4D Reconstruction

2025 VDSL 여름 세미나_염수웅

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

Presented By
염수웅

Outline

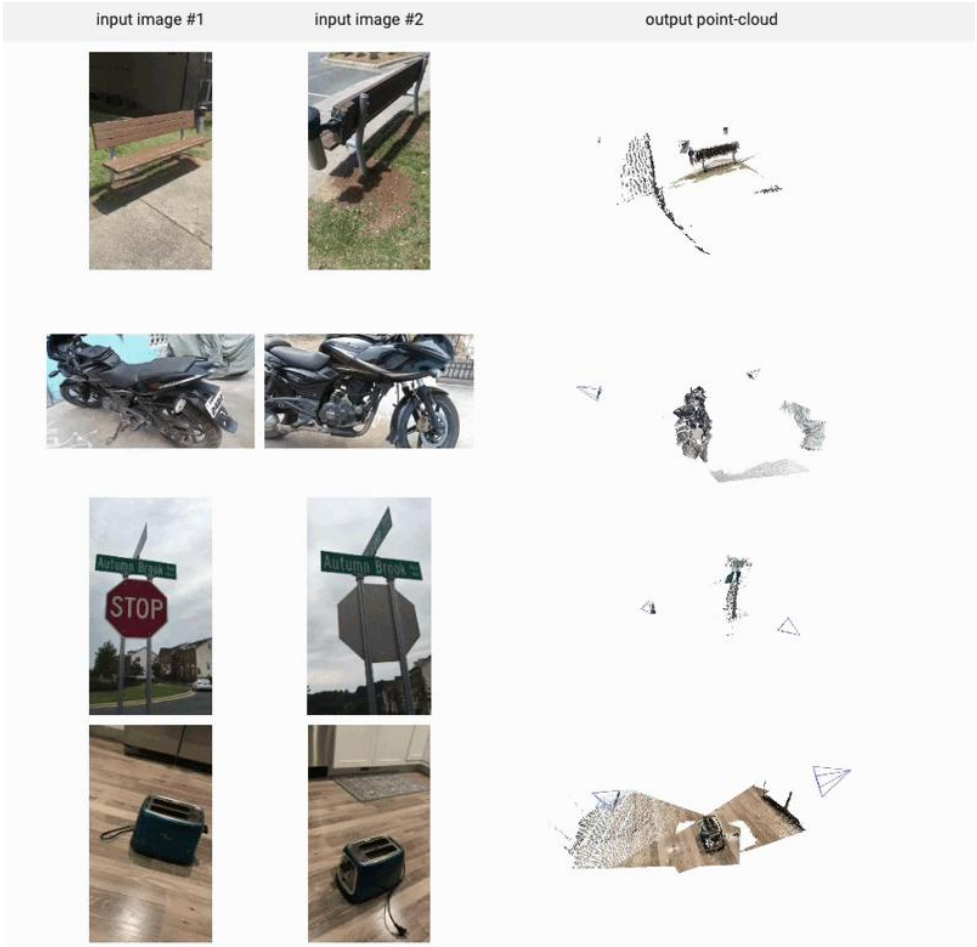
- Background
  - 3D vs 4D

- FreeTimeGS: Free Gaussian Primitives at Anytime and Anywhere for Dynamic Scene Reconstruction [CVPR 2025]

- MonoFusion: Sparse-View 4D Reconstruction via Monocular Fusion [ICCV 2025]

Background

- 3D vs 4D

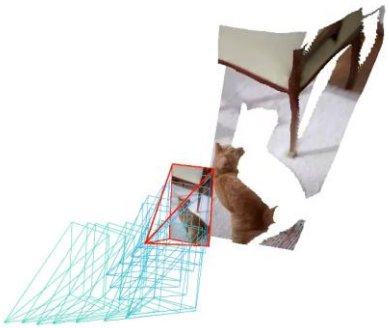


3D point cloud reconstruction

3D Gaussian reconstruction

Background

- 3D vs 4D



4D point cloud reconstruction

4D Gaussian reconstruction

FreeTimeGS: Free Gaussian Primitives at Anytime and Anywhere for Dynamic Scene Reconstruction [CVPR 2025]

Introduction

- 논문에서 다루고자 하는 task
  - 복잡한 움직임을 가지는 4D scene reconstruction
    - Motion이 크거나 비선형적 움직임 가지는 동적 객체의 spatio-temporal reconstruction
- 기존 연구
  - Implicit한 deformation field 이용하여 canonical space를 각 프레임 공간 warping
- 기존 연구의 문제점
  - RGB supervision으로 deformation field 최적화 실패
    - Canonical space와 observation space 사이 long-range correspondenc를 구축 어려움
- 해당 논문에서 제안하는 해결방법
  - 임의의 위치와 시간에서 Gaussian primitive가 나타나도록 모델링
  - 각 Gaussian primitive에 명시적인 motion function을 적용

1) https://fudan-zvg.github.io/4d-gaussian-splatting/
2) https://en.wikipedia.org/wiki/Rotations_in_4-dimensional_Euclidean_space

Method

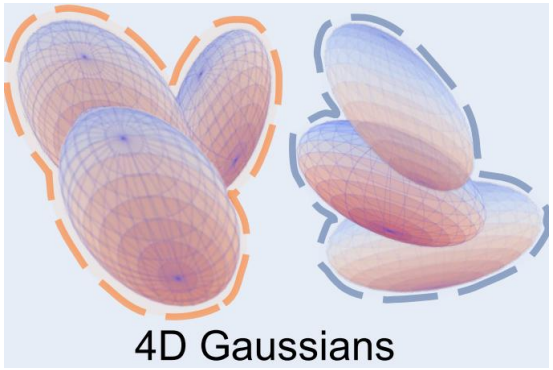
- 4D Gaussian distribution[1]을 사용한 4차원 시공간 정의
  - 3DGS는 3차원 공간을 표현하기 위해 3D Gaussian distribution 사용
  - FreeTimeGS는 time 축을 추가한 4D Gaussian distribution 으로 4차원 시공간 표현
    - Gaussian distribution formulation

$$N(x|\mu,\Sigma) = \frac{1}{(2\pi)^{D/2}}\frac{1}{|\Sigma|^{1/2}}\exp\left(-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)\right)$$

