A Receptive-Relation Understanding of Information Paradigm Change

Salvador Ruiz-de-Maya

Abstract

Virtual laboratories are used in online education, corporate training and professional skill development. There are several aspects that determine the value and effectiveness of virtual laboratories, namely (i) the cost of development which includes the cost of creating the virtual environment and designing and implementing the laboratory exercises, (ii) the benefits brought to the trainees compared with those provided by traditional physical laboratories, and (iii) the operation which includes the communication between trainers and trainees, the authentication and remote proctoring of the trainees, etc.

In this paper, a virtual laboratory system with biometric authentication and remote proctoring by employing facial recognition techniques is introduced. The general logic and basic algorithms used to enable biometric authentication and remote proctoring are described. When using this virtual laboratory system, the students log in by scanning their faces with a camera. While performing a laboratory exercise, the students sit in front of the camera and the virtual laboratory system monitors their facial expressions and head motions in order to identify suspicious behaviors. Upon detection of such suspicious behaviors, the system records a video for further analysis by the laboratory administrator. An evaluation of the feasibility of this approach is presented.

Introduction

As one of the most important implementations of virtual reality (VR), virtual laboratories (VLs) are becoming more and more popular at various levels of education and in various fields of training. There are several factors that speed up the development of VL systems. The first factor is the wide-spread adoption of the Internet which provides the possibility of remote access to VLs (including the remote control of physical devices and the communication between distance users). [1] The second factor is the development of advanced computer graphics which enables the visualization of the real world in virtual environments with CAD software or real-time rendering tools. [2,3] The third factor is the appearance of artificial intelligence which enables developers to implement virtual activities. [4] Other research in materials, electronic components and devices, etc. also boosts the development of VL systems. [5] VL systems are inherently safer and less failure prone than their physical equivalents. This renders them an alternative solution for dangerous and costly training programs (e.g., firefighter training, military training, disaster relief training, new-employee training, etc.). VL systems can be accessed remotely by the learners. Therefore, they provide an efficient way for sharing limited education resources (including experimental devices, excellent tutors and other educational materials). [6,7]

An ideal VL system should include three principal features, namely (i) a set of virtual devices that meet the requirements of the specific experiments, (ii) a virtual space in which all of the interactions between the participants and experimental devices can take place, and (iii) a set of authentication and proctoring mechanisms that can verify the users' identity and ensure the academic integrity during experiments. Besides these three features, VL systems should provide the users with a feeling of immersion in order to increase their learning interest. At the same time, VL systems would gain more widespread acceptance if they were extensible and customizable (i.e. enable the users to create their own models, avatars and other virtual features).

Numerous implementations of VLs have recently been reported. [8,9,10] They are often based on some ready-to-use VL engines, for example game engines. These engines provide various basic functions such as physics modeling, graphics rendering, sound generation, game logics, artificial intelligence, user interactions and networking.

[11,12] The main benefit of using VL engines is the convenience of development to realize immersion, distribution and collaboration. [13] At present, the developers of VLs are mainly focusing on the creation of the virtual environment (which includes a virtual space and virtual models) based on specific game engines. This kind of work can be defined as virtualization of the real world. There are two main methods for creating virtual environments. The first method is to use CAD software to manually create computer models of the real world. [14] The second method is to use 3D reconstruction techniques to automatically create the virtual environment in real-time. [15,16] The flow charts of the above two methods are illustrated in Figure 1. In the manual method, the real world is measured with traditional measurement tools in order to obtain all necessary geometric parameters. Then, built-in toolkits of the VL engines or third-party 3D software are employed to create the virtual models of the real objects. Finally, the models created are converted into a ready-to-use format for the specific VL engine. In contrast, the real-time method employs a 3D reconstruction technique in which the real world is surveyed with one or multiple non-contact scanners, the acquired raw surface data are processed and the final virtual environment is created. [17]

Although 3D reconstruction techniques provide great convenience in the creation of VLs, there is one feature that is still absent, namely a set of authentication and proctoring tools. The prevalent method for logging into a VL system is based on a username and password. The most common method for supervising an experiment is that the proctors (i.e. laboratory administrators, laboratory instructors or hired proctors) monitor the participants with a surveillance camera system. [18,19] These proctors are located at the server side of the virtual laboratory. They supervise the entire process of the experiment by monitoring video feeds on a screen. This method has two disadvantages. One shortcoming is the cost for setting up the surveillance system and the other disadvantage is the laborious nature of the work of the proctors. In order to avoid these two disadvantages, some virtual proctoring software can be employed, for example Instant-In Proctor™ [20] Proctortrack [21], Proctorfree [22] and Securexam Remote Proctor. [23] Third-party virtual proctoring tools give educational institutions greater flexibility in the timing of the laboratory experiments. In addition, after the learners sit in front of a camera, the proctoring software keeps track of their mouse clicks, faces and computer screens. The software functions just like an instructor who is located at the front of the classroom and is constantly looking at the learners, and in fact it may even be more efficient.

