2024 여름 세미나

3D Cinemagraphy and Human Image Animation from a Single Image

Sogang University
Vision & Display Systems Lab, Dept. of Electronic Engineering

Presented By
Min Seok Kang

Outline

- Background
  - Diffusion Models
  - Latent Diffusion Models – Stable Diffusion

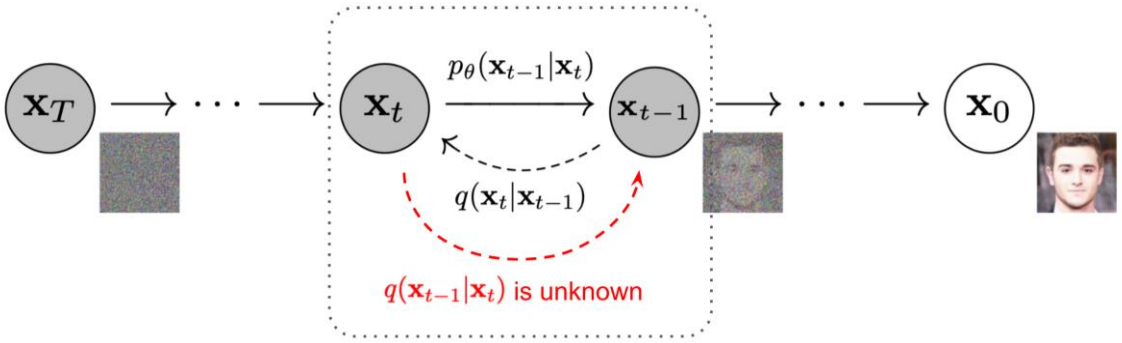
- ZHU, Shenhao, et al. **“Champ: Controllable and consistent human image animation with 3d parametric guidance.”** arXiv preprint arXiv:2403.14781, 2024.
- LI, Xingyi, et al. **“3d cinemagraphy from a single image.”** In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2023.

Background

- Diffusion Models
  - Gaussian Noise를 반복적인 denoising 과정을 거쳐서 학습된 data의 분포(image)로 변환하는 생성 모델
  - Conditional diffusion models
    - Diffusion model을 class label/text/저해상도 image로 conditioning 가능
  - Diffusion model $\hat{x}_\theta$는 아래와 같은 denoising objective로 학습

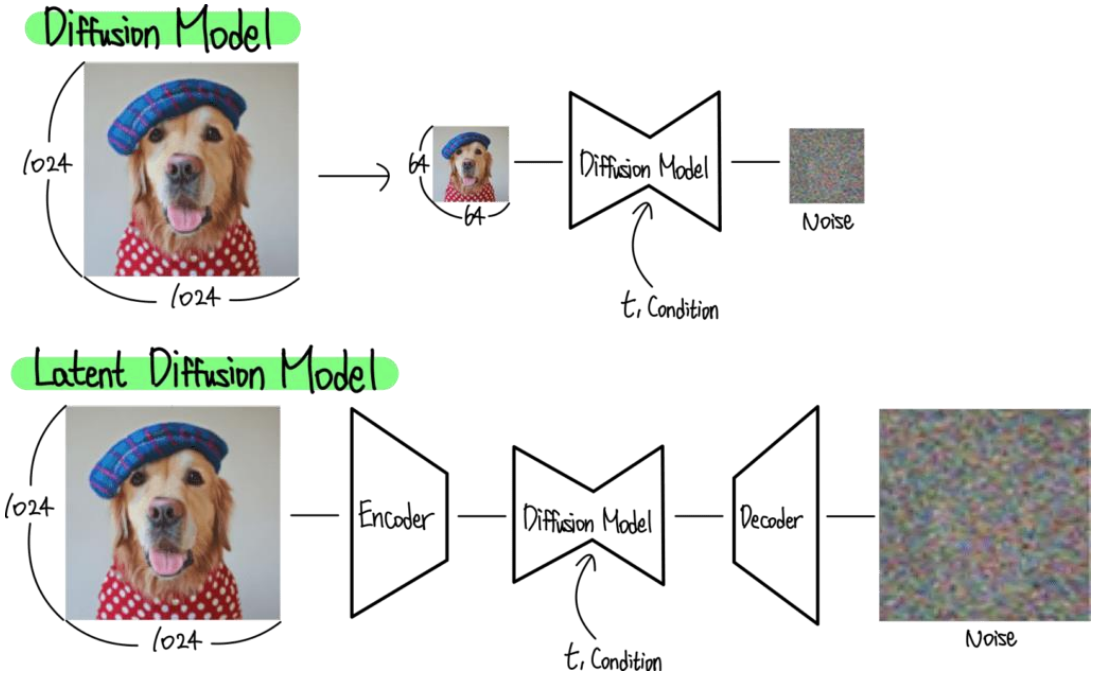
$$\mathbb{E}_{\mathbf{x},\mathbf{c},\boldsymbol{\epsilon},t}\left[w_t\|\hat{\mathbf{x}}_\theta(\alpha_t\mathbf{x}+\sigma_t\boldsymbol{\epsilon},\mathbf{c})-\mathbf{x}\|_2^2\right]$$

    - (x, c): data-condition pair, $t \sim U([0,1]), \epsilon \sim N(0, I)$(Gaussian Noise)
  - 직관적으로 diffusion model은 noise가 있는 $z_t := \alpha_t x + \sigma_t \epsilon$를 x로 denoising하는 것
    - 이 식을 reparameterization trick을 이용하여 ϵ-space에서 ϵ_θ에 대해 squared error loss를 적용
    - 원본 image 자체를 예측하는 문제에서 timestep t에서 t-1로 갈 때 제거할 noise를 예측하는 문제로 전환



Background

- Latent Diffusion Models – Stable Diffusion



- 기존 Diffusion Model은 픽셀 값을 직접 예측하고 반복적인 denoising을 통해 이미지를 생성
  - Pixel space 상에서 수행하기 때문에 엄청난 연산량을 요구함
- Pixel 값을 직접 예측하지 않고 Auto Encoder를 사용하여 압축된 latent embedding을 예측하도록 변경
  - Latent space 상에서 수행하여 더 적은 연산량과 빠른 속도를 얻을 수 있음

Champ: Controllable and Consistent Human Image Animation with 3D Parametric Guidance

CVPR 2024

Abstract

- 현재의 인공지능 기술에서 human image animation을 개선하기 위해 3D human parametric model을 활용한 새로운 방법론을 제안
- 인체의 shape 정렬과 motion 안내를 향상시키기 위해 Latent Diffusion Model의 framework 내에서 SMPL(Skinned Multi-Person Linear) model을 사용
  - SMPL model을 사용하여 신체 shape과 pose의 통합된 표현을 확립함으로써 source video로부터 정교한 인체 기하학 및 동작 특성을 정확하게 포착

Introduction

- Diffusion Model의 발전
  - 최근 Latent Diffusion 모델의 발전으로 image animation 분야가 크게 진전됨
    - 가상 현실 경험, interactive storytelling, 디지털 콘텐츠 제작에서 활용
- Human image animation
  - Skeleton, semantic maps, dense motion flows 등 인간 고유의 motion guide를 사용함
  - GAN과 Diffusion Model 기반 접근 방식이 주로 사용됨

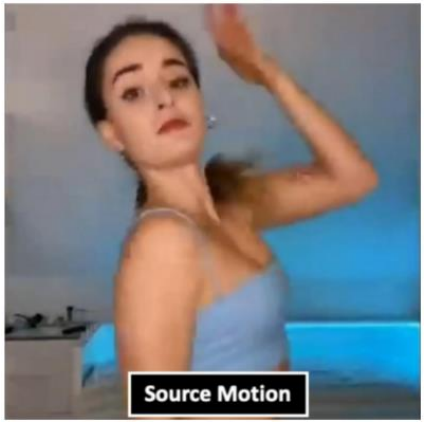
Introduction

- 기존 방법의 한계
  - GAN 기반 접근 방식
    - Warping을 사용하여 reference image를 입력 motion에 맞춰 공간적으로 변환함
    - 생성된 영상의 시각적으로 불완전한 영역을 채우고 개선하려고 하지만, motion 적용에 있어 큰 변화를 효과적으로 처리하지 못함
    - 비현실적인 visual artifact와 시간적 일관성 문제가 발생함
  - Diffusion Model 기반 접근 방식
    - Reference image와 다양한 동적인 요소를 외형과 motion의 수준에서 model의 condition으로 활용함
    - CLIP encoding된 visual feature와 Diffusion Model을 결합하여 일반화 문제를 해결함

Introduction

- 제안한 접근 방식
  - SMPL model 사용
    - Reference image의 3D 기하학을 인코딩하고 source video에서 human motion을 추출함
    - 낮은 차원의 파라미터 공간을 사용하여 shape과 pose를 통합하여 표현함
    - 이를 통해 reference image와 source video의 SMPL 기반 motion sequence 간의 기하학적인 일치가 가능함

Reference Image



Source Video Motion

Output Frame

Key Contributions

1. SMPL model과의 통합
    1. Reference image와 source video의 3D 기하학적인 일치를 통한 human animation 생성
2. Multi-Layer Motion Fusion (MLMF)
    1. Self-attention mechanism을 사용하여 shape 및 motion latent representation을 융합
3. 실험적 평가
    1. Benchmark dataset을 사용한 실험에서 고품질의 human animation 생성 능력 입증
    2. In-the-wild dataset에 대해서도 우수한 일반화 능력 확인

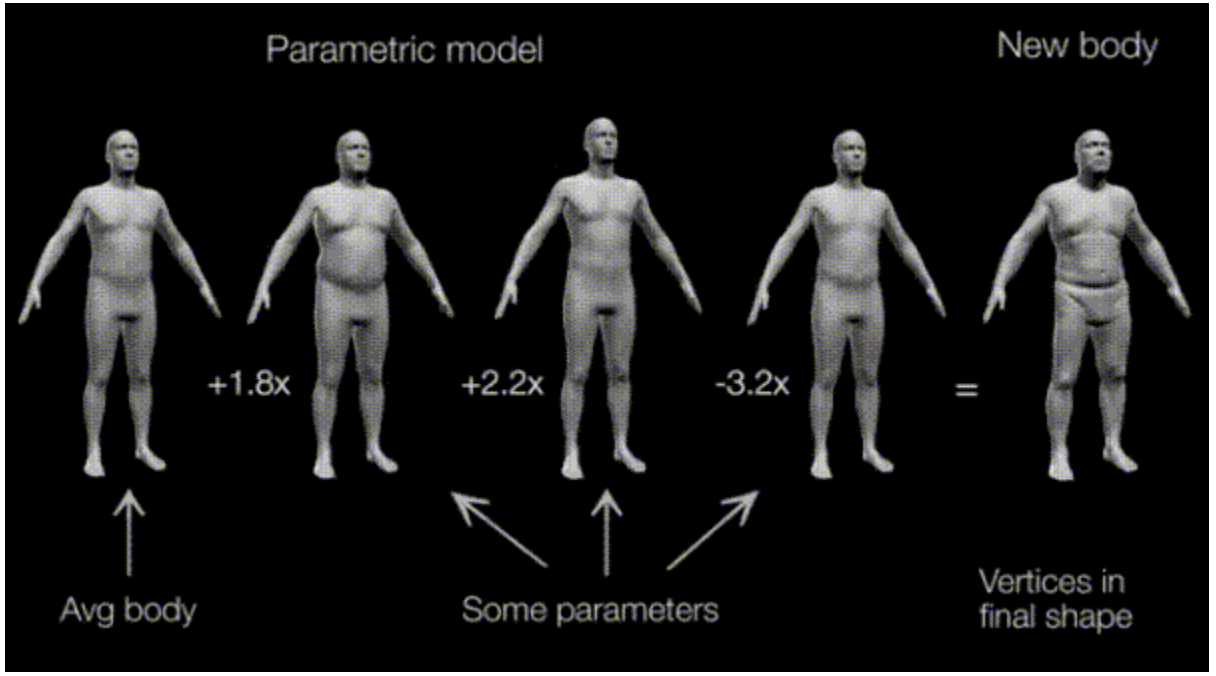
1) LOPER, Matthew, et al. SMPL: A skinned multi-person linear model. In: Seminal Graphics Papers: Pushing the Boundaries, Volume 2. 2023. p. 851-866.