    - 4D Gaussian primitive

$$\mu = (\mu_x, \mu_y, \mu_z, \mu_t)$$

$$\Sigma = RSS^TR^T$$



$$\checkmark R = L(q_l)R(q_r) = \begin{pmatrix} a & -b & -c & -d \\ b & a & -d & c \\ c & d & a & -b \\ d & -c & b & a \end{pmatrix}\begin{pmatrix} p & -q & -r & -s \\ q & p & s & -r \\ r & -s & p & q \\ s & r & -q & p \end{pmatrix}$$

- $q_l = (a,b,c,d), q_r = (p,q,r,s)$

4D Euclidean space의 rotation은 두 isotropic rotation 쌍으로 decomposition 가능[2]

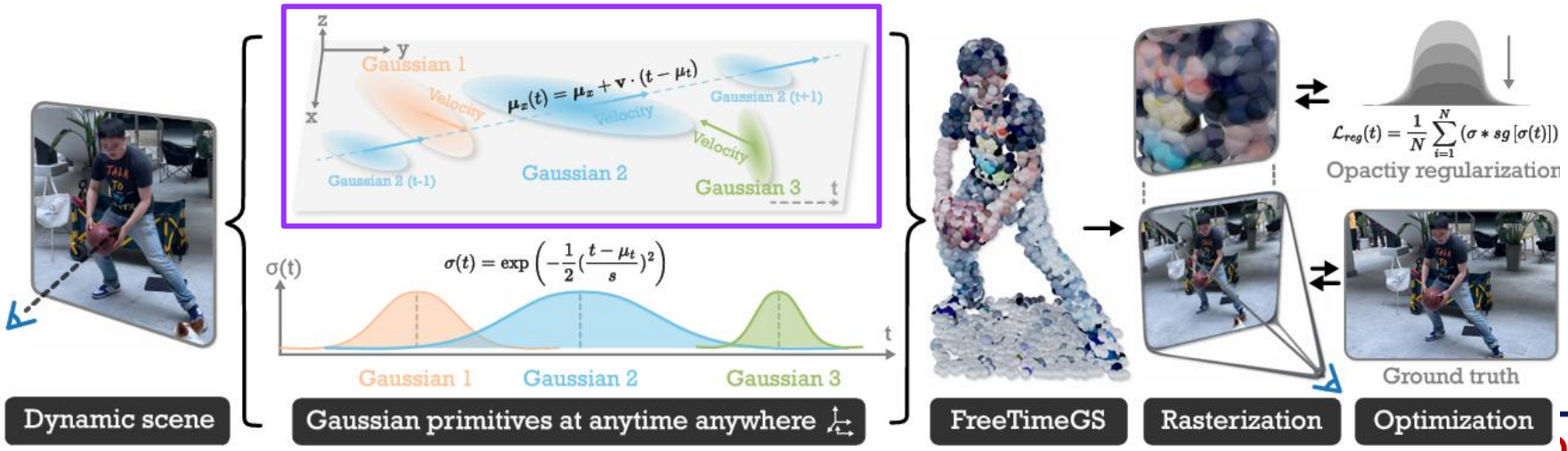
$$\checkmark S = diag(s_x, s_y, s_z, s_t)$$

Method

기존 3D Gaussian parameter

새롭게 정의된 parameter

- Gaussian primitives at anytime anywhere
  - 임의의 공간적 위치와 시간 단계에서 나타날 수 있는 Gaussian primitive 정의
  - Gaussian primitive에 대해 8개의 learnable Gaussian parameter 정의
    - Position, scale, orientation, opacity, SH coefficient, time, duration, velocity
  - 각 Gaussian primitive에 명시적인 motion function 적용
    - $\mu_x(t) = \mu_x + \mathrm{v} \cdot (t - \mu_t)$
      - 시간 t에서 3D Gaussian의 위치 계산
    - $\Sigma_x(t) = \Sigma_{1:3,1:3} - \Sigma_{1:3,4}\Sigma_{4,4}^{-1}\Sigma_{4,1:3}$



Method

- Gaussian primitives at anytime anywhere
  - Time variant alpha value
    - 3D Gaussian Splatting alpha

$$\alpha_i = \sigma_i \exp(-\frac{1}{2}(\mathrm{x}-\mu_i)^T \Sigma_i^{-1}(\mathrm{x}-\mu_i))$$

