Bringing Context into Play: Supporting Game Interaction through Real-Time Context Acquisition

Radu-Daniel Vatavu
University Stefan cel Mare of Suceava
13 University Street
Suceava 720229, Romania
raduvro@yahoo.com

Stefan-Gheorghe Pentiuc
University Stefan cel Mare of Suceava
13 University Street
Suceava 720229, Romania
pentiuc@eed.usv.ro

Tudor Ioan Cerlinca
University Stefan cel Mare of Suceava
13 University Street
Suceava 720229, Romania
tudor_c@eed.usv.ro

ABSTRACT

We present a new interaction technique that we call Context Interaction and we discuss it in relation with computer games due to their popularity. Although the HCI in gaming benefits of many devices and controllers as well as from many interaction metaphors, they only allow players to control their characters in the game and not the context of the action or the game's environment. The environment change option, if at all supported, may sometimes be carried out in special editing sessions before the actual game begins, e.g. by choosing the track for car racing. We present a simple computer vision technique that allows players to interact with the game environment in real-time and thus to perform Context Interaction. Objects placed on a table are captured by a video camera and transformed into game elements with a real-time feedback within the game. Context Interaction comes as complementary multimodal interaction to the commonly encountered game controllers. It is simple, intuitive and provides real-time feedback.

Categories and Subject Descriptors

H.5.2 [**Information Interfaces and Presentation**]: User Interfaces; I.2.10 [**Artificial Intelligence**]: Vision and Scene Understanding—*Perceptual reasoning, Video analysis*

General Terms

Context acquisition for human computer interaction

Keywords

Context-based interaction, video analysis, game interaction

1. INTRODUCTION

Computer games have always been an important part of the computer industry with an incredible audience and success over the years. The advances in dedicated hardware equipments, computer graphics and artificial intelligence techniques have led to a permanently increase in the realism level of the virtual worlds into which the players are engaging [18].

Equally important, the interaction interfaces and devices through which players are controlling their characters in the game have known a considerable development as well. Apart from dedicated hand held game controllers such as the XBox [1], PlayStation [2] or GameCube [3], other unconventional input technologies are emerging such as gesture acquisition using accelerometer-based devices (the Nintendo Wiimote [3]) or video cameras that make use of computer vision techniques to transpose the natural human body movements into action on the screen (EyeToy [4]). Other non-traditional input such as feet for mobile games [20] or even heart rate based controllers [19] have been equally proposed.

All these devices are designed to capture the users' commands either via pressing buttons on controllers, handling joysticks or interpreting natural gestures. However, users are only allowed to control their virtual characters in the game and are not provided with means of changing the game's environment where the action takes place with the same real-time feedback they are used with. Some games provide separate modules for the players to edit the game scenario (such as modifying closed loop circuits for car racing or changing architecture settings for first person shooter games) but all these options, if available at all, are limited and are all handled during separate editing modules before the actual game begins. There is absolutely no environmental change that can be controlled during the actual game with a real-time feedback of the same kind that the players receive when controlling their characters. For example, imagine a traditional car racing game within a closed loop circuit as illustrated in Figure 1: the way the circuit loops, the curves or the distances between them, these are all options that are either unavailable for the players to set or, if available, the players can only edit them before the race begins. We deal in this paper with this exact option and that is: allowing the players to edit all these environmental aspects of the game in real-time. Imagine that after performing a few racing loops and whilst keep playing, your opponent moves the 3rd curve 100 meters away or simply removes it permanently, all this without taking his eyes from the game. The 1st player could prepare a surprise himself by adding an extra loop at the end of the track or changing the difficulty of the road by adding falling stones or barriers.

MISI'07, November 15, 2007, Nagoya Japan.

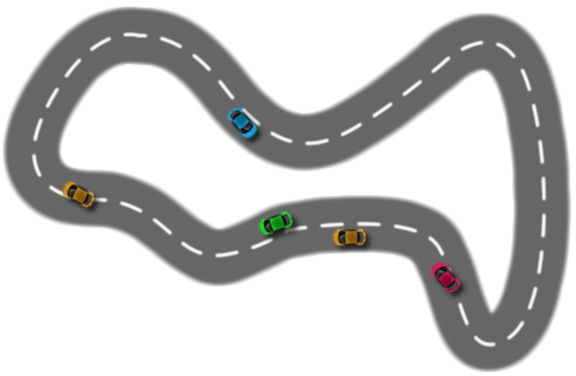

Figure 1: Closed loop circuit for a traditional car racing game.

