

HRI 시스템에서 제스처 인식을 위한 Moving Mean-Shift 기반 사용자 손 위치 보정 알고리즘

김 태 완*, 권 순 량*, 이 동 명°

A Compensation Algorithm for the Position of User Hands Based on Moving Mean-Shift for Gesture Recognition in HRI System

Tae-Wan Kim*, Soon-Ryang Kwon*, Dong Myung Lee°

요 약

본 논문은 키넥트 센서 (Kinect sensor)를 탑재한 Human Robot Interface (HRI) 시스템에서 손 위치 데이터를 측정하여 제스처 인식 및 처리성능을 높이기 위하여 Moving Mean-Shift 기반 사용자 손 위치 보정 알고리즘 ($CAPUH_{MMS}$)을 제안하였다. 또한, $CAPUH_{MMS}$ 의 성능을 자체 개발한 실시간 성능 시뮬레이터로 이동궤적에 대한 평균 오차 성능개선 비율을 다른 보정 기법인 CA_{KF} (Kalman-Filter 기반 보정 알고리즘) 및 CA_{LSM} (Least-Squares Method 기반 보정 알고리즘)의 성능과 비교하였다. 실험결과, $CAPUH_{MMS}$ 의 이동궤적에 대한 평균 오차 성능개선 비율은 양손 상하 운동에서 평균 19.35%으로, 이는 CA_{KF} 및 CA_{LSM} 보다 각각 13.88%, 16.68% 더 높은 평균 오차 성능 개선 비율을, 그리고 양손 좌우 운동에서 평균 28.54%으로 CA_{KF} 및 CA_{LSM} 보다 각각 9.51%, 17.31% 더 높은 평균 오차 성능 개선 비율을 나타낸 것이다.

Key Words : Kinect, Hands Gesture, Gesture Recognition, Compensation, Human Robot Interface

ABSTRACT

A Compensation Algorithm for The Position of the User Hands based on the Moving Mean-Shift ($CAPUH_{MMS}$) in Human Robot Interface (HRI) System running the Kinect sensor is proposed in order to improve the performance of the gesture recognition is proposed in this paper. The average error improvement ratio of the trajectories ($AEIR_{TJ}$) in left-right movements of hands for the $CAPUH_{MMS}$ is compared with other compensation algorithms such as the Compensation Algorithm based on the Compensation Algorithm based on the Kalman Filter (CA_{KF}) and the Compensation Algorithm based on Least-Squares Method (CA_{LSM}) by the developed realtime performance simulator. As a result, the $AEIR_{TJ}$ in up-down movements of hands of the $CAPUH_{MMS}$ is measured as 19.35%, it is higher value compared with that of the CA_{KF} and the CA_{LSM} as 13.88% and 16.68%, respectively.

I. 서 론

최근 유비쿼터스의 기술의 발전과 더불어 가상현실

을 기반으로 한 기술과 이를 활용한 동작인식 처리기술에 대한 연구가 매우 활발하게 진행되고 있다. 동작인식 처리기술 중 손을 이용한 제어 방법은 컴퓨터의

* First Author : Tongmyong University, Department of Computer Engineering, wanykt@nate.com, 학생회원

° Corresponding Author : Tongmyong University, Department of Computer Engineering, dmlee@tu.ac.kr, 종신회원

* Second Author : Tongmyong University, Department of Electronic Engineering, srkwon@tu.ac.kr, 정회원

논문번호 : KICS2014-06-233 Received June 12, 2014; Revised February 3, 2015; Accepted May 14, 2015

마우스나 키보드와 같은 기존의 인터페이스 제어 방법 대신 가상공간에서 손동작으로 자연스럽게, 각종 기기를 쉽게 조작할 수 있도록 한다. 손동작은 일반적으로 그 형태와 움직임이 다양하기 때문에 그 자체로 하나의 언어가 될 수 있으며, 그 의미를 더욱 강조할 수 있어 사람간의 의사소통에서도 중요한 역할을 하고 있다¹⁻³⁾.

손동작의 특징을 통한 제스처 인식 및 처리 방법은 크게 3가지 방법으로 분류된다. 첫 번째는 손의 위치 정보를 통하여 사용자가 취하고 있는 정보를 추정하는 방법, 두 번째는 획득된 사용자의 영상을 토대로 손의 위치를 모델링하는 영상처리 기반의 방법, 그리고 세 번째는 손의 위치나 움직임의 특징을 기반으로 하는 방법이다. 키넥트 센서 (Kinect sensor)를 이용한 사용자 제스처 인식 및 처리 방법은 위의 세 번째 방법 중 하나로, 주로 사용자의 위치정보를 기반으로 연구 개발되고 있다. 특히 키넥트 센서는 로봇의 Human Robot Interface (HRI) 시스템에서 사람의 눈과 같은 역할을 하며, 제스처 인식 및 처리를 이용하여 직접 로봇의 동작을 제어한다^{4,5)}.

본 논문은 키넥트 센서를 장착한 HRI 시스템에서 기존의 사용자 위치정보 인식 성능을 향상시키기 위하여 Moving Mean-Shift 기반 사용자 손 위치보정 알고리즘 (Compensation Algorithm for the Position of User Hands based on Moving Mean-shift, CAPUH_{MMS})을 제안하고, 실제 HRI 시스템에서 측정된 손 인식위치 데이터와 자체 개발한 실시간 성능분석용 시뮬레이터를 사용하여 제안한 알고리즘의 성능을 분석하였다.

