On the Design and Player Satisfaction Evaluation of an Immersive Gestural Game: The Case of Tortellino X-Perience at the Shanghai World Expo

Marco Roccetti
Computer Science Department
University of Bologna
Mura A. Zamboni 7
40127, Bologna, Italy
roccetti@cs.unibo.it

Angelo Semeraro
Computer Science Department
University of Bologna
Mura A. Zamboni 7
40127, Bologna, Italy
semeraro@cs.unibo.it

Gustavo Marfia
Computer Science Department
University of Bologna
Mura A. Zamboni 7
40127, Bologna, Italy
marfia@cs.unibo.it

ABSTRACT
The role of human-computer interaction technologies has become a prominent factor that most can determine the successful introduction of new computer games. Players, in fact, wish to experience playful exchanges with the objects and characters that compose games. To reach this aim, new technologies have come into the play that comprise the use of video cameras and gesture recognition software. The great news is that such types of technologies could be exploited not only while playing at home on a console, but also in public spaces, thus broadening the use of games to new segments of customers. Nonetheless, to the best of our knowledge, neither relevant exemplars of such specific type of games (that can be played in public spaces) have yet emerged, nor extensive measurement studies exist regarding how players enjoy games in public immersive environments (e.g., fairs, museums and exhibits), the motivation being that those technologies that support completely hands-free gaming have been commercialized only very recently. Hence, our contribution with this article is twofold: on one side we want to illustrate the main design principles we have devised to design a gestural game to be played in a public space, based on novel hand following and gesture recognition techniques. On the other side we wish to report on real measurements we took when over one hundred players enjoyed our gestural game at the Shanghai 2010 World Expo.

Categories and Subject Descriptors
H.1.2 [**Models and Principles**]: User Machine Systems – *human factors, human information processing.*

General Terms
Design.

Keywords
Hands free gaming, player experience, immersive environments, gestural gaming, Shanghai World Expo.

SIGDOC'11, October 3–5, 2011, Pisa, Italy.

1. INTRODUCTION
Major producers of game consoles are engaged in a technological race that aims at providing their respective customers with the most exciting and natural human-computer interfaces. Recent trends show that the more realistically a game can be played, the higher will be its chances of beating its competitors in the market share arena. This means that customers are no longer happy of holding joysticks, pressing keyboard buttons or sliding mice, but want to be able to play games where they perform the same natural body movements that would be performed in reality.

The first innuendos that have raised the expectations of computer players have been caused by the experience of playing with the Nintendo Wii, where realistic human body movements are supported by the combination of infrared and accelerometer technologies. The Wii has succeeded in letting players move as they were playing on a real field, especially in the case of games that required them swinging an arm, such as tennis or bowling. A further step forward has been moved with the launch of Kinect, an advanced video camera and software system that supports human body recognition on the Microsoft Xbox console. In Kinect-supported games, players can move and control their avatar, which mimics their movements, just as dreamt by many generations of youngsters.

This comes at a cost: the use of non-cheap, advanced video cameras, namely *depth sensing* cameras, which, combined with infrared sensors, are capable of estimating the distance from the real objects that are captured in real-time, thus adding 3D information to 2D images.

Besides playing at home on a console, a new alternative trend is emerging where a crowd of players joins a specific open area (we refer to it as the *gaming arena*), while waiting for their turn to have fun with a game. When a player enters the gaming arena, he/she faces a wall screen (or any other immersive display technology) where a virtual gaming world is presented. Then, a video camera is placed above the arena to record his/her movements, with particular attention to the location where arms and hands are waved. It is worth noticing that such type of settings is attracting an increasing consideration as it can be deployed within public spaces, where ludic exhibitions take place (think of museums, fairs and theatres, for example).

However, although *hands-free* gaming has been successfully introduced at home, neither relevant examples exist of using such technologies in public spaces, or open areas, probably due to the

novelty of such approach, nor player results have been yet publicly made available gathered from extensive sets of measurements taken within immersive gaming environments.

