# Building and Animating Cobwebs for Antique Sets

Fangwei Lee* Alex Ongaro Domin Lee August Meredith
DreamWorks Animation

## Introduction

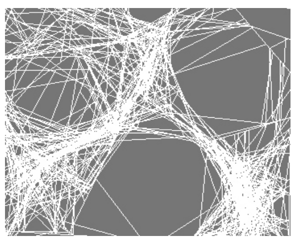
When working with antique sets, cobwebs are one of the essential elements that set up the right atmosphere. The complex nature of cobwebs makes them difficult to build by hand. We want to utilize the accuracy of simulated effects, the speed and predictability of procedural modeling and the beauty of hand modeling. This talk describes the mixed pipeline that we created, which was first introduced in DreamWorks Animation's film *Shrek Forever After* and Halloween TV special *Monsters vs Aliens: Mutant Pumpkins from Outer Space* and is used in several upcoming projects.

Before applying the new solution, our first approach was to use standard surface texturing. Unfortunately, there are three issues with this technique: first, surface textures do not create enough depth and parallax. Although this can be solved by creating more overlapping layers, painting these layers is tedious. Second, while hand-modeled webs are artistically beautiful, they do not necessarily form correct physics catenary and any small biases can break the realism that we are trying to achieve. Third, UV stretching becomes a problem with complex surfaces. To solve the above issues of modeling antique sets, we created this system which speeds up modeling and animating with procedural operators and light-weighted simulations.

## Physics for Cobwebs

Cobwebs are webs built by spiders that have collected dust. Although spiders spin webs at regular intervals, those old webs are built, destroyed, re-built, and expanded for years. Those chaotic natural events give the webs irregular and fractal structures. Thus the silky spider threads covered by dust and droplets produce the anisotropic and fuzzy texture.

### The Structure



The continuous build-and-destroy cycle creates a self-similar structure which can be defined recursively. We discovered that recursively creating line segments from two points on two line segments produces cobweb-like structure. For $L_0$ is the set of initial segments:

$$L_0 = \{S_0, S_1, S_2 ...\}$$

$$L_{n+1} = L_n + \overline{XY}\ , where\ X\ , Y \in L_n$$

### The Statics

Observationally, a hanging string forms a catenary- the theoretical shape of a hanging flexible chain or cable when supported at its ends and acted upon by a uniform gravitational force (its own weight) and in equilibrium. The curve has a U shape that is similar in appearance to the parabola:

$$y = a\cosh\left(\frac{x}{a}\right) = \frac{a}{2}\left(e^{x/a} + e^{-x/a}\right)$$

Because a parabola can be calculated much faster than catenary, we use the parabolic equation for speed and simplicity.

$$y = ax^2 + bx + c$$

Because a catenary in 3D space is actually on a 2D plane we can convert two 3D points into a vertical plane space: A(0,0), B(u,v). Given gravity factor g, we obtain:

$$a = \frac{g}{4}\ ,\ \ b = \frac{v - \frac{g}{4}u^2}{u}\ ,\ \ c = 0$$

yield:

$$y = \frac{g}{4}x^2 + \frac{v - \frac{g}{4}u^2}{u}x$$

and we use this new equation to approximate catenary curves.

### The Dynamics

Considering cobwebs as connected threads, the dynamics can be solved by cloth physics. Basic cloth simulations consist of structure, shear, and bend springs. Because spider threads have minimal thickness, the webs have no shear springs and very weak bend springs.

## Implementation

To produce the cobweb, we divide the system into five parts: a modeling toolset that allows us to freely model the web without restriction; a cobweb simulation engine; a procedural geometry shader to create fine details, which are otherwise tedious to do by hand; an animation tool; and a shader network. This approach is compatible with our standard geometry-based studio pipeline.

