REAL TIME HAND TRACKING AND 3D GESTURE RECOGNITION FOR INTERACTIVE INTERFACES USING HMM

C. Keskin, A. Erkan, L. Akarun
Computer Engineering Dept.
Boğaziçi University
E-Mail: ckeskin@ttnet.net.tr, {erkanays,akarun}@boun.edu.tr

Abstract

In this study we have developed a human-computer interaction interface (HCI) based on real time hand tracking and 3D dynamic gesture recognition using Hidden Markov Models (HMM). We propose a system, which captures and recognizes hand gestures of the user wearing a colored glove, where the hand coordinates are obtained via 3D reconstruction from stereo.

In recognition of eight defined gestures, the system was able to attain 98.75% accuracy in 160 trials, using a pair of ordinary web cams. We provide supplementary features such as an interactive training system and a self-calibration utility for the cameras. The overall model is designed to be a simple and robust gestural interface prototype for various PC applications.

1. Introduction

The use of hand gestures is a noteworthy alternative to cumbersome interface devices for human-computer interaction (HCI). In particular, visual recognition and interpretation of hand gestures provides the ease and naturalness desired for HCI. For the visual recognition of hand gestures HMM has been used prominently and effectively for various applications.

Starner and Pentland implemented one of the earliest dynamic gesture recognition systems, where they used HMM to recognize American Sign Language using a single camera [1]. Oka, Sato and Koike have developed another 2D vision based system, where they made use of the Kalman Filter for noise elimination and HMM for gesture recognition [2]. In an earlier study the authors also employed a neural network for a 3D hand gesture recognition system [3]. In this paper we describe a gestural interface based on real-time hand tracking and 3D gesture recognition. The proposed system uses two color cameras for 3D reconstruction, 2D and 3D Kalman filters for noise elimination, and HMM for gesture recognition.

Two main concerns of gesture recognition are spatio-temporal variability and segmentation ambiguity. Spatio-temporal variability means that the same gesture can differ in shape and duration even for the same gesturer. HMM is inherently capable of modeling spatio-temporal time series. On the other hand, segmentation ambiguity arises, since the start and end points of a gesture are difficult to identify. We have constructed gesture models using left-right HMM, and re-estimated the parameters of each model with the Baum-Welch algorithm [4].

Another problem arising in real time gesture recognition systems is distinguishing meaningful input sequences from unrelated hand movements, i.e. gesture spotting. Lee and Kim proposed a method to provide an adaptive threshold for classification of gestures and nongestures using HMM. They developed an ergodic threshold model, which consisted of a combination of all the states of the models in the system [5].

On the other hand, the primary issues in 3D systems are the detection of the hand in the images, the reconstruction of the world coordinates of the user's hand and the elimination of the noise present in the system. Distinguishing the hand using the skin color is a complex problem, because of the existence of other body parts in the acquired images. Using markers reduces the complexity of hand detection considerably. Therefore we use markers to separate the hand from complex backgrounds under dynamic lighting conditions.

We use 3D reconstruction from stereo images to find the world coordinates of the marker. We also provide a tool for camera calibration, which is a necessary step for this method. Because of the noise in the image retrieval procedure we apply a Kalman Filter to smooth the hand trajectory found in the 2D images. Since 3D reconstruction process is very sensitive to noise, even the filtered 2D coordinates yield a significant error. Therefore we filter the reconstructed coordinates using a 3D Kalman filter. Experiments revealed that such a double filtering method enhances the recognition results considerably.

The overall system can be seen in Figure 1. We proceed to discuss each step in detail.

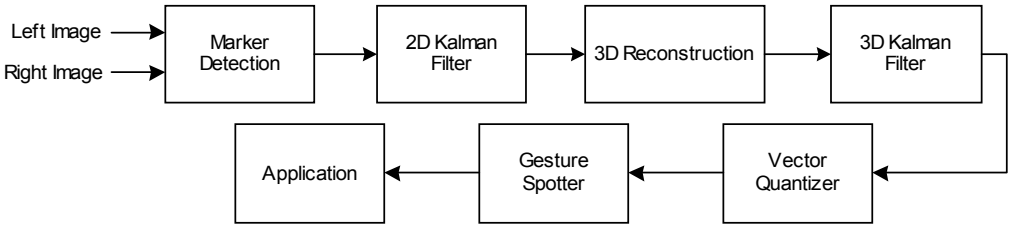

Figure 1 Flowchart of the Application

2. Real Time Finger Tip Tracking

2.1 Distinguishing The Marker

The user can select any uniform color for the marker, as long as it is not confusable with the skin color. When the user moves the marker after running the marker-detection utility, the different pixels of two consecutive scenes are obtained using a threshold. Assuming that the background is kept still, the generated map holds the pixels corresponding to the marker in the image. From this map the marker and its corresponding average hue component is found. We use this hue value for the segmentation.

