

<http://dx.doi.org/10.7236/JIWIT.2012.12.2.119>

JIWIT 2012-2-15

## 깊이 맵과 HMM을 이용한 인식 시스템 구현

### Implementation of a 3D Recognition applying Depth map and HMM

한창호\*, 오춘석\*\*

Chang-Ho Han, Choon-Suk Oh

**요약** 최근 연구에서 모션 인식을 위해 여러 가지 인식 알고리즘을 사용하였다. 예를 들면, HMM, DTW, PCA 등의 기법을 이용하여 권투 모션을 인식하는 방법을 제시했다. 이러한 방법을 이용하기 위해서 연기자로부터 3차원 데이터를 얻기 위해 액티브 마커를 사용하여 손의 위치를 얻는다. 얻은 2차원 위치 정보는 다시 스테레오 기법을 이용하여 3차원 정보로 전환하여 구한다. 본 논문에서는 3차원 모션 데이터를 얻는 방법을 깊이 맵에 대한 알고리즘을 이용하여 구하였다. 그리고 3차원 위치 데이터 정보의 정확성 나타냈으며, 그리고 모션 동작에 대한 인식을 실험을 하였고, 그 실험 결과에 대해서 언급했다.

**Abstract** Recently, we used to recognize for human motions with some recognition algorithms. examples, HMM, DTW, PCA etc. In many human motions, we concentrated our research on recognizing fighting motions. In previous work, to obtain the fighting motion data, we used motion capture system which is developed with some active markers and infrared rays cameras and 3 dimension information converting algorithms by the stereo matching method. In this paper, we describe that the different method to acquiring 3 dimension fighting motion data and a HMM algorithm to recognize the data. One of the obtaining 3d data we used is depth map algorithm which is calculated by a stereo method. We test the 3d acquiring and the motion recognition system, and show the results of accuracy and performance results.

**Key Words :** HMM, Disparity, Depth map, Stereo Vision, Recognition

#### 1. 서론

모션 인식에 대한 연구는 최근 로봇관련 분야에서 활발한 연구개발<sup>[1],[2]</sup>이 진행되고 있다. 일반 개인 컴퓨터에서 구현하는 단계에서 로봇에 적용해 인간에 대한 서비스를 향상시키는 노력이 다방면에서 이루어지고 있다.

일본과 유럽의 경우 정부 주도로 로봇 소프트웨어 개

발을 했고 추가적으로 확장하여 수행하고 있다. 미국은 특수 목적의 로봇 시장을 목표로 ERSP 및 MSRDS 등 다양한 로봇 소프트웨어 플랫폼을 개발해 시장에 제공하고 있다. 미국의 에볼루션로보틱스(ER)사에 의해 지난 2004년 개발된 ERSP는 그동안 고도의 기술력이 필요했던 비전 기반의 모바일 로봇 제어를 간단한 프로그램 작업으로 구현해 지능형 로봇 개발에 널리 사용되고 있다.

\*준회원, 선문대학교 정보통신공학과

\*\*정회원, 선문대학교 정보통신공학과(교신저자)

접수일자 2012년 2월 21일, 수정완료 2012년 3월 22일

게재확정일자 2012년 4월 13일

Received: 21 February 2012 / Revised: 22 March 2012 /

Accepted: 13 April 2012

\*\*Corresponding Author: csoh@sunmoon.ac.kr

Dept. of Information and Communication Engineering, SunMoon University, Korea

국내에서 로봇 소프트웨어 개발은 지식경제부와 OPRoS (Open Platform for Robotic Services)<sup>[3],[4]</sup>라는 로봇 소프트웨어 플랫폼 개발에 이미 많은 연구원들이 참여하여 성과를 내고 있다.

따라서 기존에 OPRoS용으로 개발된 인식 컴포넌트의 알고리즘<sup>[5]</sup>을 활용하여 마커를 사용하지 않는 마커레스(marker-less) 방법의 3차원 위치 정보를 얻는 방법을 고안하였고, 이를 바탕으로 어떤 모션을 행하였는지를 인식할 수 있도록 하였다. 모션 인식을 위한 알고리즘은 HMM 방법을 이용하였다.

실험 결과에는 마커레스 방법으로 얻은 3차원 위치 데이터를 기존 마커를 이용한 방법과 비교를 하였고, HMM 알고리즘을 이용한 모션인식에 대한 결과를 보여줬다.