Related Work

- SMPL: A skinned multi-person linear model[1)]
  - 인간의 신체 형태와 자세에 따른 형태 변화를 더 정확하게 표현할 수 있는 새로운 모델을 제시함
    - 다양한 신체의 형태와 자연스러운 자세 변화를 현실적으로 나타낼 수 있음
    - 기존의 그래픽 파이프라인과 호환 가능함
  - SMPL은 데이터에서 학습된 parameter를 사용하여 신체의 평균 템플릿, blending 가중치, 자세에 의존하는 blending 등을 포함함
    - Pose($\theta \in \mathbb{R}^{24\times3\times3}$)와 shape($\beta \in \mathbb{R}^{10}$)를 나타내는 저차원 parameter를 기반으로 함
    - 자세에 의존하는 blending은 자세 회전 행렬의 요소들의 linear function으로 표현함
    - 다양한 사람들의 다양한 자세에서 정렬된 3D mesh data를 통해 전체 모델을 훈련함
  - 입력 reference image와 source video로부터 3D mesh를 출력하여 image animation 제작에 사용함

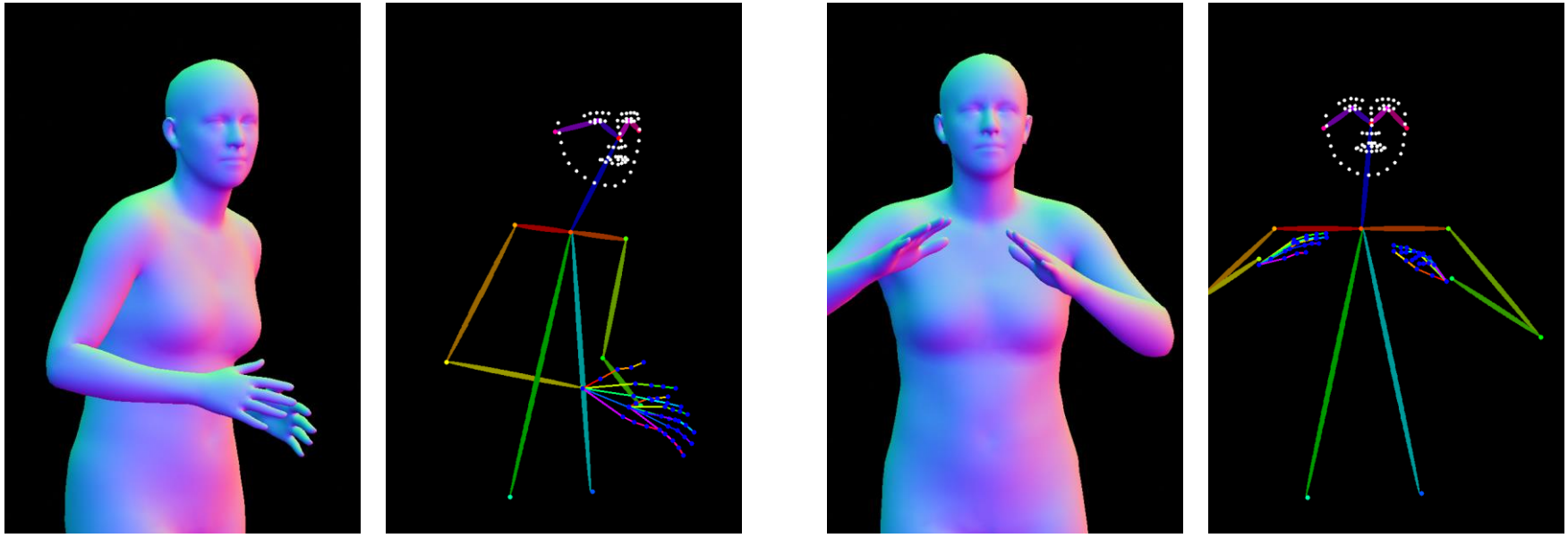
Related Work

- SMPL: A skinned multi-person linear model[1)]
  - 인간의 외형을 다양하고 자세하게 표현하는 방법
    - 인간의 신체 평균 템플릿이 존재함 (그림의 가장 왼쪽)
    - 3개의 blendshape basis가 정의되어 각각의 linear combination으로 새로운 체형을 표현함



Related Work

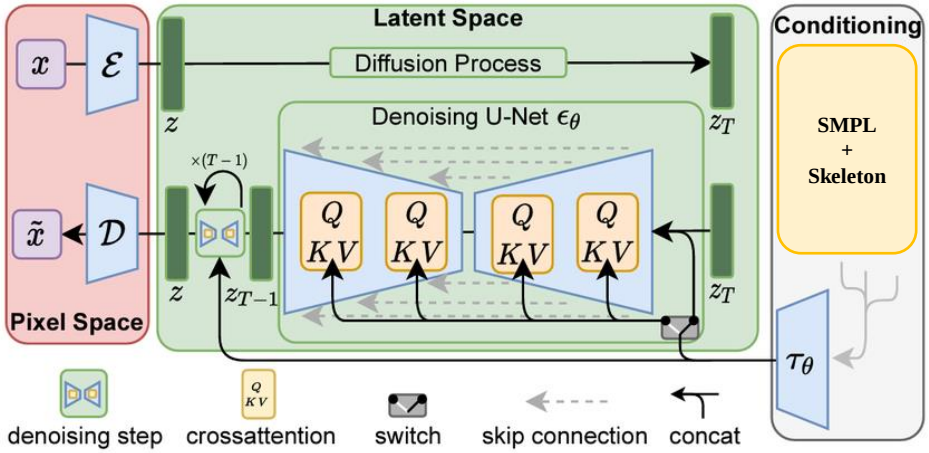
- Skeleton 추출을 위한 DWPose[1)]
  - ICCV 2023에 등재된 논문으로, 매우 정확하고 표현력 있는 skeleton을 제공함
  - Human motion의 skeleton을 diffusion model의 생성 과정에 추가함으로써, 더 높은 품질의 image animation을 제작할 수 있음
    - Source video의 normal map으로부터 skeleton을 추출함



Normal map | Skeleton output | Normal map | Skeleton output

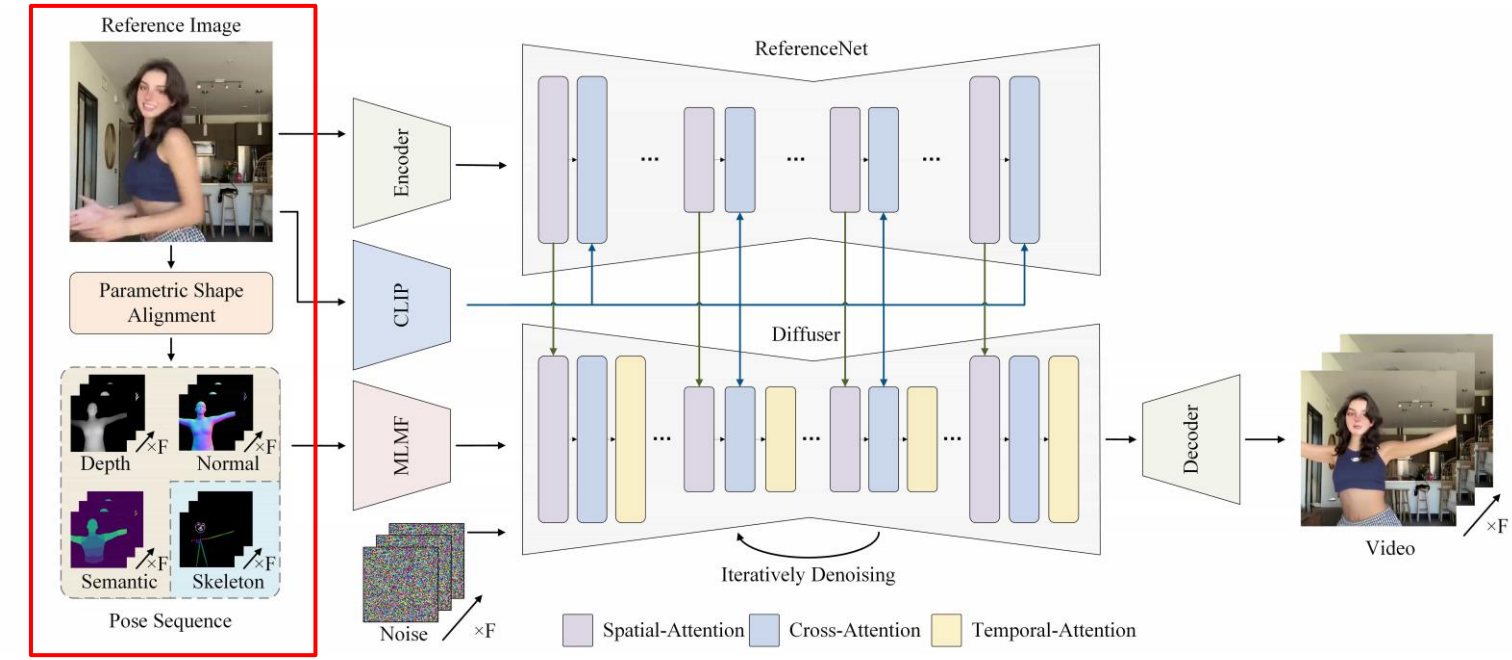
Method

- 기본적인 애니메이션 생성 방식은 Latent Diffusion Model framework를 사용함
  - Latent Diffusion Model과 동일한 3D U-Net 구조를 이용해 제작
    - Latent Diffusion Model(LDM)은 diffusion과 denoising의 두 가지 확률적 과정을 latent space에 통합하는 접근 방식임
    - 초기에는 VAE(Variational Autoencoder)를 사용하여 입력 이미지를 저차원 feature space로 인코딩함
      - 입력 이미지를 latent representation $\boldsymbol{z}_0 = \mathcal{E}(\mathrm{I})$로 변환함
    - Diffusion 과정은 z_0에 Markov process를 적용하여 다양한 noisy latent representation을 생성함
      - $\boldsymbol{z}_t = \sqrt{\overline{\alpha}_t}\boldsymbol{z}_0 + \sqrt{1-\overline{\alpha}_t}\boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$
    - Denoising 과정에서는 각 timestep t에서 $\boldsymbol{z}_t \rightarrow \boldsymbol{z}_{t-1}$로의 노이즈 $\boldsymbol{\epsilon}_\theta(\boldsymbol{z_t}, t, \boldsymbol{c})$을 예측함
    - 훈련 후 모델은 초기 상태 $\boldsymbol{z}_t \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$에서 $\boldsymbol{z}_0$로 점진적으로 denoising할 수 있음
    - Denoising된 $\boldsymbol{z}_0$는 VAE decoder $\mathcal{D}(\cdot)$를 사용하여 image space로 다시 디코딩됨



