic content and interactive processes being essential characteristics of these resources, we need to use automation software tools to simulate the process of the user's interaction with the resource. Some questions need to be considered in this process. The first question is how to extract the central content of IERs. According to the dynamic characteristics of the resources, we select screenshots for some time after interactive actions as the key content of the resources. The second question is in what way to present the preview. As an important type of multimedia data, the digital image has been widely used [24]; additionally, compared with videos, pictures take up less memory and users grasp the content in pictures faster, so we choose pictures as the preview form of IERs.

3. Methods

Users can interact with IERs by clicking buttons, dragging elements, etc. In this process, users' understanding of the resource can be improved, and the corresponding knowledge can be vivid and easy for the users to accept. However, due to the interactive and dynamic particularities of IERs, the content of IERs cannot be obtained by using the traditional methods of resource previews. We treat the preview of an IER as a preview to an interactive process problem; therefore, we propose a method to generate picture previews for IERs. The overall flow chart is illustrated in Figure 1. Firstly, extract interactive semantic information, that is, obtain interactive elements' information from the resource file, then extract dependencies, and generate an operation sequence. Secondly, use an automated software tool, which can simulate the interaction process between the user and the resource, execute the operation sequence, and save the screenshots in the process to generate picture previews of IERs.

3.1. Interactive Elements' Information Extraction. Various educational institutions have created their educational resource libraries, and each interactive resource library has its resource construction standards and norms. Since there are many types of interactive resources, before extracting interactive elements' information, we need a unified definition of the IERs. An IER can be abstracted as

$$\text{IER} = (\text{id}, E, \text{DR}), \quad (1)$$

where id is the number of the IER in the resource library, which is unique and E denotes a collection of elements e appearing in the resource,

$$E = \{e_1, e_2, \dots, e_n\}. \quad (2)$$

DR represents a collection of dependencies between elements as defined in the following equation:

$$\text{DR} = \{\langle e_i, e_j \rangle, e_i, e_j \in E\}, \quad (3)$$

where $\langle e_i, e_j \rangle$ is an ordered pair, which means that the former depends on the latter.

A single element e can be abstracted as

$$e = (\text{eid}, \text{class}, \text{type}, \text{position}, \text{actInfo}), \quad (4)$$

where eid indicates the number of the element in this resource, class is the category of the element, and type represents the visual characteristic of the element; the value of type can be *invisible*, *visible&static*, and *visible&interactive*, which respectively represent invisible, visible but not interactive, and visible and interactive; moreover, the *visible&interactive* elements can be divided into two subcategories, *visible&interactive&independent* and *visible&interactive&dependent*, the former does not depend on any other elements, but the latter may do; position means the coordinate of the element, and actInfo is the interaction information of the element:

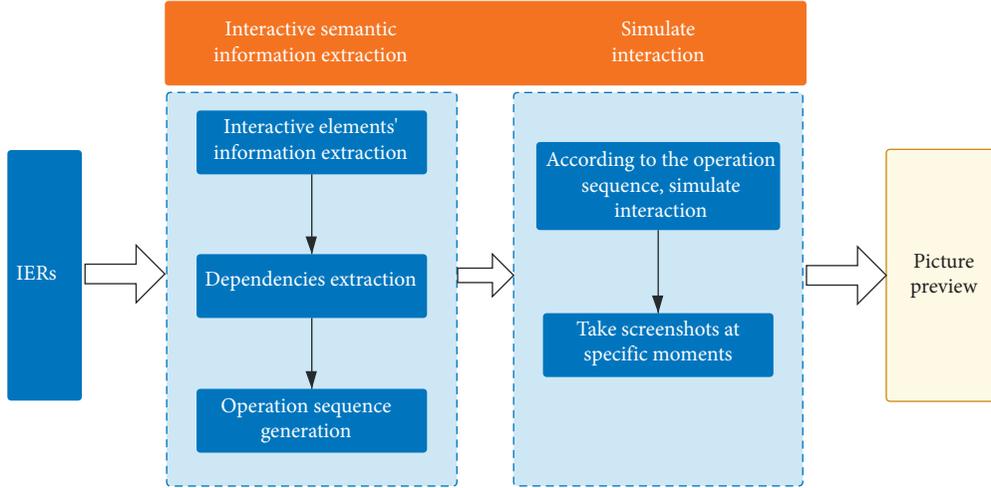


FIGURE 1: Overall flow chart.

$$\text{actInfo} = (\text{boolType}, \text{actType}), \quad (5)$$

where *boolType* takes 1 to indicate this element is visible and interactive, otherwise, takes 0; *actType* is the interaction method of the element. The values of *actType* can be *click* and *drag*; the former means that the users can interact with the element by clicking the mouse, and the latter reflects that the users can interact with the element by dragging the mouse.

In order to acquire dynamic content, to start with, we need to figure out which elements in IERs are interactive. To adapt to the general situation, we assume that the information of interactive elements in the whole library is not integrated into the resource database, which means that we do not know the names, categories, interaction modes, coordinate extraction methods, and other information of interactive elements in the whole library. Generally, we can obtain a file storing resource information in a given educational resource library; then, we need to integrate the information of interactive elements throughout the library.

Firstly, it is necessary to manually analyze some resource files to summarize the storage structure characteristics of the interactive information and then extract the names of the interactive elements by traversing a certain number of resource files in the resource library. According to the names of the interactive elements, the coordinate extraction methods and the interaction methods can be analyzed to obtain an interactive elements' information table, which lists the categories, the name attribute values, the coordinate extraction methods, and the interactive methods of the interactive elements. According to the interactive elements table, specific information about interactive elements in specific resources can be obtained. This step will be described in detail in the following application.

3.2. Dependencies Extraction. In view of the possible dependencies between the elements in an IER, we divide the dependency relationship (DR) into data dependence (DD) and control dependence (CD). DD means that if the data of an element e_i depends on the element e_j , the operation on

the element e_i needs to depend on the output data of the element e_j . CD refers to the control dependency of element e_i on element e_j , which means that e_i must be executed before e_j .

