

정규논문 (Regular Paper)

방송공학회논문지 제19권 제6호, 2014년 11월 (JBE Vol. 19, No. 6, November 2014)

<http://dx.doi.org/10.5909/JBE.2014.19.6.942>

ISSN 2287-9137 (Online) ISSN 1226-7953 (Print)

RGB 카메라 기반 실시간 21 DoF 손 추적

최준영^{a)}, 박종일^{b)‡}

RGB Camera-based Real-time 21 DoF Hand Pose Tracking

Junyeong Choi^{a)} and Jong-Il Park^{b)‡}

요 약

본 논문은 단안의 RGB 카메라를 이용하는 실시간 손 추적 방법을 제안한다. 손은 높은 degrees of freedom을 가지고 있기 때문에 손 추적은 높은 모호성을 가지고 있다. 따라서 제안하는 방법에서는 손 추적의 모호성을 줄이기 위해서 단계별 손 추적 전략을 채택하였다. 제안하는 방법의 추적 과정은 손바닥 포즈 추적, 손가락 yaw 움직임 추적, 그리고 손가락 pitch 움직임 추적, 세 단계로 구성되어 있으며, 각 단계는 순서대로 수행된다. 제안하는 방법은 손은 평면으로 간주할 수 있다고 가정하고, 평면 손 모델을 이용한다. 평면 손 모델은 손 모델을 현재의 사용자 손 모양에 맞춰서 변경하는 손 모델 재생성을 가능하게 하는데, 이는 제안하는 방법의 강건성과 정확도를 증가시킨다. 그리고 제안하는 방법은 실시간 연산이 가능하고 GPU 기반 연산을 요구하지 않기 때문에, Google Glass와 같은 모바일 장비를 포함한 다양한 환경에 적용가능하다. 본 논문은 다양한 실험을 통해서 제안하는 방법의 성능과 효용성을 입증한다.

Abstract

This paper proposes a real-time hand pose tracking method using a monocular RGB camera. Hand tracking has high ambiguity since a hand has a number of degrees of freedom. Thus, to reduce the ambiguity the proposed method adopts the step-by-step estimation scheme: a palm pose estimation, a finger yaw motion estimation, and a finger pitch motion estimation, which are performed in consecutive order. Assuming a hand to be a plane, the proposed method utilizes a planar hand model, which facilitates a hand model regeneration. The hand model regeneration modifies the hand model to fit a current user's hand, and improves robustness and accuracy of the tracking results. The proposed method can work in real-time and does not require GPU-based processing. Thus, it can be applied to various platforms including mobile devices such as Google Glass. The effectiveness and performance of the proposed method will be verified through various experiments.

Keyword : hand tracking, particle swarm optimization, human computer interaction, computer interface

a) 한양대학교 컴퓨터소프트웨어학과 (Department of Computer and Software, Hanyang University)

b) 한양대학교 컴퓨터공학부 (Department of Computer Science and Engineering, Hanyang University)

‡ Corresponding Author : 박종일 (Jong-Il Park)

E-mail: jipark@hanyang.ac.kr

Tel: +82-2-2220-0368

※ 본 연구는 미래창조과학부 및 정보통신산업진흥원의 IT융합 고급인력과정 지원사업의 연구결과로 수행되었음(NIPA-2014-H0401-14-1001)

· Manuscript received September 29, 2014 Revised October 23, 2014 Accepted October 23, 2014

I. 서론

Google Glass와 같은 모바일 장비의 발전이 꾸준히 있었으나, 그 장비를 위한 인터페이스에 대한 변화는 거의 없었다^[1]. 모바일 장비를 위한 인터페이스는 버튼 터치와 목소리 인식과 같은 간단한 상호작용만을 제공하기 때문에, 최근 많은 연구자들은 모바일 장치에 특화된 사용자 친화적 인터페이스에 대해서 많은 관심을 보이고 있다. 그리고 특히 손동작 기반 인터페이스가 가장 큰 가능성을 가지고 있다고 여겨지고 있다. 손동작 기반 인터페이스 구현을 위해서는 효과적인 손 추적 알고리즘이 필수적이지만, 손의 높은 degrees of freedom(DoF) 때문에 손은 정확히 추적하기 어렵다. 따라서 적은 연산으로 손을 정확히 추적하기 위해서 연구자들은 데이터 글러브와 같은 착용형 감지기를 사용하였다. 그러나 이런 부가적인 장비는 사용자에게 불편할 수 있기 때문에 추가적인 감지기를 요구하지 않은 비전 기반 손 추적 방법이 최근 많은 관심을 받고 있다.

본 논문은 단안의 RGB 카메라 기반 손 추적 방법은 제안한다. 대부분의 손 추적 방법은 하나의 추적 알고리즘을 사용해서 한 번의 과정을 통해서 손을 추적하는데, 이와는 대조적으로 제안하는 방법은 추적 과정을 손바닥 포즈 추적, 손가락 yaw 움직임 추적, 그리고 손가락 pitch 움직임 추적, 세 단계로 나눴다. 첫 번째 과정에서는 손바닥 포즈를 추적하고, 두 번째와 세 번째 과정을 통해서 손가락 움직임을 추적한다. 이 단계별 추적 전략은 손의 높은 DoF로 인해 발생하는 손 추적의 모호성을 줄여준다. 또한 본 논문은 추적 과정을 효율성을 극대화하기 위해서 각 단계별로 적절한 추적 알고리즘을 제시한다. 사람의 손을 평면으로 가정하고 제안하는 방법은 평면 손 모델을 사용하며, 평면 모델의 사용은 손 모델을 현재 사용자의 손 모양에 맞춰서 변형하는 손 모델 재생성을 가능하게 한다. 따라서 제안하는 방법은 사용자가 바뀌더라도 다양한 손 모양에 대해서 강인한 추적이 가능하다. 게다가 손 모델 재생성은 손 추적 알고리즘의 정확도를 향상시켜준다. 더군다나 제안하는 방법은 GPU 기반 연산 없이 CPU 연산만을 사용해서 실시간 연산이 가능하다.

II. 관련 연구

손 추적 방법은 하드웨어 기반과 비전 기반 두 종류로 나눌 수 있다. 하드웨어 기반 방법은 글러브와 같은 착용형 감지기를 사용함으로써 손을 높은 정확도로 빠르게 추적할 수 있다. 글러브 기반 방법의 예로써, 데이터 글러브 기반 방법은 감지기를 글러브에 설치하고 감지기로부터 획득한 정보를 이용해서 사용자의 손을 추적한다^[2]. 그리고 색상 글러브 기반 방법은 다양한 색상으로 칠해진 글러브를 착용하고, RGB 카메라를 이용해서 각 색상을 검출함으로써 손을 추적한다^[3]. 그러나 글러브를 착용하는 것은 사용자에게 불편할 수 있기 때문에 상대적으로 덜 불편한 밴드 기반 방법이 더 많은 관심을 받고 있다. 예를 들어서 IR 조명과 IR 카메라를 손목 밴드에 장착하고, 사용자는 손목 밴드를 착용함으로써 손을 추적하는 방법이 있었다^[4]. 또한, Sixth-Sense는 다양한 색의 손가락 밴드를 착용함으로써 손가락 끝을 검출한다^[5]. 그러나 이런 부가적인 장비는 여전히 사용자에게 불편함을 줄 수 있다.

하드웨어 기반 방법과는 대조적으로, 비전 기반 방법은 부가적인 장비를 요구하지 않는다. 그러나 손바닥 포즈가 변하게 되면 손가락으로 인한 가려짐 현상이 발생하게 되고, 이 가려짐 현상은 정교한 인식을 방해한다. 가려짐 현상을 최소화하기 위하여, 대부분의 비전 기반 방법은 손바닥과 카메라의 광축이 수직하다고 가정한다^[6,7]. 또한 이 가정은 손바닥 포즈를 고정함으로써 손의 DoF를 줄여주기 때문에, 손 추적을 더욱 빠르고 강인하게 만든다. 그러나 손바닥 포즈 제약은 손의 움직임을 제약하기 때문에 연구자들은 어떤 포즈 제약도 없는 손 추적 방법에 대한 관심이 높아졌다. 예를 들어서, 특징 기반 손바닥 포즈 계산을 이용한 손바닥 포즈 변화에 강인한 손 추적 방법이 있었다^[8]. 그러나 이 방법은 손바닥 포즈를 대략적으로 계산하기 때문에 손을 정확히 추적하는 것이 어려웠다.

