# Implementation of Artistic Curly Hair Dynamics in Houdini

Yasser Mohsen
MSc Computer Animation and Visual Effects

August 22, 2016

**Abstract**

This is an implementation of artistic curly hair dynamics according to the method introduced by Pixar to achieve the artistic hair introduced in the movie Brave. This implementation is done using Houidini, to give Houdini users the ability to apply these dynamics to their hair easily. The implementation is divided into two major sections, the dynamics of a single strain of hair, and optimizing the collision between different hair strains. For the single strain, a combination of three strings is introduced, Stretching string, Bending string, and Core string. For collisions, the operation is optimized by pruning the close particles in the one strain to reduce the amount of particles processed for a single strain from a side, and then from the other side, neighboring hair strains are pruned when possible to reduce the amount of strains to process.

# Contents

# 1 Introduction

Not only is Hair simulation a crucial part of modern creature design and production, but also a challenge in every production environment. Divided into styling, dynamics, and rendering, the production of hair present challenges for different departments of production environment. In this paper, we focus on challenges considering hair dynamics, challenges range from having a stable solution, having it balanced between being realistic and also fulfill the artistic needs, to this solution to be fast and efficient for animation needs, and also being reusable for different types of hair. Different solutions have been introduced using different perspectives. In the survey introduced by [17], different representations of hair are discussed. Some methods introduce hair using geometric representation like NURBS surface models in [13]or cylinders that represent wasps of hair in [5], others introduced the concept of using images or sketches of hair to produce and animate computer generated hair such as [18] and cartoon hair like in [9], and others, which are focused on in this paper, deal with hair as collection of dynamic strains that follow physical dynamics.

This paper introduces an implementation of curly hair dynamics using the method introduced in [8]. The method is divided into mainly two parts, the dynamics of one strain of hair, and the contact between strains of hair, and the optimization of collision detection and handling.

# 2 Previous Work

Methods have been introduced to simulate the dynamics of hair as a collection of single strains as mentioned earlier. In [16], Selle et al., building on Rosenblum et al.'s method discussed in [15] in 1991, introduced a method using a four mass spring structure that using the support of the four mass spring together to hold the curliness of a curly hair, and also keeps the structure of a straight hair. Another method as introduced in [2, 1] the concept of elastic rods is used to represent hair as a one dimensional object that it's centerline, along with its material frame, controls the shape of the hair strain at every point of time. In [7], hair model is presented in a series of rigid bodies connected by a three DOF spherical joint. Forward angular dynamics on the spherical joints control the motion of the links between these joints. These serial rigid multibody systems are combined together to create a fluid like object that follows continuum dynamics rules, where SPH method was used (SPH discussed in [10]). Another method introduced by [11], use position based dynamics through modeling hair as a chain of particles in what's called Follow The Leader method (FTL). In this method, position based dynamics approach is used based on finding the right position for a particle to fulfill the distant constraint enforced between the particles.

# 3 Artistic Simulation of Curly Hair

The method implemented in this paper was developed mainly to simulate curly hair. However, it was proved to be successful in simulating straight hair as well. Also, it's worth noticing that although this method doesn't specifically introduce an ultimately realistic hair, it gives enough control need by artists to give the desired look.

## 3.1 Single Hair Strain Dynamics

This method, introduced in [8], combines the dynamics theory of both the mass springs method introduced in [16] and the elastic rods method introduced in [1] to maintain the hair shape and simulate its motion. This combination is used to apply three main internal forces on a series of particles connected by springs that enforce these forces as follows

### 3.1.1 Stretch Spring Force

Similar to the mass spring method of Selle et al. in [16], particles are connected by linear springs. For every single strain consists of N particles, there is a set of current particle positions P = $\{p_0, p_1, ...., p_{N-1}\}$ , a set of current particle velocities V = $\{v_0, v_1, ...., v_{N-1}\}$ , and every two particles are connected by an edge $e_i = p_{i+1} - p_i$. The force applied by this spring is a standard damped linear spring force calculated as follows:

$$f_s(k_s, c_s)_i = k_s(||e_i|| - ||\bar{e}_i||)\hat{e}_i + c_s(\Delta v_i.\hat{e}_i)\hat{e}_i \quad (1)$$

where $k_s$ and $c_s$are spring and damping coefficients respectively, $\triangle v_i = v_{i+1} - v_i$ , $||.||$ denotes vector length, and $\hat{}$ denotes vector normalization. Although this equation represents both spring forces and damping forces upon the spring, these two forces are divided into two separate processes during the implementation as discussed in the implementation section. Iben et al. also suggests using a biased strain limiting approach, as discussed in [16] to control the amount of stretch applied to strains as desired by artists, by looping through the particles and limit the velocity of particles and length of stretching edges that exceed a curtain limit.