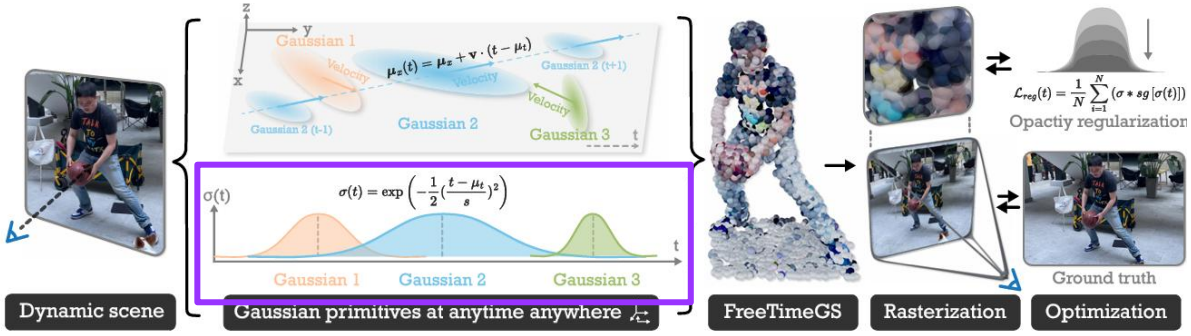
    - Time variant alpha

$$\sigma(\mathrm{x},t) = \sigma(t) * \sigma * \exp(-\frac{1}{2}(\mathrm{x}-\mu_x(t))^T \Sigma^{-1}(\mathrm{x}-\mu_x(t)))$$

$$\sigma(t) = \exp(-\frac{1}{2}(\frac{t-\mu_t}{s})^2)$$

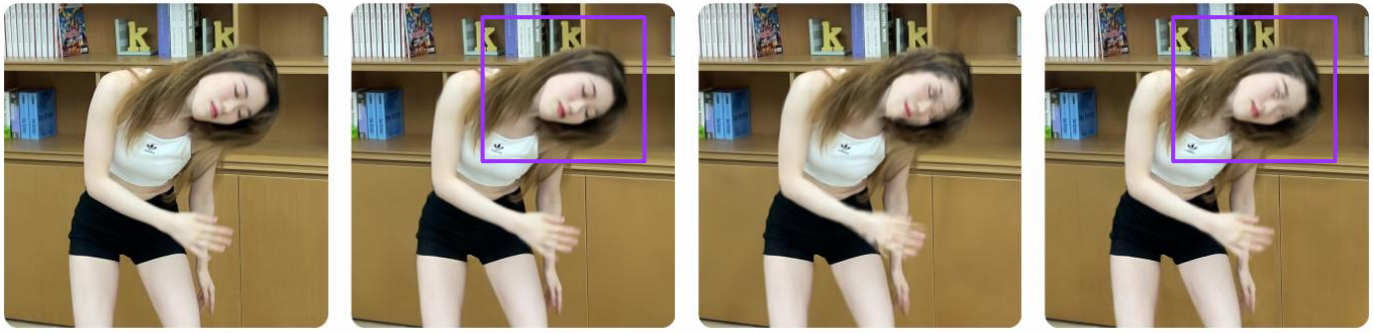
✓s: duration variable

✓전체 시간에 걸쳐 Gaussian primitive가 미치는 영향을 나타내는 weight로 작용



Method

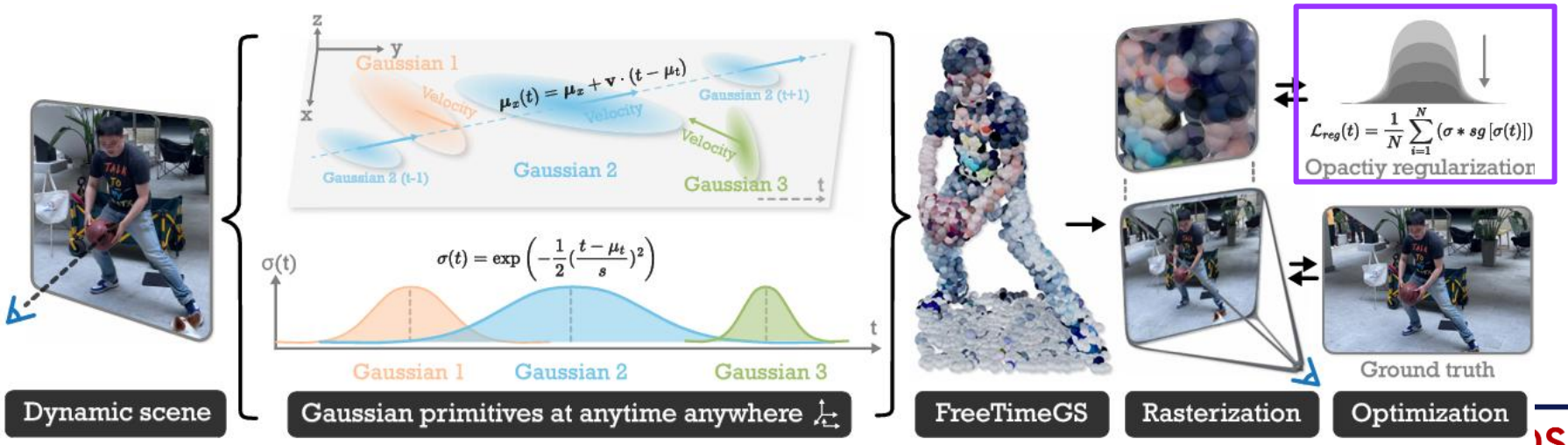
- Training
  - Objective function
    - $\mathcal{L} = \mathcal{L}_{render} + \mathcal{L}_{reg}(t)$
      - $\mathcal{L}_{render} = \lambda_{img}\mathcal{L}_{img} + \lambda_{ssim}\mathcal{L}_{ssim} + \lambda_{perc}\mathcal{L}_{perc}$
      - $\mathcal{L}_{reg}(t) = \frac{1}{N}\sum_{i=1}^{N}(\sigma * sg[\sigma(t)])$
    - 렌더링 loss만으로는 바른 움직임이나 복잡한 motion에서 품질저하 발생
      - Gaussian primitive의 opacity가 대부분 1이 나옴을 확인
        - ✓Opacity 값은 0~1 값을 가짐
      - 영향을 미치지 못해야하는 Gaussian들의 영향력이 커짐에 따라 blur한 결과 발생