최근 깊이 카메라 성능의 향상과 함께, 3D 모델 기반 손 추적 방법이 많은 관심을 받고 있다. 3D 모델 기반 방법은 3D 손 모델을 다양하게 변화시키고, 영상에서 획득한 특징과의 비교를 통해서 손 포즈를 계산하기 때문에, 손바닥 포즈 제약 없이 정확한 추적을 수행할 수 있다. 그러나 손 추적을 위해서 손 모델을 변환하고 손 포즈의 후보를 생성하는 작업이 많은 연산을 요구하며, 실시간 연산이 어려웠다

¹⁹⁾ 근래에는 GPU 기반 병렬 처리를 이용해서 실시간 연산이 가능해졌으나^{10, 11, 12)}, 이 방법은 고성능 GPU와 깊이 카메라 혹은 다중 카메라를 요구하기 때문에 모바일 폰을 포함한 다양한 환경에는 적용하기 힘들다는 단점을 발생시켰다. 또한, 깊이 영상의 노이즈 패턴을 분석한 뒤 노이즈에 강인하게 손 추적하는 방법이 있었다¹³⁾. 이 방법은 CPU i7 3.20GHz와 24GB RAM의 데스크탑 환경에서 GPU 기반 고속화 없이 12 fps의 속도로 동작하지만, 깊이 카메라를 요구한다는 단점은 여전히 남아 있다.

위에서 설명한 것과 같이, 존재하는 실시간 손 추적 방법은 손바닥 포즈를 제약하거나, 이 제약을 피하기 위해서, 깊이 카메라 혹은 다중 카메라를 사용하며, 실시간 연산을 위해서는 고성능 GPU를 요구하기 때문에, 데스크탑과 같은 환경에서만 사용할 수 있으며 모바일 폰과 같은 모바일 장비의 환경에 적용하기에는 적절하지 못하다. 그러나 본 논문에서 제안하는 방법은 손바닥 포즈 제약 없이 단안의 RGB 카메라를 기반으로 실시간으로 동작 가능하며, 또한 GPU 기반 연산을 요구하지 않기 때문에, 모바일 장비를 포함한 다양한 플랫폼에 적용될 수 있다.

III. 손 모델

제안하는 방법은 그림 1과 같은 평면 손 모델을 사용한다. 사용하는 손 모델을 각 손가락이 3 DoF, 손바닥이 6 DoF로 총 21 DoF로 구성되어 있다. 일반적으로 사람의 손가락은 뿌리에서 2 DoF 그리고 두 종류의 관절에서 2 DoF로 총 4 DoF를 가지고 있지만, 손가락의 한 관절은 다른 관절의 움직임에 비례해서 움직인다는 이론¹⁴⁾을 바탕으로 관절을 1개로 줄인 모델을 사용한다. 여기서 그림 1과 같이 관절의 각도는 J-각도, 그리고 뿌리의 두 종류 각도는 각각 Z-각도와 X-각도로 표기한다.

IV. 손 추적

그림 2는 제안하는 방법의 단계별 손 추적 과정을 보여준다. 그림 2에서 보이는 것과 같이 제안하는 방법은 손바닥 포즈 계산, 손가락 yaw 움직임 계산, 그리고 손가락 pitch 움직임 계산, 세 단계로 구성되어 있다. 손바닥 포즈 계산

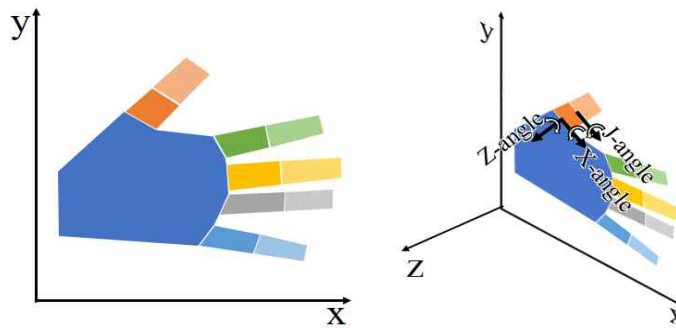


그림 1. 본 논문에서 사용하는 평면 손 모델
Fig. 1. Planar hand model used in this paper



그림 2. 제안하는 방법의 전체 과정
Fig. 2. Overall processing of the proposed method

단계에서는 제안하는 방법은 손 모델의 손바닥 포즈와 현재 입력 영상을 일치 시킨다. 그리고 손가락 yaw 움직임 계산 단계에서 손가락의 yaw 움직임을 계산하고, 마지막으로 손가락 pitch 움직임 계산 단계에서 손가락의 pitch 움직임을 계산함으로써 최종 손 포즈를 결정한다.

1. 손바닥 포즈 계산

그림 3은 제안하는 방법의 첫 번째 단계인 손바닥 포즈 계산 과정을 보여준다. 이전 입력 영상의 손 포즈는 그림 3-(a)와 같다. 그림 3-(b)에서 보이는 것과 같이 살색 영역을 검출한 다음 거리 변환을 이용해서 노이즈를 제거함으로써 손 영역이 검출된다^[8]. 그림 3-(d)는 손 영역의 외곽선(그림 3-(c))에 거리 변환을 적용함으로써 획득된다. 본 논문은 손바닥 포즈 계산을 위한 방법으로 거리 변환한 손 외곽선 영상 기반 recursively observation assignment (ROA)를 제시한다.

그림 4는 손바닥 포즈 계산을 위한 ROA를 보여준다. ROA의 첫 단계는 식 1을 이용해서 손 모델의 손바닥 포즈를 다양하게 변화시킴으로써 손바닥 포즈 후보(H_p^i)를 생성하는 것이다.

$$H_p^i = P_p + \sum_j V_p(j, i),$$

$$P_p = \begin{bmatrix} R_x \\ R_y \\ R_z \\ T_x \\ T_y \\ T_z \end{bmatrix}, V_p(j, i) = n_p^j T_i, n_{j+1} = r n_p^j. \quad (1)$$

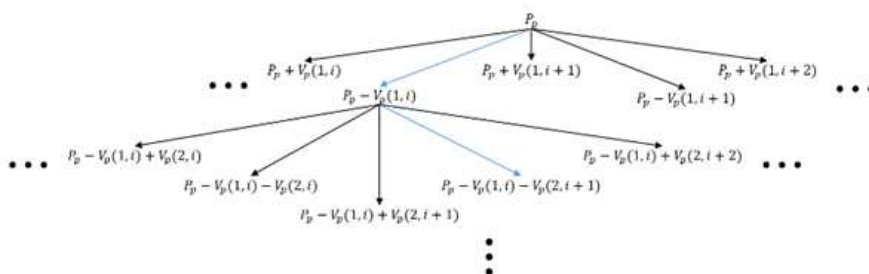


그림 4. 손바닥 포즈 계산을 위한 ROA
 Fig. 4. ROA for the palm pose estimation

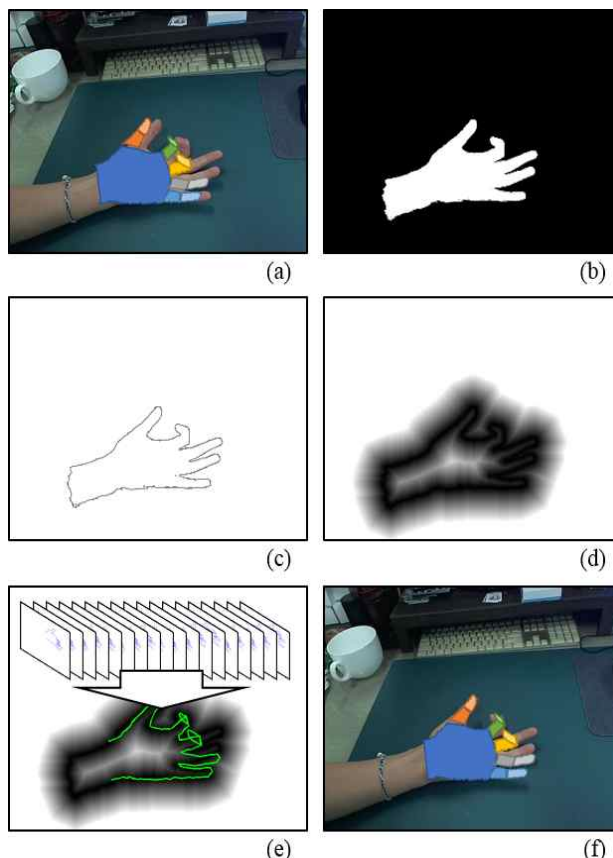
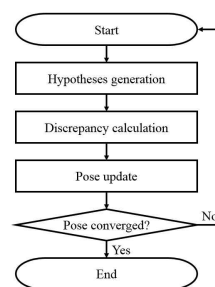


그림 3. 제안하는 방법의 손바닥 포즈 계산. (a) 이전 입력 영상의 손 포즈와 현재 입력 영상, (b) 입력 영상으로부터의 손 영역 검출 결과, (c) 손 영역의 외곽선, (d) 손 영역의 외곽선에 거리 변환을 적용한 영상, (e) 후보군과의 유사도 비교, (f) 손바닥 포즈 계산 결과

Fig. 3. The palm pose estimation of the proposed method. (a) An input scene and a hand pose of a previous frame, (b) hand region segmentation result of the input scene, (c) contour of the hand region, (d) distance transformed contour of the hand region, (e) discrepancy calculation using hypotheses, (f) the result of the palm pose estimation



이때 사용된 변수가 뜻하는 의미는 표 1과 같다.

