# Lend me your comb: faster grooming with an artist adaptive software

Jeremy Ringard*
ON Animation Studios

**Figure 1:** *The Little Prince ©2015 LPPTV LITTLE PRINCESS ON ENTERTAINMENT ORANGE STUDIO M6 FILMS LUCKY RED.*

## Abstract

In this paper, we present the solution we developped at ON Animation Studios to assist the Character FX department on the production of *Mune* and *The Little Prince*. Our goal was to design a tool that adapts to the artists rather than a solution that requires learning and practice. To do this, we provide the grooming artists the freedom to choose the interaction mode they feel the most comfortable with. Therefore, the grooming system we wrote provides three different approaches to create hair: curve sculpting, volume modeling, and sketch-based. While being different, these approaches remains compatible and can be mixed together depending of the artists' needs.

**Keywords:** hair, grooming, artist friendly, sketch

## 1 Hair grooming

The grooming solution we developped is splitted in two parts: first, a Maya plugin is used to create guide curves on the scalp, and a Guerilla Render procedural node used to generate a large amount of curves at render time. These curves are interpolated between the input guides with various additionnal parameters to control effects such as clumping or fuzzyness. This multi-resolution approach is the most common way of creating hair: more than just making the grooming task easier by reducing the complexity of data, it also makes the process of animation much faster since it just requires physics simulation of a few hundreds guides for a complete haircut.

Our approach focuses on two main challenges: Fist, we want to provide tools that make the artistic creation more natural by reducing the technical skills requirements. We also want to increase the artists productivity by allowing them to quickly adapt a haircut to various characters variations.

While the choice of a technology for artistic creation is mostly driven by the complexity of the result we want to achieve, to make a more productive tool, we also have to drive this choice by the diversity of the human skills we have access to. Every artist has a different experience and feeling: Some are used to manipulate the most common commercial software packages, and some others have a more organic approach based on drawing. To make our solution feel natural and efficient to everyone, we decided to provide a software package that includes various tools working in a very different but still intuitive manner.

*e-mail:j.ringard@onanimationstudios.com

All of these approaches generate the same data as standard Maya curves. This allows for non-exclusive choice of tools since all of them can be combined to produce the desired result.

## 2 Curves sculpting

The first toolset we wrote is mostly dedicated to artists who are experienced with grooming software packages or digital sculpting solutions. Just like in these softwares, our tool provides multiple brushes that can scatter curves on the surface of a scalp, then shape them directly by manipulating large groups of control vertices. All these brushes automatically adjust the result to make it physically coherent by conserving the curves length and preventing collisions. While the built-in brushes cover most of the requirements, we designed our API to allow for quick creation of additional brushes in order to fullfill any specific need we encounter during production.

## 3 Wig modeling

The second tool we designed has a more "modeler-friendly" approach. When creating a character, the modeling team is often requested to add simple polygonal shapes showing an approximate volume of the character's haircut. This process has the advantage of being very fast, and provide an acceptable preview of what the hair will look like without having to deal with complex guide and procedural generation. The limitation of these proxies is that it can only represent an empty surface, while an actual haircut is a more complex volume dataset. In order To use such a data for grooming, we must create a software solution which is able to reconstruct volume data.

The main concept of this tool is to give the grooming artist the ability to quickly prototype the shape of hair under the form of a polygon mesh using regular modeling tools provided with Maya. Our tool then analyses this surface and generate a coherent vector field to create a group of curves matching the internal volume of the mesh with a user defined density.

To avoid any perturbation caused by too knotty hairstyles, the software mostly relies on the mesh topology rather than its shape. To do this, it starts by identifying the strength lines available among the mesh edges. Once all these strength lines are collected, the tool analyses the vertex-to-vertex connection to establish a neighbourhood map of these lines. The neighbourhood map is eventually used to generate a clean vector field with a simple nearest neighbours interpolation algorithm. This approach makes the computa-

