ISSN 1000-9825, CODEN RUXUEW
Journal of Software, Vol.20, Supplement, December 2009, pp.202−212
© by *Institute of Software*, *the Chinese Academy of Sciences*. All rights reserved.

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel/Fax: +86-10-62562563

基于手绘草图的交互式发型生成方法*

王　斌[1,2]，孙正兴[1,2+]，张　岩[1,2]

[1](南京大学　计算机软件新技术国家重点实验室,江苏　南京　210093)
[2](南京大学　计算机科学与技术系,江苏　南京　210093)

Sketch-Based Method for Interactive Hairstyling

WANG Bin[1,2], SUN Zheng-Xing[1,2+], ZHANG Yan[1,2]

[1](State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210093, China)
[2](Department of Computer Science and Technology, Nanjing University, Nanjing 210093, China)
+ Corresponding author: E-mail: szx@nju.edu.cn, http://cs.nju.edu.cn/szx

Wang B, Sun ZX, Zhang Y. Sketch-Based method for interactive hairstyling. *Journal of Software*, 2009, 20(Suppl.):202−212. http://www.jos.org.cn/1000-9825/09024.htm

Abstract: To support hairstyling effectively, it's crucial for hairstyling tools to find a tradeoff between realism and interaction. This paper presents an interactive hairstyling method which can handle both global and local details of the hairstyle. Vector field based hairstyling method is used to generate the global shape of the hairstyle. This paper defines three different kinds of hairstyle curves which indicate the boundary constraints of vector field so that the user can control the hairstyle generation interactively. Hair wisp model is also used to represent hairstyles and several wisp based editing operations are defined, such that users can control the local details of the hairstyle. Experimental results prove that the proposed method provides more interaction and computation efficiencies without lost of hair realism.

Key words: hairstyling; wisp model; vector field; hairstyle curve; sketch-based interaction

摘　要: 发型的真实感和交互性之间的有效折衷是发型建模面临的一个重要问题.提出了一种交互式发型建模方法,使得用户可以通过交互操作处理发型的整体和局部特性.利用基于矢量场的发型建模方法来生成发型的整体形状.用户可通过三类不同的发型曲线所生成的约束来交互控制发型生成.利用发束模型表示发型并定义了若干基于发束的编辑操作使得用户可以交互控制发型的局部细节.实验结果表明,该方法在保证发型真实感的基础上提高了交互性和实时性.

关键词: 发型建模;发束模型;矢量场;发型曲线;草图交互

Hairstyle is an important feature of virtual human character in computer graphics. Hair modeling is used to

* Supported by the National Natural Science Foundation of China under Grant Nos.69903006, 60373065, 60721002 (国家自然科学基金); the National High-Tech Research and Development Plan of China under Grant No.2007AA01Z334 (国家高技术研究发展计划(863)); the Program for New Century Excellent Talents in University of China under Grant No.NCET-04-04605 (新世纪优秀人才支持计划)

Received 2009-05-15; Accepted 2009-07-23

create visually realistic virtual humans for many applications such as films, computer games and advertising. Hair modeling can be divided into three categories[1]: hairstyling, hair dynamics[2,3] and hair rendering[4]. We focus on hairstyling which is aimed to model the shape of hair. It's difficulty to ensure both hair realism and system efficiency due to the huge number of hair strands and the complexity of hairstyles. Besides, human-computer interaction should also be considered. Thus, hairstyling still remains a challenging task.

A lot of significant hairstyling methods have been proposed[1]. We only introduce the research mostly related to ours.

Noting that human hair exhibits clustering behavior, a group of hair strands that are spatially close have similar geometric features. Watanabe[5] proposed a wisp model to generate a group of similar strands. Other approaches have also been proposed since then. Yang[6], Chen[7] and Xu[8] used generalized cylinders to model a group of hair strands, the position and the shape of the wisp are controlled by its center curve. Choe[9] introduced a statistical wisp model, the wisp shape is controlled by some parameters with statistical meanings. Kim[10] introduced a multiresolution hair modeling method based on hair clusters. User can edit the hair at any level of detail. Wisp model can represent most hairstyles and it can handle local details very well. But it has some limitations in interaction. Since the entire hair usually consists of a large number of hair wisps, which makes the hairstyling process tedious and time consuming.

