Single Camera Pointing Gesture Recognition for Interaction in Edutainment Applications

Z. Černeková*
Comenius University, Bratislava
Slovakia

C. Malerczyk†
ZGDV Computer Graphics Center
Germany

N. Nikolaidis‡
Aristotle University of Thessaloniki
Greece

I. Pitas§
Aristotle University of Thessaloniki
Greece

Abstract

In this paper, a method for recognizing pointing gestures without markers is proposed. The video-based system uses one camera only, which observes the user in front of a large screen and identifies the 2D position pointed by him/her on this screen, his/her arm being in the fully extended position towards the screen. A GVF-snake is used in order to detect the pointing hand of the user, which is tracked in the following frames using the particle filters tracker. The center of gravity of the snake is used as a feature point and is transformed using linear transformation directly into the canvas coordinates. The method was tested on a large screen using applications designed for a wide range of different and even technically unversed users such as an image exploration for a virtual museum exhibit or intuitive interaction applications for gaming purposes. Experiments show very promising results for recognizing the pointing gestures by using a single camera.

1 Introduction

Human posture and activity recognition from video is a very active research area in nowadays because of its important applications in surveillance, human-computer interaction and computer animation. Hand gesture recognition is closely related to video-based interaction which is one of the most intuitive kinds of human-computer-interaction with mixed-reality applications [Malerczyk et al. November 2005]. Users are not wired to a computer, as it is necessary e.g. with electromagnetic sensors like data gloves, and maintain mostly unrestricted freedom of interaction. As a consequence, video-based interaction is the preferred kind of interaction especially for technically unversed users.

Several posture recognition systems have been presented so far [Heidemann et al. Aug. 2004], [Kehl and Gool 2004], [Liu and Fujimura 2004], [Yamamoto et al. August 2004], [Nickel et al. 2004], [Carbini et al. August 2004], [Littmann et al. 1996], [Kolesnik and Kuleßa 2001], [Richarz et al. Sept. 2006]. Many of the traditional systems are based on still cameras and background subtraction [Kehl and Gool 2004], [Liu and Fujimura 2004]; the silhouettes of the subjects are then used in posture recognition. The disadvantages of this scheme is that background subtraction is not robust and not always possible, and the method cannot distinguish postures when body parts are occluded by silhouettes.

Using multiple cameras and extracting depth information for the persons in the scene was proposed Yamamoto et al. in [Yamamoto et al. August 2004]. Four stereo cameras mounted in the corners of a ceiling that look down at an oblique angle allow to capture entire bodies and faces simultaneously. Thus arm pointing gestures are recognized without imposing restrictions on the position and orientation of the user.

In order to recognize pointing gestures many methods detect both hands and use additional information like head orientation [Nickel et al. 2004] or eye position [Carbini et al. August 2004]. Littmann et al. in [Littmann et al. 1996] show that a modular, neural network based system can achieve the visual recognition of human hand pointing gestures from stereo camera pairs. A person is positioned at one side of a table that is covered with a black 10×10 grid on a yellow surface. Several neural networks account for image segmentation, estimation of hand location, estimation of 3D-pointing direction, and necessary transforms from image to world coordinates and vice versa. The functions of all network modules can be learned from data examples only, by exploiting various learning algorithms.

The above mentioned approaches, employing multiple cameras, are rather expensive to deploy. Kolesnik and Kuleßa in [Kolesnik and Kuleßa 2001] tested a vision system which consists of a single overhead view camera and exploits a priori knowledge of the human body appearance, interactive context and environment. The user controls the motion of virtual objects by pointing with an arm extended towards the screen. However, only the horizontal coordinate of the location pointed on the screen is recognized by this method.

In [Richarz et al. Sept. 2006], the authors present a neural architecture that is capable of estimating a target point from a pointing gesture, thus enabling a user to command a mobile robot by means of pointing. They studied whether it is possible to implement a target point estimator using only monocular images from low-cost webcams. The results indicate that it is in fact possible to realize a pointing estimator using monocular image data, but further efforts are necessary to improve the accuracy and robustness of their approach.

In this paper, we focus on recognizing the cell of a grid on a

*e-mail:cernekova@fmph.uniba.sk
†e-mail:cmalerc@zgdv.de
‡e-mail:nikolaid@aiia.csd.auth.gr
§e-mail:pitas@aiia.csd.auth.gr

SCCG 2008, Budmerice, Slovakia, April 21 – 23, 2008.
© 2010 ACM 978-1-60558-957-2/08/0004 $10.00

screen that is pointed by a user, his/her hand being in the fully extended position towards the screen. The video-based module uses one camera, which observes the user in front of the screen. A GVF-snake is used to segment the pointing hand of the user in the first frame of the video. The pointing hand is subsequently tracked over time.

