특집논문 (Special Paper)
방송공학회논문지 제24권 제5호, 2019년 9월 (JBE Vol. 24, No. 5, September 2019)
https://doi.org/10.5909/JBE.2019.24.5.755
ISSN 2287-9137 (Online) ISSN 1226-7953 (Print)

단안 비디오로부터의 5차원 라이트필드 비디오 합성

배 규 호[a], 안드레 이반[a], 박 인 규[a)‡]

5D Light Field Synthesis from a Monocular Video

Kyuho Bae[a], Andre Ivan[a], and In Kyu Park[a)‡]

요 약

현재 사용 가능한 상용 라이트필드 카메라는 정지 영상만을 취득하거나 가격이 매우 높은 단점으로 인하여 5차원 라이트필드 비디오 취득에 어려움이 있다. 이러한 문제점을 해결하기 위해 본 논문에서는 단안 비디오로부터 라이트필드 비디오를 합성하기 위한 딥러닝 기반 기법을 제안한다. 라이트필드 비디오 학습 데이터를 취득하기 어려운 문제를 해결하기 위하여 UnrealCV를 활용하여 3차원 그래픽 장면의 사실적 렌더링에 의한 합성 라이트필드 데이터를 취득하고 이를 학습에 사용한다. 제안하는 딥러닝 프레임워크는 입력 단안 비디오에서 9×9의 각 SAI(sub-aperture image)를 갖는 라이트필드 비디오를 합성한다. 제안하는 네트워크는 밝기 영상으로 변환된 입력 영상으로부터 appearance flow를 추정하는 네트워크, appearance flow로부터 얻어진 인접한 라이트필드 비디오 프레임간의 optical flow를 추정하는 네트워크로 구성되어 있다.

Abstract

Currently commercially available light field cameras are difficult to acquire 5D light field video since it can only acquire the still images or high price of the device. In order to solve these problems, we propose a deep learning based method for synthesizing the light field video from monocular video. To solve the problem of obtaining the light field video training data, we use UnrealCV to acquire synthetic light field data by realistic rendering of 3D graphic scene and use it for training. The proposed deep running framework synthesizes the light field video with each sub-aperture image (SAI) of 9×9 from the input monocular video. The proposed network consists of a network for predicting the appearance flow from the input image converted to the luminance image, and a network for predicting the optical flow between the adjacent light field video frames obtained from the appearance flow.

Keywords : Deep learning, Light field, Video synthesis, View synthesis

a) 인하대학교 정보통신공학과(Inha University, Department of Information and Communication Engineering)
‡ Corresponding Author : 박인규(In Kyu Park)
E-mail: pik@inha.ac.kr
Tel: +82-32-860-9190
ORCID: https://orcid.org/0000-0003-4774-7841
※ 이 논문의 연구 결과 중 일부는 한국방송·미디어공학회 “2019년 하계학술대회”에서 발표한 바 있음.
※ 본 논문은 삼성전자 미래기술육성센터의 지원을 받아 수행한 연구결과임(과제번호 SRFC-IT1702-06).
※ This work was supported by Samsung Research Funding & Incubation Center for Future Technology(Project number SRFC-IT1702-06)
· Manuscript received July 28, 2019; Revised September 10, 2019; Accepted September 10, 2019.

Ⅰ. 서 론

라이트필드 영상은 다양한 방향에서의 빛의 정보를 취득함으로써 한 장의 영상만으로 깊이 영상 추정(depth image estimation)[17,18,19,20], 영상 재 초점(refocusing)[21], 시점 이동(view-point change) 등의 다양한 영상처리가 가능하다는 장점이 있다. 일반적으로 라이트필드 영상은 마이크로 렌즈 배열을 이용한 플렌옵틱(plenoptic)카메라 혹은 다중 카메라 배열(camera array)을 사용하여 취득한다[10]. 그러나 일반 사용자가 사용 가능하던 Lytro사의 카메라는 더 이상 지원이 되지 않고 유일하게 남아있는 Raytrix[1]사의 카메라는 라이트필드 비디오 촬영이 가능하지만 산업 현장에서 사용하기 위한 매우 고가의 제품이므로 일반 사용자가 사용하기에 어려움이 따른다. 이러한 한계점을 극복하기 위해 일반 영상으로부터 라이트필드 영상을 합성하는 다양한 기법이 소개되었다[3,4,5,11,12,13,14,15]. 하지만 해당 기법들은 모두 비디오가 아닌 정지 영상으로부터 4차원 라이트필드 영상을 합성하는 기법으로 라이트필드 비디오 합성에는 적절하지 않다.

기존의 기법 중 [3]의 경우 한 장의 라이트필드 영상을 합성하기 위해 특정 시점의 입력 영상 4장을 요구하며, [11,12,13,14,15]의 경우도 마찬가지로 한 장이 아닌 여러 장의 영상을 요구하여 일반 사용자 입장에서 이를 취득하기 어렵다는 단점이 있다. [4]의 경우 한 장의 입력 영상으로부터 깊이 추정 네트워크를 통해 깊이 영상을 추정하고 이후 컬러 합성 네트워크를 통해 angular domain에서 일관성있는 라이트필드 영상을 합성한다. 그러나 해당 기법의 경우 합성된 라이트필드 영상이 합성된 깊이 영상의 품질에 크게 의존하고 특정 데이터셋에 제한적이며 폐색(occulusion) 영역을 복원하는 데 어려움이 있다. [5]의 경우 [4]의 한계점을 극복하기 위해 깊이 영상 대신 appearance flow [16]를 사용하여 라이트필드 영상을 합성하였다. 그러나 [5] 역시 [3], [4]와 마찬가지로 정적인 물체에 대한 라이트필드 영상을 합성하였기 때문에 라이트필드 비디오를 합성하는 데는 적합하지 않다.

