Neural Human Rendering

Recent novel view synthesis method of dynamic human performance

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

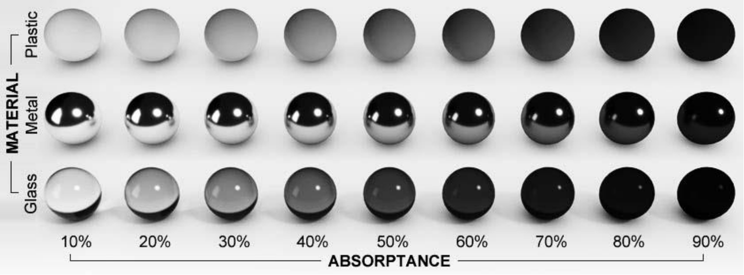
Presented By
Hosung Son

Contents

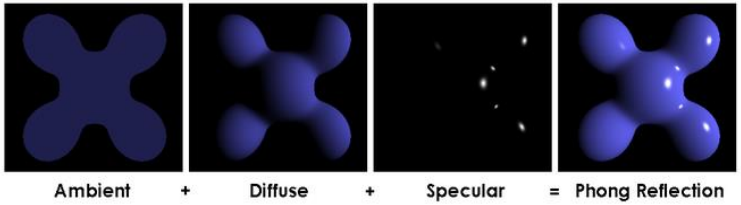
1) James T. Kajiya, The rendering equation, *ACM SIGGRAPH Computer Graphics, Vol 20, Issue 4*, 1986

What is neural rendering?

- Computer graphics in 3d rendering
  - Physical parameters from object and camera
    - Light transport
      - Absorption
      - Reflection
      - Scattering
    - Material properties
    - Camera parameters for image projection
  - All parameters should be input rendering model for high-quality reconstruction
  - Rendering equation[1)]
    - Consider only emitted. scattered light and geometry
    - More considerations are in next version of the eq.
      - Reflection, Transmission...
    - Ray tracing, ray marching, path tracing



Visual differences according to light absorptance



Phong shading model

$$I(x, x') = g(x, x')\left[\epsilon(x, x') + \int_S \rho(x, x', x'')I(x', x'')dx''\right]$$

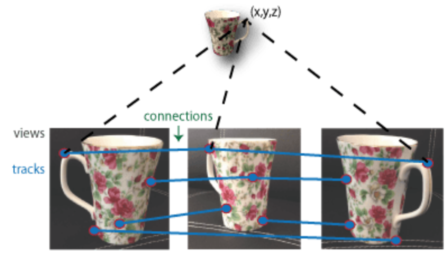
where:

$I(x, x')$	is the related to the intensity of light passing from point x' to point x
$g(x, x')$	is a "geometry" term
$\epsilon(x, x')$	is related to the intensity of emitted light from x' to x
$\rho(x, x'x'')$	is related to the intensity of light scattered from x'' to$x$ by a patch of surface at x'

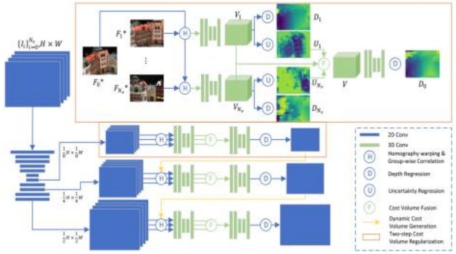
Rendering equation

What is neural rendering?

- Computer graphics in 3d rendering
  - Surface rendering
    - MVS with SfM [COLMAP]
      - Feature extracting/matching algorithm
      - Triangulation
    - MVS with Neural Networks
      - Feature extraction using DNN
      - Matching cost volume (Homography)
      - Depth estimation per view (Regression)
  - Volume rendering
    - Based on ray casting method
      1) Casting rays
      2) Sampling
      3) Shading
      4) Compositing



