

文章编号:1007-2780(2018)09-0808-08

基于 OpenGL 的交互式全景视频显示系统

周 凡*, 张 伟, 杨付正

(西安电子科技大学 综合业务网理论及关键技术国家重点实验室, 陕西 西安 710071)

摘要:随着虚拟现实和多媒体通信技术的快速发展,全景视频作为一种新兴的数字媒体,凭借其良好的临场感和沉浸感在影视娱乐、教育教学、工业仿真、医学研究等众多领域取得广泛应用。为了更好地开展全景视频的相关研究,建立交互式的全景视频显示系统成为了优化和完善全景视频业务的重要前提。本文基于 OpenGL 渲染技术,结合用户对于全景视频交互操作的需求,搭建了交互式全景视频显示系统,并在该系统上通过实验分析了本系统的交互时延和视频分辨率、帧率的关系。实验结果表明,交互产生的时延和视频分辨率、帧率的大小呈正相关关系。在本平台的设置下,交互产生的时延不超过 0.02 s,为用户提供了良好的交互性和视觉体验,为全景视频的相关研究与应用开发提供了稳定的平台。

关 键 词:全景视频;交互式渲染;显示系统;视觉体验;OpenGL

中图分类号: TN141 **文献标识码:** A **doi:** 10.3788/YJYXS20183309.0808

Interactive omnidirectional video display system based on OpenGL

ZHOU Fan*, ZHANG Wei, YANG Fu-zheng

(State Key Laboratory of Integrated Service Networks, Xidian University,
Xi'an 710071, China)

Abstract: With the rapid development of virtual reality and multimedia communication technology, omnidirectional video has become a major form of immersive media. Due to the sense of presence and immersion provided to the end users, omnidirectional videos have been widely applied in many fields including entertainment, education, industrial simulation, and medicine. To facilitate the research on omnidirectional videos, establishing an interactive omnidirectional video display system is an important precondition. Based on the OpenGL rendering technology, this paper built an interactive omnidirectional video display system according to the display demand of end users. To analyze the performance of the proposed system, the relationship between the interaction delay and the video resolution as well as the frame rate were investigated. Experimental results showed that the interaction

收稿日期:2018-05-05;修订日期:2018-06-20.

基金项目:中央高校基本科研业务费(No. JB180105);国家自然科学基金(No. 61601349, No. 61571337)

Supported by the Fundamental Research Funds for the Central Universities (No. JB180105); Supported by the National Science Foundation of China (No. 61601349, No. 61571337)

* 通信联系人, E-mail: fzhou_945@stu.xidian.edu.cn

delay is proportional to the video resolution and the frame rate and the maximum delay of the proposed system is only 0.02 s in the test condition which is able to provide end users with satisfying interactivity and visual experience. The proposed system can serve as a stable platform for conducting research on omnidirectional videos.

Key words: Omnidirectional video; interactive rendering; display system; visual experience; openGL

1 引言

随着虚拟现实技术和计算机图形学的不断发展,全景视频成为沉浸式媒体的重要表现形式,并且在虚拟现实领域扮演着重要的角色^[1]。全景视频可以给予观众极强的临场感和沉浸感,具有巨大的商业价值和广阔的发展空间,谷歌、微软、三星、华为等国内外公司相继展开了对全景视频业务的研发和应用。

全景视频,也被称为沉浸式视频和球面视频,是使用全方位摄像机或者相机阵列系统在同一时间内对场景进行全方向拍摄得到的视频记录^[2]。在虚拟环境的构造方式中,全景视频允许用户在视点不变的情况下通过改变观看角度来观察到周围全部场景。这种方式相比于传统平面视频具有极强的沉浸感,也是目前应用最广泛的虚拟环境构造方式。近年来,针对全景视频的相关研究层出不穷。当前的研究集中在全景视频的投影方式^[3-6]、编码传输^[7-8]和三维显示技术^[9-11]等领域,关于全景视频交互显示的研究相对较少。Zoric 等人^[12]研究了全景视频的交互实现,通过触屏操作和手势识别来控制视角,从而实现了全景视频观看时的视角切换,但是其交互操作适用于大屏幕和平板电脑,没有考虑到更常见的个人主机端。Neng 和 Chambel 等人^[13]研究了全景电视(Panoramic TV)中的交互式播放,实现了三维场景的导航和方向定位,但是没有考虑到鼠标等外设的交互行为。施珂奕^[14]等人研究了一种基于 OpenGL 的三维模型点坐标获取的方法,运用于三维模型交互控制时准确获取指定点的坐标。全景视频对于全景视频播放系统的交互能力提出了很高的要求,但是现有技术往往忽略了主机端的全景视频交互显示。因此本文探讨了全景视频交互和显示的关系,实现了交互式全景视频显示系统。

