Virtual 3D Model Construction with Bare-Hand-Based Interaction

Zhen Zeng[1] and Lu Hong[2]

[1] Department of Electrical Engineering, University of Michigan at Ann Arbor
[2] Department of Computer Science, University of Michigan at Ann Arbor

Abstract. Bare-hand gesture recognition is an important task in human-computer interaction applications. In this paper, we propose a novel two-level approach to efficiently perform bare-hand tracking and gesture recognition. The lower level of the approach implements the posture recognition by applying SVM to the features extracted from fingertips detection. The higher level of the approach implements the gesture recognition using Kalman filter to classify different gestures in the same category of postures.

Keywords. Human-computer interaction, gesture recognition, bare-hand detection, fingertip detection, kalman filter

1 Introduction

Human-Computer interaction has been restricted in graphic display, keyboard and mouse for a long time. But it is not a natural way to use keyboards and mice in human interaction. For example, in the field of architecture design, the sketches drawn by hand are hard to convert into the computer to construct the 3D architecture model. To achieve natural human-computer interaction, the human hand could be considered as an input device. However, hand-based interaction, an intuitive way in human interaction, is still facing different difficulties. Existing approaches involve a wide variety of methods ranging from wearable targets and sensors to electrostatic fields and multiple cameras. But the devices, such as 3D mice and cyber-gloves, used in these approaches are usually too expensive and inconvenient to be widely used in people's daily life.

In this project, aiming to use only one general camcorder to capture the hand's movements, we focus on a real time bare-hand gesture recognition system. Bare-hand means that no device and no wires are attached to the user, who controls the computer directly with the movements of user's hand. Our goal is to develop an application which could efficiently perform bare-hand tracking and gesture recognition. And the mechanism of our system is as shown in Fig. 1.

Our gesture recognition technique can be applied in architecture design, which allows designer to use bare-hand gesture to construct 3D architecture mode. For example, a designer can use bare-hand gesture and labels to construct a 3D model according to a layout plan.

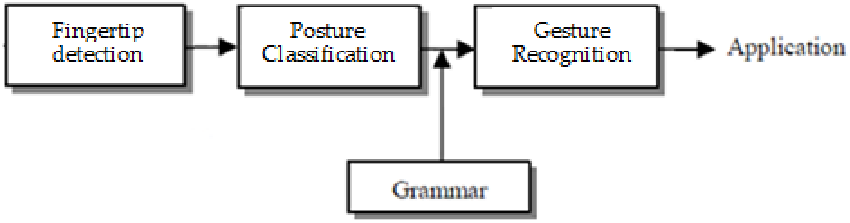


Fig. 1. Overview of our gesture recognition system: for each input frame from the video feed, we first detect the fingertips of each hand within the frame, and classify the posture according to the geometry attributes of the fingertips, such as locations and orientations. Gesture is recognized by applying kalman filter based on the model built on our defined gestures. Further human-computer interaction application could be built based on our system

2 Related Work

Gesture recognition approaches are divided into two categories – vision based approaches and electronic sensor based approaches. Vision based approaches could be further dived into two categories – 3D hand model based approaches and appearance based approaches [11]. A number of researchers have explored the use of hand gestures as a means of computer input, using a variety of technologies.

Implementation issues in bare-hand posture recognition include bare-hand segmentation, fingertip detection and posture classification. Bare-hand segmentation is the first step of posture recognition, which segments the hand out of the image. There are two proposed methods to solve this problem. One is background subtraction, which is also known as image differencing [1,9,12]. The other is skin detection, which represents skin and background pixels in a certain color space, and builds histogram-based classifier to classify skin and background [3,5,6,7,8]. Our method in bare-hand segmentation is inspired by skin detection. After segmentation, fingertip detection is performed. Hand contour matching [12] can be used to find fingertips. Another general method is fingertip shape finding [1,9], which uses a search square and an inner circle to find the shape similar to a fingertip.

In hand gesture recognition, the major concern is tracking. Context-free grammar-based syntactic analysis [2] and skeleton of hand [10] are two novel methods to perform gesture recognition. In addition, Kalman filter [8,13,14] is good at tracking objects and also can be modified as a classifier to perform gesture classification. Our work in gesture recognition is based on Kalman filter, which increases the tolerance of noise in real time. The main improvement in our algorithm is that we first predict the location of the fingertip after bare-hand segmentation, so that the fingertip shape geometry constraint needs only to be applied on the predicted region instead of the whole bare-hand area. Thus our gesture recognition system executes very fast, which is a critical fact in real time application.