## II. 모션 인식 시스템 구조

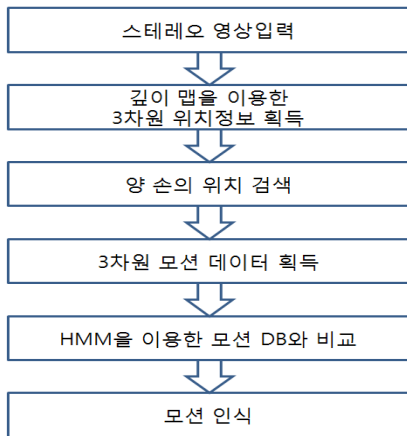


그림 1. 인식 시스템의 흐름도  
Fig. 1. Flowchart of recognition system

본 논문에서 모션 인식을 위한 시스템을 구현하기 위해 그림 1과 같은 흐름도로 구성을 했다. 먼저 웹캠을 사용하여 두 개의 스테레오 영상을 입력 받도록 하였다. 두 번째로 이 두 영상을 이용하여 3차원 데이터로 변환된 깊이 지도(depth map)를 구성하였다. 세 번째 단계에서는 이 깊이 지도에서 양손의 위치 정보를 획득하는 과정을 거치게 된다. 이 부분에서 일반적으로 마커를 이용하기도 하는데 여기서는 마커를 이용하지 않고 정보를 얻는다. 마커라는 것은 영상에서 특정 위치의 점을 인식하기

위해 고안한 방법으로 적외선 발광 다이오드를 사용했다. 그리고 그 특정 위치의 점을 인식하기 위해 적외선 카메라를 사용해야만 하는 불편한 점이 있다. 물론, 기존 상용 모션 캡처 장비와 같은 방법으로 오차가 수 mm 이내로 정확도가 높은 반면, 장비가 고가라는 단점이 있다. 그에 비해 마커레스 장비는 일반 가시광선 영역의 영상을 사용하여 얻는 방법으로 마커를 이용한 장비에 비해 정확도는 떨어질 수밖에 없다. 하지만 덜 정확한 위치정보를 인식 알고리즘을 통해 보완하고자 하는 것이 하나의 가능한 방법론이라 하겠다.

네 번째 단계에서는 연기자에 의해 움직이는 모션을 받아 들여 실시간으로 데이터를 획득하는 부분이다. 하나의 모션을 받아 들여야 그 모션이 어떤 모션에 해당한 지 분석을 할 수 있다. 따라서 진행 중 일 때는 모션을 판단하지 않는다. 다섯 번째 과정에서는 기존에 인식하고자 하는 모션들을 정리 및 저장해 놓은 데이터베이스와 비교를 한다. 여기서 비교 알고리즘으로는 HMM 기법을 사용했다. 그리고 끝으로 현재 동작한 모션이 어떤 모션인지 판단을 하게 된다.

## III. 3D 위치 정보 획득 방법

두 대의 카메라를 통해 영상을 획득한다. 획득한 영상은 스테레오 비전에 의해 3차원 정보를 만든다. 이런 과정을 구현하기 위해 각 카메라는 캘리브레이션이라는 과정을 거쳐 카메라의 특성을 분석하고, 왜곡 등을 고려한 영상으로부터 다시 스테레오 보정을 한다. 이 보정을 하고 나서 스테레오 비전 알고리즘을 적용하여 3차원 데이터를 계산하게 된다. 이렇게 얻은 정보에서 다시 양 손의 위치 정보를 추적하여 얻는다.

### 1. 3D 스테레오 비전과 캘리브레이션

스테레오 매칭 알고리즘<sup>[6]-[10]</sup>은 현재 매우 많은 연구가 계속 진행 중이며, 본 연구에서는 기본적인 매칭이론을 이용한 연구를 했다. 이 매칭 알고리즘은 삼차원 공간의 동일한 점에 해당하는 각각 영상의 점들이 서로 매칭(matching)될 때, 공간적으로 다른 평면으로부터 얻어진 2차원 이미지 쌍으로부터 3차원 세계를 복원<sup>[11]</sup>하는 과정이다. 이때 기하학 구조는 에피폴라 제약조건을 따라 3D 좌표값을 계산한다.

