ABSTRACT

Contour rendering is a hot issue in Non-photorealistic rendering research, the method in this paper used mathematical methods to achieve contours on the basis of the traditional method of free-form surfaces on the body. This method according to the principle of stylized contour rendering increased the right amount of contour lines in their neighborhood through detecting the pixels on contour lines to thickening and widening contour lines, and further improved the stylized rendering of contour lines by using texture mapping. And the method achieved texture stylized rendering without re-filling the gap between characteristics.

KEYWORDS
Computer Graphics, Non-Photorealistic Techniques, Contour Detection, Stylized Rendering.

INTRODUCTION

The importance of contour detection and rendering in non-photorealistic rendering is self-evident. When we describe the general contour and shape of the model, we first try to express its appearance through the contour of abstract objects. Especially in the simulation of real scenes, we first draw the basic contour of the object and then do the subsequent rendering. In the non-photorealistic rendering technology, we must firstly extract the contour of the object model and then the subsequent rendering should be done with the extracted contours, that is the stylized contour rendering, which aims at highlighting the global outline and the local key features, and overlooks some information that is not important and simplifies the contour lines.

In recent years, the international research achievements have been gotten in non-photorealistic technology, like stylized contour drawing, which was firstly proposed by Northrup and Markosian, uses 2D texture mapping procedure of different material to realize stylized rendering[1]. Sousa proposes a stroke similar to the style of ink painting, which simulates different styles by drawing information such as materials, tones and textures[2]. Brooks proposes a sketch-style rendering technique, which divides the input image into parts according to the local details of the image, and then processes each area independently using the user-selected NPR filter[3]. Cassidy et al. proposed for the first time to realize automatic simulation of watercolor painting. The
model uses Kubelk a-Munk synthetic model to simulate the optical effect of superimposed glaze. And a stroke style, which is similar to the pencil style, simulates the traits of rendering materials, hue and textural features[4]. And Isenberg determines the visibility of contour lines by means of rendering z-buffer parameter and removes some redundant information. There are also some representative papers at home, like an improved contour rendering technique based on image space stroke reconstruction, which is proposed by han-yu Wang in Zhejiang University[5], and Xiao-qiang Ding et al. proposed a computer-aided rendering algorithm to accelerate object rendering and maintain the visual features of fine brushwork rendering. Compared with artificial rendering works, the effect is satisfactory[6]. Si-yu Deng studied the artistic characteristics of Chinese ink and wash painting, and then improved and innovated on the basis of the existing research on non-realistic rendering of Chinese ink and wash painting, and developed a set of methods for real-time automatic conversion of three-dimensional model to two-dimensional ink and wash painting effect[7].

Non-photorealistic rendering research is highly practical, which has various applications in fields of Television Animation, Medical Science and Industrial Construction. Bing-wen Jin puts forward a method of generating stylized face animation based on example learning, which can improve the efficiency of making different kinds of face stylized animation with computer technology[8]. Chen-xi Zhu et al. based on the analysis of existing algorithms for extracting Valley and ridge lines, proposed a non-photorealistic mountain visualization algorithm based on OpenGL[9].

Although researches on stylized contour lines at home and abroad were many, investigates in time consistency of rendering were always thin. The method proposed by Bourdev parameterized the contour lines nearby the contour line of the last frame[10]. And on the basis of Bourdev’s method Kalnins focused on the improvement of sampling algorithm. Masuch proposed a method upon graphics which was not applicative when the length of arc changed[11]. Most of these algorithms focus on the first kind of contour, and seldom consider two or three kinds of contour.

**CONTOUR RENDERING**

**CATEGORY OF CONTOUR LINES**

Contour lines are divided into five types on the basis of different principles by which contour lines are produced, which are[12]:

The first kind: Contour of this kind consists of the common edges between the front of the observer and the back of the observer, and if the view vector(n) times the normal vector(m) is greater than 0(n*m>0), we call this direction as front, otherwise, we call it as back. We can divide this kind of contour lines into two lines, one of which is on the edge of the object and can show the overall appearance of the object to separate from other unrelated objects and background information, and the other of which is inside the object and can reflect the local characteristics of the object itself.

