# GeFighters: an Experiment for Gesture-based Interaction Analysis in a Fighting Game

João Marcelo Teixeira Thiago Farias Guilherme Moura
João Paulo Lima Saulo Pessoa Veronica Teichrieb

Federal University of Pernambuco, Computer Science Center, Brazil


Figure 1: GeFighters: a) scene and characters; b) gesture based interaction.

## Abstract

This paper presents GeFighters, a 3D fighting game that supports gesture based interaction. This application has been used to test and analyze gesture interaction in the context of games that need short and reliable response times. Some inherent aspects of the application have been analyzed, like the impact that interaction causes on frame renderization and response time related to the control of game characters. In order to implement the desired interaction, an input devices management platform, named CIDA, has been used.

**Keywords**: interaction, gesture recognition, games, GeFighters, CIDA

**Authors' contact**:
`{jmxnt, tsmcf, gsm, jpsml, sap, vt}@cin.ufpe.br`

## 1. Introduction

In the last decade, the innovative aspect of games has been decreasing gradually, which makes specific entertainment applications frequently repetitive. Similar types of games are published, differing only in history, visual aspects and interaction method used. The interaction is often reused from the paradigm of similar games. These games, targeting either consoles or personal computers, are controlled by joysticks, or simply by keyboard and mouse inputs. Some console games started capturing gestures through video processing techniques and as a result of the success obtained other developers followed this new approach as well.

It's a fact that the use of gesture capture increases significantly user's level of immersion, even if the visualization of the application is not done by HMDs (Head Mounted Displays) or CAVE (Cave Automatic Virtual Environment) systems. Consequently, this type of interaction acts like an extra attractive to different users, since it can be used in a large variety of games, like the ones related to sports, races, fights and simulations.

In order to attend the real-time needs of the applications, chosen techniques must correspond to speed and precision constraints, without degrading neither the gameplay nor the immersion.

This work makes use of the GeFighters (Gesture Fighters) application to test and analyze gesture interaction with games that need short and reliable response times. The application is a 3D fighting game whose interaction is based on the detection of fiducial markers (illustrated in Figure 1b), which provides information about the gesture performed. In this paper there will be analyzed some aspects inherent to the application, like the impact that interaction causes on frame renderization (variation on Frames Per Second (FPS) rate) and response time related to the control of game characters (capture delay due to image processing).

Section 2 presents games and applications in which the interaction is performed through the use of gestures, like the one proposed in this paper, as well as the technologies involved in the pattern capture process and user action interpretation. GeFighters is described in Section 3. The gesture interpretation and its mapping onto game commands, which is possible by the use of an interaction input devices abstraction layer, named

CIDA (Caothic Interaction Devices Abstraction), is explained in Section 4. Section 5 presents an analysis of the metrics used, the extraction methods and some illustrative charts. Section 6 highlights the contributions of this work and proposes future work to improve the methods and ideas developed herein.

## 2. Related Work

Conventional input devices like a joystick, a keyboard and a mouse control most of computer games. Normally, these games do not allow the player to use his/her natural movements as interaction method. This implies that he/she must learn how to control the application, associating sequences of button pushes and axes movements to actions in the game. Gestures could provide a much more intuitive way of interaction, since the user "already knows" how to control the game. In Decathlete [Freeman et al. 1998], for example, the user really has to run in order to get his/her character running. Furthermore, users with special needs would benefit themselves by the possibility of controlling games through blinks or head movements, since they generally do not have the necessary strength or coordination to use conventional input devices [Steriadis and Constantinou 2003].

Therefore, a considerable effort has been applied in researching gesture recognition, mainly for application in medicine and industry areas [Köchy et al. 1998; Myers 1998]. This stimulates the creation of commercial products, as the iMatte's iSkia [iSkia Projector 2006] and the Cybernet System's GestureStorm [GestureStorm 2006], illustrated in Figure 2. The first one presents a technology that allows television presenters to interact with projectors and screens using gestures, while the second one allows weather forecast presenters to use hand movements to illustrate their presentations.


Figure 2: Hand movements to interact with GestureStorm.

