Visual Nondestructive Rendering of 3D Animation Images Based on Large Data

Yang Zhang and Xu Zhu(✉)

Liaoning Communication University, Shenyang 110136, China
lsyyd2020@163.com, xuenne@163.com

Abstract. In the visual non-destructive rendering of three-dimensional animation images, the traditional visual non-destructive rendering method is slow, so a visual non-destructive rendering method of three-dimensional animation images based on large data is proposed. The theoretical model of pixel-by-pixel time-domain denoising process is used to denoise, and GPU is used to achieve time-domain consistent processing according to the denoising results. The nonlinear Kuwahara filter is used to smooth the three-dimensional animation image, and the first-order differential operator is used to highlight the dramatically changing pixels in the image, so as to detect the edge of the image. After obtaining the distinct contour of the three-dimensional animation image, the non-destructive rendering of the three-dimensional animation image vision is realized. In order to verify the effectiveness of this method, the average rendering speed of the proposed method is 83.2%, which is significantly higher than that of the traditional method. The experimental results show that the average rendering speed of this method is the highest, the image rendering effect of this method is better, and the effectiveness of this method is verified.

Keywords: Big data · Three-dimensional animation · Image vision · Non-destructive rendering

1 Introduction

Three-dimensional animation image usually consists of point, line, surface, volume and other geometric elements and non-geometric attributes such as gray level, color, line shape, line width, etc. [1]. From the point of view of processing technology, three-dimensional animation images are mainly divided into two categories, one is composed of lines, such as engineering drawings, contour maps, curved surface wireframe drawings, and the other is similar to the shading of photographs, that is, commonly referred to as realistic images. With the emergence of raster image display, three-dimensional animation imagery has been greatly developed and widely used. With the development of three-dimensional animation imagery, three-dimensional animation imagery has been widely used. With its application, the visual non-destructive rendering of three-dimensional animation images began to develop [2].

Graphic display devices are mostly two-dimensional raster displays and dot matrix printers. From the representation of three-dimensional solid scene to the representation of two-dimensional raster and dot matrix, image rendering is called rasterization. The

S. Liu and L. Xia (Eds.): ADHIP 2020, LNICST 348, pp. 409–420, 2021.
https://doi.org/10.1007/978-3-030-67874-6_38

raster display can be regarded as a matrix of pixels. Any graph displayed on the raster display is actually a collection of pixels with one or more colors and gray levels. For a specific grating display, the number of pixels is limited, the color and gray level of the pixels are limited, and the pixels are of size, so the grating graphics are only approximate actual graphics. How to make the grating graphics approximate the actual graphics perfectly is the content of the grating graphics to be studied [3, 4].

At present, the commonly used highlight rendering methods are depth of field rendering algorithm based on hierarchical anisotropic filtering [5] and 3D animation image texture real-time rendering system [6]. For the input depth map, the pyramid of depth map is constructed, and the discontinuous region is repaired. Combined with the diffusion pattern distribution model, the fuzzy radius parameters of each point in the scene are determined. According to the depth information, the depth map is rendered layer by layer. The radius of the filter core of each layer is consistent with the radius of the diffusion circle. Finally, the separated anisotropic Gaussian filter is used to quickly get the rendering results. The latter first designs the hardware system of the rendering system, which is composed of image client, image management node, storage node and computing node. By drawing pixels of the corresponding coordinate points of the texture of the rendered image model, the purpose of rendering the texture of the animation image is achieved. Then, the bilinear algorithm is used to calculate the texture rendering of 3D animation image.

Due to the slow speed of traditional visual non-destructive rendering method for three-dimensional animation images, a new visual non-destructive rendering method for three-dimensional animation images based on large data is proposed.

