

떨림 현상이 완화된 3차원 객체 추적

강민석, 박정식, *박종일

한양대학교 컴퓨터·소프트웨어학과

{mskang, nangsik}@mr.hanyang.ac.kr, *jipark@hanyang.ac.kr

3D Object tracking with reduced jittering

Kang, Minseok Park, Jungsik *Park, Jong-II

Department of Computer and Software, Hanyang University

요약

미리 저장된 객체의 3차원 특징점(Feature point) 좌표와 카메라 영상의 2차원 특징점 좌표를 매칭(Matching)하여 객체를 추적하는 방식의 경우, 카메라의 시점이 변할 때 특징점에서 발생하는 원근 효과(Perspective effect)가 반영되지 못하여 특징점 매칭 오류가 발생한다.

따라서 특징점에서 발생하는 원근 효과를 반영하여 정확한 카메라 포즈를 추정하기 위해 이전 프레임(Frame)의 카메라 포즈(Camera Pose)에 맞추어 텍스처가 포함 된 3차원 객체의 모델을 렌더링 하여 원근 효과를 적용한 후, 현재 카메라 영상과 특징점 매칭하여 프레임 사이의 카메라 움직임을 구하여 객체를 추적한다.

더 나아가 본 논문에서는 특징점 매칭에서 발생하는 작은 오류들로 인한 미세한 카메라 움직임은 2단계의 임계치(Threshold)를 적용하여 떨림 현상으로 간주하여 떨림 현상이 제거된 객체 추적을 수행한다. 매 프레임마다 카메라 포즈에 맞춘 추적 객체를 렌더링 하기 때문에 떨림 현상으로 간주되어 제거된 카메라 움직임은 누적되지 않고, 추적 오류도 발생시키지 않는다.

1. 서론

텍스처가 포함되어 있지 않은 객체의 3차원 추적은 객체의 3차원 모델과 카메라 영상에서 검출되는 객체의 윤곽선과 객체 3차원 모델이 카메라 포즈에 맞추어 사영(Projection)되는 윤곽선과의 비교를 통해 이루어진다. 이전 카메라 영상과 현재 카메라 영상 사이의 카메라 움직임(Camera motion)을 추정하여 카메라 6자유도(6DOF) 포즈와 객체의 위치를 예상한다[1]. 하지만 추적하려는 객체의 표면에 텍스처가 있을 경우 객체의 윤곽선뿐만 아니라 객체 표면에서도 엣지가 검출되고 이것은 엣지방식의 객체 추적을 어렵게 한다. 추적 하려고 하는 객체가 대칭형의 모양을 가지고 있고, 카메라 영상내의 객체의 움직임이 대칭축에 평행하게 회전 했을 때, 엣지를 사용한 추적 방식으로는 객체의 대칭축에 평행한 3차원 움직임을 알아낼 수 없으며 객체의 위치를 정확하게 추적하는데 있어서 오류를 일으킨다. 따라서 추적하려는 객체가 대칭형의 모양을 가지고 있다면 엣지 방식으로 추적하는데 어려움이 있다.

객체에 텍스처가 있는 경우 미리 저장시킨 객체의 3차원 특징점과 카메라 영상 내에서 추적하려는 객체의 2차원 특징점을 매칭하여 카메라 6자유도와 객체의 위치를 추정할 수 있다[2]. 객체의 3차원 특징점은 미리 저장하여 사용하게 되는데 특징점을 저장하였을 때의 카메라

시점으로 고정된 특징점을 가지게 된다. 따라서 3차원 특징점 저장 당시와 카메라 시점이 변화된 상태에서 객체를 찍게 되면 원근 효과가 반영되지 않고 이로 인하여 객체의 3차원, 2차원 특징점을 매칭 하는데 있어서 오류가 발생한다.

이러한 카메라 시점에 맞추어 원근 효과를 적용하여 정확한 객체의 위치를 추적하기 위하여 본 논문에서는 이전 카메라 포즈로 객체를 렌더링 한 영상에서 검출되는 2차원 특징점에서 3차원 좌표를 복원한 후, 현재 카메라 영상에서 검출되는 2차원 특징점과 매칭하여 카메라 모션[3], [4]을 구하고 원근효과가 반영 된 특징점 매칭을 수행함으로써 더욱 정확한 객체 추적을 수행한다.