표 1. 식 1에서 사용된 변수
Table 1. Variables used in Equation 1

Variable name	Definition
T_i	i -th column of T
$V_p(j, i)$	A variance vector for i -th hypothesis
n_p^j	A variation of the palm pose in j -th iteration
r	Decrease ratio of n_p^j at each iteration

본 논문에서 각 변수는 실험적으로 결정되었으며, n_p^0 은 5 그리고 r 은 0.8을 사용했을 때 가장 안정적으로 동작하였다. 생성된 후보는 손바닥 포즈가 변화 가능한 모든 경우의 수를 포함한다. 식 2에서 보이는 것과 같이 T 는 손바닥 포즈 변화 벡터의 모든 종류를 하나의 벡터로 나타낸 것이다. 모든 변화 벡터의 숫자는 $2^{palm DoF}$ 이고, 손바닥 포즈는 양과 음 두 종류의 방향으로 변화 가능하기 때문에, 손바닥 포즈의 변화 가능한 모든 수는 $2 \times 2^{palm DoF}$ 이다.

$$T = \begin{bmatrix} 010000011 & 1 \\ 001000010 & 1 \\ 000100001 & \dots & 1 \\ 000010000 & \dots & 1 \\ 000001000 & 1 \\ 000000100 & 1 \end{bmatrix} \cdot \quad (2)$$

현재의 손바닥 포즈와 가장 비슷한 후보를 찾기 위한 계산은 식 3과 같다(그림 3-(e)).

$$d_c = I_d \cdot \Psi_h \cdot \quad (3)$$

여기서 I_d 는 손 영역의 외곽선에 거리 변환을 적용한 영상을 뜻하고, Ψ_h 는 후보를 현재 입력 영상에 투사한 것(그림 3-(e)의 초록색 선)을 뜻하며, 식 3은 두 영상 사이의 convolution이다. 손바닥 포즈는 후보 생성과 가장 비슷한 후보를 찾는 과정을 수렴할 때까지 반복함으로써 계산된다. 생성과 비슷한 후보 검출은 병렬 연산이 가능하기 때문에, 본 논문에서는 OpenMP^[15]를 이용한 CPU 기반 병렬처리를 이용해서 고속화하였다. 그림 3-(f)는 ROA를 통해 계산된

손바닥 포즈를 보여준다. 그림 3-(f)에서 보이는 것과 같이 손바닥 포즈는 일치하지만, 손가락 포즈는 일치하지 않는 것을 알 수 있다. 특히, 약지의 yaw 움직임과 중지의 pitch 움직임은 전혀 맞지 않는다. 이 불일치는 이후 과정인 손가락 yaw와 pitch 움직임 계산 과정을 통해서 해결된다.

2. 손가락 yaw 움직임 계산

그림 5는 두 번째 단계인 손가락 yaw 움직임 계산 과정을 보여준다. 두 번째 단계에서는 손가락의 yaw 움직임(Z-각도)이 계산된다. 제안하는 방법은 손 영역(그림 5-(b))에 거리 변환을 적용함으로써 손가락 yaw 움직임 계산에 사용될 영상을 획득한다(그림 5-(c)). 그림 6은 손가락 yaw 움직임 계산을 위한 ROA를 보여준다. 이전 각도에서 시작할 경우에는 주변의 손가락에 의한 local minima에 빠지는 경우가 빈번하게 발생하기 때문에, 손바닥 포즈 계산과는 다르게 0°에서 시작한다.

손가락 yaw 움직임 계산을 위해서 n_y^j 간격으로 손가락의 Z-각도를 증가시키고 감소시킴으로써 N 개의 손가락 포즈 후보를 생성한다($N=5$). 손가락 포즈 후보 생성을 위한 공식은 다음과 같다.

$$\begin{aligned} V_r(j, i) &= n_y^j, \\ n_{y+1}^j &= r n_y^j. \end{aligned} \quad (4)$$

본 논문에서는 n_y^0 는 0°이며, n_y^j 는 매 반복마다 r 의 비율로 감소한다. 손가락 포즈 후보와 입력 영상 사이의 유사도는 공식 3과 비슷한 방법을 통해서 계산되는데, 후보의 중심점을 투사시킨 것과 거리 변환된 손 영역 영상과의 convolution을 수행함으로써 계산한다(그림 5-(d)). 그림 5-(d)에서 보이는 것과 같이 매 반복마다 손가락 포즈 후보의 변화폭은 점차 감소하며, 후보는 손가락의 중심으로 수렴한다. 제안하는 방법은 모든 손가락에 대해서 ROA를 반복함으로써 손가락의 yaw 움직임을 계산한다. 그림 5-(e)는 약지의 yaw 움직임 계산 결과를 보여주고, 그림 5-(f)는 모든 손가락에 대한 결과를 보여준다. 결과에서 보이는 것과 같이 손가락의 yaw 움직임은 정확하게 계산되었으나, 손가

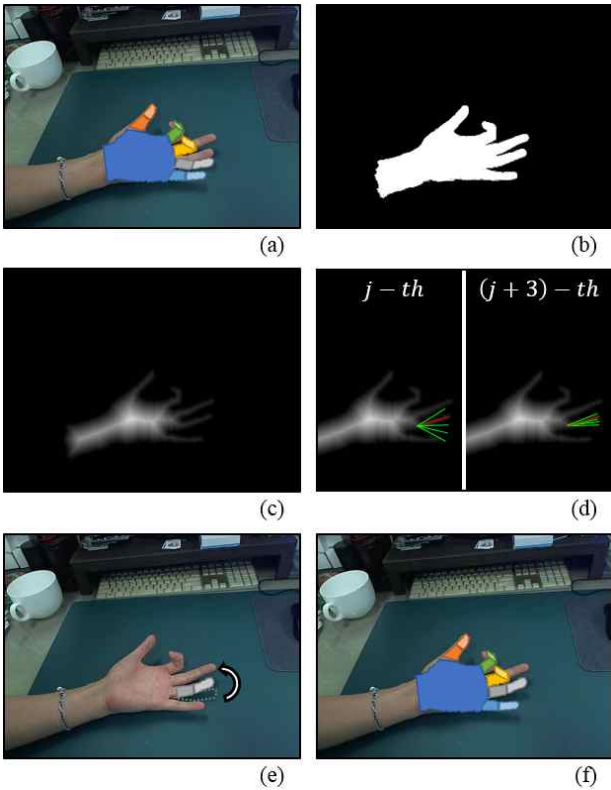


그림 5. 손가락 yaw 움직임 계산. (a) 손바닥 포즈 계산 결과, (b) 입력 영상으로부터 손 영역 검출 결과, (c) 거리 변환을 적용한 손 영역, (d) ROA 기반 손 yaw 움직임 계산, (e) 약지의 움직임 계산 결과, (f) 손가락 yaw 움직임 계산 결과

Fig. 5. The finger yaw pose estimation. (a) The result of the palm pose estimation, (b) hand region segmentation result of the input scene, (c) distance transformed hand region, (d) ROA-based finger yaw pose estimation, (e) estimation result of the ring finger, (f) result of the finger yaw pose estimation

락의 pitch 움직임은 여전히 일치하지 않는다. 다음 단계인 손가락 pitch 움직임 계산 단계에서는 pitch 움직임이 계산된다.

