FurCollide: Fast, Robust, and Controllable Fur Collisions with Meshes

Arunachalam Somasundaram
DreamWorks Animation
arun.somasundaram@dreamworks.com

Figure 1: *FurCollide* used in a wide variety of situations including fur collision with skin, thin cloth, ropes, and for grass.

ABSTRACT

We present *FurCollide*, a fast, robust, and artist friendly tool used for collision detection and collision resolution of fur curves with meshes. The tool helps artists interact with and control tens of thousands of curves with ease while providing high fidelity realistic and/or artistic collision results. This tool is in use at *DreamWorks Animation* and has been used in a wide variety of fur and/or grass collision situations in various films.

CCS CONCEPTS

•Computing methodologies →Animation;

KEYWORDS

fur, collision, mesh, pinch, fast, robust, control

ACM Reference format:
Arunachalam Somasundaram. 2017. *FurCollide*: Fast, Robust, and Controllable Fur Collisions with Meshes. In *Proceedings of SIGGRAPH '17 Talks, Los Angeles, CA, USA, July 30 - August 03, 2017,* 2 pages.
DOI: http://dx.doi.org/10.1145/3084363.3085051

1 MOTIVATION

A fur groom or a grass field can have hundreds of thousands or millions of curves. Several collision objects can interact with those curves during animation such as skin, cloth, harness, belt, ropes, ornaments etc. For high fidelity collisions we need to work with at least several thousands or tens of thousands of curves. Given the high density and fine resolution of the curves, exacting collisions with full resolution complex collision objects may be desired to avoid artifacts. Speed and robustness become very important as cleanup work can be painstaking. Several times, the collision objects can pinch or cut through the growth surface providing no physically plausible collision resolution. Procedural approaches such as shrinking the curve to avoid collisions or controlling direction of collision resolution, can produce artistically pleasing results reducing distraction. Using a physical simulation engine on a high density of curves for precise collision purposes can be slow to iterate with, artistically difficult to control, and technically difficult to deal with non-physical conditions such as pinching. *FurCollide*, using a non-physical simulation approach that can be artistically controlled, handles all of the above issues in a fast and robust way.

SIGGRAPH '17 Talks, Los Angeles, CA, USA
 978-1-4503-5008-2/17/07…$15.00
DOI: http://dx.doi.org/10.1145/3084363.3085051

2 OVERALL PROCESS

The overall steps to introduce fur motion with collision are:

- Uniformly sample curves in areas of interest and extract a percentage of the render curves (≈ 15%) to work with at reference pose using a custom *'Fur Sampler'* tool.
- Rigidly attach and move those curves with the animated growth surface using a custom *'Fur Attach'* tool.
- Perform fur collision with meshes using *FurCollide*.
- Introduce procedural wind or secondary motion, if needed, before *FurCollide* or after it while respecting collisions.
- Use these curves as motion guides in the geometry shader as mentioned in [O'Hagan et al. 2015] to deform the groom.

3 FURCOLLIDE

3.1 Collision Detection and Pinch Detection

The collision mesh is triangulated and the triangles are stored in an Octree data structure. Curves, whose bounding box lies within the bounding box of the collision object, are identified. Each segment of those curves, starting from the root (first) segment, is checked for collision with the triangles until the first intersecting segment is found. First, the appropriate octant(s) of the Octree in which the curve segment lies is hierarchically identified using a segment-octant bounding box overlap check. Next, for triangles that lie in the identified octant(s), a segment-triangle bounding box overlap check is done. If that check passes, a true segment-triangle intersection is performed. The normal of the collision object at the point of intersection, and the intersecting curve segment are both stored.

A closed collision mesh is required for pinch detection. For pinch detection, an inside test is performed to see if the root CV of the curve lies inside the collision object, where a physically plausible collision resolution is not possible. For the inside test, a ray is shot from the root CV of the curve in a random direction. If the ray intersects more back faces than front faces, then the root CV is inside the collision object and the curve is considered to be pinched.

3.2 Collision Resolution

3.2.1 Collision Rotation. The colliding curves are rotated to resolve collisions based on user parameters. The default local direction of rotation is away from the collision object in the direction of the object's colliding point normal (or its inverse if cutting through a back face of a single-sided surface). A local axis of rotation using this direction is calculated during the first hit of the curve, and used for rotation until the curve recovers back to its original state. This eliminates jitters or pops that can happen in an animation as hit points and their normals, and correspondingly their direction of collision resolution can change over frames. The user can choose the *'Attract to Growth Surface'* option to make the direction of collision resolution orient towards the closest point on the growth surface from the midpoint of the curve. This can produce the effect of patting down the fur or grass to resolve collisions. The *'Rotation Precision'* parameter (with a default value of 0.25 degrees) can be used to specify the angular rotational step size by which the curve segments should incrementally rotate by to resolve collisions.