In order for this kind of environmental interaction to succeed, it has to fulfill two conditions:

- first, the interaction should be done in real-time without any special editing modules;
- second, the interaction should take place during the actual game hence it must be simple, intuitive and not time or attention consuming.

In this paper we present a technique for achieving this kind of environmental interaction by making a correspondence between the real world environment of the user and the virtual environment of the game. In other words, we capture the context of the interaction from the real world and transform it into context in the game. Using this metaphor, we call our new interaction technique *Context Interaction*: a change in the context of the real world triggers a change in the context of the game.

Our vision of Context Interaction is illustrated in Figure 2: users play the game controlling their characters' actions with the same available game controllers, nothing changes here. Meantime, a video camera monitors the surface of a table on which various real world objects may be placed. As the figure illustrates, the camera may be placed on the top of the monitor screen facing down toward the table. The various objects from the table are located in the user's easy reach. Using simple computer vision techniques, the objects are detected and, based on various criteria of discrimination, they are transformed into context elements in the game: a castle or a group of trees in a strategy civilizations-like game; a brick wall that bounces the ball in the classical Pong game; a control point for the track in a car racing game as depicted in Figure 3. By simply changing the position of the objects on the table, the corresponding elements in the game change position as well. Also, adding or removing objects on the table triggers context changes in the game. Positioning, adding or removing objects are simple and intuitive operations which are backed up with an immediate real-time feedback on the screen. These operations can be done without even the user taking his eyes from the screen thus without interrupting the game. Context from the real world transforms into context in the game with a simple hand move. What is achieved in the end is *Context Interaction*.

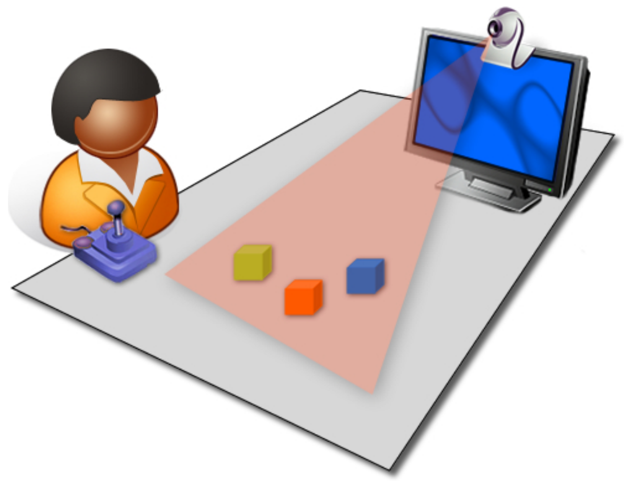

Figure 2: Context Interaction: objects are placed and moved on the surface of a table without the users interrupting their game.

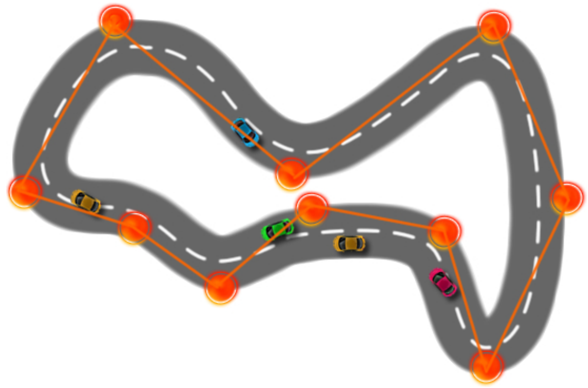

Figure 3: Control points that define the shape of a car racing track in an interpolating-like fashion; the orange control points correspond to x, y positions of real-world objects placed on the interaction table.

The paper is organized as follows: section 2 reviews related work on vision-based acquisition of human input and object detection as well as on augmented reality based games; section 3 describes our technique for interacting with the game environment using the context information as given by the presence of various real-world objects on a table surface; in section 4 we present three simple game implementations: the first one is the classical Pong game which we enhance by adding wall-like objects for the ball to bounce on; the second implementation simulates the construction of a set scenario for a strategy civilizations-like game where players may place and position various objects on a table surface in order to add castles, trees or lakes into the game scenario; the third implementation allows players to control the shape of a track circuit in a car racing game. We discuss several observations of users interacting with our game prototypes, lay our conclusions in section 5 and envisage future work for our context interaction technique.

