グラフィクスとCAD 99－8
(2000. 4. 28)

Histogram-based Hair Rendering

Waiming Kong[*1] Masayuki Nakajima[*1]

Abstract – **In this paper, we propose a algorithm to render realistic hair using photographs of real hair. Rather then using photographs directly, we create histograms of the hue, intensity and saturation values of the input hair pictures. Based on these histograms, we employ a iterative approach to change the histograms of the hair model to match the histograms of the input photograph. As the histograms of the hair model change, the appearance of the hair in the hair model will approach that of the input photograph. This technique will be useful to render realistic looking hairs for virtual characters.**

Keywords : **hair rendering, histogram, shadow**

1. Introduction

Hair expression for computer graphics is a difficult topic. Not only is it difficult to model and render realistic looking hairs, the sheer number of hair strands present demands a lot of memory and CPU power. However, many researches have been conducted in the past with different degrees of success. Essentially, hair expression can be classified under 3 major categories: rendering, modeling and animation. Hair rendering can be further divided into 2 major areas: texture-based or geometry-based. Tohjo et al.[12] presented a hair model based on anisotropic reflection and bump mapping technique. This is an efficient algorithm making use of 2d texture map to simulate hair. Kajiya[4] extended this idea by using a 3d texture to create a realistic looking Teddy bear using many hours of computer time. The texture based approach to hair rendering suffers a disadvantage as it is difficult to animate hair using texture. Watanabe and Suenaga[13] used trigonal prism and wisp model to render hair to a certain degree of success. However, the trigonal prism looks stiff and needs to be refined for better representation of human hair. LeBlanc, Turner and Thalmann[8] proposed a pixel blending solution to the aliasing problem in hair rendering while Rosenblum, Carlson and Tripp[11] proposed a superantialiasing solution. Nakajima and Agui et al.[1] generated hair image using thread model and area anti-aliasing. Their algorithm divides the hair strand into a number of fragments , which are easily stored and rendered. The hair model is improved by area-antialiasing which depends on the area occupied by the hair fragments. Since every individual hair strand needs to be stored and generated separately, the performance of this model is seriously affected. Nakajima et al.[9] improved upon the thread model by introducing fractional hair model. In this improved algorithm, the generated hair strand is rotated, then translated to produce addition fractional hairs with the same properties as the original hair strand. This group of hair strands is then lumped together and controlled as a single element. It can be seen that control of the hair is traded for processing speed in this method. Kong and Nakajima[7] introduced a fast hair rendering algorithm by classifying the hair into surface and background hairs through the use of a visible volume buffer. Goldman[3] uses fake fur to speed up rendering but it is only applicable for far away objects. The spring model is first proposed by Rosenblum, Carlson and Tripp[11] to provide a fast and simple hair animation model. Anjo, Usami and Kurihara[2] proposed a hair animation algorithm using the cantilever beam simulation. Most hair modeling algorithms concentrated on efficient GUI implementation. Kong and Nakajima[5][6] proposed a hair modeling method through the use of a hair volume which cropped the hair model to give the desired result.

Despite the different approaches toward hair expression, the use of photographs of real hair is not fully explored even though photographs provide a lot of information about the properties of hair. A naive approach will be to extract lines from the input image using image processing and use the information

*1: Graduate School of Information Science and Engineering Tokyo Institute of Technology

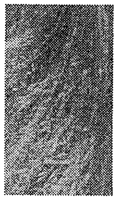


Fig. 1 Hair texture 1

obtained in hair rendering. This approach runs into difficulty in the hair strand recognition process as it is difficult to trace complete hair strands. Texture mapping can be used together with photographs to provide hair rendering but shadow due to hair strands cannot be generated with this approach due to lack of geometry. In this paper, we propose a algorithm to render realistic hair using photographs of real hair. Rather then using the photographs directly, we create histograms of the hue, intensity and saturation values of the input hair pictures. Based on these histograms, we employ a iterative approach to change the histograms of the hair model to match the histograms of the input photograph. As the histograms of the hair model change, the appearance of the hair in the hair model will approach the hair quality of the input photograph.

2. Input hair texture

From photographs of human model, the portion of the photograph that contains the hair is cropped and the cropped photograph forms the input hair picture. The input hair picture will be known as the input hair texture throughout this paper. The size of the input hair texture is not important and can be of any size. Figure 1 is an example of hair texture. From a single hair texture, we obtain the histograms of hue, lightness and saturation. We use the HLS color model as it is more intuitive than the RGB color model. The HLS color model allows the user to think in terms of making a selected hue darker or lighter. A hue is selected with hue angle H, and the desired shade, tint, or tone is obtained by adjusting the L and S. Colors are made lighter by increasing L and made darker by decreasing L. When S is decreased, the colors move toward gray. To avoid confusion with light, the lightness component in the HLS color model will be known as the intensity component for the rest of the paper.

