A Virtual Mouse Interface Based on Two-layered Bayesian Network

Myung-Cheol Roh, Sung-Ju Huh, and Seong-Whan Lee
Department of Computer Science and Engineering, Korea University,
Anam-dong, Seongbuk-ku, Seoul 136-713, Korea
{mcroh, sjhuh, swlee}@image.korea.ac.kr

Abstract

Recently, many studies on gestural control methods for substituting for keyboard and mouse devices have been conducted because of their conveniences and intuitiveness. This paper presents a Virtual Mouse interface which is a gesture-based mouse interface and Two-layered Bayesian Network (TBN) for robust hand gesture recognition in real-time. The TBN provides robust recognition of hand gestures, as it compensates for an incorrectly recognized hand posture and its location via the preceding and following information. Experiments demonstrate that the proposed model recognizes hand gestures with a recognition rate of 93.78% *and* 85.15% *for a simple and cluttered background, respectively.*

1. Introduction

The most general interaction devices used to control computers are keyboard and mouse. Recently more intuitive and convenient interaction methods have been proposed by many researchers, such as speech, gestures, etc. Gestures, especially hand gestures are a very powerful, expressive and intuitive means of communication. Consequently, significant and diverse hand gesture recognition studies have recently been conducted for Human-Computer Interaction (HCI) [1].

There are several challenging issues in applying a gesture-based interface for HCI. First of all, a gestural interface for HCI should deal with a complex background, which makes hand detection and tracking difficult. In terms of usability, the gesture commands should be simple and intuitive for users. Many previous studies have proposed various grammars and rules for gestures. One-handed gestures usually use the trajectory of the hand, the number of fingers, and the hand postures [2]. On the other hand, two-handed gestures usually use one hand for pointing, and the other for commanding [3]. Finally, it must be capable of performing in real-time and guaranteeing reliable accuracy.

This paper presents the Virtual Mouse interface, a gestural interface for a mouse device. The proposed interface provides the solution for real-time recognition of hand gestures in a cluttered environment. We propose a new probabilistic model to recognize hand gestures for a Virtual Mouse interface. The proposed model is composed of a two-layered framework: the lower-layer models hand postures with Bayesian Network (BN) and the upper-layer models hand gestures with Dynamic Bayesian Network (DBN). This hierarchical model provides a way to decouple the dependency between features for postures and gestures. Thus, the computational complexity of probabilistic inference and the number of training data can be reduced as compared with the standard HMM. Moreover, although a hand posture and its location may not be correct, the upper-layer DBN can complement these incorrectly recognized hand postures by time information. Therefore, this two-layered architecture provides more reliable performance, even in a cluttered environment.

2. Related Work

The increased demands for more convenient and intuitive interfaces impelled the introduction of numerous types of interfaces via diverse interaction methods such as speech, gestures, brain waves, etc. Of these methods, the gestural interface has been the most widely investigated, because of its convenience and practicality.

Y. Fu and T. S. Huang proposed the hMouse, a head tracking-driven virtual computer mouse [4]. They tracked and estimated a 3D head pose to control a mouse device. P. Robertson, et al. presented a vision-based virtual mouse interface for a smart room environment [5]. Utilizing a robotic head, their system tracked the user's head and hand position, and recognized hand signs to control an intelligent kiosk. They introduced a state machine for a virtual mouse, to track and recognize hand signs in real-time. L. Bretzner, et al. presented a prototype system for a vision-based HCI [2]. By recognizing hand gestures, their system executed several commands for a computer or a smart room; the user could turn on/off a television, change the channel of the television, and toggle the lamp on/off. A. Argyros and

978-1-4244-5498-3/09/$25.00 ©2009 IEEE

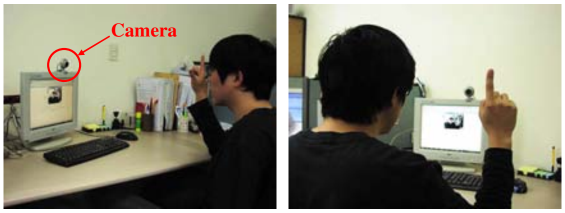

Figure 1. The environment for the proposed interface

A. Lourakis presented a vision-based interface to control a computer mouse device via hand gestures [3]. Using a reliable hand tracking algorithm with finger-tip detection, they achieved accurate and smooth movement of a mouse cursor, as well as recognition of gestures that activated mouse events. They suggested a two-handed gestural interface to solve the ambiguity of gesture spotting.