2. RELATED WORK

Computer vision has the main attraction for HCI the fact of not being intrusive and not requiring users to wear or work with additional equipments or devices. The interaction is thus sought as comfortable and natural as people are used to use their hands to gesture or to interact with objects in the real world. There are shortcomings as well that are related to video processing such as: real-time interaction requires high processing power especially when multiple cameras are involved; dependency on working scenario parameters such as lighting, user skin color, changing background; hands or objects occlusion. There exist good surveys on object tracking [30], human motion analysis [7] or visual gesture recognition [17, 21].

Techniques adopted from computer vision have been used for gaming interaction and construction of intelligent fun spaces for children [9], allowed development of new interaction techniques [14] and served as a base for augmented reality games as new candidates for the gaming technology. Augmented Reality [8] for games is novel and faces several challenges such as localization, communication, cross media augmentation or devices [10]. Previous works made AR outdoor games available such as ARQuake [22] or Human Pacman [12]. Human gestures came into play as well by using techniques for hand tracking [16], trajectory analysis [24] and hand postures recognition [25, 26] often with simple solutions that lead to intuitive interactions [28].

The work we address in this paper makes use of computer vision in order to detect objects or special markers that are placed on a common table. By computing simple features for the detected objects, we transform them into game elements: trees, walls of bricks, curves on a racing track. Our work approaches related developments in table top processing. Tabletops are a newly emerged technology which combines video projection on a common table with vision-based hand and object sensing [27] that allows for intuitive interactions to take place including gaming [29]. We present a new interaction technique based on context acquisition: context in the game adapts in direct correspondence with the real-world context with just a simple operation of a hand move. Our games environments change in real-time according to the presence and placement of the objects on the table surface. Context Interaction is thus achieved.

3. CONTEXT ACQUISITION

Our scenario consists of a simple low-cost web camera that is monitoring a table. The camera may be mounted at a small distance above the table or simply placed on the top of the screen monitor facing down. Video frames are acquired at a resolution of 640x480 with a frequency of 25fps.

A simple background subtraction technique is applied as a first step in order to detect the objects which are located on the table. In order for the approach to work, there has to be a minimum of contrast between the objects and the table. A pixel is classified as foreground if the difference between its value and the stored background value is greater than a given threshold. The approach works well, is low cost and simple enough for our demonstrative scenario although more robust approaches for background subtraction exist in the literature [13]. Using the connected components algorithm [23] objects are identified in the binary resulted image and filtered against size in order to eliminate noise. Figure 4 illustrates the objects detection process: various objects (mug, book, phone, pen, CD) are placed on the surface of the table; the objects that provide a sufficient contrast against the background are successfully segmented while the phone, mostly black, was not detected.

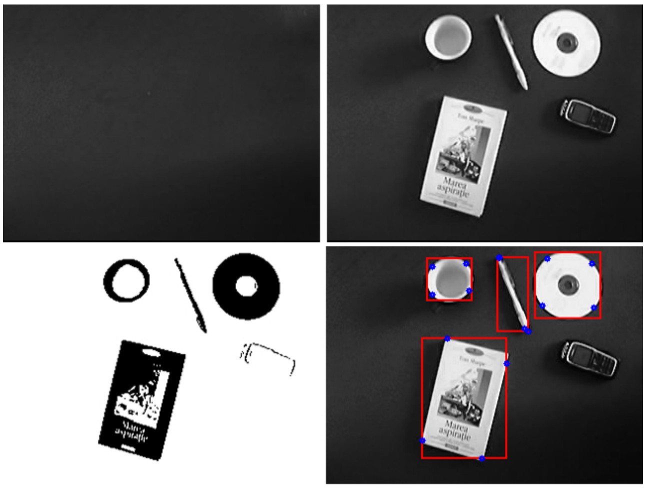

Figure 4: Objects detection on the table surface. Top row: background image and various objects placed on the table; Bottom row: background subtraction result and objects identified.