2 Visual Nondestructive Rendering of 3D Animation Images Based on Large Data

2.1 Time Domain Consistency Processing

Firstly, time-domain consistency processing is used to reduce flicker in three-dimensional animation images, and the consistency of three-dimensional animation images in time-domain, i.e. between frames, is obtained. Because noise in time domain exists between different frames, it can not be eliminated by processing a single image. Considering the real-time requirement and the characteristics of graphics hardware operation, a new solution based on large data is proposed, which achieves ideal results with the help of graphics hardware, and has certain adaptive characteristics. Firstly, the theoretical model of the pixel-by-pixel time-domain denoising process is used to denoise the image. It is assumed that the color values of each unchanged position in the three-dimensional animated image obey normal distribution, $I \sim N(\mu, \sigma^2)$. Using I_n to represent the color of (x, y) position in the n frame, suppose that a position in the three-dimensional animated image does not change from the m frame to the n frame, that is, I_m, $I_{m+1}, \ldots,$ If I_n is a sample of I, the theoretical model is as follows:

$$\begin{cases} \bar{\mu} = \dfrac{1}{n-m+1}\sum\limits_{i=m}^{n} I_i \\ S^2 = \begin{cases} S_0^2 & m = n \\ \frac{1}{n-m}\sum\limits_{i=m}^{n} I_i - \bar{\mu} & m < n \end{cases} \\ \tau = \gamma \times S^2 \end{cases} \tag{1}$$

As a basis for judging whether the pixel changes in frame $n + 1$: If the deviation of frame $n + 1$ at this position is less than π, it is considered that there is no change, the output color is $\bar{\mu}$. Otherwise, it is considered that the position changes in frame $n + 1$ and output I_{n+1} Among them, S_0^2 is a constant, which is the demarcation point of color change caused by noise and change, and can be obtained by learning; S^2 represents the value of noise change; γ is greater than 0, which is used to control the coefficients. It needs to be given according to the performance of the equipment and the characteristics of three-dimensional animation images. The theoretical model is used to define the position color.

$$\mu_{n+1} = I_{n+1} \tag{2}$$

In the formula, μ_{n+1} represents the final position color and denoises again based on the final position color:

$$S_{n+1}^2 = \frac{S_n^2 \times (N-1) + \mu_{n+1}^2}{N} \tag{3}$$

If there is noise in the position, as the processing proceeds, μ_n and S_n^2 will gradually calculate the variance of the color and noise of the position, which can be used as a basis for judging whether the pixel in a new frame has changes other than noise. It is easy to see that these two parameters have certain adaptability to the relatively slow motion and change of objects in three-dimensional animation images. Although the motion or change of objects in three-dimensional animated images will also affect μ_n and S_n^2, the two parameters will be automatically corrected as processing proceeds. The size of N reflects the sensitivity to change [7]. In addition, the selection of N also needs to consider the frame frequency of the three-dimensional animation image.

After noise reduction, GPU is used to achieve time-domain consistent processing. Firstly, the frame image of the three-dimensional animation image is loaded into the texture in the main program, and then processed by GPU line operation. At the same time, Render To Surface is used to save the operation results to the texture cache for the next operation.

2.2 Image Smoothing Processing

After the time-domain consistent processing, it is necessary to smooth the three-dimensional animation image. By smoothing the image, a large range of color consistency can be obtained, and a relatively uniform color in space can be obtained, so as to achieve a better smoothing coloring effect [8]. At the same time, smoothing processing has important local operation characteristics, which can be better processed by parallel processing. In order to reduce the loss of boundary information, a non-linear Kuwahara filter is used.

The principle of Kuwahara filtering is to select the average color of the most gentle position near each pixel as the color of the current pixel.

Firstly, take a template of size $J \times K$, where J and K belong to set $\{i \in Z | i \bmod 4 = 1, \quad i > 4\}$, and then divide the template into four $[(J+1)/2] \times [(K+1)/2]$ size regions of size 1, 2, 3 and 4. There are overlaps of $[(J+1)/2] \times 1$ or $1 \times [(K+1)/2]$ between two adjacent regions, and one common pixel in the four regions is located at the center of the template. Moving the template over the whole image range, the average and variance of each small area are calculated for each location, and then the least variance of the four regions is determined by comparison, and the $\bar{I}$ of the region is used as the color of the central position pixels of the template.

