# Cartoon Hair: Blobby Model Based Approach

酒井, 健行 / SAKAI, Takeyuki

(出版者 / Publisher)
法政大学大学院情報科学研究科
(雑誌名 / Journal or Publication Title)
法政大学大学院紀要. 情報科学研究科編
(巻 / Volume)
9
(開始ページ / Start Page)
107
(終了ページ / End Page)
112
(発行年 / Year)
2014-03
(URL)
https://doi.org/10.15002/00010530

# Cartoon Hair: Blobby Model Based Approach

Takeyuki Sakai
Graduate School of Computer and Information Sciences, Hosei University
E-mail: 12t0006@cis.k.hosei.ac.jp

Abstract

This paper describes a new hair representation technique for drawing 3D CG characters. In our approach hair models are represented by skeleton-based implicit surfaces. Implicit modeling allows representing union and division of wisps of hair by using a blobby model. Our method gets hair skeletons as input data. The skeleton is interpolated using a spline function and then implicit surface of hair shape is generated by the skeleton. And ray marching method is used to compute intersection of ray and surface of hair implicite function. Finally, non-photorealistic rendering is applied to generate resulting image as output. In our approach hair skeletons are created by operating GUI skeleton editor or tracking hand motion using depth camera. This system allows the artist to create skeleton with the help of intuitive operations. Physically-based animation of the hair model is also proposed. In our approach, spring-mass model is used to animate a solid hair shape.

## 1 Introduction

In Japanese cartoon animations, pictures are drawn usually by an artist. But nowadays we can see increasing interest in combining 3D CG characters and hand-drawn characters to make action-packed scene, for example, a battle or dance scene. In such an approach, 3D CG characters mimic hand-drawn characters.

A true real-time animation of animated characters with complex hairstyles was discussed in many papers. Early work on computer generated hair is by Kajiva and Kay [5]. In [16] authors mark that because the difficult, often unsolved problems that arise in this area, a broad diversity of approaches is used, each with strengths that make it appropriate for particular applications. For more references see also a good overview [14].

According to Anime News Network, the artwork in Anime is based primarily on manga, which is a comic book style that became popular in Japan following World War II. Seems, Anime hair modeling is more difficult problem than true real-time animation of animated characters because of the problem of a formal definition of Japanese style of hair drawing in cartoon animation. For example, Anime hair comes in bright unnatural colors. Anime is a type of Japanese animation that is characterized by its highly stylistic and sometimes even unrealistic visual imagery.

Unlike Anime hair modelling and rendering for true real-time animation of complex hairstyles involves the task of dealing with high-density hair strands and is often based on using mechanical models. There are no room to overview papers related to true real-time animation of complex hairstyles, let us only point out existing approaches, which, in principle, can be used and some of them are used for 3D Anime: spring-mass, dynamic continuum, loosely connected particles, NURBS surface shell models.

Supervisor: Prof. Vladimir Savchenko

It is necessary to note that a few publications related to the problem of Anime hair drawing are available. Most Anime art styles derive from a few influential manga artists.

Creating good looking hair by hand for Anime remains a tedious task. In this paper, we focus on hair modeling of 3D CG characters. Traditionally, characters' hairs are made by polygonal models or by using parametric surface representation. Nevertheless, polygonal models are static and there are motion limitations. These models cannot represent union and division of wisps of hair shown in hand-drawn animations, see Fig. 1. Thus, flexible hair models allowing attaining hand-drawn expressions are needed. Traditional art is a rich source of inspiration for artistic hair or cartooning. In a particular, in this paper we are trying to mimic Japanese dark angel like style drawing.

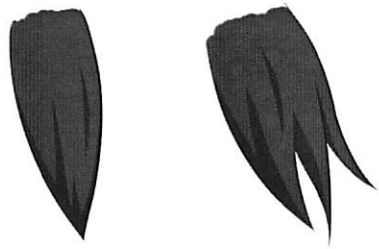

**Figure 1.** Example of hair transformation in hand-drawn animation.

## 2 Related Works

Research related to Anime like hair modeling and non-photorealistic rendering seems is less successful than photorealistic hair modeling.

