Kinect Crowd Interaction

Lei Wei, Vu Le, Wael Abdelrahman, Mohammed Hossny, Douglas Creighton, Saeid Nahavandi

Centre for Intelligent Systems Research, Deakin University, Australia

{lei.wei | vu.le | wael.abdelrahman | mohammed.hossny | douglas.creighton | saeid.nahavandi} @deakin.edu.au

Abstract. Most of the state-of-the-art commercial simulation software mainly focuses on providing realistic animations and convincing artificial intelligence to avatars in the scenario. However, works on how to trigger the events and avatar reactions in the scenario in a natural and intuitive way are less noticed and developed. Typical events are usually triggered by predefined timestamps. Once the events are set, there is no easy way to interactively generate new events while the scene is running and therefore difficult to dynamically affect the avatar reactions. Based on this situation, we propose a framework to use human gesture as input to trigger events within a DI-Guy simulation scenario in real-time, which could greatly help users to control events and avatar reactions in the scenario. By implementing such a framework, we will be able to identify user's intentions interactively and ensure that the avatars make corresponding reactions.

1. INTRODUCTION

Crowd simulation and interactions on computers have been a hot research topic in recent years, especially with the world population growth. As the increment of computational power on personal computers, researchers are enabled to incorporate more advanced simulation techniques such as more realistic animations and more convincing artificial intelligence to avatars [1, 2, 3, 26, 27 and 28]. Events and the consequent reactions in a simulation scenario are usually predefined with fixed timestamps or written within scripts, which leads to predictable and unrealistic results. There have been efforts in trying to adopt random events and more variables in the scenario to simulate uncertainty, but it still totally relies on computations rather than actual user interactions.

On the other hand, the way humans interact with the simulation scenarios are less noticed and developed. Even very sophisticated simulation software has rather simple and straightforward approach to handle user input and interactions. This leads to non-intuitive approaches for human machine interaction and usually causes rigid and non-immersive simulation procedures.

We analysed the current state-of-the-art of crowd simulation and realised that it requires to be improved by having more intelligent and autonomy to make the whole simulation procedure as realistic as possible. Avatar reactions will be affected not only by pre-defined in-scene events but also by collateral reactions from other avatars as well as external events and human interactions. Meanwhile, gesture-based human machine interaction needs to be advanced further and fit into the right niche for practical use.

In this paper we seek to fill this gap by proposing a gesture recognition framework using Microsoft Kinect system. The aim is to allow real-time events, which are triggered by human gesture recognition, to interact with avatar crowd inside off the shelf simulation environments such as DI-Guy.

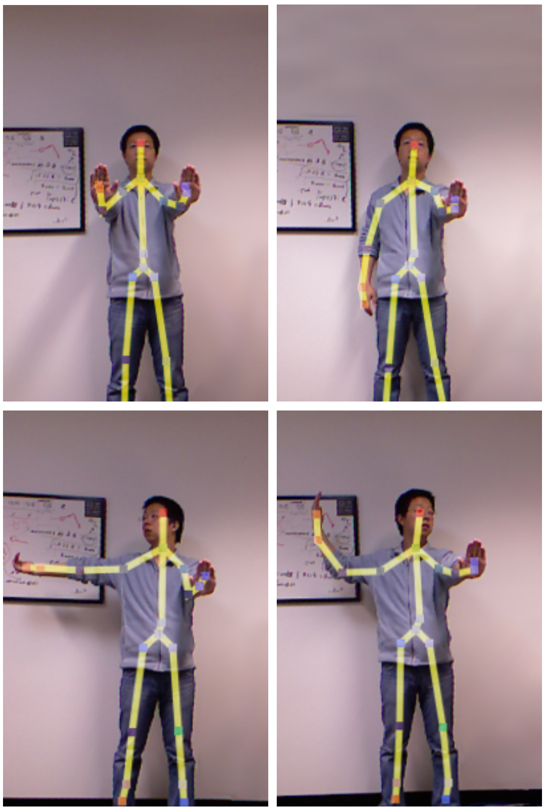


Figure 1: Figure 1: Gesture recognition using Kinect motion tracker

The rest of the paper is organised as follows. In Section 2 we review related work on crowd simulation and gesture based control system. In Section 3, we demonstrate our proposed solution to gesture-based human computer interaction, including the overall framework as well as details of each component in the framework. Chapter 4 focuses on implementation and technical details of the framework. We also discuss several real life application examples to show the advantage and capability of the proposed idea. Finally in Chapter 5 we summarize the work done and propose future work along this research and development direction.