It can be observed that hair has many properties that are similar to a fluid flow or vector field. For example, hair and vector field possess a clear direction at a specific point. Therefore, several hairstyling methods based on vector field or fluid flow have been proposed. Hadap[11] modeled static hairstyles as streamlines of fluid flow. Three different kinds of flow elements were defined to guide hair growth. Yu[12] used static 3d vector field to generate hair strands, user can superimpose several predefined vector field primitives procedurally. Hair strands are extracted by tracing the field lines of the vector field. Choe[9] also used three styling constraints which were similar to vector primitives defined in Yu[12]. Some complex hairstyle can be obtained based on those styling constraints. Vector field based hairstyling method can handle the global features of hairstyle very well. But there are two major limitations of these methods. First it's little time-consuming to calculate the vector field. Moreover, it's not intuitive enough for users to put primitives in space as they could not get a clear image of the vector field.

Recently more researchers tried to provide sketching interface for hairstyling due to its high intuitiveness. Mao[13,14] used sketches to model cartoon hairstyles. It can quickly generate hairstyle s from strokes drawn by users. However, it's not appropriate for modeling more intricate hairstyles. Malik[15] presented a user-friendly sketching interface for modeling and editing hairstyles. The user can sketch a 3D curve directly as implanting a hair wisp onto the scalp. As it's based on hair wisp model, it has the limitation discussed before. Wither[16] proposed a sketch-based interface for creating visually realistic hair. The user can sketch example hair strands over the side view of head. Certain parameters can be obtained from those strands to control a physical-based model. Fu[17] developed a new hairstyling method that combined a sketching interface and an efficient vector field solver. The user can input a set of predefined style curves to depict the global shape of hairstyle. Different boundary constraints are derived from style curves to calculate the vector field. Many kinds of visually realistic hairstyles can be obtained quickly based on the vector field. However, it costs much time calculate the vector field for the first time and it's a little time-consuming to generate an intricate hairstyle. Moreover, it couldn't handle local details of the hairstyle.

Using the idea from Fu[17] of combining vector field based hairstyling method and sketch based hairstyle curves, we present a sketch-based interactive hairstyling method. Hair wisp model is used to represent hairstyle due to its ability to control local details. All master strands are generated by vector field based hairstyling method while all member strands are created from master strands by statistical parameters. We define three different kinds of hairstyle curves to create more hairstyles by extending the style curves in Fu[17]. Hairstyle curves indicate the boundary constraints of vector field so that the user can control the creation of hairstyle interactively. We also define some hairstyle editing operations based on wisps such as cutting and curling. Interpolation algorithm is used to calculate the vector field which simplifies the computation and guarantees the real-time performance.

1 Algorithm of Hairstyle Generation

In this part we introduce the algorithm of hairstyle generation. We use hair wisp model to represent hairstyles and vector field based hairstyling method to generate master strands.

1.1 Hairstyle representation

As human hair exhibits clustering behavior, a group of hair strands that are spatially close have similar geometric features. adjacent hair strands tend to be similar, We call this group of hair strands a *wisp*. As shown in Figure 1, we use generalized cylinder to represent the boundary of a wisp. All hair strands, include a *master strand* and numerous *member strands*, locate in the boundary of the wisp.

Each hair strand is represented as a sequence of connected equaling line segments as $\{P_0,P_1,\ldots,P_n\}$. P_i (called a *strand node*) is the end point of *i*th line segment.

Master strand serves as a control curve that manipulates the overall shape the wisp. All member strands of the wisp can be created from master strand.

