

Interactive and Automatic Panning for 360° Video Playback

강경국, 조성현

대구경북과학기술원

kkk831@dgist.ac.kr, scho@dgist.ac.kr

요 약

모니터와 같은 2D 디스플레이로 360° 비디오를 감상하는 주된 방법은 전체 영상의 일부분을 잘라내어 일반적인 시야각의 영상으로 변환한 뒤에 감상하는 것이다. 사용자가 원하는 부분을 보기 위해서는 마우스 드래깅과 같은 간단한 조작을 통해 볼 수 있지만 매순간 원하는 위치를 지정해줘야 하는 단점이 있다. 이를 해결하기 위해 최근 몇 년 사이 자동으로 360 비디오에서 볼만한 위치를 찾아 이를 통해 일반 시야각의 비디오를 생성해 주는 방법들이 제시되었다. 그러나 기존 방법은 비효율적인 연산으로 인해 지나치게 높은 수행시간을 요구하며, 빠르게 움직이는 비디오는 제대로 처리하지 못하는 문제점이 있다. 특히 기존 방법의 경우 알고리즘에 의해 자동으로 찾아진 영역 외에는 사용자가 원래 360 비디오의 내용을 전혀 확인할 수 없는 문제가 있었다. 본 논문에서는 이러한 문제점을 해결하기 위한 방법을 제시한다. 본 논문에서 제시한 방법은 적은 계산량으로도 효율적으로 볼만한 영역을 찾을 수 있으며, 빠르게 움직이는 비디오에 대해 대응 가능하다. 또한 사용자의 자율성을 최대한 보장해 비디오 재생 중 사용자가 원하는 방향을 지정하면 그에 따라 비디오 재생 영역이 자동으로 업데이트된다.



(a) 360 카메라



(b) 구형 파노라마



(c) NFOV 영상

그림 1 (a) 360 카메라 (b) 구형 파노라마(spherical panorama)로 투영된 360 동영상. (c) 360 동영상 중 일부 방향만을 이용하여 생성된 normal field-of-view (NFOV) 영상

1. 서론

지난 몇 년간 가상 현실의 상업적인 발전과 더불어 360° 카메라가 많은 관심을 모았다. 360° 카메라는 모든 방향을 동시에 촬영할 수 있기에 쉽게 가상 현실 콘텐츠를 제작할 수 있어 각광을 받게 되었으며, 삼성 Gear 360, LG 360 CAM, GoPro Fusion 과 같은 다양한 카메라가 본격적으로 시장에 등장하였다.(그림 1(a)) 뿐만 아니라 Facebook 이나 YouTube 를 통해 일반 사용자들도 360° 비디오를 쉽게 접할 수 있게 되었다.

모든 방향이 동시에 녹화되어 있는 360° 비디오의 특성 상 비디오를 시청하기 위한 최적의 방법은 가상 현실 장비인 Head-Mount Display (HMD)를 착용하고 실제로 머리를 돌려가며 보는 것이지만, 실제로 많은 경우에는 HMD 를 번거롭게 착용하는 대신 일반 2D 디스플레이로 보는 경우가 많다. 모든

방향이 동시에 존재하는 360° 비디오를 2D 디스플레이로 보기 위해서는 우선 360° 비디오의 내용을 2D 디스플레이에 투영(projection)해야 한다. 기존의 대표적 방식 중 하나는 그림 1(b)와 같이 360 영상을 구형 파노라마 (spherical panorama)로 변환해서 보여주는 방식이다. 이 방식은 비디오의 모든 방향을 동시에 볼 수 있다는 장점이 있으나 영상 전반에 걸쳐 왜곡이 심하게 발생하는 문제점이 있다.

이보다 일반적인 방식은 360 영상의 특정 방향을 선택하고 그 영역만을 2D 영상으로 만들어 사용자에게 보여주는 방식이다 (그림 1(c)). 이 방식의 경우 기존의 일반적 카메라로 촬영한 것과 같이 자연스러운 시야각(NFOV, Normal Field of View)을 갖는 영상을 사용자에게 보여 줄 수 있어 YouTube 와 같은 서비스에서 흔히 적용되고 있다. 그러나 이 방식으로 비디오를 감상할 때 특정 방향의 장면만 제공되기 때문에 비디오를 처음 보는 시청자는 비디오

의 어느 부분을 봐야 할 지 고민해야 하고 비디오를 감상하는 때 순간 마우스를 이용해서 볼 방향을 번거롭게 지정해줘야 하는 문제가 있다.

