

Interactive Sketch-based Modeling of Braided Hair

Hari Hara Gowtham Jetti  and Amal Dev Parakkat 

LTCI-Telecom Paris, Institut Polytechnique de Paris, France

Abstract

Hair braids are widely used in various games and animated movies, thanks to their simplified representation and ease of animation. However, the existing research on modeling braids often relies on a limited dictionary of commonly seen hair braid patterns, constraining artists' ability to experiment by creating imaginary or creative hair braids. In this paper, we introduce a simple sketch-based interface for creating arbitrary hair braids. Our method employs a two-stage framework that first interprets a user-drawn sketch to extract the braid pattern. To accommodate arbitrarily drawn sketches, we then use a physics-inspired simulation to generate visually pleasing braids. In addition to automatically generating braids, our system allows users to interactively refine the braid pattern to create braids that match the user's imagination, facilitating experimentation and exploration of different braid structures.

CCS Concepts

• *Computing methodologies* → *Shape modeling*; • *Human-centered computing* → *Interactive systems and tools*;

1. Introduction

Realistic hair plays an important role in creating believable characters in games and movies. This inevitable role made the research on hair modeling and rendering a hot topic in Computer Graphics for decades [WBK*07]. Due to the challenges posed by the complexity of hair strands, their arrangement, intricate hair-light interactions and dynamic behaviour, it is particularly difficult to efficiently model and render hair. Thanks to the enormous research and recent advances in deep learning, there are now a lot of efficient ways to model and render hair [LOZ*24]. Due to the visual appeal, character design, ease of handling, and cultural significance, digital artists started using braided hair widely as an alternative to traditional hair. As the demand for more diverse and realistic digital characters grows, developing simple-to-use tools for creating arbitrary braided hairstyles becomes increasingly important.

While significant attention was received for hair modeling, braid modeling has received comparatively less attention. Following the seminal work on the theory of braids [Art47], research on braid modeling in Computer Graphics largely focused on repeatedly applying several basic rules, assuming that we see only a few hair braid styles in daily life. While effective for reproducing conventional braid patterns typically captured from photos [XYH*21] and RGB-D cameras [HML*14], such methods limit the creative possibilities available to digital artists to experiment and explore different imaginary braid patterns. Though it is possible to manually create custom braids using professional software like Blender and Maya, it is typically labour-intensive and error-prone, demanding meticulous attention to detail and high technical expertise.

Our main focus is to develop an interface that facilitates easy experimentation and possible exploration of braid patterns. We have two primary objectives: first, to provide a simple interface that allows users to easily specify the desired braid pattern, and second, to offer interactive editing of the underlying braid structure. To achieve the first objective, we employ a sketch-based approach, allowing users to quickly scribble the desired patterns, which are then layered to create the desired braid pattern. As the quick sketches lack precision, we deploy a physics-inspired simulation to generate visually appealing hair braids. In addition to this, the simulation also helps experiment with the braid strand thickness, adapting it as needed. For the second objective, we provide options to interactively update the automatically created braids by rearranging local layering and automatically updating the global layering of strands.

2. Proposed Approach

Our approach starts with the user scribbling a set of hair strands representing the desired braid pattern. The proposed method then uses this information to create the braided hair in three steps:

Generating pseudo braids: We create a pseudo cylindrical mesh around each user-scribbled strand. These meshes are termed "pseudo" as they may intersect other braids and do not yet represent a properly braided structure. The purpose of this cylindrical mesh representation is two-fold: 1) A cylindrical representation gives a more accurate volumetric perception of the braid structure, allowing a better understanding; and 2) The subsequent simulation step requires a mesh representation, allowing proper deformation and collision responses.

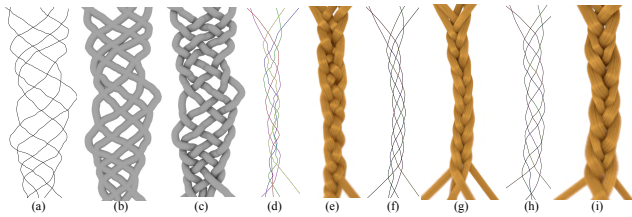


Figure 1: Left to right: (a) Input sketch, (b) Pseudo braids, (c) Layered pseudo braids, (d) Simulated braid strands, (e) Braid strands replaced with hair strands, (f) Simulated braid strands after interactive editing, (g) Edited braid strands replaced with hair strands, (h) Braid strands with interactively updated thickness, (i) Updated braid strands replaced with hair strands.