Since the user only operates on visible elements, we can consider dependencies between elements based on their visual and interactive characteristics. According to whether the elements are visible or interactive, the elements in IERs can be divided into three categories, *invisible*, *visible&static*, and *visible&interactive*. In view of the *invisible* elements being not visible, we do not take the dependencies of this type of elements with other elements into account; beyond that, the *visible&static* elements may depend on the same type of elements or *visible&interactive* elements, and the *visible&interactive* elements can only depend on the same type of elements.

Extracting the dependency relationships between interactive elements is complicated because the dependency relationships are not explicitly recorded in the resource file, and the classification of the relationship types may be different in different resource libraries. Therefore, extracting the dependency relationships between interactive elements requires realizing the classification of relationship types in practice to obtain preliminary dependencies.

Suppose we have obtained the preliminary dependency relationships, but the users finally operate directly on the *visible&interactive* elements, so we need to further filter the dependency relationships, remove the redundancies, and get the *visible&interactive* elements that the *visible&static* and *visible&interactive* elements ultimately depend on, to get the most concise dependency relationships. For example, element A depends on element B, but B depends on other elements. In this cycle, we need to find the interactive elements that element A ultimately depends on. The code for the concise dependencies is shown in Algorithm 1. Firstly, Lines 3–11 get the dependency relationships between the *visible&interactive* elements, traverse each *visible&interactive* element in the elements set, get the *visible&interactive* elements that it ultimately depends on, and then put the dependencies in a dictionary. Lines 12–20 get the

visible&interactive elements that the *visible&static* elements depend on and put the obtained dependencies into another dictionary. Line 21 clears the initial dependency relationships DR and then puts the dependency relationships stored in the two dictionaries into DR.

Algorithm 2 lists how to obtain the dependencies between *visible&interactive* elements. This is a recursive algorithm. Lines 1–4 limit the number of recursions to prevent infinite recursions. Lines 5–7 indicate that if the current element is an *invisible* element, return to the upper recursion. Lines 8–10 mean that if the current element is a *visible&interactive&independent* element, this type of element can no longer depend on other elements, so the current element is returned. Lines 11–23 are the main part of the recursion: if the current element is a *visible&interactive&dependent* element, find the elements in DR that have control dependencies on the current element (CD), and recurse with that element as the new element. Lines 24–26 indicate that if the above conditions are not met, return directly to the upper recursion.

Algorithm 3 lists how to obtain the final dependent *visible&interactive* elements of the *visible&static* elements. This is also a recursive algorithm, and the general idea is similar to Algorithm 2. Lines 11–17 indicate that if the current element is a *visible&interactive&dependent* element, it will directly return to the upper recursion, and the return value is the dependencies obtained by Algorithm 2. This step can reduce the number of recursions and speed up the algorithm running time. Lines 18–25 are the main part of the recursion: if the current element is a *visible&static* element, look for the dependent elements of the current element in DR, and then, use this element as the new element to perform recursion. Lines 26–28 indicate that if the above conditions are not met, return directly to the upper recursion.

3.3. Operation Sequence Generation. We make a definition that the performed operation sequence is composed of events triggered by the user in the user interface; different operating sequences produce different dynamic content and knowledge presentation effects; here, we define an event as

$$\text{event} = (\text{eid}, \text{actType}, \text{position}), \quad (6)$$

where *eid* is the number of the interactive element that the event affects in this resource, *actType* denotes the interaction method of the element, and *position* is the coordinate of the element. In specific, the operation sequence is composed of a set of events, that is, a series of interactive actions on *visible&interactive* elements.

In order to convey the dynamic content of the resource more comprehensively, we set a rule that all events related to *visible&interactive* elements are executed once, appearing once in the operation sequence, so that the contents of the operation sequence are obtained, and next, we need to consider the order of events. Since different operating sequences produce different dynamic content and knowledge presentation effects, we need to consider what kind of sequence can present dynamic content appropriately; we have

obtained the *visible&interactive* elements that the *visible&static* and *visible&interactive* elements ultimately depend on and will consider three aspects to generate the operation sequence, namely, the priorities of the elements, the dependencies between the elements (DR), and the layout positions in the user interface of the elements.

The element with the higher priority will be executed first, and the priority values of different categories of the *visible&interactive* elements need to be given according to different resource libraries, and specific analysis is required. After the priorities of the *visible&interactive* elements are given, we arrange all *visible&interactive* elements in the order of category priorities from high to low, getting the sequence of the initial elements in the operation sequence.

The dependency among the elements we get is usually a one-to-many relationship, that is, for a *visible&interactive* element; it may depend on several elements. We will adjust the sequence in conjunction with dependencies; in specific, if an element e_i depends on the element e_j , the priorities of the two elements are the same, but in the initial operation sequence, the event corresponding to e_j is behind the event corresponding to e_i ; then, the event corresponding to e_j is placed in the front of the event corresponding to e_i , which means that if the index of the event corresponding to e_i in the operation sequence is i , then move the event corresponding to e_i to index $i + 1$, and put the event corresponding to e_j in index i .

When a user interacts with a resource through a user interface, the order in which the user interacts with buttons has a certain relationship with the layout positions of the elements. When the button elements are arranged in an adjacent area, the user usually interacts with buttons in the order from left to right and top to bottom [25]. We apply this interaction habit to the sequence generation process.

Algorithm 4 gives the steps to generate an appropriate sequence of operations. Firstly, Lines 1–3 add all *visible&interactive* elements to the sequence of operations and then sort the elements. In specific, line 4 preliminarily sorts the elements by priorities from highest to lowest. Lines 5–14 organize the sequence according to the dependencies between the elements. Lines 15–20, for button-type elements, adjust the sequence by coordinate positions from top to bottom and left to right.

3.4. Simulate the Interaction Process. After getting the coordinates of the interactive elements and the operation sequence, the next step is to consider how to automatically interact with the resources.

