

# 虚拟演播三维实时渲染引擎控制器<sup>①</sup>

胡芯菲, 彭四伟

(北京化工大学 信息科学与技术学院, 北京 100029)

**摘要:** 研究了虚拟演播三维实时渲染引擎控制器的设计中架构、通信方式以及控制协议等相关问题. 虚拟演播三维实时渲染引擎是虚拟演播技术核心技术部件, 除了在渲染功能上与普通三维图形渲染引擎的区别, 更重要的一个区别的在于面向虚拟演播的应用的控制器的设计. 控制器的架构设计满足演播的要求, 同时演播过程中对场景的控制需要通过控制器传输到渲染模块, 控制器与渲染模块的通信方式设计为协议驱动的 TCP 方式, 它保障了渲染的实时性同时也能保证控制协议的可扩展性, 便于后续的研究及开发.

**关键词:** 虚拟演播; 控制器; TCP 通信; 协议驱动; 控制协议

## Controller of Virtual Studio Three-Dimensional Real-Time Rendering Engine

HU Xin-Fei, PENG Si-Wei

(College of Information Science and Technology, Beijing University of Chemical Technology, Beijing 100029, China)

**Abstract:** A controller of virtual studio three-dimensional real-time rendering engine is proposed in this paper, and the research focuses on the architecture, communication method and control protocol. The virtual studio three-dimensional real-time rendering engine is the core technology of virtual studio. It is different with the general three-dimensional rendering engines in the rendering function and especially in the design of controller for virtual studio application. The architecture design of the controller meets the demands of virtual studio, which is a control command transfer from the controller to the rendering module. The communication method is designed to be a protocol driven TCP communication method. It not only ensures the real-time feature of rendering, but also guarantees the scalability of control protocol, which will facilitate the follow-up research and the application development.

**Key words:** virtual studio; controller; TCP communication; protocol driven; control protocol

随着虚拟现实技术的飞速发展, 该项技术逐渐深入到我们的生产生活当中. 在军事、医疗、工业设计、娱乐、教育和通讯等诸多行业中, 虚拟现实技术在其发展和研究过程中都起到重要作用<sup>[1]</sup>. 而在现代演播技术中, 虚拟现实技术更是带来了翻天覆地的改革, 虚拟演播技术应运而生. 目前虚拟演播技术主要以商业应用开发为主, 如以色列的 ORAD(傲威)的 ProSet 高清虚拟演播室系统完成了 2008 年北京奥运会的转播, 国内的新奥特公司的 NASET 系统被中央电视台应用于 2004 年雅典奥运会的直播. 在这些商业系统中, 其虚拟演播三维实时渲染方法都是针对自己的系统进

行的相应开发, 具有庞大、复杂以及硬件依赖等特点, 对于中小型以及个人用户来说是个很大的负担, 因此将虚拟演播三维实时渲染引擎作为一个独立的个体, 研究一个轻量级、简单、可扩展的系统是很有必要的. 本文将针对虚拟演播技术, 利用 OGRE 作为基础对虚拟演播三维实时渲染引擎的设计进行研究.

伴随着虚拟现实技术和计算机图形学与演播技术的共同研究, 为现代媒体技术翻开了新的篇章, 为演播技术带来了一番天翻地覆的变革. 由虚拟现实技术和计算机图形学在媒体行业研究应用衍生而来的虚拟演播技术不仅推翻了传统演播技术的模式, 而且还为

<sup>①</sup> 收稿时间:2016-04-18;收到修改稿时间:2016-05-19 [doi:10.15888/j.cnki.csa.005531]

演播及节目的制作带来了更多的可能性和多样性。

虚拟演播技术主要分为两个部分：实景拍摄部分和虚拟场景部分<sup>[2-4]</sup>。实景拍摄是通过摄像机拍摄蓝箱或者绿箱中的实际演播过程场景，通过色键处理器去除蓝箱或者绿箱背景得到实景拍摄的场景中所需要的内容。虚拟场景部分是由虚拟演播三维实时渲染引擎经过实时渲染得到的演播过程中所需要背景画面。将实景场景和虚拟场景通过图像合成器进行合成处理，就完成了呈现在电视大屏幕上给观众观赏的电视节目<sup>[5-10]</sup>。

