Design of an Interactive Training Simulator

M. Fikret Ercan^{a,*}, Dorothy Tan^a, Allen Liu^a, Yasushi Amari^b

^a Singapore Polytechnic, 500 Dover Rd, S139651, Singapore

^b National Institute of Technology, Nagano College, 716 Tokuma, Nagano city, Nagano Prefecture 381-8550, Japan

*Corresponding Author: mfercan@sp.edu.sg

Abstract

Virtual Reality based training is becoming more and more popular because of its capabilities for immersive training experiences. Despite many advantages and capabilities, these systems have some shortcomings. A major shortcoming is the interaction with the environment. To experience immersive training environments the users use a head mounted display (HMD) which blocks all the external vision to produce a fully virtual immersive experience. The trainees can only see the virtual world but not real objects like controllers, buttons, steering wheel, pedals, or even their hands. This paper describes design of a virtual reality based training system where the users are able to see real and virtual reality images. The trainees are able to see the combination of virtual and physical elements via a head-mounted-display and camera unit. When a user looks towards the wheel, controllers, and other preselected objects, they are detected by the computer vision algorithms and blended with the virtual image.

Keywords: Computer vision, game design, object recognition, virtual reality.

1. Introduction

Driving simulators have been applied to the field of the personnel training where the use of the simulators permits in depth training experience by simulating dangerous situations without risks for men and machines ^(1,2). There are various driving simulators using game-based methods ^(3,4,5,6). However, these simulators are at high cost and they are difficult to be portable. In addition, the field of view for trainees is limited to the number of equipped monitors inside the simulator.

We consider using VR glasses which offer an immersive experience to the trainees. Witnessing the

demand of users and the growth of smart VR glasses, we leverage a new type of training along with other technologies such as image recognition, and dual rendering. The system is named DRIS (Dual Reality Interactive Simulator) for easy understanding though it can be classified as an Augmented Virtual (AR) system referring to Milgram's taxonomy ⁽⁷⁾. DRIS system supports the trainees to do the driving training including forklift, crane, and car, with immersive experience. DRIS is a platform that provides the image recognition for blending virtual reality and actual reality in the training session. It creates a new training driving experience for casual users as it allows them to perform the real-life tasks within the virtual environment and interact with the physical game controllers. The blending of real objects with virtual image is achieved by using a camera and detecting the selected objects in the video image using computer vision algorithms. Detecting objects in images is a vastly studied topic in literature and there are many well-known algorithms ^(8,9). In this study, we have experimented with many of them. However, the performance of recent object detection techniques based on deep learning and Convolutional Neural Networks (CNN) are superior to conventional methods. Therefore, we utilized them in the design of our renderer. There are various other techniques presented recently which enables better interaction with real environment while wearing VR headsets. Leap motion's hand gesture detection is one such example ⁽¹⁰⁾. However, this system requires its custom designed hardware. There are also studies based on depth cameras such as Kinect though setting and calibrating hardware makes it difficult to use by non-experts ⁽¹¹⁾.

Our approach to address the problem of high costs of game-based simulator is to combine the dual reality capabilities with a low-cost and easy-to-use drag-and-drop authoring system. The system provides the users with the ability to integrate dual reality into immersive virtual

scenarios. It also enables the users with an easy to use authoring tool for virtual environments that allow the users to create immersive training scenarios. With this set of tools domain experts can build scenarios and deploy them for training purposes. The training environment can be easily created by non-programmers using their expertise on the subject matter which makes the system very flexible when compared to others^(3,4,5,6). Furthermore, DRIS works with a low cost VR glasses⁽¹⁰⁾ making it easier and faster to perform the different steps of the training procedure. The trainees can achieve a broader view compared to traditional simulators which are limited by the number of monitors. This more efficient workflow reduces the cost and improves productivity.

2. DRIS SYSTEM

2.1 System Architecture

The DRIS system enables an expert to produce an interactive scenario without expertise in VR/AR or programming. Fig. 1 shows the block diagram of DRIS framework. Its modules are described as follows.

Game Object Assets: Game assets can be dragged and dropped into the game environment. Game assets include all the 2D and 3D art such as containers, signage, pipes, traffic lights, etc. Note that the library of game assets can be easily extended to any new training environment.

Fig.1. The block diagram of DRIS framework. The trainer creates the training scenarios. Through the learner GUI, the learner experience the scenario. The analytics engine provides the information to improve the training quality for both trainers and the trainees.

Game Object Controller: This module controls the behavior of game assets. For instance, if the trainee is driving a vehicle and collides with one specific game object asset, the penalty is given to the trainee. On the other hand, a trainee may be awarded points for performing a loading and unloading operation successfully.