Since the interaction between the user and the resource is performed by the mouse or keyboard, automated software tools can simulate the operations of the mouse and the keyboard, that is, by creating a script (a series of commands that can be executed repeatedly), the automated software tools can execute a series of mouse and keyboard actions automatically, so we use automated software tools to interact with each interactive element according to the interactive elements' operation sequence. Common automated software tools include AutoHotKey.

```

Input:
elements set  $E$ ; preliminary dependencies set DR
Output:
DR
(1) interactElementToDepElements = {}
(2) staticElementToDepElements = {}
(3) for  $e_i \in E$  and  $e_i \cdot type == visible \& interactive \& dependent$  do
(4)   dfsTimes = 0
(5)   dependentElements = []
(6)   temp = 0
(7)   temp = findRelationForInteract( $e_i$ , DR, dependentElements, dfsTimes)
(8)   if dependentElements! = Null
(9)     InteractElementToDepElements[ $e_i$ ] = dependentElements
(10)  end if
(11) end for
(12) for  $e_i \in E$  and  $e_i \cdot type == visible \& static$  do
(13)   dfsTimes = 0
(14)   dependentElements = []
(15)   temp = []
(16)   temp = findRelationForStatic( $e_i$ , DR, dependentElements, dfsTimes)
(17)   if dependentElements! = Null
(18)     staticElementToDepElements[ $e_i$ ] = dependentElements
(19)   end if
(20) end for
(21) clear DR
(22)  $DR = interactElementToDepElements \cup staticElementToDepElements$ 
(23) return DR

```

ALGORITHM 1: GenerateDR(E , DR).

There are considerations that when the interactive elements are triggered, the content of the interactive resource will change significantly and meaningfully. Only by capturing and recording those changes can users have a deeper understanding of the knowledge conveyed by the interactive resource. Therefore, in the automatic interaction process, screenshots of the user interface are taken at specific moments and the screenshots are saved. How to determine the specific moments will be explained in the fourth Section.

4. Application to Netpad

NetPad, one of the online educational resource platforms, focuses on dynamic geometry, which not only includes intelligent drawing, dynamic geometry, geometric transformation, animation, measurement, iteration, and other functions but also provides an open platform to share works and explore mathematical problems [26–28], which supports resources sharing. NetPad is popular with teachers and students for its rich teaching resources and convenient dynamic geometric drawing environment, so the number of resources on the platform is increasing dramatically.

Compared with other resources, dynamic geometric resources have distinct characteristics, containing a large number of geometric elements and hidden interactive information. If these geometric elements are associated with dynamic action, they can be interactive. Users can interact with the resource by clicking an animation button or directly dragging the interactive geometric elements and deepen

their understanding of mathematical knowledge at the same time. Figure 2 is a case of “translation of polygon.” The users can translate the polygon by clicking the animation button or dragging the green dot on the lower line segment.

4.1. Extract Interactive Element Coordinate Information and Operation Sequence. We have already defined the structure of IERs abstractly; next, we need to obtain the actual resource information (including the eid of the element, the category of the element, the position of the element in the canvas, the interaction method of the element, and the dependencies between the elements). Due to the particularity of interactive geometric educational resources, we need to use JSON format files to obtain the above resource information. The JSON file stores all the essential information of the resource, and the *content* field in the file stores the important content of the resource. When parsing the resource content, we should focus on this field, as shown in Table 1.

In the *r* field, each element information stores the following fields and attributes, as shown in Table 2.

After analyzing the JSON file structure, we need to get the actual information of the resource. However, the names of the interaction elements stored in the JSON file are unknown. The names are crucial for extracting other information of the interactive elements, so it is necessary to obtain the interactive element names of all categories as many as possible. By reading the JSON file of the first 10,000 resources in NetPad, we save the values of the *name* attribute

```

Input:
element  $e$ ; dependencies set DR; dependent interactive elements  $operateElement$ ; time of iterations  $dfsTimes$ 
Output:
 $operateElement$ 
(1) if  $dfsTimes \geq 20$ 
(2)   return
(3) end if
(4)  $dfsTimes = dfsTimes + 1$ 
(5) if  $e.type = invisible$ 
(6)   return
(7) end if
(8) if  $e.type == visible \& interactive \& independent$ 
(9)   return  $e$ 
(10) end if
(11) if  $e.type == visible \& interactive \& dependent$ 
(12)    $e_i = e$ 
(13)   for  $e_k \in \{e | e \in E \wedge \langle e_i, e \rangle \in DR\}$  do
(14)     if  $type(\langle e_i, e_k \rangle) == CD$ 
(15)        $answer = findRelationForInteract(e_k, DR, operateElement, dfsTimes)$ 
(16)       if  $answer! = Null$ 
(17)          $operateElement.append(answer)$ 
(18)       end if
(19)     end if
(20)   end for
(21)   if  $operateElement == Null$ 
(22)      $operateElement.append(e_i)$ 
(23)   end if
(24) else
(25)   return
(26) end if

```

ALGORITHM 2: Find relation for interact (e , DR, $operateElement$, $dfsTimes$).

(removing duplicate values) and analyze it to get the names of the interactive elements. Then, we analyze the examples of the JSON file according to the names of the interactive elements and get Table 3, which lists the categories, the *name* attribute values, the coordinate extraction methods, and the interactive methods of the interactive elements.

In Table 3, there are two coordinate values, (x, y) and (left, top). The former refers to the coordinate value on the coordinate system with the central point of the canvas as the origin; the latter refers to the coordinate value on the coordinate system with the point in the upper left corner of the canvas as the origin. We ultimately need to obtain the coordinate value of each interactive element on the second coordinate system, and the coordinate value of the first coordinate system can be converted into the coordinate value of the second coordinate system through mathematical calculations. Besides, the coordinate value of the point on the element (including the point on the circle, the point on the line segment, and the point on the vertical line) can be obtained through dependency analysis and mathematical calculation. For example, if we want to get the coordinate value of the point on a line segment, firstly, find the two endpoints of the line segment, and then, calculate the coordinate value.

The priority of the point on the element is set to 5, and that of the free point element is set to 4, and then, the

priorities of the variable element, the animation element, and the action element are 3, 2, and 1, respectively. We arrange the elements according to the priority from high to low and obtain the element operation sequence according to the dependencies and the layout position. Taking the case of “translation of polygon” in Figure 2 as an example, elements arranged in the operation sequence are (15, 5, 6, 13, 19), as shown in Table 4.

