Animating Still Images in 3D Space



Pusan National University
Computer Vision & Signal Processing Lab, Dept. of Electronic Engineering



Presented By
김소형

CVSP Lab
Computer Vision & Signal Processing

Outline

- 논문 선택 이유
- Background
- Optimizing 4D Gaussians for Dynamic Scene Video from Single Landscape Images [ICLR 2025]
- LoopGaussian: Creating 3D Cinemagraph with Multi-view Images via Eulerian Motion Field [ACM MM 2024]
- Conclusion

논문 선택 이유

- Contribution 논문 작성
  - 정적 이미지를 input으로 3D animation 생성
  - 3D-MOM의 문제점 지적

- 방학 계획 달성
  - 논문 잘 읽는 사람 되기

CVSP Lab
Computer Vision & Signal Processing

Background

- Task
  - Cinemagraph
    - 정적 이미지와 비디오의 결합으로, 장면 대부분은 고정되어 있고 일부는 연속적으로 반복되는 형식
      - Traditional manual generation method: 숙련된 아티스트의 수작업으로 cinemagraph 생성
      - Automatic generation method: large dataset을 통한 pre-training cinemagraph 생성





< Cinemagraph의 예시>

Background

- Task
  - 3D Cinemagraph
    - 정적인 이미지나 장면을 3차원 공간에서 표현하고, 그 일부에만 움직임을 부여하여 시각적으로 살아있는 듯한 효과를 주는 기법



< 3D Cinemagraph의 예시>

CVSP Lab
Computer Vision & Signal Processing

Background

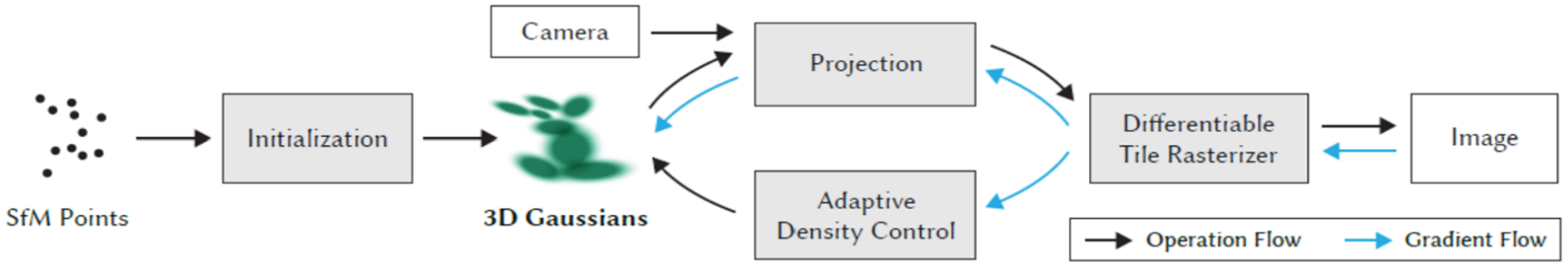
- Task
  - Optical flow
    - 연속된 두 프레임 간의 픽셀 이동을 추정하는 기법
    - 영상 속 객체의 움직임과 카메라 시점 변화를 반영



< Optical flow 시각화 이미지 >

Background

- Method
  - 3D Gaussian Splatting
    - Initialization: Sfm알고리즘을 통해 Camera pose 및 Point Cloud 정보를 얻어 3D Gaussian의 초기값으로 사용
    - Projection: 3D Gaussian을 Camera 방향에 따라 Image plane으로 projection되어 2D Gaussian 형태가 됨
      - GT 입력이미지와 비교하여 parameter를 업데이트 하기 위함
    - Differentiable Tile Rasterizer: 미분 가능한 형태의 Tile Rasterization을 통해 2D Gaussian들을 하나의 Image로 생성

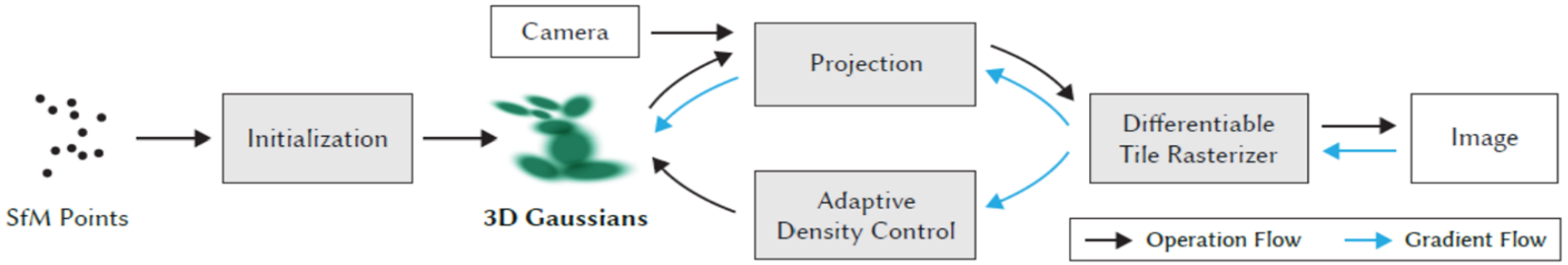


< 3D-GS의 전체 Framework >

CVSP Lab
Computer Vision & Signal Processing

Background

- Method
  - 3D Gaussian Splatting
    - Gradient Flow: 생성된 이미지와 GT 입력 이미지의 Loss를 계산하고 Gradient를 전파
    - Adaptive Density Control: Gradient를 기반으로 Gaussian의 형태를 변화시킴

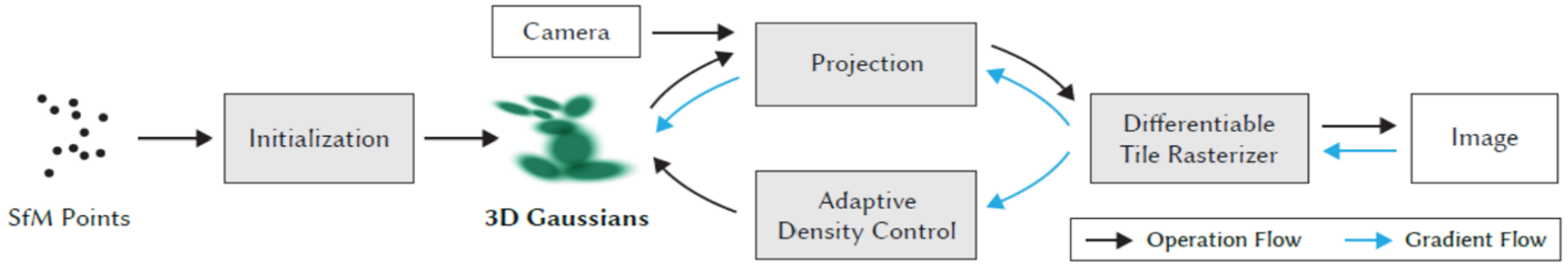


< 3D-GS의 전체 Framework >

CVSP Lab
Computer Vision & Signal Processing

Background

- Method
  - 3D Gaussian Splatting
    - Attribute: $\{p_i, q_i, s_i, \sigma_i, c_i, SH_i\}$
      - p_i : space position
      - q_i : quaternion representing rotation(3차원 상의 회전)
      - s_i : 각 축의 scaling
      - σ_i : 투명도 (opacity)
      - c_i : diffuse color
      - SH_i : spherical harmonic function

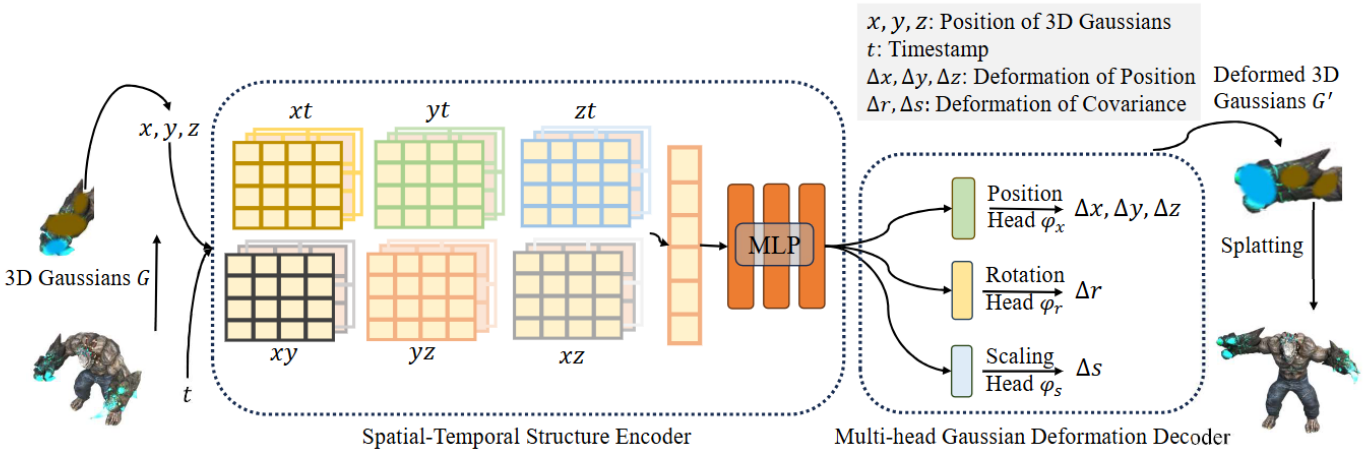


< 3D-GS의 전체 Framework >

CVSP Lab
Computer Vision & Signal Processing

Background

- Method
  - 4D Gaussian Splatting
    - Static한 3D Gaussian을 생성한 후, Gaussian의 위치정보에 Timestamp t를 추가하여 입력
    - 4차원 공간을 2D slice 형태로 분해해서 인코딩
    - 특정 Gaussian이 어느 시공간적 위치에 있는지 feature를 뽑아냄
    - 뽑아낸 Feature를 MLP의 입력으로 넣어 시간에 따른 3D Gaussian의 deformation을 학습



< 4D-GS의 전체 Framework >

Optimizing 4D Gaussians for Dynamics Scene Video from Single Landscape Images [ICLR 2025]

Introduction

- Contribution
  - Main
    - Dynamic scene video를 통한 3D 공간 가상화 방법 제안, 이를 위해 single 이미지로부터의 4D Gaussian을 optimize함
    - 3D Gaussian을 animate하기 위해, multi-view 이미지로부터 2D motion을 추정하고 이를 3D로 unprojection시켜 3D motion 생성
    - 완전한 3D 공간을 모델링하기 위해 Gaussian 기반의 공간 표현을 사용함. Volumetric representation을 통해 사실적인 깊이 인식 뿐 아니라, explicit output으로 인해 매우 실용적임



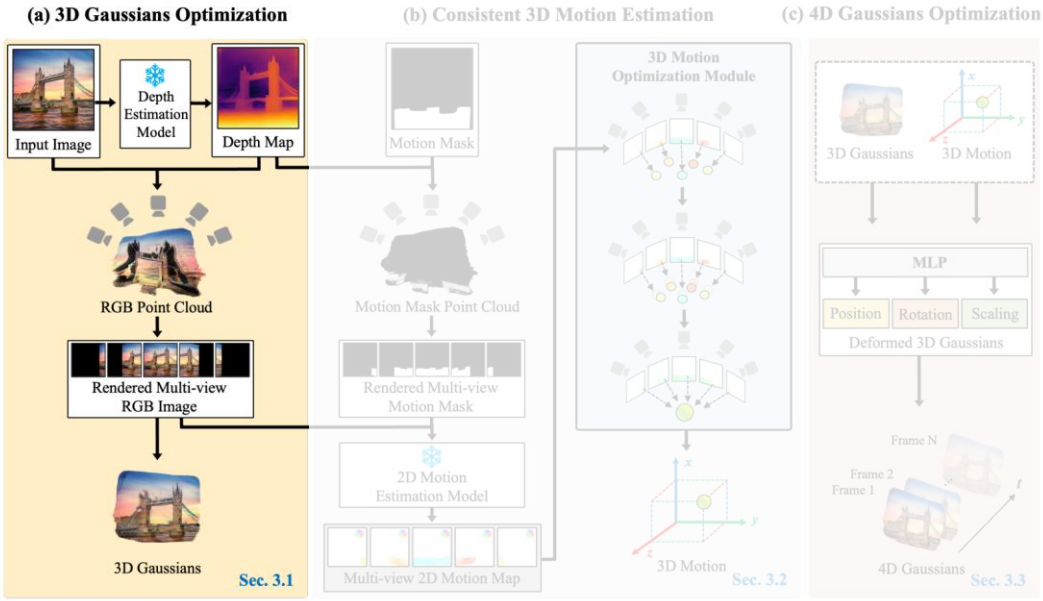
< input >



< output>

Method

- Overview
  - Multi-view Image Generation for 3D Gaussians Optimization
    - Input 이미지를 통해 depth map을 추정 후, 이미지와 depth map을 3D로 unprojection하여 RGB point cloud 생성
    - 카메라 파라미터에 따라 point cloud를 projection하여 Multi-view 이미지 렌더링
    - 렌더링된 이미지를 이용하여 3D Gaussian을 initialization 및 optimization