本文搭建的交互式全景视频显示系统,允许用户使用鼠标拖曳全景视频,多角度地观看视频内

容,营造了具有沉浸感和真实感的视觉效果。本系统以计算机图形学中的渲染流程为基础,基于 OpenGL 渲染平台和 SDL (Simple DirectMedia Layer)库进行开发,通过使用欧拉角来调整全景视频的投影显示,实现了全景视频显示系统的人机交互。

2 全景视频显示系统

全景视频的每一帧都是通过图像拼接技术得到的立体观察域,如立方体面,球面(如图1所示)等,因此全景视频在平面显示器上的显示过程需要进行映射变换,实现三维的立体图到二维显示平面的投影过程。全景视频常见的投影格式包括 ERP (Equirectangular Projection)和 CMP (Cube Map Projection) 两种。其中 ERP 将全景视频球面按照经纬坐标,投影到二维平面坐标上,生成一张宽高比为 2:1 的矩形图片^[15]。该投影技术在水平方向上可以很好地维持图像尺寸,能够有效地控制垂直方向的失真,是目前国际上制定全景视频相关标准的基础之一,已被 MPEG-I (Moving Picture Experts Group-Immersion) 选定为标准投影格式。本文以 ERP 为基础,将全景视频显示到二维屏幕上,并提出了一种基于欧拉角的矩阵变换方法,实现了全景视频的交互式显示。当用户交互式地观看视频时,该方法能将视频正确地投影在显示设备上。



图1 球面投影映射原理图

Fig.1 Spherical projection mapping diagram

2.1 全景视频的投影显示

全景视频的投影显示是交互式显示系统中的关键部分。本文依赖投影变换完成全景视频的交

交互式渲染过程。在建立投影映射关系时需要选择恰当的投影变换。投影变换创建了场景中观察点可见的一个锥体范围,通常叫做视锥体,视锥体的宽窄和纵深决定了视频内容如何投影到屏幕上。在计算机图形学中,投影变换分为正交投影和透视投影两种。相比于正交投影,透视投影会按照透视缩短效果显示视觉场景,距离远的物体会比相同大小的接近观察平面的物体的投影小^[16],真实感更强。因此本文使用了透视投影来完成全景视频的投影显示。

在透视投影的视锥体(图 2)中,靠近观察点(0,0,0)的面称为近平面,远离观察点的面称为远平面,只有落在这两个平面内的几何体才会渲染到最终图像上。视锥体将视角内的视频内容包含在内,如图 2 所示,点 (x_e, y_e, z_e) 为视锥体内的任意一点。该点和观察点(0,0,0)在近平面的交点,决定了 (x_e, y_e, z_e) 在近平面的投影位置,由此建立了视频内容的三维坐标和二维屏幕坐标的映射关系。图 2 中,利用相似三角形的性质可以确立投影点的坐标 (x_p, y_p) :

$$x_p = -n \times x_e / z_e, \quad (1)$$

$$y_p = -n \times y_e / z_e. \quad (2)$$

式(1)和式(2)建立的投影关系将全景视频的内容准确地投影到了近平面上。通过检测投影点和像素点的位置关系可以决定由哪些像素点来显示投影结果,生成正确的显示图像。

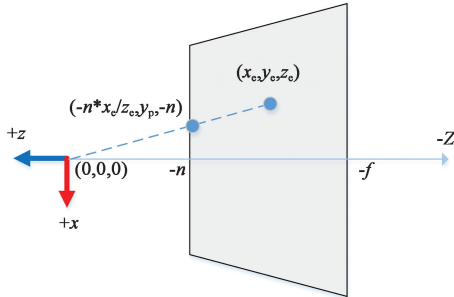


图 2 透视投影俯视图

Fig.2 Overhead view of perspective projection

2.2 全景视频的交互

全景视频显示系统的交互实现应根据终端用户的交互式操作准确完成全景视频视角切换。其中,确定正确的显示视口直接影响了视频变换视角时的合理性和准确性。本文利用了基于欧拉角的投影变换矩阵,用于计算视频显示时的坐标

