2011

# SIMPOL VR – A VIRTUAL REALITY LAW ENFORCEMENT TRAINING SIMULATOR

Stephania Loizidou Himona
*Frederick University*, com.ls@frederick.ac.cy

Efstathios Stavrakis
*Frederick University*, stathis@ucy.ac.cy

Andreas Loizides
*P.A. College*, a.loizides@faculty.pacollege.ac.cy

Andreas Savva
*University of Nicosia*, savva.a@unic.ac.cy

Yiorgos Chrysanthou
*University of Cyprus*, yiorgos@cs.ucy.ac.cy

# SIMPOL VR – A VIRTUAL REALITY LAW ENFORCEMENT TRAINING SIMULATOR

Loizidou Himona, Stephania, Frederick University, 7, Y. Frederickou Street, 1036 Nicosia, Cyprus, com.ls@frederick.ac.cy

Stavrakis, Efstathios, Frederick University, 7, Y. Frederickou Street, 1036 Nicosia, Cyprus, stathis@ucy.ac.cy

Loizides, Andreas, P.A. College, Corner of Phaneromenis and Kalvou Street, 6031 Larnaca, Cyprus, a.loizides@faculty.pacollege.ac.cy

Savva, Andreas, University of Nicosia, 46, Makedonitissas Avenue, 1700 Nicosia, Cyprus, savva.a@unic.ac.cy

Chrysanthou, Yiorgos, University of Cyprus, P.O.Box 20537, 1678 Nicosia, Cyprus, yiorgos@cs.ucy.ac.cy

## Abstract

*The paper reports the implementation of SIMPOL VR, a 3D Virtual Reality Training Simulator for the Cyprus Police Force. Central part of it is the investigation of the suitability and the applicability of Virtual Reality enhanced with realistic character animation in law enforcement training and in particular the Cyprus Police Emergency Response Unit.*

*The simulator's architecture brings together software components, comprising the software training platform, with advanced virtual reality and motion capture hardware. Computer graphics algorithms, particularly computer-generated and motion-captured character animation are used within the simulator. Facial animation, as well as modelling of the human body will also be integrated for simulating realistic 3D characters.*

*Keywords: virtual reality, training simulator, character animation.*

# 1 INTRODUCTION

Training simulators are widely used by law enforcement agencies since they allow for safely training personnel in dangerous situations. The majority of such systems are primarily video-based, while in recent years Virtual Reality simulators have started being sought as a viable and affordable alternative to their video-based counterparts.

In video-based training platforms, trainees are presented with a video stream of a pre-recorded training scenario. The simplest of these are designed for simulated firearms shooting, while more advanced systems offer more complex scenarios. For these systems multiple video branches might exist allowing the trainer to manually select subsequent behaviours according to the trainee's actions. In addition, some systems offer the ability to train multiple subjects as a team simultaneously.

Virtual Reality simulators on the other hand provide computer-generated 3D training environments that are both immersive and interactive. Training scenarios have to be dynamic and can have pre-programmed logic that allows them to respond to trainees actions instantly, without necessarily the need for a trainer to interfere. Building such a simulator from the ground up not only involves developing standard 3D software and hardware integration, but also requires algorithms that can infuse realistic locomotion on 3D characters. In this paper these software and hardware components making up a 3D projection-based VR training simulator for law enforcement agencies are presented, while simultaneously character animation techniques are chosen that can fulfil the need for believable human motion.

One of the major challenges of such training simulators is achieving realistic human character animation and modelling. To this end, an overview of the underlying algorithms that have been incorporated for enriching the simulator's characters is given. The goal is to develop new character animation techniques that allow the use of motion capture data together with physically-based animation to improve realism and interactivity of virtual 3D characters.

In the next section a review of important related work in realistic simulation of characters is presented. In Section 3 the user requirements of the VR training simulator are outlined. Section 4 describes the platform of the simulator, while Section 5 focuses on realistic character animation used in the simulator. Finally, Section 6 concludes and provides an outlook of future work in this project.

