A Virtual 3D Hair Reconstruction Method from a 2D Picture

Weng Zufeng[1] Wang Shoujue[1,2,3] Huang Lianfen[1]

[1] School of Information Science and Technology, Xiamen University
Xiamen, China 361005
devil0591@126.com
lfhuang@xmu.edu.cn

[2] Institution of Semiconductors, CAS
Beijing, China 100083

[3] Suzhou Institute of Nano-Tech and Nano-Bionics, CAS
Suzhou, China 215123

Abstract. In this paper, we present a method for reconstructing virtual 3D hair from a 2D human picture. The proposed method only needs the user to provide a 2D positive picture as the reference texture. A canny operator combined with region hair color is used to detect the edge of the hair picture. The three ellipsoidal head modal axis parameters are then calculated from the edge contour. A multi-triangle mesh is constructed by adding point sets on the edge and interior. After that the Cartesian-triangular coordinate transform is used and the corresponding depth of each texture pixel is given to obtain the initial 3D hair shape. In order to simulate a multi-level real hair strip effect we use a label map to simulate hair strip growth from the edge of the given 2D positive picture. A B-Spline function is then applied to refine the preliminary hair strips. Finally 3D hair with the texture features of the given picture is modeled. The experimental results show that this method can be used to model 3D hair strips from a 2D picture with very few manual operations, presenting good stereo vision effect.

Keywords: virtual 3D hair, 2D positive picture, Canny operator, edge contour, Cartesian-triangular coordinate transform, label map, B-Spline function, hair strips

1 Introduction

With the rapid development of computer progress in hardware and algorithms, virtual character modeling has become an intense research direction, playing more important roles in game, film, and animation development [1]. In three-dimensional computer animation, creating vivid realistic characters involves texture mapping, attitude description, ray tracing, motion analysis building models and many other aspects [2]. Hair is the most important decoration in addition to the facial features on the human head, which reflects the personality characteristics of the human model [3]. Different hair styles on the same characters can produce different vision effects, which can help to improve 3D animation viewing.

Human virtual hair developed using computer simulation has always been very challenging work because real hair strips present complex geometric structures composed of hundreds of thousands of single hairs. Each single hair can be viewed as a curve in three-dimensional space. The diameter of a single hair strip is usually between 0.05mm and 0.2mm, less than the common monitor pixel size, making it difficult to accurately display. Completely simulating all hair strips requires describing each single hair curve, which would produce a large amount of data, requiring numerous complex manual operations. Therefore, a simple intuitive method for modeling 3D virtual hair with real sense of vision effect at low manual operation is needed.

The current 3D virtual hair modeling methods are very rich. The most widely used 3D modeling software programs, such as Maya, 3DsMax and Poser etc. can produce realistic 3D hair. However these software programs are expensive and require large numbers of manual operations. The 3D hair model has a complex data structure requiring the modeling personnel to invest a long time in mastering the 3D modeling techniques. In addition, the current 3D virtual hair modeling approach emphasizes artistic sense and the modeling process is similar to painting on a canvas by manually defining the shape, color and direction of the key hair strip to obtain the final virtual hair.

This paper presents a virtual 3D hair reconstruction method directly from a 2D human picture. The proposed method only needs the user to provide a 2D positive picture as the reference texture. A canny operator combined with region hair color is applied to detect the 2D hair image edge. The three ellipsoidal head modal axis parameters are calculated from the edge contour. A multi triangle mesh is constructed by adding point sets to the edge and interior. After that the triangular coordinate system transform is used and the corresponding depth of each texture pixel is given to obtain the initial 3D hair shape. In order to simulate a multi-level real hair strip effect, we use a label map algorithm to simulate hair strip growth from the edge of the given 2D positive picture. The B-Spline function is then applied to refining the preliminary hair strips. The 3D hair with texture features from the given human picture is finally modeled and viewed from any observation angle.

2 Related Research

In modeling virtual 3D hairs, due to the complexity of real hair, the geometric and physical characteristics [4] should be fully considered. Both domestic and overseas researchers have made fruitful research into virtual 3D hair modeling.

