

# 증강 현실에서의 키프레임 기반 다수 객체 추적을 위한 적응적 키프레임 관리 알고리즘\*

김기영<sup>0</sup>, 우운택

광주과학기술원 U-VR 연구실

{kkim<sup>0</sup>, wwoo}@gist.ac.kr

## 요 약

본 논문에서는 증강 현실에서의 키프레임 기반 실시간 다수 객체 추적 알고리즘의 안정성 향상을 위한 적응적 키프레임 관리 방법을 제안한다. 영상에서 안정적으로 추적되고 있는 객체와 새롭게 등장한 객체의 키프레임 할당수를 평가 함수를 근거로 조절해, 제한된 객체 감지 과정에 소요되는 시간을 효율적으로 관리한다. 결론적으로 기존 고정식 키프레임 할당 방식보다 제한된 시간에 다수의 키프레임을 불안정한 객체에 더 할당해 영상의 잡음이나 블러에 강건한 안정적인 추적 환경을 제공한다.

## 1. 서론

최근 증강 현실과 상호작용형 응용 시스템에서 마커없이 다수 객체를 추적하는 방법에 대한 수요가 증가하고 있다 [2][3]. 일반적으로 다수 객체 추적은 단일 객체에 비해 대략 객체의 수만큼 복잡도가 증가하게 된다. 실시간성을 보장하면서 다수의 객체 6 DOF 자세값을 안정적으로 추출하는 것이 극복해야 할 핵심 부분이라고 할 수 있다.

본 논문에서는 기존의 키프레임 기반 다수 객체 추적 방식 [1]의 안정성 향상을 위한 적응적 키프레임 및 특징점 관리 방법을 제안한다. [1]에서는 일률적으로 각 객체별 최고반응 (Best-Response) 키프레임만을 고려해 영상 조건 (잡음, 블러 등)이 좋지 않을 경우, 매칭된 특징점 개수의 감소로 인해 감지가 효율적이지 못했다. 효율성과 강건성을 위해 본 논문에서는 실시간성을 보장하는 범위내에서 영상에 감지되는 객체의 수를 고려해 키프레임의 수를 동적으로 조절한다.

## 2. 키프레임 기반 다수 객체 추적

[1]의 시스템은 추적과 감지 프로세스를 각각의 코어에 할당하고, 영상 (키프레임) 탐색 방법을 결합해 실시간성과 가능 객체 수를 [2], [3]에 비해 개선했다.

기본적으로 객체의 3 차원 특징점들과 관련 키프레임들 (3 차원 특징점의 투영점 포함)이 주어졌다고 가정한다. 그림 1은 키프레임을 활용한 확장성 있는 다수 객체 추적 알고리즘의 특정 시간  $t$  와

$t+t_B$ 에서의 흐름을 나타낸다.  $(\mathbf{F}:F_1\sim F_3)$ 은 특징점을 추적하고,  $(\mathbf{B}:B_1\sim B_3)$ 은 특징점을 감지해  $(\mathbf{F})$ 에 전달한다. 각각은 카메라 영상을 공유하지만,  $(\mathbf{B})$ 는  $(\mathbf{F})$ 에 비해 상대적으로 느리다. 즉,  $(\mathbf{F})$ 는 매 프레임 단위로 호출되고  $(\mathbf{B})$ 는 그렇지 않게 된다.