기존의 라이트필드 비디오를 합성하는 연구로서 일반 DSLR 카메라와 라이트필드 카메라를 결합한 하이브리드 카메라 시스템을 고안하고, 초당 3프레임을 갖는 라이트필드 비디오를 초당 30프레임의 라이트필드 비디오로 합성하는 기법이 제안되었다[6]. 그러나 본 기법은 라이트필드 비디오를 합성하기 위해 일반 카메라뿐만 아니라 라이트필드 카메라 또한 필요하며 두 카메라 간의 시점 불일치로 인한 오차가 발생한다는 한계점이 존재한다.

본 논문에서는 이상의 한계점을 극복하여 단일 영상이 아닌 단안 비디오로부터 5차원 라이트필드 비디오를 합성하는 딥러닝 기반 기법을 제안한다. 본 논문의 구성은 다음과 같다. 2장에서는 딥러닝 네트워크를 학습하기 위해 사용한 합성 라이트필드 데이터를 소개하고, 입력 단안 비디오에서 라이트필드 비디오를 합성하는 end-to-end 딥러닝 네트워크를 제안한다. 3장에서는 제안하는 기법의 성능을 정량적, 정성적으로 평가하고 타 기법과의 비교 평가를 수행한다. 마지막으로, 4장에서 본 논문의 결론을 맺는다.

Ⅱ. 제안하는 기법

1. 합성 라이트필드 데이터셋 구성

다양한 컴퓨터 비전 분야에서 활용되고 있는 딥러닝 기반 영상처리 기법들은 각각의 기법에서 제안하는 네트워크를 학습하기 위해 다량의 영상 데이터셋를 요구한다. 하지만 특정한 목적에 정확히 부합하는 영상 데이터를 취득하기에 어려움이 많으며 특히 라이트필드 비디오 학습 데이터의 경우에는 그 희소성으로 인해 더 큰 어려움이 따른다. 본 논문에서는 이러한 한계점을 극복하기 위해 3차원 그래픽 장면의 사실적 그래픽 렌더링에 의한 합성 라이트필드 데이터를 취득하고 이를 학습 데이터로 활용한다. 합성 라이트필드 데이터를 취득하기 위해 오픈 소스 게임엔진인 Unreal Engine 4를 기반으로 하는 UnrealCV[7]를 활용하여 9×9의 angular domain을 갖는 합성 라이트필드 데이터를 취득하였으며, 학습에 사용된 3차원 그래픽 장면은 실제 도시와 유사한 외관을 가진 두 개의 장면을 사용하였다. 두 개의 장면은 [7]에서 제공하는 3차원 그래픽 장면과 본 논문에서 구성한 3차원 그래픽 장면으로 구성되어 있으며 이를 그림 1에 나타내었다. 그림 1의 (a)는 [7]에서 제공하는

그림 1. 학습에 사용된 3차원 그래픽 장면. (a) [7]에서 제공하는 3차원 그래픽 장면; (b) 본 논문에서 구성한 3차원 그래픽 장면
Fig. 1. The 3D graphic scenes used for training. (a) 3D graphic scene from [7]; (b) 3D graphic scene from ours

3차원 그래픽 장면의 일부이며, (b)는 본 논문에서 구성한 3차원 그래픽 장면의 일부이다. 각 그래픽 장면에서 구성된 도로를 따라 카메라 배열을 구성하고 이를 이동시키며 1,818장의 9×9의 SAI로 구성된 합성 라이트필드 비디오 데이터를 취득하였다.

2. 딥러닝 프레임워크

본 논문에서 제안하는 입력 단안 비디오로부터 5차원 라이트필드 비디오를 합성하는 딥러닝 프레임워크를 그림 2에 나타내었다. 전체 프레임워크는 입력 단안 비디오로부

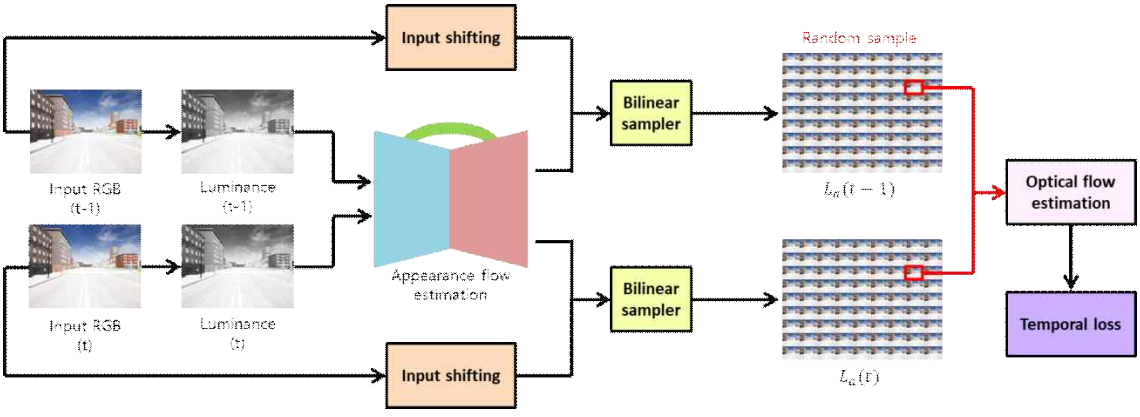

그림 2. 제안하는 프레임워크
Fig. 2. Proposed framework

터 9×9의 각 SAI에 대응되는 appearance flow를 추정하는 플로우 추정 네트워크, 그리고 추정된 플로우를 각 입력 단안 비디오에 워핑하여 얻은 합성 라이트필드 비디오 프레임간의 광학 플로우를 추정하는 광학 플로우 추정 네트워크로 구성되어 있다. 각 단계들을 아래와 같이 수식으로 표현할 수 있다:

$$\tilde{L}_a(x,y,u,v,t) = B(L_s(x,y,u,v,t), L_f(x,y,u,v,t)) \quad (1)$$

$$L_s(x,y,u,v,t) = L(x-(\eta\times\triangle u), y-(\eta\times\triangle v), 0, 0, t \quad (2)$$

$$F_{(t-1)\to t} = o(\tilde{L}_a(x,y,u,v,t-1), \tilde{L}_a(x,y,u,v,t)) \quad (3)$$