Ground Truth | Ours | *w/o* our motion | *w/o* 4d regularization

Method

- Training
  - 4D regularization
    - Opacity가 높아지지 않도록 억제하는 constraint 설계
      - $\mathcal{L}_{reg}(t) = \frac{1}{N}\sum_{i=1}^{N}(\sigma * sg[\sigma(t)])$
        - ✓모든 Gaussian에 대한 opacity값의 평균을 손실함수로 사용
        - ✓시간에 따른 opacity weight를 최소화하는 것을 방지하기 위해 stop gradient 적용
      - 해당 시간에 대한 모든 Gaussian의 opacity값을 억제



Method

- Training
  - Periodic relocation of primitives
    - Opacity 1로 몰리는 문제를 해결하면 특정 영역으로 Gaussian이 쏠리는 현상 발생
      - 낮은 opacity로 인해 목표 영역을 채우기 위해 Gaussian이 몰림
    - 더 많은 Gaussian이 필요한 영역을 측정하기 위해 sampling score 제안
      - $s = \lambda_g \nabla_g + \lambda_O \sigma$
        - ✓∇_g: 시간축이 제외된 Gaussian의 spatial gradient 정보
      - 일정 학습 iteration 마다 opacity가 threshold보다 낮은 Gaussian에 대해 sampling 수행
        - ✓Gaussian을 sampling score가 높은 영역으로 이동
        - ✓추가적인 densification 역할 수행
  - Initialization
    - 각 비디오 프레임에 대해 ROMA[1]로 매칭 후 triangulation 수행하여 3d point 계산
      - 해당 시간 스텝과 3d point를 이용하여 mean과 time을 initialize
    - 두 프레임에 해당하는 3d point사이의 translation vector를 속도로 initialize
      - 매칭되는 3d point는 knn 알고리즘으로 임의로 획득

Result

- 정량결과 및 ablation

Table 1. **Quantitative comparison on the Neural 3D Video [19] Dataset.** We report PSNR, $DSSIM_1$, $DSSIM_2$, and LPIPS to evaluate the rendering quality. [1]: only includes the *Flame Salmon* scene. [2]: excludes the *Coffee Martini* scene.

	PSNR↑	$DSSIM_1$ ↓	$DSSIM_2$ ↓	LPIPS↓
Neural Volume[1] [26]	22.80	-	0.062	0.295
LLFF[1] [29]	23.24	-	0.076	0.235
DyNeRF[1] [19]	29.58	-	0.020	0.083
HexPlane[2] [4]	31.71	-	0.014	0.075
K-Planes [10]	31.63	-	0.018	-
MixVoxels-L [42]	31.34	-	0.017	0.096
MixVoxels-X [42]	31.73	-	0.015	0.064
HyperReel [1]	31.10	0.036	-	0.096
NeRFPlayer [39]	30.96	0.034	-	0.111
Deformable-3DGS [44]	31.15	0.030	-	0.049
C-D3DGS [13]	30.46	-	0.022	0.150
SWinGS [37]	31.10	0.030	-	0.096
Ex4DGS [17]	32.11	0.030	0.015	0.048
4DGS [49]	32.01	-	0.014	0.055
STGS [21]	32.05	**0.026**	0.014	0.044
Ours	**33.19**	**0.026**	**0.013**	**0.036**

Table 4. **Ablation studies.** We include quantitative results for both entire sequence and the subsequence with fastest motion (entire/fastest)

	PSNR↑	$DSSIM_2$↓	LPIPS↓
w/o our motion	28.10/26.92	0.024/0.031	0.165/0.161
w/o 4d regularization	28.68/29.09	0.023/0.024	0.159/0.150
w/o periodic relocation	29.07/29.15	0.020/0.021	0.155/0.146
w/o 4d initialization	28.33/27.06	0.023/0.030	0.162/0.158
Ours	29.74/30.75	0.018/0.017	0.152/0.133

Table 2. **Quantitative comparison on the ENeRF-Outdoor [23] Dataset.** Green and yellow cell colors indicate the best and the second best results, respectively.

	PSNR↑	$DSSIM_2$ ↓	LPIPS↓	FPS↑
ENeRF [23]	24.96	0.107	0.299	3
4K4D [45]	25.28	0.096	0.379	220
4DGS [49]	24.82	0.089	0.317	90
STGS [21]	24.93	0.091	0.297	226
Ours	25.36	0.077	0.244	454

Table 3. **Quantitative comparison on our *SelfCap* Dataset.** We include quantitative results for both the entire image and only dynamic regions (entire/dynamic). For 4DGS and STGS, we traversed different camera near plane settings during testing to maximize floater removal. Green and yellow cell colors indicate the best and the second best results, respectively.

	PSNR↑	$DSSIM_2$↓	LPIPS↓	FPS↑
STGS [21]	24.97/25.32	0.048/0.029	0.273/0.123	142
4DGS [49]	25.98/26.75	0.036/0.019	0.237/0.104	65
Ours	27.41/29.38	0.024/0.013	0.204/0.080	467

Result

- 정성결과

Result

- Real-Time VR Demos

MonoFusion: Sparse-View 4D Reconstruction via Monocular Fusion [ICCV 2025]

Intro

- 논문에서 다루고자 하는 task
  - 소수의 sparse multi-view 상황의 4D reconstruction
- 기존 연구
  - 기존 monocular 및 multi-view 4D Gaussian Splatting 모델 존재
- 기존 연구의 문제점
  - 겹치는 view가 적은 소수의 multi-view로 reconstruction이 어려움
  - 충분한 initial point cloud 획득에 어려움 존재

