

졸업작품 연구보고서

다중 좌표 추적 기반
모션인식 시스템 개발

목원대학교 공과대학

지능로봇공학과

이 준 엽

2017년 12월

다중 좌표 추적 기반 모션인식 시스템 개발

지도교수 양 정 연

이 연구보고서를 졸업작품 연구보고서로 제출함.

2017년 12월

목원대학교 공과대학
지능로봇공학과
이 준 엽

이준엽의 졸업작품 연구보고서를 인준함.

주심위원 _____(인)

부심위원 _____(인)

부심위원 _____(인)

2017년 12월
목원대학교 공과대학
지능로봇공학과

국 문 요 약

다중 좌표 추적 기반 모션인식 시스템 개발

이 준 엽

목원대학교 공과대학 지능로봇공학과

본 연구에서는 이미 많은 사람들이 이용하고 있고, 흔히 구할 수 있는 USB Camera로부터 받아온 영상을 Computer Vision을 이용하여 많은 사람들이 간편하고, 저렴하게 이용할 수 있는 모션인식 시스템을 개발한다.

기본적인 시스템 구성은 IR LED가 장착된 하드웨어를 설계하여 USB Camera로 영상을 받아와 다중좌표 추출 기술을 이용하여 해당 좌표 및 움직임을 추적하여 여러 가지 모션을 상황에 맞게 설정 가능한 모션인식 기술을 활성화 한다.

목 차

국문 요약

1. 서 론	1
2. 좌표 추적 기반 모션인식	3
2.1 기존의 모션인식 기술	4
2.2 좌표 추적 알고리즘	7
3. 모션인식 플랫폼 설계	15
3.1 USB Camera를 활용한 수신부 설계	15
3.2 IR LED가 장착된 송신부 설계	18
4. Computer Vision을 이용한 다중좌표 추적 시스템 ...	25
5. 모션인식 시스템 구현	32
6. 실험 및 결과 고찰	35
7. 결 론	44
참고 문헌	45

그림 목 차

그림 2-1 Kalman Filter 계산 과정	6
그림 2-2 Reduce Covariance	9
그림 3-1 Solidworks 이용한 반지형 웨어러블 플랫폼 설계	12
그림 3-2 Ring Size Control	15
그림 4-1 모션인식 컨트롤러의 시스템 구성도	16
그림 4-2 시작점 좌표 초기화	18
그림 4-3 Hungarian Algorithm 적용	20
그림 6-1 거리에 따른 반지름 그래프	26
그림 6-2 IR LED의 인식 범위(각도)	27
그림 6-3 오른손 Tracking Graph	29
그림 6-4 왼손 Tracking Graph	30
그림 6-5 Tracking Graph (2D)	32
그림 6-6 Tracking Graph (3D)	33

사 진 목 차

사진 2-1 광학식 모션 캡처 시스템을 활용한 실제 사람 동작 녹화	4
사진 2-2 다양한 모션인식 디바이스	4
사진 3-1 Camera Modification	11
사진 3-2 일반 USB Camera 영상 / Negative Film 부착 후 영상	12
사진 3-3 웨어러블 플랫폼 완성품	13
사진 3-4 LR41	13
사진 3-5 IR LED - 3W 940nm LED (HBL0307)	14
사진 3-6 웨어러블 플랫폼의 착용 예	14
사진 3-7 Ring 사이즈 조절	15
사진 4-1 다중 좌표 추출을 위한 전처리 과정	17
사진 5-1 Init Position	22
사진 5-2 Motion Right Click / Left Click	23
사진 5-3 Motion Scroll Up / Scroll Down	24
사진 6-1 Matching Test	34
사진 6-2 추적 시스템에 대한 확인 영상	35
사진 6-3 양손이 겹치는 경우의 추적 확인 영상	36
사진 6-4 손의 회전에 따른 추적 확인 영상	37

표 목 차

<표 1> LR41 Specification	13
<표 2> IR LED Specification	14
<표 3> Computer Vision 개발 환경	17
<표 4> FPS Check	27

1. 서 론

INDUSTRY 4.0, 사물과 사물이 인터넷이 연결되는 사물인터넷(IoT, Internet of Things)과 빅 데이터, 인공지능이 더해지면서 또 하나의 새로운 산업혁명, 이른바 4차 산업혁명이 세간의 화두이다. 세상의 모든 생물과 사물이 인터넷으로 연결돼 산출되는 방대한 데이터를 바탕으로, 인공지능을 통해 물리적 세상과 사이버세상을 함께 연동하여 최적으로 관리하는 시대가 오고 있다[1].

4차 산업혁명이 시작되면서 영화에서만 볼 수 있던 다양한 기술들이 보다 현실이 되어 나타났다. 특히 비디오 영상분석에 대한 관심의 증대와 더불어 영상내의 특정 물체에 대한 추적 연구가 활발해지고 있다. 이에 영상처리 기술은 이미 사용되지 않는 곳이 없다 해도 과언이 아닐 정도로 일상생활 속에 녹아들어 있다. 그러나 실세계 영상의 배경색상, 조명강도, 시각변형과 더불어 카메라와 이동물체의 거리, 그리고 카메라 지터 효과로 인해 영상 내의 회색 반전과 영상 내 물체의 크기와 모양의 무작위적인 변형은 목표물의 검출 및 추적을 어렵게 한다[2].

추적을 위한 여러 가지 알고리즘 들이 존재하지만 정확한 추적이 이루어지는 장점을 가지는 알고리즘들은 계산량이 많고, 다량의 데이터를 축적해야한다는 단점 때문에 실시간으로 처리하는 추적시스템을 만들기에는 부적합하다[3].

이에 실세계의 영상대신 적외선영상을 이용하여 노이즈를 줄이고 데이터 량을 줄이는 방법을 취할 수 있다. 지금까지 적외선 영상에서의 소형표적검출은 그 중요성 때문에 많은 연구가 이루어져 왔다. 이러한 연구들은 연속영상에서 잡음의 불규

칙성을 이용하기 위해 Kalman Filter[4]를 사용한다. 이 알고리즘은 뛰어난 실시간 추적성능을 보인다.

해당 알고리즘이 뛰어난 실시간 추적성능을 보이지만 동일한 물체가 다수 존재하는 경우에는 추적하는데 어려움이 있다. 이러한 문제는 Hungarian Algorithm[5-8]을 이용하여 데이터의 다음 값을 Optimal한 값으로 수렴시켜주는 방법을 이용하였다.

이러한 영상처리를 이용한 물체추적 알고리즘은 다양한 방법들이 존재하지만 4차 산업혁명이 화두가 되면서 사람과 컴퓨터간의 상호작용(HCI, Human Computer Interaction)을 필요로 한다. 모션인식 기술은 사람과 컴퓨터를 이어주는 가장 직관적인 인터페이스라고 할 수 있다.

본 연구에서는 사람의 손 모션을 손 끝단에 달린 IR LED에 대한 적외선 연속 영상에서 Kalman Filter와 Hungarian Algorithm을 이용하여 동일한 다중 표적의 뒤섞임 속에서도 추적이 가능한 다중 소형 표적을 추적하는 알고리즘을 제안한다.

제안한 알고리즘을 이용하여 많은 사람들이 모션인식기술을 쉽게 접할 수 있도록 하는 인터페이스를 추가적으로 제안하며 이는 가정, 엔터테인먼트, 산업 현장 등 다양한 분야에서 응용될 수 있다.

본 논문은 다음과 같은 순서로 구성된다. 2장에서는 기존의 모션인식 기술과, 물체추적을 위한 좌표추적 알고리즘을 설명하고, 3장에서는 하드웨어, 4장에서는 다중 좌표 추적 시스템, 5장에서는 모션인식 시스템에 대한 제안을 하며, 6장에서는 실험 및 결과 고찰을 통해 성능을 평가하고, 마지막 7장에서는 앞의 내용을 바탕으로 결론을 맺을 것이다.

2. 좌표 추적 기반 모션인식

기존의 모션인식 기술들은 적외선을 쏘아 그 반사파를 적외선 카메라로 감지하여 물체의 위치좌표를 추출한다. 적외선 영상을 이용함으로써 조명과 같은 환경에 대해 강인한 인식기술을 제공한다[9].

물체의 좌표 추출 및 추적에 관한연구는 Vision분야에서 활발히 이루어지고 있다. Camera로부터 받아온 영상으로부터 물체를 인식하는 방법에는 색상, 모양, 질감 등의 영상정보를 사용하는 내용기반 검색기술[10]과 크기, 회전, 변화, 노이즈에도 안정감 있게 Key Point를 추출해 물체 인식이 가능한 특징점 기반 알고리즘인 SIFT, SURF와 같은 Machine Vision등이 있다[11]. 그러나 이러한 기술들은 빛의 변화에 대해 민감하거나, 계산 시간이 오래 걸린다는 단점이 있다.

본 연구에서는 특징점이 다수 존재하는 시스템이 아닌 다수의 동일 물체를 실시간 추적하기 위한 알고리즘을 필요로 한다. 따라서 최대한 간단하고 빠른 계산이 가능한 추적방식을 필요로 한다.

좌표를 추적하기 위한 방식에는 노이즈에 강하고 측정데이터를 기반으로 하여 움직임 예측이 가능한 Kalman Filter[4]가 있으며, Optimal한 Label 갱신을 위한 방식으로는 Hungarian Algorithm[12]이 있다.