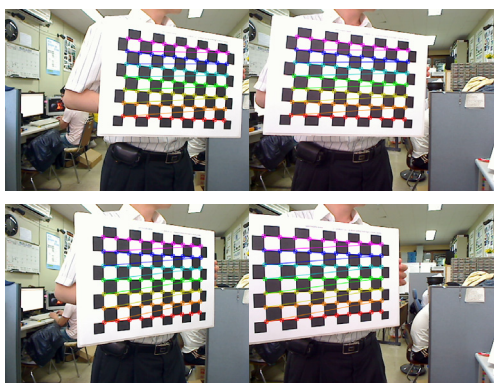


그림 2. 캘리브레이션  
Fig. 2. Calibration

3D 좌표값을 구하기 위해 카메라의 캘리브레이션이 필요하며, 카메라 캘리브레이션에는 intrinsic 파라미터와 extrinsic 파라미터가 있다. 이것은 기존에 많이 사용하는 체스판을 사용하여 내부와 외부 파라미터를 구할 수 있다. 본 연구에서는 20장의 체스판을 이용하여 캘리브레이션을 수행했다.

## 2. 스테레오 보정

스테레오 보정을 위해 Hartley 방법과 Boguet 보정 알고리즘을 사용하는데 본 논문에서는 Hartley가 제안한 알고리즘을 사용했다. 이것은 기본 행렬만을 이용하여 보정되지 않은 스테레오 영상을 조정한다. 두 영상에서 대응되는 7개 이상의 점들을 이용하여 계산된 기본 행렬을 알고 있을 때 다음과 같이 구하게 된다. 먼저, 기본 행렬을 이용하여 양쪽 에피폴을 계산한다. 식 (1)과 같이 fundamental matrix를 이용하여 구한다.

$$Fe_l = 0, \quad e_r^T F = 0 \quad (1)$$

그리고 호모그래피 Hr을 구한다. 이 호모그래피는 오른쪽 에피폴을 무한대의 점(1,0,0) T에 매핑시킨다. 오른쪽 영상의 원점을 이동시키는 이동 벡터 T와 에피폴을  $e_r^T = (f, 0, 1)$ 로 변환 시키는 회전 행렬 R을 구하여 식 (2)와 같은 호모그래피를 구할 수 있다.

$$H_r = GRT \quad (2)$$

$$G = \begin{Bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -k^{-1} & 0 & 1 \end{Bmatrix}$$

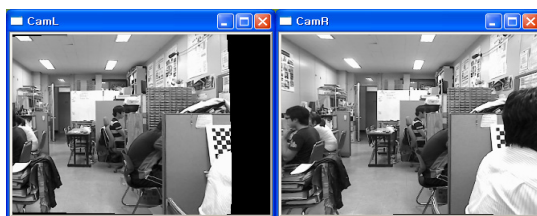


그림 3. Hartley 보정 알고리즘 결과 영상  
Fig. 3. Result of Hartley rectify algorithm

이번엔 왼쪽 에피폴을 무한대로 보내고 행 정렬되도록 만드는  $H_l$ 을 구한다. 마지막으로 왼쪽-오른쪽 대응점들의 전체 시차  $\sum_i d(H_l p_i^l, H_r p_i^r)$ 가 최소가 되는  $H_l$ 을 찾는다. 이것을 가지고 영상을 보정한다.

원본 영상에서 변형된 좌표에 매핑되면 주변 픽셀값들은 보간을 해서 픽셀값을 구하게 된다. 최종적으로 매핑을 하여 구하여 영상 보정을 마치게 된다.

## 3. 시각차 맵(Disparity)과 깊이 맵 (Depth Map)

시각차 맵<sup>[12]</sup>을 구하기 위한 방법으로 스테레오 매칭과 블럭 매칭 방법이 있다. 간단히 알고리즘을 살펴보면 Stereo Matching은 두개의 flow를 가진다. 우선 Dynamic Programming을 이용해서 disparity map을 생성하고, 생성된 disparity map 정보를 바탕으로 높은 신뢰성(higher reliability)을 가지기 위해 postprocessing을 한다. 그리고 DP(Dynamic Programming)을 사용해서 disparity map을 생성한다.

두 번째 알고리즘으로 블럭 매칭 알고리즘이 있다. 이 방법은 스테레오 조정된 왼쪽과 오른쪽 영상에서 대응점을 찾기 위해 윈도우 내부의 SAD(Sum of Absoute Difference)를 계산한다. 블록-매칭 스테레오 대응 알고리즘은 세 단계로 구성되는데, 전처리 과정, 대응점 찾기, 후처리 과정으로 이루어졌다. 전처리과정에서는 영상의 밝기를 정규화하고 질감을 향상 시키는 과정이며, 대응점 찾는 과정은 수평 에피폴라 직선을 따라 SAD 윈도우를 이동시키며 찾는 과정이다. 마지막으로 잘못된 대응점들을 제거하는 과정으로 끝난다.

