공 학 석 사 학 위 논 문

동적 베이스망 프레임워크 기반의 양손 제스쳐 인식

2007 년 8 월

부 경 대 학 교 대 학 원

컴 퓨 터 공 학 과

석 홍 일

공 학 석 사 학 위 논 문

동적 베이스망 프레임워크 기반의 양손 제스쳐 인식

지도교수 신 봉 기

이 논문을 석사 학위논문으로 제출함

2007 년 8 월

부 경 대 학 교 대 학 원

컴 퓨 터 공 학 과

석 홍 일

석흥일의 공학석사 학위논문을 인준함

2007 년 6 월 12 일

주 심 공학박사 권 기 룡 인

위 원 공학박사 김 종 남 인

위 원 공학박사 신 봉 기 인

사랑하는 가족에게

이 논문을 바칩니다.

목 차

그 림 목 차

표 목 차

Two Hands Gesture Recognition Based on
Dynamic Bayesian Network Framework

Heung-Il Suk

Department of Computer Engineering, The Graduate School,
Pukyong National University

Abstract

It is natural to use hand gestures in interacting with computers because hand gestures are freer in movements and much more expressive than any other body parts. In this paper, we define and recognize ten hand gestures including two-hand gestures as well as one-hand gestures. Skin blobs in a frame are segmented by two different skin color models combined. Each skin blob is modeled with a Gaussian model. For the tracking of the skin blobs, we exploit optical flows computed between the blobs in the previous frame and those in the current frame. The new mean of the Gaussian model for each blob in the current frame is predicted using the optical flows which give the motion information of each blob from the previous frame to the current frame. The motion of hands is defined the change of the mean of each Gaussian and the relative position between two hands, each hand and a face. A new gesture recognition model is proposed based on the dynamic Bayesian network framework which is relatively easy to represent the relationship among features and to incorporate new features or information to a model. Experimental results showed high recognition rate up to 99.59% with our small dataset in isolated gesture recognition and 84% of the detection rate, 76.36% of reliability was obtained in continuous gesture recognition. The proposed model and techniques are believed to have a sufficient potential for successful applications to other hand gestures recognition such as sign languages.

1. 서론

컴퓨터의 보급과 함께 사람과 컴퓨터간의 상호작용은 주로 키보드, 마우스와 같은 간단한 장치를 통해서 이루어져 왔다. 그러나, 컴퓨터의 성능 및 정보 표현에 대한 기술 발달로 기존의 장치만으로는 한계가 있으며, 이런 한계를 극복하기 위한 방안으로 여러 가지 새로운 방법들이 제안되고 있다. 대표적인 것으로는 음성을 이용한 방법이다[1]. 음성은 일상 생활에서 가장 많이 사용되는 의사 소통의 수단으로 컴퓨터와의 상호작용에서 또한 매우 자연스럽게 받아들여진다. 이런 이유로 음성 인식을 위한 연구는 수년간 이어져 오고 있고, 인식 성능 또한 우수하다. 그러나, 음성은 주변 잡음의 영향을 크게 받는다는 단점 때문에 사용상 많은 제약을 받는다. 반면, 음성과 함께 사람들이 자연스럽게 받아들일 수 있는 또 다른 방법은 손 동작이다. 사람은 사람 또는 사물과의 상호작용에 있어 끊임없이 손을 사용한다. 예를 들면, 특정 사물 또는 위치를 가리키거나 다른 사람과의 대화 속에서 자신의 감정이나 생각을 표현하기 위해 손을 사용한다. 그리고, 손은 신체의 다른 어떤 부분보다도 많은 것을 표현할 수 있고, 움직임 또한 자유롭다. 따라서, 손 동작은 사람들에게 매우 익숙하고 자연스러운 방법으로 컴퓨터와의 상호작용을 위한 적절한 방법이 된다.

사람의 동작들은 프로파일, 궤적 등과 같은 움직임에 대한 정보만으로 식별될 수 있음을 Johansson[2]의 실험에서 밝혀진 뒤, 컴퓨터 시각 분야에서도 사람의 행동에 대한 분석 및 이해에 대한 연구[3]가 활발히 진행되고 있으며, 손 동작 인식에 대한 연구는 Pavlovic 등[4]의 소개 논문에 잘 나타나 있다. 일반적으로, 손 동작 인식에 대한 연구는 영상 내에서 손을 찾아내는 문제, 손의 움직임을 추적하는 문제, 손 동작의 의미를 이해하는 문제

를 포함한다.

영상 내에서 손의 검출을 쉽게 하기 위해 CyberGlove와 같은 특수 장치[5, 6]를 이용하기도 하고, 영상 내의 정보만 이용하는 경우에는 피부색상 모델[7-10]을 주로 이용한다. 손의 추적을 위해서는 Kalman 필터 또는 particle 필터를 많은 연구에서 이용하고 있다. 마르코프(Markov) 체인과 가우스(Gaussian) 노이즈를 가정으로 물체의 움직임을 모델링하는 Kalman 필터[11]는 선형 동적 시스템으로, 관측 대상값이 비선형이거나, 가우스 분포를 따르지 않을 때, 그리고 다봉(multi-modal) 분포를 가질 때 추적에 실패하는 한계를 가진다. Particle 필터[12]는 Kalman 필터 보다 일반화된 동적 시스템으로 Kalman 필터의 선형적 시스템 한계를 극복하긴 하지만 계산량이 많다는 단점을 가진다.

손 동작은 시간의 흐름에 따른 손의 위치 변화로 표현되는데 이러한 시간 상의 변화 패턴을 모델링하는데 있어서 동적 프로그래밍(DTW)과 은닉 마프코프 모델(HMM)이 널리 이용되어 왔다. DTW는 미리 정의된 템플릿과 입력 패턴간의 유사성을 동적 프로그래밍 기법으로 비교하여 인식하는 방법으로 음성 인식에 많이 적용되었으나[13], 인식 대상이 많은 경우 템플릿의 개수도 증가하는 단점이 있다. HMM 역시 음성 인식을 위해 개발된 모델로 컴퓨터 시각 분야에서도 테니스 동작 인식[14], 걸음걸이를 이용한 생체 인식[15], 손의 움직임을 이용한 파워 포인트 제어[7] 등 다양한 응용에 적용되어 왔으며, Brand 등[16]은 두 손을 이용한 중국 무술 동작 인식을 위해 전형적인 HMM에 구조적 변화를 취한 couple HMM을 제안하였다.

베이스망(Bayesian Network: BN)[17]은 새로운 정보에 대한 표현이 용이하고, 정보들간의 상관 관계를 직관적인 방법으로 표현할 수 있으며, 추론 및 학습에 대한 알고리즘도 잘 정의되어 있어 최근 많은 연구에서 인식 모

델로 활용되고 있다. Du 등[18]은 두 사람간에 일어날 수 있는 5가지 행동들을 정의하고, 지역적 특성(contour, 모멘트, 높이 등)과 전역적 특성(속도, 방향, 거리)을 동적 베이스망(Dynamic Bayesian Network: DBN)의 관측값으로 하여 인식을 수행하였으며, Avilés-Arriaga 등[8]은 10가지의 한 손 동작들에 대해 손 영역, 중심 좌표, 위치 등의 변화를 특징값으로 하고, 동적 naïve 베이스 식별기를 인식 모델로 하였다. León 등[19]은 15개의 프레임을 하나의 윈도우로 정의하고, 윈도우 내에서의 손의 위치 변화를 BN에서 하나의 노드로 표현하였으며, 특정 노드의 값이 관측되지 않더라도 “Good-Bye”와 “Move Right”의 동작을 구분할 수 있음을 보였다. Yang 등[20]은 손의 움직임 패턴을 표현하는 궤적 정보를 특징값으로 하여, 연속된 사건들의 시간적 관계를 표현하는 각 계층들간에 있어서 시간적 지연을 적용한 다층 신경망인 TDNN(Time-Delayed Neural Network)을 인식 모델로 적용하여, 40개의 미국 수화 언어에 대한 실험에서 96.21%의 인식 성능을 보였다. 그리고, 시청각 음성 인식(Audio-Visual Speech Recognition:AVSR)에 대하여 coupled HMM과 factorial HMM 모델을 적용하여 인식을 수행한 Nefina 등[21]은 기존의 화자 독립적인 AVSR 모델과의 비교했을 때, coupled HMM이 factorial HMM[22]이나 기존의 다른 모델보다 인식 성능이 좋음을 보였다.

본 논문에서는 동적 베이스망 프레임워크를 이용하여 미디어 플레이어 제어를 위해 정의된 10가지의 양손 제스쳐를 인식하는 방법을 제안한다. 제안한 방법은 손의 추적을 위해 특별한 장치를 사용하지 않고, 영상 내의 색상들을 그대로 이용한다. 손과 손, 손과 얼굴 사이에서의 겹침 현상, 불연속적인 동작 및 비선형적인 동작에 대해서도 정확한 추적이 가능한 방법을 소개한다. 한 손 제스쳐 만을 인식한 기존 연구[7,8,10]와 달리, 본 논문에서 정의된 동작들은 한 손 동작뿐 아니라, 두 손을 이용한 동작들도 포함하며 특히, 한 손 동작을 취할 때 영상 내에 한 손만 있어야 한다는 제

약을 가지지 않는다. 그리고, 동적 베이스망 프레임워크[23]를 이용하여 미디어 플레이어 제어를 위한 새로운 제스쳐 인식 모델을 제안한다. 제안한 동적 베이스망을 이용하여 훈련 및 인식을 수행한 결과 독립 제스쳐 인식에서는 최대 99.57%의 높은 성능을 보였고, 연속 제스쳐에 대한 제스쳐 검출 및 인식은 84%의 인식률과 76.36%의 신뢰도를 얻었다.

논문의 구성은 다음과 같다. 2 장에서는 시스템의 전체적인 구성을 살펴보고, 3 장에서는 얼굴과 손 영역의 검출 및 추적 방법을 설명한다. 미디어 플레이어 제어를 위한 동작 정의 및 특징 추출은 4 장, 동적 베이스망을 이용한 동작 인식 모델의 정의, 추론 및 학습에 대한 것은 5 장에서 각각 살펴본다. 6 장에서는 실험 결과를 보이고, 7 장에서는 결론 및 향후 연구에 대해 살펴본다.

2. 시스템 구성

본 논문에서 제안하는 시스템의 전제적인 구성은 그림 1과 같고, 각 단계는 다음의 역할을 수행한다.

- 배경 제거: 입력 비디오 시퀀스의 각 프레임에 대하여 배경 이미지와의 차이를 계산하여 전경 이미지 추출.
- 피부색 검출: Haar-like 얼굴 검출기 및 YIQ 색상 모델을 이용.
- 영역 추적: 전경 이미지와 피부색 검출을 통해 추출된 손과 얼굴의 각 영역에 대한 가우스(Gaussian) 모델 생성 및 추적.
- 특징 추출: 각 손의 움직임을 표현하기 위한 특징 추출.
- 모델링: 각 제스쳐를 모델링하는 동적 베이스망을 훈련하고 새로운 데이터에 대해서 인식 수행.

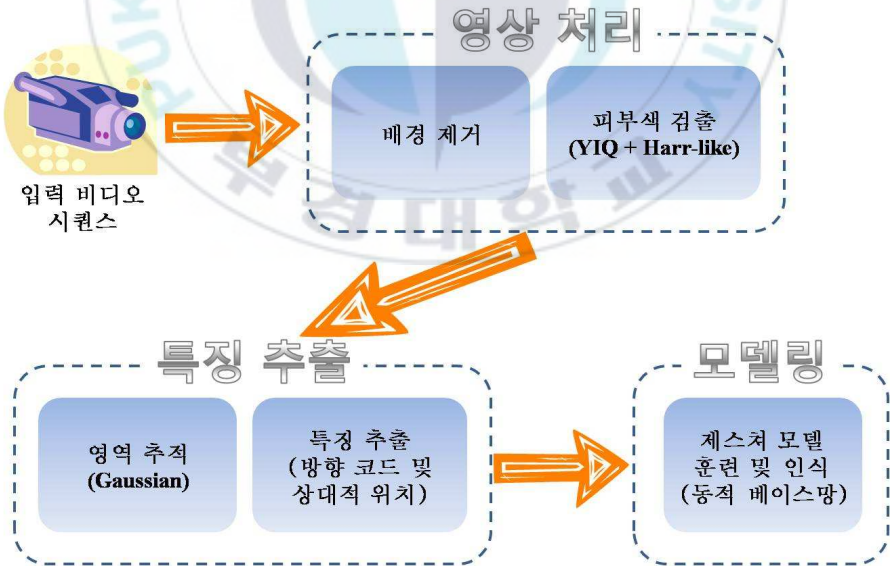


그림 1. 시스템 구성

3. 얼굴과 손 영역에 대한 검출 및 추적

얼굴과 손에 대한 검출 및 추적은 손 동작 인식을 위해 필수적인 부분으로 시스템의 성능에 큰 영향을 미친다. 검출 및 추적에 대한 본 논문에서의 해결 방안에 대해 살펴보자.