2.1 기존의 모션인식 기술

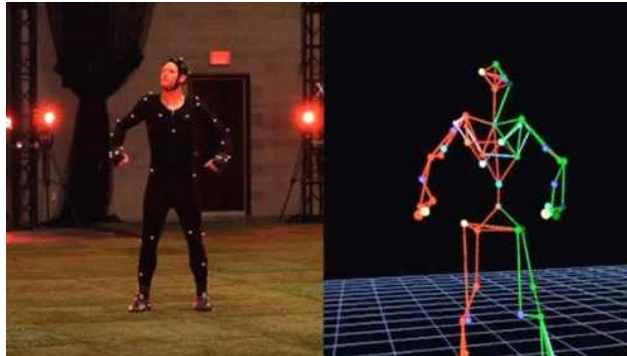


사진 2-1 광학식 모션 캡처 시스템을
활용한 실제 사람 동작 녹화

광학식 모션 캡처(Optimal Motion Capture) 시스템은 사진 2-1 [13]과 같이 모션이 기록될 사용자의 몸에 부착된 수십 개의 작은 구형 특수 마커(Marker)를 사용자의 주변에 둘러싸인 적외선 카메라들이 실시간으로 인식하여 사용자의 움직임을 실시간으로 분석하여 모션을 얻어내는 캡처 시스템[14]으로 Light Camera(적외선 방출 LED 부착 Camera)로 마커의 역반사 효과를 높여주고, 반사점(Reflective Dots)을 이용하여 3차원 공간에서 Joint의 위치를 결정한다[15].

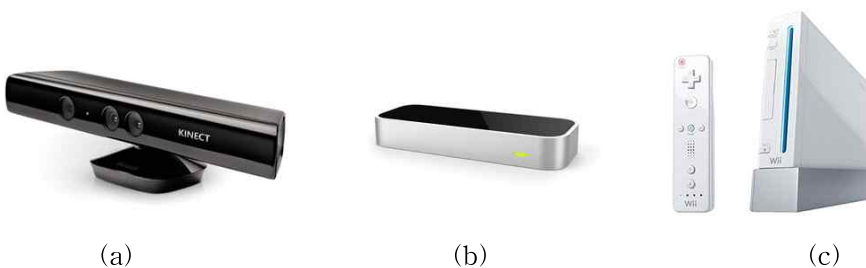


사진 2-2 다양한 모션인식 플랫폼

마이크로소프트(Microsoft)사의 게임용 동작인식기기인 키넥트(Kinect) 센서는 구조광 방식의 깊이 카메라이다. 적외선 레이저를 이용해 패턴을 조사하는 IR 방사체(Projector)와 적외선 영역의 영상을 촬영하는 IR 카메라(CMOS), 일반적인 웹캠과 같은 기능을 하는 RGB 카메라가 IR 방사체와 IR 카메라 사이에 배치되어있다[16]. 기본적인 원리는 IR 방사체에서 조사된 레이저 패턴이 물체에 투영되어 반사되면 IR 카메라가 반사된 적외선을 받아 계산하여 Depth Map을 만들어 물체와의 거리를 구하게 된다[17].

3차원 제스처 인식장치인 Leap Motion은 내부에 2개의 IR 카메라(인식 모듈), 3개의 적외선 LED(광원) 등으로 이루어져 있다. IR 카메라가 영상정보를 받아들이면 x, y, z축을 결정하게 되고, 화면상의 특징점을 추출해 이 점이 움직이는 궤도를 연속으로 추적한 뒤, 추적된 점들을 분석해 움직임을 인식하게 된다[18-20].

닌텐도 Wii는 일반적인 모션캡처 방식을 역으로 이용했다. 기존에는 동작을 인식하기 위해서 움직이는 물체에 LED를 부착한 뒤 고정된 카메라가 LED의 움직임을 인식했다. Wii는 LED를 센서바(Wii Sensor Bar)에 고정시키고 사용자가 들고 움직이는 리모컨에 IR 카메라를 달아 LED의 위치정보를 삼각측량법으로 분석하여 사용자의 움직임을 알 수 있다[21].

2.2 좌표 추적 알고리즘

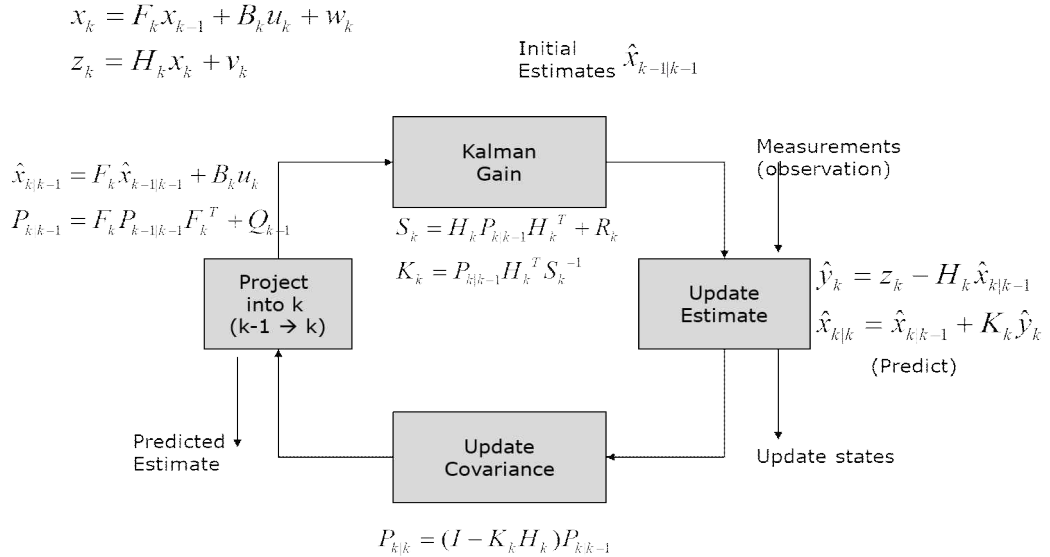


그림 2-1 Kalman Filter의 계산 과정

($k|k-1$: 추정치, $k-1|k-1$: 측정치)

칼만 필터(Kalman Filter)는 잡음이 포함되어있는 선형 역학계의 상태를 추적하는 재귀필터로 컴퓨터비전, 로봇공학, 레이더 등의 여러 분야에서 사용되며 많은 경우에 매우 효율적인 성능을 보여준다[22].

칼만 필터의 기본적인 목표는 부정확한 측정값 (Observation/Measurement) 으로부터 오차를 최소로 하는 추정치(Estimation)를 반복적으로 추정하는 방법으로 잡음이 가우시안 잡음인 경우에 최적의 추정을 할 수 있다. 실세계에는 정확한 값이 존재하지 않고 정확한 값을 기준으로 오차를 포함하는 값들만이 존재하는데 이를 칼만 필터의 수식을 반복 계산하다보면 정확한 값이 유추되는 것이다.

칼만 필터는 잡음에 의해 간섭받는 선형 동적 시스템(Linear Dynamics System)에서의 상태 벡터(State Vector) x_k 의 최적의 추정치 \hat{x}_k 를 구하기 위한 순환적인(Recursive) 알고리즘이다. 여기서 k 는 시간을 나타낸다.

칼만 필터링은 크게 예측(Prediction), 측정(Measurement), 보정(Update)의 세 단계로 나뉜다.

예측 단계에서는 다음 상태의 상태 벡터 x_k 과 상태 추정 오차의 공분산 행렬 P_k 을 예측하는 단계이다.

상태 추정 오차의 공분산 행렬을 수식으로 표현하면 아래와 같다.

$$P_{k|k-1} = E[(x_{k-1} - \hat{x}_{k-1})(x_{k-1} - \hat{x}_{k-1})^T] \quad (2-1)$$

예측 단계에서 이루어지는 상태벡터와 상태추정오차의 공분산 행렬의 예측을 나타내는 두 식은 다음과 같이 주어진다.

$$\hat{x}_{k|k-1} = F_k \hat{x}_{k-1|k-1} + B_k u_k + w_k \quad (2-2)$$

$$P_{k|k-1} = F_k P_{k-1|k-1} F_k^T + Q_{k-1} \quad (2-3)$$

위 식에서 w_k 는 시스템의 모델 잡음(Model Noise)이고, Q_k 는 w_k 의 공분산 행렬($Q_k = E(w_k^2)$)이다.

측정 단계는 아래와 같은 수식(Observation Equation)으로 표현되어 질 수 있다.

$$z_k = H_k x_k + v_k \quad (2-4)$$

z_k 는 측정 벡터, H 는 상태 벡터와 측정 벡터 사이의 관계를 나타내는 행렬(Observation Matrix), v_k 는 측정 오차를 나타내며, 모델 잡음 w_k 와 측정 오차 v_k 는 비상관(Uncorrelated)되었다고 가정한다.

마지막으로 수정단계에서는 측정단계에서 구해진 측정값을 바탕으로 상태벡터와 상태추정오차를 새로운 값으로 수정한다. 수식으로 표현하면 다음과 같다.

$$S_k = H_k P_{k|k-1} H_k^T + R_k \quad (2-5)$$

$$K_k = P_{k|k-1} H_k^T S_k^{-1} \quad (2-6)$$

$$P_{k|k} = (I - K_k H_k) P_{k|k-1} \quad (2-7)$$

$$\hat{y}_k = z_k - H_k \hat{x}_{k|k-1} \quad (2-8)$$

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k \hat{y}_k \quad (2-9)$$

위 식에서 R_k 는 측정오차 v_k 의 공분산행렬($R_k = E(v_k^2)$)을 나타내고, K_k 는 칼만 이득(Kalman Gain)을 나타낸다.

칼만 이득의 역할은 상태벡터 x_k 를 수정할 때, 측정값과 추정값 사이의 차이(Innovation)에 적절한 가중치(Weighting Factor)를 주는 역할을 한다.

결국 Kalman Filter는 측정값의 오차를 0에 가깝게 하는 것을 목표로 하기위해 측정값을 예측값으로 계산해주는 과정으로서 Kalman Gain k 를 이용하여 앞의 과정들을 반복하여 위의 Covariance항을 줄여나간다.

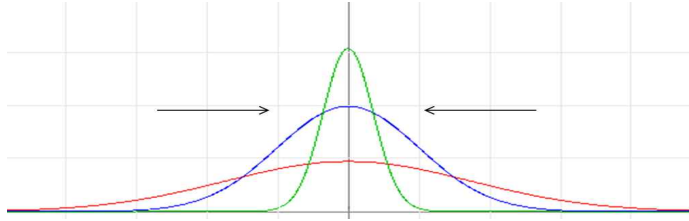


그림 2-4 Reduce Covariance

Hungarian Algorithm[5-8]은 다항시간(Polynomial Time)내에서 최적의 자원 할당을 할 수 있도록 하는 알고리즘으로 정합의 문제(Assignment Problem)에서 주로 이용된다.