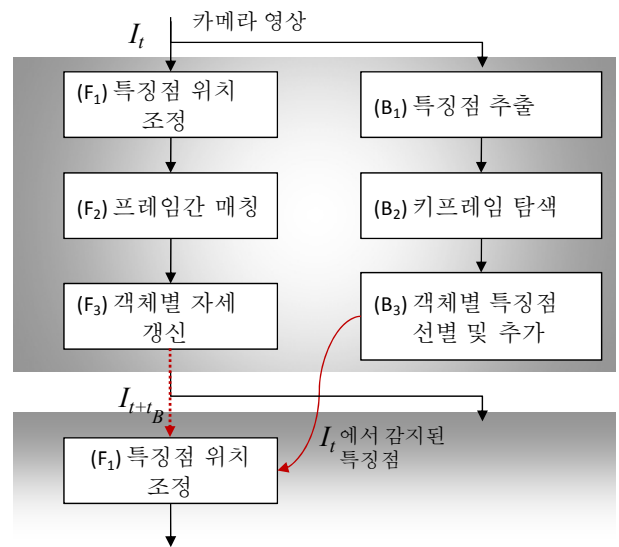


그림 1. 다수 객체 추적 알고리즘 흐름도

한 번 감지된 객체들의 자세 추정에 소요되는 시간은  $P(\mathbf{F}) = P(F_1) + P(F_2) + P(F_3)$ , (단,  $P$ 는 시간 측정 함수) 와 같고, 실질적인 추적 프레임률은  $1/P(\mathbf{F})$  FPS 과 같다.  $(\mathbf{B})$ 는  $(\mathbf{F})$ 에 현재 주어진 영상에서 감지된 객체의 특징점을 주기적으로 공급하는 역할을 담당한다. 그러므로,  $(\mathbf{B})$ 의 성능은 객체 자세의 재위치 (Re-localization)나 추적의 안정성 (드리프트)과 관계가 있다.

## 3. 적응적 키프레임 관리 방법

본 논문에서는  $(\mathbf{B})$ 의 과정 중, 특히  $(B_3)$ 의 과정을

\* 이 논문(저서)는 2010년 교육과학기술부의 재원으로 한국연구재단의 지원을 받아 <실감교류 인체감응솔루션> 글로벌프런티어연구개발사업으로 수행된 연구임 (한국연구재단-M1AXA003-20100029751).

효율적으로 관리하는 방법이다. (B<sub>2</sub>) 과정의 최종 결과는 객체별 카메라 영상과의 유사도에 따른 객체별 키프레임 리스트이다. 이를 (B<sub>3</sub>) 과정에서 (F) 과정에 전달할 특징점을 선별하는데 사용한다.

2.1 객체별 특징점 선별

(B<sub>3</sub>)에서는 객체에 주어진 키프레임 K 와 미리 저장된 3 차원 특징점들 간의 매칭을 수행해 적절한 특징점을 선택한다. 각 객체별로 두 단계 과정을 수행한다. 첫번째는 kd-Tree 를 이용한 특징점 설명자 기반의 매칭이다 (T 과정). 두번째는 기하 정보를 이용한 첫번째 단계의 아웃라이어를 제거하는 과정이다 (G 과정). Fundamental matrix 와 P-n-P 알고리즘으로 객체의 자세를 구해 아웃라이어를 제거한다.

2.2 키프레임 관리

첫번째 과정은 (B) 과정에서 실시간성을 보장하면서 사용할 수 있는 최대의 프레임 수 M 을 정하는 작업이다. 예를 들어, 목표치를 50 ms 로 정할 경우, M 은 최대 50/(P(T)+P(G)) 값을 갖는다.

두번째 과정은 M 개수의 키프레임들을 다음과 같이 i 번째 객체의 점수값 S<sub>i</sub> 를 계산해, (B<sub>3</sub>) 과정에서 사용되는 각 객체에 할당하는 작업이다.

$$S_i = \underbrace{\alpha(2 - Z(i))}_{Response} + \beta \underbrace{\frac{\sum_{j \in recent\ frames} R_j(i)}{N_{recent\ frames}}}_{Tracking\ Stability} \quad (1)$$

R(i) 함수는 i 번째 객체가 현재 영상에서 어느 정도로 추적되고 있는지에 대한 평가값이다. 최근 프레임에서의 (실제에서는 5 개) 채투영 오차의 평균이다. Z(i)는 (B<sub>2</sub>)과정에서 얻어지는 키프레임 리스트의 반응값이다. 기본적으로 L<sub>2</sub>-norm 의 2 에서 0 사이의 값으로 값이 높을수록 주어진 영상과 키프레임의 유사성이 높다. α (>0)와 β (>0)는 두 요소간 가중치이다. 즉, Z(i)가 높다는 것은 i 번째 객체가 현재 영상에서 존재할 확률이 다른 객체보다 높다는 것을 의미한다. 높은값의 객체는 한장의 키프레임만으로도 추적에 필요한 특징점을 충분히 확보가능하고, 낮은값의 객체는 다수의 키프레임들이 특징점 확보에 필요하게 된다.