Noble and Tang [8] use NURBS surfaces generated from key hair curves. In this method, the primary shape and motion of the hair is defined by an animated NURBS volume. The basic flat shading style with a two-tone approach is used to highlight the hair clumps and the hairs' natural tendency to stick together. In [7, 6], polygonal hair shapes are generated by strokes gotten from simple sketch drawn by the user.

Paper [12] describes a new hair rendering technique for Anime characters. Hair shapes are represented by using particles. Most of the rendering steps are performed on the GPU. Using the rendering order of the hair strands and the reference image, a simplified silhouette of the hair strands is calculated by applying a Sobel edge detection filter on the reference texture.

In [2], a procedural technique to generate different kinds of hair strands with strands positioning on the head is used.

Sugisaki et al. in [13] use a high-quality two-dimensional cell image of the animated character and roughly sketched images of the character's hairstyles.

Paper [3] presents a system which provides a set of tools to facilitate the positioning of the generated hair over the hand-drawn hair. System includes 4 steps required to produce

cartoonish hair: a preliminary picture drawn by the artist is required in order to fit digitized hair; some parametrically defined patches are positioned over the hand-drawn hair in 2D; some pseudo-lights are then positioned to get the proper highlights; finally, the planar look of the positioned patches can be broken using a depth perturbation technique.

## 3 Proposed approach

In our technique, hair is represented by skeleton-based implicit surfaces and non-photorealistic rendering is used for visualization [9, 10]. Drawing production pipeline includes following steps:

- Input hair skeleton.
- Interpolation of skeleton using a spline function.
- Generate hair shape using the implicit surface.
- Render hair shape by applying a non-photorealistic shading.

For creating hair skeletons we propose graphical user interface to edit skeletons. By using our GUI, artists can create hair skeletons using fingers motion.

Physically-based animation of hair wisps is implemented.

## 4 Skeleton-based Modeling

Our goal is to simplify the artist's work on the hair creation. To achieve this, we represent hair strands with implicit surfaces [1]. Implicit surface is a surface defined by those points that satisfy $f(x) = c$, see equation (1), for some constant c, where $f(x)$ is a function of several variables. In our approach, the implicit surface is represented by a blobby-like model. Blobby model is a commonly used model. Blobby model allows representing split and fusion of objects easilly. Thus we use blobby model for representing hair shapes. The blobby model represents objects by isosurface of a density field built from summation of local density functions of primitives (2).

$$f(x) - C = 0 \, (C \neq 0) \tag{1}$$

$$f(x) = \sum_{p} f_p(x), \tag{2}$$

where $f_p$ is a local function of $p$ primitive.

### 4.1 Hair Skeleton

The surface is generated from set of strokes (hair skeletons) produced by the artist. The hair skeleton enables the creation a character's hair style. Each hair skeleton represents each single wisp of hair, see Fig. 2. Each node of a skeleton has effective weight and radius of influence as shown in Fig. 3. These parameters are assigned by the user and used for deciding width of wisp and extent of blending.