II. 관련 연구

2.1 사용자 제스처 인식 및 처리

사용자 제스처 인식 및 처리 분야 중 특히 손 위치의 특징을 이용하여 순간적 표현이나 회전방향 정보를 이용한 많은 연구가 진행되고 있다⁶⁻⁹⁾. [6]은 키넥트 센서로부터 손의 위치 정보를 획득한 후 정규화하고, Dynamic Time Warping (DTW) 알고리즘을 이용하여 손의 제스처 정보를 숫자로 표현하였다. [7]은 움직이는 손의 위치 특징을 추출하여 미국인 수화 (American sign language)를 인식하였다. [8]은 손 위치정보의 특징은 순간적 표현이나 회전에 민감하다는 점을 해결하기 위하여 양 손 사이의 각도와 회전 방향 정보를 계산해서 오스트리아인 수화 (Australian sign language) 인식성능을 향상시켰다. [9]는 키넥트 센서

로부터 획득한 위치정보를 이용하여 몸의 위치좌표 및 손과 팔꿈치의 위치 좌표의 각도를 계산하여 제스처를 인식하였다. 결국 키넥트 센서를 이용한 제스처 인식 및 처리 방법의 경우, 사용자 위치정보는 순간적으로 이루어지기 때문에 인식 정확도 확보가 매우 중요하다.

2.2 모바일 로봇의 자연스러운 손동작 제어

모바일 로봇의 자연스러운 손동작 제어에서 중요한 점은 인위적인 손동작이 아닌 실제 생활환경에서의 자연스러운 손동작으로 로봇을 제어하는 것이다. 키넥트 센서를 이용해서 위와 같은 제어를 할 경우, 그림 1과 같이 깊이(Depth) 데이터로 손의 위치를 추적하고 사람의 손인지 아닌지를 검증하기 위하여 YCbCr 컬러 모델의 특정 컬러를 이용하여 손 영역을 추출한다.

[10]은 은닉 마르코프 모델 (Hidden Markov Model, HMM)을 사용하여 그림 2에서 보는 바와 같이 7가지로 자연스러운 손동작을 정의하였다. HMM은 패턴들의 유한 상태들을 시간에 따른 결합 확률로 나타내며, 바움-웰치 재 추정 알고리즘(Baum-Welch re-estimation algorithm)을 이용하여 데이터를 학습하고, 손동작의 우도 (Likelihood) 데이터와 로그 우도 (Log likelihood)의 최댓값을 구하여 동작을 인식한다.



그림 1. 키넥트 센서를 이용한 손 위치 추적과정
Fig. 1. Hand position trajectory process using Kinect sensor.



그림 2. HMM을 사용한 손동작 예
Fig. 2. Example of hand gestures based on HMM.

그리고 손동작의 속도, 손을 움직이는 거리, 그리고 다수의 손동작을 Mamdani 퍼지 이론으로 이동속도를 결정하여 모바일 로봇에 적용하였다.

2.3 보정 알고리즘

2.3.1 칼만 필터

칼만 필터 (Kalman Filter)는 과거, 현재, 미래의 상태를 예측하는 알고리즘으로서 컴퓨터 비전, 로봇 공학 등 다양한 분야에서 널리 사용된다. 칼만 필터는 측정값에 확실적인 오차가 포함되어 있고 특정 시점 상태는 이전 시점 상태와 선형적 관계를 가지는 경우에 적용이 가능하다.

칼만 필터를 이용한 보정 알고리즘의 절차 모델은 시간 업데이트 (예측 단계)와 측정 업데이트 (보정 단계)의 두 단계로 이루어지며 반복적으로 수행된다. \hat{X}_k 는 현재 추정치를 나타내며, K_k 는 칼만 이득 (Kalman gain), Z_k 는 측정된 값, \hat{X}_{k-1} 는 이전 추정치를 나타낸다. 이들 초기 값들이 설정되면 예측 단계 (추정치 예측, 오차 공분산 예측)와 보정 단계 (칼만 이득계산, 추정치 계산, 오차 공분산 계산)가 반복적으로 수행된다^{11,12}.

2.3.2 Mean-Shift 기반 보정 알고리즘

Mean-Shift는 흩어진 데이터 분포의 중심이나 무게 중심을 찾아가는 알고리즘이다. 주로 컴퓨터 비전 분야에서 특히 영상의 객체 추적, 분할, 스무딩 (Smoothing), 필터링, 엣지 검출 (Edge detection) 등에 사용되는 기법 중의 하나이다. 무게 중심을 찾기 위한 탐색 반경을 r 이라고 하였을 때 데이터 분포의 중심점은 다음 절차에 의하여 탐색된다^{12,13}.

식 (1)에서 X 는 CP_i (현재 무게 중심점)에서 탐색 반경 r 거리 내에 있는 측정된 데이터의 집합이다.

$$X = [X_1, X_2, X_3, \dots, X_n] \quad (1)$$

식 (2)를 이용하여 CP_{i+1} (다음 무게 중심점)을 구한다. $f(X)$ 는 CP_i 와 X 사이의 거리를 구하며, 구해진 거리 값의 평균을 CP_{i+1} 으로 설정한다.

