基于视频实时处理的多投影曲面拼接系统的研究

谢逸群 王慧雅 许华虎
(上海大学计算机科学与工程学院,上海 200444)

摘 要 针对多投影面系统的曲面拼接问题,提出了一种视频实时处理方法,该方法首先通过 Filter-Pipeline 链路模型很好地在普通 PC 上实现了在保证视频流畅播放的条件下,对视频源进行实时处理;然后再进行保持边缘平滑的非线形几何校正和保持边缘亮度的均衡过滤(filter)。采用这一方式构架的系统不仅具有很强的灵活性,并且还能够在取得高质量场景效果的同时,大大降低构建成本,从而提高了系统的性价比。此外,通过 F-P 模型中 filter 的增加和修改,还可以很容易对系统的功能进行修改,并为系统的扩充提供了支持。

关键词 虚拟现实系统 Filter-Pipeline 模型 非线性几何校正 边缘平滑 亮度均衡
中图法分类号: TP391.9 **文献标识码**: A **文章编号**: 1006-8961(2009)02-0286-06

Achieving Multi-projector Displays Stitching Based on the Real-time Video Processing

XIE Yi-qun, WANG Hui-ya, XU Hua-hu
(*School of Computer Engineering and Science, Shanghai University, Shanghai* 200444)

Abstract By using F-P(Filter-Pipeline) model, this paper proposed a way to stitch multi-projector displays in real-time video processing. The model takes multithreading technology which can process several frames at the same time. It ensures that the video can be played fluently with a high quality effects. With this model, the system can be easily constructed based on the common PCs and benefits us with reduced cost. Furthermore, the system constructed with F-P model can adapt to a new environment with the extensibility to add or change the filters in the model. Here, this paper also gives two examples in designing "geometrical transform filter" and "edge smoothing filter" and approved to be well done in real-time video processing.

Keywords VR-system filter-pipeline model no-linear geometrical transform edge smoothing intensity blending

1 引 言

多投影面系统[1-3]作为一种大屏幕虚拟现实系统,它可以通过对多个投影面进行拼接来展现宽视角、高分辨率的虚拟场景,以使得参与其中的观众感受到强烈的沉浸感。其在产品展示、虚拟漫游、环幕电影、教育培训等领域都存在着特有的应用价值。随着普通 PC 机处理速度的不断提高,基于 PC 的多投影面系统已经成为了现今研究发展的热点,而且其具有的远低于专业图形工作站的构件成本以及便于灵活扩展的特点更是它最大的优势,从而为其应用领域的扩展奠定了基础,但同时这也对视频源的处理提出了更高的要求。

通常视频源的处理是采用非即时处理的模式,即需要首先对各通道的视频分别进行预处理(预处理过程就是对视频进行非线性几何校正以及进行边缘亮度均衡等操作),然后将处理后的视频直接进

基金项目:上海市教委发展基金项目(A.10-0503-05-001)
收稿日期:2007-06-15;**改回日期**:2007-08-14
第一作者简介:谢逸群(1981~),男。上海大学计算机应用专业硕士研究生。主要研究方向为多媒体技术、图形学。E-mail:xyq0324@126.com

行播放。这一方法的优点在于对系统的要求不高,但其前期工作却非常的繁琐(如:需要对播放场景进行精确的测量分析),这不仅会引入人为产生的不确定性,而且效果一般、灵活性差,不能适应播放场景的变化(如:投影仪灯泡老化、位置变动,更换视频源等)。针对预处理方式的不足,本文提出了一种基于视频实时处理的方法,其思想是利用 PC 机多线程并行处理的优势,先将复杂的算法分解成若干简单的算法子块,然后通过这些子块的协同工作来同时对多帧画面进行处理,这样就可以充分利用 CPU 资源来缩短帧间处理时间,以提高视频处理速度,并保证实时处理下的视频仍然能够流畅播放。通过对视频进行实时处理不仅克服了预处理模式的缺点,而且还增强了系统的通用性和灵活性,本文就环幕电影播放系统中视频的实时处理的实现进行了研究,并给出了非线形几何校正以及边缘亮度均衡的算法。

