

ARPong: A Viability Case Study of Markerless AR-Based Interface to Computer Games with Webcam

Yuri L. B. Nogueira

Markos O. Freitas

Creto A. Vidal

Joaquim B. Cavalcante-Neto

Universidade Federal do Ceará



Figure 1: ARPong game play.

Abstract

This paper describes the ARPong, a Pong game with augmented reality interaction. It is proposed a method of paddle control using a simple webcam and markerless motion capture based on optical flow. Three algorithms are described to achieve better playability according to the environment, with no need of calibrating process.

Keywords:: Markerless Augmented Reality, Motion Capture, Human-Computer Interface, Computer Vision

Author's Contact:

{yuri,markos,cvidal,joaquimb}@lia.ufc.br

1 Introduction

The human-computer interface is one of the key points on virtual reality design that aims to make applications more immersive through an intuitive interaction, specially on games.

There have been recent efforts on game industry to produce realistic interaction, such as the Nintendo[®] Wii Remote Controller[™], the PlayStation3[®] Move[™], the PlayStation3[®] Eye[™], and the Microsoft[®] Kinect[™].

The well known Nintendo[®] Wii Remote Controller[™] and the PlayStation3[®] Move[™] are hand-held devices developed to track the player's movements using accelerometers, gyroscopes, magnetometer and other sensors. While intuitive on some applications, e.g. sword swinging, tennis or boxing games, these tools are limited to track hand-based moves, and other gadgets are needed to track other parts of the player's body. It is important to mention that video cameras are also used to capture the motion and improve the tracking of the hand-held devices.

The PlayStation3[®] Eye[™] and the Microsoft[®] Kinect[™], on the other hand, use only computer vision in conjunction with one or more cameras to track the player's movements. These tools do not introduce any special apparatus to the player's body, letting him/her free to use his/her whole body to interact with the application.

Although useful for enhancing immersion, these devices may be too expensive for a common user. Moreover, the underlying companies may not provide software development kits for third party games at low cost.

This work proposes a motion tracker using only a single webcam, easy to use and set up, with no need of markers or camera calibration. This is performed by changing the perspective of motion capture, where the software detects and recognizes the player, to a perspective where the user is responsible for reaching and modifying virtual elements.

The proposed method can be used in the rapid development of interactive games. In this work, an augmented reality version of the well known Pong game [Kent 2001] was developed to analyze and validate the proposed technique. In the ARPong game, the players use their own hands to control the paddles, while the ball is physically controlled by a physics engine.

This work is organized as follows. Section 2 reports some work related with human-computer game interface and computer vision. Section 3 presents the ARPong game and Section 4 describes the proposed techniques. Section 5 depicts the conclusion and outlines some future work.

2 Related Work

[Höysniemi et al. 2005] made a study about how intuitive gesture increases immersion in children's games. They emphasize that "intuitiveness and physical appropriateness [...] play a central role in providing a pleasant playing experience". They used the Wizard of Oz technique to identify the children's natural gestures in order to design a better game.

[Tse et al. 2007] also identified the need of intuitive interaction and used a touch-screen tabletop device and speech recognition in strategy and simulation games. Their basic idea is to replace the traditional mouse and keyboard devices for direct game scenario tapping, gestures and speaking commands in order to improve the immersiveness.

[Figueiredo et al. 2009] proposed a framework for hand-position recognition. Their work identifies hands through the recognition of color groups. Specifically, the user needs

to wear yellow gloves, as a colored marker, which usually are the largest objects with that color in the two-dimensional image captured by a webcam. They developed a guitar game to test their framework.

[Bernardes et al. 2009] proposed a technique of hand gesture identification based on skin color that could be used in game interface design. Their calibration step used 120 pictures of hands in different postures as input data to recognition algorithms. Each of those postures is mapped to a different function in an application.

[Laakso and Laakso 2006] suggest a multiplayer game system that uses a top-view camera to recognize body movements and gestures. The system needs a relatively large and clear area, since the body movements and gestures of several players had to be captured at the same time. They used a wall to project the gameplay. They concluded that cooperation, simplicity, intuitiveness, scalability and robustness are essential to a body-driven game design.

In order to avoid the need of a clear background, the work of [Sun et al. 2006] proposes a technique for background cut. Even though designed for general-purpose applications, their work can be applied to enhance the recognition of desired elements on the tracking procedure. Although that technique needs the background image on a calibration step, it is still robust enough to some background changes.

