Curvature Analysis of Sculpted Hair Meshes for Hair Guides Generation

Florian Pellegrin[123] and Andre Beauchamp[2] Eric Paquette[1]

[1] École de technologie supérieure, Montreal, Canada
florian.pellegrin.1@ens.etsmtl.ca
[2] Ubisoft La Forge, Montreal, Canada
[3] INSA Rennes, Rennes, France

Abstract. This paper proposes an approach that generates hair guides from a sculpted 3D mesh, thus accelerating hair creation. Our approach relies on the local curvature on a sculpted mesh to discover the direction of the hair on the surface. We generate hair guides by following the identified strips of polygons matching hair strands. To improve the quality of the guides, some are split to ensure they correspond to hairstyles ranging from straight to wavy, while others are connected so that they correspond to longer hair strands. In order to automatically attach the guides to the scalp of a 3D head, a vector field is computed based on the directions of the guides, and is used in a backward growth of the guides toward the scalp. This approach is novel since there is no state-of-the-art method that generates hair from a sculpted mesh. Furthermore, we demonstrate how our approach works on different hair meshes. Compared to several hours of manual work to achieve a similar result, our guides are generated in a few minutes.

Keywords: Hair synthesis, Hair modeling, Hair generation, 3D avatar, 3D computer graphics

1 Introduction

3D hair modeling is a complex task and is one of the difficult topics in the field of 3D character creation. Several types of representations are used for hair, namely, closed meshes, polygon strips and individual hair strands. Typically, artists work in two steps: first, they create hair guides, and then they either create other hair strands by interpolating the guides or they generate polygon strips along the guides. Sculpting a hair mesh with tools such as ZBrush is very efficient, as compared to the time-consuming process of manual hair guide creation. Since guide creation accounts for about fifty percent of the total time required to make a hairstyle, our goal is to automate the process by generating hair guides from a 3D hair mesh. Figure 1 illustrates the hair mesh (left) and the lines representing the hair guides (right). While 3D hair sculpting still calls for manual work, it gives freedom and control to the user in the creation process.

Despite several methods currently exist, none of them is sufficient to meet our need. For example, Yuksel et al. [1] have proposed a method that generates

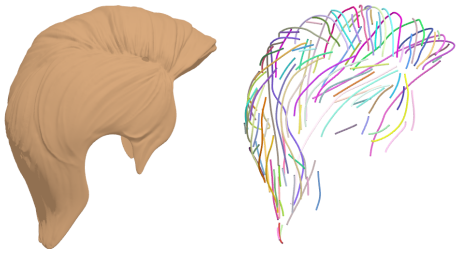


Fig. 1: Example of a desired output for a given input.

individual hair strands from a 3D mesh. The method is used by some video game productions, but it is not entirely adequate since it calls for significant manual work. Other methods use images as inputs, which makes it hard to control the generated hair. Recently, several data-driven methods require the use of 3D surfaces with identical topologies. However, such a limitation is too constraining for a game production environment. More generally, current methods are not based on meshes and seek to generate a complete hairstyle, leaving little to no room for intervention by the user. Our approach uses two inputs, namely a 3D hair mesh and a 3D head onto which the generated guides will be connected. The hair mesh is analyzed to extract polygons along the hair stands based on the curvatures of the mesh. Guides are generated along these polygons. The curvature analysis is not perfect, and so we subsequently validate if consecutive guide curves can be connected together. We then use a signed distance field to push outward hair guides which intersect the 3D head mesh. Finally, hair guides are extended through a vector field generated from the guides' growing directions in order to connect the guides to the scalp of the head. Our contributions can be summarized as follows:

- First ever approach to convert a sculpted hair mesh into guide curves;
- Extraction of guide curves through a curvature analysis and curve connections;
- A signed distance field approach to push guides out of the 3D head mesh;
- A vector field approach to connect the guide curves to the hair root region.

2 Previous work

Work related to hair generation generally falls under three categories [2]: physics-based methods, geometry-based methods, and finally, image-based methods. Physics-based methods [3,4,5,6] are no longer as popular as they used to be, because using them is difficult. Indeed, the selection of physical parameters to obtain a desired hairstyle is not intuitive.

For geometry-based methods, they favor the use of parametric representations as well as interactions with the user. Such methods are very similar to manual modeling techniques available to artists in the industry. These methods

include the modeling of parametric surfaces using splines. Here, the splines are used to represent 2D surfaces [7] or a 3D mesh [8]. As an alternative to the use of splines, the methods of Yang et al. [9] and Choe and Ko [10] favor modeling by hair clusters and statistical distribution. Unlike these preceding methods which focus on the shape of an individual hair strand, Yuksel et al. [1] propose to model the outer surface of the hair. Since one of our goals is to use a mesh to generate a hairstyle, this method is therefore related to our approach. However, the mesh in the method of Yuksel et al. [1] must be created according to certain constraints and does not allow to use any user provided mesh which our approach allows.

Image-based modeling methods are comprised of two major types: those based on a single image and those based on multiple images. Satisfactory results can be obtained with multi-view methods [11,12,13,14,15,16,17,18]. However, due to the specialized hardware often required, these methods are harder to implement than those based on a single view. Many image-based methods rely on a 2D or 3D orientation field representing the hair flow. The field is usually built using a set of Gabor filters to detect the orientation. The method by Chai et al. [19] is based on a single image and uses 2D orientation maps. However, the individual hair strands generated by their method lack physical plausibility, and this was improved by Chai et al. [20]. These methods require the prior existence of images representing a desired hairstyle and do not allow an artist to slightly alter the input to achieve different results.

Another class of image-based methods try to reconstruct a 3D hairstyle using a data set of 3D meshes. Depending on the input provided by the user, a combination of several examples in the data set is made in order to create a new mesh. The use of a data set in order to combine multiple hairstyles based on user-drawn paths on the input image was introduced by Hu et al. [21], while the need for human interaction to determine hair orientation was mitigated by the work of Chai et al. [22], which introduced the use of a CNN to automatically segment the image and detect orientations. The method by Hu et al. [23] extracts the semantic features of a hairstyle in an image to select a subset of meshes sharing these attributes from the data set. The mesh closest to the image is then retrieved and adjusted to improve the correspondence. The methods by Chai et al. [22] and Hu et al. [23] have two major problems. The approaches call for the availability of the data set at runtime, which requires a large storage space. In addition, the hairstyles inside the data set are required to share an identical mesh topology. However, such a constraint is impractical when considering several projects and numerous artists in medium to large video game studios.

The need for a data set at run time is solved by neural network methods [24,25,26]. Indeed, these methods use neural networks to generate hair and only require the hair data set for training. Zhou et al. [26] propose an auto-encoder to generate hair directly from an image. In the method, the image is segmented and the hair orientation field is extracted. This 2D field is then provided to the network to generate 3D point sequences representing the coordinates of individual hair strands. Similarly, Zhang and Zheng [25] and Ye et al. [24] propose a generative antagonist network and an auto-encoder respectively. However

the objective of these networks is to generate a 3D vector field, and not a set of hair strands directly. These methods present the same problems as those encountered using images to generate hair. Moreover some methods [24,25] generate vector fields instead of hair, and rely instead on proper hair growing algorithms.

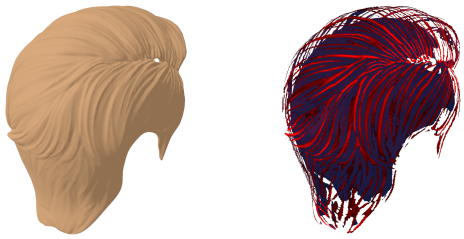
None of the methods presented fully satisfies our needs. The geometric methods are not very automated and require a significant user interaction. For their part, although promising, the methods based on one or more images are fundamentally contrary to our objective which is to use a 3D mesh as input. Finally, methods relying on a data set impose mesh topology constraints that are too limiting.

3 Hair mesh curvature analysis

The proposed approach consists in generating hair guides from a sculpted 3D hair mesh (Figure 1). The input mesh is a manifold with or without boundaries, and has enough polygons to allow visualizing smooth sculpted hair strands. Apart from the hair mesh, the user also provides a head mesh to which the guides will be "attached". To ensure that the approach is independent of the size of the hair and head, the user scales the sculpted mesh and the head mesh to their real sizes in millimeters before processing begins. The generation of the guides starts with a curvature analysis, which is used to extract regions of the mesh representing hair strands (§ 3.1). Then, splines are fitted along the regions extracted from the curvature analysis (§ 3.2). These guides are then analyzed to ensure that the strands correspond to hairstyles ranging from straight to wavy (§ 3.3). Some of these guides are then connected together as they should represent a single and longer hair strand (§ 3.5). We then fit the guides to a head mesh. Since the guides can intersect the head, the approach resolves these intersections by repulsion according to the signed distance field of the head (§ 3.6). Finally, the guides are extended through a vector field to be attached to the scalp of the head (§ 3.8).

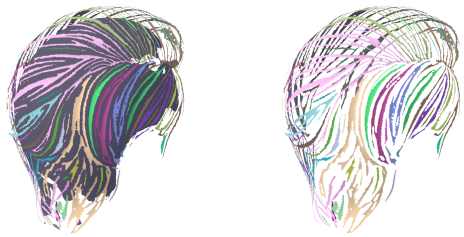
3.1 Curvature analysis

Our approach computes the curvature to find hair strands along the surface (Figure 2). We compute the Weingarten equations and its determinant to obtain the Gaussian curvature. Triangles with their vertices having a curvature lower than 0.001 are retained. Thus, only the concave regions remain (Figure 2(b)). The next step in the analysis is to separate the mesh into different groups based on the connectivity of the triangles, after which the groups are processed to remove those which are unlikely to represent hair strands. In this regard, we identify groups of connected triangles. These groups are randomly colored in Figure 3(a). A group is deleted if the diagonal of its oriented bounding box is smaller than 15 mm. Furthermore, we identify groups that are unlikely to represent hair strands as they are too wide. We retain the groups for which the geodesic distance from each internal vertex to a boundary vertex is less than a



(a) Sculpted mesh. (b) Concave regions.

Fig. 2: Identification of strands from the curvature.



(a) (b)

Fig. 3: Mesh connectivity. (a) Connectivity groups. (b) Mesh after deleting groups that are unlikely to represent hair strands.

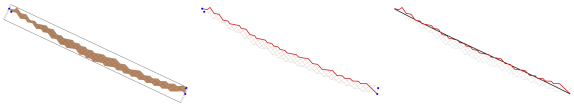
threshold set to 20 mm (Figure 3(b)). This process also removes the inner part of the sculpted mesh (the part lying close to the scalp).

3.2 Generation of guides

Our aim is to generate guides along the remaining parts of the mesh following the curvature analysis, and to that end, we implement two different strategies. Firstly, we detail the strategy to generate guides by computing the shortest path between the extremities of the groups. This approach works particularly well when the groups resulting from the analysis represent individual hair stands. Secondly, we describe an alternative way to obtain guides by computing a skeleton extraction on the parts of the mesh remaining after the curvature analysis. In our approach, both strategies are run and the output that generated the largest number of guides is automatically selected.

Shortest path strategy. For many of the groups in Figure 3(b), we observe a correspondence between a group and a hair guide. The goal here is to find the extremities of these groups and to then fit a guide from one end to the other. We iterate over each connectivity group to generate one guide per group. To find the extremities of the group, we compute the oriented bounding box. The extremities

will be close to a corner of the box. Therefore, for each corner of the box we retrieve the nearest vertex of the group (Figure 4(a)). As a result, we obtain eight vertices or less since multiple corners can share the same nearest vertex. Among these vertices, we identify the extremities by computing the shortest paths along the edges of the mesh from each vertex to all others. Thus, among the generated paths, the longest one allows to find the extremities (Figure 4(b)). Each guide is computed by fitting a NURBS of order 6 to the selected path (Figure 4(c)).



(a) (b) (c)

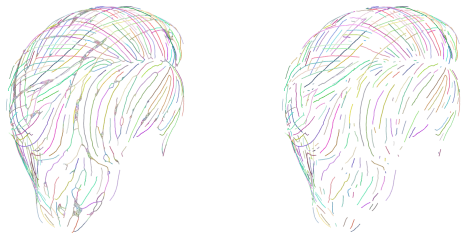
Fig. 4: Guide generation: (a) Selection of the nearest vertex for each corner of the bounding box; (b) Selected path (longest of the shortest paths between selected vertices); (c) Fitting of a spline to the vertices of the selected path.

Skeleton extraction strategy. One shortcoming of the shortest path strategy is that only one guide is generated per connectivity group. This implies a great loss of information when separate hair strands remain connected after the curvature analysis (Figure 5). To overcome this, we have developed an alternative strategy

Fig. 5: Mesh group representing multiple hair stands.

using the mesh skeleton to generate the guides. We obtain the skeleton of the mesh using a mesh contraction method similar to that of Au et al. [27] (Figure 6(a)). In order to generate splines from the skeleton, all joints connecting a bone to more than two other bones are removed (Figure 6(b)). Then, NURBS of order 6 are fitted to the polyline of each bone.

Our approach uses both the shortest path and the skeleton extraction strategies. In the case of a high-resolution mesh with fine details, the width and distance of their hair strands groups from each other are smaller than the mesh contraction fusing distance. Thus, the skeleton extraction fails at separating hair stands from each other. It is hard to predict which strategy will work best, as



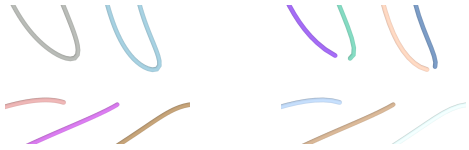
(a) Generated skeleton. (b) Disconnected skeleton.

Fig. 6: Skeleton extracted from the mesh.

results depend on how the mesh was sculpted. However, it appears that the strategies are complementary: when one performs poorly, the other produces good results. Thus, one strategy is automatically chosen based on the one that generated the greatest number of guides.

3.3 Plausibility of guides

After generating the guides, we check if they correspond to the expected shape of hair strands for straight to wavy hairstyles. We use the method proposed by Luo et al. [14] to detect U-shaped splines. In order to cut these splines, points are evenly sampled along them such that each segment's length is equal to 2 mm (guides are henceforth represented as polylines). Then, the local extrema of the curves are defined for each change of sign of the first derivative of the height values (with respect to the gravity vector). All the points labeled as extrema are then removed when the distance between two consecutive extrema is greater than 10 mm (Figure 7(b)).



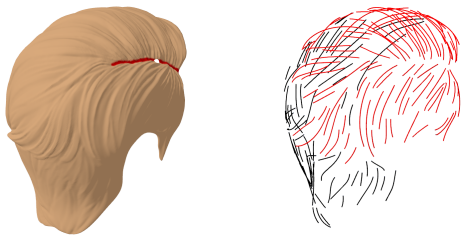
(a) (b)

Fig. 7: Splitting of implausible curves.

3.4 Direction analysis

We will now attempt to define the growing direction of each guide by setting one end as the root and the other as the tip. Most of the hair in a hairstyle

is affected by gravity and is therefore oriented from top to bottom. However, some hair strands flow from bottom to top depending on the position of the hair parting line. This phenomenon can be seen in Figure 8(a), where the guides above the parting line represent hair passing over the head going from bottom to top. In our approach, we use a parting line to determine the root of the guides that are close to it. The user manually paints the hair parting line on the sculpted mesh (Figure 8(a)). For these guides, our approach firstly defines



(a) (b)

Fig. 8: Definition of the parting line: (a): the parting line painted on the mesh; (b): the guides affected by the parting line. (in red)

the root position as its closest end to the parting line. A guide is affected by the parting line when the distance between one of its ends and the parting line is less than a user-defined parameter ($\in [100, 180]$ mm for all of our examples, and specifically 110 mm for Figure 8(b)). Secondly, the direction of the guides that are not affected by the parting line is determined according to the height value calculated from the polyline vertices in Section 3.3. For each guide, the root is associated with the highest end. The user chooses the global direction of the hair (falling or rising) beforehand in order to take into account the fact that a hairstyle may represent hair rising up. When hair is rising up, there is no parting line and the directions of the hair are reversed.

3.5 Connection of guides

Notwithstanding the fact that the guides are usable as is, some of them could be connected together to match the hair of the sculpted mesh. We perform a connection analysis which includes a subset from the method proposed by Luo et al. [14]. We follow their method to detect potential connections, but we also propose our own strategy to decide which ones to connect. The detection of potential connections considers all pairs of guides whose ends are sufficiently close to each other (less than a threshold of 30 mm). A circle is fitted to the extremities of each pair of guides and the guides form a potential connection if the fitting error is below a given threshold. For each endpoint used to create a pair, its 9 neighboring points on the curve are used to form a set containing the

10 points closest to the end of the curve. Each pair forms a set Q of 20 points used to fit a plane. The plane fitting is done by a least squares fit. The points in Q are projected onto the fitted plane, and then the least squares method is used to fit a circle. The remaining potential connections are those for which the root mean squared error is below a threshold of 1.0 mm.

