

MLESAC 움직임 추정 기반의 파티클 필터를 이용한 3D 얼굴 추적 (3D Face Tracking using Particle Filter based on MLESAC Motion Estimation)

성 하 천[†] 변 혜 란^{**}
(Hacheon Sung) (Hyeran Byun)

요 약 3D 얼굴 추적(Face tracking)은 보안감시, HCI (Human-Computer Interface), 엔터테인먼트(Entertainment) 등 컴퓨터 비전과 관련된 여러 분야의 핵심 기술로서 많은 연구가 진행되고 있다. 하지만, 광범위한 응용분야에도 불구하고 3D 얼굴 추적의 기본적인 높은 연산 비용으로 인하여 그 응용 분야가 모바일 단말기 등의 저 사양 플랫폼에는 많은 한계가 있어왔다. 본 논문에서는 이러한 3D 얼굴 추적의 연산 비용을 효과적으로 해결하고 폭 넓게 응용 분야를 확대하기 위하여 MLESAC(Maximum Likelihood Estimation by Sampling Consensus)을 이용한 움직임 추정(Motion Estimation) 기법과 기존의 파티클 필터(Particle Filter)를 결합하여 실행 속도 면에서 빠르면서도 성능 면에서도 우수한 3D 얼굴 추적 알고리즘을 제안한다.

키워드 : 3D 얼굴 추적, MLESAC, 움직임 추정, 파티클 필터

Abstract 3D face tracking is one of essential techniques in computer vision such as surveillance, HCI (Human-Computer Interface), Entertainment and etc. However, 3D face tracking demands high computational

cost. It is a serious obstacle to applying 3D face tracking to mobile devices which usually have low computing capacity. In this paper, to reduce computational cost of 3D tracking and extend 3D face tracking to mobile devices, an efficient particle filtering method using MLESAC(Maximum Likelihood Estimation SAmple Consensus) motion estimation is proposed. Finally, its speed and performance are evaluated experimentally.

Key words : 3D face tracking, MLESAC, motion estimation, particle filter

1. 서 론

3D 얼굴 추적[1-5]은 보안감시, HCI, 엔터테인먼트 등 컴퓨터 비전과 관련된 여러 분야의 핵심 기술로서 많은 연구가 진행되고 있다. 특히, 파티클 필터는 다양한 방법들 중에서도 3D 얼굴 추적에 매우 효과적인 방법으로 잘 알려져 있으며, 또한, 속도 및 성능 향상을 위하여 최근에도 다양한 연구들이 활발히 진행되고 있다. [4,5]는 [1]에서 제안된 3D 실린더 모델을 기반으로 파티클 필터와 LK(Lucas-Kanade) 알고리즘을 결합하여 PC 환경에서 실시간으로 동작 가능한 3D 얼굴 추적 알고리즘을 제안하였으며, [6]는 Random Projection과 RANSAC(Random Sampling Consensus)을 이용하여 사진에 획득한 입력 영상간의 3D 변환관계를 파티클 필터와 결합하여 역시 효과적이고 실시간으로 동작 가능한 3D 얼굴 추적 알고리즘을 제안했다. 물론, [4-6]에서 제안된 3D 얼굴 추적 기법들이 효과적이고 또한 일반 PC 환경에서 실시간으로 동작하지만, 여전히 높은 연산 비용으로 인하여 모바일 단말기 등의 저 사양 플랫폼에서는 적용하기 힘든 점이 있다. 특히, 모바일 단말기가 대중화되어 있는 오늘날 3D 얼굴 추적이 오직 PC 등의 고 사양 플랫폼에서만 적용 가능하다는 것은 3D 얼굴 추적의 응용 분야 확대의 장애가 되고 있다.

본 논문에서는 3D 얼굴 추적의 높은 연산 비용 문제를 해결하고 응용 분야를 저 사양 플랫폼으로 확대하기 위해서 기존의 파티클 필터를 효과적으로 개선하여 속도와 성능을 향상시키기 위한 방법으로 움직임 추정 기반의 파티클 필터를 제안한다.