2. LITERATURE REVIEW

Crowd encompasses many behaviours reacting to different scenarios, under different space and time. Some behaviour are influenced by the individual's personality and emotions, while other behaviours are affected by the surrounding environment and the behaviour of other people surround them, when reacting to a certain event. Panic and aggressive crowd usually reduce the overall egress rate of an evacuation, in a crowded environment [1]. Although there are existing crowd control methods, such as placing visible obstacles in the path to disperse a crowd in order to improve the flow rate [2]. In major evacuation events, such as a fire break out raises the level of panic, causing a reduction to evacuation time. Under these situations, the present of authorities to direct and control the traffic is a viable option and an effective mean to reduce the panic level [3-4].

In crowd simulation studies, authors are usually concerned with the details of animation and avatar artificial intelligence, where behaviour is trigger at predefined or randomly generated elapsed time rather than user driven control. This issue has been investigated in the work of Patil et al. [5], where a controller that allows the user to interact with a crowd in the simulation environment was developed. Their work proposed a reactive navigation field control technique to enable realistic crowd moving behaviour to resolve common congestion issues. The derived method required user to interactively specify the navigation field's direction, by using a mouse interface to sketch a path or by import motion flow fields extracted from video footage. This path renders the surrounding region into vector fields, which influence the behaviour of nearest active crowd agent by guiding and directing them towards a desired goal. This approach is similar to the proposition in Reynolds [6], where user defined flow field is employed to steer flock of agents inside the simulation environment.

Even though the computer technology was developed in the 40's, many initial researches on the interaction with the computer system took places in early 60's [7]. Some researchers began in the area of gesture-based drawing using light pen [8], followed by hand controlled mouse concept that replaced the previous light pen system for text selection and interface manipulation [9]. Concurrently, other researchers introduced the concept of 2D and 3D in the area of computer aided design (CAD) [10-11] and the introduction of video game technologies [7, 12]. These are amongst some of the important researches that lead to the development of gesture recognition, virtual reality, augmented reality, simulation and modelling disciplines, which changed the way we interact with machines today.

Despite the fact that many changes took places in the computer technology and its software application, the method to interact and manipulate the software application still relies heavily on the event trigger by the mouse and keyboard devices. Some authors tried to improve this by proposing alternative computer interfaces to perform common tasks. Such as using hand gestures to simulate a 3D mouse motion [13], moving an object through a maze [14] and controlling home appliances [15, 16]. While other authors look at facial gesture recognition to enable physicaly challenged individuals to interact with a computer [17, 18].

In simplest terms, gesture recognition is the process of interpreting human motion using computation algorithms. It has a wide-range of applications that makes it appealing to researchers [19]. Aside from providing us the ability to interact with hearing impaired people [20-21], it also allows us to interact with computer systems and controlling different appliances as seen previously in [13-18]. Although many gesture-based control methodologies have been established over time, their applications were initially limited to some specialised domains within the research community. This barrier has been knocked down by the entertainment industry, as it sees the true benefit and profit lying inside these innovative technologies, which could potentially be used to improve people's living style. With the involvement of entertainment industry, gesture recognition technology has been successfully exposes to a wider range of audiences. One of the fruitful implementation of such technology is the Nintendo Wii game console system. The system allows users to physically interact with a virtual environment in a healthy, fun and cost effective way.

Another popular candidate in the gesture recognition category is undoubtedly the low cost depth sensor, Kinect^{TM} device from Microsoft. The technology was initially developed for Xbox game consoles, until open source developers created a working software driver and made it available to the personal computer community. This simplified architecture has attracted many researchers, which initiated several fruitful research outcomes [22-25]. The software library has been used to intuitively interact with robots [22], physical rehabilitation process [23], interact with augmented reality virtual objects [24] and recognise objects inside an environment [25]. Due to its capability, portability and affordability, while allowing multiple researchers to work on the project concurrently, it was chosen in this study.

