

<https://doi.org/10.7236/IIBC.2018.18.3.35>

IIBC 2018-3-6

두 객체가 충돌하는 상황에서의 햅틱 렌더링 알고리즘

Haptic Rendering Algorithm for Collision Situation of Two Objects

김선규*, 김혜빈**, 류철***

Seonkyu Kim*, Hyebin Kim**, Chul Ryu***

요약 본 논문에서는 햅틱 렌더링을 위한 동영상의 장면을 정의하고 그 중 정지 상태의 객체와 단일 객체가 충돌하는 상황에 대한 햅틱 렌더링 알고리즘을 제안한다. 객체의 움직임을 크게 네 가지 상황으로 분류하였으며 동영상에서 쉽게 볼 수 있는 장면들을 활용하였다. 제안하는 알고리즘은 움직임 예측과 객체 추적을 활용하여 객체의 방향 변화와 모양 변화를 검출함으로써 햅틱 렌더링을 하기에 적합한 프레임인지 판별할 수 있도록 하였다. 알고리즘 실행을 통해 샘플 동영상에서 총 13개의 프레임을 추출하였으며 해당 프레임이 재생되는 시간을 계산하였다. 햅틱 편집 프로그램을 통해 상황에 알맞은 햅틱 생성 파형을 추가함으로써 계산된 재생 시간에 햅틱 효과가 나타나는 것을 확인하였다.

Abstract In this paper, we define a haptic rendering algorithm for a situation that has collision between static object and single object. We classified video scenes into four categories which can be easily seen in video sequence. The proposed algorithm can detect which frame is suitable for haptic rendering by detecting the change of direction using motion estimation and change of shape using object tracking. As a result, a total of 13 frames are extracted from the sample video and playing time of these frames were calculated. We confirmed that the haptic effect appears in expected playing time by adding the appropriate haptic generating waveform through the haptic editing program.

Key Words : Haptic, Video Sequence, Object Tracking, Motion Estimation

1. 서 론

햅틱 렌더링이란 가상 환경의 가상 객체의 성질과 사용자 사이의 상호작용에 따라 생성되는 힘을 계산하고 생성하는 방법을 말한다[1]. 최근 들어 여러 햅틱 장치를 이용하여 사용자에게 햅틱 피드백을 줄 수 있게됨에 따라, 동영상에 햅틱 기능을 추가하고 이를 햅틱 장치를 통해 사용자에게 전달하는 연구가 진행되고 있다. 가상 현실 시스템에서 낚시를 할 수 있도록 도와주는 햅틱 장치

인 FishBot을 활용하여 사용자로 하여금 실제 낚시를 하는 듯한 느낌을 햅틱 장치를 통해 느낄 수 있으며 교육 분야에서도 햅틱 기능이 활용되고 있다[2,3]. 또한, 햅틱 렌더링을 위해 비디오 프레임을 40x26 픽셀의 축각 셀로 나누어 각 셀마다 축각 정보를 입력한다[4]. 하나의 축각 셀마다 강도, 감폭, 정지 마찰, 동적 마찰의 네 가지 성질에 대한 수치를 적절히 설정하면 사용자가 해당 축각 셀이 나타내는 물체를 실제로 만지는 듯한 느낌을 받을 수 있다. [5]에서는 Microsoft Kinect를 이용하여 깊이 정보

*정회원, 동국대학교 정보통신공학과

**정회원, 동국대학교 정보통신공학과

***정회원, 동국대학교 정보통신공학과

접수일자: 2018년 4월 2일, 수정완료: 2018년 5월 2일

게재확정일자: 2018년 6월 8일

Received: 2 April, 2018 / Revised: 2 May, 2018

Accepted: 8 June, 2018

*Corresponding Author: skk205@dongguk.edu

Dept. of Information Communication Engineering, Dongguk University, Korea

가 있는 영상을 획득한다. 획득된 깊이 정보를 이용하여 영상 내의 객체와 햅틱 장치를 통해 위치를 조절할 수 있는 HIP(Haptic Interface Point)와의 충돌을 감지하여 사용자로 하여금 객체의 이동을 느낄 수 있도록 한다. 또한 촉각으로 정보를 인식하는 시각장애인들을 위한 여러 햅틱 디스플레이 관련 기술이 연구되고 있다[6,7].