# 2 RELATED WORK

Emerging inexpensive hardware and advanced graphics software have enabled high quality training simulators for a wide range of applications, recently based on Virtual Reality [1]. Law enforcement training simulation, VR-based or not, has also been an active area of research [2] and development. A variety of video-based and VR-based training simulators are currently available on the market, such as the VirTra 300 LE by VirTra Systems[1] and the law enforcement driving simulators by FAAC Incorporated[2], however these are proprietary systems.

While realism has significantly improved in VR training simulators in terms of rendering quality, animated characters have typically a limited spectrum of motion they can perform. Real-time dynamically responsive characters that feature convincing human animation could greatly improve VR training simulators.

A number of different techniques have been developed for human figure animation, each with its own advantages but at the same time its own drawbacks [3]. This interesting field has innovated tremendously moving from the first-generation [4], i.e. purely geometric models where the motion was developed using kinematic techniques, to the second generation [5], i.e. physical-based models (direct versus inverse dynamics and hybrid systems), towards the development of the third generation [6], i.e. autonomous behaviour through learning and perception. Other categorisation involves the allowance of an extra sub-level between the 2nd and the 3rd generation, i.e. motion interpolation which is basically the combination of pre-existing motion data, e.g. through motion capture or even motion by example; so that new motions can be produced. Based on findings [3], using one model alone would not solve the specific problem of realism and it is believed that a combination of techniques from the different generations (levels) could enhance the creation of autonomous virtual characters capable to react realistically to their environment.

Combining, for example, motion capture and dynamic simulation is an emerging area under exploration, in order to retain the advantages of each while avoiding their disadvantages. Generally speaking, dynamics which is based on physics, offers great realism in controlling interactively the avatar(s), a mechanism which is achieved at the expense of complicated equations of motion with even more computationally expensive solutions [7]. On the other hand, motion capture data maintains the naturalness of the human movements from which it has been created for, however, the motion is often invariant and in essence with no interactivity, due to technical issues [8]. Even though the combination of these techniques seems promising [9, 10, 11], the major problem with such a combination lies on the smooth, believable transition from one to the other and perhaps back again; needless to mention that this scenario needs to be done automatically with the minimum restrictions and/or interventions possible.

# 3 USER REQUIREMENTS FOR LAW ENFORCEMENT VR TRAINING

Prior to specifying the platform's architecture, the main user requirements have been identified. This was partly driven by the need to reproduce the functionality of the existing video-based training

[1] http://virtra.com
[2] http://faac.com/index2.html

system currently used by the Cyprus Police Force (CPF). It has also been motivated by the additional functionality the law enforcement training officers would expect the new platform to feature, considering the capabilities and limitations of Virtual Reality itself.

The trainers at the CPF have been consulted on multiple occasions and the authors had the opportunity to see their video-based training platform firsthand. Table 1, lists the pros and cons of the existing system.

| **Pros** | **Cons** |
|---|---|
| Photorealistic content (filmed videos) | Tedious and expensive to create new content |
| Easy to play back | Limited authoring capabilities |
| Easy to use interface for non-technical staff | Limited control of training scenario in real-time |
| Logging of trainees’ performance and historical data | Platform unaware of most trainees’ parameters (e.g., location, etc.) |
| Replay of trainees’ sessions for assessment | Potential of plagiarism between trainees (same scenarios for all) |
| Low computational footprint | Habituation of trainees due to repetitive training |
| | Non-immersive training scenarios (2D viewing/physical separation) |
| | Expensive and proprietary system – cannot be modified by 3rd parties |
| | Unsuitable for simulating certain critical incidents (e.g. injury, death, etc.) |

**Table 1** - Pros and cons of video-based training platform used by the Cyprus Police Force

Taking into consideration the pros and cons of the video-based platform of Table 1, the following list of desired characteristics for the VR platform have been identified:

- Reuse of content (i.e. 3D models) to author new scenarios
- Authoring of new scenarios even by non-technical staff
- Fully controllable training scenarios
- Logging of trainees’ performance and historical data
- Replay of trainees’ sessions for assessment
- Dynamic scenario execution (easy to differentiate a scenario among different trainees)
- Immersive 3D stereoscopic viewing
- Full body motion tracking data of trainee
- Open-source, modular, expandable software platform
- Reduced overall cost, mainly due to content reusability
- Wireless and remotely controlled training capability
- Ability to simulate critical incidents (e.g. death, injury, etc.)
- OS platform-independent (e.g. could enable trainers to use mobile devices)

# 4 VR TRAINING SIMULATOR PLATFORM

The law enforcement training simulator, a prototype of which is presented here, is comprised of multiple interoperating software and hardware components. All system components and reasons of being selected are outlined below.

