Methods and Algorithms for Gestures Recognition and Understanding

St.Gh. Pentiuc, Radu Vatavu, Tudor Cerlinca, Ovidiu Ungureanu

Faculty of Electrical Engineering and Computer Science
University “Stefan cel Mare”of Suceava, Romania
pentiuc@eed.usv.ro, raduvro@yahoo.com, tudor_c@eed.usv.ro, ungurean.ovidiu@gmail.com

Abstract

The paper discusses the problem of visual recognition of several hand postures and gestures. Processing is performed in a top-view scenario with a top-mounted camera that monitors the user’s hands on the working desktop. By careful choosing and controlling of the scene and lighting conditions, hands segmentation is fast and robust which increases the performances of the hand posture classifier. The chosen classifier was a multilayered Perceptron with three layers. By keeping all the processing at a low level of complexity and by considering an appropriate control of the environment, we obtain a real time 25 fps functional system with high detection and recognition accuracy results. The interpretation of gestures may be done by estimating also the kinematics parameters. To the end, is presented a Video Information System, that manages a database with a multi-level index structure for video spatio-temporal. In the frame of this system an SQL extension is presented for searchung the video spatio-temporal knowledge representation The data structure and SQL extension language were designed to support both spatial and temporal queries.

1. Introduction

The human gesture represents a source of valuable information in scene analysis and a natural mean for interacting and conveying information [1]. The video based gesture recognition systems have the main attraction of not being intrusive. On the other hand, the gesture based interfaces are looked upon as ideal with respect to the human computer interaction techniques [2, 3].

The paper discusses three steps to the visual recognition of a set of hand postures and learning their significance. Firstly it will be discussed the acquisition and recognition the hand postures selected in accordance to several commonly commands that may be performed for interacting with virtual objects inside VR environments. Posture recognition is carried out using a Multilayered Perceptron with a three layers structure. An alternative to this method is the analysis of the recorded kinematics features of the image of the hand in a video stream.

But for understanding gestures it is necessary to have also time and space information accompanying the images. A video spatio-temporal database is used in an application where data-types can be characterized by both spatial and temporal semantics. A Video Information System for indoor surveillance represents a good example of such application. A lot of models for spatio-temporal knowledge representation where designed, each of them trying to optimize the spatio-temporal queries. The best models are those who are based on the VHR tree [4] [5]. Another good model is the one that is based on the association map and frame-segment tree [6]. Spatio-temporal queries can be processed very quickly but the model wasn’t designed to minimize the storage space, which makes him unusable when the video sequences are big.

For the special case of virtual environments, appropriate human computer interfaces are in order. VR appears as an impoverished version of the physical world with incomplete sensory cues and simplified and inconsistent world models. The virtual experience is influenced by experiential, cognitive, perceptual and motor differences between users. Hence, the interaction technology should be appropriate so that the overall user experience in the virtual environment should not be diminished.

2. Gesture Recognition

The working scenario includes a top mounted camera that monitors the working area and the user’s hands. Video capture is carried out at a resolution of 320x240 and 25 fps. The working desk assures a homogenous background (see Figure 1, working desk is of blue colour) that allows for a fast and accurate segmentation of the user’s hands.

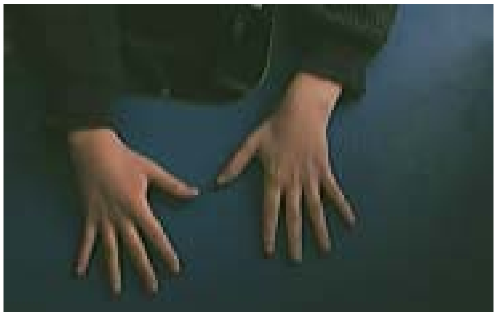


Figure 1: Camera view of the working area (top view).

Lighting is controlled in order to assure for a good contrast between the user’s hands and the working desk. The video camera auto controls the brightness and exposure settings.

2.1. Hands detection

Hands segmentation is achieved using a simple low cost skin filtering in the HSV colour space on the hue and saturation components.

$$p \textit{ is skin} \Leftrightarrow hue(p) \in \lfloor h_{low}, h_{high} \rfloor \wedge saturation(p) \in \lfloor s_{low}, s_{high} \rfloor \quad (1)$$

where p is the current pixel submitted to classification and $\lfloor h_{low}, h_{high} \rfloor and \lfloor s_{low}, s_{high} \rfloor$ are the low and high thresholds for the hue and saturation components.

The technique is very fast (the complexity order is ***O(n)*** where *n* is the dimension of the processed video frame) and assures for accurate hands segmentation under the previously mentioned working conditions. Segmentation results are given in Figure 2.