4.2. Get Picture Previews. After obtaining the coordinate of the interactive elements and the operation sequence, the next step is to consider how to automatically interact with the resources and to consider in what form to present this interaction process.

AutoHotKey is a free, open-source, simple, and powerful script-based hotkey and macro tool developed by Chris Mallett and his team that runs on the Windows platform [17]. AutoHotKey can send buttons and simulate mouse operations to the computer, and it has a lightweight feature, so we use AutoHotKey to simulate the interaction process between the user and the resource. Since pictures occupy less memory and express information more directly than videos, we choose to display the process by pictures. In the process of interaction, we use Gdip.ahk, the third-party library file of AutoHotKey, to take screenshots and save pictures.

```

Input:
elements set  $E$ ; dependencies set  $DR$ ; dependent interactive elements  $operateElement$ ; time of iterations  $dfsTimes$ 
Output:
 $operateElement$ 
(1) if  $dfsTimes \geq 20$ 
(2)   return
(3) end if
(4)  $dfstimes = dfstimes + 1$ 
(5) if  $e.type == invisible$ 
(6)   return
(7) end if
(8) if  $e.type == visible \& interactive \& independent$ 
(9)   return  $e$ 
(10) end if
(11) if  $e.type == visible \& interactive \& dependent$ 
(12)   if  $e$  in  $interactElementToDepElements$ 
(13)     return  $interactElementToDepElements [e]$ 
(14)   else
(15)     return  $e$ 
(16)   end if
(17) end if
(18) if  $e.type == visible \& static$ 
(19)    $e_i = e$ 
(20)   for  $e_k \in \{e | e \in E \wedge \langle e_i, e \rangle \in DR\}$  do
(21)      $answer = findRelationForStatic(e_k, DR, operateElement, dfsTimes)$ 
(22)     if  $answer \neq Null$ 
(23)        $operateElement.append(answer)$ 
(24)     end if
(25)   end for
(26) else
(27)   return
(28) end if

```

ALGORITHM 3: FindRelationForStatic (e , DR , $operateElement$, $dfsTimes$).

The specific moments to get screenshots are selected as follows:

For a free point, drag the free point once to capture a picture

For a point on the element, drag the point once to capture a picture

When the animation element is an animation button, for the animation button, get the number of steps and the interval time of the animation for a step so that we can get the total time required to play the animation once and divide the time into 3 equal parts, so as to obtain 4 key time points (start point, 1/3 time point, 2/3 time point, and endpoint; it should be noted that the selection of key time points can be set or modified according to actual needs). Click the animation button twice at these four-time points, once to pause, capture one picture, once to play, and capture a total of four pictures for one animation button.

When the action element is an action button, for the action button, click the button and take a screenshot.

When the variable element is a variable button, get the maximum value and minimum value of the variable button, which are divided into five points (minimum value, 1/4

value, median value, 3/4 value, and maximum value). The selection of the five points can be set according to actual needs. A picture is taken after each modification to the value, and a total of five pictures are taken for a variable button.

The interaction process is shown in Table 5.

Take the “translation of polygon” resource as an example. After extracting the position of the interactive element and generating the operation sequence and then using AutoHotKey to interact with the resource, the picture preview is obtained, as shown in Figure 3.

5. Experiments

We obtain a total of 7263 mathematics educational resources under the three categories of “Primary School Mathematics,” “Junior High School Mathematics,” and “High School Mathematics” in NetPad. We use these resources as samples to verify the proposed method, and the distribution of sample resources is shown in Figure 4.

We use AutoHotKey to automatically interact with these 7263 resources to generate picture previews; consequently, all preview pictures are stored in the database on the server side.

```

Input:
elements set  $E$ ; dependencies set  $DR$ ; elements to priorities mapping  $elementToPri$ 
Output:
operateList
(1) for  $e_i \in E$  and  $e_i.type == visible \& \& interactive$  do
(2)    $operateList.append(e_i)$ 
(3) end for
(4) array  $operateList$  by  $elementToPri$  from high to low
(5) for  $\langle e_i, e_j \rangle \in DR$  do
(6)   if  $e_i, e_j$  in  $operateList$  and  $elementToPri(e_j) == elementToPri(e_i)$ 
(7)     if  $e_j$  is behind  $e_i$  in  $operateList$ 
(8)       for  $e_k$  in front of  $e_j$  and  $e_k$  behind  $e_i$  do
(9)         put  $e_k$  in  $operateList$  position  $index(e_k) + 1$ 
(10)      end for
(11)     put  $e_j$  in  $operateList$  position  $index(e_i)$ 
(12)   end if
(13) end if
(14) end for
(15) for  $e_i$  in  $operateList$  do
(16)   if  $e_i$  is a button
(17)     array  $operateList$  by  $e_i \cdot position(left)$  from low to high
(18)     array  $operateList$  by  $e_i \cdot position(top)$  from low to high
(19)   end if
(20) end for
(21) return  $operateList$ 

```

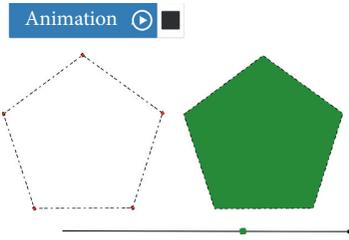
ALGORITHM 4: GetOperateList (E , DR , $elementToPri$).

FIGURE 2: A resource named “translation of polygon.”

TABLE 1: The *content* field in the JSON file.

Field name	Information	Data structure
defaultsDisplay	Display style of all elements	Array
p	Values about all variables	Array
r	Struct about all elements	Array

5.1. *Picture Preview Case.* We select two resources named “iteration of regular polygon” and “circle area” as examples.

The “iteration of regular polygon” resource is a case of iteration on a square. By analyzing the JSON file, the coordinate information and operation information of the interactive elements can be obtained. In this polygon iteration resource, there are free points, two variable buttons, and an animation button. A picture preview of the resource can be obtained by interacting with these interactive elements according to the generated operation sequence. Through the picture preview, as shown in Figure 5, we can clearly see the changes in the iteration process, which can effectively help users to choose the

required cases and to save time on manual interaction with resources.

