Pattern Recognition 43 (2010) 3059–3072

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/pr

Hand gesture recognition based on dynamic Bayesian network framework[☆]

Heung-Il Suk [a], Bong-Kee Sin [b,*], Seong-Whan Lee [a]

[a] Department of Computer Science and Engineering, Korea University, Anam-dong, Seongbuk-ku, Seoul 136-713, Republic of Korea
[b] Department of Computer Engineering, Pukyong National University, Daeyon-dong 599-1, Nam-ku, Busan 608-737, Republic of Korea

ARTICLE INFO

Article history:
Received 21 June 2009
Received in revised form
9 February 2010
Accepted 22 March 2010

Keywords:
Hand gestures recognition
Dynamic Bayesian network
Coupled hidden Markov model
Continuous gesture spotting

ABSTRACT

In this paper, we propose a new method for recognizing hand gestures in a continuous video stream using a *dynamic Bayesian network* or DBN model. The proposed method of DBN-based inference is preceded by steps of skin extraction and modelling, and motion tracking. Then we develop a gesture model for one- or two-hand gestures. They are used to define a cyclic gesture network for modeling continuous gesture stream. We have also developed a DP-based real-time decoding algorithm for continuous gesture recognition. In our experiments with 10 isolated gestures, we obtained a recognition rate upwards of 99.59% with cross validation. In the case of recognizing continuous stream of gestures, it recorded 84% with the precision of 80.77% for the spotted gestures. The proposed DBN-based hand gesture model and the design of a gesture network model are believed to have a strong potential for successful applications to other related problems such as sign language recognition although it is a bit more complicated requiring analysis of hand shapes.

1. Introduction

Since Johansson's work on human motion perception and analysis [1], many researchers in computer vision have tried to analyze and understand human motion in video. Aggarwal and Cai reviewed literatures related to human motion analysis. In the paper, they divided human motion analysis into three areas, i.e. body structure analysis, tracking, and recognition, and addressed the relationships among these areas [2]. In this paper, we focus on the recognition of human hand motions occurring in a video sequence. Pavlovic et al. [3] surveyed problems and issues in visual hands gestures. To date a large body of literatures focuses on isolated hand gesture recognition [4–10], whereas only a small number of works dealt with detecting and recognizing hand gestures from video frames [11–14].

A hand gesture can be described by a locus of hand motion recorded in a sequence of signal frames. To model these signals *hidden Markov models* (HMMs) have been widely accepted as the choice of the models with applications to video analysis problems, such as recognizing tennis motions [15], identifying humans by their gaits [16,17], browsing PowerPoint™ slides using hand commands [11] and so on. Brand et al. suggested a coupled HMM that combines two HMMs with causal, possibly asymmetric link to recognize three T'ai Chi gestures [4].

Recently, there has been an increasing interest in a more general class of probabilistic models, called the *dynamic Bayesian network* (DBN), which includes HMMs and Kalman filters as special cases. DBN is a generalized version of the Bayesian network (BN) with an extension to temporal dimension. Du et al. defined five classes of interactions that could happen between two persons and developed a DBN-based model which took local features such as contour, moment, height and global features such as velocity, orientation, distance as observations [18]. Park et al. employed a DBN to analyze the change of the poses of body parts and recognized the interaction between two persons [19]. Avilés-Arriaga et al. extracted the region and the center of a hand as input features and used a näive DBN to recognize 10 one-hand gestures [20].

Early on, Pavlovic proposed the use of DBN for gesture recognition that can be seen as a combination of an HMM and a dynamic linear system [5]. Wilson also presented modeling techniques to adapt gesture models using the DBN scheme [6]. Yang et al. used time-delayed neural network to analyze feature vectors from hand trajectories [7]. With 40 different isolated signs they achieved a recognition rate up to 96.21%. Nefina et al. compared several different methods of audio-visual speech recognition and suggested the use of coupled HMMs and factorial HMMs by showing that coupled HMMs outperformed all the other models in the performance of recognition [21]. The coupled HMM will be compared with the proposed model in our experiments.

[☆] A preliminary partial version of the paper has been presented in the 19th IAPR International Conference on Pattern Recognition, Tampa, USA, December 2008, and the 8th IEEE International Conference on Automatic Face and Gesture Recognition, Amsterdam, The Netherlands, September 2008.

[*] Corresponding author. Tel.: +82 51 629 6256; fax: +82 51 629 6210.
E-mail addresses: hisuk@image.korea.ac.kr (H.-I. Suk), bkshin@pknu.ac.kr (B.-K. Sin), swlee@image.korea.ac.kr (S.-W. Lee).

0031-3203/$ - see front matter © 2010 Elsevier Ltd. All rights reserved.
doi:10.1016/j.patcog.2010.03.016

These previous works considered recognizing isolated gestures rather than spotting gestures in a continuous stream of motion. León et al., on the other hand, used a sliding window of 10 frames to represent the local trajectory with 10 observation nodes in a BN [13]. They showed that, even though some of the observations were missing, the method could still distinguish similar gestures such as "Good-bye" from "Move-Right." Shi et al. considered to segment and recognize human activities from a continuous action stream and presented a semi-Markov model [22]. Voglar and Metaxas proposed a framework for recognition of an American sign language based on an HMM. In their experiment, they extracted the signer's arm and hand motion information using three video cameras and an electromagnetic tracking system. The method achieved a recognition rate of 94.5% in isolated single signs and 84.5% in whole sentences [12]. Recently, Yang et al. proposed a threshold model which extends conventional conditional random fields model to dealing with the task of spotting and recognizing American signs in a set of vocabulary [23].

In this paper, we propose a dynamic Bayesian network model for hand gesture recognition that can be used to control media players or slide presentation. Unlike previous systems the proposed model accepts both one and two hand gestures. Given a video sequence, it captures the hand motion trajectories and relations to the face. They are converted to time series signals, and analyzed by gesture models. In experiments with 10 isolated gestures, the proposed model achieved a recognition rate of 99.59% with cross validation. In addition, a more practical problem of continuous gesture recognition is addressed based on a cyclic spotting network connecting gesture DBNs. To simultaneously recognize gestures and detect the start and end points of embedded gestures in a sequence of motion signals we developed a Viterbi-like dynamic programming method. A test on long videos showed 84% in recall and 80.77% in precision.

In the rest of the paper, we will define 10 hand gestures and describe the methods of detecting and tracking hands, and describe features in Section 2. The proposed hand gesture recognition model and the inference and learning algorithm are explained in Section 3. Section 4 presents a circular network model for continuous gesture motion spotting and recognition for practical applications. The experimental results are presented and analyzed in Section 5. Finally Section 6 concludes the paper.

A preliminary partial version of this paper appeared in [24,25] with limited scope of isolated gesture classification. The main contribution of the current work compared to the previously published conference papers is that it analyzes the results of isolated gesture recognition by decoding the hidden states in DBNs. We then further extend the DBN-based hand gesture model to deal with continuous gestures stream by designing a gesture network model and developing a Viterbi-like dynamic programming method for more practical applications. The proposed gesture network model can detect the start and end points of the embedded meaningful gestures in a video stream. We also demonstrate many experimental results both on isolated and continuous gestures recognition.

2. Hands tracking and feature extraction

Successful dynamic hand gesture recognition requires accurate location of hands and face in space-time. The result of this step influences greatly on the performance of the target system.

2.1. Hand gesture classes

For potential applications to controlling media players or slide presentation, we define 10 different hands gestures including five two-hand gestures and five one-hand gestures as shown in Fig. 1. Each black dot in the figure represents the starting position of the hand and each directed curve the motion trajectory of the hands. In the case of one hand gestures the remaining hand not participating in the gesture may or may not appear in video frames.

2.2. Tracking

Hand detection and tracking in video, though simple, is by no means an easy task due to noise, uncertainty, and the variation in illumination conditions [26].

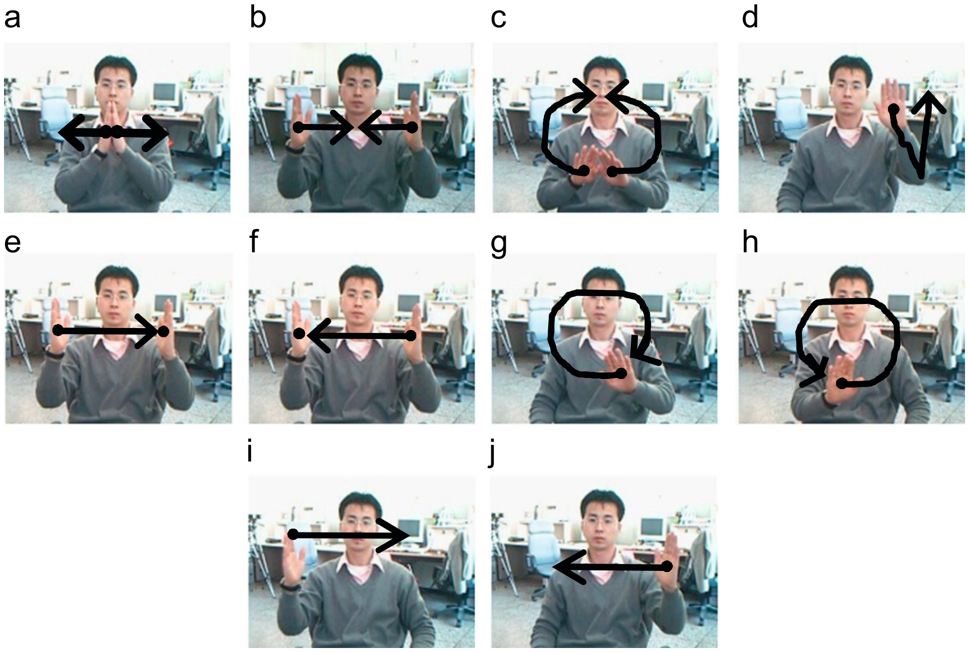

Fig. 1. Ten hands gestures: (a) open a file (OP), (b) close a file (CL), (c) play (PL), (d) pause (PA), (e) move to the last frame (ML), (f) move to the first frame (MF), (g) 10 seconds forward (TF), (h) 10 seconds backward (TB), (i) fast forward (FF), and (j) fast rewind (FR).

In this work, we employ two skin models for hands and face detection. One is the simple color range model widely used in the YIQ color space. It just limits the range of pixel values in that space. The other is the HSV color histogram built from the pixels in the face region obtained by the Haar-like face detector [27]. The latter model can reflect the current lighting conditions and the characteristics of the subject's skin color. An input pixel is regarded as a skin pixel if its likelihood of skin color in the histogram is greater than a given threshold [28].