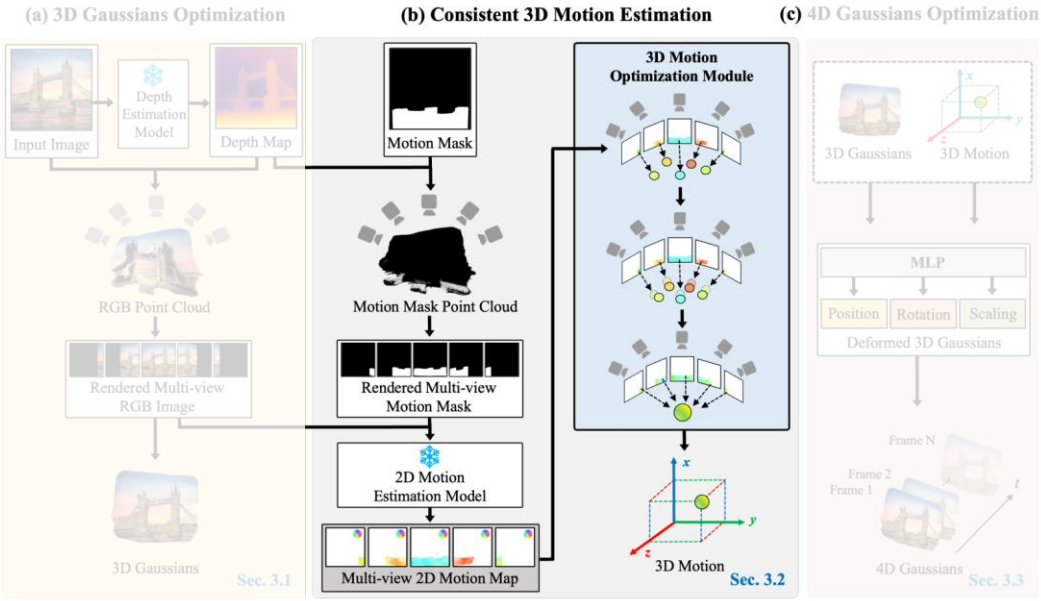


< The Overview of pipeline >

CVSP Lab
Computer Vision & Signal Processing

Method

- Overview
  - Consistent 3D Motions Estimation
    - Input 이미지에서 animation이 적용될 영역을 지정하는 초기 motion mask 생성
    - 초기 motion mask를 기반으로, 특정 카메라 viewpoint $\mathbf{E}_i$에 대응하는 motion mask $\widetilde{\mathbf{M}}_i$를 생성
    - 생성된 Multi-view 이미지 $\tilde{\mathbf{I}}_i$ 와 해당 motion mask $\widetilde{\mathbf{M}}_i$를 입력으로 하여, Eulerian flow EF를 통해 2D motion map 추정

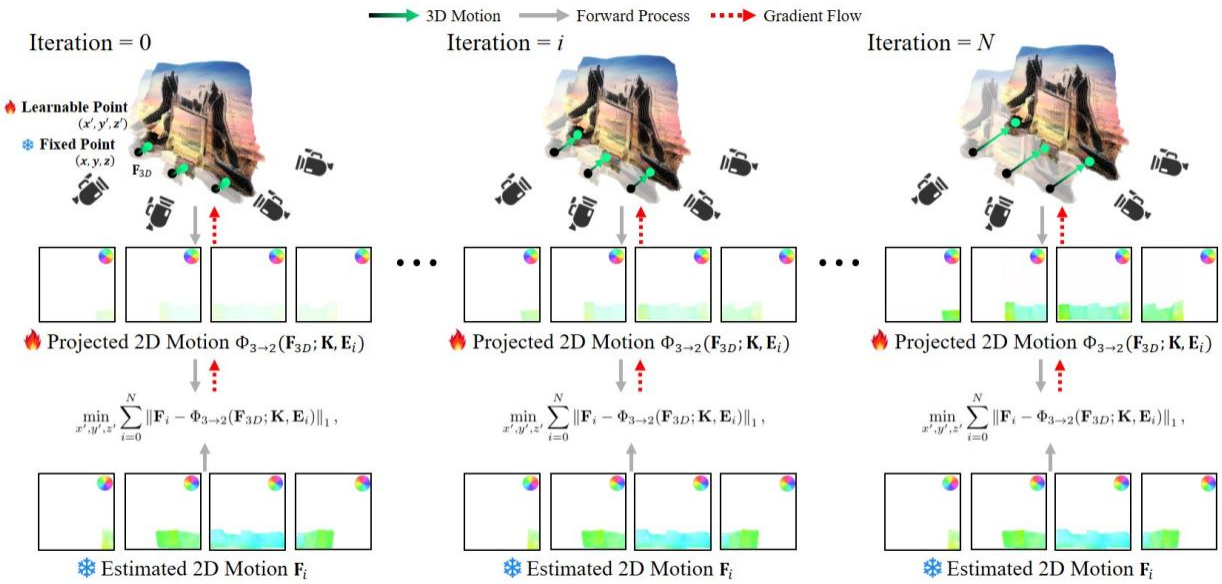


< The Overview of pipeline >

CVSP Lab
Computer Vision & Signal Processing

Method

- Overview
  - Consistent 3D Motions Estimation
    - Point cloud로부터 3D motion field 정의
    - Eulerian flow로 추정한 2D motion을 각 view의 GT로 하고, 정의된 3D motion을 view에 맞춰 projection 하고, GT 2D motion과의 차이를 L1 loss로 학습
    - 반복적인 학습 과정을 통해 3D motion field가 점차 optimize됨

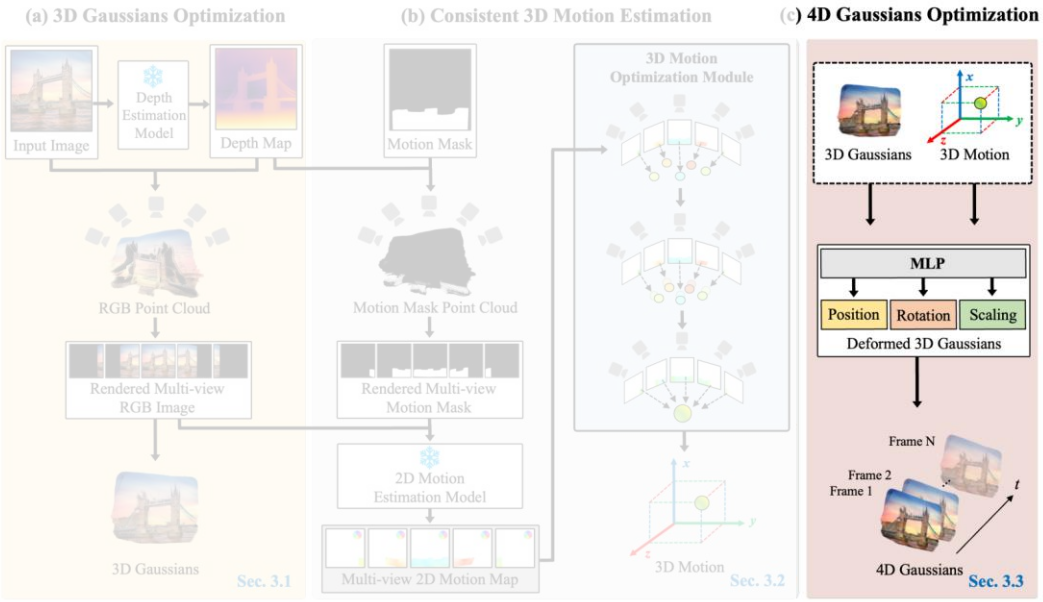


< 3D MOM의 Framework >

CVSP Lab
Computer Vision & Signal Processing

Method

- Overview
  - 4D Gaussians Optimization
    - 효율적인 학습 시간과 높은 품질을 보장하기 위해 4D-GS 채택
    - 다음 프레임 3D Gaussian의 위치 변화량의 초기값을 앞서 추정한 3D motion으로 설정
    - Multi-view 이미지 $\tilde{\mathbf{I}}$ 와 input 이미지로부터 추정한 2D animation 결과를 활용하여 $\Delta\mathbf{x}$', $\Delta\mathbf{r}$, $\Delta\mathbf{s}$ 를 추정하기 위한 학습 공동으로 진행
    - Two-stage training을 통해 4D Gaussian 학습

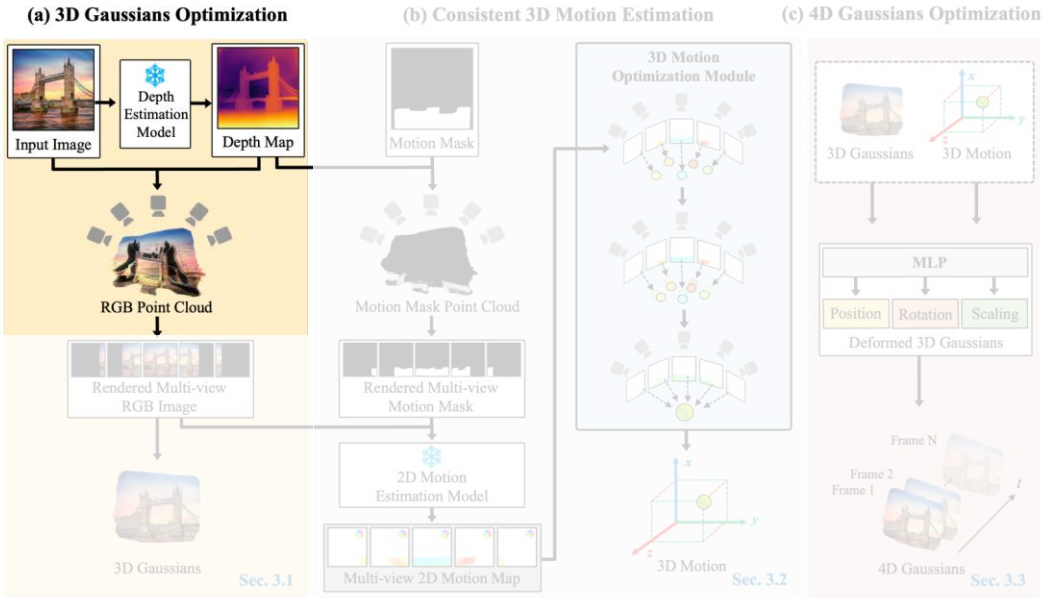


< The Overview of pipeline >

CVSP Lab
Computer Vision & Signal Processing

Method

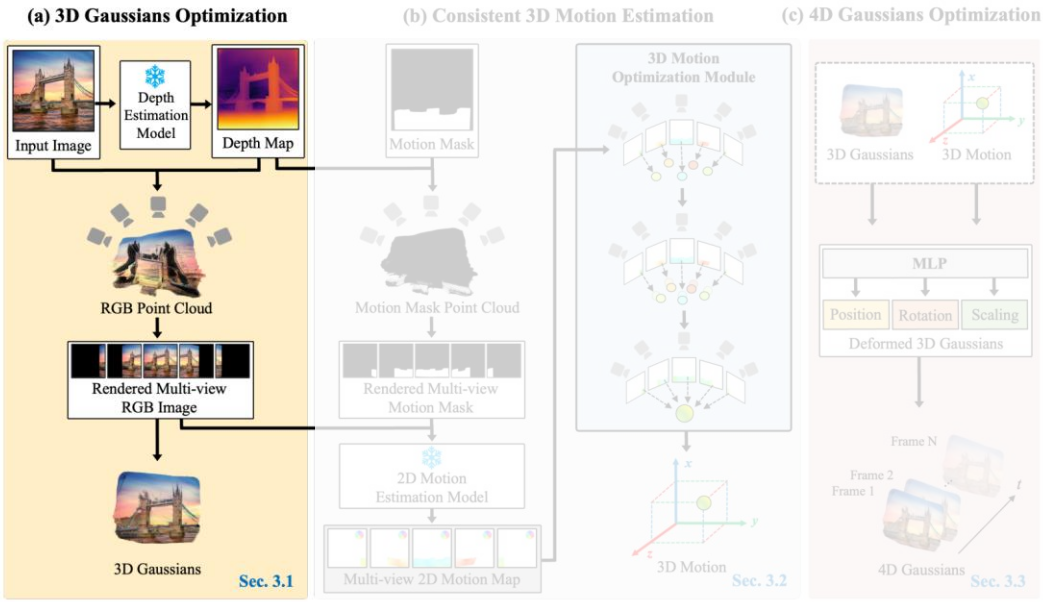
- Multi-view Image Generation for 3D Gaussians Optimization
  - Point Cloud Generation
    - Input이미지 **I**로부터 ZoeDepth를 이용하여 depth map **D**를 추정
    - 이 후, input이미지와 depth map을 3D로 unprojection시켜 point cloud P_I 생성
      - $P_I = \{(\mathbf{X}_i, \mathbf{C}_i)\} = \Phi_{2\to3}(\mathbf{I}, \mathbf{D}; \mathbf{K}, \mathbf{E_0})$
        - ✓ $\mathbf{X}_i$: 3D상의 좌표 , $\mathbf{C}_i$: RGB값, **K**: 카메라 intrinsic matrix, $\mathbf{E_0}$: 카메라 extrinsic matrix
        - ✓ $\Phi_{2\to3}$: RGB 이미지에서 point cloud로 lifting시키는 함수