Structure-from-Motion


MVS Network architecture

Point cloud results of Vis-MVSNet

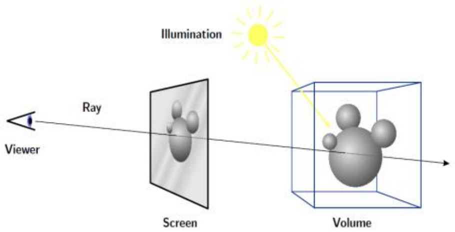
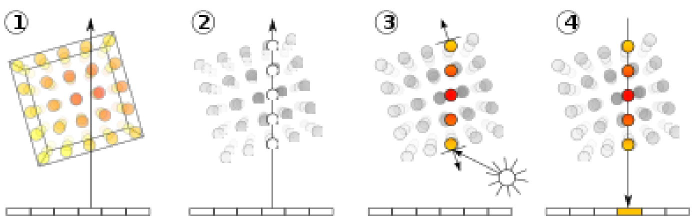


Illustration of ray casting

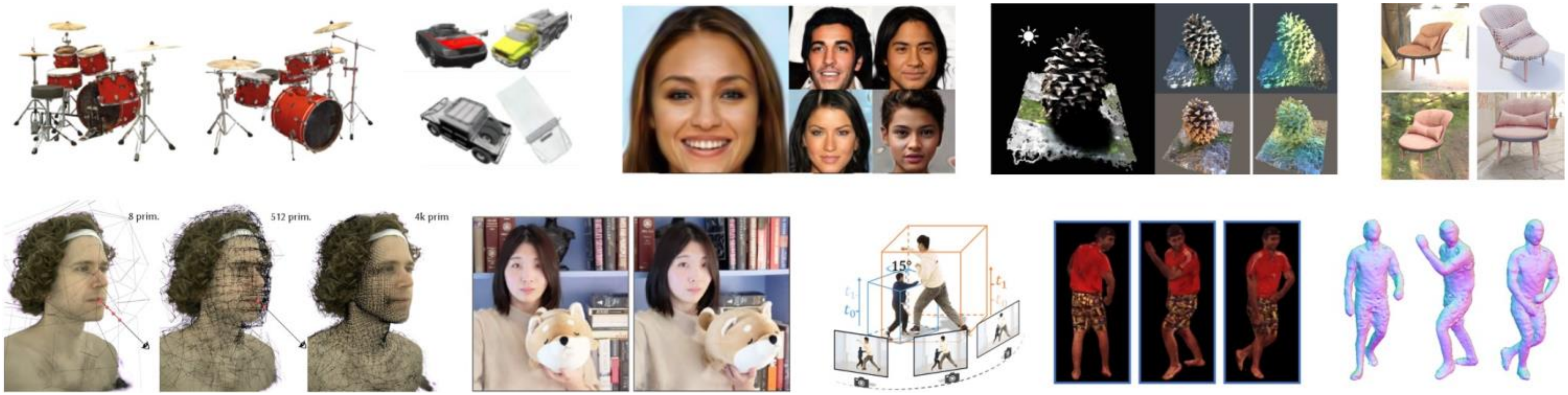

(1) Ray casting, (2) Sampling, (3) Shading, (4) Compositing

서강대학교
SOGANG UNIVERSITY

VDS LAB

What is neural rendering?