3. 손가락 pitch 움직임 계산

본 논문에서는 손가락의 pitch 움직임(X-각도, J-각도)을 계산하기 위한 방법으로 bi-directional particle swarm optimization (PSO)가 제안된다. 일반적인 PSO 기반 손 추적 방법은 식 5와 같은 움직임을 가진 파티클을 사용한다.

$$v_{i+1} = \mu(v_i + \alpha(p_i - x_i) + \beta(g - x_i)), \quad (5)$$

$$x_{i+1} = x_i + v_{i+1}.$$

여기서, x_i 는 v_i 의 속력을 가진 i 번째 파티클을 뜻하고, p_i 는 i 번째 파티클이 가졌던 포즈 중에서 가장 유사도가 높은 포즈를 뜻한다. 그리고 전체 파티클이 지녔던 포즈 중에서 가장 유사도가 높은 포즈는 g 에 저장된다. μ 는 0.73 이고, α 는 0 이상 2.8 이하의 그리고 β 는 0 이상 1.3 이하의 무작위 값을 가진다^[10,16]. 식 5의 파티클은 수렴하는 방향으로 움직이기 때문에 일반적인 PSO는 local minima에 빠질 수 있다는 단점을 가지고 있다. 특히 손가락은 가늘고 길다는 외향적 특징 때문에 손가락 포즈 계산은 쉽게 local minima에 빠진다. 예를 들어서 그림 5-(f)의 중지는 local minima에 빠진 상태이다. 따라서 일반적인 PSO를 손가락 추적에 사용하는 것은 효과적이지 못하고, 본 논문은 수렴

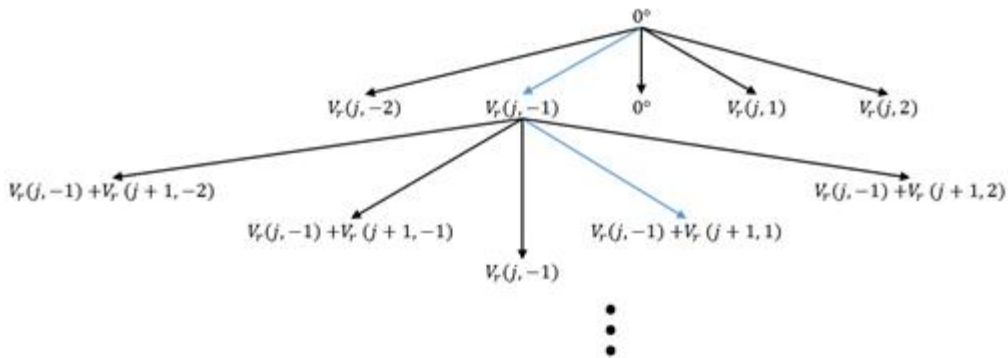


그림 6. 손가락 yaw 움직임을 위한 ROA
 Fig. 6. ROA for the finger yaw pose estimation

과 발산 두 종류의 방향으로 파티클이 움직이는 bi-directional PSO를 소개한다. 식 6은 bi-directional PSO의 파티클의 움직임을 보여준다.

$$\begin{aligned}
 v_{i+1}^+ &= \mu(v_i^+ + \alpha(p_i^+ - x_i^+) + \beta(g - x_i^+)), \\
 x_{i+1}^+ &= x_i^+ + v_{i+1}^+, \\
 v_{i+1}^- &= \mu(v_i^- + \lambda \frac{\alpha}{(p_i^- - x_i^-) + \epsilon} + \lambda \frac{\beta}{(g - x_i^-) + \epsilon}), \\
 x_{i+1}^- &= x_i^- + v_{i+1}^-.
 \end{aligned} \tag{6}$$

여기서, ϵ 는 0으로 나뉘지는 것을 막기 위해 더해졌으며, 본 논문에서는 1로 설정하였다. λ 는 표준화 상수이며 0.1로 설정하였다. 식 6에서 보이는 것과 같이 수렴 파티클(x_i^+)은 일반적인 PSO와 같이 수렴하는 방향으로 움직이며, 발산 파티클(x_i^-)은 local minima로 빠지는 것을 막기 위해서 발산하는 방향으로 움직이기 때문에, 손가락의 pitch 움직임을 강인하고 정확하게 계산할 수 있다.

손 모델의 외곽선을 투사시킨 뒤, 투사된 외곽선으로부터 수직한 방향으로 가장 가까운 손 영역의 외곽선을 찾고, 둘 사이의 거리를 구함으로써, 손가락 pitch 움직임 계산을 위한 유사도가 계산된다(그림 7)^[17]. 그림 8-(d)는 손가락 pitch 움직임 계산의 결과를 보여준다. 두 번째 단계에서 local minima에 빠져있던 중지가 세 번째 단계가 지난 다음에는 정확히 일치했다. 또한 bi-directional PSO는 병렬 연산이 가능하기 때문에, 본 논문에서는 손바닥 포즈 계산을 위한 ROA와 마찬가지로 CPU 기반 병렬 연산을 통해서 고속화 하였다.

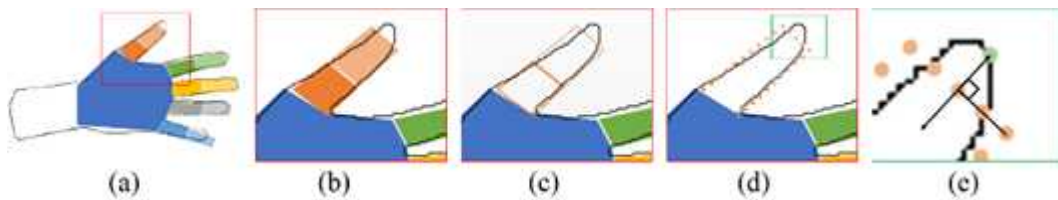


그림 7. 손가락 pitch 움직임 계산을 위한 유사도 측정 과정. (a) 손 모델 투사, (b) 투사된 엄지 영역, (c) 엄지 영역의 외곽선, (d) 외곽선의 표본 추출, (e) 수직한 방향으로 가장 가까운 외곽선 검출
 Fig. 7. Nearest edge detection processes for the discrepancy calculation of the finger pitch motion estimation. (a) Hand model projection, (b) thumb region of the projection, (c) contour of the thumb region, (d) contour sampling, (e) nearest edge detection on an orthogonal line

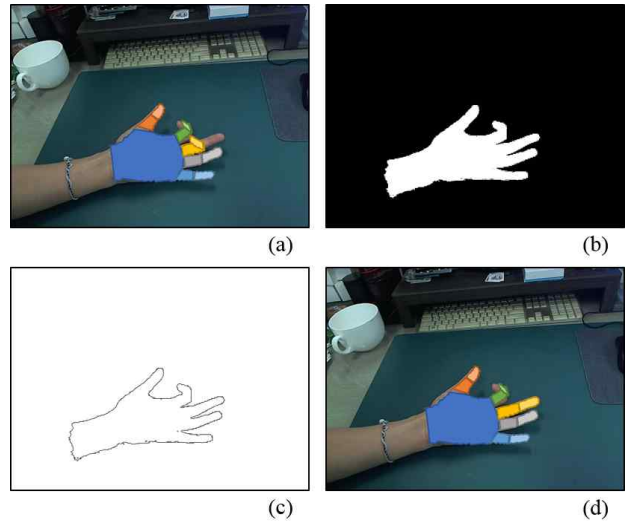


그림 8. 손가락 pitch 움직임 계산. (a) 손가락 yaw 움직임 계산 결과, (b) 입력 영상으로부터 추출한 손 영역, (c) 손 영역의 외곽선, (d) 손가락 pitch 움직임 계산 결과
 Fig. 8. The Finger pitch motion estimation. (a) The result of the finger yaw motion estimation, (b) hand region segmentation result of the input scene, (c) contour of the hand region, (d) result of the finger pitch motion estimation

V. 손 모델 재생성

사람들의 손은 각자 다른 모양을 가지고 있지만, 이런 차이를 정교한 3D 모델에 적용하는 것은 어렵다. 그러나 본 논문에서 사용하는 평면 손 모델은 현재 사용자의 손 모양에 맞춰서 변형하는 것이 가능하다. 제안하는 방법은 초기화 과정에서 평면 손 모델을 사용자의 손에 맞춰서 변형한다. 그림 9는 손 모델 재생성의 전체 과정을 보여준다. 그림