Despite conceptual simplicity, tracking hands in video is not simple because each hand may overlap with a face or other hands. We tackle this problem by adapting the method Argyros et al. [29]. Here each blob of a face and hands is described by a Gaussian template where the mean represents the location of the face or hands (Fig. 2). Then by a simple linear extrapolation from the motion vector we can predict each blob in the next frame. When an overlap occurs among blobs, it can be solved by the following rules:

- Rule 1: If a skin pixel of a blob is located within 95% confidence interval of a Gaussian distribution then this pixel is considered to be supporting the Gaussian.
- Rule 2: If a skin pixel does not support any of the Gaussians, or it is outside the 95% confidence interval of all the Gaussian distributions, then it is assigned to one of the Gaussians that is closest to it.

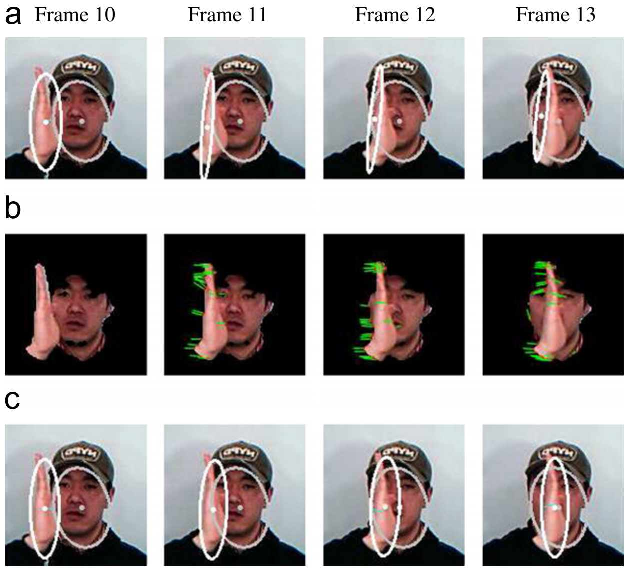

Fig. 2. A hand tracking example by two different methods: (a) Argyros et al.'s method based on prediction using the velocity of the previous two frames, (b) optical flows between the previous frame and the current frame, and (c) the proposed method based on the prediction using the average of the optical flow vectors.

Here, the confidence of a pixel for a Gaussian can be replaced by the Mahalanobis distance from the mean μ_k of Gaussian k:

$$D(\mathbf{x}) = (\mathbf{x}-\mu_k)^T \boldsymbol{\Sigma}_k^{-1} (\mathbf{x}-\mu_k) \tag{1}$$

where $\mathbf{x}$ is the coordinate of a skin-pixel and $\boldsymbol{\Sigma}_k$ is the covariance of the Gaussian representing the k-th blob.

Unlike Argyros' use of velocity for linear prediction, we propose the use of optical flow [30]. One problem of using velocity is that it often fails to track hands when the velocity changes abruptly. Whereas the optical flow measures the motion explicitly across frames and can still succeed in tracking regardless of past history.

The actual prediction is made by $\mu_k' = \mu_k + \mathbf{v}$, where $\mathbf{v}$ is mean of the optical field vectors given by

$$\mathbf{v} = \frac{1}{N} \sum_{i=1}^{N} \mathbf{f}(i)$$

and $\mathbf{f}(i) = [f_x(i), f_y(i)]^T$ denotes the i-th flow vector and N the number of flow vectors associated with the k-th blob.

For comparison Fig. 2 shows an example of hand tracking using velocity and optical flow. Fig. 2(b) shows optical flows between the consecutive frames which reflect the motion of the hand across the frames. Initially, the hand was stationary and the velocity-based method predicted for the hand to stay at the same position with no motion. Therefore, in Fig. 2(a), the velocity-based hand tracker without an explicit motion filtering consistently fails to track the accurate position of the hand. The error accumulates and finally in the frame 13 the Gaussian corresponding to the hand is out of place significantly. But the optical flow-based method tracks the hand correctly by measuring the motion explicitly as exemplified by Fig. 2(c).

2.3. Feature extraction

Among the variety of possible features, the most important information about a gesture will be the motion of hands. The motion can be described by the trajectory of a hand in space over time which in turn is represented by a sequence of positions of the hand $\mathbf{x}_t$, $t=1,\ldots$. The location $\mathbf{x}_t$ of the hand at time t has been estimated by the mean of the Gaussian fitting the corresponding blob as shown in Fig. 2. Each pair of successive hand locations defines a local motion vector. Then we can represent the whole motion trajectory by a sequence of motion vectors each of which is in turn encoded by a direction code using the scheme as shown in

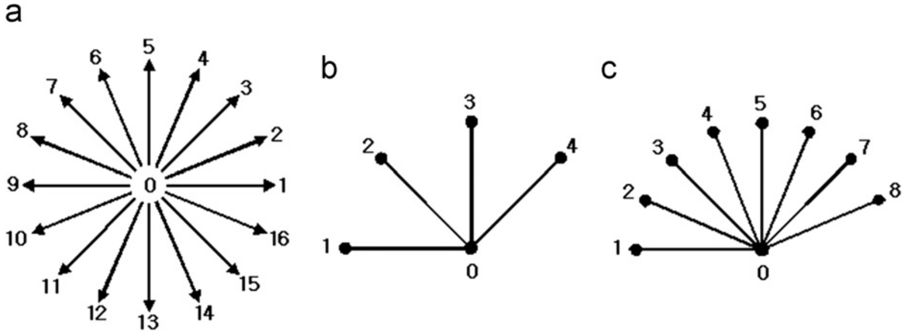

Fig. 3. Features: (a) 17 direction codes for hand motions, (b) hand–hand positional relation, and (c) face–hand positional relation.

Fig. 3(a). The central code ‘0’ denotes ‘no motion’ occurred. Given a video, we extract two chain codes one for each hand.

With the separate chain code for each hand, ambiguities can arise between gestures. For example, when there are two hands in a frame and the user is making one-hand gestures such as FF in Fig. 4(a) and FR in Fig. 4(b), the resulting chain codes are indistinguishable from those of the two-hand gestures ML in Fig. 4(c) and MF in Fig. 4(d), respectively. It is because each of the two pairs have the same chain code sequences as shown in Figs. 4(e) and (f). To avoid the ambiguities incurred by representing the motion using only the chain code, we introduce two more features: the relative position of the two hands (Fig. 3(b)) and the position of the each hand relative to the face (Fig. 3(c)). The code ‘0’ implies that two hands or a hand and a face are overlapping. The hand–hand relation and hand–face relation can also compensate for the effects of the small changes caused by user’s unconscious movement or by any unstable results from the image processing. The errors so made could affect the output code and ultimately the system performance significantly without any complementary measures.

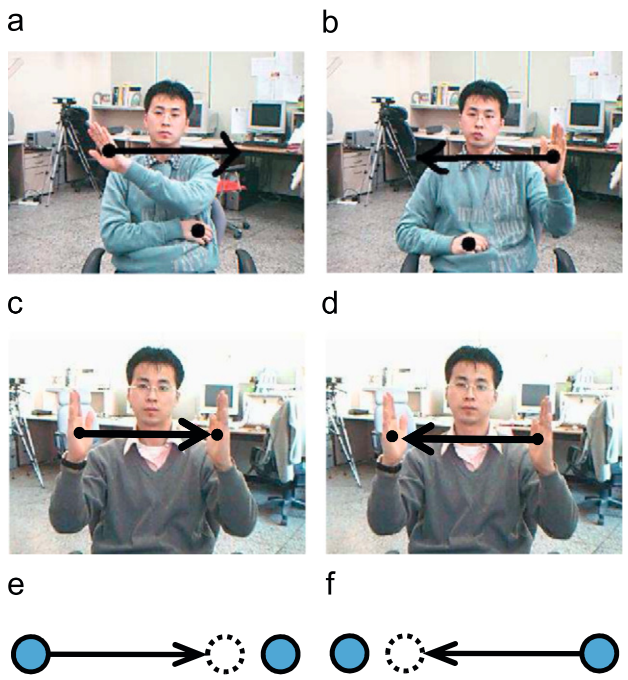

Fig. 4. Ambiguities may occur when only chain codes are used: (a) FF gesture, (b) FR gesture, (c) ML gesture, (d) MF gesture, (e) the chain-coded trajectory of (a) and (c), and (f) the chain-coded trajectory of (b) and (d).

3. Gesture modeling

3.1. Dynamic Bayesian network: DBN

Although an HMM [31] is a very useful tool for modeling variabilities in time series signals, its power is limited to a simple state space with a discrete hidden variable at a time. Suppose that there are more than two hidden variables, each representing an independent component, but those processes interact and are correlated. If we represent the system with a single hidden variable using the standard HMM of Fig. 5(a), then the state space of the hidden variable will be as big as the product of the size of all those hidden variables. Then as the number of the variables increases we will need an exponential amount of data for reliable estimation of model parameters.

The coupled HMM is an HMM variant tailored to represent the interaction of two independent processes [4]. It is essentially two HMMs coupled between the state variables across the two HMMs as shown in Fig. 5(b). In the figure, the gray square nodes denote hidden state variables and the white circle nodes observation variables. Although useful for modeling simple interacting processes, this model does not have room for common hidden variables which are believed to be shared between two variables.

The dynamic Bayesian network (DBN) [32] is a generalized framework of HMM/CHMM and Bayesian network (BN) [33]. With an appropriate design, it can make up for the weaknesses of the HMMs by factorizing the hidden variables into a set of random sub-variables. It is a model for computing effectively the joint probabilities of a set of random variables. The inference algorithms developed for Bayesian networks (BNs) can be applied to DBNs directly by taking advantage of the temporal progression.

3.2. Proposed model architecture

The 10 hand gestures defined in Section 2.1 include bimanual gestures as well as monomanual gestures. We are proposing a new design of DBN which has three hidden variables and five observable variables. The two hidden variables X^1 and X^2 model the motion of the left and the right hand, respectively, and each is associated with two observations of the features of the corresponding hand’s motion and the position relative to the face. The third hidden variable X^3 has been introduced to resolve the ambiguity between similar gestures. It models the spatial relation between hands.

