360° 비디오 재생을 위한 자동 패닝

강경국 ○ 조성현

대구경북과학기술원

{kkk831, scho}@dgist.ackr

Automatic Panning for 360° Video Playback

Kyoungkook Kang ○ Sunghyun Cho

DGIST

요 약

2D 디스플레이로 360° 비디오를 감상하는 일반적인 방법은 비디오의 일부분을 일반적인 시야각(Normal Field of View, NFoV)의 비디오로 렌더링 (rendering)하는 것이다. 이 방법을 사용하면 사용자가 자연스러운 NFoV 비디오를 감상할 수 있다는 장점이 있지만 비디오의 중요한 장면을 놓치지 않기 위해 사용자가 계속해서 보는 방향을 조정해줘야 하는 번거로움이 있다. 이를 해결하기 위해 최근 몇 년 사이 자동으로 볼만한 위치를 찾아 이를 하나의 NFoV 비디오로 생성해 주는 방법들이 제시되었다. 그러나 기존 방법들은 빠르게 움직이는 물체를 제대로 처리하지 못하거나 안정적인 경로를 찾는데 어려움을 겪었다. 본 연구에서는 360° 비디오의 optical flow와 saliency 점수를 계산하고 이를 기반으로 중요한 장면을 포함하면서도 부드럽게 변하는 최적의 경로를 찾아 사용자에게 제공하는 시스템을 제시한다.

1. 서 론

최근 몇 년간 모든 방향을 감상할 수 있는 360° 비디오가 인기를 얻고 있다. Youtube와 같은 서비스에서 이미 수많은 360° 비디오를 제공하고 있으며, 삼성 Gear 360이나 GoPro Fusion 360과 같은 360° 카메라 역시 인기를 얻고 있다.

모든 방향을 한번에 기록하는 360° 카메라의 특성 때문에 360° 비디오를 감상하는 가장 편한 방법은 head-mounted display (HMD)를 착용하고 사용자가 머리와 몸을 돌려 원하는 방향의 영상을 감상하는 것이다. 하지만 HMD는 항상 이용 가능하지 않고 비디오를 감상할 때마다 HMD를 착용하는 것은 번거로운 일이기 때문에 더 흔하게 이용되는 방법은 컴퓨터 스크린이나 스마트폰과 같은 2D 디스플레이를 통해 영상을 감상하는 것이다.

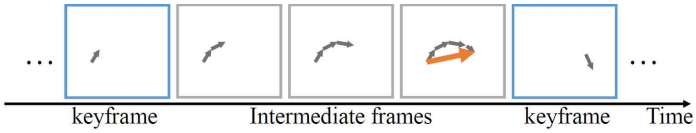
2D 디스플레이를 통해 360° 비디오를 감상하는 한 가지 방법은 전체 비디오를 구형 파노라마 (spherical panorama)로 투영하여 사용자가 모든 방향을 한번에 볼 수 있게 하는 것이다. 그러나 구형 파노라마는 영상의 상단과 하단에 심각한 왜곡을 유발한다. Youtube나 Facebook에서 사용되는 더 일반적인 방법은 비디오의 일부분을 선택하여 그 영역만을 2D 영상으로 변환하여 사용자에게 보여주는 방법이다. 이 방법은 기존의 일반적인 카메라로 촬영한 비디오처럼 일반적인 시야각(NFoV; Normal Field of View)을 갖는 자연스러운 영상을 감상할 수 있다는 장점이 있다. 비디오를 감상하는 동안에 사용자는 마우스나 터치스크린을 이용하여 원하는 방향의 영상을 감상할 수 있다. 그러나 이러한 방식은 사용자가 보는 방향을 매 순간 지정해줘야 하고 주의 깊게 영상을 감상하지 않으면 중요한 장면이 나오는 방향을 놓칠 수 있다는 문제점이 있다.

