A Fast and Robust Gesture Recognition System for Exhibit Gaming Scenarios

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

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

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

ABSTRACT

Advanced human-computer interaction schemes have gained a very important role, determining the successful introduction, or less, of new computer game consoles. Players, today, enjoy playing realistically, experiencing a whole set of interactions with objects and characters that compose virtual worlds that cannot be supported by a joystick or a mouse. Such result has been achieved with the adoption of new advanced video cameras (e.g., time-offlight cameras) and accelerometer sensors that support the recognition of a player’s gestures, thus with the devise of specialized hardware designed for such tasks. We will here show that it is possible to reach the same goal, in terms of gaming experience, solely utilizing an off-the-shelf webcam and a robust gesture recognition software system. Such result has been achieved designing and implementing a novel gesture recognition software, based on artificial vision algorithms and on contextual information that, together, lead to a fast and robust interpretation of hand gestures. This type of technology has proven to be particularly efficient for games displayed in exhibit scenarios. The efficacy of our approach is demonstrated through our game, Tortellino X-Perience, which has been enjoyed by hundreds of visitors at the 2010 Shanghai World Expo.

Categories and Subject Descriptors

K.8.0 [Personal Computing]: Games.

General Terms

Design, Experimentation.

Keywords


1. INTRODUCTION

Without any doubt, the major producers of computer games are engaged in a technological race that aims at providing their respective customers with the most exciting and realistic human-computer interface. In fact, recent trends show that the more realistically a game can be played, the higher will be its chances of beating it 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 technology that has raised the expectations from new gaming consoles has been implemented within the Nintendo Wii, where realistic human body movements are supported by the combination of infrared and accelerometer sensors. Although still being bound to the use of a controller (i.e., the Wiimote), the Nintendo Wii has been the first console to successfully let players move as they were playing on a real field. Now, a further step forward has been moved with Microsoft Kinect, an advanced video camera and software system that supports human body recognition on the Xbox console. In Kinect-supported games players can simply move and control their avatar, which mimics their movements. This has required the use of an advanced video camera, which, combined with a depth sensor, can estimate the distance from the objects that move in front of it in real-time, thus adding 3D information (i.e., depth) to 2D images.

The main contribution of this work is to provide a high level description of our new gesture recognition system that, with the sole use of one hardware component represented by an off-the-shelf webcam, supports playing computer games without the aid of any controller. The main novelty of this system lies in the use of contextual information that can be inferred by the game setting layout and the devise of three artificial vision algorithms capable of recognizing a player’s inputs: (a) a motion-tracking algorithm; (b) a hands recognition algorithm, and; (c) an action recognition algorithm. In brief, our gesture recognition system has been devised assuming that a player faces a monitor where a virtual world is displayed, while a webcam captures his or her movements from above. The motion-tracking algorithm receives the captured video stream and analyzes it, frame by frame, identifying a sub-set of key points where motion has been detected. After this step, the hands recognition algorithm, computing how close to the monitor motion is detected, determines whether a player is using one or both hands. Finally, the correctness of each of a player’s actions is checked in terms of timing and trajectory. A player, in the meantime, while moving,
can appreciate the effects of his or her actions on the monitor. This type of technology has proven to be particularly efficient for games displayed in exhibit scenarios.

In fact, all the mentioned algorithms have been implemented within a computer game, the Tortellino X-Perience, recently displayed at the 2010 Shanghai World Expo [1]-[4]. The underlying objective of the game was that of teaching players how to prepare Tortellini starting from its main ingredients (flour, water and eggs), while challenging them through a sequence of stages given by its recipe. Clearly, the preparation of a Tortellino requires walking through a very well defined number of stages where our gesture recognition system played a very important role: it supported the detection of correct movements from incorrect ones in real-time.

The rest of this paper is organized as follows. In Section II we describe the main approaches that have been, to this date, used in implementing advanced human-computer interaction schemes in commercial gaming consoles. In Section III, instead, we explain the difficulties we faced when designing our gesture recognition system for a gaming system displayed at a public exhibit, while Section V describes our Shanghai test-bed and reports on its performances. We finally conclude with Section VI.

2. RELATED WORK

A wealth of research has investigated how new and more natural means of interaction between humans and computers could be developed [5], [6], we will here focus on the prominent approaches that have appeared in commercial game consoles.