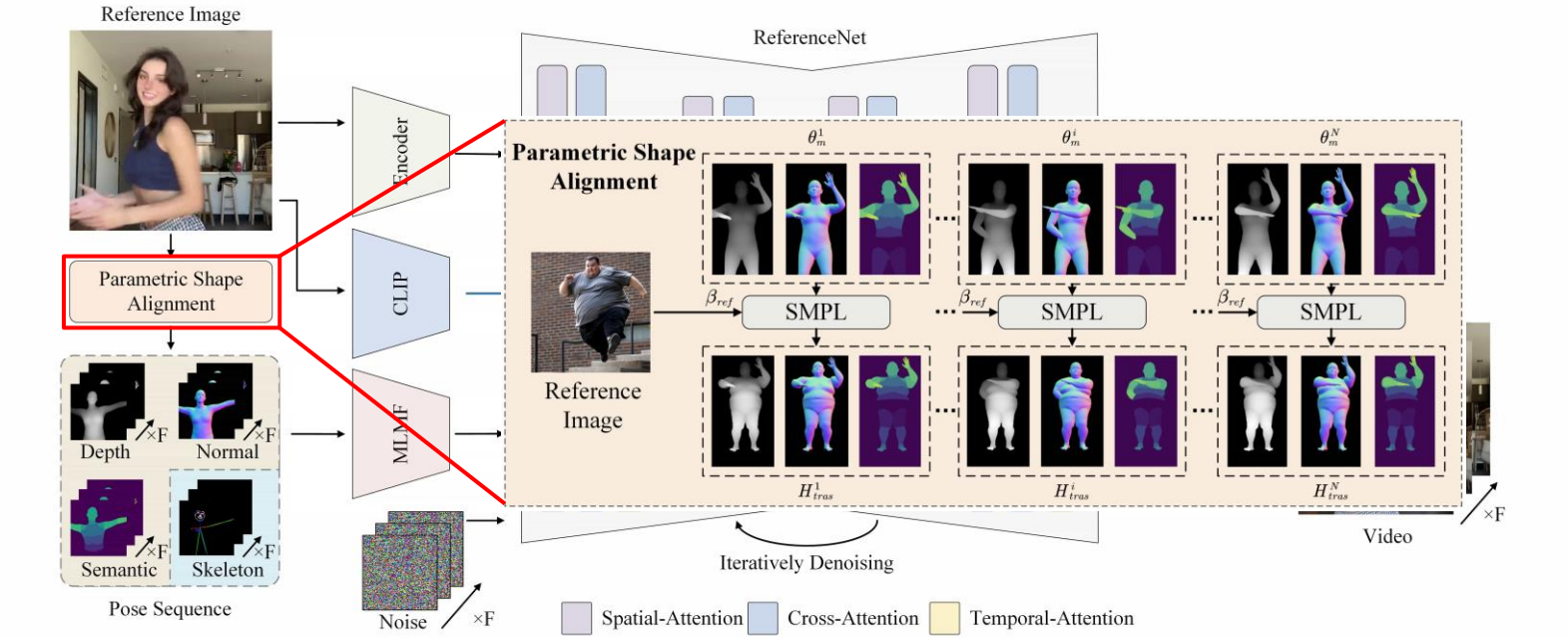
Method

- Multi-layer motion condition – SMPL to guidance condition
  - Reference human image I_{ref}와 motion video sequence frame $I^{1:N}$을 입력으로 받으면 기존 framework인 4D-Humans[1]를 사용하여 3D human parametric SMPL model H_{ref}와 $H_m^{1:N}$을 얻음
  - SMPL mesh를 렌더링하여 2D depth map, normal map, semantic map을 얻음
  - DWPose[2]를 이용해 normal map으로부터 skeleton을 얻음



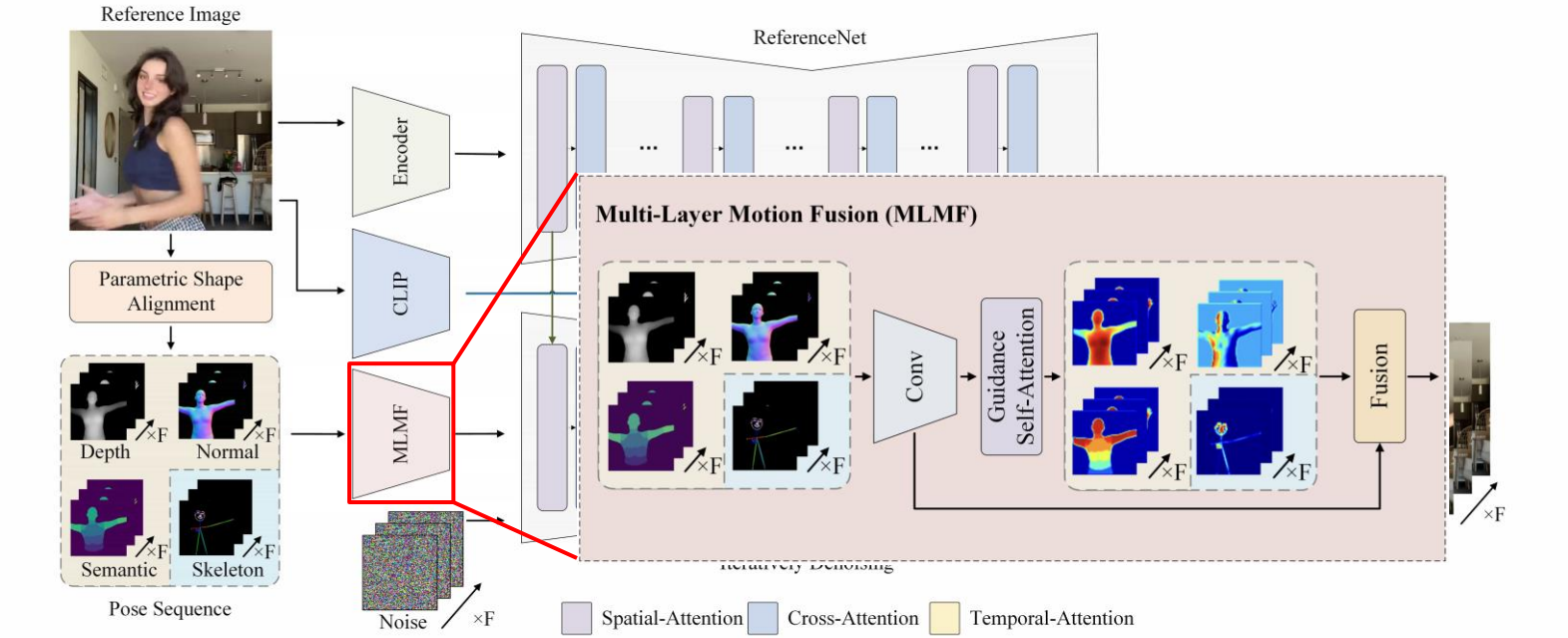
Method

- Multi-layer motion condition – Parametric shape alignment
  - Parametric human SMPL model을 사용하면 reference image의 사람과 motion sequence 사이에서 shape와 pose를 모두 정렬하기 쉬움
  - Reference image I_{ref}에서 맞춘 SMPL model H_{ref}과 N-frame motion video $I^{1:N}$의 SMPL sequence $H_m^{1:N}$가 주어지면, H_{ref}의 shape β_{ref}를 $H_m^{1:N}$의 pose sequence $\theta_m^{1:N}$에 맞춤
    - 이 과정을 통해 source video의 human shape가 reference image의 human shape와 동일해짐



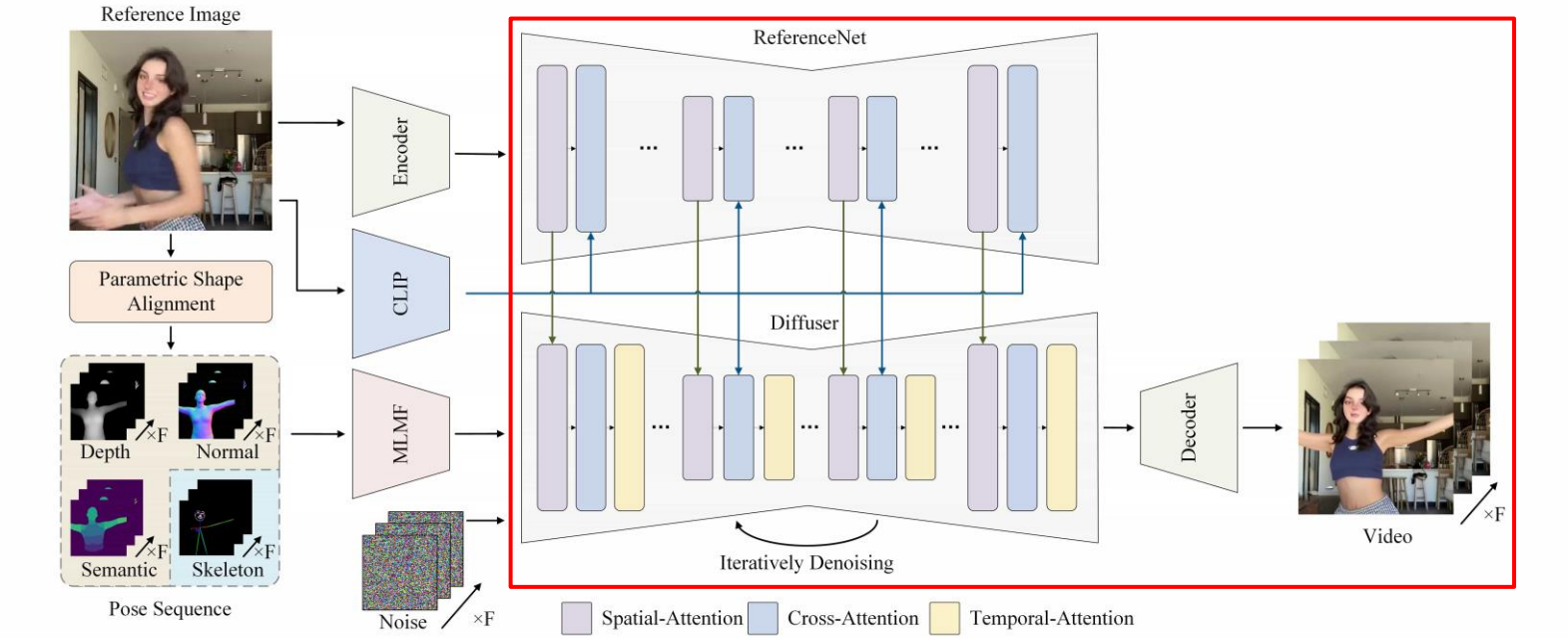
Method

- Multi-layer motion guidance – Multi-Layer Motion Fusion (MLMF)
  - Shape alignment 후 depth map, normal map, semantic map을 렌더링하고 skeleton을 추가로 제공함
  - 각 종류의 guidance마다 convolutional layer를 통해 feature를 추출하고 self-attention 모듈을 사용하여 더 정확한 특징을 파악하여 fusion을 수행함
    - 하나의 최종 guidance feature가 완성되어 denoising U-Net의 condition으로 사용됨