Within this framework, the contribution of this work is then twofold. First, we want to describe and discuss the process that led us to develop some novel hand following and gesture recognition techniques that can be easily applied to develop games to be played in public spaces. This latter aspect, in some sense, extends the scope of our work and brings an important contribution to the DOC (Design Of Communication) field, as our techniques could be more generally exploited to provide a technological support to all those performing events thant can be enjoyed publicly, where a predefined set of gestures need to be automatically recognized to permit a natural experience to customers/players. More technically speaking, we specifically developed a set of new recognition algorithms that have revealed the good property to be robust and easy to implement. For this reason, these algorithms are particularly suited for public gaming scenarios where settings can often change or adapted to new requirements or needs. [2], [3].

Second, with this paper we wish also to illustrate how over one hundred players enjoyed at the 2010 Shanghai World Expo our game *Tortellino X-Perience*, a *hands-free* game (implemented based on our algorithms) which teaches how to prepare the famous Tortellini Pasta. In particular, we provide both survey results summarizing how a large set of people, of all ages, enjoyed playing the game, and also experimental results from technical tests performed on our software. Although we neither provide nor discuss here a new methodology for measuring player's behavior in these new challenging contexts of immersive gaming environments, nonetheless we have supplied a contribution in terms of a thorough analysis of such games, through the exemplar of *Tortellino X-Perience*. Moreover, as this game was deployed in a very realistic, as well as environmentally complex, setting, we feel confident that the data we got may be of use for future studies and the development of principled theories on this subject.

The rest of this article is organized as follows. In the second Section, we provide a succinct overview of some alternative methods. In the third Section we describe the gaming scenarios we are interested in along with the algorithms we implemented in our system. In the fourth Section, we report on the use of our system at the 2010 Shanghai World Expo [1]. The fifth Section concludes this article.

2. BACKGROUND ON GESTURE RECOGNITION

In the past decade, a wealth of research has been devoted to finding new and more natural means of interaction between humans and computers. Here we simply provide a succinct overview on those technologies that have seen a widespread adoption in popular gaming consoles, while omitting other intriguing proposals that do not have any commercial counterpart [2].

A famous controller capable of supporting the tracking of a player's movements is the Nintendo Wiimote [3]. The Wiimote is equipped with an infrared camera sensor, which provides high speed tracking of up to four infrared light sources.

This infrared camera computes the distance of the Wiimote from two light emitting sources placed within the sensor bar (typically positioned above or below the TV set). The Wiimote transmits such information to the console via a Bluetooth interface and also carries an accelerometer. Seen from a different perspective, a Wiimote acts as an intermediary between the gestures a player performs and its console.

Recently, after a long marketing process that has started from the presentation of the Natal Project and finally ended with the commercialization of the Kinect sensor, computer game players have been able to enjoy body free gaming with Microsoft Xbox. Kinect is a horizontal bar that is placed either above or below the TV set. However, it comprises a specialized depth sensor in addition to the RGB camera, which is capable of recording the distance of all objects that lie in front of it [4]. This information is then processed by a software engine, which extracts human body features of players, thus enabling the interaction between physical and virtual worlds.

Sony, before Microsoft, offered to its customers mixed reality experiences with the Playstation Eye. This product is based on a digital camera that feeds the captured video stream to software algorithms, designed to infer game input commands, which implement edge detection and color tracking techniques to detect movements that should be performed only in correspondence of specific given areas. Although millions of the described consoles have been sold around the world, little or no knowledge is known concerning the opinions of gamers on *hands-free* gaming experiences.

3. ON THE DESIGN OF A GESTURE RECOGNITION SYSTEM FOR GAMES IN PUBLIC SPACES

Our hand gesture recognition system was devised to work within an immersive gaming scenario, where a game is played within an open space (termed *gaming arena*), with each new player who can join every few minutes. We have in mind not a typical console but a situation where a queue of players is waiting on his/her turn to play within that public space, like during a visit to a museum, or to a fair or to any other exhibit event.