Unfortunately, many VL implementations for remote education fail to take advantage of this useful proctoring technique, because biometric authentication techniques have not been adopted for VL implementations yet. In order to close this gap, a VL with biometric authentication and remote proctoring is introduced here.

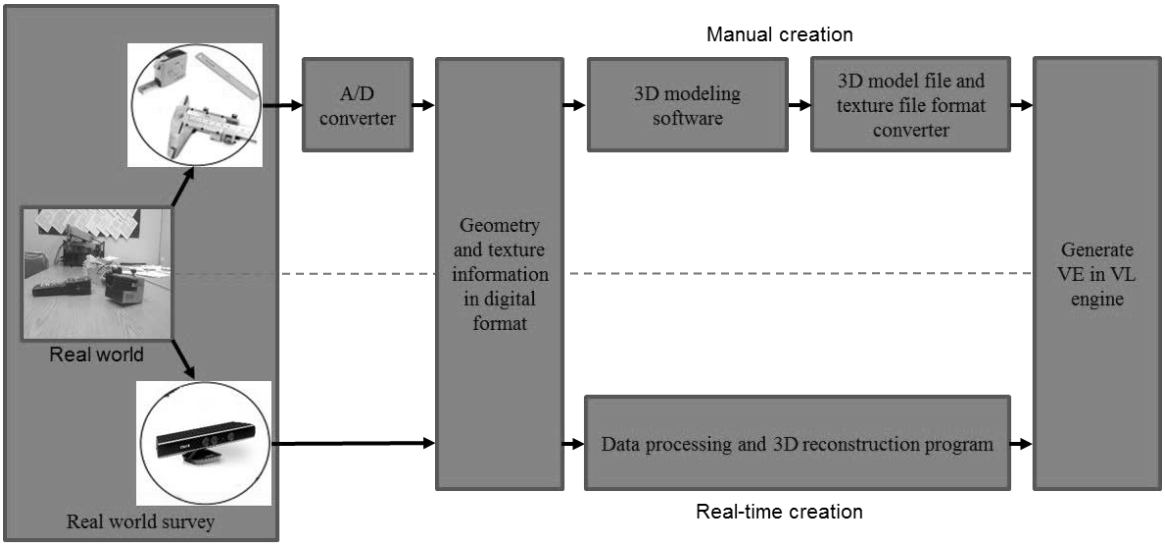

Figure 1: Popular methods for creating virtual environments.

Remote Virtual Proctor for VLs

Characteristics of Remote Virtual Proctor

In education, a proctor is a supervisor or invigilator who supervises an examination. In order to facilitate online education, remote proctoring provided by real people appeared (e.g. ProctorU [24]), but this approach is costly for the students since such services currently usually charge over $60 per student per course. With the development of the high-speed Internet and advanced computer peripherals, virtual proctors were introduced. Virtual proctors employ biometric technologies in order to identify the learners, to monitor their actions and to validate the test results without a need for real people. Recently, virtual proctoring has become one of the most effective and economical alternatives to traditional in-person proctoring or special online proctoring with real people. The advantages of virtual proctoring are: (i) Low fixed cost. The students only need to install the software and set up a webcam. Then, the software can be used repeatedly for both authentication and proctoring. The average cost for software and hardware is commonly less than $15. (ii) Convenience. It is not necessary to make an appointment with a real proctor, and thus the test can be taken at any time and from anywhere. (iii) Accurate authentication. The utilization of biometric technologies can provide accurate recognition of the participants, thus ensuring a reliable authentication. [20] Table 1 compares three kinds of proctors with respect to fixed cost (cost), operational accuracy (accuracy), difficulty of making an appointment (accessibility) and flexibility in arranging a test time (flexibility).

Design of virtual proctor with facial recognition

Advantages of facial recognition over other biometric techniques for virtual proctor development