The above detection of potential connections is derived from the method of Luo et al. [14]. We now propose to sort all potential connections by increasing fitting error. We then iterate over each potential connection and connect the extremities of the guides if none of them is already connected. Since guides are represented as polylines (Section 3.2) we fit splines to all the points of a newly connected pair to ensure smooth connections between the guides. The splines of the newly connected guides are then converted back to polylines, as shown in Section 3.3. The result of this step can be seen in Figure 9.

(a) (b)

Fig. 9: Connection of guides: (a): guides before connection; (b) guides after connection where connected guides are represented in red.

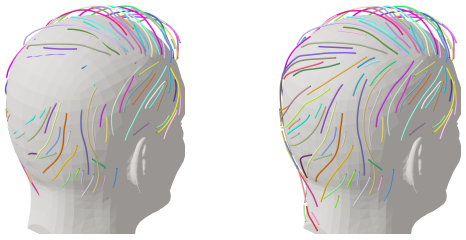
3.6 Repulsion of guides

Our approach aims to generate hair guides from any sculpted 3D hair mesh. This implies that the sculpted mesh may not correctly fit the head intended to receive the guides, and intersections may occur between the generated guides and the head. Fitting to a 3D morphable head model [28] would also require adjusting for intersections. As a solution, we propose a method to repulse the guides penetrating the interior of the head using a signed distance field. First, it is easier to set the inside apart from the outside of a mesh when the mesh is a closed manifold. Since this is rarely the case for this type of mesh, we use standard mesh closure methods. Next, we transform the mesh into a VDB [29] voxel field (Figure 10). We use a resolution fine enough to limit any loss of information during the discretization (each voxel is of size 1 mm). The VDB field is sampled to obtain the signed distance of the nearest surface to any position in the field. In addition to the distance, the gradient of the signed distance function provides the normalized vector of the direction to the surface (Figure 10(c)). Since our

(a) (b) (c)

Fig. 10: Head voxelization with signed distance field. (b) represents the VDB field for the head in (a). (c) shows the segment representing the gradient multiplied by the signed distance of a given position. For visualization purposes, (b) and (c) show 4 mm voxels instead of the effective resolution of 1 mm we use.

guides are discretized with many points, repulsing only the points inside the head would have the effect of breaking the curved appearance of the guides. Our repulsion method mitigates this by temporarily converting the guides back to splines, as shown in Section 3.2, in order to repulse the control points penetrating the interior of the head rather than the points of the discretized guides. For each control point, the distance and gradient in the VDB volume are sampled at the position of the control point, and then the control point is projected onto the surface such that $\mathbf{p} = \mathbf{p} + f(\mathbf{p})\nabla f(\mathbf{p})$ where $\mathbf{p}$ is the position of the control point, f is the distance function and ∇f is the gradient of the function. Each control point is further translated by 0.2 mm along the normal of the vertex on the surface closest to the control point. Splines are then converted back to polylines, as shown in Section 3.3. The result of the repulsion process is shown in Figure 11.



(a) (b)

Fig. 11: Repulsion of guides: (a) shows the intersection of the guides with the head mesh, while (b) shows the result of the repulsion of the guides.

3.7 Vector field generation from guides

With the objective of connecting the hair guides to the scalp, we compute a vector field describing the overall flow of the hairstyle based on the generated guides. We transform the hair mesh into a grid of voxels and we will associate these voxels with a growing direction. The generated guides do not cover the entire surface of the mesh and therefore provide sparse information about the hair directions. In order to fill in the areas of the field where no guides are detected, we describe a method to interpolate the directions within the field. After interpolation, the field could contain noise if the direction analysis (Section 3.4) partly failed, and thus has oriented some guides in the wrong direction. (i.e. some voxels in the field do not follow the general direction of their neighbors). To overcome this problem we present a smoothing approach using a median vector filter.

Similarly to what is provided in Section 3.6, the hair mesh is voxelized using the VDB structure.

For effiency reasons, this time, the size of a voxel is set to 3.0 mm. The vector field is represented as a direction for each voxels. With the voxels $\{h_i\}$ of the field H and the points $\{g_j\}$ of the guides, the directions $d(h_i)$ are defined according to Equation 1 where $\mathcal{N}(h_i)$ is the set of g_j inside voxel h_i.

$$d(h_i) = d\left(\underset{g_j \in \mathcal{N}(h_i)}{\operatorname{argmin}} \|g_j - \operatorname{center}(h_i)\|_2^2\right) \tag{1}$$

The directional field obtained (Figure 12) contains sparse information: only a

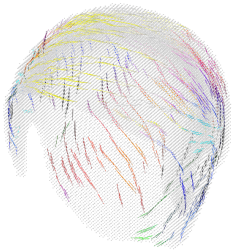


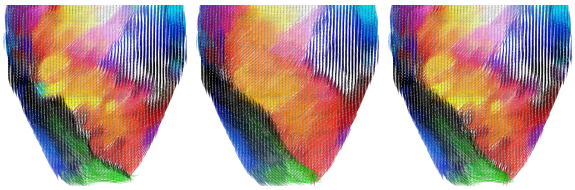
Fig. 12: Transfer of hair directions from the guides to the voxel field.

subset of voxels is associated with hair growth directions. In order to obtain a complete vector field, we leave the N voxels which were assigned a value by Equation 1 unchanged, and we interpolate from these N known directions to the other voxels using Shepard's method [30] defined by Equation 2:

$$\mathbf{d}'(h_i) = \frac{\sum_{j=1}^{N} w_j(h_i)\mathbf{d}(h_j)}{\sum_{j=1}^{N} w_j(h_i)}, \tag{2}$$

where $d'(h_i)$ is the interpolated direction for voxel h_i based on the known directions $d(h_j)$. Each direction is weighted by $w_j(h_i)$ where $p = 4$ defines the influence of the closest voxels:

$$w_j(h_i) = \frac{1}{\|\text{center}(h_i) - \text{center}(h_j)\|_2^p} \,. \tag{3}$$



(a) (b) (c)

Fig. 13: (a) the interpolated directional field, (b) result after applying the median vector filter, and (c) result when the median vector filter is applied only when there is a strong change in direction.

