Pacific Graphics (2011)
Bing-Yu Chen, Jan Kautz, Tong-Yee Lee, and Ming C. Lin (Editors)

An image-based hair styling system using layered texture pieces

Hiroki Watanabe, Yoshihiro Kanamori, Jun Mitani and Yukio Fukui

Department of Computer Science, University of Tsukuba

Abstract

We present an interactive system that allows the user to finely edit a hairstyle image so that the image can be naturally synthesized onto a given facial image. In our system, a hairstyle is composed by stacking texture pieces *(textured belt-shaped polygons) that represent hair wisps. By warping these pieces, the user can perform typical manipulations for hairstyles, such as lengthening/shortening, combing (modification of the flows of hair) and scissoring. We demonstrate the effectiveness of our system through experimental results with various hairstyle images and facial images.*

Categories and Subject Descriptors (according to ACM CCS): I.3.4 [Computer Graphics]: Graphics Utilities—Application packages

1. Introduction

Today people select their hairstyles to express their personality and satisfy their desire to change. As we sometimes experience, when a woman changes her hairstyle she may even look like a different person. Hairstyles can change individual's impression drastically [LaF].

Because hairstyles strongly influence individual's appearance that way, simulating a different hairstyle beforehand is quite beneficial. Here we need to match the hairstyle to a target face to confirm whether the hairstyle fits to the person. An intuitive option to do so is an image-based approach, that is, to select an image of an appropriate hairstyle from a catalogue and then put it on the face image. Additional image warping can adjust the hairstyle to the face shape to some extent. Indeed there are several commercial services [Hen, LLC] of such approach. However, it does not suffice; preset hairstyles cannot fulfill demands subtle but important to express the personality - "*Keep the front long.*" or "*Can you cut the sideburns a little more?*," as a customer asks a beautician at a beauty parlor.

To handle such delicate operations, we present an interactive system for image-based hair styling. In our system, a hairstyle is composed by stacking *texture pieces* (textured belt-shaped polygons) that represent hair wisps. By warping these pieces, the user can perform typical manipulations for hairstyles. Specificially, our system offers the following tools;

Scaling tool: lengthens/shortens hair (Section 3.2).
Combimg tool: modifies the flows of hair (Section 3.3).
Cutting tool: cuts hair wisps (Section 3.4).

Furthermore, our system also supports functionalities that are provided in conventional systems; global warping of a hairstyle image to roughly adjust the face shape [Miy98] and coloring of hairstyle images (Section 3.5). Our system is very simple to implement yet powerful to edit hairstyles even for casual users, as demonstrated in various results and evaluated through user tests (Section 4.1).

2. Related Work

The representation of hair has been actively studied in the field of computer graphics since its early decades because hair greatly enriches the reality of 3D characters. Please refer to a survey paper [WBK*07] for details. Here we introduce recent modeling methods for hair.

There are a number of methods for editing or modeling 3D hairstyles; sketch-based hairstyle creation and editing [FWTQ07, WBC07], polygon-based hair representation for efficient modeling [YSK09] and hair geometry acquisition from real hair [PCK*08,JMM09], to name but a few. For

3D hair styling, Ward et al. [WGL07] proposed a system that allows the user to emulate real-world operations, e.g., misting hair with water, cutting hair with a scissor and drying hair with a blower. Bertails et al. [BAQ*05] presented a new physically-based method for predicting natural hairstyles in the presence of gravity and collision.

These 3D methods have high flexibility in designing hairstyles. To check if a hairstyle fits to a certain face, we require a photorealistic head model including both a face and hair. However, preparation of such head model is not easy; suppose a casual use case where only a few images are available for an individual face and hairstyle. A realistic 3D face model may be automatically constructed from a single image using, e.g., [BV99], but automatic acquitision of a plausible 3D hairstyle model is still difficult even with the state-of-the-art methods [WYZG09, JMM09] and thus manual labors (and modeling skills as well) are required. Another concern is about editing of 3D hairstyle models, which also demands modeling skills even with special devices like Sensable PHANTOM(TM) as used in [WGL07].