식 (1)과 식 (2)의 과정은 CP_i 의 위치 변화가 없을 때 까지 반복적으로 수행한다.

$$CP_{i+1} = \frac{1}{n} \sum_{i=0}^n f(X) \quad (2)$$

2.3.3 최소자승법 기반 보정 알고리즘

최소자승법 (Least-Squares Method, LSM)은 어떤 점들의 분포를 직선이나 곡선에 대하여 수학적 도구로 수치해석 및 회귀 분석뿐만 아니라 영상처리 분야 및 이동하는 객체의 위치인식 정확도 향상 등 다양한 분야에서 사용된다. LSM에 의한 이동노드 위치좌표 값 보정방법은 다음과 같다. 먼저, 이동노드의 n 개 위치좌표 데이터 점들을 $(x_1, y_2), (x_2, y_2), \dots, (x_n, y_n)$ 라 하고, x_n 와 y_n 을 각각 독립변수와 종속변수로 두면, 1차 방정식을 근사하기 위한 오차의 제곱의 합 S 는 식 (3)과 같다.

$$S = \sum_{i=1}^n e_i^2 = \sum_{i=1}^n (y_i - a_0 - a_1 x_i)^2 \quad (3)$$

식 (3)에서 매개변수 a_0 와 a_1 에 대해 S 를 최소화 한 후 Cramer 규칙을 이용하면 미지수 a_0 와 a_1 을 각각 구할 수 있다. 구해지는 a_0 와 a_1 을 위치좌표 (x_n, y_n) 에 적용하면 보정된 위치좌표를 얻을 수 있다¹⁴.

III. Moving Mean-Shift 기반

사용자 손 인식 보정 알고리즘(CAPUH_{MMS})

Mean-Shift^{12,13}는 탐색 반경을 설정하여 현재 탐색 반경의 무게 중심을 찾는다. 이 방법은 특히 주변 상황만으로 탐색방향을 결정하기 때문에 탐색 반경의 설정이 중요하나 탐색 반경의 크기를 결정하는 것이 쉽지 않다는 단점을 가지고 있다. 탐색 반경을 너무 크게 설정하면 현재 무게 중심이 최적의 결과로 판단하여 탐색이 이루어지지 못하는 지역 최적해의 결과에 빠지게 되며, 반대로 너무 작게 설정하면 다음 탐색방향으로 이동하지 못하는 문제점이 발생한다. 따라서 손 위치 측정데이터가 탐색 반경보다 큰 이동을 하게 된다면 다음 손 위치 측정 데이터로 이동하지 못할 뿐만 아니라 실시간 손 위치 추적 및 보정 작업이 제대로 이루어지지 않을 가능성이 높아진다. 그래서 실시간으로 변화하는 손 위치의 보정에 그대로 적용하기에 문제가 있기 때문에 이를 해결하기 위한 방안이 필요하다. 따라서 본 논문에서는 위와 같은 문제를 제안하는 CAPUH_{MMS}를 이용하여 해결하였다.

3.1 CAPUH_{MMS}의 처리 절차

제안하는 CAPUH_{MMS}의 처리 절차는 그림 3과 같이 4단계로 진행된다.

1단계: 손 위치 좌표 입력 (Input of hands position coordinates) 단계로 사용자의 왼손과 오른손에 해당하는 실제 손 위치 좌표를 입력한다.

2단계: 좌표 데이터 수집 (Collection of coordinates) 단계로 1단계에서 입력된 손 위치 좌표를 일정 크기를 가진 큐 (Queue)에 저장한다.

3단계: *MMSA*의 실행 (Execution of Moving Mean-Shift Algorithm, *MMSA*) 단계로 *MMSA*는 1) 파라미터 값 초기화, 2) 탐색반경 r 안에 있는 손 위치 좌표 검색, 3) 검색된 손 위치 좌표의 합 저장, 4) 저장된 좌표들의 평균 값 계산, 5) *MMSA* 조건 확인의 총 5단계로 실행된다.

4단계: 중심점 위치좌표 설정 (Setup of Central Point Position Coordinates) 단계로 3단계의 실행이 종료 조건에 만족할 경우, 현재 좌표를 다음 중심점 위치좌표로 설정하고 $CAPUH_{MMS}$ 를 종료한다.

Procedure of $CAPUH_{MMS}$

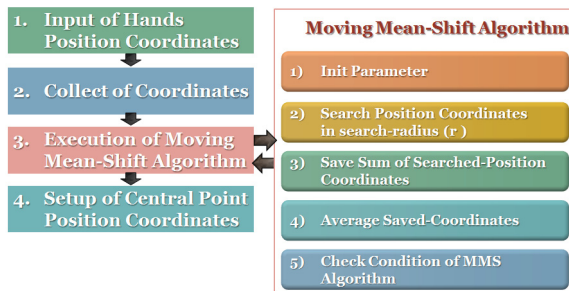


그림 3. 손 위치 보정 알고리즘의 처리 절차
Fig. 3. Procedure of the compensation algorithm for the position of user hands.

