A Simple Physics Model to Animate Human Hair Modeled in 2D Strips in Real Time

Chuan Koon Koh and Zhiyong Huang
School of Computing, NUS, Singapore 117543
Email: {kohchuan | huangzy}@comp.nus.edu.sg

Abstract

This paper presents a simple Physics model to animate human hair modeled in 2D strips in real time. A major difficulty in animating human hair results from the large number of individual hair strands in a hairstyle. To address this problem, we have presented a framework of human hair modeling based on grouping hair strands into strips. Each hair strip is modeled by one patch of parametric surface. Polygon tessellation and the alpha-mapping using hair textures are then applied. To continue that work, we present a simple Physics model to the strip-based hair model. In particular, a simple dynamic model is adapted and applied to the control point meshes. A set of dynamics equations are defined and solved for the control points. The parametric representation of hair strips can handle a deformation of any complexity and still appear smooth. Moreover, because the number of control points is much smaller than the tessellated triangles, the computation is fast to achieve real time animation. The animation of hair is controlled using event-triggered procedural animation primitives that implement wind, gravity as well as head motion. Inter hair strip collision avoidance is achieved by introducing springs between neighboring hair strips. Collision avoidance of hair and other objects are implemented using ellipsoids and reaction constraints.

Keywords: Hair animation, Physically-based modeling

1. Introduction

Hair modeling and animation are very challenging tasks in human animation due to the presences of a large number of hair, the complex interaction of light and shadow amongst them, and the small scale of a hair strand's width compared to an image pixel. Furthermore, the dynamics of hair motion requires some physics model together with collision detection and response of hair.

Various methods have been employed to model and animate human hair [Anjy92, Chen99, Dald93, Kong99, Neyr98, Wata92]. These approaches concentrate mainly on modeling hair accurately, and often require specialized rendering algorithms. As such, hardware acceleration is unlikely to be available for the above approaches, making them more suitable for off-line graphics systems.

We have presented a strip-based framework that is suitable for real-time applications [Koh00]. The main idea is to model and animate hair in 2D strips. Each hair strip, modeled by one patch of parametric surfaces in particular NURBS, represents a group of hair strands. A variety of shapes may be defined for each strip. For the rendering, we apply alpha-mapping on the tessellated polygons to achieve a realistic visual effect.

However, in [Koh00], animation is achieved by keyframing of the control points of hair strips. It is very difficult to create motion sequences with naturally-looking and Physically-plausible movement. In this paper, we present a simple Physics model to the strip-based hair model. In particular, a simple dynamic model is adapted and applied to the control point meshes. A set of dynamics equations are defined and solved for the control points. The parametric representation of hair strips can handle a deformation of any complexity and still appear smooth. Moreover, because the number of control points is much smaller than the tessellated triangles, the computation is fast to achieve real time animation. The animation of hair is controlled using event-triggered procedural animation primitives that implement wind, gravity as well as head motion. Inter hair strip collision avoidance is achieved by introducing springs between neighboring hair strips. Collision avoidance of hair and other objects are implemented using ellipsoids and reaction constraints.

2. Related Work

There are four basic problems to solve in order to produce realistic human hair: hair modeling and creation, hair motion, collision detection and response, and hair rendering [Dald93].

Since all motions are governed by Physical laws, almost all hair animation work is based on different physics models [Terz88]. Two well-known approaches include a method using one-dimensional projective differential equations and pseudo-force fields [Anjy92] and a method using mass spring model [Rose91].

An integrated system for modeling, animating and rendering hair is described in [Dald93]. It uses an interactive module called HairStyler [Thal93] to model the hair segments that represents the hairstyle. Hair motion is simulated using simple differential equations of one-dimensional angular moments as described in [Anjy92]. Collision detection is performed efficiently with a cylindrical representation of the head and body [Kuri93]. Detected collisions between hair strands and the body will respond according to the reaction constraint method [Plat88].