3차원 객체 위치의 정확한 추정은 평면 객체의 추적과 달리 노이즈(noise)에 매우 민감하다. 특징점 매칭에서 발생하는 서브픽셀 문제 등에 의해서도 카메라 포즈의 정확한 추정에 오차가 발생하며, 떨림 현상이 발생한다.

본 논문에서는 추적과정에서 발생하는 떨림 현상을 2단계의 임계치를 적용하여 추정한 후, 제거하는 방법을 제시한다. 매 프레임마다 객체를 렌더링하여 다시 추적하기 때문에 제거된 카메라 모션은 추적 오류를 발생시키지 않는다.

* 교신저자

2. 특징점 매칭을 이용한 카메라 포즈의 추정

객체의 3차원 위치와 카메라 포즈의 추정은 객체의 3차원 특징점과 카메라 영상에서의 객체의 2차원 특징점을 매칭하여 추정된다[2]. 본 논문에서는 추적 객체로 스테퍼 모터(Stepper motor)로 제작한 턴테이블(Turntable)을 활용하여 객체의 텍스처와 3차원 모델을 손쉽게 얻을 수 있는 원통형 객체를 선택하였다.

그림1과 같이 턴테이블을 이용하여 얻어낸 원통형 객체의 텍스처에서의 각각의 픽셀은 원통형 객체의 구조를 알고 있으므로, 3차원 좌표로의 전환이 가능하다.

3차원 좌표로 전환된 특징점은 카메라 영상에서의 2차원 특징점과 매칭되고, 매칭 된 특징점을 perspective-n-point[2] 알고리즘을 사용하여 카메라 포즈를 추정한다.



그림 1. 객체의 3차원 특징점과 카메라 영상의 2차원 특징점은 서로 매칭되고 카메라 포즈가 추정된다. 원근 효과가 고려되지 않은 3차원 특징점은 매칭 오류를 일으킨다.

텍스처 각각의 픽셀 위치에서 3차원 좌표로 변환된 특징점의 매칭은 원근 효과가 고려되지 않고, 이러한 3차원 특징점은 매칭 오류를 일으키고 떨림 현상이 발생한다.

3. 원근 효과와 떨림 현상이 고려된 3차원 객체 추적

perspective-n-point을 이용한 객체 3차원 위치 추정 방법의 경우 [2], 특정 카메라 시점에서의 특징점을 저장하여 사용하므로 카메라 시점이 변할 때 발생하는 원근 효과를 반영하지 못하고, 이것은 특징점 매칭 오류를 발생시킨다. 발생되는 오류는 객체의 정확한 위치를 추정하지 못하게 하므로 객체의 위치를 추적할 때 떨림 현상이 발생된다.



그림 2. 이전 프레임에서의 카메라 포즈로 객체를 렌더링 시켜서 원근 효과를 적용한 후, 현재 프레임에서의 카메라 영상과 특징점 매칭한다. 잘못 매칭 되는 특징점의 수가 현저히 적어진다.

이러한 원근 효과를 적용하고 정확한 객체 추적을 수행하기 위해,

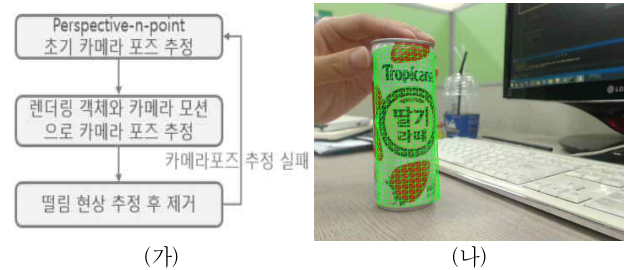


그림 3. (가) 카메라 모션을 활용한 객체 추적의 흐름도, (나) 카메라 모션을 활용하여 원근효과가 적용 된 객체의 추적

이전 카메라 프레임에서 추정된 카메라 포즈로 렌더링된 객체 영상과 카메라 모션[3], [4]을 활용하였다.

