Per-scene 4D Gaussian Reconstruction

2026 VDSL 겨울 세미나

Sogang University
Dept. of Electronic Engineering

Presented By
염수웅

Contents

- Background
- SharpTimeGS: Sharp and Stable Dynamic Gaussian Splatting via Lifespan Modulation [Arxiv 26]
- Shape of Motion: 4D Reconstruction from Single Video [ICCV 25]

Background

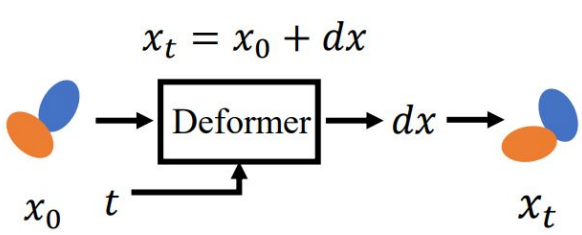
- Per-scene optimization task
  - Using Multi-View
    - Rig 형태의 동기화된 카메라들로 촬영된 다수의 비디오를 이용하여 시공간 복원
      - Multi-view consistency를 최대한 활용하는 방안으로 연구 진행
        - ✔ Photometric loss만으로도 동적 장면 복원 용이
          - View 개수가 많을 수록 정밀한 복원 가능
  - Using Monocular View
    - Hand-held 카메라로 촬영된 단안 카메라 비디오를 사용하여 시공간 복원
      - 각 time step 별 참조 view가 하나 뿐이라 복원 난이도가 높음
        - ✔ Prior를 사용하여 사전 3D 복원 및 tracking 진행 후 복원하는 방안으로 연구 진행
          - Video depth, metric depth, 3D tracking 등을 사용하여 사전에 anchor를 만듬

Multi-view 카메라 세팅

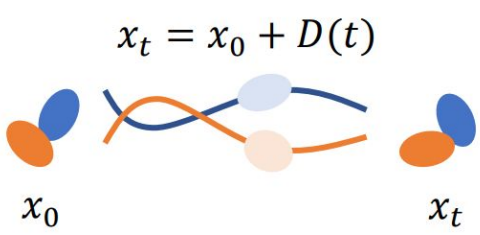
Monocular 카메라 세팅

Background

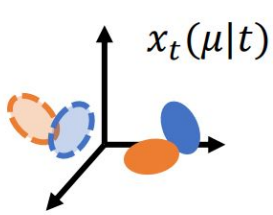
- 4D Gaussian Splatting 연구 동향
  - Deformation fields 기반 implicit 방법론
    - Canonical space를 잘 정의하여 각 프레임에 해당하는 time으로의 deformation field 학습
      - MLP 및 grid feature를 활용하여 변형되는 offset vector 학습
  - Explicit motion function 기반 방법론
    - 명시적으로 표현 가능한 함수를 사용하여 각 시간 별 motion 학습
      - Motion은 함수로 정의되며 Gaussian의 attribute를 업데이트
        - ✔ 함수가 각 장면에 맞는 움직임을 표현할 수 있도록 중심점 이동
  - 4D attribute 기반 방법론
    - 4D distribution을 따르는 Gaussian attribute를 정의하여 학습
      - 3D Gaussian에 시간축을 추가하여 4D Gaussian으로 확장



Deformation Fields 기반



Explicit motion function 기반

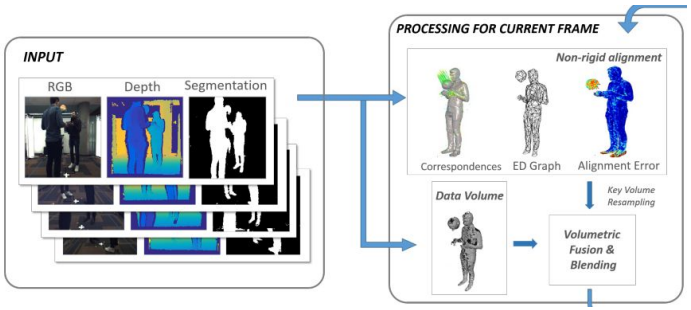


4D attribute 기반

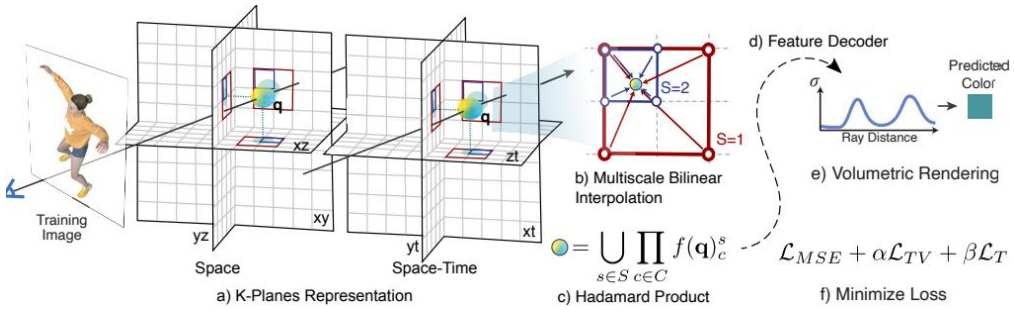
SharpTimeGS [Arxiv 26]

