Vision based user interface framework

Pavel Žák Radek Bartoň Pavel Zemčík
Graph@FIT research group
Department of Computer Graphics and Multimedia
Faculty of Information Technology
Brno, Czech Republic, 61266
[izakpa,ibarton, zemcik]@fit.vutbr.cz

Abstract — *This paper proposes a framework for creation of vision based user interfaces. The framework provides the stand-alone and configurable modules that can be included in desired application to deal both with the image processing tasks and event recognition. This approach enables development of alternative ways of human-computer interaction and also serves as a study reference for the students interested in the topic.*

1 INTRODUCTION

Over the past years the so called desktop environments have been de facto standardised as the main approach to the user interfaces. They employ elements, such as windows, menus, buttons, etc. and are typically controlled by the selection devices, such as mouse or touchpad.

The expansion of graphic accelerators and advances of computional power of modern personal computers have enabled various set of multimedia and 3D applications dealing with the complex scenes. For such applications, the traditional interaction schemes are less suitable. Such applications demand more intuitive ways that can provide the user greater immersion and make the interaction tasks more effective.

The alternative way of human-computer interaction is broughth by camera based systems. They capture the user position and actions that can be directly mapped into the interaction actions. The vision based interface could be more suitable for wide variety of interaction tasks but its development is not straightforward.

This work presents the framework that provides basic modules for building up the camera based user interfaces. It aims to enable easier development of alternative interfaces and also to serve as experimental platform for developing new interaction techniques.

The paper is structured as follows. The following Section 2 presents some of existing camera based interaction systems and schemes. Section 3 describes the proposed framework together with its basic modules. The examples of the framework use are mentioned in Section 4. Finally the Section 5 summarizes the paper and discuss the possible future work.

2 BACKGROUND

Large number of different academic and commercial camera based systems have already been presented. They exploit different methods and hardware setup to deal with the modern multimedia or 3D applications. So called virtual and augmented reality can successfully simulate the feeling of the scene immersion. They adapt the interaction techniques from real life experience and typically require specific tracking devices and environment with projective walls, such as in the case of the CAVE like systems (*Cave Automatic Virtual Environment*, [1]). The augmented reality system creation provides among others the library ARToolkit [2] that utilizes the tracking of visual marks.

In contrast using just a single camera or a stereo pair of cameras can serve for detection of the user position or pose by reasonable prize. With known user pose the various interaction tasks can be done both in desktop and virtual environment. In [3] the movement and navigation through 3D space is based just on the 2D position of the user head. The proximate 3D position of user head and hands obtained with two cameras serve as an interaction input for controlling the 3D virtual environment in [4].

The device-free interaction spaces [5] introduces the alternative area of camera based interfaces. Here the human-computer interaction is done in a limited area in front of the projective screen which is observed by several cameras. The cameras can locate precise position of user hand or a held object within the area which is the input to the interaction system reaching the virtual 3D tablet and touchscreen interface. More advanced example is the technology of Microsoft Touchlight [7].

Recently, new home entertainment kits appeared - the Sony Eye-Toy and Microsoft Kinect. They use affordable hardware devices (standard or infrared cameras) and detect the movement and pose of the user body, which serves for the interaction purposes. The commercial Camspace API, on the other hand, detects the 2,5D position and orientation of an arbitrary object handled by the user in front of the camera, which is used as the input control device.

3 FRAMEWORK DESCRIPTION

The framework aim is to provide the stand-alone modules and their mutual interfaces that can be included in a desired application to deal both with the image processing tasks and event recognition. So far, we have designed three major classes of processing modules – the object position and orientation resolver, the 3D scene controller and the virtual interaction spaces subsystem.

- The object position resolver is required to locate the object inside the camera view and return its aproximate 3D position and orientation. The framework design lets to implement the detection and localization of any arbitrary object. The current implementation supplies the detection and localization of user head using the Adaboost classifier. The module class provides the key input information to the system which can be used both for navigating or controlling task within the given environment.

- The 3D scene controller class enables the movement of a view within the virtual scene based on the object position supplied by the position resolver. The reaction to this information is configurable and can be determined to produce fuzzy or discrete events by comparing the position and orientation values to the given thresholds.

  The threshold concept is simply captured in Figure 1 but in the framework it is generalized to six degrees of freedom. For this example of the simple 1D case if we have the deviation *dev* of object position from its centre position, then the reasoning formula for viewport chane can be written as:

  ```
  IF (dev>noact_thresh) THEN
   view_change = v*(dev-noact_thresh)
  ```

  where v is the predefined constant that directly affects the velocity of viewport changes and *noact_thresh* is the smallest value of the deviation that permits any action.

  With the direct mapping of the position information to the viewport position in the scene the so called fishtank effect[6] can be reached. The screen area is in this case understood as a window into the virtual world in which the scene renedering depends on mutual position of user head and screen locations. That creates the illusion of real object presence behind the screen.

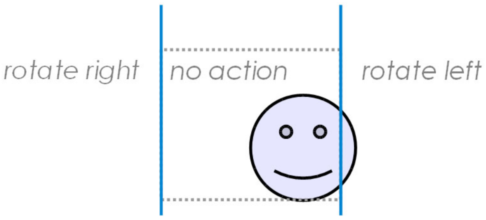

Figure 1: Simplified illustration of threshold areas used in scene controller.

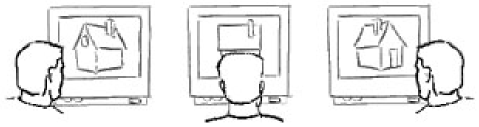

Figure 2: Illustration of the fishtank effect. Taken from [6].