카메라 모션[3], [4]을 활용한 객체 추적은 초기 카메라 포즈와 이전 프레임 영상에서 매칭되는 특징점의 3차원 좌표가 필요하다.

perspective-n-point[2] 알고리즘으로 추정된 카메라 포즈는 오차 허용 범위 내에서 카메라 포즈가 추정되므로 카메라 모션을 활용하는 객체 추적 방법에서의 초기 카메라 포즈로 사용될 수 있다. 렌더링된 객체 영상에서의 2차원 특징점들을 3차원 좌표로 복원하는 과정은 본 논문의 4번 소제목에서 설명한다.

초기 카메라 포즈와 이전 프레임에서 매칭되는 객체의 3차원 특징점을 구할 수 있었고, 카메라 모션[3], [4]을 활용하여 원근 효과가 적용된 객체의 위치를 추적할 수 있다. 더 나아가 2단계의 임계치를 적용하여 떨림 현상이 완화된 객체의 추적을 수행한다.

4. 렌더링 된 객체 2차원 특징점의 3차원 좌표 복원

렌더링 된 객체 영상에서의 특징점은 3차원의 특징점이 2차원으로 사영되었으므로 렌더링 영상의 2차원 특징점을 3차원으로 복원 시켜야 한다.

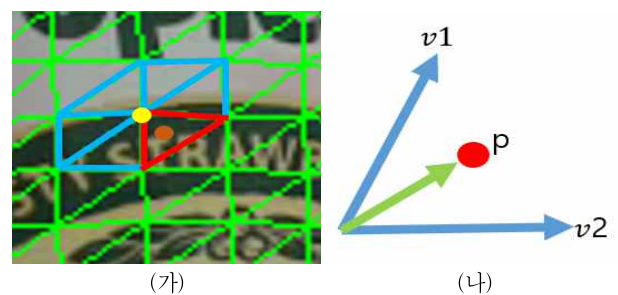


그림 4. (가)객체 모델의 사영된 2차원 좌표와 렌더링 영상에서 검출된 2차원 특징점 좌표 (나)벡터 연산을 수행하여 2차원 특징점 좌표를 3차원 좌표로 복원한다.

$$\begin{aligned} \vec{v1} &= u2 - u1, \vec{v2} = u3 - u1 \\ \vec{v3} &= p - u1 \\ t1\vec{v1} + t2\vec{v2} &= \vec{v3} \end{aligned} \tag{1}$$

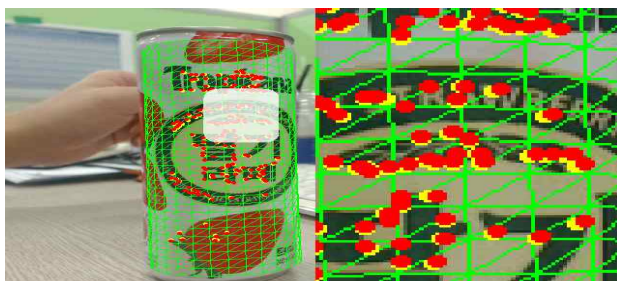
$$0 \leq t1 \leq 1, 0 \leq t2 \leq 1, t1 + t2 \leq 1 \tag{2}$$

주어진 카메라 포즈로 객체를 렌더링 하였을 때, 객체의 앞면은 그려지고 뒷면은 그려지지 않을 것이다. 3차원 객체 모델의 앞면의 모든 점과 면들을 2차원 영상으로 카메라 포즈에 맞추어 사영시킨다. 렌더링된 객체 영상의 2차원 특징점의 위치에서 사영된 3차원 객체의 모델의 점 중에서 가장 가까운 점을 찾는다. 가장 가까운 점을 포함하는 2차원으로 사영된 객체 모델의 모든 면에서 식(1)의 벡터 연산을 수행하여 조건(2)을 만족하는 t_1, t_2 과 그에 해당하는 하나의 면을 찾을 수 있고, 특징점 p 는 실수값 t_1, t_2 와 사영된 객체 모델의 좌표 u_1, u_2, u_3 로 나타낼 수 있다.