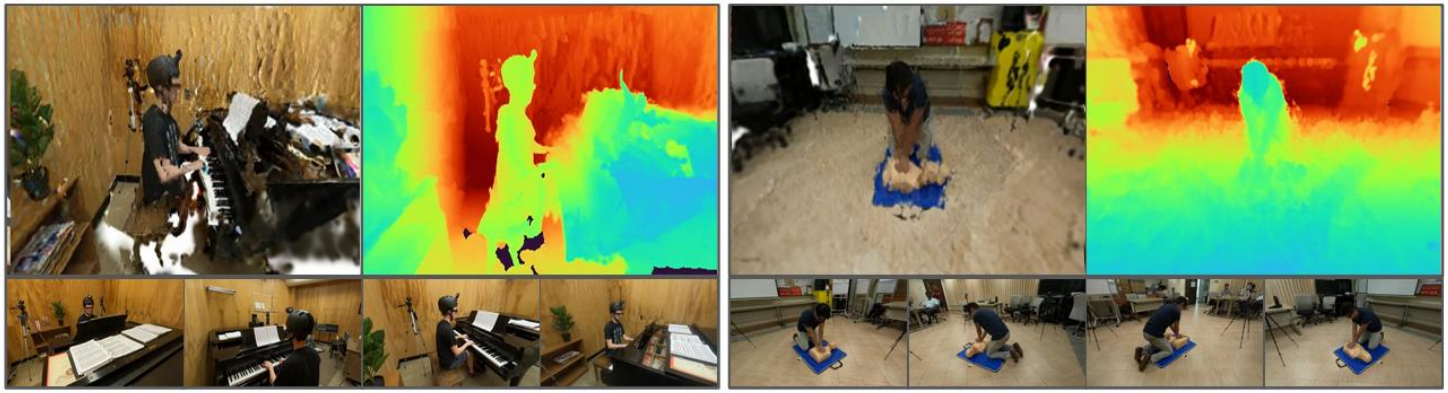


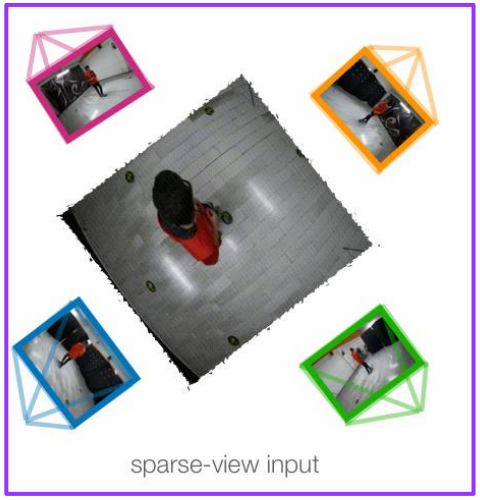
Figure 1. **Dynamic Scene Reconstruction from Sparse Views**. MonoFusion reconstructs dynamic human behaviors, such as playing the piano or performing CPR, from four equidistant inward-facing static cameras. We visualize the RGB and depth renderings of a 45° novel view between two training views. Training views are shown below for reference.

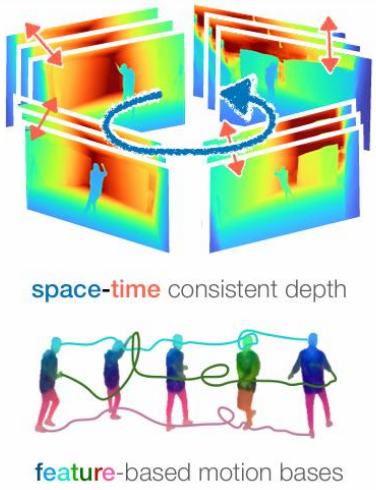
Method

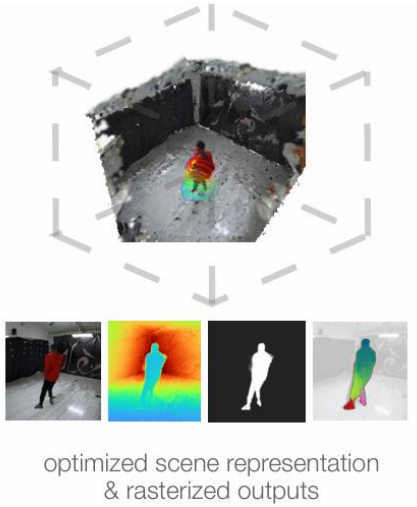
- 3D Gaussian Scene Representation
  - 첫 번째 프레임에 대한 geometrical 공간을 canonical space로 정의하여 사용
    - Gaussian Splatting을 이용하여 공간 정보 표현
    - 3D Gaussian parameter의 mean, position을 deformable parameter로 설정
      - Scale, opacity, color parameter는 모든 프레임에 대하여 변하지 않는 parameter로 설정
    - 3D Gaussian 별로 learnable한 32차원의 parameter를 임베딩
    - Canonical space의 3D Gaussian들은 motion base들을 이용하여 deformation 수행
      - 각 프레임으로 transformation 수행

Method

- Space-Time Consistent Depth Initialization
  - Multi-view pointmap prediction
    - 데이터셋에 주어진 카메라파라미터를 initial 값으로 DUSt3R 구동
      - DUSt3R의 global optimization 단계에서 camera parameter를 constraint로 사용
        - ✓이미지 쌍별 획득된 pointmap을 global coordinate로 alignment하는 과정에서 사용
      - Global scale의 정렬된 3d pointmap 획득 가능
    - Metric scale의 3d pointmap (χ_k^t)을 projectio하여 metric scale의 depth map($d_k^t(u,v)$)획득
      - $d_k^t(u,v)[u \quad v \quad 1]^T = K_k P_k \chi_k^t(u,v)$