3.1 피부 영역 검출(Skin Detection)

피부색 검출은 빛 조건의 변화에 크게 영향을 받는다. 많은 사람들이 피부색 검출에 대한 연구를 해왔으며, 지금도 활발히 연구하고 있지만 다양한 환경 또는 조건의 변화에 강인한 방법은 아직까지 알려져 있지 않다[24].

본 논문에서는 피부 영역 검출을 위하여 두 가지 방법을 결합한다. 첫 번째 방법은 가장 간단하고 널리 이용되는 것 중 하나로, RGB 색상 모델을 YIQ 색상 모델로 변환하여 I의 값이 미리 정해진 임계치의 범위를 만족하는 경우 피부색으로 선택하는 것이다. 두 번째 방법은 Haar-like 얼굴 검출기[25]로 검출된 얼굴 영역 내의 픽셀들을 이용하여, 현재의 빛 조건 및 사용자에 적합한 피부색 특성을 반영한 피부색 모델을 생성하고 이를 피부색 검출에 이용하는 것이다. 이는 Haar-like 얼굴 검출기가 다양한 빛 조건에서도 정면을 향하고 있는 얼굴을 정확히 잘 찾아 낸다는 것을 이용한 것이다. 영상 내에 있는 사용자의 얼굴을 검출하고, 검출된 얼굴 부분에서 너무 어둡거나 너무 밝지 않은 픽셀들을 이용하여 그 사용자에게 맞는 피부 색상 모델을 생성한다. 피부 색상 모델은 HSV 색상 모델에서의 색상 값에 대한 히스토그램으로 정의한다. 임의의 픽셀에 대하여 그 픽셀에 대

한 피부색으로서의 확률값을 히스토그램에서 계산하여 임계치보다 큰 경우 피부색으로 결정한다[26]. 이 방법은 조명의 변화에도 빠르게 적응하며, 임의의 사용자에 대해서 사용자에게 맞는 피부 색상 모델을 생성 및 활용하기 때문에 강인한 검출을 할 수 있다는 장점이 있다. 그림 2는 YIQ 색상 모델만을 이용한 것(그림 2(b)), 얼굴색으로 만든 색상 모델만을 이용한 것(그림 2(c)), 그리고 두 모델을 결합(그림 2(d))하여 피부색을 검출한 결과를 각각 보여주고 있다.

(a) 입력 이미지 (b) YIQ 색상을 이용

(c) 얼굴 색상을 이용 (d) (b)+(c)

그림 2. 피부색 검출 방법 비교

3.2 영역 추적(Blob Tracking)

피부 영역의 추적에 있어서 손과 손, 손과 얼굴간의 겹침 현상이 있을 때 정확한 추적이 어렵다. 이를 해결하기 위해 손 및 얼굴에 해당하는 각 영역을 하나의 가우스(Gaussian) 모델로 표현하여 추적하는 Argyros 등의 방법[27]을 이용하였다. 그러나, 현재 프레임에서의 각 영역의 위치를 예측하기 위해 이전 두 프레임에서의 변화율(velocity)을 사용한 기존의 방법과는 달리 본 논문에서는 광류를 이용한다. 이전 두 프레임에서의 변화율을 이용하는 경우, 손의 움직임이 일정한 속도를 가질 때 비교적 정확한 추적이 가능하지만 그렇지 못한 경우에는 오차를 범하게 되고, 추적에 실패하게 된다. Kalman 필터를 사용하는 경우도 있지만, Kalman 필터의 특성상 추적하고자 하는 영역이 비선형으로 움직이는 경우 추적을 하지 못하는 단점이 있다. 이에 대한 해결책으로 본 논문에서는 영역의 위치 예측을 위해 광류(optical flow)를 이용한다. 광류는 이전 프레임에서 현재 프레임으로의 움직임에 대한 정보를 직접적으로 제공해주므로 추적하고자 하는 영역이 일정한 속도를 가지거나, 선형적으로 움직여야 한다는 제약이 없어도 추적이 가능하다.

현재 프레임에서의 손과 얼굴 영역을 추적하기 위해 이전 프레임에서 검출된 영역과 현재 프레임에서 검출된 영역 사이에서 광류를 계산한 뒤, 이전 프레임에서 하나의 가우스 모델을 갱신하는데 사용되었던 영역의 픽셀들에 해당하는 광류값들을 x 방향과 y 방향에 대해 각각 정렬을 하고, 각 방향에서의 중간값 $\mathbf{m}$을 이용하여 식 (1)과 같이 현재 프레임에서의 영역 i에 대한 가우스 모델의 새로운 평균으로 예측한다.

$$N\left(\boldsymbol{\mu}_i^{predict}, \boldsymbol{\Sigma}_i\right) = N\left(\boldsymbol{\mu}_i + \mathbf{m}_i, \boldsymbol{\Sigma}_i\right) \quad (1)$$

앞서 설명한대로, 이는 이전 프레임에서 현재 프레임으로의 움직임에 대한 정보를 현재 프레임에서의 추적에 즉시 이용하므로 이전 두 프레임에서의 변화율을 이용한 방법보다 더욱 정확하고, 속도의 변화에도 민감하게 반응하여 정확한 추적이 가능하다.

그림 3은 이전 두 프레임에서의 변화율을 이용한 경우와 광류를 이용한 경우에 대한 결과를 각각 보여 주고 있다. 그림 3(a)의 변화율을 이용한 방법은 프레임 10에서의 광류(그림 3(b))에 나타난 것과 같이 프레임 9와 프레임 10사이에 손의 움직임이 없었기 때문에 프레임 10과 프레임 11사이에서도 큰 움직임이 없을 것이라 예측하고 추적을 했으나, 실제로는 움직임이 나타났다. 이로 인해 오차가 생기게 되고, 연속된 프레임에서 손의 정확한 위치를 찾지 못하여 프레임 13에서는 추적에 실패하게 된다. 그러나, 광류를 이용한 그림 3(c)는 프레임 10 이전에서의 손의 움직임과 상관없이 그림 3(b)의 광류가 이전 프레임에서 현재 프레임으로의 각 영역의 변화에 대한 움직임 정보를 제공해주기 때문에 추적에 성공하게 된다.

손과 얼굴에 대한 추적 방법은 다음과 같다. 프레임 t의 각 영역에 대하여 이전 프레임까지의 추적에서 사용 또는 생성되었던 가우스 모델이 현재 프레임의 각 영역을 지원하고 있는지 확인한다. 여기서, 가우스 모델과 영역 사이에 겹치는 부분이 있으면 가우스 모델이 해당 영역을 지원한다고 말한다. 만약, 지원하는 가우스 모델이 없는 경우에는 해당 영역에 대한 새로운 가우스 모델을 생성한다. 지원하는 모델이 있는 경우에는 이전 프레임 $t-1$에서의 피부 영역과 현재 프레임 t에서의 피부 영역 사이에서의 광류를 계산하고, 이를 이용하여 가우스 모델의 평균을 식 (1)과 같은 방법으

로 이동하여 현재 프레임에서의 피부 영역의 위치를 예측한다. 예측된 가우스 모델을 이용하여 Argyros가 제안한 방법[27]을 따라 가우스 모델의 모양을 결정짓는 평균과 공분산을 갱신한다. 이동된 가우스 모델의 평균은 이전 프레임에서 현재 프레임으로의 움직임 정보를 이용한 것이기 때문에 정확한 추적이 가능하게 된다. 갱신된 가우스 모델은 다음 프레임에서의 추적에 사용된다.

(a) 이전 두 프레임에서의 변화율(velocity)을 이용한 방법

(b) 이전 프레임과 현재 프레임간의 광류(optical flow)

(c) 이전 프레임과 현재 프레임간의 광류를 이용한 방법

그림 3. 변화율을 이용한 방법과 광류를 이용한 방법의 결과

4. 손 동작 정의 및 특징 추출

4.1 손 동작 정의

미디어 플레이어 제어를 위한 10가지 동작들을 그림 4와 같이 정의한다. 동작들은 한 손을 이용한 것(5가지)과, 두 손을 이용하는 것(5가지)을 포함한다. 그림에서의 검은색으로 채워진 원은 손의 시작 위치를 의미하고, 화살표는 손의 이동 궤적을 표시한 것이다.

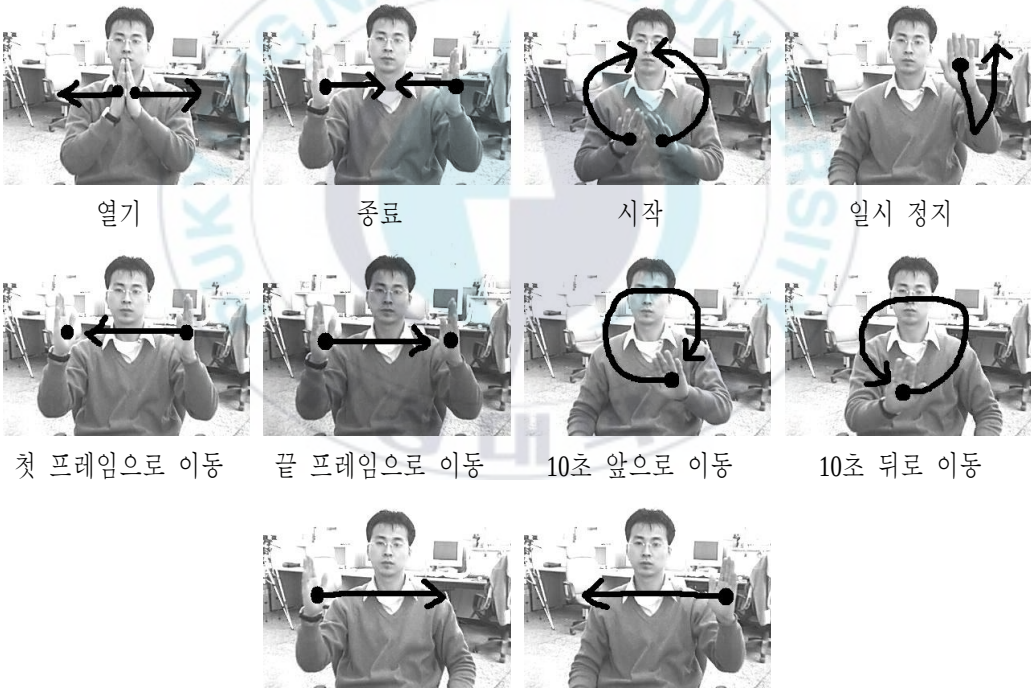


그림 4. 미디어 플레이어 제어를 위한 손 동작 정의

4.2 특징 추출(Feature Extraction)

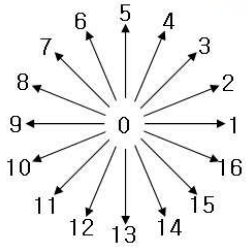
손 동작의 인식에 있어서 가장 중요한 정보는 손의 움직임 정보이다. 본 논문에서는 손의 움직임 정보를 표현하기 위해 손 영역을 모델링하는 가우스(Gaussian) 분포에서의 평균의 변화를 이용한다. 즉, 이전 프레임에서의 평균과 현재 프레임에서의 평균의 변화를 그림 6(a)와 같이 17 방향 코드로 양자화하여 표현한다. 방향 코드에서 '0'은 움직임이 없음을 의미한다. 그러나, 손의 움직임에 대한 방향 정보만으로는 서로 다른 동작간의 명확한 구분이 어렵다. 예를 들면, 그림 5와 같은 경우이다. 그림 5(c)와 그림 5(d)에서 채워진 원은 손의 시작 위치를 나타내고, 점선으로 된 원은 손의 마지막 위치를 나타낸 것이다. 그리고, 화살표는 손의 이동 경로를 의미한다.