虚拟演播三维实时渲染引擎与普通三维图形渲染引擎相比，除了在三维图形渲染上具备虚拟演播的特色，更重要的是它拥有一个适用于虚拟演播的独特的控制器。该控制器在演播过程中采用 TCP 方式向渲染模块发送控制命令对虚拟场景以及场景中的实体进行控制。

## 1 系统架构概述

虚拟演播三维实时渲染引擎的总系统架构分成三部分：场景文件系统、渲染模块、控制器模块。系统总体框图如图 1 所示。

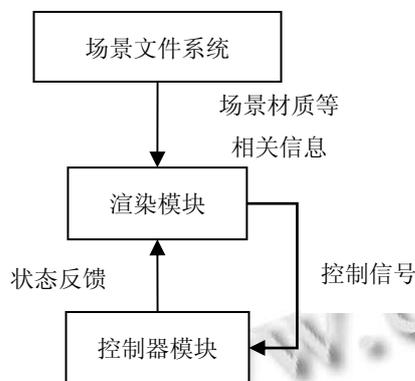


图 1 系统总体框图

**场景文件系统：**采用 XML 格式来描述一个系统中的相关信息的组织规范及定义。本文研究的虚拟演播三维实时渲染引擎的场景文件系统是为了给渲染引擎提供相关的模型文件信息、渲染过程中所需参数，以及动态特效描述文件。

**渲染模块：**渲染模块以场景文件系统提供的场景文件作为原料对虚拟背景场景以及场景内的实体进行渲染，同时接收控制命令对场景的控制进行响应并将

当前场景状态向控制器反馈。

**控制器模块：**采用协议驱动的 TCP 方式进行控制器模块与渲染模块的消息通信，同时为了保证控制器的灵活性和可扩展性设计了通用的控制协议。

## 2 控制器设计与实现

### 2.1 控制器架构

普通的三维实时渲染引擎仅仅只是完成定制的实时渲染任务，但是电视演播要求渲染引擎能够跟随实景拍摄的情况以及节目设计的适时调整并实时地完成相应的变化，让虚拟背景与实景相匹配，效果不失真<sup>[11,12]</sup>。因此，在演播过程中渲染引擎需要能够实时接收控制命令，并做出及时响应，控制模块便是完成传送控制命令这一需求的模块。

在控制部分的设计上，为了防止资源竞争以及命令冲突产生错误，采用同步方式进行控制，控制端向渲染引擎发送控制指令之后必须等到其收到渲染引擎发回的指令反馈才能继续发送下一条指令。在控制端与渲染引擎之间的消息通信采用 TCP 方式，控制端首先建立与渲染引擎的安全连接，并担任管理连接的角色。演播过程中，在控制端对命令进行设置，包括控制对象、描述动作指令的相关参数，写入控制协议定义的数据格式中，命令数据包进入待发送状态。控制端检查其是否处于命令发送预备状态，状态检查完毕将命令发送至渲染引擎。渲染引擎接收到命令后对命令进行检查，如果检查为错误命令则发送错误反馈至控制端，控制端进入新一轮命令发送预备状态；如果检查为正常命令则由渲染引擎执行命令，执行完毕将命令反馈发送给控制端，控制端恢复预备状态。经历以上过程，控制端完成一次对渲染引擎的控制命令发送。

根据以上设计，控制模块的控制流程如下：

- (1) 控制模块在建立与渲染引擎的连接并完成控制协议数据包的封装；
- (2) 控制器进行自我状态检查；
- (3) 将控制命令发送至渲染引擎，并等待接收命令反馈(错误反馈、命令正常完成反馈、命令执行异常反馈等)；
- (4) 控制模块进入下一次命令发送预备状态。