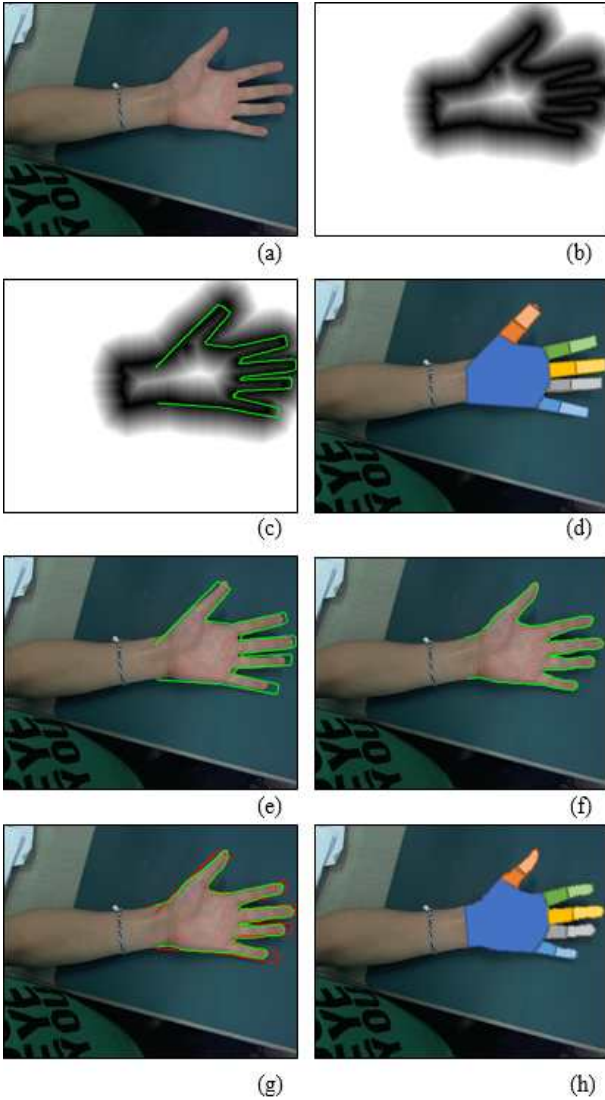


그림 9. 손 모델 재생성 과정. (a) 입력 영상, (b) 거리 변환된 손 영역의 외곽선, (c) 초기 손 모델의 투사와 거리 변환된 손 영역의 외곽선 사이의 차이 계산, (d) 손 모델 재생성을 위한 초기 추적 결과, (e) 초기 추적 후의 손 모델 투사의 외곽선, (f) 재생성된 손 모델의 외곽선, (g) 초기 모델(빨간색)과 재생성된 모델(초록색) 사이의 비교, (h) 제 생성된 모델

Fig. 9. The hand model regeneration. (a) Input scene, (b) distance transformed contour of the hand region, (c) discrepancy calculation based on convolution between the distance transformed contour of the hand region and projection of the initial hand model, (d) result of initial hand tracking for the hand model regeneration, (e) contour of the projection of the hand model after the initial hand tracking, (f) refined contour for the hand model regeneration, (g) comparison between the initial hand model (red line) and the regenerated hand model (green line), (h) the regenerated hand model

9-(a)에서 보이는 것과 같이 사용자의 손이 카메라에 촬영됐을 때, 제안하는 방법은 손 영역을 검출한 다음 손 영역 외곽선에 거리 변환을 적용하고(그림 9-(b)), 초기 포즈로 투사된 손 모델의 외곽선과의 convolution을 통해서 차이를 측정한다(그림 9-(c)). 여기서 차이가 만약 t 보다 낮으면 손이 초기 포즈의 손 모델과 거의 비슷하게 놓여 있다고 간주하고선 손 모델 재생성을 수행한다. 따라서 사용자는 반드시 그림 9-(a)와 (c)에서 보이는 것과 같이 손 모델과 비슷한 포즈로 손을 손 모델 근처에 두어야 한다.

손 모델을 재생성하기 위해서, 제안하는 방법은 손바닥 포즈 계산과 손가락 yaw 움직임 계산 과정을 수행한다. 그리고 그 결과는 그림 9-(d)와 같다. 그림 9-(e)에서 보이는 것과 같이 제안하는 방법은 손 모델을 초기 포즈로 투사시키고, 투사된 손 모델의 외곽선을 획득한다. 그리고 외곽선에서 수직인 방향으로 가장 가까운 손 영역의 외곽선 상의 점 (x_m, y_m) 을 검출한다(그림 9-(f)). 손 모델은 검출한 점을 3차원 점 (X_m, Y_m, Z_m) 으로 변환함으로써 획득이 되는데, 이때 사용하는 공식은 식 7과 같이 유도할 수 있다. 식 7은 3차원 점을 2차원 평면에 투사하는 공식을 보여주며, 본 논문에서 사용하는 모델은 평면이기 때문에 Z_m 를 0으로 설정할 수 있다. 따라서 식 7은 식 8로 유도될 수 있으며, 이 공식을 이용해서 3차원 점을 계산한다.

$$\begin{aligned}
 s \begin{bmatrix} x_m \\ y_m \\ 1 \end{bmatrix} &= I[R|t] \begin{bmatrix} X_m \\ Y_m \\ Z_m \\ 1 \end{bmatrix} \\
 &= \begin{bmatrix} f_x & s & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{11} & r_{11} & r_{11} & t_3 \end{bmatrix} \begin{bmatrix} X_m \\ Y_m \\ Z_m \\ 1 \end{bmatrix} \\
 &= \begin{bmatrix} p_1 & p_2 & p_3 & p_4 \\ p_5 & p_6 & p_7 & p_8 \\ p_9 & p_{10} & p_{11} & p_{12} \end{bmatrix} \begin{bmatrix} X_m \\ Y_m \\ Z_m \\ 1 \end{bmatrix}.
 \end{aligned} \tag{7}$$

$$\begin{bmatrix} p_{12}x_m - p_4 \\ p_{12}y_m - p_8 \end{bmatrix} = \begin{bmatrix} p_1 - p_9x_m & p_2 - p_{10}x_m \\ p_5 - p_9y_m & p_6 - p_{10}y_m \end{bmatrix} \begin{bmatrix} X_m \\ Y_m \end{bmatrix}. \tag{8}$$

그림 9-(g)는 초기 모델(빨간색)과 재생성된 모델(초록색) 사이의 외곽선 비교를 보여주며, 그림 9-(h)는 재생성된 모델을 보여준다. 결과에서 보이는 것과 같이 재생성된 손 모델(그림 9-(h))는 초기 모델(그림 9-(d))보다 현재 사용자의 손 모양과 비슷한 모양을 가지고 있다. 특히, 손가락 끝과 손목 부분이 더 잘 일치하는데, 이것은 제안하는 방법의 정확도를 향상시킨다. 다음 장에서는 실험을 통해서 이를 입증한다.

VI. 실험

이전 장에서 언급한 것과 같이 제안하는 방법은 다음과 같은 특징을 가지고 있다.

- 손 모델 재생성
- bi-directional PSO
- 단계별 추적 전략
- 실시간 연산

위의 특징을 검증하기 위해서 본 논문에서는 다양한 실험을 수행하였다. 첫 번째로 평면 손 모델의 장점을 설명하기 위해서 우리는 손 모델 재생성을 수행했을 때와 수행하지 않았을 때의 추적 결과 비교를 통해서 손 모델 재생성의 유용성을 보여준다. 두 번째로 일반적인 PSO와 bi-directional PSO의 성능 비교를 통해서 bi-directional PSO가 더욱 정확히 손가락을 추적할 수 있다는 것을 입증한다. 또한, 단계별 추적 전략은 이전 단계의 추적 결과가 다음 단계에 영향을 미칠 것이기 때문에 이를 분석함으로써 단계별 추적 전략의 한계를 분석한다. 그리고 각 단계별로 연산 시간을 측정하며, 마지막으로 제안하는 방법의 성능과 한계에 대해서 조사한다. 위의 실험 수행을 위해서 가상 데이터를 사용하였는데, 이는 손 모델을 다양하게 변형시켜가면서 만들었으며, 640×480의 해상도에 400장으로 구성되어 있다.

