EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

Gaussian Head Visualizer

Practical Machine Learning

Daniel Eskandar

César Díaz Blanco

Supervised by Berna Kabadayi and Prof. Gerard Pons-Moll

Base Gaussian Splatting Visualizer

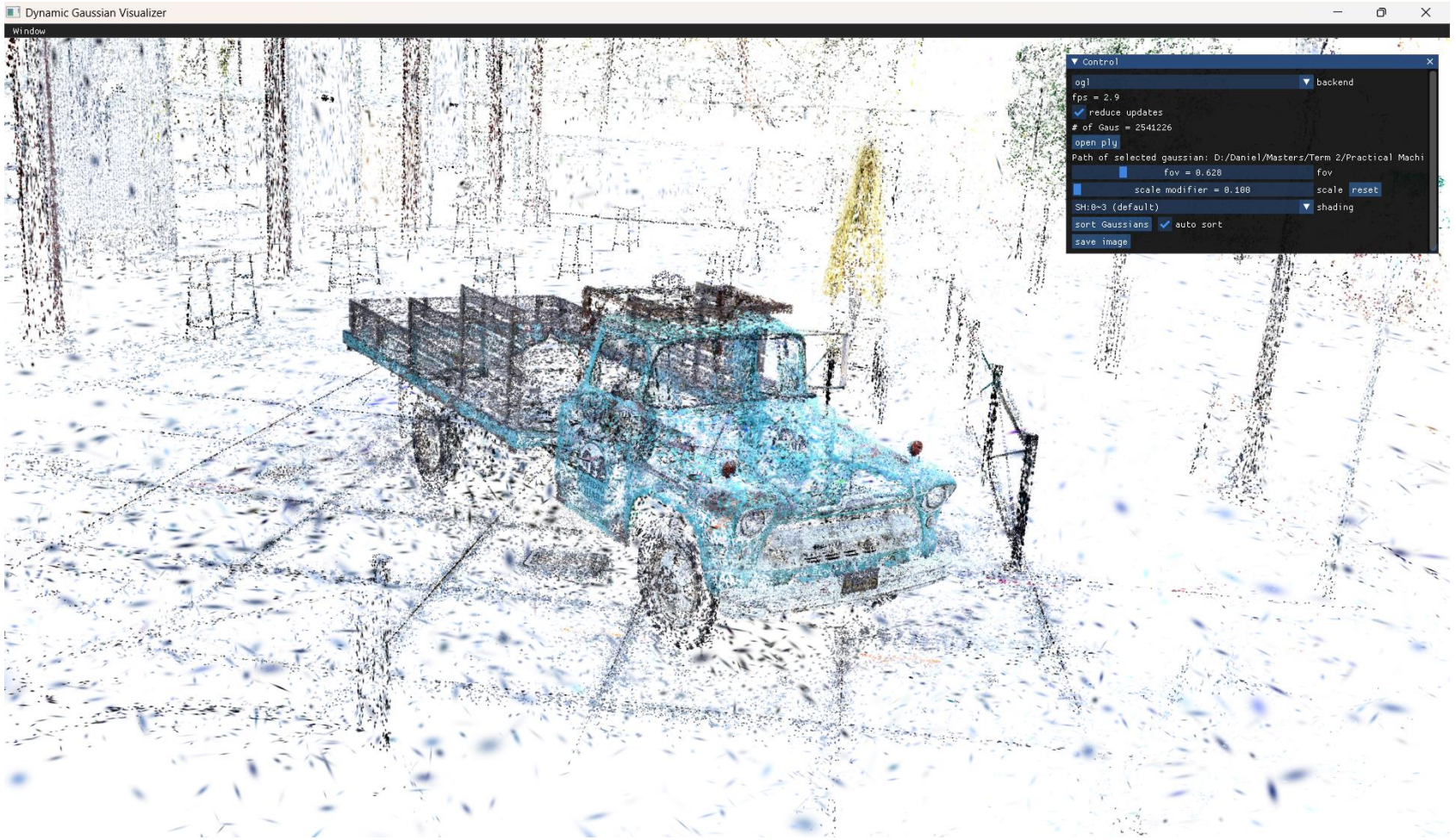
Scene Rendering

Base Gaussian Splatting Visualizer

Gaussian Ball shader

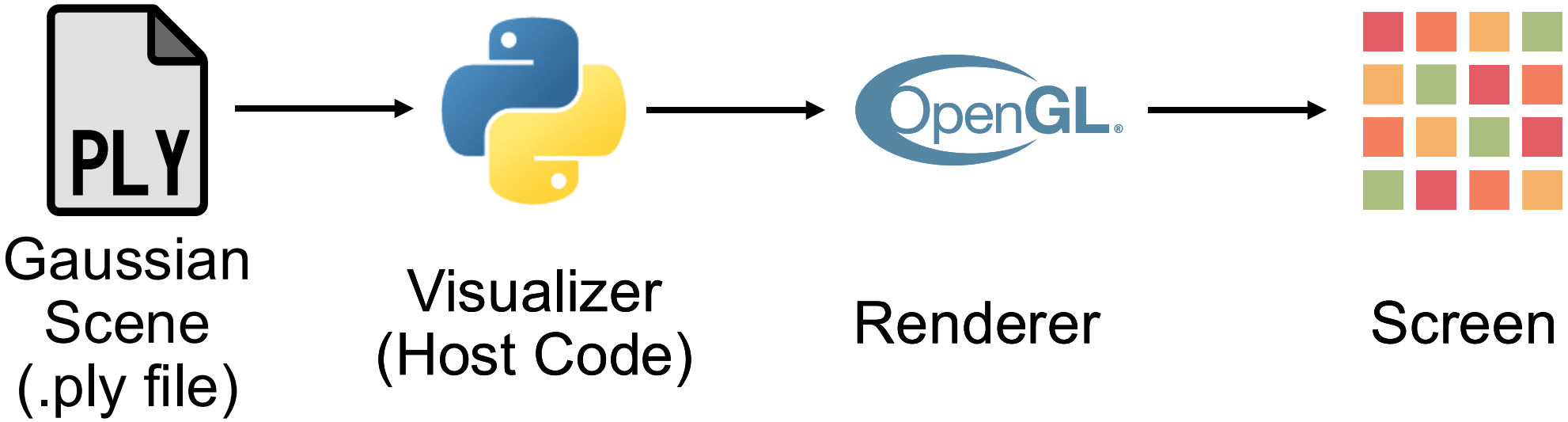