시각차 지도에서 3차원 모양으로 만들 필요가 있는데, 이것을 재구성 (reconstruction) 또는 깊이 지도 (depth map) 구성이라 한다. 식(6)의 재투영 Q 행렬을 이용하여 2차원 점을 3차원으로 변환을 할 수 있다.

Depth map을 보면 뒤 부분이 크고 맨 가운데 부분이

오히려 작다. 이것은 실제 disparity map에서 가까운 것은 크게 보이고 먼 것은 작게 보이는 perspective projection을 실제 크기에 맞게 구현했기 때문이다. 따라서 앞부분은 작게 보이고, 상대적으로 뒤 부분은 크게 보이게 된다. Perspective Projection을 변환하기 위해 그림 4와 같은 행렬을 사용했다.

```
// construct the matrix that maps from Cartesian
coordinates to disparity coordinates
double data [] = {
    mFx, 0, mClx, 0,
    0, mFy, mCy, 0,
    0, 0, (mClx-mCrx)/mDu, mFx*mTx/mDu,
    0, 0, 1, 0
};
CvMat _P = cvMat(4, 4, CV_64FC1, data);
cvCopy(&_P, &mMatCartToDisp);

//construct the matrix that maps from disparity
coordinates to Cartesian coordinates
double Q[] =
{
    1, 0, 0, -mClx,
    0, mFx/mFy, 0, -mCy*mFx/mFy,
    0, 0, 0, mFx,
    0, 0, 0, mDu/mTx,
    -(mClx-mCrx)/mTx
};
CvMat _Q = cvMat(4, 4, CV_64F, Q);
cvCopy(&_Q, &mMatDispToCart);
```

그림 4. Perspective Projection을 위한 변환 행렬  
Fig. 4. Matrix for perspective projection

#### 4. 양 손의 위치 추적 방법

생성된 깊이 맵에서 양 손의 위치를 추적하기 위해 그림 5와 같은 방법을 사용하였다. 먼저, 연기자의 모습이 보이면, body의 기본 위치를 구하고, 양손의 기본 자세를 통해 왼손과 오른 손의 위치를 구한다. 양 손의 위치는 대개 body 보다 앞쪽에 존재 하며, 각각은 왼 쪽과 오른 쪽에 위치를 하게 된다. 움직임에 따라 연속해서 그 위치를 추적하게 된다.

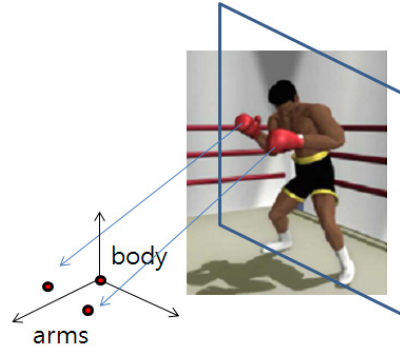


그림 5. 양 손의 위치 계산  
Fig. 5. Positions of two arms

### IV. 모션 인식 방법

본 논문에서는 권투 모션을 일반 연기자로부터 얻어 각 모션을 인식하는 방법을 구현했다. 연기자로부터 위의 방법을 이용하여 얻어진 모션 데이터는 순차적으로 저장되어 모션 그래프로 변환된다. 그리고 HMM 인식 모델을 설정하고, 모션 인식률을 얻기 위해 각 모션별로 실험을 했다.

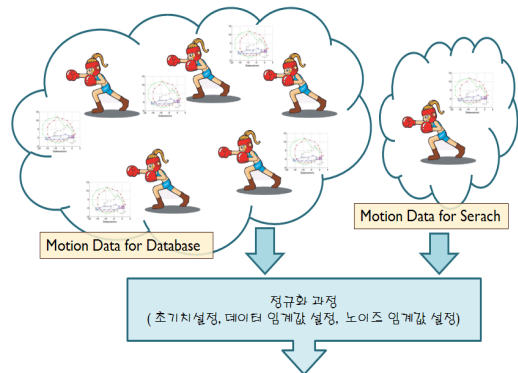


그림 6. 모션 데이터 불러오기  
Fig. 6. Load motion data

#### 1. 연기자 모션 데이터

모션 데이터를 구축하기 위해 다섯 명의 일반인으로 구성했다. 세부 동작에는 훅(Hook), 잭(Jab), 스트레이트(Straight), 어퍼컷(Uppercut) 이 있다. 최종 실험에서 사용되는 모션 그래프 개수는 모두 20개 (연기자 5명 x 4가지 세부동작)가 되며, 몇 번의 연기를 통해 가장 일반적

인 모션들을 데이터베이스에 저장한다.