수식 (2)의 $L_s(x,y,u,v,t)$는 입력 밝기 영상을 각 angular domain 좌표에 따라 angular 이동 상수 η에 비례한 수치만큼 영상의 화소 좌표를 이동시킨 영상을 나타낸다. 수식 (1)의 $L_f(x,y,u,v,t)$는 플로우 추정 네트워크로부터 얻은 시간 t에서의 각 SAI에 대응되는 appearance flow를 나타내며 bilinear sampler module [8] B를 통해 $L_s(x,y,u,v,t)$에 $L_f(x,y,u,v,t)$를 워핑하여 시간 t에서의 라이트필드 영상 $\tilde{L}_a(x,y,u,v,t)$를 얻을 수 있다. o는 시간 $t-1$과 t간의 광학 플로우 $F_{(t-1)\to t}$를 추정하는 광학 플로우 추정 네트워크를 나타낸다.

Appearance flow를 추정하는 네트워크인 플로우 추정 네트워크는 [5]에서 사용한 구조 대신 좀 더 경량 구조이면서 다양한 분야에서 그 효용을 보인 encoder-decoder 구조를 사용한다. 추정한 appearance flow를 이용해 얻은 라이트필드 비디오 프레임 $\tilde{L}_a(x,y,u,v,t)$를 학습하기 위해 [5]에서 제안한 바와 같이 아래와 같은 손실함수를 정의한다:

$$\begin{aligned}\ell_{global} = &\left|Mean(\tilde{L}_a(x,y,u,v,t)) - Mean(L(x,y,u,v,t))\right| + \\ &\left|Var(\tilde{L}_a(x,y,u,v,t)) - Var(L(x,y,u_{ij},v_{ij},t))\right|\end{aligned} \quad (4)$$

$$\begin{aligned}\ell_{global} = &\sum_{i,j=1}^{I,J}\left|Mean(\tilde{L}_a(x,y,u,v,t)) - Mean(L(x,y,u,v,t))\right| \\ &+\sum_{i,j=1}^{I,J}\left|Var(\tilde{L}_a(x,y,u,v,t)) - Var(L(x,y,u_{ij},v_{ij},t))\right|\end{aligned} \quad (5)$$

수식 (4)와 (5)에서 $Mean(L)$과 $Var(L)$은 각각 라이트필드 SAI들의 평균과 분산을 의미한다. 수식 (4)는 전체 라이트필드 SAI들의 평균과 분산에 대한 손실함수를 의미하며, 수식 (5)는 라이트필드의 SAI를 하나의 배열로 봤을 때 각 행과 열의 SAI들의 평균과 분산에 대한 손실함수를 의미한다.

광학 플로우 추정 네트워크는 그림 3와 같이 계층적 구조를 갖는 encoder-decoder구조를 사용한다. 정확한 ground truth 광학 플로우를 얻는 것에는 많은 어려움이 있기 때문에 비 지도 학습 방법으로 광학 플로우 추정 네트워크를 학습한다. 추정한 광학 플로우를 이용해 시간 $t-1$과 t에서의 라이트필드 간의 시간적 일관성을 부여하기 위해 아래와 같은 손실함수를 정의한다:

$$\begin{aligned}&\ell_{temporal} \\ &= \omega\left|\tilde{L}_a(x,y,u_{sample},v_{sample},t) - L_{warp}^{t-1\to t}(x,y,u_{sample},v_{sample})\right|\end{aligned} \quad (6)$$

$$\omega = \lambda_t \exp(-\lambda_s\left|L(x,y,0,0,t-1) - L(x,y,0,0,t)\right|) \quad (7)$$

수식 (7)은 두 입력 영상간의 차이에 따라 시간적 일관성을 다르게 부여하기 위한 가중함수이다. 즉, 시간 $t-1$에서의 프레임과 시간 t에서의 프레임간의 차이가 클 경우 시간적 일관성을 적게 부여하고 두 프레임간의 차이가 작은 경우 상대적으로 더 많은 시간적 일관성을 부여한다. 수식 (6)은 그림 3에서 얻은 광학 플로우를 사용하여 시간 $t-1$의 프레임을 시간 t로 워핑하여 얻은 라이트필드 프레임과 시간 에서 appearance flow를 통해 얻은 라이트필드 $\tilde{L}_a$ 간의 L1 손실함수이다. 또한 그림 3에서 각 단계마다 얻어지는 광학 플로우간의 일관성을 부여하기 위해 다음과 같은 손실 함수를 정의한다:

$$\begin{aligned}&\ell_{opticalconsistency} \\ &= \lambda_1\left|F_{step1} - down(F_{step3})\right| + \lambda_2\left|F_{step2} - down(F_{step3})\right|\end{aligned} \quad (8)$$

수식 (8)에서 $down(F_{step3})$는 각 단계에서의 영상 해상도에 맞게 최종 광학 플로우를 다운샘플링 하는 함수이며 λ_n은 단계 n에서의 가중치를 의미한다.