## 4.1 Hardware

The first implementation of the VR system comprises of a single-screen 3D stereoscopic projection system that uses two projectors with polarizing filters. A two-sensor magnetic motion tracker is used. One of the sensors tracks the orientation and position of the head of the trainee, while the second is used to track the orientation and position of a replica firearm, which the trainee holds. A haptic glove is used to identify whether the firearm has been triggered. The tracking space is limited to a semi-sphere of about 2m diameter, centred above the trainee and at a distance of around 1.5m from the projection screen.

In addition, a server is used to coordinate the training sessions, offering both data and communications between the rendering node used for the trainee's view and the computer used by the trainer to control the scenario. Due to the decoupling of the system and the software run on each of them, as described in the next section, all three systems use inexpensive hardware and communicate over the network. The rendering node is equipped with a commodity graphics card capable of rendering the scenarios in real-time.

## 4.2 Platform Software Design

The characteristics of the VR-based training platform have been considered together with the pros and cons of the video-based solution used by the police force in order to design the platform's architecture. This architecture enabled the integration and the support of the necessary hardware devices and set the foundation for implementing an extensible and reusable software platform.

The software design of the platform has been drafted according to the following desired characteristics:

*Usability* – The user interfaces should be easy to use, intuitive and fairly self-explanatory.
*Extensibility* – The platform software should allow for additional features integration.
*Modularity* – New features can be developed as modules that can be tested and integrated independently, thus enhancing extensibility.
*Compatibility* – The software architecture should provide an abstraction layer that separates the hardware specifics from the software modules of the platform.

To fulfil these software design requirements a client/server infrastructure has been chosen that utilises the Model-View-Controller (MVC) software design pattern in its implementation. In MVC the Model is responsible for maintaining the data used by the platform (e.g. training scenarios, 3D content, audio, user data, etc.). The Model abstracts the data store for the rest of the application enabling different storage devices to be used as required. For instance, the Model may allow storing the data in an external USB hard drive or on a remote internet server. Views within the architecture are responsible for requesting, accessing and using the data for the purpose of serving the user. Views are typically interactive user interfaces that the trainer and the trainee use. The key point about considering each interface as a View is that the underlying data is not changed and it is the individual View that is responsible for utilising the data in a specific way.

The platform has two distinct Views, the trainer's and the trainee's interface. The trainer's interface encapsulates the functionality to maintain the data (content) through commands send to the Model via the Controller, to maintain the trainee user database, to author and execute scenarios and train the officers. This View has read/write access to the data enabling trainers to perform a variety of tasks both offline and while training officers (e.g. send commands to the active training scenario to modify it dynamically).

On the other hand the trainee's user interface is a real-time stereoscopic 3D application that receives input from the trainee (e.g. position of the trainee, gun firing etc.) but has no ability to modify the data content apart from the data collected for the training session as part of the trainee's performance.

Finally, the Controller is a single component that handles all command traffic (events) between the Views as well as the Model. The Model and the Controller have been developed as a networked server application running on the same computer, while the Views are developed as network clients that can run from different machines. It must be noted that this design allows for all components of the system to run on the same computer without any change of the software.

Figure 1, shows the conceptual software architecture of the individual components and their interoperation.