2. 관련 분야 배경 지식

2.1 파티클 필터를 이용한 3D 얼굴 추적

3D 공간상에서 얼굴의 상태 θ 는 식 (1)과 같이 정의되며, t_x, t_y, t_z 는 공간상의 위치 파라미터 w_x, w_y, w_z 는 공간상의 회전 파라미터를 나타낸다.

$$\theta = [w_x \ w_y \ w_z \ t_x \ t_y \ t_z]^T \quad (1)$$

샘플링 기법을 기반으로 하는 파티클 필터[7,8]를 이

· 이 논문은 2008년 정부(교육과학기술부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업 연구임 (NRF-2008-313-D00998)

· 이 논문은 제36회 추계학술발표회에서 'MLESAC 움직임 추정 기반의 파티클 필터를 이용한 3D 얼굴 추적'의 제목으로 발표된 논문을 확장한 것임

† 비 회 원 : 연세대학교 컴퓨터학과
shc@cs.yonsei.ac.kr

** 종신회원 : 연세대학교 컴퓨터학과 교수
hrbyun@yonsei.ac.kr

논문접수 : 2009년 12월 24일

심사완료 : 2010년 4월 1일

Copyright©2010 한국정보과학회: 개인 목적이나 교육 목적인 경우, 이 저작물의 전체 또는 일부에 대한 복사본 혹은 디지털 사본의 제작을 허가합니다. 이 때, 사본은 상업적 수단으로 사용할 수 없으며 첫 페이지에 본 문구와 출처를 반드시 명시해야 합니다. 이 외의 목적으로 복제, 배포, 출판, 전송 등 모든 유형의 사용행위를 하는 경우에 대하여는 사전에 허가를 얻고 비용을 지불해야 합니다.

정보과학회논문지: 컴퓨팅의 실제 및 레터 제16권 제8호(2010.8)

용한 3D 얼굴 추적은 현재 입력 영상에서 추적 대상 즉, 얼굴의 3D 공간상의 상태 $\hat{\theta}_t$ 를 추정하는 문제로 인식된다. 현재의 입력 영상에서 $\hat{\theta}_t$ 는 관측값 $Z_{1:t}$ 에 의하여 정해지는 사후 확률밀도함수 $p(\theta_t | Z_{1:t})$ 에 의하여 추정되며, $p(\theta_t | Z_{1:t})$ 는 가중치가 적용된 파티클의 집합 $St = \{(\theta_t^{(j)}, w_t^{(j)})\}_{j=1}^N$ 의 우도(Likelihood)에 의하여 가중치 $w_t^{(j)}$ 의 pdf(Probability Density Function) 로 근사화 될 수 있다. 위의 과정은 상태 전이 모델(State transition model)과 관측 모델(Observation model)로 구성되는 동적 모델(Dynamic model)로 표현 된다.

State transition model : $\theta_t = f(\theta_{t-1}, u_t)$ (2)

Observation model : $Z_t = g(\theta_t, v_t)$ (3)

위의 식 (2),(3)에서 f 는 θ_{t-1} 에서 θ_t 로의 상태 전이 함수(Transition function), g 는 현재의 관측값 Z_t 와 상태 θ_t 의 관계를 나타내는 Measurement 함수를 정의 하며, u_t 와 v_t 는 각각 시스템 노이즈와 관측 노이즈를 정의 한다. 또한, 각 파티클의 가중치는 우도(Likelihood) $p(Z_t | \theta_t^{(j)})$ 에 의하여 아래와 같이 계산된다.

$$w_t^{(j)} = \frac{p(Z_t | \theta_t^{(j)})}{\sum_{i=1}^N p(Z_t | \theta_t^{(i)})}$$
 (4)

결과적으로, 파티클 필터의 성능은 식 (4)에 구해지는 가중치 $w_t^{(j)}$ 의 분포(distribution)에 의존적이 된다. 그런데, 식 (4)에 의하면 $w_t^{(j)}$ 는 실질적으로 식 (2),(3)에 의해서 정의된 동적 모델과 집합 S_t 의 원소 개수 N 에 의존적임을 알 수 있다. 즉, 파티클 필터의 성능에 대한 동적 모델과 N 의 관계는 아래와 같이 정의될 수 있다.