1. 손 모델 재생성의 효과

손 모델 재생성의 효과를 살펴보기 위해서, 우리는 손 모델 재생성을 수행했을 때와 하지 않았을 때 두 경우에서 제안하는 방법의 정확도를 측정하였다. 추적 결과와 가상 데이터 사이의 에러를 측정하기 위한 방법으로써, 대부분의 손 추적 방법은 Euclidean 거리를 측정한다. 그러나 제안하는 방법에서는 손 모델이 현재 사용자의 손 모양에 맞게 변형될 뿐만 아니라, 추적 결과는 각도(손바닥과 손가락)와 이동 거리(손바닥) 두 종류의 데이터로 구성되어 있기 때문에 Euclidean 거리를 측정하기 어렵다. 따라서 본 논문에서는 Chamfer 거리를 사용해서 에러를 측정하였다. Chamfer 거리는 식 3과 비슷한 방법으로 계산된다. 가상 데이터의 손 외곽선에 거리 변환을 적용한 다음 거리 변환된 가상 데이터와 추적 결과 사이의 convolution을 수행함으로써 거리를 계산한다.

표 2에서 보이는 것과 같이, 제안하는 방법은 손 모델 재생성을 사용했을 경우 1.08을, 사용하지 않았을 경우 1.67의 에러를 보여줬다. 손 모델 재생성을 사용함으로써 에러가 35% 가량 감소한 것을 알 수 있다. 그림 10과 11은 가상 데이터와 실제 영상에 대해서 손 모델 재생성을 사용했을 때와 사용하지 않았을 때의 추적 결과를 보여준다. 이전 장에서 재생성된 손 모델의 손목과 손가락 끝 부분이 현재의 사용자 손 모양과 더 잘 일치한다고 설명했는데, 그림 10을 보면 특히 손 모델 재생성을 사용했을 경우 추적 결과의 1, 3, 4, 5번째 열에서 손목과 손가락 끝이 더 정확히 일치하는 것을 알 수 있다. 또한, 손 모델 재생성으로 인한 성능 향상은 그림 11의 실제 영상의 추적 결과에서도 확인할 수 있다. 손 모델 재생성을 사용하지 않았을 경우 추적 결과에서 손가락이 일치하지 않고 local minima에 빠지는 것을 자주 확인할 수 있었으며, 손 모델 재생성을 사용했을 경우에는 정확히 추적이 성공하였다.

표 2. 손 모델 재생성의 효과
Table 2. Effectiveness of the hand model regeneration

	Error
The proposed method without the hand model regeneration	1.67
The proposed method with the hand model regeneration	1.08

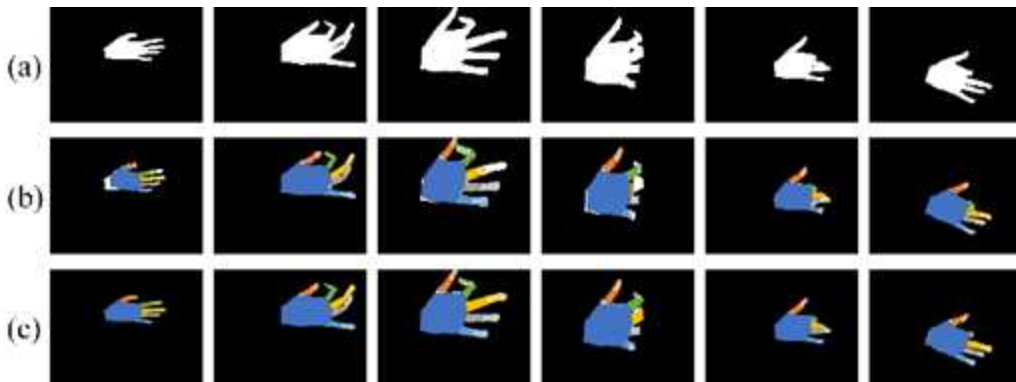


그림 10. 가상 데이터에 대한 제안하는 방법의 손 추적 결과. (a) 가상 데이터, (b) 손 모델 재생성을 사용하지 않은 경우의 손 추적 결과, (c) 손 모델 재생성을 사용했을 경우의 손 추적 결과

Fig. 10. Results of the proposed method with synthetic sequence. (a) Synthetic sequence, (b) results of the proposed method without the hand model regeneration, (c) results with the hand model regeneration

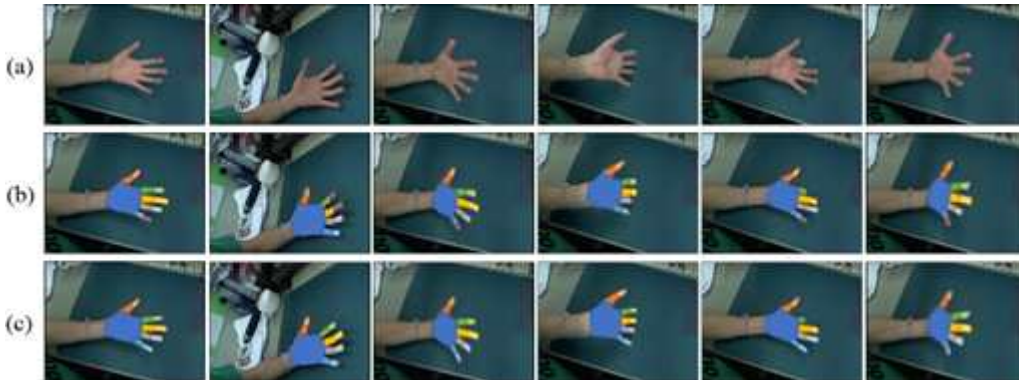


그림 11. 실제 영상에 대한 제안하는 방법의 손 추적 결과. (a) 실제 영상, (b) 손 모델 재생성을 사용하지 않은 경우의 손 추적 결과, (c) 손 모델 재생성을 사용했을 경우의 손 추적 결과

Fig. 11. Results of the proposed method with real scenes. (a) Real scenes, (b) results of the proposed method without the hand model regeneration, (c) results with the hand model regeneration

2. Bi-directional PSO의 효과

Bi-directional PSO의 효과를 입증하기 위해서 그림 12-(a)와 같이 가상 데이터는 검지의 J-각도가 급격하게 변하는 구간이 존재한다. 그림 13은 가상 데이터의 J-각도 값과 일반적인 PSO와 bi-directional PSO의 추적 결과의 J-각도 값을 보여준다. 그림 12-(b)와 (c)는 5에서 7과 16에서 18 구간의 추적 결과를 보여준다. 결과에서 보이는 것과 같이 가상 데이터의 J-각도는 0°와 90°사이를 빠르게 왔다 갔다 한다. 본 실험에서 bi-directional PSO는 두 종류의 파티클을 10개씩 사용

하며, 일반적인 PSO는 한 종류의 파티클을 20개 사용함으로써 두 방법의 전체 연산량을 비슷하게 조정하였다. 일반적인 PSO는 급격한 변화에 취약했으며, 특히 5에서 7과 16에서 17 구간에서 급격한 변화를 따라가지 못하고 local minima에 빠졌다. 이와 대조적으로 bi-directional PSO는 이런 급격한 변화에도 강인하게 J-각도를 추적하였다. 그림 14는 실제 영상에 대한 일반적인 PSO와 bi-directional PSO의 추적 결과를 보여준다. 결과에서 보이는 것과 같이 일반적인 PSO는 local minima에 빠져서 손가락을 제대로 추적하지 못했고, 이와 반대로 bi-directional PSO는 정확하게 추적하였다.