이러한 문제를 해결하고 사용자가 편안하게 360 비디오를 감상할 수 있게 하기 위해 최근 몇몇 연구가 소개 되었다[1,2,6]. 이들은 360 비디오의 내용을 분석해 불만한 지점들을 자동으로 찾고 이를 이용해 360 비디오로부터 불만한 영역들로 구성된 2D NFOV 비디오를 추출한다. 결과적으로 사용자는 불만한 영역을 직접 찾을 필요가 없이 자동으로 선택된 영역을 볼 수 있다. 하지만 알고리즘에 의해 선택되지 않은 영역은 결과 비디오에 포함되지 않아 사용자가 그 내용을 확인할 수 없는 문제가 있다.

[3]은 하나의 2D NFOV 비디오를 생성하는 대신 360 비디오 재생 시 불만한 영역을 알려주는 시각적 지표를 제공하는 방식을 제안하였다. 사용자는 시각적 지표를 활용해 직접 불만한 지점들을 선택해서 볼 수 있다. 그러나 스포츠 비디오처럼 불만한 영역의 위치가 빠르게 바뀌는 비디오에서는 사용자가 시각적 지표를 따라 방향을 빠르게 바꿔줘야 하는 단점이 있다.

본 논문에서는 두 방식의 장점을 모두 취하여 자동으로 불만한 장면을 선택해서 사용자에게 보여주는 동시에 사용자가 원하는 방향이 있다면 비디오 재생 시에 직접 개입하여 원하는 곳을 볼 수 있는 방법을 제안한다. 본 논문에서 제안하는 방식은 크게 비디오에서 불만한 영역을 찾는 전처리 과정과 미리 파악된 불만한 영역과 사용자의 인터랙션을 기반으로 자동으로 방향을 전환하며 사용자에게 비디오를 재생해주는 비디오 재생 과정으로 나뉜다.

360 비디오에서 불만한 영역을 찾는 전처리 과정을 위해서 본 연구에서는 [1,2]에서 제시한 capture-worthiness를 채택하였다. [1,2]의 경우 capture-worthiness 계산을 위해서 각각의 방향 별로 비디오 영역을 샘플링한 후, 각각의 방향마다 capture-worthiness를 따로 계산한다. 따라서 매우 짧은 비디오에 대해서도 엄청난 계산 시간이 소모되는 문제점이 있다. 본 논문에서는 이를 해결하기 위해 360 비디오에서 방향 별로 샘플링하는 과정 없이 모든 방향에 대해 동시에 capture-worthiness를 계산하는 방법을 제안한다.

또한 기존 방식[1,2]의 경우, 동영상에서 불만한 영역을 판단하기 위해 동영상을 5 초 간격으로 나눠 샘플링을 하고, 각각의 5 초 동영상에 대해 불만한 영역인지를 판단했다. 따라서 역동적으로 빠르게 움직이는 동영상에 대해서는 한계가 있었다. 본 논문에서는 보다 역동적인 동영상을 처리하기 위해 더 짧은 간격으로 샘플링하면서도 안정적으로 불만한 영역을 찾을 수 있는 방법을 제시한다.

마지막으로 불만한 영역을 재생하면서도 사용자가 인터랙티브하게 원하는 방향을 볼 수 있도록 하기 위해 본 논문에서는 재생 경로를 인터랙티브하게 업데이트하여 비디오를 재생할 수 있는 방법을 제시한다.

본 논문의 구성은 다음과 같다. 2 장은 제안하는 방법의 전처리 과정을 설명한다. 3 장은 제안하는 360 비디오 재생 방법을 설명한다. 4 장은 제안하는 방법의 실험 결과를 제시하고, 5 장은 결론을 맺는다.