Many virtual keyboards for PDAs (Personal Digital Assistants), like the one produced by Canesta [Canesta 2006] were created as result of these researches (Figure 3). They work projecting keys on a plain surface, and then capturing fingers movements to identify "pressed" keys.


Figure 3: Canesta's virtual keyboard for PDAs.

Console game manufacturers also have introduced the concept of gesture-based interaction in their systems. Sega has developed the Activator Ring [Activator Ring 2006] for the Mega Drive videogame [Mega Drive 2006]. The ring is formed by eight different sections and equipped with sensors, which correspond to the buttons of a common control. In order to interact, the user steps on the octagon and indicates character-desired movements. More recently, Sony has commercially introduced the EyeToy camera [EyeToy USB Camera 2006], which allows that some games are controlled using player gestures. The control of the not yet published Nintendo Wii [Nintendo Wii 2006] has a 6DOF (Six Degrees Of Freedom) tracker, which brings a whole new way of interaction to console games.

Despite the advantages of using gestures as game controllers are very clear, there are many details to be considered by developers. [Freeman et al. 1998] have identified some challenges of this new type of interaction. For example, the response time (the user must not notice any delay between his/her gestures and the corresponding answer provided by the computer), the reliability of the algorithms (they must be robust enough in order to support imperfect and non-intentional movements from the user) and the cost (conventional interaction devices have low cost). Furthermore, common gesture recognition devices, like data gloves and body sensors, are too much intrusive, and because of that their daily use becomes impracticable.

In this scenario, the use of traditional Augmented Reality (AR) techniques and tools for capturing and recognizing user movements is the best solution available. Software libraries like ARToolKit support the use of common low cost cameras and provide efficient pattern recognition algorithms [ARToolKit 2006]. [Bunchmann et al. 2004] have recently developed FingARtips, which focuses on the interaction with virtual objects in AR environments, based on fingers movement.

SymBall [Hakkarainen and Woodward 2005] is another example of this type of applications. It was designed to run on a camera-enabled mobile phone and it simulates a table tennis game in which the user moves the phone in order to hit the balls coming from the virtual opponent. Figure 4 shows SymBall players interacting with the game.

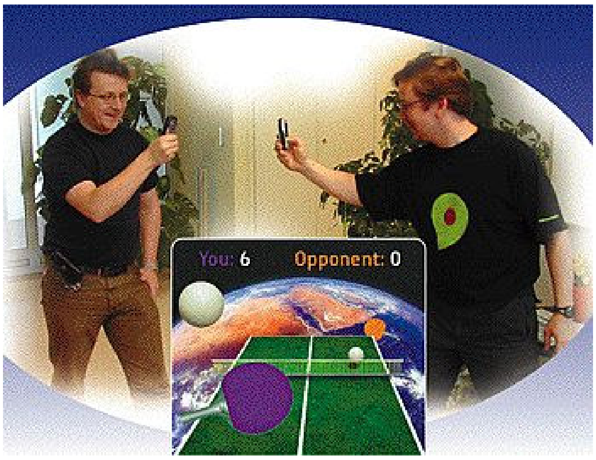


Figure 4: SymBall players.

## 3. The GeFighters Game


Figure 5: 3D fighting games: a) Tekken; b) Dead or Alive; c) GeFighters.

GeFighters is a 3D fighting game which has its conception based on well-known games like Tekken [Tekken 2006] and Dead or Alive [Dead or Alive 2006]. Figure 1a illustrates GeFighters, as well as scenes of other fighting games (Figure 5a and Figure 5b). GeFighters aims to experience and validate the use of non-conventional interaction methods, specifically gesture based ones, not pretending to be graphically sophisticated as commercial fighting games normally are.

Characters (dancers, instead of common fighters) and a virtual environment (a dancing house from the 70's) make up GeFighters. Players interact with the game performing gestures.

The game presents general aspects of comedy and fight. The fighters' arena looks like a disco house, with a great variety of ambient lights and videos playing on a huge screen, like the real ones.

Figure 5c illustrates the scene. The characters perform some dance steps while the players are not controlling them, and when there is some user interaction they perform actions that remember a fight. The winning character is the one who wins two rounds of the game.