Fig. 2. The layout of the scene editor designed for airport tow truck training application.

Trainer Interface: This module allows the expert to create the training scenarios. The user interface with menus of art assets supports drag-and-drop feature. On the trainer side, the trainers can select from a menu of game assets that fits their training purpose. A user drags and drops the game assets in order to create the virtual environment such as construction site, warehouse, and outdoor street. For each scenario, the trainer is able to select the theme and its corresponding game asset items. Once the scene is created it can be uploaded to the cloud as an xml file and it can then be downloaded to the computer of the user. The snapshots of creating a scenario and uploading to and retrieving from cloud are shown in Figure 2. Many training scenarios can be developed and shared making the system flexible and expandable.

Analytics Engine: This module supports further analysis such as a trainee's performance. It also creates visualizations and reports in order to help the trainees to know their weaknesses and the errors so that it can be avoided in the future.

The system is developed with Unity and OpenCV software tools.

Fig. 3. The allowed actions and rules set by the user.

2.2 Training Application

Virtual environments is fast gaining popularity in training and developing of staff skills. This is due to the fact that the hardware is becoming more stable and accessible. In addition, virtual environments are able to provide a hands-on experience in a safer environment without posing any risks, losses or damages. Due to long collaboration of our institution with aerospace industry, aircraft towing application is developed based on our industry partners' request. Aircraft pushback operations are critical operations where mistakes can result in substantial losses from structural or aircraft damage. Skills development and confidence in the operators in such cases is preferable before stepping into the actual tow truck. Training can be carried out anytime instead of waiting of down-time or available aircrafts.

Currently, training is handled by skilled personnel. Training processes are complicated and existing simulators are bulky (see for instance ⁽¹³⁾). The trainees can train only at the fixed place, i.e., the simulator room typically with a fixed scenario. However, if the simulator is portable, the trainees can have more access to practice. To date, there is no portable and lightweight driving simulator.

The student interns, recruited for the project, conducted a research on airports and aircraft towing to list out all the possible digital assets needed such as buildings, planes, tow trucks etc. Some of these assets can be seen in Figures 2, 3 and 4. Apparently, producing such content requires the expertise of VR developers and development time. However, once the content is developed, a trainer is able to

manipulate and construct the training scene without having to re-develop the virtual environment.

Fig. 4. A training scenario created by the user and saved in the cloud.

3. Trainee Interface and Blending Virtual and Real Image

It is the main form of interaction between the users and the system. The interface supports the trainees to interact with in-game objects and physical devices such as game controllers. First, the trainees select the corresponding training scenario for the task they want to perform. The information of the training scenario is transferred to the trainee's interface. The information includes the map, the game objects and the instructions. Then the trainees use their head mounted display as the assistant for their work. The trainees use physical game controller to control in-game vehicle. However, they cannot see game controller since the head mounted display only shows the virtual environment. In our experiments, we use HTC VIVE which comes with front mounted cameras on VR glasses for the live stream of actual scene. A number of objects are chosen so that when these objects are detected from the camera image, the renderer captures region of interest from live camera stream and blends with the virtual image. This enables the trainees to interact with the real devices such as steering wheel, game controllers and so on without the need of taking off their VR glasses.

Image Recognition module is in charge of extracting the visual features of the physical object for image recognition. This module perform the matching between the current video frame and the picture of preselected objects. Currently, the preselected object used in the experiments is

the steering wheel that a user interacts to maneuver the vehicle. There is vast research and algorithms available for detecting objects in images. We have experimented with various methods including a Haar-cascade classifier ⁽¹⁴⁾ as well as key-point descriptors such as SURF ⁽⁹⁾ and BRIEF ⁽¹⁵⁾. Key-point descriptors have been a popular topic of research in the computer vision community. There is a growing need for faster and more memory efficient key-point descriptors, for applications such as image matching, panorama stitching, tracking, and object recognition. In our application, we need to replace the long feature vector of the image in the framework by choosing a suitable descriptor which can run in real time and maintain the robustness in image recognition.

In our initial experiments, we decided to detect steering wheel as the object of interest. Once the object is detected, a circular mask is used to blend virtual image with the detected area in the image. In our empirical studies, the best performance was obtained with Haar classifier. However, training the classifier is a cumbersome process. As the object size in the image varies depending on the users' head position layers of scanning is needed for detecting the object by resizing the search window. This slows down the detection process. On the other hand, SURF algorithm was fast and showed good performance however its performance is not stable in particular when there are changes in the illumination. These performance results are not satisfactory as we wanted to achieve a seamless blending of real and virtual images. Recently, deep learning algorithms become popular among the research community due to their astounding performance. Therefore, we considered applying deep learning algorithms in our system for the object detection problem. For the deep learning experiment, we employed mobile-net ⁽¹⁶⁾. The performance obtained was overwhelmingly better when compared to other conventional methods. Table 1 shows a comparison of recognition rate obtained by various algorithms in our experiments.