2. 전처리

본 논문에서 제시하는 방법의 전처리 과정은 입력된 360 비디오로부터 각각의 시간대 별로 불만한 영역을 찾는다. 본 논문에서는 이를 위해 [1,2]에서 제시한 C3D 네트워크[4]를 기반으로 capture-worthiness를 채택하였다. [1,2]에서는 capture-worthiness를 계산하기 위해 다음과 같은 과정을 거친다. 우선 입력된 360 비디오를 5 초 간격으로 자른다. 각각의 5 초 길이의 360 비디오를 다시 방향 별로 샘플링하여 각 방향별 5 초 길이 NFOV 비디오를 얻는다. 이 각각의 NFOV 비디오를 시공간에서의 좁은 영역을 뜻하는 ST glimpse라 호칭한다. [1,2]에서는 각 방향 별 샘플링을 위해 360 비디오를 구형 좌표계로 변환하고 18 개의 경도와 11 개의 위도의 교점을 중심으로 샘플링하여 총 198 개의 ST glimpse를 만든다. 각 ST glimpse는 비디오 분류를 위해 제시된 C3D 네트워크[4]에 통과시킨다. C3D 네트워크의 결과로 나온 activation 값들을 feature로 미리 학습된 fully connected network 형태의 logistic regression classifier에 넣어 입력된 ST glimpse가 불만한 영역인지에 대한 점수를 0 부터 1 사이의 값으로 얻는다. 이에 대한 보다 자세한 내용은 [1,2]를 참고한다.

[1,2]의 경우 각 방향의 capture-worthiness를 계산하기 위해 그림 2(a)와 같이 각 방향의 ST glimpse를 샘플링하고 이를 각각 C3D 네트워크에 통과시키는 과정을 거쳤다. 이로 인해 전체 비디오에서 capture-worthiness map을 계산하기 위해서는 엄청난 계산량이 소모되는 문제점이 있다. 또한 입력 비디오를 5 초 간격으로 자르기 때문에 동적인 비디오에 대해서는 빠르게 대응하지 못하는 문제가 있다. 예를 들어 2.5 초마다 가상 카메라가 돌아야 할 지점이 바뀌어야 한다면 기존 방법으로는 좋은 결과를 얻기 힘들다.

본 논문에서는 이를 해결하기 위해 다음과 같은 방법을 제시한다. 우선 입력 비디오를 5 초 간격이 아닌 C3D 네트워크의 최소 입력 크기인 16 프레임 간격으로 자른다. 이는 약 0.5 초에 해당하는 시간으로 기존 방법 대비 훨씬 빠르게 동적인 내용을 다룰 수 있다. Capture-worthiness를 파악하기 위한 비디오가 기존 방법 대비 훨씬 짧기 때문에 기존 방법에 비해 capture-worthiness 값이 불안정하게 추정될 수 있으나, 이는 뒤의 비디오 재생 과정에서 재생의 시간적 평탄성을 고려하는 것으로 처리된다.

16 프레임 길이의 360 비디오에 대해 각 방향에 대한 샘플링없이 구형 파노라마 형태의 360 비디오를 C3D 네트워크에 통과시킨다. C3D 네트워크의 결과로 구형 파노라마 전체 영역에 대한 activation