3. A Brief Summary of the Strip-Based Hair Model

To make this paper self-contained, we first brief summarize the strip-hair model [Koh00]. We model hair in 2D strips. The motivation is to reduce the large number of geometric objects when each hair strand is individually represented. A hair strip is a group of hair strands in the shape of thin flat patch (Fig. 1d), which are modeled geometrically by NURBS surfaces. Each surface patch represents one hair strip with different shape and size for different hairstyles. Thus, all the hair strands are represented in layers of strips overlaying each other on top of the scalp. A real-world human head has around 100,000 hair strands. For simplicity, a 3D hair model typically uses less, perhaps around 20,000 strands. A hairstyle will then need around 800,000 line segments if each hair strand uses 40 segments.

For rendering, we tessellate the NURBS representation into polygon mesh. The Oslo algorithm is implemented using the multiple knot insertion for the tessellation [Cohe80, Meye91]. If more than a few knots are being inserted at once, the Oslo algorithm is more efficient than the Böhm algorithm [Böhm80]. Finally, texture maps of hair images can be applied on either one or both sides of each surface patch. The alpha map defines transparency and creates an illusion of complex geometry to the otherwise "rectangular" surfaces and adds to the final realism (Fig 1a, b, c, and e).

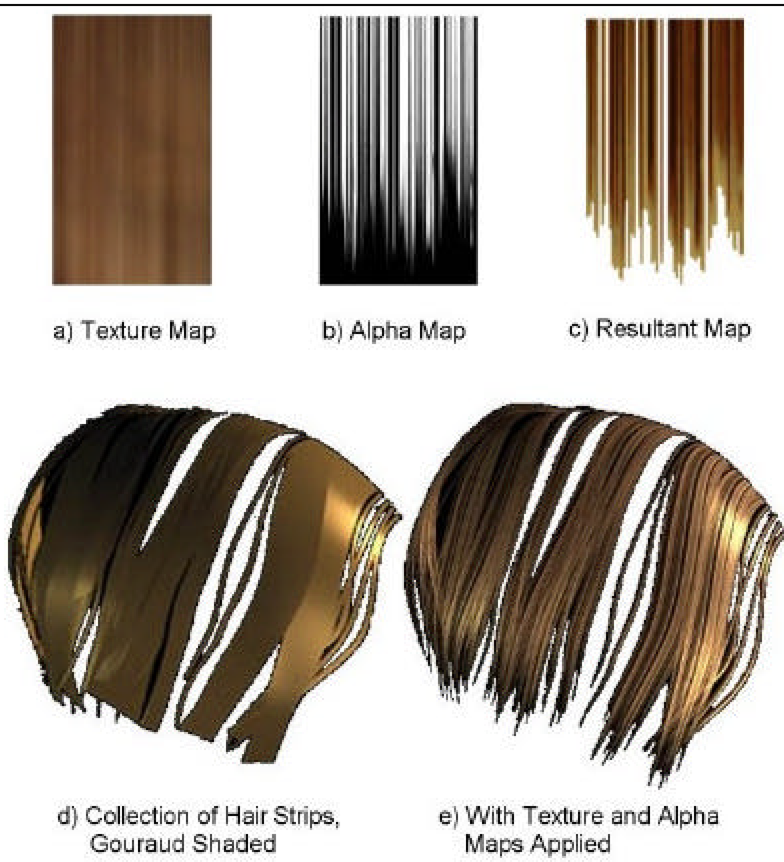

Fig 1. The Strop-Based Hair Model

4. Our Work

In this section, we start to describe our work on hair animation. First we overview the animation process using a pseudo code that indicating the equations to be used. Then, we describe the simple Physics model using equations. Finally, we discuss the collision detection in the animation.

4.1 Overview of the Animation Framework

The pseudo code of our animation process is listed as follows:

Wait for Event to trigger (Wind, Head motion etc)	
At each time step dt	
Check LOD, perform tessellation to desired detail (Appendix A)	
For each hair strand	
For each control point Pi, from top to bottom	
Compute force F_1 due to head movement and gravity	
Compute force F_2 due to wind	
Compute force F_3 due to springs	(11)
Compute $F_{total} = F_1 + F_2 + F_3$	(6)
Compute $F_{total} = F_{total\ \theta} + F_{total\ \phi}$	(5)
Compute $M_{external\ \theta}, M_{external\phi}$	(4)
Compute $M_{spring\theta}, M_{spring\phi}$	(3)
Compute M_{θ}, M_{ϕ}	(2)
Compute new position	(7)
Collision Detection using ellipsoid	(Appendix)
Collision Response if collided	(10)

The details of equations and symbols used are described in the following subsections.