정합의 문제에서 $n \times n$ Cost 행렬에 대해 Brute Force 방식으로 풀면 총 경우의 수가 $n!$ 로 $O(n!)$ 의 어마어마한 시간 복잡도(Time Complexity)를 갖게 된다. Hungarian Algorithm은 이러한 문제를 시간 복잡도 $O(n^3)$ 를 갖도록 개선된 알고리즘이다[5-8].

Hungarian Algorithm은 각 점에 해당되는 Label을 변경하면서 최적의 간선들을 찾는 과정으로 간선의 분배가 중복되지 않게 하는 것이 핵심이다.

간선의 중복 선택된 경우를 나타내는 Alternating Tree의 경우에, 간선의 기회비용을 판단하고 재조정하는 수식은 다음과 같다.

$$\Delta = \min_{x \in S, y \in Y/T} (l(x) + l(y) - w(x, y)) \quad (2-9)$$

$$l'(v) = \begin{cases} l(v) - \Delta, & v \in S \\ l(v) + \Delta, & v \in T \\ l(v), & \text{Otherwise} \end{cases} \quad (2-10)$$

위 식에서 Δ 는 다른 간선보다 최소가 되는 간선, $l(x)$ 은

$Label\ x$, $l(y)$ 는 $Label\ y$, $w(x, y)$ 는 x, y 사이 간선의 가중치 (Weight)값을 나타내며, S 집합은 x 점들 중에 Tree에 속한 점, T 집합은 y 점들 중에 Tree에 속한 점들을 나타낸다.

간선이 Alternating Tree를 만들지 않는다면 그 간선은 그 시점에서 최댓값 매칭을 보장한다. 반드시 전체 매칭 입장에서 최적값을 보장하는 간선은 아니지만, 이 Alternating Tree를 만들지 않는 간선, 즉 선택한 간선의 y 값이 Matched되지 않은 Exposed (Free, Unmatched) 한 값이라면 이 간선을 Augmenting Path라고 한다.

Hungarian Algorithm은 Augmenting Path를 n 번 찾으면 끝이 난다. 하지만 이 Augmenting Path는 각 과정에서 선택되었다고 해서 반드시 그 간선이 고정되는 것이 아니고 알고리즘을 수행하면서 계속 변할 수 있는 값으로 중요한 것은 어떤 간선이 선택되었는가가 아니라 몇 번 선택 되었는가 이며, 이러한 방식을 통해 Optimal한 다음 Step을 계산할 수 있다.

3. 모션인식 플랫폼 설계 및 제작

본 연구에서는 사람의 손에 대한 모션을 인식하는 시스템을 구현하기 위해, 수신부(수광부)는 일반적으로 사용하는 USB Camera를 개조하여 사용하였으며, 송신부(발광부)는 IR LED가 장착된 반지형 웨어러블 플랫폼을 설계하였다.

3.1 USB 카메라를 활용한 수신부 제작

본 연구에서는 일반적인 USB Camera를 IR LED 수신에 가능한 모션인식 플랫폼으로서 활용하기 위해 IR Cutter Filter를 제거하고 Negative Film을 부착하여 적외선 카메라로서 이용하였다.



사진 3-1 Camera Modification

사진 3-1의 과정은 USB Camera 렌즈부의 IR Cutter Filter를 제거하고 Negative Film을 부착하는 과정이다. Negative Film은 아날로그 카메라 필름의 검정색 부분으로서 빛을 차단해주는 효과가 있어 적외선 영상은 강하게 받아와 지지만 나머지 영상은 제대로 받아와지지 않도록 해준다. 이를 통해 사진 3-2와 같이 적외선 카메라처럼 IR LED 수신에 가능하도록 하였다.



사진 3-2 일반 USB Camera 영상 /
Negative Film 부착 후 영상

3.2 IR LED가 장착된 송신부 설계

모션을 보다 정확하게 받아오기 위해 손가락 끝에 IR LED를 부착할 필요가 있다고 판단되어 보다 편한 장착을 위해 반지형 플랫폼을 제작하였다.

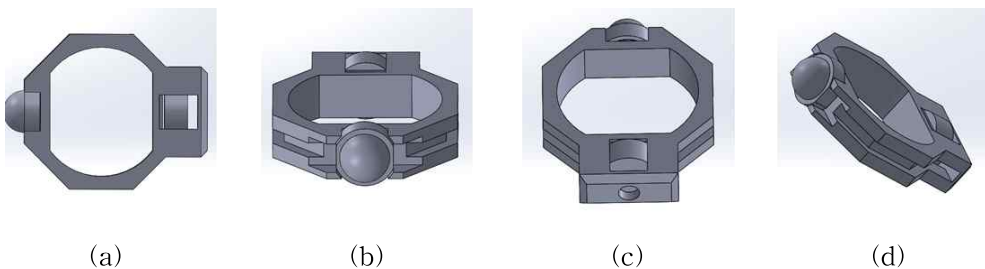


그림 3-1 Solidworks 이용한 반지형 웨어러블 플랫폼 설계

그림 3-1은 반지형 웨어러블 플랫폼의 전체적인 형태이다. (a)는 윗면, (b)는 앞면, (c)는 뒷면, (d)는 대각선면의 형태를 나타낸다.



사진 3-3 웨어러블 플랫폼 완성품

사진 3-3은 반지형 웨어러블 플랫폼의 설계를 바탕으로 하여 ABS 필라멘트를 이용하여 3D프린터로 출력한 플랫폼의 실제사진이다. 앞의 홈에는 IR LED를 부착하고, 옆면의 홈은 전선이 통과되어 정리되며, 뒷면에는 배터리를 끼워 넣을 수 있도록 제작하였다.



사진 3-4 LR41

전원으로 이용한 배터리는 소형 가전제품이나 휴대용기기, 미니 시계 등에 많이 사용되는 사진 3-4의 직경 7.9mm의 1.5V 무수은 건전지인 LR41을 이용하였다. 자세한 사양은 <표 1>에서 확인 할 수 있다.

<표 1> LR41 Specification

IEC	Japan	USA	HKG	Nominal Voltage	Nominal Capacity	Dimension(DIA)	Height	Weight
				(V)	(mAh)	(mm)	(mm)	(g)
LR41	LR41	384	AG3	1.5V	35	7.75-7.85	3.40-3.60	0.61



사진 3-5 IR LED - 3W
940nm LED (HBL0307)

사진 3-5은 반지형 웨어러블 플랫폼에 장착된 IR LED이며, <표 2>은 그에 대한 사양이다. IR LED 선정에 있어서는 강한 빛과 시야각을 중점적으로 보고 결정하였다.

<표 2> IR LED Specification

Wavelength	940nm
Viewing Angle	140°
DC Forward Voltage	1.4V~1.6V
DC Forward Current	700mA
Reverse Voltage	5.0V
Life Span	50,000 Hours
Lead Soldering Temp	260℃ for 5 second



(a)



(b)

사진 3-6 웨어러블 플랫폼의 착용 예

사진 3-6은 웨어러블 플랫폼의 착용 예이다. (a)는 잘못된 착용 예이고, (b)가 올바른 착용 예이다. 여기서 보면 손가락 별로 사이즈가 달라서 같은 사이즈를 썼을 경우에는 (a)와 같이 나올 수밖에 없음을 알 수 있다.

사람과 손가락에 따라 손가락의 사이즈는 다를 수밖에 없다 그러나 모든 사람의 사이즈에 맞춰 제작하는 것은 무리가 있다. 이에 성인 남성의 사이즈에 맞춰 큰 외형을 제작한 후 그림 3-4와 같이 사이즈 조절을 위한 링을 추가로 제작하였다.



그림 3-2 Ring Size Control

링의 사이즈를 다양하게 제작하여 손가락이 작은 사람도 맞을 수 있게 제작하였다.

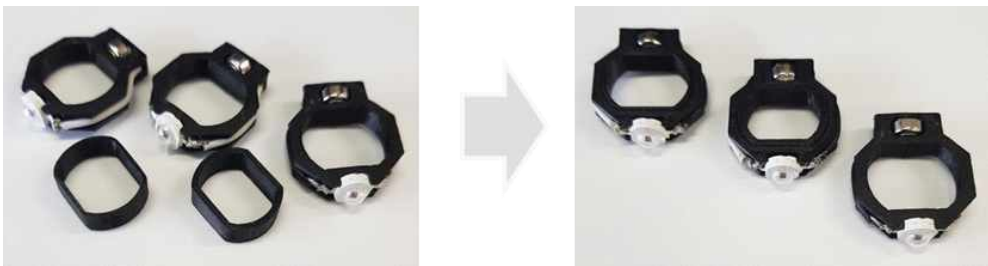


사진 3-7 Ring 사이즈 조절

위의 사진 3-7에서 볼 수 있듯이 따로 출력된 플랫폼을 외형 내부에 끼우기만 하면 사이즈를 조절할 수 있다.

4. Computer Vision을 이용한 다중좌표 추적 시스템

본 연구는 Computer Vision을 이용해 다중좌표를 추적하는 시스템을 구현하여 각 좌표들의 위치에 따라 모션을 인식하도록 하는 시스템이다. IR LED의 빛을 읽는 시스템으로서 구현에 앞서 원하는 결과를 도출하기 위한 시스템 구성도를 제안한다.