其详细控制流程如图 2 所示。

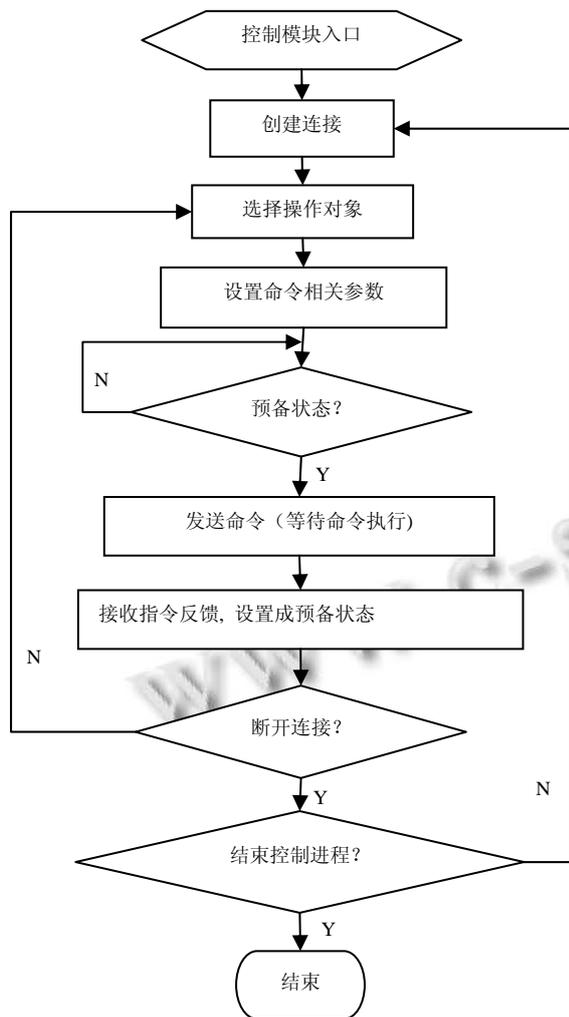


图2 控制模块流程图

控制模块的研究设计上主要包含两个部分：摄像机控制部分和场景动态变化效果控制部分。在实际的演播过程中，根据实际拍摄场景中的摄像机的摇移情况，虚拟场景中的摄像机需完成以实际对应的摇移控制，使得背景和实际拍摄的实景画面拍摄深度、角度和方位相契合，让效果更逼真。在摄像机控制部分的设计上，主要使用摄像机自身坐标来进行相对位移的计算，进行摄像机摇移。

场景控制部分则主要涵盖材质模型文件中定义的动态特效的播放控制和演播过程中的实时动态特效的设置和播放。对于已经在材质模型文件中定义好的动态特效，在演播过程中只需控制它的播放和停止。在演播过程中，可能会出现需要临时添加的动态特效，如在新闻直播中出现需要及时滚动播放的紧急通知，

在这种情况下就需要实时地在控制器上设置动态特效相关命令发送给渲染模块进行渲染，控制模块发送命令控制其播放。

## 2.2 协议驱动的 TCP 服务设计

从对虚拟演播三维实时渲染引擎的系统架构的研究可以看到，本文研究的虚拟演播三维实时渲染引擎中的控制模块和渲染模块是两个独立的模块，在演播中，使用控制模块向渲染模块发送控制命令，在这两个独立的模块之间需要进行沟通来完成命令的传送及执行，这就需要消息通信机制来协助完成。虚拟演播技术，对实时性的要求比较高，因此在消息通信机制方面也要保证通信效率来保证实时性。同时在消息传输服务的设计上，还要考虑命令执行的方式以及对消息传输协议的扩展性的支持。

消息通信主要依靠模块的传输协议及传输服务，将封装了控制信号以及其他相关信息的数据包从控制端传输到客户端。在演播过程中，控制信号的传输要保证效率及安全性，因此控制模式选择 TCP 方式进行控制信号的传输。同时，考虑到渲染过程中操作的正确性，采用同步的方式进行控制命令的传输来对虚拟场景完成控制。控制模块在每次发送控制命令之前进行自我检查，处于预备状态下才能发送命令数据包，同时修改目前的状态，直到渲染模块执行完命令接收到从渲染模块发送过来的执行完成信号再次将自己设置成预备状态，才能进行下一次的命令发送。

根据以上研究分析，控制模块的通信传输服务需要具备以下几个特点：安全、传输效率高、能够支持同步的方式，并且为了便于以后的研究及扩展，还需要能够扩展协议。本论文针对控制模块研究设计了基于 TCP 的、协议驱动的、可扩展的控制指令和消息通信方案。控制模块的核心是协议驱动的 TCP 通信模块，组成通信模块的两个基本服务类是负责数据通信的 Connection 类和负责连接请求监听的 Listener 类。