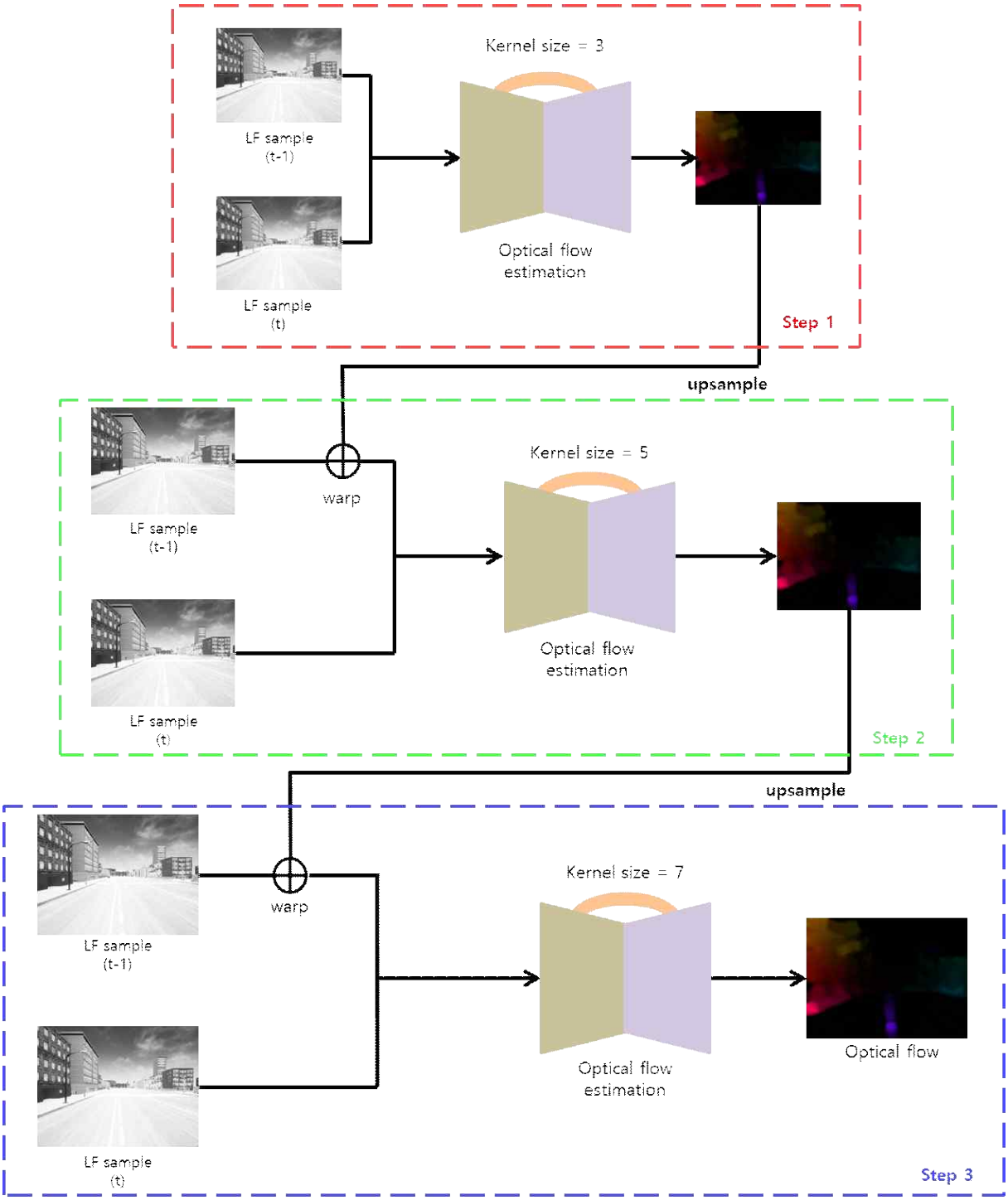

그림 3. 광학 플로우 추정 네트워크의 구조
Fig. 3. Architecture of optical flow estimation network

Ⅲ. 실험 결과

본 논문에서 제안한 기법의 정량적, 정성적 평가 결과를 아래에 나타내었다. 또한, 단일 영상으로부터 라이트필드 영상을 합성하는 최신 기법인 [4]와의 정량적인 비교 평가도 나타내었다. 제안하는 네트워크는 2의 배치 크기(batch size)를 갖고 end-to-end 방식으로 학습을 150,000회 반복 수행하였다. 본 논문에서 사용한 최적화 알고리즘으로 Adam optimizer[9]를 사용하였으며 해당 최적화 알고리즘의 기본 매개 변수를 사용하였다. 제안하는 네트워크는 TensorFlow[2]를 통해 구현하였으며, 학습 시간은 11GB의 메모리를 갖는 NVIDIA GTX 1080Ti GPU, Intel i7-6700 3.40GHz CPU, 16GB RAM 환경에서 약 16시간 정도가 소요되었다. 입력 단안 비디오의 두 장의 프레임으로부터 두 장의 라이트필드 영상을 합성하는 데 걸리는 시간은 약 0.4초 정도가 소요되었다.