< The Overview of pipeline >

Method

- Multi-view Image Generation for 3D Gaussians Optimization
  - Multi-view Image Rendering and 3D Gaussians Optimization
    - 카메라 파라미터에 따라 3D point cloud를 2D로 projection
      - $\tilde{\mathbf{I}}_i = \Phi_{3\to2}(P_I, \mathbf{K}, \mathbf{E}_i)$
        - ✓ $\mathbf{E}_i$: 특정 카메라 extrinsic parameter
    - 위 과정은 중심카메라 viewpoint $\mathbf{E}_0$부터 시작하여, 모든 카메라를 순회할 때까지 렌더링 진행



< The Overview of pipeline >

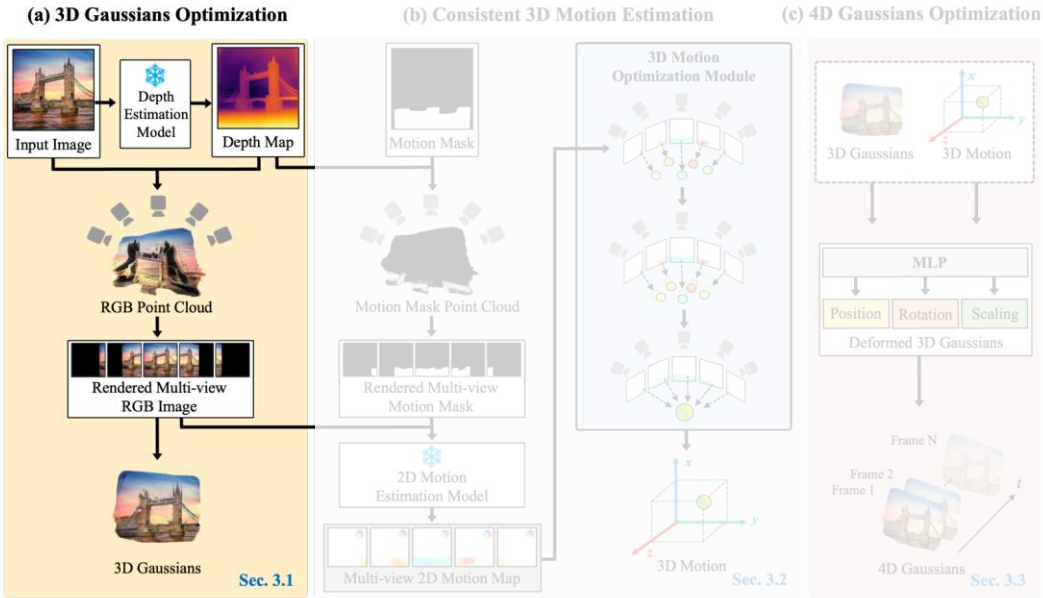
CVSP Lab
Computer Vision & Signal Processing

Method

- Multi-view Image Generation for 3D Gaussians Optimization
  - Multi-view Image Rendering and 3D Gaussians Optimization
    - 렌더링 중 발생하는 hole을 처리하기 위해 선형 보간 적용
    - 렌더링 된 multi-view 이미지는 multi-view loss의 GT 역할
    - 위를 통해 3D Gaussian의 initialization과 optimization 진행

$$L = L1 + \lambda(L_{SSIM} - L1)$$

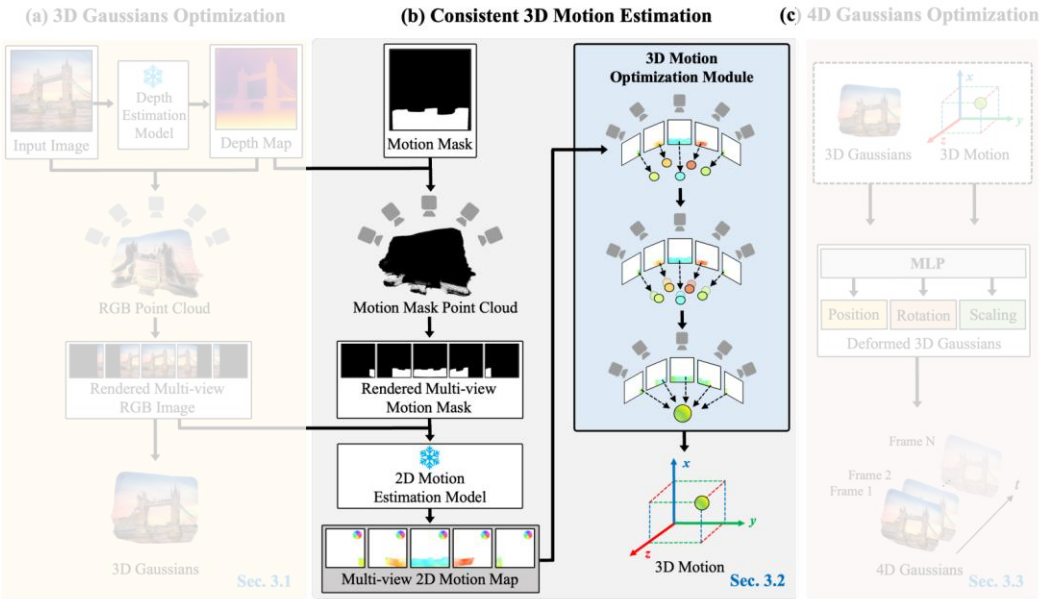
✓ λ: 두 loss의 balance를 조절하는 가중치, L_{SSIM}: 구조적 유사성을 판단하는 loss



< The Overview of pipeline >

Method

- Consistent 3D Motions Estimation
  - Multi-view 2D Motion Estimation
    - Animation이 적용될 영역을 지시하는 초기 motion mask를 LabelMe를 통해 생성
    - 초기 motion mask로부터 특정 카메라 viewpoint $\mathbf{E}_i$에 따라 motion mask $\widetilde{\mathbf{M}}_i$를 생성
      - $\widetilde{\mathbf{M}}_i = \Phi_{3\to2}(\Phi_{2\to3}(\mathbf{M_0}, \mathbf{D}; \mathbf{K}, \mathbf{E_0}); \mathbf{K}, \mathbf{E}_i)$
        - ✓ $\widetilde{\mathbf{M}}_i$: viewpoint $\mathbf{E}_i$ 따른 motion mask
        - ✓ $\mathbf{M_0}$: 초기 motion mask

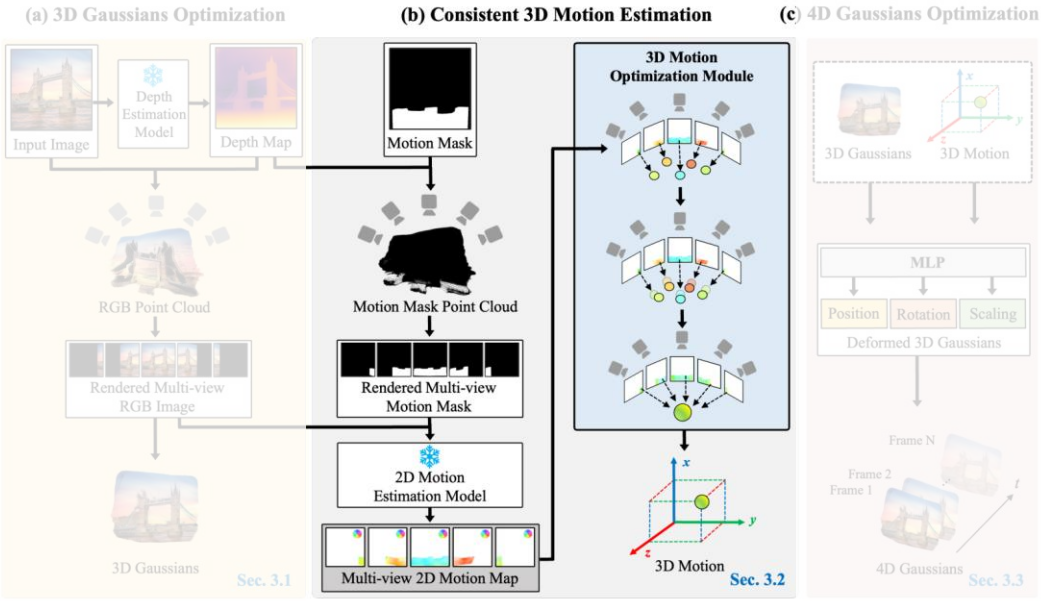


< The Overview of pipeline >

CVSP Lab
Computer Vision & Signal Processing

Method

- Consistent 3D Motions Estimation
  - Multi-view 2D Motion Estimation
    - Multi-view 이미지 $\tilde{\mathbf{I}}_i$ 와 대응하는 $\widetilde{\mathbf{M}}_i$으로 Eulerian flow EF 를 통해 2D motion map 추정
      - $\mathbf{F}_i = EF(\tilde{\mathbf{I}}_i, \widetilde{\mathbf{M}}_i)$
    - Multi-view motion maps $\mathbf{F}_i$는 각 view에 따라 독립적으로 추정한 것이므로 ***Motion Ambiguity***가 발생할 수 있음
      - ***Motion Ambiguity***: 각 view의 2D motion을 3D로 unprojection했을 때 모션이 불일치하는 문제

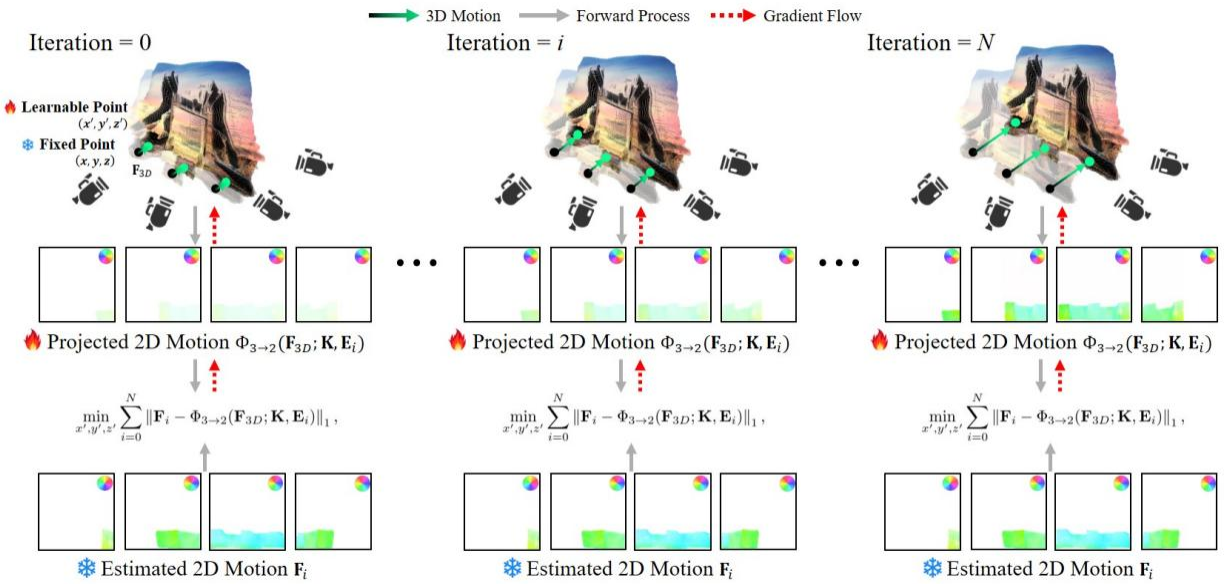


< The Overview of pipeline >

CVSP Lab
Computer Vision & Signal Processing

Method

- Consistent 3D Motions Estimation
  - 3D Motion Optimization Module
    - 이전에 생성된 Point cloud로부터 3D motion $\mathbf{F}_{3D}$ 정의
      - $\mathbf{F}_{3D} = (x', y', z') - (x, y, z)$
        - ✓ x, y, z: point의 초기 좌표
        - ✓ x', y', z': point의 추정 좌표



< 3D MOM의 Framework >

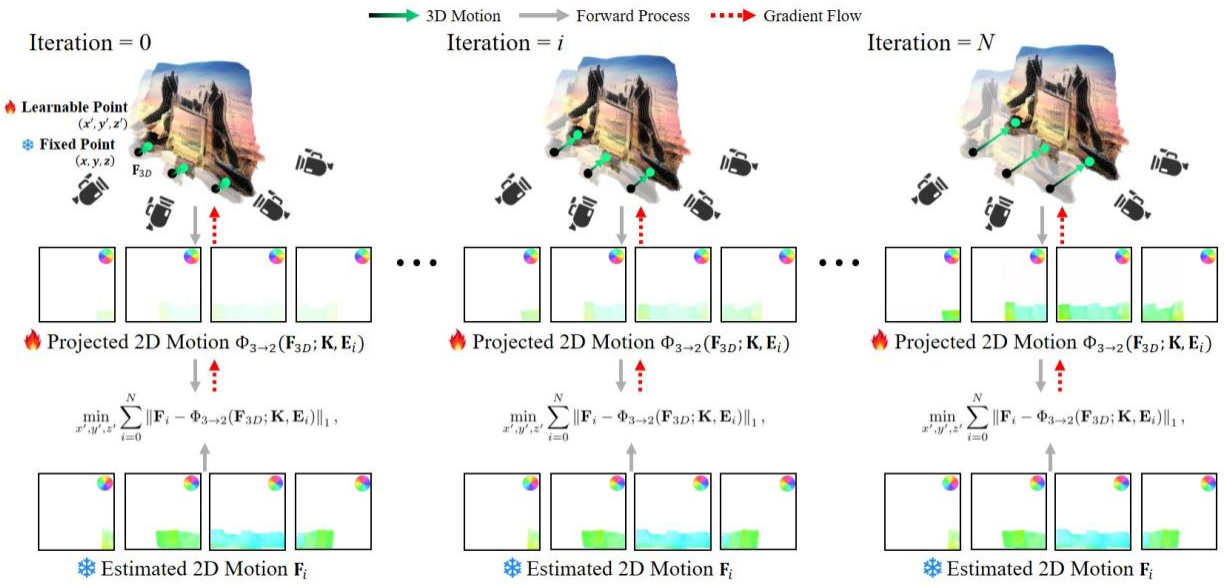
CVSP Lab
Computer Vision & Signal Processing