GeFighters was created with the main objective of evaluating an interaction method based on gestures. In order to implement efficiently the desired interaction interface, an input devices management platform, named CIDA and developed by the authors, was used [Farias et al. 2006]. CIDA provides for the game a high level abstraction of the input devices to be used and also makes the code independent from the interaction type used. More details about the gesture-based interaction method implemented are presented in Section 4.

Using CIDA allows distributing the whole game on up to three different computers: one responsible for the game processing and the other two functioning as the game controllers (capturing and interpreting the players movements). This way, besides device use flexibility, the platform also provides location abstraction for the application (the input device may not be necessarily connected to the same computer than the one running the game).

GeFighters' architecture is divided in two modules, namely OGRE and CIDA, as illustrated in Figure 6.

The OGRE module has this name because it comprehends the OGRE (Object-oriented Graphics Rendering Engine) graphics engine [OGRE 2006]. It is responsible for the visual part of the game, offering the possibility of using Direct3D or OpenGL to render the 2D and 3D models. The CIDA module functions as a bridge between user and application. This module, using a CIDA plug-in that maps user input into a virtual joystick, manages all the interaction process. Another plug-in allows the connection to controllers located in different computers, as mentioned before.

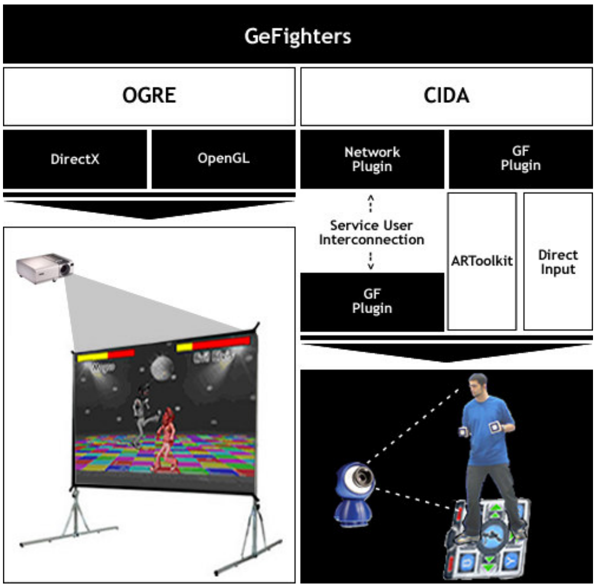


Figure 6: GeFighters' architecture.

The 3D Studio Max tool [3D Studio Max 2006] has been used in the modeling of both, a boy and a girl character. Some skin and clothes textures were used to model more realistically the dancers. The animation process was hard because it depended on the creation of a hierarchical bone structure based on human body articulations, in order to make the character movements close to reality. Figure 7 shows the bone structure of the male character. After the so called Skinning process, where the skin is attached to the positioned bones, the animation paths were defined and the models exported and further loaded by the game.

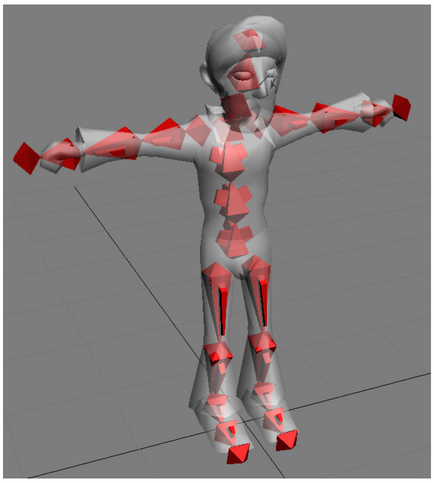

Figure 7: Character's bone structure.

Figure 8 a to j illustrate some character movements: idle, walking forward, walking backward, stooping, punching, kicking, jumping northward, rotating in the air, and giving some air attacks, respectively.