최근 이러한 문제를 해결하기 위한 연구가 소개되었다. [1,2,3] 이들은 360° 비디오를 분석하여 가장 중요한 부분들을 통과하는 최적의 가상 카[1]메라 경로를 찾는다. 이 가상 카메라 경로를 따라 NFoV 비디오가 추출되고 추출된 영상을 감상함으로써 사용자는 매순간 방향을 지정하지 않고서도 흥미로운 부분들이 포함된 영상을 감상할 수 있다.

Su et al. [1] 및 Su와 Grauman [2]은 비디오를 5초 간격으로 분할하고 분할된 영상에 대해 각 픽셀을 중심으로 하는 capture-worthiness 점수를 계산한 후 시간 축을 따라 누적된 capture-worthiness 점수가 가장 크도록 하는 경로를 계산하는 방법들을 제안했다. 하지만 이 방법들은 5초 간격으로 경로를 계산하기 때문에 역동적으로 빠르게 움직이는 동영상에 대해서는 한계가 있다. Hu et al. [3]은 컨볼루션 신경망(CNN; convolutional neural network)을 통해 영상의 가장 중요한 물체를 따라가는 가상 카메라 경로를 찾는 방법을 제안했다. 하지만 이 방법은 매 순간 하나의 중요한 물체만 존재한다고 가정하므로 여러 개의 중요한 물체가 있을 때 중요한 물체 간에 불안정적으로 이동하는 경로를 찾는 문제가 있다.

본 논문에서는 360° 비디오에서 효과적으로 중요한 장면을 통과하는 가상 카메라 경로를 찾아 NFoV 비디오를 만드는 방법을 제시한다. 본 논문이 제안하는 시스템은 비디오에서 중요한 부분을 지나가는 경로를 찾기 위해 saliency를 이용한다. 또한 본 논문은 부드러우면서도 자연스러운 경로를 위해 optical flow를 반영하여 경로를 찾는 방법을 제시한다. 이를 통하여 기존 방법에서는 다루지 못했던 역동적으로 움직이는 컨텐츠에 대해서도 효과적인 경로를 찾을 수 있으며, 또한 중요한 물체가 여러 곳에 동시에 존재하는 경우에도 가상 카메라의 경로가 현재 선택된 물체의 optical flow를 따라 감으로써 보다 안정적인 경로를 얻을 수 있다. 본 논문의 실험 결과는 본 논문이 제안하는 시스템이 기존 방법 대비 고품질의

[1] [1] **감사의 글** - 이 논문은 2017년도 정부(과학기술정보통신부)의 재원으로 한국연구재단-차세대정보컴퓨팅기술개발사업의 지원을 받아 수행된 연구임(NRF-2017M3C4A7066316).



NFoV 비디오를 생성할 수 있음을 보여준다.

그림 1. 키 프레임에서 다음 키 프레임으로의 누적 optical flow 계산.

2. Optical flow와 saliency 점수 계산

본 논문의 시스템은 $W \times H$ 사이즈의 구형 파노라마 형태의 360° 비디오를 입력 받는다. Optical flow와 saliency 점수의 효과적인 계산을 위해서 입력 비디오를 가로 크기가 360 픽셀이 되도록 다운-샘플링한다. 입력 비디오의 좌우 양 끝단에서 optical flow와 saliency 점수가 정확하게 계산되게 하기 위해 오른쪽과 왼쪽 끝의 픽셀들을 이용하여 비디오의 좌우에 각각 패딩을 한다. 본 논문에서는 양 끝단에 20픽셀씩 패딩한다. 패딩이 된 비디오에서 optical flow와 saliency 점수를 계산하였다. 본 시스템에서는 optical flow와 saliency 점수 각각의 계산을 위해 각각 Liu [4]와 Zhou et al. [5]의 방법을 이용하였다.

Liu [4]의 방법은 brightness constancy 가정을 기반으로 successive over-relaxation 방법을 이용해 연속한 두 프레임 사이의 optical flow를 계산한다. Zhou et al. [5]의 방법은 비디오를 시공간적 영역으로 over-segmentation하고 각각의 영역에 대해서 hand-crafted feature vector를 계산했다. 그리고 feature contrast에 기반하여 최종 비디오 saliency 점수를 계산했다. 최종적으로 구해진 optical flow와 saliency 점수는 다운 샘플링 된 입력 비디오와 같은 크기를 갖는다. Saliency 점수는 0과 1사이의 값을 갖고 1에 가까울수록 해당 픽셀이 더욱 salient하다는 것을 의미한다.