3.2 제안하는 *MMS* 알고리즘

*MMSA*는 식 (4)~식 (7)과 같이 처리된다. *MMS* 데이터 (중심점 데이터)는 식 (4)에 의하여 구할 수 있다.

$$MMS = \frac{1}{n} \sum_{i=0}^{n-1} M(X) \quad (4)$$

식 (4)는 다음과 같은 조건을 만족 하면 수행하게 되는데, 만약 미리 정의된 데이터 버퍼 크기가 현재 수집한 데이터 량 (n)보다 크면, 현재 입력데이터 (X_i)를 데이터 집합 (X)에 저장하여 현재 수집한 데이터 량 (n)크기만큼 *MMSA*가 수행된다. 만약, n 이 미리 정의된 데이터 버퍼 크기보다 크게 되면, X 에서 첫 번째 데이터 (X_0)를 삭제하고, X_i 를 X 에 저장한다. *MMSA*은 n 이 미리 정의된 데이터 버퍼 크기로 설정

된 후 수행된다. 식 (4)에서 $M(X)$ 를 구하는 방법은 식 (5)~식 (7)과 같으며, 식 (5)에서 X_i 가 초기 데이터 일 때, 최초 중심점 데이터 (MMS_0)는 X_0 로 설정되고 MMS_0 를 반환한다. 그러나 X_i 가 초기 데이터가 아니면 총 중심점 데이터 변수 ($TotalMMS$), 탐색반경 r 개수 변수 ($rCount$), 그리고 MMS_j 는 0으로 각각 초기화되며, 다음 조건 X_i 가 탐색 반경 r 거리 내에 있으면 식 (6)이 수행된다.

$$\begin{cases} \text{if } (i = 0) \text{ then } MMS_0 \leftarrow X_i \\ \quad \quad \quad \text{return } MMS_0 \\ \text{else then } MMS_j \leftarrow 0 \\ \quad \quad \quad \text{curr}MMS \leftarrow 0 \\ \quad \quad \quad rCount \leftarrow 0 \end{cases} \quad (5)$$

$$\begin{aligned} TotalMMS &\leftarrow TotalMMS + \frac{X_i + MMS_{j-1}}{2} \\ rCount &\leftarrow rCount + 1 \end{aligned} \quad (6)$$

식 (6)에서 X_i 와 이전 중심점 데이터 (MMS_{j-1})와의 평균을 $TotalMMS$ 와 더하여 이 값을 새로운 $TotalMMS$ 로 갱신하고, $rCount$ 값은 1 증가시킨다. 그 다음, 식 (7)에서 저장된 $rCount$ 값이 0이면 X_i 와 MMS_{j-1} 의 평균값을 반환하고, 현재 중심점 데이터 (MMS_i)와 MMS_{j-1} 이 같지 않는 조건이면 $TotalMMS$ 를 $rCount$ 값으로 평균하여 이 값을 MMS_i 와 X_i 에 각각 저장한다.

$$\begin{cases} \text{if } (rCount = 0) \\ \quad \text{then } MMS_j \leftarrow (X_i + MMS_{j-1}) / 2 \\ \quad \quad \quad \text{return } (MMS_j) \\ \text{elif } (MMS_j \neq MMS_{j-1}) \\ \quad \text{then } (MMS_j \leftarrow TotalMMS / rCount) \\ \quad \quad \quad X_i \leftarrow MMS_j \\ \quad \quad \quad \text{repeat } (equation(4) \sim (7)) \\ \text{else then return } (MMS_j) \end{cases} \quad (7)$$

위의 식 (4)~식 (7)는 반복적으로 수행되며, MMS_i 와 MMS_{j-1} 이 같다는 조건에 만족하면 MMS_j 를 반환하고 *MMSA*는 종료된다.

IV. 실험 및 성능 분석

4.1 실험 및 측정 환경

$CAPUH_{MMS}$ 의 성능을 실험하기 위해서는 먼저 손 위치인식 데이터를 측정하여야 하는데 이에 대한 측정 환경과 제안한 $CAPUH_{MMS}$ 의 성능실험을 위한 실

험 환경은 표 1과 같다.

표 1. 손 위치 데이터 측정 및 성능분석 실험 환경
Table 1. Experimental environments for measurement of hands position and performance analysis.

항목	Measurement Environment	Experiment Environment
CPU	Intel (R) Atom(TM) CPU 1.86GHz	Intel (R) Core(TM)2 Duo 3.00GHz
Memory	3.00GB	6.00GB
OS	Xubuntu 12.04LTS	Windows 8.1 Professional 64bit
API & Application	OpenNI + NITE ROS Hydro	.Net C# Framework 4.0

4.2 실험 및 성능분석 방법

4.2.1 손 위치 데이터 측정 방법

손 위치 데이터의 측정은 그림 4와 같이 HRI 시스템에 탑재된 키넥트 센서와의 거리가 160cm 떨어진 위치에서 실시되었다.

본 논문은 양 손의 상하 운동과 좌우 운동의 2가지 시나리오에 대해 제안하는 $CAPUH_{MMS}$ 의 성능을 분석하였다. 그러나 손 위치 데이터의 측정에 있어서 양 손의 제스처가 직선 형태로 이루어지더라도 정확하게 직선이 되지 않는 문제점이 발생하기 때문에, 상하 운동의 경우 양쪽 봉을 잡고 55cm 구역 내에서 상하로, 그리고 좌우 운동의 경우 양쪽 봉에 연결된 끈을 잡고 55cm 구역 내에서 좌우로 각각 반복적으로 실험을 실시하였다.