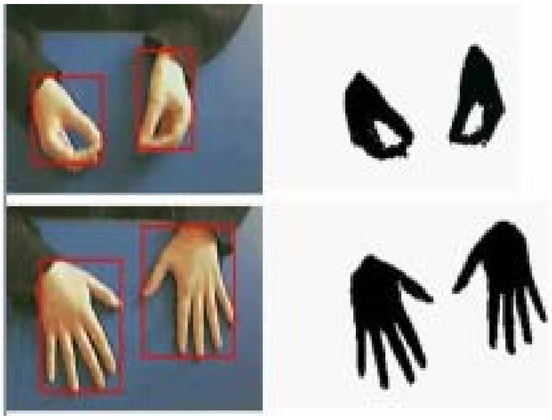


Figure 3: Segmentation results (segmentation is performed in the HSV colour space by filtering on hue / saturation).

2.2. Hands postures recognition

We have selected four hand postures by considering a few common operations encountered when interacting with virtual objects (see Figure 6) such as: selection, translation, rotation and resize. Two of these operations (selection and translation) are performed with one hand only, the other two (rotation and resize) are two-hand operations. Finally, we only have 3 distinct hand postures as presented in Figure 3.

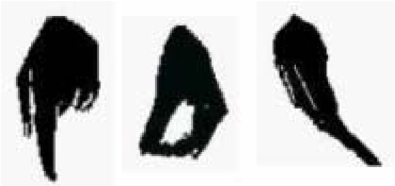
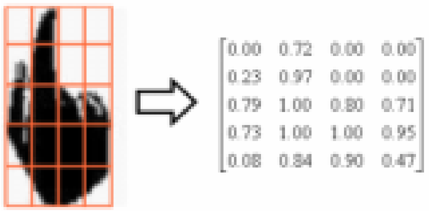


Figure 3: Hands postures selected for recognition

Recognition is performed using a multi-layered Perceptron, organized using a 3-layer structure of 39 neurons (20-16-3), as follows:

- the first layer consists of 20 input neurons coding each hand blob using 5 x 4 = 20 values, normalized in the interval [0..1] (see Figure 4)
- the second layer uses 16 hidden neurons. Experiments showed that 16 neurons in the hidden layer offer the best performance on the testing set
- the third layer with 3 neurons, each outputting a real value in the [0, 1] interval representing the probability of recognition for each of the 3 hand postures.



$$\begin{bmatrix} 0.00 & 0.72 & 0.00 & 0.00 \\ 0.23 & 0.97 & 0.00 & 0.00 \\ 0.79 & 1.00 & 0.80 & 0.71 \\ 0.73 & 1.00 & 1.00 & 0.95 \\ 0.08 & 0.84 & 0.90 & 0.47 \end{bmatrix}$$

Figure 4: Hand blob coding using a 5x4 matrix structure

The results obtained on a test set consisting of 67 images show a level of accuracy of 92%. Details with regards to the multi-layered Perceptron are given in table 1.

Table 1. Neural network details

Network structure	39 neurons distributed in 3 layers: 20-16-1
Training set	152 images
Testing set	67 images
Accuracy on the testing set	92% (61 of 67 images correctly classified)

Prior processing includes blob rotation so that the blob's longest axis should be parallel to the vertical axis, see Figure 5. This is done by computing the two axis of the ellipse of inertia having the same area and centre of mass as the hand blob.

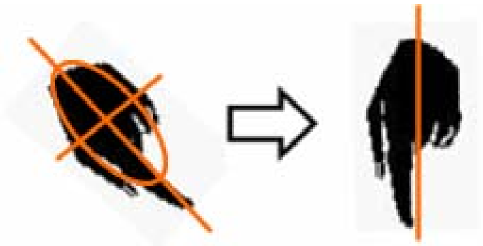


Figure 5. Blob rotation based on the axis of the ellipse of inertia.

The performances obtained overcome those presented in [8]. A neuro-fuzzy approach, as in [9], can improve the accuracy of the recognition process.

3. Learning Gestures Based on Kinematics Parameters

But gestures are not only static postures. Gestures mean posture and motion. The estimation and description of the motion in images may be done by various technique, as in [10] where is presented a method for the recognition of the human movement using temporal templates. Our approach tries to compute the optic flow at each pixel supposing some assumptions about the scene. The technique used may not represent a solution for the general problem but works robustly enough for a subset of applications.

The main stages of this technique are: hand identification, determination of a region of interest around the moving hand, target tracking, estimation of kinematics parameters that will be recorded (see figure 6).