A virtual proctor has the same functions as a local or in-person proctor. This means that a virtual proctor should administer a test and fulfill the responsibilities required by the specific test. In order to meet these requirements, a virtual proctor needs to verify the learners' identity by checking their biometric information (such as fingerprint [25], face snapshot [20], palm print [26], hand geometry [27], iris [28] and/or retina [29]) and then track their actions in real time. Among the biometrics methods, the facial recognition method is one of the preferred choices for the tasks of authentication and tracking. Firstly, the hardware for real-time facial recognition and tracking is simple and a common webcam suffices. Secondly, the sample frequency can be high when being compared with iris-recognition or retina-scanning methods because the features used for facial recognition are so notable that they can be identified very easily. Therefore, the algorithms for the facial matching are much simpler than other alternatives and the processing time for tracking the users is much shorter, thus allowing for a high sample frequency. Thirdly, facial recognition is macroscopic in scale while iris recognition and retina scanning are based on microscopic features, which renders them difficult to implement in virtual proctor applications. Hence, the virtual proctor presented here was designed based on facial recognition. The basic flow chart for the design process of the virtual proctor is depicted in Figure 2. All frames that are used to authenticate and monitor the user are sampled from a webcam located at the user's site. The user's face is detected and extracted repeatedly from a single sample frame at a specified frequency. Upon passing through the first-time matching, the user either successfully logs into the VL or fails to log in and is therefore forced to exit the VL. After passing through the authentication phase, an iteration follows. If the mismatching percentage of the facial features between the new stored template (i.e. the picture taken during the authentication phase containing the user's face and background information) and the real-time sample frame exceeds a pre-configured threshold value, a suspicious behavior is identified and a video clip is recorded, which is then used for further verification.

Table 1: Comparison of local proctor, online proctor and virtual proctor.

Type of proctoring	Cost	Accuracy	Accessibility	Flexibility
Local	High	High	Low	Low
Online by real people	Medium	Medium	Medium	Medium
Virtual	Low	Low	High	High

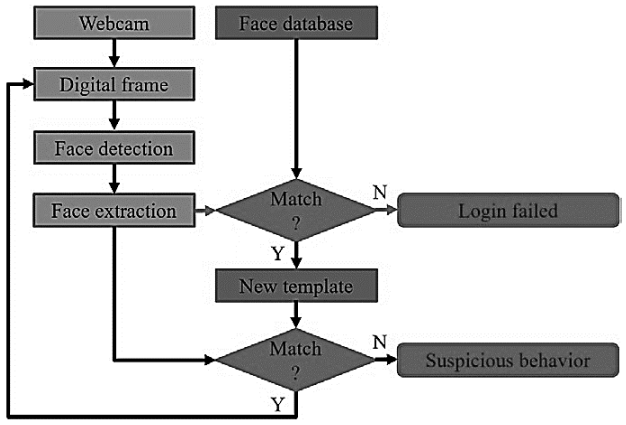

Figure 2: Flow chart for virtual proctor.

Facial Detection

Facial detection is a computer technology that has been widely employed in a variety of fields. This technology is used to identify human faces in digital images. [30] Facial detection represents a form of object detection. While object detection in general is the process of finding, separating and recognizing the objects in an image, facial detection only focuses on the task of separating the faces from the background of the images. Thus, it is easier to implement. At present, facial detection algorithms are mainly used to detect human faces in frontal view. The corresponding algorithms originated from image processing. The human faces in the image are matched bit by bit with the features of faces stored in a database. Finally, the faces are detected. Facial detection is the first step of facial recognition. Among the facial recognition algorithms, genetic algorithms [31,32] and eigenface algorithms (combined with Haar-like features [33]) are very popular. Since genetic algorithms are too time consuming to be implemented in real-time tracking, algorithms based on eigenfaces and Haar-like features were used in the work presented here to realize facial recognition and facial tracking.

Before performing the facial recognition, the faces in the images must be detected and extracted. With the development of object detection algorithms, a rapid object detection algorithm with a cascade of boosted classifiers based on Haar-like features was introduced previously. [34] This method improves the speed of facial detection while keeping the accuracy of the facial detection when being compared with the pure application of Haar-like features. [35] Therefore, the facial detection using Haar cascade classifiers was selected here.

Haar feature-based cascade classifier methods are machine learning approaches that are effective in object detection. [34] In the work presented here, the cascade function employed in this method was trained based on a large set of positive images (with faces) and negative images (without faces). After extensive training work, this classifier was used to detect faces in other images. In this method, each feature is assigned a value based on the Haar wavelet. This value can be obtained by subtracting the sum of pixels in the white area from the sum of pixels in the black area as shown in Figure 3. Three kinds of common Haar features [35,36] are shown in Figure 3. Although the total number of these features was huge, most of them are irrelevant. As shown in Figure 4, the two features in the top row are two different independent features. The left picture of the top row takes advantage of the property that the eyes were darker than both the nose and the cheeks under the ambient light condition. The right picture used the property that the eyes were darker than the bridge of the nose. These two features for eyes and nose are irrelevant. Thus, the selection of the best features from a huge pool must be resolved. Here, the Adaboost (adaptive boosting) algorithm was employed to find the best threshold for the Haar features. Adaboost is a machine learning meta-algorithm. [37] With this booster algorithm, all of the irrelevant features should be applied to all training images, but instead of applying all of the more than 6,000 features to all regions of the training images, the features were separated into many groups. In turn, these groups were then applied one-by-one to the different regions of the images as shown in Figure 5. If a rectangular region failed at the first stage, it was discarded and the remaining features were not applied to it. If it passed, the second group of features was applied and the process was continued. Any region that passed all stages was classified as a facial region. Subsequently, a set of weak classifiers was generated. Finally, all of the obtained weak classifiers were added up, and the sum of this operation was taken as the final classifier.

