Vol. 41 No. 3 Transactions of Information Processing Society of Japan Mar. 2000

Regular Paper

Visible Volume Buffer for Efficient Hair Expression and Shadow Generation

Waiming Kong† and Masayuki Nakajima†

Many researches have been conducted in hair modeling and hair rendering with considerable success. However, the immense number of hair strands present means that memory and CPU time requirements are very severe. To reduce the memory and the time needed for hair modeling and rendering, a visible volume buffer is proposed. Instead of using thousands of thin hairs, the memory usage and hair modeling time can be reduced by using coarse background hairs and fine surface hairs. The background hairs can be constructed by using thick hairs. To improve the look of the hair model, the background hairs near the surface are broken down into numerous thin hairs and rendered. The visible volume buffer is used to determine the surface hairs. The rendering time of the background and surface hairs is found to be faster than in conventional hair models by a factor of more than four with little loss of image quality. The visible volume buffer is also used to produce shadows for the hair model.

1. Introduction

As computer graphics find greater applications in the movies, commercials and computer games, there is an urgent need to render realistic looking human characters in ever shorter times. One of the most difficult features of humans to model and render is human hair. Many researches have been conducted in hair modeling[11),12)] and hair rendering with considerable success. However, the immense number of hair strands present means that memory and CPU time requirements are very severe. The memory and time requirements for hair rendering must be improved if realistic looking hairs are to find actual applications.

Many efforts have been made in the past to provide realistic and effective modeling and rendering of human hair. They are broadly divided into 2 major categories: geometry-based or texture-based. Tohjo, et al.[1)] 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 and Kay[7)] 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. Goldman[9)] uses fake fur to speed up rendering but it is only applicable for far away objects. Other researchers approached this problem through the modeling and rendering of thousands of strands of hair each with its own color and geometry. Watanabe and Suenaga[2)] 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. Agui, et al.[3)] 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. As every individual hair strand needs to be stored and generated separately, the performance of this model is seriously affected. Nakajima, et al.[4)] improved upon this thread model by introducing fractional hair model. In this improved algorithm, the generated hair strand is rotated, 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. LeBlanc, et al.[5)] proposed a pixel blending solution to the antialiasing problem in hair rendering while Anjo, et al.[6)] proposed a hair modeling algorithm using the cantilever beam stimulation.

The geometry approached to hair modeling and rendering gives realistic results with thousands of geometric hairs requiring immense memory and computer time. A efficient hair

† Graduate School of Information Science & Engineering, Tokyo Institute of Technology

model must be able to cut down the memory requirement and at the same time reduce the modeling and rendering time. To reduce the memory and the time needed for hair modeling and rendering, a visible volume buffer is proposed. From everyday observation of real human hair models, it can be observed that a human hair model can be considered as a combination of coarse background hair image and detailed surface hairs. Therefore, instead of using thousands of thin hairs, the memory usage and hair modeling and rendering time can be reduced by using coarse background hairs. To improve the look of the background hair model, the hairs near the surface is broken down into numerous thin hairs and rendered. The visible volume buffer is used to determine the surface hairs. The rendering time of the background and surface hairs is found to be faster than a conventional hair model with very little lost in image quality. The same algorithm can be used to generate shadow easily as shown in Section 4. We hope that this algorithm can find application in games and virtual characters on a network where rendering speed is more important than image quality.

2. Visible Volume Buffer

The visible volume buffer is designed to find the "visibility" of any hair or object in the 3D space as seen from the view point. The visible volume buffer is a 3D space buffer that is divided into smaller cubical cells of fixed size. Each cell stores the volume of all the hair strands that are located within the boundary of the cell. The "visibility" of any hair is determined by its opacity value. If a hair fragment has a opacity value of zero, it is not blocked from the view point and is most visible. As the opacity value increases, the hair fragment becomes less visible.

2.1 Construction of Visible Volume Buffer and Hair Classification

The visible volume buffer must be large enough to contain all the objects in its boundary. This condition must be satisfied at all times even when the hair is moving and changes its position in space. The visible volume buffer is further divided into smaller cubical cells of fixed size. The size of the cubical depends on the precision needed and the memory available. All the hair strands are classified based on their position in the real space. The hair strand is divided into small hair fragments and the x, y, z coordinates of the midpoint of the hair fragments are used to determine which of the visible volume cell contains the hair fragment. This is carried out for all the hair strands present and the volume of hair present in the visible volume cells are calculated and stored.