DI-Guy is a crowd modelling and simulation software package. It facilitates to modelling crowds and individuals with different social behaviours that change according to different situations and environments while react intelligently to ongoing events. The behaviour library in DI-Guy ranging from military formation, flee, follow, fear, aggressive, crouch, down to the micro level such as, shake head, wave, eat, talk back, gazing and facial expressions and many other features. Social behaviours are defined through action beads that trigger by predetermined time stamp. Complex behaviour for triggering events can also be defined using decision bead and script bead. Like other crowd simulation software packages, DI-Guy triggers user events through keyboard and mouse, predefined or randomly generated time stamps and sensor regions. However, such systems lack the immersive feature that allows the user to interact intuitively with the simulated crowd. In this research, our aim is to introduce a framework that allows the user to be immersed inside crowd modelling virtual environment allowing her to engage intuitive interactions with simulated crowds. The framework facilitates performing motion gestures to interact and control virtual crowds.

3. DESIGN FRAMEWORK

The proposed framework is implemented in two parts: a viewer responsible for human skeleton recognition and a plug-in to the crowd simulation software to receive parse the data into different human gestures and then trigger corresponding in-scene events. The framework is a distributed system in which different modules are communicating and synchronizing through data streams. This provides a scalable loosely coupled highly cohesive modular framework where any component can be altered or modified without redesigning the whole system. The major components of the framework are: motion capture module, motion analyser and interpreter module, and visualization and user interaction module. Figure 2 shows the flow diagram of the proposed framework.

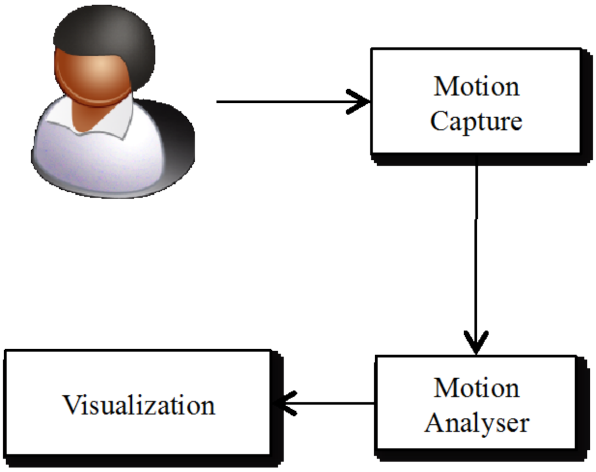
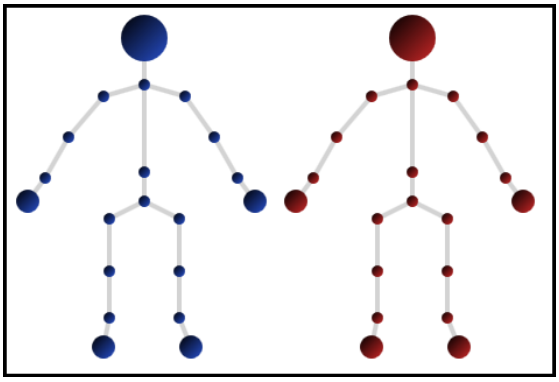


Figure 2: Overall design framework with major components.

3.1 Motion capture module

The motion capture device employed in this work is Microsoft Kinect™. This rapidly developing technology is growing a lot of interest from the research community due to its efficient real time capabilities with relatively low cost when compared to other motion capture systems. The Kinect™ device can recognize and track up to four human skeletons with Z value (Distance between skeleton joints and Kinect™ sensors). Most of the human joints can be reported such as arms and legs with exclusion of finer details such as fingers and face details. Figure 3 shows 20 human skeleton joints recognised by Kinect™.

Using these joint position values, many gestures can be formed and identified efficiently with an effective accuracy. An ethernet communication module sender module was designed to report the values of these joints to the gesture analyser module. The main challenge in this module is to filter the sent signal and execute pre-processing for easier analysis. The sender is by default deployed on a remote machine although it can also be on the same one.



Head	x1
Neck	x1
Shoulders	x2
Elbows	x2
Wrists	x2
Hands	x2
Spine	x1
Hip	x1
Thighs	x2
Knees	x2
Ankles	x2
Feet	x2

Figure 3: Recognizable human skeleton joints in Kinect.