4.2 The Physics Model for the Strip-Based Hair Model

In this section, we describe a simple dynamic model adapted and applied to the control point meshes. A set of dynamics equations are defined and solved for the control points. By working on the control points rather than the mesh points after the tessellation, we can cut the computational cost of the simulation by an order of magnitude. Of course, by working with an approximate model, we are trading accuracy for speed [Fig 2].

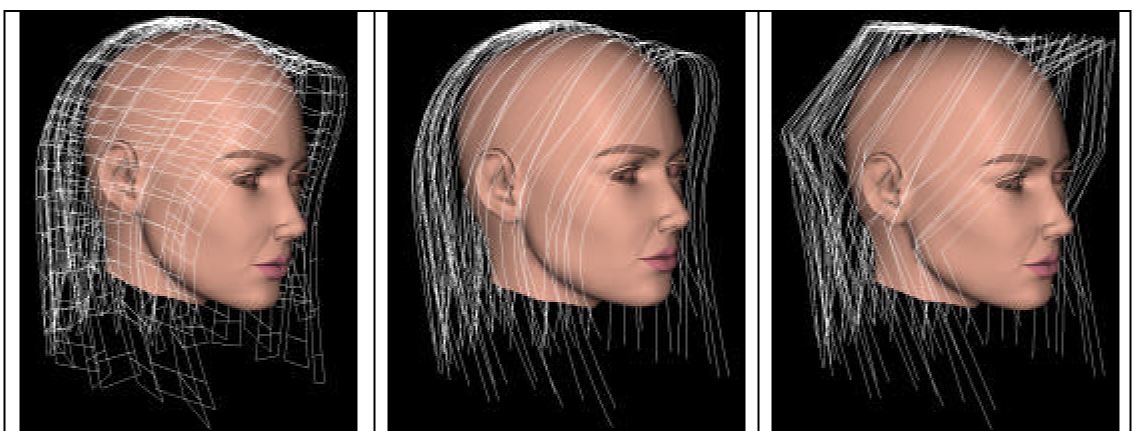

Fig 2a.	**Fig 2b.**	**Fig 2c.**
Hairstyle using hair strips	Only consider vertical rows	Only consider control pts.

The simple Physics model is similar to the model proposed by Anjyo [Anjy92] and later extended by Kurihara [Kuri93]. We brief the major equations by using the illustration of Fig 3.

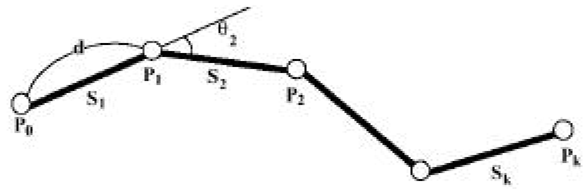

Fig 3. Illustration of a Single Strand of Hair

Consider a hair strand modeled by a series of connected line segments. Taking the polar coordinate system as shown in Fig 4, the variables $q_i(t)$ and $f_i(t)$ with time parameter t are governed by the ordinary differential equations:

$$I_i \frac{d^2\theta_i}{dt^2} + \gamma_i \frac{d\theta_i}{dt} = M_\theta,$$

$$I_i \frac{d^2\phi_i}{dt^2} + \gamma_i \frac{d\phi_i}{dt} = M_\phi, \qquad (1)$$

where I_i is the moment of inertia of the segment s_i, γ_i is the damping coefficient, M_θ and M_ϕ are the torques according to θ and ϕ component respectively.