The second kind: the raised border. We call two planes as adjacent faces when they share one line, and if the included angle of their normal vectors is less than a specific value, we call the shared line as raised border.

The third kind: the lowered border. This kind of contour line has something in common with the raised border, like they are comparatively fixed and will not change
according to the moving of the sight point position. And it is defined a contrary direction: the included angle of the normal vectors of adjacent faces is greater than a specific value.

The fourth kind: the border. Just as its name implies, this kind of contour line just has one adjacent face and only exists in unenclosed object.

The fifth kind: This kind of contour line is extracted in shadows of objects and illumination is generally needed.

The five typed above-mentioned involve almost aspects of cases in non-photorealistic technology and the first three kinds among them are our focus because that we draw the three contours in most situations and the existing literatures are based on them. And beyond that, there are other types, like border lines added manually, which are generated by human interaction and are outside the scope of this article.

**CONTOUR LINE ACQUISITION METHOD**

Contour extraction directly affects the shape rendering of object and this is especially important because of the first joint the contour acted in this simulation of this paper. There are many kinds of contour lines and we only consider the first kind in this paper which consists of the common edges between the front of the observer and the back of the observer. And we can divide this kind of contour lines into two lines, one of which is on the edge of the object and can show the overall appearance of the object, and the other of which is inside the object and can reflect the local characteristics of the object itself.

![Figure 1. Determining points on surface contour lines of objects.](image)

To obtain contour line we must draw the object model in the scene firstly. Let us suppose point P as the point which we are drawing and camera as the origin of space coordinates, then firstly we can get vector N between the origin to the point P and if we change subtly the vector N in different directions and mark the changed angle as then we can get a series of changed vector V1, V2, V3, V4, ..., Vn. Then we start from the origin point to emit light in the direction of vector V1, V2, V3, V4, ..., Vn. Vn on the basis of the light tracking technology and we can get a intersectant point for the first time.

If the point P which we are drawing is on the surface of the object, as the point a in figure 1, then there must exists a point Pm (1 ≤ m ≤ n) which is not on the surface. We mark the depth of the object as Z and the presupposed threshold as $\mu$, then there exists a relationship in the depth direction in the scene as following:
If the point \( P \) is inside the outer contour of the model then the relationship in the depth direction in the scene as following:

\[
|Z_{p_n} - Z_p| > \eta
\]  

(1)

We suppose points \( P_1, P_2, P_3, \ldots, P_n \) are on the model and they are next to the point \( P \) we are drawing and the distances between them and the point \( P \) are quite small, and now we suppose vectors \( N_1, N_2, N_3, \ldots, N_n \) are the corresponding normal vectors of these points. If point \( P \) doesn’t belong to the first kind of contour line, as the point \( b \) in figure 1, then there must exist at least two points on the model which we mark as point \( P_i \) and point \( P_j (1 \leq i, j \leq n) \), and then the angle \( \alpha \) between their normal vectors \( N_i \) and \( N_j (1 \leq i, j \leq n) \) will be quite large. On the contrary, if point \( P \) is any point beyond the contour line then the angles between vectors \( P_1, P_2, P_3, \ldots, P_n \) are quite small. So we suppose a angle as \( \theta \) then there exist two cases as following:

When \( \frac{N_i - N_j}{\| N_i \||| N_j ||} \leq \cos \theta \) we can judge that point \( P \) is on the contour line of the model; And when \( \frac{N_i - N_j}{\| N_i \||| N_j ||} > \cos \theta \) we can judge that point \( P \) is not on the model.