2.2 Marker Segmentation

We employ connected components algorithm using double thresholding to find the marker region in images acquired from the video sequence. During the connection process the area of each region is calculated. Regions smaller than a threshold are regarded as noise and are ignored.

Fingertip detection is handled by extracting the simple shape descriptors of the hand. These descriptors are the bounding box and the four outmost points of the hand defining the box. The elongation of the bounding box is used to determine the mode of the hand and the points are used to predict the location of the fingertip for different modes of the hand.

Figure 2 Hand Modes

2.3 Filtering Marker Locations

We assume that between consecutively captured frames the marker is linearly moving at constant velocity, subject to random perturbations in its trajectory. This assumption is convenient since sampling is done in short time intervals. Using Kalman Filter on this model has given satisfactory results. Figure 3 illustrates the measured and filtered trajectory of the hand motion.

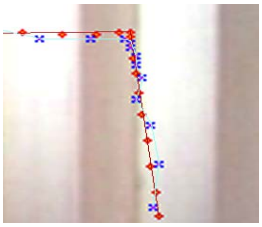

Figure 3 Effect of Kalman Filter

2.4 3D Reconstruction Of Fingertip Coordinates

For 3D reconstruction of the world coordinates, calibration matrices of both cameras are required. We have implemented a calibration utility within the system for a specific calibration object. The points are automatically found and matched using this utility.

Figure 4 Calibration Object

The world coordinates of the fingertip is generated by 3D reconstruction from stereo using a least squares approach. To smooth the trajectory 3D Kalman Filter is applied on the reconstructed coordinates.

3. Hand Gesture Classification

3.1 Gesture Spotter

To eliminate coordinate system dependence we transform the 3D coordinates of the marker in successive images into sequences of quantized

velocity vectors. HMM interprets these sequences, which are directional codewords characterizing the trajectory of the motion.

For a particular sequence to be recognized as a gesture, its likelihood should exceed that of the other models. Yet this condition is not sufficient; the candidate model's likelihood should also exceed some threshold to prevent recognition of nongestures. Setting a static threshold value is impractical, since recognition likelihoods of gestures varying in size and complexity fluctuate significantly. Therefore we have constructed an adaptive threshold model similar to the one proposed by Lee and Kim by fully connecting the states of all models.

The sequence size of a gesture performed by the user is not known in advance. Hence we regard each new input as a possible end point and check for candidate start points in an interval, whose limits are set by the user. The likelihoods of the models for each potential sequence are calculated. If the likelihood of a model exceeds that of the threshold model for one of these sequences, the corresponding gesture is recognized. (Figure 6) Otherwise classification is rejected.

4. Results

4.1 The Application

Common PC applications have a simple command interface that can be controlled via mouse selections or keyboard shortcuts. More advanced applications make use of the motion of the mouse as well, as in games or painting programs. We developed an interface that is designed to work with all such applications and linked it to a well-known painting program for experimentation.

The painting program chooses commands when the mouse clicks on the buttons and performs the commands when the mouse clicks on the canvas. In our interface gesture recognition is used to choose commands and hand tracking is used to simulate the mouse motion for drawing. The mouse click is performed by closing the fingers and extending the thumb. The pointer on the screen is controlled through the projection of the 3D coordinates to the 2D screen coordinates. Whenever a gesture is recognized the corresponding command is called by raising the related event or set of events.

The gestures can be trained and linked to the commands of the target application by the user, where the gesture-event associations are held in a configuration file. The interface has options to change the marker, train the gestures, calibrate the cameras and modify related parameters.

We have trained eight gestures and associated them with the commands given in Table 1. The second column of Table 1 shows the number of states of the corresponding HMM. While testing our system we discovered that very short sequences and sequences with less than three distinct symbols may erroneously have higher likelihoods than the threshold model, which leads to misclassifications. Therefore we do not attempt to classify such sequences.

Figure 6 shows the likelihood of the models over time for a particular gesture. For this example, the gesture is recognized to be the triangle, the gesture for brush. Table 1 shows the recognition performance of the system in 160 tests. There were 2 misclassifications, yielding a recognition performance of 98.75%.

Another remarkable observation was that the recognition occurred 98% of the times for the smallest possible sequence in the interval. This suggests that with a negligible loss in reliability the performance of the system can be increased by discarding longer sequences in the interval once the conditions discussed above are satisfied.