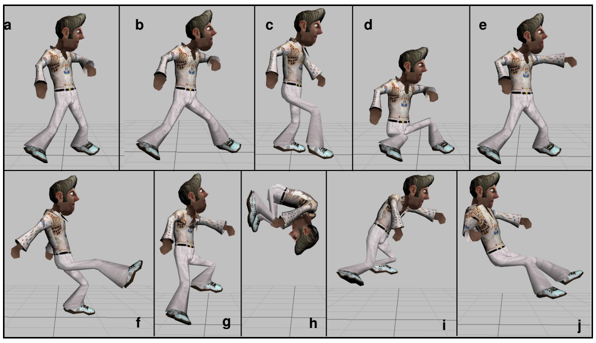


Figure 8: Character movements.

## 4. Gesture Based Interaction Interface

As personal computers and gaming consoles increase their processing power, new interaction methods can be considered. Part of these new ways of interaction originate from researches in the Virtual Reality (VR) area, using devices such as gloves, HMDs, trackers, among others.

Immersion has been taken into consideration by applications that demand unconventional ways of interaction. In these environments, the interaction has to be as natural as possible in order to conserve its immersive aspect.

One of the techniques utilized to interact with immersive environments is a gesture-based interface. Gesture based interaction can be done using gloves, trackers and even haptic devices attached to the user's hands. Another way of identifying gestures is by image capture, utilizing a camera. Video is captured and processed, frame by frame, with the purpose of extracting patterns relative to the movement realized. A set of data that can be interpreted as a gesture is obtained from a sequence of recognized patterns. This technique is used by some games, like EyeToy: Kinetic [EyeToy: Kinetic 2006], where a virtual personal trainer suggests a series of exercises and verifies if the player is practicing them properly. This game has been designed for the PlayStation 2 console and uses the EyeToy USB Camera peripheral to capture the video that will be used to identify the player's deficiencies, so that the virtual trainer may perform his/her judgment about the correctness of the exercises practiced by the player. This peripheral has not been spread, but the utilization of the interaction style offered by the games that use it is growing, together with AR and Mixed Reality (MR) applications. The main goal of these applications is to combine virtual objects with the real world, using marker patterns as the way of interaction.

Some tools and libraries have been developed in the AR, MR and computer vision areas, like ARToolKit [ARToolKit 2006], MXRToolKit [Bath and Paxman 2005] and OpenCV [OpenCV 2006], respectively. All of these can be used to detect and recognize patterns, although the last one is more generic and can realize a specific processing for detecting unconventional (markerless) patterns.

Conventional patterns are the ones based on markers, consisting usually of squares with a black border and monochromatic symbols in their central region (see marker examples in Figure 9). These symbols can be a figure indexed by pattern generation tools or an id [Fiala 2004]. Unlike this kind of pattern, the unconventional ones can be formed by any information present in the environment (e.g. faces, fingers, luminosities, symmetries, contours etc.).

GeFighters uses ARToolKit to implement its gesture based interaction interface. Two markers, containing the G and F letters as patterns, are responsible for originating the character movements, as shown in Figure 9. The user holds one marker in each hand, in a way that the patterns are always pointing to the camera.

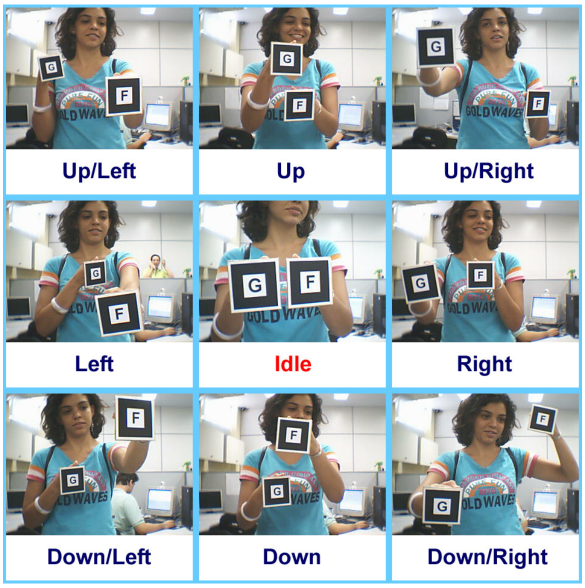


Figure 9: Direction of the X and Y axes related to the position of the markers.