3. Proposed Interface and Method

The proposed virtual mouse interface provides a way to mimic a mouse device via hand gestures, using a single hand. The only device this interface needs is a camera for capturing the input image sequences. An example of the environment for the proposed interface is illustrated in Fig. 1. The input image sequences are captured using a camera device, and the captured image sequences contain the user's upper body, including his or her face. The region being recognized is detected and tracked using image processing methods, and visual features are extracted for recognizing hand gestures. A Two-layered Bayesian Network (TBN) is proposed to handle the visual features efficiently. TBN is a hierarchical version of the Dynamic Bayesian Network (DBN), which can recognize hand gestures in two hierarchical stages: lower and upper layers.

3.1. Hand detection and Tracking

Since the hand detection technique supports initialization of the hand position and updates the new position of the hand when tracking by a hand tracker fails, detecting the hand is important for a hand gesture recognition system. In our method, we used Bayes SPM [6], because it is simple and it can models not only the skin color but also the non-skin color, and as a result it has reliable skin color detection.

We assume that the hand region being detected is the biggest skin-colored blob, excluding the head region. In order to detect a face, we used Viola and Jones' Adaboost based detector [7] which uses a boosting algorithm based on AdaBoost. Once a face is detected the head region is eliminated from the skin color image derived from the skin color detector. After morphological operations and connected component analysis, a largest blob is found, which is presumed to be the hand region.

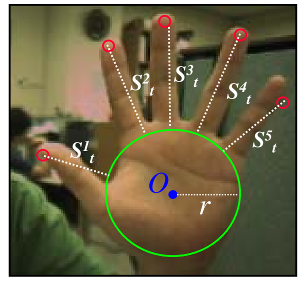

Figure 2. A visual representation of features set S_t

Initializing using the preceding detected hand region, we applied existing tracking algorithms to track the hand region over time. We apply the Bayesian Multiple-blob tracker in [8] to track the hand region. The author's tracker is based on a Bayesian solution which estimates the probability of the state of objects at time t X_t given the observation sequence from time s to time t $Z_{s:t}$.

3.2. Feature Extraction

We used two kinds of features: local and global features. The local features denote salient ones that represent the shape of the hand, whereas, global features denote temporal ones that represent the temporal evolution of hand shapes and movements.

In order to recognize hand postures, a set of local visual features from the region of the detected and tracked hand are selected. Intuitively, hand postures are composed of a variety of combinations of fingers. Thus, features which represent fingers' characteristics should be chosen. Therefore, we selected the following local features set S_t:

$$S_t = \{S_t^1, S_t^2, S_t^3, S_t^4, S_t^5\} \tag{1}$$

where $S_t^1, S_t^2, S_t^3, S_t^4,$ and S_t^5 denote the normalized length of the thumb, index, middle, ring and little fingers, respectively, at time t. Each feature of each finger is computed and normalized between 0 and 1 as follows:

$$S_t^i = \alpha^i \times \frac{|\|\overrightarrow{OF^i} - r\||}{r} \tag{2}$$

where α^i denotes the normalized constant for each finger, $\|\overrightarrow{OF^i} - r\|$ denotes the Euclidian distance between the hand center point O and the finer tip point F^i of finger i ($i \in \{1, 2, 3, 4, 5\}$), and r denotes the radius of the palm. A visual representation of the features set S_t is described in Fig. 2.

The hand center point O and radius r are extracted using the blob detection method in [9] and finer tip points F^i can be extracted via the finger detection method in [3].

For global visual features, which represent global movements of the hand being recognized, we calculate the motion orientation histogram [10]. The motion orientation histogram is constructed by applying the gradient operator to the Motion History Image (MHI). Motion orientation of MHI θ is calculated as follows:

$$\theta(i,j) = \tan^{-1} \frac{I_y(i,j)}{I_x(i,j)} \qquad (3)$$

where I_x and I_y are the image gradients of HMI in the x and y directions, and i, j are the pixel coordinates of MHI.