1. 3차원 그래픽 장면에 대한 실험 결과

본 논문에서 제안한 기법의 3차원 그래픽 장면에 대한 실험 결과를 그림 4에 나타내었다. 입력 영상은 총 162장으로 이루어진 3차원 그래픽의 단안 비디오이며, 그림 4의 첫 번째 열은 162장의 프레임 중 7장의 샘플링 된 영상을 나타낸다. 두 번째와 세 번째 열은 각각 출력 영상의 파란색 선을 따라 자른 수직 EPI를 시간 순서로 쌓은 EPI 흐름과 원본 라이트필드 비디오의 EPI 흐름을 나타내며, 네 번째와 다섯 번째 열은 각각 출력 영상의 붉은색 선을 따라 자른 수평 EPI를 시간 순서로 쌓은 EPI 흐름과 원본 라이트필드 비디오의 EPI 흐름을 나타낸다. 입력 영상의 해상도는 192×192이며, 출력 라이트필드 비디오 프레임의 해상도는 192×192×9×9이다. 그림 4에서 나타내듯이 3차원 그래픽 장면에 대하여 시간적 일관성 있는 라이트필드 비디오를 합성할 수 있음을 보였다. 다만, 수직 EPI 흐름에서 카메라에 극단적으로 가까이 있는 물체의 경우 원본 영상에서의

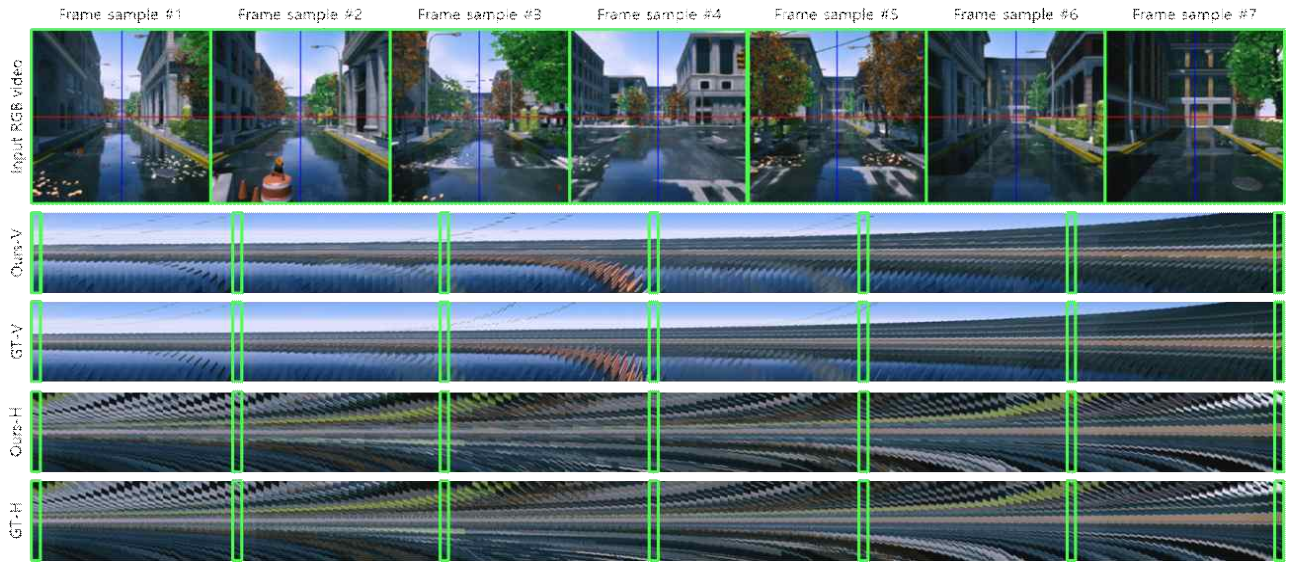

그림 4. 가상 환경 실험 영상(162 프레임)에 대한 라이트필드 비디오 합성 결과. 첫 번째 행은 입력 비디오의 샘플 프레임을 의미한다. 두 번째와 세 번째 행은 각각 제안하는 기법과 원본 영상의 수직 EPI를 시간 순서로 쌓은 결과이다. 네 번째와 다섯 번째 행은 각각 제안하는 기법과 원본 영상의 수평 EPI를 시간 순서로 쌓은 결과이다

Fig. 4. Light field video synthesis result of synthetic test scene (162 frames). First row is the sample frames of the input video. Second and third row is the vertical EPI stream of our result and ground truth, respectively. Fourth and fifth row is the horizontal EPI stream of our result and ground truth, respectively

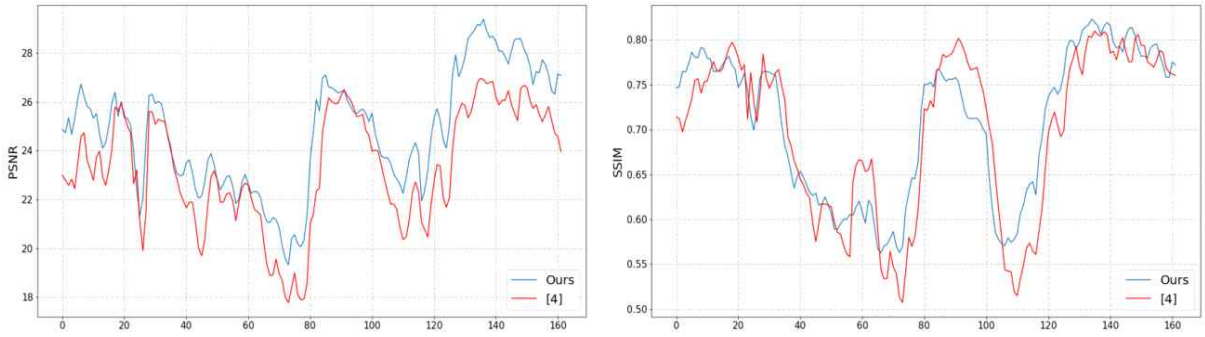

그림 5. 가상 환경 실험용 영상에 대한 정량적 비교
Fig. 5. Quantitative comparison of synthetic test scene

시점 변환에 비해 그 움직임이 적음을 보였다. 수평 EPI 흐름에서는 반대로 원본 영상과 유사한 수준의 시점 변환이 일어남을 보였다.