Connection 主要负责创建一个 TCP 连接，提供以轮询的方式接收字节流。TCP 服务通过远程主机号和端口号来建立远程连接，指定所使用的 TCP 客户端对象，启动工作进程。创建的连接规定设置工作线程在空闲时的扫描时间、两次数据接收的最小时间间隔、关闭连接时等待工作线程的结束的时间以及发送心跳的时间间隔，这几个时间参数用于判断连接是否有效、何时失效以及判断连接是否正确关闭等，以确保

连接的安全性和有效性。Connection 负责具体的数据通信过程,但不负责处理数据的逻辑协议,而是由指定的协议类的实例来负责逻辑协议的解析处理。服务将遵循通讯协议将控制命令数据包发送到渲染端,同时也发送其运行时的信息以及收到字节块和数据帧的通知。图 3 对 Connection 类的作用进行说明。

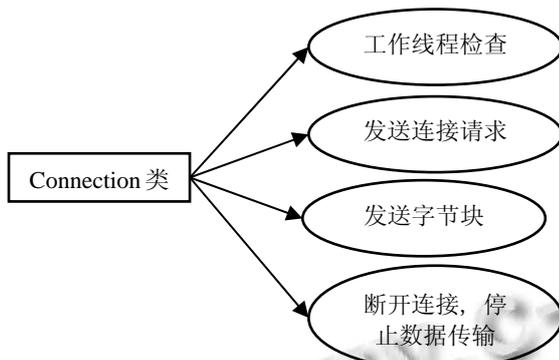


图 3 Connection 类的作用说明

Listener 负责监听指定的端口,响应连接请求,管理已建立的连接。在监听线程创建过程中,将指定要监听的端口并且准备通讯协议的协议工厂。监听启动之后,监听各个端口的状态并进行记录,对已经建立的连接进行管理,可关闭及删除指定的连接或者所有连接进行广播通知。在监听线程关闭时,同时将清理掉其所建立连接。图 4 对 Listener 类的作用进行说明。

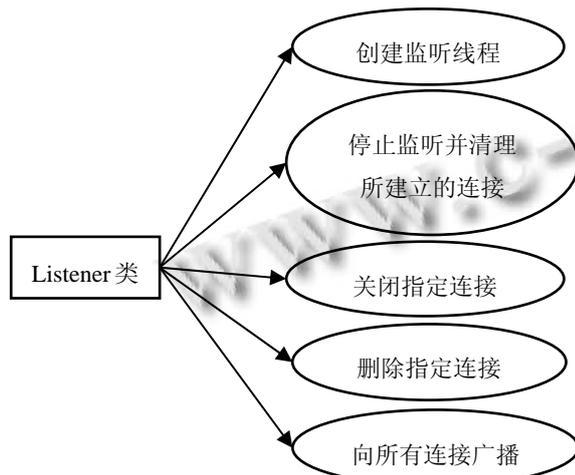


图 4 Listener 类的作用说明

为了保证 TCP 服务的灵活性和可扩展性,本文研究虚拟演播三维实时渲染引擎控制模块的消息通信机制中的 TCP 服务是可扩展协议的,因此在前面介绍的

TCP 服务框架中的通讯协议基类是可以由使用及开发者根据具体的需求进行个性化的定制设计。在研究设计中的通讯协议基类是以数据按帧为单位进行处理,通讯协议负责将一组字节块解析成一组数据帧。经过研究,数据帧的构成规则如下:首先,输入为数据片流(每次收到的一个字节块称为一个数据片),输出为数据帧流;其次,帧起始不一定在数据片的起始位置;再者,一帧可能分割到一组连续的数据片中,同时一个数据片可能包含多个帧。数据帧从控制模块传送到渲染模块的策略设计如下:

(1) 在预备状态时,等待含有帧起始的数据片,然后进入接收状态;

(2) 在接收状态下,若收到一个完整帧,则解析出该帧;