그림 4. 손 위치 데이터 측정 방법
Fig. 4. Measurement method for position data of user hands position.

4.2.2 제안하는 $CAPUH_{MMS}$ 의 실험 및 성능 분석 방법

제안한 $CAPUH_{MMS}$ 의 성능은 자체 개발한 성능 시뮬레이터를 이용하여 분석되었다. 성능 시뮬레이터에서 양 손 제스처의 상하 및 좌우 운동에 대한 평균 이동 궤적은 그림 5 및 그림 6과 같으며, 그림의 x축, y

축은 각각 손의 (x, y) 위치좌표이다.

그림 5에서 상하 운동에 대한 평균 이동 궤적은 상하 운동에 의해 측정된 상단 데이터들의 평균 위치 좌표와 하단 데이터들의 평균 위치 좌표를 연결한 선으로 표시된다. 왼쪽 선은 오른손에 대한 평균 이동 궤적, 오른쪽 선은 왼손에 대한 평균 이동 궤적을 의미한다.

또한, 그림 6에서 양 손 제스처의 좌우 운동에 대한 평균 이동 궤적은 좌우 운동에 의해 측정된 좌측 데이터들의 평균 위치 좌표와 우측 데이터들의 평균 위치 좌표를 연결한 선으로 표시된다. 위쪽 선은 왼손에 대한 평균 이동 궤적이며, 아래쪽 선이 오른손에 대한 평균 이동 궤적을 의미한다.

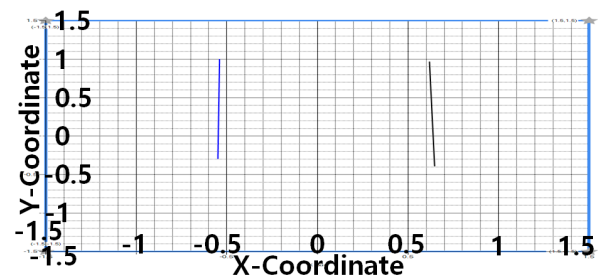


그림 5. 상하 운동에 대한 양손의 평균 이동 궤적
Fig. 5. Average trajectories in up-down movements of hands.

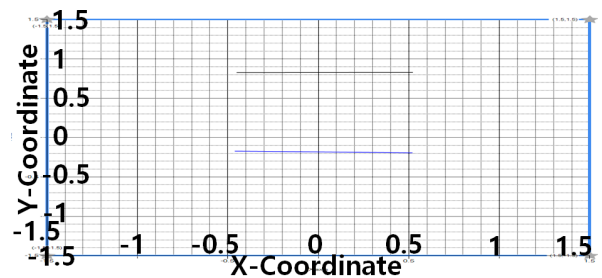


그림 6. 좌우 운동에 대한 양손의 평균 이동 궤적
Fig. 6. Average trajectories in left-right movements of hands.

측정된 (x, y) 위치 데이터와 평균 이동 궤적인 직선의 방정식 $ax + by + c = 0$ 사이의 거리에 대한 궤적 오차 (E_T (Trajectory Error))는 식 (8)과 같이 구해진다. 따라서 제안한 $CAPUH_{MMS}$ 의 상하 및 좌우 운동에 대한 평균 이동 궤적 오차 성능을 Kalman Filter based Algorithm (CA_{KF}) 및 LSM based Algorithm (CA_{LSM})의 성능과 서로 비교하고자 한다.

$$E_T = \frac{|ax_1 + by_1 + c|}{\sqrt{a^2 + b^2}} \quad (8)$$

4.3 실험 결과 및 성능분석

먼저 양손의 상하 및 좌우 운동에 대해 2,000개의 실제 좌표 데이터를 측정 한 후, 그 다음 제안한 알고리즘인 $CA_{PUH_{MMS}}$ 의 성능을 자체 개발한 실시간 성능 시뮬레이터로 다른 보정 기법인 CA_{KF} 및 CA_{LSM} 의 성능과 비교분석하였다.

4.3.1 양손 상하 운동에 대한 결과 및 성능분석

양손의 상하 운동에 대한 이동궤적의 오차율 실험 결과는 그림 7과 같다. 그림에서 2개의 패턴 중 왼쪽의 패턴은 오른손에 대한 보정 결과를, 오른쪽의 패턴은 왼손에 대한 보정 결과를 각각 나타낸 것이다. 왼손의 패턴 4개 (오른쪽 패턴) 중 실제 측정 데이터는 회색, CA_{KF} 에 의한 결과는 노란색, CA_{LSM} 에 의한 결과는 하늘색, 그리고 $CA_{PUH_{MMS}}$ 에 의한 결과는 갈색으로 각각 구분되었다. 그리고 오른손의 패턴 4개 (왼쪽 패턴) 중 실제 측정 데이터는 하늘색, CA_{KF} 에 의한 결과는 초록색, CA_{LSM} 에 의한 결과는 파란색, 그리고 $CA_{PUH_{MMS}}$ 에 의한 결과는 분홍색으로 각각 구분되었다.