Method

- Consistent 3D Motions Estimation
  - 3D Motion Optimization Module
    - Multi-view motion maps $\mathbf{F}_i$ 을 특정 view의 GT로 하고, 정의된 3D motion을 특정 view에 맞춰 projection 한 것을 L1 loss로 학습

$$\min_{x',y',z'} \sum_{i=0}^{N} \|\mathbf{F}_i - \Phi_{3\to2}(\mathbf{F}_{3D}, \mathbf{K}, \mathbf{E}_i)\|_1$$

✓ N: 카메라 trajectory를 따라 생성된 multi-view 이미지의 수

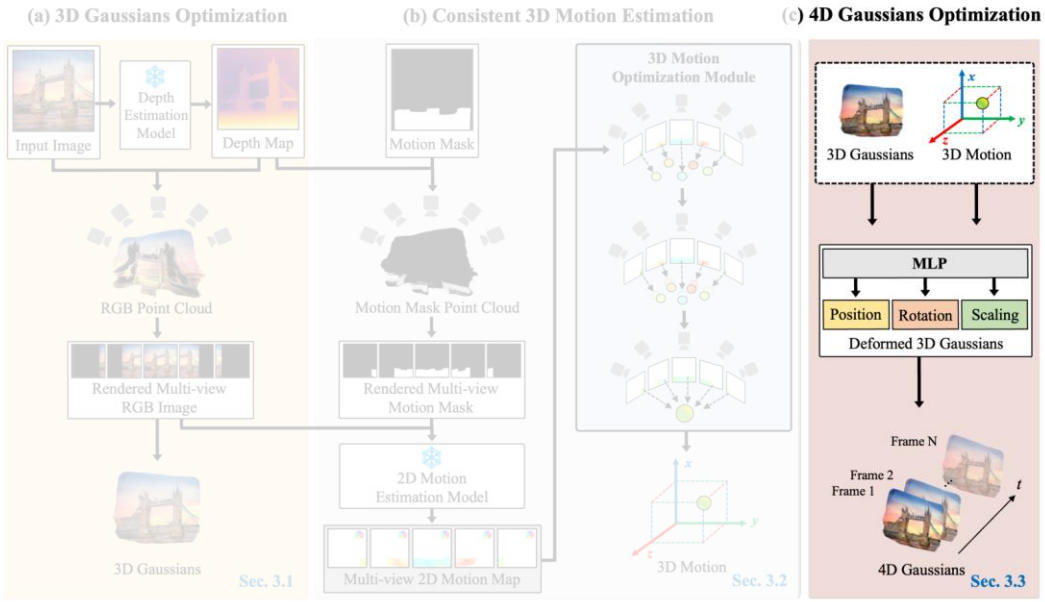


< 3D MOM의 Framework >

CVSP Lab
Computer Vision & Signal Processing

Method

- 4D Gaussians Optimization
  - 3D Motion Initialization
    - 이전 단계에서 추정된 3D motion $\mathbf{F}_{3D}$을 3D Gaussian의 위치 변화량의 초기값으로 사용
      - $(\mathbf{x}', \mathbf{r}', \mathbf{s}') = (\mathbf{x} + \mathbf{F}_{3D} + \Delta\mathbf{x}', \mathbf{r} + \Delta\mathbf{r}, \mathbf{s} + \Delta\mathbf{s})$
        - ✓ $\mathbf{x}, \mathbf{r}, \mathbf{s}$: 각각 3D Gaussian의 위치, rotation, scale의 초기값
        - ✓ $\Delta\mathbf{x}', \Delta\mathbf{r}, \Delta\mathbf{s}$: 각각 3D Gaussian의 위치, rotation, scale의 변화량

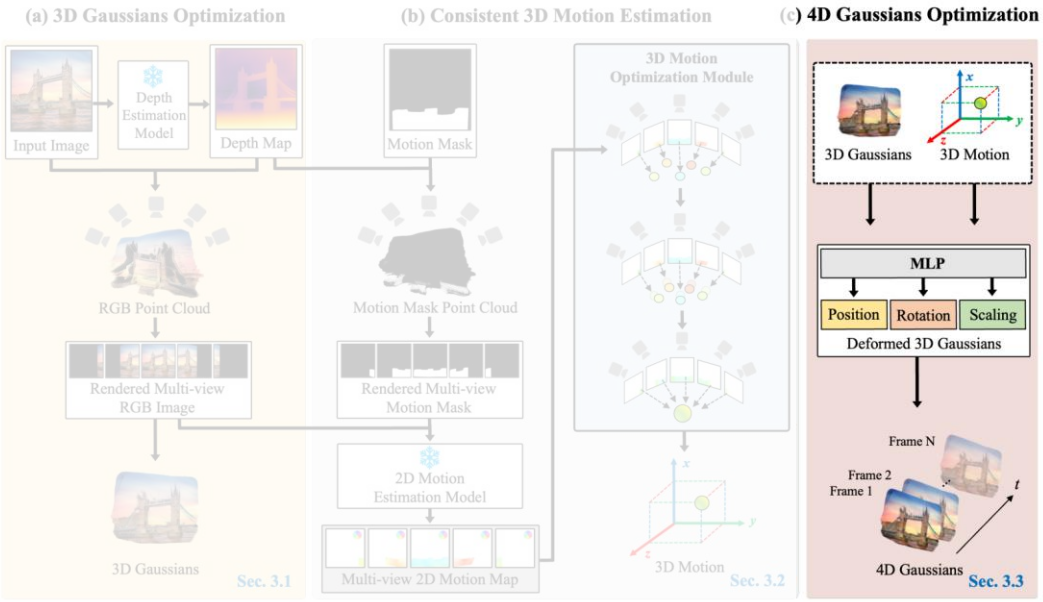


< The Overview of pipeline >

CVSP Lab
Computer Vision & Signal Processing

Method

- 4D Gaussians Optimization
  - 3D Motion Initialization
    - $\Delta\mathbf{r}$, $\Delta\mathbf{s}$ 는 4D Gaussian의 왜곡을 방지하기 위한 것으로 weak supervision으로도 충분히 학습 가능
    - 앞서 생성한 첫 프레임의 multi-view 이미지와 input 이미지의 2D 애니메이션 결과를 함께 활용하여 $\Delta\mathbf{r}$, $\Delta\mathbf{s}$ 를 학습

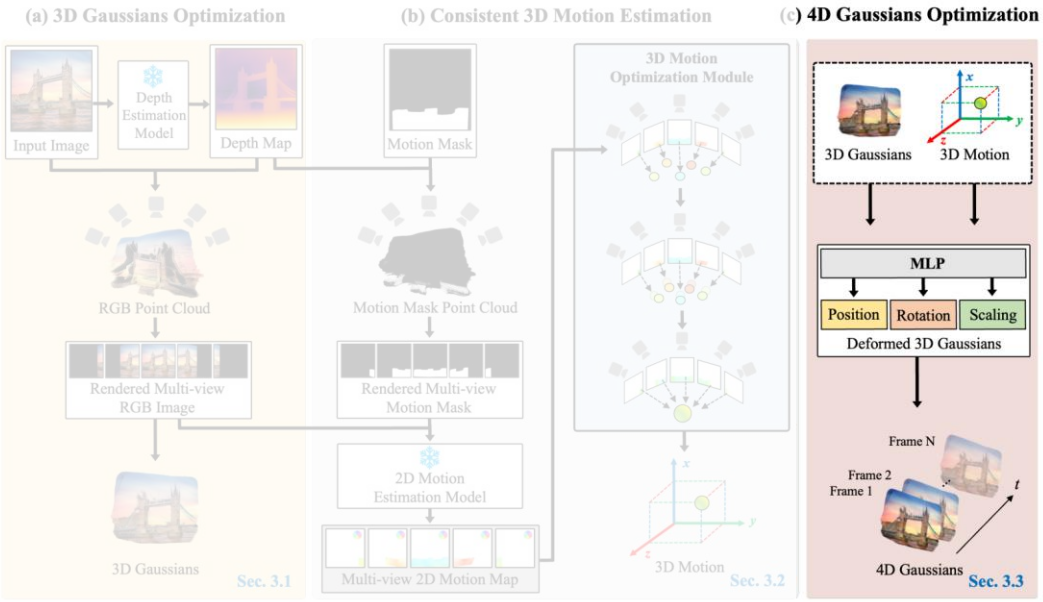


< The Overview of pipeline >

CVSP Lab
Computer Vision & Signal Processing

Method

- 4D Gaussians Optimization
  - Two-stage Training
    - 모든 multi-view 이미지 각각에 대해 영상을 생성하는 것은 중복이 심하고 매우 느림
      - viewpoint 간 겹치는 영역이 많기 때문
    - 이를 해결하기 위해 **Two-stage training** 제안

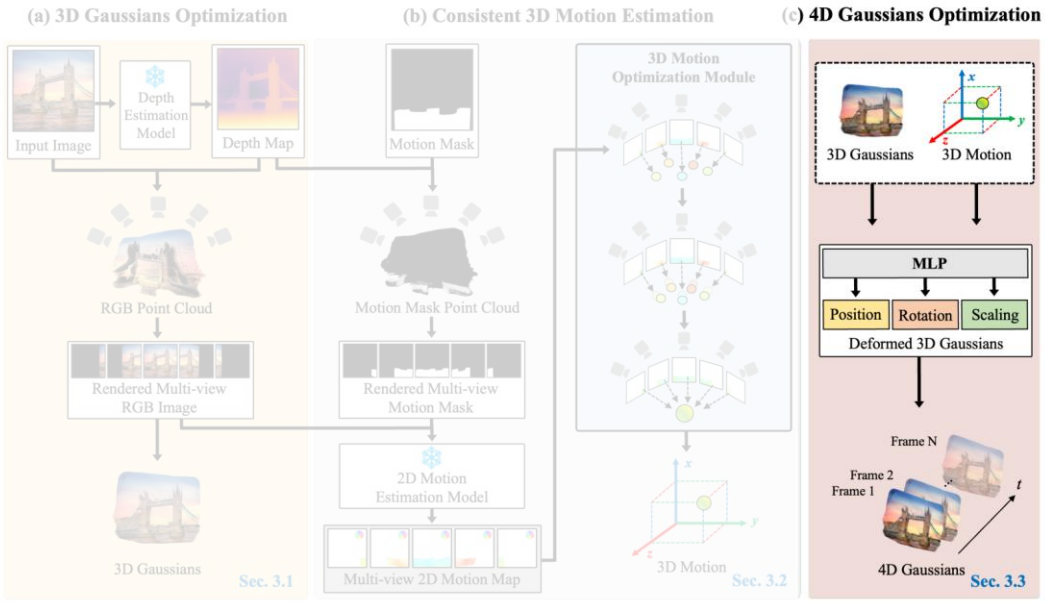


< The Overview of pipeline >

CVSP Lab
Computer Vision & Signal Processing

Method

- 4D Gaussians Optimization
  - Two-stage Training
    - Stage 1: 정지 상태에서의 전체 viewpoint 학습
      - 시간 t = 0 에 고정
      - 전체 viewpoint에 대해 3D Gaussian 학습

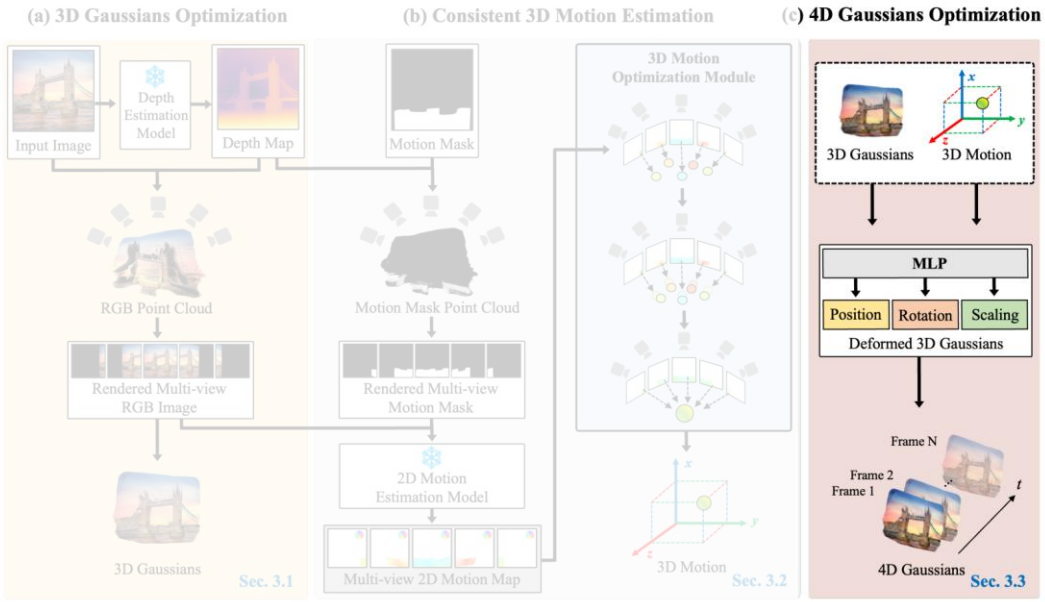


< The Overview of pipeline >

CVSP Lab
Computer Vision & Signal Processing

Method

- 4D Gaussians Optimization
  - Two-stage Training
    - Stage 2: 시간 변화에 따른 4D Gaussian 학습
      - 시간 축 전체에 걸쳐 학습
      - 모든 viewpoint가 아닌, 샘플링된 viewpoint만 사용
      - 각 프레임에 대해 motion을 포함한 4D Gaussian 학습