Suppose that the relative position of two hands has changed from Figs. 6(a) to (b). In this case, when the left hand is lowered, we can infer that the right hand has been either raised or stationary as shown in Fig. 6(c). Similarly when the right hand is raised, the left hand has been either lowered or stationary as

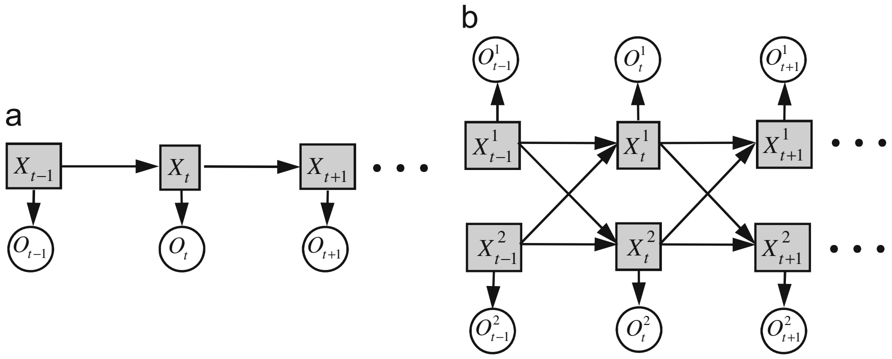

Fig. 5. Graphical representation of hidden Markov models: (a) standard hidden Markov model, and (b) coupled hidden Markov model.

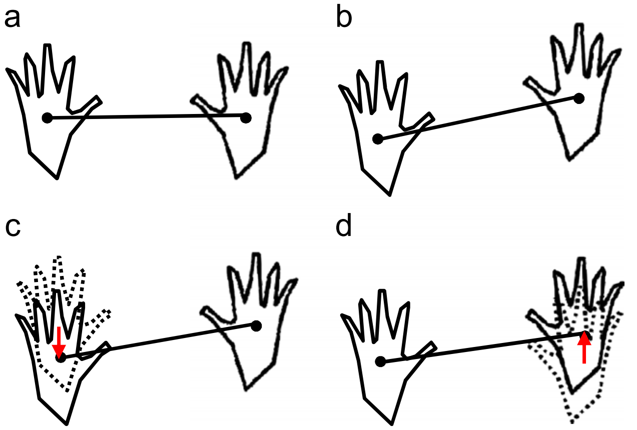

Fig. 6. Changes of the relative position of two hands: (a) the initial position of the hands, (b) after a motion, (c) the left hand lowered, and (d) the right hand raised.

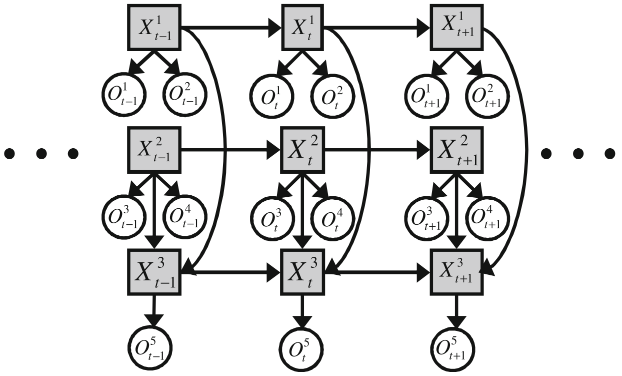

Fig. 7. The proposed dynamic Bayesian network model for hands gestures.

shown in Fig. 6(d). This means that given the value of the node X^3, i.e. the current relative position between two hands, the two nodes X^1 and X^2 are conditionally dependent, i.e.

$$X^1 \not\perp X^2 | X^3. \tag{2}$$

Using the first-order Markov assumption to simplify the motion dynamics we propose a new hands gesture model as shown in Fig. 7, with hidden variables in gray square nodes and observable variables in white circle nodes. In the figure, O^1 and O^3 denote the chain code of left and right hand motions, O^2 and O^4 the spatial relation between each hand and the face, and O^5 the spatial relation between two hands. In this model, the time dependency of the hidden variable X^3, that is, $\cdots \rightarrow X_{t-1}^3 \rightarrow X_t^3 \rightarrow X_{t+1}^3 \rightarrow \cdots$, is considered to implicitly capture the correlation between two hidden variables X^1 and X^2 as was done by the pair (X^1, X^2) in the coupled HMM. This also simplifies the proposed model and relieves it of the complication of the coupled HMM.

3.3. Inference and learning

One of the advantages of using BNs is that the graphical structure representing the conditional independence among variables allows us to compute the joint probability of a subset of variables very efficiently. The inference over a DBN is to compute the marginal probability $P(X_i|O_{1:\tau})$ of hidden variables X_i given an observation sequence $O_{1:\tau} = O_1 O_2 \dots O_\tau$.

The joint probability of variables in a BN can be factored into a product of local conditional probabilities one for each variable through conditional independencies or d-separation [33]. The full joint probability for the DBN in Fig. 7 can be computed by

$$\begin{aligned} P(X_{1:T}^{1:3},O_{1:T}^{1:5}) &= P(O_{1:T}^{1:5}|X_{1:T}^{1:3})P(X_{1:T}^{1:3}) = P(X_1^1)P(X_1^2)P(X_1^3|X_1^1,X_1^2) \\ &\quad \times \prod_{t=2}^{T} P(X_t^1|X_{t-1}^1)P(X_t^2|X_{t-1}^2)P(X_t^3|X_{t-1}^3,X_t^1,X_t^2) \\ &\quad \times \prod_{t=1}^{T} P(O_t^1,O_t^2|X_t^1)P(O_t^3,O_t^4|X_t^2)P(O_t^5|X_t^3) \end{aligned}$$

where

$$X_{1:T}^{1:3} = \begin{bmatrix} X_1^1 \\ X_1^2 \\ X_1^3 \end{bmatrix} \cdots \begin{bmatrix} X_T^1 \\ X_T^2 \\ X_T^3 \end{bmatrix} \quad \text{and} \quad O_{1:T}^{1:5} = \begin{bmatrix} O_1^1 \\ \vdots \\ O_1^5 \end{bmatrix} \cdots \begin{bmatrix} O_T^1 \\ \vdots \\ O_T^5 \end{bmatrix}$$

Learning in a DBN is the task of finding the optimal parameters $\hat{\Theta}$ that computes the maximum likelihood over the training data, i.e. $\hat{\Theta} = \operatorname{argmax}_{\Theta} P(O_{1:T}^{1:5}|\Theta)$. Here, the set of parameters includes initial state probabilities (π), state transitions probabilities (A), and output probabilities (B) just like those of HMM. Unlike the HMM, however, the DBN has many hidden variables for the distributed representation.

The proposed DBN includes three hidden variables and thus can be trained by exploiting the EM algorithm [34]. In order to determine the parameter values what we need to know are only the sufficient statistics for the variables. We can obtain them by means of the interface algorithm [32]. The update formulae for the parameters and their meaning in the perspective of maximum likelihood method are given in Appendix A.

4. Continuous gesture recognition

4.1. Design of gesture network

Isolated gesture recognition, although simple and easy to analyze, falls short of application to practical situations as it requires the knowledge of the start and the end of gesture motion. In its most general setting, a human motion is viewed as a sequence of mostly non-gestures which carry no useful information and occasional gestures of interest. A non-gesture is any meaningless pattern that fills the gap between meaningful gestures. Following the convention in other fields, a non-gesture will now be called a filler or a garbage [11,16,23].

Given the definition of a filler, the whole continuous motion can now be described as an alternating sequence of fillers and gestures. And it can be modeled by a straightforward translation into an alternating sequence of a filler model and gesture models:

< Hand Motion > := < Filler > · (< Gesture > · < Filler >)$^+$
< Filler > := Filler
< Gesture > := Open|Close|Play|Pause|
Move to the First Frame|Move to the Last Frame|
10 Seconds Forward|10 Seconds Backward|
Fast Forward|Fast Rewind

where the < Filler > and < Gesture > denote a filler and a gesture DBN, respectively, and '·' and '$^+$' denote concatenation and repetition, respectively.

One effective realization of the above whole motion model is a cyclic network of gesture DBNs. It is similar to that of recognizers for concatenated digits or phonemes [35]. In our case, it is a concatenation of gesture DBNs in parallel to one or more fillers, and then from the filler(s) back to the gesture models as shown in Fig. 8. The introduction of dummy nodes in Fig. 8, the start node '*S*' and the final node '*F*' in traversing the network, makes modeling and inference conceptually simple. The link from a dummy node

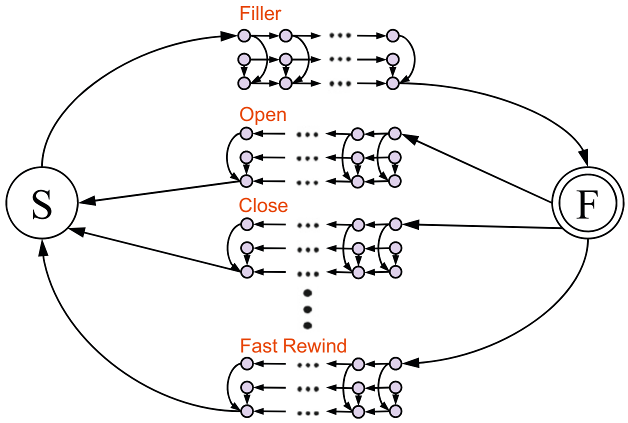

Fig. 8. Cyclic network of gesture DBNs for concatenated gestures.

to the hidden node X^1 of each DBN enables the propagation of the likelihood from the dummy node to the following filler or gesture DBNs. It does not matter which of the three hidden nodes of a DBN has a link from the dummy node, but only one of them must have it. Otherwise, the likelihood from a dummy node will be considered more than once in the computation of network search. A full explanation will be given in the forthcoming subsection.

4.2. Algorithm of continuous gesture recognition

Gesture recognition in general involves the simultaneous problems of finding the boundaries (the start and the end points) of target gestures and determining their class label. To tackle this problem we exploit the method of dynamic programming (DP) search [36]. It tries to find the best alignment between a given input sequence and the complete state sequence in which each desired gesture model gives the greatest likelihood for the corresponding segmental pattern in the motion sequence. Then the answer lies in the "best" model/state sequence that jointly maximizes the likelihood along with the segmental sequence.

Let $G=G_1G_2\dots G_K$, $K\geq 1$ be a sequence of gesture and filler DBNs for a given input frame sequence $O=O_1O_2\dots O_T$. Note that K is not known a priori. The goal is to find the best alignment to the best path $\hat{G}$ that maximizes the likelihood from the network by exploiting the method of DP search. Let one possible segmentation of O aligning to G be

$$S=S_1S_2\dots S_K=(s(1,t_1),\, s(t_1+1,t_2),\,\dots,\, s(t_{K-1}+1,t_K))$$

where $s(t_{k-1}+1,t_k)=O_{t_{k-1}+1}\dots O_{t_k}$ denotes a segment aligned to G_k and $1=t_0<t_1<\cdots<t_K=T$. Then the problem can be formulated as computing

$$P(O|Network)\triangleq\max_{K,G}P(O,G|Network)=\max_{K,G,S}P(S,G|Network) \tag{3}$$

where

$$G=G_1G_2\dots G_K,\ K\geq 1 \text{ and}$$

$$G_k=\begin{cases}\text{a gesture model} & \text{if } k=\text{even}\\ \text{a filler model} & \text{otherwise}\end{cases}$$

This is a joint optimization task of computing the maximum likelihood for the given observations by simultaneously determining the best number of gestures K, the best sequence of gestures G, and the best alignment of O to G.