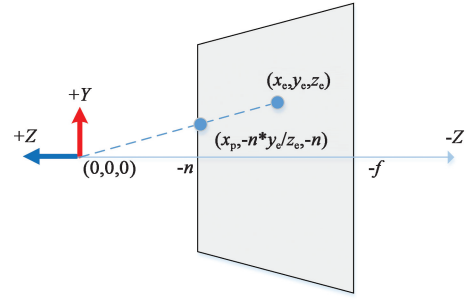


图 3 透视投影侧视图

Fig.3 Side view of perspective projection

变换。

欧拉角是一组描述三维空间物体旋转的三维向量,常用来表征刚体的运动姿态,在计算机图形学和航空航天等领域有着广泛的运用。如图 4 所示的三维坐标系中,欧拉角分别描述了绕 X 轴旋转的俯仰角(pitch),绕 Y 轴旋转的偏航角(yaw)以及绕 Z 轴旋转的翻滚角(roll)。

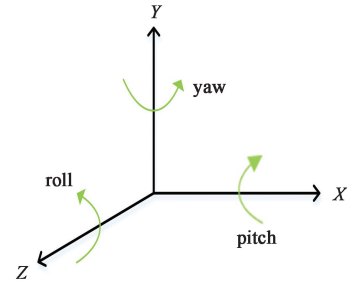


图 4 欧拉角示意图

Fig.4 Euler angles diagram

当刚体绕 X 轴转动 θ_1 ,绕 Y 轴转动 θ_2 ,绕 Z 轴转动 θ_3 时,对应各个坐标轴的旋转矩阵为:

$$\mathbf{R}_x(\theta_1) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(\theta_1) & -\sin(\theta_1) \\ 0 & \sin(\theta_1) & \cos(\theta_1) \end{pmatrix}, \quad (3)$$

$$\mathbf{R}_y(\theta_2) = \begin{pmatrix} \cos(\theta_2) & 0 & \sin(\theta_2) \\ 0 & 1 & 0 \\ -\sin(\theta_2) & 0 & \cos(\theta_2) \end{pmatrix}, \quad (4)$$

$$\mathbf{R}_z(\theta_3) = \begin{pmatrix} \cos(\theta_3) & -\sin(\theta_3) & 0 \\ \sin(\theta_3) & \cos(\theta_3) & 0 \\ 0 & 0 & 1 \end{pmatrix}. \quad (5)$$

将上述 3 个旋转矩阵相乘则得到最终的旋转矩阵:

$$\mathbf{R}(\theta_1, \theta_2, \theta_3) = \mathbf{R}_x \mathbf{R}_y \mathbf{R}_z =$$

$$\begin{pmatrix} c_2 c_3 & -c_2 s_3 & s_2 \\ c_3 s_1 s_2 + c_1 s_3 & c_1 c_3 - s_1 s_2 s_3 & -c_2 s_1 \\ s_1 s_3 - c_1 c_3 s_2 & c_1 s_2 s_3 + c_3 s_1 & c_1 c_2 \end{pmatrix}, \quad (6)$$