< The Overview of pipeline >

CVSP Lab
Computer Vision & Signal Processing

Experiment

- Experimental Setup
  - Implementation details
    - 입력 데이터 전처리
      - LabelMe를 통해 입력으로 쓸 이미지의 수동마스크 생성
        - ✓ 움직일 영역 지정
    - Multi-view 이미지 생성
      - ZoeDepth를 통해 depthmap 추정
      - 약 30개 뷰포인트의 카메라 궤적 생성
    - 3D motion optimizing
      - Optimizer: SGD
      - Iteration: 200회
      - Batch Size: 30
      - Learning Rate: 초기 0.5, 이후 Exponential decay
    - 환경
      - NVIDIA RTX 3090 GPU 1개

CVSP Lab
Computer Vision & Signal Processing

Experiment

- Experimental Setup
  - Evaulation dataset
    - Holynski et al. (2021)의 Validate dataset 사용
      - 162개의 비디오 클립
      - 27개의 장면
      - 고정카메라 시점
    - Li et al. (2023) “ 3D-Cinemagraphy”에서 정의한 4가지 카메라 궤적 적용
      - 기존 GT 비디오를 새로운 시점에서 렌더링
        - ✓ 렌더링 결과, 각 샘플당 240개의 GT 프레임 획득

Experiment

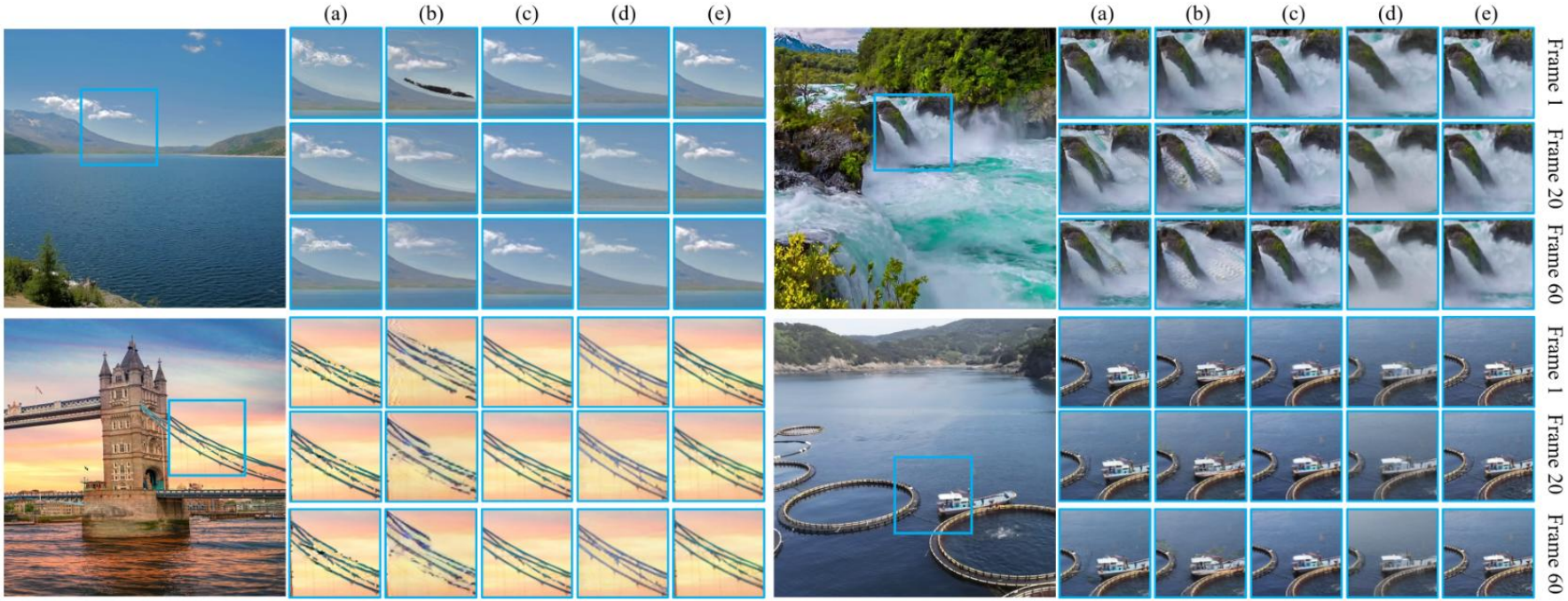
- Experimental Setup
  - Quantitative results
    - 합성 비디오의 개별 프레임의 품질을 평가
      - PSNR: 영상 화질 손실양을 평가하는 지표, MSE가 분모로 들어감으로써 화질 손실이 적을수록 PSNR 증가
      - SSIM: 두 이미지간 휘도, 대비, 구조 총 3가지 측면에서 평가
      - LPIPS: 비교할 2개의 이미지를 각각 VGG Net에 넣고, 중간 layer의 feature값들을 각각 뽑아내서, 2개의 feature가 유사한지를 측정
      - PIQE: 이미지를 블록 단위로 나눠 왜곡 가능성을 판단
        - ✓ GT없이 측정

Method	Metrics				User study (%)			
	PSNR ↑	SSIM ↑	LPIPS ↓	PIQE ↓	Immersion	Realism	Stuctural Consistency	Quality
DynamiCrafter (Xing et al., 2025)	14.98	0.81	0.23	24.58	-	-	-	-
Motion-I2V (Shi et al., 2024)	14.38	0.80	0.31	8.40	-	-	-	-
3D-Cinemagraphy (Li et al., 2023)	17.30	0.83	0.17	8.93	31.87	31.87	28.75	30.31
Make-It-4D (Shen et al., 2023)	16.98	0.81	0.20	8.30	11.87	10.31	8.43	9.06
Ours	**20.57**	**0.90**	**0.14**	**7.80**	**56.25**	**57.81**	**62.81**	**60.25**

< Quantitative results >

Experiment

- Experimental Setup
  - Qualitative results
    - (a) 3D Cinemagraphy (Li et al., 2023), (b) Make-It-4D (Shen et al.,2023), (c) DynamiCrafter (Xing et al., 2025), (d) Motion-I2V (Shi et al., 2024), (e) Our



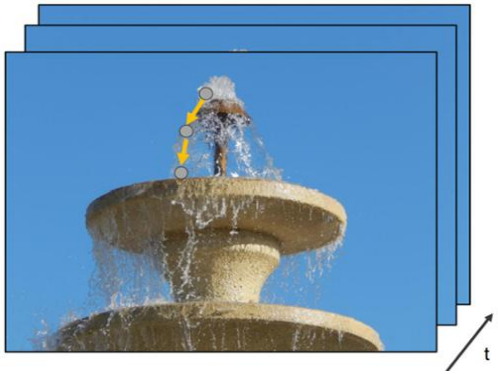
< Qualitative results >

Loop Gaussian: Creating 3D Cinemagraph with Multi-view Images via Eulerian Motion Field

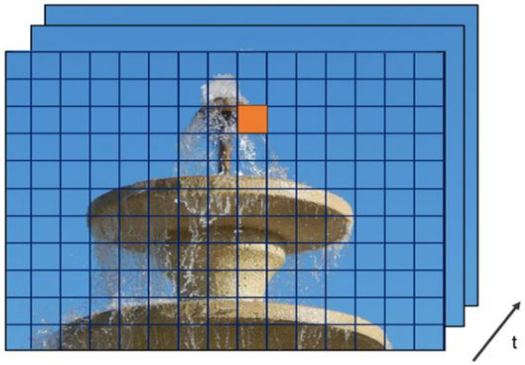
[ACM MM 2024]

Introduction

- Preliminary
  - Sparse Velocity Field Estimation
    - Lagrangian perspective: 개별 입자를 시간에 따라 추적하며, 입자가 공간을 따라 어떻게 이동하는지 봄
    - Eulerian perspective: 특정 위치에서 그 위치를 통과하는 입자들이 어떻게 움직이는 지를 봄
    - 본 논문에서는 부드럽고 non-rigid한 물체의 변형을 표현하기 위해 Eulerian perspective 채택
    - Eulerian Integration을 통해 다음 frame의 위치를 추정할 수 있음
      - $X(t+1) = X(t) + E(X\ t))$
        - ✓ $X(t)$: time t 에서 입자 위치, E: Eulerian motion field



< Lagrangian 관점 >



< Eulerian 관점 >

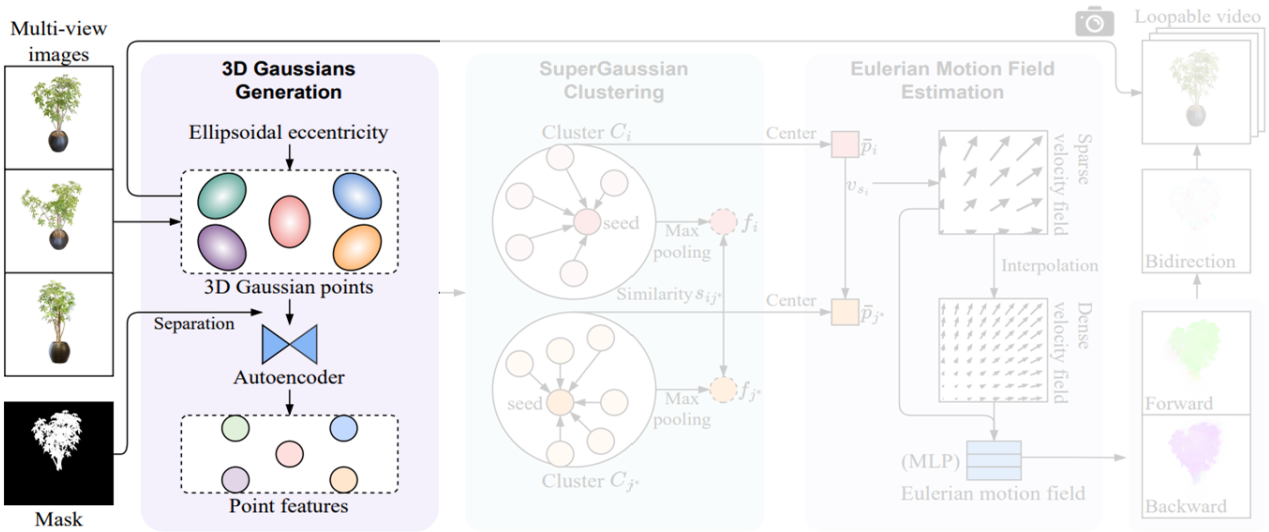
CVSP Lab
Computer Vision & Signal Processing

Introduction

- Problem & Solution
  - Problem
    - Traditional manual generation method
      - 숙련된 아티스트들의 수작업이 필요하므로 비용 문제 발생
    - Automatic generation method
      - large dataset을 통한 pre-training이 필요하여 비용 문제 발생
    - 이미지 공간에 제한되어 있고, viewing point를 바꿀 수 없음
  - Solution
    - 3D Gaussian modeling을 이용하여 cinemagraph를 2D에서 3D로 올리는 기법 제안
    - Scene representation과정에서 self-similarity를 효율적으로 활용하여 large dataset에 대한 pre-training 필요성 제거
    - 유사한 객체는 유사한 움직임을 보인다는 가정을 토대로 3D Gaussian들을 clustering한 후, motion field를 추정함

Method

- Overview
  - 3D Gaussians Generation
    - Multi-view 이미지들로 3D-GS를 통해 3D Gaussian point cloud 생성
    - 기존 3D-GS의 Loss에서 3D Gaussian의 타원 모양을 정규화하는 Ellipsoidal eccentricity 추가
    - 타원이 지나치게 날카로우면 렌더링 과정에서 tearing 발생
    - SAGA를 통해 정적, 동적 3D Gaussian을 구분하는 Mask생성 후 적용
    - Autoencoder를 통해 3D Gaussian들의 feature를 뽑아냄

