

# 3D Human Modeling through reconstruction techniques

---



***Sogang University***

*Vision & Display Systems Lab, Dept. of Electronic Engineering*



***Presented By***

**김태우**

# Contents

- Background
  - Human LBS Skinning
  - Canonical space (기준 좌표계)
  
- 관련 논문 소개
  - HUGS: Human Gaussian Splats (CVPR 2024)
  - HOSNeRF: Dynamic Human-Object-Scene Neural Radiance Fields from a Single Video (ICCV 2023)

# Background

- Linear Blending Skinning (LBS)

- Mesh 변형 기법으로, 관절(joint)의 움직임에 따라 정점(vertex) 위치를 선형적으로 계산하는 방법

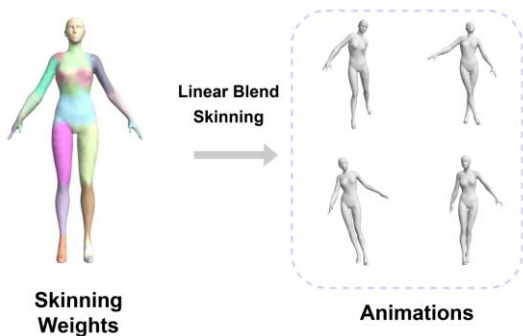
- 관절 움직임이 메쉬 변형에 직접적으로 영향을 주며, 가중치(Weight)를 기반으로 변형 정도를 결정

- 원리

- 각 Vertex가 여러 관절(Joint)의 영향을 받음

- Weight Matrix는 (Joint 개수 × Vertex 개수)로 구성

※ 각 Vertex가 특정 관절로부터 받는 선형적 영향도



<SMPL LBS 변환>

$$V_{\text{new}} = \sum_{i=1}^J W_i \cdot (T_i \cdot V_{\text{original}})$$

- $V_{\text{new}}$ : 변형 후 Vertex 위치
- $W_i$ : Vertex에 대한 관절  $i$ 의 가중치
- $T_i$ : 관절  $i$ 의 변환 행렬
- $V_{\text{original}}$ : 변형 전 Vertex 위치

<선형적 변환>

# Background

- Canonical space (기준 좌표계)

- 모델이나 mesh의 초기 상태를 정의하는 좌표 공간

- 다른 자세로 변형을 적용하기 위한 기준 공간

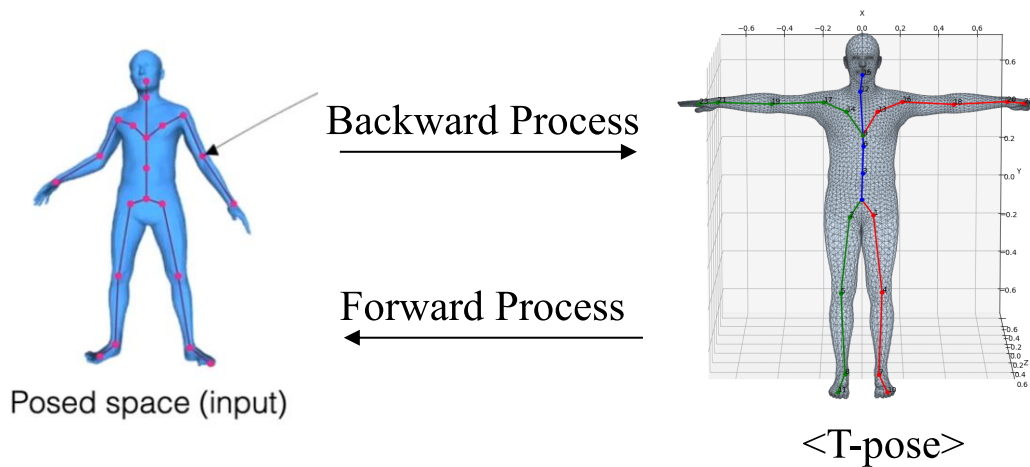
- ※ T-pose: 캐릭터가 팔을 수평으로 벌리고 선 자세

- ※ A-pose: 팔이 약간 아래로 내려간 자세

- 변형 과정

- Backward Process (역방향 과정): 포즈 공간을 Canonical Space로 변환하는 과정

- Forward Process (정방향 과정): Canonical Space에서 새로운 포즈 공간으로 변환



# Background

- Task: 4D Human and Scene Reconstruction

- 시간을 포함하여 동적인 인간과 정적인 배경 장면을 동시에 재구성

- 기존 방법의 문제점

- 기존 모델 NeuMan<sup>1)</sup>(ECCV 2022), Vid2Avatar<sup>2)</sup>(CVPR 2023) 정적 장면에 뛰어남

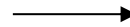
- 반면, 움직이는 인간을 포함한 동적 장면에서는 재구성 품질이 떨어짐



NeuMan [7]

Vid2Avatar [6]

<기존 방법의 문제점>



<HUGS Method>

# HUGS: Human Gaussian Splats

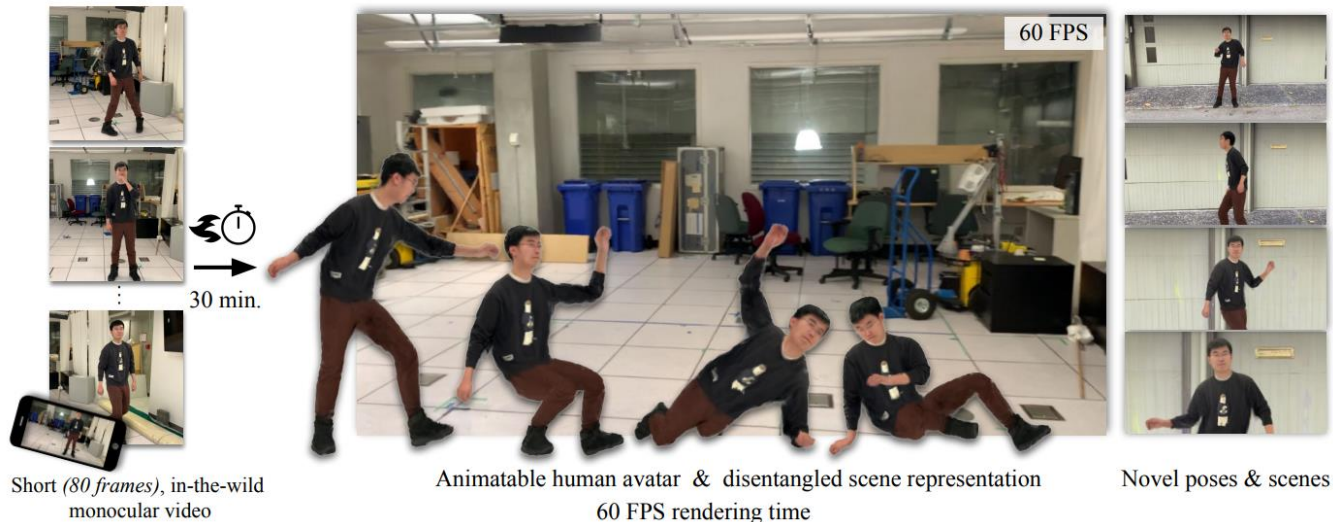
# Introduction

- Contribution

- 기존 NeRF 기반 방법에 비해 빠른 렌더링 속도
- 다중 뷰 데이터 세트에서 SOTA 성능 달성
- 50~100개의 frame 만으로 reconstruction 가능