Introduction

- 논문에서 다루고자하는 task
  - Long-term static 영역과 short term dynamic 영역의 균형을 이루는 장면 최적화
- Gaussian Splatting 이전 연구 및 문제점
  - 고전 기하학 정보 기반 복원 방법론
    - Multi-view depth를 이용한 프레임별 point cloud reconstruction
      - ;;철저하게 view 개수에 의존성이 강하며 temporal consistency를 가지기 힘듬
  - NeRF 기반 방법론
    - Static과 dynamic을 ray 상에서 분리하며 독립된 plane feature 를 학습
      - ;;복원 성능이 좋지만 실시간 렌더링이 불가능하며 학습시간이 매우 오래 걸림
        - ✔픽셀별 ray의 volume rendering을 수행함에 따라 계산량이 큼



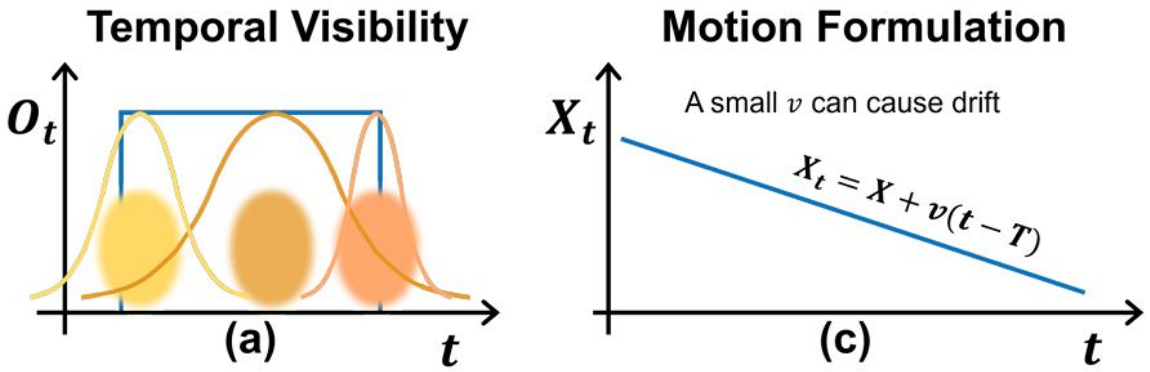
기하학 정보 기반 복원



NeRF 기반 복원

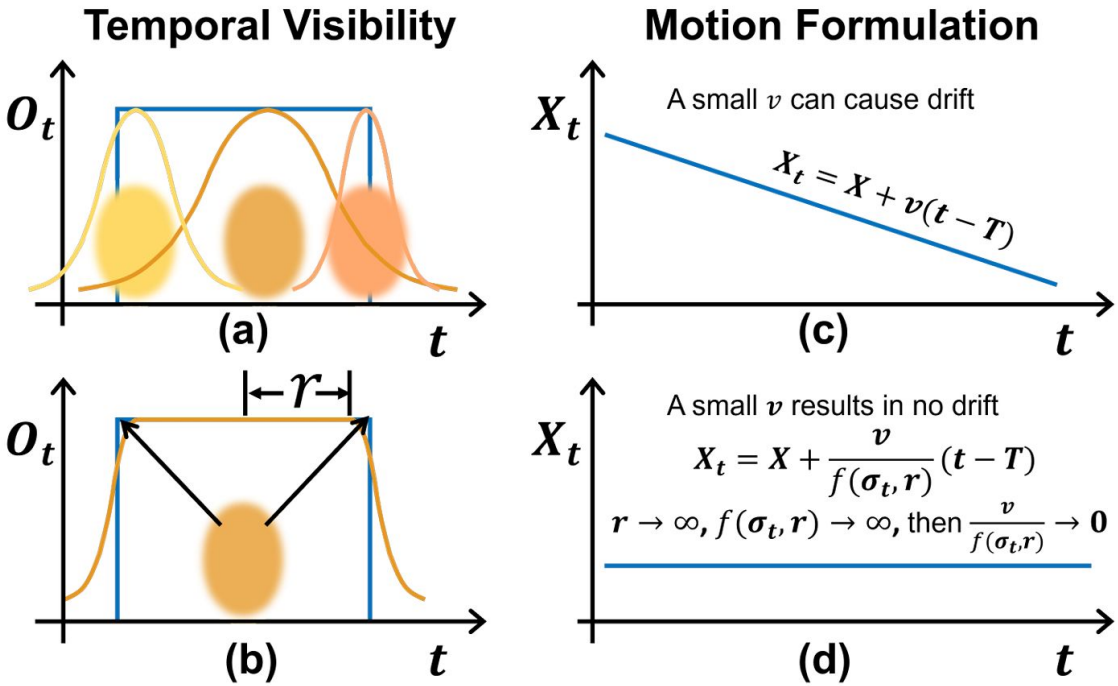
Introduction

- Gaussian Splatting 기반 연구 및 문제점
  - Canonical space 기반 deformation fields 학습 방법
    - 모든 시간의 장면을 대표하는 canonical space 정의 후 각 tine stamp로의 deformation 학습
      - Canonical space와 target space의 deformation 차이가 클수록 최적화에 어려움
  - Motion function 기반 모델링 방법
    - 시간에 따른 motion의 변화를 명시적으로 표기함으로써 4D scene 표현
      - 각 움직임이 유효한 duration을 life-span 모델링을 통해 표현
      - Static과 dynamic이 구분되는 장면에서의 최적화 어려움 발생
        - ✔ Static에 불필요하게 Gaussian attribute 증식
        - ✔ 속도가 아무리 작아도 jitter와 같은 형태로 정적 배경을 표현하게 되어짐