Playing essentially amounts to waving hands within the arena, while watching a wall screen that displays the game graphical environment. A video camera is placed above and hands are detected as those motion patterns that reach farthest away from the player's body. This may be achieved, for instance, also having a player leaning over a desk, which may further represent a restriction of the gaming arena.

The goal is that of recognizing gestures performed by hands that move freely above a given surface (e.g., a tabletop or simply the floor), and below a video camera that captures frames. We here discuss and report on the process that has led us to devise a system able to recognize players gesture within this context.

In particular, in the next Subsections we will provide details about the series of three different algorithms on which our system is based. As a preliminary comment (confirmed by the experimental results we reported at the end of this paper), we wish the reader notice that all the devised algorithm are robust enough and easy to implement. This makes them particularly suited for public gaming scenarios where settings can often change or adapted to new requirements. Each of them was implemented in Java.

To anticipate how all this work, it is worth considering that the first of these algorithms computes the luminance differences between the static scene before the hands come into play and the scene after that the player has begun to move his/her hands. Such

process leads to the identification of those specific areas that the player traverses while waving his/her forearms and hands.

Once individuated, the second algorithm filters out those areas to identify the farthest positions reached by a player's hands (termed *extreme points*). This has been done considering that the games of our interest require a player to stretch out his/her hands in front of him/her, and hence a hand can be significantly identified with the point that reaches farthest away from the player's body.

A third and final algorithm recognizes gestures while tracking those extreme points, based on the consideration that each movement a player performs involves a starting zone from which a hand commences its movement, a trajectory to be followed, and finally an ending zone where the hand completes its gesture. We describe each algorithm in detail. A comprehensive representation of the three algorithms is provided in Figure 1.

3.1 Algorithm 1: Detecting Hands

This algorithm computes the luminance difference between the initial state of the surface below and the current frame as captured by the camera above the hands.

At the very initial stage, before anyone plays, but not with each new player, a very fast calibration phase is performed. Such phase involves dividing the underlying surface rectangle into an N x M grid of blocks, and storing the maximum and minimum RGB luminance values for a subset of K pre-defined pixels within each block.

N and M are chosen large enough to provide the granularity sufficient to detect a typical hand with its arm, whose width is about 5 centimeters. K, instead, represents a trade-off between the increasing computation effort required utilizing larger values and the decreasing detection accuracy deriving from the use of smaller ones (from our experiments, the values of $N = M = 9$ and $K = 4$ has given optimal results in terms of both accuracy and speed).

Then, for every chosen pixel p in a given block, the minimum and maximum luminance values (min_p, max_p) are taken, after an observation lasting for a sufficient, but short, amount of time. This eliminates any problem due to small luminance variations that are possible, even in settings subject to constant illumination.

After that, with each new frame, the algorithm checks for the presence of a hand above any of the K pixels of each block. First, a low-pass Gaussian filter is applied to smoothen out any color peak due to small object movements and/or random brightness variations, providing as a result a filtered video frame.

Second, for each block, a check is done to verify if the luminance value of each of the K pixels p lies within the $[min_p, max_p]$ interval of reference. If that luminance value is greater than max_p, or smaller than min_p, the pixel is considered as changed. With this comparison, the number of pixels that changed is counted for each block. If this number exceeds a given threshold (e.g., 0.75 times K), then the entire block is considered as changed.

However, to be sure that the area comprised of changed blocks contains a recognizable hand, all those changed blocks must be adjacent (two blocks are pair-wise adjacent if they share an edge, or even a vertex).

To this aim, the final part of this algorithm checks whether all the changed blocks represent an area, termed *active area*, comprised of all adjacent blocks, without any form of interruption between blocks. Only in such case, and at the end of this process, we can conclude that that active area represents a player's hand. The top part of Figure 1 delivers a pictorial representation of all the process we have described.

3.2 Algorithm 2: Following Hands

Once the active areas with the two hands have been identified, the problem is to find an efficient mechanism to follow those hands.