Through analyzing we know that position vectors and normal vectors of some points are needed when we are drawing the scene, thus we don’t need to get the whole topology of three-dimensional model, which is much easier than the method of model contour drawing before because those points are in a small rectangle of the model and we don’t need to do any proper operation in advance. Hertzmann proposed that setting the vertex a binary mapping should be made on the basis of grid should first make a serial number for each vertex and then decide the front or the back of each plane and at last execute the algorithm of edge cache[13]. Sousa et al. proposed an algorithm that first numbered each vertex, then interpreted the positive and negative situation of each face, and then cached the edges[14]. Markosian proposed the real-time algorithm based on knowing the information of adjacent edges in advance. All of those methods of contour rendering mentioned above, which have better performance in smooth surface model and Polygon Model, must be based on confirming the position of each vertex and each edge and knowing topological structure of the model and ensuring connections of the points and the planes.

**STYLIZED CONTOUR RENDERING**

**PRINCIPLES TO FOLLOW IN CONTOUR RENDERING**

We conclude strategies of contour rendering with four points as following:
(1) Objects overlapping: When two objects intersect separately and in the direction of our line of sight they looks coincident then the contour line we drew should have halo and will gradually diminish slowly.

(2) Object shape of no color environment: To express the shape information of an object and without the using of color we can using contour line which is the best choice.

(3) Shadow and detail: We choose thick contour lines to describe shadow and highlight special objects in a scene or underline some details and if we are to show the sightless region covered by the overlapping of different objects.

(4) Texture: we will adopt different stylized rendering for different 6 texture, for example, we choose the straight line to express the hard surface of an object and rounded flexible curve to express its soft characteristic.

THE METHODS OF STYLIZED RENDERING

In the stylized contour rendering in Non-photorealistic Rendering there is a simple and intuitive method which is based on thickening geometric feature contour line of the object to heighten sidelines and achieve the transformation.

The thought of thickening contour line is that we can achieve our aims by regarding contour line as connecting of many dispersing points and then increasing proper points in the neighborhood of these dispersing points. Firstly, we get points on the contour line we have already got and suppose point $P_i(x_i, y_i, Z_i)$ is any point on the line. A point $P_{ij}(x_{ij}, y_{ij}, z_{ij})$, where $x_{ij} = x_i + \Delta x_y, \; \Delta x_y = (-1)^i * k_i, \; k_i = \text{rand}(m,m)$ is a integer we set, $i,j=1, 2, 3..., \; \text{the value of} \; k_i \text{is regulated parameter and the value of} \; y \text{and} \; z \text{has similar regulation. Then we can get a series of points (P1,P2,...,Pn).By Randomly and evenly increasing regulated amount of points in the neighborhood of Point P1 we can enrich the geometry contour data of the object and make the line more bold and strong and then strengthen the contrast of the contour line edge and highlight the contour feature and then can get vivid result.}

As shown in figure 2 below we can see the stylized contour rendering.

Figure 2. (a) larger value of m and n.

Figure 2.(b) moderate value of m and n.
However, stylized rendering we get by the method of thickening contour line is simplex and we get width and density of the relevant contour only through controlling the value of \( m \) and \( n \), which although highlight some details features and is hard to get more kinds of stylized border. Thus we improve the method above-mentioned by decreasing the amount of points in the neighborhood of point \( P_1 \) and just getting two points on the opposite side of the contour line of point \( P_1 \) and proceeding stylized rendering by the method of texture mapping.

The process of stylized characteristic line rendering in this paper as following: firstly we can get characteristic line of object model by the above-mentioned method, and then get stylized drawing of the characteristic line by texture mapping on the basis of different styles of art design. We suppose an endpoint on the characteristic line as point \( V \) and its unit normal vector is \( N \), then we suppose coordinate of the new endpoint is \( V' \), and thus:

\[
V' = V + |\beta| * N, \quad (\beta \neq 0)
\]