세계적인 햅틱 솔루션 전문 기업인 Immersion에서 실시한 필드테스트 결과에 의하면, 햅틱이 적용된 미디어는 사용자 경험을 훨씬 더 강화하는 경향이 있는 것으로 조사되었다[8]. 조사에 따르면 햅틱이 적용된 영화 예고편을 시청한 사용자들의 반응이 그렇지 않은 경우보다 50%이상 높은 몰입도와 만족도를 보인 것으로 조사되었다. 또한 햅틱이 적용된 예고편을 본 후 실제로 그 영화를 보러 가겠다고 답변한 응답이 그렇지 않은 경우보다 15% 이상 높은 것으로 나타났다. 이러한 리서치 결과를 통해 햅틱 기술은 소리나 영상으로 전달하기 어려운 감성적인 느낌의 전달이 가능하며 특히 비디오와 같은 콘텐츠와 결합할 경우, 훨씬 높은 정도의 몰입감을 줄 수 있음을 확인할 수 있다. 따라서 동영상에 햅틱 기능을 적용하는 특정 프로세스를 갖춘다면 사용자로 하여금 더 높은 사실감과 흥미로운 경험을 이끌어낼 수 있다.

본 논문에서는 동영상 장면들 중에서 햅틱 렌더링을 적용할 수 있는 장면을 정의하고, 그에 적합한 햅틱 렌더링 알고리즘을 제시한다. 그 중 정지 상태의 객체와 단일 객체가 충돌하는 상황에서의 알고리즘을 상세히 기술하며 햅틱 렌더링을 하기 위한 각 단계별 결과를 제시한다.

이하 본 논문의 구성은 다음과 같다. II장에서는 햅틱 렌더링 알고리즘에서 활용되는 방안에 대해 언급하고 III장에서는 햅틱 렌더링을 위한 동영상 장면들과 적합한 프레임들을 판별하기 위한 방안을 기술한다. 그리고 IV장에서는 실험 결과를 제시하며 결론 및 향후 연구내용은 V장에서 언급한다.

II. 관련 연구

본 논문에서 사용하는 햅틱 렌더링에 필요한 객체 추적 알고리즘과 움직임 예측 알고리즘에 대해 설명한다.

일반적으로 동영상에서 객체를 추적하고자 할 때 사용되는 방식으로 Meanshift 알고리즘과 CAMshift 알고리즘을 예로 들 수 있다[9,10]. 두 알고리즘은 데이터 집

합의 밀도분포(특징점, 색상 등)를 기반으로 관심 영역 객체를 추적하는 알고리즘이다. 초기에 탐색 영역의 크기와 위치를 지정하면 반복되는 색 분할 계산에 의해 색상 클러스터가 발생되고, 초기 지정한 색 영역에 기반하여 경계를 결정하여 관심 객체를 추출할 수 있다.

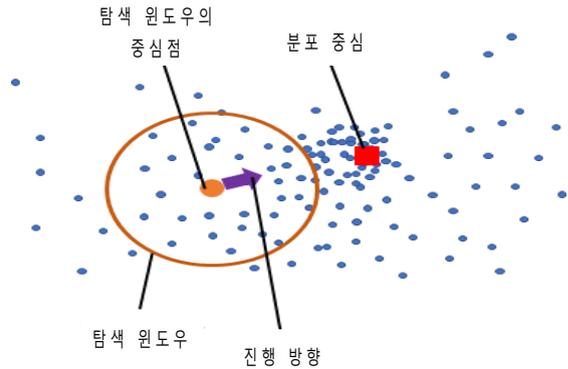


그림 1. Meanshift와 CAMshift의 동작 원리

Fig. 1. Operating principle of Meanshift and CAMshift Algorithms

그림 1과같이 동영상 내에서 탐색 윈도우를 설정한다. 탐색 윈도우의 중심점을 기준으로 탐색 윈도우 내의 데이터를 구한다. 구해진 데이터를 이용하여 데이터의 밀도가 높은 지점으로 중심점을 이동시킨다. 그 후 이동된 중심점을 기준으로 탐색 윈도우를 재설정한다. 이러한 과정을 중심점의 위치 변화가 거의 없을 때까지 반복하며 전체 영역에서 특정한 고밀도 부분을 찾는 알고리즘이 Meanshift 알고리즘이다. 그러나 복잡한 계산량에 의해 실시간 사용에는 무리가 따른다. CAMshift 알고리즘은 Meanshift 알고리즘을 스트리밍 환경에서 사용하기 위해 개선한 것으로 탐색 윈도우의 크기를 스스로 조정하는 기법을 사용하여 탐색 윈도우가 고정되어있는 Meanshift의 단점을 보완한다. 본 논문에서는 탐색 윈도우의 크기를 자동으로 조정하는 CAMshift 알고리즘의 특성을 활용하여 객체의 모양 변화를 탐지한다.