Achieving such result means determining one relevant point to follow for each of them. Some authors, in similar cases, choose the center of mass; we, instead, as relevant point to follow for each hand, have chosen the already mentioned *extreme point*.

The motivation for this is that the extreme points are those that reach the farthest positions of a player's hands, while waving them freely in the air. A large class of games, in fact, entails that a player extends his/her hands in front of him/her, and hence a relevant information for following, and then for recognizing, gestures is concerned with that precise point that reaches farthest away from the player's body. Finding both the extreme points (right and left hands) can be performed utilizing the following algorithm.

We start by mapping the underlying rectangular surface, above which hands are waved, into a Cartesian coordinate system. In this coordinate system a unit of length is taken as the linear dimension of a block. The y and x axes have their origin at the bottom leftmost corner of the surface, where y points externally away from the player's body, while x points towards the right side of the surface, as shown in the bottom leftmost part of Figure 1.

In the first step of this algorithm, each active area is transformed into a bar chart. More precisely, for any given block correspondent to a point along the x axis, a bar is inserted into the chart, whose height y equals the distance of that point from the player's body (for a given x, bars of lower height are replaced by bars of higher heights). Simply said, as shown in the leftmost chart of the bottom leftmost part of Figure 1, each bar represents the highest y value (for each x body's position), where a hand, or a part of it, was detected.

The second step of the algorithm (that is, determining the coordinates of the extreme point of the first hand) is a simple task, as that point is simply given by that bar with the maximum y value among all the represented bars in the chart.

For instance, in the aforementioned leftmost chart, the extreme point of the first hand is detected at position $x = 12$, with value $y = 7$.

