Adapting First-Person Shooter Video Game for Playing with Virtual Reality Headsets

Ilya Makarov,[1*] Oleg Konoplia,[2] Pavel Polyakov,[1] Maxim Martynov,[1]
Peter Zyuzin,[1] Olga Gerasimova,[1*] Valeria Bodishtianu[1]

[1] National Research University Higher School of Economics,
School of Data Analysis and Artificial Intelligence
iamakarov@hse.ru, {polyakovpavel96, martynov.maxim,
peter95zyuzin, olga.g3993, bodishtyanu.v} @gmail.com

[2] DigiPen Institute of Technology,
Redmond, WA, USA
olegkonoplya@gmail.com

Abstract

In this article a combination of two modern aspects of games development is considered: (i) the impact of high quality graphics and virtual reality (VR) user adaptation to believe in realness of in-game events by user's own eyes; (ii) modeling an enemy's behavior under automatic computer control, called BOT, which reacts similarly to human players. We consider a First-Person Shooter (FPS) game genre, which simulates an experience of combat actions. We describe some tricks to overcome simulator sicknesses in a shooter with respect to Oculus Rift and HTC Vive headsets. We created a BOT model that strongly reduces the conflict and uncertainty in matching human expectations. BOT passes VR game Alan Turing test with 80% threshold of believable human-like behavior.

1 Introduction

Virtual Reality is a technology that allows a person to interact with a three-dimensional computer-generated environment. That system simulates a person's physical presence and environment to allow user interaction. The success of the first Oculus Rift VR helmet mounted display lied in the technical specifications, such as low delay duration, wide-angle view, high resolution and quality of head movement tracking. Taken together, it gave the user unprecedented emotions from gameplay. However, in practice, the ability to move and rotate your head inside virtual reality is not enough to achieve full immersion effect, which is much more complicated (Muhanna 2015).

One of the contemporary problems of a low spread of all VR technologies is user's symptoms of discomfort when using virtual reality environments (VR Market 2016). In what follows we describe method of 3D scene deformation to reduce the effects of induced motion sickness. We have solved the task of inverse kinematics in HTC Vive, which allows moving in a bounded area. Besides, we have developed adaptive rotation algorithm allowing user to walk distant virtual paths being in a bounded room area.

We then focus on a First Person Shooter (FPS) game as a special genre of video games simulating combat actions with projectile-based weapons combined with an eyes perspective of a human-like model placed in virtual world. At the round start, each player is assigned to one of the competing teams and all the players are spawned at the team base point with starting weapon set and full health. Each player can move and kill other team players with weapons.

The origin of Artificial Intelligence research consists of the concept "intelligence" as computational part of the ability to reach the goals. Many games have computer-controlled agents playing against human players. The behavior of non-playable characters (NPC) is described in terms of the artificial intelligence (AI) in the game. AI provides important features of virtual reality, such as human-like behavior of NPCs and interaction with them. Unfortunately, modern game developers value graphics greater than the behavior of an enemy NPC, so called BOT, making gameplay unrealistic (Fabricatore 2007).

From the human point of view, it is not fair to access the game environment states that could not be obtained through first-person perspective of BOTs. In (Wand and Tan 2015) authors stated that users want to play against BOTs on equal conditions.

In order to satisfy expectation of the player for enemy level we focus on creating BOTs imitating human behavior in FPS game. We combined data analysis to all aspects of BOT's motion, targeting, command interaction, reasoning

* Corresponding Author

and knowledge discovery to support BOT with its human-like decision making system. We believe that combination of VR adaptation and game AI will lead to vast growth of the FPS market for VR (now, there are less than 10 FPSs for VR). Using Unreal Engine 4 we created our own FPS for VR, which then was evaluated through game Turing test of BOT humanness by humans' points of view.