To achieve their specific goals, the works described in this section need special apparatuses, advanced devices, markers, calibration steps, predefined gestures or complex assembly. This work proposes a simple, cheap, markerless, and easy to set up technique for game interaction, using only a webcam and a desktop computer.

3 Game Description

The classic Pong game is a simplified electronic two-dimensional version of a tennis game. There are two paddles, each one controlled by a user, who has to defend his/her side of the scenario, racketing a ball to the opponent's side. In the original game, the paddles only move in the vertical direction by means of a joystick.

The version of Pong described in this work, called ARPong, is an augmented reality game where the paddles and the ball are inserted in the real world. A user controls a paddle using his/her own hand, moving it through his/her side of the scenario in both vertical and horizontal directions. The background of the game is the captured image, so that the players are able to locate the virtual elements in the real world (Figure 1).

To improve the immersion feeling, the ball is physically controlled so that reflection angles and speed are defined by the force and direction the user hits the ball.

4 Proposed Techniques

The basic idea of this work is to invert the motion capture perspective, which tries to detect a player and recognize his/her body parts, to a paradigm where the player is responsible for reaching and moving the virtual elements. In the ARPong game, the paddle is not placed in a user's hand,

he/she has to grab it, which in turn starts to respond to his/her movements.

This is achieved by an optical flow technique, known as the Lucas-Kanade algorithm [Lucas and Kanade 1981]. Optical flow techniques detect directions of movements in videos, analyzing two consecutive frame images. The ARPong game uses a pyramidal Lucas-Kanade algorithm [Bouquet 2000], which provides tracking in a specified region of the image.

The tracking is performed by attaching one point to the center of each paddle. A point is then moved in the direction of the optical flow caused by the hand movement in that region (Figure 2).

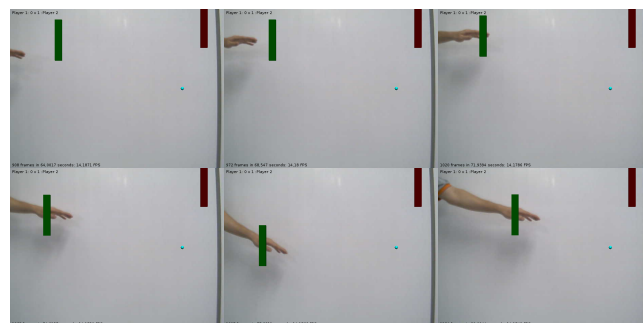


Figure 2: User reaching and moving a paddle.

Some issues have to be addressed in the development of image capture applications, such as the ARPong. The first one is the background, which can be too complex. The second is the captured image, which can be too noisy depending on the camera quality and the illumination.

These problems can confuse the optical flow calculation. Therefore, some prior image processing needs to be performed. Since only relatively wide actions need to be tracked, colors are not important and noises might generate minor undesirable movements. Thus, the first processing is the black and white image conversion along with a Gaussian filter.

That does not solve the whole problem, and three intermediate image processing algorithms were developed, each one fitting better a specific situation of the real environment: Threshold-Diff, Diff-Threshold and Borders.

4.1 Threshold-Diff

The general idea of this algorithm is to identify the objects in the image and to extract the ones that are not moving, which are considered background. That can be achieved by applying a threshold and an absolute difference between frames.

The threshold algorithm is responsible for highlighting the objects by matching shades that are alike. Furthermore, it enhances the contrast of objects with different tones. The tones under a threshold value are turned into 0, whereas tones above it are turned into 255.

The absolute difference algorithm extracts the stationary objects, since they are present in the previous and in the current image frames. The result of this processing is shown in Figure 3.

This algorithm is the most robust in attaching the paddle to

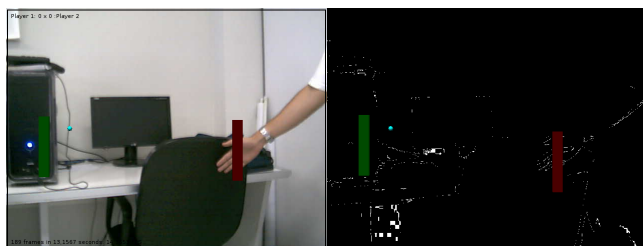


Figure 3: *Threshold-Diff algorithm.*

the hand since it is sensible to moves, but it is also the most affected by noise and, consequently, by undesired movements. Therefore, the Threshold-Diff method is more appropriate to clean backgrounds.