In order to facilitate the explanation of how the mapping is realized, the pattern of the right hand was named G and the left hand one was named F. These two patterns are mapped to two joystick axes. Although this can be implemented in several ways, it was preferred to use a vector generated by the relative position of the two markers, like a game controller stick. This way, it can be obtained information about both, the X and Y axes. CIDA is responsible for mapping this relative position vector to the axes information, allowing the application to access the input device as a virtual joystick. The pattern detection library supplies information about the spatial location and rotation of the patterns, but GeFighters uses just two dimensions, since the mapping is done to two joystick axes.

The complete character movements are based on the relative position of the markers, as stated before. The F marker works as a reference point (that is, it seems to be static) and the position of the G marker defines the movement to be interpreted. For example, in case the user is holding the G marker more to the front compared to the F marker, it means that the value of the X axis is 1 (considering that it ranges between -1 and 1, from left to right). In case the G marker is positioned below the F marker, it means that the value of the Y axis is -1.

The positioning of the markers and its mapping to specific movements is illustrated in Figure 9. It shows all positions that can be mapped to the two joystick axes. The images on the corners represent the compound positions, formed when both X and Y axes are different from 0. The central image (when both markers appear side by side, at the same height) indicates the "idle" state of the control, that is, the character is not moving. In case the player wants the character to perform a "backward jumping" movement, he/she must place the markers like is shown in the upper left image, where the G pattern is behind and above the F pattern. In case the player wants the character to stoop, he/she must place the G marker below the F marker, as can be seen in the lower central image.

When developing games it is hard to synchronize the character with the user interaction. Sometimes the character is in the middle of a complex animation and the user asks the system to do something completely different. It is mandatory to be sure that the change will be smooth; otherwise the new movement will just instantly change the character, which is visually wrong. Therefore, the axes mapping has been implemented in a way that there is an intermediary region between the movement states, which is named deadzone. This region is responsible for ensuring that there are no oscillations during the transition between two states. This is avoided by the fact that the movement is not changed in the deadzone. This way, in order to pass from the "idle" state to the "right" state, the user has to pass through the deadzone. In case the marker location values supplied by the library oscillate due to problems with image capturing, the current state of the movement will not change. In summary, the states only change from one to another when there is a significant marker movement.

Another aspect is related to the use of F and G markers. For example, if the user mistakenly tries to lift F meaning up, the system identifies it as a down.

Besides the axes, a dance carpet was used to map the joystick buttons. The user has to step on the carpet sensors to control the attack actions of the character. It is possible to punch, to kick and even execute special movements, depending on the sequence in which they are pressed.

The dance carpet has six buttons and two axes, but the axes are not used, since the information relative to the X and Y axes are obtained from the markers. The carpet was formerly developed to be used with the PlayStation console, but can also be connected to a computer through the parallel port.

## 5. Analysis of Results

An evaluation has been performed in order to analyze response time of the gesture performed during runtime, as well as frame rate. Response time and frame rate are very relevant usability requirements of interactive applications.

For the purpose of evaluating GeFighters, the source code of the application has been modified to print the response delay, starting from the beginning of the gesture until the visual recognition and processing of the movement.

This process has been automated, and the camera input was substituted by a recorded video containing a gesture movement detectable by the game.

Firstly, a small amount of data were obtained to be used in a formula, presented in Equation 1, that computes the sufficient number of samples that is significative to obtain the analysis result. This formula is based on the confidence level and accuracy desired. The $Z$ parameter stands for the normal table, $\alpha$ for the desired confidence level, $S$ for the standard deviation, $r$ for the accuracy, and $x$ for the average of data acquired. Afterwards, a new amount of samples were captured, based on the formula result, and then analyzed. Three different resolutions were used, namely 640x480, 800x600 and 1024x768, to perform the tests.

$$n = \left( \frac{100 \times Z_{\left(1-\frac{\alpha}{2}\right)} \times S}{r \times \bar{x}} \right)^2$$

Equation 1: Number of samples.

Table 1: Response time results (ms).

| | Max | Min | Avg |
|---|---|---|---|
| 640x480 | 476.4854 | 430.1145 | 453.3000 |
| 800x600 | 498.4904 | 394.4809 | 446.4857 |
| 1024x768 | 556.9253 | 455.5746 | 506.2500 |

Response time results are shown in Table 1 and are graphically presented in Figure 10. Table 1 presents the maximum and minimum response times obtained for each resolution, as well as its average.