(3) 若提取出数据帧后,没有新的帧起始,则清空接收存储,进入预备状态。

为了更好地保证本文设计的 TCP 服务的协议可扩展性同时降低以其他模块的耦合性,本文研究设计的协议类只负责帧的解析,不负责帧的处理,通过通知事件将收到的数据帧发送出去,由其他专门负责处理的对象来完成帧的处理。这样的设计方式让协议类可扩展性以及与其他功能模块的隔离性表现的比较好,但是增加了整个系统的复杂度,增加了相应的处理类和对象的管理。在本文的研究中,控制协议的通用格式中针对具体命令中的参数存在较大的差别,以这种方式设计的协议类能够让控制协议的不受到命令参数的限制。

在本节研究的 TCP 服务中以 TCP 协议为基础,总体架构是三部分构成:创建连接、数据传输、断开连接。创建连接阶段,为了保障连接的安全性及有效性,在连接之前检查相关的工作进程确定远程连接的参数及对象,连接中设置保持连接有效期的相应时间参数。在数据传输过程中,按照数据帧的传输策略进行传输,其中对于数据片存在一个缓冲区来接收字节流。在断开连接的过程中,不仅仅是单单关闭连接,同时还要确保将网络流关闭、结束工作线程才能彻底停止数据的接收。

### 2.3 渲染引擎控制协议

控制模块通过 TCP 服务将控制命令封装进数据包传输到渲染模块,让渲染模块能够及时地完成演播中的对渲染的控制需求。在对虚拟场景的控制上,控制

命令的种类及其组合有很多种,如果每一种命令都自成一格式,这种情况下对控制命令的解析只能采取穷举的方式来进行.穷举的解析方式首先对效率的影响较大,将影响虚拟演播三维实时渲染引擎的实时性.另外,对于控制命令来说,在不同的情况下定义的命令都不相同,而对于节目制作,设计的多样性和多边形决定了控制命令不能以穷举的方式出现,这样它的扩展性及灵活性将大大降低.因此,需要一个渲染引擎控制协议来规定控制指令的模式,让控制命令在一个统一的范式下进行编码封装成数据包经由TCP服务从控制端传输到渲染模块.

操作命令主要是指明在什么时候由何种对象对另外的对象进行了怎样的操作,控制协议则是针对操作命令的描述的规范.在这个协议中要规定以下信息:命令执行时间、执行者、被操作者已经相关操作信息.在虚拟演播中,操作命令是即时发送到渲染模块,由渲染模块立即实施执行,因此本节控制协议中命令执行时间不需要进行描述.而虚拟演播三维实时渲染引擎的控制模块发送的控制命令都是送达到渲染模块,由渲染模块来执行,所以渲染模块中同样也不需要申明命令执行者的信息.因此,在本文的控制协议中要规定被操作者以及操作命令的描述编码规范.

在本文提出的虚拟演播三维实时渲染引擎的研究中,控制模块主要是在演播过程中对场景的控制.场景的构成除了场景背景,还包括场景的物体,比如模型实体、摄像机、灯光等,这些都将成为控制模块的控制对象.在这些控制的对象中,根据场景归属范围将其分为两个层次,第一个层次是对于场景整体,第二个层次是场景内的所有实体的集合.基于这样的组成结构,控制协议中的控制对象的表示有两部分组成,第一部分指明操作对象在那个场景,第二部分说明操作对象是指定场景中的哪一个物体.而操作命令则由命令指令及其所需要的参数构成,图5所示则是控制协议指定的控制命令数据包格式构成.

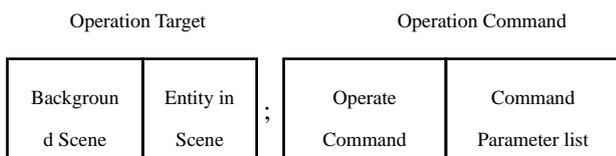


图5 控制命令数据包格式

在数据传输过程中,数据包的数据量是决定传输速率,从而影响虚拟演播三维实时渲染引擎的实时性,因此在数据包的各个部分的表示上用尽可能小的数据量进行描述.对于场景、场景内实体以及操作命令均由场景文件信息载入时存储的标号表示,操作命令使用的参数将按序封装发送给渲染端.场景文件信息解析之后的结果将以如下形式进行存储,如图6所示.