4.2 Diff-Threshold

The previous algorithm has the disadvantage of detecting undesired minor movements, as can be seen in Figure 3. The Diff-Threshold algorithm inverts the order of operations, with the aim of removing noise in pixels whose absolute value of their difference is smaller than the threshold value.

In the Threshold-Diff algorithm, since the threshold algorithm is applied first, small variations values in the same pixel around the threshold value are amplified, and detected as movements when the absolute difference is applied.

By first applying the absolute difference in consecutive frames, only the variation between two values of the same pixel is considered. Thus, the subsequent application of a threshold algorithm removes variations smaller than the threshold value. As a consequence, only wide movements are detected (Figure 4).

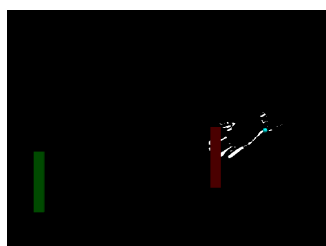


Figure 4: *Diff-Threshold algorithm.*

Despite its qualities, the Diff-Threshold algorithm can discard small desired movements, causing the hand tracking to be lost. This can lead to bad positioning of the paddles on screen when the player stays immobile and suddenly starts to move his/her hand again.

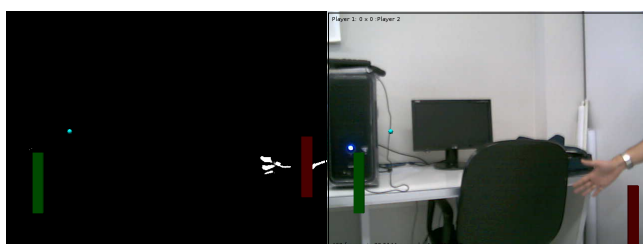


Figure 5: *“Broken” hand and bad paddle attaching on the Diff-Threshold algorithm.*

4.3 Borders

The absolute difference based algorithms may present problems with backgrounds having elements whose grayscale tone is close to the grayscale hand tone. Due to this, the hand can appear “broken” after the image processing, leading to a bad hand-paddle attachment. This is more visible in the Diff-Threshold algorithm (Figure 5).

The Borders algorithm tries to solve that problem by identifying the contours of moving objects. This algorithm uses the Diff-Threshold algorithm to remove the background and to detect the movements, and then tries to induce continuity among the “broken” elements.

After applying the absolute difference and the threshold algorithms, the Canny edge detection algorithm [Canny 1986] is employed to detect the edges of the “broken” elements. After that, the “broken” edges are dilated and their contours are inferred by the [Suzuki and Be 1985] algorithm. The resulting image of the Borders algorithm can be seen in Figure 6.



Figure 6: *Borders algorithm.*

This algorithm solves the “broken” hand problem, but it may cause undesirable paddle shaking movements due to different borders generated by a noisy image. It also inherits the stationary hand problem of the Diff-Threshold algorithm.

Figure 7 summarizes the algorithms and the complete process proposed in this work.

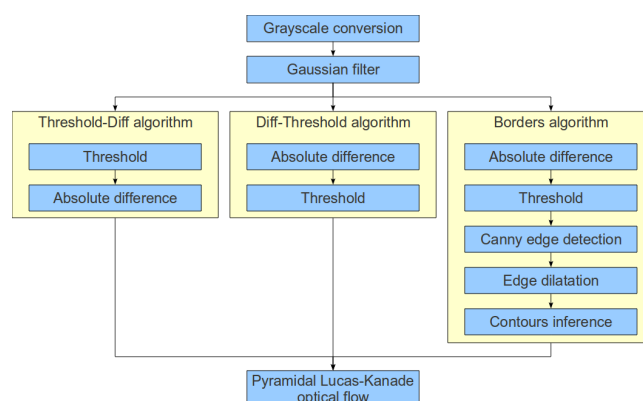


Figure 7: *Flow chart for the proposed algorithms.*

5 Conclusion

5.1 General Considerations

The frame rate of the game was limited by the camera speed, showing that the time for image processing and physical calculations is actually insignificant. This limitation lets the

game playable, but may sometimes compromise the collision detection, since the paddle can pass through the ball in the interval between two frames.