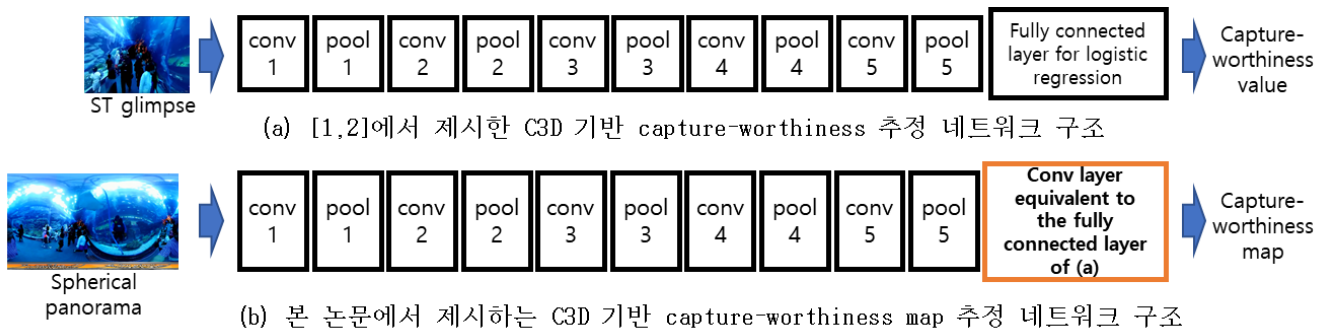


그림 2 기존 방법[1,2]에서 제시한 capture-worthiness 추정 네트워크 구조와 본 논문에서 제시한 네트워크 구조의 비교

map 이 나온다. 이 activation map 에서 각각의 ST glimpse 에 해당하는 영역의 activation 들을 crop 하여 logistic regression classifier 에 넣으면 각 ST glimpse 에 해당하는 capture-worthiness 를 계산할 수 있다. 이 logistic regression classifier 는 fully-connected network 의 형태이기 때문에 convolutional network 으로 변형이 가능하다. 따라서 제시하는 방법에서는 logistic regression classifier 의 가중치를 그대로 사용하는 convolutional layer 를 C3D 네트워크의 마지막에 붙여 16 프레임 길이의 구형 파노라마 비디오를 네트워크의 입력으로 주면 각각의 영역의 capture-worthiness 가 한꺼번에 모두 계산되는 네트워크를 만들었다. (그림 2(b))

제시하는 방법은 각 방향별 샘플링 과정이 전혀 없기 때문에 ST glimpse 샘플링에 필요한 계산 시간이 전혀 들지 않는다. 또한 convolutional layer 를 활용하여 공간적으로 인접한 ST glimpse 사이의 오버랩되는 부분에서의 중복된 계산도 없기 때문에 이에 전체적으로 capture-worthiness 를 계산하는데 필요한 계산량을 크게 낮출 수 있다. 구체적인 수행 시간은 4.3 장에서 제시한다.

구형 파노라마의 경우 ST glimpse 와는 달리 왜곡이 심하게 발생하는 영역이 생기게 된다. 왜곡은 특히 영상의 상단과 하단에서 많이 발생하게 되는데, 이러한 왜곡으로 인해 상단과 하단에서는 capture-worthiness 가 부정확하게 추정될 수 있다. 그러나 일반적인 경우 비디오의 상단과 하단은 하늘과 땅바닥에 해당하는 영역으로 capture-worthiness 가 떨어진다고 가정할 수 있다. 따라서 본 논문에서는 구형 파노라마에서 상단과 하단을 잘라낸 영역에서만 capture-worthiness 를 계산한다. 결과적으로 360 비디오의 매 16 프레임마다 18×5 크기의 capture-worthiness map 을 구하게 된다.

3. 인터랙티브 360 비디오 재생

3.1 360 비디오 자동 방향 전환

본 논문에서는 추정된 capture-worthiness 를 바탕으로 단순히 2D NFoV 비디오 하나를 추출하는

것이 아닌 사용자의 인터랙션이 가능한 360 비디오 재생 방법을 제시한다. 우선 본 절에서는 사용자 인터랙션을 소개하기에 앞서 capture-worthiness 를 바탕으로 360 비디오에서 자동으로 방향을 전환해주는 방법을 설명한다.

360 비디오에서 매 시간마다 capture-worthiness 가 높은 장면을 보여주면서도 방향 전환이 급격하게 이루어지는 것을 막기 위해 본 논문에서는 다음과 같은 에너지 함수를 최적화하는 경로 $P = \{p_1, p_2, \dots, p_T\}$ 를 구한다.

$$P = \arg \min_P \sum_t \{1 - c_t(p_t)\}^2 + \alpha \|p_t - p_{t+1}\|^2 \quad (1)$$

이 식에서 T 는 capture worthiness map 의 수로, 비디오의 전체 프레임수의 $\frac{1}{16}$ 이다. p_t 는 t 번째 시간에서의 방향을 나타낸다. c_t 는 t 번째 시간에서의 capture-worthiness map 이므로, $c_t(p_t)$ 는 p_t 방향에 해당하는 capture-worthiness 값이 된다. 오른쪽의 첫번째 항은 각 시간에서의 방향이 capture worthiness 가 높은 쪽으로 가게 하는 항이고 두번째 항은 인접한 시간에서의 방향이 비슷하게 만드는 항으로, 방향이 너무 급격하게 바뀌는 것을 막아주고 방향 전환이 부드럽게 일어나게 해 준다. α 는 두번째 항의 강도를 조절하는 파라미터이다.