In order to save development time, the pre-trained classifier from OpenCV [38] was used to implement the facial detection. There are many pre-trained classifiers, for example for faces, for eyes and for smiles. These classifiers are stored in XML format. [36] Here, the classifier haarcascade_frontalface_alt2.xml [39] was used to detect the faces.

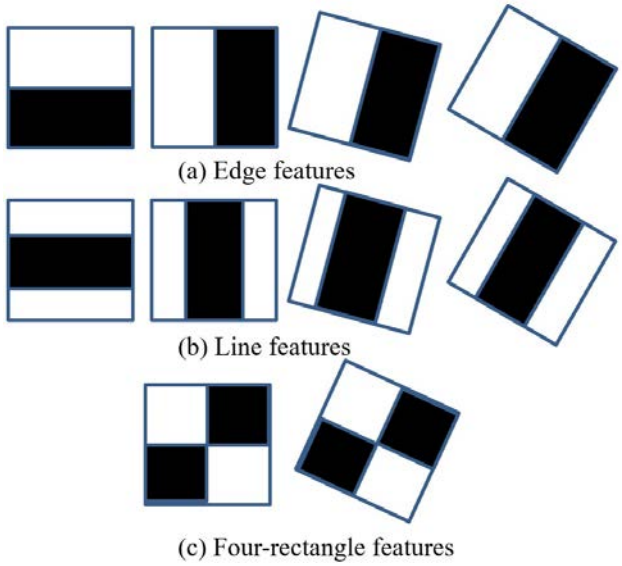

Figure 3: Common Haar features.

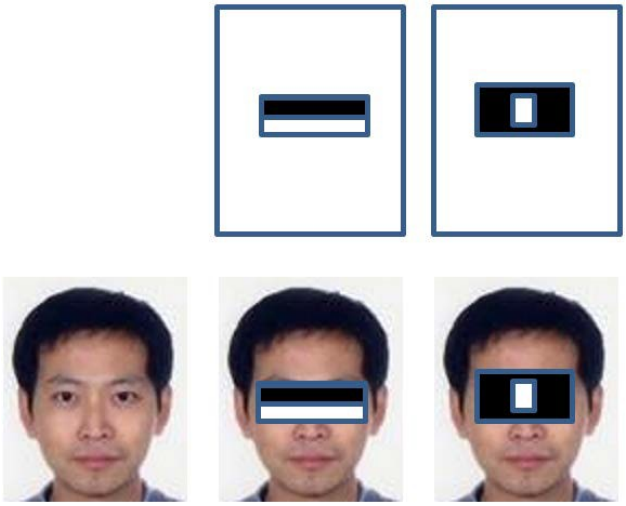

Figure 4: Irrelevant characteristics of features.

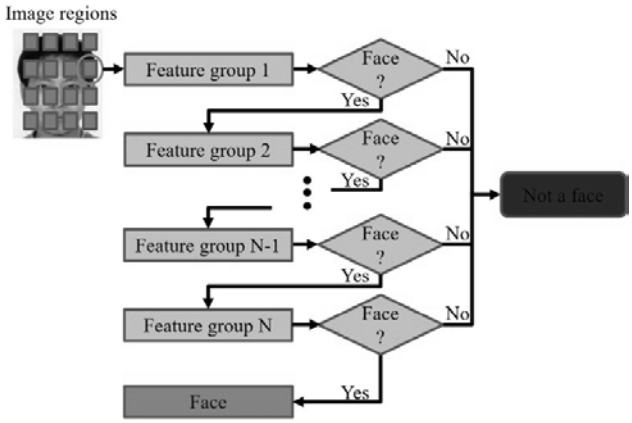

Figure 5: Face region extraction.

First, the required XML classifier was loaded. Then, the input image extracted from real-time video was loaded. The part of the code that was used to detect the faces is listed in Figure 6.