Base Gaussian Splatting Visualizer

Spherical harmonics shader with gaussian scales reduced



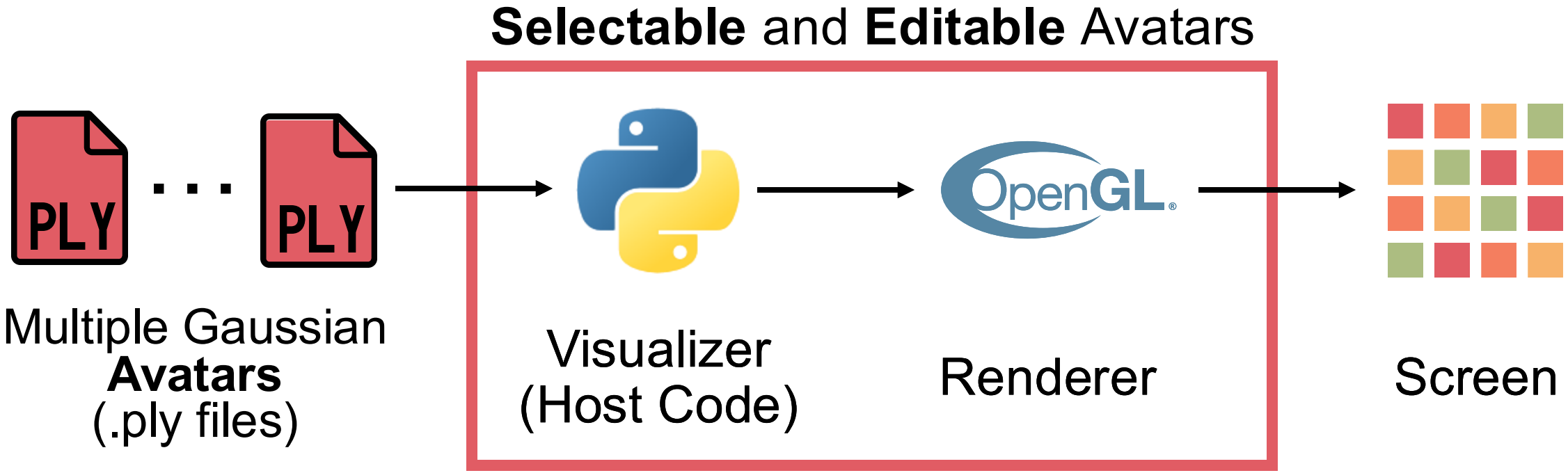
Base Gaussian Splatting Visualizer

Rendering Pipeline



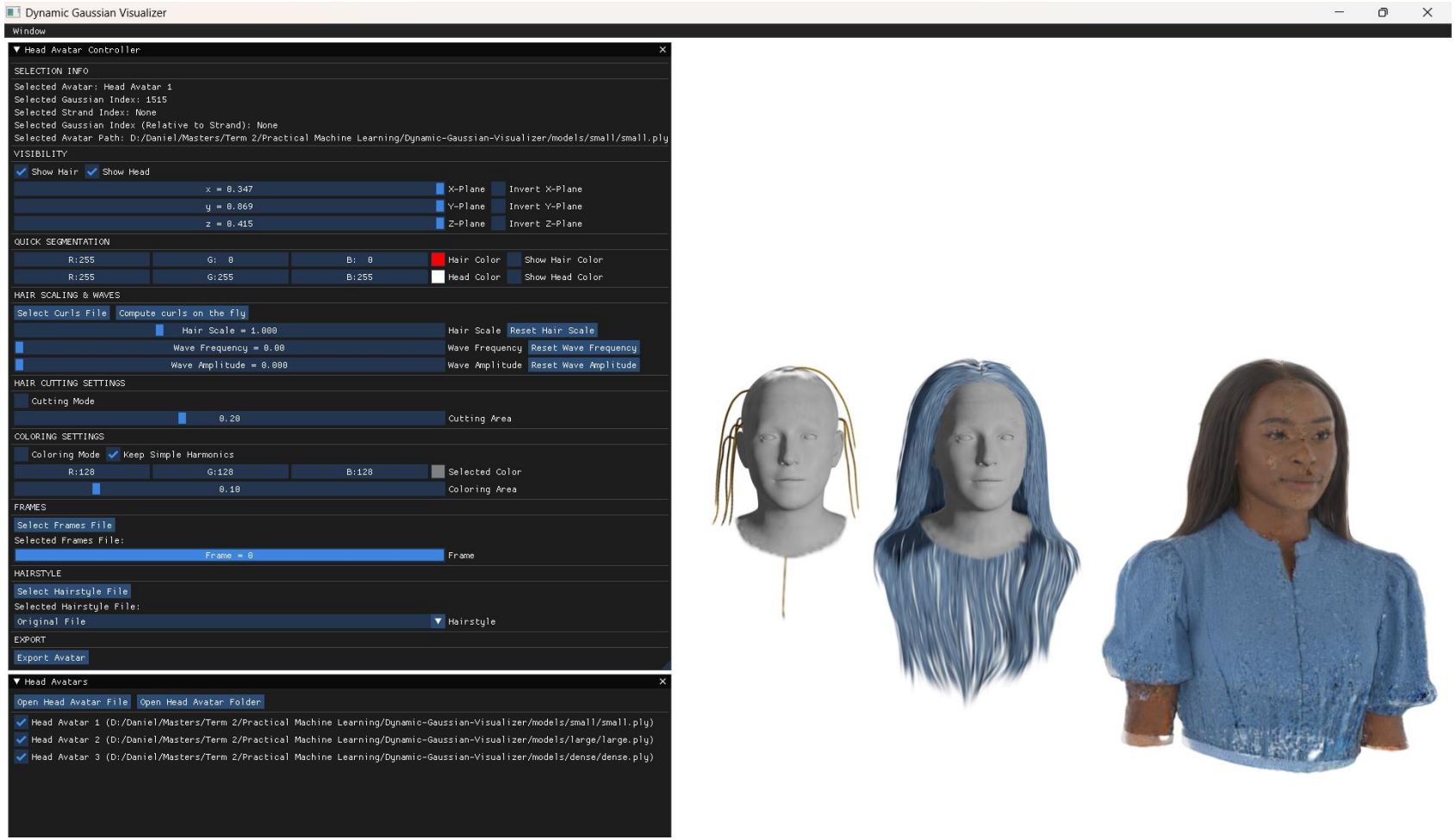
Our Interactive Gaussian Splatting Visualizer

Extended rendering pipeline for Gaussian Head Avatars



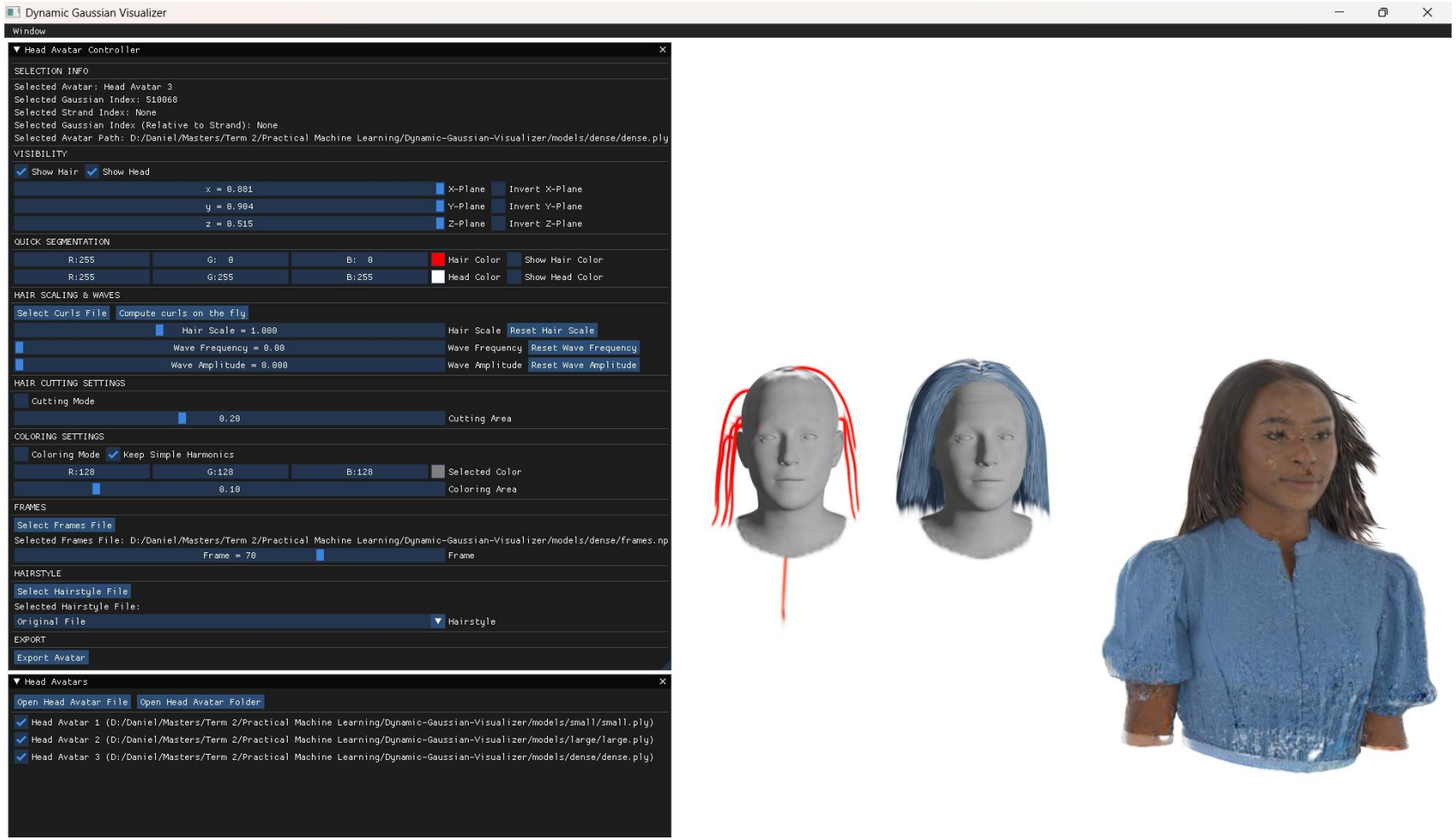
Our Interactive Gaussian Splatting Visualizer