After calculating the motion orientations for MHI, the motion orientation histogram can be constructed to represent global features. To construct the motion orientation histogram, motion orientation is calculated using equation (3), and the orientation image is divided into eight bins in polar coordinates, then, the orientations of points in each bin are summed. Thus, motion orientation histograms can be used as a one-dimensional vector of length eight, which represents global features for hand gesture recognition.

3.3. The Two-layered Bayesian Network

TBN is based on DBN and its extensions. DBN is known as one of the most expressive state-space models for recognizing various types of sequential data, including speech, gestures, etc [11].

In TBN, hand gestures are modeled in hierarchical procedures. Hand gestures are composed of the sequence of hand postures and their movements, so hand postures can be interpreted as an interim parameter between hand gestures and visual features. Thus, the hand gesture recognition problem can be decomposed into some stages. In terms of this aspect, we decompose into two-layer, one which of represents hand postures and is denoted the lower-layer and the other of which represents hand gestures and is denoted the upper-layer. The lower-layer consists of BN, which receives as input local visual features extracted from the detected and tracked hand region, and outputs the hand posture recognition results. In the lower-layer, hidden states represent the hand posture being recognized. The upper-layer is composed of DBN, whose inputs are the output from the lower layer and the global features of the hand movement,and whose output is the hand gesture recognition results. DBN describes the temporal evolution of hand postures from the lower-layer. In other words, the probabilities of hand postures, which are the outputs of the lower-layer, are vectorized as inputs of DBN, the upper-layer.

To model hand postures, we propose a particular form of BN for the lower layer. As shown in Fig. 3, the proposed BN has one hidden state node for the hand posture X_t and five hidden state nodes for the five fingers whose dependencies are independent of each other at each time instance. All five nodes for fingers $\{F_t^{index}$, F_t^{thumb}, F_t^{middle}, F_t^{ring}, $F_t^{little}\}$ have observation nodes $\{O_t^{index}$, O_t^{thumb}, O_t^{middle}, $O_t^{ring}, O_t^{little}\}$ implying the observation of five different fingers. A hidden state node for the index finger F_t^{index} is separated from the other fingers because this is the most important and frequently used finger of all. Unlike the nodes of the other fingers, the node of the index finger has an arc from the previous time slice $t-1$, meaning the state transition of the index finger. This state transition prevents abrupt changes of the index finger state over time, arising from the poorly extracted features in the image processing phase. For notational simplicity, we denote F_t^{index} and $[F_t^{thumb}, F_t^{middle}, F_t^{ring}, F_t^{little}]^T$ as F_t^1 and F_t^2, respectively.

The goal of BN in the lower-layer is to infer the state of the hidden node X_t using the observed the states of the five nodes $\{O_t^1, O_t^2\}$. The probability distribution $P(X_t|O_{1:t}^1, O_{1:t}^2)$ is formulated as:

$$P(X_t|O_{1:t}^1, O_{1:t}^2) = \frac{P(X_t, O_{1:t}^1, O_{1:t}^2)}{P(O_{1:t}^1, O_{1:t}^2)} \qquad (4)$$

The denominator of the right-hand term in equation (4) can be canceled because it is a constant. As shown in the lower layer in Fig. 3, there are intermediate nodes $F_{1:t}^1, F_{1:t}^2$ between X_t and $O_{1:t}^1, O_{1:t}^2$. These nodes represent the states of flexions of fingers. Then, $P(X_t|O_{1:t}^1, O_{1:t}^2)$ can be simplified and represented as follows:

$$\begin{aligned} P(X_t|O_{1:t}^1, O_{1:t}^2) \quad \propto \quad & \sum_{F^1}\sum_{F^2} P(O_t^1|F_t^1)P(O_t^2|F_t^2) \\ & P(F_t^1|X_t, F_{1:t-1}^1)P(F_t^2|X_t) \\ & P(X_t|O_{1:t-1}^1, O_{1:t-1}^2). \qquad (5) \end{aligned}$$