왼손 및 오른손의 상하운동에 대한 이동 궤적의 오차율은 그림 7에서 보는 바와 같이 제안한 $CA_{PUH_{MMS}}$ 가 CA_{KF} 또는 CA_{LSM} 보다 더 낮음을 확인할 수 있다.

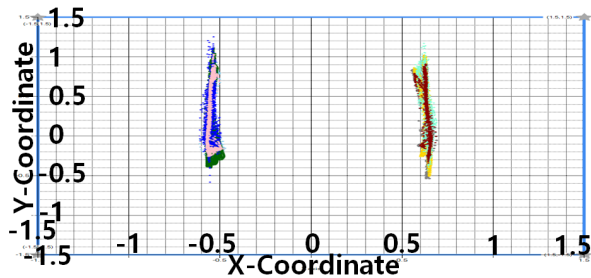


그림 7. 양손 상하 운동에 대한 이동궤적의 오차율
Fig. 7. Average error ratio of the trajectories in up-down movements of hands.

양손 상하 운동의 이동궤적에 대한 평균 오차 비교는 그림 8과 같다. 왼손의 경우 평균 오차는 실제 측정 데이터, CA_{KF} , CA_{LSM} 및 $CA_{PUH_{MMS}}$ 의 경우 1.19cm, 1.17cm, 1.20cm 및 0.99cm으로 각각 측정되었다. 오른손의 경우 평균 오차는 실제 측정 데이터, CA_{KF} , CA_{LSM} 및 $CA_{PUH_{MMS}}$ 의 경우 1.73cm, 1.58cm, 1.63cm 및 1.46cm으로 다른 보정 알고리즘 보다 감소한 것을 볼 수 있다.

양손 상하 운동의 이동궤적에 대한 평균 오차 성능 개선 비율 비교는 그림 9와 같다. 왼손 및 오른손에 대해 CA_{KF} 는 각각 1.76%, 9.18%의 오차 개선율을 나

타냈으나 CA_{LSM} 은 각각 -0.55%, 5.88%으로 왼손의 경우 오히려 보정 효과가 감소한 것을 확인할 수 있었다. 제안한 $CA_{PUH_{MMS}}$ 은 왼손의 경우 20.52%으로 CA_{KF} 및 CA_{LSM} 보다 각각 18.76%, 21.07% 더 높은 평균 오차 개선율을, 오른손의 경우 18.17%으로 CA_{KF} 및 CA_{LSM} 보다 각각 8.99%, 12.29% 더 높은 오차 개선율을 나타낸 것을 알 수 있다.

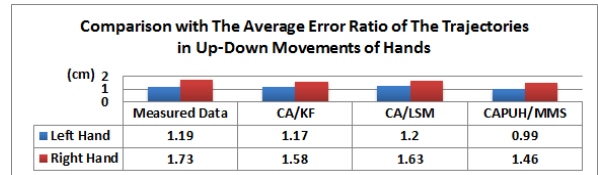


그림 8. 양손 상하 운동의 이동궤적에 대한 평균 오차 비교
Fig. 8. Comparison with the average error ratio of the trajectories in up-down movements of hands.

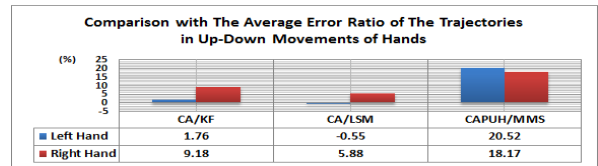


그림 9. 양손 상하 운동의 이동궤적에 대한 평균 오차 성능 개선 비율 비교
Fig. 9. Comparison with the average error improvement ratio of the trajectories in up-down movements of hands.

4.3.2 양손 좌우 운동에 대한 결과 및 성능분석

양손의 상하 운동에 대한 이동궤적의 오차율 실험 결과는 그림 10과 같다. 그림에서 2개의 패턴 중 위쪽의 패턴은 왼손에 대한 보정 결과를, 아래쪽의 패턴은 오른손에 대한 보정 결과를 각각 나타낸 것이다.

왼손의 패턴 4개 (위쪽 패턴) 중 실제 측정 데이터는 회색, CA_{KF} 에 의한 결과는 노란색, CA_{LSM} 에 의한 결과는 하늘색, 그리고 $CA_{PUH_{MMS}}$ 에 의한 결과는 갈색으로 각각 구분되었다. 그리고 오른손의 패턴 4개 (아래쪽 패턴) 중 실제 측정 데이터는 하늘색, CA_{KF} 에 의한 결과는 초록색, CA_{LSM} 에 의한 결과는 파란색, 그리고 $CA_{PUH_{MMS}}$ 에 의한 결과는 분홍색으로 각각 구분되었다.

양손의 좌우운동에 대한 이동궤적 오차율은 그림 10에서 보는 바와 같이 제안한 $CA_{PUH_{MMS}}$ 이 CA_{KF} 또는 CA_{LSM} 보다 더 낮음을 확인하였다. 이 결과는 양손 상하운동에 대한 실험 결과와 매우 유사하다.