The ARPong game was tested in a clean white wall background and in a dirty environment, both artificially illuminated by fluorescent lamps. The possibility of choosing among the three algorithms provided an adequate gameplay in both environments.

This work showed a way to use markerless augmented reality as a fast, cheap and easy to use input device in game applications. Although applied to a specific case, the ideas presented here can be used in the design of other genres of games.

The ARPong performed satisfactorily, allowing interesting and entertaining matches on all tested environments. The minor game control problems were usually solved by testing among the algorithms the one that best fitted the environment.

The Threshold-Diff algorithm is the one that best attaches the paddle to the hand, and it works best for clean backgrounds. In dirty backgrounds, the paddle may easily get stuck in other surrounding elements, and the player has to reach it to continue playing.

The Diff-Threshold algorithm works fine for both clean and dirty backgrounds. However, since it identifies only wide movements, it loses the hand tracking occasionally. Therefore, this algorithm should be used in environments where the Threshold-Diff did not perform correctly.

The Borders algorithm also works fine for both backgrounds but, since it introduces contours artificially, the paddle might shake around the player's hand. This algorithm usually works on environments where the Threshold-Diff algorithm does not work and is more robust where the Diff-Threshold algorithm is weak. As a counterpart, it generates more undesirable movements, and should be used as the last choice.

Most of the problems experienced happened due to noise, low contrast and chrominance error in the captured images. The noise causes the algorithms to identify movements where actually there are no movements. Low contrast and wrong hue, due to chrominance error, make objects with close tones to be identified as the same object, causing the algorithm to lose track of actual movements. Those issues could be attenuated by using cameras with more adequate sensors.

5.2 Future Work

Despite the good results, all the three algorithms have the limitation of not identifying a move in the direction of the arm, since there is no border movement (grayscale tone variation) in the image. To solve this problem, more studies are necessary in order to identify the hand by its specific patterns, such as shape, color and positioning relative to the body, and not only by environment movements

To smooth the paddle movement and reduce jumps and shakes, an interpolation algorithm could be applied between two direction points identified by the optical flow in consecutive frames. That makes the paddle moves continuous,

enhancing its control by the player and, therefore, the immersion feeling.

Acknowledgements

The authors would like to thank Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES), Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) and Fundação Cearense de Apoio ao Desenvolvimento Científico e Tecnológico (FUNCAP) for the financial support.

References

- BERNARDES, J., NAKAMURA, R., AND TORI, R. 2009. Design and implementation of a flexible hand gesture command interface for games based on computer vision. In *Proceedings of the VIII Brazilian Symposium on Games and Digital Entertainment*, 89–98.
- BOUGUET, J. 2000. Pyramidal implementation of the Lucas Kanade feature tracker: description of the algorithm. In *OpenCV Documentation*. Intel, Microprocessor Research Labs.
- CANNY, J. 1986. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 8, 6, 679–698.
- FIGUEIREDO, L., TEIXEIRA, J., CAVALCANTI, A., TEICHRIEB, V., AND KELNER, J. 2009. An open-source framework for air guitar games. In *Proceedings of the VIII Brazilian Symposium on Games and Digital Entertainment*, 69–75.
- HÖYSNIEMI, J., HÄMÄLÄINEN, P., TURKKI, L., AND ROUVI, T. 2005. Children's intuitive gestures in vision-based action games. *Communications of the ACM* 48, 1, 44–50.
- KENT, S. 2001. *The ultimate history of video games: from Pong to Pokémon*, 1st ed. Three Rivers Press.
- LAAKSO, S., AND LAAKSO, M. 2006. Design of a body-driven multiplayer game system. *ACM Computers in Entertainment* 4, 4.
- LUCAS, B., AND KANADE, T. 1981. An iterative image registration technique with an application to stereo vision. In *Proceedings of Imaging Understanding Workshop*, 121–130.
- SUN, J., ZHANG, W., TANG, X., AND SHUM, H. Y. 2006. Background cut. In *Proceedings of the 9th European Conference on Computer Vision*, 628–641.
- SUZUKI, S., AND BE, K. 1985. Topological structural analysis of digitalized binary images by border following. *Computer Vision, Graphics and Image Processing* 30, 1, 32–46.
- TSE, E., GREENBERG, S., SHEN, C., AND FORLINES, C. 2007. Multimodal multiplayer tabletop gaming. *ACM Computers in Entertainment* 5, 2.