Facial recognition and face tracking in virtual proctor

After the faces have been detected, face recognition algorithms are used to identify the faces (i.e. the users of the VL system). Facial recognition is an extremely difficult task for computers, even though it is an easy task for humans. One of the first automated facial recognition systems was described elsewhere. [40] In this paper, keypoints at the locations of the eyes, ears and nose were marked and used to build a feature vector. The recognition was performed by computing the Euclidean distance between the feature vectors of a probe and the reference image. This method is robust for changing illumination conditions, but the marker process is

```
image = imread(filename, CV_LOAD_IMAGE_COLOR);
        if (image.empty())
          cout << "fail to read"<<i<<".jpg" << endl;
    // Load Face cascade (.xml file)
    CascadeClassifier face_cascade;
    face_cascade.load("haarcascade_frontalface_alt2.xml");
  // Detect faces
  std::vector<Rect> faces;
  face_cascade.detectMultiScale(image,faces, 1.1, 2, 0 | CV_HAAR_SCALE_IMAGE, Size(30, 30));
```

Figure 6: Sample code for facial detection.

too complicated. Facial recognition through geometric features was introduced previously. [41] In the work presented here, a 22-dimensional feature vector on large datasets was used to realize the facial recognition.

However, the geometrical features alone may not always carry enough information for facial recognition.

In order to recognize faces efficiently, eigenfaces were employed in the work presented here. [33] The basic idea of this method is to describe a high-dimensional dataset by correlated variables. Hence, only a few meaningful dimensions account for most of the information. The principal component analysis (PCA) method finds the directions with the greatest variance in the data, called principal components.

The algorithm can be represented as follows [42]:

Let $X = \{x_1, x_2, \ldots, x_n\}$ be a random vector, x_i the observations and $x_i \in R^d$

Then, the expected value of the observations is:

$$\mu = \frac{1}{n}\sum_1^n x_i \tag{1}$$

The covariance matrix S is:

$$S = \frac{1}{n}\sum_{i=1}^n (x_i - \mu)(x_i - \mu)^T \tag{2}$$

The eigenvalues λ_i and eigenvectors v_i of S are:

$$Sv_i = \lambda_i v_i, \qquad i = 1, 2, \ldots, n \tag{3}$$

The eigenvectors are labelled in descending order. The *k* principal components are the eigenvectors corresponding to the *k* largest eigenvalues. The *k* principal components of the observed vector *x* are then given by:

$$y = W^T(x - \mu) \tag{4}$$

where $W = (v_1, v_2, \ldots, v_k)$.

The reconstruction from the PCA basis is given by:

$$x = Wy + \mu \tag{5}$$

Employing the eigenface method, facial recognition was subsequently performed using the following steps: (i) projecting all training samples into the PCA subspace, (ii) projecting the query image into the PCA subspace, and (iii) finding the nearest neighbor between the projected training images and the projected query image.

Once the facial recognition has been completed, the users can log into the VL system by scanning their faces. Then, during the following test, the users' faces are tracked and recognized in real time.

Validation and Evaluation by Integrating Virtual Proctor with a Virtual Laboratory

A VL with a gear train experiment used in an undergraduate mechanical engineering course was selected as the pilot prototype for validating the virtual proctor. Before demonstrating the validity of the implementation, suspicious behaviors were defined first. In order to explain this, a coordinate system was defined as shown in Figure 7. The behavior of "rotating head" corresponds to a rotation about either the X axis or the Y axis. The behavior of "moving relative to webcam" corresponds to a translation along the X, Y and Z directions.

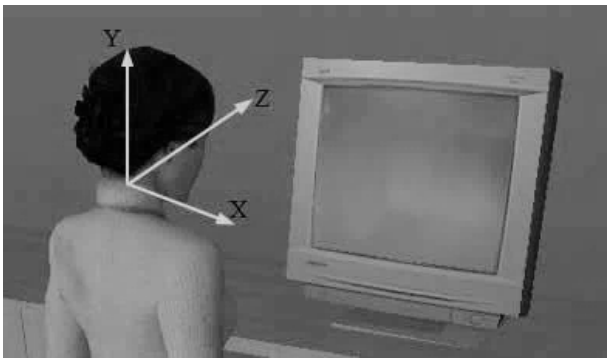

Figure 7: Explanation of suspicious behaviors.

In Figure 8, the planetary gear train experiment with the virtual proctor is depicted. First, a student logs into the VL system with his/her name. Then, the webcam scans the student's face for authentication purposes. If the authentication failed, this student is forced to exit the VL system. If the authentication was successful, the scanned frame is stored as the new template in the subsequent tracking and proctoring stage. Finally, the experiment is conducted under the supervision of the virtual proctor.