파티클 필터의 성능 \propto 동적 모델의 적합성 $\times N$ (5)

한편 식 (5)는 본 논문의 주요한 관점인 실행 속도 면에서 파티클 필터의 실행 속도 개선에 대한 실마리를 주고 있다. 일반적인 파티클 필터에서 N 은 t 번째 입력 영상에서 최종 $\hat{\theta}_t$ 를 추정하기 위해서 계산되는 가중치 $w_t^{(j)}$ 의 개수와 동일하므로 N 의 개수를 줄이는 것은 속도 향상과 직접적인 관련이 있다. 그러나, 식 (5)에 의하면 N 의 개수를 줄이면 이와 함께 파티클 필터의 성능이 감소된다. 따라서, N 의 개수를 줄이기 위해서는 동적 모델의 적합성이 먼저 향상되어야 한다.

2.2 샘플링과 MLE 기반의 움직임 추정

[9]에 의해서 제안된 MLESAC은 기존의 RANSAC 이 가지고 있던 임계값(Threshold)문제를 해결하고 샘플링 기반의 추정(Estimation) 기법의 성능을 향상시켰다. [9]에 의하면 MLESAC을 이용한 움직임 추정을 위

해서는 먼저 n 개의 대응점을 필요로 한다.

$$M_{MAP} = \arg \max_h [p(M_h | R_h)]$$

$$= \arg \max_h \left[p(R_h | M_h) \frac{p(M_h)}{p(R_h)} \right]$$
 (6)

식 (6)에서 n 개의 대응점 집합을 h 라고 하고, M_h 를 대응점 집합 h 의 움직임 가정(motion hypothesis), R_h 를 M_h 에 의해서 대응점 집합 h 에서 발생한 에러라고 한다면, 사후 확률(posterior probability) $p(M_h | R_h)$ 을 최대로 하는 M_h 를 구할 수 있다. 또한, 식 (6)에서 $p(M_h)$ 와 $p(R_h)$ 는 사전 확률(prior probability)로 이미 알려진 상수 값으로 가정 한다면, 식 (6)은 우도(Likelihood) $p(M_h | R_h)$ 의해서 아래와 같이 쓸 수 있다.

$$M_{MLESAC} = \arg \max_h [p(R_h | M_h)]$$
 (7)

또한, 식 (7)에서 전체에러 R_h 에서 i 번째 대응점의 에러를 r_i , i 가 유효한 대응점일 확률을 $p(v_i)$, M_h 에 의한 대응점들 간의 에러가 가우시안(Gaussian)이라고 가정하면 우도(Likelihood) $p(R_h | M_h)$ 는 아래와 같이 혼합 모델(Mixture model)로 표현 된다.

$$p(R_h | M_h) = \prod_{i=1}^n p(r_i | M_h)$$

$$= \prod_{i=1}^n \left(\left(\frac{1}{\sigma \sqrt{2\pi}} e^{-\frac{r_i^2}{2\sigma^2}} \right) p(v_i) + \left(\frac{1}{w_i} \right) (1 - p(v_i)) \right)$$
 (8)

식 (8)에서 w_e 는 대응점 i 가 유효하지 않을 확률 $1 - p(v_i)$ 에 대한 가중치 상수 값을 정의 한다. 또한, 연산의 효율성을 위해 식 (7)은 로그 우도(log likelihood)를 이용하여 아래와 같이 최종 정의된다.

$$M_{MLESAC} = \arg \max_h \left[\sum_i \log \left\{ \left(\frac{1}{\sigma \sqrt{2\pi}} e^{-\frac{r_i^2}{2\sigma^2}} \right) p(v_i) + \left(\frac{1}{w_i} \right) (1 - p(v_i)) \right\} \right]$$
 (9)

3. MLESAC 움직임 추정 기반의 파티클 필터

3.1 움직임 추정 기반의 파티클 필터 알고리즘

[7,8]에서 제안된 파티클 필터는 동적 모델에 의존적이다. 즉, 적합성이 낮은 동적 모델을 갖는 파티클 필터의 경우 올바른 추정을 위해서는 많은 개수의 파티클을 유지 해야 한다. 반면, 식 (5)에 의하면 향상된 동적 모델에 대해서는 적은 개수의 파티클에 의하여 좋은 성능을 보장할 수 있으며 또한, 실행 속도 역시 향상시킬 수 있다. 본 논문에서 제안하는 향상된 동적 모델은 식 (2)의 상태 전이 모델을 MLESAC 움직임 추정 모델로 변경함으로써 가능해진다. 일반적으로, 파티클 필터에서

상태 전이 모델은 이전 상태를 참조하므로 현재 상태에 대한 불확실성(Uncertainty)을 완전히 제거하기 어렵다. 이러한 불확실성은 결국 많은 파티클을 유지해야 하는 주요한 원인이 된다.