- Conclusion 요약

→ HUGS는 monocular video로 동적 인간과 scene의 고품질 4D 재구성 가능

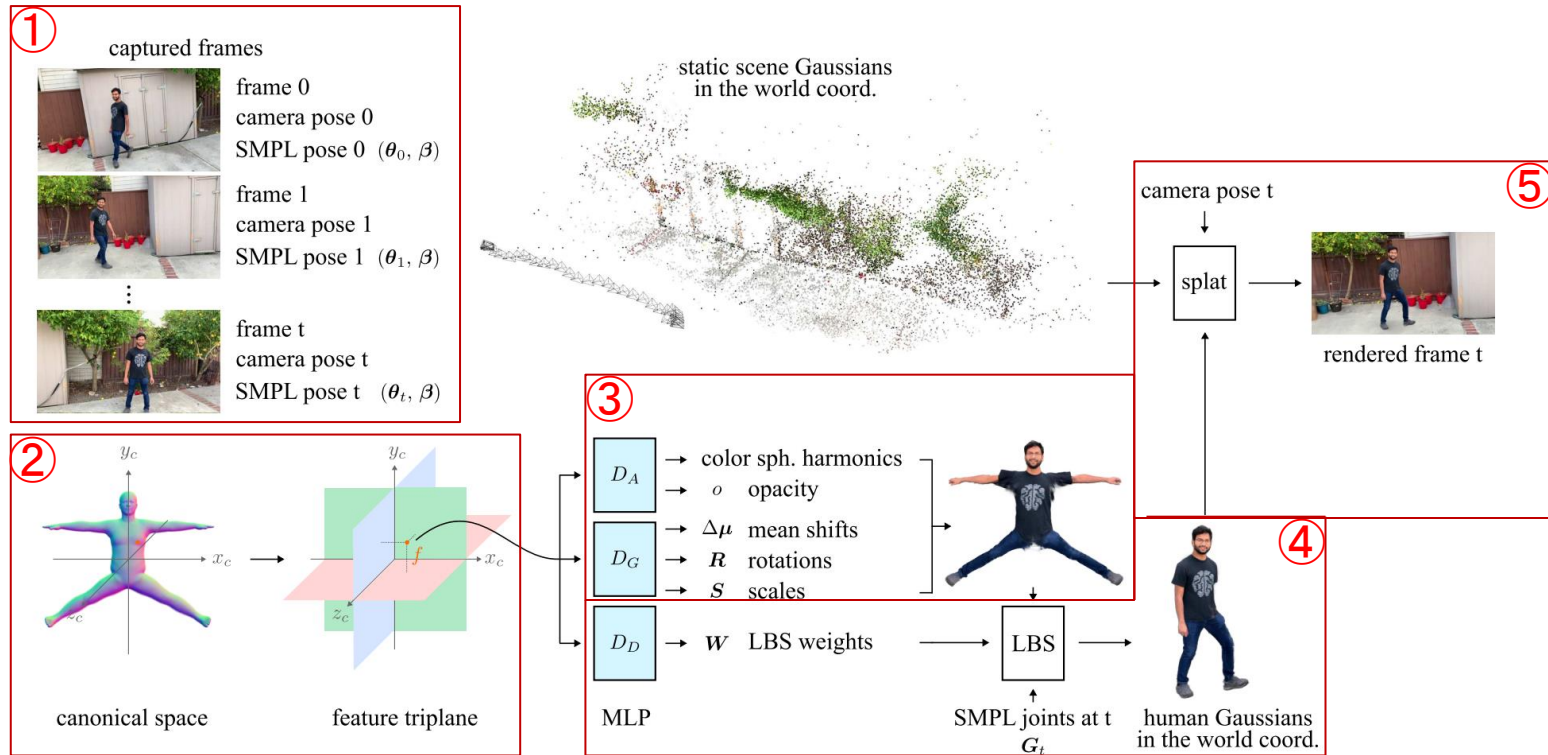


<HUGS 소개>

# Method

- HUGS overview

- 총 5파트로 나누어 설명





# Method

- Part 1. Input data

- 입력: 단일 비디오(50~100 프레임)와 카메라 pose 정보
- Human data 생성 (pose  $\theta$  parameter, shape  $\beta$  parameter)
  - Pretrained된 SMPL regressor<sup>1)</sup>를 사용
  - 각 비디오 프레임에 human pose  $\theta$  parameter, shape  $\beta$  parameter 추정

captured frames



frame 0  
camera pose 0  
SMPL pose 0 ( $\theta_0, \beta$ )



frame 1  
camera pose 1  
SMPL pose 1 ( $\theta_1, \beta$ )

⋮



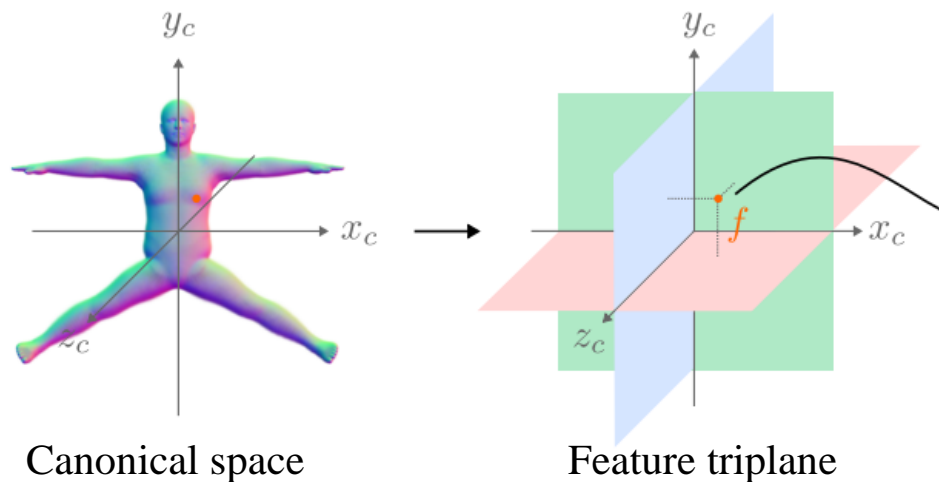
frame t  
camera pose t  
SMPL pose t ( $\theta_t, \beta$ )

<각 비디오 프레임에 대해 parameter 생성>

# Method

- Part 2. Initialization step

- Human: SMPL mesh의 정점(vertex)을 기반으로 인간의 초기 가우시안 분포를 생성
- Canonical space (표준 공간) 정의
  - Human model을 모든 pose에 대해 독립적으로 표현하기 위해 정의한 공간
  - Da-pose를 정규화된 pose로 설정
- Feature triplane
  - Canonical space 크기 만큼의 임의의 벡터 공간
  - 각 가우시안에 대해서 색상, 크기, 회전, 등을 나타내기 위해 정의된 좌표 공간



# Method

- Part 3. Setting Gaussians (Appearance)

- 각 가우시안의 특성 추출

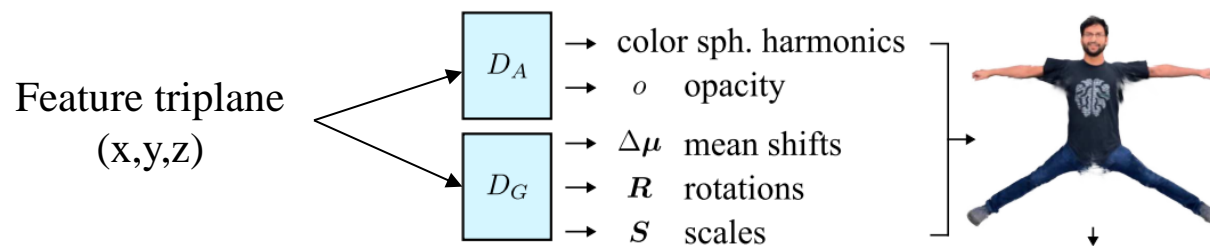
- 각 가우시안의 opacity, color, mean shifts, rotation, scales를 구하는 과정
    - MLP를 사용해 feature triplane의 좌표에 해당하는 가우시안의 특성 추출

- Feature triplane

- Triplane은 3개의 2D 평면으로 구성된 좌표 공간 (grid)

- ※ Canonical space와 동일한 크기의 공간

- 가우시안의 중심 좌표에 해당하는 값을 interpolation으로 계산하여 특징 벡터를 추출



<각 가우시안의 특성 추출>

# Method

- Part 3. Setting Gaussians (LBS weights)