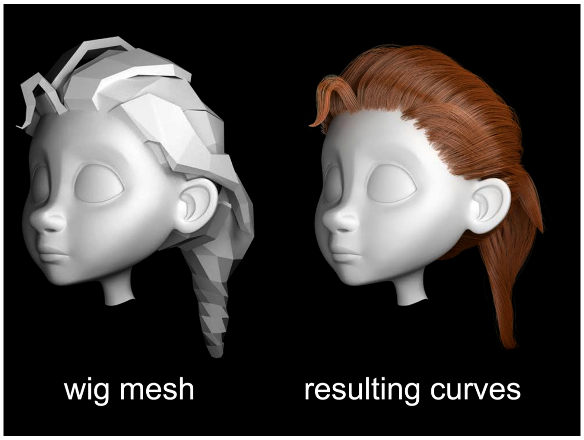


**Figure 2:** *Example of wig mesh and its result*

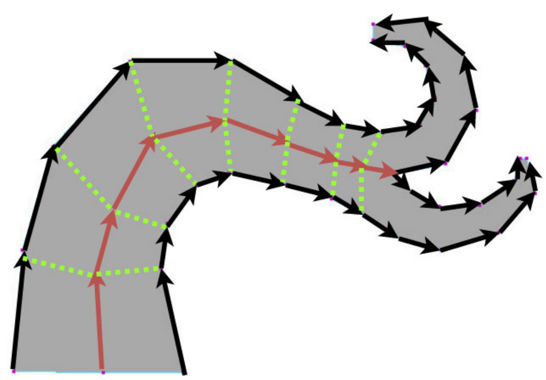

**Figure 3:** *2D view of the wig mesh analysis:: the strength lines (black) are used with neighbourhood informations (green) to produce the additional vectors (red)*

tion fast enough to generate the vector field in just a few seconds, even on high polycount meshes. The tool also offers the ability to build a non-uniform guides distrbution simply by manipulating the UV coordinates of the vertices.

This tool provides various advantages: First, it allows the artist to evaluate quickly the volumes and tweak it simply by applying transformations on few vertices. Moreover, since the user has to focus only on the bounding surface, he can easily create complex haircuts such as braids while having an accurate view of the interpenetrations. Obviously, the generated curves can then be tweaked to add details using the sculpting brushes presented on section 2.

This tool also provides a precious feature: Since it relies on a regular polygon mesh and doesn't require any construction history, the artist can save the wig and transfer it on another character. Adapting the mesh to a different head shape becomes a matter of minutes since it just requires few vertices adjustments. Grooming artists can also create their own wig catalog to quickly generate base curves for any haircut. This process is especially usefull for secondary characters, were the artists have to populate a crowd with many slightly different hairstyles.

## 4 Sketch based grooming

This third grooming solution requires virtually no technical knowledge since it's based on the exact same interaction mode as designing a character with percil and paper:

The design process of a character always starts with drawing. The artists make some sketches, adjust the lines, evaluate the silhouette and let the creation flow naturally through this process. Even for review of 3D assets, explaining the retakes through 2D paintover sketches remains the most common way to communicate between artists.

From this observation, we came up with the idea that sketch based 3D generation seems to be the most natural interaction mode. On this purpose, we developped a set of grooming tools that require almost no computer knowledge since it gives the artis the ability to create curves just by sketching hair directly on the viewport. These brushes not only create single curves, but can also create complete strands, including complex sketched width variations and twisting.

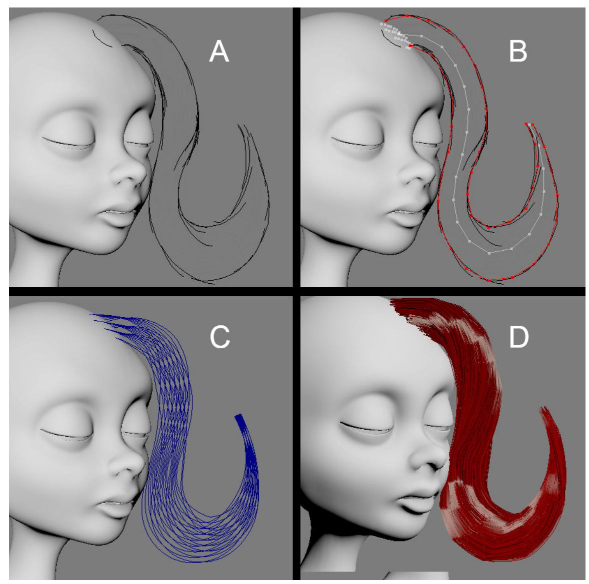


**Figure 4:** *Basic sketch based workflow: sketch made by the artist (A) sketch analysis and automatic construction of continuous lines (B) generated curves (C) shaded hair preview (D)*

The artist can then draw the silhouette of each strand directly on the 3D view of the scalp.

This proposal requires to solve 2 major challenges:

1. Sketching on a screen is a 2D interaction. The software must be able to build coherent 3D data by deducting the missing dimension.
2. The sketching process must be intuitive and should not constraint the artist. He must be able to draw naturally just like he's used to do when paper drawing. Basically, that means our software shouldn't require the artist to draw clean continuous lines: It must be able to understand a shape based on natural, multi-stroke sketches.

To solve the 2D to 3D problem, we developped various interpretations strategies to define how the brush will manage the depth dimension. These strategies include behaviors relative to the scalp collision (should the strand follow the shape of the geometry behind? should it remain straight while avoiding collision? Shall the brush allow for extreme Z-axis variations along the curves?). The artist can choose for each stroke the depth strategy he wants to use. He also can change it dynamically during the drawing process. For instance, when drawing a specific strand, we might want the first inches to be snapped on the head, and leave the rest of the strand free. Since we can't rely only on an algorithm to correctly deal with all the subtle variations that could occur in the third dimension, our tool also allows the user to correct it manually. To do this, he just has to change his point of view in the viewport. When moving the camera, the tool automatically creates a sketch of what the interpreted volume would look in this new point of view, and the user is free to correct it by drawing over. This process is not destructive because the software keeps track of all the sketches performed for each point of view. As a consequence, the additionnal corrections always apply to a volume rather than just on a 2D representation.