### 3.1.2 Bending Spring Force

Similar to the Elastic Rods method in [1], a curve between the particles representing the centerline and material frames per the point are used to compute the bending of the strain. However, as this method would detect every little move, as simple as a walk cycle, parallel transporting along smoothed curve is used to reduce the effect of small movement on the bending of the hair strain.

**3.1.2.1 Smoothing Function** Infinite Impulse Response (IIR) Bessel filter, explained in [12], is used as a smoothing function, $d_i = \varsigma(\Lambda, \alpha)_i$for the curve, where $\Lambda = \{\lambda_0, ..., \lambda_{N-1}\}$ is a set of N elements in $\mathbb{R}^3$associated with hair, like particle positions or particle velocities. Using a smoothing amount $\alpha \geq 0$, $\beta = min(1, 1 - exp(-l/\alpha))$ where $l$ is the average rest length per edge, is used to calculate the vector $d_i$ $\forall_i \in \{0...N-2\}$ that is used to achieve the smoothed curve. $d_i$is calculated as follows:

$$d_i = 2(1-\beta)d_{i-1} - (1-\beta)^2 d_{i-2} + \beta^2(\lambda_{i+1} - \lambda_i) \tag{2}$$

as noticeable the function is a recursive functions that depend on the basic condition $d_{-2} = d_{-1} = \lambda_1 - \lambda_0$ giving $d_0 = \lambda_1 - \lambda_0$. In case of positions, as illustrated in figure (1), this vector $d_i$ is to be recursively added to the position from the root to the tip to reach the smoothed curve as follows:

$$p_i^{'} = p_{i-1}^{'} + d_{i-1}$$

where $p_0^{'} = \lambda_0$, the root of the polyline.

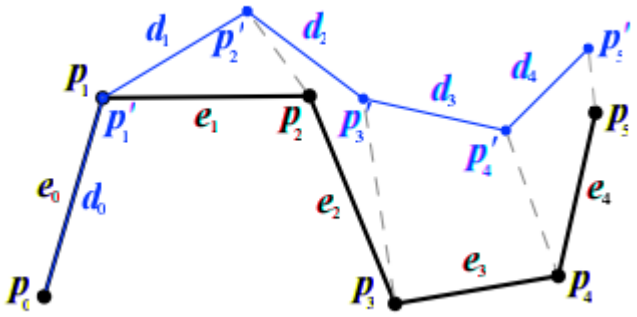


Figure 1: Smoothing Function, Original Curve in bold black line, smoothed curve in thin blue line. [8]

**3.1.2.2 Parallel Transport** The idea of parallel transport, as explained in [6] and [3], is to rotate a frame from the initial frame, root frame in our case, to maintain the parallelism of the frame to the curve. Given a curve C, an existing Frame F1 at point t-1, a tangent T1 at point t-1, and tangent T2 at t, the new frame F2's orientation can be calculated by rotating F1 about axis A with angle $\alpha$where A = T1 $\times$T2 and $\alpha = ArcCos((T1.T2)/(|T1||T2|))$. So, in other words, at each point, given the tangent of the point, the frame from the previous point is rotated so that the tangent from previous point matches the tangent on the current point, as illustrated in Figure2

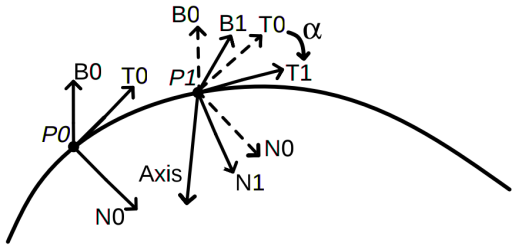


Figure 2: Computing a reference frame from the previous frame. [3]

**3.1.2.3 Applying Bending Forces** Given the new smoothed curve, the material frames are calculated per point using the parallel transport of the root frame. For initialization, a reference vector, $\bar{t}_i = \bar{F}_{i-1}^T \bar{e}_i$, is computed from the rest pose of the curve. to compute the local frame $\bar{F}_{i-1}$ we start by calculating $\bar{F}_0$from the root scalp polygon, afterwards, the frames in sequence are to be calculated by parallel transport of the previous frame along the smoothed curve as discussed previously. For each step, we calculate $F_0$from the root polygon, and $F_{i-1}$using parallel transport on the smoothed curve to get a target vector, $t_i = F_{i-1}\bar{t}_i$, that the current pose is desired to match, refer to Figure 3.