2 Virtual Reality

2.1 Simulator Sickness and User Interface

However, wearing HMD cause discomfort with symptoms varying from eye-strain to nausea and motion sickness, caused by conflicts of visual and body senses. The best way to increase user's immersion in VR is to simplify User Interface (UI). Important game information should be integrated in VR, as in real world, otherwise binocular disparity may cause eye-strain (Chen et al. 2015). For people without game experience, special systems were designed (Roupe, Bosch-Sijtsema, and Johansson 2014).

All user's movements should be fully connected with the visual part. Eye tracking is one of the natural methods for a human to interact with machine. One can use eye-tracking to identify player intensions via face emotions (Vaidya, Jin, and Fellows 2014) or to operate UI by eye movement (Stengel et al. 2015). There are prototypes of devices (Homebrew Oculus Rift Eye Tracker 2014) combining VR with eye-tracker embedded into Oculus Rift outer shell.

2.2 3D Scene Visualization in HMD

There are special lenses in HMDs, creating a wide FOV, but causing a strong Pincushion Distortion of the image (Fig. 1a). For this purpose, game engine creates an inverse Barrel Distortion (Fig. 1b). On the scheme (Fig. 1c), variable x indicates the half of the real size of the helmet's display and x' is a virtual size of the display, which user sees through the lenses. It exceeds the real size resulting in FOV increasing. In fact, the HMD size is quite small. The side effect of this method is the Pincushion Distortion.

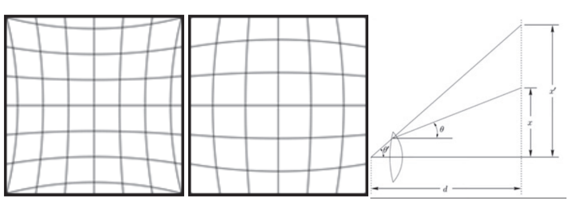

Figure 1. (a) Pincushion (b) Barrel (c) Geometry of Distortion

By using these two distortions combined, the result becomes undistorted. At the left part of the image (Fig. 2) is the scene rendered with a Barrel Distortion in advance, while at the right part we can see the undistorted result. The distortion can be computed on a GPU dividing pixels between each process (Barrel Distortion 2014).

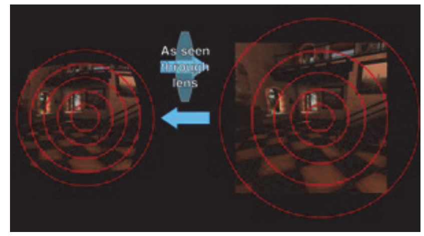

Figure 2. Combination of distortions

Oculus Rift Company listed in (Simulator Sickness 2016) general VR challenges: cut scenes, quick time events, system latency, screen door, motion blur. Controlling field of view can affect time delay before simulator sickness appears (Fernandes and Feiner 2016).

2.3 Camera Adaptive Rotation

One of the main problems of positional tracking is the size of moving area. A large virtual world is connected with the room bounded area for user motion in HMD.

We present FPS camera adaptive rotation algorithm that allows reducing the actual size of area to walk in VR HMD. Camera rotation is modified every tick of fixed time Δt. At tick n we take a current and previous tick head angle φ_n and φ_{n-1}, and calculate head angular velocity ω_n and acceleration a_n by Equations 1, 2:

$$\omega_n = \frac{\varphi_n - \varphi_{n-1}}{\Delta t} \quad (1) \qquad a_n = \frac{\omega_n - \omega_{n-1}}{\Delta t} \quad (2)$$

We take a_{max} as maximal user acceleration to calibrate velocity (system asks user to move fast his head changing his view) and compute by Equation 3 head acceleration parameter $\alpha(a_n)$ based on rotation until head's angular velocity reaches its maximum:

$$\alpha(a_n) = \begin{cases} 1 + \frac{a_n}{a_{max}}, if\ a_n \geq 0 \\ 1, \quad otherwise \end{cases} \quad (3)$$

Equation 4 is used to compute camera angular velocity ω'_n:

$$\omega'_n = \alpha(a_n) \cdot \omega_n \quad (4)$$

Equation 5 presents a derived from the first formula in Equation 1 new camera angle φ'_n:

$$\varphi'_n = \alpha(a_n) \cdot \varphi_n + \left(1 - \alpha(a_n)\right) \cdot \varphi_{n-1} \quad (5)$$

This algorithm does the linear transformation of camera yaw rotation input vector, and as a result, the virtual angle of rotation is slightly greater than the actual one (Figure 3).