2　基于 F-P 模型的视频实时处理

视频的预处理通常是采用单线程模式,也就是只有当前一帧画面的所有操作都结束以后才会对下一帧进行处理,但这种处理模式效率低,对系统资源的利用率不高,由于前后帧的间隔时间(处理完一帧画面所需要的时间)过长,因此不适合视频播放的实时处理。目前随着 PC 机对多线程的广泛支持,本文提出了一种基于多线程的过滤器-流水线(F-P)模型用来实现对视频播放进行实时处理。

2.1　Filter Pipeline 模型

F-P 模型(如图 1 所示)是采用“流水线”协同工作的方式来实现对媒体流的管理,其中参与数据处理的各个功能模块称为“过滤器(filter)”,它们被有序串联在一条链路中。按照功能不同,filters 可以被分为以下 3 类:源过滤器(SFilter)、变换过滤器(TFilter)和目标过滤器(RFilter)。其中 SFilters 负责获取数据,TFilters 负责数据的处理转换,本文对视频的处理就是在 TFilters 中实现的,而 RFilters 则决定了数据的最后去向,其可以输出到终端,也可以输出到文件等,要说明的是,这 3 个部分往往都是由多个 filter 共同工作实现的。

从图 1 可见,由于 F-P 模型独立于播放程序,因此当需要增加新的视频处理模块时,只需要在 F-P 链路中添加相应的 TFilter,而不用对播放程序进行修改,这就为系统功能的扩充提供了很好的支持。此外,TFilter 在处理每帧画面之前都会通过读取参数来对算法进行调整,这使得播放效果的实时调节成为可能,同时也提高了系统对环境的适应能力。

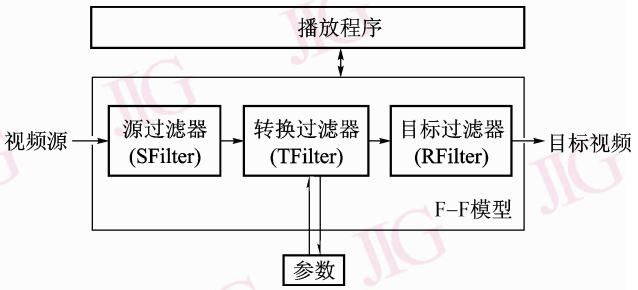

图 1　F-P 模型图

Fig. 1　F-P model

2.2　视频实时处理的实现

在视频过滤器链路中,视频流先由 SFilter 引入后,再流经链路中的每一个 TFilter,最后通过 RFilter 流向最终目标。这一模式与单线程模式的主要区别在于,其是将视频的处理交由多个过滤器共同完成,并且能够对多帧画面同时进行处理。如在图 2 中,每一帧都要流经 TFilter1 和 TFilter2,并且当TFilter2 在对第 i 帧 $\boldsymbol{F}_i$ 进行处理的同时,TFilter1 也在对下一帧 $\boldsymbol{F}_{i+1}$ 进行处理。

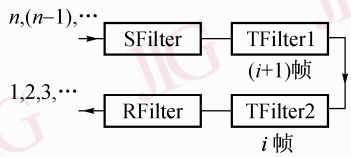

图 2　视频 filter 链路

Fig. 2　Video filter pipeline

通过分析可以很容易得到对 filter 链路性能进行评判的标准,假设链路中有 n 个 filter 分别为 f_1, f_2,⋯,f_n,处理一帧画面所耗费的时间为 $t(f_1)$, $t(f_2)$,⋯,$t(f_n)$,则

$$\begin{cases} \text{帧间延时} \\ \quad t_{\mathrm{FDT}} = \max(t(f_1), t(f_2), \cdots, t(f_n)) \\ \text{缓冲延时} \\ \quad t_{\mathrm{CDT}} = \sum_{i=1}^{n} t(f_i) \end{cases} \tag{1}$$