We detect two types of objects as they are placed or moved across the table's surface: real-world objects and black and white cardboards. The difference between these two object types is that the real-world objects are not interpreted (e.g. no difference is made between a coffee cup or a book that are placed on the table other than their relative size) while the cardboards may be semantically discriminated using their size or pattern model. Figure 5 illustrates a few cardboards that we use in our implementation. There are several reasons why we adopt this multiple-objects approach. First, we use real-world objects as they are easy-to-reach (may be anything: a pen, a book or a coffee cup) and don't need special requirements other than to provide a sufficient contrast with the table color. However, there is no discrimination between them but only at a size or shape level, i.e. small, big, rectangle-like, circle-like or irregular shape. They may be used in Context Interaction where their actual semantic meaning is not necessary for the purpose of the game: they may simply be transformed into walls of bricks that make the ball bounce in the Pong game or they may act as control points that define the shape of a circuit track in a car racing game. On the other hand, the cardboards may be printed with simple 1/0 encoding (black square/white square) patterns that can be recognized as codes and associated with specific game elements: trees, houses or fortresses when setting up the scenario of a strategy game. The patterns may be easily recognized by counting the black squares although more robust approaches as in [15] and open source libraries are available [5] that deal with complex patterns.

A special problem is caused by the user's hands while ma-

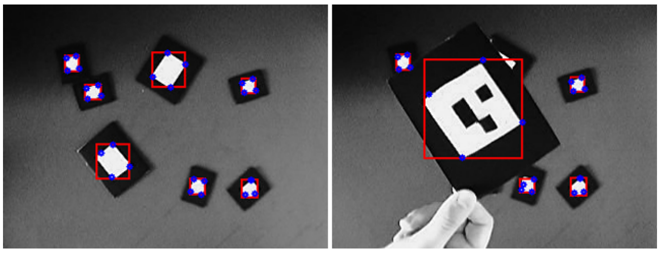

Figure 5: Black and white cardboards. Left: simple cardboards with black border and white center; Right: zoomed view of a pattern marked cardboard.

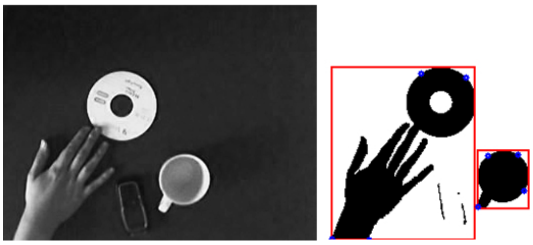

Figure 6: Hand detection and hand-object merging during objects segmentation.

nipulating objects on the table. As the processing is carried out on each video frame, the user's hand is likely to be segmented as an object while it is above the table. Even more, the hand may merge with an existing object as Figure 6 illustrates. These situations may lead to the following outcomes in the game: either the hand is detected as an object and transposed into the game context; either the hand and the object are filtered out due to their size which makes the final hand-object invisible for a small period of time in the game (whilst the hand is touching it). However, this only applies for the simple case of the real-world objects. The fact that our black and white cardboards have black margins creates a small gap between the color of the user's hand and the white center of the boards as Figure 7 shows. Such cardboards design represent a simple solution to an otherwise hard and error prone segmentation problem.

In the end, the objects either real-world or black and white cardboards are segmented and features may be computed for each of them. Such features include size and simple shape estimations for real-world objects (such as circle-like or rectangle-like) and recognition of the printed pattern for

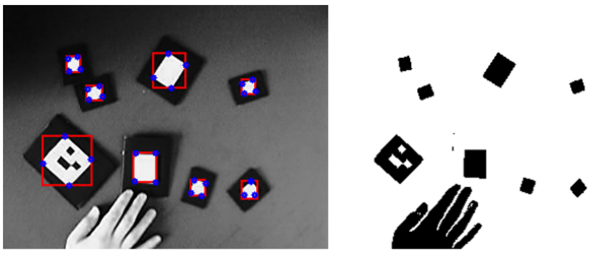

Figure 7: Successful segmentation of a cardboard even with the user's hand touching it.