Now let us assume all the paths in the spotting network are equally probable or we are using a uniform model. Then we can write Eq. (3) as a simple product of the likelihood of individual models in G:

$$P(S,G|Network)=P(S|G)P(G|Network)\triangleq\prod_{k=1}^{K}P(S_k|G_k) \tag{4}$$

In this equation, we ignore the probability of gesture transitions, i.e. the transition probabilities among hand gestures are uniform over all the gestures. If we further apply the concept of Viterbi path alignment $(S_k,\ Q_k)$ inside a model G_k, where $Q_k=q(t_{k-1}+1,t_k)=q_{t_{k-1}+1}\dots q_{t_k}$ is a legal state sequence within the model G_k. Then we can write Eq. (3) as follows:

$$P(O|Network)\triangleq\max_{K,G,S}P(S,G|Network)\triangleq\max_{K,G,S}\left[\prod_{k=1}^{K}\max_{Q_k}P(S_k,Q_k|G_k)\right] \tag{5}$$

Let us denote the current network node in path as $gr\in\{S,F\}$ and another node that immediately precedes it as gl. The pair (gl,gr) is connected via a set of parallel DBNs as shown in Fig. 8. You may regard it as a conceptual link with a label of DBNs, e.g., a link from node F to S. Let $L(gl,gr)$ be a set of models (DBNs) in the path from the node gl to gr. In addition, let us define the likelihood of initial partial sequences of length t:

$$\Delta_t(gr)=P(O_1\dots O_t,q_1\dots q_t,q_t\rightarrow gr|Network)$$
$$=\begin{pmatrix}\text{Accumulated joint likelihood of the partial sequences } O_1\dots O_t\\ \text{and the best } q_1\dots q_t \text{ reaching the node } gr \text{ at time } t\end{pmatrix}$$

Utilizing the idea of the dynamic programming principle, we can rewrite $\Delta_t(gr)$ as a recurrence relation

$$\Delta_t(gr)=\max_{(gl,m)\ \text{s.t.}\ m\in L(gl,gr)}\Delta_{t'}(gl)\times P(s(t'+1,t),q(t'+1,t)|m),\quad \begin{matrix}t=1,\dots,T\\ gr\in\{S,F\}\end{matrix} \tag{6}$$

where $s(t'+1,t)=O_{t'+1}\dots O_t$ and $q(t'+1,t)=q_{t'+1}\dots q_t$. This is what we call the '*global DP*' that performs a maximization at the level of the network model. Note that $0\leq t'<t$ and t' is to be determined probabilistically within the local DP which computes the second factor of the right hand side in Eq. (6) as described below. This equation will be revisited after describing the local DP.

In Eq. (6), the likelihood was defined for the global dummy nodes. Similarly, we can also define a similar measure for the internal states $(i,\ j,\ k)$ of each DBN as

$$\delta_t^m(i,j,k)=\Delta_{t'}(gl)\times P(s(t'+1,t),q(t'+1,t-1),q_t=(i,j,k)|m)$$
$$=\left[\max_{(a,b,c)}\delta_{t-1}^m(a,b,c)\times A_{ai,bj,ck}^m\right]\cdot B_{i,j,k}^m(O_t)$$
$$=\begin{pmatrix}\text{Accumulated joint likelihood of the partial sequences } O_1\dots O_t\\ \text{and the best } q_1\dots q_t \text{ where } q_t=(i,j,k) \text{ in the DBN } m\end{pmatrix}$$

$$\text{for}\begin{cases}t=1,\dots,T\\ m=\text{any of the gesture and filler DBNs}\\ (i,j,k)=(X_t^1,X_t^2,X_t^3),\ \text{a state triple of state spaces}\end{cases} \tag{7}$$

where the triple $(i,\ j,\ k)$ represents a state described by the three hidden variables in the DBN m and $A_{ai,bj,ck}^m$ and $B_{i,j,k}^m(O_t)$ denote the state transition $(a,b,c)\rightarrow(i,j,k)$ probability and the probability of observing O_t at the state triple $(i,\ j,\ k)$ in the model m, respectively. This is the second recurrence relation called the '*local DP*'. It aims at finding the best state transitions to a state triple $(i,\ j,\ k)$ at time t for the partial sequences $O_1\dots O_t$.

In the task of spotting and recognition in a video sequence, the segmentation boundaries of gestures are not known a priori. Let us consider a likely segment $s(t,\ t+d)$ for any $d>0$. At time $t-1$, the likelihood of a model which represents the gesture being completed will be high. During the global DP in Eq. (6), this likelihood gets forwarded to some network node gl. Then the decision as to picking the starting boundary of a new segment

$s(t,\ t+d)$ in the following model m is made by

$$\delta_t^m(i,j,k) = \max\{\delta_{t-1}^m A_{11,11,11}^m, \varDelta_{t-1}(gl) \times 1\} \times B_{1,1,1}^m(o_t) \qquad (8)$$

where it is assumed that the DBN m has a left-to-right transition topology and always starts from the state triple (1, 1, 1). Here the two expressions in the brace correspond, in order, to the DBN itself and a new transition into the DBN from outside implying a new gesture (or a segment) begins at this point in time.

Now the description of the local DP is complete. Then the global DP as shown in Eq. (6) is straightforward. Since the local likelihood of the final state triple $E^m = (N^{m,1}, N^{m,2}, N^{m,3})$ of the DBN m is equal to the right hand side of Eq. (6) for the pair of m and gl, where $N^{m,1}$, $N^{m,2}$, and $N^{m,3}$ represent the size of states on which each hidden variable could take. Therefore the actual global DP is given by

$$\varDelta_t(gr) = \max_{(gl,m \in L(gl,gr))} \delta_t^m(E^m), \quad \begin{array}{l} t = 1,\ldots,T \\ gr \in \{S,F\} \end{array} \qquad (9)$$

A complete description of all the variables employed in the above description is given in Table 1. Refer to Appendix B for the complete algorithm with an appropriate initialization and backtracking for recovering the best estimate of the gesture sequence.

In the local DP, three variables are maintained: the maximum likelihood at each state triple $\delta_t^m(i,j,k)$, the source state triple $\psi_t^m(i,j,k)$ that contributes to the maximum likelihood, and the elapsed time $\varphi_t^m(i,j,k)$ within the model m. Whereas in global DP, two variables are required: the likelihood of the best path reaching the current network node $\varDelta_t(gr)$ and the gesture model that forwarded the maximum likelihood. The pair of the gesture DBN model's label and the left network node gl is stored in $\Psi_t(gr)$. After the forward pass, the algorithm traces back the current result of forward pass to recover the best sequence of models starting from the final node 'F' in Fig. 8. The backtracking is based on the information about node transitions in Ψ_t and the elapsed time of the DBN model m in φ_t^m that evaluated the maximum likelihood at time t.

If we analyze the computational complexity of the algorithm, the majority of the computation is carried out in the local DP of the innermost loop. Let the number of states for each variable X^1, X^2, and X^3 be n on average. Then the size of the state space becomes $N = n^3$. Considering M possible gesture models including a filler model, the time complexity for a whole sequence of observations of length T is MN^2T. If we employ the left-to-right topology, it becomes linear in N.

5. Experimental results and analysis

5.1. Data description

The proposed method of gesture recognition is about statistical models with numerous parameters. They must be trained from a set of examples before being put to use. For each of the 10 gestures, we captured seven videos from seven different subjects at different times, a total of 490 video sequences for training and testing the baseline models. Another eight longer video sequences which contain 50 gestures in total were prepared for an experiment in continuous gesture recognition. All the videos were captured using a small CMOS camera at 30 frames per second, 320 × 240 in size, and 24-bit colors. The system was developed in C++ and MATLAB using Intel OpenCV library [37] and BNT [38].

5.2. Baseline gesture models

Different gesture models describe different gesture patterns of different length. This fact leads us to assign different number of states to each hidden variable of different models. In fact, this difference is a weak representation of model duration [31]. The number of states, which each hidden node can take on, is proportional to the complexity of the corresponding gesture. We determined the numbers of states of all models by evaluating the performance while varying the numbers 2–7 for the hidden nodes X^1 and X^2 and 5–15 for the hidden node X^3 considering the complexity of gestures. The detailed specification is given in Table 2. Each process corresponding to the left-hand motion, the right-hand motion, and the two-hand relative position was modeled with a left-to-right transition topology. From our experiments, a small change to the number of hidden states made little difference in performance, only small portion of test samples were failed to be correctly classified.

5.3. Isolated gesture recognition

The first set of experiments includes testing isolated gesture recognition and comparing it with a related model. Given the limited data set, we carried out 7-fold cross validation in which we randomly selected 42 sequences from 49 video sequences for each gesture for training each of the gesture models, and the rest seven sequences for testing. We repeated this process seven times. The overall performance was measured by the average rates from the seven repeated tests.

Table 1
Description of the variables employed in the algorithm of continuous gesture recognition.

Local DP	m	$m \in$ {Filler,Open,Close, ...,Fast Rewind}: gesture or filler DBN models
	(i,j,k)	DBN m's state triple representing a model configuration, $i \in S^{m,1}, j \in S^{m,2}, k \in S^{m,3}$ where $S^{m,i} : \{1,2,\ldots,N^{m,i}\}, i \in \{1,2,3\}$
	$A_{(ai,bj,ck)}^m$	State transition from (a,b,c) to (i,j,k) in DBN m $P(X_t^1 = i \mid X_{t-1}^1 = a)P(X_t^2 = j \mid X_{t-1}^2 = b)P(X_t^3 = k \mid X_t^1 = i, X_t^2 = j, X_{t-1}^3 = c)$ where $ai \in S^{m,1} \times S^{m,1}, bj \in S^{m,2} \times S^{m,2}, ck \in S^{m,3} \times S^{m,3}$
	$B_{(i,j,k)}^m\ (O_t)$	Probability of observing O_t in state triple (i,j,k) in DBN m
	$\pi_{(i,j,k)}^m$	Initial state probability reaching state triple (i,j,k) in DBN m
	$\delta_t^m(i,j,k)$	Accumulated likelihood of the best path reaching state triple (i,j,k) in DBN m at time t
	$\psi_t^m(i,j,k)$	Source state triple that maximized $\delta_t^m(i,j,k)$
	$\varphi_t^m(i,j,k)$	Duration of the best path to state triple (i,j,k) since it entered DBN m
	E^m	Final state triple $N^{m,1}, N^{m,2}, N^{m,3}$ of DBN m
Global DP	$L(gl,gr)$	A set of DBN models in the path $gl \rightarrow gr$ where $gl \rightarrow gr \in \{g_S \rightarrow g_F, g_F \rightarrow g_S\}$
	$\varDelta_t(gr)$	Accumulated likelihood of the best path reaching a network node gr at time t
	$\Psi_t(gr)$	A pair (gl,m) that produced $\varDelta_t(gr)$ where m is a DBN attached to the arc $gl \rightarrow gr$

Recognition using the DBN is done by the following classification rule:

$$\hat{\lambda} = \underset{\lambda}{\operatorname{argmax}}\, P(O^{1:5}_{1:T} | \Theta_{\lambda})$$

This classifier assumes a uniform prior over the models and chooses the model $\hat{\lambda}$ of maximum likelihood given an input features $O^{1:5}_{1:T}$. The parameter vector Θ_{λ} includes initial probabilities, state transition matrices, and output observation probability distributions one for each node of the model λ. The likelihood for the given the observation sequence can be obtained by marginalizing out all the hidden variables through an interface algorithm [32] and a junction tree algorithm [39].