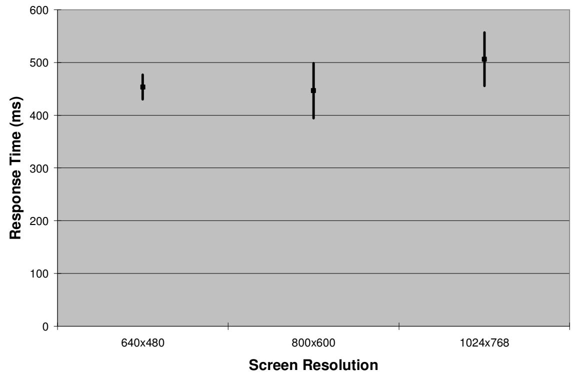


Figure 10: Response time analysis.

For the FPS performance tests, a similar one has been driven, tough not using the same formula to validate the results. A single, huge amount of data was collected, based on the delay between two frames while the application is running and processing the gesture information. The analysis results about the FPS tests performed are shown in Figure 11 and Table 2, respectively. As mentioned before, the maximum, minimum and average values are presented.

Table 2: FPS results.

| | Max | Min | Avg |
|---|---|---|---|
| 640x480 | 29.4853 | 27.2433 | 28.3643 |
| 800x600 | 25.1730 | 21.1487 | 23.1608 |
| 1024x768 | 16.8738 | 14.0324 | 15.4531 |

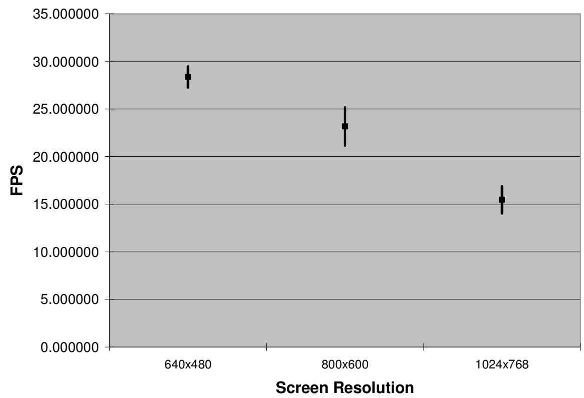


Figure 11: FPS analysis.

The hardware used for the tests was a P4 3.0GHz processor, 1GB of RAM and an nVidia FX5200 256MB GPU, on a stable and homogenous system.

Although the FPS tests were satisfactory, except in the highest resolution case, the response time results were under the expectations and could lead to a low experience related to usability aspects for a real-time game experience, such as a fighting or a car racing game.

## 6. Conclusions and Future Work

This paper presented the GeFighters game, an experiment created to analyze critical metrics about gesture-based interaction in games that have real-time constraints. The analyzed metrics were response time from interaction start to game state change and frame rate.

Some examples of well-succeeded games that use gesture have been mentioned, highlighting the inherent advantages of this interaction method.

The analysis result shows that FPS keeps satisfactory, but the obtained response time is not favorable to systems that need immediate responses. The system's bottleneck was identified as being the pattern recognition and the webcam communication bandwidth.

As future work, it is possible to isolate the capturing and pattern recognition module in a dedicated hardware, in order to reduce the image processing time. This module could provide digital outputs like a standard joystick and be used by applications through direct calls to the native communication interface or even through the implementation of a plug-in for the CIDA platform. The last option would guarantee the solution's abstraction and flexibility.

## References


ACTIVATOR RING, 2006. *Citing references: Sega* [online]. Available from: www.sega.co.jp [Accessed 31 August 2006].

ARTOOLKIT, 2006. *Citing references: Human Interface Technology Lab* [online]. Available from: www.hitl.washington.edu/artoolkit [Accessed 31 August 2006].