Comparison of multiple avatars



Our Interactive Gaussian Splatting Visualizer

Editing of multiple avatars at once



Features Overview

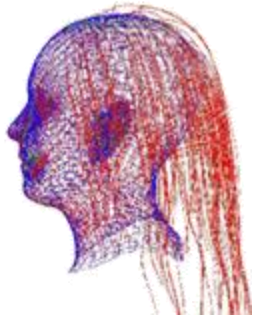
1. Multi-Avatar Display

2. Index Display

3. Visibility & Basic Coloring

4. Advanced Coloring

5. Hair Cutting



6. Axes View

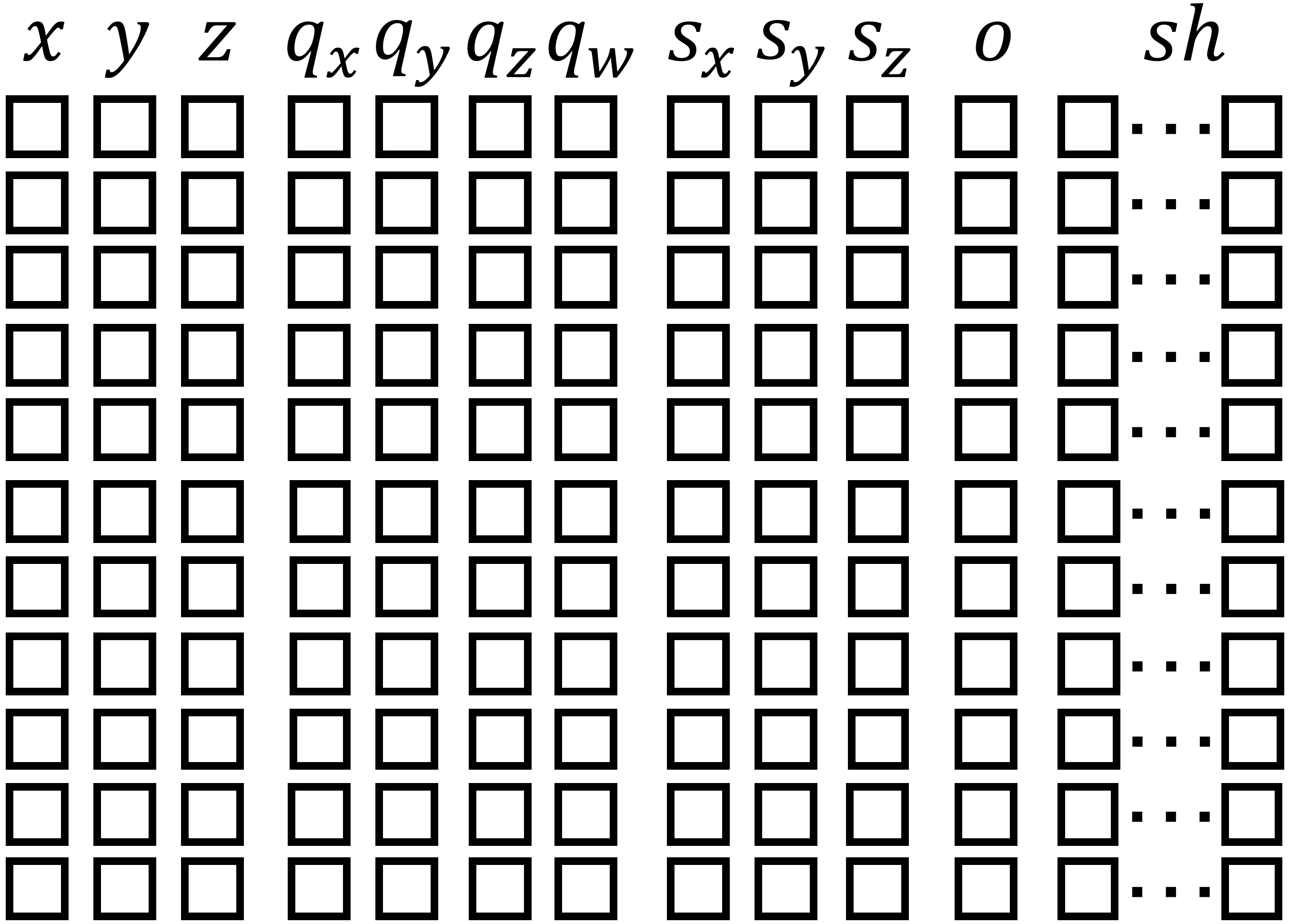
7. Curly Hair

8. Dynamic Frames

9. FLAME Integration

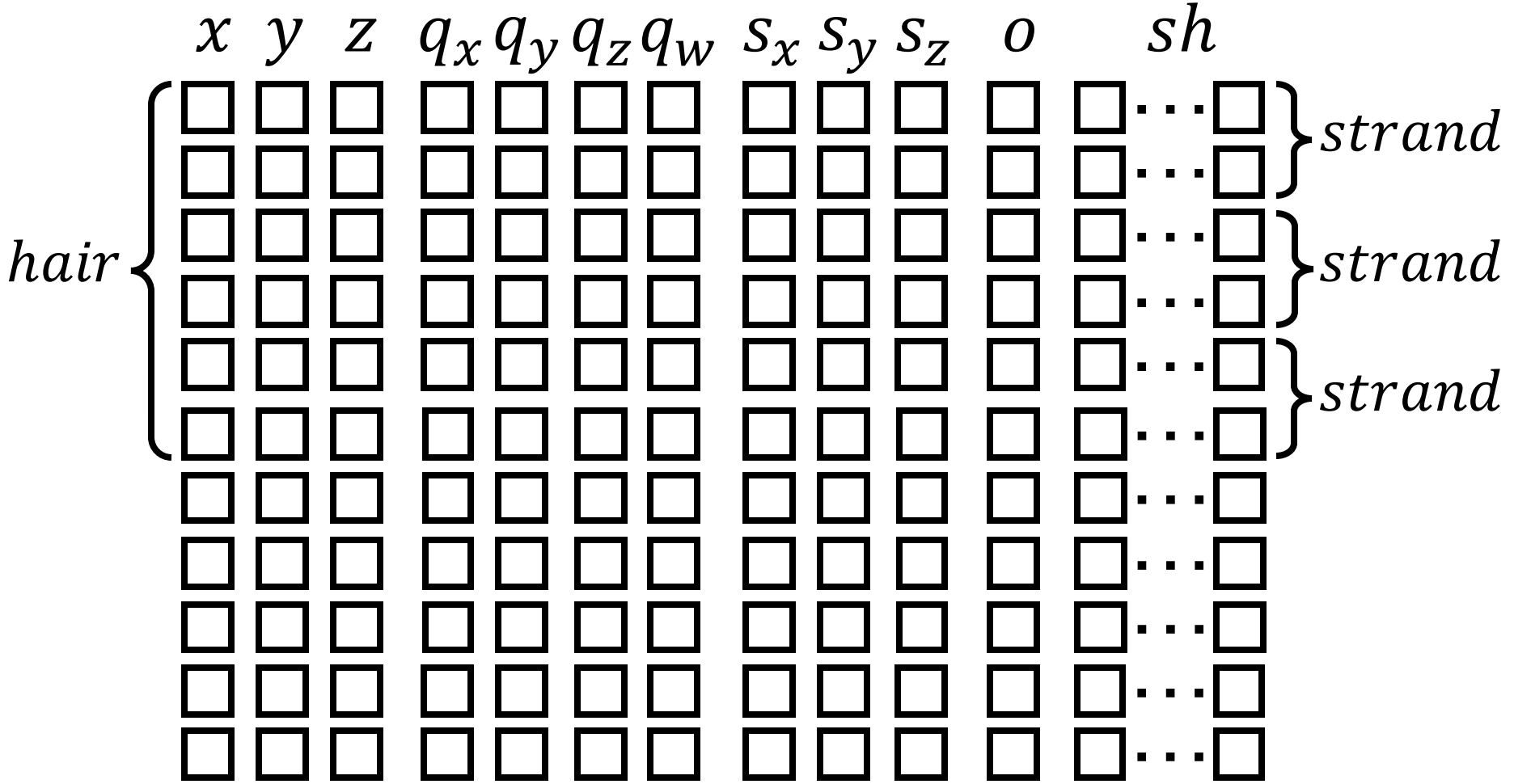
Data

.ply Gaussian Splatting Head Avatar Files