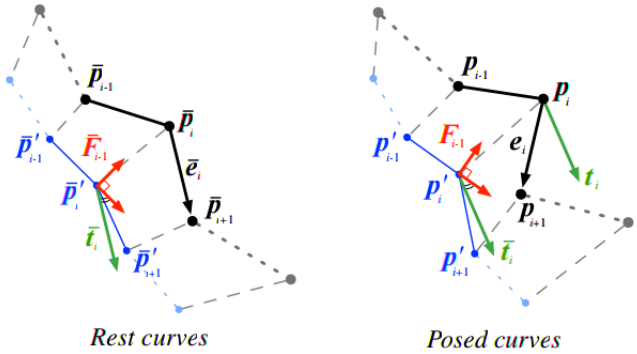


Figure 3: Rest to Posed curve, Original curve in bold black, smoothed curves in blue. [3]

Having the target vector identified the force needed to move the a point to the target position the bending force is calculated as follows:

$$f_b(k_b, c_b)_i = k_b(e_i - t_i) + c_b(\Delta v_i - (\Delta v_i.\hat{e}_i)\hat{e}_i) \tag{3}$$

where $k_b$and $c_b$are spring and damping coefficients respectively, also Iben et al. suggests giving the artists the ability to specify the coefficients for flexibility.

### 3.1.3 Core Spring Force

Although using only stretch spring and bending spring gives a working realistic hair, an undesired effect happens in extreme acceleration of characters. In high acceleration, the curls of the hair unwind and the hair becomes straight instead. To over come this undesired behavior, an extra spring is added to control the longitudinal stretch of the curls. (Another option would have been increasing the stiffness of the spring, but that would affect the flexibility of the bending of the curls). To compute the core spring forces, same smoothing described in section 3.1.2.1 is used to compute $b_i = \varsigma(P, \alpha_c)_i$, the same method is also used to calculate $\nu_i = \varsigma(V, \alpha_c)_i$ for the spring damping term, and the force is calculated as follows:

$$f_c(k_c, c_c)_i = k_c(||b_i|| - ||\bar{b}_i||)\hat{b}_i + c_c(\nu_i \cdot \hat{b}_i)\hat{b}_i \tag{4}$$

where $k_c$and $c_c$are spring and damping coefficients respectively. To maintain stability, $k_c$ must be kept smaller than $k_s$ from equation (1). To control the stiffness of the core spring, to balance between the artist's desire for the hair to bounce in simple motion like walk cycle, and control during extreme motion, Iben et al. suggests using a cubic Hermite blending function based upon the amount of longitudinal curl stretch to non-linearly change $k_c$.

## 3.2 Hair-Hair interactions

Hair and hair contacts is crucial for the simulation of realistic looking hair. Hair hair interaction gives the hair a feeling of mass and interesting interaction the adds to the realism of the hair. However, if every single particle in every single constraint is considered for detecting and calculating collisions and their impact the process would be quite expensive, and time consuming. Hence, a pruning system is introduced that work on two main levels, the level of particles for a single strain of hair, and the level of pairs of hair strains as discussed in the coming sections.

### 3.2.1 Single Hair Pruning

Single hair's particles' pruning is mainly based on having a sphere surrounding each particle, any particle that happens to exist inside this sphere is pruned, starting from the root particle that is never pruned. For a more mathematical explanation, starting from the root particle (j=0), we set k =1 and continue to increment k until the following equation is satisfied:

$$\sum_{i=j}^{k-1} ||\bar{e}_i|| > s(r_j + r_k) \tag{5}$$

where $r_j$is the radius of the $j^{th}$ particle, and s is a parameter controlling the sparsity of particles. When the equation is satisfied, all the particles between j

and k are pruned, j is set to k, k is set to k+1, and the process is repeated over and over again until reaching the end of the hair strain, as shown in Figure 4.

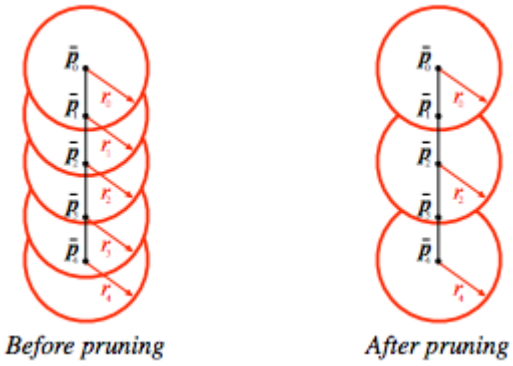


Figure 4: Single Hair Strain Pruning. [8]

### 3.2.2 Hair Pair Pruning

This part of pruning depends mainly on the fact that it's not necessarily needed to consider every two strains of hair for collisions, as other hair strains in the middle will indirectly carry the impact of one hair strain to another. To decide which hair strains effect each other, each $r_i$ is multiplied by constant $r_c$, the sum of all contacts between the spheres of particles of the two hair strains i and j are computed and name $n_{i,j}$. Then, the effect of each two hairs i, j on each others is not considered if they have $n_{i,j} < n_t$, a threshold indicating the minimum number of contacts required to be considered.