Li [5] presented methods for generating photo-realistic 3D human head models from a geometric model and recovered the 3D positions of hair contour points extracted from multiple view color images. The growth root, direction and length of every hair were then grown until the entire scalp is covered. A Coons-patch is finally constructed to approximate the hair surface from four closed 3D contour curves. Feng [6] obtained the hair cluster section by generating a closed B-spline curve and then adjusting the key hairs and the cluster section using the local B-spline properties. Huan [7] adopted the Delaunay algorithm for triangle meshes on the scalp surface and extended the curve outward from each sub triangle center to simulate hair growth using manual operations. A simple cloning was finally performed in each sub triangle to cover the scalp with hair by increasing the hair density. Chai [8] presented a hair modeling method using surface grid control to express the basic hair appearance using additional shape grids. The hair trajectory is then generated out of the grids to obtain the hair-style for the fitting shape.

Luo [9] proposed a multi-view hair reconstruction algorithm based on orientation fields with structure-aware aggregation. The structure aware aggression to the MRF matching energy was applied in their paper to enforce the structural continuities implied from the local hair orientations. Multiple depth maps from the MRF optimization were then fused into globally consistent hair geometry with a template refinement procedure. Li [10] proposed a level-of detail model to improve hair rendering efficiency. The model considered simplification on both strand density and strand curve control points, which are commonly used uniquely by traditional solutions. The hair strands can be merged into hair wisp models through K-D tree searching for the nearest control points. Takeyuki [11] described a new hair representation technique for drawing 3D CG characters. Hair models were represented using skeleton-based implicit surfaces, which allowed representing the union and division of hair wisps using a blobby model. The skeleton was interpolated using a spline function and then an implicit hair shape surface was generated by the skeleton. Finally, non-photo realistic rendering was applied to generate the resulting image as output. Li [12] introduced an approach to capture real-life hair using Kinect sensor and digital single-lens reflex (DSLR) cameras to reconstruct 3D hair models using the particle system, which collected four views of point clouds and high resolution images for real-life hair.

3 Hair Edge Detection

The texture of the front side of the 3D hair is derived completely from the photos and the hair image is required to carry out the edge detection to dig out the whole 2D hair picture. We used the Canny operator for edge detection. The main steps are as follows:

1). For each pixel in the image $I(x,y)$, using he finite difference of the first order partial derivative to calculate the horizontal directional derivative $G_x(x,y)$ and the vertical directional derivative $G_y(x,y)$. After that, the gradient magnitude $\nabla f = \sqrt{G_x^2 + G_y^2}$ is calculated by applying the directional derivative;

2). Non maximum suppression is carried out for each pixel of the edge of the 2D hair. In the neighborhood of 3×3, if $ABS(G_x(x,y)) \geq ABS(G_y(x,y))$, which means the direction of derivative is tending to component x. Otherwise, if $ABS(G_x(x,y)) < ABS(G_y(x,y))$, which means the derivative direction tends toward component y;

3). According to the second step, the $G_x(x,y)$ and $G_y(x,y)$ component proportion is interpolated. If the gradient magnitude of the current pixel is bigger than the gradient magnitude of the adjacent two pixels along the gradient direction, it is considered that the current pixel may be the edge point, and the gray value is marked as 128. Otherwise, the pixel is not the edge point and the gray value is marked as 0;

4). After the third step the 2D hair image is processed using a double threshold value. The aim is to eliminate the false edges and connect the discontinuous edges. The high threshold value and low value threshold are used to calculate the dual threshold coefficient of the image histogram. The high threshold value and the non-maximum suppression of the pixels are compared, marked as the possible edge points. For all edge points the point that is bigger than the low threshold value is iteratively searched in the 8 neighborhood.

The key point in making a 2D hair image into 3D virtual hair lies in assigning the depth value to the pixel, so the 3D model of the head has to be estimated. This paper adopts an ellipsoid to simulate the support head-form. Ordinary hairstyles can be regarded as an extension of the head scalp shape, which is close to the scalp. According to the 2D hair image edge detection results the three axis parameters of the 3D head-form ellipsoid is simulated. All points of the 2D hair image edge are traversed, the three coordinate points of the most left, the most right and the most top are calculated, respectively, marked as $A(A_x, A_y)$, $B(B_x, B_y)$, $C(C_x, C_y)$, and the central coordinates of the ellipsoid can be calculated as $O(O_x, O_y, O_z)$, where $O_x = \frac{A_x + B_x}{2}$, $O_y = \frac{2C_y - A_y - B_y}{2}$, $O_z = 0$. Assuming that the length of three ellipsoid axes are marked as R_x, R_y, R_z, where $R_x = \left|\frac{B_x - A_x}{2}\right|$, $R_y = \left|C_y - O_y\right|$, $R_z = R_x$.