式中,“帧间延时(frame delay time)t_{FDT}”是指视频播放时,相邻两帧画面之间的最大时间差,它的值为所有 filter 中处理速度最慢的那个 filter 所耗费的时

间;“缓冲延时(cache delay time)t_{CDT}”是指第 1 帧画面从 SFilter 流入开始一直到从 RFilter 流出所经历的时间,其反映了视频播放前所花费的缓冲时间,其值为所有 filter 处理时间之和。为了简化计算,通常情况下不考虑 SFilter 和 RFilter 所耗费的时间。

如果图 2 中 TFilter2 的时间大于 TFilter1 的时间,则即使当 TFilter1 已经完成了第 i 帧 $\boldsymbol{F}_i$ 的操作也不能立刻传给 TFilter2,此时第 i 帧 $\boldsymbol{F}_i$ 及其后续帧都必须等待,直到 TFilter2 处理完为止,因此对于这条链路的 t_{FDT} 就是 TFilter2 处理图像的时间。如果 t_{FDT} 值超过某个限度,则视频播放就会出现卡帧现象,虽然一般可以通过拆分速度慢的 filter 来降低 t_{FDT} 值,但同时也可能引起 t_{CDT} 值的增加,而当 t_{CDT} 值超过它的阈值后,则会因视频播放不同步而引起画面“撕裂”,并同样可能会因超过 CPU 的最大处理能力而产生视频卡帧。因此在对 filter 链路进行设计时,必须同时兼顾 t_{FDT} 和 t_{CDT},并将其控制在 CPU 能够承受的范围之内,这就对图像处理算法的速度提出了较高的要求。

3 算法及功能描述

对图像处理的顺序会直接影响到算法的复杂度以及最终实现的效果,由于相邻图像的融合区是在图像还是平面的时候确定的,因此要对图像先进行平面融合,然后通过几何校正后,再进行曲面拼接,这不仅可以降低融合算法的复杂度,而且还可以保证拼接的效果。如果采用了相反的顺序,则会增加融合算法的复杂性。

3.1 边缘亮度均衡

相邻投影面的拼接是最关键的一个部分,拼接的好坏将直接影响到最终的拼接效果。拼接方式有硬拼接和软拼接两种,其中硬拼接就是直接对图像进行拼接,这种方法容易在图像接合处产生明显的拼缝;软拼接则是先在相邻图像的边界处设置重叠区,然后通过对重叠区进行重合来消除硬拼接产生的物理拼缝,但却在重叠区域产生了明显的光学亮带。对于光带的消除以及画面之间的平滑过渡最有效的解决方法是采用边缘衰减技术[4],例如,在文献[1]中,作者提出了 alpha 衰减技术,即通过将 alpha掩模位图映射到投影面来实现光带的消除。而文献[5]则提出了一种基于相机反馈的亮度衰减映射(LAM)算法。

文献[6]以二次曲线作为衰减曲线来计算校正位图,并通过纹理映射方法来实现亮度均衡,但其算法局限于相邻投影面必须具有相同的亮度。由于这一算法容易实现,效果也比较好,因此本文借鉴其思想以及基于对 t_{FDT} 的考虑进行了以下改进:

(1) 以指数为 α 的幂函数替代二次曲线函数作为衰减曲线,然后就可以通过调节 α 的值来改变衰减程度,以适应不同宽度重叠区的拼接需要;

(2) 如果拼接画面的亮度不相等,则希望亮度大的衰减程度大些,亮度小的衰减程度小些,因此可以引入贡献因子 $\beta = I_1/I$ 进行调节,I_1 为投影面 1 的亮度,I 为相邻投影面亮度之和;

(3) 由于只处理重叠区域的像素点,因此可以提高执行速度。

通过改进可以得到以下衰减方程(其中 d 是像素点到重叠区投影面 1 边界的距离):

$$f_1(d)=\begin{cases}\dfrac{2}{\beta}d^{\alpha}\\ \dfrac{1}{\beta}[1-2(1-d)^{\alpha}]\end{cases}$$

$$f_2(d)=\begin{cases}\dfrac{1}{1-\beta}[1-2d^{\alpha}]\\ \dfrac{2}{1-\beta}(1-d)^{\alpha}\end{cases} \tag{2}$$