In the equation (5), $P(X_t|O_{1:t-1}^1, O_{1:t-1}^2)$ is calculated as $\sum_{X_{t-1}} P(X_t|X_{t-1})P(X_{t-1}|O_{1:t-1}^1, O_{1:t-1}^2)$ where X_t is independent of $O_{1:t-1}^1, O_{1:t-1}^2$ in Fig. 3 and $P(X_t|X_{t-1})$ is the state transition probability of the posture state. $P(F_t^1|X_t, F_{1:t-1}^1)$ can be simplified as $(F_t^1|X_t)P(F_t^1|F_{t-1}^1)$ where F_t^1 is dependent only on F_{t-1}^1. Note that the probability distribution of the other fingers $P(F_t^2|X_t)$ can be expressed by the product of the probabilities of all four fingers because they are all independent.

The state transition probability distribution $P(X_t|X_{t-1})$ is the pre-learned transition matrix between the posture classes. Every term we specified above can be trained via a training data set.

To model hand gestures, we apply a DBN to the upper-layer. Linking between BN for the lower-layer and DBN for the upper-layer is specified below. In the proposed DBN described in the upper layer in Fig. 3, the hidden state M_t is represented in terms of a set of random variables, each of which denotes a hand gesture. There are two observations $G_{1:t}, X_{1:t}$ represented in terms of random variables, $G_{1:t}$ denotes the orientation of a hand movement, and $X_{1:t}$ denotes the probability of a hand posture, which is the output of the lower layer BN. For the observation G_t, the motion

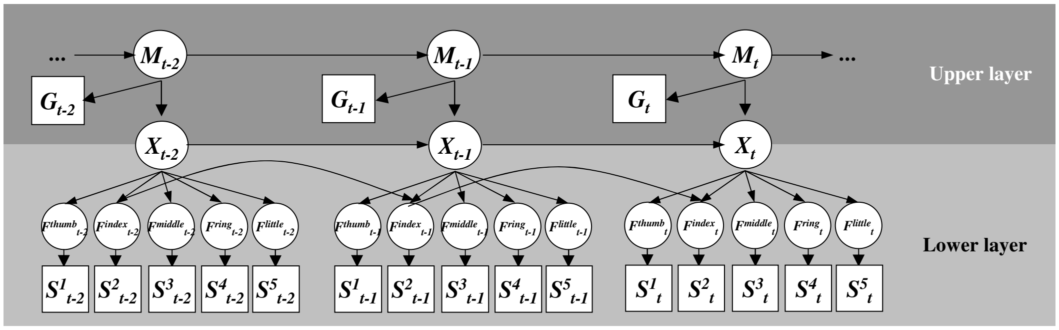

Figure 3. Graphical representation of the TBN

orientation histogram is applied, to represent the movement of hand region as an 8-direction vector. The motion orientation histogram can be extracted by accumulating the orientation of the hand movement from the image sequence. For the observation X_t, the output of the lower-layer is used as the row vector of hand posture probabilities. The lower-layer outputs the probability of each hand posture, and then, the vector for the observation can be generated by vectorizing all probabilities. Given these two observations, the marginal probability of the hidden state M_t must be calculated to recognize the hand gesture.

To compute the probability $P(M_t = j|X_{1:t}, G_{1:t})$, known as inference in DBN, we apply the forwards algorithm for the proposed DBN. Using the forwards algorithm, we can recursively calculate $P(M_t = j|X_{1:t}, G_{1:t})$ in the forward pass as follows:

$$P(M_t = j|X_{1:t}, G_{1:t}) = \frac{1}{c_t} P(M_t = j, X_t, G_t|X_{1:t-1}, G_{1:t-1}) \quad (6)$$

where

$$P(M_t = j, X_t, G_t|X_{1:t-1}, G_{1:t-1}) = P(X_t, G_t|M_t = j) \left[\sum_i P(M_t = j|M_{t-1} = i) P(M_{t-1} = i|X_{1:t-1}, G_{1:t-1})\right]$$

and

$$\begin{aligned} c_t &= P(X_t, G_t|X_{1:t-1}, G_{1:t-1}) \\ &= \sum_k P(M_t = k, X_t, G_t|X_{1:t-1}, G_{1:t-1}). \end{aligned}$$

In other words, the forward variable $P(M_t = j|X_{1:t}, G_{1:t})$ can be represented using the product of the state transition probability $P(M_t|M_{t-1})$, the previous forward variable $P(M_{t-1}|X_{1:t-1}, G_{1:t-1})$, and the conditional likelihood of the observations $P(X_t, G_t|M_t)$.

