The Design of a Virtual Reality Image Distortion Method for mobile devices

Guanlin Wu1,2, Weidong Bao1*, Jiang Cao2, Xiaomin Zhu1, Yang Ping2, Jia Song4, Quan xin3

1National University of Defense Technology, Changsha, P.R.China
2Academy of Military Science, Beijing, P.R.China
3Nanhu Lab, Jiaxing, P.R.China
4Microsoft Asia-Pacific R&D Group, Beijing, P.R.China

*Corresponding author’s e-mail: wdbao@nudt.edu.cn

Abstract. How to predict a more accurate next frame according to the current frame is still a problem. In this study, a method of generating intermediate frames in VR is proposed, which can effectively reduce the image jitter in VR games and increase the user experience; on the other hand, the inverse matrix method proposed in this study has a small amount of calculation and a fast calculation speed, and can complete the calculation in a short time. Therefore, this method can help us to complete the insertion of frames in the specified time. In this paper, firstly, the algorithm is constructed based on the principle of inverse matrix, and then the simulation experiment is carried out. The experimental results show that the scheme is a viable and effective solution.

1. Introduction

As we all know, whether in games or videos, their frame rate must be maintained above 60HZ, in order to meet human requirements, users can have a good experience. Otherwise, if you turn your head, the picture will be trembling, or it may take a long time for the videos to sync with your movements. This can create dislocation. In some cases, users will feel dizzy, even nauseous.

Although mobile VR devices are in the rising stage of development and their popularity is increasing year by year, they are still immature due to their high cost. Even the most powerful CPU of the mobile terminal is the Snapdragon 845 or the Exynos 8890 of Samsung is far from being able to reach the renderings and speeds of the computer-side I5 4590 and GTX 970. The rendering time per frame must be limited to 16.66 ms for a frame rate of 60 Hz and above, which is almost an impossible task given the GPU performance of the mobile VR device, which When the head rotates, the viewpoint needs to be changed, but no corresponding picture is rendered and sent to the display, then the picture seen by the user will still be the previous frame, which will cause image jitter. Even if the pictures are very simple, the rendering speed is very fast, we still need to consider the issue of vertical synchronization, when we receive the vertical synchronization signal to start rendering know to render after rendering, there is still a delay of 16.66ms. That is to say, even if we can render within the specified time, but in the final delivery, the current frame rendering parameters are still the head movement parameters of our rendering time, the distance from this parameter has passed 16.66ms, natural will have a delay of 16.66ms, which is still not a satisfactory delay, the user will still feel...
delayed and unfluently.

Although the popularity of mobile Virtual Reality technology is high, the user experience is far less than the computer. The main reason is that mobile hardware performance is much weaker than the computer side. Therefore, the mobile terminal brings the user experience is often unsatisfactory, there will be skipping, jitter phenomenon.

Therefore, how to predict a more accurate next frame according to the current frame is still a problem worthy of study in the field of Virtual Reality. Therefore, this paper attempts to use the idea of middle frame, the proposed matrix inverse computing method to calculate the amount of small, fast, and can be completed within a short period of time to achieve within a specified time to complete a frame insert, guaranteed speed in the calculation of shader in the program to achieve the current frame-by-pixel distortion, to achieve high-level forecasting, the above problem provides a solution.

2. Project Concept and Design

2.1. Project Ideas

In response to the above problem, when the Virtual Reality games can not render a frame within a specified time, we use to generate intermediate frames to insert, in order to reduce the games screen jitter method to deal with.

In order to take full account of the head displacement, this article uses the latest OpenGL ES computational shader to achieve. This method takes advantage of the projection process of open GL ES itself to deduct the coordinates of the corresponding position point in the real world and calculate the corresponding projection position according to the coordinates and the current viewpoint information. Thus rendering a new frame. To put it simply, the mapping relationship between pixel positions in each coordinate system is calculated mainly through the matrix transformation in scene rendering, and the mapping relationship between the current frame pixel position and the next frame pixel position is obtained. Finally, the next frame is generated according to the current frame, get distorted image output to the display device.