### Modeling

Modelers start with a clean prop without any web. First, they define the basic web structure quickly and intuitively with line segments, which form the *base curves* (figure 1). Second, we set up constraints to adhere the curves to the prop and connect to each other. The modelers place pin locators on sets to fix the web points in space and snap multiple points together to outline the rough structure. (figure 2)

After finishing the design, we run a procedure that converts *base curves* into thin polygon meshes. Cloth simulations are run with force settings on the resulting polygon web and polygon edges are extracted as newly deformed curves. This procedure deforms our original straight curves, and creates new curves in a physics-based form. (figure 3)

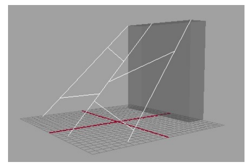


figure1: *base curves* are like beams or bones for the cobweb structure.

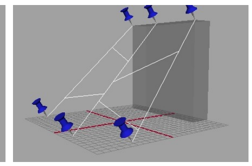


figure 2: Put pin locators to hang webs in space during simulations.

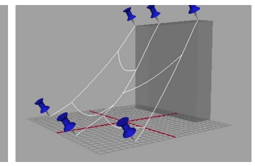


figure 3: Apply cloth simulation to *base curves.*

### Knitting Procedure

The simulated *base curves* are used as our infrastructure on which details build. We select curves to knit together and execute the knitting procedure to generate *filled curves*. New curves are generated recursively; we choose two random points on two random *base curves* or *filled curves*, and create new *filled curves* to connect them (figure 4).

We use parabola equations to approximate catenary curves. Given the positions of two points, number of segments, and gravity factor, the procedure calculates theoretical positions and moves control points on the filled curves (figure 5). After modelers are done with the cobweb design, we apply a final pass of the knitting procedure to create the *detailed curves* which represent the finest details for our cobwebs (figure 6).

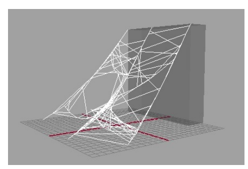


figure 4: Knit *filled curves* between *base curves.*

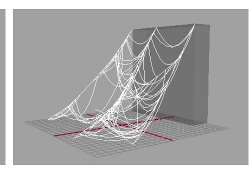


figure 5: Offset curve points with catenary equations to apply deterministic gravity effect.

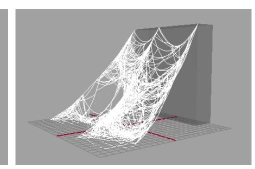


figure 6: Recursively create *detailed curves* to bring more complexity to the web.

### Animation

Cobweb animation is done in Maya with its native dynamic fields and cloth physics. To optimize the process, only the thin polygon extruded from the *base curves* are used in the simulations. Because *filled curves* are still connected to *base curves* via construction history, forces applied on the base curves, move the filled curves with them as if they are simulated as well. Using the extruded polygon as a cloth object, we define basic atmospheric environment with wind and turbulence fields (figure 8). And then we animate the constraints to detach specific components of the *base curves* at desired time during simulation.

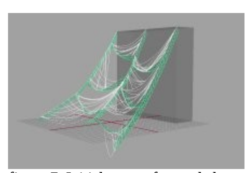


figure 7: Initial setup of extruded polygon from *base curves.*

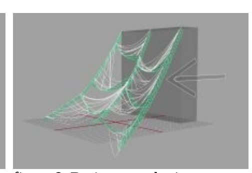


figure 8: Basic atmospheric environment setup.

figure 9: Example on table prop.

### The Texturing and Shading

Generally, cobwebs are composed of two different kinds of materials- silky spider threads, and a fuzzy layer created by dust clinging to the threads. For rendering threads, we loft geometry tubes from the curves and apply an anisotropic material. During the modeling phases, the modeler can also make *cobweb patches* by lofting between two curves. Those patch surfaces are then painted by texture artists to add another layer of artistic control and to create finer details that are too expensive or require specific art direction for the knitting approach.

## In Production

One FX artist first defined the look and behavior for the show, and then modeling department used the toolset to build the cobweb models. The modelers also lofted a few shapes between the biggest areas of the webs to make NURBS patches that the surfacing department uses to apply textures resembling the fine cobweb detail. These curves and patches are then installed to the model library and picked up automatically for any shot in production.

 The definitive version was published in SIGGRAPH 2011 Talks, https://doi.org/10.1145/2037826.2037913.