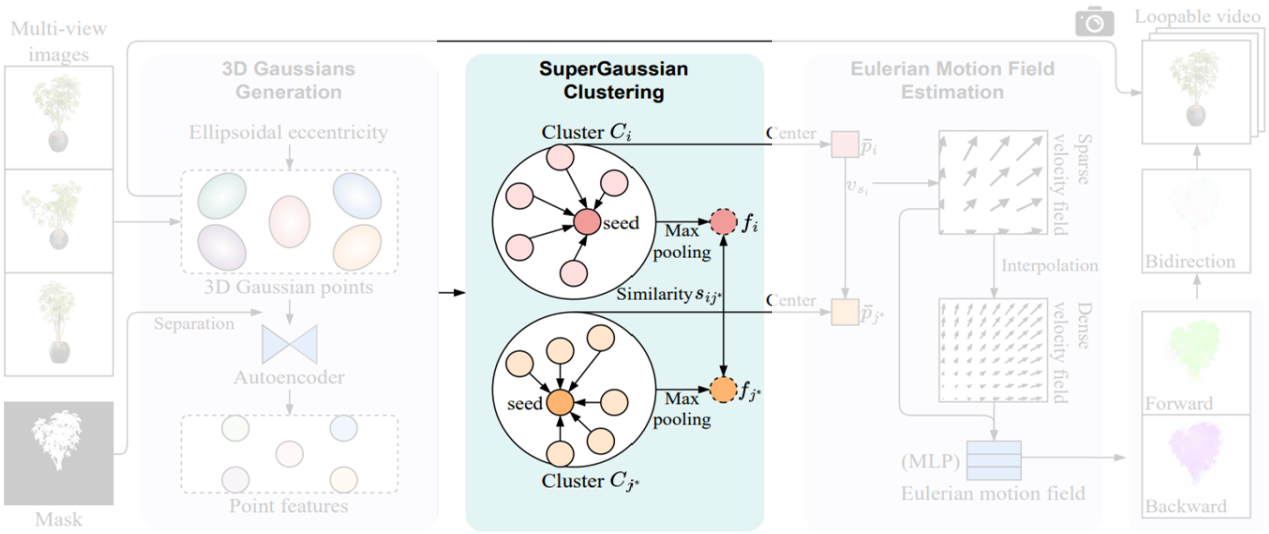


< LoopGaussian의 전체 Framework >

CVSP Lab
Computer Vision & Signal Processing

Method

- Overview
  - SuperGaussian Clustering
    - 일정 영역 내 3D Gaussian 중 1개를 랜덤하게 seed로 선정
    - seed를 중심으로 metric function을 이용하여 clustering
    - K-means clustering과 유사
    - 각 cluster에 대해 max pooling을 적용하여 global feature를 뽑아냄

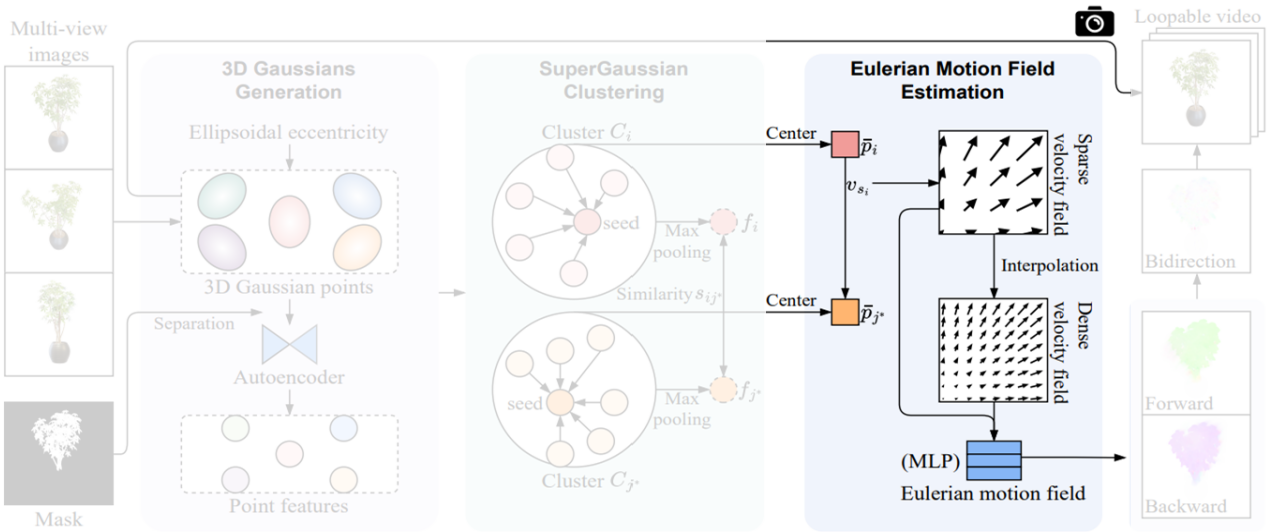


< LoopGaussian의 전체 Framework >

CVSP Lab
Computer Vision & Signal Processing

Method

- Overview
  - Eulerian Motion Field Estimation
    - Cluster들의 center position을 통해 Sparse velocity field를 추정
    - Kriging interpolation을 통해 Sparse velocity field로부터 Dense velocity field 추정
    - 모든 position에서 motion을 자연스럽게 하기 위해 MLP를 통해 Eulerian motion field 추정

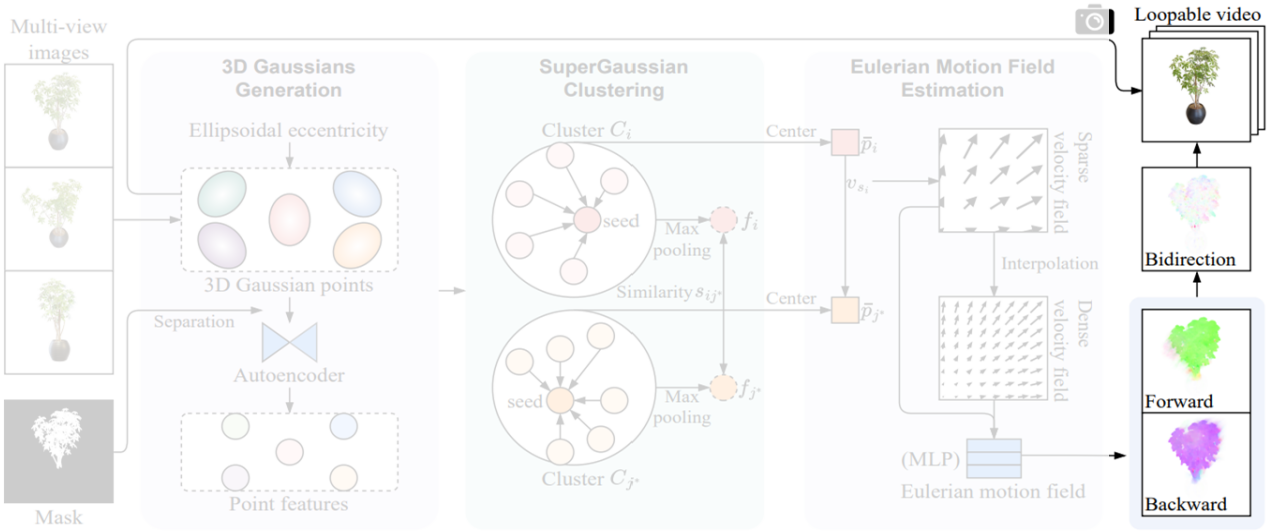


< LoopGaussian의 전체 Framework >

CVSP Lab
Computer Vision & Signal Processing

Method

- Overview
  - Loopable Dynamics Generation
    - Gaussian point의 position을 t(time)에 대해 정의
    - 이 후 Bidirectional animation technology를 새로 정의한 position에 적용
    - 각 frame에 대해 렌더링하면 최종 looping video를 형성



< LoopGaussian의 전체 Framework >

CVSP Lab
Computer Vision & Signal Processing

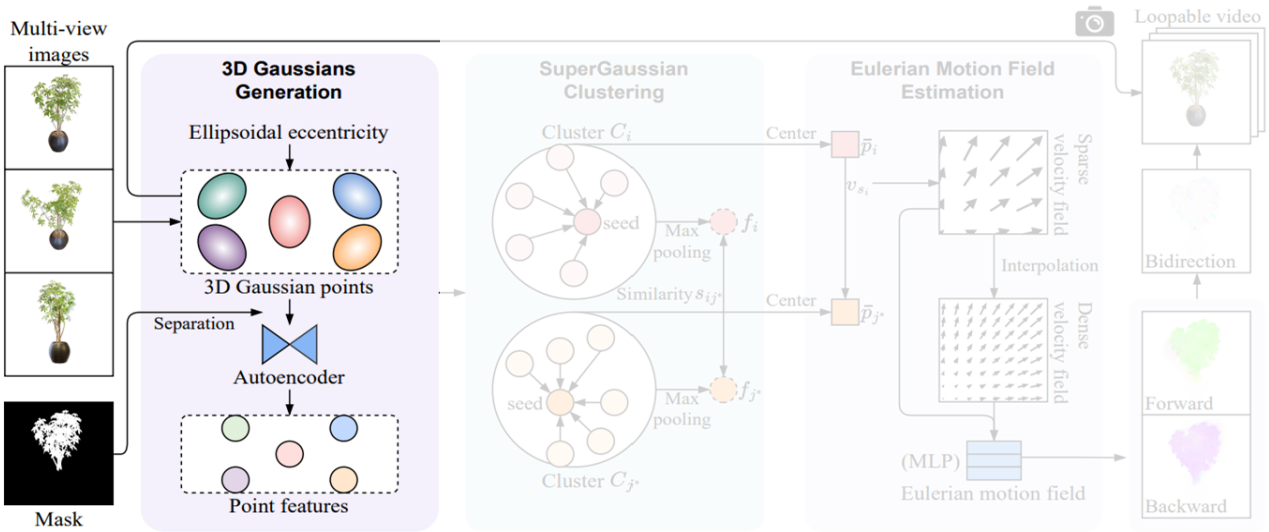
Method

- 3D Gaussian Generation
  - 3D Gaussian Generation and 3D Gaussian Optimization
    - Static Multi-view 이미지들을 가지고 3D Gaussian point cloud를 만들어 냄
    - 3D Gaussian의 타원 모양을 정규화하는 loss term을 통해 artifact 문제를 해결

$$L_{shape} = \frac{1}{|G|}\sum_{G_i \in G} 1 - \frac{\min^2(s_i)}{\max^2(s_i)}$$

$$L_{3D-GS} = \eta((1-\beta)L_1 + \beta L_{D-SSIM}) + (1-\eta)\,L_{shape}$$

✓ G: Gaussian point

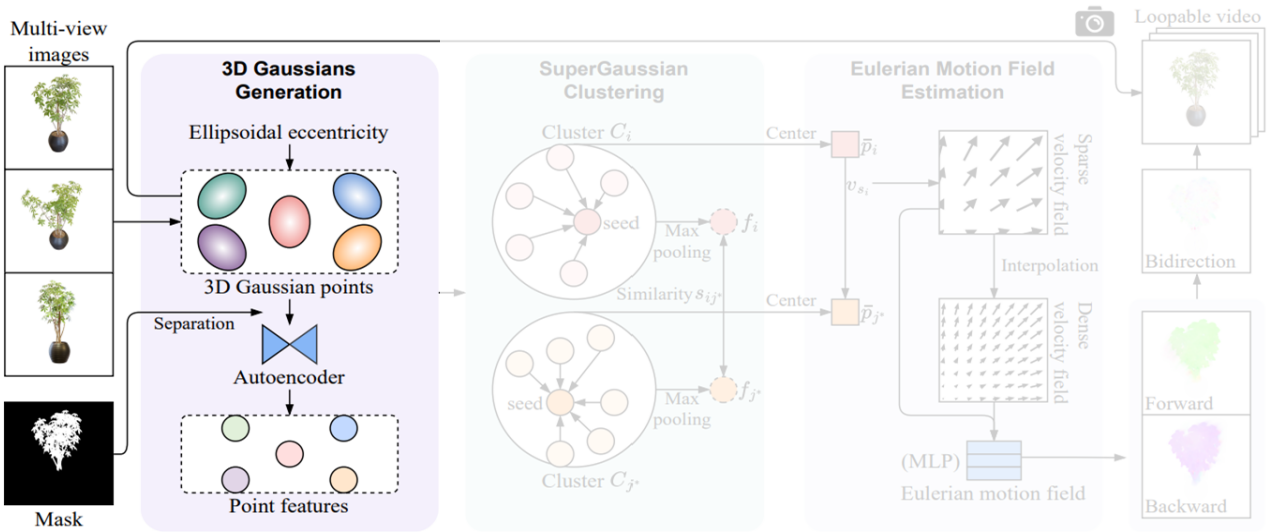


< LoopGaussian의 전체 Framework >

CVSP Lab
Computer Vision & Signal Processing

Method

- 3D Gaussian Generation
  - 3D Gaussian Separation
    - SAGA를 통해 3D Gaussian을 동적요소와 정적요소로 분류함
      - $\mathbf{M} := \{m_i\}$ = SAGA($\mathbf{G}$, I, A)
        - ✓ G: Gaussian point
        - ✓ I: Multi-view images of the training set
        - ✓ A: Manual Annotation