After filtering, the noise in the smoothed area will be greatly reduced, the color will be more uniform, and the region information in the image will be well preserved. In order to get as close as possible to the effect of smooth coloring, four times of smoothing were carried out to obtain a larger range of color consistency [9]. In the implementation, a single 5×5 Kuwahara filter needs to be completed in two steps: Firstly, the mean and variance of 3×3 region around each pixel are calculated, and the results are saved to the central pixel position of a small region. After calculating the mean and variance, the mean and variance of the four vertices in 3×3 rectangular region near each pixel are checked, and the mean of the pixel with the smallest variance is obtained as the result of this filtering. There are two reasons for this: (1) Operational efficiency: if two steps are completed in the same process, each 3×3 region will be calculated four times, and only one time is needed to divide into two processes; (2) The requirement of Pixel Shader program for program size: the number of instructions in a ps_2_0 program can not exceed 256, which makes it difficult to complete the whole algorithm.

After calculating the mean and variance, the mean and variance of the four vertices in 3×3 rectangular region near each pixel are checked, and the mean of the pixel with the smallest variance is obtained as the result of this filtering. The specific image smoothing process is shown in Fig. 1.

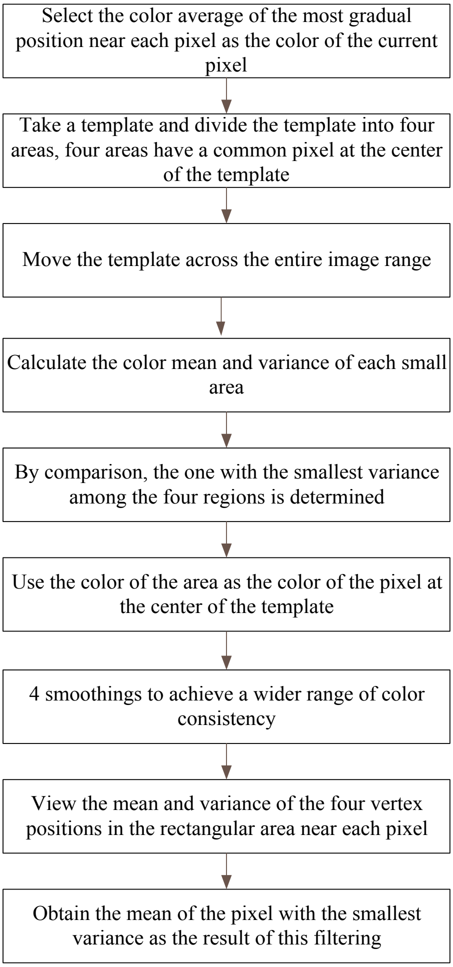

Fig. 1. Specific image smoothing process

2.3 Image Edge Detection

After the image smoothing process is completed, the image needs to be edge-detected to obtain a sharp outline of the three-dimensional animated image. The first-order differential operator is used to highlight the sharply changing pixels in the image [10]. First, calculate the I_x and I_y separately using two Sobel operators, and then take $mag(\nabla I)$ as the final result. Two Sobel operators are the horizontal Sobel operator and the vertical Sobel operator, respectively. The schematic diagram is shown in Fig. 2.

−1	0	1
−2	0	2
−1	0	1
−1	0	1
−2	0	2
−1	0	1
−1	0	1
−2	0	2
−1	0	1

(a) Horizontal Sobel operator

1	0	−1
2	0	−2
1	0	−1
1	0	−1
2	0	−2
1	0	−1
1	0	−1
1	0	−2
1	0	−1

(b) Vertical Sobel operator

Fig. 2. Schematic diagram

2.4 Image Visual Lossless Rendering Implementation

After the image edge detection is completed and the sharp outline of the three-dimensional animated image is obtained, the three-dimensional animated image is visually losslessly rendered based on the big data [11, 12]. First, real-time blanking is performed, and the three-dimensional information is transformed by a projection transformation on a two-dimensional display surface. Since the projection transformation loses depth information, it often leads to the ambiguity of the image. To eliminate ambiguity, you must eliminate occluded invisible lines or faces when drawing, customarily called eliminating hidden lines and hidden faces, or simply blanking. The projected image obtained by blanking is called the real image of the object [13]. The object of blanking is a three-dimensional object. The simplest representation of a three-dimensional object is represented by a planar polygon on the surface. The blanking result is related to the observation object and also to the viewpoint.