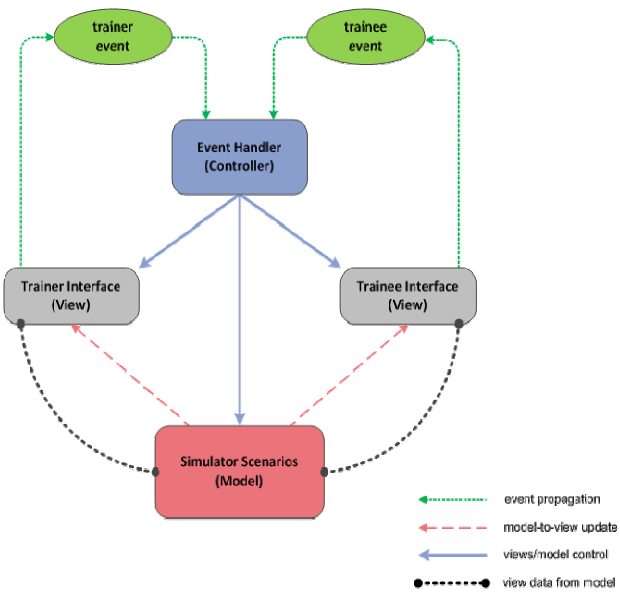


Figure 1 – Software Architecture of the VR Training Simulator

### 4.3 Platform Development and Testing

Based on the software architecture four different platform components have been implemented. Figure 2, shows the interaction of these components.

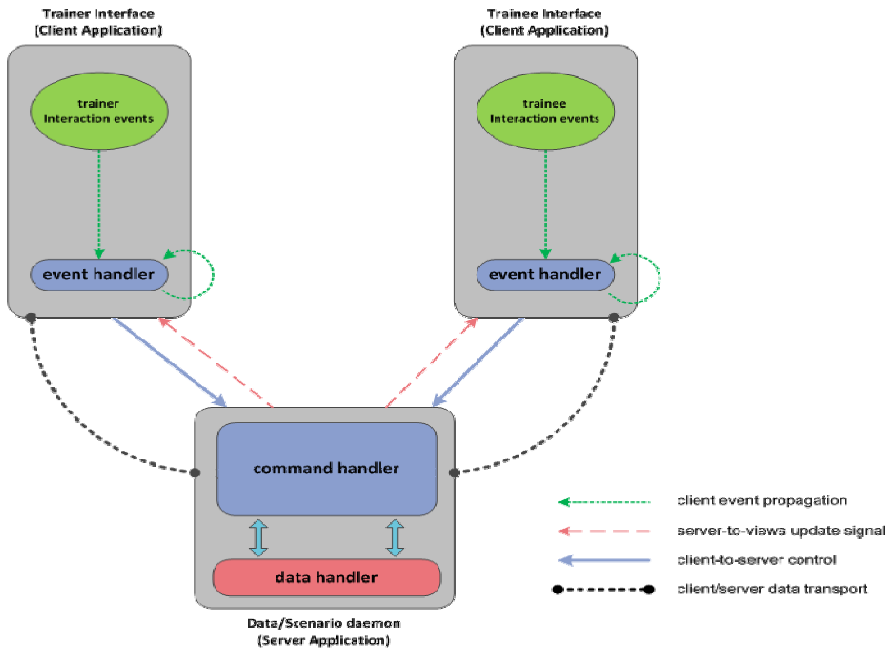


Figure 2 – VR training simulator software implementation details (Client/Server)

(a) Scenario Server (Hosting and handling training scenarios)

The scenario server implements the model of the software design. The scenario server has been implemented in the Python programming language and offers the ability to manage at a low-level the digital assets, such as 3D models, metadata descriptions (using XML) etc., that are useful for the training scenarios. The server utilises the UDP networking protocol and is accessible over a network port, offering asynchronous, multi-client communication, thus allowing several clients to connect and interact with it. The scenario server is a core system component that is aimed to be used by the other platform components, and therefore requires no direct user interaction; hence it has no graphical user interface and is the primary platform component to be launched in the training environment. The scenario server can run independent of the other system components and does not require to be restarted between training sessions, making it easy and efficient to use.

(b) Hardware Abstraction Layer (Low-level - managing hardware devices)

The hardware abstraction layer is composed of low-level components that are used by the other platform components to manage data between the diverse hardware used. One of the libraries handles the input from a haptic glove which provides the data for the triggering of the firearm of the trainee officers. Another is handling the positional and rotational data provided by the magnetic tracker, which is monitoring the hand of the trainee, thus providing information on where the officer is aiming at from the real to the virtual world. The tracker also takes the position and orientation of the head of the trainee, so that the virtual camera can be adjusted both for navigation and viewpoint tuning.