Table 1. A comparison of object detection algorithms for detecting steering wheel.

Haar	OBS	SURF	Deep learning
36%	18%	12%	99.7%

Figure 5 shows a trainee's view which incorporates virtual image and the detected object from the real world. As shown Figure 6, a trainee using DRIS system does not

need a special simulation facility. Training can be performed even at an office desk.

Fig.5. The user's perspective while controlling the tow truck. The steering wheel is detected by vision system and blended into the virtual environment.

Fig.6. A trainee is practicing the control of a tow truck driving on his desk.

In our future study, we intend to take head position of the user into account so that we can predict possible location of the object in the scene hence narrowing down the search process. In addition, we aim to integrate a hand detector so that a user will be able to see his/her hands.

4. Conclusions

This paper described design of virtual reality based training system. The system provides flexibility and portability making training available and cost-effective for various industries. It also tackles a major shortcoming of VR systems by blending virtual reality with real images making it more user friendly. The application from

aerospace industry aimed an effective use of the system for commercial training purpose. Our initial results show that DRIS system can provide an immersive experience to the user. In our future work, we aim to incorporate multiple object detection, in particular detection of a user's hands, to make simulation more immersive experience.

Acknowledgment

Authors thank Tam Nguyen and Jose Sepulveda for their contributions to this research work. This study is sponsored by Singapore Ministry of Education under the grant number MOE2015-TIF-2-T-039.

References

- (1) W. Kading: "The advanced Daimler-Benz driving simulator," SAE 9530012, 1995
- (2) H. Soma and K. Hiramatsu: "Driving simulator experiment on drivers' behaviour and effectiveness of danger warning against emergency braking of leading vehicle," Proc. of 16th ESV, Canada, 1998
- (3) Toyota Forklift Simulator. <http://www.grin.be/blog/?p=171>, retrieved 10th April 2016.
- (4) University of Valencia Forklift Simulator. <http://www.uv.es/uvweb/university-research-institute-robotics-information-communication-technologies/en/research-groups/lsym/projects/forklift-truck-simulator-forklift-1285895484292/ProjecteInves.html?id=1285898571581>, retrieved 20th May 2016
- (5) Smart Mover Ultimate Car Simulator. <http://thefutureofthings.com/5741-smart-mover-ultimate-car-simulator/>, retrieved May 2016
- (6) University of Queensland Driving Simulator. <http://www.carrsq.qut.edu.au/simulator/>, retrieved June 2016
- (7) P. Milgram and F. Kishino, "A Taxonomy of Mixed Reality Visual Displays", IEICE Trans. Information Systems, Vol. E77-D (12), pp. 1321-1329, 1994.
- (8) D. G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints", International Journal of Computer Vision, Vol. 60(2), pp. 91-110, 2004.
- (9) H. Bay, T. Tuytelaars and L. J. Van Gool, "SURF: Speeded Up Robust Features," Computer vision—ECCV, pp. 404-417, 2006.
- (10) <https://www.leapmotion.com/>, retrieved 1st Sept 2017.
- (11) C. Wang, Z. Liu and S. C. Chan, "Superpixel-Based Hand Gesture Recognition With Kinect Depth Camera," in IEEE Transactions on Multimedia, Vol. 17(1), pp. 29-39, 2015.
- (12) HTC VIVE. <https://www.vive.com/>, retrieved 30th August 2017.
- (13) L3 Aircraft pushback. <http://www.l-3training.com/applications/aviation/aircraft-pushback-training-simulator>, retrieved 25 July 2017
- (14) P. A. Viola, M. J. Jones, "Robust real-time face detection," International Journal of Computer Vision, Vol. 57(2), pp. 137-154, 2004.
- (15) M. Calonder, V. Lepetit, M. Özuysal, T. Trzcinski, C. Strecha, P. Fua: "BRIEF: Computing a local binary descriptor very fast," IEEE Trans. Pattern Anal. Mach. Intell., Vol. 34(7), pp. 1281-1298, 2012.
- (16) Andrew G. Howard, Menglong Zhu, Bo Chen, et al.: "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications." arXiv preprint arXiv:1704.04861, <https://arxiv.org/pdf/1704.04861.pdf>, retrieved 20 November 2017.