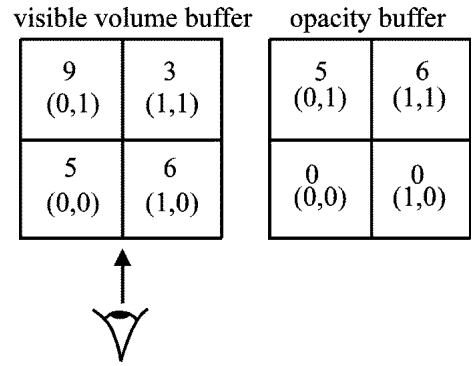

Fig. 1 Formation of opacity buffer.

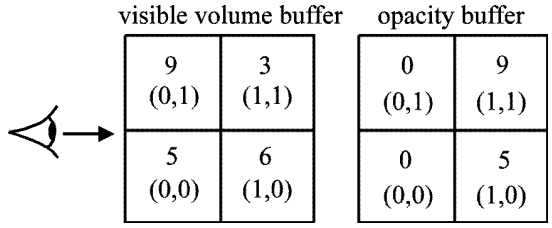

Fig. 2 Opacity buffer after change in view point.

2.2 Opacity Buffer

A opacity buffer is created to determine which of the visible volume cells is visible to the view point. The opacity value of each visible volume cell is the sum of the values of the visible volume cells that lie in a line of sight from the view point to the cell. Note that the visible volume value of the cell in question is not added to the opacity value. The opacity buffer is used to store the opacity values calculated. Consider a 2D visible volume buffer as shown in **Fig. 1**. For the cell (0,0), 5 units of hair are found to be situated in this cell. From the view point as shown in Fig. 1, the opacity buffer (0,0) is given the value of 0 as no object blocks the view point from this cell. For the cell (0,1), the opacity value is 5 as the cell (0,0) lies in the way to the view point. When the view point is shifted, the opacity buffer is changed and need to be recalculated as shown in **Fig. 2**.

3. Background and Surface Hairs

A careful observation of the hairs of human will reveal that it can be divided into 2 layers:

a outermost layer of detailed, fine hairs and a inner layer of background hairs. Based on this observation, we can speed up the modeling and rendering process using the two different types of hair. The visible volume buffer is used to find the outermost layer of hairs.

3.1 Background Hair

The major requirements of background hair are the consumption of less memory, fast in modeling and rendering time and yet provide a rough but accurate image of the hair model. The background hair can be constructed with a geometric or a non-geometric model as long as they satisfy the requirements of the background hair. A good representation of the background hair can be achieved through the use of thick hair. A thick hair approach can reduce the amount of memory needed as we can cut down the number of hair strands tremendously. Due to the reduced number of hair strands, hair modeling and hair rendering are also much faster. In applications where the detail of the hair model is not important such as when the virtual character is far away from the view point, the thick background hair model alone can be used for fast hair modeling and rendering while consuming less memory than a thin hair model.

Although the thick background hair is more efficient than thin hair, when view at close distance, the thick background hair does not look as good as the thin hair. To solve this problem, the thick background hairs at the surface are broken down into a number of thin hairs and rendered.

3.2 Surface Hair

To improve the thick background hair to portray more realism while keeping its advantage of less memory requirement and faster modeling and rendering, the visible volume buffer is used to determine the hairs that are nearest to the view point.

Hair fragments with opacity value of zero are regarded as closest to the view point and we shall call these hairs that are closest to the view point, surface hairs as compared to the rest of the hairs which will be called background hairs.

After finding the surface hairs from the opacity value, the surface hairs are broken down into a fixed number of thin hairs. The transformation of thick hairs into thin hairs is done by first reducing the radius of the hair strand to create a single strand of thin hair. The thin hair created is then translated and rotated by a random figure provided. **Figure 3** shows a strand of thick hair while **Fig. 4** shows the thin hairs generated. The thick hair is transformed into thin hairs just before the rendering process so that memory is not needed to store the thin hairs generated. Therefore, memory usage is kept to a minimum as the thin hairs generated are rendered without storing them in memory.

Fig. 3 One strand of thick hair.

Fig. 4 Breaking into thin hairs.