움직임 예측이란 영상 프레임을 작은 블록으로 분할하여 이전이나 미래의 프레임에서의 참조 블록과 가장 비슷하게 정합되는 현재 프레임에서의 블록을 찾는 것을 말한다. 이 때, 참조 블록의 위치와 현재 블록의 위치와의 차이 값을 움직임 벡터라 한다.

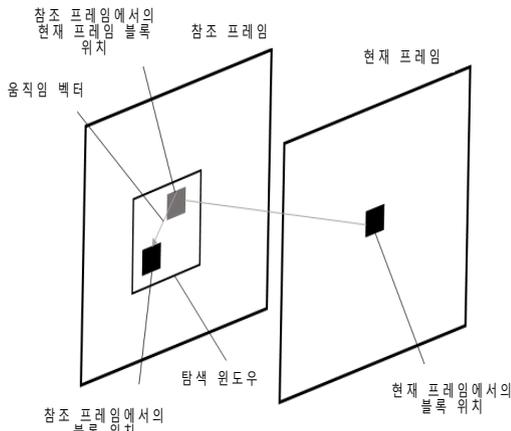


그림 2. 움직임 예측의 원리
 Fig. 2. Principle of Motion Estimation

움직임 벡터를 활용하면 동영상 프레임에서 객체가 어느 방향으로 움직이는지 쉽게 확인할 수 있다. 따라서 본 논문에서는 햅틱 렌더링이 적용될 프레임을 판단하기 위하여 움직임 벡터의 정보를 활용하여 객체의 방향 변화를 탐지한다.

III. 햅틱 렌더링을 위한 동영상 분류와 알고리즘 동작 과정

동영상 장면들 중에서 햅틱 기능을 적용할 수 있는 장면을 정의한다. 정의한 장면들 중 정지 상태의 객체에 단일 객체가 충돌하는 장면에서 햅틱 렌더링을 위한 알고리즘이 어떤 식으로 동작하는지를 살펴보기로 한다.

영화나 드라마, 광고, 학교에서 사용하는 교육용 동영상에서도 햅틱 기능이 활용되면 사용자로 하여금 특정 장면을 더욱 실감나게 감상할 수 있도록 할 수 있다. 따라서 동영상의 어떤 프레임에 햅틱 렌더링을 해야 하는지, 그리고 어떤 방안을 활용하여 햅틱 렌더링을 적용할 프레임을 추출할지에 대한 기준을 마련할 필요가 있으며 해당 장면에 어떤 순서로 햅틱 렌더링을 할지에 대한 방법을 설정할 필요가 있다. 햅틱 렌더링을 적용할 장면은 동영상에서 쉽게 볼 수 있는 장면을 활용하였다.

표 1. 햅틱 렌더링을 위한 동영상 장면

Table 1. Video scene for haptic rendering

	짧은 소리	긴 소리	불규칙
일회성 현상	차 사고, 추락	폭발	객체의 통과
연쇄적 현상	격투	건물 붕괴	큰 물체의 이동
불가측 현상	포탄 발사	지진	천동·번개

표 1과 같이 동영상의 구성을 총 아홉 가지 상황으로 분류하였으며, 가로축은 동영상에서 발생하는 소리의 패턴을 나타내며, 세로축은 화면상 확인할 수 있는 현상을 나타낸다. 본 논문에서 사용하는 햅틱 편집 프로그램은 동영상의 소리를 분석하여 이를 과형으로 나타내기 때문에 표 1의 가로축과 같은 특정 패턴을 쉽게 관찰할 수 있으며 햅틱 기능을 추가하기에 용이하다. 그러나 소리가 언제 발생하는지 정확한 시점을 알기 어려우므로 표 1의 현상들이 발생하는 시점의 프레임을 추출하여 소리만을 이용하는 햅틱 렌더링 방안과 연동한다면 더 명확한 시점에 햅틱 기능이 적용될 수 있다. 따라서 아홉 가지 상황 중, 동일한 방안을 사용하여 햅틱 렌더링을 적용할 수 있는 장면을 재분류하였다.