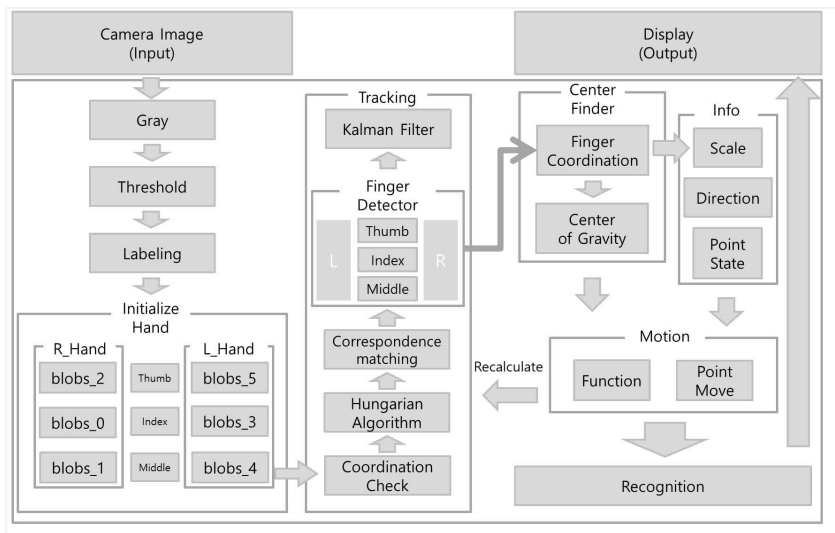


그림 4-1 모션인식 컨트롤러의 시스템 구성도

위의 그림 4-1은 시스템 구성도로서 모션인식 시스템의 전체적인 구성을 나타낸 것이다. 이번 장에서는 알고리즘의 시작 부분부터 Tracking 부분까지 즉, USB Camera를 이용하여 받아온 영상을 다중 좌표 추출, 다중좌표 추적하는 부분까지만을 작성하도록 한다.

본 연구의 Computer Vision의 개발 환경은 <표 3>와 같다.

<표 3> Computer Vision 개발 환경

항목	사양
개발OS	Microsoft Windows 10
프로그램 언어 개발 툴	Microsoft Visual Studio 2010
영상처리 개발 툴	Open CV 2.3.1
웹캠 사양	Logitech HD PRO C920R

본 연구는 USB Camera를 통해 받아온 오른손/왼손의 엄지, 검지, 중지의 총 6개의 IR LED의 빛을 받아와 영상처리를 통해 각 손가락의 좌표를 추출하여 추적하는 시스템이다.

각 손가락에서 나온 빛의 각 좌표들을 추출하기 위한 방법은 여러 가지 있을 수 있지만 본 연구에서는 Labeling을 이용하여 각 점의 좌표를 추출하였다.



사진 4-1 다중 좌표 추출을 위한 전처리 과정

사진 4-1의 과정을 거쳐 이진화 된 이미지 상에서 Labeling을 이용하여 각 점들의 위치를 파악한다. 각 점들의 영역을 확인하고 해당 영역에서의 중점을 찾는다.

$$Center(x_n, y_n) = \begin{cases} x_n = blobs_n.x + blobs_n.width/2 \\ y_n = blobs_n.y + blobs_n.height/2 \end{cases} \quad (4-1)$$

본 연구는 2D Vision상에서 점들의 중점 좌표를 찾는다. 위의 식(4-1)에서 x_n 은 n 번째 점의 x 좌표, y_n 은 n 번째 점의 y 좌표를 나타내며 $blobs_n$ 은 n 번째 blob영역을 나타낸다.

본 연구의 시스템은 손 자체를 인식하는 시스템이 아니다. 따라서 영상에 손이 찍힌 순간 어느 점이 어느 손가락인지 알 수 있는 방법이 없다. 그러므로 영상에 처음 손을 인식 시킬 때에 위치를 초기화 시켜 주어야한다.

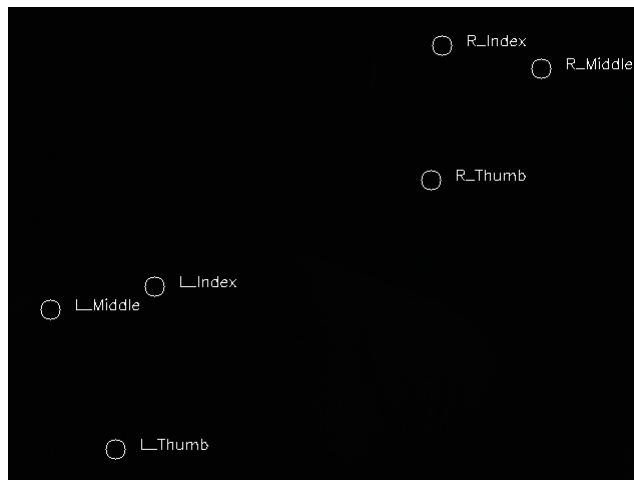


그림 4-2 시작점 좌표 초기화

여러 번의 실험에 의해 가장 편한 초기의 위치를 위의 그림 4-2과 같이 설정하였다. 위의 위치에 빛을 맞추어 초기설정을 해주어야 초기 위치에 대한 정보를 Tracking System에 넣어 줄 수 있다.

초기화된 양손의 엄지, 검지, 중지의 좌표를 Kalman Filter의 초기값으로 대입하고, Kalman Filter의 과정을 반복한다.

Kalman Filter에는 Correspondence를 알고 있는 Known Correspondence, 모르고 있는 Unknown Correspondence의 두

가지가 있다.

본 연구는 각 손가락의 점이 영상으로 보기에 모두 동일한 사이즈의 원형의 빛으로만 받아와지는 시스템이다. 따라서 Correspondence를 모르는 Unknown Correspondence에 해당하는데 Known Correspondence의 경우는 계산식이 실시간에 적합 하다고 볼 수 있으나 Unknown Correspondence의 경우 하나의 좌표 당 x, y 의 2개의 값이 존재하므로 2×2 행렬의 계산식이 필요로 된다.

본 연구에서는 오른손, 왼손의 각각 엄지, 검지, 중지의 좌표가 존재한다. 이는 총 6개의 점의 x, y 값을 계산 해주기 위해 12×12 의 행렬 계산을 필요로 하여 시스템이 매우 느려져 실시간 처리에 적합 하지 않은 상황이 생긴다.

따라서 Kalman Filter를 진행하는 과정 사이에 다음 Step의 좌표를 정하기 위한 Correspondence Matching을 위한 알고리즘을 따로 사용하였다.

Unknown Correspondence의 계산식에 비해 본 연구에서 사용할 Hungarian Algorithm은 6개의 점에 대해서 6×6 의 행렬 계산으로 끝낼 수 있다. 관련 연구에서도 살펴봤듯이 본 알고리즘은 Prior Step과 Next Step간의 Cost를 최소화 하는 Optimal한 값을 찾아내는 방식이다.

본 연구에서 추적하고자 하는 물체를 인식하기 위해서 사용된 특징은 물체의 x 좌표 $State.x$, 물체의 y 좌표 $State.y$ 이다. 물체의 크기는 동일하다고 가정하기에 이 두 가지 특징 값이 라벨(Label) 번호 n 을 가지는 물체에 대한 특징 벡터를 이루게 되는데, 식으로 표시하면 다음과 같다.

$$f_n = (State.x_n, State.y_n) \quad (4-2)$$

탐색 창 내에 2개 이상의 물체가 있을 때, 이 전 프레임에서 기억된 물체의 특징과 탐색 창 내에서 발견된 물체들의 특징을 비교해서 가장 짧은 유클리디안 거리(Euclidean Distance)를 가지는 물체를 추적하고자 하는 물체로 인식하게 된다. 이전 프레임에서 저장된 추적하고 있는 물체의 특징을 다음과 같이 나타낸다.

$$f = (State_{old}.x, State_{old}.y) \quad (4-3)$$

이전 프레임과 현재 프레임의 물체간의 거리는 다음과 같이 나타낼 수 있다.

$$D = \sqrt{(State.x_n - State_{old}.x)^2 + (State.y_n - State_{old}.y)^2} \quad (4-4)$$

위의 식에서 D 는 이전 프레임과 현재 프레임의 물체간의 거리를 나타낸다. 최적의 라벨을 찾기 위해서는 이전 프레임과 현재 프레임간의 모든 프레임간의 거리를 구하고 최솟값을 구해야 한다.

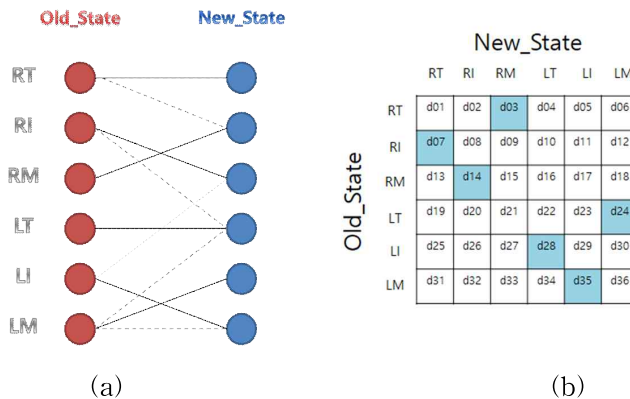


그림 4-3 Hungarian Algorithm 적용

그 최적 해를 구하기 위해 Cost를 최소로 하는 값을 빠르게 계산해내기 위해서 Hungarian Algorithm을 이용한다.

그림 4-3의 (a)와 같이 Old_State에서 New_State로 이어지는 Optimal한 값은 한가지 밖에 존재 하지 않으며, (b)의 행렬의 Rows는 Old_State, Columns는 New_State로 행렬의 원소들은 위의 식 4-4를 이용해서 구한 Old_State와 New_State간의 거리 값 D 이다.

결과적으로 현재 프레임의 탐색 창에서 발견된 물체들 중 추적하고 있는 가장 적합한 물체로 인식되는 물체의 라벨을 l_m 이라고 하면, 아래의 식과 같이 구해진다.

$$l_m = \arg \min \sqrt{(State.x_n - State_{Old}.x)^2 + (State.y_n - State_{Old}.y)^2} \quad (4-5)$$

위의 식 4-5를 통해 계산된 라벨을 이용하여 양손의 엄지, 검지, 중지의 라벨을 이전 State와 다음 State간의 거리 값이 36개 중, 최적의 거리 값을 매 순간 체크하여 라벨 번호를 갱신한다.

