

Interface for Gestural Interaction in Virtual Reality Environments

St. Gh. Pentiuc, E. G. Craciun

Stefan cel Mare University of Suceava,

str Universitatii nr. 13, RO-720229 Suceava, Romania, phone: +40 741 22 14 03, e-mail: pentiuc@usv.ro

L. Grisoni

Laboratoire d'Informatique Fondamentale de Lille,

Villeneuve d'Ascq 59650, France, e-mail: laurent.grisoni@lifl.fr

crossref <http://dx.doi.org/10.5755/j01.eee.121.5.1659>

Introduction

The Human Computer Interaction through virtual reality environments (VRE) assumes an intuitive and immersive feeling and an interaction with VRE very similar to the people interaction with the real world. The interaction interface design is in the most of cases “application dependent” and is influenced by the way the designer feels and perceives the things. In fact the people have a personal manner to accomplish it as in [1], but it can be identified some simple and general operations that the people use when they interact with a VRE: exploring the scene, choosing an object, taking it and handling it. An interaction in a VRE is a task comprising navigation, selection of the objects, moving, rotating or releasing them. These elementary actions are included after Bowman architecture [2] in the following components of a gesture interface:

- Navigation – represents a set of actions allowing the user to explore the VRE or to search something in it;
- Selection – the user chooses an object or a group of objects from a scene;
- Manipulation – comprises pick-up an object, moving it, releasing it, positioning and/or orientating it.

Beside these components there is another one, the control of the VRE, which gives the possibility to the user to change the interaction mode or the state of the system.

In the following we consider a gestural interface, having the above mentioned components, designed and implemented based on the information acquired through image processing. This interface is an instrument allowing the interaction with VRE with no other input devices such as wiimote, data gloves or similar. Analyzing the input video stream the gestural interface will recognize the user's hand, its postures and movements, and will map them onto VRE. The movements and postures of the hand will provide information for the navigation, selection and

manipulation tools. In order to control the virtual environment, the gesture depicted by the hand's trajectory must be recognized.

Another problem in the interaction with VRE concerns the necessity to connect the user's perception scale with the scale of interaction in VRE. This problem has been presented in [3], where the designed space for multi-scale selection, manipulation and navigation was created in order to reduce the time and the movements made by the user. Reality-based User Interface System (RUIS) introduced in [4] allows users to model new interfaces in order to “touch” and “sense” the virtual objects, just like in real world. Other approaches are presented also in [5] and [6].

In this paper we propose an interface for interacting with objects in a VRE, based on hand's postures recognition from a video stream. The interface has some specific tools for the selection and manipulation of virtual objects.

Gestural Interaction Interface

The overall architecture of an interaction interface that permits the interaction with a VRE through gestures made by the hand of a user is depicted in Fig.1. One of the main components of this input interface is a module able to detect the hand in the scene [7].

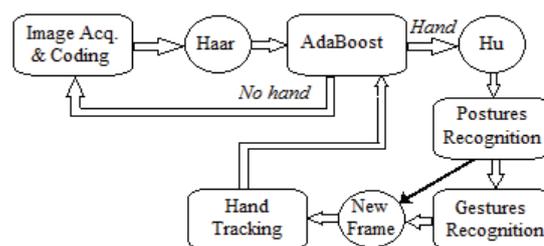


Fig. 1. Structure of the Gestural Interaction Interface

This module is based on a binary pattern classifier of the objects identified in the scene.

Initially the scene is retrieved in the RGB space color. So as to attain more detailed information from it, the image is translated into HSV space color; the reason for this operation being the fact that the representation in this color space is much closer with the human eye perception. A pattern represents a region of an image and is described by Haar-like features [8]. The classifier will decide if the analyzed pattern represents a hand or not.

The pattern classifier is based on the AdaBoost algorithm that builds a strong pattern classifier as a linear combination of weak classifiers [9]. The patterns that pass through the whole chain of weak classifiers in the AdaBoost algorithm are tagged as being a hand.