양손 좌우 운동의 이동궤적에 대한 평균 오차의 비교는 그림 11과 같다. 왼손의 경우 평균 오차는 실제 측정 데이터, CA_{KF} , CA_{LSM} 및 $CA_{PUH_{MMS}}$ 의 경우 2.76cm, 2.18cm, 2.43cm 및 2.04cm으로 각각 측정되

었다. 오른손의 경우 평균 오차는 실제 측정 데이터, CA_{KF} , CA_{LSM} 및 $CAPUH_{MMS}$ 의 경우 2.22cm, 1.99cm, 2.04cm 및 1.82cm으로 다른 보정 알고리즘 보다 감소한 것을 볼 수 있다.

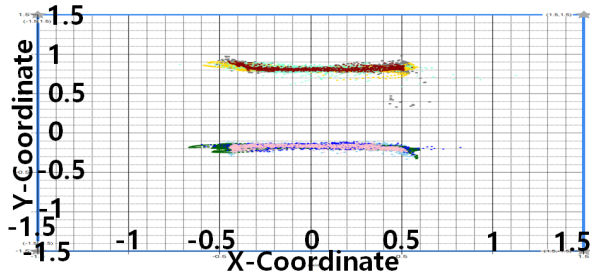


그림 10. 양손 좌우 운동에 대한 이동궤적 오차율
Fig. 10. Average error ratio of the trajectories in left-right movements of hands.

양손 좌우 운동의 이동궤적에 대한 평균 오차 성능 개선 비율 비교는 그림 12와 같다. 왼손 및 오른손에 대해 CA_{KF} 는 각각 26.51%, 11.55%의 오차 개선율을, CA_{LSM} 은 각각 13.50%, 8.95%의 오차 개선율을 나타낸다. 제안한 $CAPUH_{MMS}$ 는 왼손의 경우 35.27%으로 CA_{KF} 및 CA_{LSM} 보다 각각 8.76%, 21.77% 더 높은 오차 개선 비율을, 오른손의 경우 21.80%으로 CA_{KF} 및 CA_{LSM} 보다 각각 10.25%, 12.85% 더 높은 평균 오차 성능개선 비율을 각각 나타낸 것을 알 수 있다.

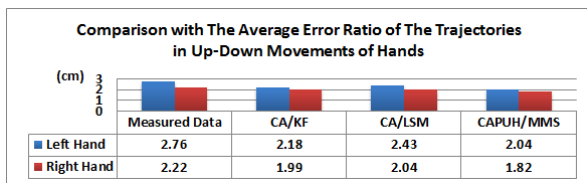


그림 11. 양손 좌우 운동의 이동궤적에 대한 평균 오차 비교
Fig. 11. Comparison with the average error ratio of the trajectories in left-right movements of hands.

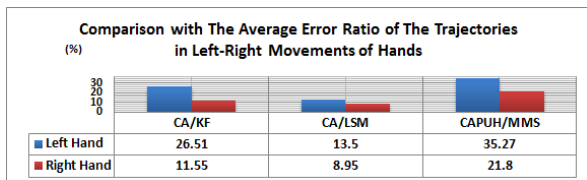


그림 12. 양손 좌우 운동의 이동궤적에 대한 평균 오차 성능개선 비율 비교
Fig. 12. Comparison with the average error improvement ratio of the trajectories in left-right movements of hands.

V. 결 론

본 논문은 키넥트 센서를 장착한 HRI 시스템에서 기존의 사용자 위치정보 인식 성능을 향상하기 위하여 Moving Mean-Shift 기반 사용자 손 위치보정 알고리즘인 $CAPUH_{MMS}$ 를 제안하였다.

성능분석을 위해 양손의 상하 및 좌우 운동에 대해 2,000개의 실제 좌표 데이터를 측정 한 후, 제안한 $CAPUH_{MMS}$ 의 이동궤적 오차율 성능을 자체 개발한 실시간 성능 시뮬레이터로 실험하여 다른 보정 기법인 CA_{KF} 및 CA_{LSM} 와 비교하였다.

$CAPUH_{MMS}$ 는 양손 상하 운동에서 왼손의 경우 20.52%으로 CA_{KF} 및 CA_{LSM} 보다 각각 18.76%, 21.07% 더 높은 평균 오차 개선 비율을, 오른손의 경우 18.17%으로 CA_{KF} 및 CA_{LSM} 보다 각각 8.99%, 12.29% 더 높은 평균 오차 개선 비율을 각각 제시하였다. 또한 $CAPUH_{MMS}$ 는 양손 좌우 운동에서 왼손의 경우 35.27%으로 CA_{KF} 및 CA_{LSM} 보다 각각 8.76%, 21.77% 더 높은 평균 오차 개선 비율을, 오른손의 경우 21.80%으로 CA_{KF} 및 CA_{LSM} 보다 각각 10.25%, 12.85% 더 높은 평균 오차 개선 비율을 각각 나타낸 것을 알 수 있다.

이번 실험에서의 특징은 첫째, 왼손의 상하운동의 경우, CA_{LSM} 은 오히려 실제 측정 데이터 보다 0.55% 오차가 증가하는 경우가 발생하였지만, 제안한 $CAPUH_{MMS}$ 는 20.52%의 평균 오차 개선 비율을 보였다라는 점이다. 두 번째, 양손의 좌우 운동이 상하 운동에 비해 전반적으로 CA_{KF} , CA_{LSM} 및 제안한 $CAPUH_{MMS}$ 의 평균 오차 개선 비율이 더 높게 나왔다는 점이다. 그 이유는 실험에서 양손의 좌우 움직임이 상하 움직임에 비해 다소 불안정했기 때문에 나온 것으로 추정된다.