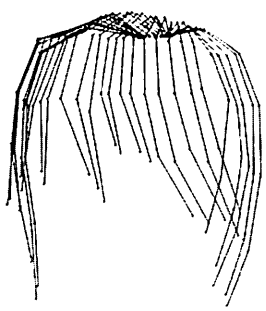

**Figure 2.** Example of 3D hair skeletons.

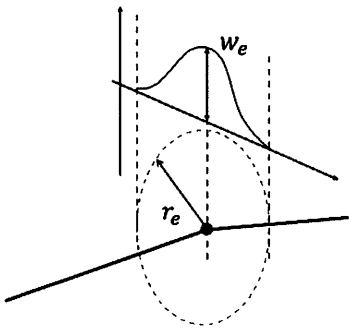


**Figure 3.** Parameters of skeletons. $r_e$: radius of influence, $w_e$: effective weight.

### 4.2 Interpolation of Skeletons

Skeletons are interpolated using the cubic spline function in order to attain smoothed hair shape as it shown in Fig. 4. We suppose that the ordinate are given by the user approximately and the cubic spline function is used for interpolating the hand-drawn hair or $N$ data points smoothly. The spline passes through each data point forming a smooth function $S_j(t_j)$, were local functions of each interval are defined by (3), $j = 0 \ldots N-1$. In function (3), $t_j$ is defined by (4) and parameters $a_j$, $b_j$, $c_j$, and $d_j$ are defined in conformity with constrains (5), (6), and (7).

$$S_j(t_j) = a_j t_j^3 + b_j t_j^2 + c_j t_j + d_j (0 \leq t_j \leq 1) \tag{3}$$

$$t_j = t - j (j = 0, 1, \ldots, N-2) \tag{4}$$

$$S_j'(1) = S_{j+1}'(0) \tag{5}$$

$$S_j''(1) = S_{j+1}''(0) \tag{6}$$

$$S_0''(0) = S_{N-2}''(1) = \mathbf{0} \tag{7}$$

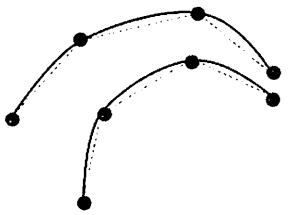

**Figure 4.** Spline interpolation.

### 4.3 Hair Shape Function

The hair shape which is the isosurface of a density function is modeled using the blobby primitives. Blobby modeling has a feature that blobby shapes placed close to one another can be blended together. The isosurface represents points of a constant value of the function (8) and this function is defined by the difference between a threshold and summation of density functions of each edge. The density function used in the paper is Gaussian like function approximated by the quadratic function (9). Density decays with increasing distance between a point and an edge of a skeleton as it shown in Fig. 5.

$$f_{\text{surface}}(v) = \sum_{e \in S} f_{\text{density}} \left( \frac{D_e(v)}{r_e} \right) w_e - r_{\text{threshold}} \tag{8}$$

$$f_{\text{density}}(x) = \begin{cases} 0 & (1 < x) \\ \frac{3}{2}(1-x)^2 & (\frac{1}{3} < x \leq 1), \\ 1 - 3x^2 & (x \leq \frac{1}{3}) \end{cases} \tag{9}$$

where $D_e$ is distance between the edge and point, $r_e$ is radius of influence, $w_e$ is weight value and, $S$ is a set of edges of a skeleton.

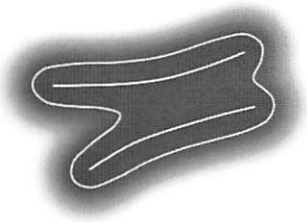

**Figure 5.** Skeleton-based hair shape.

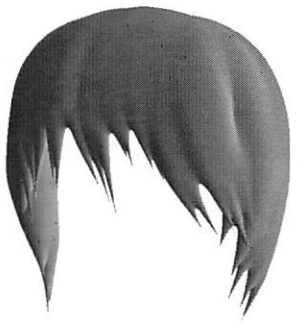

**Figure 6.** Example of hair modeling using the density function (9).

## 5 Non-photorealistic Rendering

Hair models represented by implicit surfaces are rendered using ray marching algorithm. Ray marching algorithm is a kind of ray tracing technique, where calculating ray/surface intersection is produced numerically.

### 5.1 Calculating the Shape Function

To calculate the density function of the skeleton, the distance between a point of a ray and a point interpolated by the spline function in accordance to skeleton points has to be calculated. In this case, the function (10) of 6th degree must be minimized. We use numerical technique descussed in [15] that combines quadratic minimization and Newton's method to find the distance.

$$f(t) = (S_x(t) - v_x)^2 + (S_y(t) - v_y)^2 + (S_z(t) - v_z)^2, \quad (10)$$

where $S_x$, $S_y$, $S_x$ are components of the cubic spline function and $v_x$, $v_y$, $v_z$ are coordinates of a ray point.