3.2 Motion analyser and interpreter module

The second module is the motion analyser and interpreter. This module is the core part of the proposed framework and is responsible for receiving joint positional values streamed from the motion capture module, identify and analyse it, and resend pre-defined orders to the visualization module. This is required in order to decouple the visualization and the processing of the data. Also, the DI-Guy™ software requires this decoupling in the form of environment plugins.

The motion stream is first split into frames. Each frame is identified using a pair of header and footer. The gestures are extracted, as will be seen in the next section, from the stream. This is determined by monitoring one or several joints through consecutive number of frames. The main challenge here is the ability to handle the case of lost packets during gesture recognition.

Upon receiving real-time joint data from remote or local sender, the received data will first be verified to ensure its integrity. If the received data is corrupted, we propose several approaches to handle the situation, depending on the number of consecutively corrupted frames as well as the fidelity requirement of the specific application. When only few frames were corrupted, we propose to first analyse the previous frames and determine if there was an abrupt motion during the corrupted frames. If not, a linear interpolation algorithm will be applied to estimate the corrupted frame data and make sure the reconstructed data generate smooth motion with previous and current frames. If the analysis shows that there was an abrupt motion during the corrupted frames, a warning message will show and inform the user to redo specific actions for the previous few seconds.

We developed a general routine for parsing all receiving skeleton joint positions/orientations from the user and map them onto predefined gesture libraries. Ideally the gesture library could have as many gestures as possible. However, due the update rate and input data accuracy, we generally store gestures with distinct limb and body features. A brief list of available gestures in our current library is listed in Figure 4, and it is still expanding.

Gestures	Involved joints
Left/Right hand up (over the head)	Left/Right hand, head
Left/Right hand waving (between neck and hip)	Left/Right hand, Left/Right elbow, neck, hip
Left/Right hand down (lower than hip)	Left/Right hand, neck, hip
Cross hands	hands, elbows, neck, hip
Prepare to Fight	hands, elbows, feet, knees, neck, hip
Prepare to Run	hands, elbows, feet, knees, neck, hip
Prepare to Shoot	hands, elbows, neck, hip
Stop other people	Left/Right hand, neck, hip
Let other people go	hands, neck, hip
Bruce Lee style	hands, elbows, shoulders, feet, neck, hip

Figure 4: Recognizable gestures and related joints.

The gestures are recognised based on analysis of relative positions and joint angles of related joints on the human skeleton. We first analyse joints by regions such as upper limbs, lower limbs, body, head, and then use pre-defined criteria on joints values to evaluate the gesture. After that, gestures from different regions are combined to compose the final gesture. Each gesture has one or several corresponding events which will trigger in-scene avatar actions as well as collateral actions. As the user continuously making new gestures, the avatars are driven by continuous events and react accordingly.

3.3 Visualization and user interaction module

The third module is dedicated for visualizing the scenarios and providing an interactive environment for the user. This is represented in DI-Guy™.

The DI-Guy™ software has an open architecture which supports functionality extensions in the form of plugins. Plugins can be compiled according to DI-Guy interfacing requirements and invoked when the simulation starts and then responds to the simulation scenario events. We identify two challenges in this module. The first challenge is to fit the virtual character joints with the supplied joints from the Kinect™ device, while the second challenge is to define character reactions to the user movement to form a two way interaction environment. The plugin also is responsible for controlling the virtual scene characters and calling artificial intelligence (AI) routines. Once each gesture is recognised, the corresponding events will be triggered in the scene and affect some or all avatar actions. Some avatars' reactions may also be affected collaterally by other avatars' reactions. All the reactions to certain events are generated and controlled by AI routines in the crowd simulation package. DI-Guy employs the LUA language to for AI scripts.