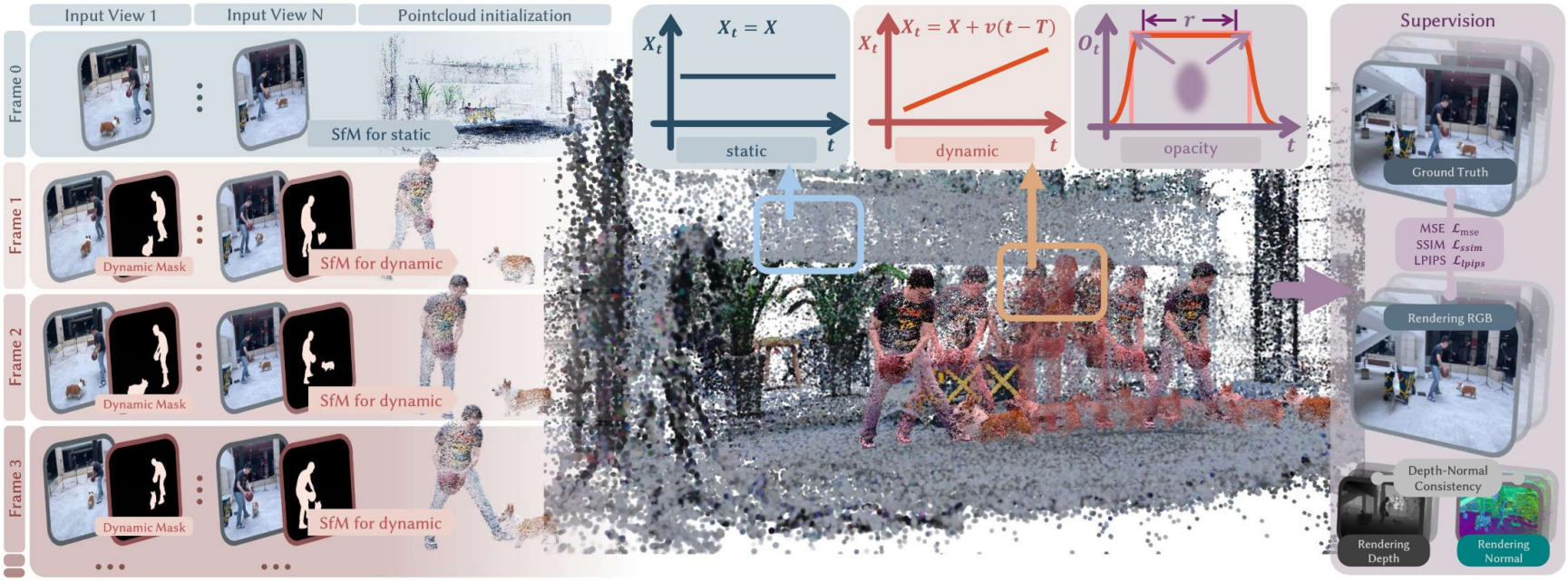
Introduction

- 문제 해결을 위해 논문에서 제안하는 방법
  - Life-sapn에 learnable parameter를 추가
    - Life-span을 flat한 형태로 제어 함으로써 정적 영역의 Gaussian 증식을 억제
    - Motion fuction에 life-span 파라미터를 적용함으로써 속도와 수명의 관계를 모델링
      - 정적 영역의 Gaussian이 jitter없이 고정되도록 유도



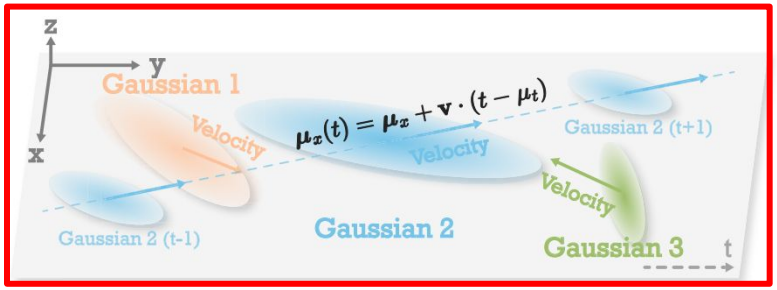
Method

- Overall Pipeline



- Velocity aware initialization
- Life-span modulated 4D Gaussian representation
- Velocity-lifespan-aware densification

Method



- 4D Gaussians with Lifespan Modulation
  - Life-span modulated motion dyanmics
    - Life span parameter를 정의 후 defomation function에 적용