### 5.2 Bounding Volume

Ray marching algorithm is time-consuming to calculate ray/-surface intersection. Using bounding volume, limiting ray range is needed to reduce searching time.

At first, cubic spline function is converted to cubic Bezier function. And bounding volume of each skeleton segment is made by convex envelope of Bezier curve. Segment of cubic Bezier curve is defined by (11) with control points $\boldsymbol{v}_{j0}$, $\boldsymbol{v}_{j1}$, $\boldsymbol{v}_{j2}$, $\boldsymbol{v}_{j3}$. To solve function (11) for $t_j$, equation (12) is obtained.

$$\begin{aligned}\boldsymbol{B}_j(t_j) =& t_j^3 \boldsymbol{v}_{j3} + 3t_j^2(1-t_j)\boldsymbol{v}_{j2} \\ &+ 3t_j(1-t_j)^2 \boldsymbol{v}_{j1} + (1-t_j)^3 \boldsymbol{v}_{j0}\end{aligned} \quad (11)$$

$$\begin{aligned}\boldsymbol{B}_j(t_j) =& (\boldsymbol{v}_{j3} - 3\boldsymbol{v}_{j2} + 3\boldsymbol{v}_{j1} - \boldsymbol{v}_{j0}) t_j^3 \\ &+ (\boldsymbol{v}_{j2} + 2\boldsymbol{v}_{j1} + \boldsymbol{v}_{j0}) t_j^2 \\ &+ (\boldsymbol{v}_{j1} - \boldsymbol{v}_{j0}) t_j + \boldsymbol{v}_{j0}\end{aligned} \quad (12)$$

Then parameters of spline function in equation (3) and bezier function (12) are used to define the equation (13). Control points $\boldsymbol{v}_{j0}$, $\boldsymbol{v}_{j1}$, $\boldsymbol{v}_{j2}$, $\boldsymbol{v}_{j3}$ are obtained from equation (13).

$$\begin{cases} \boldsymbol{v}_{j3} - 3\boldsymbol{v}_{j2} + 3\boldsymbol{v}_{j1} - \boldsymbol{v}_{j0} = \boldsymbol{a}_j \\ \boldsymbol{v}_{j2} + 2\boldsymbol{v}_{j1} + \boldsymbol{v}_{j0} = \boldsymbol{b}_j \\ \boldsymbol{v}_{j1} - \boldsymbol{v}_{j0} = \boldsymbol{c}_j \\ \boldsymbol{v}_{j0} = \boldsymbol{d}_j \end{cases} \quad (13)$$

And axis-aligned bounding box of each edge segment of skeleton is created from control points of Bezier curve. Because Bezier curve control points have convex envelope, bounding box of the all skeleton is created by fitting control points.

### 5.3 Non-photorealistic Shading

In our technique, shading is done in cartoon style. This shading is based on typical toon shading by implementing non-photorealistic lighting technique where a color intensity is calculated at first and after that binary painting is applied. Shading is applied in a local coordinate system defined the head orientation of a character.

The specular intensity is calculated by using only latitude angular difference between normal vector and the direction of light. Process of shading consist of the following steps:

1. Normal vector and the light direction are converted into local coordinate system of the head.
2. Each 3D vector is converted into 2D vector including a latitudinal component, which is in accordance to Z axis (15), and a longitudinal component, which is projected onto local XY plane (16).
3. Dot product of 2D normal and light directional vectors defined in the step 2 defines the specular intensity (14).

Diffuse intensity is calculated in accordance to the longitude angular difference between the normal vector and the light direction (19) and the latitude angle of the normal (18). Diffuse intensity is defined by multiplication of the latitudinal component of the normal vector and the dot product of normal and light directions projected onto the local XY plane (17). These intensities are shown in Fig. 7.