본 시스템은 계산의 효율성을 위해서 본 연구에서는 4 프레임 간격의 키 프레임에 대해서 최적의 경로를 찾는다. 따라서 하나의 키 프레임에서 다음 키 프레임으로의 누적 optical flow를 계산해야 한다. 그림 1은 누적 optical flow를 계산하는 과정을 나타낸다. Optical flow와 saliency 점수를 계산한 후 좌 우의 패딩 된 부분을 제거한다.

3. 가상 카메라 경로 계산

앞에서 계산된 saliency와 optical flow를 바탕으로 본 논문에서 제시하는 시스템은 360° 비디오에서 매 시간마다 saliency 점수가 높은 장면을 보여주면서도 사용자가 보기 편하도록 방향 전환이 급격하게 이루어지지 않는 경로를 찾는다. 구체적으로 T개의 키 프레임 $F^{key} = \{f_1^{key}, \dots, f_T^{key}\}$과 이에 해당하는 saliency 점수 맵 $\{s_1, \dots, s_T\}$ 와, optical flow 맵 $\{o_1, \dots, o_T\}$이 주어졌을 때 최적의 경로 $P = \{p_1, \dots, p_T\}$를 찾는다. 이 때 p_t는 t번째 키 프레임에서 다운 샘플링 된 구형 파노라마상의 픽셀 좌표이다. 본 논문에서는 다음과 같은 에너지 함수를 최소화하여 통해 최적의 경로를 찾는다.

$$E(P) = \sum_{t=1}^{T} |1 - s_t(p_t)| + \alpha \sum_{t=1}^{T-1} \|(p_{t+1} - p_t) - o_t(p_t)\| \qquad (1)$$

그림 2. 본 논문에서 제시하는 수식 1의 효과.

이 식에서 T는 키 프레임의 수로 전체 프레임 수의 $\frac{1}{4}$이다. $s_t(p_t)$과 $o_t(p_t)$는 각각 t번째 키 프레임의 좌표 p_t에서의 saliency 점수와 optical flow를 나타낸다. α는 첫번째 항과 두번째 항의 상대적 강도를 조절해주는 파라미터로 본 연구에서는 0.1을 사용했다.

에너지 함수의 첫번째 항은 saliency 항으로 각 키 프레임에서의 경로가 가장 salient한 쪽으로 가게 하는 항이고 두번째 optical flow 기반 temporal smoothness 항으로 가상 카메라의 경로가 비디오 내의 움직이는 물체나 360° 비디오에서의 카메라의 모션을 따라가게 한다.

수식 1의 두번째 항은 Deselaers et al.[6]와 Lai et al.[7] 등에서 시간적으로 부드러운 카메라의 경로를 찾기 위해 사용된 다음 수식과 매우 유사하다.

$$\alpha \sum_{t=1}^{T-1} \|p_{t+1} - p_t\| \qquad (2)$$

그러나 본 시스템의 optical flow 기반 temporal smoothness 항은 다음과 같은 주목할 만한 효과가 있다. 첫째로 optical flow가 물체의 움직임을 반영하기 때문에 가상 카메라 경로가 움직이는 물체를 효과적으로 추적할 수 있도록 해 준다. 둘째로 경로가 여러 salient 영역 사이를 빠르게 오가는 것을 막아준다. 여러 개의 salient 영역이 있을 때 수식 1의 첫번째 항만을 최소화하면 경로가 salient 영역 사이를 불안정하게 이동할 수 있다. 이 때 수식 2를 사용하게 되면 경로의 빠른 변화는 방지할 수 있으나 빠르게 움직이는 물체를 추적하는 데 어려움을 겪게 된다. 하지만 본 시스템의 optical flow 기반 temporal smoothness 항은 경로가 물체의 optical flow를 따라가도록 하기 때문에 빠르게 움직이는 물체를 추적하면서도 경로가 중요한 물체 사이를 빠르게 움직이는 것을 방지할 수 있다.