본 논문에서 제시하는 방법은 이를 최적화하는 경로 P 를 구하기 위해 동적 프로그래밍 기법을 적용하여 최적해를 구한다. 구해진 최적의 경로를 따라서 비디오 재생 시 각 시간마다 최적의 방향으로 선택된 비디오 영역을 보여주는 식으로 자동 방향 전환을 실행한다.

[1,2]의 경우 역시 마찬가지로 capture-worthiness 를 최적화하는 경로를 동적 프로그래밍을 통해서 구해 NFoV 비디오를 생성한다. 그러나 [1,2]의 경우는 급격한 전환을 막기 위해 인접한 시간 사이의 방향 전환의 양을 바로 인접한 ST glimpse 로만 갈 수 있도록 hard constraint 로 제약하였다. 또한 5 초 간격으로 샘플링하기 때문에 매우 느린 전환만 가능하다. 본 논문에서 제시하는 방법은 hard constraint 대신 soft constraint 인 수식 (1)의 두번째 항을 이용하여 좀 더 자유로운 방향 전환이 가능하도록 하였으며, 5 초 간격 대신 16 프레임 간격으로 샘플링하기 때문에 동적인 비디오에 대해서도 대응

이 가능하다.

3.2 사용자 인터랙션을 통한 방향 전환

360 비디오를 감상할 때의 장점 중 하나는 마우스를 이용하여 쉽게 사용자가 원하는 방향을 볼 수 있다는 것이다. 즉, 기존의 수동적인 비디오 시청과는 달리 사용자가 원하는 영역을 능동적으로 찾아볼 수 있다는 점이다. 기존의 360 비디오로부터 NFoV 비디오를 추출해 주는 방법[1,2]의 경우 사용자가 봐야할 지점을 자동으로 찾아주는 장점이 있지만 알고리즘이 선택하지 않은 영역은 볼 수 없다는 단점이 있다.

본 논문에서는 사용자의 자율성을 보장하기 위해서 비디오를 재생하는 도중 사용자가 원하는 지점을 제시해 주면 그 지점을 기준으로 방향 경로를 업데이트하는 방식을 제안한다. 이를 위해 본 연구에서는 그림 6 와 같은 그래픽 유저 인터페이스(GUI)를 만들었다. 사용자는 GUI 를 통해 360 비디오를 감상하다가 원하는 방향이 있다면 마우스를 드래깅하여 원하는 방향으로 비디오의 방향을 전환한다. 또는 GUI 하단에 함께 보여지는 구형 파노라마 영상의 원하는 지점에 마우스를 클릭하면 그 지점으로 GUI 상단의 NFoV 비디오의 방향도 업데이트된다.

사용자가 지정한 방향을 따라 새로운 경로를 구하기 위해서 본 연구에서는 다음과 같은 사항을 고려하였다.

- 사용자가 선택한 지점이 capture worthiness 가 높은 영역인 경우 capture worthiness 가 높은 영역을 계속 따라가며 보여준다.
- 사용자가 선택한 지점이 capture worthiness 가 높지 않은 영역인 경우에도 일정 시간 이상 이 지점을 보여준다. 일정 시간이 지난 후에는 capture worthiness 가 높으면서 가까운 영역으로 경로를 이동한다.