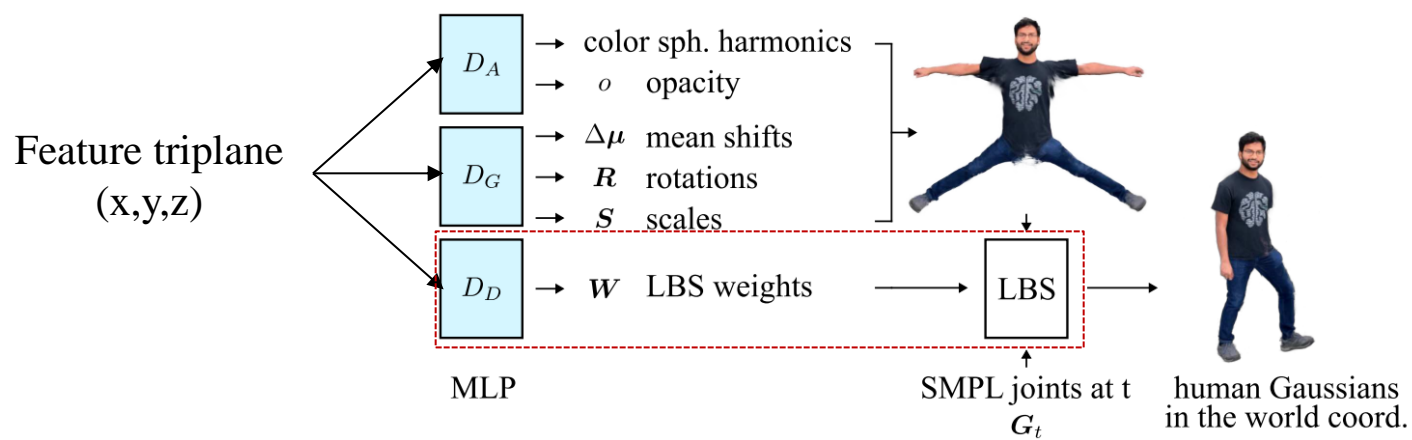
- Apply LBS weights to gaussians

- Initialization: SMPL의 vertex와 동일한 위치의 가우시안이므로, LBS weights를 상속
- LBS 이동 계산은 가우시안의 중심 좌표 기준

- LBS weights 계산 방법

- MLP를 사용해 feature triplane의 좌표에 해당하는 가우시안의 LBS weight 추정
  - ※ SMPL 외부의 좌표(머리카락, 옷)의 가우시안에도 LBS 계산 가능

- 각 프레임에 대한 SMPL pose 를 입력하여, LBS 변환



<각 가우시안의 LBS 추정>

# Method

## • Part 3. Setting Gaussians (LBS weights loss)

### • 학습된 가우시안들에 대해 새로운 LBS 추정 방법

- 가우시안의 중심 이동 발생, SMPL 외부의 가우시안 → 새로운 LBS 값 필요

### • 정규화

-  $W$ : 이전 iteration의 LBS 수치

- SMPL 메시의 가장 가까운  $k$ 개의 정점에서 거리에 따라, LBS 가중 평균 값을 사용

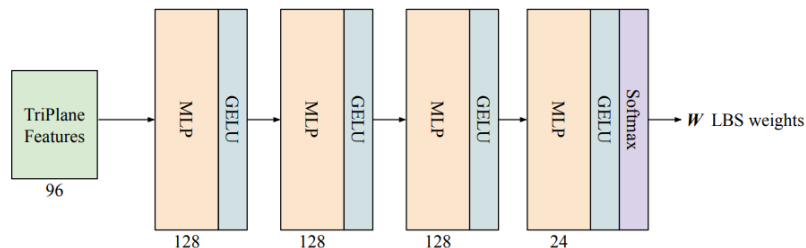
- 해당 정규화 term과 차이를 줄이는 방향으로 MLP 학습

$$\hat{W}_i = \frac{\sum_{j \in N_i} \omega_j W_j}{\sum_{j \in N_i} \omega_j}$$

정규화 term

$$L_{LBS} = \|W - \hat{W}\|_F^2$$

LBS loss

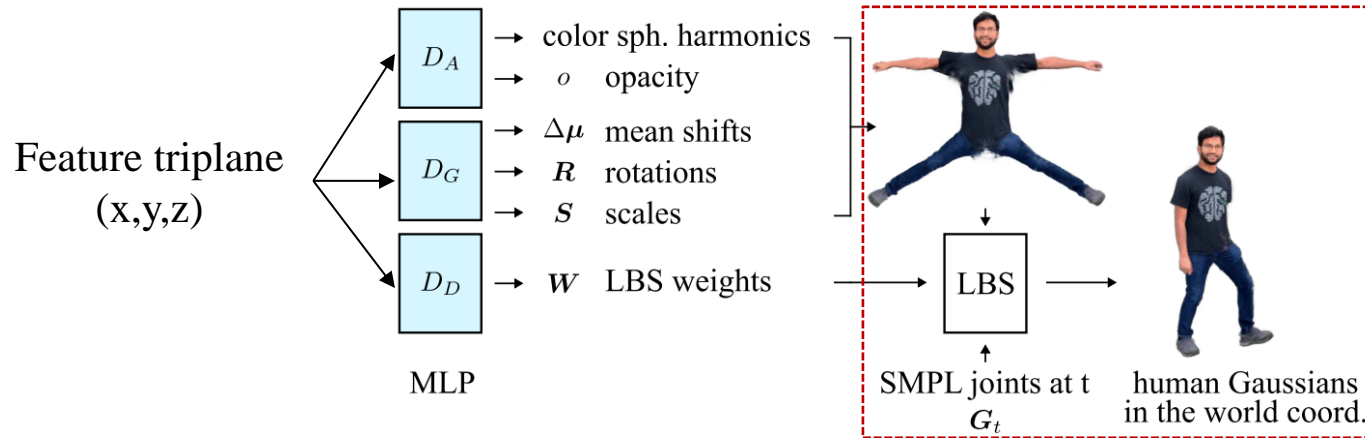


<LBS 추정 MLP 구조>

# Method

- Part 4. Apply LBS to Gaussians

- 각 프레임에 대한 SMPL pose 를 입력하여, LBS 변환
- 학습된 LBS 가중치를 사용: 3D 가우시안에 대해 위치, 회전, 스케일을 변환
  - 해당 pose 에 대한 가우시안 아바타 생성
- Test 과정에는 LBS inference가 필요 없음 (저장된 가중치 사용)



<Apply LBS in Gaussian model>

# Method

- Part 5. Render & Optimazation

- 렌더링된 이미지와 실제 촬영된 이미지를 비교하여 loss 계산

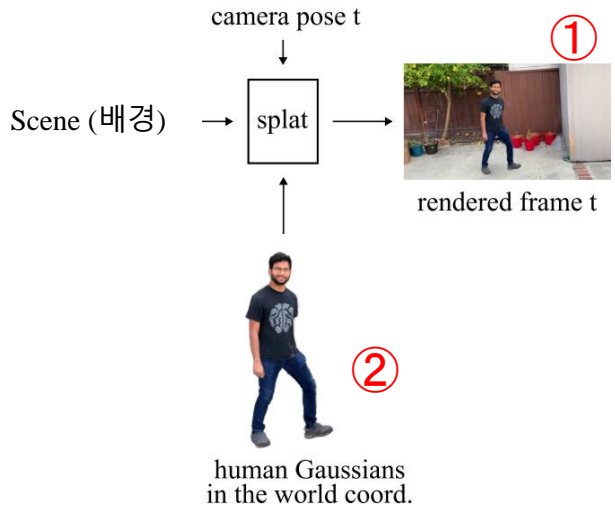
- $L_1(I_{GT}, I_R), L_{SSIM}(I_{GT}, I_R), L_{VGG}(I_{GT}, I_R)$  사용