5. 모션인식 시스템 구현

다중 좌표 추적 시스템을 통해서 손가락의 모든 점을 제대로 Tracking 하고 있다. 이를 이용하여 양손 세 손가락의 움직임에 따른 모션을 제안한다.

모션에는 여러 가지가 있겠지만 본 연구에서는 직관적으로 쓸 수 있고 사람들에게 익숙한 마우스를 컨트롤 하는 시스템을 기본으로 하여 구현하였다.

$$\begin{cases} Center_R = (Point_{RT} + Point_{RI} + Point_{RM}) / 3 \\ Center_L = (Point_{LT} + Point_{LI} + Point_{LM}) / 3 \end{cases} \quad (5-1)$$

위의 식에서 $Center_R$ 은 오른손의 중점을, $Center_L$ 은 왼손의 중점을, $Point_{RT}, Point_{RI}, Point_{RM}$ 은 각각 오른손의 엄지, 검지, 중지의 중점 좌표를 $Point_{LT}, Point_{LI}, Point_{LM}$ 은 왼손의 엄지, 검지, 중지의 중점 좌표를 나타낸다. 각 점들 간의 거리를 계산하고, 그림 5-1과 같이 각 점을 선으로 이은 삼각형을 만들어 준다. 중점의 좌표를 중점으로 하여 점들 간의 거리 중 가장 긴 거리를 반지름으로 하는 원을 만들어주어 움직임의 형태를 보다 명확히 보여준다.

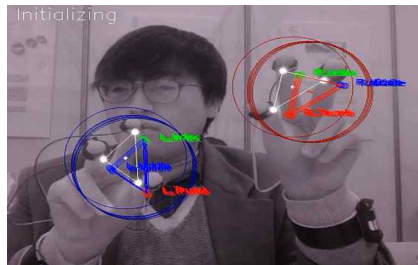
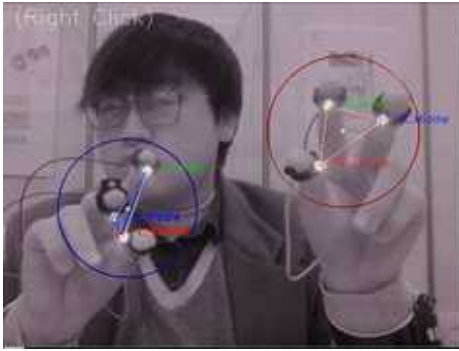


사진 5-1 Init Position

$$\begin{cases} Center_R = (Point_{RT} + Point_{RI} + Point_{RM}) / 3 \\ Center_R \times Scale \rightarrow Point_{Mouse} \end{cases} \quad (5-2)$$

앞의 식에서 $Scale$ 은 영상의 좌표와 화면상의 좌표를 맞추어주기 위한 보정 값, $Point_{Mouse}$ 는 마우스가 화면상에서 움직이는 좌표를 나타낸다.



(a)



(b)

사진 5-2 Motion Right Click / Left Click

$$\begin{cases} Point_{LT} > Center_L - Shift_{click} \\ \leq ngth_{L-(Thumb - Index)} \leq D_{Left Click} \end{cases} \quad (5-3)$$

$$\begin{cases} Point_{LT} > Center_L - Shift_{click} \\ \leq ngth_{L-(Middle - Thumb)} \leq D_{Right Click} \end{cases}$$

(5-4)

위의 식에서 $Shift_{click}$ 은 click 구현을 위한 보정 값이고, $D_{Left Click}$, $D_{Right Click}$ 은 각각 좌 클릭과 우 클릭에 모션 동작 임계점이다. $\leq ngth_{L-(Thumb - Index)}$, $\leq ngth_{L-(Middle - Thumb)}$ 은 각각 엄지와 검지, 엄지와 중지 사이의 거리를 나타내며, 식 (5-3)

과 식(5-4)을 만족하는 경우 해당동작을 수행한다.

사진 5-2의 (a)와 같이 엄지와 중지의 관계가 식 (5-3)을 만족하는 경우 Right Click, (b)와 같이 엄지와 검지의 관계가 식 (5-4)을 만족하는 경우 Left Click을 수행한다.



(a)



(b)

사진 5-3 Motion Scroll Up / Scroll Down

$$\begin{cases} Point_{L-Thumb} \leq Point_{L-Center} - Shift_{scroll} \\ \leq ngth_{L-Radiou} > L_{Scroll Up} \\ (L-Radiou = L_{\Delta}(Max)) \end{cases} \quad (5-5)$$

$$\begin{cases} Point_{L-Thumb} \leq Point_{L-Center} - Shift_{scroll} \\ \leq ngth_{L-Radiou} < L_{Scroll Down} \\ (L-Radiou = L_{\Delta}(Max)) \end{cases} \quad (5-6)$$

위의 식에서 $Shift_{scroll}$ 은 scroll 구현을 위한 보정 값이고, $L_{Scroll Up}$, $L_{Scroll Down}$ 은 각각 스크롤에 대한 모션 동작 임계점이다. $L-Radiou$ 은 왼손의 각 손가락간의 거리 중 가장 긴 부분을 나타내고, $\leq ngth_{L-Radiou}$ 는 그 거리를 나타낸다.

사진 5-3의 (a)와 같이 가장 긴 변의 길이가 식 (5-5)을 만족하면 Scroll Up, (b)와 같이 식 (5-6)을 만족하는 경우

Scroll Down을 수행한다.

기본적인 틀은 이런 식으로 잡았지만, 모든 좌표의 위치를 추적하고 있기 때문에 여러 가지 다른 방식의 모션도 충분히 제작 가능성을 확인하였다.

6. 실험 및 결과 고찰

본 연구를 진행함에 있어 제안한 시스템을 실험을 통해 최적화를 진행하고, 제대로 동작함을 확인하는 작업을 진행한다.

보다 정확한 인식 및 정리된 영상을 위해 물리적인 환경보상 외에 소프트웨어적인 환경보상을 진행한다.

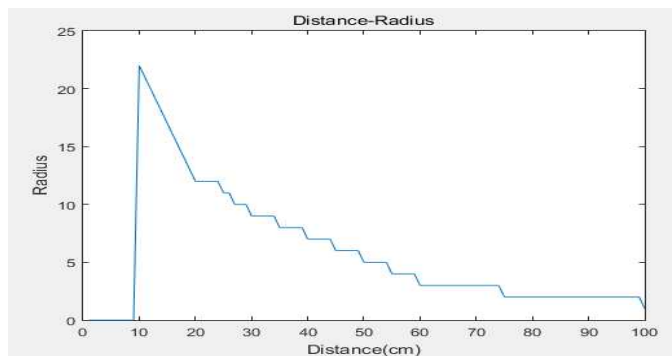


그림 6-1 거리에 따른 반지름 그래프

위의 그림 6-1의 그래프를 보면 IR LED의 밝기는 기본적으로 밝아 대략 1m 정도까지는 제대로 영상이 들어오는 것을 볼 수 있다. 거리에 따라 점의 크기가 줄어들기는 하지만 일반적인 조명과 같은 노이즈와는 어느 정도의 차이가 존재한다. 따라서 대강의 크기를 알고 있는 점의 크기 범위를 지정하여 해당 크기의 Blob을 제외한 나머지 Label을 전부 제거 한다.

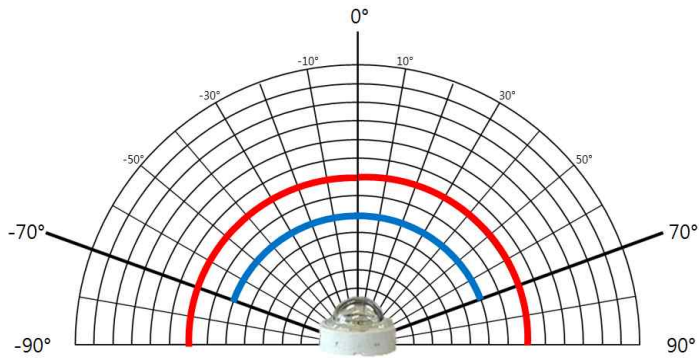


그림 6-2 IR LED의 인식 범위(각도)

웨어러블 플랫폼의 인식 실험을 진행하였다. 그림 6-2에서 확인 가능하듯이 웨어러블 플랫폼에 사용된 IR LED의 시야각 (Viewing Angle)은 파란색 부분이 나타내고 있는 140°이다.

그러나 손으로 모션을 취했을 때 손의 각도가 다양하게 벌어지기 마련이다. 따라서 어느 정도의 각도까지 IR LED의 빛이 인식 가능한지에 대한 실험한 결과 빨간색 부분이 나타내는 $\pm 90^\circ$ 정도까지의 위치에 카메라가 있어도 잘 인식됨을 알 수 있었다. 따라서 웨어러블 플랫폼을 이용한 손 모션의 자유도는 어느 정도 보장된다 할 수 있다.

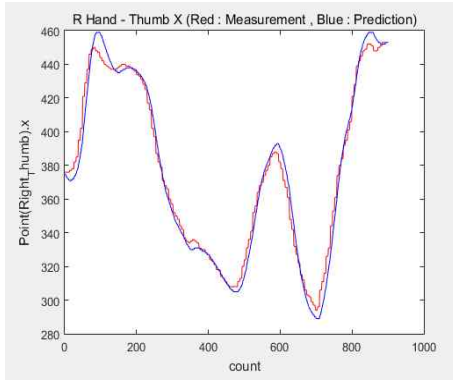
<표 4> FPS Check

회차	실행 시간	fps (Frame per Sec)
1	0.0451	22.1682
2	0.0449	22.2597
3	0.0444	22.5173
4	0.0455	21.9665
5	0.0467	21.4246

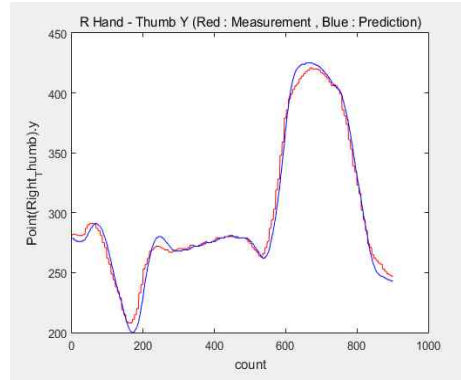
위의 <표 4>는 2.60GHz, Intel(R) Core(TM) i7-4720HQ 컴퓨터에서 측정한 실행 시간 및 fps이다.