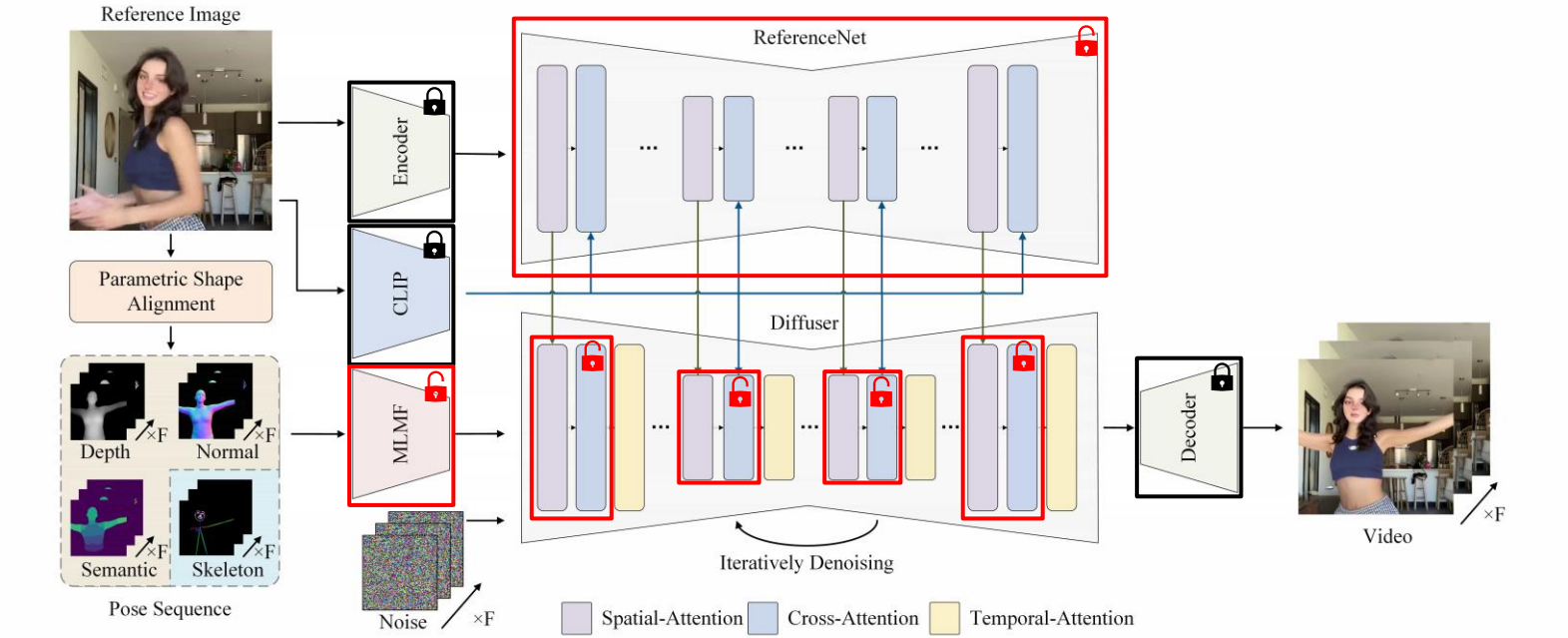
Method

- ReferenceNet 및 네트워크 구조
  - ReferenceNet을 통해 생성될 비디오의 캐릭터와 배경을 reference image와 일관되게 유지함
    - ReferenceNet은 denoising U-Net과 동일한 구조임
    - VAE와 CLIP 인코더를 통해 인코딩된 reference image embedding을 입력으로 받아서 생성된 비디오에서 일관된 시각적 품질을 유지함
  - 자연스러운 비디오를 위해 시간 축으로의 temporal attention 모듈을 추가함



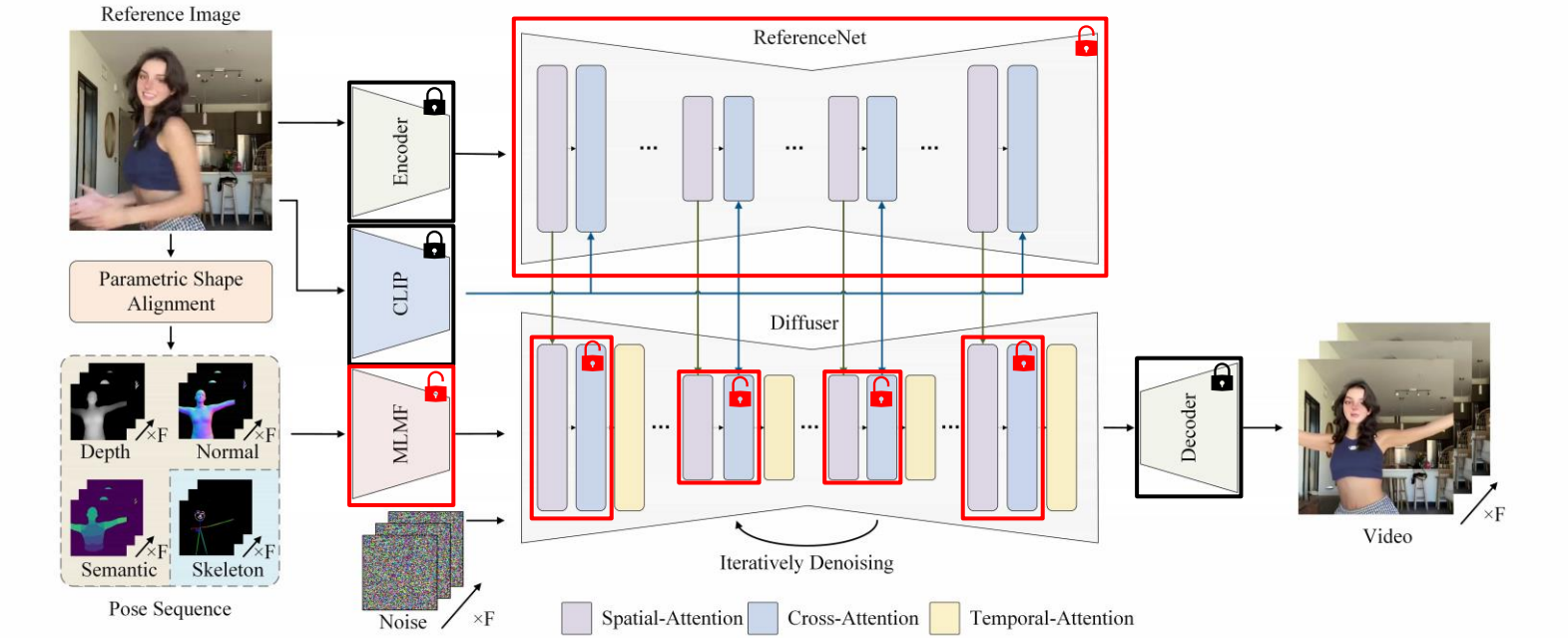
Method

- Two-stage training – 1st stage
  - 첫 번째 단계에서는 이미지만을 대상으로 학습을 진행하고 motion module은 제외함
    - VAE encoder와 decoder, CLIP image encoder의 가중치를 고정함
    - Guidance encoder, denoising U-Net, ReferenceNet은 학습 중 업데이트함



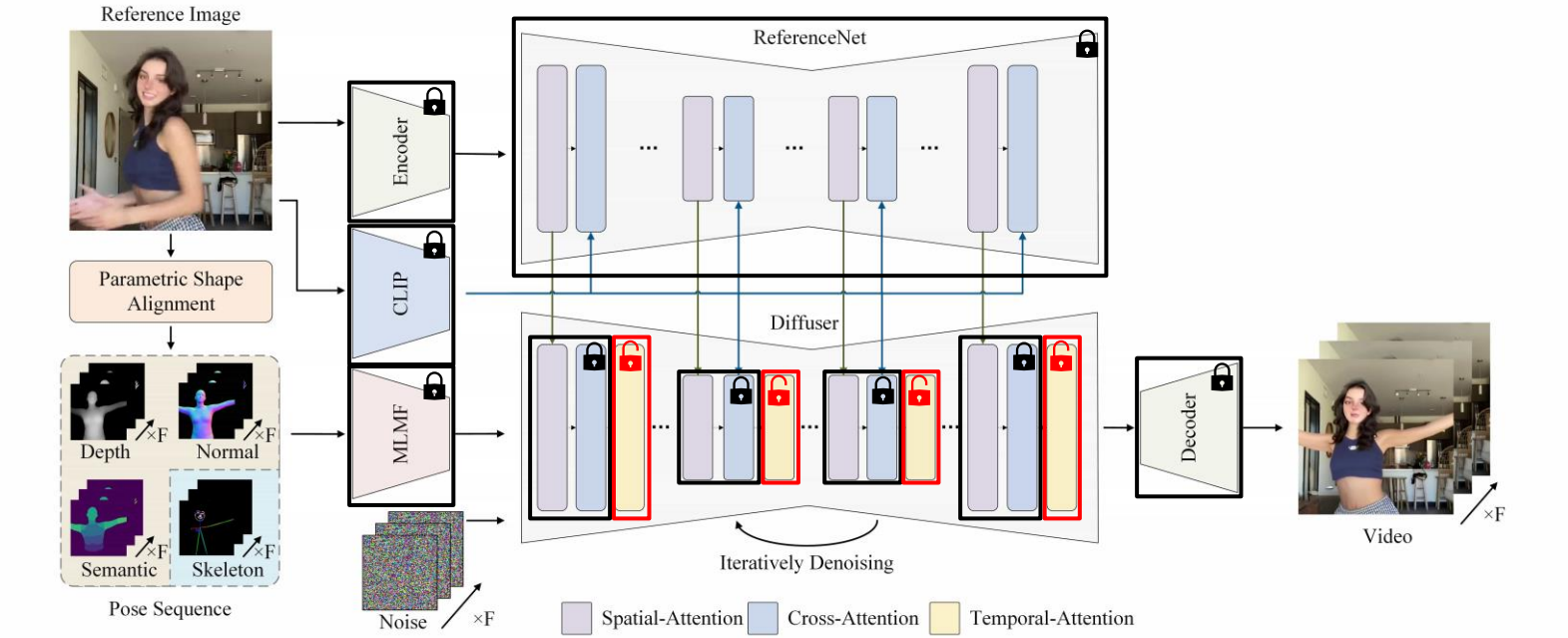
Method

- Two-stage training – 1st stage
  - 첫 번째 단계에서는 이미지만을 대상으로 학습을 진행하고 motion module은 제외함
    - Human video에서 랜덤으로 프레임을 선택하여 reference image로 사용하고, 동일 video에서 또 다른 이미지를 target image로 선택함
    - Target image에서 추출한 multi-layer guidance를 Guidance network에 입력하여 고품질의 animation image를 생성하도록 훈련함



Method

- Two-stage training – 2nd stage
  - 두 번째 단계에서는 motion module을 도입하여 모델의 시간적 일관성과 유연성을 강화함
    - Motion module(Temporal-Attention)을 도입하고 AnimateDiff[1]의 기존 weight로 초기화함
    - 24 frame으로 구성된 video segment를 입력 데이터로 사용함
    - 첫 번째 단계에서 학습된 Guidance encoder, denoising U-Net, ReferenceNet는 고정함



Experimental Results

- Dataset
  - 약 5,000개의 고화질 human video를 온라인에서 수집, 총 100만 frame 이상을 포함함
    - Bilibili: 2,540개 video
    - Kuaishou: 920개 video
    - Tiktok & Youtube: 1,438개 video
    - Xiaohongshu: 430개 video
  - 다양한 연령, 인종, 성별의 인물을 포함하며, 전신, 반신, 클로즈업 샷을 포함한 다양한 실내 및 실외 배경에서 촬영함
  - 다양한 춤 스타일을 보여주는 댄서들의 영상을 포함하여 다양한 의상과 움직임을 분석함
  - Test set으로는 Image Animation 분야의 기존 벤치마크와 일치시키기 위해 MagicAnimate[1]에서 사용된 동일한 test set를 TikTok 평가에 사용함