Then the level details are simplified. The drawing complexity of the 3D scene is very high. A complex scene may contain dozens or even millions of polygons. It is very difficult to image the complex scene. Hierarchical detail simplification is to improve the speed of image drawing by reducing the complexity of the scene.

Simplifying the level of detail requires the use of hierarchical detail display simplification technology, which reduces the geometric complexity of the scene by simplifying the surface details of the scene one by one without affecting the visual effect of the picture, thereby improving the efficiency of the rendering algorithm. This technique typically establishes geometric models of several different approximation degrees for a primitive polyhedral model. Compared with the original model, each model retains a certain level of detail. When observing objects from close proximity, a fine model can be used. When observing objects from a distance, a rougher model is used. In complex scenes, the complexity of the scene can be reduced, and the speed of image generation can be greatly improved. This is the basic principle of hierarchical detail display and simplification technology [14, 15].

However, it should be noted that when the viewpoint changes continuously, there is a significant jump between the two different levels of the model, and it is necessary to form a smooth visual transition between the adjacent levels of the model, that is, the geometric transition. The generated sequence of photorealistic images is visually smooth. The study of hierarchical detail techniques focuses on how to model the different levels of detail of the original mesh model and how to create geometric transitions between adjacent hierarchical polygon mesh models.

Hierarchical detail display and simplification is based on the geometric model of the object scene. By reducing the geometric complexity of the scene, that is, reducing the number of scene patches that the realistic graphics algorithm needs to render, the efficiency of rendering the realistic image is improved. Requirements. However, for 3D animated images, a technique capable of real-time realistic graphics rendering is required, and this technique is required to be applied to a general computer [16–18].

In recent years, technologies that meet this requirement have begun to emerge, that is, image-based rendering techniques. It starts from some pre-generated realistic images and generates realistic images at different viewpoints through certain operations such as interpolation, blending, and deformation. After generating realistic images at different viewpoints, it is necessary to render non-photorealistic images of these realistic images, including scientific data visualization processing and artistic style rendering processing [19].

Visualization of scientific data pays more attention to highlighting important information and neglecting secondary information so that the most important data can be more clearly expressed; artistic style rendering processing pays more attention to the personalization and artistic expression of 3D animated images, combined with scientific data. Visual processing and artistic style rendering processing realize visual lossless rendering of 3D animated images.

3 Experimental Research

In order to detect the visual lossless rendering method of three-dimensional animated images based on big data proposed in this paper, a comparative experiment was designed.

3.1 Experimental Parameters

The parameters of this experiment are shown in Table 1:

Table 1. Experimental parameters

Project	Data	Environment
Data sources	3D animated image library	Software environment: 3D animated image support database
Hardware accelerated frame rate	18.54 fps	
Algorithm frame rate	0.06 fps	
Experiment platform	MATLAB R2014a	
Memory	256 MB RAM	Hardware environment: NVIDIA GeForce FX 5600 graphics card, Intel(R) Celeron(R) CPU 2.40 GHz processor
Rendering scene	Sponza scenes	
Data source approach	Get the actual parameters	
Experiment process	Visually lossless rendering of 3D animated images	
Evaluation basis	Rendering speed	
Operating system	Microsoft Windows XP Professional (5.1,Build 2600)	

3.2 Experimental Process

The 3D animated image is supported by the database to collect the experimental 3D animated images, and the 3D animated images are subjected to time domain uniform processing, image smoothing processing, and image edge detection, thereby realizing image visual lossless rendering and comparing the rendering speed. In order to ensure the validity of the experiment, image depth of field rendering algorithm based on hierarchical anisotropic filtering (Method 1), real time rendering system of 3D animation image texture (Method 2) is compared with the big data-based three-dimensional animated image visual lossless rendering method proposed in this paper.