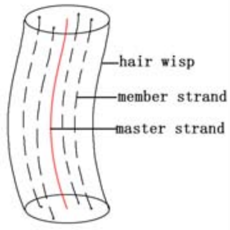


Fig.1 A hair wisp

1.2 Generation of master strands based on vector field

We define a discrete vector field in a 3D uniform grid around the head model. If vectors at some grid points (these vectors are called *vector field constraints*) are known, we calculate the entire vector field by interpolation algorithm. For each grid point G_i, the vector at G_i is:

$$Vi = \sigma \cdot \frac{1}{Num} \sum_{G_j \in Neigh(G_i)} V_j \tag{1}$$

$Neigh(G_i)$ is the set of neighboring grid points where the vector has already been calculated. *Num* is the size of $Neigh(G_i)$. σ is a decrease parameter which is used to smooth the vector field.

Once the entire 3d vector field is calculated, all master strands can be generated based on the 3D vector field. The growth of a master strand starts at its root on the scalp and then goes on a certain steps along the direction of the accumulated vector of the vector field. Assume that the current strand node is p, then the next strand node $pNext$ can be calculated as: $pNext=p+vl$. l denotes the length of unit line segment and $v=v(p)/||v(p)||$,$v(p)$ is the vector at p, it can be interpolated from its surrounding grid points

$$v(p) = \sum_{G_i \in sur(p)} Weigh(G_i) \cdot V_i \tag{2}$$

where $Sur(p)$ is the set of surrounding grid points of p. $Weigh(G_i)$ is weight of V_i, it's in inverse proportion to the distance from p to G_i. Keep calculating all strand nodes until $||v(p)||<s$. s is a predefined parameter which is used to

control the length of hair.

Hair-scalp collision detection should be considered in hair growth because the strand nodes may grow into scalp. Consider a strand node *p* whose distance to the scalp is smaller than *l*, if *v*(*p*) points inwards (the angle between *v*(*p*) and *p*’s nearest scalp outer normal is greater than 90^{o}), its direction is replaced with its projection on the tangent plane of *p*’s nearest scalp point, its length remains unchanged.

1.3 Generation of member strands

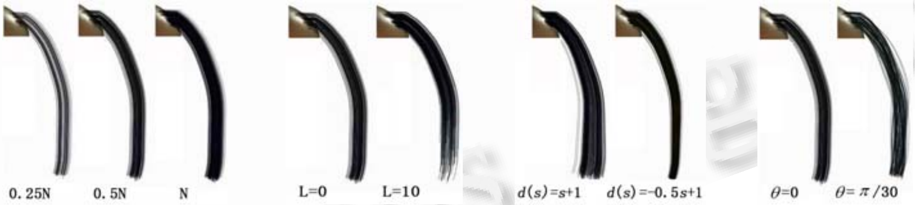
As we discussed in section 1.1, master strand controls the overall shape of the wisp while member strands are all similar to the master strand. We define several parameters, such as density, length, deviation and twist, to measure the degree of similarity among hair strands. Once the shape of the master strand is determined, we can generate all member strands by those parameters.

(1) **Density**: The density denotes the number of strands in a wisp. The scalp is divided into several regions, which has roughly same size. Each region corresponds to a wisp and roots of all strands distributed in the region uniformly, so the density also means the number of roots. We set the maximum number of roots per wisp to *N*. Fig.2(a) shows wisps whose density is 0.25*N*, 0.5*N* and *N* respectively.

(2) **Length**: Length distribution denotes the length variation between master strand and member strands. Since hair strand is represented as connected equilong line segments, we can use the number of strand nodes to denote its length. Let *n* denote the number of strand nodes in master strand. The number of strand nodes of the member strand could locate within [*n*–*L*,*n*]. Fig.2(b) shows two wisps with two different *L*.

(3) **Deviation**: Deviation denotes the offset from member strand to master strand, we adopt a function *d*(*s*) to measure the deviation of *k*th member strand node to its corresponding master strand node. *s*=*k*/*n*, *n* denotes the number of strand nodes in master strand. Fig.2(c) shows two wisps with different deviation function *d*(*s*).

(4) **Twist**: Twist is a rotation value set per strand node. A member strand node may be rotated about the corresponding master strand node for s certain angle. Fig.2(d) shows two wisps with different rotation angle θ.