Experimental Results

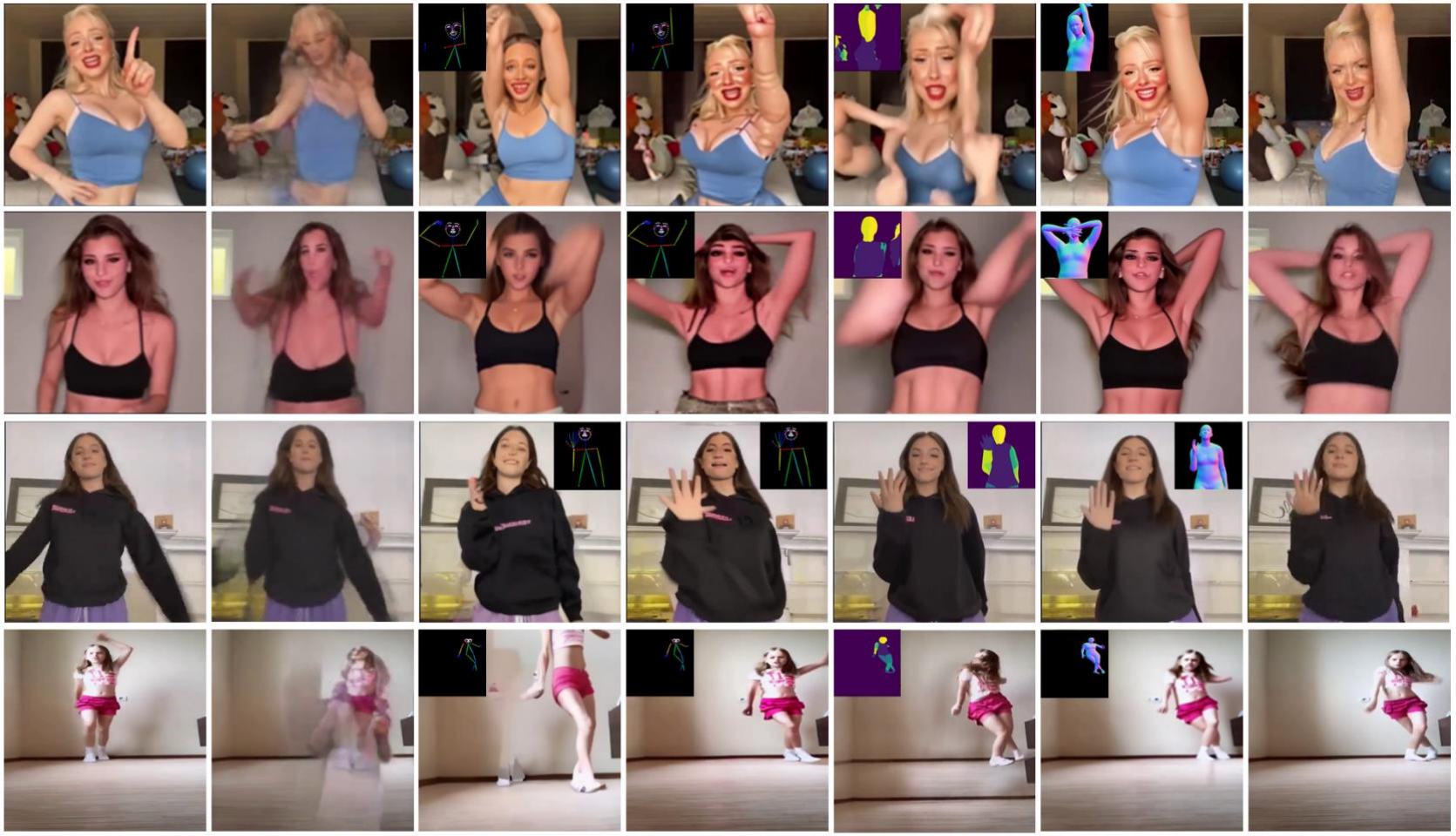
- Qualitative Results

Experimental Results

- Qualitative Results

Experimental Results

- Qualitative Results



Experimental Results

- Quantitative Results
  - TikTok dataset에 대해서 기존 모델들에 비해 더 좋은 성능을 보임
    - LPIPS (Learned Perceptual Image Patch Similarity): 이미지 패치를 pre-trained network에 입력하고, 중간 layer에서 추출된 feature vector 간의 유사성을 측정함
    - FID-VID (Frechet Inception Distance for Videos): 생성된 video와 실제 video의 Inception network에서 추출된 feature vector의 평균과 공분산 행렬을 비교하여 Frechet 거리를 계산함
    - FVD (Frechet Video Distance): Video의 프레임을 단일 이미지로 취급하지 않고, 전체 video sequence의 feature vector를 추출하여 Frechet 거리를 계산함

Method	L1 ↓	PSNR ↑	SSIM ↑	LPIPS ↓	FID-VID ↓	FVD ↓
MRAA	3.21E-04	29.39	0.672	0.296	54.47	284.82
DisCo	3.78E-04	29.03	0.668	0.292	59.90	292.80
MagicAnimate	3.13E-04	29.16	0.714	0.239	21.75	179.07
Animate Anyone	-	29.56	0.718	0.285	-	171.9
Ours	3.02E-04	29.84	0.773	0.235	26.14	170.20
Ours*	**2.94E-04**	**29.91**	**0.802**	**0.234**	**21.07**	**160.82**

(*): TikTok dataset에 대해서만 fine-tuning을 진행한 모델

Experimental Results

- Quantitative Results
  - 서로 다른 motion guidance에 따른 성능의 영향 비교
    - w/o. SMPL: 오직 skeleton map만 사용하였을 때
    - w/o. geo.: geometric information에 해당하는 depth map과 normal map을 제외하였을 때
    - w/o. skl.: SMPL에서 제공된 depth map, normal map, semantic map만 사용하였을 때
  - SMPL-driven guidance (depth, normal, semantic map)과 skeleton map을 모두 함께 사용하였을 때 가장 좋은 성능을 보임

Method	L1 ↓	PSNR ↑	SSIM ↑	LPIPS ↓	FID-VID ↓	FVD ↓
Ours (w/o. SMPL)	4.83E-04	28.57	0.672	0.296	30.06	192.34
Ours (w/o. geo.)	4.06E-04	28.78	0.714	0.276	29.75	189.07
Ours (w/o. skl.)	3.76E-04	29.05	0.724	0.264	34.12	184.24
Ours	3.02E-04	29.84	0.773	0.235	26.14	170.20

3D cinemagraphy from a single image

CVPR 2023

Abstract

- 단일 이미지로부터 3D cinemagraphy를 생성하는 새로운 기술을 제안
  - 3D cinemagraphy란, 정지된 이미지로부터 시각적인 콘텐츠 animation과 카메라 움직임을 포함하는 3D 효과 video를 생성하는 기술임
- 2D 이미지와 3D 사진 촬영 방법을 단순히 결합하면, 뚜렷한 artifact와 일관성 없는 animation이 발생함
- 이를 해결하기 위해 scene을 3D 공간에서 표현하고 animation화하는 새로운 방법을 제안

Introduction

- Cinemagraphy 발전의 배경
  - 스마트폰 카메라의 보급으로 온라인에 많은 사진이 업로드됨
  - YouTube와 TikTok 같은 비디오 공유 플랫폼의 인기가 상승함
  - 사람들은 정적인 이미지보다 동영상을 선호하게 됨
- 제안한 문제점
  - Cinemagraphs는 정적인 카메라를 기반으로 하기 때문에 3D 감각을 제공하지 못함
  - 기존 방법을 단순히 결합하면 visual artifact나 일관성 없는 animation이 발생함
- 제안한 목표
  - 단일 이미지로부터 현실감 있는 scene animation과 카메라 움직임을 포함한 3D cinemagraphy 구현

Introduction

- 제안한 방법론
  - Scene 표현
    - Feature-based Layered Depth Images (LDIs)로 scene을 표현함
    - LDIs를 feature point cloud로 변환함
  - Scene animation
    - 2D motion을 3D scene flow로 변환하여 animation화함
    - Depth 값 예측을 통해 2D motion을 3D로 전환
  - Hole 문제 해결
    - 3D symmetric animation 기술을 사용하여 point cloud를 양방향으로 이동함
    - 새로운 view를 합성하여 hole이 발생하는 문제를 해결함

Key Contributions

1. 새로운 task 제안
    1. 단일 이미지로부터 3D Cinemagraphs를 생성하는 새로운 task를 제안함
    2. Image animation과 novel view synthesis를 3D 공간에서 공동으로 해결하는 새로운 framework를 제안함
2. 3D symmetric animation 기술 설계
    1. Point가 전진하면서 발생하는 hole 문제를 해결하기 위한 3D symmetric animation 기술을 설계함
3. 유연한 framework
    1. 사용자 정의 mask와 flow hint를 이용해서 motion estimation을 보강함으로써 제어 가능한 animation을 구현 가능함

LI, Xingyi, et al. "3d cinemagraphy from a single image." In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2023.

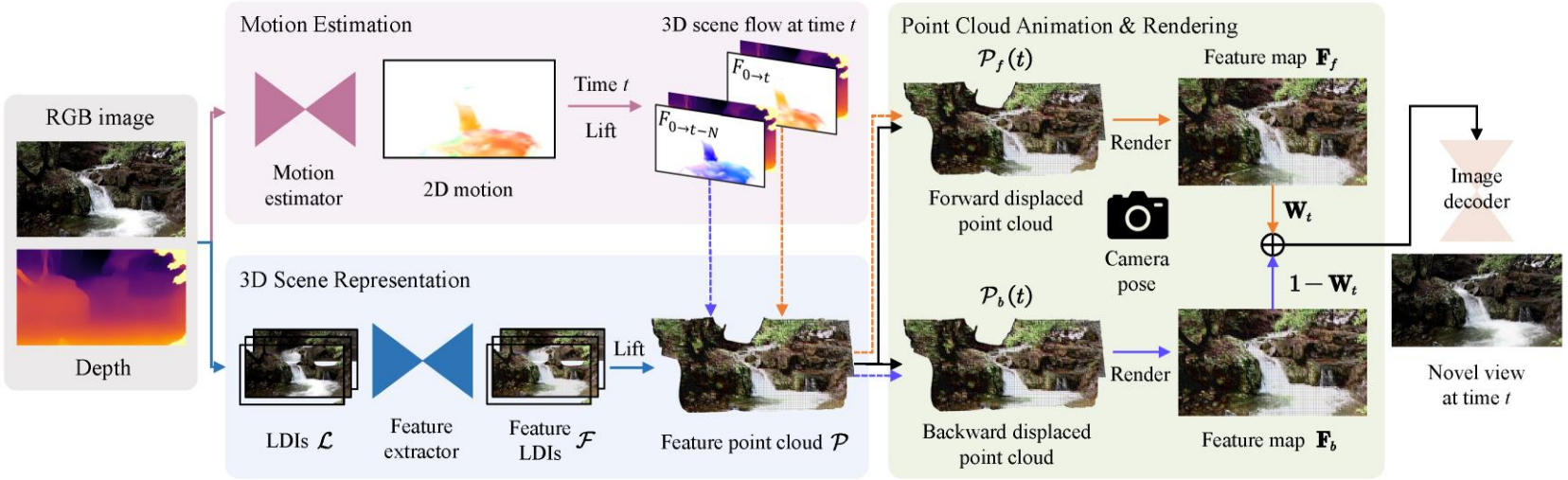
1) JHOU, Wei-Cih; CHENG, Wen-Huang. Animating still landscape photographs through cloud motion creation. *IEEE Transactions on Multimedia*, 2015.