3.3 Experimental Results

The rendering process is shown in the flowchart in **Fig. 5**. In the visible volume buffer formation process, all the hair strands are divided into smaller hair fragments and the volume of the hair fragment is added to the visible volume cell in which the hair fragment is located. Next, we calculate the opacity values of all the visible volume cells. For any hair strand, if the opacity value of any of the visible volume cells in which its hair fragments are located is 0, it is labeled as a surface hair and broken into thin hairs and rendered. Otherwise, it is labeled as background hair and rendered.

The images of the hair model generated by the background and surface hair model utilized a $100 \times 100 \times 100$ visible volume buffer and the results are shown in **Fig. 6** and **Fig. 7**. Both figures are 700×700 pixels image.

Figure 6 shows the side view of the hair model. The hair model consists of 2222 strands of hair and 279 strands of hair are found to be visible when view from the side. Each visible thick hair of radius of 3.0 units is broken down

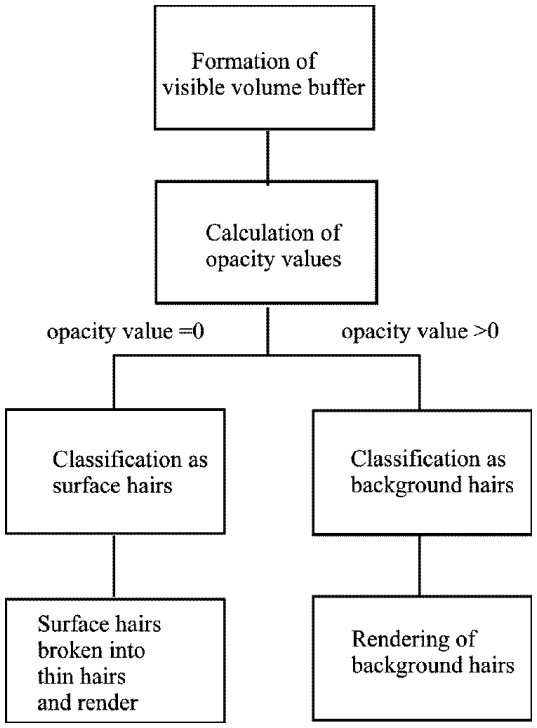

Fig. 5 Rendering flowchart.

into 10 thin hairs with a radius of 2.0 units. If "gap" or "hole" is found in the hair image due to the reduced number of hair strands, we can either increase the number of surface hairs or increase the radius of the background hairs. In Fig. 6, 66 seconds are needed to render 2790 thin hairs while 57 seconds are needed to render 1943 thick hairs on an onyx workstation with 200 MHz R4400 CPU and 500 megabytes of memory. Figure 7 shows the hair model when viewed at an angle. 273 strands of hair are found to be visible resulting in a rendering time of 68 seconds for 2730 thin hairs and 58 seconds for 1949 thick hairs.

The time taken for classification of all the hair strands in the visible buffer is less than 1 second which is small when compared to the total rendering time. Memory requirement for the background and surface hair algorithm is efficient as we need only to store 2222 strands of hair for the hair model found in Fig. 6 and Fig. 7. We only need to render 4733 strands of hair for Fig. 6 and 4679 strands of hair for Fig. 7. To compare our results with a thin hair model, consider that breaking every single thick hair into 10 thin strands will result in a thin hair model of 22220 strands of hair. The images rendered from the 22220 strands thin hair model

Table 1 Rendering time for hair.

Fig.	Number of thin hairs	Number of thick hairs	Time for thin hairs	Time for thick hairs
6	2790	1943	66 s	57 s
7	2730	1949	68 s	58 s
8	22220	0	516 s	0 s
9	22220	0	528 s	0 s
11	2790	1943	67 s	57 s
13	2730	1949	68 s	58 s

are shown in **Fig. 8** and **Fig. 9**. As shown in **Table 1**, the improvement in rendering time for Fig. 6 will be 409 percents over the thin hair model shown in Fig. 8 with very little lost in image quality. Figure 7 records an 419 percents improvement over Fig. 9. Modeling time is also reduced as the number of hair modeled is reduced tremendously. If an aliasing problem is present in the visible volume buffer, we can reduce the aliasing through the use of a averaging filter or use smaller cubical cells. We made use of a z buffer algorithm and Kajiya shading to yield the experimental results.