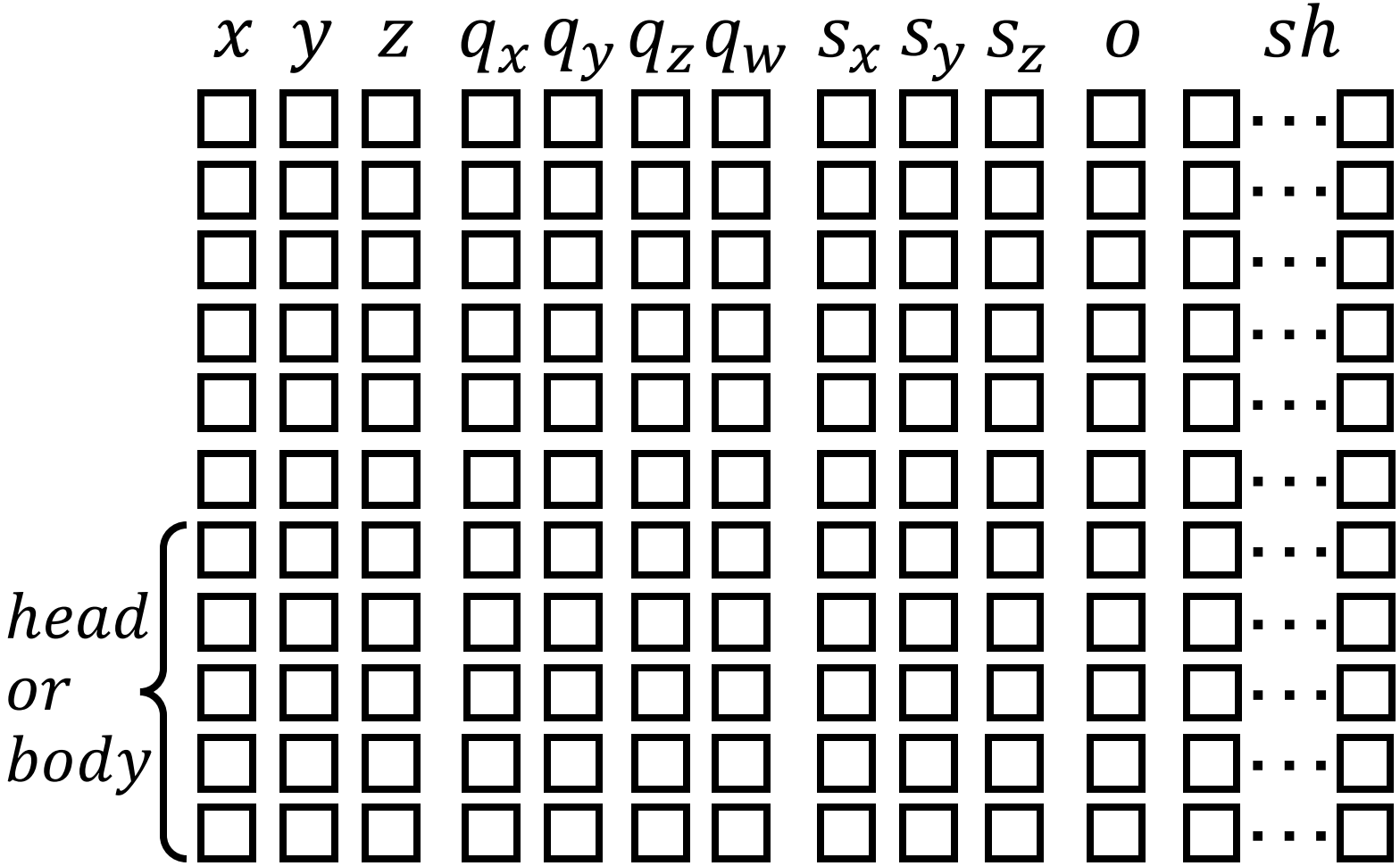
Data

.ply Gaussian Splatting Head Avatar Files



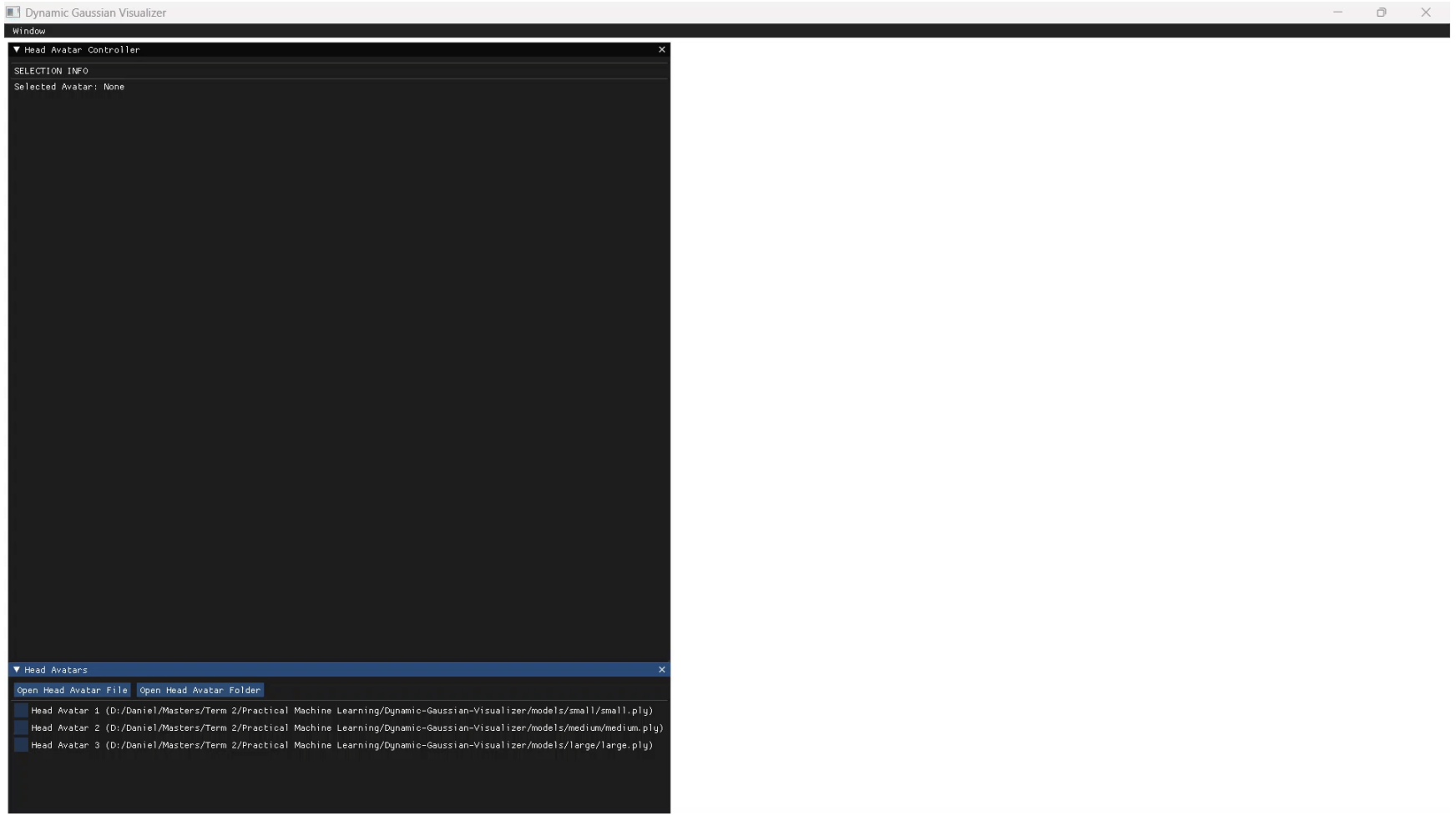
Data

.ply Gaussian Splatting Head Avatar Files



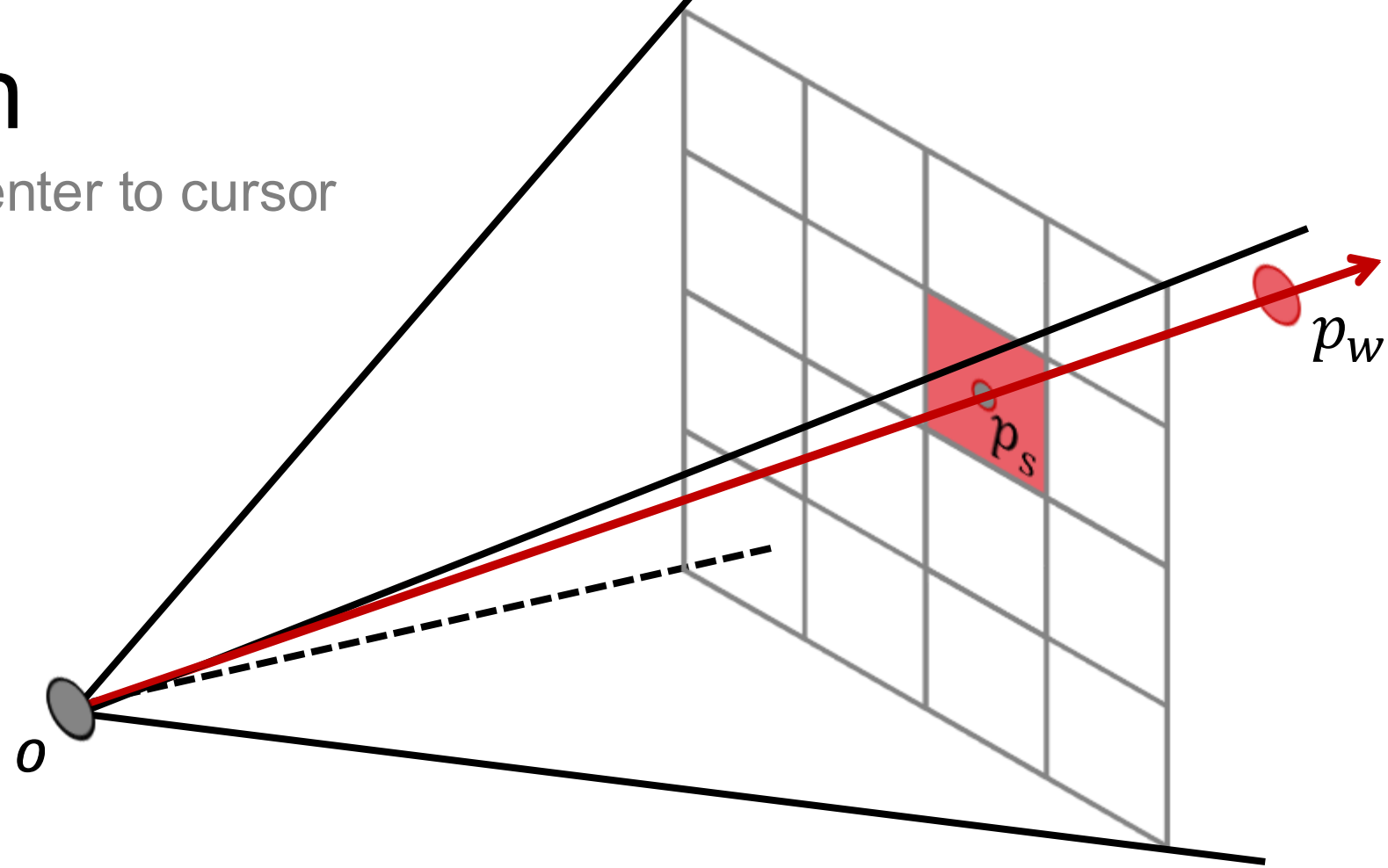
Multi-Avatar Display

Checkbox for displaying head avatars



Avatar Selection

Compute ray from camera center to cursor



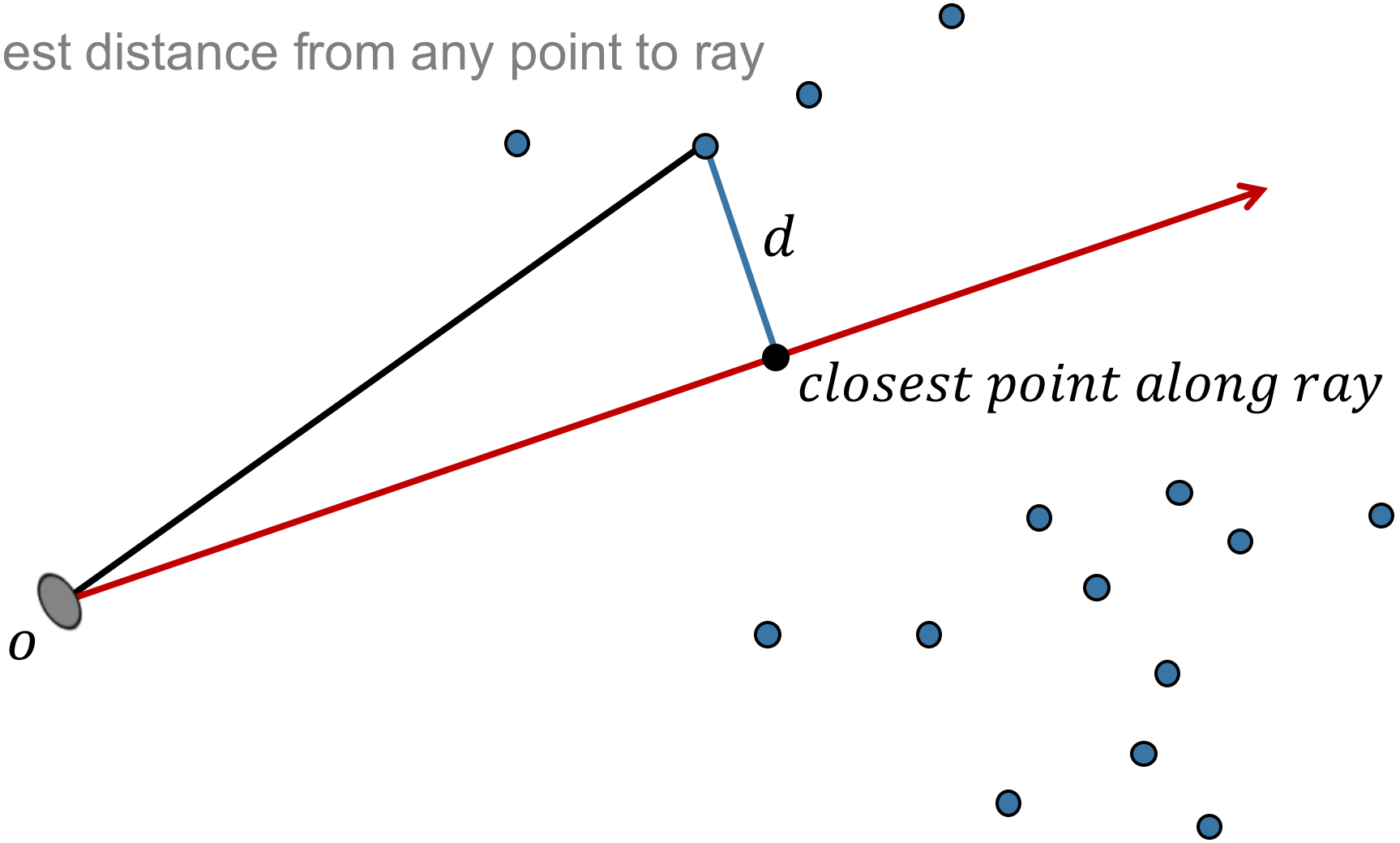
O: Camera origin (camera position)

p_s: Screen point (2D)

p_w: World position (3D) obtained by unprojecting p_s.