- Training is  $\sim 100\times$  faster than NeRF based

- Scene and Human Losses

- 사람 영역은 pretrained segmentation model로 분리하여 loss 계산

- ☼ 랜덤 배경 이미지 사용



$$\mathcal{L} = \underbrace{\lambda_1 \mathcal{L}_1 + \lambda_2 \mathcal{L}_{ssim} + \lambda_3 \mathcal{L}_{vgg}}_{\textcircled{1} \text{ scene + human}} + \underbrace{\lambda_1 \mathcal{L}_1^h + \lambda_2 \mathcal{L}_{ssim}^h + \lambda_3 \mathcal{L}_{vgg}^h}_{\textcircled{2} \text{ human}} + \lambda_4 \mathcal{L}_{LBS}$$

$\lambda_1 = 0.8, \lambda_2 = 0.2, \lambda_3 = 1.0, \lambda_4 = 1000$

<Final loss term>

# Experiments

- NeuMan Dataset<sup>1)</sup>

- 휴대폰 카메라로 촬영 (Wild dataset)
- 10~20초 길이의 6개 영상으로 구성
- 카메라 팬(pan) 동작으로 멀티 뷰 구성
  - 카메라를 한 위치에 고정한 상태에서 수평으로 회전하면서 촬영
- 80% training frames, 10% validation frames, and 10% test frames



<NeuMan Dataset>



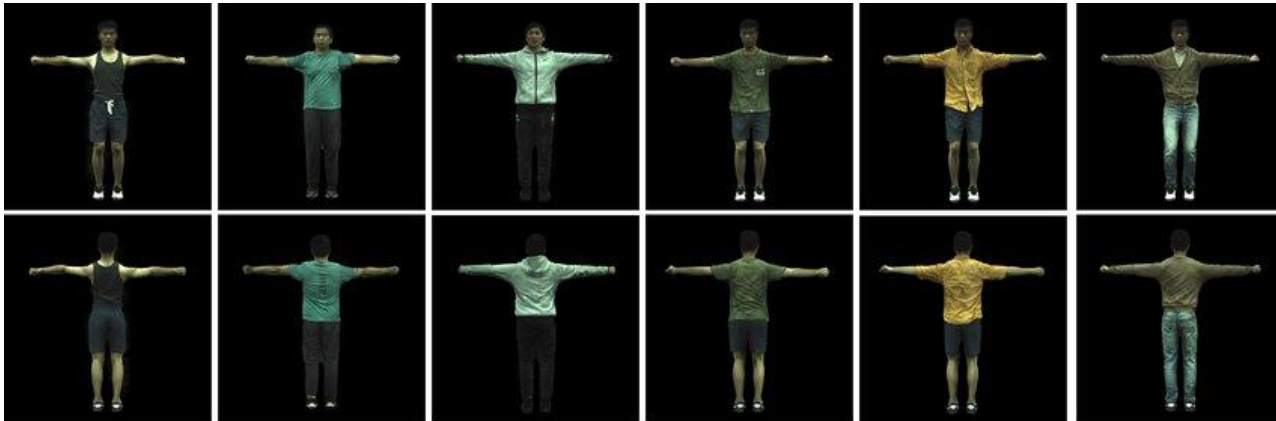
# Experiments

## • ZJU-MoCap Dataset<sup>1)</sup>

- 고정된 다중 뷰 카메라 구성 (Lab dataset)
- 동적 대상: 다양한 움직임을 보여주는 6명의 피사체

## • HUGS training

- 총 23개의 뷰 시스템에서 camera 1 으로 촬영된 이미지만을 사용
  - ※ 평가 데이터: 나머지 22개의 카메라로 촬영된 영상을 활용
- 카메라 행렬(camera matrices), body pose, segmentation 정보를 평가에 사용



<ZJU-MoCap Dataset>

# Experiments

- NeuMan, ZJU Mocap 데이터 세트에 대해 정량적 평가
  - 평가 지표: PSNR, SSIM, LPIPS

	Seattle			Citron			Parking			Bike			Jogging			Lab		
	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓
NeRF-T	21.84	0.69	0.37	12.33	0.49	0.65	21.98	0.69	0.46	21.16	0.71	0.36	20.63	0.53	0.49	20.52	0.75	0.39
HyperNeRF	16.43	0.43	0.40	16.81	0.41	0.56	16.04	0.38	0.62	17.64	0.42	0.43	18.52	0.39	0.52	16.75	0.51	0.23
Vid2Avatar	17.41	0.56	0.60	14.32	0.62	0.65	21.56	0.69	0.50	14.86	0.51	0.69	15.04	0.41	0.70	13.96	0.60	0.68
NeuMan	23.99	0.78	0.26	24.63	0.81	0.26	25.43	0.80	0.31	25.55	0.83	0.23	22.70	0.68	0.32	24.96	0.86	0.21
HUGS	25.94	0.85	0.13	25.54	0.86	0.15	26.86	0.85	0.22	25.46	0.84	0.13	23.75	0.78	0.22	26.00	0.92	0.09

## <NeuMan Dataset>

	377			386			387			392			393			394		
	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓
NeuralBody	29.11	0.97	0.04	30.54	0.97	0.05	27.00	0.95	0.06	30.10	0.96	0.05	28.61	0.96	0.06	29.10	0.96	0.05
HumanNerf	30.41	0.97	0.02	33.20	0.98	0.03	28.18	0.96	0.04	31.04	0.97	0.03	28.31	0.96	0.04	30.31	0.96	0.03
MonoHuman	30.77	0.98	0.02	32.97	0.97	0.03	27.93	0.96	0.03	31.24	0.97	0.03	28.46	0.96	0.03	28.94	0.96	0.04
HUGS	30.80	0.98	0.02	34.11	0.98	0.02	29.29	0.97	0.03	31.36	0.97	0.03	29.80	0.97	0.03	30.54	0.97	0.03