이를 위해서 사용자가 지정한 시점에서부터 다음과 같은 에너지 함수를 최적화하는 경로를 구한다.

$$P = \arg \min_p \sum_{t=t_0}^T \{ (1 - w_t) |1 - c_t(p_t)|^2 + \alpha \|p_t - p_{t+1}\|^2 + w_t \|p_t - p_u\|^2 \} \quad (2)$$

이 식에서 t_0 는 사용자가 방향을 전환한 시점이다. p_u 는 사용자가 지정한 방향이다. w_t 는 시간 축에서의 가중치로 다음과 같이 정의된다.

$$w_t = \exp\left(-\frac{|t - t_0|^2}{2\sigma_{t_0}^2}\right) \quad (3)$$

$\sigma_{t_0}^2$ 는 w_t 가 t_0 에서 떨어질수록 1에서 0으로 떨어지는 속도를 결정하는 파라미터로 t_0 에서 사용자가 지정한 지점의 capture worthiness에 의해 결정된다.

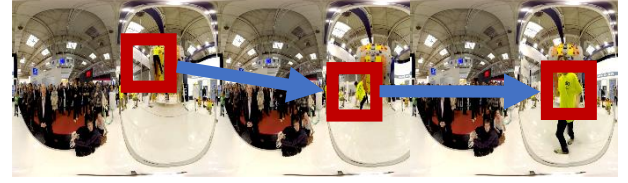


그림 3 시간의 흐름에 따른 원본 360° 비디오와 시스템의 결과

$$\sigma_{t_0}^2 = \max \left\{ M^2 (1 - c_{t_0}(p_u))^2, \tau \right\} \quad (3)$$

M 은 사용자 파라미터로 본 논문이 제시하는 시스템에서는 10을 사용했다. τ 는 식 (3)에서 0으로 나누는 것을 방지하기 위한 작은 상수이다.

식 (2)에서의 첫번째 항과 두번째 항은 식 (1)의 항과 유사하다. 식 (2)의 세번째 항은 사용자가 원하는 지점을 일정 시간 이상 보여주기 위한 항으로 사용자 인터랙션을 위해 추가되었다. w_t 는 사용자가 선택한 지점의 capture worthiness에 따라 변경되어 최적의 방향 전환 경로가 앞에서 고려한 사항을 만족하도록 한다.

4. 실험 결과 및 분석

4.1 capture-worthiness 데이터 셋

C3D feature를 통해 360 비디오의 각 영역별 capture worthiness를 구하기 위한 logistic regression classifier를 학습하기 위한 데이터 셋은 [1,2]에서 제공하는 데이터 셋을 사용하였다. 데이터 셋은 YouTube에서 다운받은 360° 비디오와 360 카메라가 아닌 일반 카메라로 촬영된 2D NFoV 비디오 (HumanCam)로 구성되어 있다. 360° 비디오는 총 길이 7.3 시간의 86개의 비디오로 구성되어 있고 HumanCam은 총 길이 343 시간의 NFoV 비디오로 구성되어 있다.



그림 4 사용자가 원하는 방향으로의 개입과 그에 따른 새로운 경로의 생성 결과

[1,2]에서와 마찬가지로 classifier 를 학습하기 위해 360 비디오의 임의의 방향을 무작위로 잘라내서 생성한 2D NFoV 비디오 데이터셋을 만들었다. 360 비디오의 임의의 방향을 잘랐기 때문에 대부분의 경우는 의미없는 내용이 담겨 있다. 따라서 이런 비디오는 capture worthiness 가 0 이라고 가정할 수 있다. 반대로 사람이 직접 촬영한 2D NFoV 비디오의 경우 촬영자가 찍고자 하는 대상을 촬영한 경우가 대부분이기 때문에 capture worthiness 가 1 이라고 가정할 수 있다. 따라서 360 비디오에서 무작위로 잘라낸 NFoV 비디오와 사람이 직접 촬영한 NFoV 비디오 사이를 구분하기 위한 logistic regression classifier 를 학습하고 이를 이용하여 capture worthiness 계산에 활용하였다.

4.2 자동 방향 전환 결과

그림 3 은 원본 360° 비디오와 시스템이 만들어 낸 결과를 시간순으로 배열하였다. 붉은 색 사각형을 통해서 가상의 카메라가 가리키는 위치를 표시하였다. 더 다양한 비디오에 대한 결과는 그림 6 에 제시되어 있다.