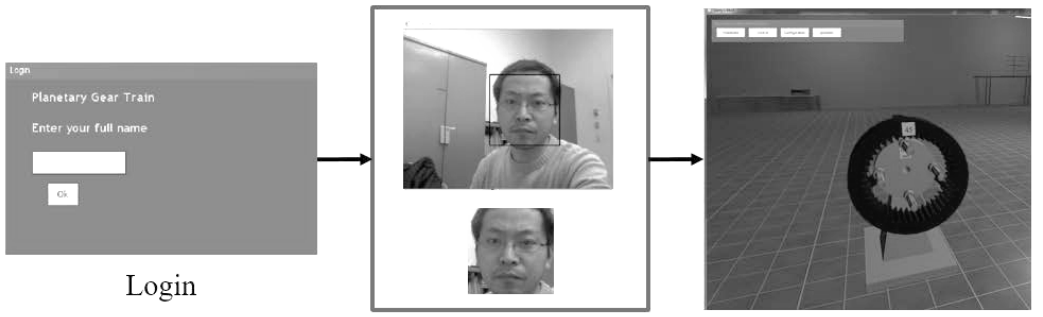

Figure 8: Implementation of virtual proctor in VL.

In order to realize real-time tracking, the sampling frequency of the videos was set to 15 frames per second. In addition, any rotations or translations are relative to the reference template captured during authentication (refer to 'New template' in Figure 2). In order to test the accuracy of correctly identifying cheating behaviors, 50 cheating attempts were acted out before putting the virtual proctor into practice. The test results corresponding to different criteria are listed in Table 2.

In the planetary gear train experiment, six questions were posed to the students. In a pilot study involving 30 students, the students answered all of these questions individually during the experiment. The student receives 1 point for each correct answer. Therefore, the maximum score is six and the lowest possible score is zero. The histogram of the scores obtained by the participants is shown in Figure 9. There was no cheating behavior found in the pilot study.

In order to obtain the students' attitudes about this experiment, an anonymous survey was administered. In this survey, 5 rating levels were set: excellent, very good, good, fair and poor. 27 of the 30 participating students provided feedback. A distribution pie chart is given in Figure 10. The conclusion that can be drawn is that the VL system with virtual proctor works well, even though it is not perfect. More details about the gear train experiments implemented using the VL system can be found elsewhere. [43]

Conclusions and Future Work

In this paper, a VL system with a virtual proctor was introduced. In order to test and evaluate the proposed VL system, a prototype in the form of a planetary gear train experiment was implemented and piloted. This VL system with virtual proctor was found to work well, and thus the virtual proctor represents a viable alternative to human proctors. In addition, a tool for detecting suspicious behaviors was implemented based on the matching percentage of features between a stored frame which is taken as a template and a real-time sampled frame. It was concluded that the virtual proctor can provide high accuracy in detecting suspicious behavior if the threshold value of the matching percentage is configured appropriately.

At present, the method introduced here was only used to identify and track the users' faces. The definition for suspicious behaviors was also fairly simple (including only frontal views of faces). Therefore, future work on the virtual proctor will focus on expanding the categories that can be monitored and tracked, such as keystrokes, sounds, gestures and the users' computer screens.

Although the virtual proctor is not perfect in its current form, it still represents a new feature for VL systems. It holds certain promise, especially if more advanced algorithms and biometric technologies are introduced.

Table 2: Test results corresponding to different criteria based on 50 cheating attempts.

Criteria (rotation Rθ in degrees and translation Td in millimeters)	Rθ ≤ |3| Td ≤ |10 |	Rθ ≤ |5| Td ≤ |20 |	Rθ ≤ |10| Td ≤ |40|	Rθ ≥ |10| any translation
Accuracy	100%	≤ 91%	≤ 63%	≤ 30%

Note: No cheating attempt is recorded if both the rotation and translation values are less than their respective thresholds.

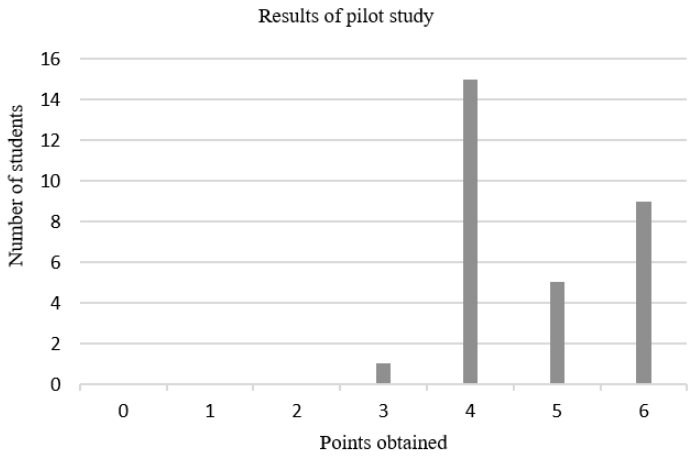

Figure 9: Results of pilot study.