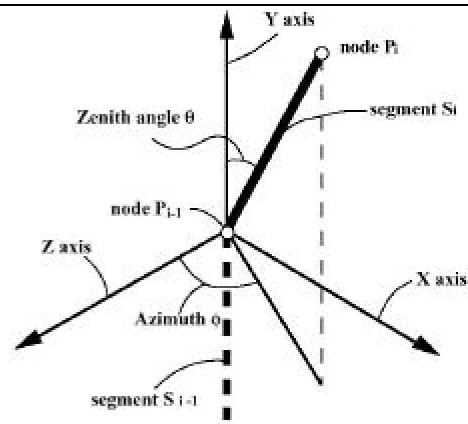

Fig 4. Polar co-ordinate system for a hair segment

The torque M_θ and M_ϕ applied to the segment s_i are derived from the hinge effect $M_{\theta\,spring}$, $M_{\phi\,spring}$ between two segments, and external moment $M_{\theta\,external}$, $M_{\phi\,external}$ from external force, such as gravity, inertial force and wind:

$$M_\theta = M_{\theta\,spring} + M_{\theta\,external},$$
$$M_\phi = M_{\phi\,spring} + M_{\phi\,external}, \qquad (2)$$

$M_\theta$spring and $M_\phi$spring are defined as

$$M_{\theta\,spring} = -k\theta(\theta - \theta_0),$$
$$M_{\phi\,spring} = -k\phi(\phi - \phi_0), \qquad (3)$$

$$M_{external\theta} = u\,F_\theta,$$
$$M_{external\phi} = v\,F_\phi, \qquad (4)$$

where u is (1/2)d, v is the half length of the segment that is the projection of s_i onto the ϕ plane. F_θ, F_ϕ are the "θ, ϕ -component" of force **F** respectively.

$$F_{total} = F_{total\,\theta} + F_{total\,\phi} \quad , \qquad (5)$$

The above θ component F_θ of the applied force **F** is the scalar value defined by $F\theta = (\mathbf{F}, \mathbf{V}_\theta)$, where $\mathbf{V}\theta$ is the unit vector on the θ plane that is perpendicular to the segment s_i.

Similarly, the ϕ component F_ϕ is defined by $F_\phi = (\mathbf{F}, \mathbf{V}_\phi)$, where $\mathbf{V}_\phi$ is the unit vector on the ϕplane that is perpendicular to the projected segment of s_i onto the ϕ plane.

The external force **F** is defined as

$$F = \rho\, d\,(\mathbf{g} + \mathbf{a}) + d\,\mathbf{f} \ , \qquad (6)$$

where **g** is the acceleration due to gravity, **a** is the acceleration due to the movement of the head itself, and **f** is the density of the applied force, such as wind.

In the numerical simulation, equation (1) is discretized as

$$\theta_i^{n+1} - 2\theta_i^{n\text{-}1} + \gamma_i \Delta t\,(\theta_i^{n} - \theta_i^{n\text{-}1}) = (\Delta t)^2 M_\theta \,,$$
$$\phi_i^{n+1} - 2\phi_i^{n\text{-}1} + \gamma_i \Delta t\,(\phi_i^{n} - \phi_i^{n\text{-}1}) = (\Delta t)^2 M_\phi \,, \qquad (7)$$

The calculation starts with the segment s_1, and the new angle of s_i is successively determined using (7).

4.3 Collision Detection and Avoidance

In order to properly achieve hair animation, the model must include a collision detection and treatment process. Our model takes into account the two types of collisions involved:

a) Collision avoidance between hair strips and head

The goal is to have a quick and robust method to avoid hair strip penetration inside the head object.

Ellipsoids are used to approximate the bounds for the head object. Using this method we can quickly test if the animation path of the hair strip collides with the bounds [see Appendix B].

When point **P** is inside of the ellipsoid, let the point **T** on the surface of the ellipsoid be the nearest point from **P**. If a hair strand is inside of the body, the reaction constraint method [6] is applied to keep hair outside of the body. Let $\mathbf{F}_{\mathbf{input}}$ be the applied force to node point P. Then unconstrained component of **F** input is

$$F_{unconstrained} = F\ input - (F\ input \bullet N)\ N, \qquad (8)$$

where N is the normal vector at point T. The constrained force to avoid the collision is

$$F_{constrained} = -(k\ PT + c\ V \bullet N)\ N, \qquad (9)$$

where V is the velocity of point P, T is the nearest point on the surface from point P, *k* is the strength of the constraint and *c* is the damping coefficient.