Computer Vision을 진행함에 있어 영상처리에 다양한 알고리즘의 계산을 필요로 하는 실시간 시스템에서 실행 시간 및 초당 프레임 수(fps, frame per sec)의 확인은 중요한 과정이라고 할 수 있다.

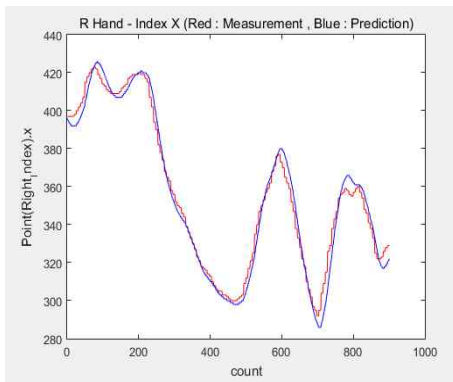
본 연구의 실시간 추적 시스템에서 fps 확인을 위해 중요 계산 부분이 500번 돌아가는 동안의 한 프레임에 걸리는 시간의 평균을 이용하였다. 이를 이용하여 초당 프레임 수를 계산하고, 여러 번의 실험을 통해 확인을 해본결과 결과가 매우 비슷하게 나왔기 때문에 5회 정도의 데이터만을 첨부 하였다.



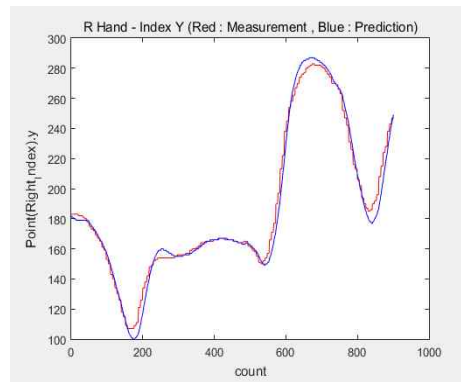
(a)



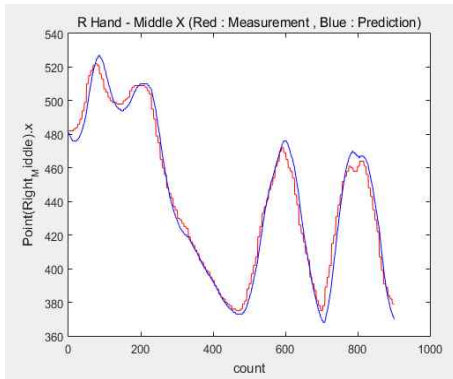
(d)



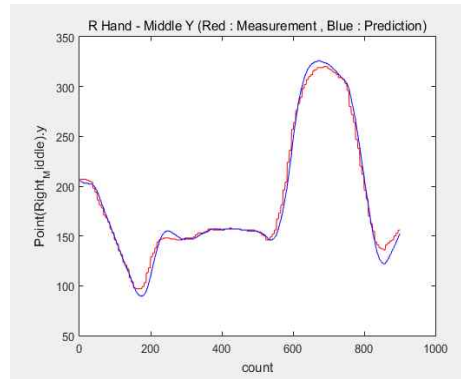
(b)



(e)

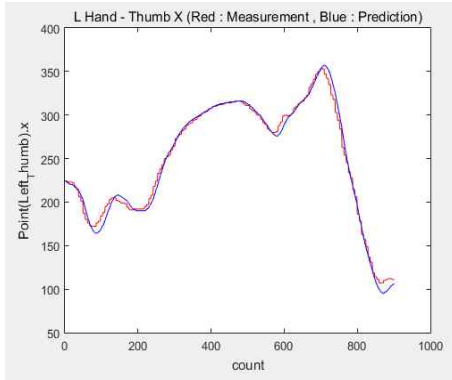


(c)

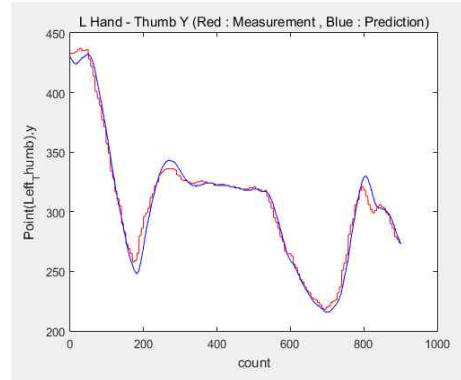


(f)

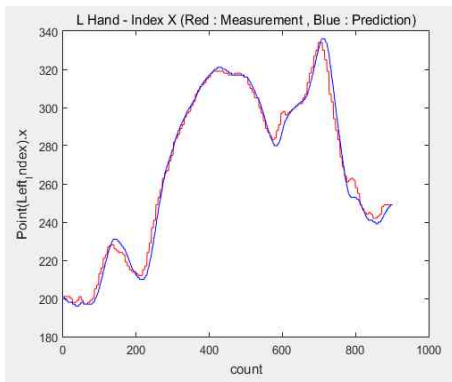
그림 6-3 오른손 Tracking Graph



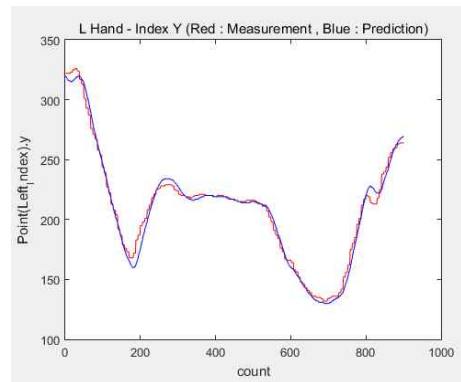
(a)



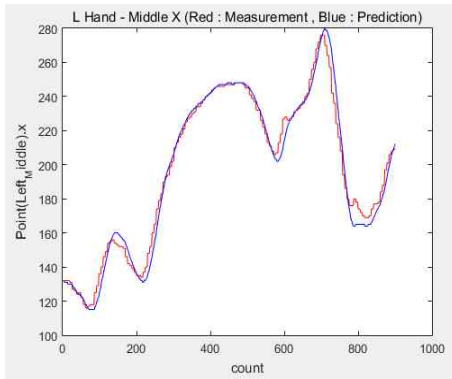
(d)



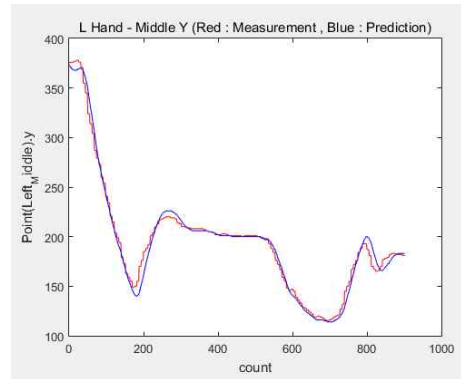
(b)



(e)



(c)



(f)

그림 6-4 왼손 Tracking Graph

그림 6-3는 오른손의 추적 상태에 대한 그래프로 (a)는 오른손 엄지에 대한 x좌표, (b)는 오른손 검지에 대한 x좌표, (c)는 오른손 중지에 대한 x좌표, (d)는 오른손 엄지에 대한 y좌표, (e)는 오른손 검지에 대한 y좌표, (f)는 오른손 중지에 대한 y좌표를 나타낸 값이다.

그림 6-4은 왼손의 추적 상태에 대한 그래프로 (a)는 왼손 엄지에 대한 x좌표, (b)는 왼손 검지에 대한 x좌표, (c)는 왼손 중지에 대한 x좌표, (d)는 왼손의 엄지에 대한 y좌표, (e)는 왼손 검지에 대한 y좌표, (f)는 왼손 중지에 대한 y좌표를 나타낸 값이다.

각각의 그래프에서 붉은색의 그래프가 측정값, 푸른색의 그래프가 추정값을 나타낸다. 그래프를 확인해 보면 측정값에는 노이즈로 인한 굴곡이 존재하는 반면 추정값은 Kalman Filter를 통해 노이즈가 제거된 상태로 잘 추적 되고 있음을 확인할 수 있다. 앞의 그림 6-3와 6-4의 그래프는 손에 대한 측정값과 추적 시스템에 의해 추적된 추정값의 데이터를 비교한 결과를 나타낸 것이다.

위의 데이터를 토대로 2D Graph, 3D Graph로 나타내면 다음과 같다.

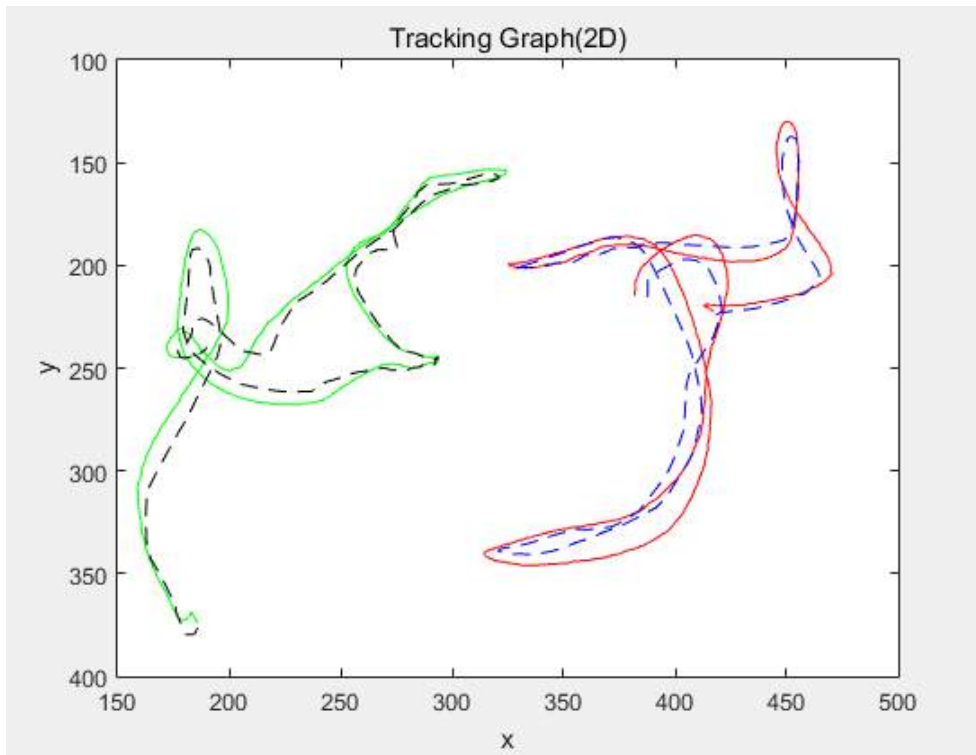


그림 6-5 Tracking Graph (2D)

점선으로 된 데이터는 측정값을 나타내고, 실선으로 된 데이터는 추정값을 나타낸다. 빨간색과 파란색 그래프가 오른손, 녹색과 검정색 그래프가 왼손을 나타낸다.