On the other hand, image-based 2D hair styling is easy to use and nice to grasp an overview. In Miyata's method [Miy98], a hairstyle image is integrated with a given face image by warping. As a preprocessing, the outline of a face image and the curve in a hairstyle image along the face are specified so that the hairstyle image can be adjusted to match face images. Chen et al. [CZ06] proposed a method that constructs a vector field from a hairstyle image and then warps it to synthesize slightly different hairstyles as, for example, non-photorealistic illustrations. These methods can warp the global shape of a hairstyle, but cannot handle fine editing of hairstyles. Common commercial services such as the virtual preview [Hen] and a typical iPhone application [LLC] share the same drawback.

In our image-based hair styling system, the user can finely manipulate wisps of hair based on a novel representation, *layered texture pieces*. A texture piece represents a hair wisp, and thus our model can be regarded as a 2D variant of 3D hair wisp models [PCP01, BKCN03].

3. Our System

We first describe the overview of our system. Figure 1 shows the overview of our system. The inputs of our system are a hairstyle image and a face image. The hairstyle and the face can be of the same person or another. The face image is requested to provide the entire face with no hidden part. As a hairstyle image, we use an image where regions except hair are erased using photo-retouching software. To construct an editable hairstyle model, the user first extracts belt-shaped small images using mouse strokes. In this paper, we refer to these small images as *texture pieces*. These texture pieces are layered to construct a hairstyle model (Figure 2). The layered texture pieces form the core of our system to enable local manipulation of hairstyles. By making a number of hairstyle models and storing them to a database, the user can select favorite hairstyles from the database.

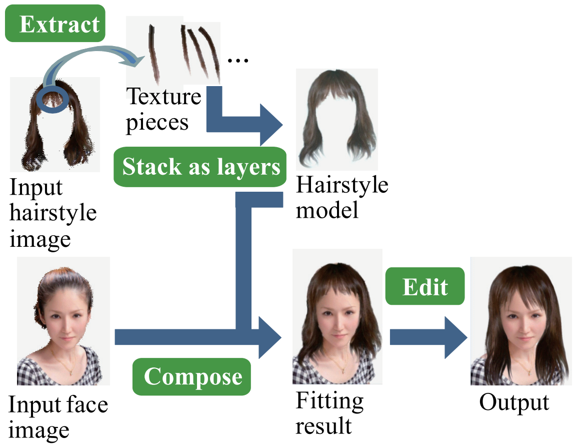


Figure 1: *Overview of our system. Texture pieces are extracted from the input hairstyle image. Then, a hairstyle model is constructed by layering texture pieces and is integrated with the face image. The user can finely manipulate the hairstyle model to satisfy his/her demand.*

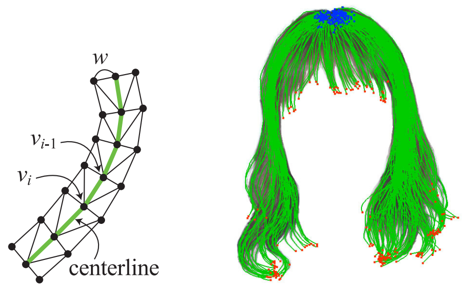


Figure 2: *Texture piece (left, wireframe only) and a hairstyle model consisting of layered texture pieces (right). In the hairstyle model, the centerlines are colored green, top vertices blue and tip vertices red.*

3.1. Texture Pieces

The key idea in our system is the use of what we call a *texture piece*, which is a belt-shaped polygon textured with a hair-wisp image that is extracted from the given hairstyle image. By layering a number of texture pieces while allowing them to overlap each other, our hairstyle model is represented and rendered. Texture pieces enable fine-scale manipulations of hairstyles, as described in the following sections.

