An interactive video-based learning environment that supports learning analytics for teaching 'Image Editing'

Alexandros Kleftodimos[1,2] and Georgios Evangelidis[2]

[1] Department of Digital Media and Communication
Technological Education Institute of Western Macedonia, Kastoria, Greece
kleftodimos@kastoria.teikoz.gr
[2] Department of Applied Informatics, School of Information Sciences
University of Macedonia, Thessaloniki, Greece
{akleftodimos, gevan}@uom.gr

Abstract. The use of online videos is a common practice amongst education professionals and the interactive features found in these videos are constantly evolving. In a previous research paper we presented a roadmap on how open source technologies and open internet resources can be used to build a video based learning environment that supports learning analytics. This paper describes how an environment for teaching image editing techniques is build using similar principles. The video based environment is used in educational settings and a dataset of learner activity behaviors is obtained. Cluster analysis is then used for identifying groups of students with similar viewing and activity behaviors and an attempt is made to interpret the results.

Keywords: Interactive Educational Videos, Video Learning Analytics, Open Educational Resources

1 Introduction

Nowadays, educational video distribution over the Internet is a widespread practice. Together with the increase in the use of educational videos there is also an increase in the features that accompany these videos. Interactivity in educational videos is a relatively new trend with the level and types of interactivity to be in constant evolution. This is evident from the appearance of new tools and online platforms for creating and hosting interactive videos. Examples of such platforms are Zaption[1], EdPuzzle[2], EduCanon[3], Vialogues[4] and Raptmedia[5]. These platforms support a different set of features (e.g., in-video quizzes, time based

[1] http://www.zaption.com
[2] http://edpuzzle.com
[3] https://www.educanon.com
[4] http://vialogues.com
[5] http://www.raptmedia.com

SE@VBL 2016 workshop at LAK'16, April 26, 2016, Edinburgh, Scotland

discussions, branching videos etc.) and aim at making the video learning process a more interactive experience. However, most of these applications either do not come for free (e.g., Raptmedia) or offer only a limited set of features for free (e.g., Zaption, Educanon). Moreover, these applications are not open source meaning that they are not open to further development or customization by independent developers.

In a previous research paper [3] we provided a roadmap on how open source tools and open internet resources can be used to build interactive video based learning environments that could incorporate a range of interactive features and also support learning analytics. In this paper we describe how we used similar principles to build an environment for learning image editing techniques with the use of video and 'image editing' activities.

More specifically, in Section 2 we describe the details of a video based learning environment for learning 'image editing techniques' together with the educational settings in which the environment was used. In Section 3 we describe a module for storing learner viewing behaviours and actions. In Section 4 cluster analysis is carried out in order to identify groups of students with similar behaviour. The paper concludes in Section 5.

2 Building a video based environment for teaching 'Image Editing techniques'

In [3] we described how open source technologies and open internet resources are used for creating time based interactive videos. By using the API of Media Element js [6] actions can be initiated when specific time points (or intervals) are reached in the video timeline or when certain video events occur (e.g., pause, resume, start/end of video, volume change). An action that can be initiated is the retrieval (or storage) of content from (or to) a database and this is the basis for building time based interactive videos. Using this technique various features could be incorporated into a video based environment, like, in-video quizzes, subtitles, sections and table of contents, video and web content aggregation, discussions, etc.

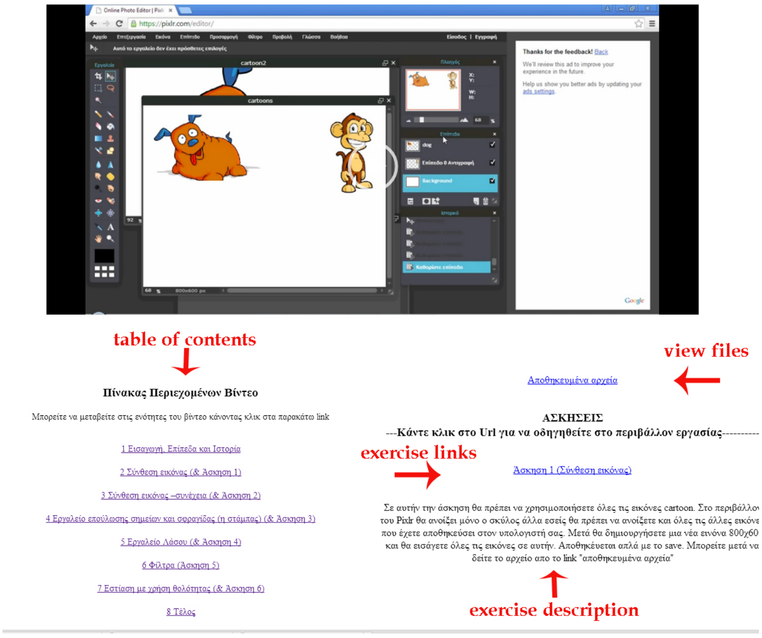
In this section we describe the features of an environment that is build using similar principles. This environment combines an instructional video together with a web based image editing tool, for teaching image editing techniques. The web based tool that is chosen is Pixlr[7], a tool that is free to use and which has similar features to Photoshop. Pixlr also provides an API that enables web developers to use the application on their own site.