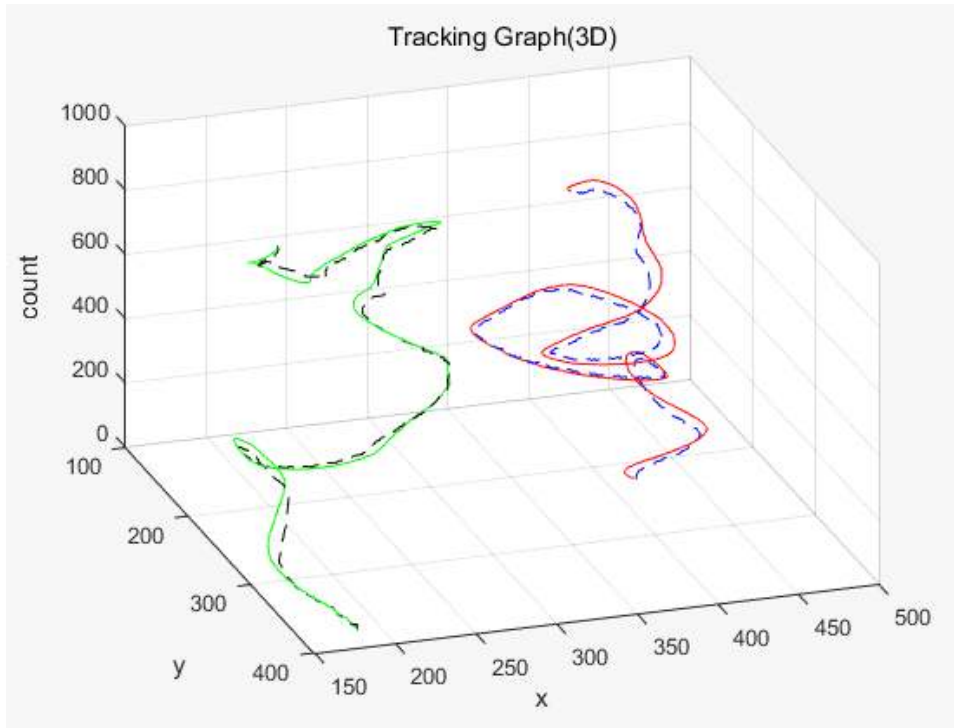
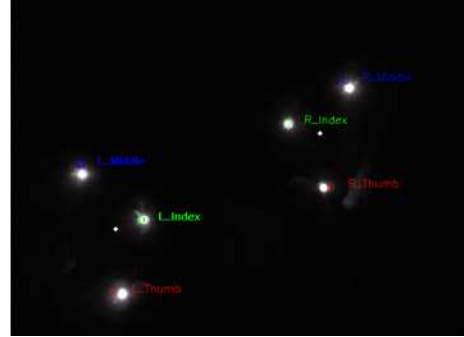


그림 6-6 Tracking Graph (3D)

점선으로 된 데이터는 측정값을 나타내고, 실선으로 된 데이터는 추정값을 나타낸다. 빨간색과 파란색 그래프가 오른손, 녹색과 검정색 그래프가 왼손을 나타낸다. z축은 알고리즘이 돌아간 횟수를 나타낸다.



(a)

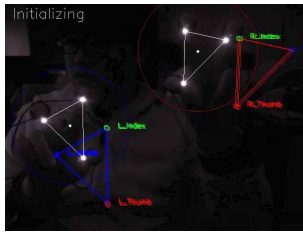


(b)

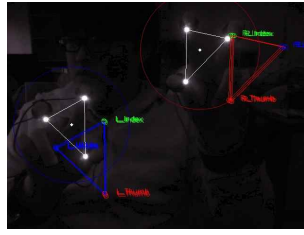
사진 6-1 Matching Test

사진 6-1은 추적 알고리즘 적용 전과 후에 대한 차이를 보여준다. (a)의 경우는 추적 알고리즘 적용 전으로 손가락이 겹치는 경우 각 손가락의 좌표가 서로 엇갈려 제대로 추적이 되지 않는 것을 확인할 수 있다. (b)의 경우 추적 알고리즘 적용 후에 손가락의 위치가 (a)와 같지만 좌표가 서로 엇갈리지 않고 제대로 추적이 되고 있는 것을 확인할 수 있다.

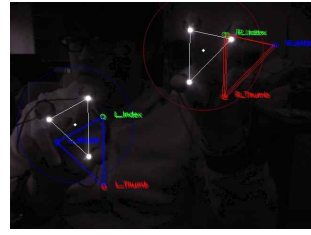
본 연구의 추적 시스템에서 확인해야 할 중요한 포인트는 여러 가지가 있다. 그 중에서 추적 시스템이 제대로 동작하는지, 추적 시스템에 있어서의 데이터 겹침 현상에 대한 추적 상태 확인, 손의 회전에 따른 각 손가락 좌표의 추적 상태 확인의 경우를 중점적으로 확인하였다.



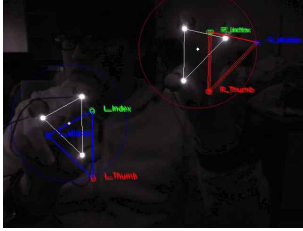
frame 100



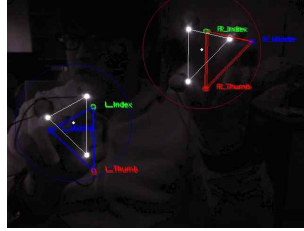
frame 112



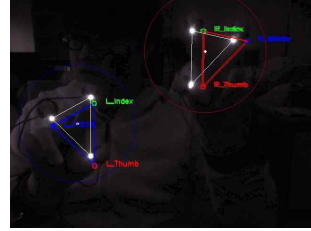
frame 124



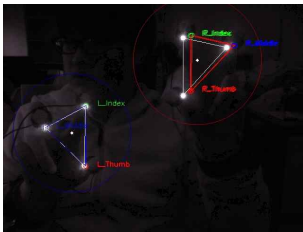
frame 136



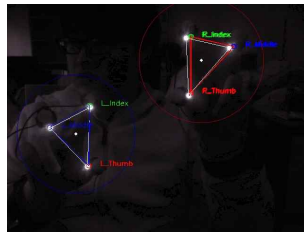
frame 148



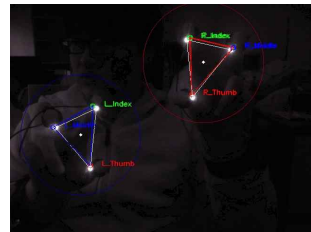
frame 160



frame 172



frame 184

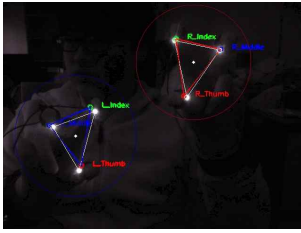


frame 196

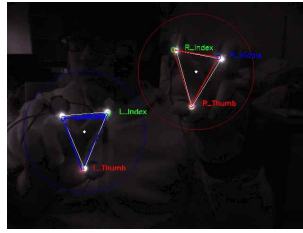
사진 6-2 추적 시스템에 대한 확인 영상

사진 6-2를 보면 양손의 하얀색 삼각형을 왼손의 파란색 삼각형과 오른손의 빨간색 삼각형이 각각 찾아들어가는 것을 확인 할 수 있다.

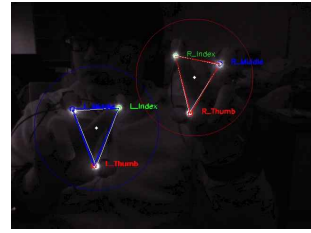
각 손가락의 끝단의 좌표는 엄지는 붉은색, 검지는 녹색, 중지 푸른색으로 제대로 추적 되고 있음을 확인할 수 있다.