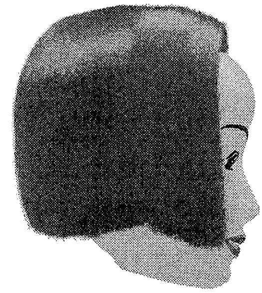


Fig. 2 Initial hair model image

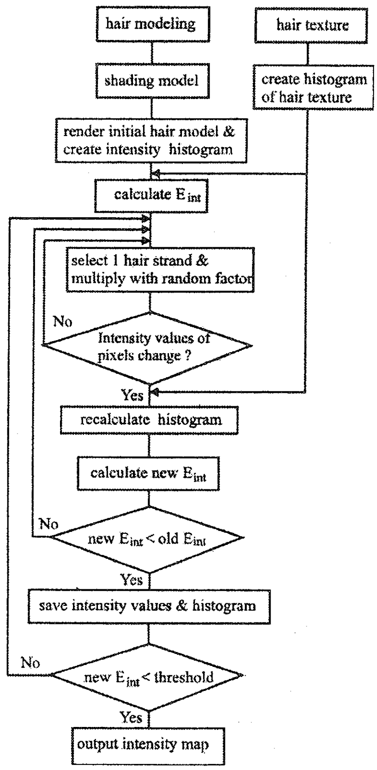
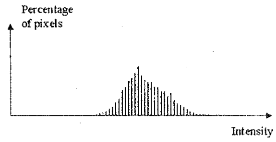


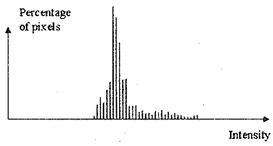
Fig. 3 Intensity flowchart

3. Hair rendering by histograms

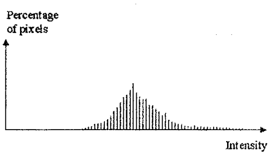
Figure 2 shows a hair model image rendered with Kajiya[4] shading model for intensity with constant hue and saturation values. To improve upon the hair image, we change its histograms toward the histograms of the hair texture. The hair rendering process is completed when the histograms of the rendered hair image approaches the histograms of the



(a) Hair texture intensity histogram

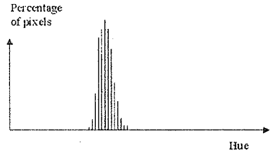


(b) Hair model intensity histogram before iterative change

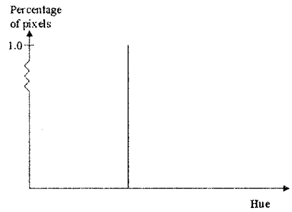


(c) Hair model intensity histogram after iterative change

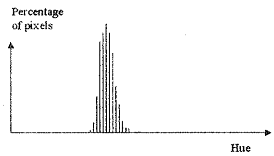
Fig. 4 Intensity histograms



(a) Hair texture hue histogram



(b) Hair model hue histogram before iterative change



(c) Hair model hue histogram after iterative change

Fig. 5 Hue histograms

hair texture. There is a need to process separately for the histograms of the intensity, hue and saturation and combine the outputs to form the final hair image.

4. Intensity histogram

The flowchart to change the intensity histogram is shown in Figure 3. The light intensity which is calculated by

$$I_{light} = \Psi_{specular} + \Psi_{diffuse} + \Psi_{ambient} \qquad (1)$$

where the specular and diffuse components are obtained from Kajiya shading model [4] and ambient light is considered as constant.

In this paper, one hair strand is divided into 15 hair fragments. The hair fragment is rendered onto the screen with superantialiasing and z buffer. For each hair fragment, we project the hair fragment onto the screen and find the pixels which are contained in the hair fragment. The intensity of the pixel is calculated by

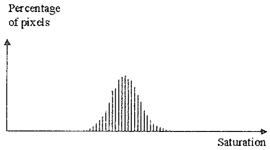
$$I_{pixel} = I_{light} * I_{initial\ intensity} \qquad (2)$$

where I_{light} is calculated by equation (1) and $I_{initial\ intensity}$ is the original intensity given to the fragment. The original intensity for all hair strands is obtained from the median intensity value of the hair texture. This is repeated for all the hair strands to obtain a initial hair image as shown in Figure 2. From the initial hair image, we create the intensity histogram. The error between the hair image intensity histogram and the hair texture intensity histogram is given by

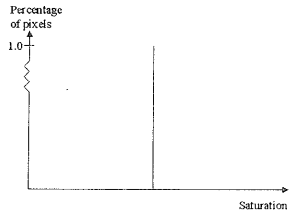
$$E_{int} = \sum_{k=0}^{99}(I(hair\ model)_k - I(hair\ texture)_k)^2 \qquad (3)$$

where $I(hair\ model)_k$ is the percentage of total pixels with intensity k in the hair model image and $I(hair\ texture)_k$ is the percentage of total pixels with intensity k in the hair texture. Figure 4 shows the intensity histograms with the percentage of total pixels as the vertical axis and the intensity value as the horizontal axis.