(a) Density (b) Length (c) Deviation (d) Twist

Fig.2 Defined parameters of member strands

Once the master strand is generated as $\{P_0,P_1,\ldots,P_n\}$ and all these parameters are determined, we can easily generate all member strands. To generate the *k*th strand node P_k', we first find a plane *R* which is perpendicular to the plane passes though P_{k-1}, P_k, P_{k+1}. R also passes through P_k and the midpoint of P_{k-1} and P_{k+1}.Denote the intersection point of $P'_{k-1}+\overline{P_{k-1}P_k}\cdot t$ and *R* as *I*.

$$P'_k = twist(P_k + t_k, \theta_i), \quad t_k = d(s)\cdot\overline{P_k I} \tag{3}$$

Where $Twist(p,\theta)$ is a twist function that rotate *p* about the master strand for θ.

1.4 Algorithm description

Our algorithm is described as follows:

Input: Hairstyle curves.

Step 1. generate vector field constraints of each hairstyle curve

Step 2. compute the entire 3d vector field using interpolation algorithm
Step 3. create master strands of hair wisps by tracing the 3d vector field
Step 4. create member strands from master strands by several parameters
Step 5. edit local details of hairstyle
Output: Hair styles.

2 Sketch-Based Interactive Haristyling

Let’s look back the process of hairstyle generation. Vector field, which is calculated from vector field constraints, controls the global shape of the hairstyle. Besides, as each hair wisp is controlled by its master strand, we can say that master strands control the local details of the hairstyle. If we can control vector field and master strands interactively, both global and local details of the hairstyle can be well handled.

We define sketch-based hairstyle curves to control the vector field and several editing operations to control the master strands.

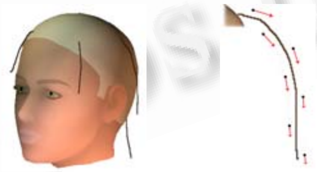
2.1 Definition of hairstyle curve

To support interactive global hairstyling, sketched hairstyle curves are used to depict the global shape of hairstyle. Like Fu[14], we define three different kinds of hairstyle curves: straight curve, ponytail curve and curl curve. They can generate different kinds of vector field constraints. User can sketch several hairstyle curves directly on head model from different views.

2.1.1 Straight curve

Straight curve is a smooth curve which starts from the scalp. It controls the local orientation and length of hair. The key of transforming 2D stroke to 3D curve is to decide the depth of each sample point. User can draw a stroke starting from the scalp. The 3d ray from viewpoint to the start point of stroke and the scalp intersect at a point p. Take p as the start point of straight curve, other sample points of the stroke have the same depth as p.

User can draw several straight curves from different views (Fig.3(a)). Then the vector field constraints are generated (Fig.3(b)). Every straight curve assigns a vector to the neighboring grid points as the constraints of vector field. The direction of the vector is set as the curve’s tangent while the length decreases from start to end in order to control the length of hair strands.



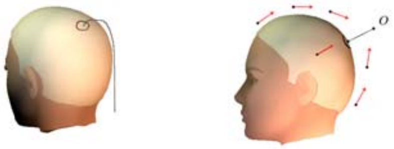
(a) Straight curves (b) Constraints

Fig.3 Straight curves

2.1.2 Ponytail curve

A ponytail curve consists of two separate curves: a straight curve and a circle on scalp (Fig.4(a)). The straight curve has already been discussed in section 2.1.1. Each point of the circle is the intersection point of the 3d ray from viewpoint to corresponding sample point and the scalp.

As shown in Fig.4(b), we first calculate the center point of the circle O, it’s reasonable to assume that O is on scalp, move O along its outer normal vector for a distance. The circle assigns a vector v to each neighboring grid point G of scalp. The direction of v is from G to O while the length of v is set as an uniform value. If v points inwards the scalp (the angle between v and G’s nearest scalp outer normal is greater than 90°), the direction is replaced with its projection on the tangent plane of G's nearest scalp point.