(c) Trainee Client (VR user interface software for trainees)
The Trainee Client is a component that has been built to satisfy the need for an interface between the platform and the trainee police officer. The Trainee Client implements one of the possible arbitrary views of the software architecture and has a number of distinct features that set it apart from the other views of the platform. The trainee client connects the real world, where the officers are being trained, with the 3D virtual worlds using an immersive 3D stereoscopic projection system. It tracks the position of the trainee officer using the magnetic tracking device and integrates interaction data from the haptic glove that the officers can wear in order to detect triggering of their replica training weapon. The client is a 3D application that executes training scenarios fetched from the scenario server and it is implemented in the Panda3D engine. Figure 3, shows a trainee using the prototype simulator. The hardware integration has been done by developing wrapper Python libraries that form the hardware abstraction layer.

Figure 3 – A trainee session on our prototype VR simulator

(d) Trainer Client (Desktop graphical user interface for trainers)
The Trainer Client is the other main component that has been developed to implement another view of the software architecture. This client enables the trainer officers to connect to the scenario server and execute commands on the training session, essentially manipulating the user space of the trainee officers. It can also be used to author scenarios and to monitor the performance and progress of the trainees. The trainer client is in general a component that can modify the data assets, inserting new content and removing old content, in contrast to the trainee client that can only read content and can only write data relevant to the performance of the trainee.

# 5 REALISTIC HUMAN CHARACTERS

Dynamics of motion, motion captured data as well as the combination of the two techniques are analysed in the following sections.

## 5.1 Motion Graphs technique (Motion Capture)

The use of the Motion Graphs, a technique originally introduced by Kovar in 2002 [12], for tackling the realistic motion of characters using motion capture data has been engaged. Despite its very wide adoption by the computer animation community there have not been available software implementations of the technique and therefore it has to be developed from scratch. Alongside with the core Motion Graphs software library that has been implemented in the C++ language, auxiliary libraries for parsing and handling motion capture data files of Biovision (BVH) file format have been created. This choice of supporting BVH files allows all the developed software to take advantage of the existing and freely available motion capture databases, such as the Carnegie Mellon University Motion Capture Database - CMU[3] which counts for over 2500 captured motions. It has also been planned that the implemented BVH software library will be able to support captured data from a newly acquired optical motion capture system.

In addition, a graphical application in C++ using OpenGL that allows simultaneous viewing and playback of multiple BVH motion files in real-time has been developed. This software allows for the application of the motion graphs algorithm on motions that the user loads and has been used for debugging and previewing the results. Figures 4(a) and (b) below, show sample outputs of the motion control application.

## 5.2 Physically-based animation (Dynamics)

To tackle character animation using dynamics of motion different physics engines such as Open Dynamic Engine, Havok and PhysX have been employed and tested. Eventually though, the Open Dynamic Engine[4] has been utilised. This is one of the standard libraries used by the community for simulating rigid body dynamics. To gain experience with the libraries, simple 3D applications where physics are applied to objects of our 3D environment have been built and experiments with physically-based rigid body animation took place.