\( \beta \) is the weight. As shown in figure 3 below, we move the two endpoints of characteristic lines \( V_1 \) and \( V_2 \) along the normal vectors \( N_1 \) and \( N_2 \) of vertex of 3-d frame model to points \( V_1' \) and \( V_2' \) and we attach the four points in turn then we can get a quadrangle. Then we can randomly select interested art style 2D texture and keep the four vertices of the texture we selected corresponding, respectively, to points \( V_1' \), \( V_2' \), \( V_1 \) and \( V_2 \) and mapped to the quadrangle and thus we can get contour lines of different styles. Figure 4 shows the texture which stylized rendering uses. The method in this paper can ensure continuity between the characteristic line and achieve real-time rendering effect without extra compute time in filling the gap between characteristic lines, as shown in Figure 5.
THE RENDERING

Figure 4. The texture which stylized rendering uses.

Figure 5. (a) the contour line we first get.

Figure 5. (b) the contour line after using rendering technique and thickening.

Figure 5. (c) the contour line after stylized rendering in this paper.

CONCLUSION

We discussed the detection technique and rendering technique of contour lines in this paper and the method of stylized rendering. The method in this paper obtained contour lines by using mathematical method on the basis of conventional free-form surface method and further made controllable change on the width, thickness and concentration of the contour lines which we have obtained and then improved the contour line by using Texture Mapping Technique to realize stylized rendering without re-filling the gap between characteristics.
ACKNOWLEDGMENT

The research is partially supported by Humanities and Social Sciences Planning Fund of the Ministry of Education “Research on Cross-media Semantic Retrieval Methods of Digital Library Based on Deep Learning” (No.17YJA870010).

REFERENCES

1. Brooks S. Mixed media Painting and Portraiture. IEEE Transactions on Visualization and Computer Graphics, 2007, 13(5):1041-1054.
2. Sousa M C, Foster K, Wyvill B, et al. Samavati F. Precise Ink Drawing of 3D Models. Computer Graphics Forum, 2003, 22(3):369-373.
3. Cassidy J Curtis, Sean E Anderson, Joshua E Seims, Kurt W Fleischer, David H Salesin. Computer generated watercolor. Proceedings of SIGGRAPH 97:421-430.
4. Northrup J D. Artistic silhouettes: a hybrid approach[C]/International Symposium on Non-photorealistic Animation & Rendering. ACM, 2000.
5. Wang Aoyu. Research on Some Technologies of Non-Realistic Rendering [D]. Zhejiang University, 2005.
6. Ding Xiaoqiang, Hu Taocheng, Peng Ren, Yu Jinhui. Computer Aided Drawing of Fine Brushwork Figure Painting [J]. Journal of Computer Aided Design and Graphics, 2015, 27 (07): 1189-1195.
7. Deng Siyu. Research on rendering method of three-dimensional model based on GPU [D]. Jilin University, 2018.
8. Jin Bingwen. Research on Stylized Face Animation Generation Method Based on Example Learning [D]. Zhejiang University, 2017.
9. Zhu Chenxi, Feng Qian, Rao Simin. An OpenGL-based non-photorealistic mountain visualization algorithm [J]. Electronic Technology and Software Engineering, 2018 (09): 54.
10. Lee Markosian, Michael A. Kowalski, Daniel Goldstein, Samuel J. Trych in, John F. Hughes, Lubomir D. Bourdev. Real-time Non-Photorealistic Rendering. Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques, August 1997.pp.415~420.
11. Wang Aoyu, Tang Min et al. Overview of contour detection techniques in non-photorealistic rendering [J]. Computer Applications and Software, 23(5), 2006:102-104.
12. Winkenbach G A. Computer-Generated Pen-and-Ink Illustration[C]/Conference on Computer Graphics & Interactive Techniques. 1994.
13. A. Hertzmann, D.H. Salesin. Image Analogies[C]. Proceedings of SIGGRAPH, 2001.
14. Sousa M C, Buchanan J W. Computer-Generated graphite pencil rendering of 3D polygonal models[A]. In:EUROGRAPHICS'99 Conference Proceeding. London, England, 1999:195-207.07.