The supervised training of this classifier consists in employing the AdaBoost algorithm with a training set of patterns representing images of various hands' posture, and images with different objects appearing in the scene (not hands). The training phase is finished when the resulted strong classifier has an acceptable error rate.

This classifier was included in the designed interaction interface and we built an application that allows the user to interact with a VRE in real-time constraints.

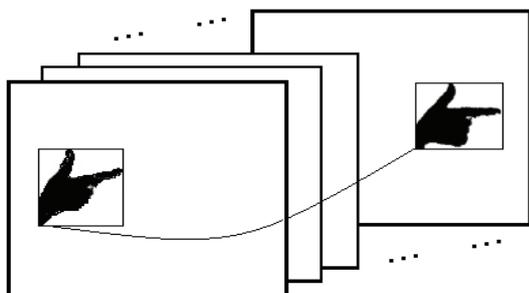


Fig. 2. During the tracking the size and the orientation of the bounded region are re-estimated

The aim of the interface is to make possible the real-time interaction of a user with a VRE. But the real time implies some restrictions on the response time that requires a reduced execution time. In order to have a high computational speed, the detection component analyzes each frame obtained from the video device by using the pattern classifier until a hand is detected. Starting from the next frame, the hand detected area is being tracked using Camshift algorithm [10], because the tracking is less time consuming than the detection algorithm. The tracking algorithm estimates for each pixel in the following frames the probability to be in the hand. The bounded tracked region is shifted to a new estimate location. Each time the shifted operation takes place, new values for tracked region size and orientation must be re-estimated (Fig. 2). This calculation is made by selecting the scale and orientation that are best fit to the hand-probability pixels inside the new bounded location.

Finally, the hand must be included in the VRE as a virtual object representing the input tool (a virtual hand). The movement of the virtual hand in VRE is synchronized with the movement of the real user hand, detected by the classifier and then real-time tracked. In the developed

module, when the hand effectuates a translation movement, the trajectory path of the tracked hand is represented by the coordinates of the start and end locations. For mapping the real hand coordinates onto the equivalent ones in VRE a bi-dimensional transformation vector is used.

The problem of distance perceived by the user in VRE isn't the goal of this work, therefore choosing a parameter for mapping the distance between the two environments is one to one pixel. The rotation and the orientation of the hand as specific problems for manipulation technique are presented in the next subsection.

The interface encapsulates modules for the coordination of the relevant actions across the virtual system. The bridge between the two environments, the real world and the VRE, is accomplished by virtual reality methods that compute the information acquired in the image processing stage.

Postures and gestures recognition

The user can select, grab, rotate, and release virtual objects through the selection and manipulation tools of a module based on hand posture recognition.

The module receives from the hand detection module the coordinates of the region containing a hand (*imgROI*) and calls the following the next function:

```

function postureRec(imgROI)
1. h ← handSegmentation(imgROI)
2. cont ← contourExtraction(h)
3. pattern ← momentsHu(cont)
4. (iclas, accuracy) ← postRec(w, pattern)
5. if accuracy >  $\epsilon$  then return iclas
6. else return 0
end

```

The postRec(*w*,*x*) function classifies the pattern *x* based on the coefficients of discriminant functions stored in the matrix *w*. The patterns represent valid postures described by the Hu moments. The classes contain patterns describing hand postures having the same significance. The matrix *w* was previously computed based on a training set. The function returns the ID of the posture class or 0 if the classification accuracy is not adequate.

A command in a virtual system may be a posture or a gesture. A posture may be extracted from a single frame, but a gesture is represented by the trajectory of a hand extracted from many frames. To recognize a specific gesture geometry is a more complex problem. Selecting good features and classification techniques for gesture recognition still represents a challenge, since the trajectory of the same shape may differ also by scaling, translating or rotating. Several recently techniques use for classifying gestures the Hidden Markov Model [11] or 1\$ recognizer proposed by [12].