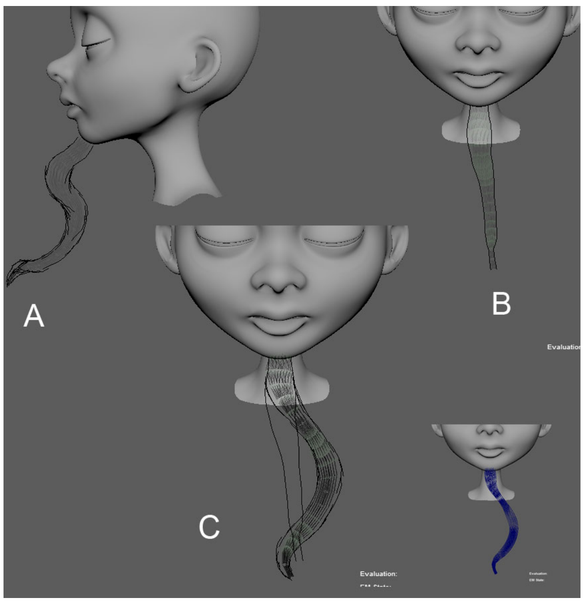


**Figure 5:** *Example of depth corrections: view from the first sketch (A) reconstructed sketch after moving to front view (B) manually corrected shape (C)*

The challenge of allowing natural sketching has been identified by observing the artists while they draw some paintovers during reviews: rather than drawing perfectly clean lines, they usually perform a large amount of strokes, go over the previous lines, adjust the curvature, or change their mind and start a new stroke with a different shape. Basically, a drawn line is never continuous but rather made of many shorter lines that we called "substrokes". Based on this observation, we developped an algorithm to create clean lines interpretation from these substrokes in realtime. To do this, our software retrieves every substroke under the form of a list of 2D points connected by straight lines. These lines can be interpreted as sparse vectors that can be associated weights based on various parameters:

1. Accidental strokes can happen while drawing. To avoid any perturbation, our algorithm finds the vectors that don't seem relevant, based on attributes such as their distance from the rest of the sketch or their length (a long substroke is less likely to be accidental)

2. Time is also a relevant information: since we're able to keep track of the drawing order, we can figure out which substroke has been performed last among a set of substrokes. This information is critical since it makes the software able to understand when a recent substroke has been sketched for correcting the older ones.

Using this data, the algorithm can build a continuous 2D line showing an accurate interpretation of the stroke. The software can display this interpretation in realtime so the user can evaluate its validity, and keep sketching if he wants to feed the tool with more data in order to get a more accurate result.

For the conversion into 3D curves, our software converts the 2D lines into 3D using depth interpretation algorithm, then builds a 3D structure made of multiple cylinders whose radius match the sketched silhouette width. These cylinders are eventually used as bouding volumes to grow curves.

this sketch-based solution gives pretty good results and allows to quickly research and design haircuts. As usual, all the curves generated through this process can be modified using the regular sculpt brushes. The sketch interpretation system proved to be satisfying enough to be integrated into some other tools of our pipeline, mostly for geometry deformation and rig control.

## 5 Future works

The next step of this work will be oriented towards the finalling of the groom. Although the procedural approach makes perfect sense to generate hundred thousands of curves, it creates strong limitation on the level of detail the artists can control: This process is still quite linear since it splits the grooming process into two separated parts. The guide curves grooming on one side, and the procedural configuration on the other side.

Such a linear approach doesn't give the users the ability to manually groom individual hair fibers after the procedural generation. We would like to avoid this limitation by allowing the artist to use the presented tools directly on a several hundred-thousands curves groom. We started exploring this approach by developping a specific GPU node that can store and display a large amount of shaded curves. This node can be combed in realtime using GPU-accelerated brushes directly inside Maya.

This process will raise new challenges regarding animation: By getting rid of the guide curves system, we might loose the advantage of having a consistant low resolution representation of the haircut for dynamics simulation. Since simulating such a large amount of curves would be computationally expensive, we have to figure out a way to decimate the haircut and be able to generate guide curves just for simulation. This process would introduce a notion of LOD in simulation that could lead to great optimizations. It would ease the work of the simulation team by providing the ability to define the level of detail needed for each character on a per-shot basis, by using low density setups for background characters.

## Acknowledgements

We would like to acknowledge Alexis Casas for his help on the redaction of this paper, Boris Laprade for his illustrations and the Guerilla Render team for their support in the developpement of the procedural generator.