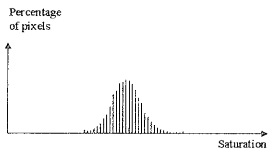
E_{int} is minimized so that the intensity histogram of the generated hair model image approaches that of the hair texture. To minimize E_{int}, one hair strand is randomly selected and the intensity value is multiply by a random factor. The random factor is a random number that is generated to jitter the intensity value by a small amount. The hair strand is then rendered



(a) Hair texture saturation histogram



(b) Hair model saturation histogram before iterative change



(c) Hair model hue histogram after iterative change

Fig. 6 Saturation histograms

with the new intensity value and the pixels affected are calculated. If the intensity values of the pixels in the hair image are changed, the intensity histogram and E_{int} are recalculated. If the new E_{int} is less than the old E_{int}, the intensity values of the hair strands are stored and the same process is repeated by randomly choosing another hair strand. The iterative process is stopped when E_{int} falls below a threshold value. At the end of the process, the intensity histogram of the hair image should approach that of the hair texture and the intensity map which contains the intensity values of all the pixels in the hair image is obtained. Figure 4 shows the intensity histograms of the hair texture and of the hair model image before and after the change in histogram.

4.1 Hue and saturation histograms

The procedure for the hue histogram is similar to the intensity histogram but without the shading model. For the randomly selected hair strand, the hue value is multiplied by a random factor to give new hue value. E_{hue} is calculated as shown in Equation 4 and minimize to produce the desired results.

Table 1 Rendering time for hair model images

	Rendering time
Figure 7	1173s
Figure 8	931s
Figure 10	1485s
Figure 11	1283s
Figure 13	1387s
Figure 14	1113s

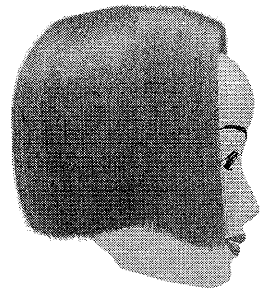


Fig. 7 Image 1 by hair texture 1

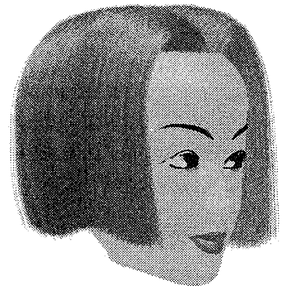


Fig. 8 Image 2 by hair texture 1

$$E_{hue} = \sum_{k=0}^{359} (H(hair\ model)_k - H(hair\ texture)_k)^2 \quad (4)$$

where $H(hair\ model)_k$ is the percentage of total pixels with hue k in the hair model image and $H(hair\ texture)_k$ is the percentage of total pixels with hue k in the hair texture.

The same procedure is applied to the saturation histogram. The hue and saturation histograms are shown in Figure 5 and Figure 6. The hair model image after change in histograms is shown in Figure 7.

5. Results

Figure 2 shows the hair model image rendered with Kajiya shading and common hue and saturation values for all hair strands. The intensity histogram of Figure 2 is shown in Figure 4b. All the pixels in Figure 2 share the same hue and saturation values as shown in the hue and saturation histograms in Figure 5b and Figure 6b. The histograms after the iterative change are much closer to the histograms of the hair texture as shown in Figure 4c, 5c and 5c. Comparing Figure 7 with Figure 2 , we observe that the new hair image reflects the hair quality of the hair texture and the presence of pseudo shadow gives it more realism.

All the generated images are 700x700 pixels images and they are rendered with threshold values of 10 units for E_{int}, E_{hue} and E_{sat} on a Pentium II 400 MHz computer. The hair model consists of 60000 hair strands of radius 0.25 pixels. The rendering time is shown in Table 1. The average time taken to render a image is about 20 minutes. To reduce the rendering time, We can increase the threshold values of E_{int}, E_{hue} and E_{sat}. Figures 7, 8,10, 11,13 and 14 are generated with 3 different types of hair texture chosen to represent the wide variety of hair colors. The histogram approach can generate not only light color hair models but is also capable of producing dark hair model.