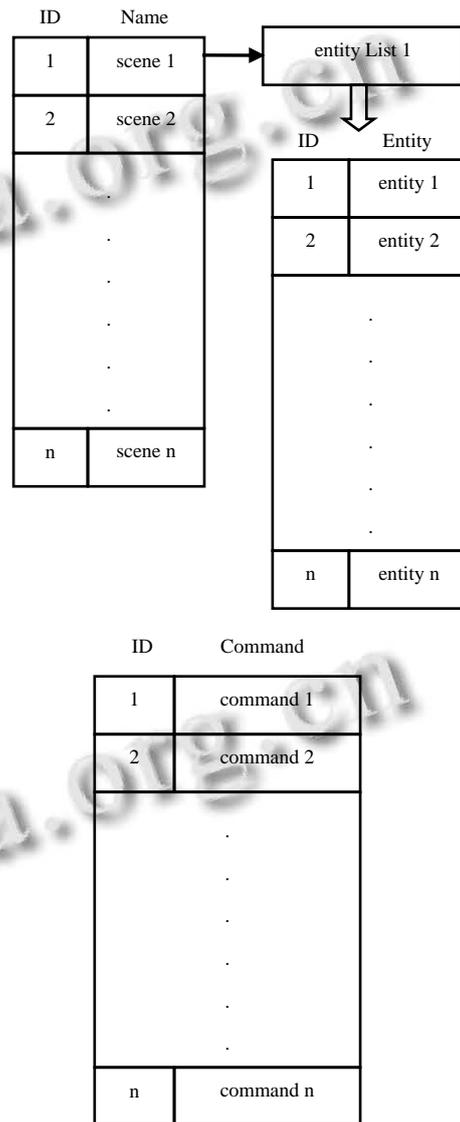


图6 场景文件解析及命令存储形式

在演播过程中,控制命令的控制对象以及控制命令的编码主要转化成为其对应的序列标号,传输到渲染模块后使用标号对应于场景文件和命令的存储结构体进行解析以执行.控制命令编码遵循如下策略:

- (1) 场景位不可空缺,场景内实体位可缺省.场景

内实体位缺省时表示其对整个场景包含场景内所物体的操作; 否则操作命令的执行对象是指定场景内的指定实体;

(2) 操作命令中命令位补课空缺, 参数序列可空缺. 参数序列的按照命令构成的参数初始化顺序进行编码, 解析后按序对相应的对象中的成员进行设置让控制命令对控制对象执行.

## 2.4 实验

在控制器模块, 实际演播过程中的控制信号将会根据发展和设计产生变化或者进行相应的增加, 本文的研究在控制协议方面, 研究设计了通用的渲染引擎控制协议. 产生变化或者增加的控制信号, 按照控制协议中规定的何时进行编码封装, 即可通过 TCP 服务将控制信号数据包从控制端传送到渲染模块.

在实验中, 设计了三个实验对虚拟演播三维实时渲染引擎的控制器进行了. 第一个实验是在演播过程中添加渲染虚拟场景内的实体的实验, 实验结果如图 7 所示; 第二组实验是对摄像机的摇移控制, 实例中摄像机的坐标位置为(0,10,-100), 控制命令位将摄像机从当前位置摇移到原点, 对五个不同的时刻进行了截图, 每张截图对应的是摄像机向 Z 轴负方向移动 20 个单位, 实验结果如图 8 所示; 第三组实验是通过控制器向渲染模块场景内实体的圆形动态特效的实验, 实验结果如图 9 所示, 图 10 为圆形动态特效的运动合成结果.



图 7 第一组实验结果

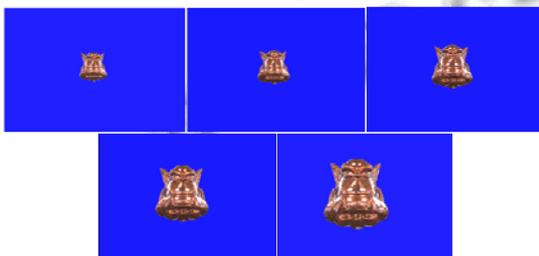


图 8 第二组实验结果

图 7 至图 9 三组图给出了相应的针对控制模块给出的三个实验结果. 在图 7 中, 左图给出了虚拟演播三维实时渲染引擎渲染得到的空间背景场景图, 右图给出了演播过程中控制器向渲染模块发出实体渲染的信息, 渲染模块实时渲染出空间背景中实体的截图,

