Example-Based Hair Geometry Synthesis: Supplemental Material

Lvdi Wang[1] Yizhou Yu[2] Kun Zhou[3] Baining Guo[1,4]
[1]Tsinghua University [2]University of Illinois at Urbana-Champaign [3]Zhejiang University [4]Microsoft Research Asia

A Correcting Reversed Strands

All existing image-based hair acquisition methods [Paris et al. 2004; Wei et al. 2005; Paris et al. 2008] first construct a 3D orientation field and then trace individual hair strands from this field. However, it is difficult for an algorithm to confidently distinguish the root and tip of a hair strand from an orientation field. As a result, a significant portion of the reconstructed hair strands have reversed directions. Note that this problem does not affect static rendering, but may cause problems in animation and our example-based analysis and synthesis.

Given a hair strand γ with n_s vertices, we first define its *average tangent vector* $\mathbf{v}_g(\gamma)$ as

$$\mathbf{v}_g(\gamma) = \frac{\sum_{i=2}^{n_s}(\gamma(i) - \gamma(1))}{\| \sum_{i=2}^{n_s}(\gamma(i) - \gamma(1))\|}. \quad (1)$$

We then define the *scalp direction* of γ, $\mathbf{v}_s(\gamma)$ as the 2D projection of $\mathbf{v}_g(\gamma)$ onto the tangent plane of the scalp surface at the root of γ. The scalp direction of all hair strands forms a 2D vector field on the scalp surface.

We provide a stroke-based interface to easily correct reversed hair directions. The user draws a few strokes (typically 1 to 10) indicating a sparse vector field. From these strokes, we compute a full-resolution guidance field using scattered data interpolation. For each hair cluster in the finest level of the hierarchy, we compare the scalp direction of each of its strands with the corresponding direction in the guidance field. If the dot product between the two is negative, we mark this hair strand as *reversed*. Finally, if the cluster has more reversed strands than correct ones, we invert the direction of the entire cluster.

To correct a reversed hair strand geometrically, we cannot simply invert its vertex order because 1) its original tip may not be located exactly on the scalp surface, and 2) its original root in most cases should not still stay on the scalp surface after the inversion. To meet the above constraints yet preserve the original hair geometry as much as possible, we apply the following simple method. For an original hair tip above the scalp surface, we incrementally extend the strand while gradually bending it toward the scalp until it hits the scalp; for an original hair root, we simply remove the portion that is close to the scalp surface.

B The Combing Tool

Our combing tool is designed to help the user to modify large scale hair shapes in an efficient and intuitive way. The user applies combing by drawing strokes on the screen. Each stroke defines a local 2D vector field in the screen space. The hair strands are first projected to the screen space, deformed accordingly by the local vector field, and then projected back using their original depth information.

Typically, the user only needs to comb the coarsest level of the hierarchy where hair strands often have very little local detail. Hair strands at finer levels are then computed by transfering the original displacements to the combed center strands. Sometimes, when the user has to directly edit finer levels of the hierarchy, or when the coarsest level still has important details to be preserved, we further apply the technique described in Section 3.1 of [Xu et al. 2007]

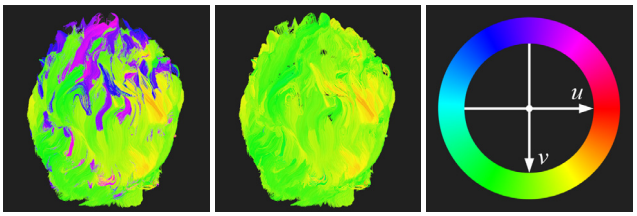

Figure 1: *Correcting Reversed Strands. From left to right: input hairstyle before correction; hairstyle after correction; the color code used to visualize the scalp direction of hair strands. In this case, the user only needs to draw one stroke.*

Hairstyle	*Num. of strands*	*Overall time (sec.)*
Tangled	72859	43
Puffy	84587	51
Wavy	69661	46
Curly	10000	11
Spiky	39695	17

Table 1: *The overall synthesis time.*

to enforce the detail-preserving ability of the combing tool. More specifically, we perform the following steps for each hair strand with n vertices:

- Define per-vertex local frames on these n vertices.
- Select $\lfloor n/p \rfloor$ vertices as control vertices, or *handles*. The parameter p is an integer that typically ranges from 5 to 10.
- Modify the position of these handles using the aforementioned combing algorithm.
- Compute the new local frames on the modified handles and interpolate them over the rest of the vertices.
- Solve the linear system described in Xu et al.'s work [2007] to obtain the position of all the vertices.

Please see our companion video for a live demo.

C Detailed Timing

We have measured the overall synthesis time of the five hairstyles in our paper on a Pentium 4 3.0GHz processor with 2GB memory, as shown in Table 1.

References

PARIS, S., HECTOR M. BRICE N., AND SILLION, F. X. 2004. Capture of hair geometry from multiple images. In *SIGGRAPH '04*, ACM, 712–719.

PARIS, S., CHANG, W., KOZHUSHNYAN, O. I., JAROSZ, W., MATUSIK, W., ZWICKER, M., AND DURAND, F. 2008. Hair photobooth: geometric and photometric acquisition of real hairstyles. In *SIGGRAPH '08*, ACM, 1–9.

WEI, Y., OFEK, E., QUAN, L., AND SHUM, H.-Y. 2005. Modeling hair from multiple views. *ACM Trans. Graph. 24*, 3, 816–820.

XU, W., ZHOU, K., YU, Y., TAN, Q., PENG, Q., AND GUO, B. 2007. Gradient domain editing of deforming mesh sequences. In *SIGGRAPH '07*, ACM, 84.