그림 3의 첫번째 결과에서는 가장 두드러지는 사람의 행동을 따라 가상의 카메라가 이동하게 되고 두번째 결과에서는 기장과 배의 움직임을 따라 이동하는 것을 확인할 수 있다. 각각의 결과에서 시스템은 사람이 찍은 비디오와 가장 유사한 비디오를 출력하게 된다.

4.3 계산 시간

표 1 실행 시간 비교

방법	AutoCam[1,2]	실험 결과
60 초 비디오 처리시간	200 분 이상	10 분 이내

표 1 은 전처리 과정에서 소모되는 실행 시간을 [1,2]에서 제시한 방법과 비교한 결과이다. [1,2]에서 제시한 방법이 5 초 단위로 샘플링하는 것에 비해 본 논문에서 제시한 방법은 16프레임 단위로 샘플링함에도 불구하고 월등히 빠른 속도로 capture-worthiness 를 구할 수 있음을 알 수 있다.

4.4 사용자 인터랙션

그림 4 는 그림 3 의 첫번째 결과에서 왼쪽 방향으로 마우스 드래깅을 통해 사용자가 원하는 화면을 지정해준 결과이다. 사용자가 화면을 지정해줬을 때 그 화면에서 시작하는 경로를 새로 찾아 줌으로써 사용자가 자율적으로 경로를 선택할 수 있다.



그림 5 빠른 움직임을 담고 있는 360 비디오에 대한 시스템의 결과

4.5 역동적 비디오 생성

그림 5 는 시스템이 원본 영상의 볼 지점이 빠르게 바뀌는 경우의 결과를 시간 순으로 배열하였다. 2 장에서 제시한 것처럼 16 프레임 간격으로 capture-worthiness 를 구하기 때문에 공이 빠르게 움직이는 영상에서도 공의 움직임을 올바르게 쫓는 결과를 출력하게 된다.

5. 결론

본 논문에서는 360 비디오를 2D 디스플레이 상에서 시청할 때 360 비디오로부터 불만족 영역을 자동으로 찾아 방향 전환을 해 주는 방법을 제시하였다. 본 논문에서 제시하는 방법은 기존 방법[1,2] 대비 월등히 빠른 속도로 360 비디오 상의 불만족 영역을 찾을 수 있으며, 동적인 비디오에 대해서도 대응 가능하다. 또한 사용자의 자율성을 최대한 보장하기 위한 사용자 인터랙션이 가능한 그래픽 유저 인터페이스 및 사용자의 선택을 반영한 경로 업데이트 방법 역시 제시하였다.

감사의 글

이 논문은 2017년도 정부(과학기술정보통신부)의 재원으로 한국연구재단-차세대정보·컴퓨팅기술개발사업의 지원을 받아 수행된 연구임(NRF-2017M3C4A7066316).



그림 6 시간의 흐름에 따른 원본 360° 비디오와 시스템의 결과. 원본 영상에서의 노란색 박스는 결과 동영상 이 가리키는 위치를 표시

참고문헌

Learning a Deep Agent for Piloting through 360° Sports Videos. In CVPR 2017

- [1] Yu-Chuan Su, Dinesh Jayaraman, & Kristen Grauman. Pano2Vid: Automatic cinematography for watching 360° videos. In ACCV 2016
- [2] Yu-chuan Su, Kristen Grauman. Making 360° Video Watchable in 2D: Learning Videography for Click Free Viewing In CVPR 2017
- [3] Yen-Chen Lin, Yung-Ju Chang, Hou-Ning Hu, Hsien-Tzu Cheng, Chi-Wen Huang, Min Sun. Tell Me Where to Look: Investigating Ways for Assisting Focus in 360° Video In CHI 2017
- [4] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, Manohar Paluri. Learning Spatiotemporal Features with 3D Convolutional Networks. In ICCV 2015
- [5] Xiao, J., Ehinger, K.A., Oliva, A., Torralba, A.: Recognizing scene viewpoint using panoramic place representation. In: CVPR 2012
- [6] Hou-Ning Hu, Yen-Chen Lin, Ming-Yu Lui, Hsien-Tzu Cheng, Yung-Ju Chang, Min Sun. Deep 360 Pilot: