Real-Time Hair Rendering with Hair Meshes

Supplemental Document

Gaurav Bhokare
University of Utah
USA

Eisen Montalvo
University of Utah
USA

Elie Diaz
University of Utah
USA

Cem Yuksel
Cyber Radiance
USA

PROCEDURAL HAIR STYLING

To introduce small-scale variations between hair strands we apply a set of procedural styling operations using the styling coordinates. Here, we describe the styling operations we use in our implementation. Though our styling operations can generate various hairstyles, our approach is not limited to these particular operations. Note that these styling operations are not applied to the root vertex.

We use 5 procedural styling operations: *fuzz*, *fizz*, *kink*, *curl*, and *clumping*, as explained below. In our formulations below we assume a normalized w coordinate, such that $w = 0$ at the root and $w = 1$ at the hair tip. Most of our styling operations use a noise function to determine the direction and magnitude of the perturbation. We use the gradient of Perlin noise [Perlin 2002] in our implementation.

Fuzz adds a random perturbation to each hair strand. The same perturbation direction is used for all vertices of a hair strand, determined by the hair strand index or its uv coordinate at the root. Its amplitude a_{fuzz}, defined in object space, increases from root to tip, the rate of which is controlled by a user-defined exponent parameter b_{fuzz}. The resulting fuzz perturbation of a vertex can be written as

$$\mathbf{d}^{\tau}_{\text{fuzz}} = a_{\text{fuzz}}\ (w)^{b_{\text{fuzz}}}\ \begin{bmatrix}\cos\phi & \sin\phi & 0\end{bmatrix}^T\ (\xi_a)^{c_{\text{fuzz}}}\ , \tag{1}$$

where $\phi = 2\pi\xi_\phi$ is a random angle, $\xi_a, \xi_\phi \in [0, 1]$ are random numbers associated with the hair strand, and c_{fuzz} is another exponent parameter that controls the magnitude distribution.

Frizz computes a perturbation direction at the root uv coordinate of each hair strand using the noise function and applies the same perturbation to the entire hair strand. It also uses amplitude a_{frizz} and exponent b_{frizz} parameters, along with noise frequency f_{frizz}, resulting

$$\mathbf{d}^{\tau}_{\text{frizz}} = a_{\text{frizz}}\ (w)^{b_{\text{frizz}}}\ \boldsymbol{\Psi}\left(\begin{bmatrix}u\, f_{\text{frizz}} & v\, f_{\text{frizz}} & 0\end{bmatrix}^T\right)\ , \tag{2}$$

where $\boldsymbol{\Psi}$ is the noise function that returns a direction for the given 3D position in texture space. Notice that frizz samples the noise direction at the root, where $w = 0$.

Kink works similar to frizz, but computes a different perturbation for each hair vertex. Kink uses amplitude a_{kink} and exponent b_{kink} parameters as well, but it has two frequency parameters: f^{uv}_{kink} for the u and v coordinates and f^{w}_{kink} for the w coordinate, such that

$$\mathbf{d}^{\tau}_{\text{kink}} = a_{\text{kink}}\ (w)^{b_{\text{kink}}}\ \boldsymbol{\Psi}\left(\begin{bmatrix}u\, f^{uv}_{\text{kink}} & v\, f^{uv}_{\text{kink}} & w\, f^{w}_{\text{kink}}\end{bmatrix}^T\right)\ . \tag{3}$$

SIGGRAPH Conference Papers '24, July 27-August 1, 2024, Denver, CO, USA
2024. ACM ISBN 979-8-4007-0525-0/24/07…$15.00
https://doi.org/10.1145/3641519.3657521

Table 1: *Parameters of the procedural styling operations.*

Fuzz	a_{fuzz}	amplitude
	b_{fuzz}	amplitude exponent
	c_{fuzz}	distribution exponent
Frizz	a_{frizz}	amplitude
	b_{frizz}	amplitude exponent
	f_{frizz}	noise frequency
Kink	a_{kink}	amplitude
	b_{kink}	amplitude exponent
	f^{uv}_{kink}	noise frequency for u and v
	f^{w}_{kink}	noise frequency for w
Curl	a_{curl}	amplitude
	b_{curl}	amplitude exponent
	c_{curl}	distribution exponent
	f_{curl}	noise frequency
	r_{curl}	number of spiral rotations
Clumping	a_{clump}	thickness
	b_{clump}	clumping exponent
	n_{clump}	noise amount
	f_{clump}	noise frequency
	m_{clump}	clump center grid resolution
	p_{clump}	percentage
	v_{clump}	clump variation

Curl applies a spiral function to achieve a curly hairstyle. In addition to amplitude a_{curl} and exponent b_{curl} parameters, it receives the number of spiral rotations r_{curl}, the noise frequency f_{curl} for shifting the phase, and a second exponent parameter c_{curl} that shifts the spiral rotations towards the tip or the root, resulting

$$\mathbf{d}^{\tau}_{\text{curl}} = a_{\text{curl}}\ (w)^{b_{\text{curl}}}\ \begin{bmatrix}\cos\theta & \sin\theta & 0\end{bmatrix}^T \tag{4}$$

$$\theta = 2\pi\ r_{\text{curl}}\ \left(\Psi_w\left(\begin{bmatrix}u\, f_{\text{curl}} & v\, f_{\text{curl}} & 0\end{bmatrix}^T\right) + (w)^{c_{\text{curl}}}\right)\ , \tag{5}$$

where Ψ_w is a scalar noise function.

Clumping is defined on a regular grid of size $m_{\text{clump}} \times m_{\text{clump}}$ covering the uv texture space. The uv value at the center of a grid cell defines a *reference strand*. Vertices $\mathbf{p}$ of hair strands within the same grid cell (based on their uv coordinates) are moved towards the corresponding vertex of the reference strand $\mathbf{p}_c$ using

$$\mathbf{d}_{\text{clump}} = (\mathbf{p}_c - \mathbf{p})\left(1 - \frac{h\, a_{\text{clump}}}{\|\mathbf{p}_c - \mathbf{p}\|}\right)(w)^{b_{\text{clump}}}\ , \tag{6}$$

where $h \in [0, 1]$ is the texture-space distance of the hair strand from the clump center normalized by the texture-space clump size,

a_{clump} is the object-space clump thickness parameter, and b_{clump} is the exponent parameter. For breaking the uniformity of this grid structure, we find the clump center of a hair strand after adding a noise offset to its uv coordinate.

We compute the object-space position of the reference strand $\mathbf{p}_c$ by sampling the **hair mesh texture** at the corresponding position of the clump center $\mathbf{p}_c^\star$. For evaluating $\mathbf{p}_c^\star$, we precompute tip positions $\mathbf{p}_{c\text{-tip}}^\star$ and store them in a 2D texture. This process involves rendering the tip-layer faces of the hair mesh onto a texture of $m_{\text{clump}} \times m_{\text{clump}}$ resolution using the uv coordinates of the hair mesh tip vertices. To break the uniformity of the grid structure, we apply the same noise function above after finely tessellating each tip-layer face tessellation. The empty texels of this texture are filled by copying neighboring non-empty texel values.

We use another parameter $p_{\text{clump}} \in [0, 1]$ to determine the percentage of hair strands to be included in clumping. Whether a hair strand should be included in a clump is determined by a random number in $[0, 1]$ that can be generated based on the hair strand index (within a clump) or its original uv coordinate.

Furthermore, we use a variation parameter v_{clump} to overwrite the uvw coordinate of a hair strand by interpolating its original uvw coordinate and the uvw coordinate of the clump center by v_{clump}. Frizz and kink operations use this updated uvw coordinate while evaluating their noise functions. In our implementation, curl directly uses the uvw of the clump center, regardless of v_{clump}. Thus, when $v_{\text{clump}} = 0$, all hair strands within a clump get the same styling perturbations, except for fuzz.

REFERENCES

Ken Perlin. 2002. Improving Noise. *ACM Trans. Graph.* 21, 3 (jul 2002), 681–682. https://doi.org/10.1145/566654.566636