표 2. 재분류된 동영상 장면과 그에 따른 햅틱 렌더링 방안

Table 2. Reclassified video scene its haptic rendering method

장면	방안
추락, 건물 붕괴	움직임 벡터 활용
차 사고, 격투	객체의 충돌 감지
폭발, 포탄 발사, 천동·번개	색상 변화 감지
객체의 통과	객체의 사라짐 감지

표 2는 각각의 동영상 장면과 해당 장면에서 햅틱 렌더링을 할 수 있는 특정 프레임을 추출하기 위한 방안을 정리한 것이다. 정지 상태의 객체에 단일 객체가 충돌하는 장면에서는 객체의 움직임 벡터의 방향성을 활용하여 햅틱 렌더링에 적합한 프레임을 판별할 수 있다. 또한, 차 사고 장면이나 격투 장면에서는 두 개 이상의 움직이는 객체가 충돌하는 장면이 많으므로, 객체간의 충돌을 감지하여 판별할 수 있다. 폭발이나 천동·번개가 치는 장면에서는 프레임에서 색상이 급격하게 변화하기 때문에 색상의 변화를 감지하는 방안을 활용한다. 기차나 헬기 등이 지나가는 객체가 통과하는 장면에서는 해당하는 객체를 추적하며, 추적 중인 객체가 사라지는 순간을 잡아내

어 이 프레임에 햅틱 렌더링을 하는 방안을 활용한다.

이와 같이 크게 동영상상의 장면을 네 가지 범주로 분류하여 각 상황에 적합한 알고리즘을 적용하여 햅틱 렌더링을 진행한다. 본 논문에서는 네 가지 범주 중, 정지 상태의 객체에 단일 객체가 충돌하는 장면에서 햅틱 렌더링을 하기 위한 프레임을 검출하기 위한 알고리즘을 제안한다. 제안한 알고리즘의 순서도는 그림 3에 나타나있다.

프레임 검출을 위해 움직임 예측, 객체 추적 순으로 처리가 진행되며, 햅틱 렌더링에 적합한 프레임이 검출되고 나면 TSComposer[11]를 이용하여 해당 프레임이 재생되는 시간에 햅틱 효과를 추가한다.

첫 번째의 움직임 예측 단계에서는 각 프레임별로 움직임 벡터의 방향을 확인한다. 즉, 동영상을 프레임 단위로 분할하여 저장하며 분할된 프레임에 움직임 예측을 적용하여 움직임 벡터를 추출하고, 움직임 벡터의 방향에 변화가 있는 프레임을 검출한다. 예를 들어, 움직임 벡터의 방향이 아래에서 위로, 위에서 아래로 바뀐 프레임을 검출하게 된다. 또한 객체가 움직임을 멈춘 프레임도 검출한다. 이 단계에서 움직임 벡터의 방향 변화를 관찰하는 것만으로는 프레임 검출에 문제가 발생할 수 있다. 이는 객체가 어딘가에 부딪혀서 방향의 변화가 발생한 것인지 아니면 단순히 공중에서 방향 변화가 발생한 것인지 판별하기 쉽지 않기 때문이다.

두 번째 객체 추적 단계에서는 움직임 예측 단계에서 프레임 검출을 진행할 때 움직임 벡터 방향이 단순히 변경되었을 때 발생하는 문제점을 해결하기 위해 객체 추적을 진행한다. 객체가 벽이나 바닥에 부딪힌 상황을 판별하기 위해 CAMshift 알고리즘을 활용하여 움직임 벡터의 방향 변화가 있는 프레임에서 객체의 모양 변화가 발생하는지를 판별한다. 만약 객체의 모양 변화가 발생한다면 충돌이 있다는 것으로 판단하고 햅틱 렌더링을 진행하며 그렇지 않은 경우 객체 추적을 계속한다. 객체가 바닥에 닿아 정지된 경우에도 움직임 벡터의 방향 변화가 있는 것으로 판단할 수 있는데, 이 경우 CAMshift 알고리즘으로는 객체의 모양 변화를 감지할 수 없다. 따라서 움직임 예측 단계에서 움직임 벡터의 방향 변화가 검출된 프레임 이후의 세 개 프레임을 더 관찰하여 방향 변화가 없고 정지된 상황이라 판단되면 해당 프레임에 햅틱 렌더링을 진행한다.