3 Technical Part

We present an efficient approach to recognize bare-hand gestures. The features that we use for representation are the geometry attributes of the detected fingertips. Then posture classification gives corresponding gesture hypothesis, and Kalman filter is used to update the probability of each hypothesis through the video sequences. Gesture is determined by choosing the hypothesis with the highest probability.

3.1 Low level hand posture recognition

3.1.1 Fingertip detection

Bare-hand is first segmented from the background by skin-tone detection [3,5,7]. By plotting the distributions of R-G, and G-B for both hand and background (Fig. 2(a)), we got the criteria for segmentation as

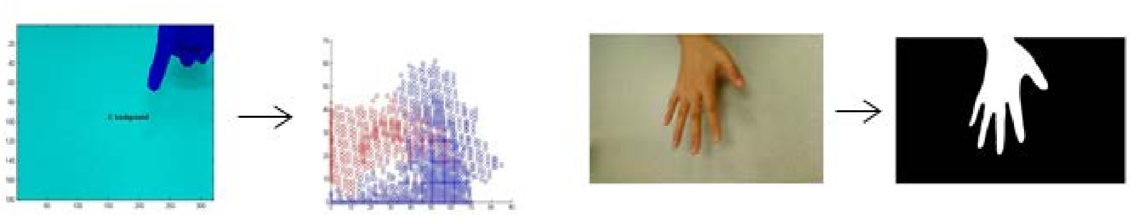
$$R>40 \ \& \ G>20 \ \& \ B>10 \ \& \ R-G>30 \ \& \ R>B$$

To smooth the binary image, we apply mean filter to get rid of the glitches around the edge of the hand (Fig. 2(b)). We narrow down the search region of the fingertip shape finder by first predict the fingertip location (Fig. 3).

The final representation of the hand features includes: number of hands; number of detected fingertip on each hand; radius and location of each fingertip; the orientation (direction pointing from fingertip to palm center) of each fingertip. And they are passed on to the next stage as explained in detail in the following section.

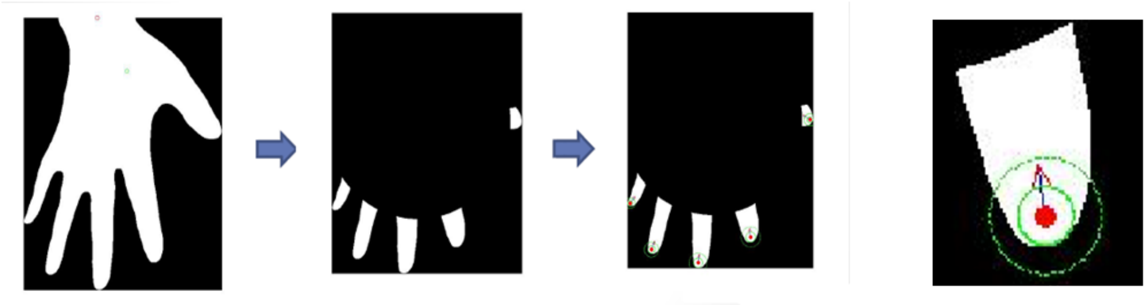
3.1.2 Posture classification

In our implementation, there are five different categories of postures: one tip, one hand with two tips, one hand with more than two tips, two hands with one tip per hand and two hands with more than two tips per hand. The features of each posture output by fingertip detection include position, rotation and radius of each detected fingertips. To classify postures into these five categories, we apply one-versus-all SVM to these features. The tested gestures and postures are as defined in Table 1.



(a) (b)

Fig. 2. (a) Manually label the background and bare-hand. X-axis is R-G, and Y-axis is G-B. Blue dots are hand, and red dots are background; **(b)** Sample result of bare hand segmentation in a traditional lighting condition.