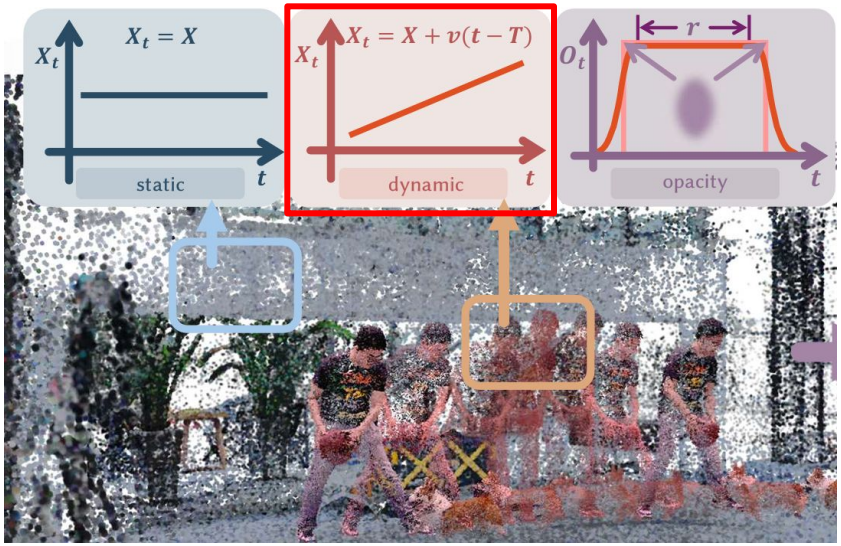
$$X_t = X + \frac{v}{f(\sigma_t, r)}(t - T)$$

$$f(\sigma_t, r) = 1.0 + \max\{1.0, (\sigma_t + r)^2\}$$

- Life-span modulated temporal visibility

$$O_t = O \cdot l(t)$$

$$l(t) = \begin{cases} \exp\left(-\left(\frac{|t-T|-r}{\sigma_t}\right)^2\right), & |t - T| > r \\ 1, & |t - T| \le r \end{cases}$$



X_t : time stamp t에서의 Gaussian 중심점
v: 각 Gaussian별 속도 attribute
T: 각 Gaussian 별 canonical time

σ_t: 각 Gaussian 별 canonical time의 variance
r: 각 Gaussian 별 temporal radius
O_t : time stamp t에서의 Gaussian opacity
O: 각 Gaussian별 canonical opacity

Method

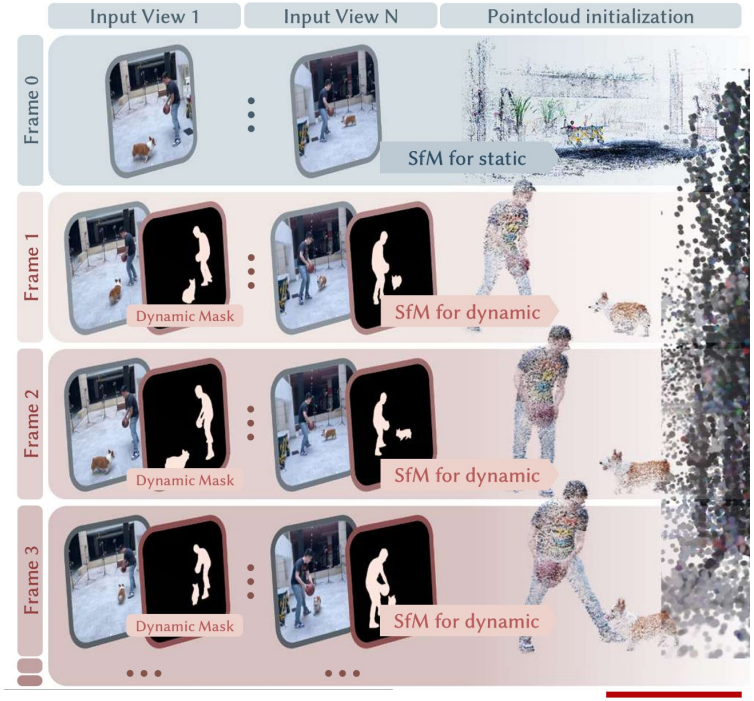
- Velocity-lifespan-aware densification
  - 동적 객체의 학습량 부족에 따른 재구성 품질 저하 방지를 위한 densification
    - 정적 영역은 매 time stamp마다 존재하기 때문에 학습이 자주 이루어짐
    - 동적 객체의 경우 각 time stamp에서 한번씩만 등장하기 때문에 학습이 부족함
    - 움직임의 속도와 temporal consistency를 바탕으로 densification 수행
      - 초기 학습의 1/3 동안은 AbsGS와 동일한 densification 수행
        - ✓3DGS보다 효과적으로 복제함으로써 충분한 양의 Gaussian을 생성
      - 이후 2/3 동안 각 canonical time의 Gaussian을 고정 후 시공간적 분포를 정교화
        - ✓$s = \lambda_e E + \lambda_o O + \lambda_l (1 - \exp(-\frac{\|v+1\|}{f(\sigma_t, r)}))$
        - ✓렌더링 퀄리티가 낮은 time에 대한 densification
        - ✓Opacity가 높은 영역에 대하여 densification
        - ✓수명이 짧고 빠르게 움직이는 Gaussian에 높은 score를 부여하여 densification
          - 주로 물체가 빠르게 움직이는 구간에서 품질저하가 발생하기 때문에 필요