최종적으로 S<sub>i</sub> 값에 따라 정렬하고, 정규화 과정을 수행한 다음, 높은 S<sub>i</sub> 객체에 많은 키프레임들을 할당한다. 그 후, 2.1 의 단계를 키프레임별로 수행한다. α 의 가중치를 높히는 것은 새롭게 등장한 객체나 특징점의 수가 적은 객체에 키프레임을 더 할당한다는 의미이다. β 의 가중치를 높히는 것은 이미 안정적으로 추적중인 객체에 키프레임을 할당하지 않는다는 의미이다.

4. 실험 결과 및 분석

실험을 위한 하드웨어로 CPU 2.9 GHz 쿼드코어, 엔비디아 GTX 285 그래픽카드, 640×480 해상도의 FireflyMV 카메라를 사용했다. (B<sub>1</sub>) 과정의 속도를 높이기 위해 SIFTGPU [4]를 사용했다. 그림 2는 실험에서 사용한 다수 객체들과 이를 추적한 결과를 시각화한 특정 프레임을 보여준다. 표 1은 각 모듈에서의 평균 수행 시간 측정된 결과를 보여준다.

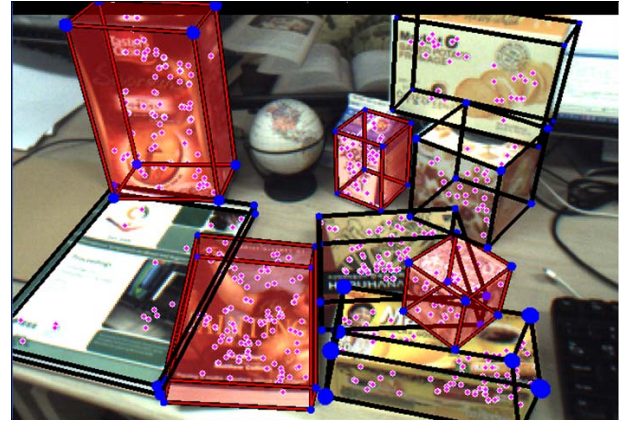


그림 2. 다수 객체 추적 영상, 총 9 (메모리에는 총 40 개의 객체각 적재) 개의 객체 자세가 (F) 과정에서 추적 속도 18 ms 로 갱신된 것으로, 모델과 특징점을 시각화한 것임.

표 1. 각 모듈에서의 수행 시간 (9 개 객체) 측정

	[1]의 방법	제안된 방법
(F)	2.3 + 4.36 + 14.48 ms	변화 없음
(B <sub>1</sub> ,B <sub>2</sub> )	20.56 + 12.20 ms	변화 없음
(B <sub>3</sub> )	18.82 ms	28.84 ms
# 키프레임	9 (9 개 객체)	14 (5 개 객체)

5. 결론

본 논문에서 제안한 키프레임 관리 방법은 듀얼 코어 방식의 프로그래밍이 가능한 모바일 장비나 저사양의 데스크탑에서 추적 및 감지 속도를 보장하는데 사용될 수 있을 것이다.

참고문헌

[1] K. Kim, V. Lepetit, W. Woo, "Keyframe-based Modeling and Tracking of Multiple 3D Objects," ISMAR, pp. 193-198, 2010.  
 [2] Y. Park, V. Lepetit, W. Woo, "Multiple 3D Object Tracking for Augmented Reality," ISMAR, pp. 117-120, 2008.  
 [3] D. Wagner, D. Schmalstieg, and H. Bischof, "Multiple target detection and tracking with guaranteed framerates on mobile phones," ISMAR, pp. 57-64, 2009.  
 [4] C. Wu, "SiftGPU: A GPU Implementation of Scale Invariant Feature Transform (SIFT)," <http://cs.unc.edu/~ccwu/siftgpu>, 2010.