During experiments on 20 small virtual mazes, the average area of the playing room reduces by four times in comparison to the same virtual area; however, the algorithm creates extra camera acceleration that sometimes can cause motion sickness faster than usual. Results show that motion sickness appears after 10 minutes of playing in average for 16 participants, comparing to 14 minutes mean result for experiment without using the algorithm.

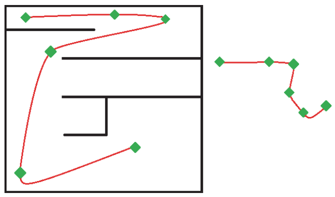

Figure 3. (a) Virtual map (b) Walking in a room

2.4 Inverse Kinematics

Inverse Kinematics allows copying actual player body movements to animate the character in the virtual world for a multiplayer game. Two hand-controllers of HTC Vive track user's hands motions, which then are transferring to wrist's joints in the skeletal mesh. We created animation controlling certain leg's joints without tracking from the VR sensors. Animation is based on inverse kinematics for user's head movements in a plane parallel to the ground. All three controllers create natural character movements and synchronizing it with actual user movements.

3 Game AI in First-Person Shooter

3.1 State-of-the-Art

General approaches to implement game AI are still finite-state machines (van Hoorn, Togelius, and Schmidhuber 2009) and rule-based systems (da Silva and Vasconcelos 2006). Developers often script NPC's behavior by certain game scenarios (Cole, Louis, and Miles 2004), but do not provide an interactive learning or a freedom of choice for BOT. Online learning system RETALIATE coordinating team tactics was presented in (Smith, Lee-Urban, and Munoz-Avila 2007); interesting results were obtained for imitation approach using Markov's chains (Tence et al. 2013) or genetic algorithms (Polceanu et al. 2016). A real-time behavioral learning video game NERO was presented (Stanley, Bryant, and Miikkulainen 2003). The contemporary results on evolution algorithms can be found in (McPartland and Gallagher 2011; Makarov, Tokmakov, and Tokmakova 2015).

3.2 Modeling BOT Behavior

We now briefly describe our model of the BOT based on SARSA algorithm of choosing between exploring, combating and retreating. Our idea was to implement self-learning BOT based on in-game rewards corresponding to BOT's actions and generating reinforcement learning policies. Each of the strategies was implemented via a self-organizing neural network for real-time reinforcement learning, called FALCON (Tan 2004).

Let us consider the targeting algorithm in details. The set of weapons that accessible to BOT includes rifle, shot-gun, machine-gun and knife, each having advantages and disadvantages in different game situations. Reward value is calculated using the following formula: $r = (a + b \cdot dist) \cdot damage$; where *dist* is a normalized value of the distance between the BOT and the enemy, and *damage* is normalized value of damage that the BOT inflicts to the enemy.

The experiments for one hundred of weapon usages show average successes over 72%. The comparison with original FALCON results in the following stats: Sniper Rifle was used more efficiently for long ranges with enemy's higher velocity; Shot Gun was used more optimal for short-range distances and with greater amount of reward increasing it by 50%. The example of weapon selection based on a modified FALCON showed its applicability to human-like learning in FPS games (Makarov et. al. 2016a).

In order to reproduce how real people aim in FPS games, we implemented aim-assisting techniques, such as Gravity, Sticky, and Lock targeting (Vicencio-Moreira et al. 2014) helped to make aiming process believable when shooting at a group of enemies. The resulting model combines adaptive ray-casting algorithms with several approaches of targeting resulting in human-like shooting described by "first seen – first shot".