## <ZJU-MoCap Dataset>

# HOSNeRF: Dynamic Human-Object-Scene Neural Radiance Fields from a Single Video

# Background

- Object reconstruction in 3D

- 기존 문제점

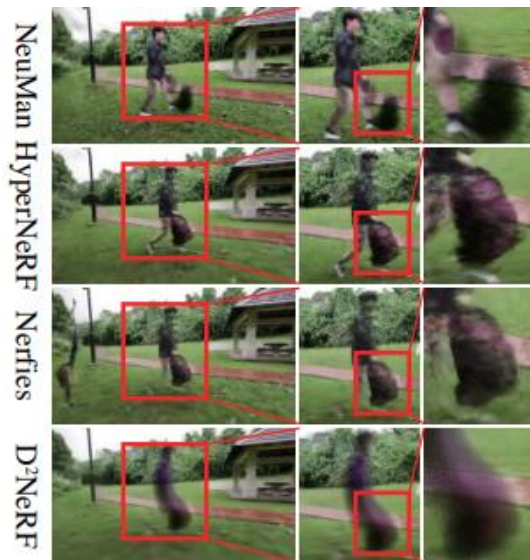
- 객체의 복잡한 변형 처리 부족

- ※ 사람-객체 상호작용이 복잡해질 경우, 객체가 깨지는 현상 발생

- 객체 상태 변화에 대한 대응 부족

- ※ 사람: 손에 쥐고 있던 객체를 놓을 경우, 형상 변화에 대해 반영하지 못함

- ※ 배경: 기존 배경에 새롭게 추가된 물체를 추가하지 못함



<객체의 복잡한 변형 처리 부족> 20

<객체 상태 변화에 대한 대응 부족>

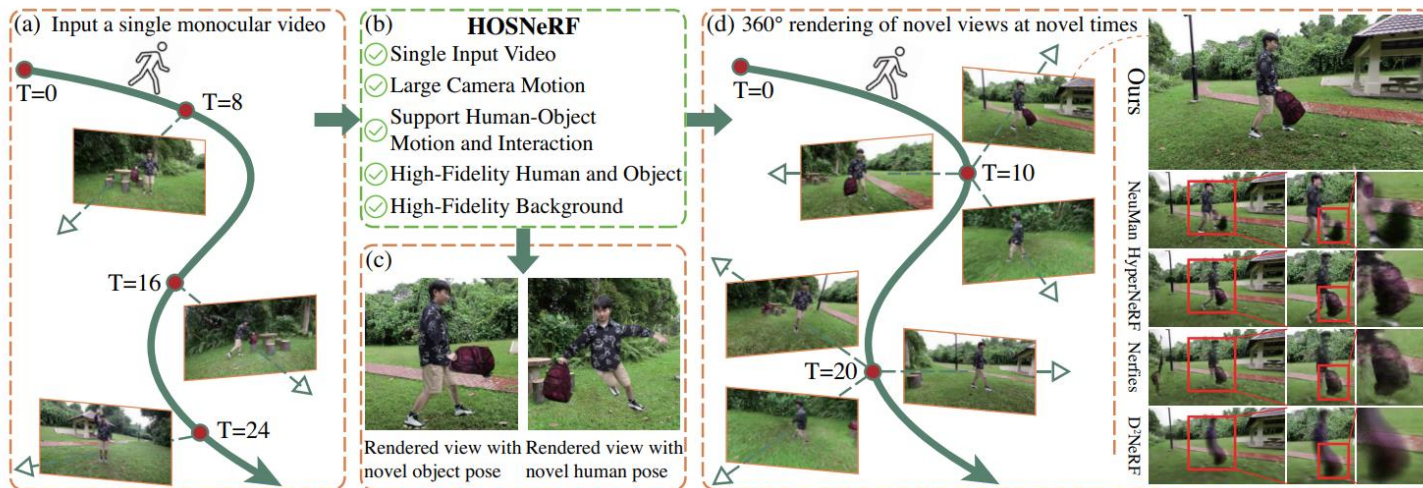
# Introduction

- Contribution

- Framework 최초 제시: 동적 인간-물체 상호작용 rendering (single video)
- Object bone 도입: 인간-물체 상호작용을 고품질 rendering 가능
- Object 상태 변화 처리 가능: State-Conditional Representation 개발

- Conclusion 요약

→ 두 개의 데이터 세트에 대해, SOTA 성능 달성 (LPIPS: 40%~50% 향상)



<HOSNeRF>

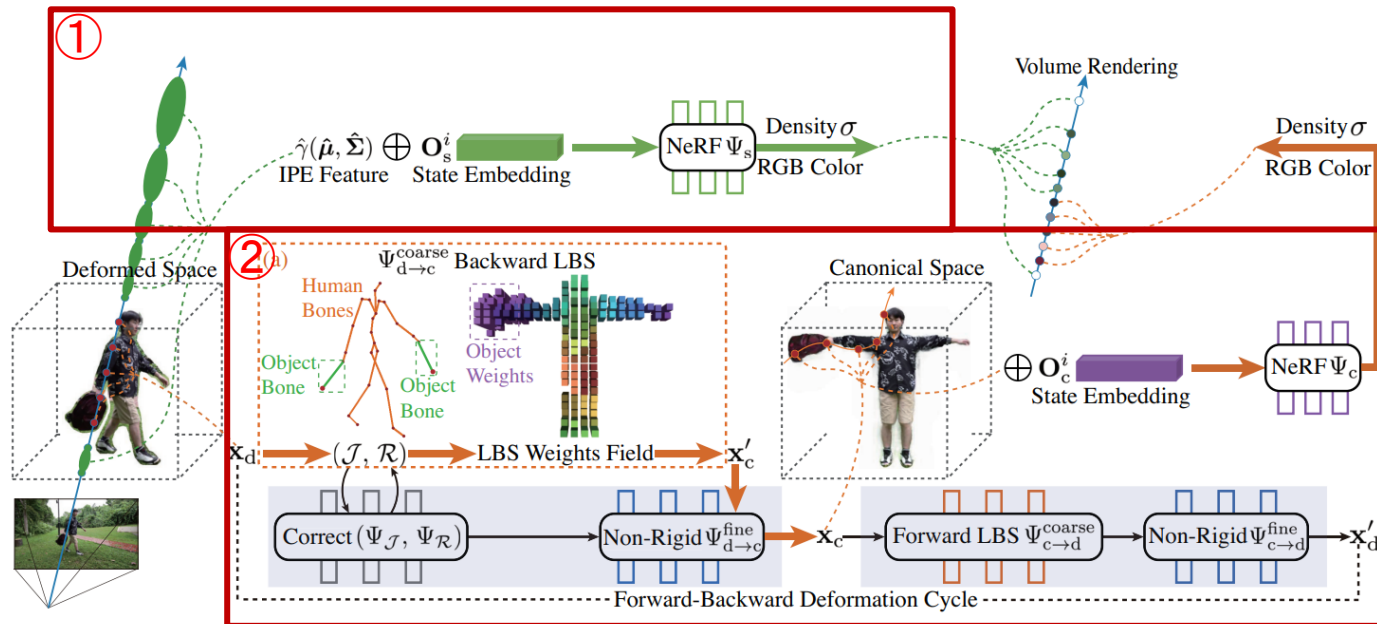
# Method

- HOSNeRF overview

- 크게 2파트로 구분

- Background 처리 MLP (Base: Mip-NeRF 360<sup>1)</sup>)

- Human 처리 MLP (Base: HumanNeRF<sup>2)</sup>) (하위 3파트)



<HOSNeRF overview>

# Method

- Input: Multi-frame (Monocular video)

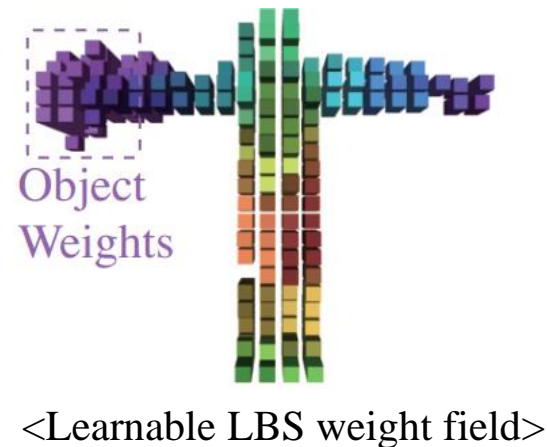
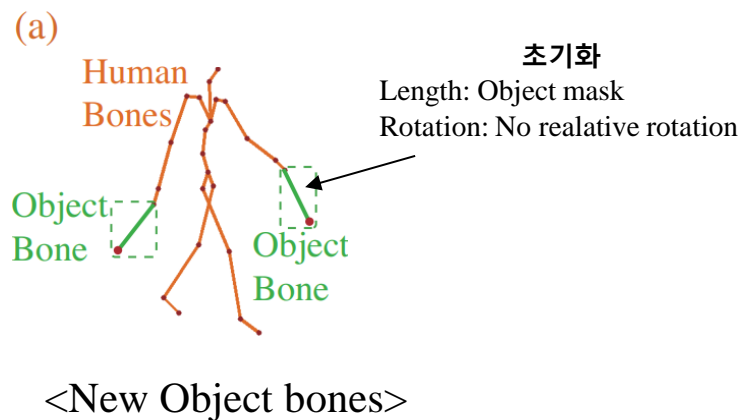
- Output

- Human pose (Use ROMP<sup>1)</sup> model)
    - Human-object segmentation mask (Mask-RCNN)

- Part 1: Dynamic Human-object Model

- Object Bones 도입

- Object LBS 생성 목적: 부드러운 object motion 생성 가능
    - New Hierarchical pose set,  $J=(J_{\text{human}}, J_{\text{object}})$ ,  $R=(w_{\text{human}}, w_{\text{object}})$