FurCollide has the following three modes of curve rotation:

- *Stiff:* A colliding curve is rotated at the root about the calculated local axis, keeping its original shape. This mode can be used for short curves.
- *Bend:* All the segments of a colliding curve are rotated to bend about the calculated local axis. The overall shape of the curve is maintained except for a bend from the root to the tip. This mode can be used for longer curves.
- *Soft:* Each segment of a colliding curve can be rotated about a different local axis. Each colliding segment can affect the curve CVs further down the curve with a falloff based on *'Affected Connected CVs'* and *'CV FallOff'* parameters. This can change the shape of the curve, but can help it navigate tightly in complex situations or keep curve deformations closer to the colliding segments. Length correction is applied to account for localized CV deformations.

3.2.2 Collision Shrinking. An option to shrink the colliding curves along their length as the curves rotate is provided. Shrinking is only active in the *'Stiff'* and *'Bend'* modes. *'Scale Minimum'* parameter is used to set the minimum scale that a curve can shrink to. *'Scale Angle'* parameter is used to set the rotation angle at which the minimum scale is achieved. The fur segments are scaled, using a linear ramp, from a value of 1.0 to the *'Scale Minimum'* parameter value as the rotation angle ranges from 0.0 to the *'Scale Angle'* parameter value. If the fur segment's rotation angle is greater than the value specified by *'Scale Angle'*, the scaling is clamped at the value specified by *'Scale Minimum'* parameter. This allows the segments to be scaled down as they rotate further with a clamp on scaling. This can sometimes help reduce the look of fur cutting through one another at widely different angles to avoid distraction.

3.2.3 Recovery. Once the curves clear collision, they are allowed to recover back towards their original state over time using the *'Recovery Rate'* parameter. That parameter controls how quickly the curve recovers back (by rotating and lengthening) to its original default shape and state. The recovery stops if the curve hits a collision object during recovery, as collision overrides recovery.

3.2.4 Plasticity. The amount of recovery can be controlled using the *'Plasticity'* parameter (0.0 allows full recovery, 1.0 keeps the curves in their state of maximum rotation, an in-between value is a linear ramp on the recovery amount). This parameter helps with leaving an imprint of the collision object onto the fur curves.

3.2.5 Pinch Resolution. Pinched curves have two options:

- *Stop Solving:* The local rotation of the pinched curves are left at the prior state to pinching. This helps avoid unnecessary jitters trying to solve the impossible.
- *Put to Growth Surface:* Rotate to the growth surface.

The pinched curves can start recovery once they come out of pinching using the *'Recover Pinched Fur'* parameter.

4 WIND / SIMPLE SECONDARY MOTION

Procedural Wind and or simple secondary motion (e.g. jiggle) can be added to the input curves before passing it to *FurCollide*. Depending on the nature of the incoming motion (if not too extreme for example), this works in most cases. Such motion can also be added after running the curves first through collision using *FurCollide*. A *'collider distance'* attribute is first stored on each simulated collision free curve CV using a custom *'Fur Collider Distance'* tool. This distance is the closest distance of that CV's segments to the collision object. Any motion imposed on the curve CVs after that is restricted to be within its collider distance using a custom *'Region of Influence'* tool. This helps maintain the collision free state when motion is introduced, similar to the technique described in [O'Hagan et al. 2015]. The restriction of motion can sometimes lead to curve bunching artifacts which can be improved by using a higher density of curves. This technique leads to accurate collisions even during extreme conditions such as high noisy fur wind.

5 IMPLEMENTATION AND USAGE

FurCollide, and associated tools such as *'Fur Sampler'*, *'Fur Attach'*, *'Fur Collider Distance'*, and *'Region of Influence'* tools are implemented as custom nodes in a third party procedural package and are multi-threaded. The artist can control the various parameters of the *FurCollide* node to achieve the desired artistic collision results. This fur collision workflows helps to resolve collision for both rigid skin-bound curves and curves that have motion such as procedural wind or simple secondary motion. The tool does not require the fur to start at a collision free state. The user can also turn off the collisions of certain curves at any frame, maintaining their prior local state for future frames. For fur collision with non-polygonal objects such as a braid of hair, a collision mesh is built around the braid and used to collide with the fur curves. On an average, when colliding about 50,000 curves, 8 CVs each, with a high resolution collision object consisting of about 130,000 triangles, *FurCollide* operates at about 4 frames per second using 16 CPUs. The tool has been used in a wide variety of fur collision situations including collision with skin, cloth, braid, rope, belt, harness, and other accessories and/or for grass in various films at *DreamWorks Animation*.

REFERENCES

Colleen O'Hagan, Arunachalam Somasundaram, and Jason P. Weber. 2015. Hair Smash. In *ACM SIGGRAPH 2015 Talks (SIGGRAPH '15)*. ACM, New York, NY, USA, Article 35, 1 pages. DOI: http://dx.doi.org/10.1145/2775280.2792543