E: GT이미지와 렌더링 이미지의 오차
O: Gaussian opacity

$\|v+1\|$: 값이 커질수록 s가 커짐
$f(\sigma_t, r)$: 값이 작을수록 s가 커짐

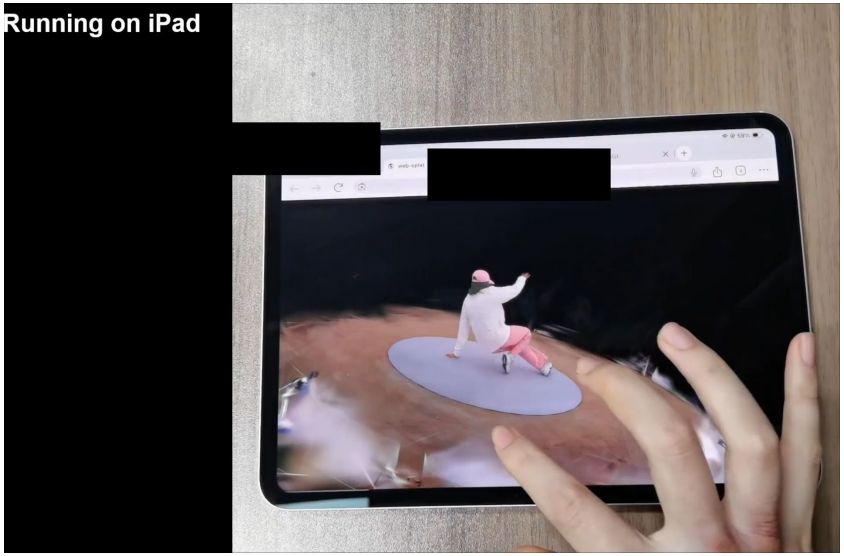
Method

- Velocity-aware initialization
  - Optical flow와 SAM을 이용한 dynamic mask 획득
    - Optical flow를 이용하여 움직이는 구간을 구분 후 key point를 생성하여 SAM2 구동
  - 전체 프레임에서의 COLMAP 구동
    - 모든 프레임에서 COLMAP을 개별적으로 구동하여 각 time의 point cloud 획득
      - 마스크와 카메라 파라미터를 이용하여 정적인 영역과 동적인 영역의 point 분리
  - 속도 초기화
    - 인접한 프레임 point 사이의 knn 구동
      - 가까운 점과의 거리를 구한 후 time으로 나눔
        - ✔ FreeTimeGS와 동일하게 수행
        - ✔ Time은 프레임의 time stamp로 초기화
  - Life-span 초기화
    - 정적 영역에 대해서는 전체 프레임을 커버
    - 동적 영역은 세 프레임을 커버하도록 설정



Experiments

- Video demo results



Experiments

- Quantitative results

Table 1. Quantitative comparison on Neural3DV [19] Dataset, ENeRF-Outdoor [22] Dataset, and SelfCap [46] Dataset. We report PSNR, $SSIM_2$ [42], and LPIPS [50] to evaluate the rendering quality. Values in boldface denote the best result in the corresponding column.

Method	Neural3DV			ENeRF-Outdoor			SelfCap		
	PSNR ↑	$SSIM_2$ ↑	LPIPS ↓	PSNR ↑	$SSIM_2$ ↑	LPIPS ↓	PSNR ↑	$SSIM_2$ ↑	LPIPS ↓
Deformable-3DGS [47]	31.15	0.970	0.049	24.26	0.801	0.318	25.85	0.920	0.312
Ex4DGS [18]	32.11	0.970	0.048	24.89	0.817	0.305	24.96	0.920	0.299
4DGS [48]	32.01	0.972	0.055	24.82	0.822	0.317	25.86	0.923	0.245
STGS [20]	32.05	0.972	0.044	24.93	0.818	0.297	24.77	0.894	0.291
FreeTimeGS [41]	33.19	0.974	0.036	25.36	0.846	0.244	27.50	0.951	0.201
Ours	**33.57**	**0.977**	**0.031**	**25.82**	**0.872**	**0.233**	**28.14**	**0.960**	**0.192**

- Ablation study

Table 2. Ablation study on SelfCap [46] Dataset (Partial). We report PSNR, $SSIM_2$, and LPIPS to evaluate the rendering quality.