Related Work

- Single-image animation
  - 다양한 방법들이 정지된 이미지를 animation화하는데 사용됨
  - 일부 연구는 특정 객체를 물리적인 simulation으로 animation화함
    - Animating Still Landscape Photographs Through Cloud Motion Creation (2015)[1)]
    - 일상적인 이미지에 대해서는 잘 일반화되지 않음
  - 다른 연구들은 reference video를 참고하여 정적인 객체에 motion을 부여함
    - 추가적인 video를 입력으로 넣어줘야 하므로, 해당 논문의 목표와 맞지 않음
  - 기존의 방법들은 2D space에서 작동하여 카메라 움직임을 생성할 수 없음
- Novel view synthesis
  - 2D 이미지와 해당 카메라 pose를 이용하여 새로운 카메라 view를 렌더링함
  - 최근에는 NeRF, 3D Gaussian Splatting 등이 높은 품질의 결과를 생성함
    - 하지만, 이러한 방법들은 dense views를 입력으로 가정함
  - 단일 이미지 입력을 다루는 방법들도 존재하지만, 일반적으로 정적인 장면을 가정함

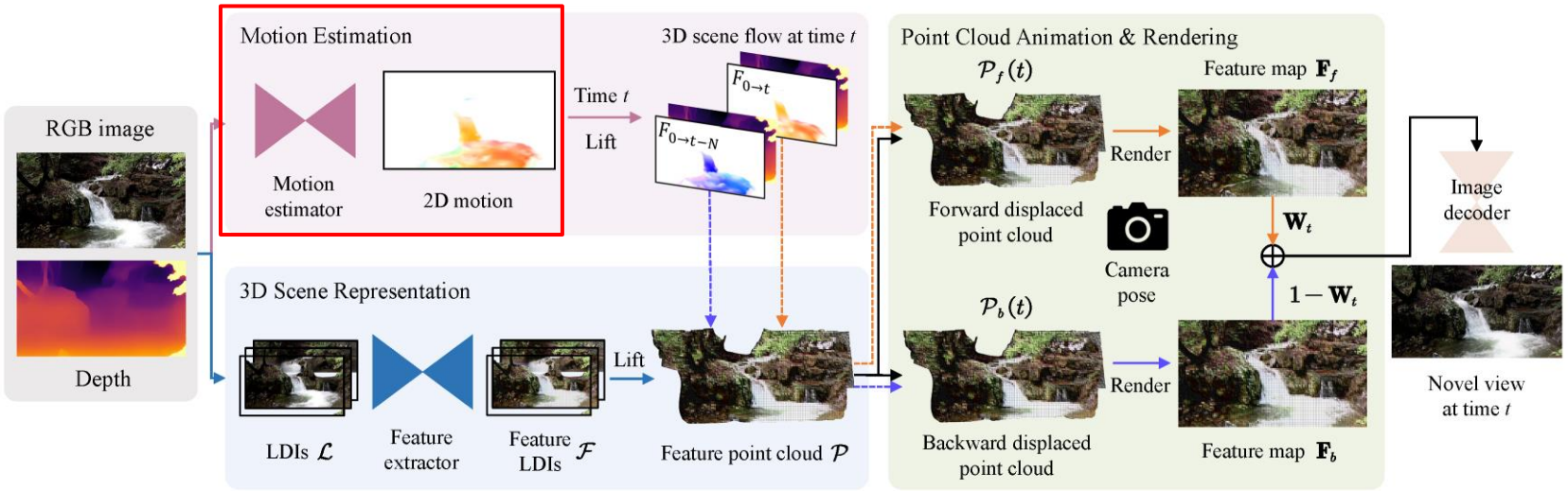
Method

- 전체적인 개요
  - 단일 정지 이미지에서 scene animation과 카메라 움직임을 동시에 가능하게 하는 것이 목표임
- Overall pipeline
  1. Motion field와 depth map 예측
  2. RGBD 입력을 여러 layer로 분리하여 feature LDI 생성
  3. Animation화 및 novel view synthesis 수행



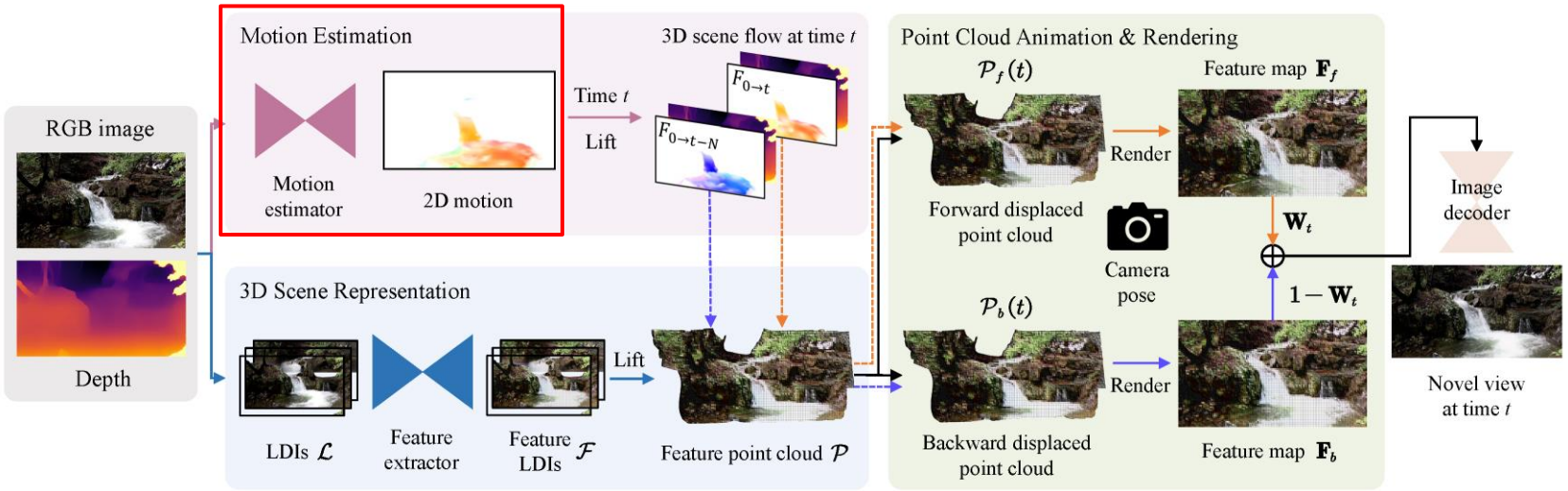
Method

- Motion Estimation
  - 기존 optical flow를 사용하려면 최소 한 쌍의 이미지가 필요하므로 단일 이미지에는 불가능함
  - 기존 방법론에 따라서 일정한 속도의 Eulerian flow field를 사용함
    - “Animating pictures with Eulerian motion fields[1)]” 논문에서 제안한 Eulerian flow field는 단일 RGB 이미지를 optical flow로 mapping해주는 motion estimator를 사용함
    - Frame t에서 frame $t+1$로의 optical flow map: $\boldsymbol{F_{t\rightarrow t+1}}(\cdot) = \boldsymbol{M}(\cdot)$
      - 여기서 M은 scene의 Eulerian flow field로, 각 pixel이 다음 frame에서 어떻게 움직일지를 알 수 있음



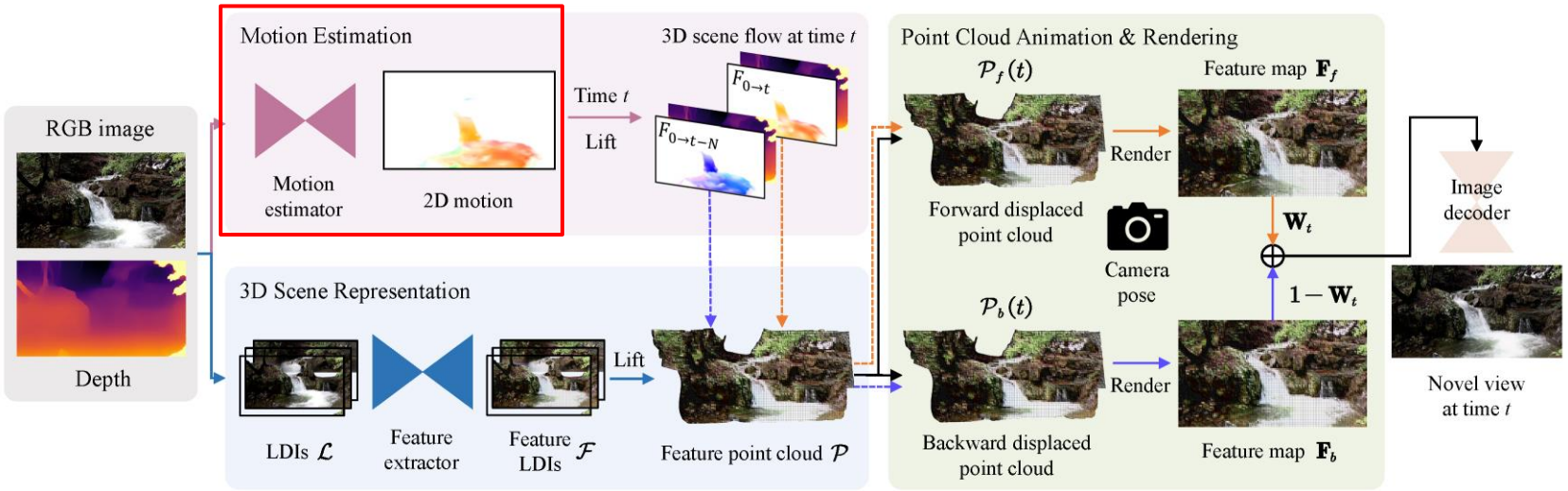
Method

- Motion Estimation
  - Euler integration을 통해 next frame을 얻을 수 있음
    - $x_{t+1} = x_t + M(x_t)$
      - x_t는 시점 t에서의 pixel들의 위치 좌표를 나타냄
  - 연속된 frame 사이의 optical flow가 동일하므로, 이 정보를 사용해서 displacement field를 추정함
    - Displacement field: 특정 시간 동안의 각 pixel의 이동 경로(flow)를 나타내는 field
    - $F_{0\to t}(x_0)$: 시점 0에서 t까지의 displacement field



Method

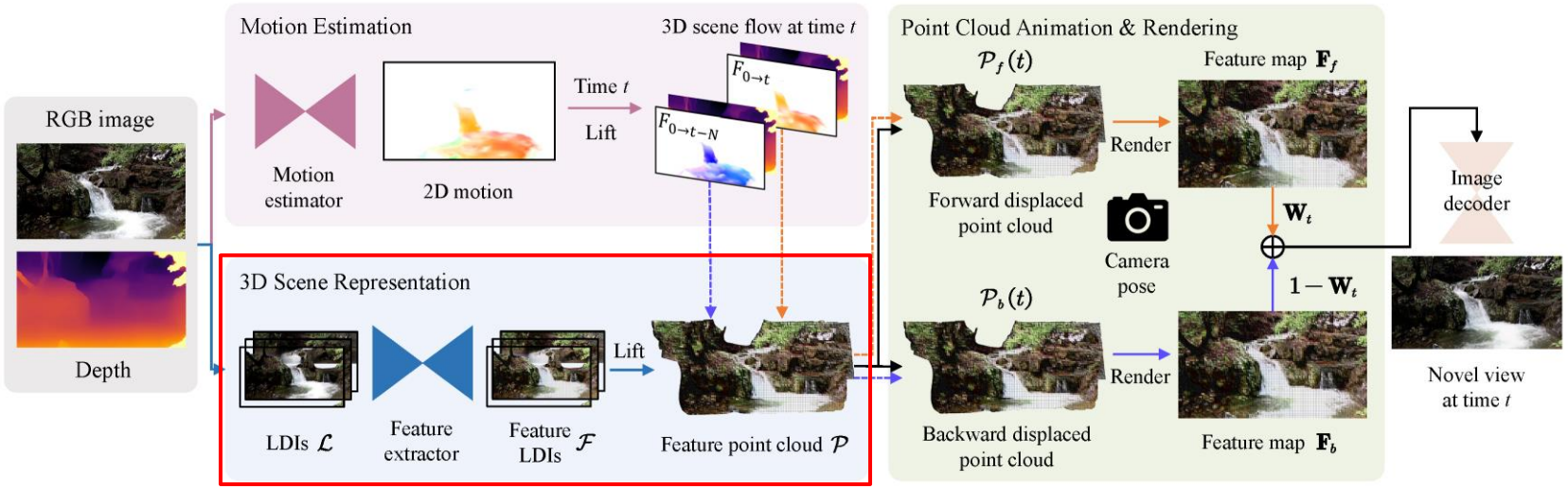
- Motion Estimation
  - $F_{0\to t}(x_0)$를 재귀적 적용을 통해 구하는 방법
    - $F_{0\to t}(x_0)$를 계산하기 위해 이전 시점 $t-1$까지의 displacement field $F_{0\to t-1}(x_0)$에 현재 frame의 optical flow $M\big(x_0 + F_{0\to t-1}(x_0)\big)$를 더할 수 있음
      - $\boldsymbol{F_{0\to t}(x_0) = F_{0\to t-1}(x_0) + M\big(x_0 + F_{0\to t-1}(x_0)\big)}$
    - 이 방법을 통해 초기 frame부터 특정 시점까지의 각 pixel의 이동 경로를 효율적으로 추적 가능함
      - 단일 이미지에서 animation을 생성하는 데 중요한 역할을 함



1) Rene Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), pages 12179–12188, 2021.