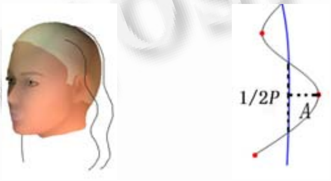
(a) A ponytail curve (b) Constraints of the circle

Fig.4 Ponytail curve

2.1.3 Curl curve

Curl curve is a wavy curve sketched by user. It is used to generate curly hair. The sketching method is the same as straight curve.

To generate the vector field constraints of a curl curve, we first find its inflexions and then generate the center curve (Fig.5(b)). The center curve is treated as a straight curve which generates vector field constraints as we discussed in section 2.1.1.



(a) Curl curves (b) Analysis of curl curve

Fig.5 Curl curve

When the user intents to generate curly hair from curl curves, straight hair is first generated by center curves of curl curves, then we use offset function in Yu[13] to make straight hair curly. Almost all kinds of curly hair can be represented by offset function:

$$Offset(t)=A\sin(2\pi Pt+\phi)+Bias \tag{4}$$

Where t is a parametric variable and A, P, $Bias$, ϕ are predefined const. For a hair strand $\{P_0,P_1,\dots,P_n\}$, we define a local coordinate system at each strand node P_i. The direction of z-axis is from P_i to P_{i+1}. We use the vector from P_i to the centric of the scalp as an UP vector. Both x- and y- axis can be derived from the UP vector and z-axis. We parameterize the strand, for a certain point p on the strand; its corresponding t value is the length from p to the root of the strand. With two sets of different predefined parameters, offset function can calculate different offsets at p along x- and y- axis respectively. We can obtain a new strand node by moving p along both x- and y- axis.

We can easily find that $offset(t)$ is similar to sine curve with A and $1/P$ corresponding to amplitude and period respectively. See Fig.5(b), we can obtain A and P from curl curve directly. A is set as the distance from an inflexion to the center curve and the distance between every two intersection point of curl curve and center curve is $1/2P$. By predefinition of different $Bias$ and ϕ, we can get two different offset functions with the same A and P but different $Bias$ and ϕ.

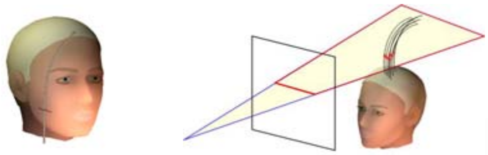
2.2 Wisp based editing operations

We also define several wisp based interactive operations, which are similar to real-world hairstyling. As we mentioned in section 1.3, member strands can be created from the master strand, which decides the position and overall shape of the wisp. Therefore, the entire hair wisp can be modified by editing its master strand.

User can choose several wisps to edit freely, after editing the master strands of those chosen wisps, we only need to update all member strands.

2.2.1 Cutting operation

Similar to the cutting operation in Malik[9], if some wisps are too long, we can cut those hair strands by drawing a relatively straight cutting stroke across the chosen master strands (Fig.6(a)). As shown in Fig.6(b), a cutting plane passed through the viewpoint and the cutting stroke intersects with the chosen master strand at point I. let I be the new end node of the master strand. All strand nodes between I and the original end node is abandoned.

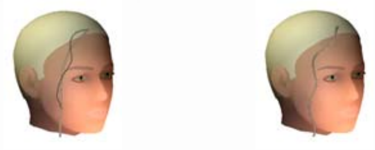


(a) Cutting Stroke (b) Cutting master strands

Fig.6 Cutting operation

2.2.2 Curling operation

User can sketch a wavy curve to curl the chosen master strands (Fig.7(a)). The curling operation is the same as creating curly hair from curl curve. We can obtain parameter A and P for offset function from the curling stroke as we did in section 2.1.3. Then offset function is then used to make the original straight hair curly (as we discussed in section 2.1.3). Fig.7(b) shows the new shape of master strand after curling.