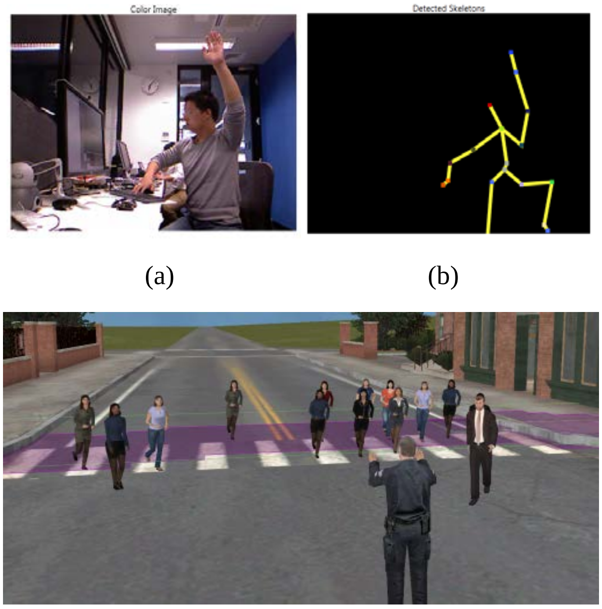
The user interacts with the environment in a hands-free basis using Kinect™ motion tracker. The environment also can support the introduction of other input devices such as a haptic device or a 3D mouse.

4. TECHNICAL DETAILS AND APPLICATION EXAMPLES

The viewer is implemented based on Kinect SDK via C# programming language. This SDK facilitates interactive identification of the user's skeleton and returns the corresponding joint positions. The plug-in is implemented in C++ programming language through reserved interfaces in DI-Guy. The two parts communicate through TCP/IP communication sockets.

The proposed framework could support various applications including but not limited to civil, military training, and teleportation. The examples in this paper focus on interactions with crowds especially in the domain of civil law enforcement. Two test cases were developed: a police officer organizing traffic for pedestrians and two police officers evacuating civilians from an explosion in a building.

The first case study shown in Figure 4 involves a police officer and a crowd of civilians. There are some disasters happened in the scene and civilians are running. The user will need to act as the police officer and use his/her own gestures to guarantee the safety of the civilians. The scene has predefined action beads which connects gestures of the police officer with reactions of the crowd, i.e. when the police officer waves forward, the crowd will run towards him, when the police officer waves backward, the crowd will run away from him, when the police officer raises both of his arms and show stop, the crowd will stop and wait for further commands. The police officer's gestures are triggered by user interactions instead of the predefined timestamps. In this test case, we identify three gestures from the Kinect recognised posture: left hand up corresponds to wave forward; right hand up corresponds to wave backward; both hands up correspond to stop. The gesture parsing procedure is done by analysing relative positions between wrist and head.

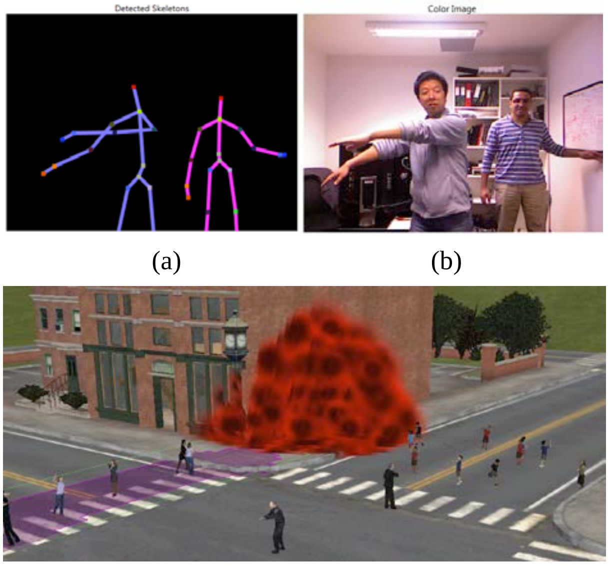


(a) (b)

(c)

Figure 5: Using Kinect to control the crowd reactions in a DI-Guy scenario. (a) Actual human gesture. (b) Detected skeletons. (c) Crowd reactions to recognised commands parsed from detected skeletons

A more advanced case where two police officers are controlling an emergency evacuation of an explosion in a building is shown in Figure 5. This is more challenging as the timing of the gestures is important and a good perception of the whole scene is required. Besides, civilians further from the police officer will not see his gestures and respond. On the other hand, they will start to wonder why civilians closer to the police officer are running and later realize the hazardous situation. Therefore, user interactions will not only directly affect certain avatars, but also cause collateral reactions to other avatars. This procedure involve more processing power over artificial intelligence as well as extra control parameters such as area of influence, other avatars' action analysis, decision and action. Besides, there are cases the two police officers make different commands for evacuation. In such cases, the avatars will have to make their decisions and act quickly based on each of the two officers, as well as other avatars' actions. This case would provide great immersion in simulating real-life emergency situations and help train authorities as well as civilians.