제안하는 기법과 기존의 단일 영상으로부터 라이트필드 영상을 합성하는 연구인 [4]와의 정량적인 비교 평가를 그림 5에 나타내었다. 비교에 사용한 실험 데이터는 그림 4에 사용한 3차원 그래픽의 단안 비디오를 사용하였으며, 입력 비디오의 프레임당 독립적으로 [4]의 기법을 적용하여 라이트필드 비디오를 생성하였다. 그림 5에서 알 수 있는 바와 같이 제안하는 기법은 [4]와 비교했을 때 PSNR과 SSIM 측면에서 월등함을 보였다. 또한, 실험 데이터에 대한 평균 PSNR과 SSIM을 표 1에 나타내었다. 표 1에서 나타내듯이 평균 PSNR과 SSIM 수치가 각각 약 1.5dB, 0.01만큼 높음을 보였다. 이와 같이 프레임당 독립적으로 라이트필드 영상을 생성하여 비디오를 생성하는 고전적인 기법에 비해, 라이트필드 비디오 자체를 시간 축에서 일관성 있게 생성하는 제안하는 기법이 효과적임을 알 수 있다.

표 1. 가상 환경 실험용 영상에 대한 평균 PSNR(dB), SSIM
Table 1. Average PSNR (in dB) and SSIM from synthetic test scene

Method	PSNR	SSIM
Srinivasan et al. [4]	23.31	0.699
Proposed	24.79	0.710

2. 실제 환경 영상에 대한 실험 결과

본 논문에서 제안한 기법의 정성적인 성능 분석을 위해 실제 환경 영상에 대해 실험한 결과를 그림 6에 나타내었다. 본 논문에서 실험에 사용한 실제 환경 영상으로는 다양한 컴퓨터 비전 분야에서 활용되고 있는 KITTI 데이터셋을 사용하였다. KITTI 데이터셋에서 제공하는 데이터의 형태는 라이트필드 영상이 아닌 스테레오 영상이기 때문에 정성적인 실험 결과만을 그림 6에 나타내었다. 그림 6에 나타낸 바와 같이 합성 라이트필드 데이터를 사용해 학습한 네트워크를 사용하여 실제 환경의 데이터에 대해서도 시간적으로 일관성 있는 라이트필드 비디오를 합성해 낼 수 있음을 보였다.

3. Ablation Study

본 논문에서 제안한 손실 함수의 효과를 분석하기 위해 ablation study를 수행하였다. 각각 화소 단위의 L1 손실함수(L1), 시간적 일관성 손실함수와 global 손실함수(O+G), 시간적 일관성 손실함수와 local 손실함수(O+L), global-local 손실함수(G+L), 그리고 시간적 일관성 손실함수와 global-local 손실함수(O+G+L)를 사용하여 실험을 수행하여 그 결과를 그림 7과 표 2에 나타내었다. 실험 결과 L1 손실함수와 비교했을 때 global 손실 함수의 효과는 적지만, local 손실함수와 함께 사용할 시 그 성능이 향상됨을 보였다. 또한, 시간적 일관성을 부여하지 않을 시 평균 PSNR과 SSIM이 소폭 하락하는 것을 보였으며 그림 7에서 나타내듯이 시간적 일관성을 부여하지 않을 시 부여했을 때 보다 그 안정성이 더 떨어지는 것을 보였다.