제안한 $CAPUH_{MMS}$ 는 제스처 인식 및 처리의 성능 향상에 많은 도움을 줄 것으로 예상하고 있다. 향후 제안한 $CAPUH_{MMS}$ 를 HRI 시제품에 실제 적용 한 후 다양한 손의 제스처의 패턴을 통해 보다 정밀한 손 위치를 보정 할 수 있는 알고리즘을 제시 할 예정이다.

References

[1] S. C. Kim and I. C. Park, "A study on the ubiquitous home network interface system by application of user's gesture recognition method," *J. Korean of The Science of Emotion & Sensibility (KJSOS)*, vol 8. no 3, pp. 265-276, Sep. 2005.

[2] F. K. H. Quek, "Unencumbered gestural interaction," *IEEE Multimedia*, vol. 3, pp. 36-47, Dec. 1996.

[3] S. Mitra and T. Acharya, "Gesture recognition: A survey," *IEEE Trans. Systems, Man, and Cybernetics*, Part C: Applications and Reviews, vol. 37, no. 3, pp. 311-324, 2007.

[4] G.R.S. Murthy and R.S. Jadon, "A review of vision based hand gestures recognition," *J. Int'l Information Technol. and Knowledge Management (JIITKM)*, vol. 2, no. 2, pp. 405-410, July-December 2009.

[5] H. J. Kim, *Kinect for windows 1: Implement of the innovative user interfaces of embedded devices (2012)*, Retrieved Feb. 1, 2015, from [http://www.mdstec.com/bbs/press_room/upload/Kinect for Windows_4.pdf](http://www.mdstec.com/bbs/press_room/upload/Kinect%20for%20Windows_4.pdf)

[6] P. Doliotis, et al., "Comparing gesture recognition accuracy using color and depth information," in Proc. *4th Int. Conf. Pervasive Technol. Related to Assistive Environments (PETRA'11)*, no. 20, pp. 1-7, USA, May 2011.

[7] M. Yang, N. Ahuja, and M. Tabb, "Extraction of 2D motion trajectories and its application to hand gesture recognition," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 24, no. 8, pp. 1061-1074, Aug. 2002.

[8] E. J. Holden, G. Lee, and R. Owens, "Australian sign language recognition," *J. Machine Vision and Appl.*, vol. 16, no. 5, pp. 312-320, 2005.

[9] S. Y. Cho, et al., "Hand gesture recognition from kinect sensor data," *J. Korea Broadcast Eng. (KJBE)*, vol. 17, no 3, pp. 447-458, May 2012.

[10] A. R. Kim and S. Y. Rhee, "Motion control of a mobile robot using natural hand gesture," *J. The Korean Inst. Intell. Syst. (KJIS)*, vol. 24, no 1, pp. 64-70, Feb. 2014.

[11] G. Welch and G. Bishop, *An Introduction to the Kalman Filter* (2006), http://www.cs.unc.edu/~welch/media/pdf/kalman_intro.pdf

[12] T. W. Kim, "A study on the localization compensation algorithm using Mean kShift/Kalman Filter in random walk/waypoint mobility model environment," Master's Thesis, The Graduate

School of Tongmyong University, Feb. 2014.

[13] Konstantinos G. Derpanis, *Mean shift clustering*, *Computer Vision Notes* (2005), http://www.cse.yorku.ca/~kosta/CompVis_Notes/mean_shift.pdf/

[14] M. K. Jung and D. M. Lee, "Performance analysis of the localization compensation algorithm for moving objects using the least-squares method," *J. KICS*, vol. 39C, no. 1, pp. 9-16, Jan. 2014.

김 태 완 (Tae-Wan Kim)



2012년 2월 : 동명대학교 컴퓨터공학과 (학사)
 2014년 2월 : 동명대학교 컴퓨터미디어공학과 (석사)
 2014년 3월~현재 : 동명대학교 컴퓨터미디어공학과 (박사과정)
 <관심분야> 센서네트워크, 임베디드, 위치추정시스템, 영상처리

권 순 량 (Soon-Ryang Kwon)



1982년 : 동아대학교 전자공학과 졸업 (학사)
 1984년 : 부산대학교 전자공학과 졸업 (석사)
 1999년 : 충남대학교 전자공학과 졸업 (박사)
 1984년~1999년 : 한국전자통신연구원 책임연구원

1999년-현재 동명대학교 전자공학과 교수
 <관심분야> RTLS, RFID/USN, 홈네트워크, 이동통신시스템, 무선 가시광 통신

이 동 명 (Dong Myung Lee)



1982년 2월 : 숭실대학교 전자계산학과 (학사)
 1990년 8월 : 숭실대학교 전산공학과 (석사)
 1997년 8월 : 숭실대학교 전자계산학과 (박사)
 1982년 3월~2000년 2월 : 한국전자통신연구원 책임연구원

2000년 3월~현재 : 동명대학교 컴퓨터공학과 교수
 <관심분야> 이동통신시스템, 센서네트워크, 위치추정 시스템