알고리즘 1 움직임 추정 기반의 파티클 필터

Initialization: 초기 단계에서 가중치가 동일한 N 개의 파티클 집합 $S_0 = \{(\theta_0^{(1)}, 1), \dots, (\theta_0^{(N)}, 1)\}$ 을 발생 시킨다.
 For $t = 1, 2, \dots, T$
 Motion Estimation: 알고리즘 2 에 의해서 움직임 $\tilde{\theta}_t$ 를 추정한다.

$$\tilde{\theta}_t = ME_{MLESAC}(S_{t-1})$$

$$S_{t-1} = \{(\theta_{t-1}^{(1)}, w_{t-1}^{(1)}), \dots, (\theta_{t-1}^{(N)}, w_{t-1}^{(N)})\}$$

For $n = 1, 2, \dots, N$

- (1) Particle Generation: 움직임 추정 $p(\tilde{\theta}_t)$ 에 의해서 파티클 $(\theta_t^{(n)}, 1)$ 을 발생시킨다.
- (2) Weight Computation: likelihood $p(Z_t | \theta_t^{(n)})$ 에 해서 가중치 w_t^n 을 계산한다.

End

Weight Normalization: 각 파티클의 가중치를 정규화한다.

$$w_t^{(j)} = \frac{p(Z_t | \theta_t^{(j)})}{\sum_{i=1}^N p(Z_t | \theta_t^{(i)})}$$

Final Estimation: 가중치에 의해서 최종 움직임을 추정한다.

$$\hat{\theta}_t = \arg \max_{\theta_t} w_t^{(n)}$$

End

알고리즘 1에서 제안된 움직임 추정 기반의 파티클 알고리즘은 기존의 파티클 필터와는 달리 $t-1$ 번째의 파티클 집합 S_{t-1} 은 MLESAC 움직임 추정 $\tilde{\theta}_t$ 를 구하기 위해서 사용되며 t 번째 파티클 집합 $S_t = \{(\theta_t^{(1)}, 1), \dots, (\theta_t^{(N)}, 1)\}$ 는 $p(\tilde{\theta}_t)$ 에 의하여 발생된다는 점이다. 결과적으로 사전에 획득 한 지식 $p(\tilde{\theta}_t)$ 에 의해서 발생된 파티클 집합 S_t 는 높은 확률로 최종 상태 추정 $\hat{\theta}_t$ 의 주변에 밀집하여 분포하게 되며 보다 적은 수의 파티클에 의해서도 3D 얼굴 추적 성능을 향상 시키고 동시에 빠르게 3D 얼굴 추적을 가능하게 한다.

3.2 파티클 필터를 위한 MLESAC 움직임 추정

본 논문에서 제안된 알고리즘 1은 움직임 추정을 기반으로 하고 있다. 물론, 움직임 추정을 위해서는 현존하는 수많은 움직임 추정 기법이 가능하다. 하지만, 몇 가지 이유로 인하여 MLESAC은 특히 저 사양 플랫폼에서 파티클 필터에 적용하기 적합한 움직임 추정 방법이라고 할 수 있다.

첫째, MLESAC은 파티클 필터와 같이 샘플링을 기반으로 하고 있기 때문에 기존의 다른 움직임 추정 방법과 비교하여 빠르게 실행된다.

둘째, MLESAC 기반의 움직임 추정에서 파티클 필터의 상태 θ 가 움직임 가설로 대체 가능하다. 그러므로, 기존의 MLESAC에서 움직임 가설을 구하기 위해서 필요했던 특징점 및 대응점 추출이 생략 가능하다.