Avatar Selection

Compute closest distance from any point to ray



Gaussian Index Display

Selecting singular Gaussians with cursor

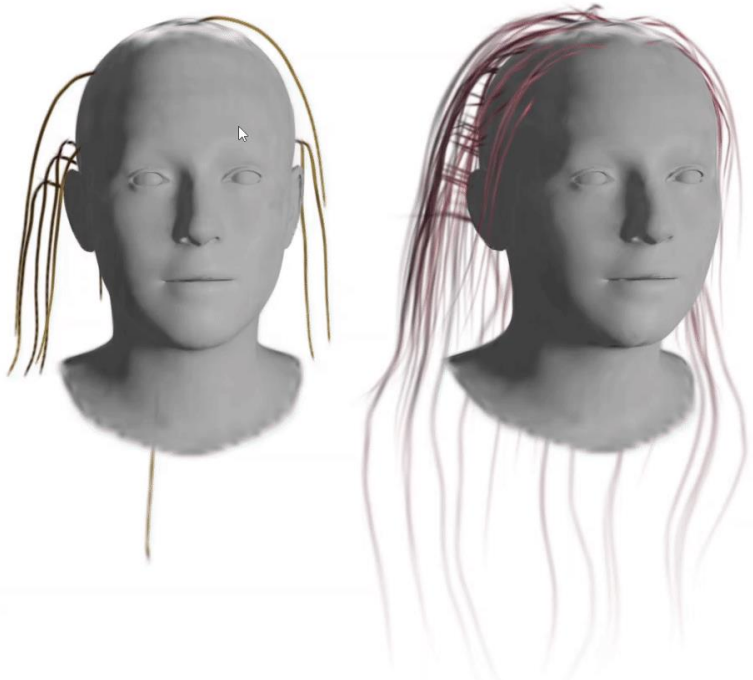
Dynamic Gaussian Visualizer

Window

Head Avatar Controller

SELECTION INFO

Selected Avatar: Head Avatar 1
Selected Gaussian Index: 4826
Selected Strand Index: None
Selected Gaussian Index (Relative to Strand): None



Visibility and Coloring of hair or face

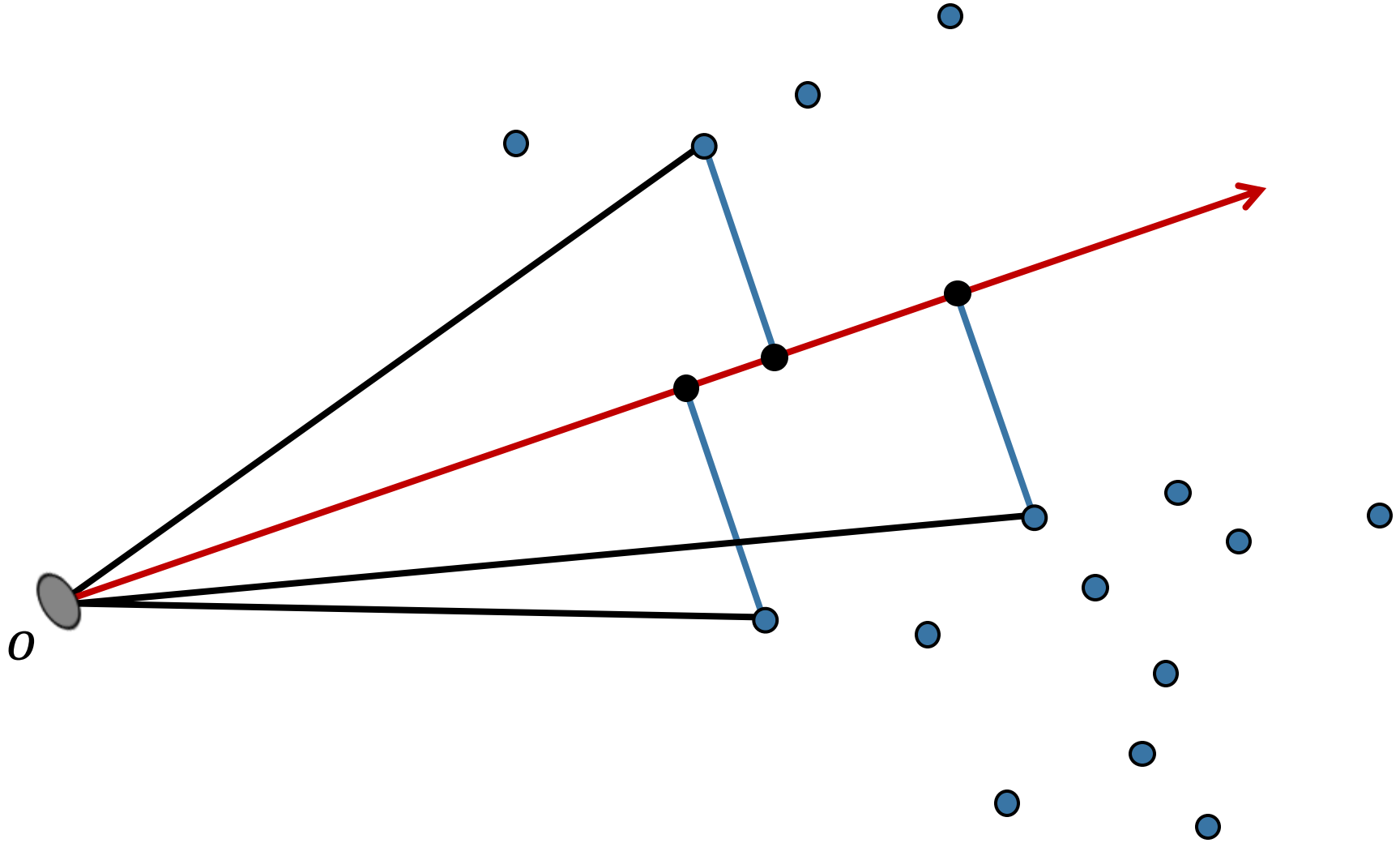
Enabling, disabling, and coloring all hair or head gaussians.

Visibility

Basic Coloring

Region Coloring

Compute closest distance from multiple Gaussians to the casted ray



Region Coloring

Color Gaussians at some threshold distance from the casted ray

Region Coloring

However, front and back are colored since depth along ray is not specified

Slicing

Fast to compute and sufficient for coloring only one face

X-Axis Slice

Y-Axis Slice

Z-Axis Slice

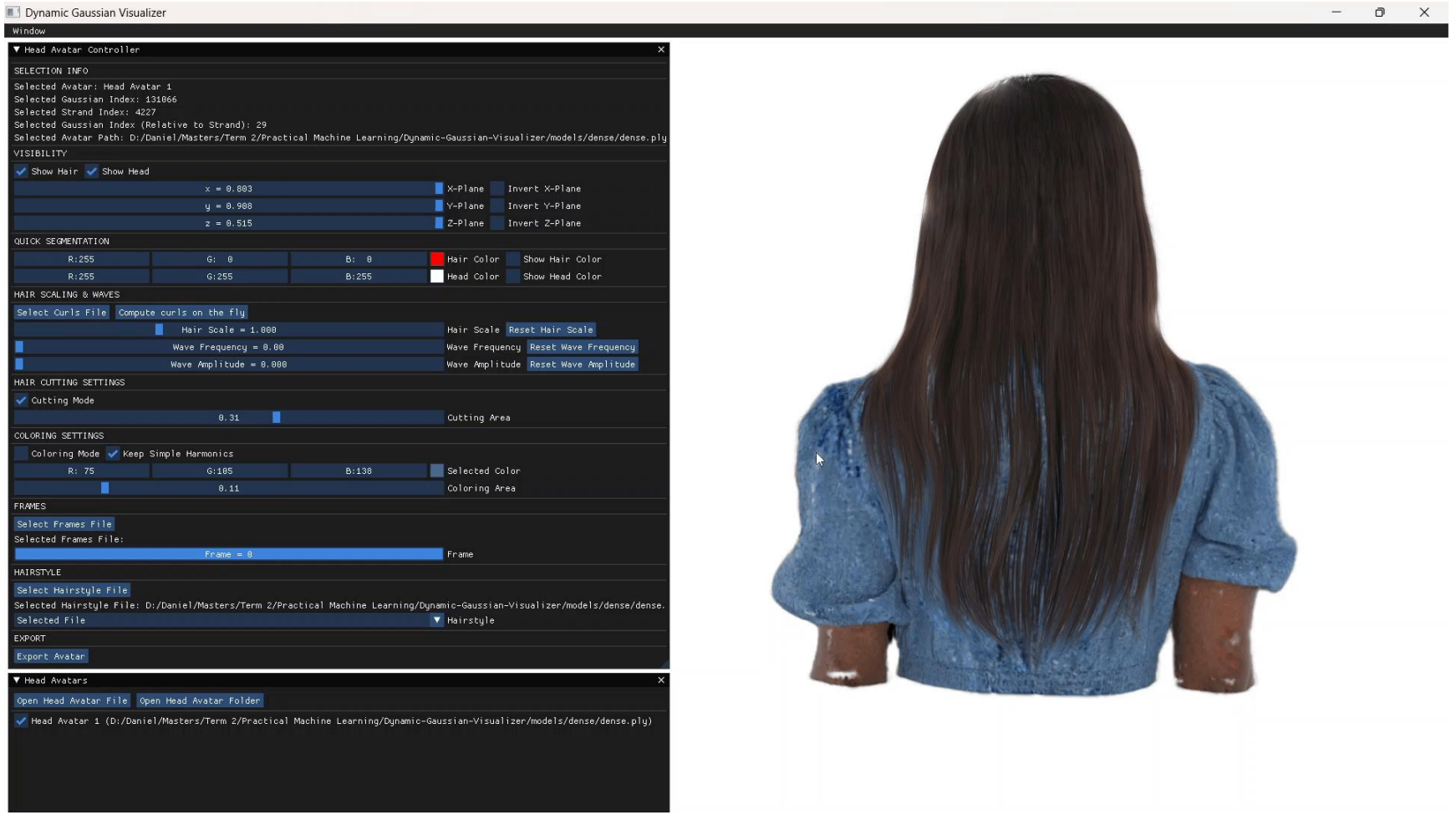
Region Coloring

Fix artifacts or segment by assigning solid colors

Segmented Corrected Original

Hair Cutting

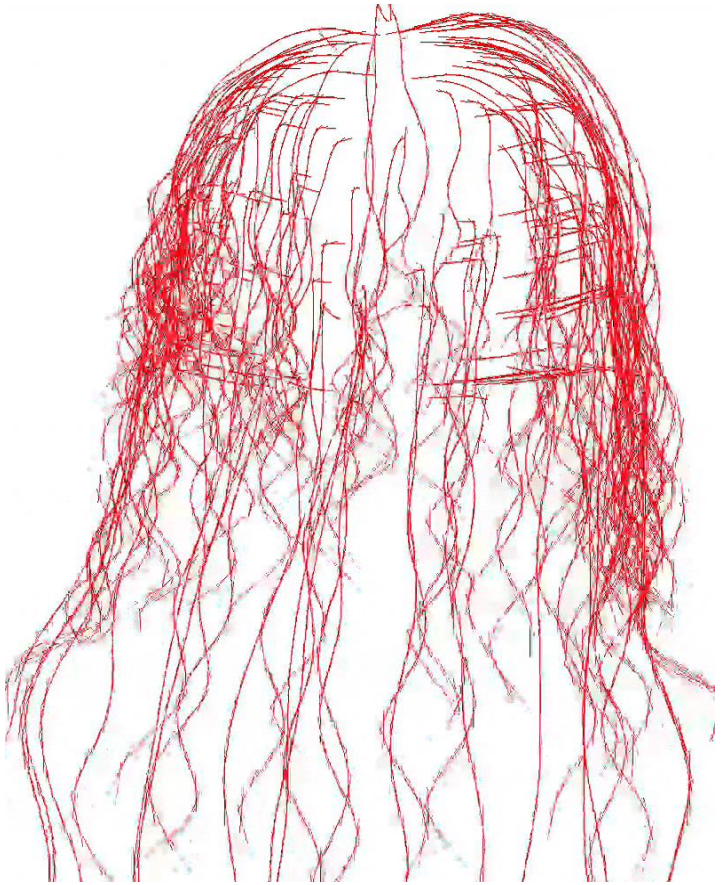
Cut the selected Gaussians (zero Opacity) and its downstream Gaussians