5.3.1. *Modeling with standard HMMs and coupled HMMs*

In the first experiment, we created standard HMMs and coupled HMMs for the 10 gestures to compare the performance with that of the proposed DBNs. They observe the same chain codes of each hand's trajectory. We assigned uniform values to the probability distributions of the hand which is not participating in one-hand gestures to ignore its unintentional motion. When tested on a selected set of video showing only one hand in one hand gestures and the other hand is out of scene, the standard HMMs and coupled HMMs recorded the hit ratio of up to 97.55% and 97.35%, respectively. As expected a large number of hidden states are required for the standard HMMs to recognize the 10 hand gestures. The highest recognition rate was obtained from HMMs with 18 states for hand gestures of '*Open*,' '*Close*,' '*Move to the First Frame*,' '*Move to the Last Frame*,' '*Fast Forward*,' '*Fast Rewind*' and more than 30 for hand gestures of '*Play*,' '*Pause*,' '*10 Seconds Forward*,' '*10 Seconds Backward*.' The detailed cross-validation results are shown in Fig. 9(a) and (b) where each bar represents the hit ratio for the corresponding subset of data. The low recognition rate for the subset 2 in Fig. 9(b) is attributed to the errors in the image processing that failed to detect a large part of the skin regions in some frames because of sudden changes in lighting conditions. Consequently, the means and the covariances of the Gaussians describing the blob shape and position were unstable leading to jerky gestures. This caused a great perturbation to the chain codes.

5.3.2. *Dynamic Bayesian network with additional information*

Just like the standard HMMs and coupled HMMs, we created 10 DBNs for the target gestures but with the addition of the relative position between two hands as required by the proposed model. This information, although not absolutely required as an input (it can be missing), is important for inferencing in the proposed DBN models. The DBN recorded the recognition rate of 98.98% (Fig. 9(c)).

Although the performance of HMMs and CHMMs is comparable to that of the proposed DBN in isolated hand gestures recognition the state space of DBN is much smaller than the other models. It is known that we can theoretically model any complex pattern with an HMM with an arbitrarily large number of states. An important issue here is how to effectively reduce the number of states down to a manageable level while retaining the modeling power. Here does the DBN excels the conventional HMMs.

Another advantage of the proposed model is that it can accept both one- and two-hand gestures whether two hands are in view or not. When tested on an additional set of six one-hand gesture sequences, we obtained the results detailed in Table 3. Both the standard HMMs and the coupled HMMs recognized only one of the six one-hand gestures while they worked well for the input sequences in which only one hand was in view for the one-hand gestures. On the other hand, the DBN could still recognize all of them correctly. The tragic failure of the standard and the coupled HMMs comes from the unintended motion information of the

Table 2
The number of states of the three hidden nodes for each gesture DBN.

Gestures	Hidden nodes		
	X^1	X^2	X^3
OP (*Open*)	3	3	6
CL (*Close*)	3	3	6
PL (*Play*)	4	4	8
PA (*Pause*)	5	5	10
MF (*Move to the First Frame*)	3	3	6
ML (*Move to the Last Frame*)	3	3	6
TF (*10 Seconds Forward*)	6	6	12
TB (*10 Seconds Backward*)	6	6	12
FF (*Fast Forward*)	3	3	6
FR (*Fast Rewind*)	3	3	6

Table 3
Performance comparison of the standard HMM, the coupled HMM and the DBN for the one-hand gesture sequences in which both hands were in view: ○ (hit), × (miss) where the misrecognition labels are given in parentheses. Hand motion which is not related to the one-hand gestures can hinder their discrimination from some of the two-hand gestures.

One-hand gesture data	Recognition results	
	Standard HMM and Coupled HMM	DBN
FF (*Fast Forward*)	× : ML (*Move to the Last Frame*)	○
PA (*Pause*)	○	○
TF (*10 Seconds Forward*)	× : PA (*Pause*)	○
TB (*10 Seconds Backward*)	× : PA (*Pause*)	○
FR (*Fast Rewind*)	× : MF (*Move to the First Frame*)	○
TF (*10 Seconds Forward*)	× : PA (*Pause*)	○
Hits	1	6

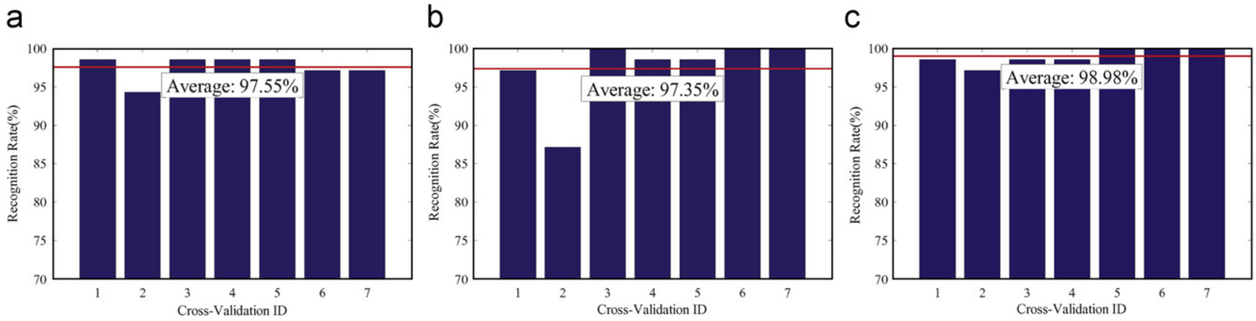

Fig. 9. Comparison of recognition rates among standard HMMs, coupled HMM, and DBN: (a) standard HMM with observations of chain codes, (b) coupled HMM having observations of chain codes, and (c) DBN having observations of chain codes and hand-hand relative positions.

Table 4
Hand gesture recognition results.

Gestures	No. of test gestures	No. of hits	No. of misses	Recognition rates (%)
OP (*Open*)	49	49	0	100
CL (*Close*)	49	49	0	100
PL (*Play*)	49	49	0	100
PA (*Pause*)	49	49	0	100
MF (*Move to the First Frame*)	49	48	1	97.59
ML (*Move to the Last Frame*)	49	48	1	97.59
TF (*10 Seconds Forward*)	49	49	0	100
TB (*10 Seconds Backward*)	49	49	0	100
FF (*Fast Forward*)	49	49	0	100
FR (*Fast Rewind*)	49	49	0	100
Sums and rates	490	488	2	99.59

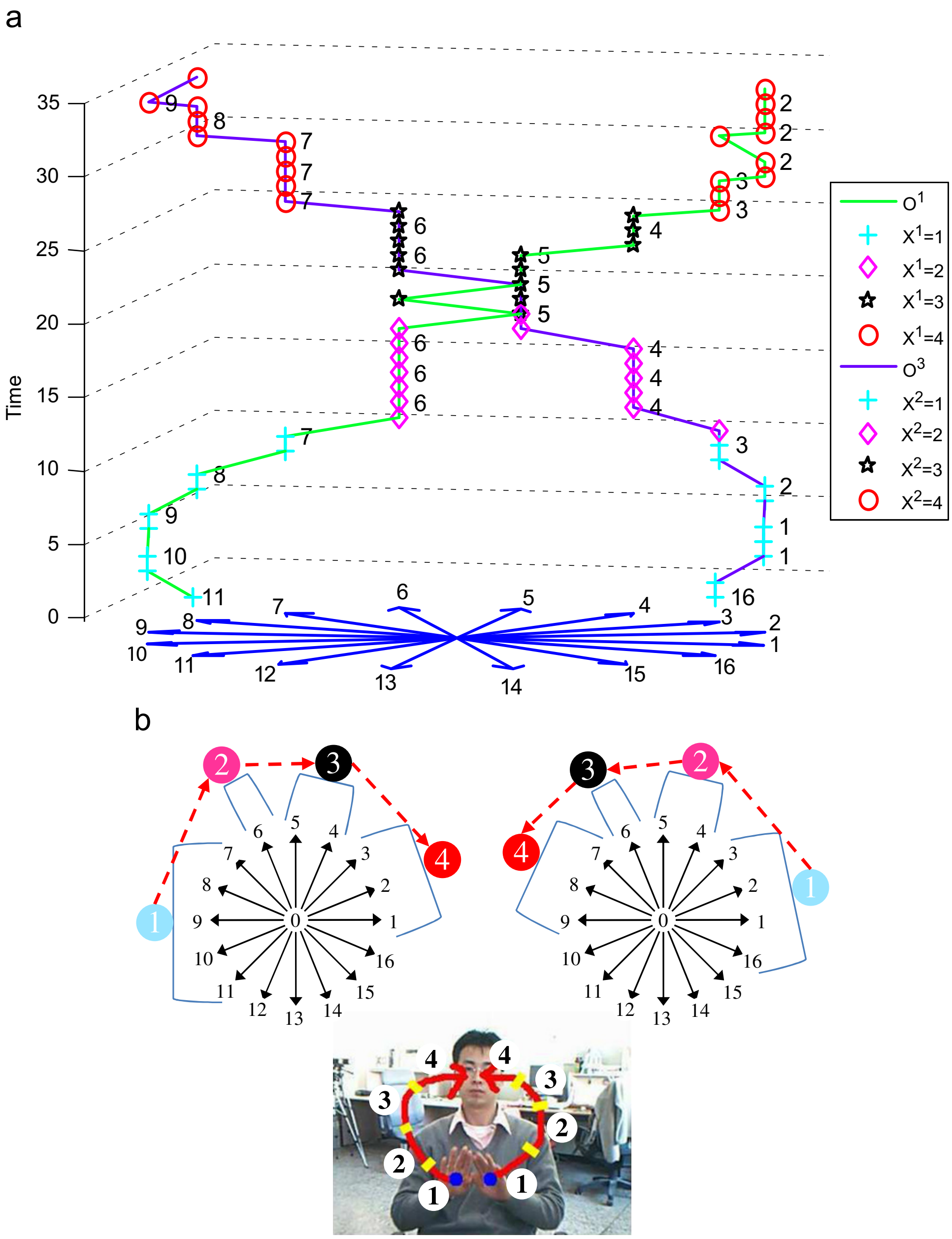

Fig. 10. Decoding result of a gesture sample of PL(*Play*). (a) Visualization of decoding the states of two hidden nodes X^1, X^2 given the observations O^1 and O^3 which represent each hand's motion. The numbers next to the markers denote directional codes of each hand at the time and the different markers the state of hidden nodes X^1, X^2 as shown in the bottom of the figure. (b) Motion segmentation and their state mapping for hidden nodes X^1, X^2. Refer to the online color version for clear view and understanding of this figure.

hand not part of the one-hand gestures. This tells us that the additional information helps disambiguate the gestures FF (*Fast Forward*), ML (*Move to the Last Frame*), FR (*Fast Rewind*), and MF (*Move to the First Frame*) which are ambiguous when only chain codes are used.