- Neural rendering

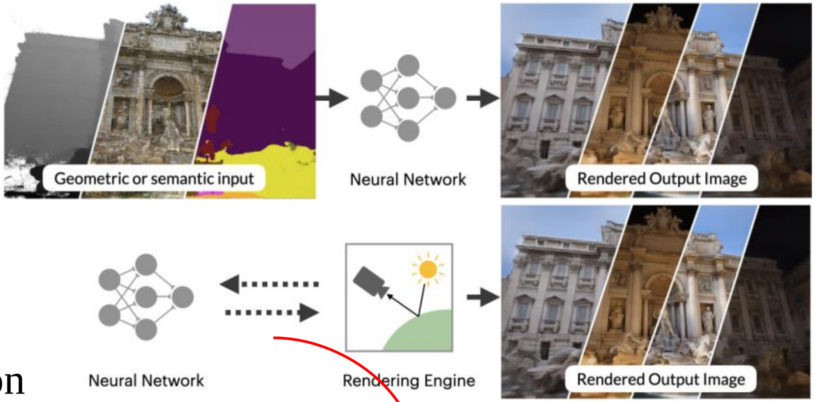


Many kinds of neural rendering tasks

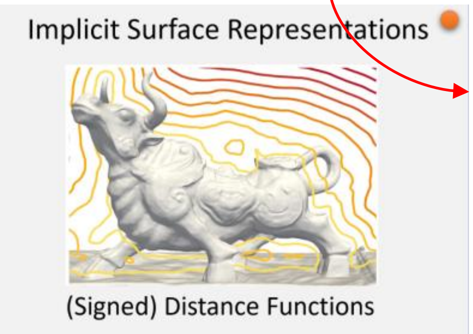
- Shape and appearance rendering combining two insights.
  - Classical Computer Graphics
  - Deep Neural Network
- Optimizing functions with Neural network such as MLPs
  - Non-linear optimization
  - Optimization strategies

Fundamentals of Neural Rendering

- Key points of Neural rendering
  - Disentanglement of camera capturing process and 3D scene representation
  - Neural rendering methods should be differentiable for training
- Scene representation
  - Surface representation
    - Explicit → point cloud, polygon mesh
      - $S_{explicit} = \left\{ \begin{pmatrix} x \\ y \\ f_{explicit}(x,y) \end{pmatrix} \middle| \begin{pmatrix} x \\ y \end{pmatrix} \in \mathbb{R}^2 \right\}$
    - Implicit → zero level set of implicit function
      - $S_{implicit} = \left\{ \begin{pmatrix} x \\ y \\ z \end{pmatrix} \in \mathbb{R}^2 \middle| f_{implicit}(x,y,z) = 0 \right\}$
  - Volumetric representation
    - Densities, opacities or occupancies
    - Multi-dimensional features
      - colors, radiance



Difference between 2D and 3D neural rendering



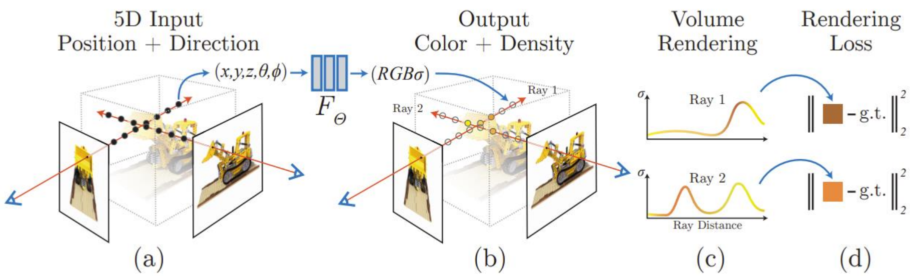
differentiable scheme from computer graphics which are motivated by physics.

Implicit surface representation of SDF

1) B. Mildenhall, NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis, ECCV2020
2) Kajiya,, Ray tracing volume densities. Computer Graphics(SIGGRAPH) (1984)
3) Max, N., Optical models for direct volume rendering. IEEE Transactions on Visualization and Computer Graphics (1995)

Novel view synthesis

- View synthesis of static contents
  - Neural Radiance Fields (NeRF[1)])



– Volume rendering with radiance fields[2)]

$$C(\mathbf{r}) = \int_{t_n}^{t_f} T(t)\sigma(\mathbf{r}(t))\mathbf{c}(\mathbf{r}(t),\mathbf{d})dt\,, \text{ where } T(t) = \exp\left(-\int_{t_n}^{t}\sigma(\mathbf{r}(s))ds\right)$$

$C(\mathbf{r})$: expected color
$\sigma(\mathbf{x})$: volume density
$\mathbf{r}(t) = \mathbf{o} + t\mathbf{d}$: camera ray
t_n: near bound
t_f: far bound
$\delta_i = t_{i+1} - t_i$: distance between samples

Quadrature rule

– Uniform ray sampling

$$t_i \sim \mathcal{U}\left[t_n + \frac{i-1}{N}(t_f - t_n),\ t_n + \frac{i}{N}(t_f - t_n)\right]$$