알고리즘 2는 알고리즘 1에서 제안된 파티클 필터를 위한 MLESAC 움직임 추정을 보여 주고 있다. 앞에서 언급 되었듯이 알고리즘 2는 [9]에서 제안된 기존의 MLESAC 기반의 움직임 추정과는 달리 대응점 추출 과정을 생략하고 파티클 필터의 이전 상태에서 바로 움직임 가설을 샘플링한다. 즉, 알고리즘 2에서 샘플링된 $\theta_{t-1}^{(n)}$ 는 기존의 MLESAC에서의 움직임 가설 M_n 에 해당한다

알고리즘 2 파티클 필터를 위한 MLESAC 움직임 추정

Input: $t-1$ 에서 N 파티클을 갖는 집합 S_{t-1} 을 입력으로 한다.

$$S_{t-1} = \{(\theta_{t-1}^{(1)}, w_{t-1}^{(1)}), \dots, (\theta_{t-1}^{(N)}, w_{t-1}^{(N)})\}$$

For $n = 1, 2, \dots, N$

- (1) Sampling: 이전 파티클 집합의 가중치를 적용하여 새로운 파티클의 상태 $\theta_{t-1}^{(n)}$ 샘플링한다.
- (2) Feature Extraction: 입력 영상에서 $\theta_{t-1}^{(n)}$ 에 의하여 템플릿 특징 벡터 F 와 대응되는 특징 벡터 F' 을 구한다.

$$F = [v_1 \ \dots \ v_M], F' = [v'_1 \ \dots \ v'_M]$$

- (3) MLE Computation: 식(9)에서 $r_t = v_t - v'_t$ 로 하여 $MLE^{(n)}$ 을 계산한다.
- (4) Threshold decision: 만약, 정해진 임계값 T 에 대하여 아래 식을 만족하면 반복문을 벗어난다.

$$MLE^{(n)} \geq T$$

End

Output: 최대값을 갖는 MLE 에 해당하는 $\theta_{t-1}^{(n)}$ 을 최종 움직임 $\tilde{\theta}_t$ 로 추정 한다.

4. 실험 및 평가

4.1 실험 및 평가를 위한 구현

본 논문에서 제안한 알고리즘을 실제 3D 얼굴 추적에 적용하기 위하여 [1]에서 제안한 3D 실린더 모델과 [10]에서 제안한 특징 추출 방법을 이용하였다. PC 기반의 구현은 Windows Visual C++를 이용하여 Intel Pentium 4 CPU 2.80 GHz를 탑재한 PC에서 구현되었으며, 구현된 알고리즘은 BU Data Set[1]에서 평가되었다. 그림 1은 3D 얼굴 추적을 위한 특징 추출 모습을 보여 주고 있으며 추출된 특징은 본 논문에서 제안된 알고리즘 1과 2를 이용하여 3D 얼굴 추적이 가능하도록 구현되었다.

그림 2는 본 논문에서 제안된 알고리즘에 BU Data Set에서 3D 얼굴 추적 실행의 예를 보여주고 있다.

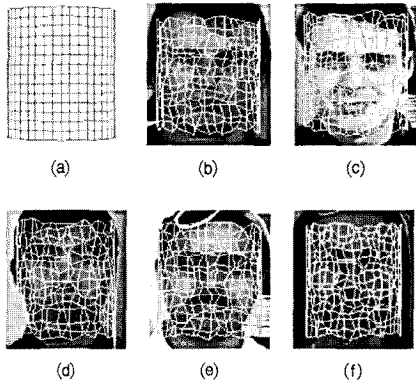


그림 1 3D 실린더 모델과 BU data set의 각 인물에 대한 특징 선택, (a) 3D 실린더 모델, (b),(c), (d),(e),(f) 각 jam1, jim1, llm1, sam1,vam1의 첫 번째 입력영상에서 선택된 특징들(직선이 교차된 지점)

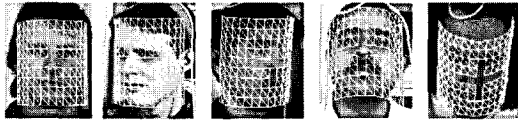


그림 2 BU data set에서의 3D 얼굴 추적의 예

4.2 성능 평가 방법

알고리즘의 성능 평가를 위해서 PC에서 구현된 3D 얼굴 추적을 이용하여 BU Data set에서 3D 얼굴 추적 성능 및 실행 속도가 측정되었다.