(a) '빨리 감기' 명령 제스쳐 (b) '뒤로 감기' 명령 제스쳐

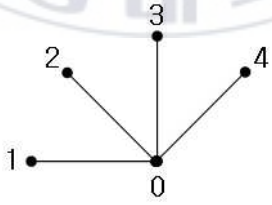
(c) (a)의 궤적 (d) (b)의 궤적

그림 5. 방향 코드만을 이용한 경우 발생하는 모호성

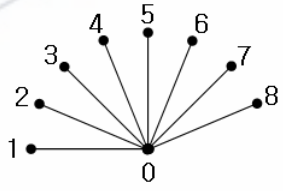
그림 5(a)는 ‘빨리 감기’ 명령이지만 손의 움직임에 대한 궤적 정보를 양자화 한 방향 코드만을 이용하면 그림 5(c)와 같이 표현될 것이고, 이는 ‘끝 프레임으로 이동’ 명령으로 인식되게 된다. 그림 5(b) 또한, ‘뒤로 감기’ 명령이지만 궤적 정보를 양자화 한 방향 코드만을 이용할 경우 그림 5(d)와 같이 표현되고, 이는 ‘첫 프레임으로 이동’ 명령으로 인식되게 된다. 이런 문제를 해결하기 위해 오른손과 왼손의 상대적 위치에 대한 정보를 그림 6(b)와 같은 경우 중 하나로 나타낸다. ‘0’은 두 손이 겹쳐져 있는 경우를 의미한다. 마지막으로 사용한 특징은 얼굴과 각 손의 상대적인 위치로 9가지의 경우 중 하나로 그림 6(c)와 같이 표현된다. 이는 고정되어 있어야 하는 손이 미세하게 움직이는 경우 ‘첫 프레임으로 이동’, ‘마지막 프레임으로 이동’에 대한 동작이 ‘종료’ 동작으로 잘못 인식될 수 있음을 해결하기 위한 것으로, 실제 방향 코드와 두 손의 상대적 위치만을 특징으로 하여 본 논문에서 제안한 모델을 훈련하고 인식 한 결과 잘못 인식된 것들의 대부분은 ‘첫 프레임으로 이동’, ‘마지막 프레임으로 이동’에 대한 동작들이 ‘종료’ 동작으로 인식된 것이었고, 세 번째의 특징을 추가하여 인식한 결과 더 높은 인식률을 보였다.



(a) 손의 움직임



(b) 양손의 상대적 위치



(c) 얼굴-손 상대적 위치

그림 6. 특징 표현

영상 처리 과정에서의 오차나 사용자의 미세한 움직임에 의해 방향 코드와 상대적 위치 값이 바뀌게 되고, 이로 인해 인식률에 영향을 미치는 것은 양자화된 값을 사용하는 것에 따른 문제점이 될 수도 있다. 그러나, 절대적 위치 좌표를 사용하는 경우 카메라와 사용자 사이의 거리 변화에 따라 관측 데이터의 확률 분포의 변화율이 커져서 인식률은 더 나빠질 수 있다.

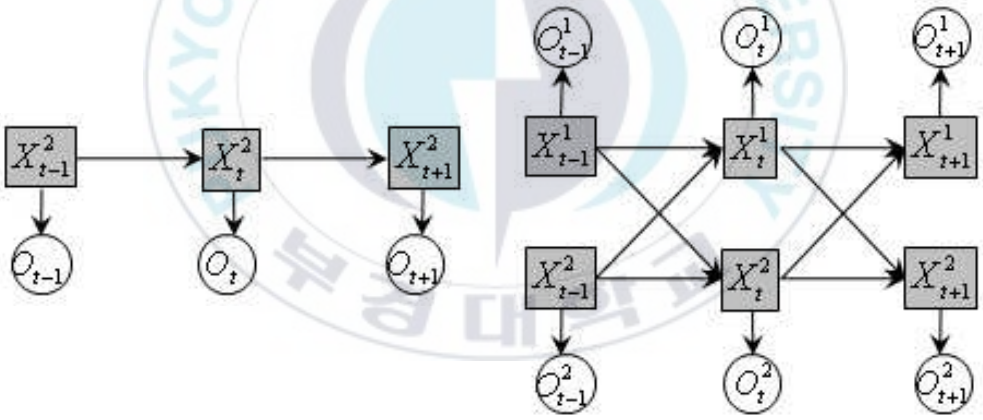
5. 인식 모델

5.1 동적 베이스망(Dynamic Bayesian Netowrk: DBN)

시계열 데이터를 모델링 하는데 있어서 가장 널리 이용되는 것 중 하나는 은닉 마르코프 모델(hidden Markov model: HMM)[28]이다. 음성 인식을 위해 처음 제안된 HMM은 음성 인식뿐 만 아니라 컴퓨터 시각 분야를 포함한 여러 분야에서 우수한 성능을 보여왔으며, 다양한 변형 모델도 제안되어 왔다[23]. 그러나, 모델의 구조적 특징이 고정되어 있다는 제약이 있다. 두 개 이상의 은닉 노드에 대한 마르코프 과정을 표준 HMM과 같이 하나의 상태 노드로 표현한다고 가정하자. 이때, 각 마르코프 과정에서의 노드가 가질 수 있는 상태들을 모두 곱한 수만큼의 상태수를 표준 HMM(그림 7(a))에서는 하나의 상태 노드로 표현할 수 있어야 하므로 상태수가 지수적으로 증가하게 된다. 상태수가 급격히 증가함에 따라 상태들에 대한 확률 분포를 훈련하기 위해서는 많은 양의 훈련 데이터를 필요로 한다. Coupled HMM(그림 7(b))의 경우에는 마르코프 과정의 수가 증가함에 따라 노드간의 coupling 수가 증가하게 되는데 이 또한 계산량을 높이는 원인이 된다. 그리고, HMM은 새로운 정보 또는 특징이 추가가 어렵다.

HMM의 단점을 보완 및 해결하고 일반화 된 것이 DBN[23]이다. DBN은 그래프 모델(Graphical Model)[29]의 하나인 베이스망(Bayesian Network: BN)에 시간 정보가 추가된 모델로서 시계열 데이터를 모델링 하기에 유용한 도구이며, 최근 많은 관심을 받고 있다. BN은 사이클이 없는 방향성 그래프로 각 노드는 하나의 확률 변수를 나타내고, 방향성을 가진 에지는 변수들 간의 통계적 의존성 즉, 부모 노드의 값이 주어졌을 때 해당 변수의 확률 분

포를 표현한다. BN은 변수들간의 조건부 독립성(*d*-분리[30])를 이용하여 변수들에 대한 결합 확률 분포를 효율적으로 표현 및 계산한다. 조건부 독립성은 모델의 구조와 해당 모델에 대한 추론 및 학습을 수행하는데 필요한 계산을 간단하도록 해 준다. BN과 같은 그래프 모델에서는 확률 변수들 간의 조건부 독립성을 확인하기 위해 수식적 분석을 할 필요 없이 그래프 상에 나타난 노드들의 관계로부터 직접적으로 쉽게 알 수 있다는 장점을 가진다. DBN은 BN의 노드들에 대해 $t-1$ 시간의 노드 값이 t 시간에서의 노드의 값에 영향을 준다는 마르코프 과정을 적용하여 노드들 간의 시간적 의존 관계를 추가로 표현한 것으로, BN에서의 추론 알고리즘들을 그대로 적용할 수 있다.



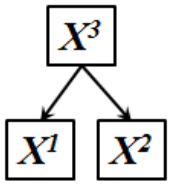
(a) 표준 HMM (b) Coupled HMM

그림 7. 동적 베이스망으로 표현한 표준 HMM과 coupled HMM

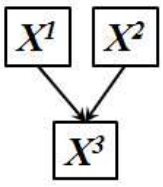
5.2 제안하는 양손 제스쳐 모델

본 논문에 정의된 동작들은 양손을 모두 사용하는 동작뿐만 아니라, 한 손 동작들도 포함하고 있다. 양손 제스쳐의 경우 오른손과 왼손의 각 동작들은 서로 연관성을 가지며 이런 동작들은 coupled HMM으로 모델링 될 수도 있지만 사실, 두 손의 움직임이 완전한 연관성을 가지는 것이 아니며, 어느 정도의 독립성을 가진다. Coupled HMM은 각 손을 모델링하는 은닉 노드들간에 완전한 연결을 이루고 있기 때문에 각 손을 모델링하는 노드들간의 의존성이 너무 높다. 본 논문에서는 두 손의 상대적 위치를 관측 데이터로 가지는 은닉 상태 노드 X^3를 추가하여 각 손의 움직임을 모델링하면서 두 손의 완화된 연관성을 표현하는 새로운 DBN을 정의한다.

은닉 상태 노드를 추가할 때 각 손을 모델링하는 기존의 두 노드(X^1, X^2)들과의 관계에 대해서는 다음과 같은 두 가지 가능성이 있다. 첫째는 그림 8(a)와 같이 두 손의 상대적 위치를 모델링하는 은닉 노드 X^3에서 각각의 손을 모델링하는 노드 X^1과 X^2로 화살표를 연결하는 것이고, 둘째는 그림 8(b)와 같이 각각의 손을 모델링하는 노드 X^1과 X^2에서 손의 상대적 위치를 모델링하는 은닉 노드 X^3로 화살표를 연결하는 것이다.



(a) 손의 상대적 위치를 모델링하는 은닉 노드 X^3에서 각 손을 모델링하는 노드 X^1과 X^2로의 연결



(b) 각 손을 모델링하는 노드 X^1, X^2에서 손의 상대적 위치를 모델링하는 은닉 노드 X^3로의 연결

그림 8. 각 손을 모델링하는 노드들(X^1, X^2)와 새로운 노드 X^3와의 관계

손의 상대적 위치가 그림 9(a)에서 그림 9(b)로 바뀌는 경우를 살펴보자. 이때, 그림 8(a)의 경우는 두 손의 상대적 위치가 바뀐 결과가 손의 움직임에 영향을 미치는 것을 표현한다. 이는 상대적 위치 변화의 결과가 그림 9(c)와 같이 왼쪽에 있던 손이 이전 위치보다 내려간 것인지, 아니면 그림 9(d)와 같이 오른쪽에 있던 손이 이전 위치보다 올라간 것인지를 명확히 설명할 수 없다. 그러나, 그림 8(b)는 각 손이 특정 방향으로 움직인 결과 즉, 왼쪽 손이 내려갔기 때문에 또는, 오른쪽 손이 올라갔기 때문에 두 손의 상대적 위치가 바뀌었다는 것을 설명할 수 있기 때문에 더 옳다는 것을 알 수 있다. 따라서, 그림 8(b)와 같은 노드간의 관계를 새로운 제스쳐 인식 모델에 반영한다.

(a) 초기 상대적 위치 (b) 변화된 상대적 위치

(c) 왼손이 내려간 경우 (d) 오른손이 올라간 경우

그림 9. 두 손의 상대적 위치의 변화

시간의 흐름에 따른 은닉 노드들 간의 관계에 대해서 1차 마르코프 과정을 가정하면 그림 10과 같이 도식화 할 수 있으며, 이를 양손 제스쳐 동

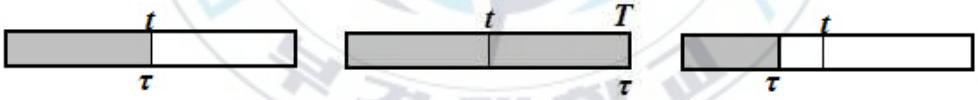
작에 대한 인식 모델로 제안한다. 그림 10에서 회색 노드들은 관측될 수 없는 은닉 노드들을 나타내고, 흰색의 노드들은 관측 가능한 노드들을 나타낸다. 두 손의 상대적 위치를 모델링하는 노드(X^3)의 시간적 관계 모델 $P(X_t^3 \mid X_{t-1}^3)$은 coupled HMM에서 완전히 연결되어 있던 두 노드 X^1과 X^2의 높은 의존성을 완화시켜 서로간의 의존성을 간접적으로 표현한다. 따라서, 시간 $t-1$에서의 각 손의 움직임 정보와 시간 t에서의 각 손에 대한 움직임의 정보가 노드 X_{t-1}^3과 X_t^3에 의해 간접적으로 연결되므로 coupled HMM에서 완전히 연결되어 있던 이들 두 노드사이의 연결은 제거된다.

그림 10. 손 동작 인식을 위한 동적 베이스망 모델

5.3 추론(Inference)

DBN에서의 추론은 관측 데이터가 주어졌을 때 은닉 노드가 임의의 값을 가질 주변 확률 분포 $P(X_t \mid O_{1:t})$를 계산하는 것이다. 이는 τ의 값에

따라 그림 11과 같이 다른 종류의 추론으로 분류될 수 있다[23]. 그림에서 회색으로 칠해진 부분은 관측된 데이터의 양을 표시한다. 첫 번째, $t=\tau$인 경우(그림 11(a))는 필터링에 해당한다. 이는 시간 1부터 시간 t까지의 데이터를 관측했을 때, 시간 t에서의 노드 X_t가 특정값을 가질 확률 $P(X_t|O_{1:t})$을 계산하는 것이다. 두 번째, $t<\tau$인 경우(그림 11(b))는 평활화로 시간 τ까지의 데이터를 모두 관측했을 때, 시간 τ 이전인 과거 시간에서의 노드 X_t가 특정값을 가질 확률 $P(X_t|O_{1:T})$를 계산하는 것으로 DBN의 파라미터 값을 오프라인으로 훈련할 때 사용된다. 마지막으로 $t>\tau$인 예측(그림 11(c))은 관측 데이터가 시간 t이전까지 주어졌을 때, 시간 t에서의 노드 X_t가 특정값을 가질 확률 $P(X_t|O_{1:t-1})$를 계산한다. 이는 Kalman 필터에서와 같이 미래의 상태를 미리 예측할 필요가 있는 경우에 사용된다. 본 논문에서는 관측 데이터가 주어졌을 때, DBN에 대한 관측 데이터의 우도를 계산하기 위해 필터링을 적용하고, 파라미터 학습을 위해 평활화를 적용한다.