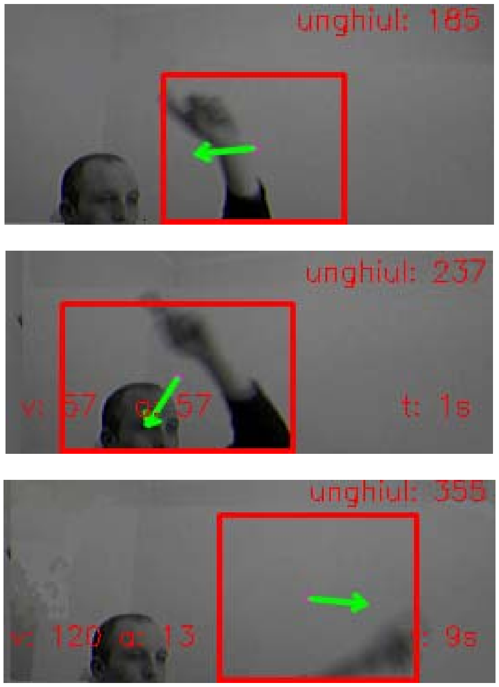


Figure 6: Three frames with kinematics parameters of the left hand oscillatory motion

The result are a file with the recorded values of the coordinates, velocity and acceleration of the gravity center of the hand . This file represents the training set of patterns that will be used in the description of the movement. This approach may be very useful in interpreting a class of gesture that may be differentiated only by velocity (e.g. an horizontal slow movement of the hand may be interpreted as a pointing gesture, while a rapid horizontal movement may be interpreted as a negation).

4. Spatio-Temporal Knowledge Representation Model

Typically, a video spatio-temporal database will store the following information:

- time stamp information
- spatial properties of each object which appears in the video sequence
- spatial and temporal relationships among the objects in the video sequence

The model we developed is based on the VHR indexing tree which was originally designed by Lei Chen & Vincent Oria [4]. The spatio-temporal knowledge representation model is build upon a multi-level indexing structure.

The first level of the indexing structure is a linked list of distinct objects which appears in the video sequence. Every object in the list is defined as a 3-tuple: <*IDi*,*SIi*,*LKFi*> where: *IDi* is the object's unique identifier, *SIi* describes the spatial properties of the object, *LKFi* is a list of frames which contains the current object. Each node in the list contains a unique identifier of the frame and a pointer to the corresponding root of the VHR tree.

The second level of the indexing structure is the VHR tree, which will store the following information:

- the unique identifier of each frame in the video sequence
- scene objects defined by their spatial properties
- spatial and temporal relationships among objects in the video sequence

The VRH tree is an extension to HR tree and was designed to minimize the disk space needed for data storing and optimize the spatio-temporal queries.

For every frame in the video sequence, the HR model computes the difference between R trees of current and previous frames. The major advantage of VHR model is that he computes the difference between R trees of the current frame and R trees of the two last frames. The main idea of both HR and VHR trees is to reuse the R trees at different moments of time.

We have focused on the representation of the following classes of spatio-temporal relationships:

- temporal relationships: *together, before, after*
- spatial relationships which are divided in two different categories: directional (*left, right, up, bottom*) and topological (*partial covered, total covered, touched*)

According to [2], the spatio-temporal relationship between two objects *Ai* and *Aj* in the frame interval $FI = [FK_i, FK_f]$, may be expressed as $A_i(r, fz, Ik)$, where *r* is a directional or topolocigal relationship and fz is a fuzzy member who has a value between 0 and 1. Taking into consideration the spatial distance between two objects, the fuzzy member fz will specify how much this distance satisfy a specific spatial relationship. In figure 7 it can be seen different cases for a ***Object1***(*left,fz,1)****Object2*** relationship between two objects in a video sequence.

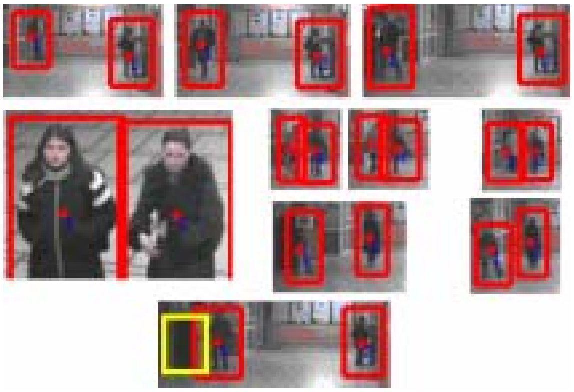


Figure 3: Different cases of objects relationship in video sequence.

As it is shown we have focused on the representation of two different spatial relationships: between two moving objects and between one moving object and one stationary object. In our case, the stationary object (the coffee maker) is marked with yellow rectangle.

The temporal relationships can be defined as follows:

- *together*: Two objects *Obj1* and *Obj2* shows up *together* only if there is a frames interval $FI = [FK_i, FK_f]$ in which both objects appears.
- *before*: An object *Obj1* appears *before Obj2* only if two frames interval $FI1$, $FI2$ exists for which $FK1_f \le FK2_i$ and $Obj1 \in FI1$, $Obj2 \in FI2$.