The “circle area” resource is created to help students understand the concept of “circle area.” There are two variable buttons in the resource, where the variable ruler p controls the number of rings that divide the circle, and the variable ruler m controls the opening and closing of the rings. As shown in Figure 6, the picture preview of this resource can reflect the construction process of a solid circle clearly. It is composed of countless circles with different radiuses. Then, each circle is disassembled to obtain a line, and the lines of these circles are superimposed together to form a triangle; then, the area of the circle equals the area of the triangle. The interest of this resource can be reflected in the picture preview.

5.2. *Picture Preview Experience.* We invited 11 people from Guangzhou University to participate in the experiment, who are familiar with the interactive operations of NetPad and can objectively evaluate the effect of picture previews. We randomly selected 15 resources from the above 7263 resources, with titles highly related to the content and containing at least 3 interactive elements. These participants, respectively, open the links of the resources, manually interact with the resources, understand the content of the resources, record the time required for the interaction, and then evaluate the picture previews of the resources. The rating score interval is [1, 5]. The higher the score is, the better the picture preview summarizes.

We summarize the rating result, as shown in Figure 7. Analyzing the data in Figure 7, we can see that the scores for all resources are above 3.7, and most scores are above 4.

TABLE 2: Fields and attributes of elements in the r field.

Fields name	Attributes	Information
Fun	name	Name of element
	arg	Dependent element information
Display	show	Visible or not
	position	Coordinate value

TABLE 3: Interactive elements' information.

Interactive categories	Name attributes of interactive elements	Coordinate extraction	Interactive modes
Free point	Point	arg attribute: (x, y)	
Point on element	PointOnCircle PointOnLine PointOnVLine	PointOn PointPar PropElement	Through dependency analysis
	Variable Animation Action	PointOnVLine PointPar Variable AnimationVar ButtonFun	
		Position attribute: $(left, top)$	Drag the point once Click the button at specific moments

TABLE 4: Coordinate information of interactive elements in "translation of polygon."

Element id (eid)	Element name	Coordinate (left, top)
15	PointOnLine	(691, 703)
5	Point	(11, 352)
6	Point	(251, 175)
13	Point	(1056, 704)
19	AnimationVar	(46, 26)

TABLE 5: Interaction process by AutoHotKey.

AutoHotKey
(a) Read the operation sequence and interactive elements' information
(b) Open the browser and enter the resource display page
(c) Enter the canvas maximize mode
(d) According to the operation sequence, interact with the interactive elements in turn and use Gdip to get screenshots
(e) Save the screenshots and close the browser

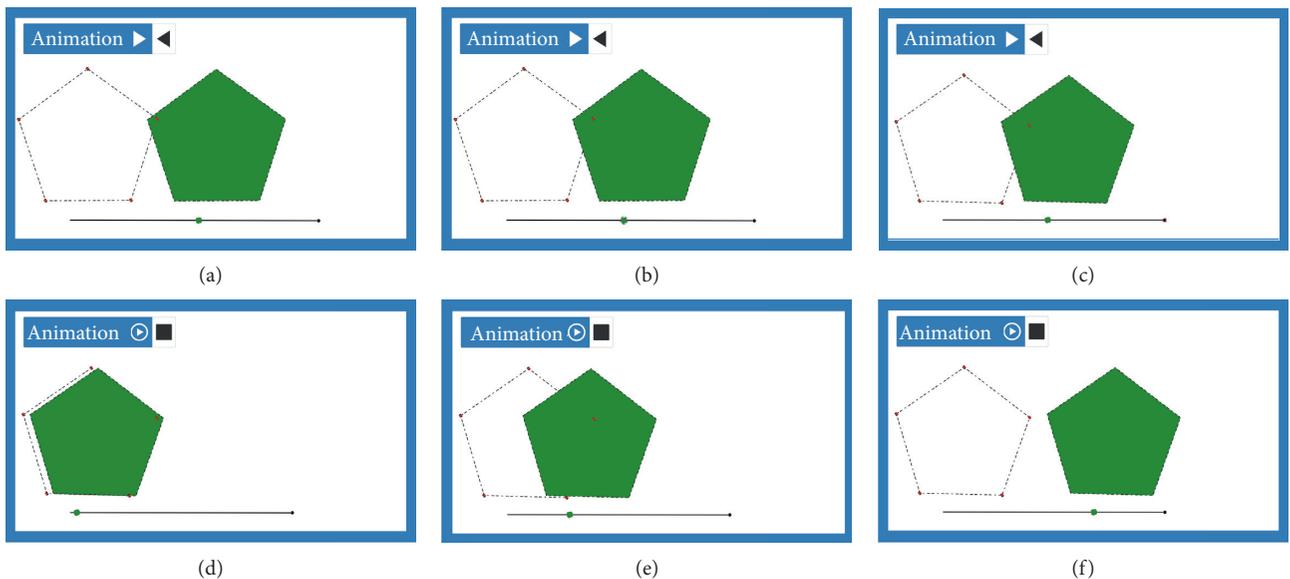


FIGURE 3: The picture preview of the interactive process. (a) Initial state. (b) Move the point on the line. (c) Move the free points. (d)–(f) Click the animation button.