The system starts by defining a subdivision threshold and a cylinder radius, controlling the resolution and the volumetric properties of the braids. First, the user-drawn sketch, interpreted as a polyline, undergoes a subdivision. At each of these subdivided points, we create a circular cross-section using local coordinate frames. The cross-sections are then connected to form a continuous cylindrical surface as in generic sweep surfaces. To help with the collision detection during the simulation phase, we keep the subdivision threshold less than the cylinder radius. Figures 1(a) and (b) shows a sample sketch and corresponding pseudo braids.

Layering pseudo braids: The generated pseudo meshes that intersect (as all the braid strands lay on the drawing plane) are then layered appropriately by determining the over-under patterns to create a woven braid structure. This layering of pseudo braids prevents visual intersections by adjusting their positions along the drawing plane as shown in Figure 1(c). To do this, the system first selects the strand with the most intersections and alternatively labels each intersection point as ‘up’ or ‘down’. These labels are then propagated to other strands, ensuring alternating layering at intersections. While doing this layering, as usually seen in the literature, we assume that no more than two braid strands intersect at a point. Finally, the intersection points are translated based on their cylinder radius and the normal. This translation is smoothly propagated along the strands to maintain a natural appearance while avoiding overlaps.

Simulating layered braids: As shown in Figure 1(a), the user sketch could be rough, creating gaps in braids and making it visually unrealistic. To avoid this and to consider braid thickness, the layered pseudo braids are simulated as a spring system. The underlying uniformly resampled strokes are used to perform a massless-spring simulation by considering the points as particles following a force-based model and semi-implicit Euler time integration method with collision handling. To mimic our action in real life to tighten the braids, in addition to spring force and gravity, a tangential force is added to the strand ends. Figures 1(d) and (e) shows the effect of simulation on the layered pseudo braid shown in Figure 1(c).

In addition to being able to automatically generate braided hair, our system also provides options to interactively edit them to match the user’s imagination. Our current implementation allows users to update the local layering and braid thickness. To update local layering, users can select the intersection point to flip the ordering of the strands at that position. The effect of this flipping is then propagated

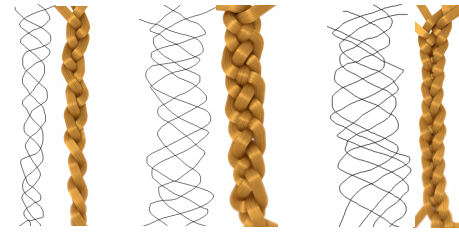


Figure 2: Three strand, four strand and five strands braids generated using our approach along with the user drawn sketches.

to other intersection points. In addition, the interface also allows users to manually override the ordering at any intersection point. Figure 1(g) shows an example where the local layering of Figure 1(d) is interactively updated to create a fishtail braid. The thickness adjustment increases the cylinder size around the strands and subsequently rearranges themselves accordingly. Figure 1(h) shows the effect of assigning varying thickness to different braid strands. More examples of braided hair structures automatically generated using our system are shown in Figure 2.

3. Future work

The proposed system offers a simple and interactive way to generate different aesthetically pleasing braid structures and provides the option to interactively edit them. However, our current implementation has two main limitations:

- Our current method of alternatively applying ‘up’ and ‘down’ labels at intersection points cannot always guarantee a consistent braid arrangement, which in some cases results in an unintended disentanglement at certain places. The current implementation does not automatically address this issue, requiring users to manually verify the resulting braid.
- We currently use a simplified cylindrical structure to represent braid strands, which, in reality, is much more complex due to the repositioning of individual hair strands.

We are currently working on an extension to address these issues. In addition, incorporating the options to animate braids and simulating the braids loosening over time would be interesting directions for further research.

References

- [Art47] ARTIN E.: Theory of braids. *Annals of Mathematics* 48, 1 (1947), 101–126. 1
- [HML*14] HU L., MA C., LUO L., WEI L.-Y., LI H.: Capturing braided hairstyles. *ACM Transactions on Graphics (TOG)* 33, 6 (2014), 1–9. 1
- [LOZ*24] LUO H., OUYANG M., ZHAO Z., JIANG S., ZHANG L., ZHANG Q., YANG W., XU L., YU J.: Gaussianhair: Hair modeling and rendering with light-aware gaussians. *arXiv preprint arXiv:2402.10483* (2024). 1
- [WBK*07] WARD K., BERTAILS F., KIM T.-Y., MARSCHNER S. R., CANI M.-P., LIN M. C.: A survey on hair modeling: Styling, simulation, and rendering. *IEEE transactions on visualization and computer graphics* 13, 2 (2007), 213–234. 1
- [XYH*21] XIAO C., YU D., HAN X., ZHENG Y., FU H.: Sketch-hairsalon: Deep sketch-based hair image synthesis. *arXiv preprint arXiv:2109.07874* (2021). 1