(a) **(b)**

Fig. 3. (a) ***Left***: find wrist center by calculating the middle points of the crossing line of wrist; find connected regions (blob) in binary image, which indicates potential hands; palm center is defined as centroid of each blob; ***Middle***: cut hand with circles centered at palm center to leave fingers only; a series of radius are used, and we choose the one which gives the largest number of similar area fingers; ***Right***: we draw a vector from the palm center to the finger center, and the intersection between the vector and the edge of the finger is defined as the predicted fingertip location; radius of the fingertip is defined as half of the smallest distance from the finger center to the edge point of the finger. **(b)** Visualization of an example of a detected fingertip (marked as red dot), and the estimated radius; Then we verify each predicted fingertip by calculating the filled percentage of the small and large green circles drew around the fingertip. If *filled percentage of small circle* >0.83 & 0.35< *filled percentage of large circle* <0.77, then it is marked as a fingertip. Our approach of fingertip shape finder is inspired by the previous works [1, 7].

3.2 High level hand gesture recognition

3.2.1 Kalman Filter

The Kalman filter is a recursive estimator, which means that only the estimated state from the previous time step and the current measurement are needed to estimate the current state. This property of Kalman filter enables it to be used in motion tracking, in which case we can consider the estimated state as estimated position of moving object and the current measurement as the actual position of moving object.

Let y_t represent the measurement and x_t represent the estimate state at time step t. For the standard Kalman filter, the state transition from *t* to *t+1* can be expressed with the equation

$$x_{t+1} = Ax_t + \omega_t \tag{1}$$

where A is the state transition matrix and ω_t is a noise term. This noise term is a Gaussian random variable with zero mean and a covariance matrix Q, so its probability distribution is

$$p(\omega) \sim N(0, Q) \tag{2}$$

The measurement y_t can be expressed in the below equation

$$y_t = Hx_t + v_t \tag{3}$$

where H is an matrix which relates the state to the measurement. Much like ω_t, v_t is the noise of the measurement, which is also assumed to have a normal distribution expressed by

$$p(v) \sim N(0, R) \tag{4}$$

There are two steps, time update step and measurement update step within each recursion of standard Kalman filter algorithm. *Time update step:*

$$\hat{x}_{t+1|t} = A\hat{x}_{t|t} \tag{5}$$

$$P_{t+1|t} = AP_{t|t}A^T + Q \tag{6}$$

where $\hat{x}_{t+1|t}$ is the mean of x_{t+1} conditioned on $y_0, y_1, \dots, y_t$, $P_{t+1|t}$ is the associated covariance matrix of $\hat{x}_{t+1|t}$. *Measurement update step:*

$$K_{t+1} = P_{t+1|t}H^T\left(HP_{t+1|t}H^T + R\right)^{-1} \tag{7}$$

$$\hat{x}_{t+1|t+1} = \hat{x}_{t+1|t} + K_{t+1}(y_{t+1} - H\hat{x}_{t+1|t}) \tag{8}$$

$$P_{t+1|t+1} = P_{t+1|t} - K_{t+1}HP_{t+1|t} \tag{9}$$

where K_{t+1} is the Kalman gain matrix.

The Kalman filter model requires a representation of the moving object in state space. Two obvious parameters required in each state are the p_x and p_y coordinates of the moving object. In addition, the state will include the velocity components v_x and v_y, and the acceleration components a_x and a_y. Since we need the difference of radius and orientation of fingertips to distinguish some gestures, we include d_r and d_o in the state. So the state at any point can be represented with the vector $\left[p_x, p_y, d_r, d_o, v_x, v_y, a_x, a_y\right]^T$. The state transition matrix is derived from the theory of motion which can be expressed with the equations:

$$\begin{aligned} p(t+1) &= p(t) + v(t)\Delta t + \frac{a(t)\Delta t^2}{2} \\ v(t+1) &= v(t) + a(t)\Delta t \\ a(t+1) &= a(t) \end{aligned} \tag{10}$$

where $p, v, a, \Delta\mathrm{t}$ are position, velocity, acceleration and the time difference between frames. We also assume that $\Delta t = 1$.

As shown in Table 1, there is only one gesture in one hand with two tips and two hands with more than two tips per hand, which means Reset and Zoom can be recognized during posture recognition process. All we need to do in gesture recognition is to classify three gestures in one tip category, two gestures in one hand with more than two tips category and two hands with one tip per hand category.