(a) 필터링 (b) 평활화 (c) 예측

그림 11. 동적 베이스망에서의 추론

관측 데이터가 시간 t까지 주어졌을 때 시간 t에서의 은닉 노드에 대한 확률 분포를 계산하는 필터링에 대해 좀더 자세히 살펴보자. 예를 들어, DBN에서의 은닉 노드가 N개의 확률 변수들의 집합($X_t^i, i\in\{1,\cdots,N\}$)으로 표현되고, 이 확률 변수들은 이산값을 가지며, 관측 데이터는 M개의 확률 변

수들의 집합($O_t^i, i \in \{1,\cdots,M\}$)으로 표현된다고 가정하자. BN에서의 결합 확률 분포는 *d*-분리[30]에 따른 조건부 확률 분포의 곱으로 간단히 표현될 수 있다는 사실을 DBN에 그대로 적용할 수 있다. 즉, 시간 t–1의 BN이 주어졌을 때, 시간 t의 BN은 시간 t–1 이전의 모든 BN과 조건부 독립이 되므로 시간 t에서의 BN에 대한 확률 분포 계산을 위해서는 시간 t–1의 BN만 고려하면 된다. 따라서, 정의된 DBN에 대한 결합 확률 분포는 식 (2)와 같다.

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

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

본 논문에서는 추론을 위해 interface 알고리즘[23]을 이용한다. V_t를 시간 t에서의 노드들의 집합, E를 DBN에서의 노드 간의 에지 집합이라 할 때, $E^{tmp}(t)$를 식 (3)과 같이 이웃한 두 시간에서의 BN의 노드간 에지 집합이라 정의하자.

$$E^{tmp}(t) = \{(u,v) \in E \mid u \in V_{t-1}, v \in V_t\} \tag{3}$$

Interface 알고리즘은 DBN에서 이웃하는 두 시간에서의 BN(2 time-slice Bayesian network: 2TBN)을 고려했을 때, 시간 $t-1$에서 시간 t로 연결된 화살표를 가진 노드들의 집합은 과거(시간 $t-1$이전의 모든 BN)과 미래(시간 t이후의 모든 BN)을 d-분리 하기에 충분하다는 사실을 이용한다. 이때, 과거와 미래를 d-분리 해주는 노드들의 집합을 interface라 한다. 특히, 필터링에서는 전진 interface라하고, 평활화에서는 후진 interface라 한다. $ch(v)$를 노드 v의 자식 노드들의 집합이라 했을 때, interface는 다음과 같이 정의된다.

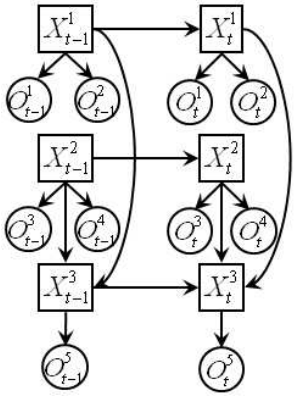
$$\text{전향 interface} \quad : \quad I_t^{\rightarrow} = \left\{ u \in V_t \mid (u,v) \in E^{tmp}(t+1),\ v \in V_{t+1} \right\}$$

$$\text{후향 interface} \quad : \quad I_t^{\leftarrow} = \left\{ \begin{array}{l} v \in V_t \mid (u,v) \in E^{tmp}(t) \text{ or} \\ \qquad \exists w \in ch(v) : (u,w) \in E^{tmp}(t), u \in V_{t-1} \end{array} \right\}$$

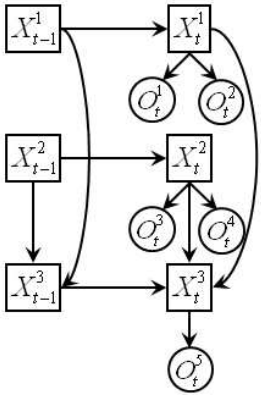
그리고, interface 알고리즘은 DBN을 접합(junction) 트리로 변환 할 때, interface에 속하는 노드들을 포함하는 clique 노드가 반드시 존재해야 한다는 조건을 가지는데, 이를 만족하는 접합 트리를 구성한 뒤에는 BN에서 가장 널리 이용되는 접합 트리 알고리즘(Junction Tree Algorithm: JTA)[31]을 적용할 수 있다.

본 논문에서 제안한 DBN에 대한 접합 트리의 구성 과정 및 이웃한 두 시간에서의 BN에 대한 interface는 그림 12와 같으며, 회색으로 칠해진 노드들은 interface를 나타낸다. DBN에서 시간 t-1에서의 BN이 주어지면 $t-1$ 이전 시간의 BN과 시간 t에서의 BN은 d-분리에 의해 조건부 독립이 되므로, 시간 t에서의 필터링을 위해서는 그림 12(a)와 같은 두 시간에서의 BN(2TBN)만 고려하면 된다. 그리고, 시간 $t-1$에서의 interface 노드들($X_{t-1}^{1:3}$)이 주어지게 되면 시간 $t-1$에서의 관측 노드들($O_{t-1}^{1:5}$)과 시간 t에서의 BN 또한 d-분리에 의해 조건부 독립이 되므로 그림 12(b)와 같이 2TBN에서 $t-1$

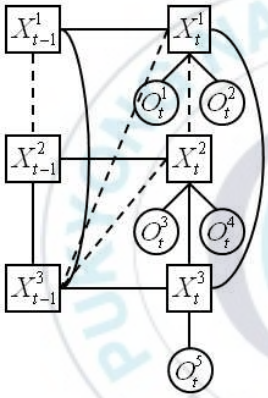
시간에서의 관측 노드들을 제거한 1.5TBN으로 간단히 할 수 있다. 그림 12(c)와 그림 12(d)는 1.5TBN에 대한 교화된(moralized) 그래프와 삼각화된(triangulated) 그래프를 보여주고 있으며, 접합 트리 알고리즘이 적용될 최종적인 접합 트리는 그림 12(e)와 같다. 그림 12(e)에서 clique 노드 ($X_{t-1}^1, X_{t-1}^2, X_{t-1}^3, X_t^1$)와 ($X_{t-1}^3, X_t^1, X_t^2, X_t^3$)은 시간 t−1에서의 interface $X_{t-1}^{1:3}$과 t에서의 interface 노드 $X_t^{1:3}$을 각각 포함하고 있으므로 interface 알고리즘에 대한 조건을 만족한다. 그리고, 이들 interface 노드들은 이웃한 시간 (t−2, t−1)과 (t, t+1) 사이에서 정의되는 1.5TBN에 대한 접합 트리를 연결시켜주는 분리자가 된다.



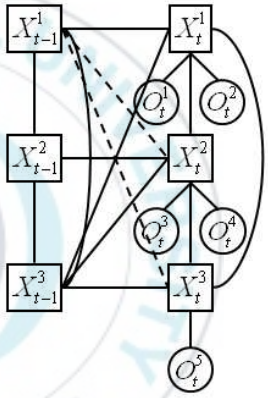
(a) 2 시간에서의 BN (2TBN)



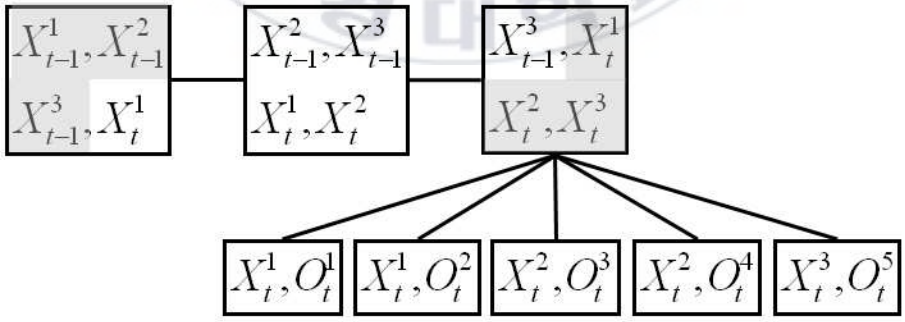
(b) 1.5 시간에서의 BN (1.5TBN)



(c) 교화된(moralized) 그래프



(d) 삼각화된(triangulated) 그래프



(e) 제안된 DBN에서의 2TBN에 대한 접합 트리(결과)

그림 12. Interface 알고리즘 적용을 위한 접합 트리 구성 과정

5.4 학습(Learning)

DBN에서의 학습은 훈련 데이터 $O_{1:T}^{1:5}$가 주어졌을 때, 식 (4)와 같이 주어진 관측 데이터에 대한 DBN의 우도를 최대로 하는 파라미터의 값을 결정하는 것이다.

$$\hat{\theta} = \arg\max_{\theta} P\left(O_{1:T}^{1:5} \mid \theta\right) \tag{4}$$

위 식에서 파라미터 θ는 HMM을 포함한 모든 상태 공간 확률 모델에서와 마찬가지로, 은닉 노드들의 초기 확률 분포(π), 은닉 노드들에서의 상태 전이 확률 분포(A), 은닉 노드들에서의 관측값 출력 분포(B)를 포함한다. 하지만, 은닉 상태를 하나의 확률 변수로 표현하는 HMM과는 달리, DBN은 여러 개의 확률 변수들로 은닉 상태를 표현한다. 여러 개의 확률 변수를 사용함으로 해서 상태의 분산된 표현을 이용할 수 있고, 모든 상태들의 조합을 하나의 확률 변수로 표현해야 하는 HMM에 비해 계산량을 훨씬 많이 줄일 수 있으며, 적은 수의 데이터로도 훈련이 가능하다.

본 논문에서 정의된 DBN은 은닉 노드들 즉, 관측할 수 없는 값들을 가지는 노드들을 포함하고 있으므로 훈련 데이터에 대한 DBN의 우도를 최대로 하기 위한 파라미터의 값을 결정하기 위해 EM 알고리즘[32]을 적용한다. EM 알고리즘은 관측되지 않은 값 또는 관측될 수 없는 값을 현재의 파라미터 값을 이용하여 추정한 뒤, 추정된 값과 실제 관측된 값으로 구성되는 완전한 데이터에 기반하여 우도가 최대가 되도록 파라미터의 값을 갱신하는 반복 알고리즘이다. 파라미터의 값을 결정하기 위해서는 훈련 데이터로부터 각 노드에 대한 충분 통계량만 계산하면 되는데, 이들 충분

통계량은 기대값을 이용하여 수식화 될 수 있다. 기대값은 앞서 설명한 interface 알고리즘을 이용하여 계산한다.

본 논문에서 정의된 DBN의 파라미터를 갱신하는 수식은 다음과 같다.

$$\pi = P\left(X_1^q = i\right) = E\left[X_1^q = i\right] \quad where,\ q \in \{1,2,3\}$$

$$A = \begin{cases} P\left(X_t^1 = i \mid X_{t-1}^1 = j\right) = \dfrac{E\left[X_t^1 = i \mid X_{t-1}^1 = j\right]}{E\left[X_{t-1}^1 = j\right]} \\ P\left(X_t^2 = i \mid X_{t-1}^2 = j\right) = \dfrac{E\left[X_t^2 = i \mid X_{t-1}^2 = j\right]}{E\left[X_{t-1}^2 = j\right]} \\ P(X_t^3 = i \mid X_{t-1}^3 = j, X_t^1 = k, X_t^2 = l) = \dfrac{E\left[X_t^3 = i \mid X_{t-1}^3 = j, X_t^1 = k, X_t^2 = l\right]}{E\left[X_{t-1}^3 = j, X_t^1 = k, X_t^2 = l\right]} \end{cases}$$

$$B = P\left(O_t = y \mid X_t = i\right) = \frac{E\left[O_t = y \mid X_t = i\right]}{E\left[X_t = i\right]}$$

$$where,\ (O_t, X_t) \in \left\{(O_t^1, X_t^1), (O_t^2, X_t^1), (O_t^1, X_t^2), (O_t^2, X_t^2), (O_t^1, X_t^3)\right\}$$

그러나, 만약 훈련 데이터가 충분하지 못해 특정값에 대한 관측이 훈련 데이터에 포함되어 있지 않다면, 해당값의 확률 분포를 표현하는 파라미터에 대한 훈련 결과는 0의 확률값을 가지게 된다. 그릭고, 실제 적용에서 그 값이 관측될 경우 모델 전체에 대한 확률은 0이 되게 된다. 이런 결과를 막기 위해 확률 분포에 대한 바닥치(floor smoothing)를 설정하고, 확률 분포를 재정규화 한다.