The output force which is applied to point P is a summation of F unconstrained and F constrained.

$$F_{output} = F_{input} - (F_{input} \bullet N)\ N - (k\ PT + c\ V \bullet N)\ N, \qquad (10)$$

b) Collision avoidance between hair strips

There are 2 main reasons for this. The first is to give volume to the hair. Second, intersecting, over-compressed or over-stretched hair strips may produce visual artifacts. This is done by introducing springs within each hair strip and also between neighboring hair strips.

For each connected spring z,

$$F_{spring} \mathrel{+}= -k_z * x_z , \qquad (11)$$

where k is the spring constant (controls stiffness), and x is the distance stretched from initial rest length. Thus, a repulsion force from the spring is introduced when two strips are closer than a proximity factor. That factor can be adjusted depending on hair density and shape. Similarly, an attractive spring force prevents neighboring hair strips from moving too far away from each other.

5. Results

Our implementation uses Java3d version 1.2.1 on Windows 2000 for software, on a AMD 900 Duron processor with 256MB RAM and a Riva TNT graphics chipset for hardware.

Some animation results are rendered and shown in Fig.5 in snapshots. Practically, we have achieved interactive hair animation with collision detection and good visual quality. This is possible because of results can be rendered with acceleration any low-end 3D graphics card. The latest family of consumer 3D graphics card can even perform keyframe interpolation in hardware, thus there is still potential for more speed gain.