改进后的算法可以适用于不同宽度重合区的拼接以及相邻投影面亮度不等的情况。本文通过在视频链路中新增一个该算法的 filter 来实现边缘亮度均衡,并且将该 filter 放置在非线性几何校正 filter 之前。

3.2 几何校正及边缘平滑

在环幕系统中,由于投影仪被架在高处,从上往下投影到柱形弧面上,因此投影画面会产生畸变也被称为非线性几何失真[7](图 3)。

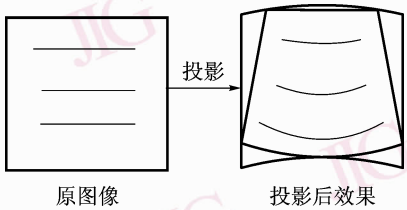

图 3 非线性几何失真

Fig. 3 No-linear geometrical distortion

对于非线性几何失真,投影仪自带的光学校正功能只能对梯形失真进行有限的校正,而由弧面引起的失真则需要通过对视频源进行几何变换来进行

校正。本文通过在视频 filter 链路中添加一个专门的 TFilter 来进行非线形几何校正处理,校正过程如图 4 所示。

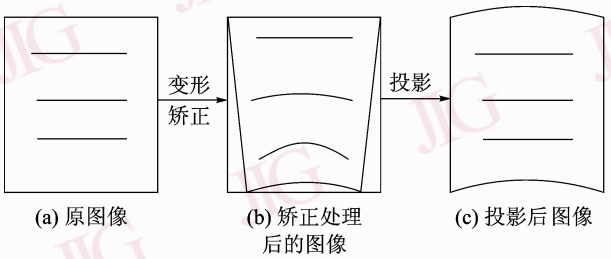

图 4　几何校正示意图

Fig. 4　Geometrical transform

为了提高系统对场景变化的适应能力,有必要在 filter 中引入人机交互接口,以使得用户可以在系统调试阶段通过参数的设置来实时校正播放效果。从图 4 中还可以看到,由于经校正后的图像在边缘处会出现锯齿,其对图像的拼接会直接产生影响,因此在对图像进行几何校正的同时,也要能够保持边缘的平滑[8]。

变形后图像之所以会在边界产生锯齿是因为边界上点的颜色与边界外的填充色反差太大,而解决锯齿问题的主要思路就是要通过降低边界内与边界外颜色的反差来淡化锯齿。由于通过对边界点进行分类并选取不同的模板来拟合边界周围的点,可以促使边界内外的颜色平滑过渡,从而可消除锯齿。保持边缘平滑图像几何校正算法步骤如下:

(1) 读取用户参数,在原画面中确定变换窗口和限定校正后图像的位置及大小,并且对任意点 p,计算以下映射关系函数:

$$f:(x_p,y_p)\rightarrow(\hat{x}_p,\hat{y}_p) \tag{3}$$

(2) 先从上游 filter 读入一帧画面,然后使用步骤(1)中的映射函数对图像进行双线性插值[9],并且按顺时针顺序记录下新的边界像素点的位置,同时对新边界外的像素点填充黑色;

(3) 用下式

$$K=|x_1-x|+|x-x_2|+3\times(|y_1-y|+|y-y_2|) \tag{4}$$

计算边界点(x, y)的近似斜率,并进行分类;

(4) 对不同类的边界点,采用不同的带权模板混合其周围像素点的颜色来填充该边界点[8];

(5) 将处理完的该帧画面传送给下游 filter;

(6) 重复步骤 1 到步骤 5 直到处理完视频的最后一帧画面;

通过以上步骤就可以对视频实时进行保持边缘平滑的非线性几何校正;然后就可以通过选用网格画面[3]作为参考来对投影效果进行手工校正,同时通过调节参数对投影效果进行调整,直到相邻画面的网格对齐;最后锁定参数。