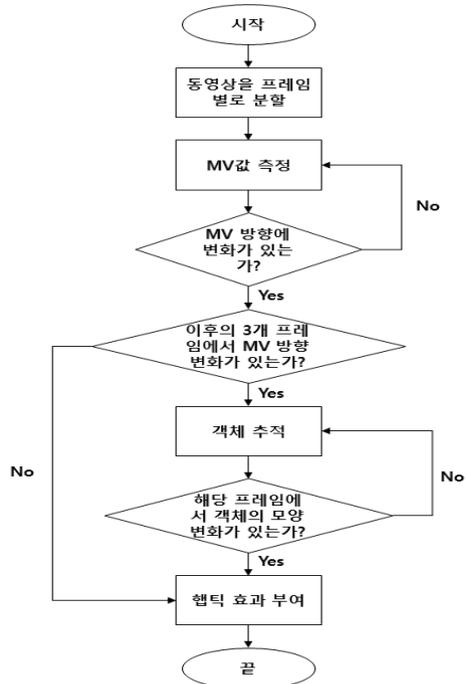


그림 3. 햅틱 렌더링 알고리즘의 순서도
Fig. 3. Flowchart of haptic rendering algorithm

마지막으로 움직임 예측 단계와 객체 추적 단계를 통해 검출된 프레임에 햅틱 렌더링을 하기 위해 동영상의 초당 프레임 재생 수를 이용하여 검출된 프레임이 재생되는 시간을 계산한다. 계산된 시간에 햅틱 편집 프로그램인 TSComposer를 이용하여 햅틱 기능을 추가한다.

IV. 실험 및 결과

움직임 예측과 객체 추적을 통해 검출된 프레임과 해당 프레임이 재생되는 시간에 햅틱 효과가 추가된 결과를 볼 수 있다. 정지 상태의 객체에 단일 객체가 충돌하는 상황과 부합하는 공이 낙하하는 동영상을 이용하였다.

그림 4는 공이 낙하하다가 바닥에 닿고 다시 튕겨 올라가는 상황을 나타낸 것이다. 그림 4의 (a)는 공이 바닥에 닿은 상황이지만, 움직임 벡터의 방향이 아래방향으로 유지되는 것을 확인할 수 있다. 그림 4의 (b)부터 움직임 벡터의 방향에 변화가 발생한 것을 확인할 수 있는데, 이와 같은 프레임을 햅틱 렌더링을 하기에 적합한 프레임으로 판단한다. 이와 같이 샘플 동영상에서 움직임 예측을 통해 객체의 방향 변화가 발생한 프레임을 총 12개

검출할 수 있다. 그림 4의 (d)는 공이 정지한 상황을 나타낸다. 이 프레임 또한 움직임 벡터의 방향에 변화가 발생하였기 때문에 적합한 프레임이라 판별할 수 있으므로 그 이후의 추가적인 세 개의 프레임을 관찰하여 객체가 정지하였음을 확인하였다.

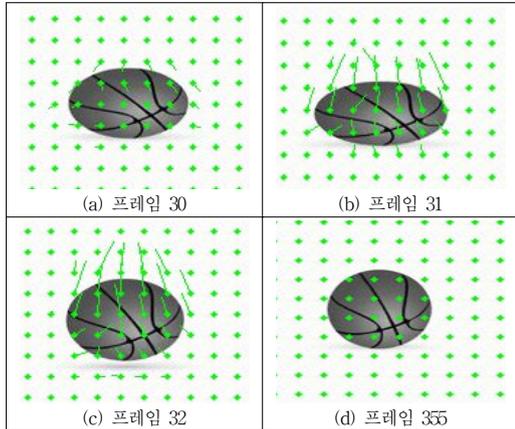


그림 4. 움직임 벡터의 방향 변화가 있는 프레임의 예
 Fig. 4. Example frames which have direction change of motion vector

그림 5는 객체 추적을 실행한 결과를 나타낸다. (a)는 공이 최고점에 위치할 때의 상황을 나타낸 것이며, (b)는 공이 바닥에 튕겨져 올라가는 상황이다. 두 결과를 통해 CAMshift 알고리즘이 객체가 바닥에 튕길 때 객체의 모양 변화에 따라 탐색 윈도우의 크기를 변화시키는 것을 확인하였다. 따라서 움직임 벡터의 변화가 있는 시점의 프레임에서 객체의 모양 변화가 있다는 점을 판단하였다.