A texture piece has a width parameter w, centerline vertices $\{\mathbf{v}_i\}$ $(i = 1, 2, \cdots N$, where N is the number of vertices), and outline vertices. The positions for a pair of left and right outline vertices are calculated by offsetting a centerline vertex $\mathbf{v}_i$ with $\pm w$ in the direction perpendicular to the centerline. Their positions are re-calculated when the texture piece

(a) Before (b) After

Figure 3: *Artifact handling around tips of texture pieces. (a) Blocky artifacts in the red oval can be reduced by adjusting alpha values around tip vertices (b).*

is manipulated. Each vertex in a texture piece has a texture coordinate and an alpha value.

The alpha values are used for matting with a given face image and other texture pieces. The alpha values are set as 1 for the centerline and are attenuated towards the outline. By default, the alpha values at outline vertices are set as 0.8. Note that, if we simply layer texture pieces, blocky artifacts might occur around the tips of hair wisps (Figure 3(a)). To address this issue, we linearly attenuate the alpha values from the vertices at 25% in length from the tips, and set the alpha values of the tip vertices zero.

Extraction of texture pieces is currently done by a manual operation as follows. The user first specifies the width of the piece, and then draws a mouse stroke along the flow of hair. The stroke is smoothed and used as the centerline of a new texture piece. The order of layering of texture pieces is the same as that of extraction.

This manual extraction is the bottleneck in our current work flow, which we wish to eliminate in future work. For example, the time required to construct a hairstyle model is about 15 minutes in the case of the developer of this system. However, we do not think it is critical because this operation is required only once for a single hairstyle model and the model can be reused.

3.2. Scaling Tool

We first explain the scaling tool among the three tools provided in our system. The user first selects texture pieces by drawing a mouse stroke across the centerlines of the pieces. The user can lengthen or shorten the selected pieces by just dragging the mouse downward to the tips or upward to the top of the head, respectively.

To roughly guide the direction of shrinking and lengthening, our system uses a quadratic curve that is fitted to x and y coordinates separately for the centerline vertices using least-squares. We denote the curve as $C(\gamma) = (x(\gamma), y(\gamma))$.

The new vertex positions $\mathbf{v}'_i = (x'_i, y'_i)$ in scaled centerlines are calculated as follows. Let g be the scaling factor calculated as a ratio of the distance from the top vertex $\mathbf{v}_1$ to the start position of the mouse drag and the distance from $\mathbf{v}_1$ to the current position of the mouse drag, then

$$x'_i = x(g\,i) + (x(i) - x_i)\,d \quad (1)$$

(a) Before (b) Scattered (c) Modified

Figure 4: *Modification of growing directions of texture pieces. (b) If we fit a curve to each centerline individually, the growing directions of texture pieces become scattered. (c) This can be modified by adding an extra control point for each piece.*

$$y'_i = y(g\,i) + (y(i) - y_i)\,d \quad (2)$$

where d is a coefficient to control the amount to take into account the differences between original vertex positions (x_i, y_i) and the corresponding points on the fitted curve $(x(i), y(i))$. If the scaling fator g is larger than one (i.e., lengthening) and the differences are allowed to be large, the centerline gradually leaves the fitted curve and becomes winding. Thus we set $d = 1$ if $g > 1$. On the other hand, if g is less than or equal to one (i.e., shortening), we set $d = g$ to make the centerline close to the fitted curve gradually.

Note that naïvely fitting a curve to each centerline causes the pieces to grow to scattered directions when lengthening (Figure 4(b)). To control the growing direction, we add a control vertex $\mathbf{v}_c$ from each tip of a selected piece to the direction of the mouse drag when applying least-squares. Let $\hat{\mathbf{m}}$ be the unit vector that represents the direction of the mouse drag, then

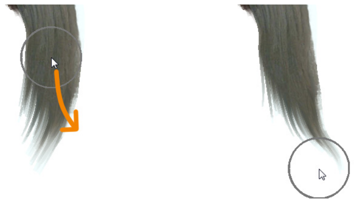
$$\mathbf{v}_c = \mathbf{v}_N + \alpha\,\hat{\mathbf{m}} \quad (3)$$

where α is a weight and $\mathbf{v}_N$ is an end point of the centerline. Note that α should be determined according to the number of vertices in the centerline; whilst the control vertex works well if the number of vertices is small, the control vertex affects little if there are many vertices, unless it is added far away from the tip. Consequently, we set $\alpha = N$ in our implementation. This works well to control the growing direction, as shown in Figure 4(c).