其中: s_i 代表 $\sin(\theta_i)$, c_i 代表 $\cos(\theta_i)$ 。

假设观察点(人眼或相机)的位置如下:

$$\mathbf{R}_c = [0, 0, z]^T, \quad (7)$$

由式(6)和式(7)相乘,可以得到观察点 $\mathbf{R}_c$ 所观察到的位置和方向 $\mathbf{R}$:

$$\mathbf{R} = [s_2 z, -c_2 s_1 z, c_1 c_2 z]^T. \quad (8)$$

式(8)代表了观察点所观察的位置和方向。随着用户鼠标的拖动,欧拉角发生变化,矩阵 $\mathbf{R}$ 也随之改变,得到新的观察方向。值得注意的是,由于显示屏幕是二维平面,鼠标拖拽只涉及 X 轴和 Y 轴方向上的变换,故只需考虑欧拉角的两个维度, Z 轴上没有发生旋转,所以式(8)中的矩阵没有和 θ_3 有关的分量。通过引入欧拉角以记录用户使用鼠标拖动后全景视频的旋转量,可以根据式(8),实时更新全景视频的观察方向,以便完成后续的渲染操作。

3 基于 OpenGL 的交互式全景视频播放器

本文搭建了一种基于图形学渲染流程的全景视频显示系统。该显示系统的外部依赖项包括 OpenGL 和 SDL 库。OpenGL 作为一个跨平台的应用程序编程接口(API, Application Programming Interface)^[17], 提供了完备的三维图形开发库。SDL 作为一个多媒体应用编程接口,可用于处理音视频文件和获取鼠标等外设的状态信息。本文提出的全景视频显示系统根据 OpenGL 的渲染流程构建渲染框架,同时调用 SDL 库完成来自外设的交互式操作响应。

本文提出的全景视频显示系统的核心是循环渲染模块。该模块包括了顶点处理、图元处理和片段处理 3 个环节,其具体实现流程如图 5 所示。

顶点处理环节(Vertex Operation)负责处理物体顶点的相关数据,计算顶点在屏幕上的位置信息,并通过光照的计算来判断顶点的颜色。图元处理(Primitive Assembly)负责将顶点数据和几何图元关联起来,进而生成对应的片元,片段处理(Fragment Operation)负责处理片元的最终颜

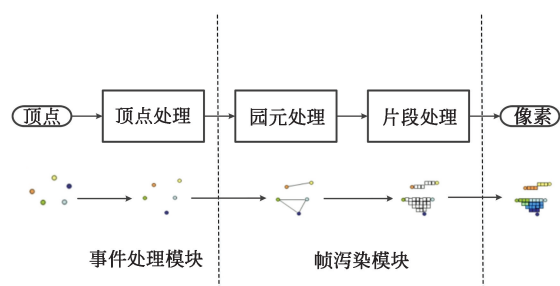


图5 循环渲染模块流程图

Fig.5 Flowchart of loop rendering module

色和深度值,使用纹理映射的方式对顶点处理环节所计算的颜色值进行补充。

在本系统的循环渲染模块中,由于鼠标交互的事件处理和后续的渲染流程相对独立,因此进一步划分为两个子模块:事件处理模块和帧渲染模块。其中,事件处理模块对应渲染流程中的顶点处理环节,负责处理鼠标等外设的输入事件,完成场景的坐标变换;帧渲染模块对应了渲染管线的图元处理和片段处理环节,通过着色器来完成显示器窗口图像的渲染工作。在搭建循环渲染模块之前,需要建立渲染模型,为渲染管线提供顶点数据。本显示系统使用的渲染模型是通过 OpenGL 生成的球形立体背景,半径为 20,并划分球面的经纬度,便于进行坐标定位。其中纬度划分为 512 份,经度划分为 256 份,保证了 2:1 的宽高比,为后续的片段处理做好准备。本文选取的全景视频源,采用 H.264 编码,分辨率为 3840×1920 ,这和球形背景的经纬度比例契合,均满足 2:1 的宽高比。

3.1 事件处理

在全景视频显示系统中,一旦监测到鼠标移动事件,鼠标的二维偏移量便会转换为场景物体的三维偏移量,从而执行物体的坐标变换。本系统利用式(8)建立起全景视频交互时的坐标投影映射关系,便于在帧渲染模块中显示出正确的图像。

在事件处理模块中,三维场景的矩阵变换遵循一定的步骤。在建立场景模型之后,依次进行模型变换(Model Transformation),视图变换(View Transformation),投影变换(Projection Transformation)^[11]。模型变换负责操纵模型和其中的特定对象,变换使得模型移动到需要的位置,包含了平移,旋转和缩放等三大类变换。在建