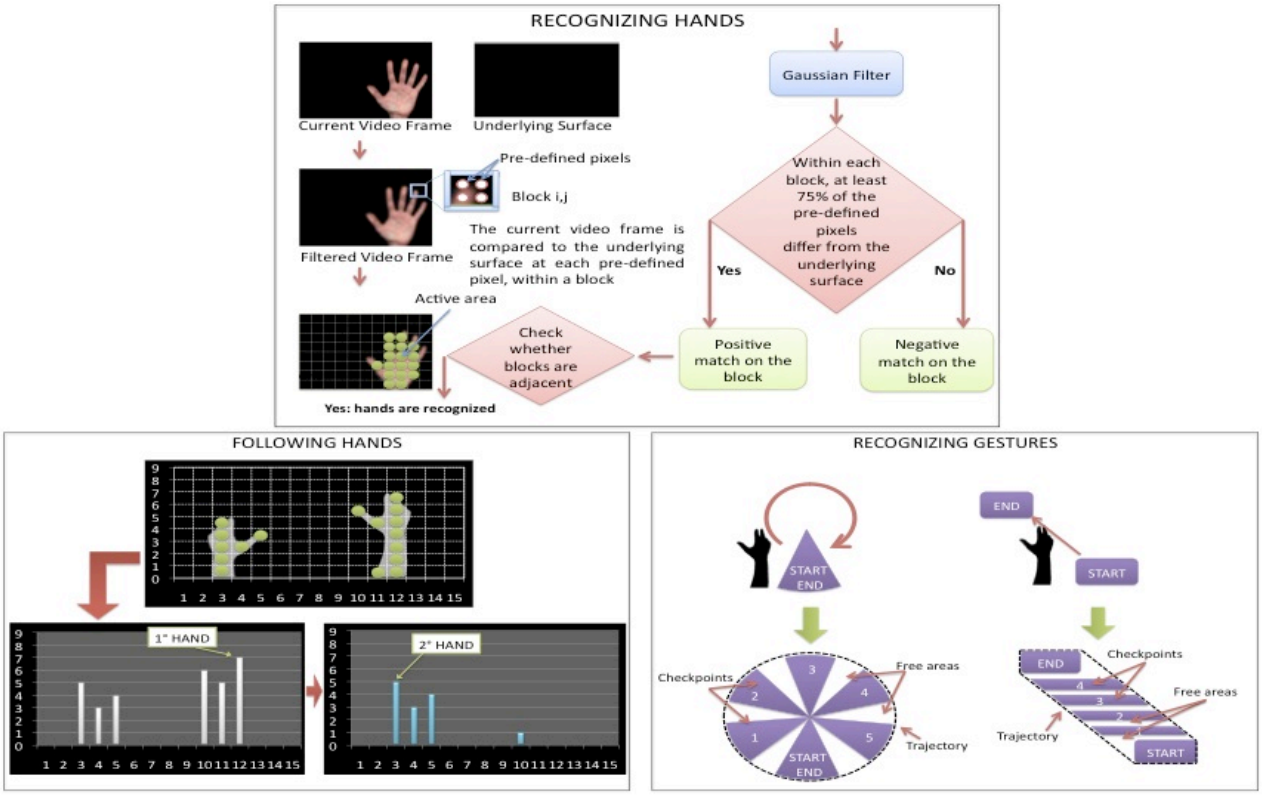

Figure 1. Algorithms for detecting hands, following hands, recognizing actions (anticlockwise from top).

Trickier is the problem of finding the extreme point of the second hand, as it cannot simply be individuated in correspondence with the position where the second (or third) highest y value is found in the chart. These, in fact, could correspond to local maximum values that are representations of lower points of still the first hand. In essence, we are seeking for another maximum y value, while excluding all those bars that still pertain to the first recognized hand.

Thus, with the aim of filtering out all these possible *fake* maximum values, our algorithm constructs a new bar chart, based on the first one, where each new bar is created as follows. The height (y) of each bar of the old chart is contrasted with a so-called *minimum* (whose value is progressively updated).

We start with the next bar at the left (or at the right) of that for which the global maximum was found. If the difference between the height of this current bar and the *minimum* is positive, then a new bar is inserted in the new chart, whose height equals the value of this difference.

If the difference is negative, instead, the bar is not imported in the new chart. Obviously, the *minimum* represents the y value of the lowest bar encountered so far, and is updated as long as the search proceeds, with each new bar under consideration.

This methodology applied to the leftmost chart of Figure 1 yields the correspondent rightmost chart of the same Figure.

In fact, if we move along the direction of decreasing x values, we obtain a new chart where at position $x = 11$, we have $y(11) = max(5 - 7, 0) = 0$, as the minimum y value encountered so far corresponds to the value of the global maximum, that is 7. After that, the new minimum y value is updated to 5, and hence for the position $x = 10$, we obtain a new bar in the new chart, whose y value is equal to $max(6 - 5, 0) = 1$. Again, at the position $x = 5$, the minimum value computed until that point is 0 and hence $y(5) = max(4 - 0, 0) = 4$. If we iterate this until the entire old chart is completely scanned, a new chart is produced with $y(4) = 3$, $y(3) = 5$, respectively. On the new chart, a new global maximum can be searched. That new maximum y value corresponds to the extreme point of the second hand. In our example, it is identified at the position $x = 3$. With the extreme point of each hand, we can now follow hands and their movements.

3.3 Algorithm 3: Recognizing Actions

To recognize actions, we exploit the fact that each gesture requires the player move his/her hand between an origin and a destination, along a given trajectory connecting the two areas.

Moreover, the action of following a given trajectory must be done within a given time period, after which that movement has been performed too slowly to be considered correct.

This is true for all the movements that a player is required to perform within a very large gamut of games where hands can be moved freely.

Hence, the idea behind our algorithm is that of tracking the extreme point of each hand, while verifying that this point starts

its movement from the origin, completes it in correspondence with the destination, and traverses a set of checkpoints set along the chosen trajectory (see the bottom rightmost part of Figure 1).

Therefore, each trajectory followed by an extreme point is recognized as correct if it flows within a stripe of a given size (in essence, a certain degree of tolerance is allowed).

This mechanism was devised to make our algorithm able to consider as correct a wider set of movements with slightly different trajectories that differentiate only for a few geometric differences. This scheme can be applied to trajectories of either linear or circular shape.