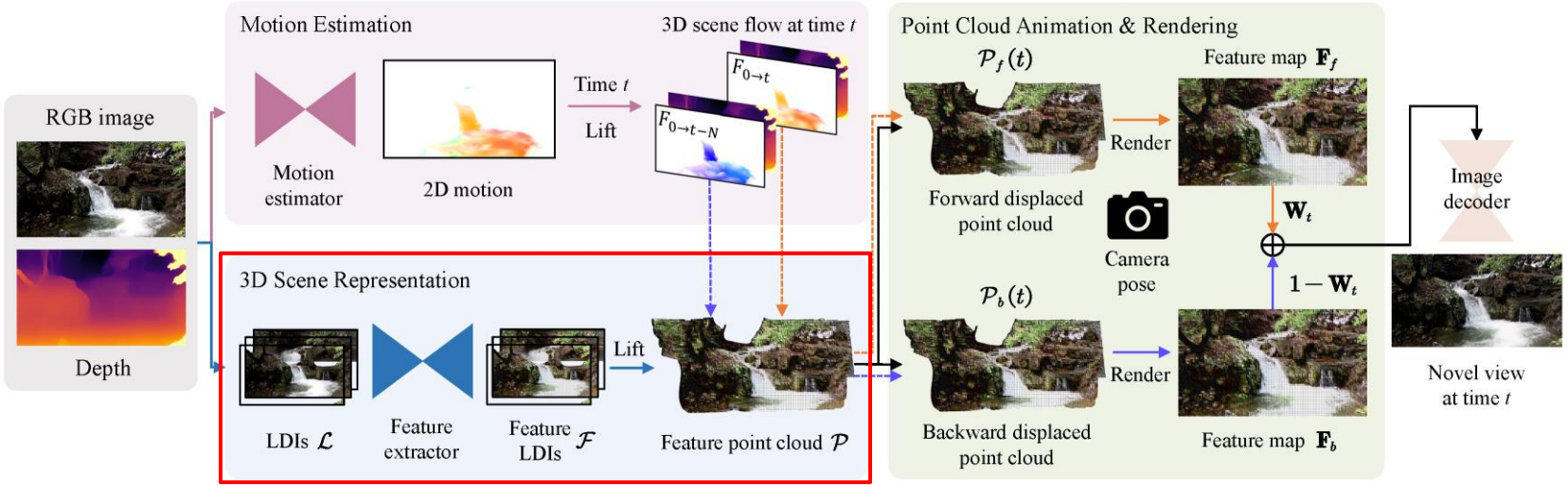
Method

- 3D scene representation
  - 이전의 단일 이미지 animation 방법론들은 대부분 2D space에서 작동해서 카메라 움직임 불가
    - 3D space로 전환하여 단일 이미지로부터 장면의 기하학적 구조를 추정함
  - Step 1. 깊이 추정
    - DPT(Depth Prediction Transformer)[1]를 사용하여 monocular depth estimation을 수행함
      - 논문 발표 기준 state-of-the-art monocular depth estimator임



Method

- 3D scene representation
  - Step 2. LDI(Layered Depth Image) 생성
    - 예측한 depth map을 포함한 RGBD 입력을 depth 불연속성에 따라 여러 layer로 분리함
    - Depth 범위를 여러 간격으로 나눈 후, 각 layer의 color 및 depth 정보를 포함한 LDI 생성
      - Layered depth image $\mathcal{L} = \{\boldsymbol{C}_l, \boldsymbol{D}_l\}_{l=1}^{L}$
    - LDI image의 각 layer의 가려진 영역을 3D Photo[1]의 pretrained inpainting model을 이용해 보완함



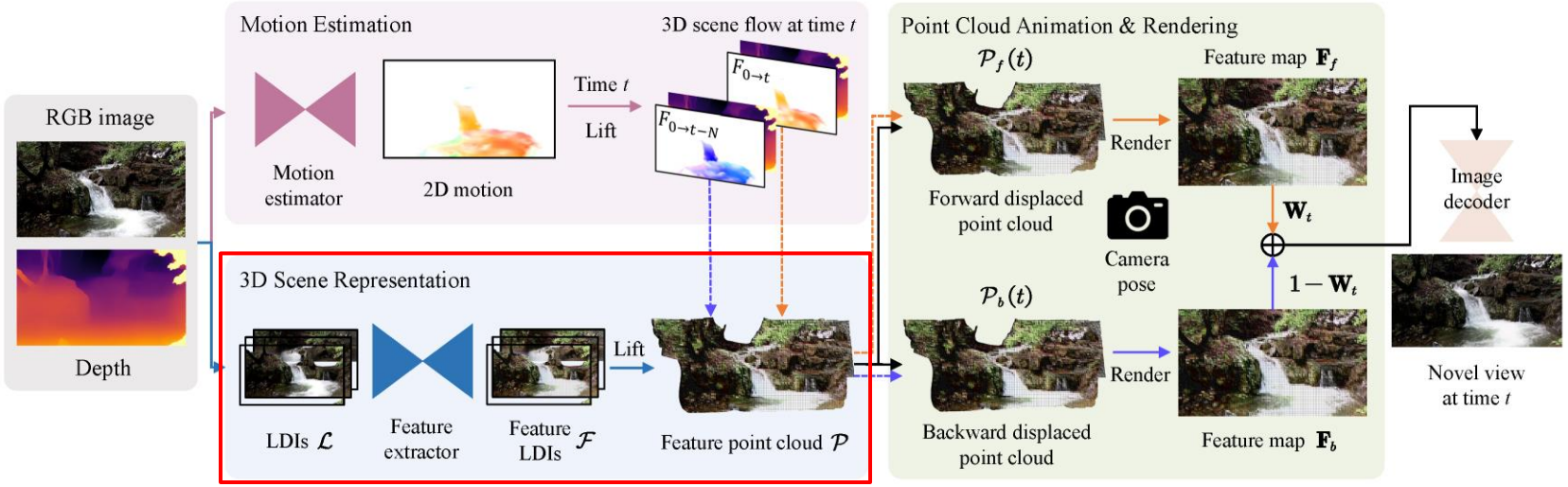
Method

- 3D scene representation
  - Step 3. Feature point cloud 생성
    - 2D feature extraction network를 사용하여 각 inpainted LDI color layer에 대해 2D feature map을 얻음
    - 각 layer를 대응하는 depth 값으로 3D space로 전환하여 feature point cloud $\mathcal{P}$를 생성함

$$\mathcal{P} = \{(\boldsymbol{X}_i, \boldsymbol{f}_i)\}$$

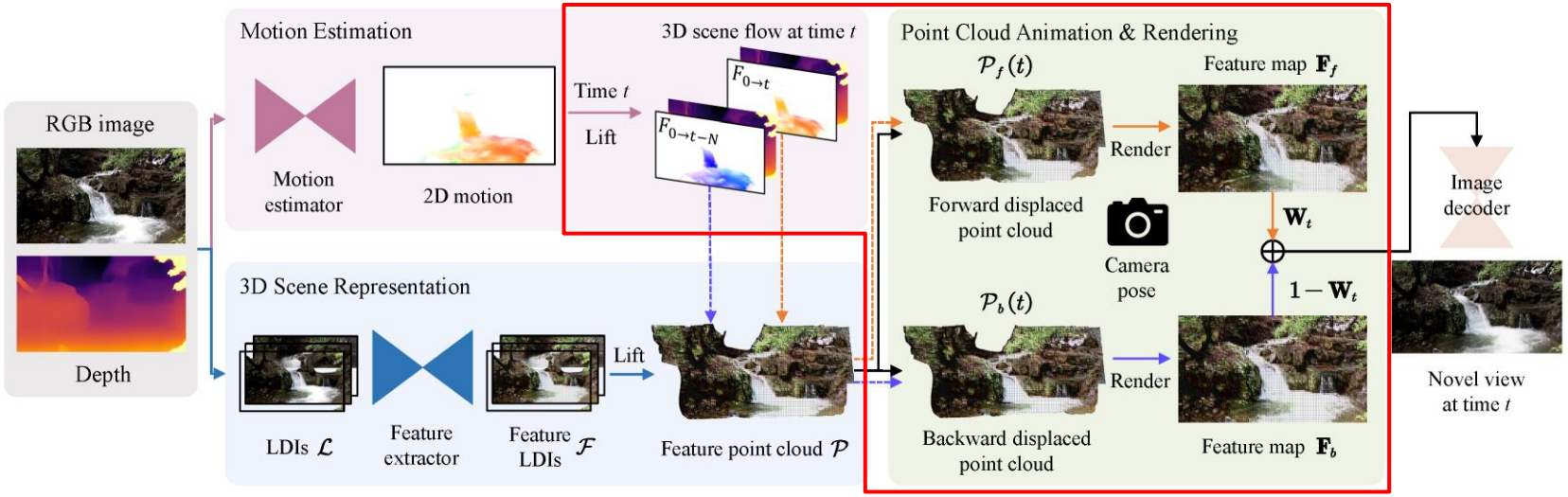
✓ $\boldsymbol{X}_i$: i-th 3D point의 3D 좌표

✓ $\boldsymbol{f}_i$: i-th 3D point의 feature vector



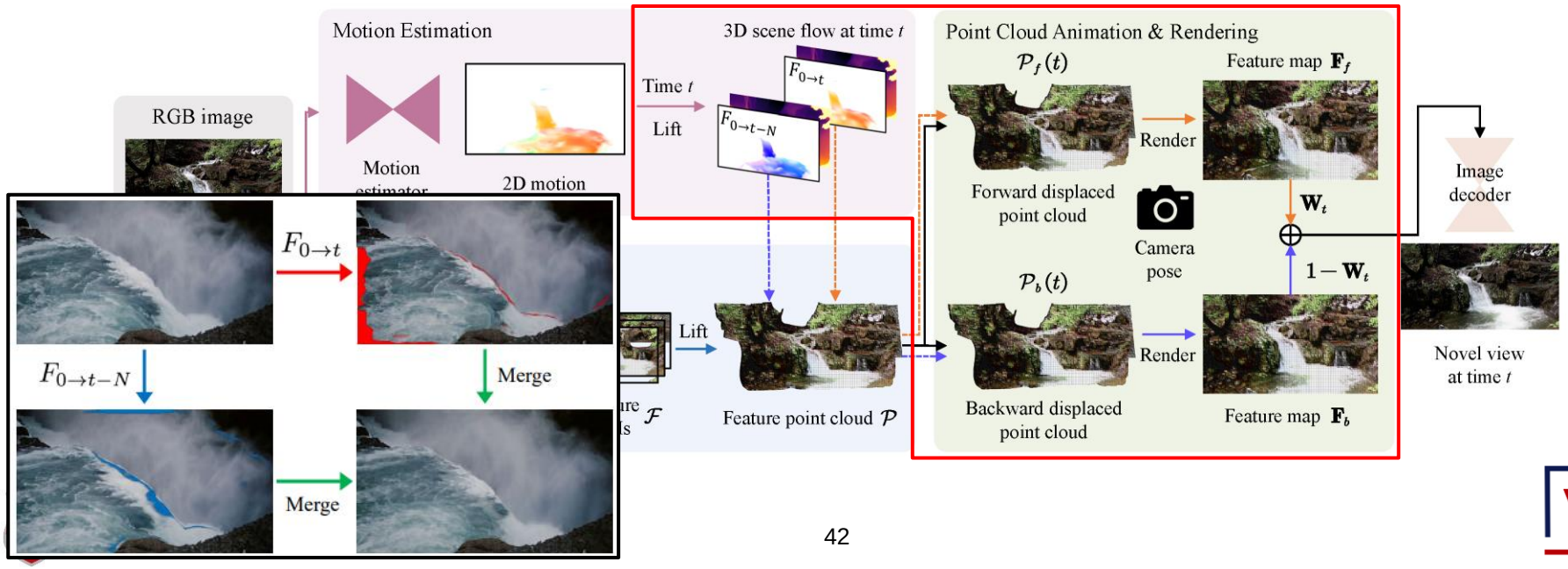
Method

- Point cloud animation & rendering
  - 이제 2D displacement field $F_{0\to t}$와 3D feature point cloud $\mathcal{P}$를 얻었음
  - 2D displacement field를 estimated depth와 함께 3D scene flow로 변환함
    - 3D scene flow를 이용해 시점 t에서의 3D feature point cloud를 얻을 수 있음
      - $\mathcal{P}(t) = \{(\boldsymbol{X}_i(t), \boldsymbol{f}_i)\}$
    - 하지만 여전히 3D point가 이동하면서 원래 위치에 빈 공간이 발생하는 hole 문제가 생김
      - 이를 3D symmetric animation을 통해 해결함