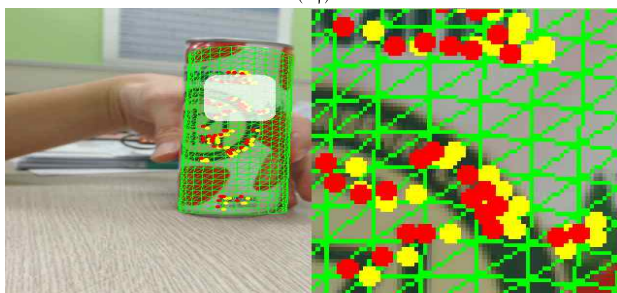
식(1)에서 2차원으로 사영된 객체 모델 특정 면의 세 점을 u_1, u_2, u_3 이고, p 는 렌더링 영상에서의 2차원 특징점 위치이다. v_1, v_2, v_3 는 객체 모델로부터 사영된 점 u_1, u_2, u_3 와 특징점 p 으로부터 생성되는 벡터이고, t_1, t_2 는 식(1)과 조건(2) v_3 를 만족시키는 임의의 실수 값이다.

객체모델의 사영된 2차원 좌표 u_1, u_2, u_3 와 그에 대응하는 객체 모델의 3차원 좌표를 알므로, 사영된 2차원 특징점 p 와 t_1, t_2 값으로 벡터 연산을 수행하여 2차원 특징점 p 의 3차원 위치를 복원할 수 있다

5. 2단계의 임계치를 이용한 떨림 현상의 판단



(가)



(나)

그림 5. (가)떨림 현상에 의해 발생하는 카메라 모션은 일관 되지 않는 방향으로 발생하고 (나)객체의 움직임에 의해 발생하는 카메라 모션은 일관된 움직임을 보인다.

카메라 모션[3],[4]의 추정에는 이전 카메라 포즈로 렌더링된 객체 영상의 특징점과 현재 카메라 영상의 특징점을 매칭함으로써 이루어진다. 이 때, 매칭 되는 특징점들 각각의 매칭 쌍의 위치가 서로 멀리 떨어져 있다면 객체가 많이 움직인 것이고 가까이 있다면 객체는 적게 움직인 것이다.

$$\sum_{i=1}^N \frac{(x_p(i) - x_c(i))^2 + (y_p(i) - y_c(i))^2}{N} \leq T_1 \quad (3)$$

$$\left| \sum_{i=1}^N \frac{(x_p(i) - x_c(i))}{N} \right| + \left| \sum_{i=1}^N \frac{(y_p(i) - y_c(i))}{N} \right| \leq T_2 \quad (4)$$

식(3), (4)의 $x_p(i), x_c(i), y_p(i), y_c(i)$ 는 각각 이전 프레임의 카메라 포즈로 렌더링된 영상과, 현재 프레임의 카메라 영상에서 검출되는 특징점들이 서로 매칭되는 매칭쌍의 위치이다. N 은 매칭되는 특징점의 개수, T_1, T_2 는 떨림 현상인지, 매우 적은 객체의 움직임인지, 객체의 움직임인지를 판단하는 임계값을 나타낸다. 식(3)으로 이전 프레임과 현재 프레임 사이에서 객체가 얼마만큼 움직였는지 측정한다. 객체의 움직임이 임계치 T_1 보다 클 경우에는 객체가 움직인 것으로 판단하며, 임계치 T_1 보다 작은 움직임을 보였을 경우에는 떨림 현상이거나 매우 작은 객체의 움직임으로 간주한다.

식(3)의 임계치 T_1 보다 작은 움직임은 떨림 현상인지 매우 작은 객체의 움직임인지 판단하기 위해 식(4)로 검사하게 된다. 매칭쌍에서 식(3)의 T_1 보다 작은 객체의 움직임이 발견되었지만 모든 특징점 쌍에서 관찰된 객체 움직임에서 일관성이 발견되면 식(4)의 값은 큰 값을 가지게 될 것이고 식(3)의 임계치 T_1 보다 적게 움직였다 할지라도 객체의 움직임 일 것이다. 반대로 떨림 현상일 경우에는 매칭쌍에서 일관 되지 않게 움직임이 발생되므로 식(4)의 특징점 쌍들의 더하는 값들이 부호가 다르기 때문에 식(4)의 값들이 많이 상쇄되어 작은 값을 가지게 될 것이다. 즉, 특징점의 매칭쌍으로부터 계산되는 식(3), (4)의 값이 임계값 T_1, T_2 보다 작은 값을 가져야만 떨림 현상으로 간주할 수 있을 것이다. 식(2), (3)으로 판단한 떨림 현상은 제거하여 떨림 현상이 감소된 객체 추적을 수행하게 된다.