---

[3] http://mocap.cs.cmu.edu
[4] http://www.ode.org

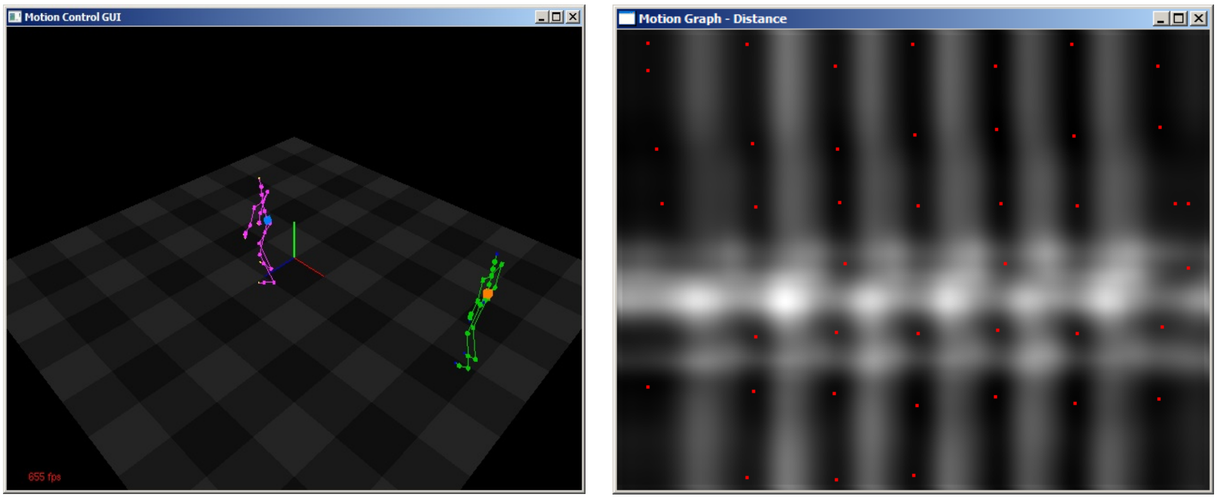

Figure 4 – On the left, (a) our character motion control 3D application, showing two characters with different motion captured sequences. On the right, (b) candidate transition points (marked with red) between the two motions shown in Figure 4(a), as calculated by the Motion Graphs algorithm.

### 5.3 Dynamics and Motion Capture Motions

The idea of combining motion capture data with dynamic equations of motion control is relatively new [13]. The objective of such a combination would be to capture the subtle details of realistic human motion while at the same time maintaining the physical realism of the movement. However, the main problem with such a combination lies on the smooth, automatic transition from one method to the other as well as identifying when it is more appropriate to use each method. The emphasis should be on bringing the real and the virtual worlds closer together by bridging their associated strengths. In particular, the motion is initiated from motion capture data using the motion graphs technique and whenever a collision or an abnormal situation emerges the dynamics take over.

Specifically, it has been identified that the motion capture data is best suited for commonly performed actions, e.g. walking, running, jumping etc. However, motion capture data are fixed pre-recorded sequences that cannot be directly used in arbitrary synthetic conditions, such as the introduction of obstacles in the 3D environment.

Physics, on the other hand, may be applied to the motion captured data in order to synthesise new motions that take into consideration novel or extreme conditions that did not exist in the recorded data. For instance, along the walk path of a 3D character when motion capture data is used an object may be introduced in real-time. Physically based simulation can assist in generating the impact with the injected object by applying forces to the object and the character.

Thus, physically-based animation is important in situations of simulating motions that are very difficult and sometimes even impossible to record, e.g. and impact of an object to a character that causes serious injury or death. The nature of the application in hand, training the police special forces, makes the employment of dynamics an actual necessity given the extreme scenarios that can occur.

In the developed platform, both of the aforementioned methods (motion captured data and dynamics) can co-exist and a solution to the problem of transitioning from one to the other will be the major next concern. Switching between the two methods is currently done manually.

# 6 CONCLUSIONS AND FUTURE WORK

In this paper the user requirements, hardware, software architecture and implementation of a VR law enforcement training simulator has been presented. In addition, we have outlined the use of data-driven and physically-based animation for realistic human character animation and we are currently investigating possible ways of combining the two into a hybrid approach to further increase the realism of human locomotion in training simulators.

This alloy of hardware, software and algorithmic components is envisioned to form in the future a replacement VR training platform for the Cyprus Police Force, where scenarios will be easy to author and police officers will be better trained in an immersive 3D environment.

In our current implementation we are utilizing publicly available motion capture data from the CMU library for character animation and a magnetic motion tracker that has only two sensors. We have recently acquired a 12-camera full body optical motion capture and tracking system to replace the limited two-sensor magnetic tracker used in the prototype, which will also allow us to create our own purpose-made motion data for the virtual characters.