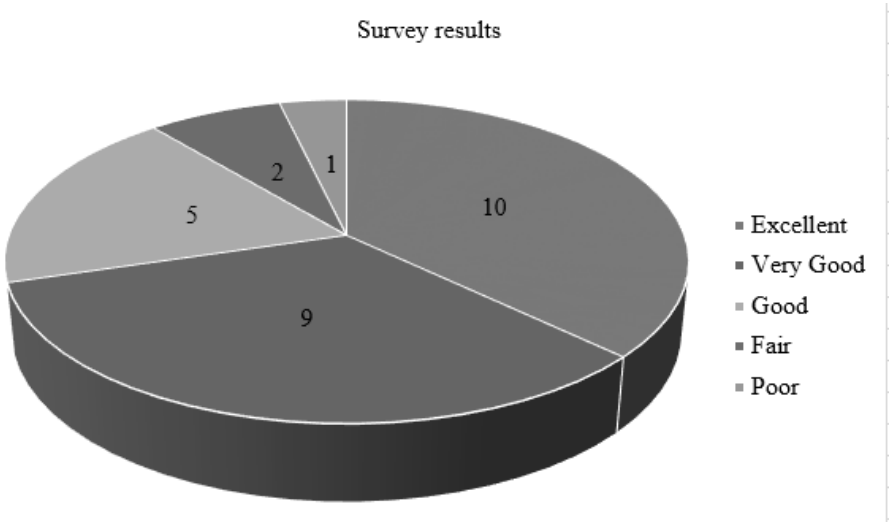

Figure 10: Survey results.

References

1. Steuer, J., Biocca, F. & Levy, M. R., 1955, "Defining virtual reality: Dimensions determining telepresence", Communication in the Age of Virtual Reality, pp. 33-56.

2. Jayaram, S., Connacher, H. I. & Lyons, K. W., 1997. "Virtual assembly using virtual reality techniques", Computer-Aided Design, Vol. 29, No. 8, pp. 575-584.

3. Zhang, Z., Zhang, M., Tumkor, S., Chang, Y., Esche, S. K. & Chassapis, C., 2015, "Real-time 3D reconstruction for facilitating the development of game-based virtual laboratories", Computers in Education Journal, Vol. 7, No. 1, pp. 85-99.

4. Luck, M. & Aylett, R., 2000, "Applying artificial intelligence to virtual reality: Intelligent virtual environments", Applied Artificial Intelligence, Vol. 14, No. 1, pp. 3-32.

5. De Mauro, A., 2011, "Virtual reality based rehabilitation and game technology", EICS4Med 2011, Vol. 1, pp. 48-52.

6. http://www.wecanchange.com/elementary-school/resources/virtual-labs, accessed in January 2016.

7. Weimer, J., Xu, Y., Fischione, C., Johansson, K. H., Ljungberg, P., Donovan, C. & Fahlén, L. E., 2012, "A virtual laboratory for micro-grid information and communication infrastructures", Proceeding of the 3rd IEEE PES International Conference and Exhibition, October 14-17, 2012, Berlin, Germany, pp. 1-6.

8. Aziz, E.-S., Chang, Y., Esche, S. K. & Chassapis, C., 2013, "A multi-user virtual laboratory environment for gear train design", Computer Applications in Engineering Education, Vol. 22, No. 4, pp. 788-802.

9. Barham, W., Preston, J. & Werner, J., 2012, "Using a virtual gaming environment in strength of materials laboratory", Proceedings of Computing in Civil Engineering, June 17-20, 2012, Clearwater Beach, FL, USA, pp. 105-112.

10. http://www.virtualgamelab.com, accessed in January 2016.

11. Baba, S. A., Hussain, H. & Embi, Z. C., 2007, "An overview of parameters of game engine", IEEE Multidisciplinary Engineering Education Magazine, Vol. 2, No. 3, pp. 10-12.

12. Thorn, A., 2010, "Game Engine Design and Implementation", Chapter 1, 1st Ed., Jones & Bartlett Publishers.

13. Zhang, Z., Zhang, M., Tumkor, S., Chang, Y., Esche, S. K. & Chassapis, C., 2013, "Integration of physical devices into game-based virtual reality", International Journal of Online Engineering, Vol. 9, No. 5, pp. 25-38.

14. Chang, Y., Aziz, E. S., Esche, S. K. & Chassapis, C., 2011, "A game-based laboratory for gear design", Proceedings of the 118th Annual Conference & Explosion of ASEE, Vancouver, Canada, June 26-29, 2011.

15. Zhang, Z., Zhang, M., Aziz, E.-S., Esche, S. K. & Chassapis, C., 2013, "Real-time 3D model reconstruction and interaction using Kinect for a game-based virtual laboratory", Proceedings of the ASME 2013 International Mechanical Engineering Congress & Exposition, San Diego, USA, November 15-21, 2013.