The most revolutionary and widespread controller that has been to this date created is, beyond any doubt, the Nintendo Wiimote controller. The Wiimote is equipped with an infrared camera sensor, which combined with two light emitting sources placed within the sensor bar positioned above or below the TV set, is used to locate its position in real-time. Such controller, in addition, also carries an accelerometer, used to register any sudden acceleration that is experienced by a player’s hand while engaged in a game. Clearly, using such technology requires players holding a Wiimote, hence demands the use of a hardware device that acts as a broker between the gestures a player performs and a Wii console. This hardly suites a scenario where hundreds or even thousands of different people can play a game using the same console during a single day, as was the scenario we dealt with in Shanghai.

Very recently, computer game players have been able to enjoy body free gaming with the Microsoft Kinect sensor and the Xbox. Kinect is a horizontal bar that is placed either above or below the video screen, containing a depth sensor in addition to an RGB camera, capable of recording the distance of all objects that lie in front of it. Depth information is then processed by a software engine that extracts, in real-time, the key human body features of players, thus enabling the interaction between the physical world and the virtual one. We will show that similar results can also be achieved simply using a normal video camera.

Also Sony has supported mixed reality experiences with the EyeToy and the Playstation Eye. Both of such products are based on a digital camera, using a very similar hardware approach to ours. However, our two approaches differ on the software side, as the Playstation Eye focuses on the detecting motion patterns above certain given areas, while our algorithms also recognize the position and the gestures performed by both hands.

In summary, the main experiences that may be found in commercial computer gaming consoles for implementing advanced human-computer interaction schemes either adopt complex hardware interfaces needed to detect and track movements, or reduce the accuracy of the tracking and gesture recognition processes, hence, reducing the realism with which players are supposed to play.

3. COMPUTER GAMES IN EXHIBITIONS

Our new gesture recognition system is based on a set of artificial vision algorithms tailored to suite exhibit gaming scenarios, where a computer game is displayed and can be played in a public area within a museum or a fair, for example.

Such types of scenarios introduce one advantage and one disadvantage, in terms of human-computer interaction design. On one side a clear advantage is given by the possibility of controlling the entire setting layout, and, hence, the context within which a player interacts with a game (e.g., the gaming arena area, position, illumination level, etc.). On the other side, instead, such scenarios pose new challenges, as no a-priori assumptions can be made on the players that, eventually, will play a game (e.g., skill level, physical features, etc.).

We devised a very traditional layout where a visitor plays waving his or her hands within a restricted area, while facing a video screen that displays the game graphical environment (Figure 1). The only difference with other well-known commercial gaming consoles is given by the video camera, which in our case is placed above the gaming arena, while, in Microsoft Kinect or Playstation Eye, for example, lies in front of the player. Such choice gives us the possibility to accurately track the horizontal movements that are performed by a player’s hands. The gaming layout, hence, represents here a contextual piece of information that can be tuned and put to good use to optimize the efficiency of the gesture recognition system in identifying and interpreting a player’s movements for a given game.

Such type of scenario, however, also represents an obstacle to the use of off-the-shelf technologies for gesture recognition. Just as an example, consider the case where we adopted the Playstation Eye gesture recognition system to support our game: before every time a new player started a game a new calibration phase would have been needed. In fact, the Playstation Eye requires an initial calibration phase where the body dimensions of a player are estimated. Another popular technique in legacy gesture recognition systems is the use of color recognizers to track hands (also requiring an initial calibration phase). Clearly, any of such solutions that require an initial calibration phase would result to be unfeasible in a context where a new player can join every few minutes.

In summary, a gesture recognition system that can efficiently cope with such scenario should be able to put to good use the a-priori assumptions that can be made on its setting layout, while being robust to its variety of users.

3.1 Algorithms: High Level Description

Our gesture recognition system is based on three algorithms, which, working in cascade, recognize the actions that are performed by a player at each given game stage.
The first algorithm worries about efficiently tracking any motion event that occurs over the game arena. To fulfill this aim, motion patterns are only searched for at specific locations (e.g., pixels). This approach is justified by the fact that a coarse-grained check for the movement of an arm or hand is sufficient for its detection. It is not necessary, then, to spend computational resources with no reason to check whether every single pixel that composes the game arena experienced a color change, thus revealing a motion pattern. Our approach reveals to be computationally efficient for an additional reason: it naturally enables the use, given the limited number of pixels that are kept under observation, of a parallelized approach to check for movement patterns that occur over different areas of the game arena.