Axes View Shader

The gaussians' axes shader is helpful for visualizing hair gaussians

Spherical harmonics shader



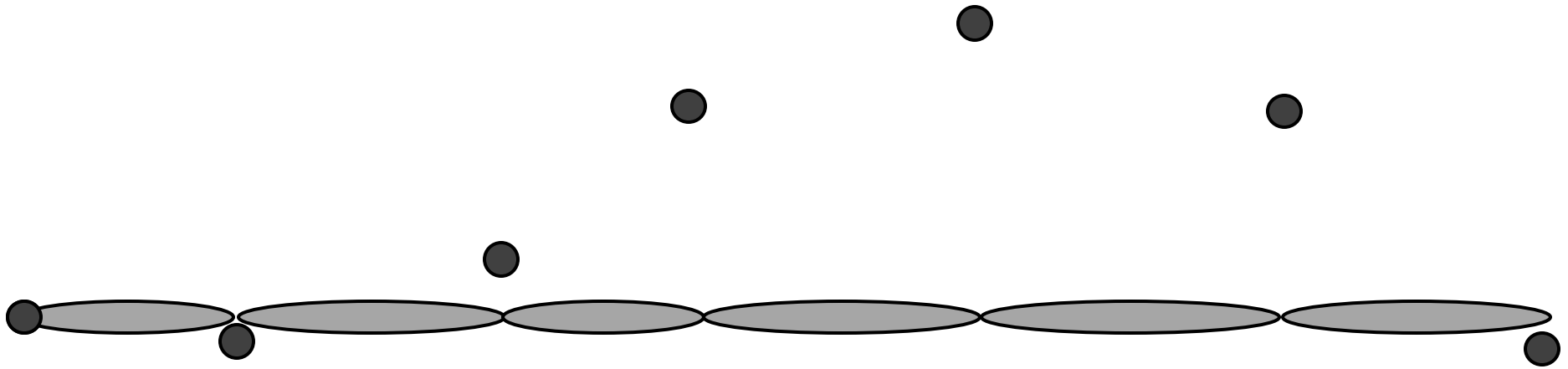
Gaussians' axes shader

Curly Hair

Hair gaussians to hair points

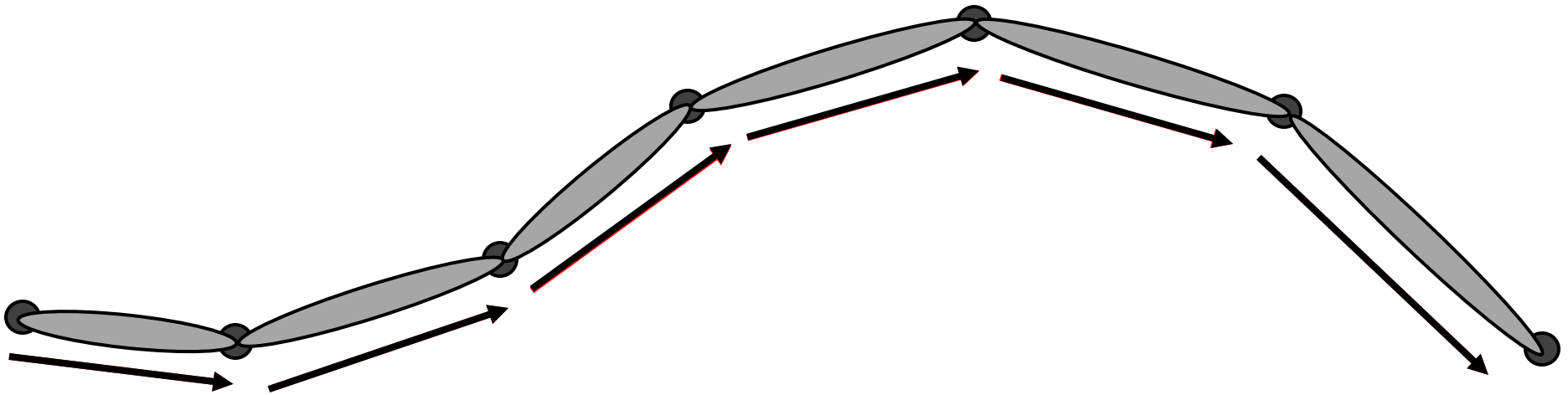
Curly Hair

Hair points are pushed in the gaussian’s y and z directions



Curly Hair

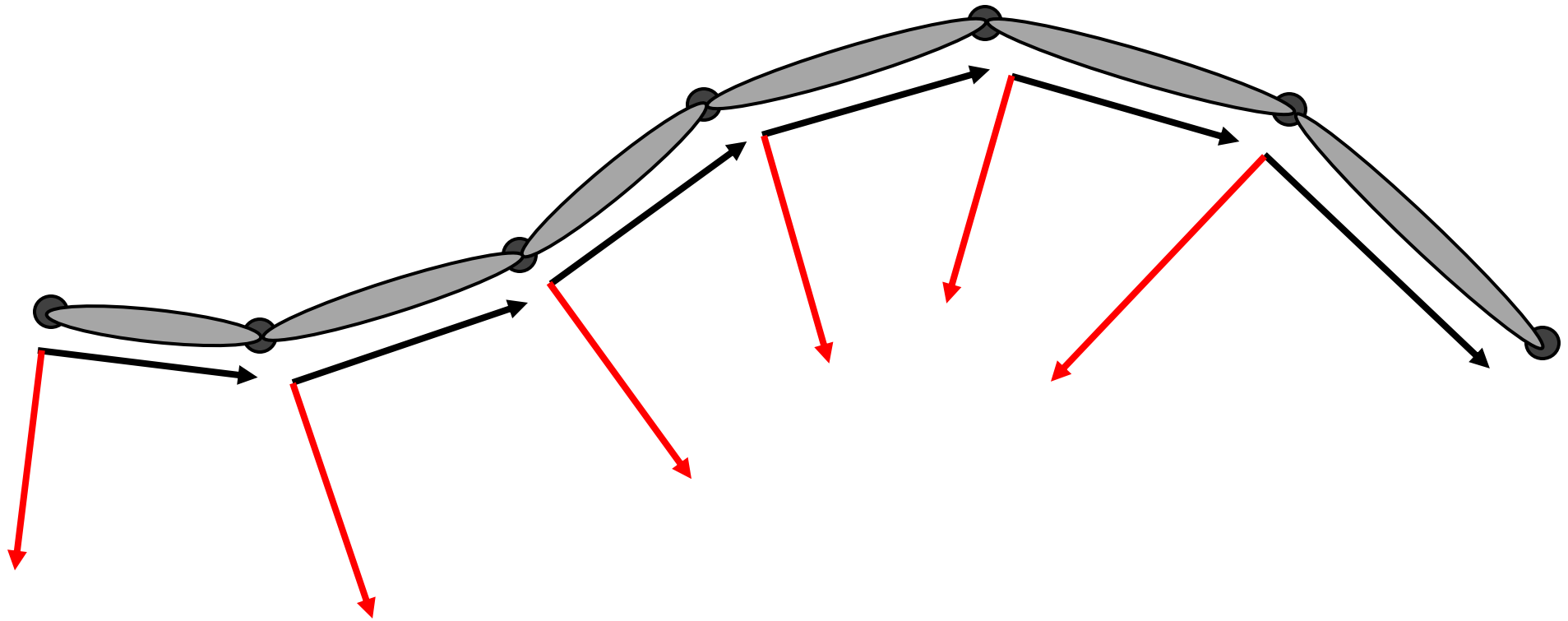
Tangent, normal, and binormal vectors computation



$$\boldsymbol{T_i = h_{i+1} - h_i}$$

Curly Hair

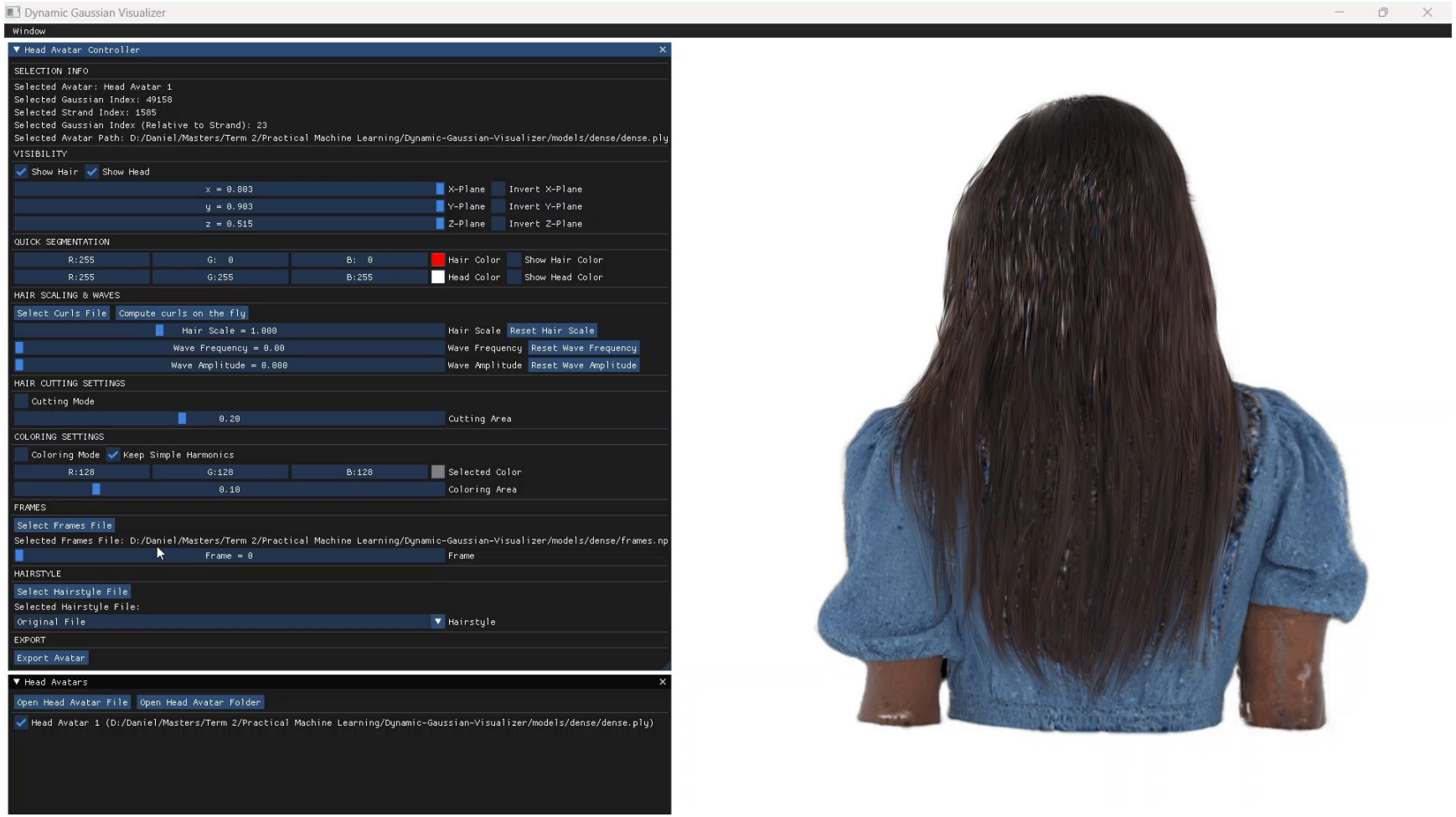
Tangent, normal, and binormal vectors computation



$$\boldsymbol{T_i = h_{i+1} - h_i} \qquad \boldsymbol{N_i} = (-T_{i,y}, T_{i,x}, 0) \qquad \boldsymbol{B_i = T_i \times N_i}$$