After the above four steps we can obtain the edge image of the whole 2D human head picture. However, this edge image contains the 2D hair edge and also the face and facial features edges. Further processing is therefore needed to filter them. The processing methods are shown as follows:

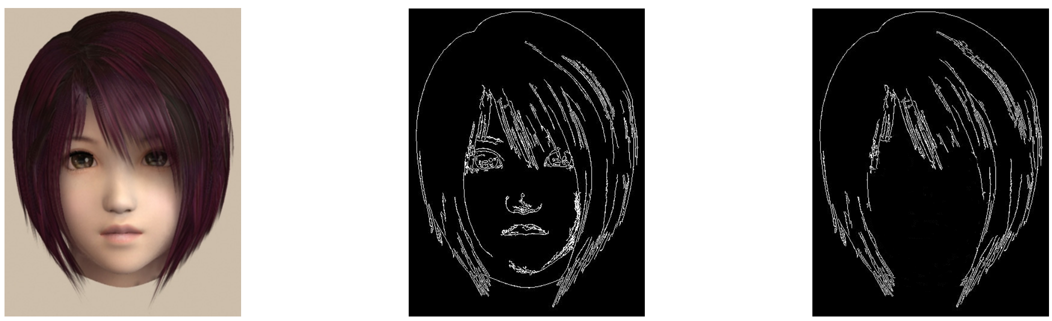
1). Marking the positions of the two eyes by manual, marked as $P(P_x, P_y)$ and $Q(Q_x, Q_y)$, setting the radius around the eyes as r, then eliminating the edge points within the radius r;

2). Selecting three points within the edge image, which are the right side of the left-most point $A(A_x, A_y)$, the left side of the rightest point $B(B_x, B_y)$ and the below side of the top point $C(C_x, C_y)$, then the average RGB value of these tree points are calculated, marked as the hair color $clrHair(R_1, G_1, B_1)$;

3). Traversing the edge image, for each point, getting its color $clrCurrent(R_2, G_2, B_2)$ from original RGB image. The distance between $clrHair$ and $clrCurrent$ is calculated, marked as:

$$s = 1 - \frac{R_1 \cdot R_2 + G_1 \cdot G_2 + B_1 \cdot B_2}{\sqrt{R_1 \cdot R_1 + G_1 \cdot G_1 + B_1 \cdot B_1}} + \frac{\sqrt{(R_1 - R_2) + (G_1 - G_2)\cdot(B_1 - B_2)\cdot(R_1 - R_2) + (G_1 - G_2)\cdot(B_1 - B_2)}}{255\sqrt{3}} \tag{1}$$

When s is less than the given threshold, it is judged to be the edge point of the hair. Otherwise, the current edge point is eliminated. The vision effect of these processing methods can be shown as follows:



(a) (b) (c)

Fig. 1. The edge detection of the given positive 2D hair

4 Multi Triangle Mesh and Pixel Depth

In the last section the 2D hair image edge detection is realized, which provides a texture reference for the 3D virtual hair. However, as shown in figure 1, the edge detection is not able to maintain the full continuity of the edge, and cannot obtain the texture information from the 2D hair. According to the 2D hair edge information the points are set in the corresponding position of the source hair image, and the multi triangle mesh is connected using the given points set. In order to cover the main 2D hair texture region the mesh points are adjusted.

The multi triangle mesh effect occurs when the vertex position of a certain sub-triangle changes, the internal pixels will change accordingly, so the local 2D hair image deformation is realized. Figure 2 shows the conversion between the triangular vector coordinate the Descartes coordinate:

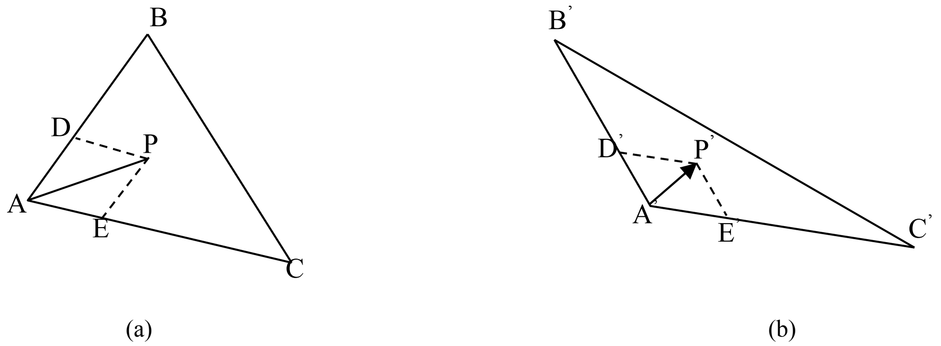


Fig. 2. Triangular-Cartesian coordinates system transform

As shown in figure 2, according to the law of parallel quadrilateral, there is $\overrightarrow{AP} = \overrightarrow{AD} + \overrightarrow{AE}$, assuming $u = \frac{\overline{AE}}{\overline{AC}}$, $v = \frac{\overline{AD}}{\overline{AB}}$ as the triangular coordinate of point P inside the ΔABC, we can obtain:

$$\overrightarrow{AP} = u \cdot \overrightarrow{AC} + v \cdot \overrightarrow{AB} \quad \textbf{(2)}$$

The left side and right side of equation (2) respectively, multiply $\overrightarrow{AB}$ and $\overrightarrow{AC}$:

$$\begin{cases} \overrightarrow{AP} \cdot \overrightarrow{AB} = u \cdot \overrightarrow{AC} \cdot \overrightarrow{AB} + v \cdot \overrightarrow{AB} \cdot \overrightarrow{AB} \\ \overrightarrow{AP} \cdot \overrightarrow{AC} = u \cdot \overrightarrow{AC} \cdot \overrightarrow{AC} + v \cdot \overrightarrow{AB} \cdot \overrightarrow{AC} \end{cases} \quad \textbf{(3)}$$

Solving simultaneous equations (3):

$$\begin{cases} u = \dfrac{\left(\overrightarrow{AP} \cdot \overrightarrow{AB}\right)\left(\overrightarrow{AB} \cdot \overrightarrow{AC}\right) - \left(\overrightarrow{AP} \cdot \overrightarrow{AC}\right)\left(\overrightarrow{AB} \cdot \overrightarrow{AB}\right)}{\left(\overrightarrow{AC} \cdot \overrightarrow{AB}\right)\left(\overrightarrow{AB} \cdot \overrightarrow{AC}\right) - \left(\overrightarrow{AC} \cdot \overrightarrow{AC}\right)\left(\overrightarrow{AB} \cdot \overrightarrow{AB}\right)} \\ v = \dfrac{\left(\overrightarrow{AP} \cdot \overrightarrow{AB}\right)\left(\overrightarrow{AC} \cdot \overrightarrow{AC}\right) - \left(\overrightarrow{AP} \cdot \overrightarrow{AC}\right)\left(\overrightarrow{AC} \cdot \overrightarrow{AB}\right)}{\left(\overrightarrow{AC} \cdot \overrightarrow{AB}\right)\left(\overrightarrow{AB} \cdot \overrightarrow{AC}\right) - \left(\overrightarrow{AC} \cdot \overrightarrow{AC}\right)\left(\overrightarrow{AB} \cdot \overrightarrow{AB}\right)} \end{cases} \quad \textbf{(4)}$$

Using the triangular coordinate u and v, the formula for calculating the Cartesian coordinate is:

$$P\left(P_x, P_y\right) = A + u \cdot \overrightarrow{AC} + v \cdot \overrightarrow{AB} \quad \textbf{(5)}$$

As shown in figure 2 when the ΔABC shape is changed to $\Delta A^{'}B^{'}C^{'}$ the triangular coordinate values of the points both inside and on the side of the given triangle will not change while the Cartesian coordinate values change. The triangular coordinates of point P and point $P^{'}$ are the same as formula (5), which can be used to calculate the different Cartesian coordinates. The positive side of the 2D hair image can be locally deformed by applying these formulas.

However, because the user only provides a single positive 2D photo, which only contains the front half part texture of the given hair image, the back half part of the 3D hair is completely without texture support. We assume that the 3D shape of the back part of the hairstyle is expressed in accord with the ellipsoidal head-form. The multi triangle mesh is extended, such as taking the back half part of the ellipsoidal depth value less than O_z, and mirror mapping the mesh of the front half part to the back of the virtual 3D head-form. The remaining 2D points are randomly completed within the elliptical interface.

According to the third section in this paper the 3D coordinate of the elliptical center point is $O\left(O_x, O_y, O_z\right)$, and the tree axis parameters are R_x, R_y, and R_z. Setting a certain point $P\left(P_x, P_y\right)$ of the 2D hair texture, according the formula of ellipsoid:

$$\left(\frac{P_x - O_x}{R_x}\right)^2 + \left(\frac{P_y - O_y}{R_y}\right)^2 + \left(\frac{P_z - O_z}{R_z}\right)^2 = 1 \quad (6)$$

The corresponding depth coordinate value P_z can be calculated. Traversing all pixels of the 2D hair texture inside the edge contour, according to the front and back sides of the given two multi triangle mesh, the preliminary 3D hair-style can be obtained by combining with the depth value.