## 2. 모션 그래프

대량의 모션 데이터로부터 적절한 데이터를 부분적으로 선택하고 이를 새로운 순서로 연결하여 새로운 동작 시퀀스를 생성하는 것이다. 이러한 작업을 효율적으로 수행하기 위해 모션 데이터를 방향성 그래프로 표현하고 저장한다. 모션 데이터가 입력으로 주어지면, 그 안의 모든 프레임들 중에 서로 자세와 속도가 비슷한 프레임 쌍을 찾아내고 이들 간의 전이 에지를 연결함으로써 모션 그래프를 구성할 수 있다. 마크로로부터의 정보를 각각의 모션으로 분류하여 데이터베이스에 저장한다. 저장된 모션들은 각각의 그래프로 구성되어 인식시스템에서 사용된다.

그림 7에서 x,y,z축을 기준으로 그래프가 보이고 있는데, 이것은 월드 좌표계이며, 모션 캡처 시스템에서 가져온 그림이다. 각 모션들에 대한 움직임이 보이고 있다.

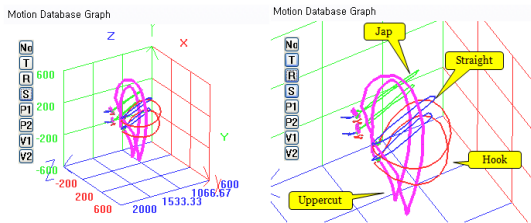


그림 7. 네 가지 모션의 그래프  
Fig. 7. Four kind of motion graph

## 2. HMM 모션 인식

은닉 마르코프 모델(Hidden Markov Model)이란 마르코프 모델이 감추어졌다는 의미이다. 관찰이 불가능한 숨겨진 확률론적 과정을 관찰이 가능한 기호로 발생시키는 다른 확률론적 과정을 통하여 모형화하는 이중의 확률론적 과정이다. HMM은 통계적 모형의 하나로서 시계열로 표현되는 신호의 패턴을 인식하거나 필기체를 인식하는 문제에 널리 이용되었다. 어떤 객체의 패턴을 표현하는 모형으로서, 그 객체의 패턴을 심볼 스트링 형태로 분석한다. 통계적 특징을 훈련하여 입력 스트링에 대해 관찰할 확률이 얼마인가 하는 최고 확률을 계산하는 방법이다. 본 연구에서는 구해진 최고의 관찰된 확률을 통해 패턴의 스트링을 인식하게 된다.

HMM의 모델 학습과정은 관측열에 대하여 최대 확률을 가지는 적합한 모델을 선택하고, 이 모델의 내부 상태들 중에서 최적의 상태 순서를 결정한 후, 관찰열 X에 대한  $P(X|\lambda)$ 를 최대화하는 모델  $\lambda = (\pi, A, B)$ 의 매개변수를 결정한다. 마르코프 매개변수는 다음 식과 같이 세 가지 매개변수를 갖는다.

- ① 사전 확률 (prior probability) : 초기 상태 확률 백터라고도 하는데, HMM을 처음 시작할 때 어느 상태에서 시작할 지를 결정하는데 사용한다.

$$\pi_i = P(q_1 = s_i), \sum_{i=1}^t \pi_i = 1 \quad (3)$$

- ② 상태 전이 확률 (transition probabilities) : A는 HMM이 시행중 다음 상태를 결정하는데 사용된다.  $a_{ij}$ 는 시간 t-1에 상태  $s_i$ 에서 시간 t에  $s_j$ 로 이동할 확률이다. 조건은  $a_{ij}$ 의 모든 합은 1이어야 한다.

$$A = \{a_{ij}\}, 1 \leq i, j \leq t$$

$$a_{ij} = P(q_t = s_j | q_{t-1} = s_i), \sum_{j=1}^t a_{ij} = 1 \quad (4)$$

- ③ 관찰 심볼 확률 (emission probabilities) : B는 HMM이 어느 상태에 도달했을 때 그 상태에서 관찰되는 심볼을 결정하는데 쓰인다.  $b_{ik}$ 는 상태  $s_i$ 에서 심볼  $v_k$ 가 관측될 확률이다.

이렇게 구해진 모델은 확률 추정하게 된다. 방법은 3단계로 이루어진다.