For a first semester course called 'Image and Video Editing Principles' an instructional video is produced in order to teach students the basics of image editing using Pixlr. Students are required to perform a login procedure in order to enter a platform with video lessons. One of the video lessons is the lesson

[6] http://mediaelementjs.com

[7] https://pixlr.com

on 'Image Editing using Pixlr'. Next to the video link there is also a link to a page containing detailed instructions. Students are strongly advised to visit the instructions webpage first before clicking on the video link. When the video link is clicked learners are directed to an environment that contains the video together with other features as shown in Figure 1.



Πίνακας Περιεχομένων Βίντεο

Μπορείτε να μεταβείτε στις ενότητες του βίντεο κάνοντας κλικ στα παρακάτω link

1 Εισαγωγή, Επίπεδα και Ιστορία

2 Σύνθεση εικόνας (& Άσκηση 1)

3 Σύνθεση εικόνας –συνέχεια (& Άσκηση 2)

4 Εργαλείο επούλωσης σημείων και σφραγίδας (η στάμπας) (& Άσκηση 3)

5 Εργαλείο Λάσου (& Άσκηση 4)

6 Φίλτρα (Άσκηση 5)

7 Εστίαση με χρήση θολότητας (& Άσκηση 6)

8 Τέλος

Αποθηκευμένα αρχεία

ΑΣΚΗΣΕΙΣ

---Κάντε κλικ στο Url για να οδηγηθείτε στο περιβάλλον εργασίας----------

Άσκηση 1 (Σύνθεση εικόνας)

Σε αυτήν την άσκηση θα πρέπει να χρησιμοποιήσετε όλες τις εικόνες cartoon. Στο περιβάλλο
του Pixlr θα ανοίξει μόνο ο σκύλος άλλα εσείς θα πρέπει να ανοίξετε και όλες τις άλλες εικόνε
που έχετε αποθηκεύσει στον υπολογιστή σας. Μετά θα δημιουργήσετε μια νέα εινόνα 800x60
και θα εισάγετε όλες τις εικόνες σε αυτήν. Αποθηκέυεται απλά με το save. Μπορείτε μετά να
δείτε το αρχείο απο το link "αποθηκευμένα αρχεία"

Fig. 1. A video based learning environment for teaching 'Image Editing'

The video occupies the top of the page and under the video there is an area with a table of contents (on the left) with links to different topic sections. By sections we mean the logical segmentation of video content into segments that cover a particular subtopic. The video lesson on image editing is divided into seven logical sections and the starting points of these sections are stored in the database. The first section is a general introduction to Pixlr and the six sections that follow present different image editing techniques. The techniques are presented by descriptive exercise implementations. We also defined one extra dummy section (or marker point) in the end in order to provide a link (in the table of contents) to the end of the video.

On the right side of the webpage (under the video) there is an area where exercise descriptions appear together with links that learners have to follow to complete the exercises. Each exercise (six in total) is associated with a different section and appears only during that section. Students are prompted to follow the link in order to complete a similar exercise as the one described in the related section. When the link is followed the Pixlr environment opens at a different browser window (or tab) together with the related image files that are needed in order to complete the exercise. For the autumn semester of 2015-2016, the exercises were part of an assignment that counted for the 15% of the final mark.