6. 실험 결과 및 고찰

6.1 실험 환경

실험 데이터는 PC 카메라를 이용하여 촬영하였고, 10가지의 동작을 그림 4와 같이 정의하였다. 각 동작에 대하여 7명이 서로 다른 날 각각 7번씩 촬영하여 총 490개의 비디오 데이터를 수집하였다. 초당 30 프레임으로 촬영되었으며, 프레임은 320×240의 크기이고, 24비트 컬러 색상이다. 실험에 사용된 프로그램 코드는 Visual C++ 6.0과 Matlab 7.0으로 작성하였으며, Intel OpenCV 라이브러리와 Bayesian Network Toolbox(BNT)[33]를 이용하였다.

6.2 영상 처리 및 특징 추출

그림 13은 영상 처리의 과정 및 각 단계에서의 결과를 보여주고 있다. 그림 13(a)는 입력 프레임이고, 그림 13(b)는 배경 이미지와의 차이를 계산한 결과를 이진화 한 전경 이미지이다. 그림 13(c)는 YIQ 색상 모델에서 I 요소값이 미리 정의된 임계치 값의 범위를 만족하는 픽셀들을 이용하여 피부색을 검출한 결과를 보여주고 있으며, 그림 13(d)는 Haar-like 얼굴 검출기를 이용하여 검출된 얼굴의 위치를 나타내고 있다. 검출된 얼굴 영역에 있는 픽셀들의 RGB 컬러 색상을 HSV 컬러 색상으로 변환한 뒤, 색상값에 대한 히스토그램의 분포를 나타내고 이를 이용하여 검출된 피부 영역이 그림 13(e)에 나타나 있다. 히스토그램은 사용자 개인에 대한 피부 색상 분포 및 빛 조건의 변화에 따른 피부색 변화를 반영한다. 그림 13(f)는 YIQ와

Haar-like 얼굴 검출기를 각각 적용한 결과들을 결합하여 검출된 피부 영역들을 나타내고 있다. 현재 프레임에서 검출된 피부 영역과 이전 프레임에서 검출된 피부 영역 사이에서 광류를 계산한다(그림 13(g)). 계산된 광류에 대하여 x방향과 y방향에 대하여 각각 정렬을 하고, 각 방향에서의 중간값을 이용하여 각 피부 영역의 분포를 표현하는 가우스(Gaussian) 모델의 평균을 그림 13(h)와 같이 이동하여 현재 프레임에서의 손 및 얼굴에 대한 위치를 예측한다. 예측된 가우스 모델을 이용하여 Argyros 등[27]이 제안한 방법을 따라 그림 13(i)와 같이 손과 얼굴에 해당하는 픽셀들을 분리하고, 분리된 픽셀들로 가우스 모델의 평균 및 공분산 행렬에 대한 값들을 갱신(그림 13(j))하며, 갱신된 값은 다음 프레임에서의 추적에 이용된다.

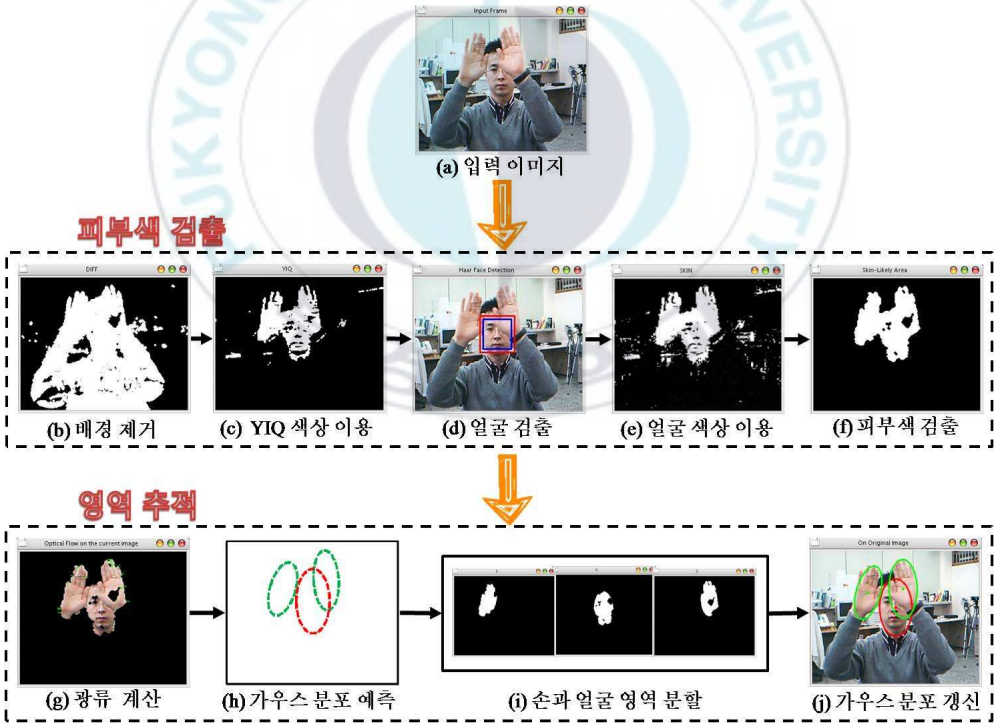


그림 13. 영상 처리 과정 및 단계별 결과

6.3 독립 제스쳐 인식(Isolated Gesture Recognition)

제스쳐 모델의 성능은 교차 검증 방법을 이용하여 평가하였다. 각 제스처에 대해 42개의 데이터로 모델을 훈련시키고, 제외된 나머지 7개의 데이터로 테스트하였다. 총 7번의 교차 검증 실험에 대한 평균값을 인식률로 결정하였으며, 은닉 노드가 가질 수 있는 상태값의 범위는 동작의 복잡성을 고려하여 가변적으로 하였다. 입력 비디오 영상에 대한 손 제스쳐 동작 인식은 식 (5)와 같이 비디오 영상에서 추출한 특징값들을 DBN의 관측 데이터로 하였을 때, 이들 관측 데이터에 대한 우도가 가장 큰 것으로 선택하였다.

$$\hat{\lambda} = \arg\max_{\lambda} \left\{ P(O_{1:T}^{1:5} \mid \theta_{\lambda}) \right\} \tag{5}$$

$$\text{where, } P = \prod_{t} \sum_{X^3} \sum_{X^2} \sum_{X^1} P(x_t^1, x_t^2, x_t^3, o_t^1, o_t^2, o_t^3, o_t^4, o_t^5 \mid x_{t-1}^1, x_{t-1}^2, x_{t-1}^3, \theta)$$

$$= \left\{ \begin{array}{l} \sum_{X^3} P(o_1^5 \mid x_1^3) P(x_1^3 \mid x_1^1, x_1^2) \\ \times \sum_{X^2} P(o_t^3, o_t^4 \mid x_t^2) P(x_1^2) \\ \times \sum_{X^1} P(o_t^1, o_t^2 \mid x_t^1) P(x_1^1) \end{array} \right\} \times \prod_{t=2}^{T} \left\{ \begin{array}{l} \sum_{X^3} P(o_t^5 \mid x_t^3) P(x_t^3 \mid x_{t-1}^3, x_t^1, x_t^2) \\ \times \sum_{X^2} P(o_t^3, o_t^4 \mid x_t^2) P(x_t^2 \mid x_{t-1}^2) \\ \times \sum_{X^1} P(o_t^1, o_t^2 \mid x_t^1) P(x_t^1 \mid x_{t-1}^1) \end{array} \right\}$$

6.3.1 방향 코드만을 이용한 coupled HMM

손의 움직임에 대한 정보를 방향 코드로 양자화하여 각 동작을 coupled HMM으로 모델링하였다. 한 손동작에 대해서는 동작을 수행하지

않는 다른 한 손의 움직임을 고려하지 않기 위해 이에 대한 확률 분포 및 상태 전이 확률은 동일(uniform)하게 지정하여 실험하였다. 그 결과, 한 손 동작에 대해 이미지상에 한 손만 나타나는 경우와 두 손을 이용하는 동작에 대해서는 97.35%의 인식률(그림 14)을 보였다. 그림에서 교차 검증의 2번째 테스트에서 인식률이 다른 것보다 상대적으로 낮은 결과를 보였는데 이는 영상 처리에서 발생한 오차로 인한 것이었다. 특히, 영상에서의 손 영역의 피부색을 검출할 때 빛의 영향으로 손의 일부 영역을 일시적으로 잃어버리면서 손 영역을 모델링하는 가우스 분포의 평균값이 진동하는 결과를 보였다. 평균값의 변화는 방향 코드에 직접적으로 영향을 주기 때문에 이는 원래의 제스쳐가 다른 제스쳐로 잘못 인식되는 결과를 초래하였다.

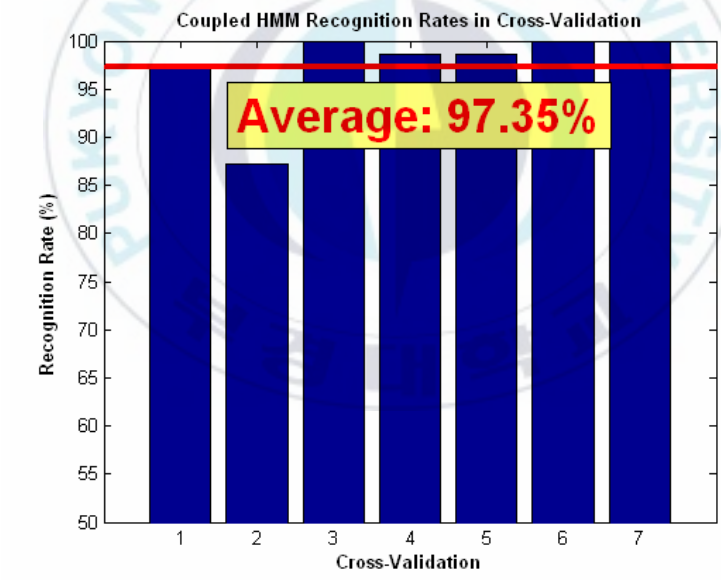


그림 14. 방향 코드를 이용한 coupled HMM 인식률

6.3.2 두 손의 상대적 위치 정보를 추가한 DBN

방향 코드와 두 손의 상대적 위치 정보를 사용한 DBN에 대해서는 평균 98.16%의 성능(그림 15)을 보였다. Coupled HMM을 이용하여 모델링한 실험의 2번째 교차 검증 테스트에서 영상 처리에서의 오차에 영향을 받아 인식률이 떨어진 반면, 두 손의 상대적 위치 정보를 추가한 DBN에서는 손 영역 추적에서의 오차로 발생한 방향 코드의 변화에 대한 에러를 두 손의 상대적 위치가 보완해주어 인식률을 높일 수 있었다.

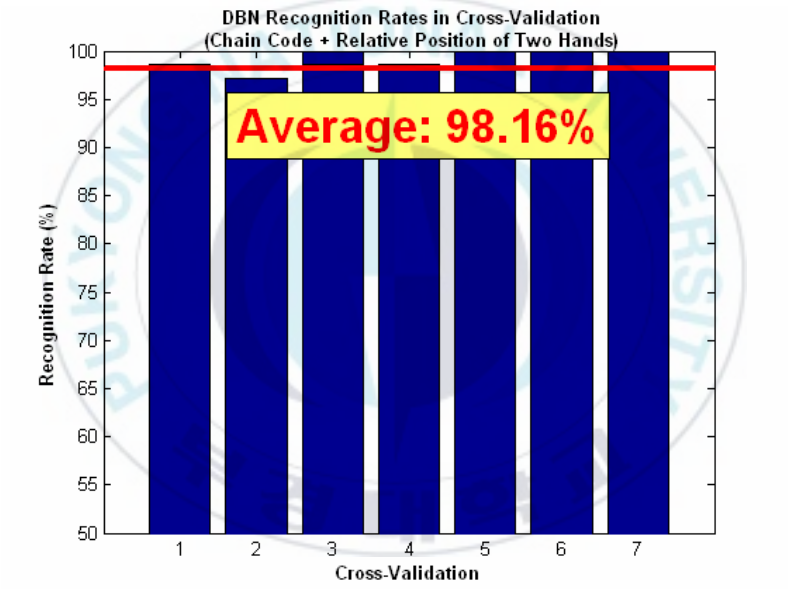


그림 15. 방향 코드와 두 손의 상대적 위치를 이용한 DBN의 인식률

한 손 제스처에 대해 두 손이 모두 이미지 상에 나타난 비디오 영상에서는 coupled HMM과 제안한 DBN의 성능에서 큰 차이를 보였다(표 1). 방향 코드 만을 이용하여 손의 움직임을 표현한 coupled HMM은 6개의 한 손 제스쳐 실험 비디오에 대해 하나만 정확하게 인식을 하였고, 나머지는 다

른 제스쳐로 잘못 인식을 하였다. 이는 방향 코드 만을 사용하였기 때문에 발생한 모호성으로 인한 것이다. 이런 모호성을 없애기 위해, 두 손의 상대적 위치에 대한 정보를 추가한 DBN은 coupled HMM과는 달리, 한 손 제스처에 대해 두 손이 모두 이미지 상에 나타난 경우에도 좋은 인식률을 보였다. 이는 두 손의 상대적 위치에 대한 추가된 정보가 방향 코드 만을 사용하여 인식한 경우에 발행하는 동작의 모호성을 제거할 수 있음을 보여준다.