For BOT navigation we choose composite Bezier curves and include visibility measure on Voronoi-based navigation mesh to make BOT trajectory smooth and human-like. The relative difference of the smooth low-visibility path and optimal path lengths does not exceed 10-12% (Makarov and Polyakov 2016).

3.3 Measuring AI Quality

Models performing intelligence tests have different purposes and applications (Hernández-Orallo et al. 2016). One of criterions to verify the quality of a game AI is the level of indistinguishability of computer-controlled and human players. We use the evaluation of such quality-based level of BOT humanness through Alan Turing test for game BOTs (Hingston 2009). In the competition, BOTs and humans take part in several rounds of FPS game, whereby the judges try to guess which enemies are humans (The 2K BotPrize 2012). In 2012 two teams have succeeded passing 50% human-like behavior barrier (Karpov, Schrum, and Miikkulainen 2012).

We use Turing test with 16 participants using VR HMDs to evaluate BOT AI quality in our FPS. All of human players have had at least 2 years' experience of playing FPS multiplayer games to judge correctly a behavioral aspect. Results show 80% judging accuracy computed as a ratio of correct judgments made by a judge to the total number of judgments (Makarov et al. 2016b). It became possible not only because of advanced AI but also due to increased immersion of human judges wearing VR HMDs.

4 Summary and Discussion

We made an overview of VR and eye-tracking technologies and proposed new methods of moving in virtual world under room constraints when wearing HTC Vive HMD.

Reward-based model for simulating human-like decision making in path planning, current goal selection and adaptive targeting as core FPS gameplay elements were considered. We were the first to verify BOT model through the VR game Alan Turing test (The 2K BotPrize 2012). It became possible due to advanced AI and increased immersion of human judges wearing VR HMDs. Combining adapted VR interface and game AI seems to be the reasonable approach of making FPS games friendly with VR and can increase a number of FPS video games for VR.

Aknowledgements

The reported study is to be funded by RFBR according to the research project proposal № 17-01-01534. The section 3.3 was prepared within the framework of the Basic Research Program at the National Research University Higher School of Economics (HSE) and supported within the framework of a subsidy by the Russian Academic Excellence Project '5-100'.

References

Chen, W., Chao, J.-G., Chen, X.-W., Wang, J.-K., Tan, C. 2015. Quantitative Orientation Preference and Susceptibility to Space Motion Sickness Simulated in a Virtual Reality Environment. *Brain Research Bulletin,* 113, 17-26

Cole, N., Louis, S.J., Miles, C. 2004. Using a Genetic Algorithm to Tune First-person Shooter BOTs. *CEC*, 1, 139-145

da Silva, F.S.C., Vasconcelos, W.W. 2006. Rule Schemata for Game Artificial Intelligence. *LNCS* 4140, 451-461

Fabricatore, C. 2007. Gameplay and Game Mechanics: A Key to Quality in Videogames*. Proceedings of OECD Expert Meeting on Videogames and Education*, 1-18

Hernández-Orallo, J., Martínez-Plumed, F., Schmid, U., Siebers, M., Dowe, D.L. 2016. Computer Models Solving Intelligence Test Problems: Progress and Implications, *AI*, 230, 74-107

Hingston, P. 2009. A Turing Test for Computer Game BOTs. *IEEE Transactions on CIAIG*, 1 (3), 169-186

Karpov, I.V., Schrum, J., Miikkulainen, R. 2012. Navigation via Playback of Human Traces. *Believable Bots,* 151-170

Makarov, I., Tokmakov, M., Tokmakova, L. 2015. Imitation of Human Behavior in 3D Shooter Game. In *Supplementary Proceedings of 4th IC AIST*, Vol. 1452 of CEUR WP, 64-77

Makarov, I., Polyakov, P. 2016. Smoothing Voronoi-based Path with Minimized Length and Visibility using Composite Bezier Curves. In *Supplementary Proceedings of 5th IC AIST*, Vol. 1710 of CEUR WP, 191-202

Makarov I., Zyuzin, P., Polyakov, P., Tokmakov, M., Gerasimova, O., Guschenko-Cheverda, I., Uriev, M. 2016a. Modelling Human-like Behavior through Reward-based Approach in a First-Person Shooter Game, in: *Proceedings of 3rd Workshop on EEML*, Moscow, Russia. Vol. 1627 of CEUR WP, 24-33.