We define variable γ as follows:

$$\gamma_t(j) = \frac{P(M_t = j|X_{1:t}, G_{1:t})}{\sum_{j=1}^{N} P(M_t = j|X_{1:t}, G_{1:t})} \quad (7)$$

where N is the number of defined gestures and the dominator $\sum_{j=1}^{N} P(M_t = j|X_{1:t}, G_{1:t})$ is the normalization factor.

Using $\gamma_t(j)$, we can discriminate the recognized gesture $\hat{j}$ associated with the given observations $X_{1:t}, G_{1:t}$.

$$\hat{j} = \arg\max_{1 \le j \le N} [\gamma_t(j)], \qquad 1 \le t \le T. \quad (8)$$

Fig. 3 shows a graphical representation of the proposed TBN. As shown in the Fig. 3, the lower layer BN inputs visual features, and outputs hand posture recognition results as a vector representing the probabilities of each hand posture. In turn, the upper layer DBN inputs the vector which is the output of the lower layer and the orientation histogram of the hand movement, and outputs the hand gesture recognition result.

This hierarchical model decouples the dependency between features of postures and features of gestures. Therefore it has the advantage of reducing the computational complexity of probabilistic inference. In addition to this computational advantage, TBN can alleviate the inaccurate feature extraction problem by taking into account the temporal constraint. Usually, a cluttered background causes errors in detection of hands from an image, and these errors may successively affect the steps of feature extraction and recognition.

The problem of the training TBN involves a parameter estimation. The parameter $\hat{\Theta}$ is estimated by the EM (Expectation Maximization) algorithm, to maximize the likelihood of input sequences $L(\Theta, O)$, where Θ is the parameter vector of the model and O is the training sequence. In the TBN, because of the two-layered architecture, the parameters of the lower-layer and upper-layer can be trained separately. This separation of training between the lower and upper layers can reduce the computational complexity.

4. Experimental Results and Analysis

In order to evaluate the proposed method, two data sets were used. First one is a data set we built. We built a video data set because there is no proper database of hand gesture for the virtual mouse. The video data was captured using a USB web camera, at 30 frames per second, 320×240 in size and 24-bit colors. In the data set, there are two different background conditions: simple and cluttered backgrounds. The videos under the simple background condition were taken in where the background was surrounded by blue fabric and the videos under the cluttered background were taken in an office. For each background condition, video sequences of 15 times of 11 gestures were recorded for each subject.

Second one is a data set from S. Marcel [12]. The data set contained four hand gestures (Click, Rotate, Stop-Grasp-Ok, No). The size of images was 58×62 pixels and 8-bits colors, containing a cropped hand region in a uniform background. 15 sequences for each hand gesture are included in the data set.

Besides the data sets, we took additional 40 videos for training purpose. In the TBN, the probability density of each hand posture was trained using 80 images, and the probability density of each hand gesture was trained using 40 video sequences.

4.1. Definition of the Hand Postures and Gestures

A set of hand postures and gestures must be defined for the virtual mouse application. We defined one right-hand posture and gestures for the experiments. The hand postures are defined as specific hand shapes consisting of extending fingers. The list of defined hand postures are described in Fig. 4. Hand gestures are defined as continuous deformations of these hand postures. We defined commands for the hand gestures as shown in Fig. 5.

4.2. Experimental Results and Analysis

A series of experiments were performed to evaluate the proposed TBN and to compare it with the previous methods: Finite State Machine (FSM) [13] and Hidden Markov Model (HMM) [14].