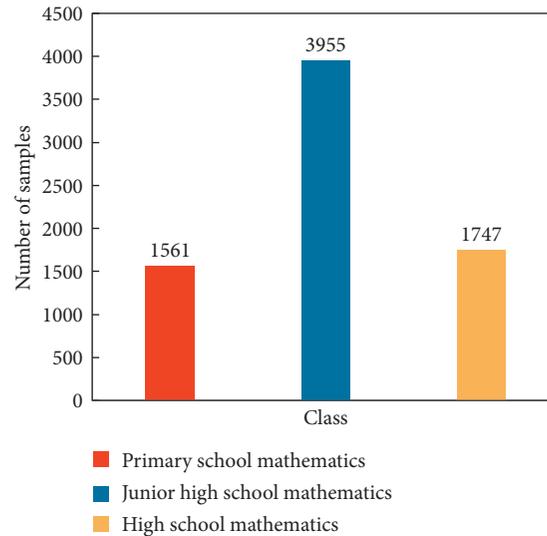
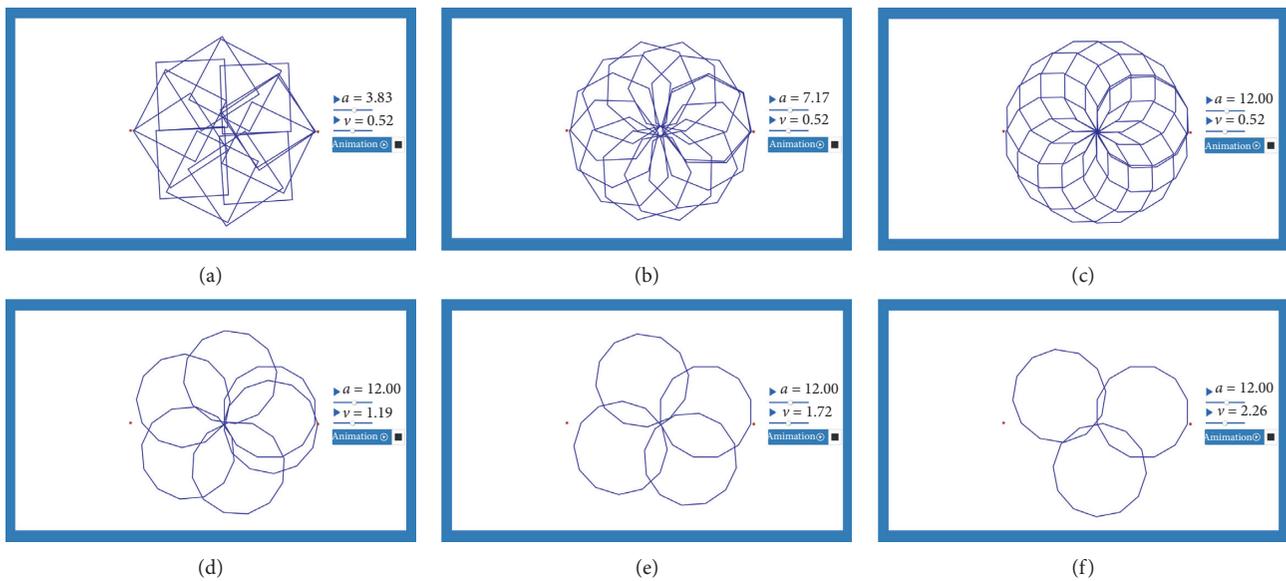


FIGURE 4: Distribution of samples.

FIGURE 5: The picture preview of “iteration of regular polygon.” (a)–(c) Click the variable ruler a ; change the number of sides of the polygon. (d)–(f) Click the variable ruler v ; change the depth of the iteration.

The average score is 4.248. This proves that the picture previews generated by our proposed method can represent most of the content of mathematical educational resources.

We will explain how much time can be saved for users by picture previews. According to the opinions of the participants, we set the time required to read one picture as 1 s, and the time required to read the resource preview corresponding to a resource is multiplying the number of pictures by the time required for a single picture.

During the experiment, the time required for each participant to manually interact with the resource was recorded. The percentage of time saved in the experiment is shown in Figure 8.

As shown in Figure 8, using the picture previews, more than 85% of the time is saved in understanding most resources, and the average time saving is 87.3%. Both the average time saving and the average picture score verify the effectiveness of our proposed method.

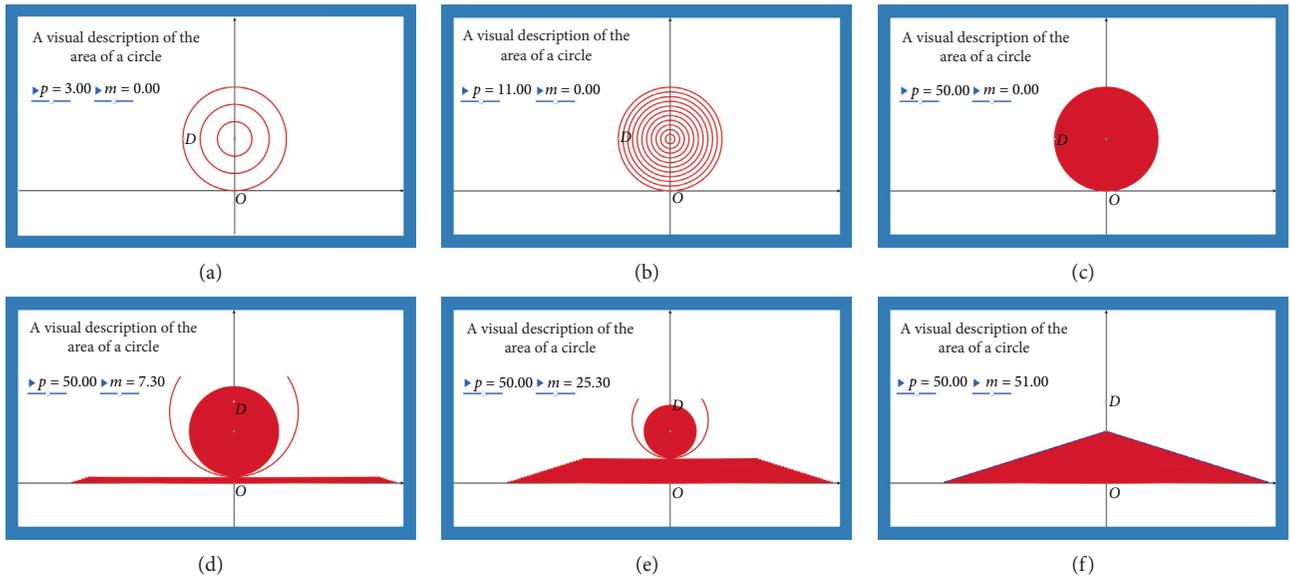


FIGURE 6: The picture preview of “circle area.” (a)–(c) Click the variable ruler p ; show the formation of the circle. (d)–(f) Click the variable ruler m ; show the decomposition of the rings.