The remainder of the paper is organized as follows: In Section 2, the setup used in our method is described. In Section 3, a description of the proposed method is provided. Implementation is addressed in Section 4 and the possible applications are presented in Section 5. Experimental results on pointing gesture recognition are presented and commented in Section 6 and conclusions are drawn in Section 7.

2 System Setup

Our testing environment is equipped with a single uncalibrated firewire camera connected to the computer feeding the system with greyscaled images. The camera is placed on the top of the screen at approximately 2 meters from ground, thus observing the user from the front. The user stands in front of a screen of size 2.0m width and 1.5m height located at approximately two meters in front of him. The position of the user is somewhat pre-defined with respect to the camera set-up. There is a marker on the floor indicating the most suitable position of the user. It is extremely important that the user be provided with a visual feedback of his own pointing action. Different kinds of feedback can be presented to the user at the interface level, according to the specific application requirements. In our applications we have used two basic feedbacks; a small red point (understandable as a laser pointer metaphor) and a magnifying glass-based feedback, where the area around the pointed point is zoomed, providing an effect of holding a virtual magnifying glass with the pointing hand. In Figure 1 one can see the testing environment with the frontal camera and the user pointing with fully extended arm towards the screen.

Figure 1: The testing environment setup with the frontal camera. Exploring Hieronymus Bosch's "The Haywain" triptych with a virtual magnifying glass.

3 Proposed method

The first task to be solved is segmentation of pointing hand in the first frame. The static environment and the fixed camera at our setup allows using background subtraction. However, due to non-uniform lighting, the shadows cast by the user may cause problems. Therefore, in order to properly detect the silhouette of the user, we decided to used an active contour (snake). We have chosen a snake [Xu and Prince 1998] that uses the gradient vector flow (GVF) field, computed as a diffusion of the gradient vectors of a gray-level or binary edge map derived from the image, as its external force. Advantages of the GVF snake over a traditional snake include its insensitivity to initialization and its ability to move into boundary concavities.

The snake is applied in the first frame in order to localize the pointing hand. For the initialization of the snake a circle is used. The user is asked to point at a predefined area at the start of the session and the circle is centered at the area where the projection of the hand resides in such pointing gesture. The center of gravity of the snake is used as the *hand* ($\mathbf{x}_{hd}$) position. In the rest of the frames the points are tracked by a particle filters tracker [Zhou et al. 2004], which showed the best performance during the testing of the method.

Since, the position of the user is approximately set and the position and dimensions of the area where the user is pointing (canvas) as well as the camera position are fixed, his hand can appear only in a certain sub-area of the camera image. To find this area the user is asked in the beginning of the session to point at the the top right corner and the bottom left corner of the canvas and the image coordinates of those two points are recorded. Then the camera image coordinates of the hand $\mathbf{x}_{hd}$ (for a certain frame), with respect to the predefined sub-area, are transformed into normalized canvas coordinates $\mathbf{Y}_{hd}$ in range [0..1] using and easily defined linear transformation L_t.

$$\mathbf{Y}_{hd} = L_t(\mathbf{x}_{hd}) \tag{1}$$

In Figure 2 one can see an output of the particle filters tracker with the pointing hand detected in a camera frame.

Figure 2: Pointing hand detected and tracked in a frame acquired in the frontal camera setup.

4 Implementation

The complete hand pointing recognizer was implemented in C++. We have implemented the core of the method which was integrated into the hand pointing recognition system developed in ZGDV Darmstadt, Germany [Malerczyk et al. November 2005].

In order to speed up the calculation of the GVF snake in the first frame, which is time consuming, we calculate the gradient vector flow field only in a predefined image window, where we expect the hand would appear and not on the whole image.

Four types of trackers were tested, namely: Kanade-Lucas-Tomasi (KLT) tracker, elastic graph matching [Stamou et al. 11-14 September, 2005], modal analysis tracker [Krinidis et al. 2007] and particle filters tracker [Zhou et al. 2004], to obtain the best tracking results. The best performance in terms of precision and speed, was achieved by the particle filters tracker.