- ① 평가 (evaluation) : 관측열  $X = \{x_1, x_2, \dots, x_T\}$ 와 모델  $\lambda = (\pi, A, B)$ 에 대하여 주어진 HMM에서 관찰되어진 순서의 확률  $P(X|\lambda)$ 를 계산하는 문제이다. 이 경우는 전방향 알고리즘과 후방향 알고리즘을 이용하여 해결이 가능하다.
- ② 디코딩 (decoding) : 관측열  $X = \{x_1, x_2, \dots, x_T\}$ 와 모델  $\lambda = (\pi, A, B)$ 에 대하여 최적의 상태순서  $Q = \{q_1, q_2, q_3, \dots, q_N\}$ 를 생성할 확률이 가장 높은 은닉상태들 간의 순서를 찾는 문제이다. 일반적으로 동적 프로그래밍 기법의 하나인 비터비 알고리즘을 사용한다.
- ③ 학습 : 관찰된 관측열  $X = \{x_1, x_2, \dots, x_T\}$ 에 대하여  $P(X|\lambda)$ 를 최대로 하는 모델  $\lambda = (\pi, A, B)$ 의 매개변수를 결정하는 문제이다. 매개변수의 추정은 Baum-Welch 알고리즘을 사용하여 수행할 수 있다.

먼저 HMM을 이용한 전체 프로세스는 그림 8과 같다. 모션 데이터를 읽어 들여 초기화를 거쳐 평가단계를 거쳐 인식단계로 넘어간다.

HMM을 초기화 하는 과정에서 전이 행렬 (transition matrix)과 관찰 행렬 (observation matrix)은 훈련 과정을 통해 확률 값이 결정된다. 마코프 모델(markov model)에서는 3x3 전이 행렬의 의미는 3가지 상태를 갖고 있으며 상태가 전이될 확률을 갖는 행렬을 의미한다. 즉, 본 논문에서는 상태를 모두 27가지 3차원 체인코드로 설정하여 수행을 하므로, 27x27 형태의 행렬이 필요하다. 하지만 은닉 마코프 모델(HMM)에서는 3x3 전이 행렬과 3x27 관찰 행렬로 구성하여 마치 3x3행렬에 숨겨진 듯한 모델로 표현을 하고 있다.

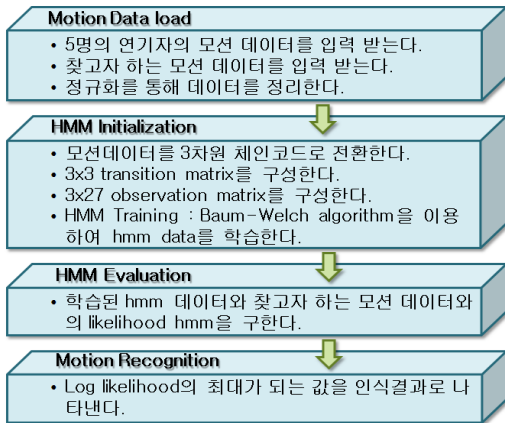


그림 8. HMM 인식 프로세스  
Fig. 8. HMM recognition process

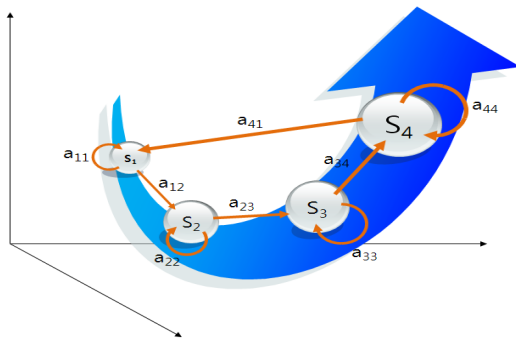


그림 9. 상태 전이 다이어그램 (어퍼컷)  
Fig. 9. State transfer diagram (uppercut)

그림 9는 어퍼컷의 마코프 모델을 보여주고 있는데 여기서는 3개가 아닌 4개의 상태를 보여주고 있다. 실제로는 4개로는 어퍼컷을 구성할 수 없으며, 이것을 HMM에서는 s1, s2, s3, s4 (, s5, s6, ... )중, 각 상태에서 27개의 상태 중 관찰된 상태가 시간에 따라 진행되는 것을 의미한다. 이 s1, s2, s3, s4, ..를 묶어서 상태열이라 한다.

상태열은 모션 그래프로부터 3차원 체인 코드를 이용하여 데이터를 변환하여 얻을 수 있다. 이것을 HMM에 적용하여 매개변수를 구하고 전이확률을 구하여 HMM 모델을 생성한 후, 인식할 때는 학습 모델과 비교하여 likelihood 값으로 판단한다. 다음은 HMM 평가 단계이다. 학습된 hmm 데이터와 찾고자 하는 모션 데이터의 HMM-likelihood를 구하며, 그 값 중 가장 큰 값을 모션이 가까운 것으로 인식한다.