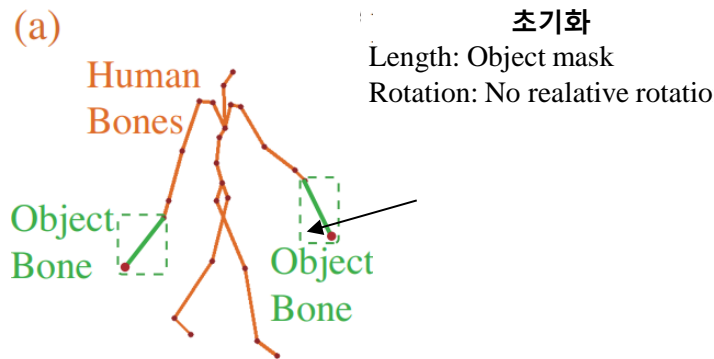


# Method

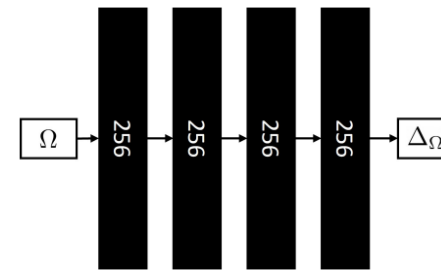
- Part 1: Dynamic Human-object Model

- New object bone pose 계산 방법

- Pose correction MLP: Pose를 입력 받은 후, 비선형적 변환을 통해 Pose 변화량 계산



<New object bone pose>



<Pose correction module / HumanNeRF<sup>1</sup>>



# Method

- Part 2: Deform to Canonical Space

- Learnable LBS weight field

- Input: Latent code (random variable)  $R^{256}$

- Output:  $32 \times 32 \times 32 \times 27$  (24 + 2 object bones + 1 background)

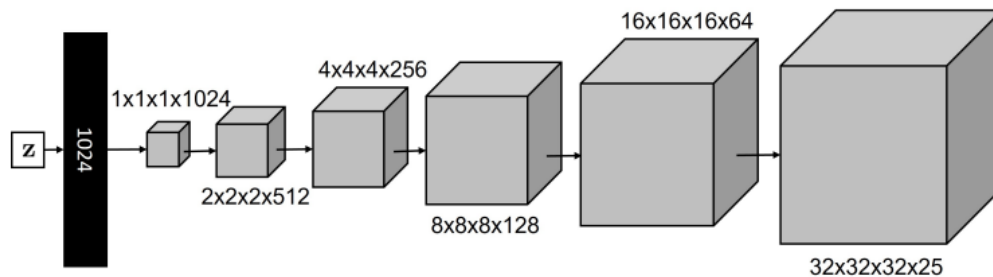
- CNN parameter explain the LBS weights

- 1개의 채널은 각 관절에 대한 weights (27개 값)

- ※ 3차원 공간으로 표현

- Canonical  $x, y, z$  기준으로 표현: 간접적으로 3D LBS weight map 생성

- 해당 weight 값을 바탕으로 LBS 적용



<LBS: CNN module / HumanNeRF<sup>1</sup>>



<LBS weight field>

# Method

- Part 2: Deform to Canonical Space

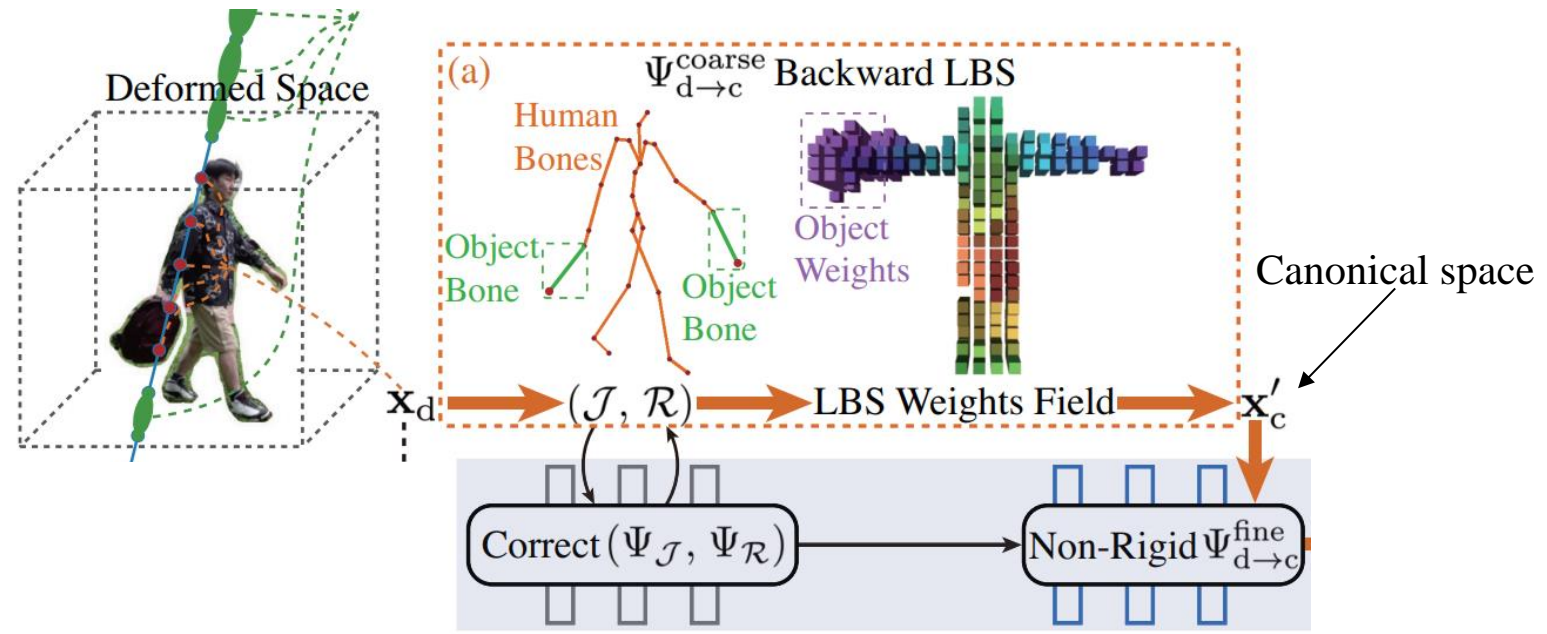
- Deformed space (World) → Canonical space (coarse)