立模型矩阵时,本系统使用 SDL 库函数来获取用户鼠标的运动状态(如移动,滚轮等)和鼠标在二维屏幕上的位置信息(包括 X , Y 坐标)。当用户移动鼠标时,更新相应的位置信息,得到鼠标的实时坐标。

通过对鼠标移动事件的监听,可以计算出鼠标拖曳全景视频时的二维偏移量,用 (X_{old}, Y_{old}) 和 (X_{new}, Y_{new}) 来分别表示鼠标拖动前后的位置坐标,则二维偏移量表示为:

$$\Delta X = X_{new} - X_{old}, \Delta Y = Y_{new} - Y_{old}. \quad (9)$$

将二维偏移量转换为全景视频在球体的偏移角度,如图 6 所示,计球体半径为 R ,则

绕 X 轴偏移角度为:

$$\theta_1 = 2 \times \tan^{-1}(\Delta Y / 2 \times R). \quad (10)$$

绕 Y 轴偏移角度为:

$$\theta_2 = 2 \times \tan^{-1}(\Delta X / 2 \times R), \quad (11)$$

式(10)和式(11)建立了二维平面和三维球体的坐标映射关系,将用户使用鼠标拖曳全景视频

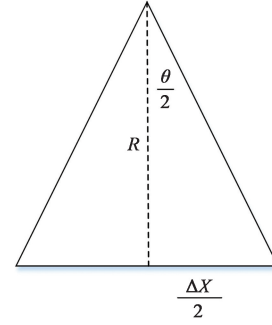


图 6 二维偏移量和角度的关系图

Fig.6 Two-dimensional offsets and angles diagram

时产生的二维偏移量映射到物体场景对应的三维偏移角度。在得到三维偏移角度后可根据式(8)得到正确的模型矩阵。值得注意的是,在模型矩阵的建立过程中会带来万向锁(Gimble Lock)的问题,即整个场景被限制在只能绕垂直轴旋转,丢失了一个表示维度。

在式(6)中取 $\theta_2 = \pi/2$,有:

$$R\left(\theta_1, \frac{\pi}{2}, \theta_3\right) = \begin{pmatrix} 0 & 0 & -1 \\ \sin(\theta_1 - \theta_3) & \cos(\theta_1 - \theta_3) & 0 \\ \cos(\theta_1 - \theta_3) & -\sin(\theta_1 - \theta_3) & 0 \end{pmatrix}, \quad (12)$$

式(12)中,刚体绕 Y 轴旋转 $\pi/2$ 后,失去了一个自由度,导致旋转轨迹受到限制,产生错误的运动姿态,出现了万向锁的问题。本文为了解决万向锁带来的问题,限制偏移角度 θ_1 的范围为 $(-2\pi, 2\pi)$,偏移角度 θ_2 的范围为 $(-\pi/2, \pi/2)$,从而避免了万向锁现象,计算出正确的模型矩阵。

将事件处理环节得到的模型矩阵与视图矩阵和投影矩阵进行乘法运算可以得到三维场景的矩阵变换。其中,视图矩阵负责确定观察点的位置和观察方向^[17]。视频的清晰度受相机和场景的距离影响,较小的距离会使得视频模糊,因此需要调整两者的间距,经过实验,本文发现将相机设置在 $(0, 0, 12)$ 的位置上图像颗粒感不明显,用户体验较好。投影矩阵决定了三维场景如何被投影到屏幕上。本系统采用了透视投影变换,因此利用透视除法,在观察平面指定一个裁剪窗口形成一个视锥体(图 7)。在具体实现时,由于人类的双目视觉(Binocular Vision)通常具有约 114° 的可视区域^[18],因此视锥体中的视场角(FOV, Field of View)取 110° ;近平面

和远平面距离观察点的间隔分别设置为 0.1 和 100。为了避免由于近平面和远平面各自的宽高比设置的不恰当所导致的场景失真现象,本文将近平面和远平面各自的宽高比和播放器界面的宽高比设为一致。