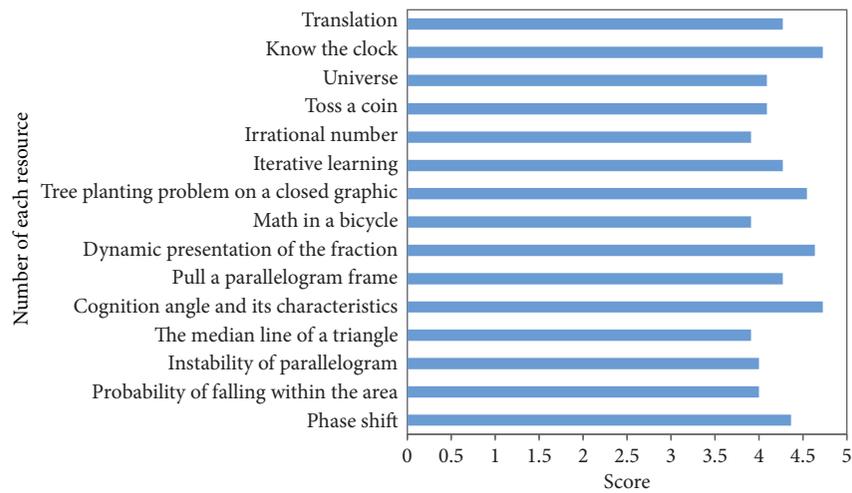


FIGURE 7: Rating of the picture previews.

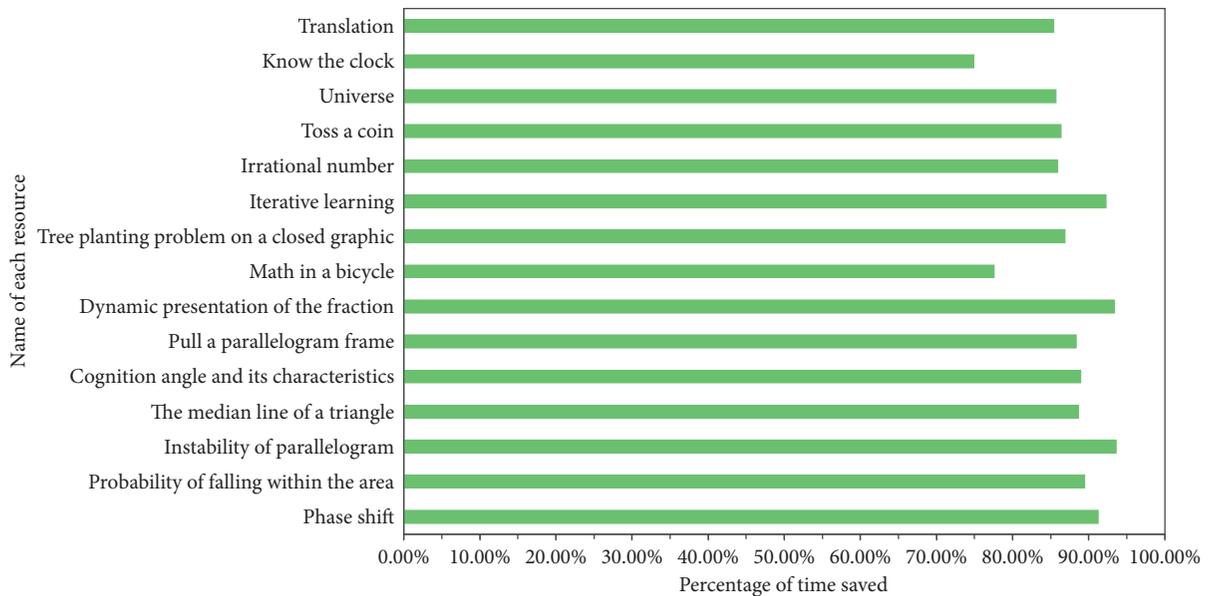


FIGURE 8: Percentage of time saved.

6. Conclusions

Due to the interactive characteristics of IERs, it requires a certain cost to understand the content of the resources. Therefore, we propose a method for generating picture previews of IERs. Firstly, we obtain interactive elements' information from the file that stores interactive resource information, extract dependencies, and then generate an operation sequence. Secondly, we execute the operation sequence using an automated software tool, in order to simulate the interaction process between the user and the resource, and save the screenshots in the process to generate picture previews of IERs. Then, we test our method on the resources in NetPad. The experimental results show that our picture previews are of great help for users to understand the content of interactive resources. It can save users 87.3% of the time on average and help users understand 88.6% of the content of resources.

In the future, we will further try to apply the picture previews to other tasks of IERs, such as resource classification and resource retrieval with pictures.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding this paper.

Acknowledgments

This work was supported by the National Key R&D Program of China (no. 2018YFB1005100), Guangzhou Academician and Expert Workstation (no. 20200115-9), and National Natural Science Foundation of China (no. 61772140).