Figure 1, 9 and 12 show the input hair textures used to generate the experiment results. In hair texture images, shadows are present in different intensities . This shadow will be passed onto the hair model as the histograms converge in the iteration process. As this shadow is not generated by the light sources of the hair model, this shadow is called "pseudo" shadow. This pseudo shadow helps to give a 3 dimensional look to the hair model and adds realism to the image. The pseudo shadow is evenly distributed in the hair model due to the random selection of hair strand. By cropping the input hair texture, we can control the amount of pseudo shadow present to give desired results.

6. Conclusion

This paper proposes a histogram-based approach to hair rendering. The only input required is a photograph of real hair which is easily available. The rendered hair images bear close resemblance to the hair texture images demonstrating the effectiveness of the proposed algorithm. By carefully selecting and cropping the input texture image, we can modify and obtain any desirable hair image. This technique will be useful to render hairs for virtual characters. We believe that with some modification, the histogram-based rendering can also be used for other natural object rendering such as trees and plants. For future work, we are looking at extracting other useful information from the input hair texture image to be used for rendering. The rendering time for this algorithm should also be shortened for practical application.

Reference

[1] T. Agui, Y. Miwa and M. Nakajima, "Computer animation of the movements of human hair blown by the wind using a stochastic model," IPSJ, Vol 32, No. 6, pp. 749-755, 1991 (in Japanese).

[2] K. Anjyo, Y. Usami and T. Kurihara, "A simple method for extracting the natural beauty of hair," Computer Graphics, 26, (2), pp. 111-120, 1992.

[3] D. Goldman, "Fake fur rendering," Computer Graphics Proceedings, pp. 127-134,1997.

[4] J. Kajiya and T. Kay, "Rendering fur with three dimensional textures," Computer Graphics,23,(3),pp. 271- 280,1989.

[5] W.M. Kong and M. Nakajima, " Generation of hair model from 2D image," The Journal of the Institute of Image Information and Television Engineers, Vol. 52,(9), pp. 128-131,1998.

[6] W.M. Kong and M. Nakajima, " Generation of 3D hair model from multiple images," The Journal of the Institute of Image Information and Television Engineers, Vol. 52,(9), pp. 109-114,1998.

[7] W.M. Kong and M. Nakajima, "Visible volume buffer for efficient hair expression and shadow generation," IEEE Computer Animation, pp.58-65,1999.

[8] A. M. LeBlanc, R. Turner and D. Thalmann, "Rendering hair using pixel blending and shadow buffers," The Journal of Visualization and Computer Animation, Vol. 2, pp. 92-97, 1991.

[9] M. Nakajima, S. Saruta and H. Takahashi, "Hair image generating algorithm using fractional hair model," Signal Processing: Image Communication 9, pp. 267-273, 1997.

[10] W. Reeves, D. Salessin and R. Cook. " Rendering antialiased shadows with depth maps,"Computer Graphics,21,(4), pp. 283-291, 1987.

[11] Rosenblum, Carlson and Tripp, "Simulating the structure and dynamics of human hair : modeling, rendering and animation," The Journal of Visualization and Computer Animation, Vol. 2, pp. 141-148, 1991.

[12] H. Tohjo, M. Miyamoto, K. Murakami and K. Hirota, "Representation of glossy hairs by computer graphics," Technical Report of IEICE, IE89-34, pp. 57-64, 1989.

[13] Y. Watanabe and Y. Suenaga, "Human hair rendering based on trigonal prism and wasp model," IEICE Transaction, Vol. J73-D-II No. 3 pp. 367-373, 1990.

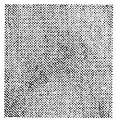


Fig. 9 Hair texture 2

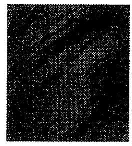


Fig. 12 Hair texture 3

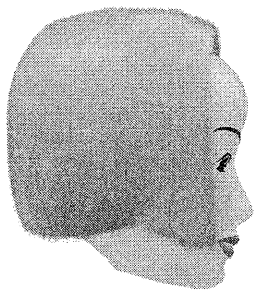


Fig. 10 Image 1 by hair texture 2

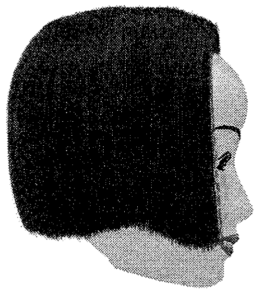


Fig. 13 Image 1 by hair texture 3

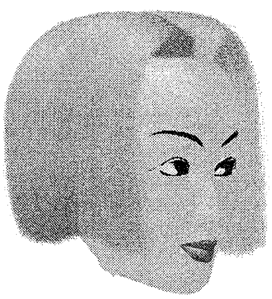


Fig. 11 Image 2 by hair texture 2

Fig. 14 Image 2 by hair texture 3