## V. 실험 결과

깊이 정보를 얻는 방법에는 스테레오 매칭 방법과 블럭 매칭 방법이 있는데 여기서는 스테레오 매칭 방법만을 사용했다. 정확도를 측정하기 위해 각각 거리 측정을 해서 표로 나타냈으며, 모션인식에 대한 결과를 보였다.

### 1. 위치 정보 정확도 실험

그림 10은 스테레오 매칭 결과를 보여주는 영상이다. 입력 변수 중 MaxDisparity 값은 80이다. 이 값에 따라 전체 밝고 어두운 정도가 달리 나타난다. 영상 처리 속도는 약 3.20 fps 나온다. (Core2 Quad CPU 2.4GHz)

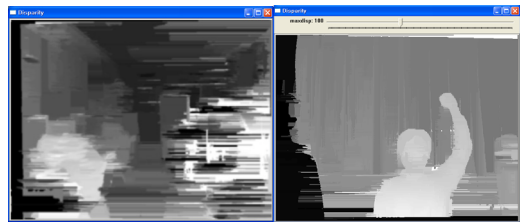


그림 10 스테레오 매칭 결과  
Fig. 10. Results of stereo matching algorithm

그림 11은 스테레오 매칭 결과 데이터를 깊이 맵으로 변환한 결과 영상이다. 깊이 맵에서 먼저 연기자의 body를 찾아내어 기본 거리를 측정하게 된다.

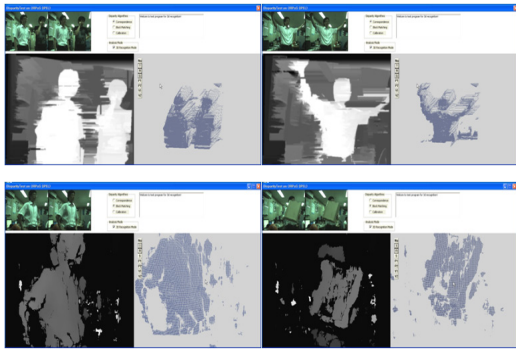


그림 11. 깊이 맵 결과 영상  
Fig. 11. Results of depth map

그림 12는 기존 연구에서 거리 측정을 위한 실험결과를 그래프로 나타낸 것이며, 측정 방법은 90cm에서 30cm씩 이동하면서 가운데 보이는 사람의 body 영역의 거리 값을 평균을 하여 측정 결과 값을 출력했다.

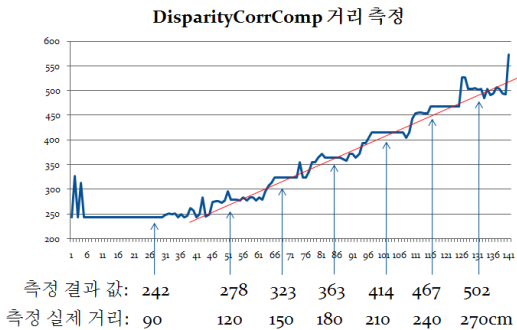


그림 12. SM 거리 측정 그래프  
Fig. 12. Graph of distance experiment with SM

표 1. SM 거리 측정 값  
Table 1. Result values of distance with SM

실제거리	90	120	150	180	210	240	270
측정결과	242	278	323	363	414	467	502
보정값	91.4	116.6	148.1	176.1	211.8	248.9	273.4
오차(cm)	1.4	-3.4	-1.9	-3.9	1.8	8.9	3.4

오차는 측정된 결과 값을 선형으로 보정하여 나온 결과 값과 비교했다. 수행속도는 270ms정도 나왔다. 결과표를 보면 최대 8.9cm의 오차를 보여준다. 연기자의 body가 측정된 후 이를 기본으로 좌,우측 손의 위치를 계산하여 양 손의 위치를 찾아낸다.

표 2. HMM을 이용한 모션 인식 결과  
Table 2. Recognition result for HMM

연기자 \ 모션	Hook	Jab	Straight	Uppercut
Person 1	○	○	○	○
Person 2	○	○	○	○
Person 3	○	○	○	×
Person 4	×	○	○	○
Person 5	○	○	○	○
인식률	90 %			

## 2. 모션 인식 실험 결과

본 연구에서는 깊이 맵 방법을 통해 얻은 양손의 위치 정보를 3차원 모션 그래프로 변환하였고, 이 권투 모션 데이터를 마르코프 모델을 이용한 모션 인식 알고리즘에 적용한 결과는 최대 90% 성능을 보였다. 물론 마르코프 모델을 훈련하는 과정에서 확률로 최대치를 추정하므로 실험할 때마다 결과가 항상 같게 나오지 않는다. 또한 양손의 위치를 얻는 데이터도 오차 범위가 있으나, 보통 80%~90%의 결과로 나타나고 있다. 인식의 성능이 80% 이상의 결과를 보여주고 있는 것은 3차원 모션 입력데이터가 좀 더 정확할 필요가 있는 것으로 생각된다.