3.3. Combing Tool

Using the combing tool, the user can change the flow of hair along mouse strokes as shown in Figure 5. As the user drags the mouse, the selected texture pieces are warped while keeping their lengths of centerlines.

In the combing tool, a texture piece is selected if its centerline intersects with a circle centered at the current position of the mouse cursor. The centerline vertices $\mathbf{v}_i$ located within the circle are moved to the new positions $\mathbf{v}'_i$ along the direction $\mathbf{m}$ of the mouse drag, while keeping the distance



(a) Before (b) After

Figure 5: *Close-up illustration of the combing tool. The flow of hair is modifed along the mouse stroke (orange).*

between vertices $\mathbf{v}_i$ and $\mathbf{v}_{i-1}$.

$$\mathbf{v}'_i = \frac{(\mathbf{v}_i - \mathbf{v}_{i-1}) + \mathbf{m}}{\|(\mathbf{v}_i - \mathbf{v}_{i-1}) + \mathbf{m}\|} \|\mathbf{v}_i - \mathbf{v}_{i-1}\| + \mathbf{v}_{i-1} \qquad (4)$$

After this operation, $\mathbf{v}_i$ is smoothed by averaging the positions of the previous and next vertices so that the centerline does not become jagged due to hand movement.

3.4. Cutting Tool

The user can cut hair with the cutting tool when the user wants to erase certain parts of a hairstyle. When a mouse stroke crosses texture pieces, we set the alpha values of the region lower than the cross section zero.

This tool works differently from shortening by the scaling tool. Suppose a texture piece whose centerline is straight up to a certain length and then becomes curly. In this case shortening will keep the curled region whilst the cutting tool will cut it off.

3.5. Automatic Fitting of Hairstyle Models and Coloring

Our system also provides automatic fitting of preset hairstyle models to a given face image, similarly to Miyata's work [Miy98]; when a hairstyle model is selected, the hairstyle is automatically warped to roughly fit to the face image. For automatic fitting, two oval-shaped point sets are speficied both for the hairstyle model and the given face image. The oval for the hairstyle model is arranged manually beforehand whilst the oval for the face image is automatically calculated in runtime using the face detection capability implemented in OpenCV. The hairstyle model is stretched or shrinked so that the two ovals coincide with each other. The user can further adjust the hairstyle model by scaling, rotating or translating it based on the oval.

The user can change the color of hair as well. Our system represents the hair image in *Lab* color space. The user can change the chroma of the hair by changing the *ab* color components.

4. Result

Our prototype system was written in C++, using OpenGL, GLUT, GLUI, OpenCV and DevIL. All experiments were

(a) (b) (c)

Figure 6: *Illustration for automatic fitting. The hairstyle in (b) is warped so that the blue oval coinsides with (a) the red oval on the given face.*

(a) (b) (c)

Figure 7: *Editing result with a female face. Face image size is* 1065×1600*, hairstyle image size is* 500×500*, and the number of texture pieces is 200.*

conducted on a PC with Intel Core 2 4300 1.80GHz, an NVIDIA Quadro FX 570 graphics card and 2GByte memory. The user could perform all operations at interactive rates, as demonstrated in the accompanying video.

Figures 7 and 8 show editing results using our system. The time for editing was about five minutes for each. Figures 7(a) and 8(a) are the results of automatic fitting of respective hairstyle images onto face images. For Figure 7, the user cut and evened up the forelock, and gave a volume by spreading the middle of the hair using the combing tool. On the other hand, for Figure 8, the user shortened the hair using the scaling tool (Figure 8(b)), and also lengthened the hair using the scaling tool and changed the flow of the hair (Figure 8(c)).