– Discretized representation of volume rendering[3)]

$$\hat{C}(\mathbf{r}) = \sum_{i=1}^{N} T_i(1-\exp(-\sigma_i\delta_i))\mathbf{c}_i\,, \text{ where } T_i = \exp\left(-\sum_{j=1}^{i-1}\sigma_j\delta_j\right)$$

Volume rendering eq. from NeRF[1)]

$=$

$$C(\mathbf{r}) = \sum_{i=1}^{D}\left(\prod_{j=1}^{i-1}(1-\alpha_j)\right)\alpha_i\mathbf{c}(\mathbf{x}_i),$$
$$\alpha_i = 1 - \exp(-\sigma(\mathbf{x}_i)\Delta t_i),$$

Volume rendering eq. from HumanNeRF

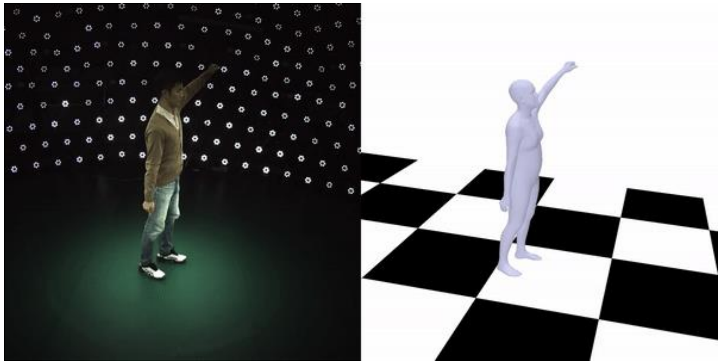
→ Differentiable!!

1) Peng, Neural Body: Implicit Neural Representations with Structured Latent Codes for Novel View Synthesis of Dynamic Humans, *CVPR2021*

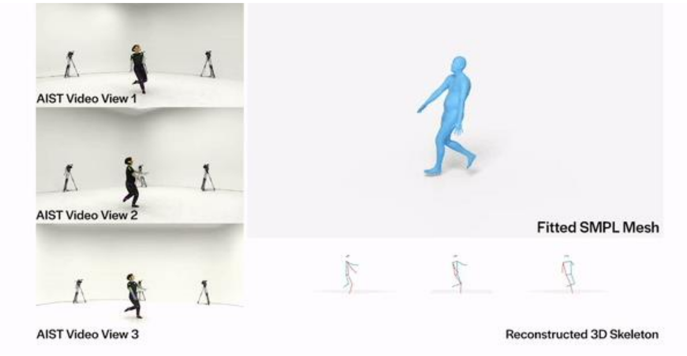
2) R. Li, AI Choreographer: Music Conditioned 3D Dance Generation with AIST++, *ICCV2021*

Novel view synthesis

- View synthesis of Non-static contents
  - Human performance dataset (Multi-view)



ZjU-MoCaP[1)]



AIST++[2)]

  - "In the wild" monocular videos from Youtube

story

invisible

way2sexy

1) K. Park, A Higher-Dimensional Representation for Topologically Varying Neural Radiance Fields, ACM Transactions on Graphics(TOG), Vol 40(6), 2021
2) Peng, Neural Body: Implicit Neural Representations with Structured Latent Codes for Novel View Synthesis of Dynamic Humans, *CVPR2021*

Novel view synthesis

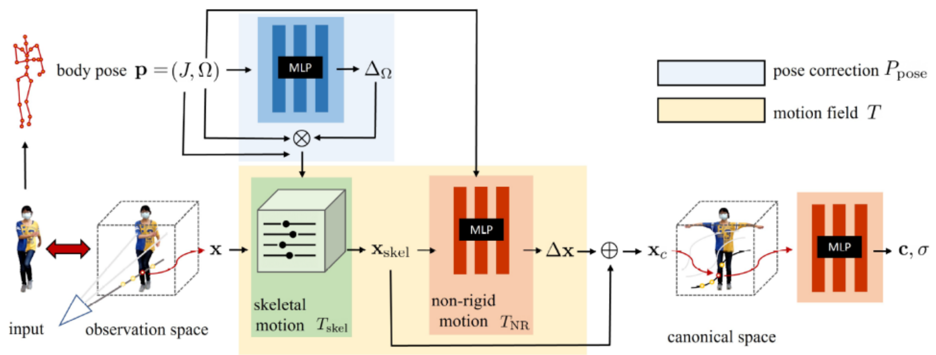
- View synthesis of Non-static contents
  - Time varying in motion
    - Since human performance is time varying, it is difficult to generate generalized appearance.
    - If the motion is highly dynamic, unintentional artifacts could be input to model.
  - Deformation
    - Human face
      - expression
    - Clothes
    - Etc...
  - Occlusion
    - If not a multi-view contents, occlusion is very likely to occur.
    - Cross section of human body parts