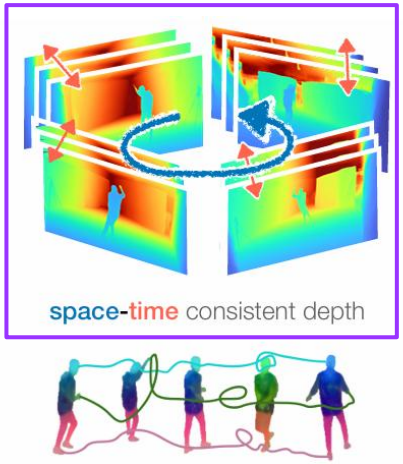


Method

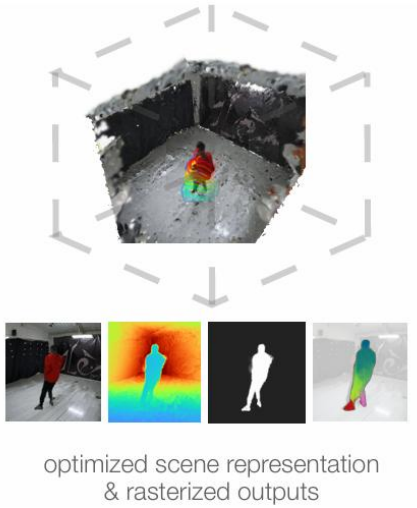
- Space-Time Consistent Depth Initialization
  - Spatio-temporal alignment of monocular depth with multi-view consistent pointmaps
    - Multi-view 추정기를 통한 사람에 대해서 낮은 품질의 depth가 잡히는 경향 존재
      - 학습 데이터에 등장하는 사람은 주로 동적 객체로서 out-lier 취급하기 때문
    - 단일 이미지에 대해 기하학 구조를 잘 예측하는 monocular depth를 활용
      - Projection되어 구해진 depth를 활용하여 monocular depth의 일관성 최적화

$$\checkmark \arg\min_{\{a_k^t, b_k^t\}} \sum_{t=1}^{T} \sum_{k=1}^{K} \sum_{u,v \in BG_k^t} \left\| \left(a_k^t m_k^t(u,v) + b_k^t\right) - d_k^t(u,v) \right\|^2$$

sparse-view input



feature-based motion bases



optimized scene representation
& rasterized outputs

Method

- Grouping-based Motion Initialization
  - Time-viarying rigid transformations
    - Canoncial space의 Gaussian을 각 프레임에 해당하는 공간으로 rigid transformation 수행
      - $x_s = R_{0\to t}x_0 + t_{0\to t}, R_t = R_{0\to t}R_0$
  - Motion bases
    - 3D motion에 대한 learnable basis trajectory를 정의하여 rigid transformation 학습
      - $T_{0\to t}^{(i)} = \sum_{b=1}^{B} w^{(i,b)} T_{0\to t}^{(b)}$
        - ✓Gaussian별로 basis trajectory의 weighted sum을 통해 각 time으로 transformation
      - Weight를 DINOv2 feature를 사용하여 초기화 진행
        - ✓3D pointmap에 feature를 임베딩
        - ✓Pointmap의 feature를 보고 k-means clustering 수행
          - B개의 cluster를 구성하여 basis의 개수 결정
          - 각 cluster의 중심과 Gaussian들의 중심에 대하여 L2 distance를 이용하여 weight 초기화

Method

- Grouping-based Motion Initialization
  - Optimization

$$-\mathcal{L}_{recon} = \left\|\hat{I} - I\right\|_1 + \left\|\hat{M} - M\right\|_1 + \left\|\hat{F} - F\right\|_1 + \left\|\hat{D} - D\right\|_1$$

Photometric loss, mask loss, feature loss, depth loss를 결합하여 사용

✓각 pointmap에 임베딩된 feature또한 렌더링 후 DINIOv2 feature로 supervision

$$-\mathcal{L}_{rigid} = \sum_{neighbors\ i} \left\|\hat{X}_t - \hat{X}_t^{(i)}\right\|_2^2 - \left\|\hat{X}_{t'} - \hat{X}_{t'}^{(i)}\right\|_2^2$$

시간 t로 deformation되어진 Gaussian에 대하여 주변 Gaussian과의 차이 억제

✓$\hat{X}_t$: 시간 t에서의 3D Gaussian 위치

✓$\hat{X}_{t'}$: 시간 t'에서의 3D Gaussian 위치

✓i: Knn 알고리즘을 통해 구한 주변 Gaussian들

Result

- 정량결과

Method	$\mathcal{L}_{\text{feat}}$	$\mathbf{d}_n$	$\mathbf{T}_{0\to t}^{(b)}$	↑PSNR	↑SSIM	↓LPIPS	↑IoU
Baseline	✗	✗	✗	26.19	0.915	0.077	0.60
+ $\mathcal{L}_{\text{feat}}$	✓	✗	✗	25.39	0.933	0.087	0.63
+ Our depth / no $\mathcal{L}_{\text{feat}}$	✗	✓	✗	29.55	0.944	0.037	0.73
+ Our depth / $\mathcal{L}_{\text{feat}}$	✓	✓	✗	29.31	0.941	0.041	0.75
+ Motion bases (**Ours**)	✓	✓	✓	**30.40**	**0.947**	**0.037**	**0.81**

Table 3. **Ablation study of pipeline components.** We ablate our choice of feature-metric loss, spacetime consistent depth, and feature-based motion bases. While the proposed depth and feature-based motion bases considerably improve 4D reconstruction (evaluated by photometric errors), we find that our feature loss helps learn better motion masks (evaluated by IoU).