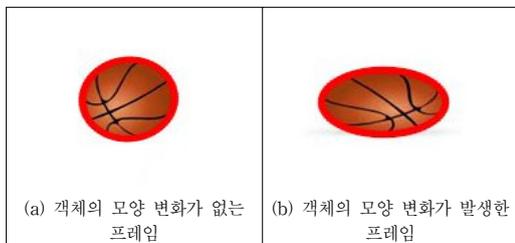


그림 5. 객체 추적을 통한 탐색 윈도우의 변화
 Fig. 5. Change of search window through object tracking

움직임 예측과 객체 추적을 통해 검출된 프레임에 햅틱 효과를 추가하기 위해 TSComposer를 사용하였다. TSComposer는 동영상의 사운드를 분석하여 파형으로

나타내며, 특정 파형이 발생하는 지점에 사용자가 햅틱 파형을 추가하는 프로그램이다. 햅틱 파형을 추가하기 위해 검출된 프레임이 재생되는 시점을 초당 프레임 재생수와 프레임 넘버를 이용하여 계산하였다. 계산 결과는 표 3과 같다.

표 3. 햅틱 렌더링 알고리즘을 통해 검출된 프레임 재생 시간
 Table 3. Playback Time of detected frames through the haptic rendering algorithm

프레임 넘버	재생 시간
31	1.03
77	2.56
123	4.10
169	5.63
209	6.96
241	8.03
270	9.00
295	9.83
313	10.43
327	10.9
339	11.3
348	11.6
355	11.83

계산된 재생 시간에 그림 6과 같이 TSComposer를 이용하여 햅틱 효과를 추가하였다. 추가된 햅틱 효과는 샘플 동영상의 상황에 가장 적합한 임펄스 형태의 파형을 사용하였다.

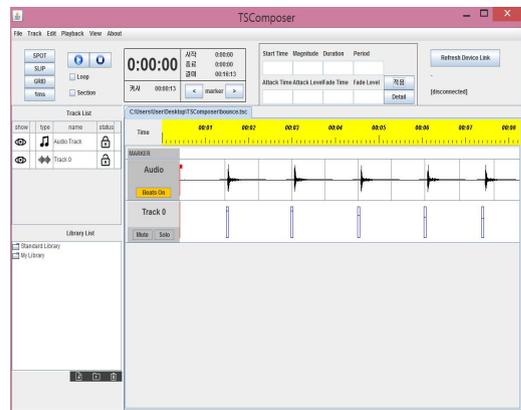


그림 6. TSComposer를 활용하여 햅틱 효과를 추가한 결과
 Fig. 6. Result of adding haptic effect by using TSComposer

TSComposer를 이용하여 햅틱 렌더링 알고리즘을 통해 검출된 프레임에 햅틱 파형을 추가한 후 TSPlayer[11]라는 햅틱 동영상 재생 어플리케이션을 이용하여 동영상을

재생한 결과 공이 바닥에 닿는 순간 진동 효과를 확인할 수 있었다.

V. 결론

본 논문에서 동영상 장면들 중에서 햅틱 렌더링을 할 수 있는 장면을 정의하고, 그에 대한 햅틱 렌더링 알고리즘을 제시하였다. 네 가지로 분류한 동영상의 종류 중, 정지 상태의 객체에 단일 객체가 충돌하는 상황에서의 알고리즘을 상세히 기술하였고 햅틱 렌더링을 위한 각 단계별 결과를 제시하였다. 움직임 예측을 통해 움직임 벡터를 추출하며 벡터의 방향성을 이용하여 움직임 벡터의 방향 변화를 판단한다. 또한 객체 추적 방식을 연동하여 움직임 벡터의 방향 변화가 있는 프레임에서 객체의 모양 변화를 판별하여 적합한 프레임을 검출한 후 검출된 프레임의 재생 시간을 계산하여 햅틱 렌더링을 실시한다. 제안된 방법을 사용할 경우, 동영상이 재생되는 동안 비교적 정확한 시점에 햅틱 효과가 발생하는 것을 확인할 수 있었다.