4. PLAYING WITH TORTELLINO X-PERIENCE IN SHANGHAI: GAME RULES AND USER SATISFACTION

We developed an educational cooking game where the player is instructed on how to cook a typical Italian recipe: the Tortellini Pasta. The cooking steps the player has to imitate using hands are as follows:

- Initially, we have an empty cooking board, with flour and eggs at its right hand side. The player is required to bring the ingredients at the center of the board;
- The yolks and the albumen of the opened eggs float squeezed within the flour. The action required is to use hands to knead the ingredients, adding some water, ending up with a smooth ball of dough;
- The ball of dough needs to be rolled out with a rolling pin;
- A thin foil of dough is lying on the cooking board. It needs to be cut into squares, using a cutting wheel;
- Each square of dough has to be stuffed with meat and cheese;
- The pasta has to be closed to obtain a Tortellino. This is done by joining different pairs of opposite angles of the pasta;
- The Tortellino is ready to be cooked.

Figure 2, 3 and 4 below illustrate the real case of use of our game during the Shanghai World Expo [7], following some of the rules we have just described. We now supply an analysis of such game, which, thanks to the environmentally complex setting where it was deployed, provided us the opportunity of collecting user experience information from heterogeneous groups of players. To this aim, we here present the most prominent result coming from our Chinese experience. We supply two types of data: the first type is concerned with the quantitative behavior of our system when stressed under the pressure of a continuous queue of customers wishing to play.

Table I. Tortellino X-Perience speed results.

Average Delay	Number of Experiments
< 20 ms	103
> 20 ms	4

The second type of results, more traditionally take qualitative measurements of the degree of satisfaction of players during their gaming sessions.With reference to the first type, we first checked the speed and robustness of our system. A first set of experiments assessed the speed of the cascade of our algorithms, measuring the time they took to return a result on a given action performed by a player. These results are shown in Table I, where we can appreciate that on over 100 events detected by our algorithm, less than 4% required more than a 20 milliseconds processing time, never, however, exceeding a 30 milliseconds delay.

Figure 2. Explaining the actions.

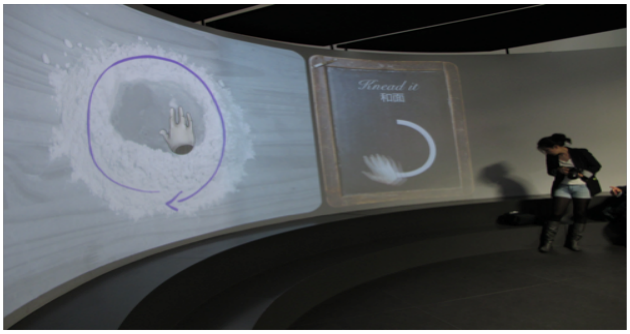

Figure 3. Virtual cooking board.

A second set of experiments was employed to assess the robustness of our algorithms in positively identifying a player's hands, and to contrast it with a gesture recognition system that employs color-tracking techniques implemented with the OpenCV libraries.

Figure 4. Playing the game in Shanghai.

Table II reports the results obtained from using both recognition systems over 50 trials and with 4 different players.

Table II. Tortellino X-Perience robustness results.

Gesture Recognition System	Positive Matches
Custom	88%
OpenCV-based	81%

Analyzing the results, we have concluded that OpenCV-based is slightly outperformed by our gesture recognition system for the following reason. Most misses while using OpenCV, over 70%, have been observed while the third player was engaged in the game. The same problems have not been observed with the same player when a recalibration phase has been performed before the beginning of a game. For this reason we are confident in saying that the calibration phase required by the color tracking algorithm used in OpenCV, in this case, was the source of its underperforming.

Table III. Tortellino X-Perience survey results.