표 1. 한 손 제스쳐 동작에 두 손이 영상에 나타난 경우의 성능 비교

한손 제스쳐 데이터	**인식 결과**	
	Coupled HMM	**DBN**
빨리 감기	끝 프레임으로 이동(X)	**빨리 감기(O)**
일시 정지	**일시 정지(O)**	**일시 정지(O)**
10초 뒤로 이동	일시 정지(X)	**10초 뒤로 이동(O)**
10초 뒤로 이동	끝 프레임으로 이동(X)	**10초 뒤로 이동(O)**
뒤로 감기	첫 프레임으로 이동(X)	**뒤로 감기(O)**
10초 앞으로 이동	끝 프레임으로 이동(X)	**10초 앞으로 이동(O)**
맞힌 수	**1**	**6**

6.3.3 얼굴과 각 손의 상대적 위치 정보를 추가한 DBN

두 손의 상대적 위치 정보를 포함한 DBN에서 인식에 실패한 경우들을 분석을 한 결과 '종료', '끝 프레임으로 이동', '첫 프레임으로 이동' 동작 간의 혼동으로 인한 것이 대부분이었다. 이는 영상 처리에서의 오차로 인

한 것도 있었고, 고정되어 있어야 하는 손을 사용자가 무의식적으로 조금씩 움직여서 그런 것도 있었다. 사용자의 무의식적인 움직임이 인식 결과에 영향을 미치는 것을 줄이기 위한 방안으로 얼굴과 각 손과의 상대적 위치 정보를 추가하였다. 최종적인 DBN을 그림 10과 같이 정의하고 실험한 결과 표 2와 같이 99.59%로 성능을 높일 수 있었다

표 2. 손 제스쳐 동작 인식 성능

동작	테스트 수	맞힌 수	틀린 수	인식률(%)
열기	49	49	0	100
종료	49	49	0	100
시작	49	49	0	100
일시 정지	49	49	0	100
첫 프레임으로 이동	49	48	1	97.59
끝 프레임으로 이동	49	48	1	97.59
10초 앞으로 이동	49	49	0	100
10초 뒤로 이동	49	49	0	100
빨리 감기	49	49	0	100
뒤로 감기	49	49	0	100
평균 인식률	490	488	2	99.59

6.3.4 은닉 노드 상태 디코딩(Hidden State Decoding)

관측 데이터가 주어졌을 때, 이들 관측 데이터에 대해 제안된 모델이 각 제스쳐를 잘 모델링하고 있는지를 확인하기 위해 은닉 노드들의 최적의 상태 시퀀스를 디코딩해 보았다. 최적의 상태 시퀀스는 관측 데이터 $O_{1:T}^{1:5}$에 대해 데이터를 모델링하고 있는 DBN에서 시간 1부터 시간 T까지의 상태 노드들의 모든 경로들 중에서 최대의 우도를 출력하는 하나의 상태 노드들의 경로를 의미한다. 최대의 우도를 출력하는 상태 노드들의 시퀀스를 결정하기 위해서는 접합 트리 알고리즘에서 상태의 전이를 계산할 때 시간 $t-1$에서의 상태들로부터 시간 t의 상태로의 전이를 계산할 때, 모든 상태로부터의 전이를 고려하는 것이 아니라, 하나의 최대 전이값을 선택하여 계산한다. 이는 '최적화의 원칙'에 기반한 동적 프로그래밍 기법으로 HMM에서의 Viterbi 알고리즘과 동일한 방식이다. 최적의 상태 시퀀스를 결정하는 것은 6.4절의 연속 제스쳐 인식에서 특정 제스쳐의 존재 여부를 결정하는 데에서도 이용된다.

'실행' 제스쳐에 대한 디코딩 결과가 표 3에 나타나 있다. 표 3은 각 손의 움직임을 모델링하는 두 노드 X^1, X^2의 방향 코드에 대한 은닉 노드의 상태 변화를 디코딩한 결과이며, 그림 16은 은닉 노드가 방향 코드의 공간을 어떻게 분할하고 있는지를 보여주고 있다. 그림 16에 나타난 바와 같이 은닉 노드의 각 상태는 손의 움직임을 왼쪽, 왼쪽 위, 위, 오른쪽 위 방향의 4개 공간을 분할하여 표현하고 있다. 주어진 관측 데이터에 대해 각 손의 움직임을 모델링하는 은닉 노드들이 관측 데이터를 대체로 균일하게 분할하면서 상태 전이를 하고 있는 것으로 보아 훈련된 모델이 제스쳐를 잘 모델링하고 있다고 할 수 있다.

표 3. 실행 제스쳐의 방향 코드(O^1, O^3)와 은닉 노드(X^1, X^2)의 상태 디코딩

O^1	11	10	10	9	0	8	8	7	7	6	6	6	0	6	6	6	5	6	5	5	5	4	0	4	3	3	3	2	2	3	2	2	2	2	1	0
O^3	16	16	1	1	0	2	2	3	3	3	4	4	0	4	4	5	5	5	5	6	6	6	0	6	7	7	7	7	7	8	8	8	9	8	8	8
X^1	3	3	3	3	3	3	3	3	3	2	2	2	2	2	2	2	1	1	1	1	1	1	1	1	4	4	4	4	4	4	4	4	4	4	4	4
X^2	4	4	4	4	4	4	4	4	4	2	2	2	2	2	2	2	2	1	1	1	1	1	1	1	3	3	3	3	3	3	3	3	3	3	3	3

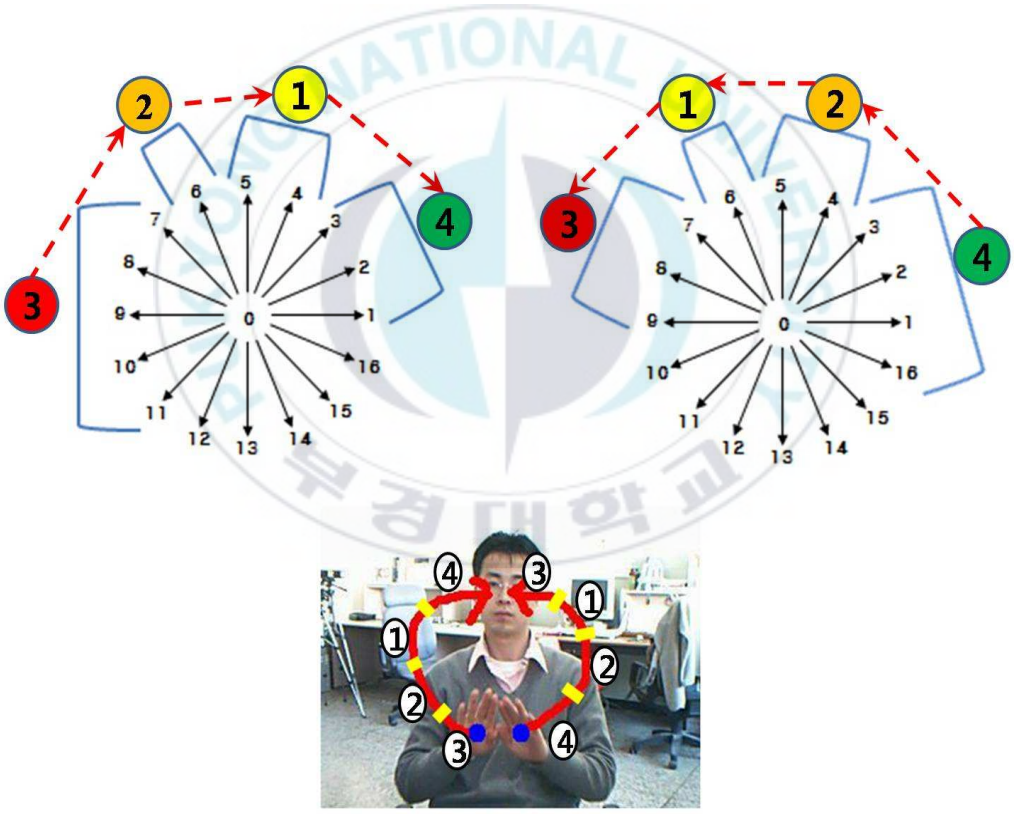


그림 16. 실행 제스쳐에 대한 은닉 노드 $\boldsymbol{X}^1$, $\boldsymbol{X}^2$의 코드 분할 및 상태 전이

6.4 연속 제스쳐 인식(Continuous Gesture Recognition)

연속 제스쳐 인식은 하나 또는 그 이상의 제스쳐를 포함하고 있는 입력 비디오 시퀀스에서 미리 정의된 의미 있는 제스쳐를 검출하고, 검출된 제스쳐가 어떤 제스쳐인지를 인식하는 문제이다. 연속 제스쳐 인식을 위해 두 가지 방법이 사용된다. 첫 번째 방법은 제스쳐의 평균 동작 길이만큼의 윈도우의 크기를 미리 결정해 놓고, 이 윈도우를 일정 프레임 간격으로 이동하면서 각 제스쳐 모델에 대한 우도를 계산하여 우도가 임계치보다 클 때 최대 우도를 출력하는 모델로 제스쳐를 인식하는 방법이다[36]. 그러나, 이 방법은 고정된 크기의 윈도우를 사용하기 때문에 제스쳐를 수행하는 사람들의 각기 다른 동작 속도를 수용하지 못하는 단점이 있다. 두 번째는 유한 상태 네트워크(finite state network: FSN)와 동적 프로그래밍(dynamic programming: DP) 기법을 이용하는 방법이다[37]. FSN은 유한개의 네트워크 상태 노드와 각 상태 노드를 연결하는 화살표를 이용하여 이미 알고 있는 지식 정보를 표현하는 방법이며, DP는 Bellman[38]의 '최적화의 원칙(principle of optimality)'을 기본 개념으로 하는 검색 방법이다. 본 논문에서는 두 번째 방법을 이용하여 연속 제스쳐 인식에 대한 문제를 해결한다.

6.4.1 연속 제스쳐 인식을 위한 네트워크 구조

하나의 제스쳐를 취하기 위해서는 그 제스쳐를 취하기 위한 시작 자세로의 움직임이 필요하며, 한 제스쳐의 마지막과 다음 제스쳐의 시작 부분 사이에서도 두 제스쳐를 연결하기 위한 움직임이 필요하다. 이러한 중간

과정의 의미없는 제스쳐를 ‘필러(filler)’라 한다. 그림 17은 연속 제스쳐에 대한 이런 지식 정보를 FSN으로 표현한 것이다. 그림에서 ‘S(Start)’, ‘F(Final)’, ‘N(Null)’을 ‘네트워크 상태 노드’라 한다. ‘S’는 네트워크 순회를 위한 시작 상태 노드를 의미하고, 두 개의 원이 겹쳐져 있는 ‘F’는 네트워크 순회의 마지막 상태 노드를 의미한다. 그리고, 두 개의 네트워크 상태 노드 사이에 연결되어 있는 사각형 내부의 모델들은 5장에서 제안한 양손 제스쳐 모델(DBN)들이다. ‘S’와 ‘F’ 사이에 연결되어 있는 ‘Filler’는 새로운 제스쳐의 시작을 위한 준비 과정 또는 임의의 두 제스쳐를 연결하는 의미 없는 제스쳐에 대한 DBN이며, ‘F’와 ‘N’ 사이에 연결되어 있는 ‘G1~G10’은 정의된 10가지의 제스쳐를 각각 모델링하고 있는 DBN들이다. ‘N’에서 ‘S’으로의 점선으로 된 화살표는 널 전이를 표현한 것으로 아무런 계산 없이 다음 노드로의 전이를 표현한다. 이렇게 함으로 해서 순환 네트워크 모델이 형성되게 되고, 이는 임의의 개수만큼의 제스쳐 동작에 대한 검출 및 인식이 가능하도록 한다.