(a) Curling stroke (b) Master strand after curling

Fig.7 Curling operation

2.2.3 Drag operation

User can use mouse or pen to drag the chosen wisps (as shown in Figure 8). A strand node p may be dragged to a new position, denote as p', namely we translate p by a vector $v=p'-p$. Other nodes should also be translated to new positions to maintain the smoothness of hair strand. The translation vector can be linearly interpolated from v.

Fig.8 Drag operation

If p is quite close to the end node of the strand, we translate the end node by v. For the kth node of the strand, we translate it by $(k/n)v$ where k is the index of the node and n is the number of strand nodes. If p is not close to the end node of the strand, denote the index of p as i. We translate the kth node of the strand by $(k/i)v$ if $k<i$ and $(n-k)/(n-i)v$ if $k>i$.

2.2.4 Re-Generate operation

User can sketch some new straight curves around the chosen wisps to re-generate those wisps (as shown in Figure 9). The new straight curve represents the new position and shape of the chosen wisps. The principle of re-generate operation is the same as the generation of straight hair by straight curves. The chosen wisps are replaced with new wisps.

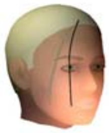


Fig.9 Re-Generate operation

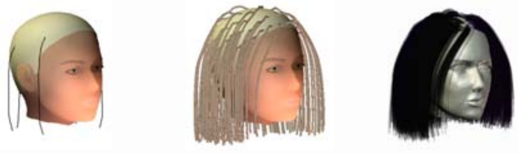
3 Experiments and Evaluations

We used C++ programming language and OpenGL graphics library to implement our method on a PC with an Intel Pentium (4) 2.40 GHz CPU. We used existing software to render the results.

The scalp is divided into 100 regions, namely there're 100 wisps each of which consists of 100 hair strands. The length distribution parameter L=10, the deviation function $d(s)=s+1$ and twist angle θ=0.

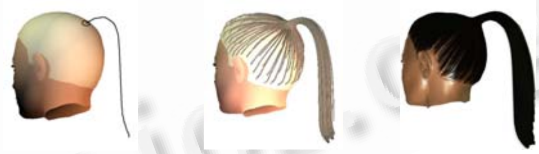
3.1 Experimental verification

We first did three groups of experiments to validate the feasibility of hairstyle curves. As shown in Fig.10(a), user sketched several straight curves in the first experiment. Fig.10(b) and Fig.10(c) are the results, where Fig.10(b) shows all master strands and Fig.10(c) shows the hairstyle after rendering. It took about 4 seconds to create this hairstyle. In the second experiment, user sketched a ponytail curve as shown in Fig.11(a). Fig.11(b) and Fig.11(c) are the results. Time cost is about 6 seconds. At last, user sketched a few curl curves as shown in Fig.12(a) and the results are shown in Fig.12(b) and Fig.12(c). Time cost is about 9 seconds.



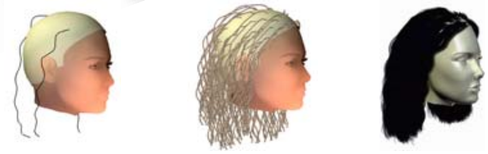
(a) Straight curves (b) Master strands (c) Hairstyle

Fig.10 Illustration of straight hair generation



(a) Ponytail curve (b) Master strands (c) Hairstyle

Fig.11 Illustration of ponytail generation



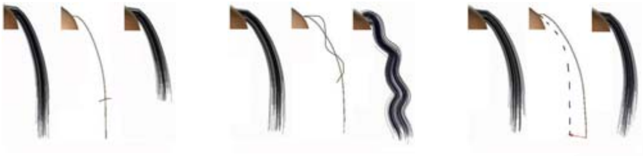
(a) Curl curves (b) Master strands (c) Hairstyle

Fig.12 Illustration of curly hair generation

These three experiments show that we can quickly create different kinds of hairstyle by sketching different hairstyle curves. It usually takes 3s~5s to create straight hair, 6s~7s for ponytail and 7s~10s for curly hair. The time cost is suitable for human-computer interaction.