The next set of test results is about the overall performance in isolated gesture recognition (see Table 4). With all the input features considered, the proposed model recognized 99.59% of the input gestures (the rightmost column). Prior to this we first tested the model without the face-hands relation features (O^2, O^4). According to the analysis of the cases in which the DBN failed, e.g., CL (*Close*), ML (*Move to the Last Frame*), and MF (*Move to the First Frame*), most of the errors came from jerky motions of the hands or failures in detecting skin pixels. But with the inclusion of the relative position between a face and two hands, the effect of noise was reduced and the recognition rate reached up to 99.59% as presented in Table 4.

5.3.3. *Hidden states decoding*

Although the performance figures are quite high, they do not tell anything about the models' internal workings. Fortunately, however, we can at least estimate the hidden information probabilistically. The Viterbi algorithm is the very tool for the task when using HMMs. We decoded the best state sequence given an input to check whether the DBN characterizes the gestures to our intuition. Here the best state sequence is a complete state sequence for each hidden node from time 1 to T with the maximum likelihood of the given gesture model for the observation. This best state sequence can be obtained from the junction tree algorithm in which we consider only the maximum likelihood path from among the states at time $t-1$ leading to the current state at time t instead of summing up the likelihood of all the possible paths. Finding the best state sequence is the basic for the recognition of continuous gestures considered in Section 4.

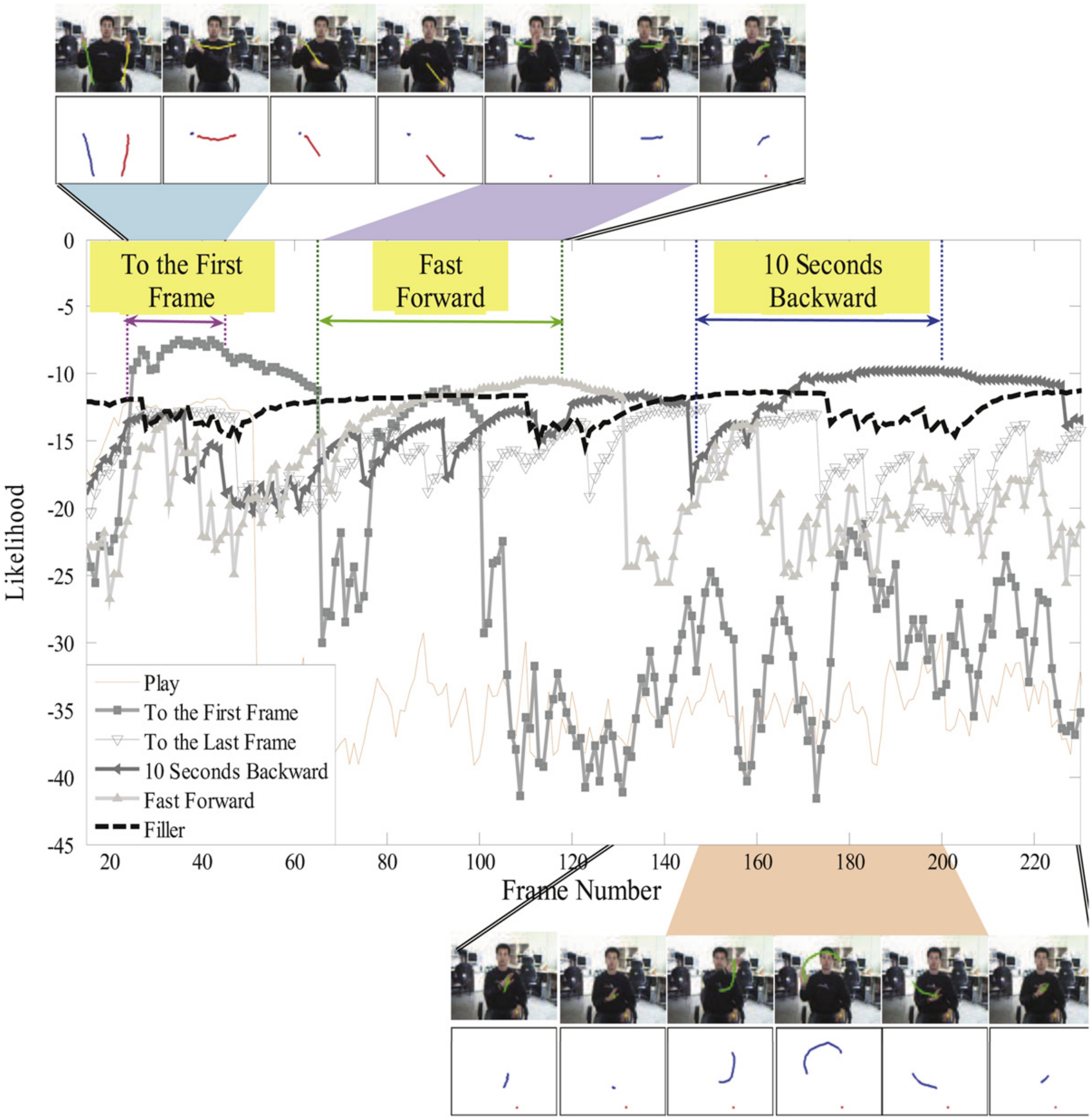

Fig. 11. Normalized per-symbol likelihood of gesture models with the corresponding partial observations for the video Seq. VIII. At the beginning of a gesture the output likelihood from the corresponding gesture model usually reports a poor match. However, around the end point of each gesture, it definitely beats all the rest even though it does not peak there. After that point the likelihood decreases slowly or drops abruptly due to following gestures. We here visualize the likelihoods of only six of ten gesture models to keep the graph less cluttered. Those small curve segments at the bottom represent the trajectory of two hands up to the frames.

The decoding result for the gesture sample of PL (*Play*) is given in Fig. 10. The values next to the markers in the figure are the chain codes (O^1, O^3) for two hands. The markers represent the sequence of the state labels of the corresponding hidden nodes (X^1, X^2) modeling the motion of each hand. Fig. 10(b) illustrates the segmentation of hand motion trajectories into four parts in sequence and the result of state labelling. Each state covers a small part of the whole motion. According to this result, the DBN models the gesture well enough since the model makes transitions and segments the gesture motion at highly intuitive positions in space. Note that the state change of the two hands are made in step after a certain length of time in each state. Finally, it should be noted that the state labels in the sequences are irrelevant to any physical interpretation. Rather, the consistency of the order is relevant.

5.4. Continuous gesture recognition

5.4.1. A view into gesture network internal decoding

Unlike the case of decoding a single isolated model as shown in Fig. 10, it is not easy to analyze, not to mention evaluate, the inner workings of the cyclic network of Fig. 8 for spotting and recognition of gestures from a continuous video stream.

In Fig. 11, we plotted the temporal evolution of the normalized likelihood at the final state triple of several gesture DBNs given an input sequence. To avoid messing up, we plotted the curves of only the six significant DBNs as indicated in the legend. The normalized likelihood is the per-vector average of the local segmental likelihood of a model for the partial sequence. Fig. 11 also presents several input key frames and the corresponding segmental trajectories of two hands at the top and the bottom for an annotation of the plot in the middle. The vertical lines mark the start and end points of gestures spotted by decoding the network of gesture DBNs. At the end of a gesture, the corresponding model computes the maximum likelihood among the set of models and can tell us the starting time. But the start time is often obscured by other overlying curves at that time. Note that, at the beginning of a gesture, the final state triple of the corresponding gesture DBN usually reports a poor match. But around the end point of a gesture it definitely beats all the rest even though it does not peak there. Generally it is followed by a slow or abrupt drop due to ensuing non-gestures.

Table 5
Performance of continuous gesture recognition: substitution (*S*), deletion (*D*), insertion (*I*).

No. of input gestures	Recognition results					
	Hits	Error types			Recall (%)	Precision (%)
		S	*I*	*D*		
50	42	5	5	3	84	80.77

5.4.2. Performance of continuous gesture recognition

In spotting-based continuous gesture recognition, there are three types of errors. First, an 'insertion error' (*I*) occurs when the system detects a gesture even though there is no gesture performed. Second, a 'deletion error' (*D*) occurs when there exists a gesture but the system fails to report it. The third type of error is called the 'substitution error' (*S*) which happens when the system detects a gesture but mistakes it for others possibly with wrong boundaries.

A recognition test was made on eight different long video sequences containing 50 gestures performed continuously in random sequence. The result is summarized in Table 5. The conventional evaluation criteria for spotting or detection include recall and precision rates. The recall rate is a statistical measure of the completeness of search. It is calculated by dividing the number of true positives (H) by the total number of true positives (H) and false negatives ($S+D$) as follows:

$$\text{Recall}(\%) = \frac{H}{H+S+D} \times 100(\%)$$

The recall does not include the false positives, i.e. insertion errors. The following criterion, precision rate, is a complementary measure taking the insertion errors into account.

$$\text{Precision}(\%) = \frac{H}{H+S+I} \times 100(\%)$$

Table 6 presents the results of spotting and recognizing meaningful hand gestures from one video sequence out of eight test video sequences which contain different gestures performed continuously in random sequence. In each table, the top row (consisting of three sub-rows) shows the ground truth, the middle row the decoding result, and the bottom row the discrepancy in boundary points or misclassification. Most of the gestures were spotted correctly with the segmentation errors of 2.9 frames for the start point and 2.1 frames for the end point on average. Among others, all the insertion errors came from over-segmentations at the boundaries between fillers and ensuing CL (*Close*), MF (*Move to the First Frame*), or ML (*Move to the Last Frame*) gestures. In fact, the boundaries were highly confusing and thus took a large share of the total errors. To overcome this kind of problem, we may need to consider the length of each gesture and the duration of each state.