NeRF with face deformation [HyperNeRF[1)]]

NeRF with dynamic motion [NeuralBody[2)]]

Paper – HumanNeRF [CVPR2022]

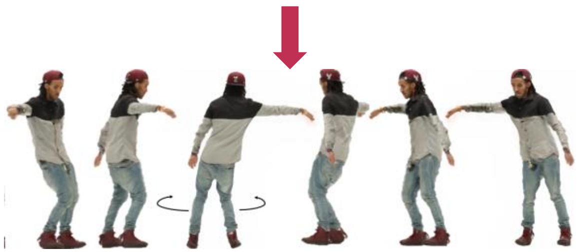
- Overall architecture



HumanNeRF teaser video

- Input
  - Monocular video of complex human performance
  - Human, camera pose at each frame (<u>Not use template</u>)
- Output
  - Free-viewpoint rendering for any frame in the sequence
- Components
  - Pose refiner: $\Delta\Omega$
  - Motion fields: T_{skel}, T_{NR}
  - Canonical Volume: F_c

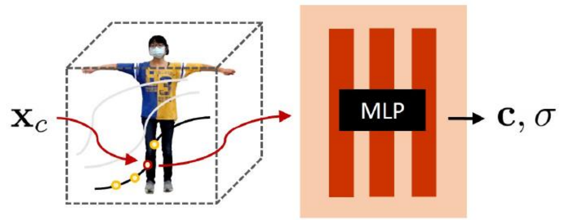
Monocular video



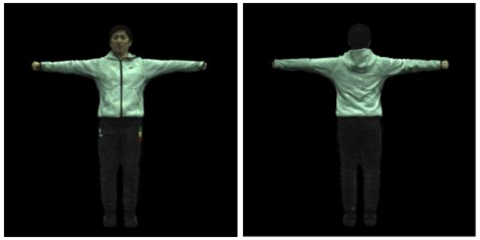
free-viewpoint rendering

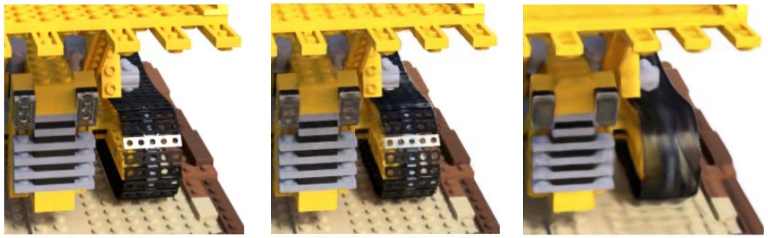
Paper – HumanNeRF [CVPR2022]

- 3D canonical space
  - Canonical volume F_c
    - Continuous field with an MLP
    - Outputs color **c** and density $\boldsymbol{\sigma}$
    - $F_c(\mathbf{x}) = \mathrm{MLP}_{\theta_c}(\gamma(\mathbf{x}))$



Canonical space


Optimized canonical appearance
[ZjU-MoCaP-Subject387]