Then we did several experiments to validate the editing operations based on wisps. Fig.13(a), Fig.13(b) and

Fig.13(c) show cutting, curling and drag operations respectively. Each image shows the original wisp, the editing operation and the result wisp from left to right. It's not necessary to validate the re-generate operation as it has the same principle with hair generation. The results show that our editing operations can edit the local details of hairstyles by editing master strands.

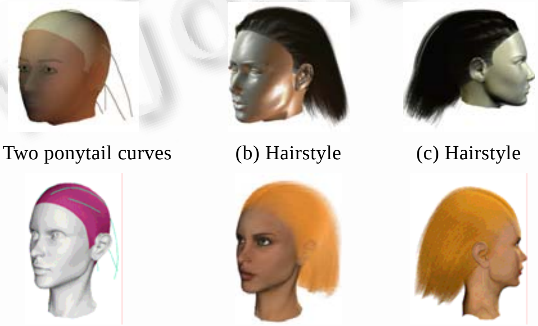


(a) Cutting (b) Curling (c) Drag

Fig.13 Editing operation

3.2 Contrast experiments

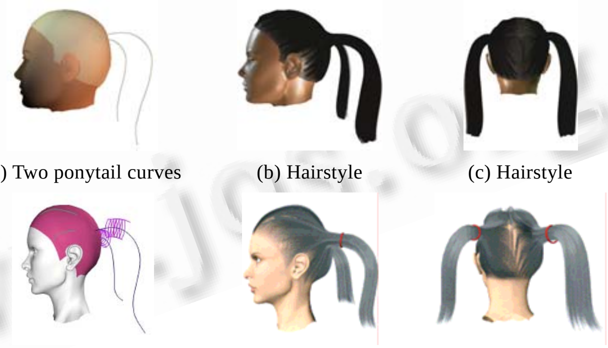
As shown in Figure 14 and Figure 15, we did two contrast experiments to compare out method with Fu[17].



(a) Two ponytail curves (b) Hairstyle (c) Hairstyle

(d) Style curves in Fu[17] (e) Hairstyle (f) Hairstyle

Fig.14 First contrast experiment with Fu[17]



(a) Two ponytail curves (b) Hairstyle (c) Hairstyle

(d) Style curves in Fu[17] (e) Hairstyle (f) Hairstyle

Fig.15 Second contrast experiment with Fu[17]

Fig.14(d), Fig.14(e) and Fig.14(f) are experimental results from Fu[17]. Fig.14(d) shows the style curves sketched by user, it consists of ten stream curves. Fig.14(e) and Fig.14(f) are the result hairstyle at different views. We sketched five straight curves to generate the similar hairstyle (Fig.14(a)). Fig.14(b) and Fig.14(c) show the generated hairstyle at different views. It took about 6 seconds to create this hairstyle.

Fig.15(d), Fig.15(e) and Fig.15(f) are experimental results from Fu[17]. Fig.15(d) shows the style curves sketched by user, it consists of four stream curves and two ponytail primitives. Fig.15(e) and Fig.15(f) are the result hairstyle at different views. It cost 3 minutes to generate the hairstyle. We sketched two ponytail hairstyle curves to generate the similar hairstyle (Fig.15(a)). Fig.15(b) and Fig.15(c) show the generated hairstyle at different views. It took about 8 seconds to create this hairstyle.

The contrast experiments show that we can generate similar hairstyle with much less time cost, besides, our hairstyle curves are easier to sketch.

Figure 16 show three kinds of hairstyles generated by our method. All experiment results prove that our method provides more interaction and computation efficiencies without much lost of hair realism.

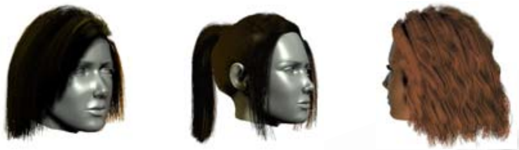


Fig.16 Three kinds of hairstyle generated by our method

4 Conclusions

We present an interactive hairstyling method that combines hair wisp model, vector field based hairstyling and sketch-based hairstyling curves. Users can sketch several different hairstyle curves to generate different kinds of hairstyles. Users can also edit local details of the hairstyle by wisp based editing operations. Experiment results show that our method provide more interaction and computation efficiencies without much lost of hair realism.