As shown in Figure 13(a), the interpolated field contains a little bit of noise related to a few guides having directions that are too different from that of their neighbors. To correct this type of artifact, we apply a median vector filter over the field. The objective of this filter is to find the median direction among the directions around the center of voxel h_i:

$$\mathbf{d}_m(h_i) = \underset{\mathbf{h}\in\mathcal{N}(h_i)}{\text{argmin}} \sum_{\mathbf{h}_k\in\mathcal{N}(h_i)} \|\mathbf{d}'(h) - \mathbf{d}'(h_k)\|_2^2 \tag{4}$$

where this time $\mathcal{N}(h_i)$ corresponds to the list of voxels h_k such that $\|\text{center}(h_k) - \text{center}(h_i)\|_2^2 < 30$ mm and $\mathbf{d}_m(h_i)$ is the median direction among the $\mathcal{N}(\langle\rangle)$ neighbors. With direction $\mathbf{d}_m(h_i)$ the result in Figure 13(b) is obtained. Although the previously mentioned noise is eradicated, a loss can be noted in the variation of the directions. To overcome this, we define Equation 5 such that:

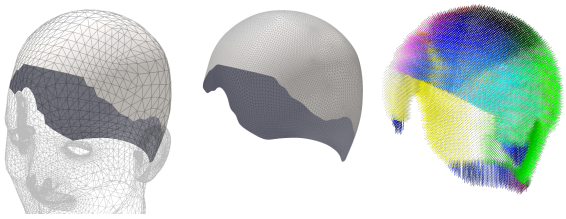
$$\hat{\mathbf{d}}(h_i) = \begin{cases} \mathbf{d}_m(h_i) & \angle\mathbf{d}'(h_i), \mathbf{d}_m(h_i) > 18^\circ \\ \mathbf{d}'(h_i) & \text{otherwise} \end{cases} \,. \tag{5}$$

Thus, the median direction is used only when it is different enough from the interpolated direction.

3.8 Scalp connection

A hairstyle without scalp connection looks unrealistic and makes hair simulation [31] difficult. Our last step grows the root of the guides up to the scalp.

For this, the polygons belonging to the scalp are first defined by the user (Figure 14(a)). Generally, the heads of the same project share the same mesh topology. Thus, the scalp identification can be transferred to all heads. Only the scalp polygons are kept, and then subdivided until the polygons have an area less than 8 mm^2 (Figure 14(b)). The vector field S (Figure 14(c)) represents the direction to the nearest vertex of the scalp, while the vector field H represents the direction of the hair in the hairstyle. The vector field S is composed of the same voxels as H. We then grow root ends of the guides through H and S. Until the scalp



(a) (b) (c)

Fig. 14: Computation of directions to scalp.

is reached, we iteratively add new points $\mathbf{p}_i$ to each guide, where $\mathbf{p}_1$ represents the root point of the guide and $\mathbf{d}_1$ is the reversed growing direction of the root point. The positions $\mathbf{p}_i$ and directions $\mathbf{d}_i$ of these points are calculated as:

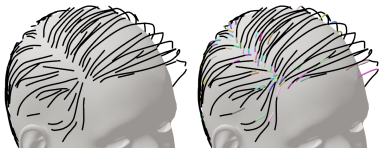
$$\mathbf{d}_{i+1} = t[S(\mathbf{p}_i) - \frac{1}{2}H(\mathbf{p}_i)] + (1-t)\mathbf{d}_i \tag{6}$$

$$\mathbf{p}_{i+1} = \mathbf{p}_i + \delta\mathbf{d}_{i+1} \tag{7}$$

where t is the effect rate of the new direction $S(\mathbf{p}_i) - \frac{1}{2}H(\mathbf{p}_i)$ such that $t \in]0, 1]$, δ is the step size, with $\delta > 0$, and $S(p)$ and $H(p)$ correspond to the direction of the closest point to p in each field. We use $t = 0.15$ with a step size $\delta = 2.0$ mm. To connect the guides exactly to the scalp surface, the extension of a guide stops when a point penetrates inside the head. This is detected using the signed distance field calculated in Section 3.7. Figure 15 shows the guides before and after they are connected to the scalp.

4 Results

We present the results of the entire process of our approach based on six different meshes in Figures 16 and 17. These 3D sculpts were manually adjusted to a single head. The hairstyles represented are diverse: various hair lengths, with or without parting lines and different levels of detail. As shown in Figure 18, the sculpting of more guides involves the use of a mesh with more polygons. For each



(a) (b)

Fig. 15: Guides before and after the scalp attachment.

example in Figure 16 and 17 the user manually specified the parting line and adjusted its influence distance (in the range of $[100, 180]$ mm). Apart from the manual adjustment related to the parting line, the guides are all generated with the same parameters. As our approach is the only one using sculpted meshes to generate hair our results can presently not be compared with those of state-of-the-art methods.