16. Zhang, Z., Zhang, M., Chang, Y., Esche, S. K. & Chassapis, C., 2014, "An efficient method for creating virtual spaces for virtual reality", Proceedings of the 2014 ASME International Mechanical Engineering Congress and Exposition, Montreal, Canada, November 14-20, 2014.

17. Mouragnon, E., Lhuillier, M., Dhome, M., Dekeyser, F. & Sayd, P., 2006, "Real time localization and 3D reconstruction", Proceedings of the 2006 IEEE Conference on Computer Vision and Pattern Recognition, New York, USA, June 17-22, 2006, Vol. 1, pp. 363-370.

18. https://www.utoledo.edu/business/InfoTech/ITLabs.html, accessed in January, 2016.

19. http://www.scholarsqatar.com/virtual-lab/, accessed in January, 2016.

20. http://www.biomids.com/proctoring/, accessed in January, 2016.

21. http://www.proctortrack.com/, accessed in January, 2016.

22. http://proctorfree.com/exam-proctoring, accessed in January, 2016.

23. http://remoteproctor.com/rpinstall/orgselector/orgselector.aspx, accessed in January, 2016.

24. http://www.proctoru.com/, accessed in January, 2016.

25. http://www.softwaresecure.com/remote-proctor-pro-faq/, accessed in January, 2016.

26. Rasmussen, K. B., Roeschlin, M., Martinovic, I. & Tsudik, G., 2014, "Authentication using pulse-response biometrics", Proceeding of Network and Distributed System Security (NDSS) Symposium 2014, San Diego, California, USA, February 23-25, 2014.

27. Bača, M., Grd, P. & Fotak, T., 2012, "Basic principles and trends in hand geometry and hand shape biometrics", INTECH, November 28, 2012.

28. http://proctorfree.com/blog/the-future-is-now-iris-recognition-technology-makes-id-cards-redundant, accessed in January, 2016.

29. Proctor, R. W., Lien, M. C., Salvendy, G. & Schultz, E. E., 2000, "A task analysis of usability in third-party authentication", Information Security Bulletin, Vol. 5, No. 3, pp. 49-56.

30. https://facedetection.com/, accessed in January, 2016.

31. Panigrahy, M. P. & Kumar, N., 2012, "Face recognition using genetic algorithm and neural networks", International Journal of Computer Applications, Vol. 55, No. 4, pp. 8-12.

32. Hjelmås, E. & Low, B. K., 2001, "Face detection: A survey", Computer Vision and Image Understanding, Vol. 83, No. 3, pp. 236-274.

33. Menezes, P., Barreto, J. C. & Dias, J., 2004, "Face tracking based on Haar-like features and eigenfaces", Proceedings of IFAC/EURON Symposium on Intelligent Autonomous Vehicles, Técnico, Lisboa, Portugal, July 5-7, 2004.

34. Viola, P. & Jones, M., 2001, "Rapid object detection using a boosted cascade of simple features", Proceedings of the 2001 IEEE Computer Society Conference, Kauai, Hawaii, December 8-14, 2001, Vol. 1, pp. I-511-520.

35. Wilson, P. I. & Fernandez, J., 2006, "Facial feature detection using Haar classifiers", Journal of Computing Sciences in Colleges, Vol. 21, No. 4, pp. 127-133.

36. http://docs.opencv.org/master/d7/d8b/tutorial_py_face_detection.html#gsc.tab=0, accessed in January, 2016.

37. Freund, Y., Schapire, R. & Abe, N., 1999, "A short introduction to boosting", Journal of Japanese Society for Artificial Intelligence, Vol. 14, No. 5, pp. 771-780.

38. http://opencv.org/, accessed in March, 2016.

39. https://github.com/Itseez/opencv/blob/master/data/haarcascades/haarcascade_frontalface_alt2.xml, accessed in March, 2016.

40. Kanade, T., 1973, "Picture processing system by computer complex and recognition of human faces", Doctoral Dissertation, Kyoto University, 3952, pp. 83-97.

41. Brunelli, R. & Poggio, T., 1992, "Face recognition through geometrical features", Proceedings of Computer Vision - ECCV'92, Second European Conference on Computer Vision, Santa Margherita Ligure, Italy, May 19-22, 1992, pp. 792-800.

42. http://docs.opencv.org/2.4/modules/contrib/doc/facerec/facerec_tutorial.html#id3, accessed in January, 2016.

43. Chang, Y., Aziz, E.-S., Zhang, Z., Zhang, M., Esche, S. K. & Chassapis, C., 2016, “Usability evaluation of a virtual educational laboratory platform”, Computers in Education Journal, Vol. 7, No. 1, pp. 24-36.