## VI. 결론

기존 연구에서는 마커를 이용한 3차원 위치 정보를 획득하는 방법을 이용하여 모션을 인식하는 시스템을 구현했다. 본 연구에서는 마커가 없는 깊이 맵 방법을 이용하여 모션을 인식하는 시스템을 구현하였다. 두 시스템 모두 성능 면에서는 별 차이가 없었다. 다만, 본 연구에서 모션 인식 결과가 기존 연구 실험결과(95%)보다 떨어지는 이유는 3차원 위치 정보의 오차에 기인한다고 볼 수 있다. 하지만, 인식률 차이는 큰 의미가 없어 보인다. 앞으로 계속적인 연구를 통해 다양한 모션에 대해서도 인식할 수 있는 방법을 모색하며, 또한 마커리스 방법에서 3차원 위치 정보의 정확도를 높일 수 있어야 하겠다.

## 참고 문헌

[1] Jean, Grace V. Road warriors: robots get smarter,

- but who will buy them? National Defense, March 2008.
- [2] Kageyama, Yuri. "Walking, talking female robot to hit Japan catwalk." The Seattle Times. 16 March 2009.
- [3] OPRoS 팀, www.opros.or.kr
- [4] Park,H.S., "Development of robot S/W platform verification and estimation automation technology", OPRoS combination workshop 2008. 11.
- [5] Han,C.H., Oh,C.S., "Development of a 3D Object Recognition Component for OPRoS", IWIT, Vol11-3-12, 2011. 6
- [6] Trucco, Emanuele. Introductory Techniques for 3D Computer Vision. Prentice Hall Inc, 1998.
- [7] H. Jeong and S.C. Park, "Generalized Trellis Stereo Matching with Systolic Array," In Lecture Notes in Computer Science, Vol.3358, 2004, pp.263-267.
- [8] Abhijit S. Ogale and Yiannis Aloimonos, "Shape and the Stereo Correspondence Problem", DCV 65,3, pp. 147- 162, 2005.
- [9] Sukjune Yoon, Sung-Kee Park, Sungchul Kang, Yoon Keun Kwak, "Fast correlation-based stereo matching with the reduction of systematic errors", Pattern Recognition Letters 26, pp. 2221- 2231,2005.
- [10] Michael Bleyer, Margrit Gelautz, "A layered stereo matching algorithm using image segmentation and global visibility constraints", ISPRS Journal of Photogrammetry & Remote Sensing 59, pp. 128- 150, 2005.
- [11] Hansung Kim, Kwanghoon Sohn, "3D reconstruction from stereo images for interactions between real and virtual objects", Signal Processing: Image Communication 20, pp. 61- 75, 2005.
- [12] K. Muhlmann, D. Maier, I. Hesser and R. Manner, "Calculating Dense Disparity Maps from Color Stereo Images, an Efficient Implementation", DCV 47, 1/2/3, pp. 79-88, 2002.
- [13] Bradski, Gary; Kaehler, Adrian. Learning OpenCV: Computer Vision with the OpenCV Library. O'Reilly Media Inc, 2008.

## 저자 소개

### 한 창 호



- 1993년 청주대 전자공학과 (학사).
- 1995년 동 대학원 전자공학과 (석사).
- 2005년~2009년 선문대학교 대학원 전자공학과 (박사).
- 2009년~ 2011년 선문대학교 시간강사.
- 2012년 ~ 현재 선문대학교 정보통신공학과 계약교수.

<관심분야는 영상처리, 모션캡처, 자동검사, 로봇 비전>

### 오 춘 석



- 1980년 서강대학교 전자공학과 (학사).
- 1986년 Marquette University 컴퓨터공과 (석사).
- 1992년 University of Arizona 컴퓨터공과(박사).
- 1980년~1993년 전자통신연구원(ETRI)

선임연구원.

- 2001년 미국 Optical Sciences Center Visiting Scholar.
- 1993년~현재 선문대학교 전자정보통신공학부 교수.

<주관심분야는 화상처리, 자동검사, 머신비전, 광학시스템설계>