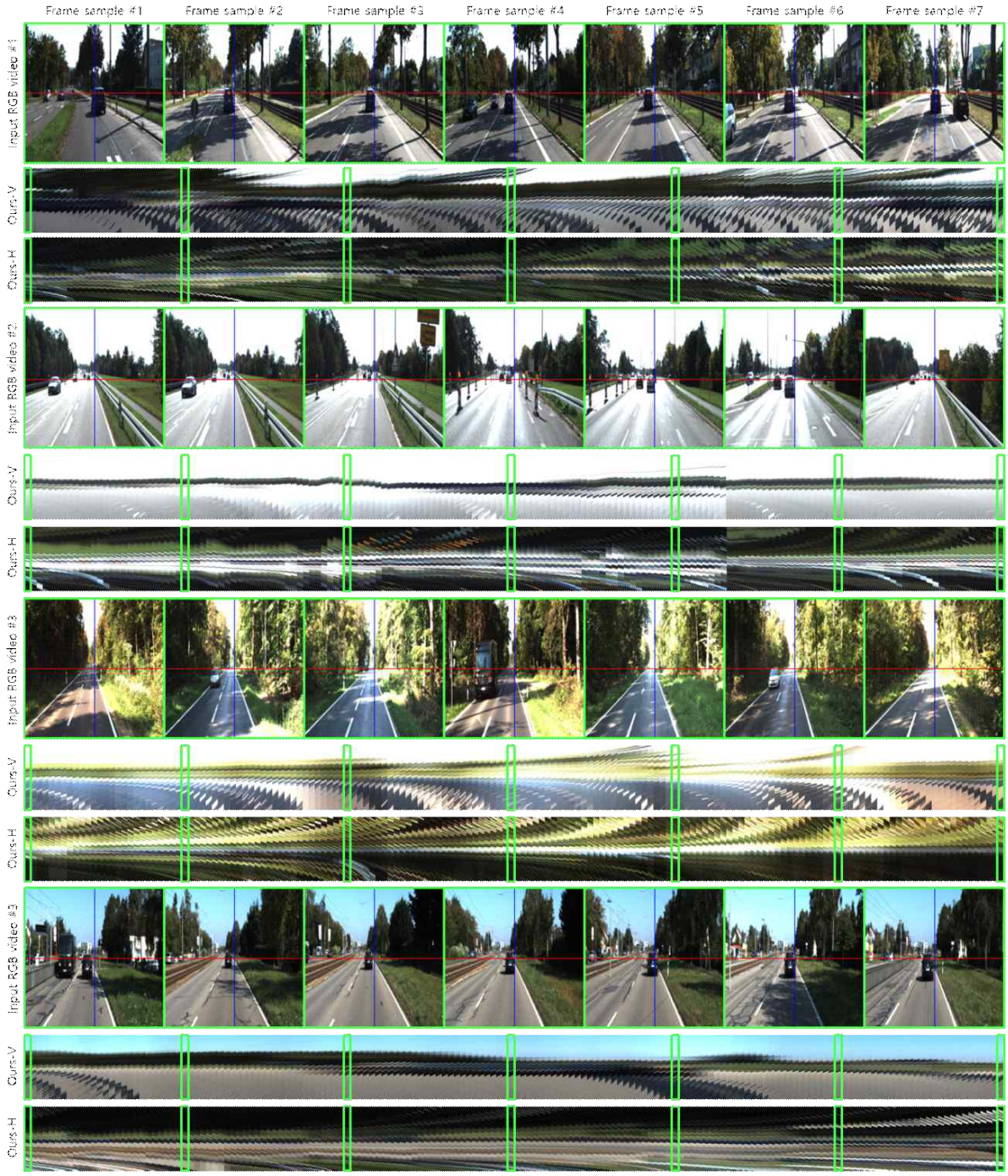

그림 6. 실제 환경에 대한 라이트필드 비디오 합성 결과. 각 비디오는 162개의 프레임을 갖는 KITTI 데이터셋의 영상이다. 3개의 각 행은 입력 단안 비디오 샘플, 수직 EPI 흐름, 수평 EPI 흐름을 나타낸다

Fig. 6. Light field video synthesis result for real scene. Each video consists of 162 frames of the KITTI dataset. Each 3 rows represent the input single video sample, vertical EPI stream, and horizontal EPI stream, respectively

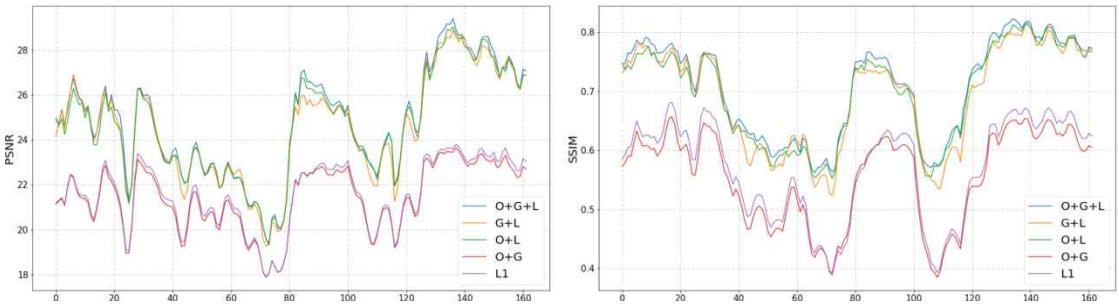

그림 7. 손실 함수에 따른 정량적 성능 비교
Fig. 7. Quantitative comparison of each loss function

표 2. 각 손실 함수에 따른 평균 PSNR(dB), SSIM
Table 2. Average PSNR (in dB) and SSIM of each loss function

Metric	L1	O+G	O+L	G+L	O+G+L
PSNR	21.61	21.45	24.63	24.56	24.79
SSIM	0.57	0.56	0.70	0.69	0.71

Ⅳ. 결 론

본 논문에서 제안한 기법은 합성 라이트필드 데이터를 활용해 딥러닝 네트워크를 학습하고 학습된 네트워크로부터 appearance flow를 얻은 뒤 이를 사용해 라이트필드를 합성하며, 마찬가지로 학습된 네트워크로부터 광학 플로우를 얻은 뒤 이를 사용해 시간적 일관성을 부여한다. 실험 결과를 통해 정량적, 정성적으로 기존의 기법보다 우월함을 보였다.

또한 실험 결과를 통해 합성 라이트필드 데이터를 활용해 학습한 네트워크로 3차원 그래픽 데이터 외에 실제 환경의 데이터에 대해서도 라이트필드 비디오를 합성해 낼 수 있음을 보였다. 향후 다양한 환경의 가상 환경 데이터를 구성하고 이를 활용하여 다양한 환경의 실제 환경에 대해서도 라이트필드 비디오를 합성해 낼 수 있다.

참 고 문 헌 (References)