### 3.2.3 Collision handling

When particles and hairs that are considered for collision are identified, contacts are detected between two particles whenever their spheres overlap, i.e. $||p_i - p_j|| < r_i + r_j$. To handle this contact, the method used in both [16] and [4] is adapted. For every two particles where a contact is detected, a penalty force spring is applied between these two particles, and breaks when its length reaches a threshold. To avoid instabilities, the spring stiffness is dynamically increased during initial contact while full damping is being used; before the spring breaks, the stiffness is firstly decreased to zero, and then followed by damping.

# 4 Implementation

In this paper, an implementation using Houdini is being discussed to achieve the results desired by the method introduced by Iben et al. in [8]. The implementation is based mainly on the use of Houdini DOPs, VOPs, and VEX. As

this paper is only concerned with the implementation of hair dynamics, the hair tool in Houdini is used for all the other elements. The hair tool in Houdini is divided into skin, guides, dynamics, clumps, and rendering and visualization. This goal of this implementation is add a new dynamics implementation to the built in Houdini hair tool. The addition to the Houdini hair tool is divided into two parts: operation on the rest pose, to make it ready for processing, and the solver that loops and operates on the data represent by the rest pose, and apply different forces to it step after step.

## 4.1 Pre-Solver Calculations on Rest Pose

To qualify the given curve to be a guide for hair using the discussed method, the curve goes through some calculations to set the parameters needed during the calculations of the forces. This calculations are being added to the guides node, as shown in Figure 5. As shown in the figure, in this stage, the curve is being resampled to give smoother calculations between different curvatures. $\bar{e}_i$ and $||\bar{e}_i||$from Equation (1) are then being calculated by the node calculate_eLen. In the same node, guides, the pose curve is smoothed using method described in section (3.1.2.1) using a vex node that loops through the points and recursively compute the smoothed curve, also out of the smoothed curve, $\bar{b}_i$ from equation (4)are extracted. After computing the initial frame using polyframe node, a VOP SOP node is then used to parallelly transport the frame to the points in sequence. Given the new frames, another VOP SOP network is used to calculate the reference vector $\bar{t}_i$. Now, the curve is ready to be used for the calculation of the dynamics of the hair.

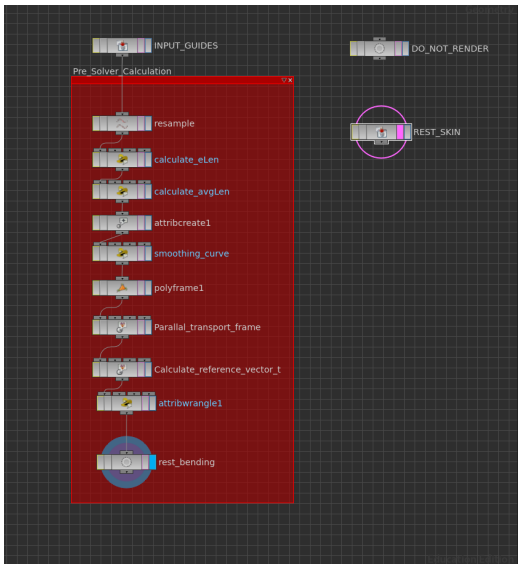


Figure 5: pre-solver calculations on the Rest pose

## 4.2 Solving Dynamics

All dynamic calculations are done in a DOP network that exist in the guide_dynamics node. Running 30 steps frame, as suggested by Iben et al. in [8], the dop network consists mainly of three POP solvers, one POP solver is used to integrate spring forces, with 15 sub-steps per network step, another one is used to integrate damping forces, with 10 sub-steps per spring force step = 150 sub-steps per network step, and the third is only used to calculate the new positions produced by the new velocities coming out of the other two solvers, with one sub-step per network step. This recursion is built based on the algorithm suggested in [8] and shown in Figure 6.

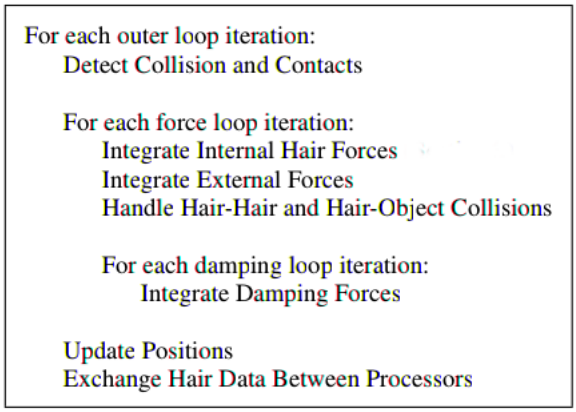


Figure 6: Dynamics Calculation Algorithm. [8]