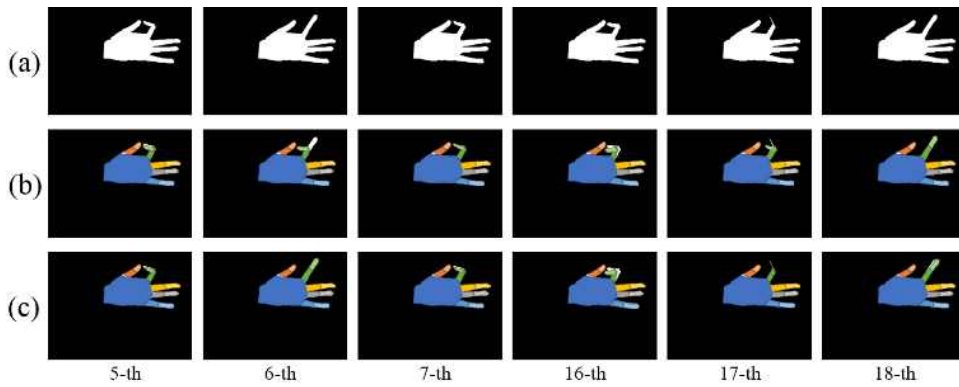


그림 12. Bi-directional PSO의 효과. (a) 가상 데이터, (b) 일반적인 PSO의 결과, (c) bi-directional PSO의 결과
 Fig. 12. Effectiveness of bi-directional PSO. (a) The synthetic sequence, (b) results of general PSO, (c) bi-directional PSO

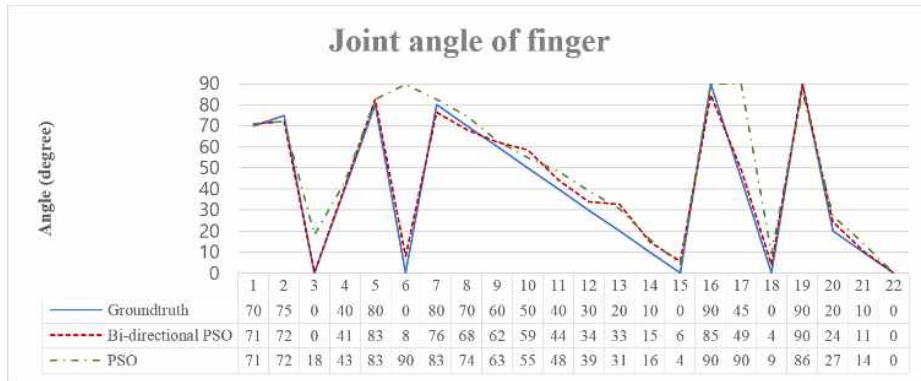


그림 13. Bi-directional PSO와 일반적인 PSO 사이의 성능 비교
 Fig. 13. Effectiveness comparison between Bi-directional PSO and general PSO

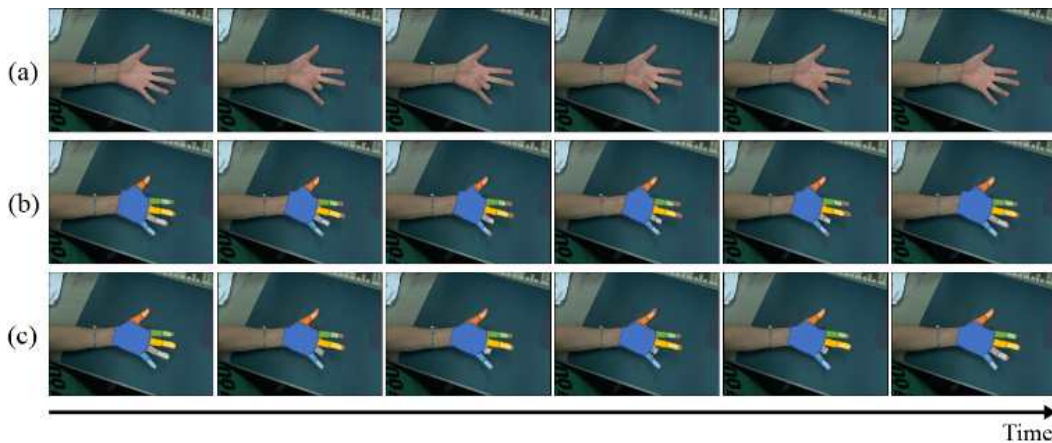


그림 14. 실제 영상에 대한 bi-directional PSO의 성능. (a) 실제 영상, (b) 일반적인 PSO의 결과, (c) bi-directional PSO의 결과
 Fig. 14. Effectiveness of bi-directional PSO with real scenes. (a) Real scenes, (b) results of general PSO, (c) bi-directional PSO

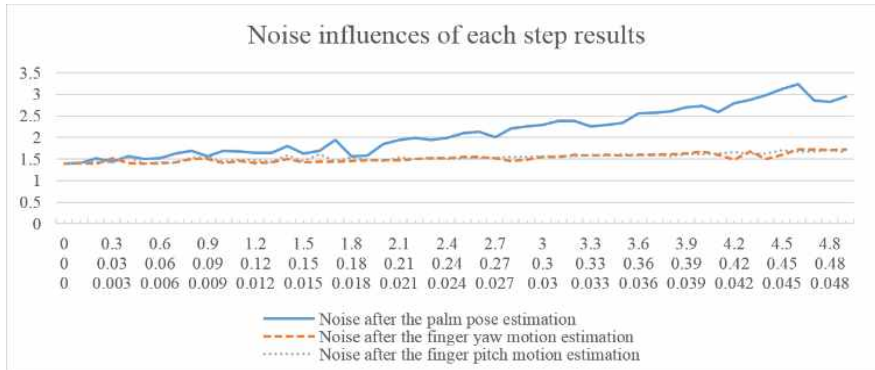


그림 15. 각 단계 결과의 노이즈 영향 분석
 Fig. 15. Noise influences analysis of each step results

3. 단계별 추적 전략 분석

제안하는 방법은 단계별 추적 전략에 기반 하기 때문에, 이전 단계의 추적 결과가 다음 단계의 추적 결과에 영향을 미친다. 그 영향을 분석하기 위해서 우리는 세 종류의 실험을 수행하였다. 첫 번째로 손바닥 포즈 계산 단계를 수행한 후 가우시안 노이즈를 손바닥 포즈 계산 결과에 추가한 뒤, 손가락 yaw와 pitch 움직인 계산 단계를 수행하였다. 두 번째로 손바닥 포즈 계산과 손가락 yaw 움직임 계산 단계가 끝난 후 손가락 yaw 움직임 계산 결과에 노이즈를 추가하였다. 세 번째로 모든 단계가 끝난 후 손가락 pitch 결과에 노이즈를 추가하였다. 여기서 가우시안 노이즈의 평균은 0 그리고 translation, rotation, 손가락 각도의 표준 편차를 각각 0에서 0.05, 0.5, 5까지 변화하도록 하였다. 그림 15는 세 종류 실험에서의 추적 결과의 Chamfer 거리 기반 에러를 보여준다. 그림에서 보이는 것과 같이 노이즈가 증가하면 에러 또한 증가한다. 또한 손바닥 포즈 계산의 노이즈가 전체 손 추적 결과에 가장 큰 영향을 미치는 것을 알 수 있다. 이것은 손바닥 포즈 계산 결과가 제안하는 방법의 성능에 있어서 가장 중요하다는 것을 알 수 있다.

전체 손 추적 성능에 손바닥 포즈 계산이 가장 큰 영향을 미치기 때문에, 우리는 두 종류의 실험을 통해서 손바닥 포즈 계산의 정확도를 측정하였다. 하나는 제안하는 방법의 모든 단계를 이용해서 손을 추적하는 것이고, 다른 하나는 손바닥 포즈 계산 단계만을 이용해서 손을 추적하는 것이다. 또한

우리는 두 종류의 실험에 대해서 두 종류의 에러를 측정하였다. 하나는 손바닥과 손가락을 포함한 손 전체 영역에 대한 에러를 측정한 것이고, 다른 하나는 손가락을 제외한 손바닥 영역만의 에러를 측정한 것이다. 표 3은 두 종류의 실험의 결과를 보여주며, 표를 통해서 우리는 두 종류의 결과를 도출할 수 있다. 하나는 손바닥 영역의 에러가 손가락 영역보다 적다는 것이며, 이는 손가락 포즈 계산이 상당히 정확하다는 것을 의미한다. 그리고 손바닥 포즈 계산의 강건함을 알 수 있다. 손바닥 포즈 계산 과정은 손바닥 영역뿐만 아니라 손가락 영역의 정보도 이용하기 때문에, 손가락 포즈가 일치하지 않을 경우에는 정확도가 떨어진다. 그러나 손가락 포즈가 전혀 맞지 않음에도 불구하고 손바닥 포즈 계산 과정만을 이용해서 추적했을 때도 손바닥은 정확히 추적되었다.