It has been developed a set of SQL operators in order to support both spatial and temporal queries. The SELECT statement syntax which is listed below is pretty similar with the one from the most of the DBMS.

```
SELECT FRAMES|COUNT [ALL|FROM
start_frame TO [end_frame]]
WHERE condition
```

Some of the conditions that are supported by our SQL language extension are listed below:
1. NUMBER OF OBJECTS IS EQUAL|
BIGGER|SMALLER|BETWEEN(minNrObj,maxNrObj)
[WITH|THAN RefValue]
2. OBJECT Obj1 APPERARS [IN AREA (X1,X2,Y1,Y2)]
3.RELATIONSHIP Obj1 fuzzy/spatial /temporal relationship Obj2

Logical operations are also supported by using AND, OR, NOT keywords.

As examples of queries that can be resolved by this SQL extension: *find all frames in which object **a** appears* or *find all frames in which object a appears in the left of object b and object c appears in the right of coffee_maker*.

The Video Information System that has been developed was tested with video sequences recorded in different locations of the "Stefan cel Mare" University of Suceava. The main window of the application is presented in figure 7. The application consists of four modules:

1. moving target detection module
2. video spatio-temporal database builder based on the VHR tree
3. SQL engine for query processing
4. query builder which helps users to build their own SELECT statements using graphical controls

To increase the speed of VIS, users can define one or more exclusion areas, which are rectangular areas where moving object will never be present. . At this moment, the SQL engine which is responsible with query translation and processing is not integrated into a DBMS but in our Video Information System.

For personalized queries, users can give a name to each object present in the video sequence. Even the static objects can have names: coffee maker, access door 1, access door 2, etc.

5. Conclusions

Visual recognition of a set of hand postures is achieved with a real time gesture based interacting system with virtual reality. The hand postures have been selected in correspondence with a few operations commonly performed in virtual environments, such as: object selection, translation, rotation and resizing of virtual objects.

The working scenario includes a top-mounted camera that monitors the user's hands on the working desktop. Scene parameters (such as lighting conditions, working area complexity, etc.) have been carefully chosen which leads to fast and robust hands detection that increases further the performances of the postures classifier. We managed to keep all the processing at a low level of complexity and ended up with a real time 25 fps functional system with high detection and recognition accuracy results.

In this paper is presented the background and the overall structure of a VIS elaborated in the University of Suceava. The system was experimented in the University video surveillance system.

The main contributions that may be revealed are the SQL language we defined, SQL engine, and data structure we implemented and improved for spatio-temporal relationship and knowledge representation.

Future work will be focused on the integration of the SQL engine into PostgreSQL [3] or to develop a framework in ActiveX similar to [6]. The engine will also be improved, in order to be able to translate the natural language into SELECT statements. All modules will be integrated in a collaborative environment [7].

6. References

[1] Melinda M. Cerney, Judy M. Vance, "Gesture Recognition in Virtual Environments: A Review and Framework for Future Development", *Iowa State University Human Computer Interaction Technical Report* ISU-HCI-2005-01, 28 March, 2005

[2] Matthew Turk, "Gesture Recognition, chapter 10", *http://ilab.cs.ucsb.edu/projects/turk/ TurkVEChapter.pdf*, 2005

[3] Axel Mulder, "Hand Gestures for HCI, Hand Centered Studies of Human Movement Project", *Technical Report* 96-1, School of Kinesiology, Simon Fraser University, February 1996

[4] Lei Chen, Vincent Oria, "A multi-level index structure for video databases", *Multimedia Information Systems*, pag, 28-37, 2002

[5] Lei Chen, Vincent Oria, "MINDEX: An Efficient Index Structure for Salient Object-based Queries in Video Databases", *Multimedia Systems*, 2004, pag. 56-71

[6] Mesru Koprulu, Nihan Kesim, "Spatio -Temporal Querying in Video Databases", *Information Science*., p. 131-152, 2004

[7] Mohamed Minout, Esteban Zimanyi, "Algebra-to-SQL Query Translation for Spatio-Temporal Databases", *DEXA* pag. 904-913, 2004

[8] Sorin Vlad, Nicolae Morariu, "Using Neuroshell Neural Networks Simulator for Prediction Problems Solving", *Advances in Electrical and Computer Engineering*, ISSN 1582-7445 - No 1, 2005

[9] Alexandru Bârleanu, "Fault Diagnosis of Wall Hang Boliers Using a Neuro-Fuzzy Model", *Advances in Electrical and Computer Engineering* ISSN 1582-7445 - No 1, 2005

[10] Bobick, A. and J. Davis, "The recognition of human movement using temporal templates," *IEEE Transaction on Pattern Analysis & Machine Intelligence*, 23(3), March 2001.