### 4.2.1 popsolver_External_and_Internal (force loop in Figure (6))

As shown in Figure 7, the internal forces are being integrated by the popsolver_External_and_Internal combined with external forces (i.e. gravity). Internal forces are being divided into three parts, **Stretch Spring:** where the first part of equation (1) regarding the spring force $kforce = k_s(||e_i|| - ||\bar{e}_i||)\hat{e}_i$, **Bending Spring:** the parts where sections (3.1.2.1), and (3.1.2.2) are implemented, and target vector $t_i$ and the $kforce = k_b(e_i - t_i)$ from equation (3) are calculated, **Core Spring**: using $b_i$ extracted from the smoothed curve, in this part the core spring force, the first term in equation (4), is calculated. The three forces are merged, and integrated in the popsolver_External_and_Internal solver producing the new velocities. As discussed earlier, this solver runs 15 sub-steps for every simulation step.

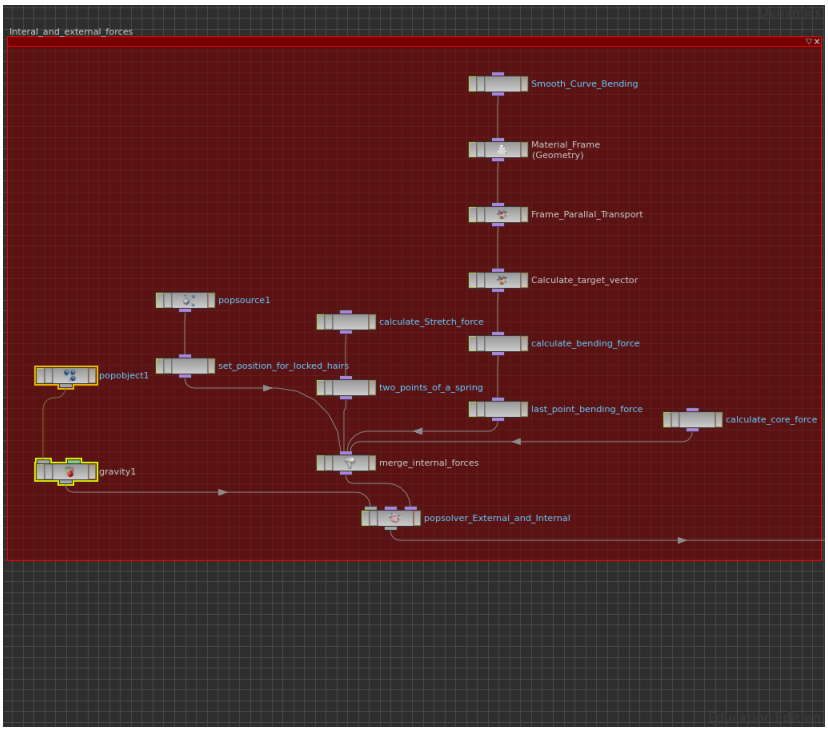


Figure 7: Internal Forces

#### 4.2.2 popsolver_damp (damping loop in Figure (6))

Figure (8) shows the damping forces divided into three parts representing the damping terms in the the three equations (1), (3), and (4). These forces and then merged and integrated by the solver popsolver_damp to produce the new velocities, that gets update every sup-step to help damping the spring forces more efficient. This solver runs 150 sub-steps per simulation step, = 10 sub-step per each spring force sub-step.