4　实验结果分析

实验时,PC 机配置为 P4 2.80GHz CPU,1 G 内存;图形加速卡为 NVIDIA Quadro FX 540,128 MB 显存;视频源采用每帧画面大小为 1 024 × 768 的 AVI 文件。

图 5 是实验中采用的 F-P 链路图,它清楚地表现了视频流的处理流程,从图中可以看到,链路中包含了 5 个 TFilter,其中 3 个是系统本身具有的对视频进行音视频流分离、MPEG4 解码以及颜色空间变换的 TFilter,而另外两个 filter 则是在本文中讨论的边缘亮度均衡过滤器和非线性几何校正以及边缘锯齿的消除过滤器。图 6 是对视频文件进行拼接前后的效果对比图,图 6(a)是未进行融合处理拼接后的效果,可以很清楚地看到相邻画面之间有着明显的拼缝。而图 6(b)则是在进行边缘亮度均衡之后拼接的效果,由该图可以看到图像之间平滑过渡连成一体,已经看不出明显的拼缝。

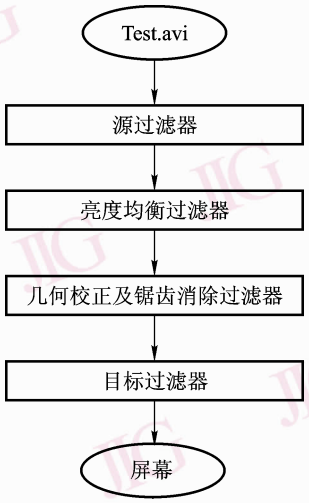

图 5　视频 F-P 图

Fig. 5　Video F-P image

对用于衡量 filter 优劣的两个时间指标 t_{FDT} 与 t_{CDT} 分别进行了实验,实验的视频源采用大小为 1

024 × 768，帧速率为 25fps，数据速率为 537kbps、24bits 采样大小的 AVI 文件，测试算法为对每个像素点进行 R，G，B 三色互换。最后得到的测试实验结果如表 1、表 2 所示：

表 1 t_{FDT} 测试实验数据表

Tab. 1 t_{FDT} testing result table

测试方式	$O(K\times(1\,024\times768))$ K 个(1 024 × 768)的二重循环			$O(1\,024\times768\times K)$ 三重循环	
	$K\leqslant5$	$6<K\leqslant12$	$K>12$	$K<6$	$K>6$
实验效果	流畅	基本流畅偶有跳帧	跳帧次数随 K 值增加	流畅	跳帧次数随 K 值增加

表 2 t_{CDT} 测试实验数据表

Tab. 2 t_{CDT} testing result table

复杂度	filter 个数					
	1 个	2 个	3 个	4 个	5 个	6 个
$O(5(1\,024\times768))$	流畅	流畅	比较流畅	有明显卡帧	很卡	很卡
$O(4(1\,024\times768))$	流畅	流畅	流畅	比较流畅	有明显卡帧	很卡
$O(3(1\,024\times768))$	流畅	流畅	流畅	比较流畅	有轻微卡帧	比较卡

为了拼接效果更加直观，在 t_{FDT} 和 t_{CDT} 的实验中，仅考虑了测试使用的“三色互换 filter”，而忽略了其他 filter 所耗费的时间，在这一前提下得出了表 1 和表 2 中的结果。从表中可以看到，在用 filter 处理图像的时间复杂度小于 $O(4\times1\,024\times768)$ 且 filter个数小于等于 3 的情况下视频可以很流畅的播放。对于本文讨论的两个 filters，其中非线性几何校正 filter 处理图像的时间复杂度小于 $O(m\times n/2+2\times(m+n))$，亮度均衡 filter 处理图像的时间复杂度小于 $O(m\times n/2)$，可是这两个 filter 的图像处理时间 t_{FDT} 以及 t_{CDT} 都在系统可以承受的范围内，并且通过实验也证明，其具有比较好的拼接处理效果。对于非线性几何校正 filter 来说，由于其时间复杂度相对比较高，因此在配置稍差的 PC 机中可以将其中的几何校正与锯齿消除功能进行分离，以提高播放的流畅性。