3.3 Experimental Results

The comparison results of rendering speed of different methods are shown in Fig. 3.

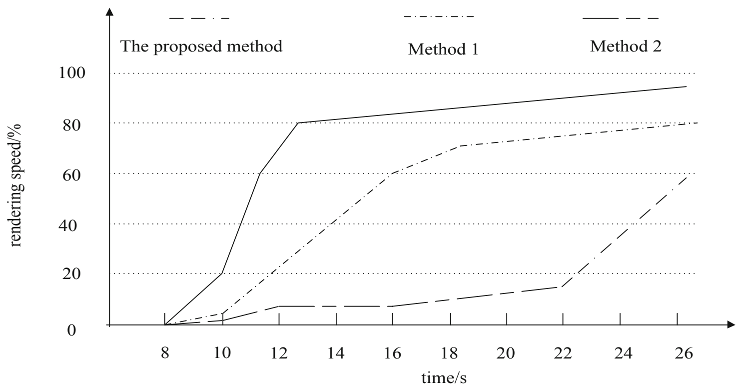

Fig. 3. Rendering speed comparison

It can be seen from Fig. 3 that the maximum speed of visually non-destructive rendering of 3D animation images using Method 1 is about 80%; the maximum speed of visually non-destructive rendering of 3D animation images using Method 2 is about 60%; and the use of 3D based on big data Animated image visual lossless rendering method The highest rendering speed for visually lossless rendering of 3D animated images has reached more than 90%. By comparing the average rendering speed, it can be seen that the average rendering speed of the 3D animation image visual lossless rendering method based on big data is higher than the existing method, which proves the superiority of the method. This is because the proposed method uses pixel by pixel time domain noise reduction to deal with the noise in the image, and uses GPU to achieve time-domain consistent processing according to the denoising results, which reduces the noise interference on image rendering and improves the rendering speed.

In order to further verify the effectiveness of the proposed method, taking the line elements in the 3D animation image as the object, different methods are used to render the lines in the 3D animation image, and the results are shown in Fig. 4.

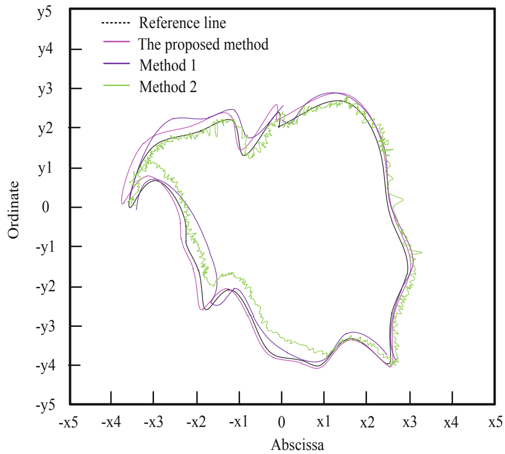

Fig. 4. Comparison of different rendering methods

From the analysis of Fig. 4, it can be seen that the difference between the lines rendered by the proposed method and the reference lines is small, and the lines can be completely presented, which is basically consistent with the reference lines. However, in method 1, there is a phenomenon that the ends of the lines are not connected and there is a gap; although method 2 obtains complete closed lines, there are a lot of rough lines. Therefore, the proposed method can render the image better.

4 Conclusion

Image rendering is the process of transforming three-dimensional light transfer processing into a two-dimensional image. Scenes and entities are expressed in three-dimensional form, which is closer to the real world and easy to manipulate and change. Before rendering the 3D animated image, you need to prepare 3D geometric model information, 3D animation to define information and material information. The 3D geometric model is obtained by 3D scanning, 3D interactive geometric modeling and 3D model library to obtain 3D animation definitions. Motion design, motion capture, motion calculation and dynamic deformation are used to obtain materials from scanned photos, computer calculated images and human paintings. Get it out of the picture. The work to be done in image rendering is to generate images by geometric transformation, projection transformation, perspective transformation and window clipping, and then through the acquired material and light and shadow information. After the image is rendered, the image information is output to an image file or a video file, or to a frame buffer of the display device to complete the graphics generation. The experimental results show that the visual lossless rendering method based on big data and 3D animated image can reduce the amount of calculation and improve the rendering speed, and the image rendering effect is good.