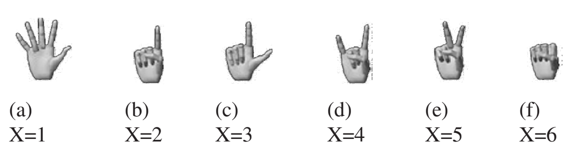

Figure 4. Descriptions of hand postures

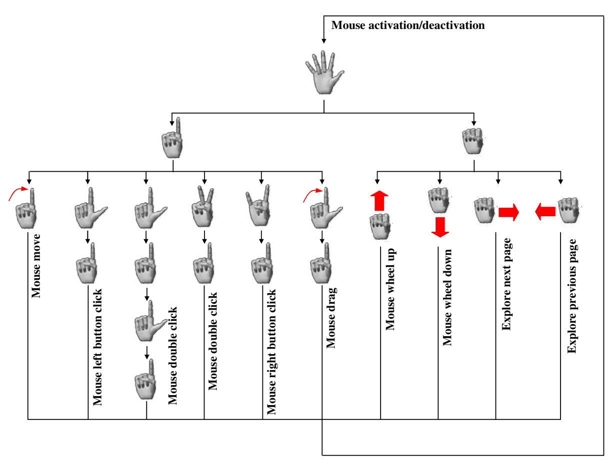

Figure 5. Description of hand gestures

Table 3. The comparison of recognition results with Marcel's dynamic hand posture data set[12]

Methods	N	C	Errors S	I	D	*Accuracy* (%)
FSM [13]	60	55	5	4	0	85.0
HMM [14]	60	57	3	3	0	90.0
The proposed method	60	58	2	1	0	95.0

The hand gesture recognition rates using TBN are presented in Table. 1. In this table, S (substitution), I (Insertion), D (Deletion) values indicate the cases where a gesture sequence is classified incorrectly, where a wrong gesture is detected in a correctly recognized gesture sequence, and where any gesture is not detected in an input sequence. The recognition accuracy is calculated as

$$\frac{N-S-I-D}{N} \times 100 = \frac{C-I}{N} \times 100 \tag{9}$$

where N is the number of test data and C is $N-(S+D)$, the number of correctly recognized gestures. The comparison of recognition results between previous methods and the proposed TBN is presented in Table. 2. Compared with the previous methods proposed by M. Yeasin and S. Chaudhuri [13] and A. Ramamoorthy, et al. [14], the proposed method is robust to cluttered backgrounds, while the likelihood of the true gesture is generally reduced in the case of a cluttered background. Because of the architecture of the two-layered framework, which recognizes hand postures and gestures hierarchically, our method can effectively recognize hand gestures, even though the posture layer may fail to recognize the exact hand posture with ambiguous features.

In addition to the experiments, we also have tested in a data set from S. Marcel [12]. The comparison of recognition results with this data set are shown in Table 3.

5. Conclusions

In this paper, we proposed a Virtual Mouse interface, a gesture-based mouse interface and, to utilize the Virtual Mouse interface, we also proposed Two-layered Bayesian Network (TBN) for robust hand gesture recognition in real-time. The proposed interface provides a convenient and intuitive way to perform mouse actions via single handed gestures. The TBN has the advantage of recognizing hand gestures in cluttered environments, which normally makes extracting reliable visual features difficult. Due to the two-layered architecture, we can decouple the dependency between features for postures and features for gestures. In the experiments, the Virtual Mouse Interface based on the TBN showed reliable recognition rates on both simple and cluttered environments.

Table 1. The recognition rates using the proposed TBN

Gestures	Simple background *N*	*C*	Errors *S*	*I*	*D*	*Accuracy* (%)	Cluttered background *N*	*C*	Errors *S*	*I*	*D*	*Accuracy* (%)
Mouse Activation	150	149	1	2	0	98.00	150	142	8	11	0	87.33
Mouse Move	150	147	3	7	0	93.33	150	145	5	15	0	86.67
Left Button Click	150	144	6	9	0	90.00	150	143	7	19	0	82.67
Double Click$_1$	150	138	12	4	0	89.33	150	136	14	13	0	82.00
Double Click$_2$	150	139	11	2	0	91.33	150	138	12	16	0	81.33
Right Button Click	150	146	4	2	0	96.00	150	137	13	8	0	86.00
Mouse drag	150	143	7	3	0	93.33	150	138	12	13	0	83.33
Mouse Wheel Up	150	145	5	1	0	96.00	150	142	7	15	1	84.67
Mouse Wheel Down	150	144	6	1	0	95.33	150	140	10	9	0	87.33
Explore next page	150	147	3	7	0	93.33	150	144	6	12	0	88.00
Explore previous page	150	145	5	2	0	95.33	150	142	7	11	1	87.33
Total	1650	1587	63	40	0	93.76	1650	1547	101	142	2	85.15