Method	**PSNR ↑**	**SSIM ↑**	**LPIPS ↓**	**IOU ↑**	**AbsRel (↓)**
SOM	16.73	0.554	0.491	0.287	0.578
Dyn3D-GS	23.31	0.776	0.316	—	0.273
MV-SOM	21.56	0.541	0.433	0.482	0.413
MonoFusion	**25.73**	**0.847**	**0.158**	**0.943**	**0.188**

Table 2. **Quantitative analysis of 45° novel-view synthesis on Panoptic Studio.** We benchmark our method against state-of-the-art approaches by evaluating both the dynamic foreground region and the entire scene. Notably, the evaluation is conducted on novel views where the cameras are at least 45° apart from all training views. We additionally evaluate the geometric reconstruction quality with absolute relative (AbsRel) error in rendered depth.

Dataset	Method	Full Frame				Dynamic Only			
		PSNR ↑	SSIM ↑	LPIPS ↓	AbsRel ↓	PSNR ↑	SSIM ↑	LPIPS ↓	IOU ↑
Panoptic Studio	SOM [54]	17.86	0.687	0.460	0.491	18.75	0.701	0.236	0.358
	Dyn3D-GS [38]	25.37	0.831	0.266	0.207	26.11	0.862	0.129	—
	MV-SOM [54]	26.28	0.858	0.241	0.331	26.80	0.883	0.161	0.886
	MonoFusion	**28.01**	**0.899**	**0.117**	**0.149**	**27.52**	**0.944**	**0.022**	**0.965**
ExoRecon	SOM [54]	14.73	0.535	0.482	0.843	15.63	0.559	0.450	0.294
	Dyn3D-GS [38]	24.28	0.692	0.539	0.612	24.61	0.673	0.384	—
	MV-SOM-DS [54]	28.37	0.906	0.079	0.398	28.23	0.931	0.063	0.872
	MV-SOM [54]	26.91	0.890	0.138	0.474	27.31	0.919	0.078	0.845
	MonoFusion	**30.43**	**0.927**	**0.061**	**0.290**	**29.71**	**0.947**	**0.017**	**0.963**

Table 1. **Quantitative analysis of held-out view synthesis.** We benchmark our method against state-of-the-art approaches by evaluating the novel-view rendering and geometric quality on both the dynamic foreground region and the entire scene, across the held-out frames from input videos. MV-SOM is a multi-view version of Shape-of-Motion [54] that we construct by instantiating four different instances of single-view shape of motion, and optimize them together. On Panoptic Studio, groundtruth depth for computing the AbsRel metric is obtained from 27-view optimization of the original Dynamic 3DGS, and for ExoRecon, we project the released point clouds obtained via SLAM from Aria glasses. When evaluating single-view baselines, SOM [54], we naively aggregate their predictions from the four views and evaluate this aggregated prediction against the evaluation cameras.

Result

- 정성결과

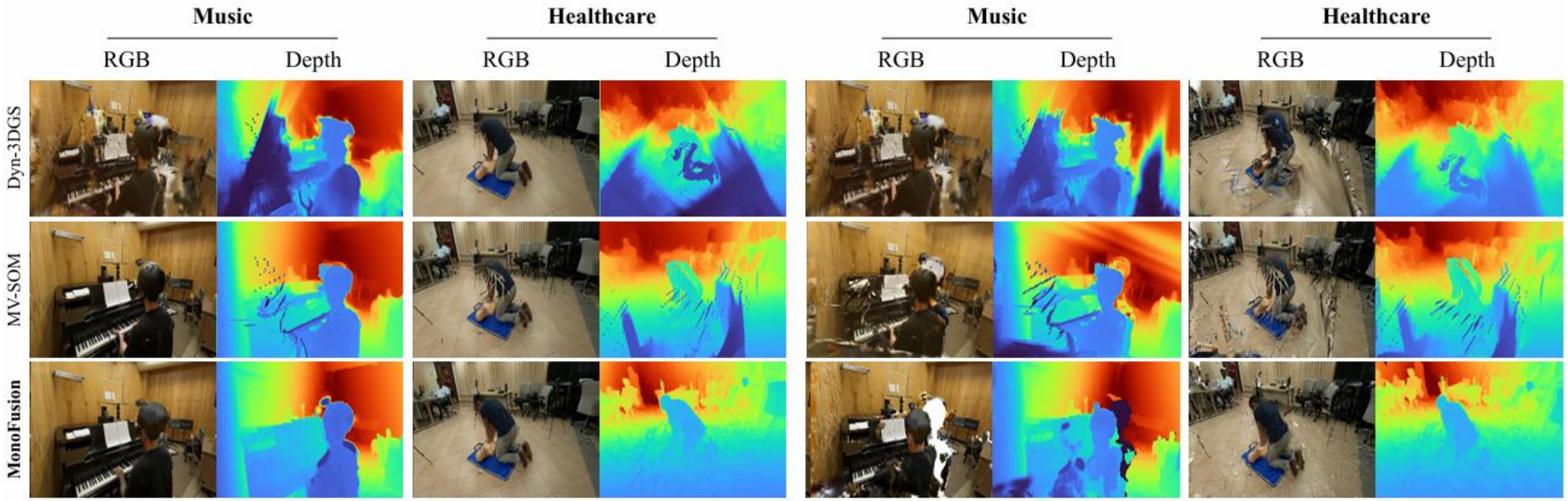
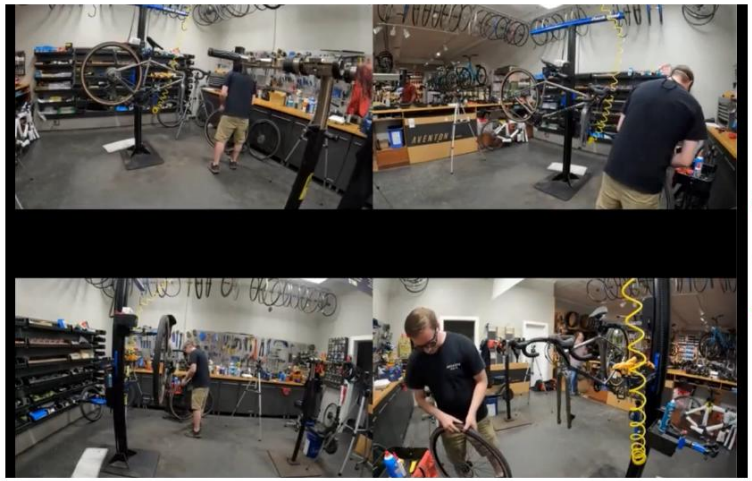


