Modelling Hair Collisions
Using Spline-Based Cylinder Chains

Stuart Golodetz
Oxford University Computing Laboratory
stuart.golodetz@comlab.ox.ac.uk

January 12, 2007

Abstract

This paper discusses a method of modelling collisions between long hairs. The hairs are modelled as cubic B-splines, to which chains of cylinders are fitted for the purposes of collision detection. The head on which the hairs rest is modelled as an ellipsoid. The paper explains how to fit cylinder chains to splines, how to test whether and where two cylinder chains intersect, and how to test whether and where a cylinder chain intersects an ellipsoid. It further shows how to make use of these results to model (a necessarily limited amount of) long hair on a human head.

1 Introduction

The modelling of hairs and the collisions between them has long been a difficult computational problem. Whilst remarkable progress has been made in modelling short hair and fur, the modelling of longer hair, in particular, remains a research problem. When one considers that the average human has well over 100,000 hairs, it is easy to see why dealing with the problem is so hard.

This paper should not, and will not, pretend to provide any groundbreaking solutions to the difficulties faced. Its purpose is merely to present my research in this area, in the hope that it may help someone to have more penetrating insights into the problem.

2 Related Work

In the process of deciding how to approach the problem, I read a number of research papers on hair modelling. Many of them were too advanced for my current state of knowledge, but I did derive a certain amount of inspiration from [2].

Although my approach ultimately bears very little similarity to that in their paper, I am nonetheless indebted to the authors for providing some food for thought.

3 Initial Insights

A first step in any sort of modelling is to decide on the desired representations of the entities involved. Hairs, being curvy by nature,[1] can be represented reasonably well by cubic B-splines [3]. This representation suffices for basic hair rendering, but is far from ideal for collision detection. Collision testing using the underlying spline representations of the hairs would be difficult and time-consuming.

A better approach is to find some bounding approximations for our hairs which can be tested more efficiently. Since hairs are long and thin, it makes sense to represent them with chains of bounding cylinders (see Figure ?). Testing cylinders against each other is a relatively efficient geometric operation, as we will see in §5.1.

We also need a representation for the head on which our hairs will rest. Heads tend to be relatively ellipsoidal, so we represent them as such. Testing cylinders against ellipsoids is fairly straightforward and efficient (see §6.1), so this representation is compatible with our representation for hairs.

[1] Even so-called ‘straight’ hair curves over the head to some extent.

Even with reasonable representations for the hairs and head, the number of hairs involved in serious hair modelling means that a naive approach will result in a painfully slow program. I have not attempted to deal with the problem in this paper, but I hope that a future hierarchical approach will yield better results (see §8).

4 Fitting Cylinder Chains to Splines

TODO

5 Inter-Chain Intersections

This section describes how to determine whether and where two chains of cylinders intersect. The method described will involve testing pairs of cylinders from the two chains against one another, so a way of doing so is described first.

5.1 Inter-Cylinder Intersections

Consider two arbitrary cylinders of finite height. As Figure ? shows, they intersect only if the shortest distance between their axes (each of which is a finite line segment) is less than the sum of their radii. Furthermore, if they do intersect, then knowing the endpoints of the unique shortest line segment between the axes (if any - the axes may be parallel and overlapping) will be helpful.

Accordingly, we develop a procedure to find the unique shortest line segment between the axes (if any), or the shortest distance between them (otherwise). This is too short a paper to provide a detailed derivation of the algorithm, but a good reference is available on the web [1].

Given such a procedure, it is trivial to determine whether or not two cylinders intersect by comparing the distance returned to the sum of the radii of the cylinders.

5.2 Extension to Chains

We can extend this to cylinder chains in the obvious way by testing the cylinders of one chain against those of the other in a pairwise fashion. Figure ? illustrates the process.

5.3 Complexity Analysis

Suppose one cylinder chain contains m cylinders and the other n cylinders. Then the complexity of the chain intersection algorithm is $O(mn)$, since we have to test all possible cylinder pairs.

This is potentially problematic. On average, we would expect that $m \in \theta(n)$, making the method quadratic (in either m or n). Given a head with h hairs, and bearing in mind that each of the $\binom{h}{2}$ pairs of hairs need testing against each other, we can see that our method would have a complexity of $O(h^2n^2)$, which is far from ideal. On reflection, however, we observe that since a cylinder chain is merely an *approximation* to the spline representation of a hair, its length (e.g. n) can be bounded above without any ill-effect. If we ensure that the length of a chain is no more than some constant, then our overall complexity becomes $O(h^2)$, albeit with a large constant factor. This is manageable for reasonable values of h, though probably not when dealing with the full head, when $h > 100000$.

6 Chain-Ellipsoid Intersections

This section describes how to determine whether and where a chain of cylinders intersects an ellipsoid. The method described involves testing each cylinder in the chain against the ellipsoid, so a way of doing so is described first.