Finally, color is defined using intensity and threshold parameters given by the user. The color is separated into specular and diffuse components (21,22), after that each component is divided according to threshold parameters that allows forming a light and dark areas, and then 2 parts are mixed according to (20). An example of shading illustrating so called angel halo is shown in Fig. 8.

$$I_{\text{specular}} = N_{\text{latitude}} L_{\text{latitude}} + N_{\text{longitude}} L_{\text{longitude}} \quad (14)$$

$$V_{\text{latitude}} = \boldsymbol{V} \cdot \boldsymbol{E}_z \quad (15)$$

$$V_{\text{longitude}} = \sqrt{(\boldsymbol{V} \cdot \boldsymbol{E}_x)^2 + (\boldsymbol{V} \cdot \boldsymbol{E}_y)^2} \quad (16)$$

$$I_{\text{diffuse}} = I_{\text{latitude}} I_{\text{longitude}} \quad (17)$$

$$I_{\text{latitude}} = 0.5(N_{\text{latitude}} + 1) \quad (18)$$

$$I_{\text{longitude}} = 0.5\left(\frac{\boldsymbol{L} \cdot \boldsymbol{E}_x}{L_h}\frac{\boldsymbol{N} \cdot \boldsymbol{E}_x}{N_h} + \frac{\boldsymbol{L} \cdot \boldsymbol{E}_y}{L_h}\frac{\boldsymbol{N} \cdot \boldsymbol{E}_y}{N_h} + 1\right), \quad (19)$$

where $\boldsymbol{N}$ is the normal vector, $\boldsymbol{L}$ is the normalized light direction, $\boldsymbol{V}$ is the vector $\boldsymbol{N}$ or $\boldsymbol{L}$, $\boldsymbol{E}_x$, $\boldsymbol{E}_y$, $\boldsymbol{E}_z$ are components of the basis vector defined in the local coordinate system of the head.

Color $C$ is defined as follows:

$$C = C_{\text{specular}} + C_{\text{diffuse}} \quad (20)$$

$$C_{\text{specular}} = \begin{cases} c_{\text{specular}} & (t_{\text{specular}} < I_{\text{specular}}) \\ 0 & (I_{\text{specular}} \le t_{\text{specular}}) \end{cases} \quad (21)$$

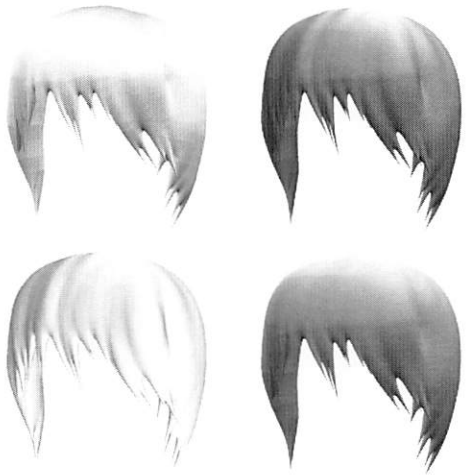

**Figure 7.** Shaded and highlighted areas according to $I_{\text{specular}}$ (top left), $I_{\text{deffuse}}$ (top right), $I_{\text{longitude}}$ (bottom left), and $I_{\text{latitude}}$ (bottom right) values.

$$C_{\text{diffuse}} = \begin{cases} c_{\text{diffuse}} & (t_{\text{diffuse}} < I_{\text{diffuse}}) \\ c_{\text{shade}} & (I_{\text{diffuse}} \leq t_{\text{diffuse}}) \end{cases} \tag{22}$$

Where $t$ is threshold of intensity, $c$ is color.

**Figure 8.** Result of shading.

## 6 User Interface using Depth Camera

Our model cannot be made on existing 3DCG software without any extending. We propose a graphical user interface for creating hair skeletons. Screen caputure of our system is shown in Fig. 9. The artist have to create or edit hundreds of skeleton nodes to define positions and parameters. It is tedious for designers. Thus we need a user interface allowing intuitive operations to create skeletons.