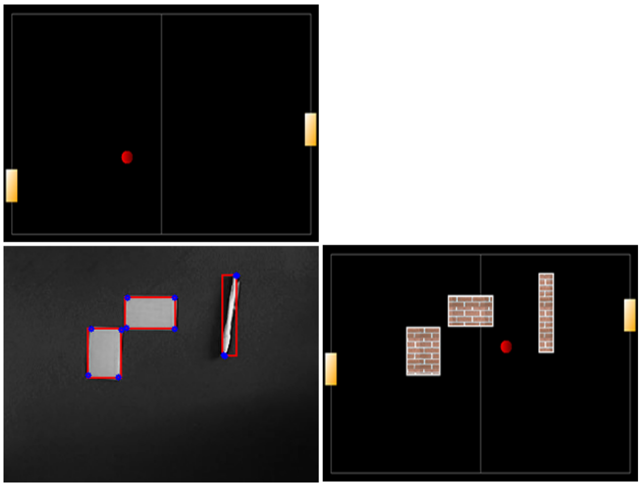

Figure 8: The Pong game. Top row: classical Pong game; Bottom left: various objects placed on the table; Bottom right: Pong game enhaced with Context Interaction where the objects transform into walls of bricks that bounce the ball.

the cardboards, if available. These features may be interpreted according to the particular game scenario which makes the objects transform into game elements or influence the game environment. Sample game prototypes are discussed in the next section.

4. SAMPLE PROTOTYPES

We further present and discuss a few implementations of our Context Interaction technique: the classical Pong game; a setup scenario building for a strategy game that we named the Dragons Castle and a car racing game. We discuss what types of objects are suited for each implementation ranging for every day real-world objects for the Pong game due to its simple nature up to coded cardboards for the strategy game.

4.1 Reviving Pong

Pong is a classical game [6] that we enhance with context interaction. In the classical game, two players control each a palette on the left and right sides of a board trying to return the ball toward the opponent. The context information that we bring in is represented by walls of bricks which correspond to objects being placed and moved on the table surface. A snapshot of Pong is presented in Figure 8.

Pong is a simple game and we chose it to demonstrate the fact that there isn't always the need for discrimination between real-world objects in order for them to be included as game elements. In this case, each object is transformed into walls of bricks that make the ball bounce. The only feature we compute is the size of the objects in terms of $width \times height$ so that the corresponding game walls are proportionally sized.

4.2 The Dragons Castle

The Dragons Castle is a simulation of a strategy civilizations-like game where players may modify the game environment

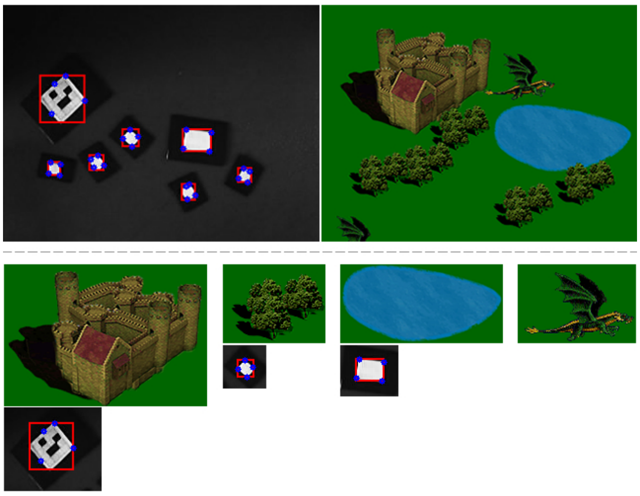

Figure 9: The Dragons Castle game. Top row: objects detected on the table and corresponding elements in the game; Bottom row: correspondence between markers and game elements (castle, trees, lake); the dragons are characters in the game and don't have corresponding objects.

by adding, removing or repositioning elements such as castles, trees or lakes. In this case, discrimination is required between the objects placed on the table hence we use the black and white cardboards. Discrimination is handled using size and pattern codes. Figure 9 illustrates a snapshot of the game as well as the game environment elements (castle, lake, trees) and their corresponding real-world objects. The size of the objects is a good criteria as it directly relates to the actual game elements and common sense judgments such as a lake is bigger than a tree, etc.

4.3 Car Racing

In the Car Racing game, the players may influence the shape of the circuit track on which the cars are competing. Objects placed on the table, either real-world or black and white cardboards, are converted into control points that interpolate the circuit track. We used a Catmull-Rom spline [11] as interpolating curve with a minimum of 3 control points required. As with the Pong game, there are no special requirements regarding the objects that may be used as they only represent control points. Moreover, the restrictions are even less than in the case of the Pong game where the objects sizes were proportionally depicted in the sizes of the brick walls. Figure 10 illustrates the Car Racing game. The changes in the circuit track shape (curves, straight paths) are performed in real-time with an immediate feedback on the screen.