In the designed, programmed and implemented application, the developed gesture recognition module allows user to define new types of gestures and modify or delete the old ones. Taken into account the dynamic behavior of the user, the HMM solution for gesture

recognition is fully justified as in other recognition processes ([13], [14]). The procedure follows:

```

procedure GestureRecognition
1. m:=n
2. for i=1 to n do
3.    $d \leftarrow$  distance from  $P_i$  to its neighbor
4.   if  $\text{dist}(P_i, P_{i+1}) < \varepsilon_1$  then eliminates( $P_i$ ); m:=m-1
5. end for
6. for i=2 to m-1 do
7.    $d' \leftarrow$  distance between the segment  $[P_{i-1}, P_{i+1}]$  and
   the point  $P_i$ .
8.   if  $d' < \varepsilon_2$  then eliminates( $P_i$ )
9. end for
10. *) compute the orientations of the segments in gesture
    trajectory
    
$$\theta_t = \arctan \frac{y_{t+1} - y_t}{x_{t+1} - x_t}, t = 1, \dots, T-1, \quad (1)$$

    where  $(x_t, y_t)$  are the palm center coordinates at time
    instant  $t$ , and  $T$  being the time length of gesture
11. let gesture  $\mathbf{g} \leftarrow (\theta_1, t_1, \theta_2, t_2, \dots, \theta_{T-1}, T-1)$ 
12. let  $\Lambda = \{\lambda_1, \lambda_2, \dots, \lambda_M\}$  be a set of HMM models which
    corresponds to gesture classes  $C_1, C_2, \dots, C_M$ 
13. for i=1 to M do
14. *) compute the probability  $P(\mathbf{g} | \lambda_i)$  that the gesture  $\mathbf{g}$  is
    being generate by the model  $\lambda_i$ .
15. if  $P(\mathbf{g} | \lambda_i) > P(\mathbf{g} | \lambda_k)$  then k=i
16. end for
17. decision:  $\mathbf{g} \in C_k$ 
end

```

The procedure has three main stages: data processing and reducing (steps 1-9), feature extraction (steps 10-11), training (step 12) and recognition (steps 13-17).

A gesture is a time-space shape obtained from the trajectory of the palm center. The class C_k for gesture \mathbf{g} may be determined in accordance with HMM method such as the probability that the gesture \mathbf{g} is being generated by the model λ_k is maximum.

Experimental results

The interface for interacting with VRE was programmed using specific modules for image processing from the OpenCV library and virtual reality modules from the OpenGL library.

Each tool of the gestural interface was evaluated by several experiments and the performance parameters were determined. It was tested and evaluated individually the hand detection classifier and the posture recognizer. The experimental data were captured during a task where some users were asked to perform hand movements in the front of the video camera and interact within the VRE displayed on the front desktop. The interaction were recorded on a movie and analyzed after experiment by the detection classifier and posture recognizer. In this approach the system would have to detect hand and recognize postures used by different people for selecting, manipulating, and releasing objects in VRE. Each class is afterwards evaluated, calculating the specificity, the sensitivity and the accuracy of recognition for each class.

It was counted the correct classified postures (TP - True Positives), the incorrectly identified postures (FP - False Positive), the correctly rejected postures (TN - True Negative) and the incorrectly rejected postures (FN - False Negative). The parameter *accuracy* computed as $(TP+TN)/N_{tot}$, where N_{tot} is the total number of cases, determines how well the classifier tests correctly the data set, *specificity*, computed as $TN/(TN+FP)$, calculates the ratio of correctly rejected images, and *sensitivity*, computed as $TP/(TP+FN)$, indicates the ratio of correctly postures identified.

The hand detection classifier had the *accuracy* = 0.89, *specificity* = 0.96 and *sensitivity* = 0.75. Due to the fact that the working environment was a variant one, and it had a low cost video input device, the hand detection classifier performance wasn't high qualified.

The experiment involves three different types of posture for *selecting*, *grabbing* and *releasing* objects.