Dynamic Frames

Dense model with 12k strands and 647k total gaussians

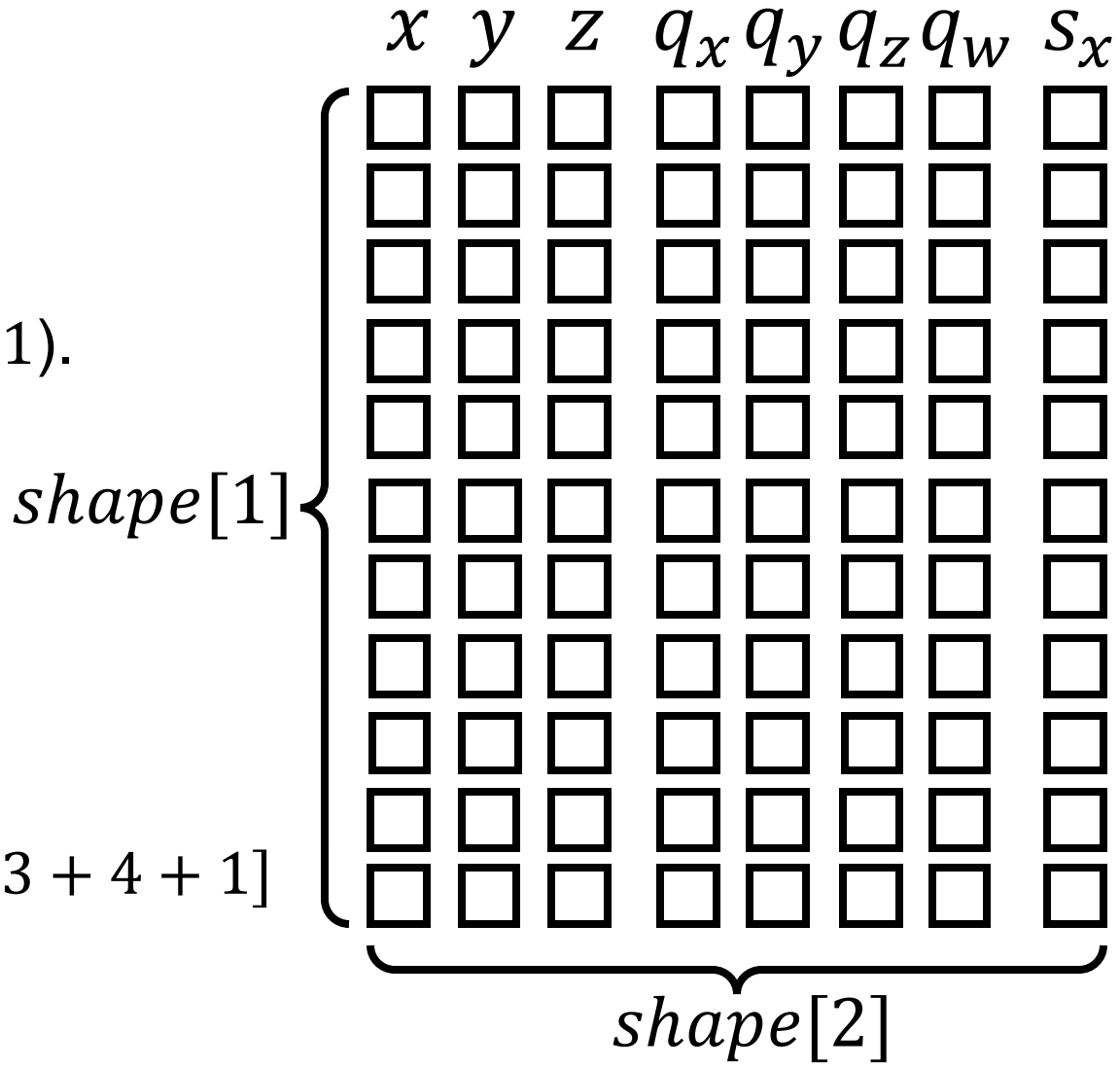


Dynamic Frames

Optimization for rendering the frames

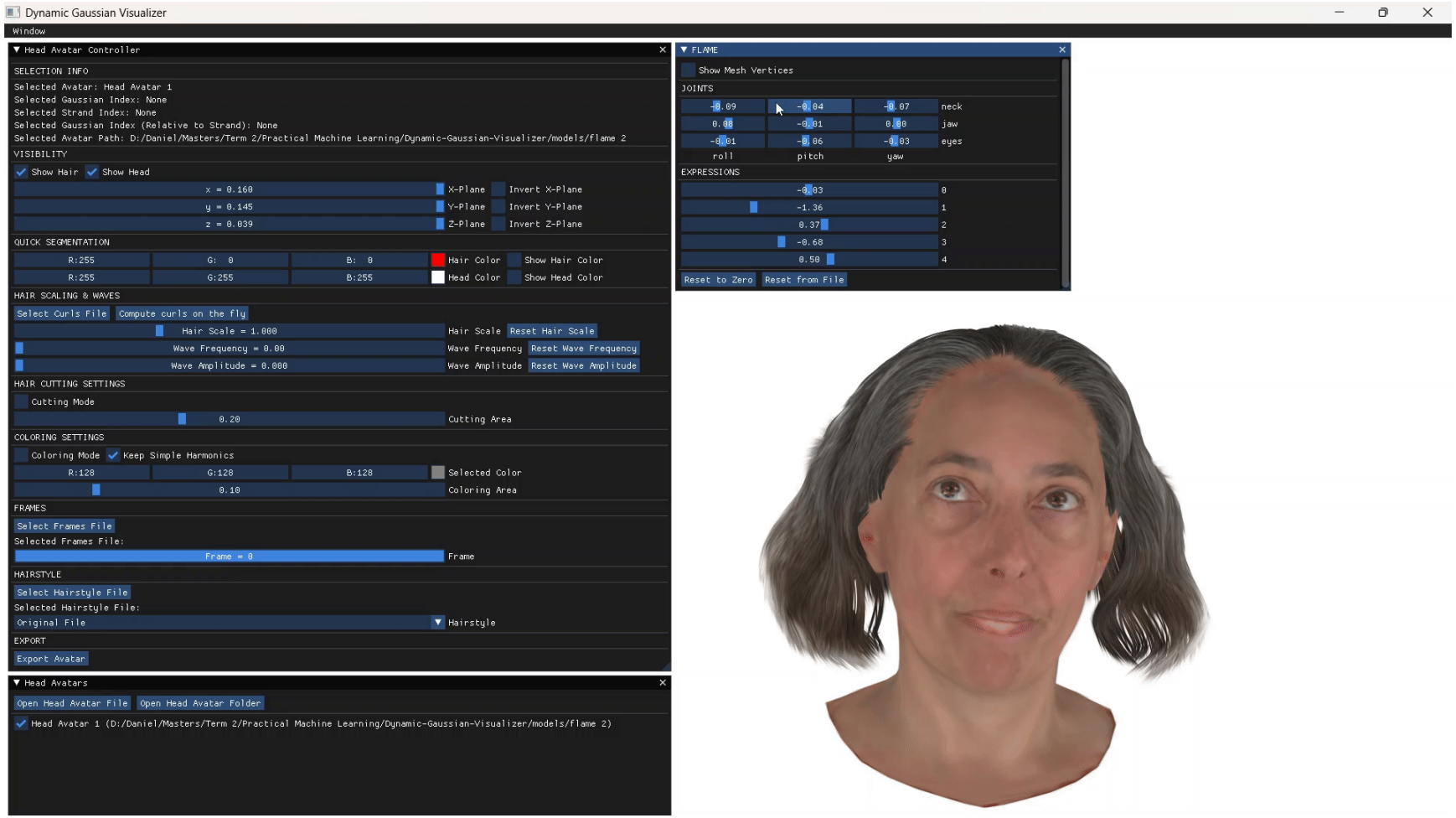
- Meant for hair gaussians
- Save only s_x (s_y and s_z fixed at 0.0001).
- Load data as single tensor.

$$shape = [\#\ of\ frames, \#\ of\ gaussians, 3 + 4 + 1]$$



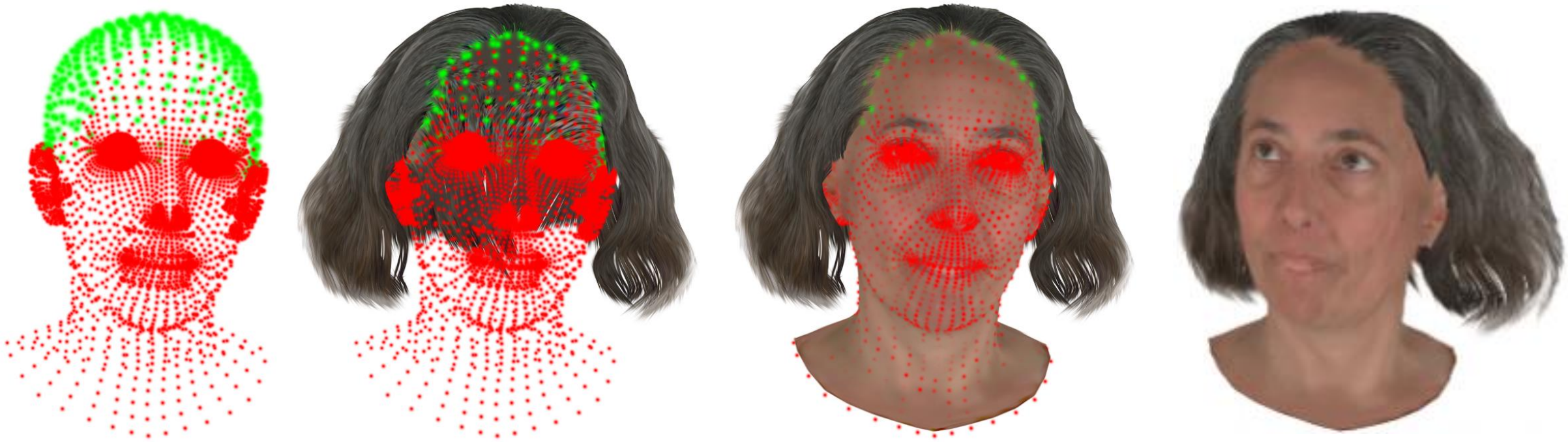
FLAME Integration

- FLAME: parametric model for head pose and expression.
- Gaussians bound to FLAME mesh.
- Parameters drive avatar animation.