References

1. Song, L.: The early stage of the visual effects of movies - the concept and skills of the early stage shooting of the new visual effects of movies. Modern TV Technol. **10**(11), 150–155 (2017)
2. Morikawa, T., et al.: Image processing analysis of oral cancer, oral potentially malignant disorders, and other oral diseases using optical instruments. Int. J. Oral Maxillofac. Surg. **49** (4), 515–521 (2020)
3. Jinming, L.: Optimized simulation of feature point matching for 3D reconstruction of multi-vision animation images. Comput. Simul. **34**(39), 341–344 (2017)
4. Xixi, W.: Three-dimensional visual representation method in virtual display design of metallurgical automation. Autom. Instrum. **21**(22), 147–149 (2018)
5. Zhiheng, O.Y., et al.: A layered image depth of field rendering algorithm using anisotropic filtering. Optical Technique **044**(004), 469–475 (2018)
6. Suran, K., Junping, Y.: Design of image texture real-time rendering system for three-dimensional animation. Modern Electron. Technique **041**(005), 102–105 (2018)
7. Liang, S., Dawei, L., Liang, L., et al.: Design and implementation of disparity visualization adjustment method for three-dimensional animation production. J. Comput. Aided Des. Graph. **29**(37), 1245–1255 (2017)
8. Liu, S., Sun, G., Fu, W. (eds.): eLEOT 2020. LNICST, vol. 339. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-63952-5
9. Qian, D., Jingshuang, W.: Talking about the script creation and technical application of three-dimensional animation film and television works. Tomorrow Fashion **10**(12), 361–361 (2018)
10. Lu, M., Liu, S.: Nucleosome positioning based on generalized relative entropy. Soft. Comput. **23**(19), 9175–9188 (2018). https://doi.org/10.1007/s00500-018-3602-2
11. Deming, Z.: A brief analysis of the real reproduction of three-dimensional animation in the historical documentary “palace museum”. J. Res. Guide **18**(25), 282–282 (2017)
12. Dongli, X.: A brief talk on the visual language performance of graphic design elements in animation. Time Educ. **14**(15), 198–198 (2017)
13. Bing, L., Shugang, L.: Efficient polarization direction measurement by utilizing the polarization axis finder and digital image processing. Opt. Lett. **43**(12), 2969–2972 (2018)
14. Megibow, A.J., Kambadakone, A., Ananthakrishnan, L.: Dual-energy computed tomography: image acquisition, processing, and workflow. Radiologic Clin. North Am. **56**(4), 507–520 (2018)
15. Engelkes, K., Friedrich, F., Hammel, J.U., Haas, A.: A simple setup for episcopic microtomy and a digital image processing workflow to acquire high-quality volume data and 3D surface models of small vertebrates. Zoomorphology **137**(1), 213–228 (2017). https://doi.org/10.1007/s00435-017-0386-3
16. Xiaoyuan, Y., Jingkai, W., Ridong, Z.: Random walks for synthetic aperture radar image fusion in framelet domain. IEEE Trans. Image Process. **27**(2), 851 (2018)

17. Mujika, K.M., Méndez, J.A.J., de Miguel, A.F.: Advantages and disadvantages in image processing with free software in radiology. J. Med. Syst. **42**(3), 1–7 (2018). https://doi.org/10.1007/s10916-017-0888-z
18. Haiyang, L., Xiaofeng, H., Xu, L.: Remote real-time rendering system based on graphics cluster. J. Syst. Simul. **31**(005), 886–892 (2019)
19. Liu, S., Pan, Z., Cheng, X.: A novel fast fractal image compression method based on distance clustering in high dimensional sphere surface. Fractals **25**(4), 1740004 (2017)