4.4 Discussion

We have presented our game prototypes to several users and although we don't have a thoroughly evaluation of the Context Interaction technique, we drawn several ideas from our direct discussions with them. First of all, there is a general first impression of excitement and surprise - how is that done? - sort of fashion. Otherwise, they find the interaction simple and intuitive as they have the immediate feedback on the screen.

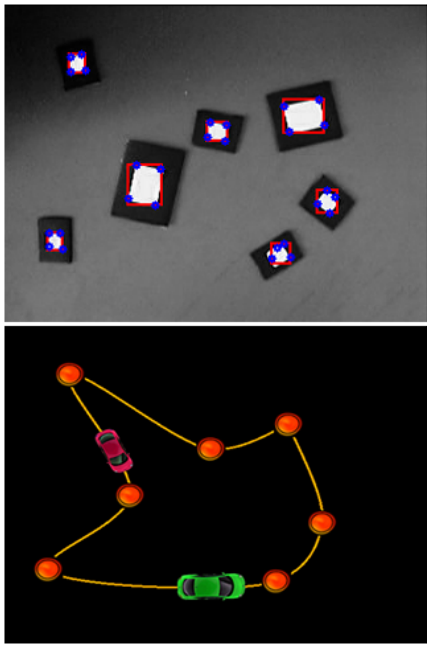

Figure 10: The Car Racing game. Objects detected on the table correspond to control points that determine the shape of the racing track.

The Car Racing game was the most popular and users found changing of the racing track shape very simple and intuitive. Pong was popular as well but for another reason actually as users tried to combine normal game interaction, i.e. the up/down movement of the palette, with context interaction by trying to reflect the ball before it actually got to their palettes. This soon changed into the users leaving the up/down palette controlling keys and trying to bounce the ball with the objects on the table which sort of defeated our context interaction purpose but it showed that they adopted context interaction as a suitable alternative.

We encountered several issues as well. One of them was related to the fact that the context interaction area on the table was not physically marked by any sort hence the users had trouble placing the various objects close to the margins of the video camera view. Objects that were outside the camera view even by a small distance were not detected hence they got removed from the game. Actual delimitation of the interaction area on the surface of the table is a feature that needs to be catered for. Other issues were caused by occlusions: while aiming for an object on the table, the user's hands may partially or totally occluded other objects which either made their size change or they were removed from the game for the short time while the occlusion took place.

5. CONCLUSIONS

We described in this paper Context Interaction, a new interaction technique that comes as complementary to other existing interaction paradigms and devices. We apply our technique to the special application case of video games due to their popularity among computer users. Context Interaction proves simple, intuitive and is done with real-time feedback. Moreover, Context Interaction is carried out independently of the actual game interaction: users may modify the game environment by a simple object move operation on the surface of the table without pausing the game.

Our technique depends on computer vision algorithms in order to detect objects that are being placed or moved across the table. A drawback we encountered during implementation was caused by erroneous detection when the user's hand was touching the objects because that led to the objects being filtered out from the game. We overcame this problem using black and white cardboards with a black border. Moreover, by introducing the cardboards we showed the possibility of using pattern coded markers.

Future work will address enhanced segmentation techniques in order for the user's hands to also come into play as our prototypes experiments showed that this would be beneficial for the interaction process. Other application scenarios besides games may reveal as interesting investigation as well.

6. ACKNOWLEDGMENTS

The work presented is supported by INTEROB CEEX-131 2006 funding grant and the AUF Bourse 1021FR58ML/2006.

7. REFERENCES

[1] Microsoft Corp. XBox, www.xbox.com (Aug. 2007).

[2] Sony Playstation, www.playstation.com (Aug. 2007).

[3] Nintendo. www.nintendo.com (Aug. 2007).

[4] Sony Inc. EyeToy, www.eyetoy.com (Aug. 2007).

[5] ARToolKit, http://artoolkit.sourceforge.net (Aug. 2007).

[6] PONG-Story, www.pong-story.com (Aug. 2007).

[7] J. K. Aggarwal and Q. Cai. Human motion analysis: a review. *Comput. Vis. Image Underst.*, 73(3):428–440, 1999.

[8] O. Bimber and R. Raskar. Modern approaches to augmented reality. In *SIGGRAPH '05: ACM SIGGRAPH 2005 Courses*, page 1, New York, NY, USA, 2005. ACM Press.