Ground Truth | Complete Model | No Positional Encoding

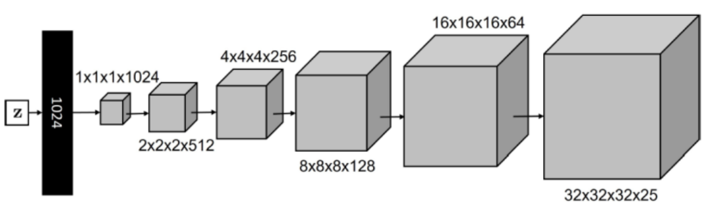
Positional Encoding

  - Positional encoding $\gamma(\mathrm{x})$
    - From NeRF[1], basic implementation is inefficient in the required number of samples per ray.
    - Positional encoding maps each input 5D coordinate into a higher dimensional space
      → It enables the MLP to represent higher <u>frequency functions</u>
      - Sinusoidal embedding function

$$\mathbf{x} \mapsto \{\sin(2^0\pi\mathbf{x}), \cos(2^0\pi\mathbf{x}), \cdots, \sin(2^{L-1}\pi\mathbf{x}), \cos(2^{L-1}\pi\mathbf{x})\}$$

Paper – HumanNeRF [CVPR2022]

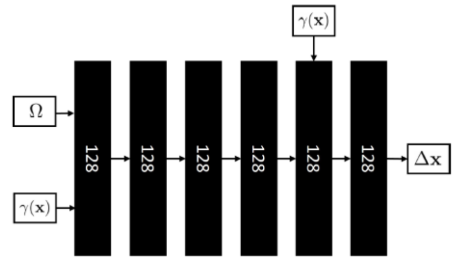
- Motion fields
  - Transformation between observation field and canonical space
  - To handle complex human movement with complex deformation by decomposing the motion field two parts
    - Skeletal motion field T_{skel} : Inverse (volumetric) linear-blend skinning
    - Non-rigid motion field T_{NR} : Complex deformation of non-rigid human appearance
  - Skeletal motion field T_{skel}[1)]
    - Skeletal motion field provides the coarse deformation driven by standard skinning.
    - $T_{skel}(\mathbf{x}, \mathbf{p}) = \sum_{i=1}^{K} \omega_o^i(\mathbf{x})(R_i\mathbf{x} + \mathbf{t}_i)$
    - $\omega_o^i(\mathbf{x}) = \frac{\omega_c^i(R_i\mathbf{x}+\mathbf{t}_i)}{\sum_{i=1}^{K} \omega_c^k(\mathbf{x})(R_k\mathbf{x}+\mathbf{t}_k)}$
      - Solving for a single set of weight volumes $\{\omega_c^i(\mathbf{x})\}$ in canonical space can lead better generalization as it avoids over-fitting.
    - $\{\omega_c^i(\mathbf{x})\} \coloneqq W_c(\mathbf{x}) = \mathrm{CNN}_{\theta_{\mathrm{skel}}}(\mathbf{x}; \mathbf{z})$



1) K. Park, HyperNeRF: A Higher-Dimensional Representation for Topologically Varying Neural Radiance Fields

Paper – HumanNeRF [CVPR2022]

- Motion fields
  - Non-rigid motion field T_{NR}
    - Output an offset Δx to the skeletal motion
    - $\Delta \mathrm{x}(\mathrm{x}, \mathbf{p}) = T_{NR}(T_{skel}(x, \mathbf{p}), \mathbf{p})$
    - $T_{NR}(x, \mathbf{p}) = \mathrm{MLP}_{\theta_{NR}}(\gamma(\mathrm{x}); \Omega)$
    - Estimate difficult deformation of high-deformable region in dynamic human motion



Non-rigid motion field network

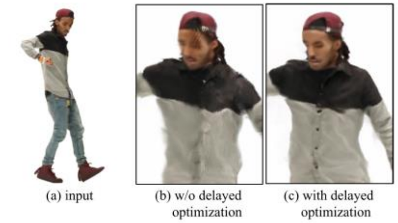


Figure 8. Without delayed optimization and strong decoupling of skeletal and non-rigd deformations, generalization to unseen views is poor (b). With delayed optimization, the decoupling leads to good generalization (c).