In our approach, hair skeleton for our model can be created by tracking finger motion using depth camera. Depth camera tracks 2 fingers and skeletons parameters depend on fingertips positions and distance between each fingertip.

2 fingertips are defined by 2 nearest convex (white areas shown in Fig. 9) areas of the image. Process of searching fingertips is as follows:

1. Search a point, which defines a pixel image position, closest to the camera.
2. Expand the area (one pixel area on the image) around the pixel found at step 1 at a time.
3. Terminate expanding of the area if relative depth of the point exceeds a given threshold or is becoming less than relative depth of outside points.
4. Define this area as the first fingertip area.
5. Search for another point which is the nearest one excluding the first fingertip area.
6. Expand the area around the point found in the step 5 as it has been done in steps 2-3.
7. Define this area as the second fingertip area.
8. Two fingertips are defined by the centers of corresponding areas.

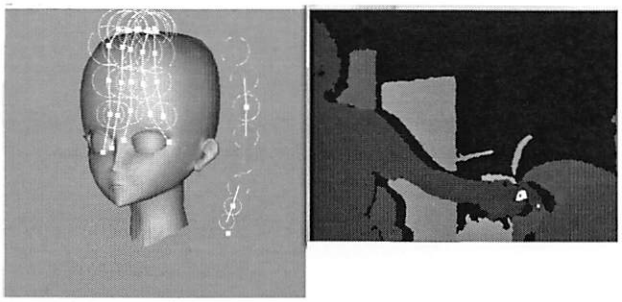

**Figure 9.** Screen capture of our GUI. Left - skeleton editor. Right - preview of depth data detected by camera.

## 7 Physically-based Animation

In our approach, spring-mass model is implemented for hair animation. Spring-masss models are used for simulation of soft-body object, see [4] and references herein. Hair simulation is one of the soft-body simulations used spring-mass model, see [11].

In spring-mass model, soft-body objects are represented by mass points and springs connecting neighbor mass points. And solving motion equations for mass points, motion of object is calculated. In Fig. 10, struct of mass/spring connection in our method is shown. One skeleton node has one primary mass point and 8 sub mass points. The primary mass point is set same location of skeleton node. And sub mass points are set around the primary mass points distanced according to the blob radius which is the skeleton parameter. And mass points are connected by set of springs for restoring deformation.

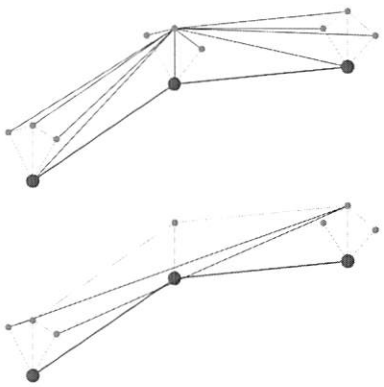

**Figure 10.** Mass points connecting. A connection between one sub mass point and all neighbor points (top). Releasing connections for producing deformations (bottom).

Equation (23) shows force applied to the mass point $p$. The flowchart of calculations is shown in Fig. 11. In our method, Runge-Kutta 4th order method is used for time integration.

$$\boldsymbol{f}_p = \sum_{s \in S} k_s \left( l_s - l_{s,\text{natural}} \right) \boldsymbol{d}_{s,p} + \boldsymbol{g} m_p - c_{\text{damp}} \boldsymbol{v}_p, \tag{23}$$

where $S$ defines a set of springs, $k_s$ is a spring constant of the

spring $s$, $l_s$ is the spring length, $\boldsymbol{d}_{s,p}$ is direction vector from $p$ to the other side point of spring $s$, $\boldsymbol{g}$ is gravity acceleration, $m_p$ is mass of point $p$, $c_{\text{damp}}$ is damping coefficient, and $\boldsymbol{v}_p$ is velocity of $p$.