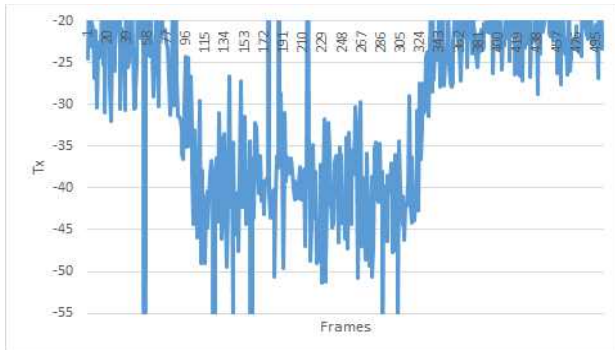
6. 실험결과

perspective-n-point[2] 알고리즘으로 객체를 추적하는 방식, 렌더링된 객체와 카메라 모션을 활용하여 원근 왜곡을 제거한 후 객체를 추적하는 방식[3], [4], 마지막으로 본 논문에서 제시하는 렌더링된 객체와 카메라 모션을 활용하고 임계값을 지정하여 떨림 현상을 개선한 방식의 3가지 추적 방식으로 객체를 추적하였다. 동일한 실험 영상을 사용하여 각각의 방식들이 객체의 위치를 어떻게 추적하는지 비교하고 제안된 방법에서 떨림 현상이 개선되는지를 실험하였다.

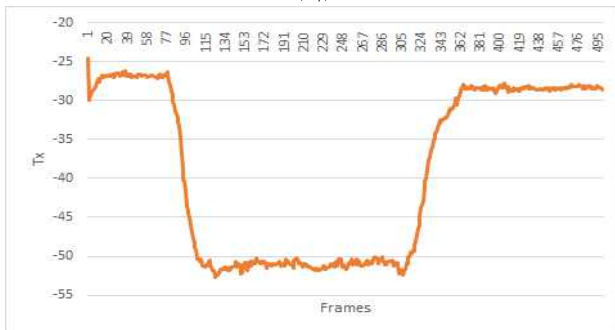
perspective-n-point[2] 알고리즘 방법의 객체 추적에서는 추적 객체의 크기가 다양하게 나타나므로 다중 스케일(Scale)의 특징점 검출과 매칭이 필요하여 SIFT[5]를 활용하였다.

렌더링된 객체와 카메라 모션을 활용하는 방법에서는 초기 포즈의 추정에는 perspective-n-point[2]를 사용하였고, 카메라 모션을 구하는 과정에서는 초기 카메라 포즈에서 같은 스케일에서의 객체의 움직임을 활용하므로 단일 스케일의 특징점을 검출하는 FAST[6]로 특징점을 검출하였다. 매칭에 있어서는 렌더링된 객체와 실제 카메라 영상의 특징점 매칭이기 때문에 강한 매칭이 필요하였고, SIFT[5] 기술자(Descriptor)로 매칭을 수행하였다. 마지막으로 본 논문에서 제안하는 식(3), (4)을 활용하여 떨림 현상을 개선시켜 보았다.

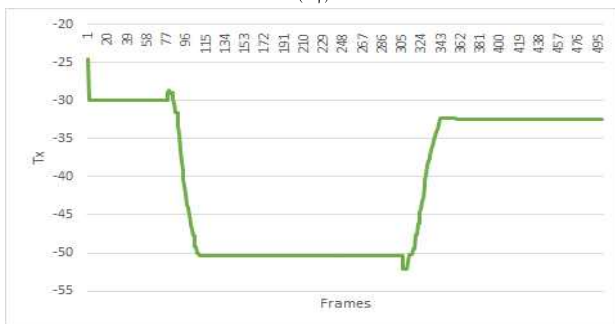
각각의 방법의 차이점을 확인하기 위해 500프레임 크기의 동일한 영상을 사용하여 실험을 진행했으며, 객체가 정지했을 경우의 떨림 현상 완화를 측정하기 위해 실험 영상에서 객체를 3번 정지 시켜보았다.