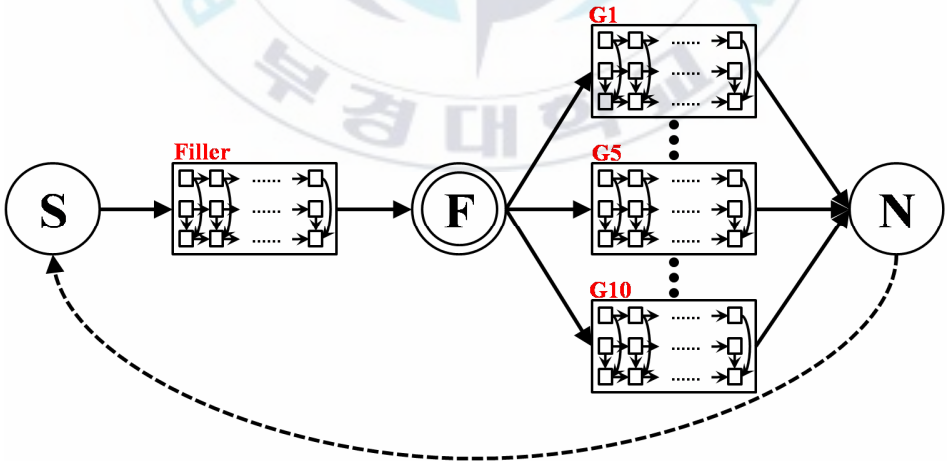


그림 17. 연속 제스쳐 인식을 위한 네트워크 모델

6.4.2 연속 제스쳐 인식 알고리즘

연속 제스쳐 인식은 주어진 입력 비디오의 모든 프레임에서의 관측값에 대해 최대의 확률값을 출력하는 제스쳐의 시퀀스를 결정하는 문제이다. 이때, 모든 프레임에서의 관측값에 대한 최대 확률값을 출력하기 위해서는 t 시간에서의 관측값에 대해 각 제스쳐를 모델링하고 있는 DBN에서의 출력 확률 또한 최대가 되어야 한다. 이는 최적화 문제로 정의될 수 있으며, DP를 이용하여 해결할 수 있다. 여기서, 각 제스쳐를 모델링하는 DBN 내부에서의 DP를 '지역(local) DP'라 하고, 연속 제스쳐에 대한 지식 정보를 표현하고 있는 FSN에서의 DP를 '전역(global) DP' 라고 한다.

입력 비디오에 대한 연속 제스쳐 인식 알고리즘이 그림 18에 나타나 있다. 알고리즘에서 $nl \rightarrow nr$은 FSN을 순회할 때, 왼쪽의 네트워크 상태 노드(left network state node: nl)에서 오른쪽의 네트워크 상태 노드(right network state node: nr)로의 전이를 나타내며, 이때 거쳐가는 제스쳐 모델들은 m으로 표현된다. 시간 t에서의 관측값에 대해 지역 DP와 전역 DP를 각각 수행한다.

지역 DP는 각각의 제스쳐 모델에 대해서 t 시간에 은닉 상태 ($X^1=i$, $X^2=j$, $X^3=k$)에 머무를 최대 확률을 계산한다. $\delta_t^m(i,j,k)$ 를 제스쳐 모델 m에 대해 t 시간에서 상태 (i, j, k)의 최대 출력 확률값을 저장하는 변수, $\psi_t^m(i,j,k)$ 를 그때의 조건 정보(제스쳐 모델 내부에서의 상태 전이를 한 것인지, 네트워크 노드 전이를 한 것인지에 대한 정보)를 저장하는 변수, $\varphi_t^m(i,j,k)$ 를 제스쳐 모델 m에 얼마나 오랫동안 머물러 있었는가를 저장하는 변수로 각각 정의한다. 최대 확률값을 출력하는 후보는 크게 두 가지 경우가 있다. 첫째는 시간 $t-1$에서도 동일 제스쳐 모델에 머물러 있다가 현

재 상태 (i, j, k)로 모델 내부에서의 상태 전이 $(a, b, c) \rightarrow (i, j, k)$를 하는 경우 ($\delta_{t-1}^{m}(a,b,c)A_{(ia,\,jb,kc)}^{m}$)이고, 두 번째는 시간 t-1에는 다른 제스쳐 모델에 머물러 있다가 현재의 제스쳐 모델로 새롭게 진입하는 네트워크 모델에서의 전이이다. 다시 말하면, FSN 네트워크의 상태 노드간 전이를 통하여 새로운 DBN 모델로의 진입에 의한 초기 상태 확률이 최대가 되는 경우 ($\Delta_{t-1}\pi_{(i,j,k)}^{m}$)이다.

전역 DP는 각 제스쳐 모델을 통과한 뒤, 다음 제스쳐 모델로의 이동에 대한 정보를 저장하는 과정이다. $\Delta_t(nr)$는 nl에서 nr로의 전이 과정에 연결되어 있는 여러 제스쳐 모델 중에서 네트워크 노드 전이에 대한 최대 확률값($\max_m \delta_t^m(E_m)$)을 저장한다. 이때의 최대 출력값에 대한 후보들은 각 제스쳐 모델의 마지막 상태 노드 E_m에서의 확률값들이다. $\Psi_t(nr)$은 어떤 제스쳐 모델 m이 $\Delta_t(nr)$에 최대 확률값을 제공하는가를 저장한다.

입력 비디오 시퀀스에 대한 마지막 프레임까지의 계산을 마친 뒤, 매 시간마다 수행되었던 지역 DP 및 전역 DP의 결과들을 이용하여 역추적을 수행하면 어떠한 제스쳐들이 포함되어 있으며, 각 제스쳐의 시작 위치와 마지막 위치를 찾아낼 수 있다. 입력 비디오 시퀀스의 마지막 프레임에서의 출력 확률값이 최대가 되는 DBN 모델을 FSN 순환에 대한 제약 조건을 따라 마지막 상태 노드 'F'에서 선택하고, 그 모델의 최대 확률에 대한 시작 지점을 그림 18에 나타나 있는 알고리즘에서의 지역 DP에서 정의된 변수 $\varphi_t^m(i,j,k)$에 저장된 정보를 이용하여 계산한다. 그리고, 이 선택된 DBN의 시작 지점으로 진입할 때의 정보 즉, 어떤 DBN에서 현재의 DBN으로 전이가 일어났는지에 대한 정보를 전역 DP에서 정의된 변수 $\Delta_t(nr)$를 이용하여 획득한다. 이들 과정을 반복하면 입력 비디오 시퀀스의 관측값들에 대한 최대 확률을 출력하는 연속 제스쳐 시퀀스를 검출할 수 있다.

function Network_Model_DP

for $t = 1:T$

for each transition($nl \rightarrow nr$) in network model

for each model in between $nl \rightarrow nr$

// local DP

for *states*($i,\ j,\ k$)

$$\psi_t^m(i,j,k) = \arg\max_{(\hat{a},\hat{b},\hat{c})} \left\{ \delta_{t-1}^m(a,b,c) A_{(ia,jb,kc)}^m, \Delta_{t-1}\pi_{(i,j,k)}^m \right\} \times B_{(i,j,k)}^m(O_t^{1:5})$$

$$\delta_t^m(i,j,k) = \max_{(\hat{a},\hat{b},\hat{c})} \left\{ \delta_{t-1}^m(a,b,c) A_{(ia,jb,kc)}^m, \Delta_{t-1}\pi_{(i,j,k)}^m \right\} \times B_{(i,j,k)}^m(O_t^{1:5})$$

$$\varphi_t^m(i,j,k) = \begin{cases} 1 & \text{if, } \Delta_{t-1}\pi_{(i,j,k)}^m \text{ is the maximum} \\ \varphi_{t-1}^m(\hat{a},\hat{b},\hat{c}) + 1 & \text{otherwise} \end{cases}$$

end

end

// global DP

$$\Delta_t(nr) = \max_m \delta_t^m(E_m)$$

$$\Psi_t(nr) = \arg\max_m \delta_t^m(E_m)$$

end

end

end

그림 18. 네트워크 모델 DP 알고리즘

그림 19는 하나의 입력 비디오에 대한 6개의 제스쳐 모델들의 정규화된 확률값 변화를 보여주고 있다. 그림의 복잡성을 낮추기 위해 일부 제스쳐 모델에 대한 확률값들만 나타내었다. 정규화된 확률값이란 식(6)과 같이 해당 모델 고유의 확률값의 계산을 위해 사용된 프레임의 수로 나눈 값을 의미한다.

$$\text{normalized model likelihood} = \frac{\delta_t^m(E_m) - \Delta_{t-\varphi_t^m(E_m)}}{\varphi_t^m(E_m)} \quad (6)$$

그림에서 상단에 위치한 숫자 '5', '9', '8'은 입력 비디오 시퀀스에 이들 제스쳐가 포함되어 있다는 것을 나타내고 있으며, 역추적을 통하여 검출된 각 제스쳐의 시작 위치와 마지막 위치는 수직 점선으로 표현되었다. 입력 비디오의 프레임 시퀀스에서 특정 제스쳐가 포함되어 있는 부분에서는 해당 제스쳐의 확률값이 다른 제스쳐 모델에서의 확률값보다 높게 나타남을 확인 할 수 있다.

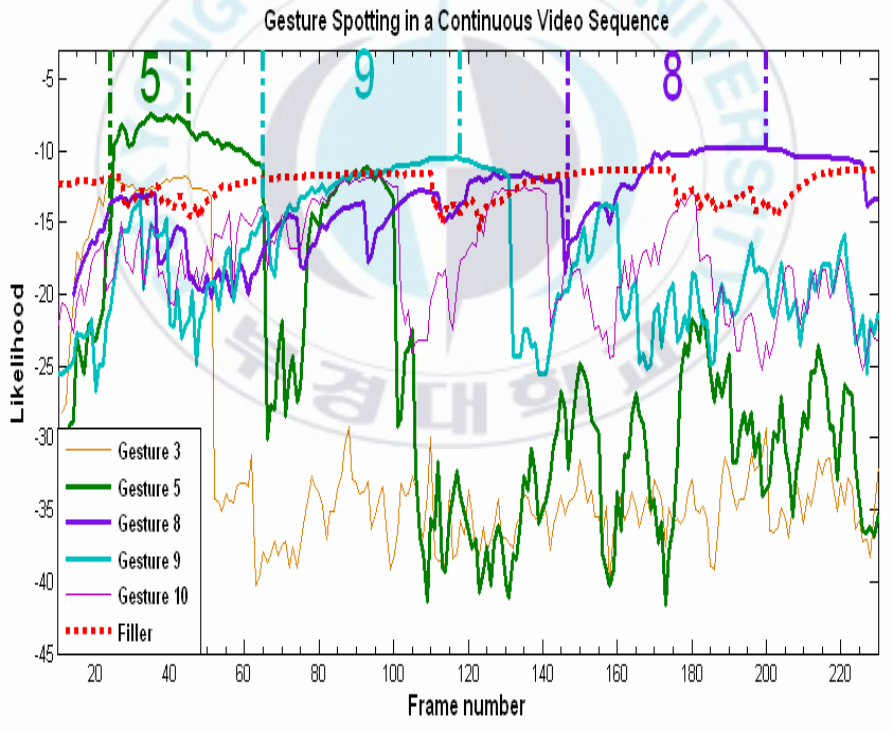


그림 19. 비디오 시퀀스에 대한 각 모델의 우도 변화

6.4.3 연속 제스쳐 인식 성능

연속 제스쳐 인식에서는 3가지 타입의 에러가 있다. 첫째, 존재하지 않는 제스쳐가 존재한다고 결정하는 삽입(insertion) 에러, 둘째, 입력 비디오 내에 제스쳐가 존재함에도 존재하지 않는다고 결정을 내리는 삭제(deletion) 에러, 마지막으로 제스쳐가 존재하며 그 위치도 정확히 찾았으나 다른 제스쳐로 잘 못 인식하는 대체(substitution) 에러이다. 각각 다른 수의 제스쳐를 포함하고 있는 8개의 입력 비디오에 대한 인식 결과가 표 4에 나타나 있다. 표에서 인식률(detection)은 식 (7)과 같이 입력된 제스쳐와 정확하게 인식한 제스쳐의 수에 대한 비로 계산된다. 인식된 제스쳐들의 시작 위치와 마지막 위치를 손으로 직접 결정한 부분들과 비교했을 때, 오차의 평균과 표준 편차는 각각 0.09프레임, 5.51 프레임을 나타났다. 즉, 대체로 정확한 위치를 찾았다고 할 수 있다.

$$\text{Detection}(\%)=\frac{\text{\# of correctly recognized gestures}}{\text{\# of input gestures}} \tag{7}$$

인식률 계산에서는 삽입 에러는 고려하지 않았지만, 존재하지 않는 제스쳐의 삽입으로 인해 원래 존재하는 제스쳐에 대한 인식 성능에 영향을 미칠 수도 있다. 이런 점을 감안하여 삽입 에러까지 함께 고려한 성능 평가 척도인 신뢰도(reliability)는 식 (8)과 같이 계산된다.

$$\text{Reliability}(\%)=\frac{\text{\# of correctly recognized gestures}}{\text{\# of input gestures} + \text{\# of insertion errors}} \tag{8}$$

표 4. 연속 제스쳐 인식 성능

입력 제스쳐 수	인식 결과					
	Hit	Substitution	Insertion	Deletion	Detection (%)	Reliability (%)
50	42	5	5	3	84	76.36

7. 결론 및 향후 과제

컴퓨터의 성능 및 정보 표현에 대한 기술 발달로 인해 컴퓨터와의 상호작용을 위한 새로운 방법들이 요구되고 있다. 손은 신체의 다른 어떤 부분보다도 움직임이 자유로우며, 많은 것을 표현할 수 있으므로 컴퓨터와의 상호작용을 위해 손을 이용하는 것은 적절한 방법이 된다. 본 논문에서는 미디어 플레이어를 제어하기 위한 10가지의 손 동작을 정의하고, 인식하는 방법을 제안하였다.