$$\text{추적율} = \frac{\text{추적된 영상의 개수}}{\text{입력 영상의 전체 개수}} \quad (10)$$

추적 성능은 위의 식 (10)에 의해서 정의된 추적율로 평가되었다. 식 (10)에서 “입력 영상의 전체 개수”는 BU Data Set 전체 이미지 개수(5명의 인물, 인물당 9개의 동영상 샘플, 각 동영상당 200개의 이미지 프레임)인 9000개가 되며, “추적된 영상의 개수”는 각 동영상에서 3D 얼굴 추적이 최초 실패하기 전까지 추적에 성공한 이미지 개수이다. 또한, 추적의 성공과 실패는 식 (11)에 의해서 판단되었다.

$$e_i^2 = [a_i, -\tilde{a}_i]^T \Sigma^{-1} [a_i, -\tilde{a}_i] \quad (11)$$

위의 식 (11)에서 a_i 는 i 번째 영상에서 3D 얼굴 추적에 의한 포즈 추정이며 \tilde{a}_i 는 이때의 Ground truth 값이다. Σ 는 모든 영상에서의 포즈 변화에 대한 공분산 행렬로 정의된다. 얼굴 추적은 e_i 가 정해진 임계값을 넘지 않으면 성공적인 것으로 판단하였다. 본 논문의 실험에서는 e_i 를 3.0으로 정하였다.

4.3 성능 및 속도 측정 결과

아래의 표 1은 MLESAC 움직임 추정 기반의 파티클 (MLESAC-PF) 필터와 기존의 상태 전이 모델에 의한 파티클 필터(STANDARD-PF)에서의 3D 얼굴 추적에 대한 성능 및 속도에 대하여 BU Data set에서 측정된 결과를 보여주고 있다.

표 1 MLESAC-PF와 STANDARD-PF의 성능 및 속도 비교

알고리즘 구분 파티클수	STANDARD-PF		MLESAC-PF	
	추적율(%)	추적 속도 (fps)	추적율(%)	추적 속도 (fps)
50	63.8	61	87.9	50
70	73.2	47	89.2	39
90	78.9	39	90.5	30
130	81.4	28	92.8	23
170	84.3	23	89.6	18
210	86.4	20	94.1	15

표 1과 그림 3의 결과에 의하면 MLESAC-PF에 의한 추적율은 50개의 파티클 만을 사용하더라도, STANDARD-PF에서 210개를 사용했을 때 보다 좋은 성능을 보이고 있다.

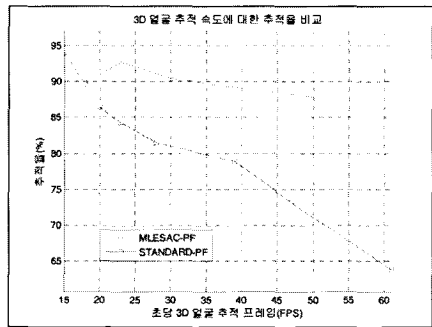


그림 3 3D 얼굴 추적 실행 속도와 성능에 대한 비교

그러므로, MLESAC-PF는 저 사양 플랫폼에서 3D 얼굴 추적의 속도 향상을 위해서 N 을 감소시킬 때 발생하는 성능 저하를 효과적으로 방지할 수 있으며, 이것은 본 논문에서 제안하는 MLESAC 움직임 추정 기반의 파티클 필터가 모바일 기기 등의 저 사양 플랫폼에 적합한 이유가 된다.

4.4 모바일 단말기에서의 3D 얼굴 추적

본 논문에서 제안된 알고리즘이 저 사양 플랫폼에서 효과적으로 동작 가능함을 보이기 위해서 일반적으로 사용되는 모바일 단말기에 탑재되어서 평가되었다. 제안된 알고리즘을 모바일 단말기에 탑재하기 위해서 실수



그림 4 SPH-M4650에서의 3D 얼굴 추적

연산의 정수 변환, 효율적인 메모리 사용 등의 일반적인 최적화 기법이 적용되었으며 “XSCALE PXA300”을 탑재한 “Windows Mobile 6.0”기반의 삼성 “SPH-M4650” 단말기에 구현 되었다.