5 结 论

本文主要针对多投影系统中最主要的边缘亮度均衡及几何校正两个问题进行讨论，并且给出了简单可行的 filter 实现算法，用户可以通过修改 filter 的参数来对投影面进行整体调整或局部细调，直到取得最佳效果；filter 提供的这些接口也为以后通过增加高像素摄像头或照相机反馈来自适应调节模块的开发提供了支持。此外，采用 F-P 构架的系统的最大特点就是具有良好的灵活性，其不仅对播放环境的适应力强，并且有利于进行扩充，其实现效果已经接近，甚至超过了一些专业硬件设备。随着 PC 机速度的进一步提高以及多核 PC 机的普及，更可以通过改进 filter 中的算法来进一步提高拼接效果，这也决定了其必将成为今后研究的热点。

(a) 未进行融合处理的图像

(b) 融合后图像

图 6 融合效果示意图

Fig. 6 Result images

但在本文构建的系统中也存在着一些不足之处，有待进一步研究与改进，比如：①在锯齿消除算法中没有对图像内部的一些高频部分变形后产生的锯齿进行考虑，其实可以通过边缘检测算子先把这些边界提取出来，再进行平滑处理；②在亮度均衡方面还应该对投影面内的亮度均衡以及投影面之间可能产生的颜色不一致问题进行考虑。这些问题需要在今后的研究中继续完善。

参考文献(References)

1 Raskar Ramesh, Brown Michael S, Yang Rui-gang, *et al*. Multi-projector displays using camera-based registration [A]. In: Proceedings of IEEE Conference on Visualization [C], San Francisco, CA, USA, 1999: 161-168.

2 Majumder Aditi, He Zhu, Towles Herman, *et al*. Achieving color uniformity across multi-projector displays [A]. In: Proceedings of IEEE Conference on Visualization [C], Salt Lake City, UT, USA, 2000: 117-124.

3 Yang Rui-gang, Gotz David, Hensley Justin, *et al*. A reconfigurable multi-projector display system [A]. In: Proceedings of IEEE Conference on Visualization [C], San Diego, CA, USA, 2001: 167-174.

4 Stone Maureen C. Color and brightness appearance issues in tiled displays [J]. Computer Graphics and Applications, 2001, **21**(5): 58-66.

5 Majumder Aditi, Stevens Rick. Color nonuniformity in projection-Based displays: analysis and solutions [J]. Visualization and Computer Graphics, 2003, **10**(2): 177-188.

6 Jia Qing-xuan, Ruan Rui-qing, Sun Han-xu, *et al*. Realizing of intensity balancing for multi-projector display system [J]. Journal of System Simulation, 2006, **18**(supp. 2): 478-481. [贾庆轩, 阮瑞卿, 孙汉旭等. 多投影面显示系统亮度均衡的实现 [J]. 系统仿真学报, 2006, **18**(增刊 2): 478-481.]

7 Jia Qing-xuan, Song jing-zhou, Sun Han-xu, *et al*. Design and achieving aigh immersive virtual system [J]. Engineering Science, 2006, **8**(8): 33-38. [贾庆轩, 宋荆洲, 孙汉旭等. 高临场感多投影面虚拟环境系统的设计与实现 [J]. 中国工程科学, 2006, **8**(8): 33-38.]

8 Jiang Hua. Edge smoothing of abnormally window [J]. Coumputer Programming and Maintence, 2004, (5): 68-74. [江华. 不规则窗体边缘的平滑方法 [J]. 电脑编程技巧与维护, 2004, (5): 68-74.]

9 Yang Guo-peng, Yu Xu-chu, Li Qiang. Research status in image interpolation [J]. Application Research of Computers. 2006, **23**(supp.): 777-778. [杨国鹏, 余旭初, 李强. 图像插值方法研究现状 [J]. 计算机应用研究, 2006, **23**(增刊): 777-778.]