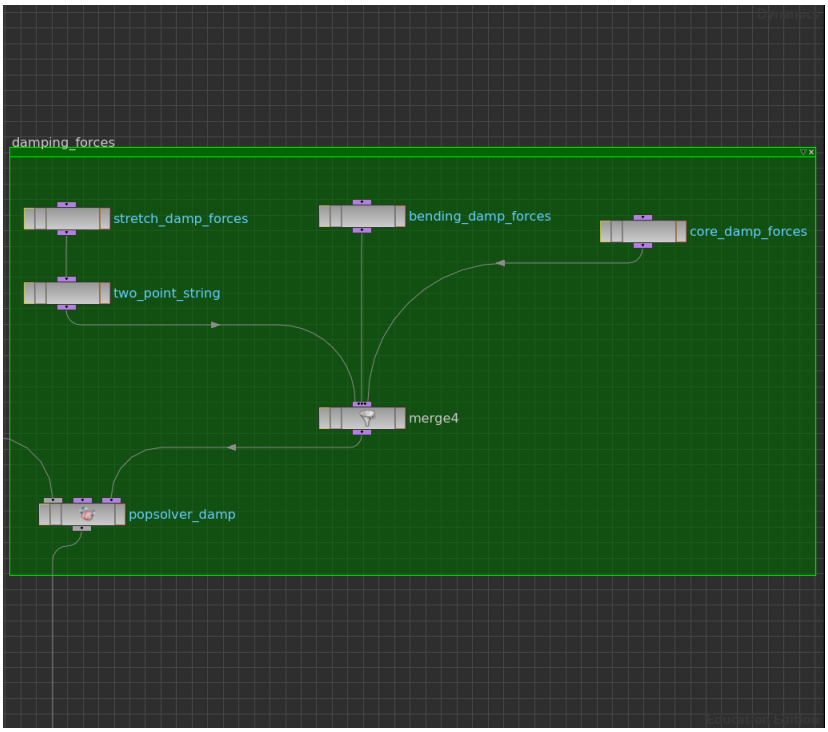


Figure 8: Damping Forces

### 4.3 Handling Collisions

For simplicity, POP impact node was used in this implementation to handle the collisions between the particles. For the ball representing the head, houdini static object was used to handle collision between hair particles and the ball.

## 5 Results and Discussion

### 5.1 Results

The implementation produced a hair dynamics system that is able to handle a curve submitted to the built in hair system, and apply the dynamics method discussed in this paper. Different curliness of hair guides need different parameters to be handled properly. The system gives more flexible dynamics handler than the houdini built in as it gives more access to how the curls are handled, unwound, and put to bounce. Unlike the built in system in Houdini, the implemented system gives strains the ability to bounce in trivial movements, like walk cycle, without having the curl unwound despite the elasticity of the strain. Although, it's more physically realistic to have the curls unwinding in such a bounce, this unwinding is not always desired by the artists, as mentioned in

section (3.1.2). This system also allows to keep the strain flexible and elastic with keeping it's curl and length even under extreme motion of character. Check Figure

## 5.2 Future work

### 5.2.1 Performancestretchiness

The system can still be investigated further to reduce calculations time. The investigation should go through better ways to design the system to run faster, better use of houdini to make the most of its parallel, and multi thread abilities, and also the best sub step handling of the process depending of the type hair simulated. The performance of the system is very critical in the animation environment as animators will often need the hair to be present when animation includes moving the hair or is actually revolved around it.

### 5.2.2 Collision system

Although a simple pop collision handling is used in the implementation, a more efficient collision system is desired. This system should also handle the strain of hair that get stuck together due to collision rather than just bouncing upon each other. Iben et al., in [8], suggests using a penalty spring forces to handle these collisions as mentioned in section (3.2.3). The basic idea of the method is also discussed and explained in [14] for more details.

### 5.2.3 User Interface

So far, this implementation is only an implementation to the method in [8]. However, this implementation needs to be converted to an easily reusable tool. A tool to handle different types of hair, giving the user the ability to change the variables that makes the flexibility offered by this method is available in the user's hands. Similar to the "add dynamics" tool in the Hair houdini shelf, the user should be able to choose between applying the built in houdini dynamics, using wires, and the proposed method. Should the user choose the proposed method, the pre-solution addition in section (4.1) is appended to the guides node, and a dop network containing the solution network explained in section (4.2) is added to the scene.

# 6 Conclusion

Hair simulation is a critical part of every computer animation environment due to it's impact in making characters more appealing and realistic. For more than a quarter of a century, new methods have been emerging to provide the industry with new ways to represent hair. Different needs, different productions, different animation styles, and different characters, all of these are elements that caused the variation of hair simulation methods that were discussed to range