(가)



(나)



(다)

그림 6. 동일한 영상에서 각각의 방법으로 추적된 객체의 위치를 프레임에 따라 나타낸 모습. (가)perspective-n-point, (나)렌더링 객체와 카메라 모션, (다)카메라와 모션과 떨림 현상 임계치를 적용한 객체의 위치 추적.

그림6은 각각의 프레임에서 추적되는 객체의 실제 3차원 좌표 중 가로 방향의 좌표 값이며, 각각의 방법으로 수행한 실험 결과는 그림6의 (가),(나),(다)에서 확인 할 수 있다.

perspective-n-point[2]을 활용하여 객체를 추적한 방법에서는 원근 효과가 적용되지 않아서 특징점 매칭 오류가 발생하게 되고, 이러한 문제로 떨림 현상이 매우 크게 나타났다. 그림6의 (가)에서 객체의 위치는 매우 큰 변동폭을 보이고 있으며, 계속해서 떨림현상이 크게 발생한다.

렌더링 된 객체와 카메라 모션을 활용한 방법을 사용하였을 경우에는 [2]를 이용한 방법보다 떨림 현상이 작게 나타났으나, 객체가 정

지했을 경우에 미세한 떨림 현상이 계속해서 발생하는 것을 그림6의 (나)에서 확인할 수 있다.

마지막으로 본 논문에서 제안하는 임계치로 떨림 현상을 판단하여 판단된 떨림 현상을 제거하는 방법에서는 객체가 정지 하였을 때, 효과적으로 떨림 현상을 개선시킨 것을 그림6의 (다)에서 확인 할 수 있었다.

7. 결론

옛지 방식의 객체 추적은 객체가 표면에 텍스처를 포함하고 있거나, 대칭형의 형태를 가지고 있을 경우에는 추적하는데 있어서 어려움이 있다.

perspective-n-point을 이용한 객체 추적의 방법은 카메라 시점의 변화에 따라 특징점에서의 원근 효과가 고려되지 않고, 매칭 오류가 일어나며 정확한 카메라 포즈가 추정되지 않는다.

이러한 원근 효과를 적용하여 특징점을 매칭 하기 위해 3차원 객체의 모델을 렌더링하고 카메라 모션을 활용하여 객체 추적을 수행하였다. 3차원 객체의 추적의 경우에 평면객체와는 다르게 매우 정확한 카메라 포즈를 요구하지만 특징점 매칭에서 서브픽셀 문제 등으로 발생하는 미세한 오류로 인해 객체의 정확한 위치가 파악되지 않아서 떨림 현상이 발생하게 된다.

이러한 떨림 현상을 완화시키기 위하여 본 논문에서는 떨림 현상을 임계치를 사용하여 제거하는 방법을 제안했다. 제시된 방법이 효과적으로 객체 추적에서 발생하는 떨림 현상을 감소시킬 수 있는 것을 실험 결과를 통해 확인할 수 있었다.

감사의 글

본 연구는 미래창조과학부 및 정보통신산업진흥원에서 추진하는 "정보통신연구기반구축사업"으로 수행한 연구결과임(과제번호 I2221-14-1005)

참고문헌

- [1] Lepetit, Vincent, and Pascal Fua. Monocular model-based 3D tracking of rigid objects. Now Publishers Inc, 2005.
- [2] Hartley, Richard, and Andrew Zisserman. Multiple view geometry in computer vision. Cambridge university press, 2003.
- [3] Bregler, Christoph, and Jitendra Malik. "Tracking people with twists and exponential maps." Computer Vision and Pattern Recognition, 1998. Proceedings. 1998 IEEE Computer Society Conference on. IEEE, 1998.
- [4] Drummond, Tom, and Roberto Cipolla. "Application of lie algebras to visual servoing." International Journal of Computer Vision 37.1 (2000): 21-41.
- [5] Lowe, David G. "Object recognition from local scale-invariant features." Computer vision, 1999. The proceedings of the seventh IEEE international conference on. Vol. 2. Ieee, 1999.
- [6] Rosten, Edward, and Tom Drummond. "Machine learning for high-speed corner detection." Computer Vision - ECCV 2006. Springer Berlin Heidelberg, 2006. 430-443.