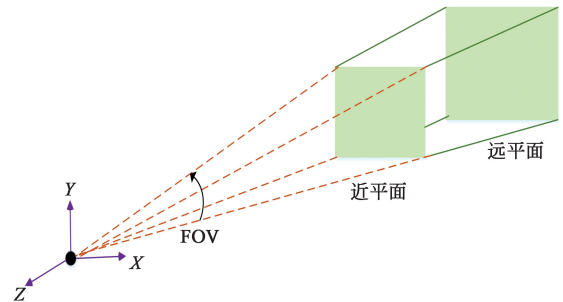


图 7 透视投影变换原理图

Fig.7 Perspective projection transformation diagram

将全景视频交互时产生的变换矩阵输入到渲染管线进行渲染,生成场景的二维投影,并映射到播放器的窗口中,从而得到全景视频交互时的正确显示。

3.2 帧渲染

帧渲染是循环渲染模块中的关键部分,此部分完成了视频中每一帧的绘制工作并将渲染结果在二维平面显示器上进行显示。在计算机图形学渲染流程中,帧渲染环节位于顶点处理之后,是事件处理环节的后续步骤。事件处理环节建立了用户交互操作时的变换矩阵,并调用顶点着色器来处理顶点相关的数据,将三维模型的顶点数据输入顶点着色器来完成相应的坐标变换。顶点着色器输出的数据被送入帧渲染环节,依次进行图元处理和片段处理。

图元处理时的关键步骤是进行图元装配,即将事件处理环节后离散的点按照一定的规则装配成点,线或者面片。本系统利用 OpenGL 中的相关函数进行图元处理,按照三角形连接的原则进行图元装配。在完成图元装配后,通过光栅化处理,将装配好的图元转换为片段。片段包含光栅化之后的顶点和像素信息,在全景视频显示系统中,通过片段着色器来对片段进行操作,对顶点的颜色值及纹理坐标进行插值运算,确定片段的颜色值,并进行纹理应用和凹凸映射。为保证显示系统的渲染结果正确,进行片元处理时,需要进行隐藏面消除。本系统采用了逐片元处理的模式来进行隐藏面消除,根据片元之间的遮挡关系,判断片元的可见性。通常使用深度检测的方法,利用片元携带的深度信息,来决定帧缓存中的内容是否更新。进行深度检测时,在相同的片段位置对当前片段的深度值与帧缓冲区中的深度值进行对比。若当前片段的深度值大于同一位置帧缓冲的深度值,则此片段被遮挡;若当前片段的深度值小于同一位置帧缓冲的深度值,则使当前片段通过深度测试,对相应位置的帧缓存内容进行更新。

帧缓存中的数据以纹理的方式进行处理,通过纹理贴图的方法将各帧数据贴到球体背景上。纹理贴图中,可以将当前纹理坐标周围的纹理单元的加权平均值应用到这个纹理坐标上,通常称为线性过滤,但是该技术会产生闪烁效果(锯齿假影),且容易带来额外的内存开销。因此本系统采用了 Mip 贴图技术,即为每一个特定几何图形选择具有最佳过滤效果的纹理。在片元处理阶段完成纹理映射后,帧缓存中的数据被映射到显示器窗口相应的部分上,完成显示器窗口的渲染流程。

通过本实验平台,用户可以在播放全景视频的同时,使用鼠标等外设拖曳、移动视频,多视角地观看全景视频,增强了沉浸感和真实感。相比于市面上现有的全景视频播放器,本平台秉承开源至上的原则,公开了核心代码,因此具有一定的拓展性。用户可以在拖曳的基础上,实现更多的功能和效果,例如控制拖动速度、进行三维场景漫游等。

4 显示效果与分析

本文选取了运行 Windows 10 64 位操作系统的主机作为测试平台,分析了该全景视频显示系统的性能。测试平台的相关配置参数如表 1 所示。

表 1 测试平台硬件配置

Tab.1 Test platform hardware configuration

硬件名称	配置
显卡型号	NVIDIA GeForce GTX 970
处理器型号	Intel i7 6700K 4.00 GHz
操作系统	Windows 10 64 位
显示器分辨率	3 840×2 160