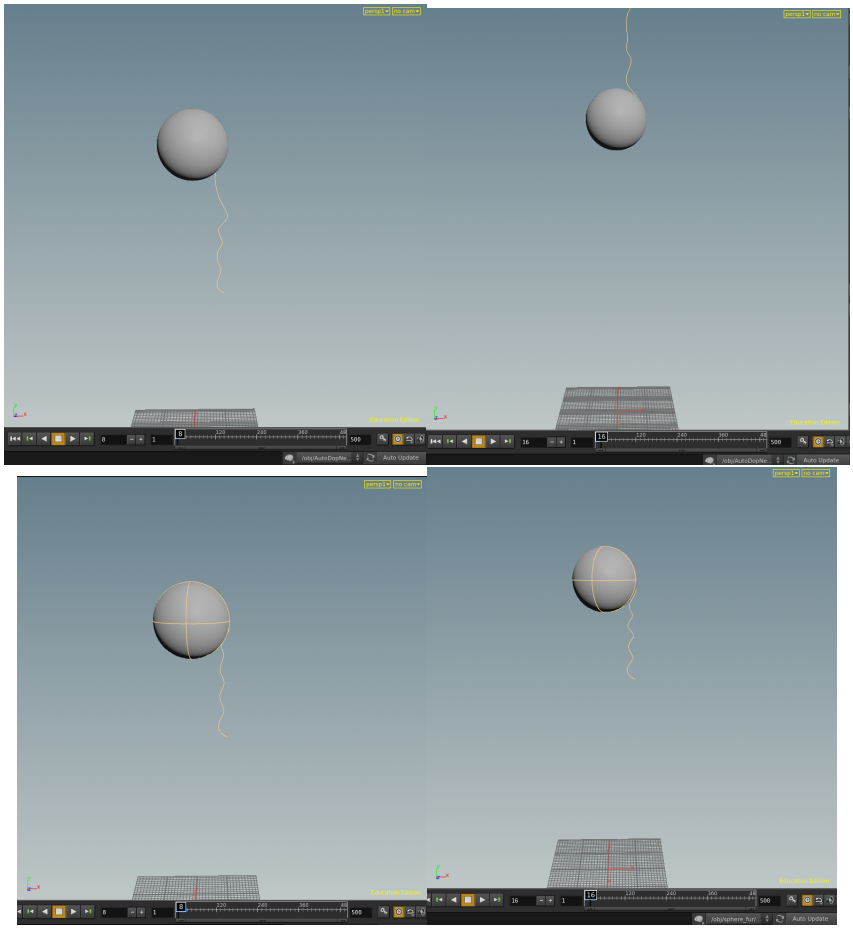

Figure 9: Comparison between Houdini built in wire solver for hair, and the proposed method. The two on the top are from Houdini built in wire solver in two different times using the same properties, on the buttom the proposed method in the same times, using their own same properties. Notice that the wire solver could give better output by raising the stiffness which would affect the appealing output of the simulation.

between cartoon hair, realistic hair, volume hair, geometric represented hair, or strain represented hair. A system has been implemented to simulate hair dynamics. The system consists mainly of a set of particles connected by three springs representing a strain of hair. A linear damping stretch spring as the main spring connecting the particles together. A bending spring that is responsible for maintaining the bending of the hair strain by parallel transporting a material frame along a smoothed curve, which by turn make the simulation more stable and causing less strain unwinding in secondary motion. And a core spring that is responsible for handling keeping the length and winding of the hair strain even in extraordinary motion of the character. The implementation is made in Houdini to allow users to use it easily through the interface of houdini. The implementation is added as part of the hair built in system of Houdini. The implementation is based mainly on vops, vex snippets, and pop solvers that make pre-force calculations, calculate forces, and integrate forces respectively. Although the implementation should an appealing potential and flexibility to handle different types of hair and give different desired behavior, either being realistic or not, there is still more to advance, more to optimize, and more to integrate with Houdini in this implementation. At the end, this is one implementation of one method, there are, as discussed before, a lot of different method that were explored and introduced throughout the years. Each method has its own uses, and best practices. The method implemented in here was firstly developed to deal with curly hair in a way that is not necessarily realistic. However, it showed the ability to adapt with different kinds of hair, and has been used in the same production to produce different types of hair. Given the advantage of the proposed method, some other methods and models are definitely worth studying as they give more variety for different uses, and fit better in other different situations.