4. Shadow Generation

The visible volume buffer can also be used to generate shadow easily. By replacing the view point with the light source and utilizing the same visible volume buffer described in the previous sections, we can construct the opacity buffer for shadow. The shadow opacity buffer is constructed in a similar way as the opacity buffer. The shadow opacity value of each visible volume cell is the sum of the values of the visible volume cells that lie in a line of sight from the light source to the cell. Note that the visible volume value of the cell in question is not added to the shadow opacity value. The shadow opacity buffer is used to store the shadow opacity values calculated. If the shadow opacity buffer is zero, the cell is clearly in full view of the light source and is not in shadow. Light intensity of the cell is governed by the equation:

$$I = \begin{cases} 1.0 - obv/st & obv/st < 1.0 \\ 0.0 & obv/st \geq 1.0 \end{cases} \quad (1)$$

where I is the light intensity, obv is the opacity buffer value and st is the shadow threshold.

By adjusting the value of the shadow threshold, we can adjust the amount of shadow present to give desired results. The shadow generated by hair on the head is show in **Fig. 12** while **Fig. 10** shows the head without shadow. Figure 10 is rendered in 131 seconds and Fig. 12 is rendered in 186 seconds. **Figure 11** and

Fig. 6 Background and surface hair image 1.

Fig. 8 Thin hair image 1.

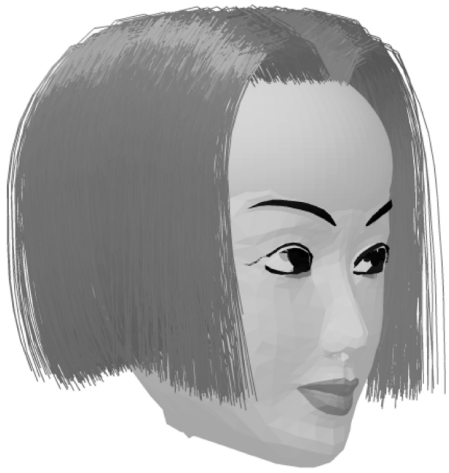

Fig. 7 Background and surface hair image 2.

Fig. 9 Thin hair image 2.

Fig. 13 show the hair model with shadow as compared to the shadowless images in Fig. 6 and Fig. 7.

Past researches on z-buffer based shadow algorithm [5),9)] require the calculation of the projection onto a shadow map for each and every strand of hair. With this method, very accurate shadows can be obtained but the calculations may incur heavy CPU cost. In comparison, our algorithm is much faster as it requires the formation of the visible volume buffer and the opacity buffer just once and it can be used to produce shadow as well as generating the background and surface hairs for fast rendering. The results in this paper are produced with a volume buffer of $100 \times 100 \times 100$. The time needed to render the hairs excluding the head is shown in Table 1. By comparing the times needed for rendering in Table 1, we can see that the visible volume buffer is a efficient method to produce shadows for hair model. To produce shadow in finer details will require the use of a bigger volume buffer and this will increase the memory usage and time needed for rendering accordingly.

5. Conclusion

An efficient model for hair is needed if it is to find applications in multimedia, virtual reality and in entertainment. Polygonal hair may be useful for fast rendering but the quality of the hair generated leaves much room for improvement. To create realistic hair, geometry is needed but the sheer numbers of hair present is the main obstacle hindering its application. The large number of hairs represent large mem-

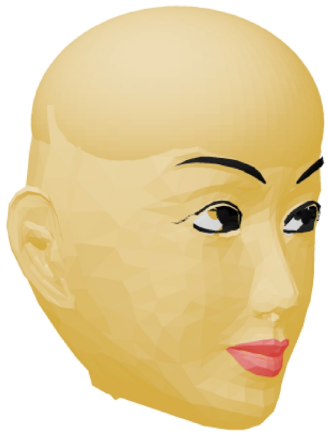

Fig. 10 Head without shadow.

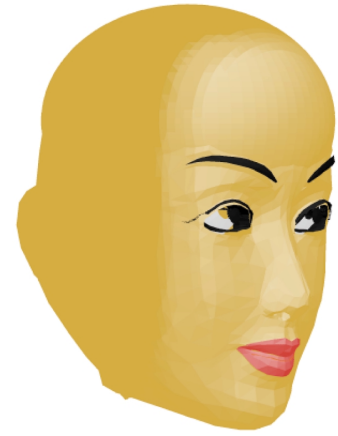

Fig. 12 Head with shadow.