5 The Refinement of the 3D Hair Strips

The above methods obtain the preliminary virtual 3D hair with stereo vision effect. The texture of the preliminary virtual 3D hair reflects the hair style image of the given human positive photo. The three-dimensional shape of the preliminary virtual 3D hair meets the ellipsoidal head form. However, common human hair has a real sense of vision effect and is not a single smooth surface, but has hair strips in multiple levels. These multiple hair strips include shape characteristics from attachment to the scalp, uneven surface thickness, irregular hair strips detail distributions and so on.

Given a two-dimensional photo, the human eyes can easily discern the different depth levels in the hair texture details. However in computer graphics, the depth information cannot be extracted from a single two-dimensional photo. The hair strips can be simplified into many three-dimensional space curves. To achieve a more realistic vision effect, detailed 3D hair strip refinement methods can be realized as follows:

(1). Adjusting low thresh value and high thresh value of the Canny operator, and using methods shown in chapter 3 to obtain more edge information inside the multi-triangular mesh, which can reflect the detailed changes in the two dimensional hair texture;

(2). Using a label map the same size as the original two-dimensional photo and initializing all data with '0', then marking all edge points as '1';

(3). For each point marked as '1', traversing three directions such as down, down-left, down-right in its eight-neighborhood, when meeting '1', changing it to '2', then recording its 3D coordinates in an array named $Array1$;

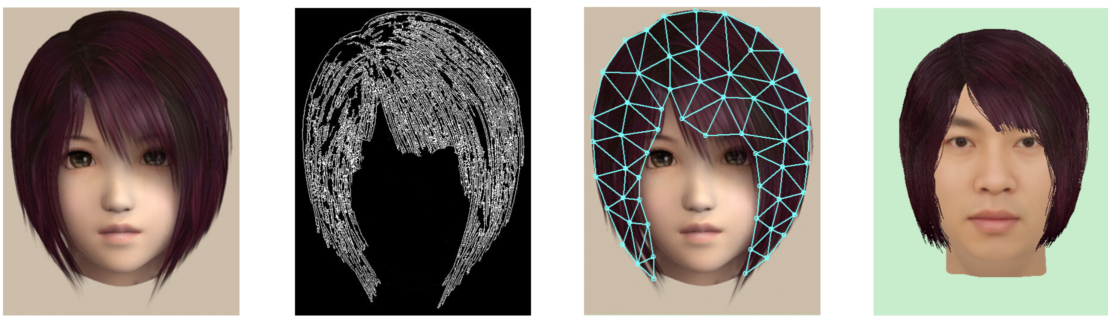
(4). Assuming that $Array1$ has N 3D coordinates, in order to make these hair strips continuity, every $\frac{N}{6}$ key shape-points of 3D coordinates should be get from $Array1$ together with the start 3D coordinates, were restored in a new array named $Array2$;

(5). By now, there are 7 coordinates in $Array1$, which can be used as shape-points of the relevant B-Spline functions [13][14][15] to adjust the shape of the single hair strip;

(6). With the next point marked as '1', repeat from step (3). Jump over step (4) and step (5) when meeting '2', and then repeat from step (3) again;

(7). The processing ends when there is no point marked as '1' in the label map.

The whole processing from 2D positive photo to virtual 3D hair with strips is shown as below:



(a) (b) (c) (d)

Fig. 3. Whole processing from 2D positive photo to virtual 3D hair

From Fig. 3 we can see the whole process from 2D positive photo to virtual 3D hair. Fig. 3(b) shows the hair edge image of Fig. 3(a), which indicates the change in texture of the given hair. From Fig. 3(c) and Fig. 3(d), we can see that the 3D hair with many hair strips has the same positive texture image as Fig. 3(a). Furthermore, 3D hair observed from multiple perspectives is shown in Fig. 4.

(a) (b) (c) (d)

Fig. 4. 3D Hair observed from multiple perspectives

In comparison with previous papers [9][12], we used only a positive 2D human picture rather than multiple view hair capture or Kinect sensors and DSLR cameras to obtain the whole 3D hair. We also used very few manual operations during the proposed method (only in triangle mesh adjusting). It has to be pointed out that the whole back half of the 3D hair was created with no image support and created totally from the information from the front half of model.