Varying the segmentation threshold in a certain interval, the accuracy for the posture recognition is illustrated in the Fig. 3.

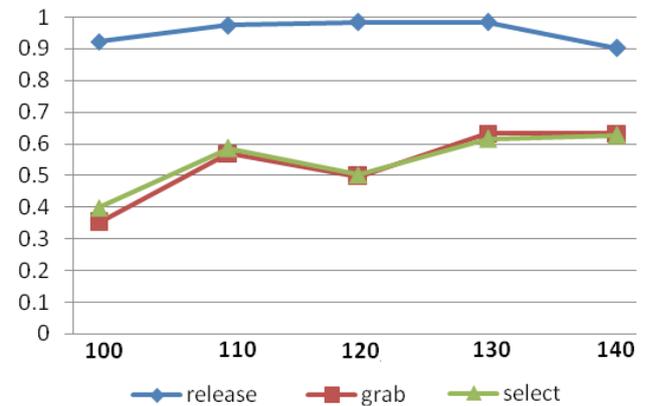


Fig. 3. System accuracy measured for a threshold varying in the interval [100,140]

The system accuracy is satisfactory for the *released posture*; in contrast some problems occur when we have to analyze the other two postures. The *selection* and the *grabbing postures* are very similar according to the features we choose. So, in these cases the recognizer for posture will mislead us to false positive *selection posture* instead of true positive *grabbing posture*. To skip this problem we propose several solutions for further development such as a combination between Hu moments with other features, like in [6]. The evaluation of the proposed method provides an average accuracy rate of about 0.70, with the maximum value for the release posture 0.95. Therefore the presented method is a primary way of dealing with posture recognition. Fast and easy to implement, our method still has its merits despite the fact that the accuracy is not comparable to other procedures described in current literature. The features which have been used are scaling, rotation and translation invariant and these did make possible to obtain a good performance in selection and manipulation virtual objects.

Considering that we want to control the system further, we have developed an instrument which allows recognizing the user's gesture. For each gesture we

acquired 100 samples for the training set and another 10 samples for evaluation/test set.

The training step was performed just once, but the number of hidden states was varied for the HMM from N=4 until N=10. It was not experimented a greater number of hidden states because it was obtained a high edge for N=6, and after that the system performance was constant decreasing. The achieved performance was of 95.3% and is due to the fact that the chosen gestures work very well with the orientation feature.

Conclusions

The paper analyzes the problem of designing and implementing a gestural interface for interacting with VRE, centered on providing some specific tools for each of the interface component. In this aim it was designed, programmed and implemented an application able to detect and recognize postures of the hand and elementary gestures for interacting with VRE: selection, grabbing, release.

The proposed interaction techniques are dealing with video acquisition, image processing, pattern recognition, extraction of different types of information, and use of it in a more intuitive way. The accuracy, sensitivity, and specificity of the designed interface were evaluated in some experiments and the results are very good.

Acknowledgements

This paper was supported by the project "Knowledge provocation and development through doctoral research PRO-DOCT - Contract no. POSDRU/88/1.5/S/52946", project co-funded from European Social Fund through Sectorial Operational Program Human Resources 2007-13.