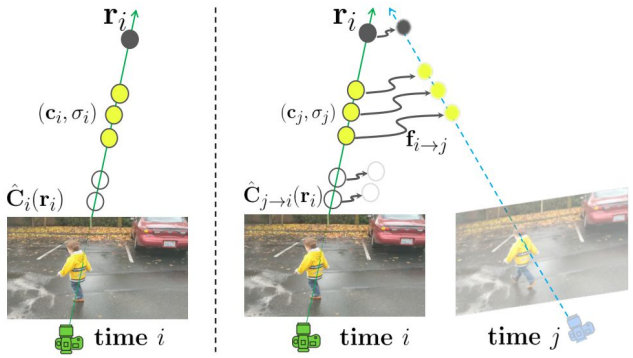
4DGS 사용 결과

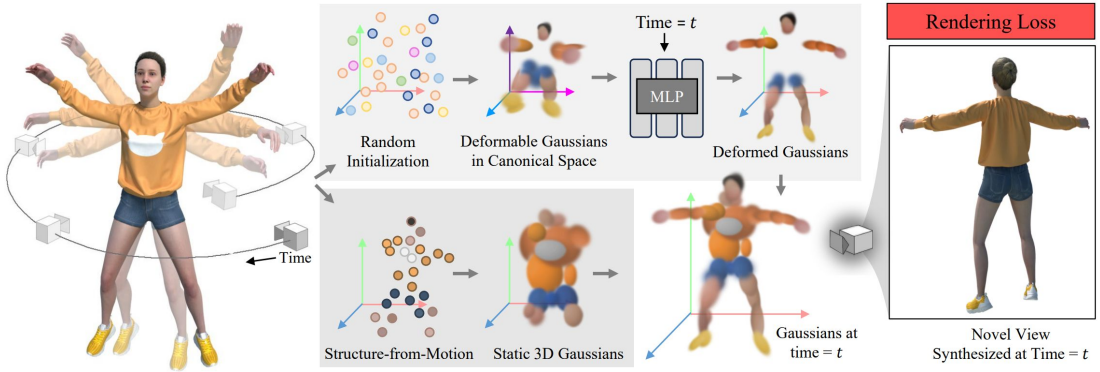
Method	PSNR ↑	$SSIM_2$ ↑	LPIPS ↓
w/o our representation	25.96	0.907	0.299
w/o lifespan r	26.76	0.927	0.321
w/o our densification	26.82	0.919	0.317
w/o our initialization	26.83	0.927	0.297
full model	**27.36**	**0.947**	**0.244**

Shape of Motion [ICCV 25]

Introduction

- 논문에서 다루고자하는 task
  - Monocular scene에서 3D tracking이 수행될 수 있는 explicit 4D Gaussian 모델 개발
- Scene flow 기반 방법 및 문제점
  - 인접한 프레임 사이의 scene flow를 모델링하여 전체 4D scene을 표현
    - 단기 장면의 흐름을 모델링하기 때문에 비디오 전체에 지속적인 움직임 복원에 한계
- Implicit deformation fields 기반 방법 및 문제점
  - Canonical space를 만들고 각 time stamp로의 offset vector를 학습
    - Implicit neural fields를 활용하며 view가 한적적인 monocular에서 특히 수렴이 어려움
      - 간단한 scene 및 잘 세팅되어진 장면이 아닌 casual scene에서의 한계점이 명확함



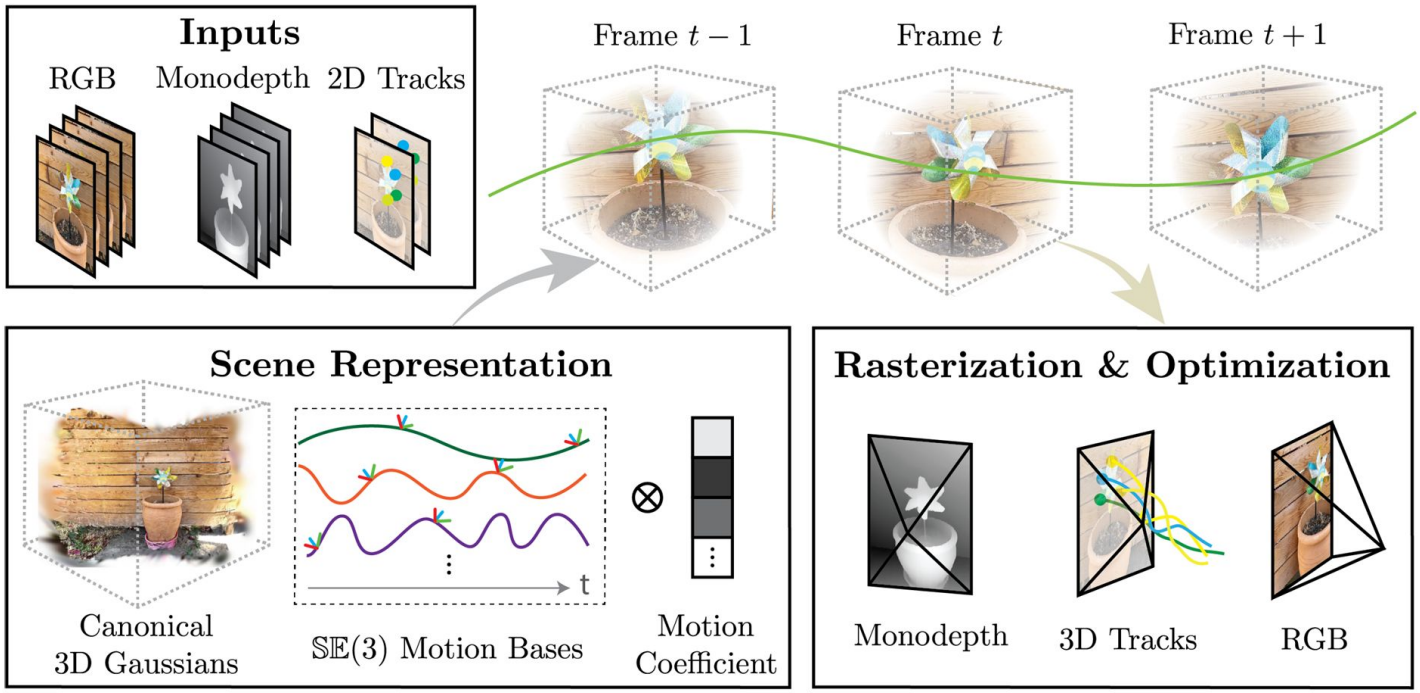


Introduction

- 문제 해결을 위해 논문에서 제안하는 방법
  - 물체의 움직임을 explicit한 강체 운동으로 표현
    - 이미지 공간에서 움직임이 아무리 복잡하더라도 그 기저는 강체 운동의 결합으로 표현
      - 강체의 운동은 물체의 모양이 변하지 않고 위치와 방향이 바뀌는 운동
      - 복잡한 움직임도 결국 쪼개어 보면 단순한 물리 법칙을 따르는 3D 운동의 합임
  - Monocular가 가지는 view 부족 한계를 극복하기 위한 prior 활용
    - Monocular depth와 2D tracking 을 이용하여 scene을 초기화
      - Monocular 4D scene의 경우 장면별 최적화에 필요한 contraint가 매우 부족
      - 부족한 constraint를 data-driven prior 활용하여 보완해야함
        - ✔ 기하학적 구조와 일관된 motion을 표현하는데 있어 큰 단서를 제공
          - 비록 노이즈를 포함할지라도 그 이상의 가치있는 표현 정보를 제공함