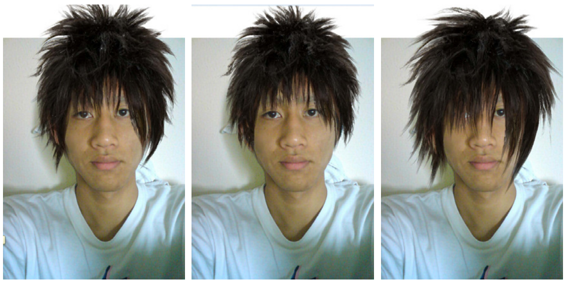
4.1. User Tests

To evaluate the controllability of our system, we conducted user tests where four subjects were requested to edit each given hairstyle image (Figure 9, right) so that it becomes similar to a reference (Figure 9, left). About five-minute instructions, the subjects edited images up to 20 minutes using the three tools until satisfied.

Figure 10 shows the results and the corresponding times required for editing. Overall the subjects success-

(a) Subject #1 (b) Subject #2 (c) Subject #3 (d) Subject #4

Figure 10: *Results of the user test where four subjects were requested to edit the given hairstyle image (Figure 9, right) to make it similar to the reference (Figure 9, left). The times required for editing are; (a) 6 min 21 sec, (b) 9 min 29 sec, (c) 11 min 38 sec and (d) 7 min 59 sec.*



(a) (b) (c)

Figure 8: *Editing result with a male face. Face image size is* 480×680*, hairstyle image size is* 600×600*, and the number of texture pieces is 150.*

Figure 9: *Reference image (left) and a hairstyle image (right) used for the user test.*

fully performed operations such as "separating the forelock", "changing the length of hair wisps" and "making the inward flow of hair to outward" within from five minutes to about ten minutes.

Next we also requested the subjects to select their favorite hairstyles and edit them freely. Also in this case the subjects were given 20 minutes and allowed to edit until satisfied. As shown in Figure 11, the subjects obtained satisfactory results in at most about ten minutes.

Here we introduce the comments from the subjects. Regarding the scaling tool, a subject said, "*This tool was easy to use as I could scale the hair in natural directions.*" Another subject criticized that hair wisps are lengthened unnaturally when the tool is used near the top although it works intuitively around tips. This is probably because we calculate the scaling factor as a ratio of distances calculated from the top vertices and thus the factor changes drastically if calculated near the top. About the cutting tool, subjects said "*It works pretty well as if I were cutting with a scissor.*" and "*If used for finishing after using the scaling tool, the hair wisps were evened up beautifully.*" For changing the flow of hair, the combing tool is important. About this tool, subjects said, "*I could use it intuitively.*" "*The user will use it freely to some extent if the user gets used to.*" On the other hand, there are also critical comments; "*It requires some training to use.*", "*The shape of a wisp can be drastically changed if this tool is used near the top.*" Another important comment is that, the wisps can get clustered when using this tool, which may cause gaps between hair wisps. This problem can be avoided by adjusting the radius of influence of the mouse, by increasing the number of texture pieces, and probably by introducing collision handling for centerlines of texture pieces.

5. Conclusion and Future Work

We have presented an interactive system for editing hairstyle images. The key idea of our approach is to introduce a textured belt-shaped polygon called a *texture piece* to represent a hair wisp, and to model a hairstyle with layered texture pieces to enable partial manipulations of the hairstyle. Specifically, we have presented three tools for fine manipulations; the scaling tool for lengthening/shortening hair wisps, the combing tool for changing the flow of hair and the cutting tool for scissoring hair wisps. As demonstrated in the user tests, our system can be used well even by novice users.

In future work, we would like to develop a method to reduce the manual labor to extract texture pieces from a hairstyle image. Additionally, we would like to handle the shading of hair better; when using the scaling tool, the highlights on hair wisps are also scaled incorrectly in our current system. The illumination differences between the face image and the hairstyle image should be corrected to improve the realism of synthesized results.