Once a motion pattern is detected, the next step is that of recognizing which part of a player’s body generated such event. Perhaps, the most important component of our gesture recognition system, our hand recognition algorithm, works as follows. First of all, such algorithm, relying on the contextual information provided by the exhibit setting and by the video camera position, works under the assumption that only the arms and the hands of a player can occupy the gaming area. Hands are detected as those motion patterns that reach farther out towards the video screen (clearing now, why the video camera is set above a player’s head). This is achieved, for example, having a player lean over a desk, which represents the gaming arena, while playing. Second, it estimates whether the player is using only one or both hands by comparing the distance from the player of the two patterns that reach farther out towards the video screen.

Finally, the action recognition algorithm kicks in. Such algorithm, differently from many others designed for similar scopes, not only checks whether a hand reaches a certain position, but also checks whether a hand reaches a certain position following a given trajectory. In this way, it is possible to define valid trajectories that the player should follow to receive points, and invalid ones that represent mistakes. Almost all the tasks required within the main stages of the game that we developed can be summarized as actions where there is an origin, a destination point and a trajectory that should be followed.

4. SHANGHAI GAME EXHIBIT

Our gesture recognition system, more than simply being assessed in a set of experiments performed in a lab, has been tested by several hundreds of people visiting our game stand in Shanghai, at the 2010 Universal Expo. Spending a few words on the context where this game has been played for the first time, before moving on to its design, we here report that it has been created with the intent of describing an important piece of the cultural heritage of Bologna, Italy, to foreign players. For this reason we chose to challenge players in preparing a Tortellino, as it represents one of the most famous and traditional dishes of our city.

4.1 A Cooking Game

When designing our game we had to consider that: (a) not many people are familiar with cooking at all; (b) many players in China may be unaware of what a Tortellino is. Hence we designed a game that teaches as it is played. The game is divided in stages, where within each stage a player is taught the correct movements he or she should perform. Hence, while a player accomplishes a series of movements, mimicking the actions that have been previously displayed, the gesture recognition system checks for their correctness. If incorrect, the player is asked to retry. If correct, instead, the game moves on to the next stage.

Figure 2 summarizes, with a flow diagram, how each stage is organized. At first a player watches a video displaying a Sfoglina, a traditional Bologna Tortellini chef, that explains and shows what gestures should be performed. At this point, if, for example, the player needs to mix the flour, water and eggs altogether, he or she will receive points in reward and be able to proceed with the recipe only if performing the correct operation the right number of times.

4.2 Cooking a Tortellino

Cooking Tortellini, as cooking any type of food, follows a very well defined recipe, recipe that can be divided into successive phases, one per each stage of the game. In particular we implemented the following list of phases:

- The cooking board is empty, while flour and eggs appear at its right hand side. The player is required to move the ingredients at the center of the cooking board;
- The yolks and the albumen of the opened eggs float squeezed within the flour. The required action is to knead the ingredients, adding some water, ending up with a smooth ball of dough;
- The ball of dough is sitting over the cooking board. The dough, at this point, needs to be rolled out with a rolling pin;
- A thin foil of dough is lying on the cooking board. This should now be cut in squares, using a cutting wheel;
- Now, each square of dough should be filled with the ingredients (meat and cheese) that sit at the side of the cooking board;
- Each Tortellino, now open, needs to be closed joining the first pair of opposite angles;
- Each half-closed Tortellino becomes ready to be cooked after closing the second pair of its opposite angles.

The pictures taken from our testbed, and shown in Figures 3, 4 and 5, represent an example of the third stage of our game. In particular, Figure 3 shows the Sfoglina explaining how the
dough should be rolled out. Figure 4, instead, depicts a digital representation of two hands holding a rolling pin. Once the player succeeds performing the correct movements, a stop motion video shows all the steps required to obtain a thin foil from a ball of dough (Figure 5).

Figure 4. Actions performed by a player.

Figure 5. Result of a player’s actions.

7. REFERENCES

5. CONCLUSION
The goal of this paper was the description of a novel game, based on an innovative hands-free gesture recognition system. Such game, Tortellino X-Perience, has been successfully demonstrated at the Shanghai World Expo, where hundreds of visitors have been challenged in preparing the best traditional Tortellini dish.

6. ACKNOWLEDGMENTS
We express our gratitude to all our undergraduate students (CS Class: Multimedia Systems and Applications, University of Bologna). They assisted us during the implementation of the system with intense dedication. Special thanks are devoted to Articolture, a Bologna-based media agency, which implemented the graphical engine of the game.