향후 연구를 진행하며 좀 더 우수한 움직임 예측 방안을 연구할 것이며, 햅틱 렌더링 알고리즘의 각 단계를 자동화하여 유기적으로 햅틱 렌더링이 진행될 수 있도록 햅틱 렌더링 알고리즘을 개선할 예정이다.

References

- [1] Yi Li, Youngeun Lee, Young J. Kim. "Six-degree-of-freedom Haptic Rendering using Translational and Generalized Penetration Depth Computation." The Journal of Korea Robotics Society, 8.3 (2013.9): 173-178.
DOI: <https://doi.org/10.1109/whc.2013.6548423>
- [2] Young-Dae Lee, Jeong-Jin Kang, Chan-Woo Moon. "Development of the Fishbot using Haptic Technology." The Journal of the Institute of Internet, Broadcasting and Communication, 10.4 (2010):77-82.
- [3] Hyun-Seob Cho. "Reproduced for the Haptic Interface Electrical Shock Feeling Technology Development." Conference of the Korea Academia-Industrial cooperation Society(JKAIS), (2014.05): 178-180.
- [4] Mee Young Sung, Kyung Koo Jun, Dongju Ji, Hwanmun Lee, Kikwon Kim, "Touchable Video and Tactile Audio.", 2009 11th IEEE International Symposium on Multimedia, pp.425-431
DOI: <https://doi.org/10.1109/ism.2009.79>
- [5] Shahzad Rasool, Alexei Sourin, "Haptic Interaction with Video Streams Containing Depth Data.", 2015 International Conference on Cyberworlds(CW), pp.173-180
DOI: <https://doi.org/10.1109/cw.2015.43>
- [6] Jungil Jung, Jinsoo Cho. "Design and Implementation of Real-time Haptic Display System." The Institute of Electronics Engineers of Korea - Computer and Information, 48.2 (2011.3): 1-8.
- [7] Sang-Youn Kim, Seong-Man Cho. "A Haptic Navigation System for Visually Impaired Persons." Journal of Korea Multimedia Society, 14.1 (2011.01): 133-143.
DOI: <https://doi.org/10.9717/kmms.2011.14.1.133>
- [8] "콘텐츠 몰입도를 높여주는 햅틱 기술의 진화", 한국광고총연합회 광고정보센터 매거진, 2014년 10월 10일 수정, 2017년 5월 24일 접속,
<https://www.adic.or.kr/journal/column/show.do?ukey=348406>
- [9] Y.Cheng, "Mean shift, mode seeking, and clustering," IEEE Trans. Pattern Anal. Machine Intell., 17:790-799, 1995.
DOI: <https://doi.org/10.1109/34.400568>
- [10] Bradski G.: "Computer Vision Face Tracking for Use in a Perceptual User Interface", Intel Technology Journal, 1998, 2(Q2), pp. 1-15.
- [11] Tactual Sense로부터 제공받은 햅틱 편집 프로그램과 햅틱 동영상 재생 어플리케이션

저자 소개

김 선 규(정회원)



- 2017년 동국대학교 정보통신공학과 공학사
- 2017년 ~ 현재 동국대학교 정보통신공학과 석사과정
<주관심분야: 영상 처리, 햅틱 렌더링>

김 혜 빈(정회원)



- 2018년 동국대학교 정보통신공학과 공학사
<주관심분야: 영상 처리, 동영상 압축>

류 철(정회원)



- 1989년 Florida Institute of Technology 컴퓨터공학과 공학사
- 1991년 Polytechnic Institute of NYU 전기공학과 공학석사
- 1997년 Polytechnic Institute of NYU 전기공학과 공학박사
- 1998년 ~ 1999년 LG 정보통신연구소 선임연구원
- 1999년 ~ 현재 동국대학교 정보통신공학과 교수
<주관심분야 : Visual Communication, High Resolution Reconstruction, Joint Source Coding>