Method

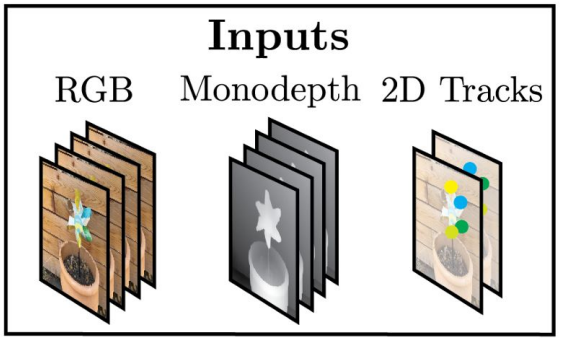
- Overall Pipeline



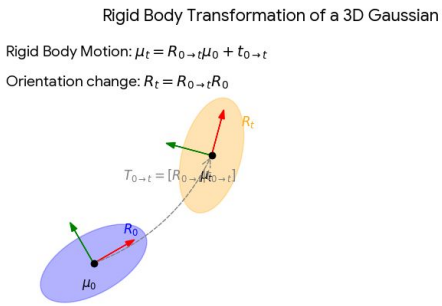
- Initialization using depth and 2D tracks
- Dynamic Scene representation
- Optimization

Method

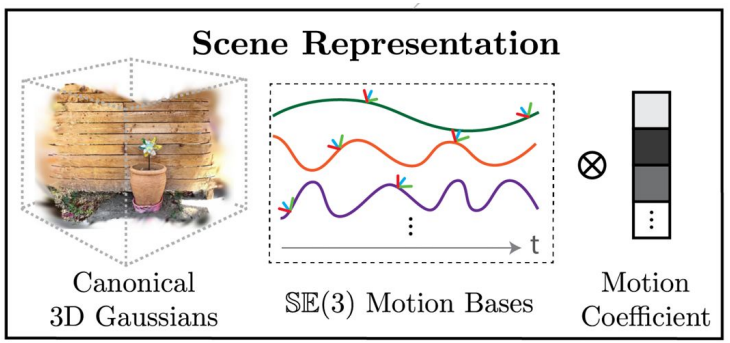
- Initialization using depth and 2D tracks
  - COLMAP 및 MegaSaM을 이용하여 카메라 파라미터 및 point cloud 획득
  - Depyh Anything을 이용하여 프레임별 relative depthmap 획득
    - Point cloud를 활용하여 획득된 depth map을 scaling하여 reconstruction에 활용
  - TAPIR을 활용하여 2D 이미지상에서 트래킹 수행
    - 트래킹된 2D 픽셀 좌표를 scalin된 depth map을 활용하여 3D reconstruction 수행
    - TrackAnyhting을 통해 획득된 motion mask를 이용하여 움직이는 물체만을 트래킹
  - Canonical sapce 초기화
    - 트래킹된 점들이 가장 많은 3D 공간을 canonical space로 초기화
      - Motion mask를 사용하여 static 영역 reconstruction 후 Gaussian 초기화
      - 트래킹된 3D point는 dynamic 영역으로 Gaussian 초기화



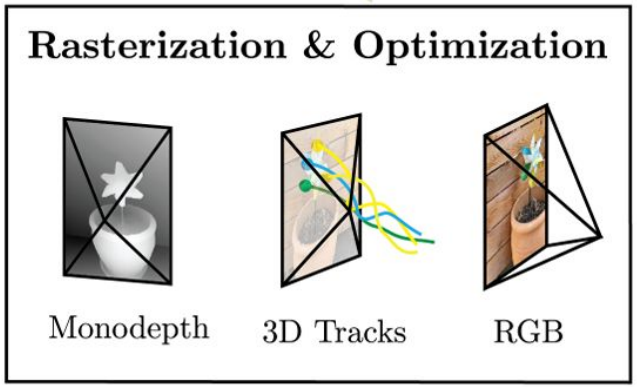
Method



- Dynamic scene representation
  - Rigid transformation을 이용한 움직임 모델링
    - 아무리 복잡한 움직임도 단순한 gigid motion의 결합으로 모두 표현할 수 있음을 가정
    - 시점 t에서의 Gaussian의 위치와 회전을 SE(3) 변환으로 모델링
      - $\mu_t = R_{0\to t}\mu_0 + t_{0\to t}, R_t = R_{0\to t}R_0$
      - Gaussian의 움직임에 적용된 회전 변환을 rotation 파라미터에도 동일하게 적용
        - ✓강체의 움직임을 강제
  - Motion bases를 활용한 transformation 계산
    - 각 Gaussian의 transformation은 전역적으로 공유되는 B개의 기제 궤적들의 합으로 표현
      - $T_{0\to t} = \sum_{b=1}^{B} \omega^{(b)} T_{0\to t}^{(b)}$
        - ✓$T_{0\to t}^{(b)}$: b 번째 기저 (모든 Gaussian이 공유)
        - ✓$\omega^{(b)}$: 각 Gaussian 고유의 motion coefficient