6. Conclusion

This paper has discussed a dynamic Bayesian network (DBN)-based framework for hand gesture recognition. The use of DBN is not new in the area of the general class of human activity

Table 6
A sample segmentation result for video sequence containing eight gestures: OP (*Open*), PL (*Play*), PA (*Pause*), MF(*Move to the First Frame*), ML (*Move to the Last Frame*), TF (*10 Seconds Forward*), TB (*10 Seconds Backward*), FF (*Fast Forward*), *S* (substitution), *D* (deletion), *I* (insertion).

Seq. V												
	Input gestures and ground truth	Gesture	OP	PL	FF	TB	TF		ML	PA		MF
		Start	24	81	145	208	269		358	409		495
		End	41	115	177	260	324		393	450		528
	Detected gestures and frame alignment	Gesture	OP	PL		TB	TF	OP	ML	TB	OP	MF
		Start	28	83		209	269	339	358	433	474	495
		End	41	116		260	315	352	393	467	489	527
	Frame alignment error	Start	4	2	D	1	0	I	0	S	I	0
		End	0	1		0	−9		0			−1

recognition. But the technology still leaves room for further developments for systematic modeling and extension to real world complex patterns.

The one idea of the proposed method is the introduction of DBN tailored for hands gesture recognition. This contrasts with the fixed architecture of coupled hidden Markov model which is not deemed effective for other than tight-coupled two-party interactions. Another key feature is the DBN-based network design that can comprise a generic framework for modeling and inferencing in arbitrarily complex pattern recognition problems.

Although stochastic models are useful for describing the noisy and incomplete observations, accurate and reliable input is an important factor for the successful recognition. We applied two skin color models to detect skin pixels in each frame: the YIQ color model commonly employed to detect skin pixels and the histogram-based color model built from the pixels in the face region. The skin blobs are then tracked across frames by applying the modified method of Argyros et al. [29]. Instead of simplistic linear prediction, we computed the optical flow for an explicit prediction and accurate tracking of hand motion.

We also proposed a new hands gesture model having three hidden variables which together take five observations: chain codes of each hand's motion, relative position between the face and each hand, and relative position of two hands. We tested the DBN-based system performance with a data set which was captured from seven different subjects at different times, in total 490 video sequences. The DBN model showed the recognition rate of 99.59% in isolated gesture recognition with a cross-validation technique.

For continuous gesture recognition we designed a cyclic network of gesture DBN models including filler gesture model which links two successive gestures. Inference over the network is a dynamic programming search that spots gestures and recognizes them. The system showed the recall rate of 84% with the precision of 80.77%.

All the features used are discrete and thus may be possible to lose some important information that can be useful for better performance. However, we believe this effort is a useful and informative milestone for future research efforts on more complex gestures such as sign languages and whole body gestures.

Acknowledgement

This work was supported by the Korea Research Foundation Grant funded by the Korean Government (MOEHRD, Basic Research Promotion Fund) (KRF-2006-311-D00197). This research was also supported by the Intelligent Robotics Development Program, one of the 21st Century Frontier R&D Programs funded by the Ministry of Knowledge Economy of Korea.

Appendix A. Learning parameters in the proposed DBN

$$\hat{\pi}_i^q = E[X_1^q = i] = \frac{P(O_1^{1:5}, X_1^q = i|\Theta)}{P(O_1^{1:5}|\Theta)} \quad \text{where } q \in \{1,2\} = \left(\begin{array}{l}\text{Expected ratio of the number of transitions}\\ \text{starting in state } i \text{ at time } t=1\end{array}\right) \tag{10}$$

$$\begin{aligned}\hat{\pi}_{ijk}^3 &= E[X_1^3 = k|X_1^1 = i, X_1^2 = j] = \frac{P(O_1^{1:5}, X_1^1 = i, X_1^2 = j, X_1^3 = k|\Theta)}{P(O_1^{1:5}, X_1^1 = i, X_1^2 = j|\Theta)}\\ &= \left(\begin{array}{l}\text{Expected ratio of the number of transitions starting}\\ \text{in state } k \text{ given } (X^1, X^2) = i,j \text{ at time } t=1\end{array}\right)\end{aligned} \tag{11}$$

$$\hat{A}_{ij}^1 = \frac{E[X_t^1 = j|X_{t-1}^1 = i]}{E[X_{t-1}^1 = i]} = \frac{\sum_{t=2}^T \frac{P(O_t^{1:5}, X_t^1 = j, X_{t-1}^1 = i|\Theta)}{P(O_t^{1:5}|\Theta)}}{\sum_{t=2}^T \frac{P(O_t^{1:5}, X_{t-1}^1 = i|\Theta)}{P(O_t^{1:5}|\Theta)}} = \left(\begin{array}{l}\text{Expected ratio of the number of transitions}\\ \text{from state } i \text{ to state } j\end{array}\right) \tag{12}$$

$$\hat{A}_{gh}^2 = \frac{E[X_t^2 = h|X_{t-1}^2 = g]}{E[X_{t-1}^2 = g]} = \frac{\sum_{t=2}^T \frac{P(O_t^{1:5}, X_t^2 = h, X_{t-1}^2 = g|\Theta)}{P(O_t^{1:5}|\Theta)}}{\sum_{t=2}^T \frac{P(O_t^{1:5}, X_{t-1}^2 = g|\Theta)}{P(O_t^{1:5}|\Theta)}} \tag{13}$$

$$\begin{aligned}\hat{A}_{klmn}^3 &= \frac{E[X_t^3 = n|X_{t-1}^3 = m, X_t^1 = k, X_t^2 = l]}{E[X_{t-1}^3 = m, X_t^1 = k, X_t^2 = l]} = \frac{\sum_{t=2}^T \frac{P(O_t^{1:5}, X_t^3 = n, X_{t-1}^3 = m, X_t^1 = k, X_t^2 = l|\Theta)}{P(O_t^{1:5}|\Theta)}}{\sum_{t=2}^T \frac{P(O_t^{1:5}, X_{t-1}^3 = m, X_t^1 = k, X_t^2 = l|\Theta)}{P(O_t^{1:5}|\Theta)}}\\ &= \left(\begin{array}{l}\text{Expected ratio of the number of transitions}\\ \text{from state } m \text{ to state triple } (k,l,n)\end{array}\right)\end{aligned} \tag{14}$$

$$\begin{aligned}&\hat{B}_{iy} = \frac{E[O_t = y|X_t = i]}{E[X_t = i]} = \frac{\sum_{t=1}^T \frac{P(O_t = y, X_t = i|\Theta)}{P(O_t^{1:5}|\Theta)}}{\sum_{t=1}^T \frac{P(X_t = i|\Theta)}{P(O_t^{1:5}|\Theta)}}\\ &\text{where } (O_t, X_t) \in \left\{\begin{array}{l}(O_t^1, X_t^1), (O_t^2, X_t^1), (O_t^3, X_t^2),\\ (O_t^4, X_t^2), (O_t^5, X_t^3)\end{array}\right\} = \left(\begin{array}{l}\text{Expected ratio of the number of}\\ \text{observing symbol } y \text{ in state } i\end{array}\right)\end{aligned} \tag{15}$$

where t=1,…,T. In the above formulas $\hat{\pi}_i^q$ denotes the initial state probabilities, $(\hat{A}_{ij}^1, \hat{A}_{gh}^2, \hat{A}_{klmn}^3)$ the state transition probabilities of each hidden node, and $\hat{B}_{iy}$ the observation probabilities.

Appendix B. Pseudo-code for continuous gesture recognition

Algorithm 1. Continuous gesture detection and recognition

```
Input:
  $\mathbf{O}=O_1,O_2,\ldots,O_t,\ldots,O_T$
  Gesture DBN $m \in \{\text{Filler},\text{Open},\text{Close},\ldots,\text{Fast Rewind}\}$
Output:
  Gesture sequence $G=G_1G_2\ldots G_K$
  Observation segments $S_g=S_{g1}S_{g2}\ldots S_{gK}=(s(1,t_1),\ s(t_1+1,\ t_2),\ldots,s(t_{K-1}+1,\ t_K))$
Initialize:
  $\Delta_0(S)=1,\ \Delta_0(F)=1,\ \delta_0^m(1,1,1)=0$
DPSearch(O)
for $t$=time 1 to $T$ do
  for $gr$=each network node $\{F,S\}$ do
    for arc $e \in \{(gl \to gr)\}$ do
      for $m$=each gesture DBN attached to $e$ do
        $\delta_t^m(1,1,1)=\max_{(1,1,1),gl}\{\delta_{t-1}^m(1,1,1)A^m_{(11,11,11)},\Delta_{t-1}(gl)\times 1\}\times B^m_{(1,1,1)}(O_t)$
        $\psi_t^m(1,1,1)=\operatorname{argmax}_{(1,1,1),gl}\{\delta_{t-1}^m(1,1,1)A^m_{(11,11,11)},\Delta_{t-1}(gl)\times 1\}\times B^m_{(1,1,1)}(O_t)$
        $\varphi_t^m(1,1,1)=\begin{cases}1 & \text{if, } \delta_{t-1}^m(1,1,1)A^m_{(11,11,11)}<\Delta_{t-1}(gl)\times 1\\ \varphi_{t-1}^m(\psi_t^m(1,1,1))+1 & \text{otherwise}\end{cases}$
        for $(i,j,k)$=each state triple excluding $(1,1,1)$ of DBN $m$ do
          $\delta_t^m(i,j,k)=\max_{(\hat{a},\hat{b},\hat{c})}\{\delta_{t-1}^m(a,b,c)A^m_{(ai,bj,ck)}\}\times B^m_{(i,j,k)}(O_t)$
          $\psi_t^m(i,j,k)=\operatorname{argmax}_{(\hat{a},\hat{b},\hat{c})}\{\delta_{t-1}^m(a,b,c)A^m_{(ai,bj,ck)}\}\times B^m_{(i,j,k)}(O_t)$
          $\varphi_t^m(i,j,k)=\varphi_{t-1}^m(\psi_t^m(i,j,k))+1$
        end for
      end for
      $\Delta_t(gr)=\max_{(gl\ \text{s.t.}\ m\in L(gl,gr))}\{\delta_t^m(E^m)\}$
      $\Psi_t(gr)=\operatorname{argmax}_{(gl\ \text{s.t.}\ m\in L(gl,gr))}\{\delta_t^m(E^m)\}$
    end for
  end for
end for
$G=\emptyset$ // gesture sequence
$S_g=\emptyset$
$g=F$ // final node in the network
$t=T$
while $t\neq 0$ do
  $m=\Psi_t(g).m$
  $G=m+G$ // concatenation
  $S_g=(t-\varphi_t^m(E^m)+1,t)+S_g$ // concatenation
  $g=\Psi_t(g).gl$
  $t=t-\varphi_t^m(E^m)$ // subtraction
end while
```

References

[1] G. Johansson, Visual perception of biological motion and a model for its analysis, Perception and Psychophysics 14 (1973) 201–211.