References

- [1] H. Guan, X. L. Qin, and Y. S. Rao, "Research and design of an open platform for dynamic mathematics and digital resources," *Journal of Harbin Institute of Technology*, vol. 51, no. 5, pp. 14–22, 2019.
- [2] J. Kopf, A. Shamir, and P. Peers, "Content-adaptive image downscaling," *ACM Transactions on Graphics*, vol. 32, no. 6, pp. 173.1–173.8, 2013.
- [3] A. C. Öztireli and M. Gross, "Perceptually based downscaling of images," *ACM Transactions on Graphics*, vol. 34, no. 4, pp. 77.1–77.10, 2015.
- [4] N. Weber, M. Waechter, S. C. Amend, S. Guthe, and M. Goesele, "Rapid, detail-preserving image downscaling," *ACM Transactions on Graphics*, vol. 35, no. 6, pp. 205.1–205.6, 2016.
- [5] E. S. L. Gastal and M. M. Oliveira, "Spectral remapping for image downscaling," *ACM Transactions on Graphics*, vol. 36, no. 4, pp. 145.1–145.16, 2017.
- [6] M. A. Alsmirat, F. Al-Alem, M. Al-Ayyoub, Y. Jararweh, and B. Gupta, "Impact of digital fingerprint image quality on the fingerprint recognition accuracy," *Multimedia Tools and Applications*, vol. 78, no. 3, pp. 3649–3688, 2018.
- [7] C. Strobbe, B. Frees, and J. Engelen, "An accessibility checker for libreoffice and openoffice.org writer," *Lecture Notes in Computer Science*, vol. 7382, pp. 484–491, 2012.
- [8] Y. K. Meena and D. Gopalani, "Evolutionary algorithms for extractive automatic text summarization," *Procedia Computer Science*, vol. 48, pp. 244–249, 2015.
- [9] B. T. M. Anh, N. T. My, and N. T. T. Trang, "Enhanced genetic algorithm for single document extractive summarization," in *Proceedings of the ACM International Symposium on Information and Communication Technology*, pp. 370–376, Hanoi, Vietnam, December 2019.
- [10] S. Banerjee, P. Mitra, and K. Sugiyama, "Multi-document abstractive summarization using ILP based multi-sentence compression," in *Proceedings of the AAAI International Joint Conference on Artificial Intelligence*, pp. 1208–1214, Buenos Aires, Argentina, July 2015.
- [11] R. Nallapati, F. F. Zhai, and B. W. Zhou, "SummaRuNNer: a recurrent neural network based sequence model for extractive summarization of documents," in *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 3075–3081, San Francisco, CA, USA, February 2017.
- [12] M. T. Nguyen, T. V. Cuong, and N. X. Hoai, "Exploiting user comments for document summarization with matrix factorization," in *Proceedings of the ACM International Symposium on Information and Communication Technology*, pp. 118–124, Hanoi, Vietnam, December 2019.
- [13] J. Ramos, N. Nedjah, L. de Macedo Mourelle, and B. B. Gupta, "Visual data mining for crowd anomaly detection using artificial bacteria colony," *Multimedia Tools and Applications*, vol. 77, no. 14, pp. 17755–17777, 2018.
- [14] Z. S. Wang, M. Kumar, J. B. Luo, and B. X. Li, "Extracting key frames from consumer videos using bi-layer group sparsity," in *Proceedings of the ACM International Conference on Multimedia*, pp. 1505–1508, Scottsdale, AZ, USA, November 2011.
- [15] X. Chen, Y. F. Zhang, Q. Y. Ai, H. T. Xu, and Z. Qin, "Personalized key frame recommendation," in *Proceedings of the International ACM SIGIR Conference*, pp. 315–324, Shinjuku, Japan, August 2017.
- [16] A. D. Zhao, Y. Lai, Y. Liu, and H. B. Leng, "Key frame extraction algorithm for surveillance video based on golden section," in *Proceedings of the ACM International Symposium on Signal Processing Systems*, pp. 78–81, Beijing, China, February 2019.
- [17] C. Ma, X. P. Chen, Y. S. Rao, and M. G. Lin, "Generating summarized preview for Education Resource based on exploring and comparing GUIs," in *Proceedings of the KSI Research International Conference on Software Engineering and Knowledge Engineering*, pp. 141–146, San Francisco, CA, USA, July 2016.
- [18] Y. L. Song, M. Redi, J. Vallmitjana, and A. Jaimes, "To click or not to Click: automatic selection of beautiful thumbnails from videos," in *Proceedings of the ACM International on Conference on Information and Knowledge Management*, pp. 659–668, Indianapolis, IN, USA, October 2016.
- [19] J. Y. Chen, X. P. Chen, L. Ma, Z. Q. Jie, and T. S. Chua, "Temporally grounding natural sentence in video," in *Proceedings of the Association for Computational Linguistics Conference on Empirical Methods in Natural Language Processing*, pp. 162–171, Brussels, Belgium, November 2018.
- [20] M. Liu, X. A. Wang, L. Q. Nie et al., "Attentive moment retrieval in videos," in *Proceedings of the International ACM SIGIR Conference*, pp. 15–24, Ann Arbor, MI, USA, June 2018.

- [21] Y. T. Yuan, L. Ma, and W. W. Zhu, "Sentence specified dynamic video thumbnail generation," in *Proceedings of the ACM International Conference on Multimedia*, pp. 2332–2340, Nice, France, October 2019.
- [22] B. B. Gupta and M. Quamara, "An overview of Internet of Things (IoT): architectural aspects, challenges, and protocols," *Concurrency and Computation: Practice and Experience*, vol. 32, no. 21, 2020.
- [23] H. Wang, Z. Li, Y. Li, B. B. Gupta, and C. Choi, "Visual saliency guided complex image retrieval," *Pattern Recognition Letters*, vol. 130, pp. 64–72, 2020.
- [24] C. Yu, J. Li, X. Li, X. Ren, and B. B. Gupta, "Four-image encryption scheme based on quaternion Fresnel transform, chaos and computer generated hologram," *Multimedia Tools and Applications*, vol. 77, no. 4, pp. 4585–4608, 2018.
- [25] Z. Liu, J. Mao, C. Wang, Q. Ai, Y. Liu, and J. Y. Nie, "Enhancing click models with mouse movement information," *Information Retrieval Journal*, vol. 20, no. 1, pp. 53–80, 2017.
- [26] Y. Wang, Y. S. Rao, H. Guan, and Y. Zou, "NetPad: an online DGS for mathematics education," in *Proceedings of the IEEE International Conference on Computer Science and Education*, pp. 305–309, Houston, TX, USA, December 2017.
- [27] H. Guan, Y. S. Rao, Y. Wang, X. X. Li, and R. X. Chen, "Design and implementation of web-based dynamic mathematics intelligence education platform," in *Proceedings of the IEEE International Conference on Digital Home*, pp. 141–147, Guilin, China, November 2018.
- [28] Y. Rao, J. Zhang, Y. Zou, Y. Sun, X. Chen, and S. Xu, "An advanced operating environment for mathematics education resources," *Science China Information Sciences*, vol. 61, no. 9, pp. 1–3, 2018.