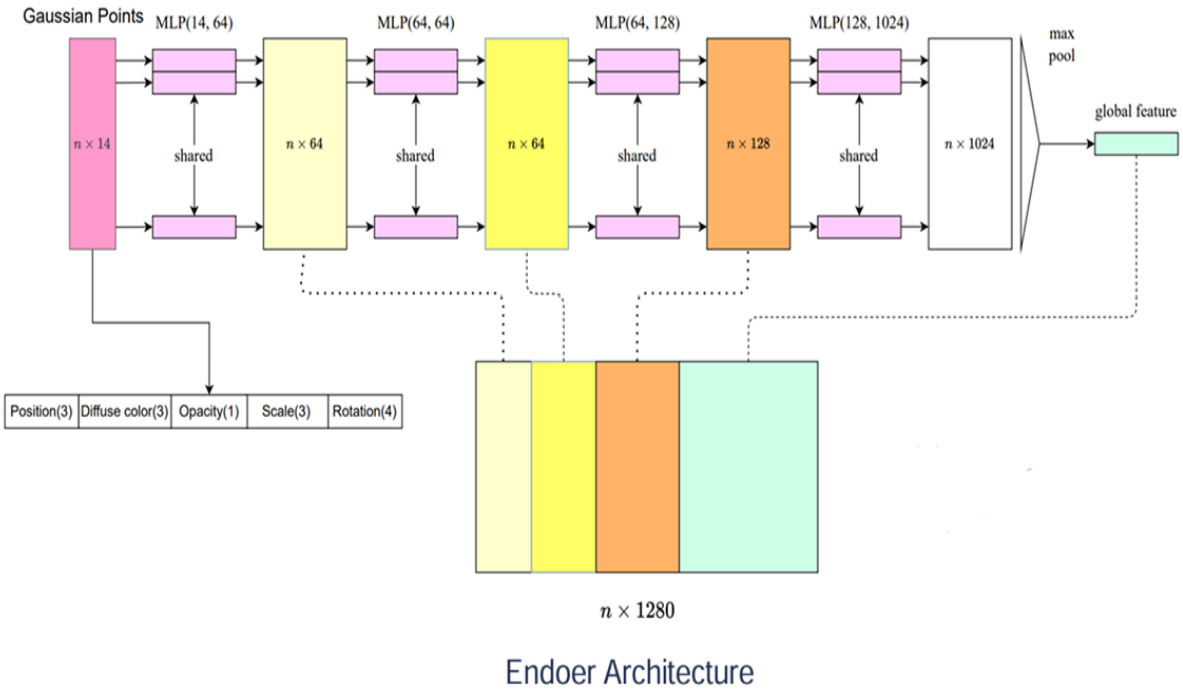


< LoopGaussian의 전체 Framework >

CVSP Lab
Computer Vision & Signal Processing

Method

- 3D Gaussian Generation
  - Autoencoder
    - 3D Gaussian의 Attribute들을 모두 넣어 n × 14 행렬이 입력이 됨
    - 중간 layer들의 출력 feature와 최종 출력 feature를 maxpooling한 global feature를 concat
    - Decoder는 Encoder와 대칭구조

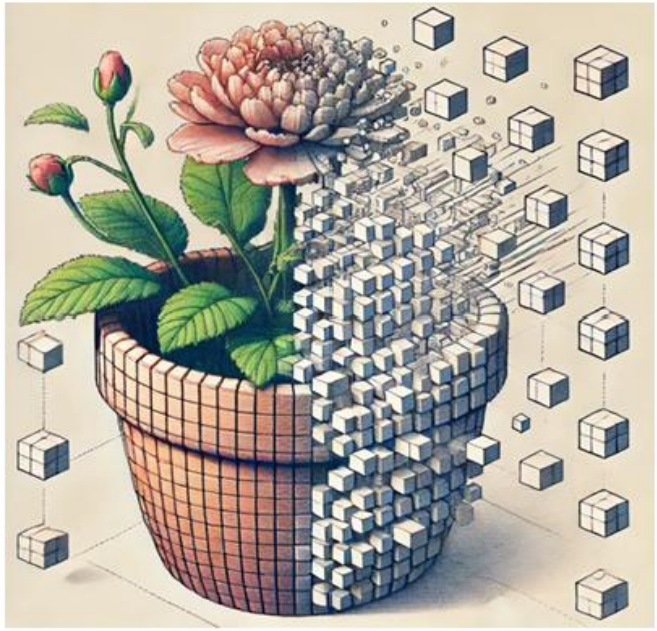


< Autoencoder의 Architecture 모습 >

CVSP Lab
Computer Vision & Signal Processing

Method

- SuperGaussian Clustering
  - Voxelization
    - 3D 공간을 voxel단위로 쪼갬
    - Voxel resoultion(R)을 변수로 두고 clustering 과정에서 metric function에 쓰이게 됨
    - voxel 내 3D Gaussian 중 1개를 랜덤하게 seed로 선정



< Voxelization의 형상화 그림 >

CVSP Lab
Computer Vision & Signal Processing

Method

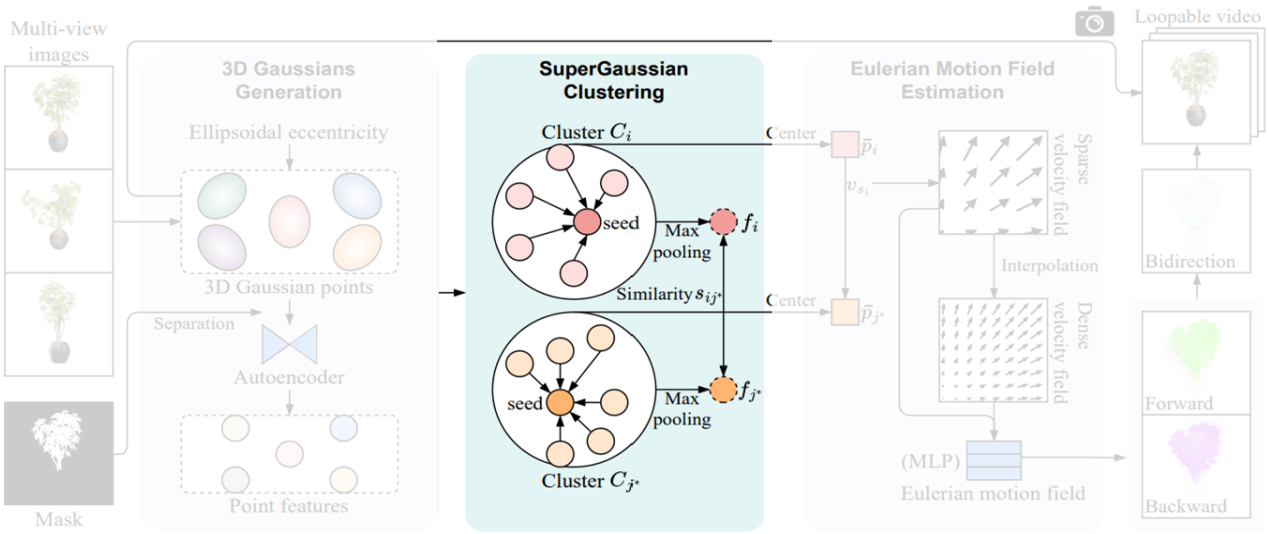
- SuperGaussian Clustering
  - SuperGaussian
    - 3D Gaussian들의 position과 feature 기반으로 distance가 더 작은 seed로 clustering

$$SG^* = \underset{SG}{argmin} \sum_{k=1}^{K} \sum_{SG(G_i)=k} D(G_i, G_{k'})$$

$$D(G_i, G_j) = 1 - \frac{|f_i \cdot f_j|}{\|f_i\| \cdot \|f_j\|} + \mu \frac{\|p_i - p_j\|}{R}$$

✓ R: voxel의 resolution

    - 각 Cluster에 대해 max pooling으로 global feature를 뽑아냄



< LoopGaussian의 전체 Framework >

CVSP Lab
Computer Vision & Signal Processing

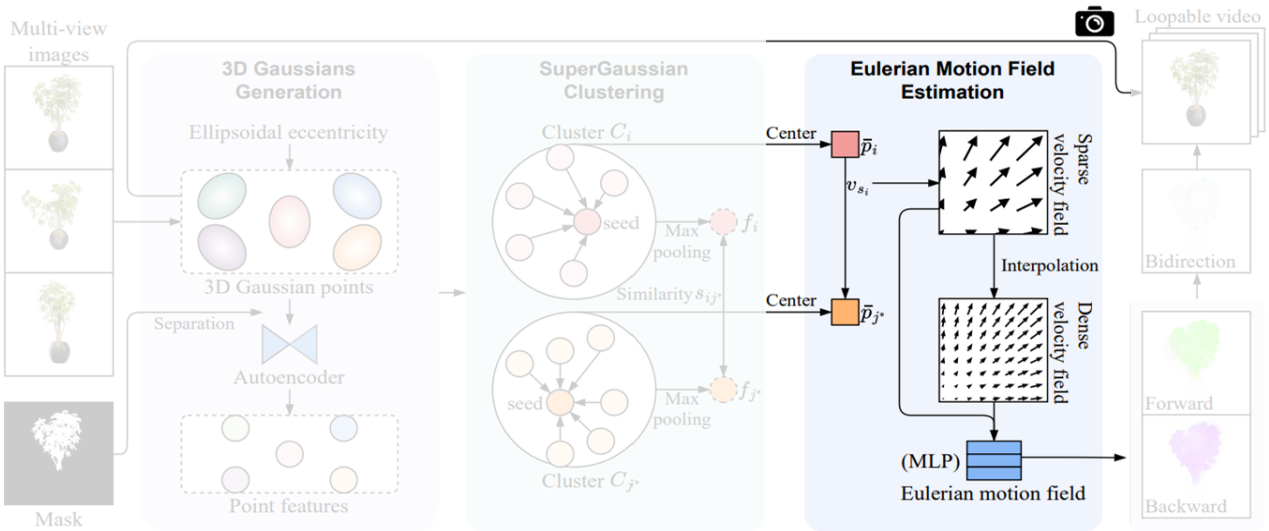
Method

- Eulerian Motion Field Estimation
  - Sparse Velocity Field Estimation
    - Global feature간의 가장 큰 유사도를 보이는 두 cluster를 구하고, 두 cluster 각각의 cluster 내부 모든 Gaussian point들의 Center position을 뽑아냄

$$s_{ij} = \frac{|f_i \cdot f_j|}{\|f_i\| \cdot \|f_j\|} \quad \forall_i \in [1, K],\ j \in [1, K]$$

✓ s_{ij}: f_i와 f_j 사이의 cosine similarity

✓ f_i, f_j: C_i와 C_j의 feature



< LoopGaussian의 전체 Framework >

CVSP Lab
Computer Vision & Signal Processing

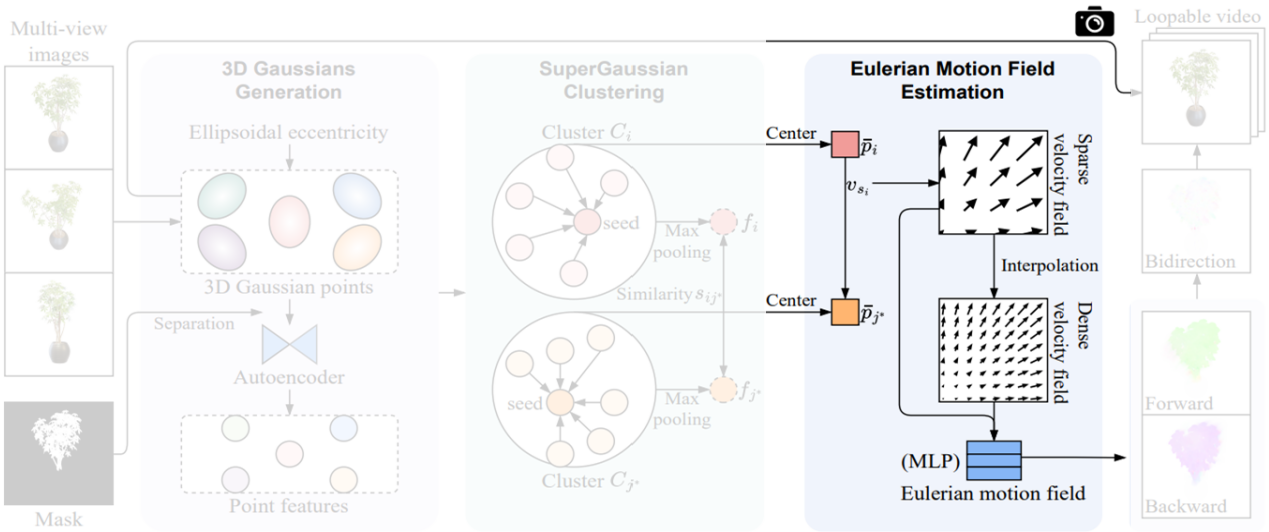
Method

- Eulerian Motion Field Estimation
  - Sparse Velocity Field Estimation
    - v_{sparse}를 Center position의 차로 정의

$$v_{s_i} = \bar{p}_{j^*} - \bar{p}_i \quad \forall_i \in [1, K]$$

✓ $\bar{p}_{j^*}$, $\bar{p}_i$: C_{j^*}, C_i 의 center position

✓ K: Cluster C의 총 개수

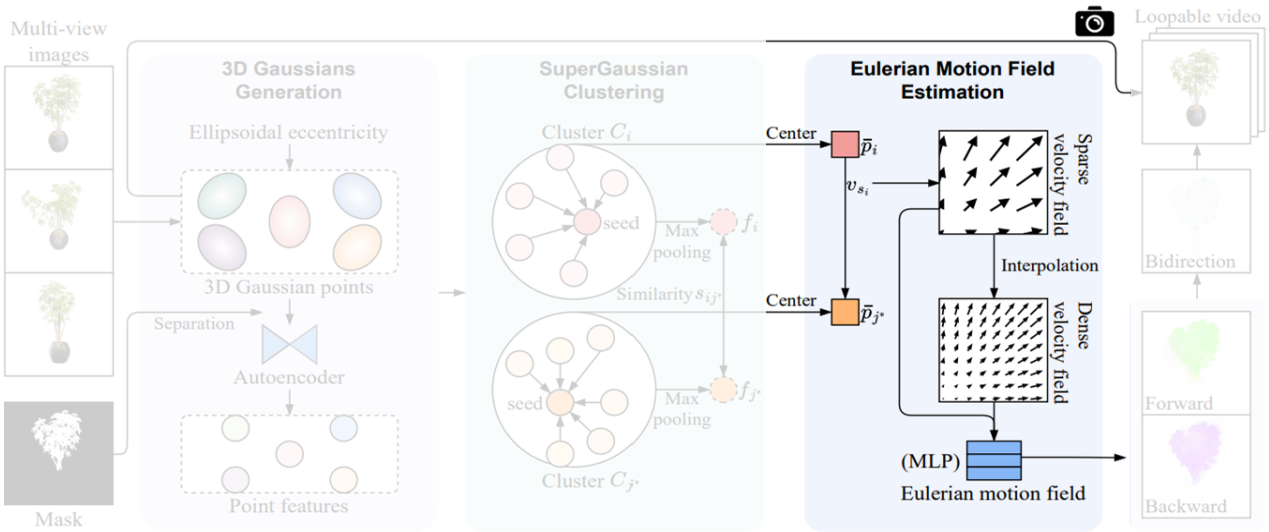


< LoopGaussian의 전체 Framework >

CVSP Lab
Computer Vision & Signal Processing

Method

- Eulerian Motion Field Estimation
  - Dense Velocity Field Estimation
    - v_{sparse} 에서 Kriging interpolation을 통해 Gaussian point의 v_{dense} 를 구함
      - $v_{dense} := \{v_{d_i}\}_{I=1}^{N}$
      - Kriging interpolation: 공간적 상관성을 고려하여 값을 추정하는 통계 기반 보간법
    - 이를 통해 모든 Gaussian point의 position에서 속도를 정의할 수 있게 됨