During the generation tests we used a computer with an Intel Xeon W-2135 processor, 64 GB of RAM and an NVIDIA GeForce RTX 2070 8 GB graphics card. Our method is implemented in the Houdini software and most of the code is written in VEX. All VEX code is automatically parallelized by Houdini when possible. The generation time varies between 17 and 124 seconds for the presented results. Specifically, Figure 19 compares the time spent on the major steps of our approach.

In order to evaluate our approach, a qualitative survey was established and sent to a hair modeling expert working in the video game industry. The evaluation considered a complete hairstyle creation by an artist and thus considered the impact of using these guides to model a 3D hairstyle. According to the feedback, the time saved increases with the complexity of the hairstyle. Moreover, the method does not alter the creative process of the artists and allows to obtain good guides.

Our approach generates curves based on a shortest path and a skeleton extraction. Figure 20 compares curves generated with both approaches.

Although the results are satisfying, our approach has some limitations. As in the case with other recent research, our approach performs well for simple hairstyles, but would struggle for complex ones. For example, the hairstyle in Figure 17(c) has hair sections that have opposite flow directions. In such a case, one could separate the hair in multiple meshes and apply our approach with the use of the parting line to orient the hair correctly. As our method is designed to provide control to the user at each step, for Figure 17(c) we selected the incorrectly oriented guides and reversed their direction. Things get even worst for hairstyles with curly hair, as well as for hair tied in braids [32] or buns. Indeed, when the internal structure of the hair is not represented by the external structure of the mesh, the results will not be as useful.

The length of the guides is sometimes strongly heterogeneous. This is notably visible on Figure 16(a), where neighboring guides can have significantly different lengths. On the other hand, in Figure 17 the results obtained show that the

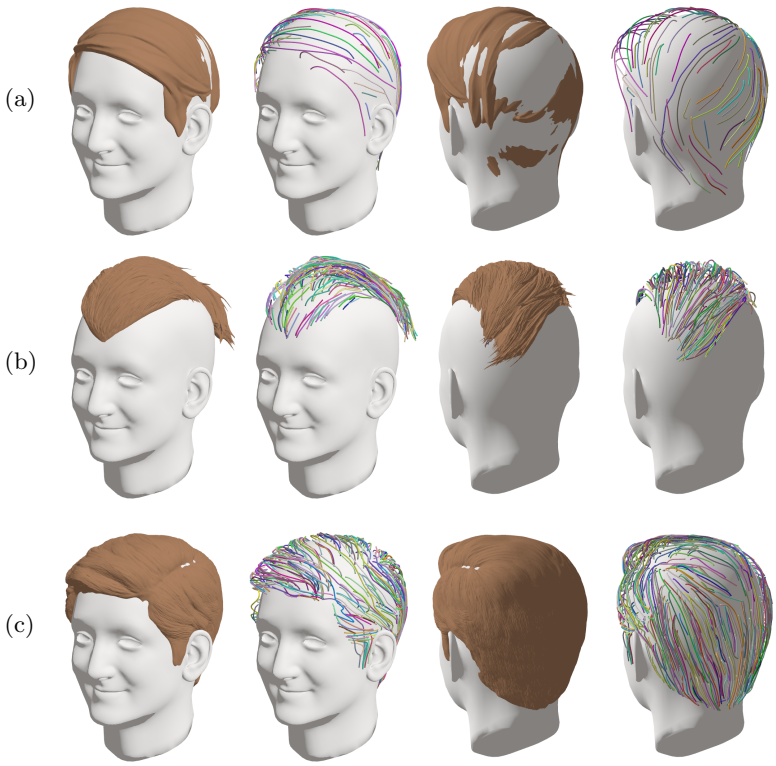


Fig. 16: Results of the approach for short hair.

step involving the connection of the guides to the scalp hardly works with long hair. Indeed, on Figure 17(a), some guides are connected to the cheeks and the ears. On Figure 17(c), several guides connect to the scalp before reaching the ponytail rallying area. Despite apparent flaws, the guides match the meshes well and follow their shapes correctly.

As a result of the curvature analysis, hair strands are sometimes too close to each other. Consequently, the skeleton extraction method (Section 3.2) is not able to contract each group individually. Therefore, after deleting points with more than two connections, there are sometimes a few segments left to generate guides. The result of this phenomenon is illustrated in Figure 21(b). We propose an alternative that grows hair from the scalp to solve this situation. Indeed, guides can be traced through the vector field generated in Section 3.7, as shown in Figure 21(c).