[1] *Raytrix 3D Light Field Cameras*, https://raytrix.de/products
[2] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, and M. Kudlur, “Tensor-flow: A system for large-scale machine learning,” In Proc. of 12th Symposium on Operating Systems Design and Implementation, volume 16, pages 265-283, 2016.
[3] N. K. Kalantari, T. ‐C. Wang, and R. Ramamoorthi, “Learning-based view synthesis for light field cameras,” ACM Transactions on Graphics, 35(6): 193, 2016.
[4] P. P. Srinivasan, T. Wang, A. Sreelal, R. Ramamoorthi, and R. Ng, “Learning to synthesize a 4D RGBD light field from a single image,” In Proc. of IEEE International Conference on Computer Vision, pages 2243-2251, 2017.
[5] A. Ivan, Willem, and I. K. Park, “Synthesizing a 4D spatio-angular consistent light field from a single image,” arXiv preprint arXiv:1903.12364, 2019.
[6] T. ‐C. Wang, J. ‐Y. Zhu, N. K. Kalantari, A. A. Efros, and R. Ramamoorthi, “Light field video capture using a learning-based hybrid imaging system,” ACM Transactions on Graphics, 36(4): 133, 2017.
[7] W. Qiu and Y. Alan,”UnrealCV: Connecting computer vision to unreal engine,” In Proc. of European Conference on Computer Vision, pages 909-916, 2016.
[8] M. Jaderberg, K. Simonyan, and A. Zisserman, “Spatial transformer networks,” In Proc. of Advances in Neural Information Processing Systems, pages 2017-2025, 2015.
[9] D. P. Kingma and J. B. Adam, “Adam: A method for stochastic optimization,” In Proc. of International Conference on Machine Learning, 2015.
[10] B. Wilburn, N. Joshi, V. Vaish, E. ‐V. Talvala, E. Antunez, A. Barth, A. Adams, M. Horowitz, and M. Levoy, “High performance imaging using large camera arrays,” ACM Transactions on Graphics, 24(3), pages 765-776, 2005.
[11] G. Wu, M. Zhao, L. Wang, Q. Dai, T. Chai, and Y. Liu, “Light field reconstruction using deep convolutional network on EPI,” In Proc. of IEEE Conference on Computer Vision and Pattern Recognition, pages 6319-6327, 2017.
[12] H. Wing Fung Yeung, J. Hou, J. Chen, Y. Ying Chung, and X. Chen, “Fast light field reconstruction with deep coarse-to-fine modelling of spatial-angular clues,” In Proc. of European Conference on Computer Vision, pages 137-152, 2018.
[13] T. Zhou, R. Tucker, J. Flynn, G. Fyffe, and N. Snavely, “Stereo magnification: Learning view synthesis using multiplane images,” ACM Transactions on Graphics, 37(4):65:1-65:12, 2018.
[14] P. P. Srinivasan, R. Tucker, J. T. Barron, R. Ramamoorthi, R. Ng, and

N. Snavely, “Pushing the boundaries of view extrapolation with multi-plane images,” In Proc. of IEEE Conference on Computer Vision and Pattern Recognition, pages 175-184, 2019.

[15] B. Mildenhall, P. P. Srinivasan, R. Ortiz-Cayon, N. K. Kalantari, R. Ramamoorthi, R. Ng, and A. Kar, “Local light field fusion: Practical view synthesis with prescriptive sampling guidelines,” ACM Transactions on Graphics, 38(4):29:1-29:14, 2019.

[16] T. Zhou, S. Tulsiani, W. Sun, J Malik, and A. A. Efros, “View synthesis by appearance flow,” In Proc. of European Conference on Computer Vision, pages 286-301, 2016.

[17] H. Schilling, M. Diebold, C. Rother, and B. Jhne, “Trust your model: Light field depth estimation with inline occlusion handling,” In Proc. of IEEE Conference on Computer Vision and Pattern Recognition, pages 4530-4538, 2018

[18] C. Shin, H. ‐G. Jeon, Y. Yoon, I. S. Kweon, and S. J. Kim, “Epinet: A fully-convolutional neural network using epipolar geometry for depth from light field images,” In Proc. of IEEE Conference on Computer Vision and Pattern Recognition, pages 4748-4757, 2018.

[19] Williem, and I. K. Park, “Robust light field depth estimation for noisy scene with occlusion,” In Proc. of IEEE Conference on Computer Vision and Pattern Recognition, pages 4396-4404, 2016.

[20] Williem, I. K. Park, and K. M. Lee, “Robust light field depth estimation using occlusion-noise aware data costs,” IEEE Transactions on Pattern Analysis and Machine Intelligence, (10):2484-2497, 2018.

[21] R. Ng, M. Levoy, M. Brdif, G. Duval, M. Horowitz, and P. Hanrahan, “Light field photography with a hand-held plenoptic camera,” Computer Science Technical Report CSTR, 2(11):1-11, 2005.

저 자 소 개

배 규 호

- 2018년 2월 : 인하대학교 정보통신공학과 학사
- 2018년 3월 ~ 현재 : 인하대학교 정보통신공학과 석사과정
- ORCID : https://orcid.org/0000-0003-1239-8522
- 주관심분야 : 컴퓨터비전 및 그래픽스 (라이트필드, 증강현실), deep learning

Andre Ivan

- 2017년 2월 : B.S. in Department of Computer Science, Bina Nusantara University
- 2019년 8월 : 인하대학교 정보통신공학과 석사
- ORCID : https://orcid.org/0000-0003-0365-3042
- 주관심분야 : 컴퓨터비전 및 그래픽스 (스테레오 비전, 라이트필드, 증강현실), deep learning, 모바일 GPGPU

박 인 규

- 1995년 2월 : 서울대학교 제어계측공학과 학사
- 1997년 2월 : 서울대학교 제어계측공학과 석사
- 2001년 8월 : 서울대학교 전기컴퓨터공학부 박사
- 2001년 9월 ~ 2004년 2월 : 삼성종합기술원 멀티미디어랩 전문연구원
- 2007년 1월 ~ 2008년 2월 : Mitsubishi Electric Research Laboratories (MERL) 방문연구원
- 2014년 9월 ~ 2015년 8월 : MIT Media Lab 방문부교수
- 2018년 7월 ~ 2019년 6월 : University of California, San Diego (UCSD) 방문학자
- 2004년 3월 ~ 현재 : 인하대학교 정보통신공학과 교수
- ORCID : https://orcid.org/0000-0003-4774-7841
- 주관심분야 : 컴퓨터비전 및 그래픽스, deep learning, GPGPU