Table 2. The comparison of recognition results between previous methods and the proposed TBN

Methods	Simple background *N*	*C*	Errors *S*	*I*	*D*	*Accuracy* (%)	Cluttered background *N*	*C*	Errors *S*	*I*	*D*	*Accuracy* (%)
FSM [13]	1650	1568	77	156	5	85.58	1650	1465	185	179	0	77.94
HMM [14]	1650	1590	60	77	0	91.70	1650	1503	136	148	11	82.12
The proposed method	1650	1587	63	40	0	93.76	1650	1547	101	142	2	85.15

Acknowledgments

This work was supported by the Korea Science and Engineering Foundation (KOSEF) grant funded by the Korea government (MEST) (No. 2009-0060113).

References

[1] S. Mitra, T. Acharya, Gesture recognition: A survey, IEEE Trans. on Systems, Man, and Cybernetics 37 (3) (2007) 311–324.

[2] L. Bretzner, I. Laptev, T. Lindeberg, S. Lenman, Y. Sundblad, A prototype system for computer vision based human computer interaction, Tech. Rep. CVAP251, Department of Numerical Analysis and Computer Science, KTH (Royal Institute of Technology), Stockholm, Sweden (2001).

[3] A. Argyros, M. Lourakis, Vision-based interpretation of hand gestures for remote control of a computer mouse, in: Proc. European Conference on Computer Vision, Vol. 3979, Graz, Austria, 2006, pp. 40–51.

[4] Y. Fu, T. S. Huang, hMouse: Head tracking driven virtual computer mouse, in: Proc. of the 8th IEEE Workshop on Applications of Computer Vision, Austin, Texas, USA, 2007, pp. 30–36.

[5] P. Robertson, R. Laddaga, M. Kleek, Virtual mouse vision based interface, in: Proc. of the 9th International Conference on Intelligent User Interfaces, Funchal, Madeira, Portugal, 2004, pp. 177–193.

[6] J. Brand, J. Mason, A comparative assessment of three approaches to pixellevel human skin-detection, in: Proc. 15th IEEE/IAPR International Conference on Pattern Recognition, Vol. 1, Barcelona, Spain, 2000, pp. 1056–1059.

[7] P. Viola, M. Jones, Robust real-time face detection, International Journal of Computer Vision 57 (2) (2004) 137–154.

[8] M. Isard, J. Maccormick, BraMBLe: a bayesian multiple-blob tracker, in: Proc. 8th IEEE International Conference on Computer Vision, Vol. 2, Vancouver, Canada, 2001, pp. 34–41.

[9] I. Laptev, T. Lindeberg, Tracking of multi-state hand models using particle filtering and a hierarchy of multi-scale image features, in: Proc. 3rd International Conference on Scale-Space and Morphology in Computer Vision, Vancouver, Canada, 2001, pp. 63–74.

[10] M. Vafadar, A. Behrad, Human hand gesture recognition using motion orientation histogram for interaction of handicapped persons with computer, in: Proc. of International Conference on Image and Signal Processing, Vol. 5099, Normandy, France, 2008, pp. 378–385.

[11] K. Murphy, Dynamic bayesian networks: Representation, inference and learning, Ph.D. thesis, University of British Columbia (2002).

[12] S. Marcel, http://www.idiap.ch/resources/gestures/.

[13] M. Yeasin, S. Chaudhuri, Visual understanding of dynamic hand gestures, Pattern Recognition 33 (11) (2000) 1805–1817.

[14] A. Ramamoorthy, N. Vaswani, S. Chaudhury, S. Banerjee, Recognition of dynamic hand gestures, Pattern Recognition 36 (9) (2003) 2069–2081.