FLAME Integration

Hair not represented in FLAME and cannot be bound to the mesh.



Solution:

- Root-to-nearest vertex anchoring.
- Strand follows vertex transform.

Summary

An extended Gaussian Splatting visualizer for Gaussian Head Avatars

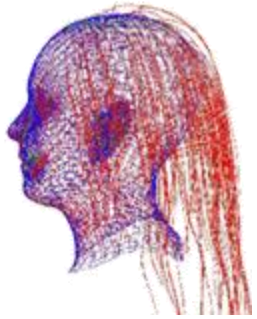
1. Multi-Avatar Display

2. Index Display

3. Visibility & Basic Coloring

4. Advanced Coloring

5. Hair Cutting



6. Axes View

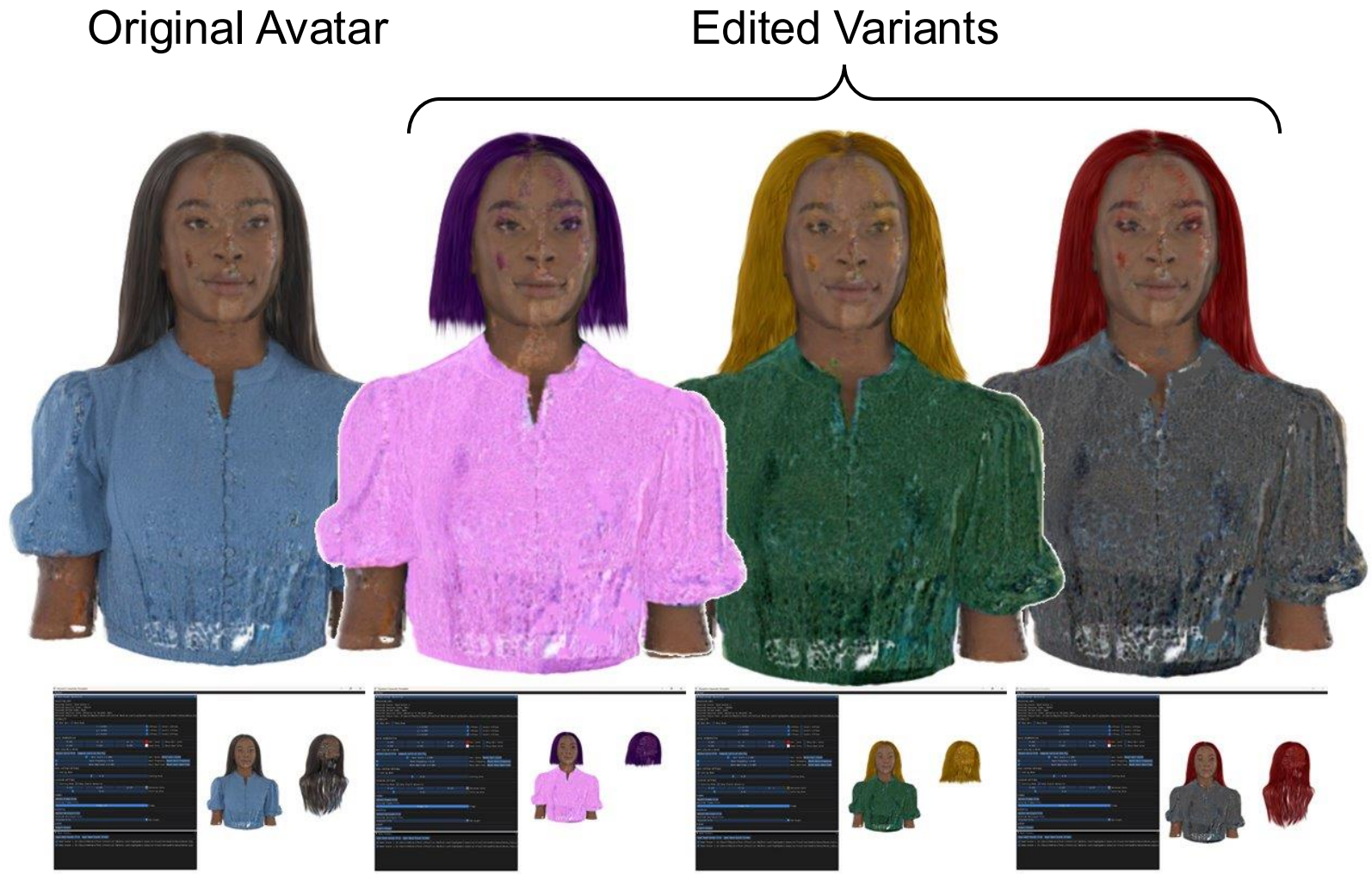
7. Curly Hair

8. Dynamic Frames

9. FLAME Integration

Summary

An extended Gaussian Splatting visualizer for Gaussian Head Avatars

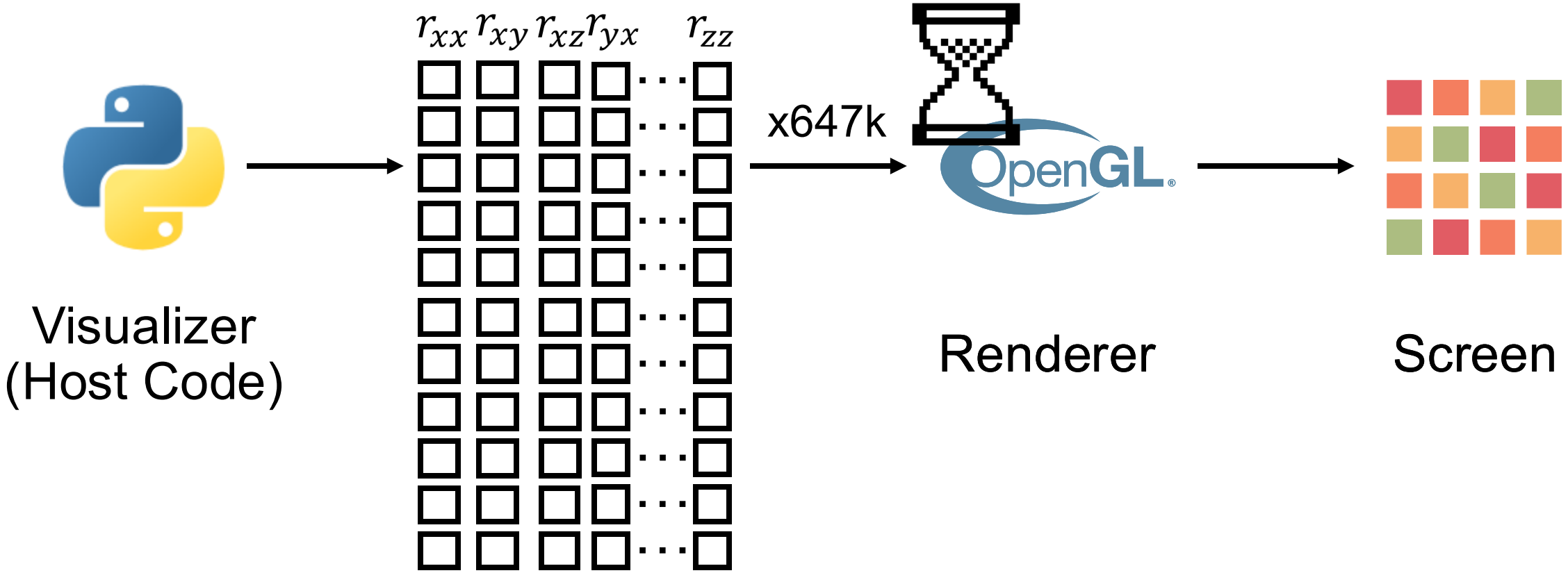


Acknowledgements

- The foundation for this work is based on the original repository GaussianSplattingViewer.
- Code for binding Gaussian files with the FLAME model was sourced from the GaussianAvatars repository.
- Huge thanks to Berna Kabadayi for her guidance!

Sending rotations to the renderer

Rotations as quaternions or matrices



Curly Hair

From TNB Frames to Quaternions

1. The quaternion which represents the rotation from v_1 to v_2:

$$q_{v_1 \to v_2} = \left(||v_1|| ||v_2|| + v_1 . v_2, v_1 \times v_2 \right)$$

2. To rotate the 1st canonical vector to the tangent vector we do:

$$q_{e_1 \to T} = \left(1 + T_x, 0, -T_z, T_y\right) = (1 + T_x, N)$$

3. The formula to go from a quaternion rotation to a matrix is:

$$\begin{pmatrix} 1 - 2(q_j^2 + q_k^2) & 2(q_i q_j - q_k q_r) & 2(q_i q_k + q_j q_r) \\ 2(q_i q_j + q_k q_r) & 1 - 2(q_i^2 + q_k^2) & 2(q_j q_k - q_i q_r) \\ 2(q_i q_k - q_j q_r) & 2(q_j q_k + q_i q_r) & 1 - 2(q_i^2 + q_j^2) \end{pmatrix}$$

4. We need to rotate the 2nd canonical vector by $q_{e_1 \to T}$. Once done, we compute the quaternion to go from the rotated 2nd canonical vector to the normals.
5. Lastly, we have the three quaternions for each of the rotations which together give us the final quaternion.