It will also be investigated whether the Level Of Detail (LOD) of the virtual characters to be integrated into the platform is sufficient, and if not, a method such as subdivision splines [14], which can generate smooth surfaces defined by an arbitrary topology of grids, will be used to provide the required level of realism [15].

Given the current application, it is vital for virtual characters' emotions to be synthesised as realistically as possible, given that trainees will use the information provided to try and ”interpret” the suspects' emotions and subsequently take the ”required” action. Therefore, emotional facial expressions are also going to be added to the system based on mathematical modelled muscle deformations [16].

Designing, developing and integrating the core building blocks of this VR simulator have been the primary concerns of the work presented. Although, trainers have been consulted for understanding the features that should be included in the simulator, formal user experience and usability testing is necessary for verifying that the overall system fulfils the targeted training goal of police officers.

# 7 ACKNOWLEDGEMENTS

This project is funded by the Cyprus Research Promotion Foundation under contract ΤΠΕ/ΠΛΗΡΟ/0308(ΒΙΕ)/03. The motion capture data used in this project was obtained from mocap.cs.cmu.edu.

## References

1. Craig, A., Sherman, R.W. and Will, D.J. (2009). Developing Virtual Reality Applications: Foundations of Effective Design, Morgan Kaufmann.
2. Hormann, J.S. (1995). Virtual Reality: The Future of Law Enforcement Training, FBI Law Enforcement Bulletin, Vol. 64, Issue 7, pp7-12.
3. Loizidou Himona, S., Chrysanthou, Y. and Loizides, A. (2008). Human Figure Animation: A Historical Perspective, 22nd European Conference on Modelling and Simulation (ECMS), Cyprus, pp562-569.
4. Zeltzer, D. (1993). Knowledge-based animation, In ACM SIGGRAPH/SIGART, Workshop on Motion, pp187-192.
5. Duy Le, D., Boulic, R. and Thalmann, D. (2003). Integrating Age Attributes to Virtual Human Locomotion, International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Vol.XXXIV-5/W10.
6. Shao, W. and Terzopoulos, D. (2005). Autonomous Pedestrians, Eurographics/ACM SIGGRAPH Symposium on Computer Animation, pp19-28.
7. Bruderlin, A. and Calvert, T. (1989). Goal-Directed Dynamic Animation for Human Walking, ACM Computer Graphics, pp233-242.
8. Menache, A. (2000). Understanding Motion Capture for Computer Animation and Video Games, Morgan Kaufmann.
9. Yin, K. K., Loken, K. and Van de Panne, M. (2007). SIMBICON: Simple Biped Locomotion Control, International Conference on Computer Graphics and Interactive Techniques, ACM SIGGRAPH, pp105-115.
10. Sok Won, K., Kim, M. and Lee J. (2007). Simulating Biped Behaviours from Human Motion Data, International Conference on Computer Graphics and Interactive Techniques, ACM SIGGRAPH, pp79-87.
11. Chai, J. and Hodgins, J. (2007). Constrained-based Optimisation Using a Statistical Dynamic Model, International Conference on Computer Graphics and Interactive Techniques, ACM SIGGRAP, pp8-17.
12. Kovar, L., Gleicher, M. and Pighin, F. H. (2002). Motion Graphs, SIGGRAPH, pp473-482.
13. Zordan, V., Majkowska, A., Chiu, B. and Fast, M. (2005). Dynamic Response for Motion Capture Animation, ACM Transactions on Graphics, Vol.24, No.3, pp697-701.
14. Savva, A. and Stylianou, V. (2010). Surface generation from a non-structured set of vertices, Proc International Conference on Applications of Mathematics in technical and Natural Sciences (AMiTaNS'10), Published by the American Institute of Physics, Bulgaria, pp426-434.
15. Ma, W. (2005). Subdivision surfaces for CAD – An overview, Computer Aided Design, 37(7), pp693-709.
16. Loizides, A., Loizidou Himona, S. and Chrysanthou, Y. (2008). Visualisation Tool for Representing Synthetic Facial Emotional Expressions, 12th International Conference on Information Visualisation (IV08), UK, pp 135-140.