Effect of delayed optimization

  - Delayed optimization of non-rigid motion field[1)]
    - Disable non-rigid motions at the beginning of optimization until 100K (in practice)
    - Hann window was applied to frequency bands of positional encoding → prevent over-fitting

$$\omega(\alpha) = \frac{1 - \cos(\mathrm{clamp}(\alpha - j, 0, 1)\pi}{2}, \qquad \alpha(t) = L\frac{\max(0, t - T_s)}{T_e - T_s}, \qquad j \in \{0, \cdots, L-1\}$$

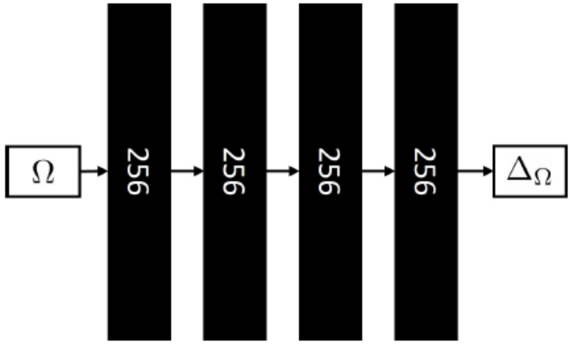
- $\omega(\alpha)$: weight for each frequency band j of positional encoding
- $\alpha(t)$: width of a truncated Hann window

$$\gamma_\alpha(x) = \{w_0 \sin(2^0 \pi \mathbf{x}), w_0 \cos(2^0 \pi \mathbf{x}), \cdots, w_{L-1} \sin(2^{L-1} \pi \mathbf{x}), w_{L-1} \cos(2^{L-1} \pi \mathbf{x})\}$$

- If $\alpha = 0$, non-rigid motion completely be disabled

Paper – HumanNeRF [CVPR2022]

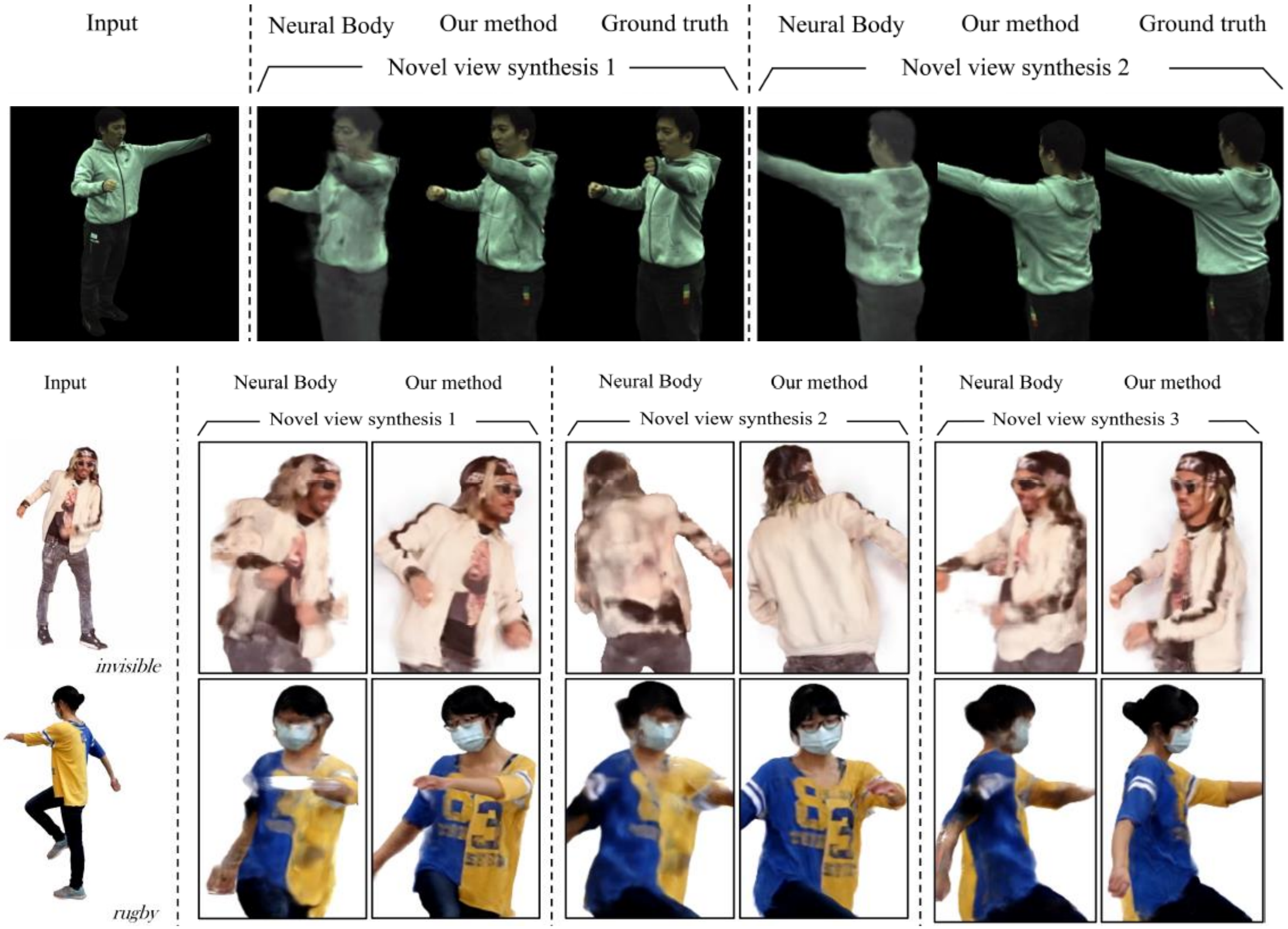
- Pose refinement module
  - Pose refinement
    - Initial human poses from ‘pose detector’ is not accuracy.
    - MLP based network outputs difference of joint rotative vectors ΔΩ.
    - Network better explains the observations and improve the skeleton-driven deformation.
    - Pose correction function P_{pose}
      - Consider 23 joints except for the root. (body orientation)
      - Instead, describe changes of global body orientation as camera rotations.