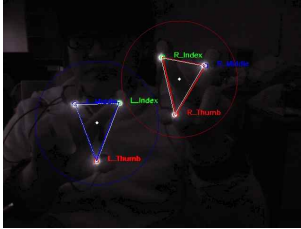
frame 220



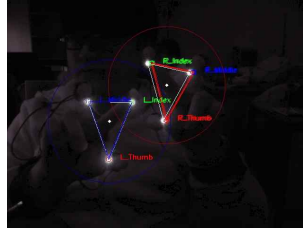
frame 280



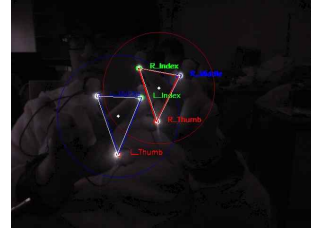
frame 340



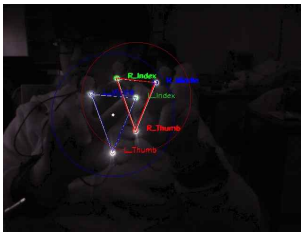
frame 400



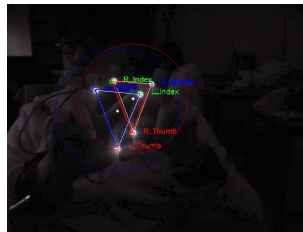
frame 460



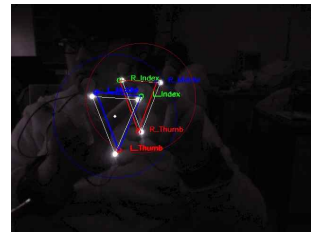
frame 520



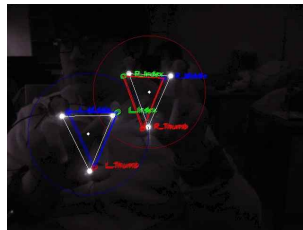
frame 580



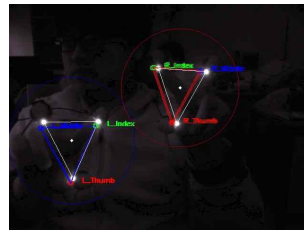
frame 640



frame 700



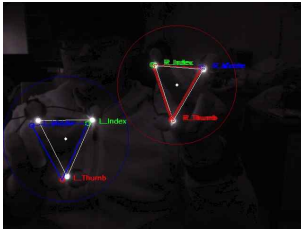
frame 760



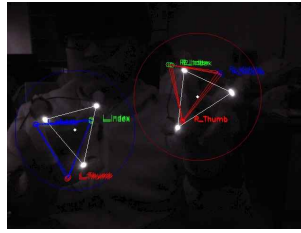
frame 820

사진 6-3 양손이 겹치는 경우의 추적 확인 영상

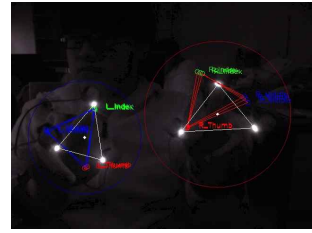
사진 6-3을 보면 양손이 겹쳐져서 손가락의 좌표가 서로 섞임에도 불구하고 각각의 데이터가 엇갈리지 않고 제대로 추적됨을 확인할 수 있다.



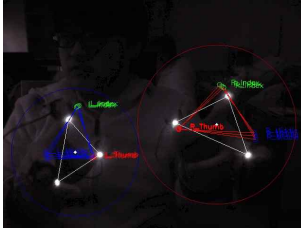
frame 832



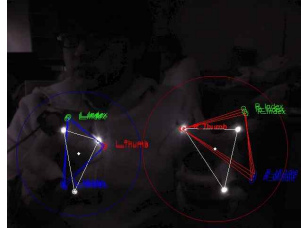
frame 856



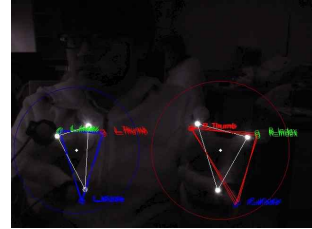
frame 880



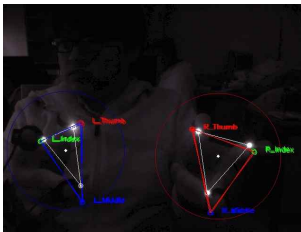
frame 904



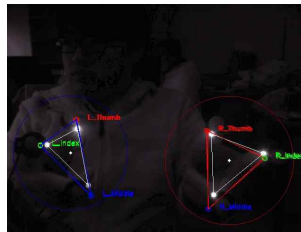
frame 928



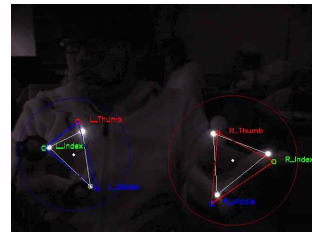
frame 952



frame 976



frame 1000



frame 1024

사진 6-4 손의 회전에 따른 추적 확인 영상

사진 6-4를 보면 오른손의 경우 시계방향으로 왼손의 경우 반시계 방향으로 각각 회전을 시킨 상태에서도 각 손가락의 데이터가 제대로 추적되고 있음을 확인할 수 있다.

7. 결 론

본 연구에서는 기존의 모션인식 디바이스가 아닌 대중화 되어있는 USB Camera를 이용해 손가락의 좌표를 추적하고, 그것을 기반으로 다양한 모션이 제작 가능하도록 개발한 모션인식 시스템에 대해 제안했다.

시스템 구현을 위한 알고리즘으로 특징점이 존재하는 다수의 물체를 추적하는 것이 아닌 다수의 동일 물체에 대한 추적이 가능한 이동물체 추적 방법을 제안하였다.

기존의 USB Camera의 IR Cutter Filter를 제거하고 Negative Film으로 교체해 주어 적외선 영상으로서 이용함으로써 환경에 대한 노이즈와 영상정보량을 줄여서 보다 색상이나 조명등에도 강인한 특징을 나타내며, Kalman Filter 알고리즘은 현재 물체의 정보로 이동 물체의 위치 및 상태를 예측할 수 있다는 장점이 있었다. 동일 물체 추적을 위한 Optimal하고 빠른 시스템을 위해 Hungarian Algorithm은 빠른 계산 속도를 보여 주었으며, 이를 Kalman Filter와 결합함으로써 실시간에서 정확하게 손가락을 추적하는 시스템의 구현이 가능하였다.

제안된 알고리즘의 성능을 확인해본 결과 모든 손의 데이터가 존재하는 한 제대로 추적을 하는 것을 확인할 수 있었지만, 손을 너무 빠르게 움직여서 데이터가 분산되거나 손과 손이 겹쳐 데이터가 가려지는 경우 추적에 실패하는 것을 확인할 수 있었다.

향후 알고리즘을 개선하여 처리속도를 개선하여 빠르게 물체를 추적할 수 있는 알고리즘과, 데이터가 소멸 시에는 계산만 하고, 모든 데이터가 있는 경우만 시스템을 갱신한다면 보다 효율적인 모션인식 시스템을 구축할 수 있을 것으로 사료된다.

참고 문헌

- [1] 김윤이, 강승연, 김대식, 김유진, 박재준, 박주언, 송경희, 오성익, 유혜영, 이동우, 이보인, 이상현, 이효석, 임동균, 정수근, 조동연, "빅픽처 2017 4차 산업혁명과 고립주의의 역설", 생각정원, p.25, 2016.11
- [2] 이금분, "공분산 행렬과 칼만필터를 결합한 고속 이동 물체 추적 방법", 한국정보통신학회논문지, Vol. 19, No. 6 : 1477-1484, 2015.6
- [3] 김형태, 설성욱, 이철현, 강창순, 남기곤, "칼만 필터를 이용한 다중 차량 추적 알고리즘", 대한전자공학회 학술대회, 955-958, 1998.11.
- [4] R. E. Kalman, "A New Approach to Linear Filtering and Prediction Problems", Transactions of the ASME - Journal of Basic Engineering, 82 (Series D): 35-45, 1960.
- [5] Harold W. Kuhn, "The Hungarian Method for the Assignment Problem", Naval Research Logistics Quarterly, Kuhn's Original Publication, pp. 83-97, 1955.
- [6] J. Munkres, "Algorithms for the Assignment and Transportation Problems", Journal of the Society of Industrial and Applied Mathematics, Vol. 5, No. 1, pp. 32-38, 1957.

- [7] Ford, L. R.; Fulkerson, D. R. "Maximal flow through a network" (PDF). Canadian Journal of Mathematics. 8: 399 - 404, 1956.
- [8] Edmonds, Jack; Karp, Richard M. "Theoretical improvements in algorithmic efficiency for network flow problems". Journal of the ACM. Association for Computing Machinery. 19 (2): 248 - 264, 1972.
- [9] 주재흠, "적외선 연속 영상에서 다중 소형 표적 추적 알고리즘", 신호처리 시스템학회 논문지, Vol.14, No. 1, pp,33-38, 2013.
- [10] 노진구, "내용기반 영상정보 검색기술에 관한 이론적 고찰", 한국도서관정보학회, Vol. 31, No. 1 : 229-258, 2000.
- [11] David G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints", International Journal of Computer Vision, Vol. 60, No. 2, pp. 91-110, 2004.
- [12] Nandashri D, Smitha P, "An Efficient Tracking of Multi Object Visual Motion using Hungarian Method", International Journal of Engineering Research & Technology, Vol. 4, No. 04, 2015.
- [13] <http://ultimatefifa.com/2012/fifa-13-motion-capture-session/>

- [14] 최태준, 유석호, 이동열, 이완복, "광학식 모션 캡처 (Optical Motion Capture)방식을 이용한 디지털 캐릭터 움직임" 한국콘텐츠학회논문지, Vol. 7, No. 8, 2007.
- [15] 안정현, "스테레오 영상을 이용한 광학식 모션캡처 후처리 소프트웨어 개발", 한국과학기술원, 61-62, 2000.
- [16] 이진호, "다중 키넥트를 활용한 향상된 인체추적 방법", 세종대학교 일반대학원, 2014.
- [17] 황원욱, "Kinect를 이용한 재할 프로그램 구현", 중앙대학교 대학원, 2013.
- [18] 주향한, 조민수, 인승교, 조규원, 민준기, "립 모션 컨트롤러를 이용한 야구 게임 개발", KIISE Transactions on Computing Practices, Vol. 21, No. 5, pp. 343-350, 2015. 5
- [19] <http://www.Leanp Motion.kr>
- [20] <https://www.slideshare.net/JonghoonSeo/20-24091955>
- [21] <http://lg-sl.net/product/scilab/sciencestorylist/IQEX/readSciencestoryList.mvc?sciencestoryListId=IQEX2008070004>
- [22] https://ko.wikipedia.org/wiki/%EC%B9%BC%EB%A7%8C_%ED%95%84%ED%84%B0