영상 처리 단계에서 손의 정확한 검출을 위해 얼굴 검출을 통해 생성한 피부색 모델과 YIQ 색상 모델을 결합한 방법을 적용하였다. 손과 얼굴 영역의 추적을 위해서는 Argyros 등[27]의 방법을 이용하되, 이전 프레임에서의 피부 영역과 현재 프레임에서의 피부 영역들간의 광류(optical flow)를 계산하고, 이를 현재 프레임에서의 손의 위치를 예측하는데 적용하여 손의 비선형적 또는 비연속적 움직임에 대해 더욱 강건한 추적을 할 수 있었다.

손 동작의 모델링을 위해서 손의 움직임을 표현하는 방향 코드, 두 손의 상대적 위치 및 얼굴과 손의 상태적 위치를 특징으로 사용하였다. 두 손의 상대적 위치를 특징으로 사용함으로써 한 손 동작을 하는 과정에 다른 한 손이 영상 내에 존재할 경우, 두 손 동작으로 잘못 인식하는 것을 막을 수 있었으며, 얼굴과 손의 상대적 위치는 사용자가 무의식 중에 고정되어 있어야 하는 손을 움직여서 다른 동작으로 인식되는 것을 방지하였다. 손 동작 인식 모델은 은닉 마르코프 모델(Hidden Markov Model: HMM)이나 coupled HMM의 일반적 형태인 동적 베이스망(Dynamic Bayesian Network: DBN)을 이용하였으며, 위의 특징들을 관측값으로 하여 훈련 및 인식을 수행하였다.

제안된 DBN에 대한 독립 제스쳐 인식(isolated gesture recognition) 성능은 490개의 실험 비디오 데이터로 교차 검증 방법을 이용하여 평가하였다. 각 동작에 대한 49개의 데이터 중 42개의 데이터로 훈련을 하고, 제외된 나머지 7개의 데이터로 테스트를 하여 평균 99.59%의 인식률을 얻었다. 그리고, 연속 제스쳐(continuous gesture)에 대한 제스쳐 검출 및 인식은 논문에서 정의된 각 제스쳐에 대한 DBN과 제스쳐와 제스쳐 사이의 움직임을 모델링하는 필러(filler) DBN으로 구성된 유한 상태 네트워크(finite state network: FSN)로 연속 제스쳐 인식 네트워크를 정의하였다. 제스쳐의 검출 및 인식을 위해 두 단계의 동적 프로그래밍(dynamic programming) 기법을 적용하여 84%의 인식률과 76.36%의 신뢰도를 얻었다.

본 논문에서는 결합 확률 분포 계산을 위해 접합(junction) 트리 알고리즘을 사용하여 정확한 값을 이용한 추론을 하였다. 이에 따라 추론을 하는데 있어 많은 시간이 소요된다. Variational 방법[34]이나 샘플링 방법[35]을 이용하는 근사적 추론 방법을 이용하게 되면 계산 시간을 줄일 수 있다. 또한, 본 논문에서는 상대적으로 간단한 동작들을 정의하고 인식하였는데, 제안된 DBN 모델을 확장하여 수화와 같은 복잡한 손 동작 인식에 적용할 수 있을 것이다.

참고 문헌

[1] L. Rabiner, *Fundamentals of Speech Recognition*, Prentice-Hall, 1993.

[2] G. Johansson, "Visual perception of biological motion and a model for its analysis," *Perception and Psychophysics*, vol. 14, pp. 201-211, 1973.

[3] J. K. Arggarwal and Q. Cai, "Human motion analysis - a review," *Computer Vision and Image Understanding*, vol. 73, pp. 428-440, 1999.

[4] V. I. Pavlovic, R. Sharma, and T. S. Huang, "Visual Interpretation of Hand Gestures for Human-Computer Interaction A Review," *IEEE Transaction on Pattern Analysis and Machine Intelligence*, vol. 19, pp. 677-695, 1997.

[5] D. Xu, "A Neural Network Approach for Hand Gesture Recognition in Virtual Reality Driving Training System of SPG," In *Proceedings of IEEE International Conference on Pattern Recognition*, vol. 3, pp. 519-522, 2006.

[6] B. Domer, “Chasing the colour glove: visual hand tracking,” *Technique Report*, Department of Computer Science, Simon Fraser University, 1994.

[7] H. K. Lee and J. H. Kim, "An HMM-Based Threshold Model Approach for Gesture Recognition," *IEEE Transaction on Pattern Analysis and Machine Intelligence*, vol. 21, pp. 961-973, 1999.

[8] H. H. Avilés-Arriaga, L. E. Sucar, and C. E. Mendoza, “Visual Recognition of Similar Gestures,” In *Proceedings of IEEE International Conference on Pattern Recognition*, vol. 1, pp. 1100-1103, 2006.

[9] M. H. Yang and N. Ahuja, “Recognizing Hand Gestures Using Motion Trajectories,” In *Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition*, vol. 1, pp. 23-25, 1999.

[10] S. F. Wong and R. Cipolla, “Continuous Gesture Recognition Using a Sparse Bayesian Classifier,” In *Proceedings of IEEE International Conference on Pattern Recognition*, vol. 1, pp. 1084-1087, 2006.

[11] D. A. Forsyth and J. Ponce, *Computer Vision: A Modern Approach*, Chapter 19, pp. 520-573, Prentice Hall, 2003.

[12] A. Blake and M. Isard, "Condensation: Conditional density propagation for visual tracking," *International Journal of Computer Vision*, pp. 5-28, 1998.

[13] C. S. Myers and L. R. Rabiner. “A comparative study of several dynamic time-warping algorithms for connected word recognition,” *The Bell System Technical Journal*, vol. 60, pp. 1389-1409, 1981.

[14] J. Yamato, J. Ohya, and K. Ishii, “Recognizing human action in time-sequential images using Hidden Markov Model,” In *Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition*, pp. 379-385, 1992.

[15] H. I. Suk and B. K. Sin, “HMM-Based Gait Recognition with Human Profiles”, In *Proceedings of Joint IAPR International Workshops SSPR 2006 and SPR2006*, Hong Kong, China, pp. 596-603, 2006.

[16] M. Brand, N. Oliver, and A. Pentland, “Coupled hidden Markov models for complex action recognition,” In *Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition*, pp. 994-999, 1997.

[17] F. V. Jensen, *Bayesian Networks and Decision Graphs*, Chapter 1, pp. 3-34, Springer, 2001.

[18] Y. Du, F. Chen, W. Xu, and Y. Li, “Recognizing Interaction Activities using Dynamic Bayesian Network,” In *Proceedings of IEEE International Conference on Pattern Recognition*, pp. 618-621, 2006.

[19] R. León, “Continuous Activity Recognition with Missing Data,” In *Proceedings of IEEE International Conference on Pattern Recognition*, vol. 1, pp. 439-446, 2002.

[20] M. H. Yang and N. Ahuja, “Recognizing Hand Gestures Using Motion Trajectories,” In *Proceedings of IEEE Computer Vision and Pattern Recognition*, vol. 1, pp. 23-25, 1999.

[21] A. V. Nefian, L. Liang, X. Pi, X. Liu, and K. Murphy, “Dynamic Bayesian

Networks for Audio-Visual Speech Recognition," *Journal of Applied Signal Processing*, vol. 11, pp. 1-15, 2002.

[22] Z. Ghahramani and M. I. Jordan, "Factorial hidden Markov models," *Machine Learning*, vol. 29, pp. 245-273, 1997.

[23] K. P. Murphy, *Dynamic Bayesian Network: Representation, Inference and Learning*, Ph.D. Dissertation, University of California, Berkeley, 2002.

[24] P. Kakumanu, S. Makrogiannis, and N. Bourbakis, "A Survey on Pixel-Based Skin Color Detection Techniques," *Pattern Recognition*, vol. 40, pp. 1106-1122, 2007.

[25] P. Viola and M. J. Jones, "Robust Real-Time Face Detection," *International Journal of Computer Vision*, vol. 57, pp. 137-154, 2004.

[26] G. R. Bradski, "Computer vision face tracking for use in a perceptual user interface," *Intel Technology Journal* Q2, pp. 1–15., 1998.

[27] A. A. Argyros and M. I.A. Lourakis, "Real-time tracking of multiple skin-colored objects with a possibly moving camera," In *Proceedings of European Conference on Computer Vision*, vol. 3, pp. 368-379", 2004.

[28] L. R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, pp. 257-285, 1989.

[29] R. G. Cowell, A. P. Dawid, S. L. Lauritzen, and D. J. Spiegelhalter, *Probabilistic Networks and Expert Systems*, Springer, 1999.

[30] C. M. Bishop, *Pattern Recognition and Machine Learning*, Chapter 8, pp. 359-422, Springer, 2007.

[31] C. Huang and A. Darwiche, "Inference in Belief Networks: A Procedural Guide," *International Journal of Approximate Reasoning*, vol.15, pp. 225-263, 1994.

[32] Dempster, A. P., N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the Royal Statistical Society*, vol. 39, pp. 1-38, 1977.

[33] http://bnt.sourceforge.net/

[34] T. Jaakkola, *Variational Methods for Inference and Estimation in Graphical Models*, Ph.D. Dissertation, Massachusetts Institute of Technology, 1997.

[35] W. R. Gilks, S. Richardson, and D. J. Spiegelhalter, *Markov Chain Monte Carlo in Practice*, Chapman and Hall, 1996.

[36] H. Kang, C. W. Lee, and K. Jung, "Recognition-based gesture spotting in video games," *Pattern Recognition Letters*, vol. 25, pp. 1701-1714, 2004.

[37] B. K. Sin, *An HMM-Based Statistical Approach For Modeling On-line Cursive Script*, Ph. D. Dissertation, Korea Advanced Institute of Science and Technology, 1994.

[38] R. Bellman, *Dynamic Programming*, Princeton University Press, Princeton, 1957.

감사의 글

누구 보다 먼저 감사의 마음을 전하고 싶은 분이 있습니다. 항상 바쁘신 와중에도 따로 시간을 내어주시면서 개인 강의도 해주시고, 저의 질문에 항상 성심성의껏 답변해 주시면서 학문적으로 정말 많은 가르침을 주시면서 석사 과정을 마칠 수 있도록 도와주신 분입니다. 학업적 가르침뿐만 아니라, 테니스도 무료로 가르쳐 주셨고, 졸업 후 진로도 걱정해주시고 챙겨주신 지도 교수님이신 신봉기 교수님 감사 드립니다. 그리고, 미국 생활 동안 혼자 지내는 저를 위해 맛있는 반찬과 음식들을 제공해주신 사모님께도 고마움을 전하고 싶습니다.

부족한 저의 논문을 심사 및 지도·조언을 해 주신 권기룡 교수님과 김종남 교수님께도 감사의 마음을 전합니다. 지금은 졸업하고 열심히 직장 생활을 하고 있는 함께 연구실 생활을 했었던 동기들 윤희, 효연, 윤성, 대종, 병문, 헌호, 그리고 대학원 생활 동안 많은 조언과 도움을 주신 창수 선배, 길호 선배, 주현 선배, 윤도 선배, 실험 비디오 촬영을 도와준 현준씨, 병제, 태윤이, 기윤이 모두들 고맙습니다.

사실 마지막이 아니라 가장 먼저 감사의 마음을 전했어야 할 우리 가족들... 공부한답시고 집안에 아무런 도움도 되지 못한 아들에게 물심양면으로 챙겨주시고 홀로 많은 힘든 일들을 처리하시는 저의 어머니께 죄송스럽고, 고맙고, 사랑한다는 마음을 전합니다. 누나 진숙이, 동생 진미, 최근 한 가족이 되어 누나, 동생과 행복한 가정을 꾸리고 있는 매형과 매제, 그리고 힘들어 좌절할 때마다 한결같이 곁에서 응원해주고 힘을 북돋워준 여자 친구 영주에게도 고맙다는 말을 전합니다.

모두가 행복했으면 좋겠습니다.

2007년 7월

석 흥 일 드림