[2] J. Aggarwal, Q. Cai, Human motion analysis—a review, Computer Vision and Image Understanding 73 (3) (1999) 428–440.

[3] V. Pavlovic, R. Sharma, T. Huang, Visual interpretation of hand gestures for human–computer interaction: a review, IEEE Transactions on Pattern Analysis and Machine Intelligence 19 (7) (1997) 677–695.

[4] M. Brand, N. Oliver, A. Pentland, Coupled hidden Markov models for complex action recognition, in: Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition, San Juan, Puerto Rico, June 1997, pp. 994–999.

[5] V. Pavlovic, Dynamic Bayesian networks for information fusion with applications to human–computer interfaces, Ph.D. Dissertation, University of Illinois at Urbana-Champaign, 1999.

[6] A. Wilson, Adaptive models for the recognition of human gestures, Ph.D. Dissertation, Massachusetts Institute of Technology, 2000.

[7] M. Yang, N. Ahuja, Recognizing hand gestures using motion trajectories, in: Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Fort Collins, USA, vol. 1, June 1999, pp. 23–25.

[8] B. Miners, O. Basir, M. Kernel, Understanding hand gestures using approximate graph matching, IEEE Transactions on Systems, Man, and Cybernetics—Part A: Systems and Humans 35 (2) (2005) 239–248.

[9] T.-P. Tian, S. Sclaroff, Handsignals recognition from video using 3D motion capture data, in: Proceedings of IEEE Workshop on Motion and Video Computing, vol. 2, 2005, pp. 189–194.

[10] A. Just, S. Marcel, A comparative study of two state-of-the-art sequence processing techniques for hand gesture recognition, Computer Vision and Image Understanding 113 (2009) 532–543.

[11] H.-K. Lee, J.-H. Kim, An HMM-based threshold model approach for gesture recognition, IEEE Transactions on Pattern Analysis and Machine Intelligence 21 (10) (1999) 961–973.

[12] C. Voglar, D. Metaxas, A framework for recognizing the simultaneous aspects of American sign language, Computer Vision and Image Understanding 81 (3) (2001) 358–384.

[13] R. León, Continuous activity recognition with missing data, in: Proceedings of IEEE International Conference on Pattern Recognition, Quebec, Canada, vol. 1, August 2002, pp. 439–446.

[14] M. Bhuyan, D. Ghosh, P. Bora, Continuous hand gesture segmentation and co-articulation detection, Lecture Notes in Computer Science: Computer Vision, Graphics and Image Processing 4338 (2006), pp. 564–575.

[15] J. Yamato, J. Ohya, K. Ishii, Recognizing human action in time-sequential images using hidden Markov Model, in: Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Urbana-Champaign, USA, June 1992, pp. 379–385.

[16] H.-I. Suk, B.-K. Sin, HMM-based gait recognition with human profiles, in: Proceedings of Joint IAPR International Workshops SSPR2006 and SPR2006, Hong Kong, China, August 2006, pp. 596–603.

[17] A. Kale, A. Rajagopalan, A. Sundaresan, N. Cuntoor, A. RoyChowdhury, V. Krueger, R. Chellappa, Identification of humans using gait, IEEE Transactions on Image Processing 13 (9) (2004) 1163–1173.

[18] Y. Du, F. Chen, W. Xu, Y. Li, Recognizing interaction activities using dynamic Bayesian network, in: Proceedings of IEEE International Conference on Pattern Recognition, Hong Kong, China, vol. 1, August 2006, pp. 618–621.

[19] S.-H. Park, J. Aggarwal, A hierarchical Bayesian network for event recognition of human actions and interactions, ACM Journal of Multimedia Systems 10 (2) (2004) 164–179.

[20] H. Avilés-Arriaga, L. Sucar, C. Mendoza, Visual recognition of similar gestures, in: Proceedings of IEEE International Conference on Pattern Recognition, vol. 1, Hong Kong, China, August 2006, pp. 1100–1103.

[21] A. Nefina, L. Liang, X. Pi, X. Liu, K. Murphy, Dynamic Bayesian networks for audio-visual speech recognition, Journal of Applied Signal Processing 11 (1) (2002) 1–15.

[22] Q. Shi, L. Wang, L. Cheng, A. Smola, Discriminative human action segmentation and recognition using semi-Markov model, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, Anchorage, USA, June 2008, pp. 1–8.

[23] H.-D. Yang, S. Sclaroff, S.-W. Lee, Sign language spotting with a threshold model based on conditional random fields, IEEE Transactions on Pattern Analysis and Machine Intelligence 31 (7) (2009) 1264–1277.

[24] H.-I. Suk, B.-K. Sin, S.-W. Lee, Robust modeling and recognition of hand gestures with dynamic Bayesian network, in: Proceedings of 19th IAPR/IEEE International Conference on Pattern Recognition, Tampa, USA, December 2008, pp. 1–4.

[25] H.-I. Suk, B.-K. Sin, S.-W. Lee, Recognizing hand gestures using dynamic Bayesian network, in: Proceedings of 8th IEEE International Conference on Automatic Face and Gesture Recognition, Amsterdam, The Netherlands, September 2008, pp. 1–4.

[26] P. Kakumanu, S. Makrogiannis, N. Bourbakis, A survey on pixel-based skin color detection techniques, Pattern Recognition 40 (3) (2007) 1106–1122.

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

[28] G. Bradski, Computer vision face tracking for use in a perceptual user interface, Intel Technology Journal 2 (1998) 1–15.

[29] A. Argyros, M. Lourakis, Real-time tracking of multiple skin-colored objects with a possibly moving camera, in: Proceedings of European Conference on Computer Vision, Prague, Czech Republic, vol. 3, May 2004, pp. 368–379.

[30] S. Beauchemin, J. Barron, The computation of optical flow, ACM Computing Survey 27 (3) (1995) 433–466.

[31] L. Rabiner, A tutorial on hidden Markov models and selected applications in speech recognition, Proceedings of the IEEE 77 (1989) 257–285.

[32] K. Murphy, Dynamic Bayesian network: representation, inference and learning, Ph.D. Dissertation, University of California, Berkeley, 2002.

[33] F. Jensen, Bayesian networks and decision graphs, Springer, 2001 (Chapter 1, pp. 3–34).

[34] A. Dempster, N. Laird, D. Rubin, Maximum likelihood from incomplete data via the EM algorithm, Journal of the Royal Statistical Society 39 (1) (1977) 1–38.

[35] C. Lee, L. Rabiner, A frame-synchronous network search algorithm for connected word recognition, IEEE Transactions on Acoustics, Speech, and Signal Processing 37 (11) (1989) 1649–1658.

[36] H. Ney, S. Ortmanns, Progress in dynamic programming search for LVCSR, Proceedings of the IEEE 88 (8) (2000) 1224–1240.

[37] ⟨http://sourceforge.net/projects/opencvlibrary/⟩.

[38] ⟨http://bnt.sourceforge.net/⟩.

[39] C. Huang, A. Darwiche, Inference in belief networks: a procedural guide, International Journal of Approximate Reasoning 15 (3) (1994) 225–263.

About the Author—HEUNG-IL SUK received the B.S. and M.S. degrees in computer engineering from Pukyong National University, Busan, Korea, in 2004 and 2007, respectively. Between 2004 and 2005, he was a visiting researcher at the Computer and Vision Research Center in the University of Texas at Austin. He is currently a Ph.D. student in the Department of Computer Science in Korea University. His research interests include machine learning, computer vision, and brain-computer interfaces.

About the Author—BONG-KEE SIN received B.S. degree in mineral and petroleum engineering from Seoul National University, Seoul, Korea, in 1985, and M.S. degree in computer science from Korea Advanced Institute of Science and Technology or KAIST in 1987. Then he had worked for the Software Research Labs of Korea Telecom until February 1999. Between 1991 and 1994, he continued his study for his Ph.D. in computer science in KAIST. In March 1999, he joined the faculty of the Department of Computer Multimedia Engineering in Pukyong National University, Busan, and is now a full professor. His general research interest includes statistical pattern recognition methods, machine learning, and applications to dynamic computer vision and other sequential signals.

About the Author—SEONG-WHAN LEE is the Hyundai Motor Chair Professor at Korea University, where he is the head of the Department of Brain and Cognitive Engineering and the director of the Institute for Brain and Cognitive Engineering. He received the B.S. degree in computer science and statistics from Seoul National University, Seoul, Korea, in 1984, and the M.S. and Ph.D. degrees in computer science from KAIST in 1986 and 1989, respectively. From 1989 to 1995, he was an assistant professor in the Department of Computer Science, Chungbuk National University, Cheongju, Korea. In 1995, he joined the faculty of the Department of Computer Science and Engineering, Korea University, Seoul, as a full professor. Dr. Lee was the winner of the Annual Best Student Paper Award of the Korea Information Science Society in 1986. He obtained the First Outstanding Young Researcher Award at the Second International Conference on Document Analysis and Recognition in 1993, and the First Distinguished Research Award from Chungbuk National University in 1994. He also obtained the Outstanding Research Award from the Korea Information Science Society in 1996.

A Fellow of the IEEE, IAPR, and Korean Academy of Science and Technology, he has served several professional societies as chairman or governing board member. He was the founding Co-Editor-in-Chief of the International Journal of Document Analysis and Recognition and has been an Associate Editor of several international journals; Pattern Recognition, ACM Transactions on Applied Perception, IEEE Transactions on Affective Computing, Image and Vision Computing, International Journal of Pattern Recognition and Artificial Intelligence, and International Journal of Image and Graphics, etc. He was a general or program chair of many international conferences and workshops and has also served on the program committees of numerous conferences and workshops. His research interests include pattern recognition, computer vision, and brain informatics. He has more than 250 publications in international journals and conference proceedings, and authored 10 books.