Although Pixlr could also be opened as separate iframe in the same webpage and next to the video lesson, we chose not to follow this option since we wanted to give enough browser space to the video and the application and have menu actions related to the Pixlr software be clearly visible. After completing the exercise, the learner can save it by pressing the 'Save' menu item in Pixlr. The image file of the completed exercise is saved on the learning environment web server and not on the students computer (or the students account on the Pixlr web server). This is achieved by using the Pixlr API. We chose this option since students would have to deliver the same exercises and we wanted to make sure that cheating is avoided and that students use their accounts to view the videos and complete the exercises. Learners are able to view the stored files whenever they want by pressing on a relevant link on the environment interface. Students that used this environment to complete the assignment were told in the instructions that images above 1000 width or 1000 pixel height would not be stored on the server and that was done in order to avoid delays and storage problems. An image with a relevant informative message was saved in case the student attempted to save an image that exceeded these dimensions.

3 Storing learner activity

Analysis of video viewing data is a relatively recent trend and there is a small but increasing body of on-going research that focuses on the topic (e.g., [1, 2, 4]). In our previous research we described a system module that is responsible for capturing and storing learner activity while watching educational videos and performing interactive activities. Events triggered during video execution are stored in the database and are associated with a specific viewing session. Most of the events are triggered by the Html 5 video player API but some are also created. The events that are specific to the Media Element API and are used for tracking learner activity are the following: a) loadeddata, called when the video is loaded, b) seeked, called when the learner 'seeks' by moving the video play head, c) play, called when the video starts playing or resumes after a pause, d) pause, called when the video is paused by the viewer, e) ended, called when the video reaches its end, f) volumechange, called on sound volume change, and, g) muted, called on sound mute. A set of properties are also retrieved when these events occur such as the video time, the current date and time, etc. The full list of properties, events and methods of the Media Element js API is provided in the Media Element web page.

The 'timeupdate' is an event that fires few times in every second and it can be used to track the state of the video. This event can also be used to build more events. For example, if at some instance the current video time is higher than the video time recorded a second before by two seconds or more, then we can safely assume that a jump took place (in our setting we use the threshold of 5 seconds to record a jump). A backward or forward jump can take place if the learner pauses the video at a certain point and resumes at another or when he/she moves the play head (or slide bar) to a new position in the video timeline. The events that are created using the 'timeupdate' event are the following: a) section enter, b) time section enter, and, c) jump event. 'Section enter' event is called when the play head enters one of the educator defined sections. As already mentioned, sections defined by the educator reflect different conceptual topics. Sections can also play the role of marker points in the video. When a marker point is reached, or when the video play head lands in a section after a jump, an entry is stored in the database together with the current date and time. Another way of splitting the video is through equal time intervals (rather than different conceptual topics). In this case the markers are inserted in equal time intervals. The 'time section enter', event is triggered when the playhead reaches one of these intervals either during sequential video execution or after a jump. The time interval is set by the educator and stored in a general parameters database table. For the specific video the interval was two minutes meaning that the 36 minute video was split into 18 intervals. Finally, the jump events are called when a jump takes place and two entries are stored in the database (with two different codes). The first entry indicates the video time point of where the playhead was before the jump, and the other entry the video time point after the jump.

The section enter events together with the events pause, play and jump can give us a good estimate of the video portions viewed by learners. One advantage of logging viewing activity into a database is that many things can be quickly calculated by using the SQL query language (e.g., number of jumps and logical sections visited).

Actions that are related to the Pixlr activities are also stored in a separate database table. The additional actions that are stored in the database are the following: a) visiting the instructions page, b) following one of the exercise related links (six actions for the six different exercises), c) saving an image file (again six 'save' actions for the six different activities), d) viewing the image files stored on the server, and, e) clicking on a section link of the table of contents in order to be directed to a particular section.

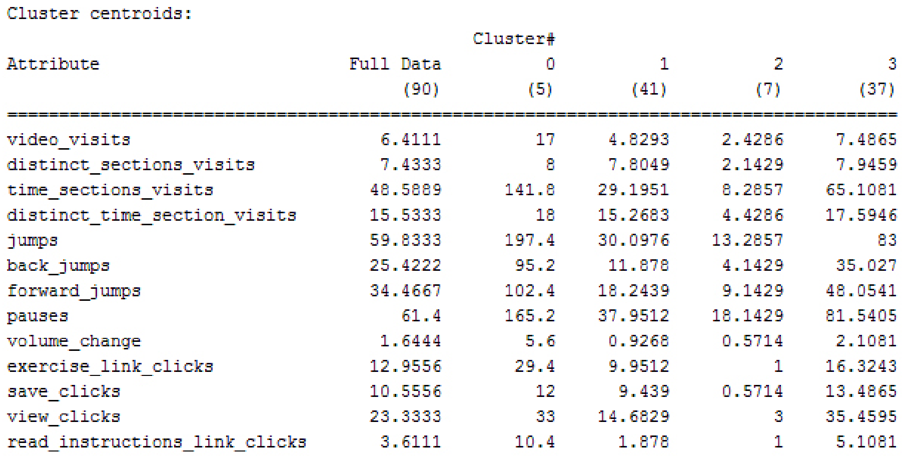
4 Using cluster analysis to obtain insights about learner behavior