(a) Subject #1 (b) Subject #2 (c) Subject #3 (d) Subject #4

Figure 11: *Results of the user test where four subjects freely edited their favorite hairstyles. Upper row: initial settings of selected hairstyles, lower row: edited results. Each column corresponds to the input and the result of a subject. The times required for editing are; (a) 4 min 2 sec, (b) 10 min 23 sec, (c) 7 min 30 sec and (d) 5 min 32 sec.*

References

[BAQ*05] BERTAILS F., AUDOLY B., QUERLEUX B., LEROY F., LÉVÊQUE J.-L., CANI M.-P.: Predicting natural hair shapes by solving the statics of flexible rods. In *Eurographics 05, Short papers, August, 2005* (2005), Eurographics. 2

[BKCN03] BERTAILS F., KIM T.-Y., CANI M.-P., NEUMANN U.: Adaptive wisp tree - a multiresolution control structure for simulating dynamic clustering in hair motion. In *ACM-SIGGRAPH/EG Symposium on Computer Animation (SCA)* (July 2003). 2

[BV99] BLANZ V., VETTER T.: A morphable model for the synthesis of 3D faces. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques* (1999), SIGGRAPH '99, pp. 187–194. 2

[CZ06] CHEN H., ZHU S.-C.: A generative sketch model for human hair analysis and synthesis. *Pattern Analysis and Machine Intelligence, IEEE Transactions on 28*, 7 (july 2006), 1025 –1040. 2

[FWTQ07] FU H., WEI Y., TAI C., QUAN L.: Sketching hairstyles. In *Proceedings of the 4th Eurographics workshop on Sketch-based interfaces and modeling* (2007), ACM, pp. 31–36. 1

[Hen] HENKEL JAPAN CO. LTD.: Virtual Preview. https://virtual-preview.com/. 1, 2

[JMM09] JAKOB W., MOON J. T., MARSCHNER S.: Capturing hair assemblies fiber by fiber. *ACM Trans. Graph. 28*, 5 (2009), 1–9. 1, 2

[LaF] LAFRANCE M.: First impressions and hair impressions study. 1

[LLC] LLC.HASSHOKUKOU: Try hairstyle. http://www.hi-ho.ne.jp/hasshokukou/hairstyle/hairstyle.htm. 1, 2

[Miy98] MIYATA K.: Hairstyle simulation system. In *ACM SIGGRAPH 98 Conference abstracts and applications* (1998), SIGGRAPH '98, p. 278. 1, 2, 4

[PCK*08] PARIS S., CHANG W., KOZHUSHNYAN O. I., JAROSZ W., MATUSIK W., ZWICKER M., DURAND F.: Hair photobooth: geometric and photometric acquisition of real hairstyles. *ACM Trans. Graph. 27* (August 2008), 30:1–30:9. 1

[PCP01] PLANTE E., CANI M.-P., POULIN P.: A layered wisp model for simulating interactions inside long hair. In *Eurographics Workshop on Computer Animation and Simulation (EGCAS)* (sep 2001), Marie-Paule Cani Daniel Thalmann N. M.-T., (Ed.), Computer Science, EUROGRAPHICS, Springer. Proceedings of the EG workshop of Animation and Simulation. 2

[WBC07] WITHER J., BERTAILS F., CANI M.: Realistic hair from a sketch. In *Shape modeling international* (2007). 1

[WBK*07] WARD K., BERTAILS F., KIM T. Y., MARSCHNER S. R., CANI M. P., LIN M.: A survey on hair modeling: Styling, simulation, and rendering. *IEEE Transactions on Visualization and Computer Graphics 17*, 2 (2007), 213–234. 1

[WGL07] WARD K., GALOPPO N., LIN M.: Interactive virtual hair salon. *Presence: Teleoper. Virtual Environ. 16* (June 2007), 237–251. 2

[WYZG09] WANG L., YU Y., ZHOU K., GUO B.: Example-based hair geometry synthesis. In *ACM SIGGRAPH 2009 papers* (2009), SIGGRAPH '09, pp. 56:1–56:9. 2

[YSK09] YUKSEL C., SCHAEFER S., KEYSER J.: Hair meshes. *ACM Trans. Graph. 28* (December 2009), 166:1–166:7. 1