- Input:  $(J, x_d)$ ,  $x_d$  는 Human vertex로 가정

- ※ Joint 위치 값을 알면, T-pose 같은 canonical space로 변환 가능

- LBS weight 값을 통해, canonical space 의 좌표로 변환

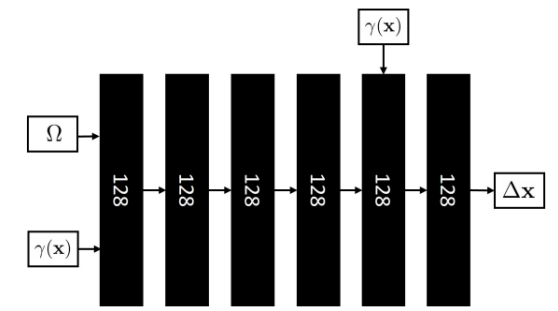
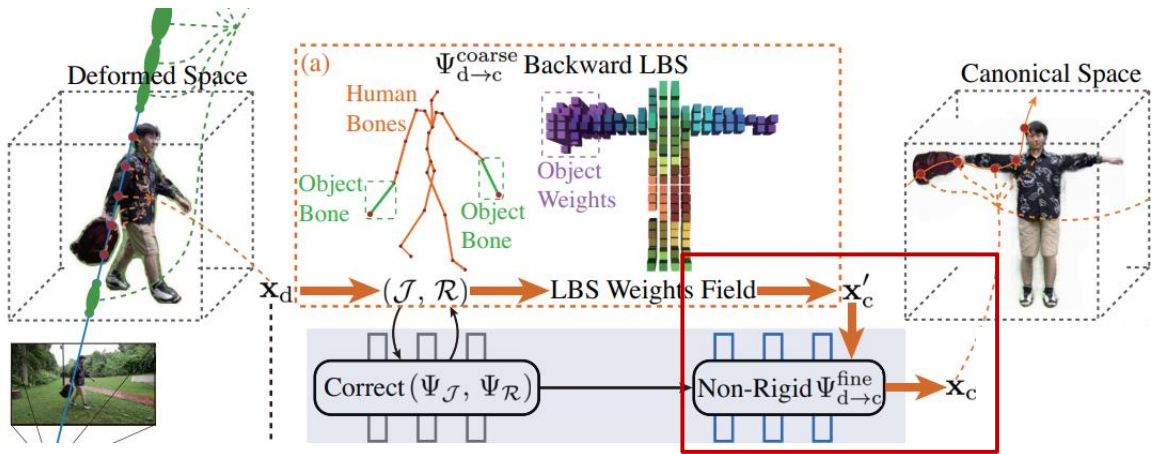
- Coarse  $x'_c$  를 MLP에 입력: canonical space 에 mapping 성능 강화 (fine)



<Deformed space to canonical space>

# Method

- Part 2: Deform to Canonical Space
  - Canonical space (coarse)  $\rightarrow$  Canonical space (fine)
    - Canonical space 에 mapping 성능 강화 목적
  - Input: Canonical  $x'$ , Pose (J)
    - Pose correction output을 입력 받아, correction module 가능 구현



< Canonical space (coarse)  $\rightarrow$  Canonical space (fine) >

< Non-rigid motion MLP / HumanNeRF<sup>1</sup> >

# Method

- Part 3: Object state embedding space

- Object 상태를 표현하는 embedding space를 정의

- State 수: Object에 대한 현재 상황 수

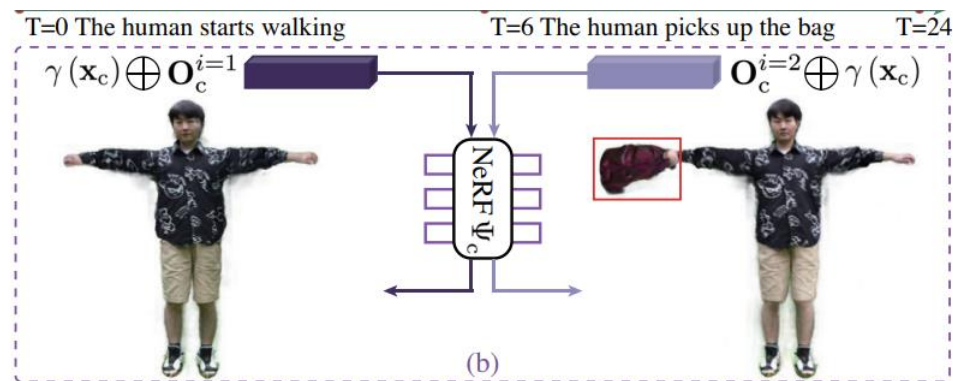
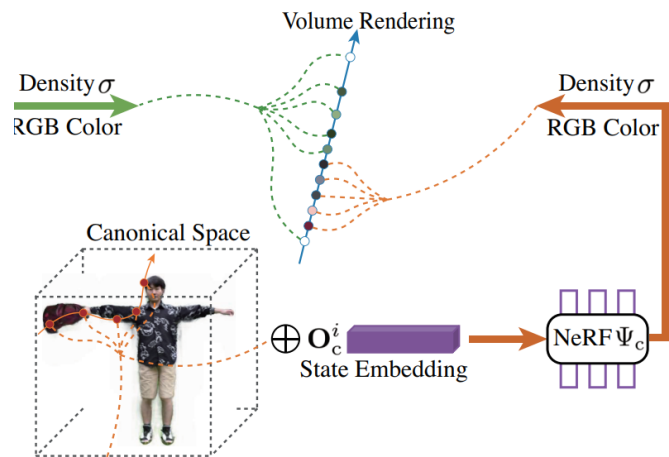
- 예: 가방을 쥐고 놓은 Scene  $\rightarrow$  3개의 state (잡기 전, 잡은 상황, 놓은 후)

- Object set:  $\{O_c^i\}$  ( $i = 1, 2, \dots, N$ )

- N: State 수

- $O_c^i$ : 1 x 64 크기로 학습 가능한 파라미터

$$\Psi_c (\text{concat} (\gamma (\mathbf{x}_c), \mathbf{O}_c^i)) \mapsto (\mathbf{c}, \sigma)$$



<Object state embedding space>

# Method

- Part 4: Scene reconstruction

- Mip-NeRF 360<sup>1)</sup>을 기반으로 한 정적 배경 모델링

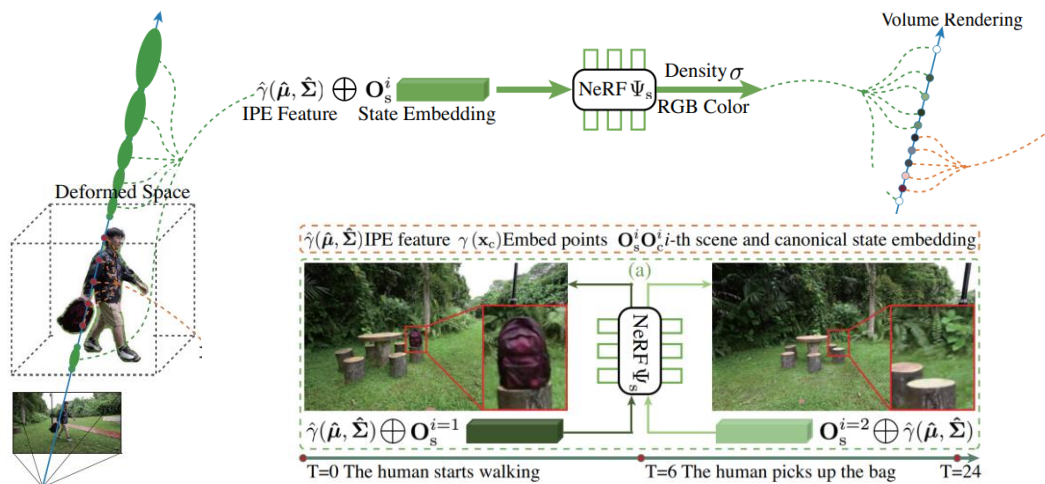
- 픽셀에 대한 정보:  $\mu, \Sigma$  로 표현

- Object state processing

- 상태 임베딩(space embedding) 추가

- $\Psi_s \left( \text{concat} \left( \hat{\gamma} \left( \hat{\mu}, \hat{\Sigma} \right), \mathbf{O}_s^i \right) \right) \mapsto (\mathbf{c}, \sigma)$