6 Conclusion

We proposed a method for reconstructing 3D hair from a given 2D human positive photo. The Canny operator together with region hair color is used to detect the edge contour of the hair image. The tree axis parameters of the ellipsoid head model are then calculated to support such hair. The preliminary 3D hair is obtained using two positive and negative multiple triangle mesh sets. The preliminary 3D hair refinement is applied using a label map algorithm together with the relevant B-spline functions. The final 3D hair is then obtained with multi-level hair strips. The experimental results show that only a single positive photo is needed by the proposed method, which can obtain virtual 3D hair with realistic photo visualization using very few manual operations.

Our method is however only suitable for the positive covered hair condition and focuses on reducing the number of manual operations and hair strips rather than beautiful visualization. We will continue to improve this method in future research.

References

[1] Rahman Rabiatul Adawiyah Abdul, Bade Abdullah, "Geometric modeling in 3D computer graphics and visualization," in *Proceedings of Computer Games and Allied Technology 08 (CGAT 08) - Animation, Multimedia, IPTV and Edutainment*, pp. 103-108, 2008.

[2] B. Yosuke, B. Y. Chen, N Tomoyuki, "Animation hair with loosely connected particles," *Computer Graphics Forum*, Vol. 22, No. 3, pp. 411-418, 2003.

[3] N. Ducheneaut, M. H. Wen, N. Yee, et al., "Body and mind: a study on avatar personalization in three virtual worlds," in *Proceedings of the 27th International Conference on Human Factors in Computing Systems*, New York: ACM Press, pp. 1151-1160, 2009.

[4] W. Kelly, B. Florence, T. Y. Kim, "A survey on hair modeling: Styling, simulation and rendering," *IEEE Transactions on visionization and Computer Graphics*, Vol. 13, No. 2, pp. 213-233, 2007.

[5] X. L. Li, H. B. Zha, "Realistic Human head modeling with hairstyle reconstruction," *Journal of Computer-Aided Design & Computer Graphics (in Chinese)*, Vol. 18, No. 8, pp. 1117-1122, 2006.

[6] C. F. Feng, Y. M. Jiang, C. Deng, "The modeling of real hair based on B-spline," *Microelectronics & Computer (in Chinese)*, Vol. 25, No. 5, pp. 161-164, 2008.

[7] N. V. Huan, D. N. Toan, "Morphing and repairing a 3D scalp geometry in building hair models and simulation," in

Proceedings of the 2009 2nd International Congress on Image and Signal Processing, pp. 305-311, 2009.

[8] M. L. Chai, Y. M. Weng, Q. M. Hou, Z. Ren, "Surface Mesh Controlled Fast Modeling," *Journal of Computer-Aided Design & Computer Graphics (in Chinese)*, Vol. 24, No. 8, pp. 975-983, 2012.

[9] L. J. Luo, H. Li, S. Paris, T. Weise, M. Pauly, S. Rusinkiewicz, "Multi-View Hair Capture Using Orientation Fields," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1490-1497, 2012.

[10] K. Li, W. X. Jin, G. H. Geng, Y. Han, "Level-of-detail modeling with artist-defined constrains for photorealistic hair rendering," *International Conference on Virtual Reality and Visualization (ICVRV)*, pp. 225-228, 2013.

[11] T. Sakai, V. Savchenko, "Skeleton-based anime hair modeling and visualization," in *Proceedings of International Conference on Cyberworlds (CW)*, pp. 318-321, 2013.

[12] Z. R. Li, C. Sun, W. S. Lee, I. J. Kim, "Hair modeling using Kinect sensors and DSLR cameras," *International Conference on Ubiquitous Robots and Ambient Intelligence (URAI)*, pp. 22-27, 2014.

[13] F. Z. Shi, *Computer Aided Geometric Design and Non-Uniform Rational B-spline*, Beijing: Higher Education Press, 2001.

[14] F. H. Tseng, H. H. Cho, L. D. Chou, H. C. Chao, "Efficient power conservation mechanism in spline function defined WSN terrain," *IEEE Sensors Journal*, Vol. 14, No. 3, pp. 853-864, 2014.

[15] A. Sakatoku, T. Osada, G. Kitagata, N. Shiratori, T. Kinoshita, "3D symbiotic environment for agent-aided collaborative work," *Journal of Internet Technology*, Vol. 13, No. 1, pp. 127-136, 2012.