5. Conclusion

In this study we have worked on real time 3D hand gesture recognition. First we have applied several image processing algorithms to detect markers in bitmap images acquired from simple web cameras. Then we have applied 3D reconstruction from stereo images. Next we have trained HMMs to recognize different gestures using the input sequences. In addition we have implemented an adaptive threshold model for gesture spotting. Combining all this work, we have developed a simple and robust framework where we can use gestures as inputs for interfaces for various PC applications.

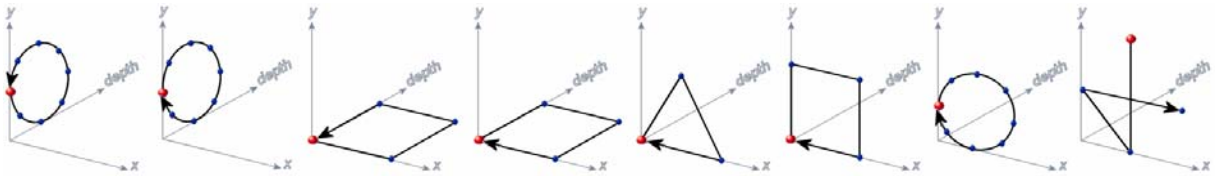

Figure 5 Gestures

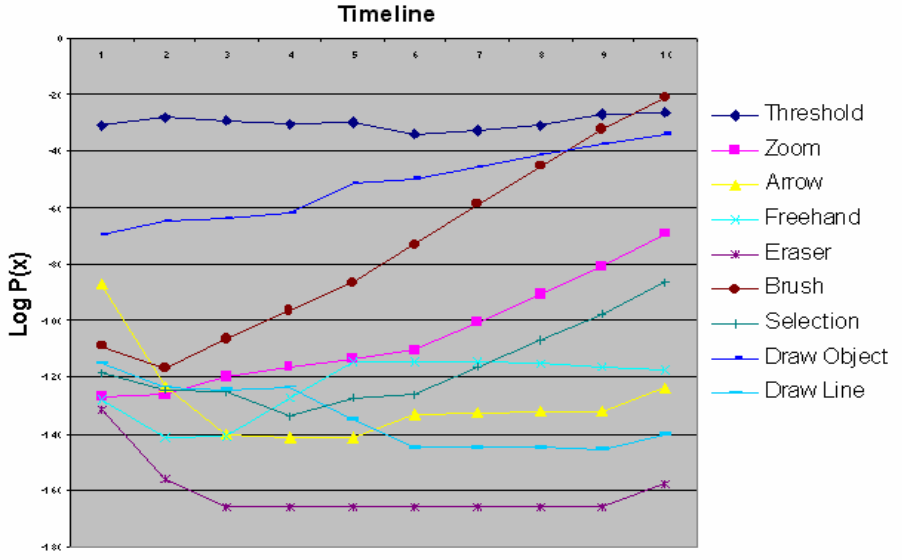

Figure 6 Likelihoods of the Models

Gesture	# States	# Training	# Trials	# Correct	# Wrong	Rate %	Tool
	8	65	20	20	0	100	Zoom
	4	56	20	20	0	100	Arrow
	8	59	20	20	0	100	Freehand
	4	59	20	20	0	100	Eraser
	3	60	20	20	0	100	Brush
	4	56	20	19	1	95	Selection
	8	57	20	19	1	95	Draw Object
	3	57	20	20	0	100	Draw Line

Table 1 Descriptions and Recognition Rates of Gestures

REFERENCES

[1] T. Starner and A. Pentland, "Real Time American Sign Language Recognition from Video Using Hidden Markov Models," Technical Report TR-375, MIT's Media Lab., 1995

[2] K. Oka, Y.Sato and H. Koike, "Real-Time Fingertip Tracking and Gesture Recognition", *IEEE Computer Graphics and Applications*, November-December, 2002, pp. 64-71.

[3] Y. Sato, M. Saito, and H. Koike, "Real-time input of 3D pose and gestures of a user's hand and its applications for HCI," *Proc. 2001 IEEE Virtual Reality Conference (IEEE VR2001)*, pp. 79-86, March 2001.

[4] L.R Rabiner, "A tutorial on Hidden Markov Models and Selected Applications in Speech Recognition," *Proc IEEE*, vol.77, pp.257-285, 1989

[5] H. Lee and J.H Kim, "An HMM-Based Threshold Model Approach for Gesture Recognition," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 21, no. 10, pp. 961-973, 1999.