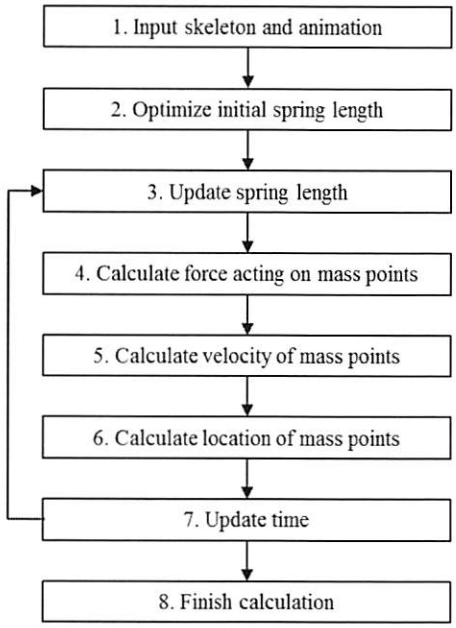


**Figure 11.** Flowchart of hair animation.

### 7.1 Optimization of natural spring length

Hair behavior is affected by gravity and hair shape is changed from shape defined by inputed skeleton even if head doesn't move. But artists cannot design a hair skeleton in consideration of deformation by gravity. Thus hair shape should not deform from initial shape when any force without gravity doesn't affect a hair. Therefore springs' natural lengths are adjusted to balance between force from spring and gravity befor starting simulation.

Balancing between force from spring and gravity means that we have satisfy the equation (24). To solve the equation (24) for $l_{s,\text{natural}}$, we can get hair model that doesn't change from default shape with gravity.

$$\boldsymbol{f}_p = \sum_{s \in S} k_s \left( l_{s,\text{initial}} - l_{s,\text{natural}} \right) \boldsymbol{d}_{s,p} + \boldsymbol{g} m_p = \boldsymbol{0}, \qquad (24)$$

where $l_{s,\text{initial}}$ is initial length of spring $s$, and $l_{s,\text{natural}}$ is natural length balancing with gravity.

This system of equation has less linear equations than demanded unknown variables. Singular value decomposition to solve this system of equations is used.

## 8 Result

Our method allows representing hand like drawn transformations by only editing skeletons, see Fig. 12. We can see one thick hair wisp divided into a number of thin wisps. Using our method, producing such transformations do not require artists to modify the hair surface directly.

The hair model, non-photorealistic rendering, and animation are shown in Fig. 13. It illustrates simulation of hair motion such as shaking a characters head from side to side. Hair flows are in concert with head motion. Then hair wisps divide and unite. Binary painting produces results resembling cartoon painting and allows creating a horizontal specular slat, so called angel halo.

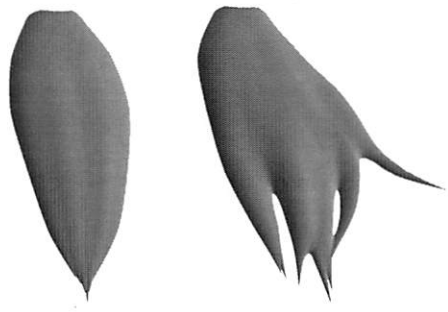

**Figure 12.** Example of simulation of transforming a hair wisp.

## 9 Conclusion and Future Work

For non-photorealistic characters hair, a hair shape model based on the use of implicit surfaces, non-photorealistic hair shading, physically-based hair dynamics are proposed. Developed hair skeleton editor allows simplifying model design. We can say that the proposed technique produces reasonable visual results, that is, modeling hair wisps with skeletons has shown to be a powerful and promising approach.

Nevertheless, our system is still under development. Currently, a basic blobby function is used for representing hair shape. Normally, partings of hair wisps are sharp, as we can see in Fig. 1. But using the basic blobby function produces hair wisps blended smoothly including partings. Therefore we need to modify the function specialized for hair shape partings to reach hand drawn shape rendering.