6.1 Cylinder-Ellipsoid Intersections

Consider an arbitrary cylinder of finite height, represented by its radius r and the line segment $(\mathbf{e_1}, \mathbf{e_2})$ defining its axis, and an arbitrary ellipsoid, represented by an orthogonal Cartesian coordinate system with its origin at the ellipsoid's centre (see Figure 6.1).

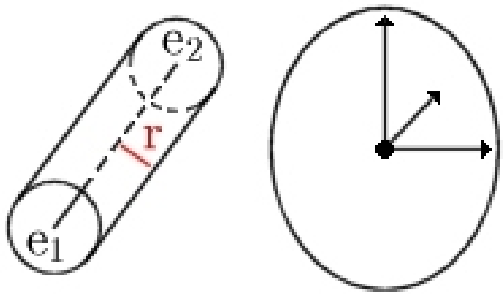

Figure 6.1: The representation of cylinders and ellipsoids

Testing the cylinder against the ellipsoid is equivalent to testing its axis against a version of the ellipsoid expanded by r in all directions (see Figure ?)[2].

Furthermore, we can do this latter test in the (expanded) ellipsoid's local coordinate system. In that system, the ellipsoid is a unit sphere, centred at the origin, so we thereby reduce our test to that of a line segment against a sphere.

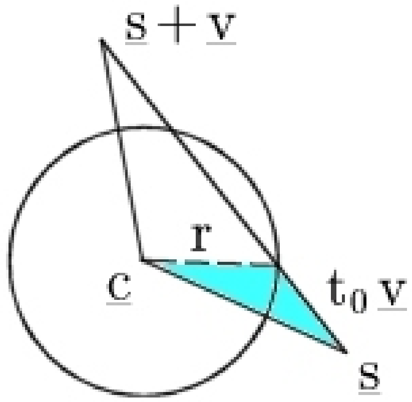

Figure 6.2: Determining where the line with equation $\ell(t) = \mathbf{s} + t\mathbf{v}$ intersects the sphere centred at $\mathbf{c}$ with radius r

Consider Figure 6.2. To determine where the line $\ell(t) = \mathbf{s} + t\mathbf{v}$ intersects the sphere with centre $\mathbf{c}$ and radius r, we use the cosine rule to derive that

$$r^2 = |\mathbf{c} - \mathbf{s}|^2 + |t_0\mathbf{v}|^2 - 2(\mathbf{c} - \mathbf{s}) \cdot (t_0\mathbf{v})$$

Rearranging gives us the simple quadratic

$$t_0^2\left(|\mathbf{v}|^2\right) + t_0(-2(\mathbf{c} - \mathbf{s}) \cdot \mathbf{v}) + \left(|\mathbf{c} - \mathbf{s}|^2 - r^2\right) = 0$$

which we can solve to find t_0 and hence the intersection point(s). If there are two intersection points, the answer we want to return is the line segment joining them.

To do the same thing for the line segment $(\mathbf{e_1}, \mathbf{e_2})$, we simply let $\mathbf{s} = \mathbf{e_1}$ and $\mathbf{v} = \mathbf{e_2} - \mathbf{e_1}$. We then note that the line segment can be represented by an interval on the infinite line, namely $[0, 1]$, since $\ell(0) = \mathbf{e_1}$ and $\ell(1) = \mathbf{e_2}$. Similarly, the line segment joining the two intersection points (which also lies on ℓ) can be represented by the interval $[t_1, t_2]$, say, where t_1 and t_2 are the values of t at the points where the infinite line intersects the sphere and $t_1 \leq t_2$. The answer we want to return is then the line segment represented by the intersection of these two intervals, namely $[\max(0, t_1), \min(1, t_2)]$. Figure ? illustrates the intention.

6.2 Extension to Chains

The extension of the method to a chain of cylinders is trivial: we simply test the ellipsoid against each cylinder in the chain in turn.

6.3 Complexity Analysis

Suppose the cylinder chain contains n cylinders. Then the complexity of the chain intersection algorithm is simply $O(n)$, since we just have to test each cylinder against the ellipsoid.

7 Modelling the Hair

TODO

8 Conclusions and Further Work

TODO

9 Acknowledgements

I would like to thank both Stephen Cameron and Irina Voiculescu, my DPhil supervisors at Oxford, for helping me with this project.

Any mistakes/catastrophic blunders which may subsequently be found in this paper are of course my own!

[2]This is a configuration-space approach to the problem.

References

[1] Dan Sunday, *Distance between Lines and Segments with their Closest Point of Approach*, http://www.geometryalgorithms.com/Archive/algorithm_0106/algorithm_0106.htm.

[2] Nadia Magnenat Thalmann, Stephanie Carion, Martin Courchesne, Pascal Volino, Yin Wu, *Virtual Clothes, Hair and Skin for Beautiful Top Models*, MIRALab, University of Geneva, 1998.

[3] Irina Voiculescu, *Splines Sets I-VIII*, http://web.comlab.ox.ac.uk/internal/courses/materials04-05/splines/lectures.html. Oxford University Computing Laboratory, *Splines and Computational Geometry* course, 2005.