Posture	Gesture	Description	Meaning
One tip	Double Click	Double click with one finger on the desk plane.	Double clicking on a rectangle produces a square.
	Lift	Move one finger from the desk plane to higher place.	Lifting a rectangle or a square produces a cuboid and a circle produces a cylinder.
	Rotate	Draw an arc using one finger on the desk plane.	Rotate a 3D object.
One hand with two tips	Zoom in and Zoom out	Two fingers in one hand moving towards means Zoom in and moving outwards means Zoom out.	Zoom in or Zoom out a rectangle.
One hand with more than two tips	Throw	Put one hand on a 3D object and quickly move the hand out of the scene.	Throwing a 3D object erases the object from the scene.
	Move	Put one hand on a 3D object and then slowly raise the object, move to another place and put the object down.	Move a 3D object from one place to another place.
Two hands with one tip per hand	Rectangle	Two fingers of each hand move outwards on the desk plane.	Draw a rectangle on the desk plane.
	Circle	Two fingers of each hand draw a circle on the desk plane.	Draw a circle on the desk plane.
Two hands with more than two tips per hand	Reset	Two hands put on the desk plane.	Clear all objects in the scene.

Table 1. We define 10 gestures out of 5 posture categories in total

For each gesture, which can't be recognized after posture recognition, we apply different Kalman filters with different parameters to it. Here are some common parameters applied to different Kalman filters in our implementation:

$$Q = 0.01I_{8\times8}, P = 100I_{8\times8}, H = [I\ 0]_{4\times8}$$

$$R = \begin{bmatrix} 0.2845 & 0.0045 & 0.0045 & 0.2845 \\ 0.0045 & 0.0045 & 0.0045 & 0.0045 \\ 0.0045 & 0.0045 & 0.0045 & 0.0045 \\ 0.2845 & 0.0045 & 0.0045 & 0.2845 \end{bmatrix}$$

$$y = \left[p_x, p_y, d_r, d_o\right]^T$$

One Tip. *Lift.* Lift can be considered as uniform motion in a straight line. Recalling the definition of the state transition as $x_{t+1} = Ax_t + \omega_t$, we can set A as A_{lift}. And set the initial state as x_{lift1}. *Double Click.* We assume that the difference of fingertips' radius is opposite in each time step since the finger moves towards and outwards the camera in Double Click. To model the motion in Double Click, we set the acceleration as the opposite of half of the initial velocity. *Rotate.* To

$$A_{lift} = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 1/2 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1/2 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$x_{lift1} = \left[p_{x1}, p_{y1}, d_{r1}, 0, p_{x1} - p_{x0}, p_{y1} - p_{y0}, 0,0\right]^T$$

$$A_{double} = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 1/2 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1/2 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$x_{double1} = \left[p_{x1}, p_{y1}, d_{r1}, 0, p_{x1} - p_{x0}, p_{y1} - p_{y0}, -\frac{(p_{x1} - p_{x0})}{2}, -\frac{p_{y1} - p_{y0}}{2}\right]^T$$

$$A_{rotate} = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 1/2 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1/2 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$x_{rotate1} = \left[p_{x1}, p_{y1}, d_{r1}, d_{o1}, p_{x1} - p_{x0}, p_{y1} - p_{y0}, 0,0\right]^T$$

Fig. 4. Model parameters of posture lift and rotate

distinguish Rotate from Lift, we assume that the difference of rotation in Rotate is a constant rather than 0.

One Hand with More than Two Tips. Since in Throw the hand moves very fast, it's hard to track using Kalman filter. However, we can easily distinguish Throw from Move by simply comparing the velocity.

Two Hands with One Tip per Hand. In this category, we use the same Kalman filter in each hand. *Rectangle****.*** The transition matrix and initial state are the same as those in Lift, which means that $A_{rectangle} = A_{lift}$ and $x_{rectangle1} = x_{lift1}$. We apply Kalman filter to each hand. *Circle.* The transition matrix and initial state are the same as those in Rotate, which means that $A_{circle} = A_{rotate}$ and $x_{circle1} = x_{rotate1}$. We apply Kalman filter to each hand.

3.2.2 Gesture recognition

In each category, given a set of gestures$\{G_i\}, i = 1, \ldots, N, N = 2 \; or \; 3$, the posterior distribution over the gestures at time step *m* can be expressed by Bayes Rule,