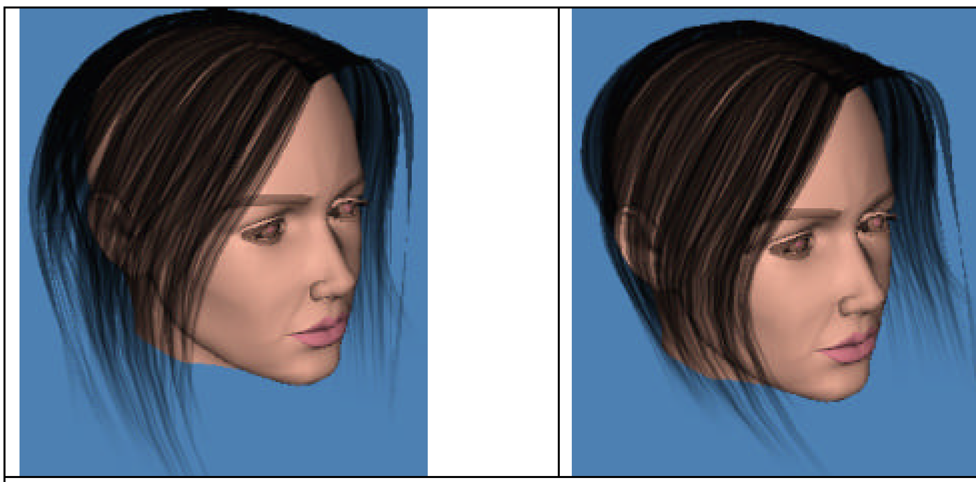

Fig5a. Wind blowing sideways

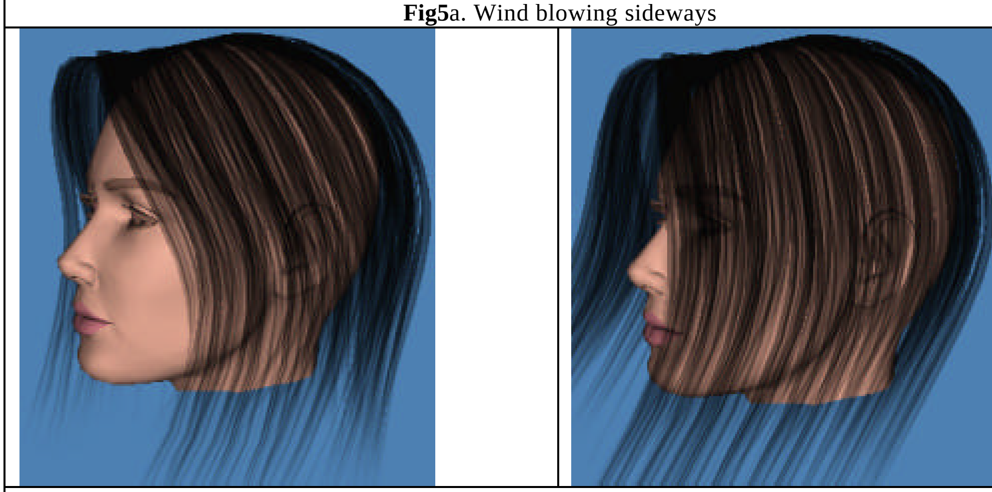

Fig5b. Wind blowing from behind

6. Conclusion

We have proposed and implemented a simple Physics model applied on the control points of the 2D strip-based hair model. Because of the simplicity of the model and much smaller number of control points involved in the simulation, we have achieved the real time animation with naturally-looking and Physically-plausible animation results.

We are still working on increasing the performance by introducing flocking behavior to the animation of hair strips, thus cutting down simulation time needed.

7. Acknowledgement

The first author was under the scholarship of PSA Corporation, Singapore. This work was partly supported by the Academic Research Grant (RP3982704) of National University of Singapore.

Reference

[Anjy92] K Anjyo, Y. Usami, and T. Kurihura. A Simple Method For Extracting The Natural Beauty Of Hair, SIGGRAPH (92), pp. 111-120.

[Böhm80] W. Bohm. Insert New Knots into B-spline Curves, Journal of Computer Aided Design, 12 (4), pp. 199-201, 1980.

[Chen99] L. H. Chen, S. Saeyor, H. Dohi, and M. Ishizuka. A System of 3D Hair Style Synthesis Based on the Wisp Model, The Visual Computer, 15 (4), pp. 159-170 (1999).

[Cohe80] E. Cohen, T. Lyche, and R. Risenfeld. Discrete B-Splines and Subdivision Technique in Computer-Aided Geometric Design and Computer Graphics, CGIP, 14 (2), 1980, pp. 87-111.

[Dald93] A. Daldegan, T. Kurihara, N. Magnenat Thalmann, and D. Thalmann. An Integrated System for Modeling, Animating and Rendering Hair, Proc. Eurographics (93), Computer Graphics Forum, Vol.12, No3, pp.211-221.

[Koh00] C K Koh, Z Y Huang, Real-time Animation of Human Hair Modeled in Strips, In: Computer Animation and Simulation, Springer-Verlag 2000 pp.101-110.

[Kong99] W. Kong and M. Nakajima. Visible Volume Buffer for Efficient Hair Expression and Shadow Generation, Computer Animation (99), (May 1999, Geneva, Switzerland). IEEE Computer Society.

[Kuri93] T. Kurihara, K. Anjyo, and D. Thalmann. Hair Animation with Collision Detection, Models and Techniques in Computer Animation (93), Springer-Verlag, Tokyo, pp. 128-138.

[Meye91] A. Meyer. A Linear Time Oslo Algorithm, TOG (10), 1991, pp. 312-318.

[Neyr98] F. Neyret. Modeling, Animating, and Rendering Complex Scenes Using Volumetric Textures, IEEE Transactions on Visualization and Computer Graphics, 4(1), pp. 55-70 (January-March 1998).

[Plat88] J. C. Platt and A. H. Barr. Constraint Methods for Flexible Models, SIGGRAPH (88), pp. 279-288.

[Rose91] R. E. Rosenblum, W E. Carlson, and I. E. Tripp. Simulating the Structure and Dynamics of Human Hair: Modeling, Rendering and Animation. The Journal of Visualization and Computer Animation, 2 (4), 1991, pp. 141-148.