5 Applications

The single camera pointing gestures recognitions module can be used to provide input to a number of different applications whose goal is to create an intuitively usable experience for any (even technically unversed) user of the system, who is curious enough to explore virtual worlds with a new interaction paradigm like the pointing recognition system. For the creation of new scenario content it is important to have standardized and easy to use authoring tools and rendering components at hand. We use the instantreality-framework [Ins], [Behr et al.] for the rendering part of the applications. The instantreality-framework is a high-performance Mixed Reality (MR) system, which combines various components to provide a single and consistent interface for AR/VR developers. The framework provides a comprehensive set of features to support classic Virtual Reality (VR) and advanced Augmented Reality (AR) equally well. The goal is to provide a very simple application interface while still including the latest research results in the fields of high-realistic rendering, 3D user interaction and total-immersive display technology [Ins]. The instantreality-framework uses X3D/VRML as the programming language for the virtual worlds the user interacts with. Like most traditional toolkits, it uses a scenegraph to organize the data, as well as spatial and logical relations. In addition to the scene description, VR applications need to deal with dynamic behavior of objects, and the user interaction via non-standard input devices. The use of X3D/VRML as an application programming language leads to a number of advantages over a proprietary language [Behr et al.]:

- It is integral to an effcient development cycle to have access to simple yet powerful scene development tools. With X3D/VRML, the application developer can use a wide range of systems for modeling, optimizing and converting.
- The interface is well defined by a company-independent ISO standard.
- Due to platform independence, development and testing can even be done on regular standard desktop computers.
- VRML and JavaScript are much easier to learn than the low-level interfaces often provided by traditional VR toolkits.
- There are a great number of books and tutorials available.

5.1 Object Exploration Scenarios

As a consequence of using an open X3D/VRML environment one may easily conceive of many different immersive 2D and 3D applications (e.g. see Figures 1 and 3) that benefit from using a pointing based input device. Nevertheless, using a static pointing posture for interaction is slightly different to the use of traditional input devices like mouse or keyboard due to the fact that obviously no selection events like a mouse click are possible without the interpretation of dynamic hand movements. Therefore, status event changes are simulated using a time driven position evaluation within the scenario applications. Within a Java script node of the application the continuous input stream of pointing positions are observed. A selection event is automatically generated if the user is pointing at an approximate position on the screen for longer than a given time span or if the speed of the users hand movement drops below a given threshold.

Figure 3: Interaction with a three-dimensional object using virtual buttons to point at for rotation (3D-Model by courtesy of the Picture Gallery of the Academy of Fine Arts Vienna).

A first chosen scenario application addresses the exploration of a digitized painting using a virtual magnifying glass. For this scenario the well known triptych "The Haywain" (Figure 1) by the Netherlandish painter Hieronymus Bosch (c. 1450-1516) was used. Paintings of Hieronymus Bosch perfectly fit for an exploration using a virtual magnifying glass since Bosch is well known for his complex painted panels featuring fantastic and very detailed portrayals of demons, fools and other creatures from Eden to hell. The application directly starts with the full screen exploration of the painting. While no menu bars or other objects disturb the visual impression of the digitized painting, the visitor is able to focus solely on the painting and its details. An additional post processing step in the pointing gesture tracking module allows an extremely stable position of the magnifying glass, if the user is bringing an interesting detail into focus.

As an application dealing with the presentation of three-dimensional objects we have chosen the exploration of a bust of the Greek mythological creature Medusa. The user is able to look upon the bust from every angle by rotating it using four virtual buttons arranged at the right hand side and the bottom of the screen (see Figure 3). A small red cursor ensures permanent visual feedback during the interaction.

5.2 Multimodal Interaction

The recognition and tracking of a pointing gesture is obviously suitable to be used for multimodal interaction combining speech and gesture modalities. While the instantreality-framework provides a sensor node for speech recognition, the fusion of spoken input text and pointing based selection events is performed within a generic Java script node of the application's scene graph. As a first proof-of-concept for the

multimodal sensor fusion we have developed an interactive version of the well known number placement puzzle Sudoku (see Figure 4, for further information on Sudoku in general see e.g. http://en.wikipedia.org/wiki/Sudoku). Integers can be entered into empty cells of the current game by selecting a cell by pointing at it and saying the word "number" followed by an integer between one and nine. To enable conversation and discussion between users during the game play and to avoid unwanted misunderstandings of the system the speech recognition is grammar based. In addition to spoken integer input like "number one", "number two", etc. while pointing at an empty cell of the Sudoku board, additional commands can be used like "delete that" or "remove this" to undo wrongly entered numbers and "start new game" or "reset game" to control the status of the puzzle.

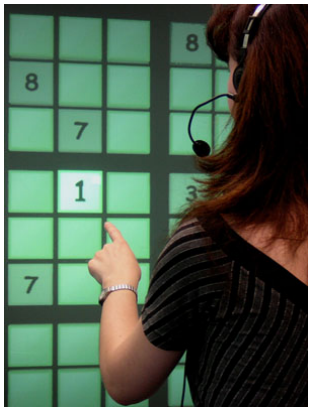

Figure 4: Solving a Sudoku puzzle using multimodal intercation like the fusion of pointing and speech recognition within the rendering application.

6 Experimental results

Experiments were conducted to show that the proposed pointing recognition system can indeed be used for human computer interaction. We have tested the method on the scenario application of the virtual exploration of the "The Haywain", using a virtual magnifying glass, as described in Section 5. The users reported that they were satisfied with the experience and the performance of the system.