In current realization of hair animation, a hair wisp moves independent of other wisps. There is no collision detection check between each hair wisps or between hair and characters' body. We need to implement collision check processing hair wisps. In practice, hair wisps can intersect bounds of other wisps because hair wisps are blended with other wisps. But between hair and characters' body, a hair wisp must not have intersections with a character body.

In current realization of our approach, each hair wisp moves separately. Thus, spread hair shape appearance becomes too different form artists designing. It's preferable for hair to move loosely at the thick hair wisp level. In future, we need to take into consideration spring inter connection of wisps to realize solid hair to mimic artists' hand drawing.

Non-photorealistic shading is still a problem with our system and must be extended and improved. In our approach, toon shading is applied, but only one style and it doesn't guaranty reaching perfect such as hand-drawn expression results.

## References


[1] Jules Bloomenthal and Brian Wyvill, editors. *Introduction to Implicit Surfaces*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1997.

[2] V.J. Cassol, F.P. Marson, and S. Raupp Musse. Procedural hair generation. In *Games and Digital Entertainment (SBGAMES), 2009 VIII Brazilian Symposium on*, pages 185–190, 2009.

[3] Martin Côté, Pierre-Marc Jodoin, Charles Donohue, and Victor Ostromoukhov. Non-Photorealistic Rendering of Hair for Animated Cartoons. In *Proceedings of GRAPHICON'04*, 2004.

[4] Xavier Provot Institut and Xavier Provot. Deformation constraints in a mass-spring model to describe rigid cloth behavior. In *In Graphics Interface*, pages 147–154, 1996.

**Figure 13.** Example of hair animation and rendering using our method.


[5] J. T. Kajiya and T. L. Kay. Rendering fur with three dimensional textures. In *Proceedings of the 16th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '89, pages 271–280, New York, NY, USA, 1989. ACM.

[6] 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.

[7] Xiaoyang Mao, H. Kato, A. Imamiya, and K. Anjyo. Sketch interface based expressive hairstyle modelling and rendering. In *Computer Graphics International, 2004. Proceedings*, pages 608–611, 2004.

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

[9] Takeyuki Sakai and Vladimir Savchenko. Skeleton-based anime hair modeling and visualization. In *Cyberworlds (CW), 2013 International Conference on*, pages 318–321, 2013.

[10] Takeyuki Sakai and Vladimir Savchenko. Skeleton-based cartoon hair modeling using blobby model. In *SIGGRAPH Asia 2013 Posters*, SA '13, pages 17:1–17:1, New York, NY, USA, 2013. ACM.

[11] Andrew Selle, Michael Lentine, and Ronald Fedkiw. A mass spring model for hair simulation. In *ACM SIGGRAPH 2008 Papers*, SIGGRAPH '08, pages 64:1–64:11, New York, NY, USA, 2008. ACM.

[12] Jung Shin, Michael Haller, and R. Mukundan. A stylized cartoon hair renderer. In *Proceedings of the 2006 ACM SIGCHI International Conference on Advances in Computer Entertainment Technology*, ACE '06, New York, NY, USA, 2006. ACM.

[13] Eiji Sugisaki and Yizhou Yu. Simulation-based cartoon hair animation. In *In 13th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision (WSCG' 05*, pages 117–122, 2005.

[14] Pascal Volino and Nadia Magnenat-Thalmann. Real-time animation of complex hairstyles. *IEEE Transactions on Visualization and Computer Graphics*, 12(2):131–142, 2006.

[15] Hongling Wang, Joseph Kearney, and Kendall Atkinson. Robust and efficient computation of the closest point on a spline curve. In *In Proceedings of the 5th International Conference on Curves and Surfaces*, pages 397–406, 2002.

[16] Kelly Ward, Florence Bertails, Tae yong Kim, Stephen R. Marschner, Marie paule Cani, and Ming C. Lin. A survey on hair modeling: styling, simulation, and rendering. In *IEEE TRANSACTION ON VISUALIZATION AND COMPUTER GRAPHICS*, pages 213–234, 2006.