Pose refinement network architecture

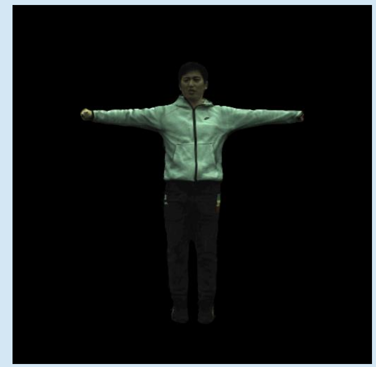
Paper – HumanNeRF [CVPR2022]

- Qualitative results



Paper – HumanNeRF [CVPR2022]

- Renderings
  - Zju-MoCaP - subject 387

Frame inputs	Free-view (Frame 1)	T-pose
		

- Performance

	Subject **377**			Subject **386**			Subject **387**		
	PSNR ↑	SSIM ↑	LPIPS* ↓	PSNR ↑	SSIM ↑	LPIPS* ↓	PSNR ↑	SSIM ↑	LPIPS* ↓
Neural Body [48]	29.11	0.9674	40.95	30.54	0.9678	46.43	27.00	0.9518	59.47
Ours	30.41	0.9743	24.06	33.20	0.9752	28.99	28.18	0.9632	35.58

	Subject **392**			Subject **393**			Subject **394**		
	PSNR ↑	SSIM ↑	LPIPS* ↓	PSNR ↑	SSIM ↑	LPIPS* ↓	PSNR ↑	SSIM ↑	LPIPS* ↓
Neural Body [48]	30.10	0.9642	53.27	28.61	0.9590	59.05	29.10	0.9593	54.55
Ours	31.04	0.9705	32.12	28.31	0.9603	36.72	30.31	0.9642	32.89

Paper – HumanNeRF [CVPR2022]

- Limitations
  - Occlusion
    - It has artifacts when part of the body is not shown in the video (occlusion)
  - Pose correction failure
    - If the initial pose estimate is poor or the image contains strong artifact such as motion blur
  - Non-rigid motion covering
    - Authors assume non-rigid motion is pose-dependent, but it is not always true
    - E.g. clothes shifting due to wind or due to follow-through after dynamic subject motion
  - Only consider human scene, not background scene