(a) (b)

(c)

Figure 6: Using Kinect to control the crowd reactions in a DI-Guy scenario (Two users). (a) Actual human gesture. (b) Detected skeletons. (c) Crowd reactions to recognised commands parsed from detected skeletons

5. CONCLUSION AND FUTURE WORK

To summarize, this framework could provide fast, affordable yet reliable environments for real time interactive crowd simulations. It supports direct user gesture input and allow for direct and collateral avatar reactions based on artificial intelligence. The gesture libraries provide most commonly used gesture recognitions and it could be easily expanded with more requirements. Several application examples based on commercial simulation software DI-Guy are also demonstrated to show the capabilities of the proposed framework.

In the future, we are going to further expand this framework to other applications, which involves interactive human gestures to perform both virtual and real-life control of human as well as robotics. We are also working on using Kinect input to control haptic devices and use them as force output equipment for educational and entertainment purposes.

ACKNOWLEDGEMENT

This project is supported by the Centre for Intelligent Systems Research in Deakin University.

REFERENCE

1. Helbing, D., Farkas, I. and Vicsek, T. (2000) "Simulating dynamical features of escape panic," *in Nature 407 International Weekly Journal of Science*, Sep, pp.487-490.

2. Johansson, A. and Helbing, D. (2007) "Pedestrian flow optimization with a genetic algorithm based on boolean grids," *in Pedestrian and Evacuation Dynamics 2005*, Waldau, N., Gattermann, P. Knoflacher, H. and Schrekenberg, M (2007). Springer, Berlin, pp. 267-272.

3. Tsai, J., Fridman, N., Bowring, E., Brown, M., Epstein, S., Kaminka, G., Marsella, S., Ogden, A., Rika, I., Sheel, A., Taylor, M. E., Wang, X., Zilka, A. and Tambe, M. (2011) "ESCAPES – Evacuation simulation with children, authorities, parents, emotions, and social comparison," *in Proceeding of the 10th International Conference on Autonomous Agents and Multiagent Systems*, Taipei, May, pp.457-464.

4. Smith, C. A and Ellsworth, P. C. (1985) "Patterns of cognitive appraisal in emotion," *in Journal of Personality and Social Psychology*, vol. 48, No. 4, pp. 813-838.

5. Patil, S., Berg, J. V. D., Curtis, S., Lin, M. and Manocha, D. (2011) "Directing crowd simulations using navigation fields," *in IEEE Transactions on Visualization and Computer Graphics*, vol. 17, no.2, Feb, pp. 244-254.

6. Reynolds, C. W. (1999) "Steering behaviors for autonomous characters," *in Proceeding of the Game Developer Conference*, San Francisco, pp. 763-782.

7. Myers, B. A. (1998) "A brief history of human computer interaction technology," *in ACM Interactions*, vol. 5, no. 2, March, pp. 44-54.

8. Sutherland, I. E. (1963) "SketchPad: a man-machine graphical communication system," *in AFIPS Spring Joint Computer Conference*, vol. 23, pp. 329-346.

9. English, W. K., Engelbard, D. C. and Berman, M. L. (1967) "Display selection techniques for text manipulation," *in IEEE Transaction on Human Factors in Electronics*, vol. HFE-8, no. 1, pp. 5-15.

10. Ross, D. and Rodriguez, J. (1963) "Theoretical foundations for the computer-aided design system," in *AFIPS Spring Joint Computer Conference*, vol. 23. pp. 305-322.

11. Johnson, T. (1963) "Sketchpad III: three dimensional graphical communication with a digital computer," in *AFIPS Spring Joint Computer Conference.* 1963. vol. 23. pp. 347-353.