$$p(G_i|y_1, y_2, \dots, y_m) = p(G_i) \prod_{t=1,\dots,m} p(y_t|G_i) \tag{11}$$

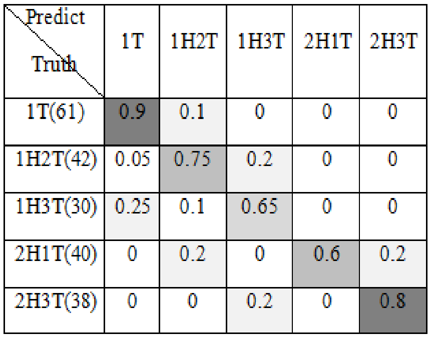
where y_t is the observation. If we have no information about the correctness of the gestures from the first frame, the prior probability $p(G_i)$ in Equation 11 is uniformly distributed over all the gestures. For each time step, the likelihood of observation y_t is modeled by a normal distribution with mean at predicted state $\hat{x}_{t|t}$. So the likelihood of gesture G_i at time step t is,

$$p(y_t|G_i) = \prod_{j=0}^{n} exp - \frac{\|\hat{x}_{t|t} - y_t\|^2}{2\sigma^2} \tag{12}$$

4 Experiment Results

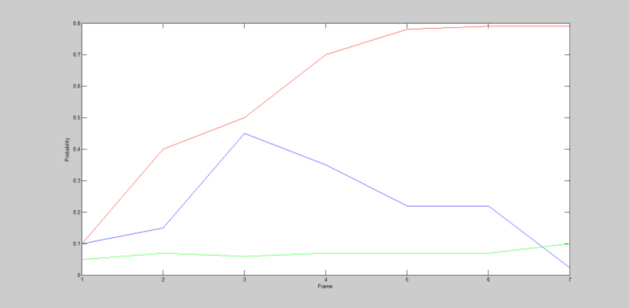
4.1 Posture Recognition

The execution time of fingertip features extraction is approximately 0.07s, and posture classification only takes about 0.0033s, which makes our application capable of real time performance. As shown in Fig. 5(a), the accuracy of posture classification is high. But there are still confuses with other postures and confuses usually happen between similar postures. For example, confuses happen between one tip and one hand with two tips when the fingertip detection only detects one tip in the latter case. Also the experiment result above shows that the accuracy becomes lower as the number of tips grows. This may result from that when the number of tips grows; the noise grows which leads to lower accuracy.



Predict Truth	1T	1H2T	1H3T	2H1T	2H3T
1T(61)	0.9	0.1	0	0	0
1H2T(42)	0.05	0.75	0.2	0	0
1H3T(30)	0.25	0.1	0.65	0	0
2H1T(40)	0	0.2	0	0.6	0.2
2H3T(38)	0	0	0.2	0	0.8

(a)

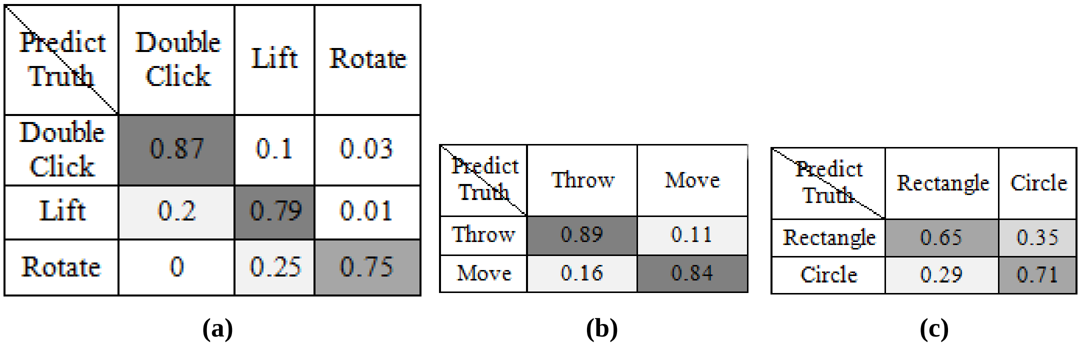


(b)

Fig. 5. (a) Confusion matrix of posture classification. 1T = One Tip, 1H2T = One Hand with Two Tips, 1H3T = One Hand with More than Two Tips, 2H2T = Two Hands with One Tip per Hand, 2H3T = Two Hands with More than Two Tips per Hand. The number in the bracket is the number of test cases. **(b)** Posterior Probability Distribution after each frame in one tip category

4.2 Gesture Recognition

We tested our approach on 101 videos. As shown in Fig. 5(b), all gestures are equally likely in the first frame. Gestures with low accuracy drop significantly in the next few frames, while one with the highest accuracy gradually stands out among the rest. As shown in Fig. 6, the values in the diagonal are comparably high, which represent the accuracy of gesture recognition. But there are still confuses with other. The accuracy in Two Hands with One Tip per Hand is lower