- The third class provides the modules that enables build-up of virtual desktop widgets around the user. The virtual interaction spaces, as we note them, react to the user movement, its direction, or speed in given area of the image. So it enables execution of user interactive commands.

  The implementation is based on the optical flow detection when the interaction spaces are rectangle areas inside the vector field of computed optical flow. In every such area the mean value of optical flow velocity, direction or duration are measured. Depending on the required behaviour, these values are again compared to given thresholds which enables emmiting of interaction events. These events are then triggered in the similar way as in the classic desktop environment.

  The functional areas can be structured to form virtual buttons, scrollbars, touchpads, or movable objects and therefore it can be used to build complex interface layout. The layout can be positioned at fixed location within the image area or also it can be positioned around the user which enables him free movement together with the interfacing availability.

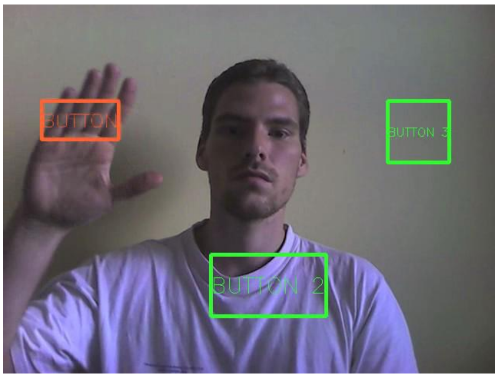

Figure 3: Example of virtual buttons created using interaction spaces.

4 USE EXAMPLES

The sectional modules of each class presented in previous section are combinable in several ways to reach the desired behaviour of the final user interface. For example, the head position and orientation is passed to the 3D scene controller and so the user can browse the 3D scene only with the head movements. The browsing can be set to act as pure transitional or rotational observer, to create the fishtank effect, or to act as their combination resulting with immersive free moving scene view. The possibilities to use the framework are demonstrated on several examples:

1. The first example uses the combination of head position and orientation resolver and scene controller, but in the limited space of 2D desktop environment with the aim of mouse movement emulation. The position of the user head is passed to the scene controller that directly alters the position of the mouse cursor. The information of head orientation is also used to control the click events, when the click is executed with user head nod.

2. The free movement through the virtual 3D scene can be reached similarly by using the head position and orientation resolver and scene controller. The controller is configured so that the input position and orientation changes of input vector are directly mapped to the changes of the virtual viewport in the observed scene. This means that as the user moves the head the translation in the virtual scene is done, similarly with changing head orientation the rotation of the viewport is altered.

3. The on screen menus and widgets are built up from virtual desktop controls created from the interaction spaces. The example application is a particle simulation that uses the widgets for setting up the simulation and visualization parameters. With combination of scene resolver as in previous example the resulting application could be independent on other input devices, visually attractive and having original user interface.

5 CONCLUSIONS

The framework for creation of vision based user intarfaces has been presented. The modular design extends its usability in wider application areas and also the framework can be used as an experimental or educational tool. So far three major classes are implemented allowing the basic functionality for browsing or controlling virtual spaces with object movement detection or with virtual desktop widgets.

The future work will focus on user pose detection and its integration into the framework structure together with reasoning from dynamical gestures.

Acknowledgments

This work has been supported by Higher Education Development Fund of Ministry of Education, Youth and Sports, grant no. FR2814/2010/G1, and EU FP7 project *TA2: Together Anywhere, Together Anytime* ICT-2007-214793, grant no 214793. Supported also by BUT FIT grant FIT-10-S-2 and the research plan MSM0021630528.

References

[1] Carolina Cruz-Neira, Daniel J. Sandin, and Thomas A. DeFanti. Surround-screen projection-based virtual reality: the design and implementation of the cave. In *SIGGRAPH '93: Proceedings of the 20th annual conference on Computer graphics and interactive techniques*, pages 135–142, New York, NY, USA, 1993. ACM.

[2] Hirokazu Kato and Mark Billinghurst. Marker tracking and hmd calibration for a video-based augmented reality conferencing system. In *IWAR '99: Proceedings of the 2nd IEEE and ACM International Workshop on Augmented Reality*, page 85, Washington, DC, USA, 1999. IEEE Computer Society.

[3] Ginés García Mateos and Sergio Fructuoso Munoz. Tierra inhospita: exploring a virtual world

with your face. In *ACE '05: Proceedings of the 2005 ACM SIGCHI International Conference on Advances in computer entertainment technology*, pages 383–384, New York, NY, USA, 2005. ACM.

[4] Flavia Sparacino, Christopher Wren, Ali Azarbayejani, and Alex Pentland. Browsing 3-d spaces with 3-d vision: body-driven navigation through the internet city. *3D Data Processing Visualization and Transmission, International Symposium on*, 0:224, 2002.

[5] Daniel Stodle, Olga Troyanskaya, Kai Li, and Otto J. Anshus. Tech-note: Device-free interaction spaces. In *In Proceedings of IEEE International Symposium on 3D User Interfaces 2009*, 2009.

[6] Colin Ware, Kevin Arthur, and Kellogg S. Booth. Fish tank virtual reality. In *CHI '93: Proceedings of the INTERACT '93 and CHI '93 conference on Human factors in computing systems*, pages 37–42, New York, NY, USA, 1993. ACM.

[7] Andrew D. Wilson. Touchlight: an imaging touch screen and display for gesture-based interaction. In *ICMI '04: Proceedings of the 6th international conference on Multimodal interfaces*, pages 69–76, New York, NY, USA, 2004. ACM.