在实体的渲染过程中并未影响到空间场景背景渲染结果. 图 9 说明的是向渲染模块发出摄像机摇移的信号, 让摄像机从当前的位置沿 Z 轴负方向移动到原点位置, 摄像机摇移过程中, 摄像机镜头离物体越来越近, 因此显示出来的实体越来越大. 图 9 是演播过程中控制模块向渲染模块发送实体的圆形动态特效的指令, 指令中包含背景场景及实体的编号、圆形动态特效的指令以及唯一表示圆形运动的相应, 渲染模块渲染得到其圆形运动八个时刻的截图, 在图 10 是对运动轨迹的合成效果图, 从图 10 可以看出指令执行的效果确实是圆形的运动轨迹. 通过以上的实验可以看到, 通过控制模块发送给渲染模块的指令能够被正确执行.



图 9 第三组实验结果

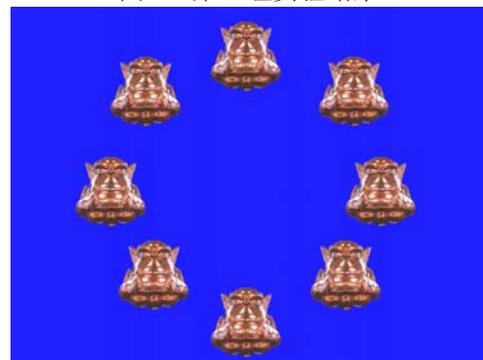


图 10 第三组实验圆形动态特效合成图

### 3 结语

本文重点研究设计了虚拟演播三维实时渲染引擎的控制器,在虚拟演播三维实时渲染引擎的总体系统架构的基础下,结合演播的需求,设计控制器。同时为了保证控制器与渲染模块的通信,设计了协议驱动的TCP通信方式,在该设计下,控制协议在通用控制协议框架下,能够根据实际演播过程中的要求将控制协议进行具体的实例化,使得其更灵活实用。

虚拟演播技术是媒体技术的发展前进的一大步,将虚拟演播三维实时渲染引擎作为一个独立的个体进行研究,设计一个轻量级、简单易用且扩展性良好的虚拟演播三维实时渲染引擎为虚拟演播技术,便于技术的发展及传播。

#### 参考文献

- 1 Gibbs S, Arapis C, Breiteneder C, et al. Virtual Studios: An overview. *IEEE Multimedia*, 1998, 5(1): 18–35.
- 2 Blond L, Buck M, Galli R, et al. A virtual studio for live broadcasting: The mona lisa project. *IEEE Multimedia*, 1996, 3(2): 18–29.
- 3 梁骥. 数字虚拟演播室的研究与应用探索. *中国有线电视*, 2013,(4):501–506.
- 4 陈蕙. 全媒体交互式演播室的设计与建设. *电视技术*, 2014, 38(2):77–79.
- 5 刘跃. 电视事业战略发展与虚拟演播室技术. *中国传媒科技*, 2012,(15):70–73.
- 6 Rodríguez GR, Montesa AR. A method for obtaining a virtual object within a virtual studio from a real object: EP. EP2736247. 2014.
- 7 Griesert A. Method for generating a picture in a virtual studio: US. US20020030692. 2002.
- 8 Lin SW, Chou HC, Lin CM, et al. Method and system for media production in virtual studio: US. US20080232778. 2008.
- 9 刘霞,张砚. 虚拟演播室系统新议. *电视技术*, 2011,35(2): 90–91.
- 10 朱巍峰,李震阳. 数字化三维虚拟演播室系统的设计与实现. *计算机技术与发展*, 2010,20(12):213–216.
- 11 齐越,胡晓峰,张茂军. 虚拟演播室:结构及关键技术. *中国图象图形学报*, 2000,5(6):457–460.
- 12 Jeong JS, Park C, Han JJ, Im MS, Jang RH, Kim M, Yoo KH. Development of a 3D virtual studio system for experiential learning. *Advanced Computer Science and Information Technology*, Springer Berlin Heidelberg, 2011, 195(10): 78–87.