However, there are still some issues to be addressed. Comparing to the large variety of hairstyles in real world, we can only create three different types of hairstyles. More kinds of hairstyle curves should be defined in future work. Hair-scalp detection costs too much time during the process of hairstyle generation. We could optimize the time cost by preprocess the scalp model.

References:

[1] Ward K, Bertails F, Kim TY. A survey on hair modeling: Styling, simulation, and rendering. IEEE Trans. on Visualization and Computer Graphics, 2007,13(2):213−234.

[2] Bertails F, Audoly B, Querleux B, Leroy F, Leveque JL, Cani MP. Predicting natural hair shapes by solving the statics of flexible rods. In: Proc. of the Eurographics. 2005.

[3] Bertails F, Audoly B, Cani MP, Querleux B, Leroy F, Leveque JL. Super-Helices for predicting the dynamics of natural hair. ACM Trans. on Graphics (Proc. of the SIGGRAPH 2006), 2006:1180−1187.

[4] Moon JT, Marschner SR. Simulating multiple scattering in hair using a photon mapping approach. ACM Trans. on Graphics (Proc. of the SIGGRAPH 2006), 2006,25(3):1067−1074.

[5] Watanabe Y, Suenaga Y. A trigonal prism-based method for hair image generation. IEEE Computer and Applications, 1992,12(1): 47−53.

[6] Yang XD, Xu Z, Wang T. The cluster hair model. Graphical Models, 2000,62(2):85−103.

[7] Chen L, Saeyor S, Dohi H, Ishizuka M. A system of 3D hairstyle synthesis based on the wisp model. The Visual Computer, 1999, 15:159−170.

[8] Xu Z, Yang XD. V-HairStudio: An interactive tool for hair design. IEEE Computer Graphics and Applications, 2001,21(3):36−42.

[9] Choe B, Ko HS. A statistical wisp model and pseudophysical approaches for interactive hairstyle generation. IEEE Trans. on Visualization and Computer Graphics, 2005,11(2):160−170.

[10] Kim TY, Neumann U. Interactive multiresolution hair modeling and editing. ACM Trans. on Graphics, 2003,21(3):411−418.

[11] Hadap S, Thalmann NM. Interactive hair styler based on fluid flow. Computer Animation and Simulation, 2000:87−100.

[12] Yu. Y. Modeling realistic virtual hairstyles. In: Proc. of the Pacific Graphics 2001. 2001. 295−304.

[13] Mao X, Kato KH, Atsumi A, Anjyo K. Sketch interface based expressive hairstyle modeling and rendering. In: Proc. of the Computer Graphics lnt’l. 2004. 608−611.

[14] Mao X, Isobe S, Anjyo K, Imamiya A. Sketchy hairstyles. In: Proc. of the Computer Graphics Int'l. 2005. 142−147.

[15] Malik S. A sketching interface for modeling and editing hairstyles. In: Proc. of the Eurographics Workshop on Sketch-Based Interfaces and Modeling 2005. 2005. 185−194.

[16] Wither J, Florence B, Marie PC. Realistic hair from a sketch. In: Proc. of the IEEE Int'l Conf. on Shape Modeling and Applications. 2007. 33−42.

[17] Fu H, Wei Y, Quan L, Tai CL. Sketching hairstyles. In: Proc. of the Eurographics Workshop on Sketch-Based Interfaces and Modeling. 2007.

WANG Bin was born in 1985. He is a master degree candidate. His current research areas are computer graphics.

SUN Zheng-Xing was born in 1964. He is a Ph.D. candidate, and a professor and doctorial tutor. His current research areas are computer graphics and computer vision.

ZHANG Yan was born in 1980. She is a Ph.D. candidate. Her current research areas are computer graphics and computer vision.