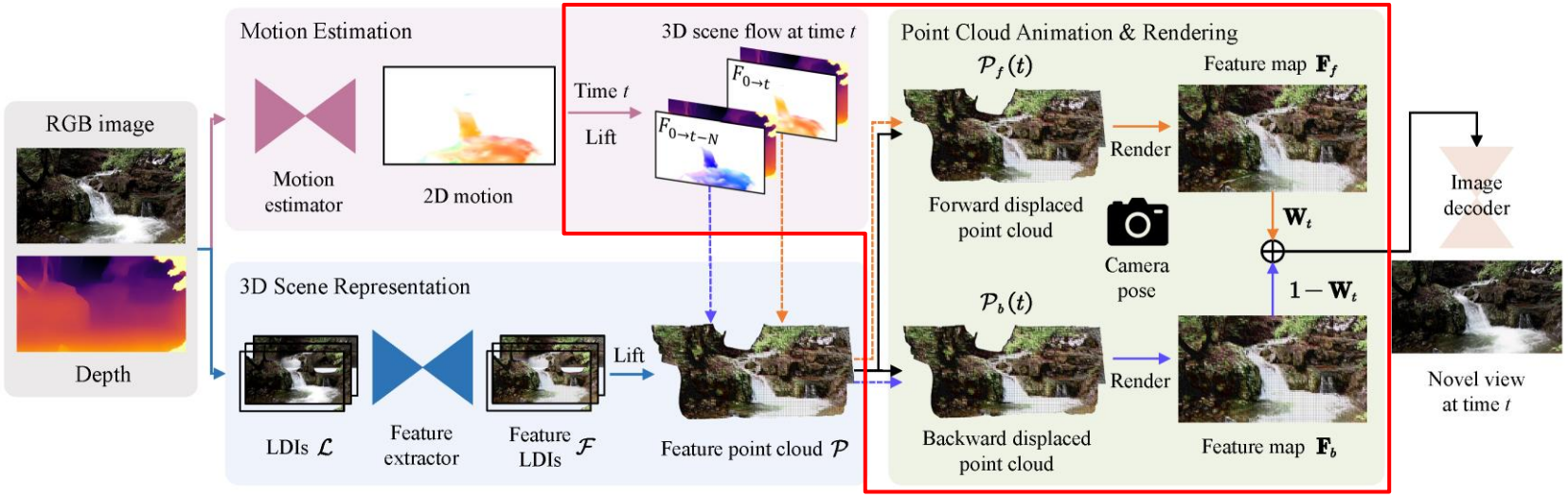
Method

- Point cloud animation & rendering
  - 3D symmetric animation
    - Point cloud를 양방향(forward + backward)으로 이동시켜서 서로 보완함
      - Backward field로서 $F_{0\to t}$ 대신 $F_{0\to t-N}$를 사용함
    - Forward displacement field $F_{0\to t}$와 backward displacement field $F_{0\to t-N}$를 각각 사용하여 양방향 point cloud $\mathcal{P}_f(t) = \{(\boldsymbol{X}_i^f(t), \boldsymbol{f}_i)\}$, $\mathcal{P}_b(t) = \{(\boldsymbol{X}_i^b(t), \boldsymbol{f}_i)\}$를 생성함
      - 양방향 point cloud를 merge하여 hole이 발생하는 문제를 해결할 수 있음



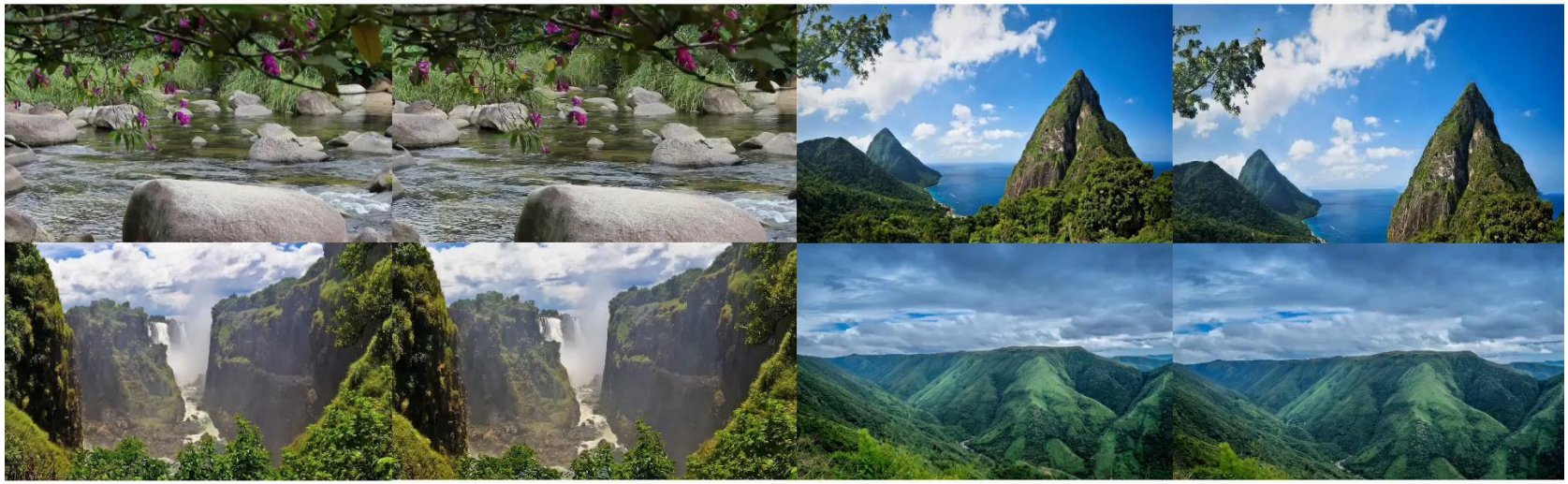
Method

- Point cloud animation & rendering
  - Neural rendering
    - Camera pose와 intrinsic parameter를 사용하여 양방향 point cloud를 image plane에 투영함
    - 이를 통해 2D feature map $\boldsymbol{F}_f$와 $\boldsymbol{F}_b$, depth map $\boldsymbol{D}_f$와 $\boldsymbol{D}_b$, alpha map $\boldsymbol{\alpha}_f$와 $\boldsymbol{\alpha}_b$를 얻음
    - 양방향 feature map을 가중치 $\boldsymbol{W}_t$를 이용해 합성하여 최종 feature map $\boldsymbol{F}_t$를 생성함
    - $\mathbf{W}_t = \frac{(1-\frac{t}{N})\cdot\boldsymbol{\alpha}_f\cdot e^{-\mathbf{D}_f}}{(1-\frac{t}{N})\cdot\boldsymbol{\alpha}_f\cdot e^{-\mathbf{D}_f}+\frac{t}{N}\cdot\boldsymbol{\alpha}_b\cdot e^{-\mathbf{D}_b}}$
    - 최종 decoder를 거쳐서 시점 t에서의 novel view를 얻음

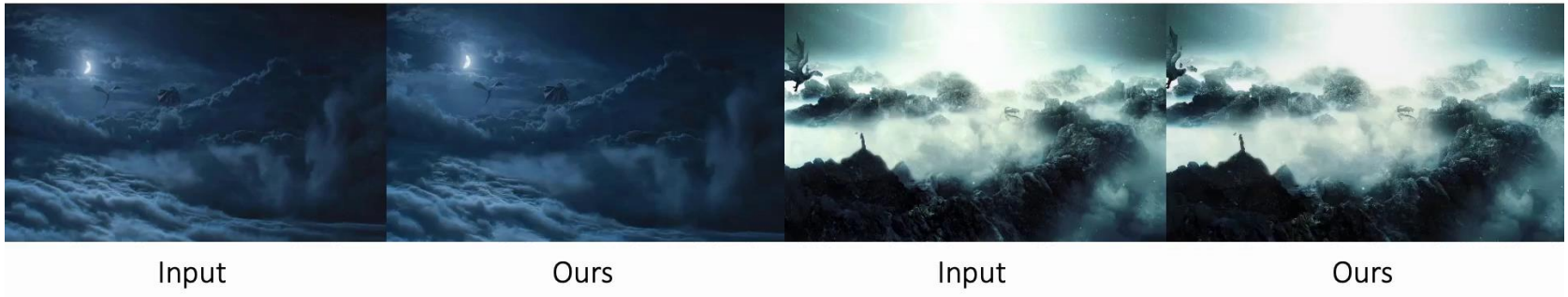


Experimental Results

- Real-world Photos



- Computer-Generated Imagery



Experimental Results

- Paintings

- Synthetic Images Generated by Stable Diffusion

LI, Xingyi, et al. “3d cinemagraphy from a single image.” In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2023.
1) HOLYNSKI, Aleksander, et al. Animating pictures with eulerian motion fields. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2021. p. 5810-5819.
2) Meng-Li Shih, Shih-Yang Su, Johannes Kopf, and Jia-Bin Huang. 3d photography using context-aware layered depth inpainting. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020.

Experimental Results

- Quantitative Results
  - 단일 이미지로부터 사실적인 parallex 효과를 가진 cinemagraph를 생성하는 기존 task가 없기 때문에 직접적인 비교는 어려움
  - 우수성을 검증하기 위해 몇 가지 baseline을 설정함 → **제안한 방법론의 우수성 확인 가능**
    - 2D animation → Novel View Synthesis: Animating pictures with Eulerian motion fields[1)]를 이용해 2D animation을 수행한 이후 3D Photo[2)]를 이용해 NVS 수행
    - Novel View Synthesis → 2D animation: 3D photo를 사용해 NVS를 수행한 다음 1)을 이용해 각 viewpoint를 animation화
      - MA(Moving Average): Viewpoint마다 motion field가 달라지므로 이 변동성을 완화하기 위해 사용
    - Naïve Point Cloud Animation: Pixel을 직접 3D 공간으로 투영 후 flow를 이용해 point cloud 이동 및 렌더링
      - 3DSA(3D Symmetric Animation): 위에서 사용한 대칭 기술을 추가하여 baseline을 개선

Method	PSNR↑	SSIM↑	LPIPS↓
2D Anim. [19] → NVS [52]	21.12	0.633	0.286
NVS [52] → 2D Anim. [19]	21.97	0.697	0.276
NVS [52] → 2D Anim. [19] + MA	22.47	0.718	0.261
Naive PC Anim.	19.46	0.647	0.243
Naive PC Anim. + 3DSA	20.49	0.660	0.237
Ours	**23.33**	**0.776**	**0.197**

Any Questions?