References

1. **Goldin-Meadow S.** Talking and Thinking with our Hands // *Current Directions in Psychological Science*, 2006. – No. 15(1). – P. 34–39.
2. **Bowman D. A., Kruijff E., Laviola J. J., Pou, I.** 3D User Interfaces: Theory and Practice. – Addison-Wesley, Boston, MA, USA, 2005. – 487 p.
3. **Peck S. M., North C., Bowman D.** A Multiscale Interaction Technique for Large, High-Resolution Displays // *Proceedings 3DUI '09*. – Washington, DC, 2009. – P. 31–38.
4. **Takala T. M., Pougliose R., Rauhamaa P., Takala T.** Reality-based User Interface System (RUIS) // *Proceedings of IEEE 3DUI Symposium*. – Singapore, 2011. – P. 141–142.
5. **Bielskis A. A., Denisovas V., Drungilas D., Gricius G., Dzemydienė D.** Multi-Agent-Based Human Computer Interaction of E-Health Care System for People with Movement Disabilities // *Electronics and Electrical Engineering*. – Kaunas: Technologija, 2010. – No. 7(103). – P. 77–82.
6. **Serafinavičius P.**, Estimating Characteristic Points of Human Body for Automatic Hand Pointing Gesture Recognition // *Electronics and Electrical Engineering*. – Kaunas: Technologija, 2007. – No. 8(80). – P. 83–86.
7. **Craciun E. G., Pentiu S. G.** Hand Detection using a Set of Classifiers Based on Haar-Like features // *Proceedings Sisteme Distribuite*. – Suceava, 2009. – P. 90–93.
8. **Takeshi M., Toshimitsu K., Osamu H.** Joint Haar-like Features for Face Detection // *Proceedings ICCV*. – Beijing, 2005. – P. 1619–1626.
9. **Viola P., Jones M. J.** Robust real-time face detection // *International Journal of Computer Vision*, 2004. – P. 137–154.
10. **Mahmoud E., Ayoub A., Omer R., Bernd M.** Posture and Gesture Recognition for Human-Computer Interaction // *Advanced Technologies InTech*. Online: <http://www.intechopen.com/articles/show/title/posture-and-gesture-recognition-for-human-computer-interaction>.
11. **Rabiner L. R.** A Tutorial on Hidden Markov Models, and Selected Applications in Speech Recognition // *Proceedings of the IEEE*, 1989. – Vol. 77. – P. 257–286.
12. **Wobbrock J. O., Wilson A. D., Li Y.** Gestures without libraries, toolkits or training: A \$1 recognizer for user interface prototypes // *Proceedings UIST '07*. – Newport, New York, 2007. – P. 159–168.
13. **Pentiu S. G., Schipor O. A., Danubianu M., Schipor M. D., Tobolcea I.** Speech Therapy Programs for a Computer Aided Therapy System // *Electronics and Electrical Engineering*. – Kaunas: Technologija, 2010. – No. 7(103). – P. 87–90.
14. **Vatavu R. D.** The Effect of Sampling Rate on the Performance of Template-based Gesture Recognizers // *Proceedings ICMI*. – Alicante, 2011. – P. 271–278.

Received 2011 10 31

Accepted after revision 2012 01 03

St. Gh. Pentiu, E. G. Craciun, L. Grisoni. Interface for Gestural Interaction in Virtual Reality Environments // *Electronics and Electrical Engineering*. – Kaunas: Technologija, 2012. – No. 5(121). – P. 97–100.

The main contribution of the paper is the concept, design and experimentation of an interface able to satisfy all the main function that the user needs to interact with VR environments. The interface was aimed to supply a unified gestural paradigm. It deals with video acquisition, image processing, features extraction and pattern classification. The results of the experiments with the posture and gesture recognizing application validated our approach. Ill. 3, bibl. 14 (in English; abstracts in English and Lithuanian).

St. Gh. Pentiu, E. G. Craciun, L. Grisoni. Gestų interakcijos sąsaja virtualios realybės aplinkoje // *Elektronika ir elektrotechnika*. – Kaunas: Technologija, 2012. – Nr. 5(121). – P. 97–100.

Pateikiama sąsajos, užtikrinančios vartotojo poreikius sąveikaujant virtualios realybės aplinkoje koncepcija, dizainas ir eksperimento rezultatai. Sąsaja skirta aprūpinti unifikotą gestų paradigmą. Joje panaudotas vaizdo paėmimas, vaizdų apdorojimas, bruožų išskyrimas ir vaizdų atpažinimas. Eksperimento rezultatai su laikysenos ir gestų atpažinimo aplikacija patvirtino autorių požiūrį. Il. 3, bibl. 14 (anglų kalba; santraukos anglų ir lietuvių k.).