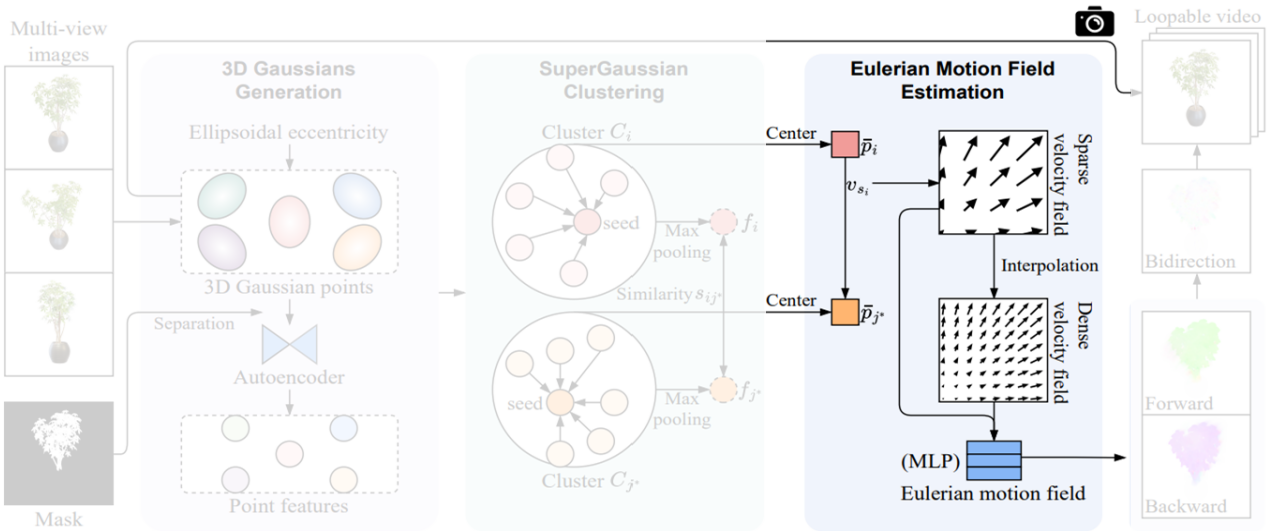
< LoopGaussian의 전체 Framework >

CVSP Lab
Computer Vision & Signal Processing

Method

- Eulerian Motion Field Estimation
  - Eulerian Motion Field Estimation
    - MLP의 input으로 각 cluster의 Center position과 모든 Gaussian point들의 position을 넣음
    - Center position에서의 정답은 Sparse Velocity Field가 되고, Gaussian point position에서의 정답은 Dense Velocity Field가 됨
    - 최종적으로 MLP가 3D 공간 내 모든 position에서 정의되는 Eulerian Motion을 추정할 수 있음

$$v = \vec{E}_G(p) := \mathrm{MLP}(p)$$



< LoopGaussian의 전체 Framework >

CVSP Lab
Computer Vision & Signal Processing

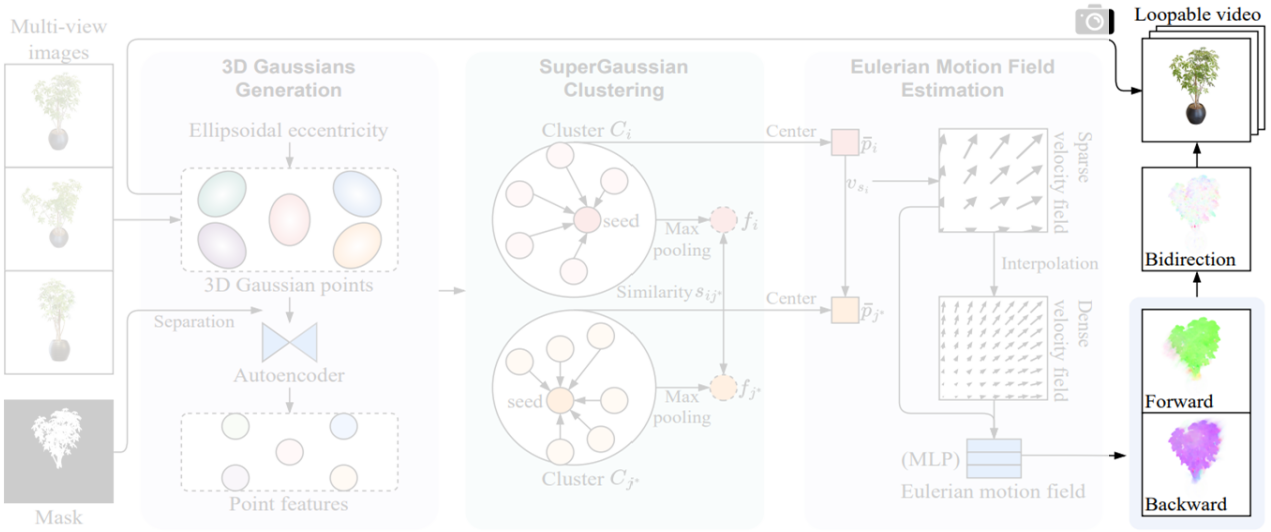
Method

- Loopable Dynamics Generation
  - Bidirectional animation technology
    - Eulerian Motion Field를 이용하여 특정 frame에서의 Gaussian point의 position을 추정

$$p_i(t) = p_i(0) + \sum_{\tau=0}^{t-1} \psi \odot \vec{E}_G(p_i(\tau))$$

✓ ψ: 움직임 크기를 제어하는 하이퍼파라미터

✓ $\odot$: Hadamard product(원소별 곱)

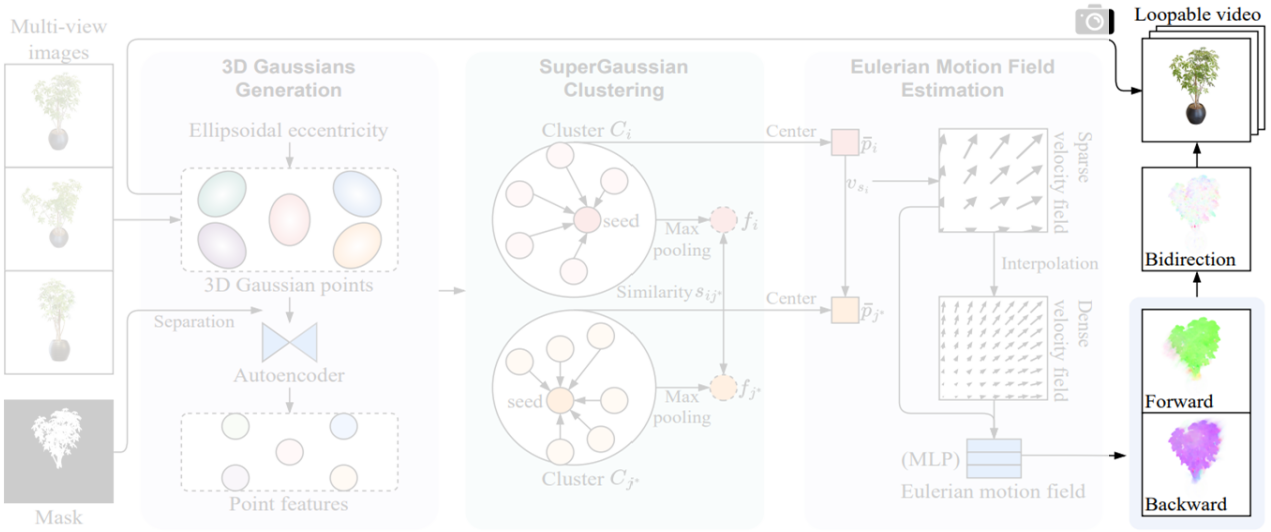


< LoopGaussian의 전체 Framework >

CVSP Lab
Computer Vision & Signal Processing

Method

- Loopable Dynamics Generation
  - Bidirectional animation technology
    - 추정한 position을 토대로 정방향과 역방향의 position을 구한 후, 그 둘을 선형보간 한 position 생성
      - $\hat{p}_i(t) = \alpha p_i(t) + (1-\alpha)p_i(t-T)$
        - ✓ T: 전체 프레임
    - 최종 생성된 position을 통해 Camera parameter에 따라 각 frame에서 렌더링하면 loopable video 생성



< LoopGaussian의 전체 Framework >

CVSP Lab
Computer Vision & Signal Processing

Experiment

- Dataset
  - 일부는 NeRF synthetic dataset 이용, 나머지는 자체 제작
    - 이미지
      - Test: 200개의 뷰를 가진 multi-view 이미지 데이터셋
      - Train: 100개의 뷰를 가진 multi-view 이미지 데이터셋
      - Validation: 100개의 뷰를 가진 multi-view 이미지 데이터셋
    - GT 비디오
      - 해상도: 900 x 900 픽셀
      - 프레임 수: 48 프레임





Experiment

- Implementation details
  - 실험 환경
    - GPU: NVIDIA RTX 4090 1개
    - Framework: PyTorch
  - Hyperparameter
    - β = 0.2, η = 0.9로 설정
      - $L_{3D-GS} = \eta((1-\beta)L_1 + \beta L_{D-SSIM}) + (1-\eta)\, L_{shape}$
    - λ = 0.04, μ = 0.5로 설정
      - $D(G_i, G_j) = 1 - \frac{|f_i \cdot f_j|}{\|f_i\| \cdot \|f_j\|} + \mu \frac{\|p_i - p_j\|}{R}$
        - ✓ λ는 R(장면 해상도)을 조절하는 하이퍼파라미터
    - ω = 1.2로 설정
      - $p_i(t) = p_i(0) + \sum_{\tau=0}^{t-1} \psi \odot \vec{E}_G(p_i(\tau))$
        - ✓ ω는 ψ(움직임 크기를 제어)를 조절하는 하이퍼파라미터

Experiment

- Results
  - Quantitative evaluation
    - Optical flow에 대한 이미지 품질 평가
      - 2번째 프레임부터 Optical flow 계산
      - 각 프레임 ↔ 이전 프레임의 flow 계산
      - 평균을 내어 종합 Optical flow map 획득
    - 최종 비디오에 대한 비디오 품질 평가
      - FVD: FID를 기반으로 하여 비디오의 품질과 다양성을 측정하는 데 사용
        - ✓ FID: 두 이미지의 feature vector의 distribution이 얼마나 유사한지를 distance로 나타내는 지표

Methods	PSNR↑	SSIM↑	LPIPS ↓
Li [19]	22.959	0.915	0.233
Ours	**24.868**	**0.928**	**0.208**

< Quantitative comparison of average optical flowmaps >

Methods	FVD ↓
Li [19]	1174.948
Ours	**933.824**

< Quantitative comparison of generated videos >

Experiment

- Results
  - Qualitative evaluation
    - Baseline인 Li et al. (2023) “ 3D-Cinemagraphy” 과 비교



< 3D Cinemagraph(좌), LoopGaussian(우) >

CVSP Lab
Computer Vision & Signal Processing

Conclusion

- Contribution paper
  - 3D-MOM은 유체에 대한 motion 잘 추정하지만, 다른 객체들에 대한 motion은 잘 추정하지 못함
    - 배도 유체처럼 늘어지는 실험 결과를 볼 수 있음



< 바다 위의 배 이미지에 대한 3D-MOM의 output >

CVSP Lab
Computer Vision & Signal Processing

Conclusion

- Contribution paper
  - 3D-MOM의 motion 추정 부분을 Diffusion model 기반의 motion 추정 모델로 대체하여 실험 진행
    - 바다 위의 배처럼 유체가 아닌 객체에서도 자연스러운 motion을 추정하는 효과를 기대

감사합니다

CVSP Lab
Computer Vision & Signal Processing