[Tann98] C. C. Tanner, C. J. Migdal, and M. T. Jones. The Clipmap: A Virtual Mipmap, SIGGRAPH (98), pp. 151-158.

[Terz88] Terzopoulos D and Fleischer K, Deformable Models, The Visual Computer 1988, Vol 4. No. 6 pp. 306-33

[Thal93] N. Magnenat Thalmann and A. Daldegan. Creating Virtual Fur and Hair Styles for Synthetic Actors. In Communicating with Virtual Worlds, Springer-Verlag, Tokyo, 1993, pp. 358-370.

[Wata92] Y. Watanabe and Y. Suenaga. A Trigonal Prism-Based Method For Hair Image Generation, CGA (12), No. 1, 1992, pp. 47-53.

Appendix A: Outline of Surface Tessellation By Forward Differencing

The method of finding a point on a NURBS surface, called the de Boor algorithm, can easily become a bottleneck in a graphics program. By restricting ourselves to using only uniform B-Spline surfaces means we can easily use a numerical technique called Forward Differencing.

Given an equation for a curve, find a starting point on it and then follow its direction to find each subsequent point. This approach is *much* faster than the de Boor algorithm. First we have to convert the control vertices to standard equations (power basis) and set up the starting points.

Fig 6 shows the geometrical meaning of the power basis form of a B-Spline curve segment. The coefficients a, b, c, and d ``tell'' the curve to start at d, go in the c direction, but curve toward b, and change the curvature at a rate of a. With a *Uniform* B-Spline, the conversion to power basis can be read from the geometry. This is the same as the natural Bezier conversion.

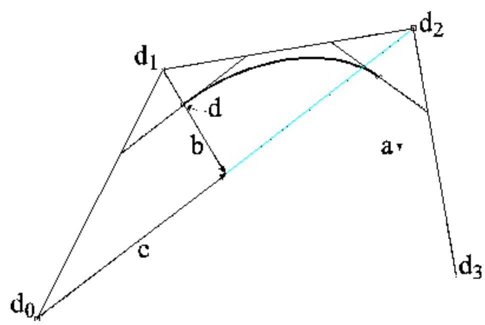

Fig 6. The power basis form of a cubic curve ($\mathbf{a}*t^3 + \mathbf{b}*t^2 + \mathbf{c}*t + \mathbf{d}$) has an interesting relationship to the geometry of the UBS CVs. For instance, **b** is the vector from d1 to half way between d0 and d2.

The Forward Differencing approach to evaluating a curve is, basically, to first find the starting point and then follow the direction.

Initialize Forward Differences
$f(t) = at^3 + bt^2 + ct + d$, $\quad$ $s = 1/(n-1)$
$f = d,\ l = 0$ $\quad$ s = 1st point on curve, l = length so far
$\Delta f = f(s)-f(0) = as^3 + bs^2 + cs$
$\Delta^2 f = \Delta f(s) - \Delta f(0) = 6as^3 + 2bs^2$
$\Delta^3 f = \Delta^2 f(s) - \Delta^2 f(0) = 6as^3$

Iterate, following the curve
for i = 1 to n
 if (l>lmax) exit
 $f = f + \Delta f$
 $l = l + |\Delta f|$
 $\Delta f = \Delta f + \Delta^2 f$
 $\Delta^2 f = \Delta^2 f + \Delta^3 f$

We stop when the accumulated length has exceeded the the length of that particular hair strand.

The Forward Differences are set up first in the u direction, and then curves are followed in the v direction. Doing the v direction last means that points on the surface are indexed in the same order as the CVs.

Appendix B

The routine below returns the ray/sphere intersection. The inputs are the ray's origin and normalized direction vector, as well as the sphere's origin and radius.

```
double intersectSphere ( Point rO, Vector rV, Point sO, double sR )
{
    Vector Q = sO - rO;
    double c = length of Q;
    double v = Q * rV;
    double d = sR*sR - (c*c – v*v);
    if (d < 0.0) return -1.0;          // no intersection
    return v - sqrt(d);                // distance to 1st intersecting pt
}
```

For an ellipsoid, we perform a stretch in 2 axes before performing the same test above.