BATH, W., AND PAXMAN, J., 2005. UAV localisation & control through computer vision. *In: Proceedings of the Australasian Conference on Robotics & Automation, 5-7 December 2005 Sydney.*

BUCHMANN, V., VIOLICH, S., BILLINGHURST, M. AND COCKBURN, A., 2004. FingARtips – gesture based direct manipulation in augmented reality. *In: Proceedings of the International Conference on Computer Graphics and Interactive Techniques in Australasia and South East Asia, 15-18 June 2004, Singapore*. New York: ACM Press, 212-221.

CANESTA'S VIRTUAL KEYBOARD FOR PDAS, 2006. *Citing references: Canesta Inc.* [online]. Available from: www.canesta.com [Accessed 31 August 2006].

DEAD OR ALIVE, 2006. *Citing references: Dead or Alive* [online]. Available from: www.deadoralivegame.com/ [Accessed 31 August 2006].

EYETOY: KINETIC, 2006. *Citing references: EyeToy: Kinetic* [online]. Available from: www.eyetoykinetic.com [Accessed 31 August 2006].

EYETOY USB CAMERA, 2006. *Citing references: Sony Playstation* [online]. Available from: www.us.playstation.com [Accessed 31 August 2006].

FARIAS, T., TEIXEIRA, J.M., RODRIGUES, C.E., PESSOA, S., COSTA, N., TEICHRIEB, V. AND KELNER, J., 2006. CIDA: an interaction devices management platform. *In: Proceedings of the Symposium on Virtual Reality, 2-6 May 2006 Belém*. Porto Alegre: SBC, 271-284.

FIALA, M., 2004. *ARTag, an improved marker system based on ARToolkit*. NRC Technical Report, National Research Council of Canada, Canada.

FREEMAN, W.T., ANDERSON, D.B., BEARDSLEY, P.A., DODGE, C.N., ROTH, M.W.C.D. AND YERAZUNIS, W.S., 1998. Computer vision for interactive computer graphic. *IEEE Computer Graphics and Applications*, 18(3), 42-53.

GESTURESTORM WEATHER MAP MANAGEMENT SYSTEM, 2006. *Citing references: Cybernet Systems Corporation* [online]. Available from: www.cybernet.com [Accessed 31 August 2006].

HAKKARAINEN, M. AND WOODWARD, C., 2005. SymBall - camera driven table tennis for mobile phones. *In: Proceedings of the ACM SIGCHI International Conference on Advances in Computer Entertainment Technology, 2005*. 1-2.

ISKIA PROJECTOR, 2006. *Citing references: Imatte* [online]. Available from: www.imatte.com [Accessed 31 August 2006].

KÖCHY, K., KRAUSS, M. AND NEUMANN, P., 1998. Interactive manipulation of realtime visualisation from medical volume data by using 2-handed VR-techniques. *In: Proceedings of the EuroPACS, October 1998 Barcelona*. 221-224.

MEGA DRIVE, 2006. *Citing references: Sega Mega Drive/Sega Genesis Wikipedia* [online]. Available from: en.wikipedia.org/wiki/Sega_Genesis [Accessed 31 August 2006].

MYERS, B.A., 1998. A brief history of human computer interaction technology. *ACM Interactions*, 5(2), 44-54.

NINTENDO WII, 2006. *Citing references: Nintendo* [online]. Available from: www.nintendo.com/home [Accessed 31 August 2006].

OGRE – OBJECT-ORIENTED GRAPHICS RENDERING ENGINE, 2006. *Citing references: OGRE* [online]. Available from: www.ogre3d.org [Accessed 31 August 2006].

OPENCV - OPEN SOURCE COMPUTER VISION LIBRARY, 2006. *Citing references: Intel Corporation* [online]. Available from: www.intel.com/technology/computing/opencv/ [Accessed 31 August 2006].

STERIADIS, C.E. AND CONSTANTINOU, P., 2003. Designing human-computer interfaces for quadriplegic people.

*ACM Transactions on Computer-Human Interaction*, 10(2), 87-118.

TEKKEN, 2006. *Citing references: Tekken Official* [online]. Available from: www.tekken-official.jp/ [Accessed 31 August 2006].

3D STUDIO MAX, 2006. *Citing references: Autodesk* [online]. Available from: www.autodesk.com/3dsmax [Accessed 31 August 2006].