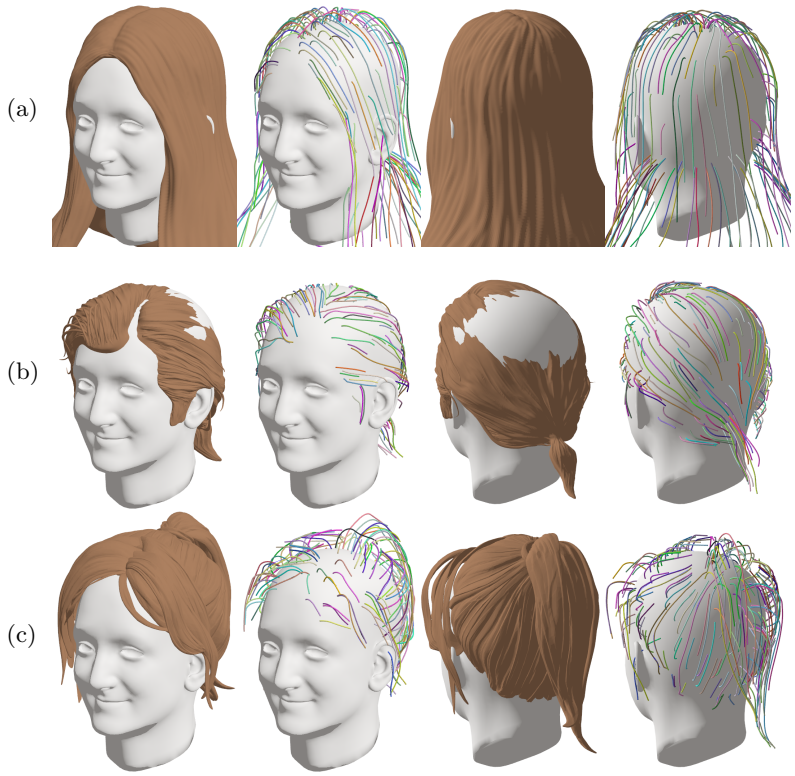


Fig. 17: Results of the approach for long hair.

5 Conclusion

Artists are able to quickly model hair using 3D sculpture software. The modeled hair represents the 3D volume of a hairstyle. The objective of our approach is to generate hair guides from the mesh of a hair volume. First, an analysis of curvature is performed to extract the polygons along hair strand regions. Then, two complementary strategies generate guides according to the polygons extracted from the curvature analysis. The first strategy separates the polygons into groups according to their connectivity and then uses the shortest path along edges to identify a guide for each group. The second strategy is based on the reconstruction of a skeleton for the extracted polygons to fit guides to the bones. The result with most guides between these two strategies is selected and used as input for the next steps. These guides are then analyzed to define their root ends. Then, they are connected together when possible. Beyond the generation of guides, we also perform the fitting of the guides to the mesh of a head. Indeed, our method corrects the guides to solve any collision with the head. This is done by pushing back the guides according to a signed distance field, obtained through

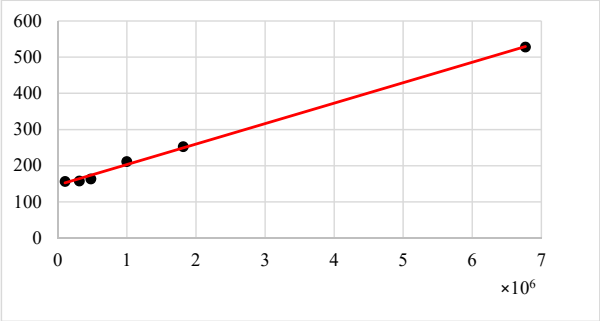


Fig. 18: Number of generated guides according to the number of triangles.

a voxelization process of the head. Subsequently, a vector field is constructed by interpolating the directions of the guides, and is then used to trace connections from the guides to the scalp. Thus, this approach allows a user to obtain guides connected to a head. Moreover, these guides can be directly used in 3D software that uses guides to create dense hair.

During the generation process, we build a dense vector field representing the hair flow and use it to attach the guides to the scalp of a head. Nevertheless, it would be interesting to exploit this field for other purposes. Our method could loop on itself and use this generated field as new input information to improve the result of some steps. Also, this field together with an efficient hair growth algorithm would allow our approach to generate not only guides, but also individual hair strands from a mesh.

6 Acknowledgment

This work was supported by Ubisoft Inc. and École de technologie supérieure. We would also like to thank SideFX™ for providing Houdini™ licenses for research.

References

1. Yuksel, C., Schaefer, S., Keyser, J.: Hair meshes. ACM Trans. Graphic. **28**(5), 1–7 (2009)
2. 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), 213–234 (2007)
3. Hadap, S., Magnenat-Thalmann, N.: Interactive hair styler based on fluid flow. In: Computer Animation and Simulation 2000, pp. 87–99. Springer (2000)

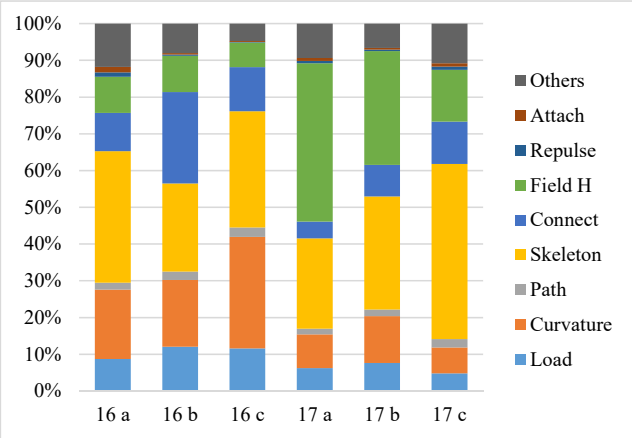


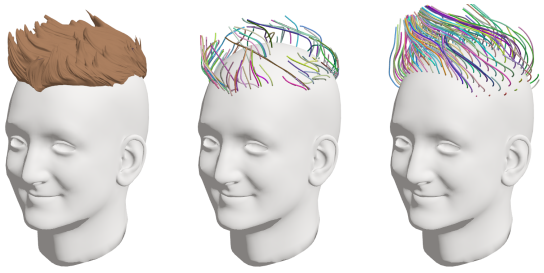
Fig. 19: Comparison of the generation time. The meshes used are those from Figure 16(a-c) and Figure 17(a-c).