In order to obtain accuracy of the system the user was asked to hold a laser pointer while operating the pointing system. The distance between the trace of the laser pointer on the canvas and the point identified by the system (and presented to the user as feedback) was used to measure the accuracy. The maximum distance between the two points was about 10 cm.

These experiments show that very good results can be achieved for pointing gesture recognition using only a single camera.

7 Conclusions and discussion

A method for the recognition of pointing gestures without markers using only a single camera was presented in this paper. The proposed method uses a GVF snake to detect the pointing hand and subsequently tracks obtained features over the video. The purpose of our method is to recognize 2D position pointed by the user on a screen to enable intuitive video-based interaction in applications like gaming (chess playing, puzzle solving, sudoku) or virtual museums (selecting a part of painting in order to obtain information for this part).

Very good results were achieved by the proposed system. Future work includes improving the performance of the method, providing automatic initialization of the session and integrating the method to other applications that have been already implemented, as described in Section 5.

Acknowledgement

This work has been conducted in conjunction with the 'SIMILAR' European Network of Excellence on Multimodal Interfaces of the IST Programme of the European Union (www.similar.cc).

This research was partially supported from Slovak Ministery of Education grant No. VEGA 1/3083/0.

References

Behr, J., Dähne, P., and Roth, M. Utilizing x3d for immersive environments. In *Proc. 2004 Web3D*.

Carbini, S., Viallet, J. E., and Bernier, O. August 2004. Pointing gesture visual recognition for large display. In *Proc. 2004 Int. Conf. Pattern Recognition, Cambridge*.

Heidemann, G., Bekel, H., Bax, I., and Saalbach, A. Aug. 2004. Hand gesture recognition: self-organising maps as a graphical user interface for the partitioning of large training data sets. In *Proc. 1996 17th Int. Conf. on Pattern Recognition (ICPR 2004)*, vol. 4, 487–490.

instantreality - advanced mixed reality technology, project homepage.

Kehl, R., and Gool, L. V. 2004. Real-time pointing gesture recognition for an immersive environment. In *Proc. 2004 IEEE 6th Int. Conf. on Automatic Face and Gesture Recognition (FGR04)*.

Kolesnik, M., and Kulessa, T. 2001. Detecting, tracking and interpretation of a pointing gesture by an overhead view camera. In *B.Radig, editor, LNCS: Pattern Recognition*.

Krinidis, M., Nikolaidis, N., and Pitas, I. 2007. 2d feature point selection and tracking using 3d physics-based deformable surfaces. *IEEE Trans. Circuits and Systems for Video Technology*.

Littmann, E., Drees, A., and Ritter, H. 1996. Visual gesture recognition by a modular neural system. In *Proc. 1996 Int. Conf. on Artificial Neural Networks*, 317 – 322.

Liu, X., and Fujimura, K. 2004. Hand gesture recognition using depth data. In *Proc. 2004 IEEE 6th Int. Conf. on Automatic Face and Gesture Recognition (FGR04)*.

Malerczyk, C., Dhne, P., and Schnaider, M. November 2005. Exploring digitized artworks by pointing posture recognition. In *Proc. 2005 6th Int. Symposium on Virtual Reality, Archeology and Cultural Heritage, Pisa, Italy*.

Nickel, K., Seemann, E., and Stiefelhagen, R. 2004. 3d-tracking of head and hands for pointing gesture recog-

nition in a human-robot interaction scenario. In *Proc. 2004 IEEE 6th Int. Conf. on Automatic Face and Gesture Recognition (FGR04)*.

Richarz, J., Martin, C., Scheidig, A., and Gross, H. M. Sept. 2006. There you go! - estimating pointing gestures in monocular images for mobile robot instruction. In *Proc. 2006 IEEE 15th Int. Symposium on Robot and Human Interactive Communication (ROMAN 2006)*, 546–551.

Stamou, G., Nikolaidis, N., and Pitas, I. 11-14 September, 2005. Object tracking based on morphological elastic graph matching. In *Proc. of 2005 IEEE Int. Conf. on Image Processing (ICIP 2005), Genova, Italy*.

Xu, C., and Prince, J. L. 1998. Snakes, shapes, and gradient vector flow. *IEEE Trans. Image Processing 7*, 3, 359–369.

Yamamoto, Y., Yoda, I., and Sakaue, K. August 2004. Arm-pointing gesture interface using surrounded stereo cameras system. In *Proc. 2004 Int. Conf. Pattern Recognition, Cambridge*.

Zhou, S. K., Chellappa, R., and Moghaddam, B. 2004. Visual tracking and recognition using appearance-adaptive models in particle filters. *IEEE Trans. Image Processing 13*, 11, 1491–1506.