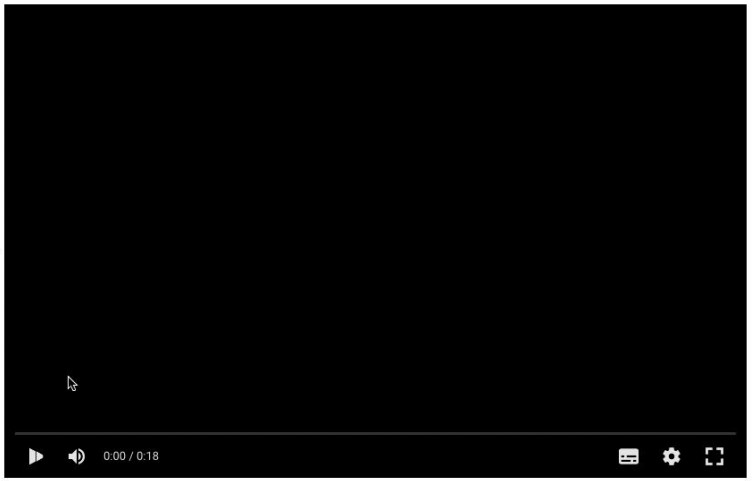
Figure 4. **Qualitative analysis of held-out view synthesis on ExoRecon.** We show qualitative results of held-out view synthesis (**left**) and a 5° deviation from the static camera position at the held-out timestamp (**right**). As compared to other multi-view baselines, our method does dramatically better at interpolating the motion of dynamic foreground (left), even from new camera views (right). We posit that Dynamic 3DGS suffers because of lack of geometric constraints and MV-SOM has duplicate foreground artifacts because of conflicting depth initialization from the four views.

Video Result



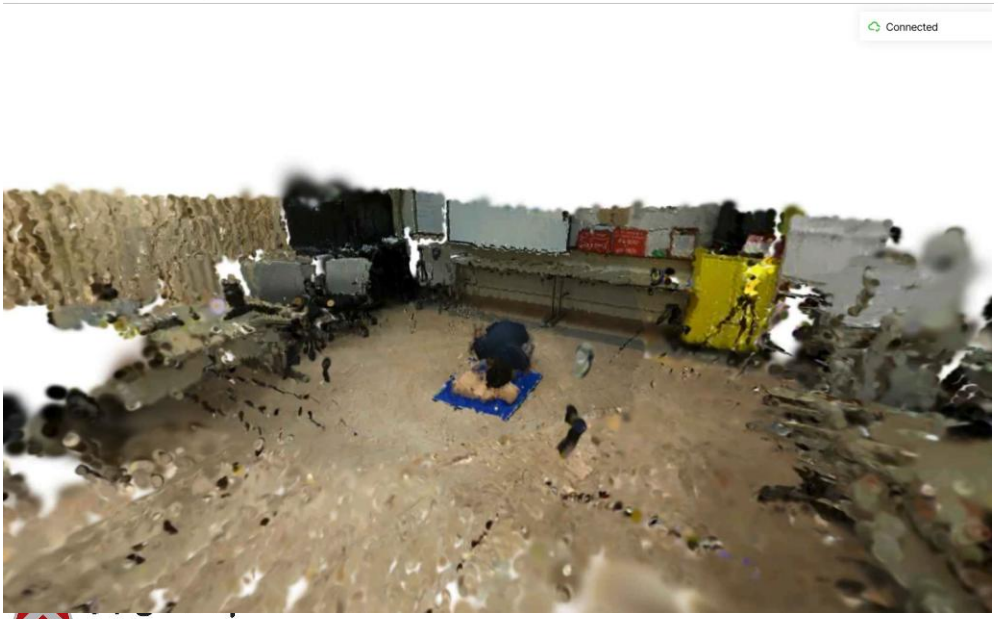
anytime +

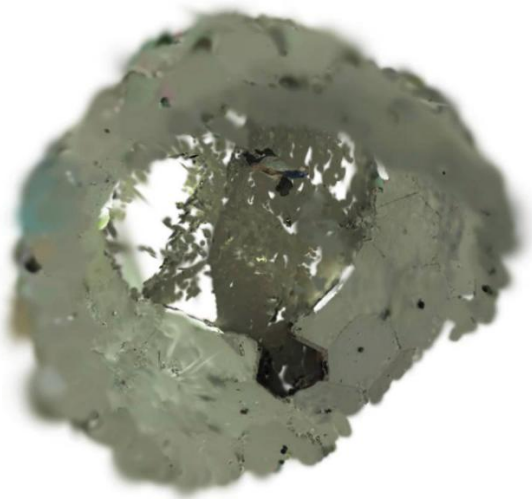
anywhere



Input: sparse-view 2D observation

Output: free-view 4D reconstruction





감사합니다