Age Range	Did you enjoy the game?	Was it easy and intuitive to play?	Prefer a remote control?	# of answers
< 18	3.8	4.8	30 NO 3 YES	33
18-50	4.3	4.5	64 NO 9 YES	73
50-70	4.5	4.7	17 NO 0 YES	17
> 70	4.8	4.3	4 NO 0 YES	7

Finally, taking into consideration qualitative aspects, we performed a survey among a large sample of players that played a game. We asked the following set of simple to understand questions:

1. Did you enjoy the game?
2. Was it easy and intuitive to play?
3. Would have you preferred playing it with a remote control (or something similar to control the game)?

Players were asked to give a score, between 1 and 5, to answer the first two questions, while they were asked to simply give a positive or negative answer to the last one. The outcome of this survey is reported in Table III (answers are clustered by the age ranges of participants). The second column, which reports the average score of answers to the first question, clearly shows there has been a high appreciation for the game, which has especially been enjoyed by older people, we believe because of its theme. The third column, instead, shows that, when answering the second question, there has been no significant difference between people of different ages: the innovative human-computer interface has been appreciated, on average, by all. This fact has also been confirmed by the answers we received to the third question, summarized in the fourth column, where most players clearly affirmed that they preferred our *hands-free* interaction scheme to the use of a controller, as a remote control. Interestingly, we obtained a few preferences for the use of a remote control among players whose age ranged between the late thirties to the late forties; therefore all people belonging to the generation that most has used such type of hardware interface (with the exception of 3 over-70 who were not able to respond to this question).

5. CONCLUSIONS

We have discussed on the design process and on the user experience of a *hands-free* gaming system deployed in an immersive and public environment. We have specifically reported on a real case study of a food game (termed the Tortellino X-Perience) we developed, which has been publicly enjoyed by over one hundred players at the Shanghai 2010 World Expo [1, 6, 8]. We feel that our work brings an important contribution to the DOC (Design Of Communication) field, as our techniques can be generally expoited to provide a support to all those performing events to be enjoyed publicly, where a predefined set of gestures need to be automatically recognized.

6. ACKNOWLEDGMENTS

Our thanks to the Italian Projects DAMASCO (FIRB) and ALTER-NET (PRIN).

7. REFERENCES

1. M. Roccetti, G. Marfia, M. Zanichelli, “The Art and Craft of Making the Tortellino: Playing with a Digital Gesture Recognizer for Preparing Pasta Culinary Recipes,” *ACM Computers in Entertainment*, ACM, 8(4), December 2010.
2. S. Mitra, T. Acharaya, “Gesture Recognition: A Survey,” *IEEE Trans. On Sys., Man and Cyb.*, 37(3): 311-324, May 2007.
3. J. C. Lee, “Hacking the Nintendo Wii Remote,” *Pervasive Computing,* IEEE, (7)3:39-45, July-Sept. 2008.
4. A. D. Wilson, “Depth-Sensing Video Cameras for 3D Tangible Tabletop Interaction,” *Proc. of the Second Annual IEEE International Workshop on Horizontal Interactive Human-Computer Systems,* Newport (RI), 2007, 201-204.
5. R. Y. Wang, J. Popovic, “Real-time hand-tracking with a color glove,” *ACM Trans. Graph.,* 28(3):1-8, August 2009.
6. J. Giles, “One per Cent: Video Games Teaches You To Make the Perfect Tortellini, *New Scientist*, Jan. 2011, accessed online January 26th, 2011: http://www.newscientist.com/blogs/onepercent/2011/01/computer-game-that-teaches-you.html
7. H. H. Aviles-Arriaga, L.E. Sucar, C.E. Mendoza, “Visual Recognition of Similar Gestures,” *Proc. of the 18th International Conference on Pattern Recognition,* Hong Kong, 2006, 1100-1103.
8. M. Roccetti, G Marfia, “Recognizing Intuitive Pre-defined Gestures for Cultural Specific Interactions: An Image-based Approach”, *Proc. 3rd IEEE International Workshop on Digital Entertainment, Networked Virtual Environments, and Creative Technology*, Las Vegas (NV), 2011, 1-5.