Makarov I. et al. 2016b. First-Person Shooter Game for Virtual Reality Headset with Advanced Multi-Agent Intelligent System, *24th ACM IC on Multimedia*, Amsterdam, Netherlands, ACM

McPartland, M., Gallagher, M. 2011. Reinforcement Learning in First Person Shooter Games. *IEEE Trans. on CIAIG,* 3 (1), 43-56

Muhanna, A.M. 2015. Virtual Reality and the CAVE: Taxonomy, Interaction Challenges and Research Directions. *J. KSU – Computer and Information Sciences,* 27 (3), 344-361

Polceanu, M., García, A.M., Jiménez, J.L., Buche, C., Leiva, A.J.F. 2016. The Believability Gene in Virtual Bots. *29th FLAIRS IC*, Miami, United States, AAAI Publications, 346-349

Roupe, M., Bosch-Sijtsema, P., Johansson, M. 2014. Interactive Navigation Interface for Virtual Reality using the Human Body. Computers, Environment and Urban Systems, 43, 42-50

Smith, M., Lee-Urban, S., Munoz-Avila, H. 2007. RETALIATE: Learning Winning Policies in First-Person Shooter Games. *Proceedings of 22nd AAAI IC on AI*, 1801-1806

Stanley, K.O., Bryant, B.D., Miikkulainen, R. 2005. Real-time Neuroevolution in the NERO Video Game. *IEEE Transactions on Evolutionary Computation,* 9 (6), 653-668

Stengel, M., Grogorick, S., Eisemann, M., Eisemann, E., Magnor, M.A. 2015. An Affordable Solution for Binocular Eye Tracking and Calibration in Head-mounted Displays. *Proceedings of 23rd ACM IC on Multimedia,* 15-24

Tan, A.H. 2004. FALCON: a Fusion Architecture for Learning, COgnition, and Navigation. *IEEE IJC on NN,* 4, 3297-3302

Tence, F., Gaubert, L., Soler, J., De Loor, P., Buche, C. 2013. Chameleon: Online Learning for Believable Behaviors Based on Humans Imitation in Computer Games. *Comp. Anim. Virtual Worlds*, 24, 477–495

Vaidya, A.R., Jin, C., Fellows, L.K. 2014. Eye spy: The Predictive Value of Fixation Patterns in Detecting Subtle and Extreme Emotions from Faces. *Cognition*, 133 (2), 443-456

van Hoorn, N., Togelius, J., Schmidhuber, J. 2009. Hierarchical Controller Learning in a First-Person Shooter. *IEEE Symposium on CIG*, 294-301

Vicencio-Moreira, R., Mandryk, R.L., Gutwin, C., Bateman, S. 2014. The Effectiveness of Aim-assist Techniques in First-Person Shooter Games. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, NY, USA, 937-946

Wang, D., Tan, A.H. 2015. Creating Autonomous Adaptive Agents in a Real-time First-Person Shooter Computer Game. *IEEE Transactions on CIAIG*, 7 (2), 123-138

Barrel Distortion in the Oculus Rift. Accessed: 2014-01. https://www.youtube.com/watch?v=B7qrgrrHry0

Homebrew Oculus Rift Eye Tracker. Accessed: 2014-06. http://jdarpinian.blogspot.ru/2014/06/homebrew-oculus-rift-eye-tracker.html

Simulator Sickness. Accessed: 2016-01. https://developer.oculus.com/documentation/intro-vr/latest/concepts/bp_app_simulator_sickness/

The 2K BotPrize: Publications. Accessed: 2012-11. http://botprize.org/publications.html

VR Market - Global Forecast to 2022. Accessed: 2016-07. http://www.marketsandmarkets.com/Market-Reports/reality-applications-market-458.html