4. Rosenblum, R.E., Carlson, W.E., Tripp III, E.: Simulating the structure and dynamics of human hair: modelling, rendering and animation. The Journal of Visualization and Computer Animation **2**(4), 141–148 (1991)
5. Stam, J.: Multi-scale stochastic modelling of complex natural phenomena. Ph.D. thesis, University of Toronto (1996)
6. Yu, Y.: Modeling realistic virtual hairstyles. In: 9th Pacific Conference on Computer Graphics and Applications. Pacific Graphics 2001, pp. 295–304. IEEE (2001)
7. Liang, W., Huang, Z.: An enhanced framework for real-time hair animation. In: 11th Pacific Conference on Computer Graphics and Applications, 2003. Proceedings., pp. 467–471. IEEE (2003)
8. Noble, P., Tang, W.: Modelling and animating cartoon hair with nurbs surfaces. In: Proceedings Computer Graphics International, 2004., pp. 60–67. IEEE (2004)

(a) Curves shortest path.

(b) Curves skeleton.

Fig. 20: Generated curves.



(a) (b) (c)

Fig. 21: Limitation of guide generation by direct curvature analysis and alternative solution.

9. Yang, X.D., Xu, Z., Yang, J., Wang, T.: The cluster hair model. Graphical Models **62**(2), 85–103 (2000)
10. Choe, B., Ko, H.S.: A statistical wisp model and pseudophysical approaches for interactive hairstyle generation. IEEE Transactions on Visualization and Computer Graphics **11**(2), 160–170 (2005)
11. Bao, Y., Qi, Y.: Realistic hair modeling from a hybrid orientation field. The Visual Computer **32**(6), 729–738 (2016)
12. Cao, C., Wu, H., Weng, Y., Shao, T., Zhou, K.: Real-time facial animation with image-based dynamic avatars. ACM Trans. Graphic. **35**(4) (2016)
13. Luo, L., Li, H., Paris, S., Weise, T., Pauly, M., Rusinkiewicz, S.: Multi-view hair capture using orientation fields. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition, pp. 1490–1497. IEEE (2012)
14. Luo, L., Li, H., Rusinkiewicz, S.: Structure-aware hair capture. ACM Trans. Graphic. **32**(4), 1–12 (2013)
15. Paris, S., Briceno, H.M., Sillion, F.X.: Capture of hair geometry from multiple images. ACM Trans. Graphic. **23**(3), 712–719 (2004)
16. Paris, S., Chang, W., Kozhushnyan, O.I., Jarosz, W., Matusik, W., Zwicker, M., Durand, F.: Hair photobooth: geometric and photometric acquisition of real hairstyles. ACM Trans. Graphic. **27**(3), 30 (2008)
17. Wei, Y., Ofek, E., Quan, L., Shum, H.Y.: Modeling hair from multiple views. ACM Trans. Graphic. **24**(3), 816–820 (2005)
18. Zhang, M., Chai, M., Wu, H., Yang, H., Zhou, K.: A data-driven approach to four-view image-based hair modeling. ACM Trans. Graphic. **36**(4), 156–1 (2017)
19. Chai, M., Wang, L., Weng, Y., Yu, Y., Guo, B., Zhou, K.: Single-view hair modeling for portrait manipulation. ACM Trans. Graphic. **31**(4), 1–8 (2012)
20. Chai, M., Wang, L., Weng, Y., Jin, X., Zhou, K.: Dynamic hair manipulation in images and videos. ACM Trans. Graphic. **32**(4), 1–8 (2013)
21. Hu, L., Ma, C., Luo, L., Li, H.: Single-view hair modeling using a hairstyle database. ACM Trans. Graphic. **34**(4), 1–9 (2015)
22. Chai, M., Shao, T., Wu, H., Weng, Y., Zhou, K.: Autohair: Fully automatic hair modeling from a single image. ACM Trans. Graphic. **35**(4) (2016)
23. Hu, L., Saito, S., Wei, L., Nagano, K., Seo, J., Fursund, J., Sadeghi, I., Sun, C., Chen, Y.C., Li, H.: Avatar digitization from a single image for real-time rendering. ACM Trans. Graphic. **36**(6), 1–14 (2017)

24. Ye, Z., Li, G., Yao, B., Xian, C.: Hao-cnn: Filament-aware hair reconstruction based on volumetric vector fields. Computer Animation and Virtual Worlds **31**(4-5), e1945 (2020)
25. Zhang, M., Zheng, Y.: Hair-gan: Recovering 3d hair structure from a single image using generative adversarial networks. Visual Informatics **3**(2), 102–112 (2019)
26. Zhou, Y., Hu, L., Xing, J., Chen, W., Kung, H.W., Tong, X., Li, H.: Hairnet: Single-view hair reconstruction using convolutional neural networks. In: European Conference on Computer Vision (ECCV), pp. 235–251 (2018)
27. Au, O.K.C., Tai, C.L., Chu, H.K., Cohen-Or, D., Lee, T.Y.: Skeleton extraction by mesh contraction. ACM Trans. Graphic. **27**(3), 1–10 (2008)
28. Ghafourzadeh, D., Fallahdoust, S., Rahgoshay, C., Beauchamp, A., Aubame, A., Popa, T., Paquette, E.: Local control editing paradigms for part-based 3D face morphable models. Computer Animation and Virtual Worlds p. e2028 (2021)
29. Museth, K.: Vdb: High-resolution sparse volumes with dynamic topology. ACM Trans. Graphic. **32**(3), 1–22 (2013)
30. Shepard, D.: A two-dimensional interpolation function for irregularly-spaced data. In: Proceedings of the 1968 23rd ACM National Conference, p. 517–524. Association for Computing Machinery, New York, NY, USA (1968)
31. Jiang, J., Sheng, B., Li, P., Ma, L., Tong, X., Wu, E.: Real-time hair simulation with heptadiagonal decomposition on mass spring system. Graphical Models **111**, 101,077 (2020)
32. Sun, C., Ramachandran, S., Paquette, E., Lee, W.S.: Single-view procedural braided hair modeling through braid unit identification. Computer Animation and Virtual Worlds **32**(3-4), e2007 (2021)