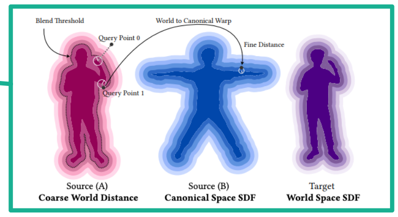
Relightable & Animatable Neural Avatars



1. Terminology

World Space: Coordinate landscape independent of camera orientation, instead $\exists$ a world origin.

Screen Space: Coordinates once mapped to a 2D image, from a camera orientation.

Canonical Space: Coordinates mapped to a specific pose

Signed distance function (SDF): distance from point to nearest surface of human, positive if inside human, negative if outside.

2. Objective

Given sparse-view or monocular video of a human performer, create an avatar that's:

- Animatable: we can put into new poses
- Relightable: will have correct light emitted under new lighting conditions (background, position)

3. Hierarchical Distance Query

For each point $x \in \mathbb{R}^3$, we wish to compute it's SDF.

a) Coarse distance query

- Creates initial SDF using KNN

b) Inverse warping

- Transform world space $\rightarrow$ canonical space using Linear Blend Skinning

c) Fine distance query

- MLP to correct the SDF based on which pose the human had

d) Smooth distance blending

- Combine coarse & fine distances
- on edges, take coarse SDF since very similar to fine.

4. Geometry

Our renderer needs : Surface point x_s

Surface normal n_s

Light visibility $V(x_s, \omega_i)$

which can be easily found from the SDF....

a) take a ray $r(t) = o + td$

b) march along it until find $SDF = 0 \Rightarrow x_s$

c) compute surface normal off $\frac{\partial SDF}{\partial x}$

5. Reflectance (emitted colour of each vertex)

Microfacet BRDF model $R_s(x_s, \omega_i, \omega_o, n_s)$

where R_s = reflectance at vertex s

x_s = Vertex s position

ω_i = Incoming light source direction

ω_o = Outgoing (towards camera) light direction

n_s = Surface normal of x_s

Within BRDF, $\exists$ 2 MLPs :

- $A(x'')$ computes albedo (neutral colour) $\alpha = (r, g, b)$

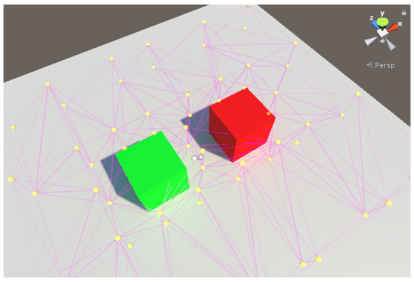
x'' = canonical space of x_s

- $\Gamma(x'')$ computes roughness (shininess) $\gamma \in \mathbb{R}$

hen BRDF maps α, γ to an output colour for vertex x_s

Light Probe

- A 3D Space containing a number of light paths from Sources & reflections
- To approximate lighting at a point x_s, take lighting from these nearby light rays



- Capture light directions & how 'wide' the light in that direction is

→ we can tell which light will hit parts of the avatar by using Distance Field Soft Shadow model.

Rendering

Lets fire camera rays in the required direction & take the light emitted from the first point of human intersection. The light emitted:

$$L_o = \sum_{\omega_i} L_s(\omega_i) \cdot R_s(x_s, \omega_i, \omega_o, n_s) \cdot V_s(x_s, \omega_i) \cdot n_s \cdot \omega_i \cdot \Delta\omega_i$$

= Sum over all incoming (→ human) light sources, the corresponding emitted light

L_o = output colour of human surface point x_s

L_s = light source direction at x_s

V_s = indicates if light from direction ω_i reaches x_s (or if blocked)

$\Delta\omega_i$ = width of incoming light beam

Loss

- The difference between the estimated avatar in a pose in certain lighting to ground truth (from the training video)
- Some additional regularisation