In this section, we present some metrics related to learners' activity that were used to carry out cluster analysis with Weka [8]. The method of clustering is used in order to reveal different groups of students with similar viewing and activity behaviours and then an attempt is made to interpret the results.

More specifically, the following metrics are obtained from the dataset: a) number of video visits, b) distinct section visits (logical sections defined by the

[8] http://www.cs.waikato.ac.nz/ml/weka

educator), c) time section visits, this indicator is proportional to the time spend on active video viewing, d) distinct time sections visited, the video duration is 36 minutes and it is split into 18 equal (2 minute) time intervals, e) jumps, the number of forward and backward jumps that took place during video viewing, f) backward jumps, g) forward jumps, h) pauses, the number of times that the learner pressed the pause button, i) volume change, the number of time the learner attempted to change the volume, j) exercise link clicks, the number of times that an exercise link was clicked by a learner, k) save clicks, the number of times that a learner saved an image file, l) view files clicks, the number of times that the learner viewed the 'saved' image files, and, m) read instruction clicks, the number of times that the learner visited the instructions page. These metrics are then used for carrying out cluster analysis. The analysis is carried out using the K-means algorithm. To determine the optimum number of clusters we used the 'within sum of squared error SSE' in order to assess the clustering results. For each observation, the error is the distance to the nearest cluster. To get SSE, the errors are squared and summed up. To obtain the optimum K, we start from one cluster and continue adding clusters until diminishing returns are achieved, meaning no significant reduction in within SSE. Of course another important factor in accepting the resulting clusters is that we should be able to interpret them. The results that were obtained after following this process with Weka are shown in Figure 2.



```
Cluster centroids:
                                            Cluster#
Attribute                      Full Data           0           1           2           3
                                    (90)         (5)        (41)         (7)        (37)
========================================================================================
video_visits                      6.4111          17      4.8293      2.4286      7.4865
distinct_sections_visits          7.4333           8      7.8049      2.1429      7.9459
time_sections_visits             48.5889       141.8     29.1951      8.2857     65.1081
distinct_time_section_visits     15.5333          18     15.2683      4.4286     17.5946
jumps                            59.8333       197.4     30.0976     13.2857          83
back_jumps                       25.4222        95.2      11.878      4.1429      35.027
forward_jumps                    34.4667       102.4     18.2439      9.1429     48.0541
pauses                              61.4       165.2     37.9512     18.1429     81.5405
volume_change                     1.6444         5.6      0.9268      0.5714      2.1081
exercise_link_clicks             12.9556        29.4      9.9512           1     16.3243
save_clicks                      10.5556          12       9.439      0.5714     13.4865
view_clicks                      23.3333          33     14.6829           3     35.4595
read_instructions_link_clicks     3.6111        10.4       1.878           1      5.1081
```

Fig. 2. Results from Cluster Analysis using Weka

Cluster 0 comprises of students that have a large number of video views, large number of jumps and time section visits. These students visited all logical sections of the video and also all the time sections (18 in total). However, the troubling fact is that these students also exhibit a large number of exercise visits and a large number of visits to the instructions page. A session where video viewing and exercises are performed without trouble is expected to be in line with the following scenario: in the beginning, the learner visits the instruction page, and, after carefully reading the instructions, he/she proceeds to view the related video. After (or while) viewing the second section of the video that is associated with exercise 1, the learner presses on the exercise link and performs the first exercise in the Pixlr environment. After completing the exercise the learner saves the image file and views the file by pressing the 'view-files' link. In the same manner the learner views the subsequent sections and performs the rest of the exercises. Ideally, the learner would click the exercise link only once in order to perform the exercise. The high number of clicks to the exercise link and the high number of file views as well as the high number of visits to the instruction page are a sign that the learner is disorientated in the environment.