[9] A. Bobick, S. Intille, J. Davis, F. Baird, C. Pinhanez, L. Campbell, Y. Ivanov, A. Schutte, and A. Wilson. The kidsroom: A perceptually based interactive and immersive story environment. *Presence, Teleoperators and Virtual Environments*, 8(4):367–391, 1999.

[10] W. Broll, J. Ohlenburg, I. Lindt, I. Herbst, and A.-K. Braun. Meeting technology challenges of pervasive augmented reality games. In *NetGames '06: Proc. of 5th ACM SIGCOMM Workshop on Network and System Support for Games*. ACM Press, 2006.

[11] M. D. Carmo. *Differential Geometry of Curves and Surfaces*. Prentice-Hall, 1976.

[12] A. D. Cheok, K. H. Goh, W. Liu, F. Farbiz, S. W. Fong, S. L. Teo, Y. Li, and X. Yang. Human pacman: a mobile, wide-area entertainment system based on physical, social, and ubiquitous computing. *Personal Ubiquitous Comput.*, 8(2):71–81, 2004.

[13] A. M. Elgammal, D. Harwood, and L. S. Davis. Non-parametric model for background subtraction. In *ECCV '00: Proc. of the 6th European Conf. on Computer Vision-Part II*, London, UK, 2000.

[14] J. Höysniemi, P. Hämäläinen, L. Turkki, and T. Rouvi. Children's intuitive gestures in vision-based action games. *Commun. ACM*, 48(1):44–50, 2005.

[15] H. Kato and M. Billinghurst. Developing ar applications with artoolkit. In *ISMAR*, 2004.

[16] M. Kolsch and M. Turk. Fast 2d hand tracking with flocks of features and multi-cue integration. In *CVPRW '04: Proc. of the 2004 Conf. on Computer Vision and Pattern Recognition Workshop*, Washington, DC, USA, 2004. IEEE Computer Society.

[17] J. LaViola. A survey of hand posture and gesture recognition techniques and technology. In *TR CS-99-11, Brown University*, 1999.

[18] D. Marshall, T. Ward, and S. McLoone. From chasing dots to reading minds: the past, present, and future of video game interaction. *Crossroads*, 13(2), 2006.

[19] V. Nenonen, A. Lindblad, V. Häkkinen, T. Laitinen, M. Jouhtio, and P. Hämäläinen. Using heart rate to control an interactive game. In *CHI '07: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 853–856, New York, NY, USA, 2007. ACM Press.

[20] V. Paelke, C. Reimann, and D. Stichling. Foot-based mobile interaction with games. In *ACE '04: Proceedings of the 2004 ACM SIGCHI International Conference on Advances in computer entertainment technology*, pages 321–324, New York, NY, USA, 2004. ACM Press.

[21] V. Pavlovic, R. Sharma, and T. Huang. Visual interpretation of hand gestures for human-computer interaction a review. *IEEE Trans. on PAMI*, 19(7), 1997.

[22] W. Piekarski and B. Thomas. Arquake: the outdoor augmented reality gaming system. *Commun. ACM*, 45(1):36–38, 2002.

[23] W. K. Pratt. *Digital Image Processing, 3rd ed.* John Wiley & Sons, Inc, New York, 2001.

[24] R.-D. Vatavu, L. Grisoni, and S.-G. Pentiuc. Gesture recognition based on elastic deformation energies. In *Proc. Gestures Workshop, Portugal*, May 2007.

[25] R.-D. Vatavu, S.-G. Pentiuc, C. Chaillou, and L. G. abd Samuel Degrande. Visual recognition of hand postures for interacting with virtual environments. In *Proc. Int.Conf.DAS, Romania*, May 2006.

[26] A. Wexelblat. An approach to natural gesture in virtual environments. *ACM Trans. Comput.-Hum. Interact.*, 2(3):179–200, 1995.

[27] A. Wilson. Playanywhere: A compact tabletop computer vision system. In *Symposium on User Interface Software and Technology (UIST)*, 2005.

[28] A. Wilson. Robust vision-based detection of pinching for one and two-handed gesture input. In *Proc. ACM UIST*, 2006.

[29] A. Wilson and D. C. Robbins. Playtogether: Playing games across multiple interactive tabletops. In *IUI Workshop on Tangible Play: Research and Design for Tangible and Tabletop Games*, 2006.

[30] A. Yilmaz, O. Javed, and M. Shah. Object tracking: A survey. *ACM Comput. Surv.*, 38(4):13, 2006.