표 3. 손바닥 추적의 정확성
 Table 3. Accuracy of the palm pose estimation

Steps used for tracking	Region for accuracy calculation	Error
Using the whole of the proposed method	Palm and fingers	1.08
	Palm	0.53
Using only the palm pose estimation	Palm and fingers	2.56
	Palm	0.91

4. 연산 시간

제안하는 방법은 손바닥 포즈 계산, 손가락 yaw 움직임 계산, 그리고 손가락 pitch 움직임 계산, 총 3단계로 구성되

표 4. 제안하는 방법의 연산 시간
Table 4. Processing times of the proposed method

Steps of the proposed method	Processing time
Palm pose estimation	24.165 ms
Finger yaw motion estimation	0.090 ms
Finger pitch motion estimation	11.912 ms
Total	36.167 ms (27.65 fps)

어 있다. 표 4는 각 단계의 연산 시간을 보여준다. 실험을 위해 사용된 컴퓨터는 CPU i7 3.5GHz, 16GB RAM, 윈도우 7 64 bits로 구성되었으며, 사용된 영상은 640×480의 해상도를 가지고 있다. 제안하는 방법은 최적화가 이뤄지

지 않았고, GPU 기반 고속화를 사용하지 않았음에도 불구하고 실시간 연산이 가능했다. 표 4에서 보이는 것과 같이 손바닥 포즈 계산 과정이 가장 많은 연산 시간을 가지는데, 이는 위에서 묘사한 것과 같이 손바닥 포즈 계산 단계가 전체 성능에 가장 큰 영향을 미치기 때문에 손바닥 포즈 계산에 가장 큰 비중을 두었기 때문이다.

5. 제안하는 방법의 성능과 한계

그림 16은 실제 영상에 대한 제안하는 방법의 추적 결과를 보여준다. 결과에 보이는 것과 같이 제안하는 방법은 다



그림 16. 다양한 실제 영상에 대한 제안하는 방법의 결과
Fig. 16. Various results of the proposed method with real scenes



그림 17. 제안하는 방법의 한계
Fig. 17. The limitation of the proposed method

양한 손바닥과 손가락 포즈에 대해서 정확히 추적했다. 특히, 첫 번째 줄 그림의 손가락과 같이 서로 겹쳐진 경우에도 불구하고, 추적을 성공하였다. 또한 두 번째와 세 번째 줄에서와 같이 제안하는 방법은 넓은 범위의 pitch와 yaw 각도를 가진 손바닥에 대해서 정확히 추적하였다.

제안하는 방법은 단안의 RGB 카메라를 이용해서 손을 추적하기 때문에, 손 영역의 깊이 차이를 인지할 수 없다. 또한 일반적인 사람의 손가락은 4 DoF를 가진데 비하여 제안하는 방법은 3 DoF의 손가락 모델을 사용한다. 따라서 제안하는 방법은 그림 17에서와 같이 손가락에 의한 가려짐에 취약하며, 접힌 손가락을 정확히 추적하기 어렵다. 그림 16과 17의 실제 영상은 단순한 배경을 가지고 있는데, 제안하는 방법은 RGB 화소 값을 기반으로 손을 검출하기 때문에, 살색을 가진 물체가 존재하는 복잡한 배경에 대해서는 정확히 동작하기 어렵고 추적 결과가 불안정하다는 단점을 가지고 있다.

VII. 결론

본 논문에서는 단안의 RGB 카메라를 이용하는 손 추적 방법을 제안하였다. 제안한 방법은 25 fps가 넘는 속도로 동작하며, 손가락의 빠른 움직임에 강인하게 동작 가능하다. 각 단계에서 사용하는 알고리즘의 효율성은 다양한 실험을 통해서 입증되었으며, 각 단계에서 수행하는 역할에 적합하게 구성되었다. 또한 손 모델 재생성은 현재 사용자의 손 모양에 맞춰서 손 모델을 변화시킴으로써 강인한 추적이 가능하게 해준다. 무엇보다도 제안하는 방법은 CPU 기반 연산만을 사용하기 때문에, 비록 현재 장비들의 성능이 약간 부족하지만, 추후에는 모바일 폰을 포함한 다양한

환경에 적용 가능할 것으로 예상된다.

우리는 입력 영상으로부터 손 영역을 검출하는 과정에 상대적으로 관심이 덜 했기 때문에, 제안하는 방법은 복잡한 배경과 가려짐에 취약하다. 따라서 우리는 현재 단안의 RGB 카메라 기반 강건한 손 검출 방법에 연구 중에 있다. 또한 모바일 폰과 같은 장비에 구현을 위한 최적화 작업을 진행 중에 있다.

참고 문헌 (References)

- [1] Google Glass, <http://www.google.com/glass/>.
- [2] J.-H. Kim, N. D. Thang, and T.-S. Kim, "3-D hand motion tracking and gesture recognition using a data glove," Proc. of IEEE International Symposium on Industrial Electronics, pp. 1013-1018, 2009.
- [3] R. Y. Wang and J. Popović, "Real-time hand-tracking with a color glove," ACM Transactions on Graphics, vol. 28, no. 3, 2009.
- [4] D. Kim, O. Hilliges, S. Izadi, A. Butler, J. Chen, I. Oikonomidis, and P. Olivier, "Digits: freehand 3D interactions anywhere using a wristworn gloveless sensor," Proc. of the 25th annual ACM Symposium on User Interface Software and Technology, pp. 167-176, 2012.
- [5] P. Mistry and P. Maes, "SixthSense: a wearable gestural interface," Proc. of ACM SIGGRAPH ASIA 2009 Sketches, 2009.
- [6] Z. Zhao and A. Elgammal, "Spatiotemporal pyramid representation for recognition of facial expressions and hand gestures," Proc. of 8th IEEE International Conference on Automatic Face & Gesture Recognition, pp. 1-6, 2008.
- [7] C. Cao, Y. Sun, R. Li, and L. Chen, "Hand posture recognition via joint feature sparse representation," Optical Engineering, vol. 50, no. 12, 127210, 2011.
- [8] J. Choi, J. Park, H. Park, and J.-I. Park, "iHand: an interactive bare-hand-based augmented reality interface on commercial mobile phones," Optical Engineering, vol. 52, no. 2, 027206, 2013.
- [9] B. Stenger, A. Thayananthan, P. H. S. Torr, and R. Cipolla, "Filtering using a tree-based estimator," Proc. of 9th IEEE International Conference on Computer Vision, vol. 2, pp. 1063-1070, 2003.
- [10] I. Oikonomidis, N. Kyriazis, and A. A. Argyros, "Efficient model-based 3D tracking of hand articulations using kinect," Proc. of the 22nd British Machine Vision Conference, 2011.

- [11] I. Oikonomidis, N. Kyriazis and A.A. Argyros, "Tracking the articulated motion of two strongly interacting hands", Proc. of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 18-20, 2012.
- [12] S. Sridhar, A. Oulasvirta, and C. Theobalt, "Interactive markerless articulated hand motion tracking using RGB and depth data," Proc. of IEEE International Conference on Computer Vision, pp. 2456 - 2463, 2013.
- [13] C. Xu and L. Cheng, "Efficient hand pose estimation from a single depth image," Proc. of IEEE International Conference on Computer Vision, pp. 3456-3462, 2013.
- [14] K. Li, I.-M. Chen, S. H. Yeo, and C. K. Lim, "Development of finger-motion capturing device based on optical linear encoder," Journal of Rehabilitation Research & Development, vol. 48, no. 1, pp. 69-82, 2011.
- [15] OpenMP Fortran Application Program Interface, <http://www.openmp.org/>.
- [16] M. Clerc and J. Kennedy, "The particle swarm - explosion, stability, and convergence in a multidimensional complex space," IEEE Transactions on Evolutionary Computation, vol. 6, no. 1, pp. 58-73, 2002.
- [17] T. Drummond and R. Cipolla, "Real-time visual tracking of complex structures," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 24, no. 7, 2002.

저 자 소 개

최 준 영



- 2007년 : 한양대학교 전자전기컴퓨터공학부 학사
- 2009년 : 한양대학교 전자컴퓨터통신공학과 석사
- 2009년 ~ 현재 : 한양대학교 전자컴퓨터통신공학과 박사 과정
- 주관심분야 : 증강현실, 컴퓨터그래픽스비전, 3차원 영상처리, 인간컴퓨터 상호작용

박 종 일



- 1987년 : 서울대학교 전자공학과 학사
- 1989년 : 서울대학교 전자공학과 석사
- 1995년 : 서울대학교 전자공학과 박사
- 1992년 ~ 1994년 : 일본 NHK 방송기술연구소 객원연구원
- 1995년 ~ 1996년 : 한국방송개발원 선임연구원
- 1996년 ~ 1999년 : 일본 ATR 지능영상통신연구소 연구원
- 1999년 ~ 현재 : 한양대학교 공과대학 컴퓨터공학부 교수
- 주관심분야 : 가상현실, 컴퓨터그래픽스비전, 3차원 영상처리, 인간컴퓨터 상호작용