- IPE encoded + scene state embedding을 MLP의 입력으로 사용



<Object state aware scene reconstruction>

# Method

- Part 5: Training Objectives

- Cycle loss

- 변형(deformation)이 forward, backward process 모두 일관되게 동작하도록 규제

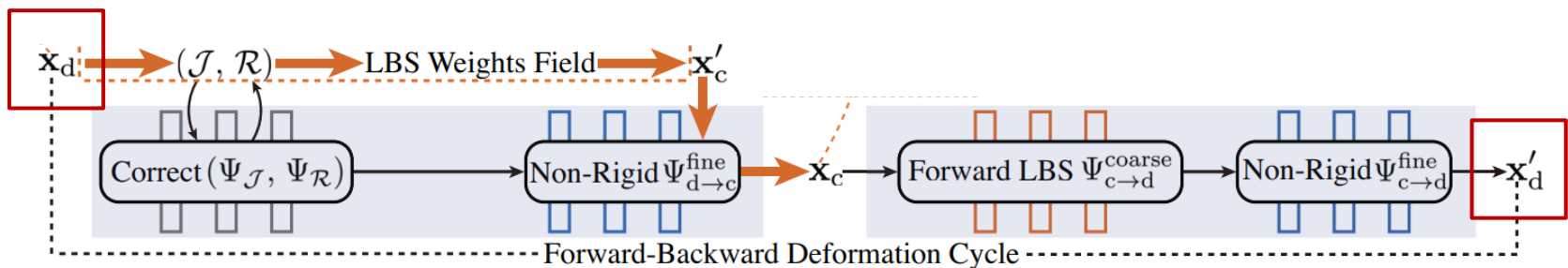
- 원리

- Deformed 공간에서 Canonical 공간로 mapping (Backward)

- Canonical 공간에서 다시 Deformed 공간으로 역 mapping (Forward)

- 처음 deformed 공간의 좌표와 역 mapping 결과가 일치하도록 학습

$$\mathcal{L}_{\text{Cycle}} = \frac{1}{2N} \sum_{i=1}^N \|\mathbf{x}_d^i - \hat{\mathbf{x}}_d^i\|_2^2$$



<Deformation cycle>

# Method

- Part 5: Training Objectives

- Human 과 background를 각각 렌더링하여, 합칠 경우 경계 artifact 발생

- 해결 방법: Human, background 분리 mask를 사용해 모델을 각각 학습

- Foreground-Background 렌더링 병합 과정

- 각각의 모델(Scene, Human-Object)에서 광선(ray)을 따라 샘플링된 점을 정렬

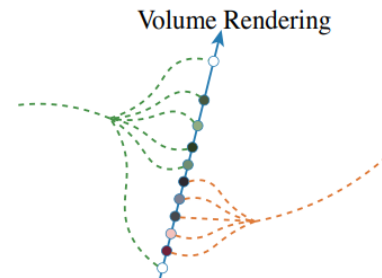
- ※ 샘플 포인트 별, 카메라 원점으로부터의 거리 계산

- 학습 과정: 총 3단계로 진행

- Stage 1 (Scene model)

- Using background pixels only

- $L_{MSE}$ : Photometric MSE Loss



<거리 기반 sampling point 정렬>

# Method

- Part 5: Training Objectives

- Stage 2 (Human-object model)

- Using random solid background with Human-object regions

- Image relate loss

- ☼  $\mathcal{L}_{\text{MSE}}$  : Photometric MSE Loss

- ☼  $\mathcal{L}_{\text{LPIPS}}$  : Perceptual Loss

- Canonical, HOI model Loss

- ☼  $\mathcal{L}_{\text{Cycle}}$  : Forward-Backward Deformation 간의 일관성을 유지

- ☼  $\mathcal{L}_{\text{Flow}}$  : Pretrained 모델로 2D Optical Flow를 예측하여 프레임 간 일관된 움직임 유지

- Stage 3: Full model fine-tuning

- Human-Object와 Background 모델을 결합해 전체를 한번에 fine-tuning

$$\begin{aligned}\mathcal{L} &= \omega_{\text{MSE}} \cdot \mathcal{L}_{\text{MSE}} + \omega_{\text{LPIPS}} \mathcal{L}_{\text{LPIPS}} \\ &+ \omega_{\text{Cycle}} \cdot \mathcal{L}_{\text{Cycle}} + \omega_{\text{Flow}} \cdot \mathcal{L}_{\text{Flow}}\end{aligned}$$

$$\omega_{\text{MSE}}: 0.2, \omega_{\text{LPIPS}}: 1.0, \omega_{\text{Cycle}}: 0.01, \omega_{\text{Flow}}: 0.01$$

<Final loss term>



# Experiment

- HOSNeRF Dataset

- 데이터 특징

- 인간이 특정 객체를 잡거나 놓는 등의 상호작용 과정을 학습할 수 있는 데이터

- 데이터 구성

- 6개의 scene으로 구성

- 비디오 길이: 60초 ~ 120초

- 전체 중, 16개의 프레임을 선택하여 novel views와 novel timesteps에서 테스트

Scene	No. of objects	No. of states
BACKPACK	1	3
TENNIS	2	3
SUITCASE	1	4
PLAYGROUND	3	5
DANCE	3	7
LOUNGE	3	5

<HOSNeRF Dataset 세부사항>

# Experiment

- HOSNeRF Dataset, NeuMan 데이터에 대해 정량적 평가
  - 평가 지표: PSNR, SSIM, LPIPS

	BACKPACK			TENNIS			SUITCASE			PLAYGROUND			DANCE			LOUNGE		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
K-Planes [7]	19.05	0.345	0.557	19.31	0.690	0.536	18.64	0.575	0.602	17.92	0.446	0.635	18.17	0.328	0.623	24.21	0.824	0.453
D <sup>2</sup> NeRF [57]	20.52	0.384	0.608	23.97	0.737	0.540	20.99	0.592	0.645	21.23	0.510	0.616	19.92	0.407	0.647	27.13	0.858	0.509
Nerfies [32]	19.56	0.515	0.559	22.12	0.818	0.443	19.01	0.643	0.555	21.14	0.726	0.533	19.37	0.674	0.524	25.90	0.914	0.342
HyperNeRF [33]	19.62	0.359	0.587	21.26	0.742	0.510	19.41	0.604	0.607	21.67	0.558	0.578	19.30	0.412	0.601	27.25	0.925	0.332
NeuMan [14]	21.21	0.479	0.478	23.17	0.768	0.442	20.84	0.611	0.551	21.46	0.546	0.551	21.19	0.529	0.490	28.40	0.917	0.341
Ours (base)	21.51	0.764	0.271	24.02	0.910	0.326	21.10	0.829	0.395	22.20	0.787	0.348	21.84	0.785	0.266	26.54	0.962	0.243
Ours w/ state	22.38	0.786	0.252	23.98	0.910	0.323	21.43	0.834	0.390	22.52	0.796	0.341	22.43	0.796	0.258	27.51	0.965	0.245
Ours w/ object	21.56	0.767	0.269	23.98	0.909	0.327	21.08	0.830	0.396	22.14	0.787	0.347	21.88	0.785	0.269	26.80	0.962	0.246
Ours w/ state, object	22.33	0.785	0.253	24.08	0.909	0.320	21.51	0.833	0.386	22.55	0.796	0.338	22.38	0.796	0.259	27.52	0.965	0.245
Ours w/ state, object, mask	22.48	0.790	0.245	24.19	0.911	0.321	21.60	0.834	0.382	22.77	0.799	0.335	22.48	0.802	0.251	27.73	0.968	0.227
Ours (full)	22.56	0.792	0.243	24.15	0.911	0.320	21.74	0.836	0.382	22.67	0.801	0.336	22.63	0.804	0.248	27.74	0.968	0.227

<HOSNeRF Dataset>

	SEATTLE			PARKING			BIKE			JOGGING			LAB			CITRON		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
NSFF [22]	21.84	0.69	0.37	21.98	0.69	0.46	21.16	0.71	0.36	20.63	0.53	0.49	20.52	0.75	0.39	12.33	0.49	0.65
HyperNeRF [33]	16.43	0.43	0.40	16.04	0.38	0.62	17.64	0.42	0.43	18.52	0.39	0.52	16.75	0.51	0.23	16.81	0.41	0.56
NeuMan [14]	23.98	0.77	0.26	25.43	0.79	0.31	25.52	0.82	0.23	22.68	0.67	0.32	24.93	0.85	0.21	24.71	0.80	0.26
Ours	26.68	0.91	0.14	27.20	0.93	0.12	26.09	0.93	0.10	25.04	0.89	0.16	24.93	0.94	0.10	24.44	0.90	0.18

<NeuMan Dataset>