그림 4는 BU Data set에서 50개의 파티클을 사용하였을 때 각각 1, 40, 80, 120,160, 200 번째 프레임에서의 추적 결과를 보여주고 있다. 이때의 3D 얼굴 추적 속도는 평균 8 fps로 실행되었다. 표 1에 의하면 파티클 필터의 개수가 50개일 때 추적율은 87.9%이다. 따라서, 본 논문에서 제안된 MLESAC 움직임 추정 기반의 파티클 필터에 의한 3D 얼굴 추적이 모바일 단말기 등의 저 사양 플랫폼에서 충분히 효과적임을 알 수 있다.

5. 결론

본 논문에서는 저 사양 플랫폼 기반의 3D 얼굴 추적을 위한 MLESAC 움직임 추정 기반의 파티클 필터를 제안했으며, 실험 및 평가에 의하여 충분히 효과적임을 입증했다. 물론, 본 논문에서 제안하는 방법 이외에 3D 얼굴 추적의 성능과 속도를 향상시킬 수 있는 방법은

다양하게 존재하지만, 본 논문에서 제안한 MLESAC 움직임 추정 기반의 파티클 필터는 서로 상호 보완적인 측면에서 최적화되었다는 점에서 의미가 있다. 즉, 기존의 파티클 필터는 MLESAC에 의하여 불확실성을 줄일 수 있으며, 또한, MLESAC은 파티클 필터에 의하여 대응점 추출을 생략함으로써 보다 효율적인 알고리즘이 된다. 이러한, 상호 보완적 성능 향상은 실제 모바일 단말기에서 구현으로 충분히 효과적임을 입증되었다. 또한, 본 논문에서 제안한 알고리즘이 현재 본 논문의 구현에서 적용된 특징 추출 및 최적화 보다 향상된 방법들이 적용된다면 훨씬 효과적일 것으로 생각된다.

참고 문헌

- [1] M. Cascia, S. Sclaroff, V. Athitsos, Fast, reliable head tracking under varying illumination: an approach based on registration of texture-mapped 3d model," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol.22, no.4, pp.322-336, 2000.
- [2] J. Xiao, T. Kanade, J. Cohn, "Roust full motion recovery of head by dynamic templates and re-registration technique," *Int. J. Imaging Syst. Technol.*, vol.13, pp.85-94, 2003.
- [3] G. Aggaward, A. Veeraraghavan, R. Chellappa, "3d facial pose tracking in uncalibrated videos," *International Conference on Pattern Recognition and Machine Intelligence*, vol.3776, pp.515-520, 2005.
- [4] S. Chio, D. Kim, "Robust head tracking using 3D ellipsoidal head model in particle filter," *Pattern Recognition*, vol.41, no.9, pp.2901-2915, 2008.
- [5] J. Sung, T. Kanade, D. Kim, "Pose Robust Face Tracking by Combining Active Appearance Models and Cylinder Head Models," *Int. Journal of Computer Vision*, vol.80, pp.260-274, 2008.
- [6] L. Lu, X. Dai, and gd Hager, "Efficient particle filtering using. RANSAC with application to 3D face tracking," *Image Vision. Computing*, vol.24, no.6, pp.581-592, 2006.
- [7] A. Doucet, S.J. Godsill, C. Andrieu, "On sequential Monte Carlo sampling methods for Bayesian filtering," *Statist. Comput.*, vol.10, no.3, pp.197-209, 2000.
- [8] M. Isard and A. Blake, "Condensation-conditional density propagation for visual tracking," *Int. Journal of Computer Vision*, vol.29, no.1, pp.5-28, 1998.
- [9] P.H.S. Torr and A. Zisserman, "MLESAC: a new robust estimator with application to estimating image geometry," *Computer Vision and Image Understanding*, vol.78, no.1, pp.138-156, 2000.
- [10] H. Sung, H. Byun, "Fast 3D face tracking using feature selection and particle filter," *Proc. of the KIISE Korea Computer Congress 2009*, vol.36, no.1(C), pp.546-551, 2009. (in Korean)