结合设备特点和平台处理能力,本文选择了 24 个具有不同视频内容的 H.264 编码视频序列作为测试序列,视频分辨率有 720×360、960×480、1 280×640、1 920×960、2 160×1 080、3 840×1 920 等共计 6 种,每种分辨率的视频序列包括 25 fps (Frames Per Second)、30 fps、35 fps、40 fps 等 4 种不同规格的帧率。

相比于普通的全景视频播放器,本系统添加了鼠标拖曳的功能,因此对于拖曳的时延进行测试,从而探讨交互操作对于播放器性能的影响。测试时采用控制变量法,分别设置视频分辨率和帧率为单变量,统计每种情况下鼠标拖动产生的平均时延,为了更好地逼近时延的真实值,实验中对每种情况进行了 30 次拖动,并计算单次拖动的平均时延。实验结果以折线图的形式呈现,图 8 对应同一帧率不同分辨率下的单次拖动平均时延,图 9 对应同一分辨率不同帧率下的单次拖动平均时延。

从图 8 可以看出,当帧率一定时,随着视频分辨率的增大,平均拖动时延总体呈上升趋势;从图

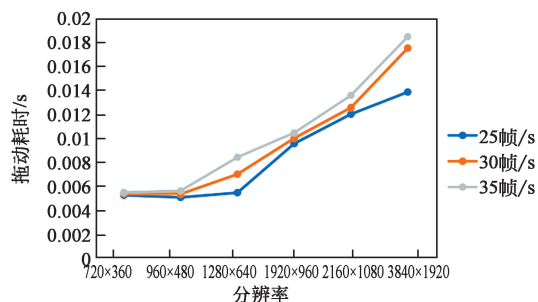


图 8 不同分辨率下的单次拖动平均时延

Fig.8 Average delay for single drag at different resolutions

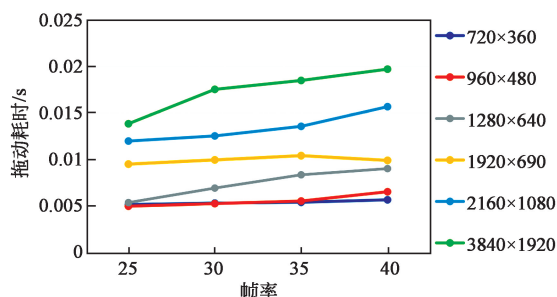


图 9 不同帧率下的单次拖动平均时延

Fig.9 Average delay for single drag at different frame rates

9 则可以看出,当分辨率一定时,随着视频帧率的增大,平均拖动时延也会出现小幅上升,即随着视频分辨率和帧率的增加,全景视频渲染的时间开销也逐步增加。在开销最大情况下(即以 40 fps 的帧率播放 4K 视频序列)本系统的交互时延仅为 0.02 s。由于人眼的视觉暂留效应(Persistence of Vision),在鼠标拖动的过程中人眼看到的图像能够保留约 0.1~0.4 s。因此本系统的交互时延远小于视觉暂留的时间范围,保证了交互过程的实时性。

综上所述,本系统能够按照用户的交互式操作指令进行全景视频的渲染显示。

5 结 论

随着虚拟现实技术的迅速发展,用户对全景视频的需求日益增长。本文以此为出发点,基于 OpenGL 图形渲染流程设计了一种交互式全景视频显示系统,并依托该系统实现了全景视频播放器。实验结果表明该播放器能够实时处理用户的交互式输入,输出用户请求的全景视频视口内容,在本平台的设置下,交互时延和视频分辨率、帧率的大小呈正相关关系,且交互时延不超过 0.02 s,为全景视频研究提供了工作基础和实验平台。

参 考 文 献:

- [1] QIAN F, JI L S, HAN B, *et al.* Optimizing 360 video delivery over cellular networks [C]//*Proceedings of the 5th Workshop on All Things Cellular: Operations, Applications and Challenges*. New York: ACM, 2016: 1-6..
- [2] GHIMIRE S. Production of 360° video: Introduction to 360° video and production guidelines [D]. Helsinki: Helsinki Metropolia University of Applied Sciences, 2016: 8-12.
- [3] SZELISKI R, SHUM H Y. Creating full view panoramic image mosaics and environment maps [C]//*Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques*. New York: ACM Press/Addison-Wesley Publishing Co., 1997: 251-258.
- [4] NG K T, CHAN S C, SHUM H Y. Data compression and transmission aspects of panoramic videos [J]. *IEEE Transactions on Circuits and Systems for Video Technology*, 2005, 15(1): 82-95.
- [5] HUANG K C, CHIEN P Y, CHIEN C A, *et al.* A 360-degree panoramic video system design [C]//*Technical Papers of 2014 International Symposium on VLSI Design, Automation and Test*. Hsinchu, Taiwan, China: IEEE, 2014: 1-4.
- [6] FU C W, WAN L, WONG T T, *et al.* The rhombic dodecahedron map: An efficient scheme for encoding panoramic video [J]. *IEEE Transactions on Multimedia*, 2009, 11(4): 634-644.
- [7] LIU Y F, YANG L, XU M, *et al.* Rate control schemes for panoramic video coding [J]. *Journal of Visual Communication and Image Representation*, 2018, 53: 76-85.
- [8] ZONG T Y, TU Z Z, AI L, *et al.* Panoramic video delivery based on Laplace compensation and Sphere-Markov

- probability model [C]//*Proceedings of 2018 IEEE International Conference on Consumer Electronics (ICCE)*. Las Vegas, NV, USA: IEEE, 2018: 1-4.
- [9] 孔令胜,南敬实,荀显超. 平面三维显示技术的研究现状[J]. *中国光学与应用光学*, 2009, 2(2): 112-118.
KONG L S, NAN J S, XUN X C. Research status quo of flat 3-D display technology [J]. *Chinese Journal of Optics and Applied Optics*, 2009, 2(2): 112-118. (in Chinese)
- [10] 赵星,王芳,杨勇,等. 集成成像三维显示系统显示性能的研究进展[J]. *中国光学*, 2012, 5(3): 209-221.
ZHAO X, WANG F, YANG Y, *et al.* Research progress of display performance of integral imaging three-dimensional display system [J]. *Chinese Optics*, 2012, 5(3): 209-221. (in Chinese)
- [11] CHENG Y. Progress in display technologies and their applications [J]. *Chinese Journal of Optics and Applied Optics*, 2010, 3(1): 27-32.
- [12] ZORIC G, BARKHUUS L, ENGSTRÖM A, *et al.* Panoramic video: design challenges and implications for content interaction [C]//*Proceedings of the 11th European Conference on Interactive TV and Video*. Como, Italy: ACM, 2013: 153-162.
- [13] NENG L A R, CHAMBEL T. Get around 360° hypervideo [C]//*Proceedings of the 14th International Academic MindTrek Conference: Envisioning Future Media Environments*. Tampere, Finland: ACM, 2010: 119-122.
- [14] 施珂奕,邓春健,邹昆. 基于 OpenGL 的三维模型点坐标拾取方法[J]. *液晶与显示*, 2016, 31(7): 708-713.
SHI K Y, DENG C J, ZOU K. Point pick-up in three-dimension model based on OpenGL [J]. *Chinese Journal of Liquid Crystals and Displays*, 2016, 31(7): 708-713. (in Chinese)
- [15] RAKKOLAINEN I, RAISAMO R, TURK M, *et al.* Casual immersive viewing with smartphones [C]//*Proceedings of the 20th International Academic Mindtrek Conference*. Tampere, Finland: ACM, 2016: 449-452
- [16] HEARN D D, BAKER M P, CARITHERS W R. *Computer graphics with OpenGL* [M]. 4th ed. Upper Saddle River: Prentice Hall Press, 2010.
- [17] SHREINER D, SELLERS G, KESSENICH J, *et al.* *OpenGL Programming Guide: The Official Guide to Learning OpenGL* [M]. version 4.3 8th ed. Reading: Addison-Wesley, 2013.
- [18] HOWARD I P, ROGERS B J. *Binocular Vision and Stereopsis* [M]. New York, USA: Oxford University Press, 1995.

作者简介:周凡(1994—),男,江苏南通人,硕士研究生,研究方向为计算机图形学与虚拟现实。E-mail: fzhou_945@stu.xidian.edu.cn