본 논문에서는 수식 1을 최적화하기 위해서 동적 프로그래밍 기법을 적용했다. 구해진 최적의 해는 키 프레임에 대해서 최적의 경로로 구성되고 이를 선형적으로 보간(interpolation)하여 모든 프레임에 대한 경로 $G = \{g_1, \dots, g_N\}$를 구할 수 있다. 이 때 N은 총 프레임의 수이다. 최적의 경로 G는 수식 1의 두 번째 항의 효과로 부드럽게 최적화되지만 두 개의 연속한 키 프레임에 기반하였기 때문에 약간의 흔들림이 남아있을 수 있다. 뿐만 아니라 선형 보간법은 모든 프레임에 대한 부드러운 경로를 보장하지 않는다. 이러한 흔들림을 억제하고 부드러운

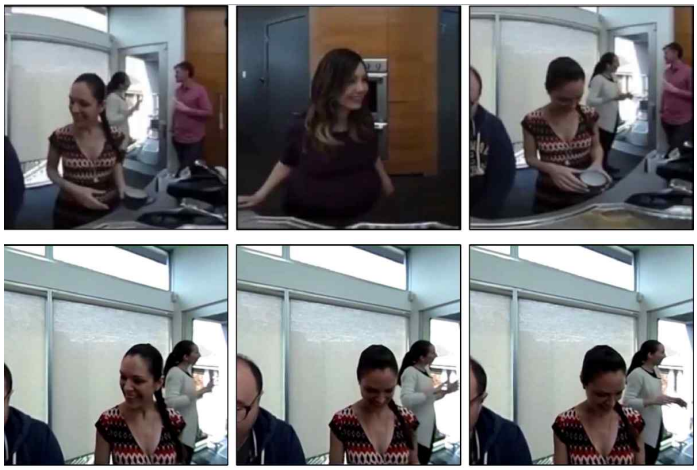


그림 3. Su et al. [1]과 본 논문의 시스템의 결과

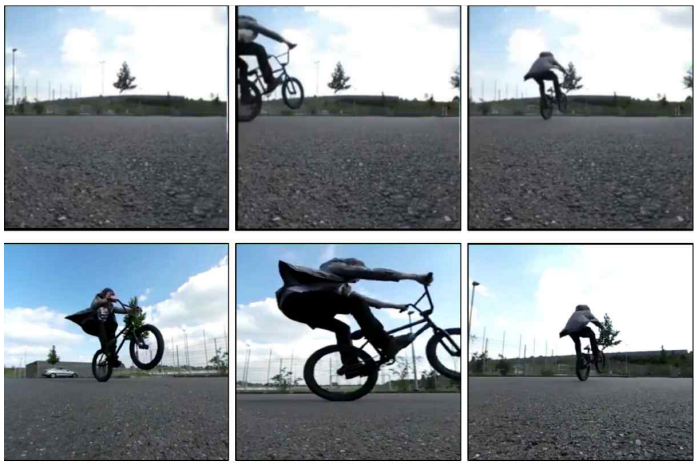


그림 4. Hu et al. [3]과 본 논문의 시스템의 결과.

경로 $\hat{G} = \{\hat{g}_1, \ldots, \hat{g}_N\}$를 만들기 위해서 본 연구에서는 다음과 같은 에너지 함수를 최소화하여 개선된 최종 경로를 구한다.

$$E^{ref}(\hat{G}) = \sum_{t=1}^{N} \|\hat{g}_t - g_t\|^2 + w_v \sum_{t=1}^{N-1} \|\hat{g}_{t+1} - \hat{g}_t\|^2 + w_a \sum_{t=2}^{N-1} \|\hat{g}_{t+1} - 2\hat{g}_t + \hat{g}_{t-1}\|^2 \quad (3)$$

에너지 함수의 첫번째 항은 데이터 항, 두번째 항과 세번째 항은 각각 속도와 가속도를 기반으로 하는 smoothness 항이다. w_v 와 w_a는 smoothness 항의 계수로 본 연구에서는 각각 **10**과 **20**을 사용했다. 수식(3)은 단순한 이차 함수로 최소 자승법을 이용하여 최적의 해를 구할 수 있다.