Method



- Optimization
  - Reconstruction loss 사용
    - $L_{recon} = \left\|\hat{I} - I\right\|_1 + \lambda_{depth}\left\|\hat{D} - D\right\|_1 + \lambda_{mask}\left\|\hat{M} - M\right\|_1$
      - Photometric loss와 depth loss를 사용하여 텍스처 및 기하학 정보 학습
      - Mask loss를 통해 dynamic 객체에 대한 움직임 간접 학습
  - Rasterization 3D Trajectories
    - ${}^{w}\hat{X}_{t\to t'}(p) = \sum_{i\in H(p)} T_i \alpha_i \mu_{i,t'}$
    - $\hat{U}_{t\to t'}(p) = \prod(K_t, {}^{c}\hat{X}_{t\to t'}(p))$
  - Tracking loss
    - $L_{track-2d} = \left\|U_{t\to t'} - \hat{U}_{t\to t'}\right\|_1$
    - $L_{track-depth} = \left\|\hat{d}_{t\to t'} - \hat{D}(U_{t\to t'})\right\|_1$
      - 동적 객체dp tracking loss를 통해 직접적으로 supervise함으로써 움직임 학습

Experiments

- Video demo results

Experiments

- Quantitative results

Method	3D Tracking			2D Tracking			View Synthesis		
	EPE ↓	$\delta^{.05}_{3D}$ ↑	$\delta^{.10}_{3D}$ ↑	AJ ↑	$<\delta_{\text{avg}}$ ↑	OA ↑	PSNR ↑	SSIM ↑	LPIPS ↓
T-NeRF [25]	-	-	-	-	-	-	15.60	0.55	0.55
HyperNeRF [77]	0.182	28.4	45.8	10.1	19.3	52.0	15.99	0.59	0.51
DynIBaR [60]	0.252	11.4	24.6	5.4	8.7	37.7	13.41	0.48	0.55
Deformable-3D-GS [122]	0.151	33.4	55.3	14.0	20.9	63.9	11.92	0.49	0.66
DynMF [53]	0.188	22.9	53.8	5.5	9.5	60.5	16.54	0.59	0.49
CoTracker [45]+DA [121]	0.202	34.3	57.9	24.1	33.9	73.0	-	-	-
TAPIR [16]+DA [121]	0.114	38.1	63.2	27.8	41.5	67.4	-	-	-
DELTA (world) [74]	0.159	32.5	55.3	24.7	34.1	68.9	-	-	-
SpatialTracker (world) [113]	0.125	37.7	63.9	24.9	36.9	73.5	-	-	-
Ours	**0.082**	43.0	**73.3**	34.4	**47.0**	86.6	16.72	0.63	0.45
Ours + 2DGS[34]	0.097	**47.3**	71.3	**35.8**	**47.0**	**87.3**	**16.75**	**0.65**	**0.40**

Table 1. **Evaluation on iPhone dataset**. Our method achieves SOTA performance all tasks of 3D point tracking, 2D point tracking, and novel view synthesis. The baselines that perform best on 2D and 3D tracking (TAPIR [16]+DA [121], CoTracker [45]+DA [121], DELTA [74], SpatialTracker [113]) are unable to synthesize novel views of the scene, while the methods that perform best in novel view synthesis struggle with or fail to produce 2D and 3D tracks. Our method achieves a significant boost in all three tasks above baselines. We include training details about "Ours + 2DGS [34]" in the supplement.

- Metric
  - EPE: GT와 예측된 3차원 사이의 거리
  - $\delta^{0.5}_{3D}$: 예측된 3D 지점이 실제 위치로부터 5cm 이내의 오차범위에 들어오는 포인트 비율
  - AJ(Average Jaccard) : 위치 정확도와 가시성을 동시에 계산하여 정확도 측정
  - $< \delta_{avg}$: 추적중인 포인트가 실제 정답 위치로 부터 얼마나 떨어져 있는지 평가

Experiments

- Ablation study

Methods	$\mathbb{SE}(3)$	Motion Basis	2D tracks	Initialization	EPE↓	$\delta_{3D}^{.05}$ ↑	$\delta_{3D}^{.10}$ ↑
Ours (Full)	✓	✓	✓	✓	**0.082**	43.0	**73.3**
Transl. Bases		✓	✓	✓	0.093	42.3	69.9
Per-Gaussian $\mathbb{SE}(3)$	✓		✓	✓	0.083	**43.6**	70.2
Per-Gaussian Transl.			✓	✓	0.087	41.2	69.2
No $\mathbb{SE}(3)$ Init.	✓	✓	✓		0.111	39.3	65.7
No 2D Tracks	✓	✓			0.141	30.4	57.8

Table 4. **Ablation Studies on iPhone dataset.**

- Off-the-shelf 방법들을 통한 initial이 가장 중요한 역할 수행
  - Initialization이 없을 때 트래킹 성능이 급격하게 감소
    - 4D Gaussian Splatting에서 intial이 얼마나 중요한 역할을 하는지 확인 가능
      - ✔ 특히 constraint가 부족한 monocular scene에서 그 효과가 부각되어짐