2.2. Basic algorithm step design

More specifically, the present solution generates the next frame according to the current frame, the current state of the helmet and the state of the helmet at the next frame moment according to the inverse transformation of the matrix transformation when the scene is rendered, and outputs the next frame directly to the display device. Specifically, the following steps:

Step 1: The current rendering frame information is read, and the coordinates of the rendered frame in the screen space coordinate system \(\left[ x_{\text{original}}, y_{\text{original}} \right] \) are read to read the state of the current and next frames of the device;

Step 2: Perform the necessary preprocessing on the read-in data, and convert the pixel coordinate \(\left[ x_{\text{original}}, y_{\text{original}} \right] \) in the screen space coordinate system to the coordinate \((x, y, z, w)^T\) Generate rotation matrix \( R, R' \);

Step 3: Perform matrix inverse transformation on each pixel coordinate \((x, y, z, w)^T\) of the current frame according to the projection matrix \( P \), the rotation matrix \( R \) and the left and right eye viewpoint translation matrixes \( T_L, T_R \) to obtain the coordinates \( (x_1, y_1, z_1, w_1)^T \);

Step 4: Matrix transformation is performed on the corresponding coordinate points \((x_1, y_1, z_1, w_1)^T\) in the world coordinate system according to the left and right eye viewpoint translation matrices \( T_L, T_R \), rotation matrix \( R' \) and projection matrix \( P \) at the next frame moment, get the coordinate \((x', y', z', w')^T\) corresponding to the standard equipment coordinate system of the next frame;

Step 5: the coordinates \((x', y', z', w')^T\) obtained in step 4 are transformed linearly and finally converted into coordinates \((x_{\text{final}}, y_{\text{final}})^T\) in the screen coordinate system.
Step 6: the pixel RGB value in each pixel coordinate position \((x_{original}, y_{original})^T\) of the current frame is assigned to the pixel coordinate position \((x_{final}, y_{final})^T\) corresponding to the next frame to obtain the final distorted image.

3. Program realization and effect evaluation

3.1. Experimental environment settings

The experimental development environment of this verification experiment is: Intel's third-generation Core i5-3550 3.30GHz quad-core CPU (processor), NVIDIA GeForce GTX 650 GPU (graphics), 8G memory, the software environment is Eclipse2015 (software development tools), Microsoft Visual Studio2015 (Software development tools) and OpenGL ES 3.1 (embedded system open source graphics library). Operating environment: RK3288 (mobile processor models), ARM Mali-T764 (GPU models), 2GB of memory, 16GB of storage space.

3.2. Program Implement the process and results in a specific device

The program is mainly used in mobile Virtual Reality devices, such as VR helmets. When the device is running the VR program, the main thread of the device renders the image frame output with a frequency of at least 60 Hz, and the sub-thread runs the program method all the time. When the device cannot render a frame within the specified time, the sub-thread sends the distorted image backbuffer, the final output to the display device, thus ensuring that the device can run at a sufficient frame rate, and reduce the picture jitter.

Based on the program, the related results in the image processing are as shown in Figure 1-3 below:

![Figure 1. The image before and after the distortion of the current frame view](image1.png)

![Figure 2. The actual next frame view of the contrast map before and after the image distortion](image2.png)
Figure 3. Shows the next frame view by the method of the invention twisting before and after the image is distorted

4. Conclusion
In general, on the one hand, this approach is a method of generating intermediate frames in Virtual Reality. This method can effectively reduce the image jitter in VR games and increase the user experience. On the other hand, the matrix inverse algorithm has many advantages, such as the calculation is small, fast computing speed, it can complete the operation in a short time and achieve the completion of a frame within the specified time to insert.

References
[1] Chen S. E. and Williams L.(2003) View Interpolation for Image Synthesis. Computer Graphics (SIGGRAPH’93 Proceedings), pp. 279-288.
[2] Adelson S. J. and Hodges L. F.(2003) Stereoscopic ray-tracing. The Visual Computer, pp. 127-144.
[3] Laveau S. and Faugeras O.(2008) 3-D Scene Representation as a Collection of Images and Fundamental Matrices. INRIA, Technical Report No. 2205.