Cluster 3 consists of learners that have a lower score on the mentioned indicators. However, still the number of exercise link clicks in the particular cluster is significantly higher than the number of exercises (6 exercises). Thus these learners also faced some problems. By looking in the dataset at a number of sessions from learners in this cluster we observed that a lot of these sessions were 'troubled' at the beginning (e.g., multiple clicks to the exercise links, visits to the instruction page, high number of jumps) but normalized after a while.

Cluster 1 is a less 'troubled' cluster. This cluster is characterized by low number of exercise visits. The exercise visits in this cluster are close to the number of exercises. There are also less visits to the instruction page and also fewer jumps. Moreover, there are fewer distinct time section visits and this means that a number of students in this cluster skipped some time segments of the video probably because it was clear to them how they should proceed.

Finally **Cluster 2** consists of learners that did not finish the assignment since the number of exercise and 'save' clicks are less than the number of the assignment exercises.

During the completion of the assignment a number of students contacted the educator in order to express problems that they were facing. The students encountered two types of problems mainly because they did not understand the instructions. One problem was that students did not understand that in order to perform the 6 exercises they would have to follow 6 different links that appeared in the exercise area during the related video sections. Another problem was that some of them tried to store images with dimensions larger than the defined limit. An image with a relevant message was saved in this case but not all students understood what this message was about. Out of the 10 students that expressed their problems by email 2 of them were members of cluster 0, 7 were members of cluster 3 and 1 from cluster 1.

A survey was also handed out to the students via Google forms in order for students to evaluate the environment. 61 students took part in the survey out of the 90 that attempted the assignment. One of the survey questions asked students whether they faced any of the two problems mentioned above (or any other problem). Out of the 22 students who stated that they had at least one of the two problems described above, 16 were found to be members of cluster 3, 1 in cluster 0 and 4 in cluster 1.

Thus, we observe that the clustering results are in line with the questionnaire and the emails. However, the data suggests that more students faced problems than those who stated so in the survey. To summarize, the clustering scheme revealed clusters of students with differences in the scores of the various indicators. In these educational settings the higher scores in these indicators were associated with problems that students faced during the assignment completion. Performance issues are not discussed in this study since once the students got acquainted with the environment features they completed the activities with success.

5 Conclusions

The purpose of this paper was to present a video based learning environment that supports learning analytics for teaching ‘image editing techniques’. After using this environment in educational settings as part of an assignment we obtained a dataset of viewing & activity behaviours. Indicators from this dataset were used in a clustering scheme to obtain groups of learners with similar characteristics. The clustering scheme helped us distinguish between learners that seem to have completed the assignment without any problems and those who encountered problems. This clustering scheme could again be used in the future as a method to assess the environment after having made all the necessary amendments in order to overcome the problems that were caused mainly due to misunderstanding of the instructions.

References

1. Giannakos, M., Chorianopoulos, K., N., C.: Making sense of video analytics: Lessons learned from clickstream interactions, attitudes, and learning outcome in a video-assisted course. In: The International Review of Research in Open and Distance Learning. vol. 16 (1), pp. 260–283 (2015)
2. Kim, J., Guo, P.J., Seaton, D.T., Mitros, P., Gajos, K.Z., Miller, R.C.: Understanding in-video dropouts and interaction peaks inonline lecture videos. In: Proceedings of the First ACM Conference on Learning @ Scale Conference. pp. 31–40. L@S '14, ACM, New York, NY, USA (2014), http://doi.acm.org/10.1145/2556325.2566237
3. Kleftodimos, A., Evangelidis, G.: An interactive video-based learning environment supporting learning analytics: Insights obtained from analyzing learner activity data. In: Li, Y., Chang, M., Kravcik, M., Popescu, E., Huang, R., Kinshuk, Chen, N.S. (eds.) State-of-the-Art and Future Directions of Smart Learning, pp. 471–481. Lecture Notes in Educational Technology, Springer Singapore (2016), http://dx.doi.org/10.1007/978-981-287-868-7_56
4. Li, N., Kidzinski, L., Jermann, P., Dillenbourg, P.: How Do In-video Interactions Reflect Perceived Video Difficulty? In: Proceedings of the European MOOCs Stakeholder Summit 2015. pp. 112–121. P.A.U. Education (2015)