마지막으로 구해진 최적의 경로를 따라 각 시간마다 선택된 지점을 중심으로 NFoV 비디오를 렌더링(rendering)하여 자동으로 방향 전환되는 비디오를 생성한다.

4. 실험 결과 및 분석

본 논문에서는 [1,2,3]의 데이터 셋에 포함된 다양한 비디오에 대해 실험을 진행했다. 그림 2는 빠르게 움직이는 여러 물체가 있는 비디오에 대한 결과를 시간 순서대로 배열했다. 첫 번째 행은 기존 방법에서 일반적으로 사용된 수식 2를 사용한 결과이고 두 번째 행은 본 논문에서 제시하는 optical flow를 기반으로 한 smoothness 항을 사용하는 결과이다. 본 논문의 시스템은 물체의 optical flow를 따라가는 경로를 제공하기 때문에 중앙의 붉은 유니폼을 입은 선수를 더 효과적으로 추적하는 결과를 보여준다.

그림 3은 Su et al.[1]과 본 논문의 결과를 각각 첫 번째와 두 번째 행에 배열했다. Su et al.[1]의 방법은 5초 간격으로 경로를 계산해야 하기 때문에 빠르게 움직이는 자전거를 추적하는데 실패한다. 반면 본 논문이 제시하는 시스템은 4 프레임 간격의 키프레임에 대해서 optical flow를 활용하여 경로를 구하기 때문에 자전거를 효과적으로 추적하는 것을 확인할 수 있다.

그림 4는 Hu et al.[3]과 본 논문의 결과를 각각 첫번째와 두 번째 행에 배열했다. Hu et al.[3]의 방법은 하나의 중요한 물체를 따라가도록 설계되어 있으며 여러 개의 중요한 물체가 존재하는 경우 가상 카메라의 경로가 물체 사이에 불안정적으로 빠르게 전환하여 사용자가 보기 불편한 결과를 만든다. 반면에 본 논문의 결과는 비디오 내의 물체의 움직임이 고려되어 하나의 중요한 물체를 안정적으로 추적하는 비디오를 생성한다.

5. 결론

본 논문에서는 360° 비디오를 보다 편하게 감상하기 위해 영상을 분석해 자동으로 중요한 부분을 따라 방향 전환되는 비디오를 생성하는 방법을 제시하였다. 또한 본 논문에서 제시하는 방법은 optical flow를 활용하여 기존 방법에 비해 빠르게 움직이는 비디오나 여러 개의 중요한 물체가 나오는 비디오에 대해서도 대응 가능하다.

참 고 문 헌

[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] Hou-Ning Hu, Yen-Chen Lin, Ming-Yu Liu, Hsien-Tzu Cheng, Yung-Ju Chang, Min Sun. Deep 360 Pilot: Learning a Deep Agent for Piloting through 360 Sports Video. In CVPR 2017

[4] Ce Liu. Beyond pixels: exploring new representations and applications for motion analysis. Ph.D. Dissertation. MIT

[5] Feng Zhou, Sing Bing Kang, and Michael F Cohen. Time-mapping using space-time saliency. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 3358-3365

[6] Feng Liu and Michael Gleicher. Video Retargeting: Automating Pan and Scan. In Proceedings of the 14th ACM international Conference on Multimedia.

[7] Wei-Sheng Lai, Yujia Huang, Neel Joshi, Christopher Buehler, Ming-Hsuan Yang, and Sing Bing Kang. Semantic-driven generation of hyperlapse from 360 video. IEEE Transactions on Visualization and Computer Graphics