Fig. 11 Hair with shadow 1.

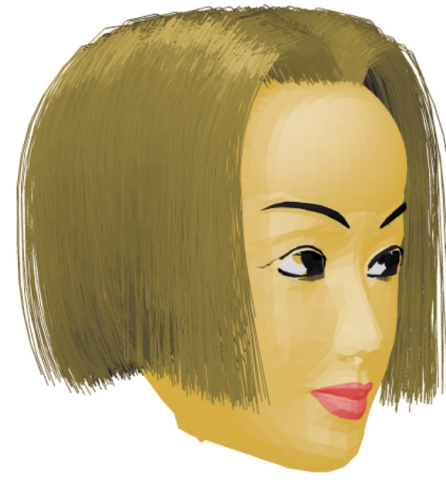

Fig. 13 Hair with shadow 2.

ory requirement and longer time needed for hair modeling and rendering. To alleviate these problems, the visible volume buffer is proposed for fast rendering of hair images with shadow. We believe that this algorithm can produce images for time critical applications such as games and virtual characters on the network. To improve the quality of the hair images generated, the visible volume buffer can be used together with more sophisticated rendering techniques to render realistic looking hair images in faster time.

References

1) Tohjo, H., Miyamoto, M., Murakami, K. and Hirota, K.: Representation of glossy hairs by computer graphics, Technical Report of IEICE, IE89-34, pp.57–64 (1989).
2) Watanabe, Y. and Suenaga, Y.: Human hair rendering based on trigonal prism and wasp model, *IEICE Trans.*, Vol.J73-D-II, No.3 pp.367–373 (1990).
3) Agui, T., Miwa, Y. and Nakajima, M.: Computer animation of the movements of human hair blown by the wind using a stochastic model, *IPS Japan*, Vol.32, No.6, pp.749–755 (1991) (in Japanese).
4) Nakajima, M., Saruta, S. and Takahashi, H.: Hair image generating algorithm using fractional hair model, *Signal Processing: Image Communication*, 9, pp.267–273 (1997).
5) LeBlanc, A.M., Turner, R. and Thalmann, D.: Rendering hair using pixel blending and shadow buffers, *Journal of Visualization and Computer Animation*, Vol.2, pp.92–97 (1991).
6) Anjyo, K., Usami, Y. and Kurihara, T.: A simple method for extracting the natural beauty of hair, *Computer Graphics*, Vol.26, No.2, pp.111–120 (1992).

7) Kajiya, J. and Kay, T.: Rendering fur with three dimensional textures, *Computer Graphics*, Vol.23, No.3, pp.271–280 (1989).
8) Rosenblum, Carlson and Tripp: Simulating the structure and dynamics of human hair: modeling, rendering and animation, *Journal of Visualization and Computer Animation*, Vol.2, pp.141–148 (1991).
9) Goldman, D.: Fake fur rendering, Computer Graphics Proceedings, pp.127–134 (1997).
10) Reeves, W., Salessin, D. and Cook, R.: Rendering antialiased shadows with depth maps, *Computer Graphics*, Vol.21, No.4, pp.283–291 (1987).
11) Kong, W.M. and Nakajima, M.: Generation of hair model from 2D image, *Journal of the Institute of Image Information and Television Engineers*, Vol.52, No.9, pp.128–131 (1998).
[12] Kong, W.M. and Nakajima, M.: Generation of 3D hair model from multiple images, *Journal of the Institute of Image Information and Television Engineers*, Vol.52, No.9, pp.109–114 (1998).

(Received September 3, 1999)
(Accepted February 4, 2000)

Waiming Kong is a doctor-course student in Tokyo Institute of Technology, Japan. His research interests include image processing and computer graphics. He received his B.E. degree and M.E. degree from Tokyo Institute of Technology in 1992 and 1998, respectively.

Masayuki Nakajima received the B.E.E. degree and Dr. Eng. degree from the Tokyo Institute of Technology, Tokyo, Japan, in 1969 and 1975, respectively. Since 1975 he had been with the Department of Imaging Science and Engineering, Tokyo Institute of Technology, Yokohama, Japan. He is now a professor at the Department of Computer Science, the Faculty of Graduate School of Information Science & Engineering, Tokyo Institute of Technology, Tokyo, Japan. His fields of interest are computer graphics, pattern recognition, image processing and virtual reality. He received the best author award from IPSJ in 1998.