# 7 Appendices

## 7.1 Code Snippets

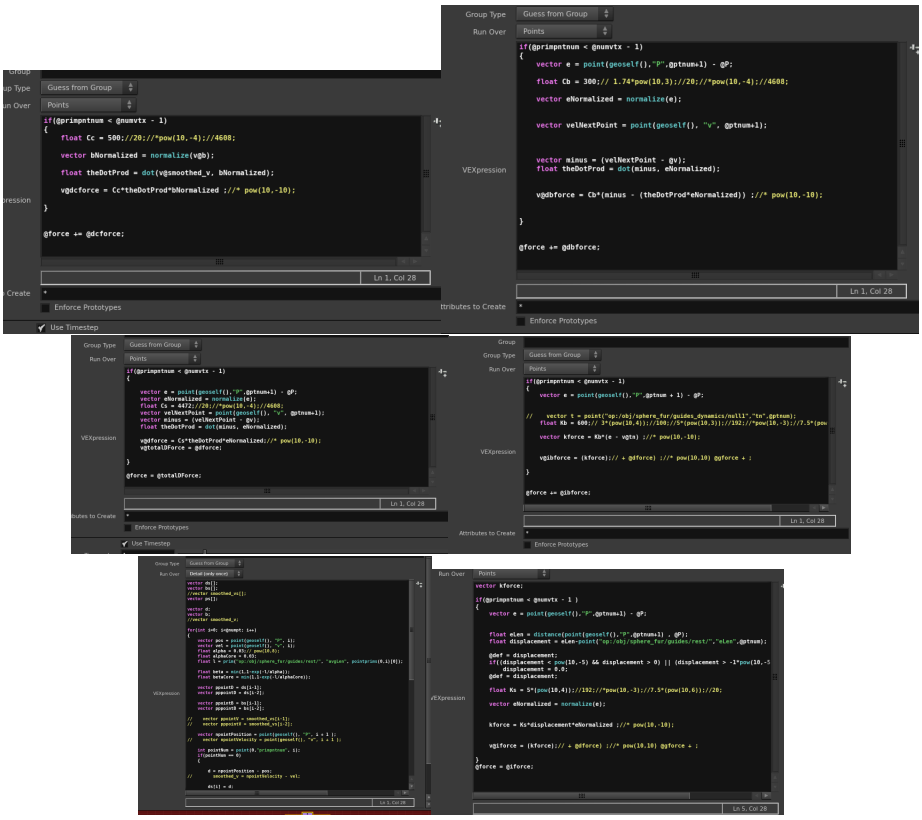

Figure 10: Some VEX Code Snippets from Geometry Wranglers

# References


[1] Miklós Bergou, Max Wardetzky, Stephen Robinson, Basile Audoly, and Eitan Grinspun. Discrete Elastic Rods. *ACM Transactions on Graphics (SIGGRAPH)*, 27(3):63:1–63:12, aug 2008.

[2] Florence Bertails, Basile Audoly, Marie-Paule Cani, Bernard Querleux, Frédéric Leroy, and Jean-Luc Lévêque. Super-helices for predicting the dynamics of natural hair, August 2006. accepted to Siggraph'06.

[3] Jules Bloomenthal. Graphics gems. chapter Calculation of Reference Frames Along a Space Curve, pages 567–571. Academic Press Professional, Inc., San Diego, CA, USA, 1990.

[4] Johnny T. Chang, Jingyi Jin, and Yizhou Yu. A practical model for hair mutual interactions. In *Proceedings of the 2002 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, SCA '02, pages 73–80, New York, NY, USA, 2002. ACM.

[5] Byoungwon Choe and Hyeong-Seok Ko. A statistical wisp model and pseudophysical approaches for interactive hairstyle generation. *IEEE Transactions on Visualization and Computer Graphics*, 11(2):160–170, March 2005.

[6] Mark DeLoura. *Game Programming Gems 2*. Charles River Media, Inc., Rockland, MA, USA, 2001.

[7] Sunil Hadap and Nadia Magnenat-Thalmann. Modeling dynamic hair as a continuum. *Computer Graphics Forum*, 20(3):329–338, 2001.

[8] Hayley Iben, Mark Meyer, Lena Petrovic, Olivier Soares, John Anderson, and Andrew Witkin. Artistic simulation of curly hair. In *Proceedings of the 12th ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, SCA '13, pages 63–71, New York, NY, USA, 2013. ACM.

[9] X. Mao, S. Isobe, K. Anjyo, and A. Imamiya. Sketchy hairstyles. In *Proceedings of the Computer Graphics International 2005*, CGI '05, pages 142–147, Washington, DC, USA, 2005. IEEE Computer Society.

[10] J.P. Morris. *An Overview of the Method of Smoothed Particle Hydrodynamics*. Berichte der Arbeitsgruppe Technomathematik. Univ., Fachbereich Mathematik, 1995.

[11] Matthias Müller, Tae-Yong Kim, and Nuttapong Chentanez. Fast simulation of inextensible hair and fur. In *VRIPHYS 12: 9th Workshop on Virtual Reality Interactions and Physical Simulations, Darmstadt, Germany, 2012. Proceedings*, pages 39–44, 2012.

[12] M. Najim. *Digital Filters Design for Signal and Image Processing*. ISTE. Wiley, 2013.

[13] P. Noble and W. Tang. Modelling and animating cartoon hair with nurbs surfaces. In *Computer Graphics International, 2004. Proceedings*, pages 60–67, June 2004.

[14] Rick Parent. *Computer animation : algorithms and techniques.* The Morgan Kaufmann series in computer graphics and geometric modeling. San Francisco ; London : Morgan Kaufmann Publishers, c2002., 2002.

[15] Robert E. Rosenblum, Wayne E. Carlson, and Edwin Tripp. Simulating the structure and dynamics of human hair: Modelling, rendering and animation. *The Journal of Visualization and Computer Animation*, 2(4):141–148, 1991.

[16] Andrew Selle, Michael Lentine, and Ronald Fedkiw. A mass spring model for hair simulation. *ACM Trans. Graph.*, 27(3):64:1–64:11, August 2008.

[17] Kelly Ward, Florence Bertails, Tae-Yong Kim, Stephen R. Marschner, Marie-Paule Cani, and Ming Lin. A survey on hair modeling: Styling, simulation, and rendering, Mar-Apr 2007. To appear.

[18] Yichen Wei, Eyal Ofek, Long Quan, and Heung-Yeung Shum. Modeling hair from multiple views. *ACM Trans. Graph.*, 24(3):816–820, July 2005.