Predict / Truth	Double Click	Lift	Rotate
Double Click	0.87	0.1	0.03
Lift	0.2	0.79	0.01
Rotate	0	0.25	0.75

(a)

Predict / Truth	Throw	Move
Throw	0.89	0.11
Move	0.16	0.84

(b)

Predict / Truth	Rectangle	Circle
Rectangle	0.65	0.35
Circle	0.29	0.71

(c)

Fig. 6. **(a)** Confusion matrix of gestures in One Tip posture category**;** **(b)** Confusion matrix of gestures in Two Hands with One Tip per Hand posture category; **(c)** Confusion matrix of gestures in One Hand with more than two Tips per Hand posture category

than other categories. The reason may be that we use the same Kalman filter on each hand, which results in a lower combining accuracy.

5 Conclusion

Gesture recognition is an important topic in computer vision, and lots of human-computer interaction could benefit from this technique. Tremendous works have been done previously, and many of them are taking advantage of wearable devices, motion sensors, multiple cameras and depth camera. From the human-computer interaction point of view, it is crucial to develop bare-hand tracking and gesture recognition algorithm, so that user could get rid of the inconvenient devices and save some space at the same time. In order to achieve this, we represent the features of the bare hand in a novel way. The number, location, radius, orientation of fingertips are the attributes that we used. And the posture classification and gesture recognition based on kalman filter based on the feature representation give us promising result. While there is still some confusion between the gestures belonging to the same posture category, in order to get more accurate models for kalman filter, we could extract the features of each frame from the test videos and estimate the corresponding parameters of each model. And to make our algorithm more robust, we need to test it on more datasets with different lighting conditions, background, and people from different races. Finally, an useful 3D simple prototype building application could be developed based on our gesture recognition system.

6 References

[1] Von Hardenberg, C. and Bérard, F. Bare-hand Human Computer Interaction. Workshop on Perceptive User Interfaces, Orlando, Florida, 2001.

[2] Qing Chen, Nicolas D. Georagnas, Emil M. Petriu. Real-time Vision-based Hand Gesture Recognition Using Harr-like Features. IMTC, Warswa, Poland, 2007.

[3] Pushkar Dhawale, Masood Masoodian, Bill Rogers. Bare-Hand 3D Gesture Input to Interactive Systems. SIGCHI, New York, USA, 2006.

[4] Khalid Youssef. Bare Hand Tracking Comprehensive Literature Review.

[5] Teofilo Emidio de Campos. 3D Hand and Object Tracking for Intention Recognition. DPhil Transfer Report, 2003.

[6] Nasser H. Dardas, Mohammad Alhaj. Hand Gesture Interaction with a 3D Virtual Environment. The Research Bulletin of Jordan ACM, ISSN: 2078-7952, Volume II (III), P86-94.

[7] Elena Sanchez-Nielsen, Luis Anton-Canalis, Mario Hernandez-Tejera. Hand Gesture Recognition for Human-Machine. WSCG, 2004.

[8] Thomas Coogan, George Awad, Junwei Han, Alistair Sutherland. Real Time Hand Gesture recognition Including Hand Segmentation and Tracking. ISVC, 2006.

[9] Julien Letessier, Francois Berard. Visual Tracking of Bare Fingers for Interactive Surfaces. UIST, 2004.

[10] Bogdan lonescu, Didier Coquin, Patrick Lambert, Vasile Buzuloiu. Dynamic Hand Gesture Recognition Using the Skeleton of the Hand. EURASIP, 2005.

[11] Pragati Garg, Naveen Aggarwal and Sanjeev Sofat. Vision Based Hand Gesture Recognition. World Academy of Science, Engineering and Technology, 2009.

[12] Duanduan Yang, Lianwen Jin, Junxun Yin. An Effective Robust Fingertip Detection Method for Finger Writing Character Recognition System. 4th International Conference on Machine Learning and Cybernetics, Guangzhou, 2005.

[13] B.Stenger, P.R.S. Mendonca and R. Cipolla. Model-Based Hand Tracking Using an Unscented Kalman Filter. British Machine Vision Conference, volume I.

[14] William F. Leven and Aaron D. Lanterman. Unscented Kalman Filters for Multiple Target Tracking with Symmetric Measurement Equations. Proceedings of SPIE, the International Society for Optical Engineering, Vol. 5810, pp. 56-67.