12. Levy, S. (1984) "*Hackers: heroes of the computer evolution,"* Anchor Press/Doubleday, New York.

13. Bretzner, L. and Lindeberg, T. (1998) "Use your hand as a 3-D mouse, or, relative orientation from extended sequences of sparse point and line correspondences using the affine trifocal tensor," *in Proceeding of the 5th European Conference on Computer Vision*, vol. 1406, Berlin, Jun, pp. 141-157.

14. Sepehri, A., Yacoob, Y. and Davis, L. S (2006) "Employing the hand as an interface device," *in Journal of Multimedia*, vol. 1, no. 7, Nov, pp. 18-29.

15. Do, J. H., Jung, J. W., Sung, H. J., Jang, H. and Bien, Z. (2006) "Advanced soft remote control system using hand gesture," *in Proceeding of the 5th Mexican International Conference on Artificial Intelligence*, vol. 4293, Nov, pp. 745-755.

16. Lee, D. W., Lim, J. M., Sunwoo, J. Cho, I. Y. and Lee, C. H. (2009) "Actual remote control: universal control using hand motions on a virtual menu," *in IEEE Transaction on Consumer Electronics*, vol. 55, no. 3, Aug, pp. 1439-1446.

17. Dalka, P. and Czyzewski A. (2009) "Lip movement and gesture recognition for multimodal human-computer interface," *in Proceeding of the International Conference on Computer Science and Information Technology*, Oct, pp. 451-455.

18. Varona, J. Yee, C. M. and Perrales F. J. "Hands-free vision-based interface for computer accessibility," *in Journal of Network and Computer Applications*, vol. 31, no. 4, Nov, pp. 357-374.

19. Mitra, S. and Acharya, T. (2007) "Gesture recognition: a survey," *in IEEE Transactions on Systems, Man, and Cybernetics*, vol. 37, no. 3, pp. 311-324.

20. Liang, R. H. and Ouhyoung M (2008) "A real-time continuous alphabetic sign language to speech conversion VR system," *in Computer Graphics Forum*, vol. 14, no. 3, Aug, pp. 67-76.

21. Vamossy, Z., Toth, A. and Benedek, B. (2007) "Virtual hand – hand gesture recognition system," *in Proceeding of the 5th International Symposium on Intelligent Systems and Informatics*, Subotica, pp. 97-102.

22. Bergh, V. D., Carton, D., Nijs, D. R., Mitsou, N. Landiedel, C. Kuehnlenz, K., Wollherr, D. Gool, V. L. and Buss, M. (2011) "Real-time 3d hand gesture interaction with a robot for understanding directions from humans," *in the 20th IEEE International Symposium on Robot and Human Interactive Communication*, Atlanta, Aug, pp. 357-362.

23. Chang, Y. J., Chen, S. F. and J. D. Huang (2011) "A Kinect-based system for physical rehabilitation: a pilot study for young adults with motor disabilities,*" in Journal of Research in Developmental Disabilities*, vol. 32, no. 6, Nov, pp. 2566-2570.

24. Santos, E. S., Lamounier, E. A and Cardoso, A. (2011) "Interaction in augmented reality environments using kinect," *in Symposium on Virtual Reality 2011*, Uberlandia, May, pp.112-121.

25. Devereux, D., Mitra, B. Holland, O. and Diamond, A. (2011) "Using the Microsoft kinect to model the environment of an anthropomimetic robot," *in Proceeding of the 2nd IASTED International Conference on Robotics*, Pittsburgh, 1-8.

26. Narain, R., Golas, A., Curtis S. and Lin, M. C (2009) "Aggregate dynamics for dense crowd simulation," *in ACM Transaction of Graphics (Proceeding of ACM SIGGRAPH Asia)*, vol. 28, no. 5, Dec, pp. 122:1-122:8.

27. Pelechano N., Allbeck J. M. and Badler N. I. (2007) "Controlling individual agents in high-density crowd simulation," in *Eurographics/ACM SIGGRAPH Symposium on Computer Animation*, Aug, pp. 99-108.

28. Guy, S. J., Kim, S., Lin, M. C. and Manocha, D. (2011) "Simulating heterogeneous crowd behaviors using personality trait theory,*" in Eurographics/ACM SIGGRAPH Symposium on Computer Animation*, vol. pp. 43-52.