文章编号:1001 - 9081(2005)03 - 0729 - 03

基于实时渲染技术具有水彩画风格的音乐可视化方法

李　华[1],胡春晖[2],顾　明[1]

(1. 清华大学 软件学院,北京 100084; 2. 微软亚洲工程院,北京 100080)

(li-h02@mails.tsinghua.edu.cn)

摘　要:针对目前音乐可视化效果单一的问题,实现了一个与音乐充分结合,具有水彩画风格的动画效果。主要应用基于 alpha 通道的水彩画效果实现技术,基于优化物理模型、逐段拼接以及组合反走样等方法的动态场景仿真技术,借助 Direct3D 实现动画场景。结果显示,该方法可以在较低的 CPU 占用率和较小内存请求下达到良好效果。

关键词:alpha 通道;实时渲染;音乐可视化;组合反走样

中图分类号: TP391.9　**文献标识码**:A

Music visualization with aquarelle style based on real-time rendering

LI Hua[1], HU Chun-hui[2], GU Ming[1]

(1. *School of Software, Tsinghua University, Beijing* 100084, *China*;
2. *Microsoft Research Asia Advanced Technology Center, Beijing* 100080, *China*)

Abstract: Most of current music visualizations are monotone. In this article, a aquarelle-style music visualization integrated perfectly with music was implemented. It adopts some key technologies, such as aquarelle effect based on alpha channel, tidy physical model, piecemeal joint, and composite anti-aliasing, etc. The example of music visualization by bloom, stalk, dimple and goldfish shows that this method can get good effect with low CPU occupancy and small memory request.

Key words: alpha channel; real-time rendering; music visualization; composite anti-aliasing

0　引言

音乐可视化作为一种将听觉形象化的技术,在当前各种流行的媒体播放软件中一直占有着重要的地位,其中所涉及到的实时渲染、自然效果仿真、计算复杂性等是计算机图形学及多媒体方面研究的热点[1,2]。纵览目前比较流行的媒体播放软件,其音乐可视化效果主要有波形图、色彩变换、烟雾特效等表现形式,画面较为单一,还没有具有完整主题的可视化效果出现。本文实现了一个主题鲜明、与音乐充分结合、具有水彩画风格的可视化效果,表现为在音乐的控制下,场景中的茎逐渐生长,花自然开放,再加上游动的金鱼和花瓣落下后产生的涟漪,构成了一幅赏心悦目的图画。

本文利用 Direct3D 技术,通过控制多级纹理的 alpha 值的变化和层次的多少,使得整个场景呈现出水彩画的效果;通过优化物理模型,逐段拼接以及组合反走样等方法,使得场景中出现的花朵、花茎、涟漪、金鱼等动态变化的元素在模型精细和光滑方面有优良表现。结果表明,这些方法的综合应用,可以在较低 CPU 占用率、较小内存使用下完成与音乐完美结合的、具有水彩画效果的可视化效果。同时,利用 COM 插件技术与 media player 结合起来,构成一个有机的整体。

1　基于 alpha 通道的水彩画效果实现

场景整体是水彩画风格的,水彩画效果的实现是整个可视化场景的基础。

32 位 RGB 模式下的 BMP 位图,采用 4 个字节存储位图每个像素的信息,红蓝绿三个颜色分量各占用一个字节的存储空间,剩余的一个字节空间并不存储和颜色相关的信息,即与颜色通道无关,它记录着表征像素颜色透明程度的值,即 *alpha* 值。在位图的显示过程中,会根据该值对源像素和目标像素的颜色执行加权平均:输出像素颜色 = 目标像素颜色 × (1 − *alpha*) + 源像素颜色 × *alpha*,达到二者颜色融合的目的。水彩画效果的实现正是以这一基本运算为基础,在多级纹理合成时,每一层纹理都与已合成结果再进行加权平均,即第一层首先与背景加权平均,其运算结果作为背景再与下一层纹理加权平均:

B_0 = 初始背景颜色

$B_n = B_{n-1} \times (1 - alpha) + L_n \times alpha$

B_i 为第 i 次合成的纹理;L_i 为第 i 层纹理。

利用此迭代过程,同时适当旋转各个花瓣的相对位置,造成花瓣在局部的纹理层数不同,就产生了浓淡搭配的水彩画效果。此方法较运用粒子系统模拟或者画笔模拟等实现方式[3],具有计算复杂度低、算法直接、实现容易等特点,更加符合实时性的要求。

2　动画场景的生成

2.1　花的模拟

对于花的模拟,采用的方法是,首先生成一个二维矩形模型,借鉴三维地形图的生成算法[4],纵横划分为网格面,然后把某种花瓣的纹理坐标采用线性插值的方法逐点映射到网格的各个节点上(图 1)。而花瓣在先期制作好后,由于要生成水彩

收稿日期:2004 - 08 - 29;**修订日期**:2004 - 11 - 17

作者简介:李华(1979 -),男,山西长治人,硕士研究生,主要研究方向:计算机图形学; 胡春晖(1975 -),男,湖北黄石人,高级工程师,硕士,主要研究方向:计算机图形学、网络; 顾明(1962 -),女,辽宁沈阳人,副教授,主要研究方向:操作系统、中间件技术、分布式应用系统支撑平台.

画效果,根据前述生成算法,需要为每个像素增加 *alpha* 值。这里利用了 32 位 BMP 图每个像素空闲一个字节的特点,需要完全透明的部分则把相应的 *alpha* 值设为 0,否则以反比于离位图中心点的距离、满足线性关系的方式设置该值。按照这种方法处理在性能上更优于直接采用透明 PNG 图构造花瓣纹理的方法,这样,在后续的渲染处理中就可以产生水彩画的效果。

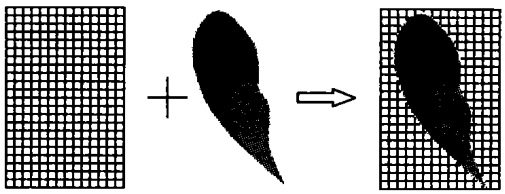


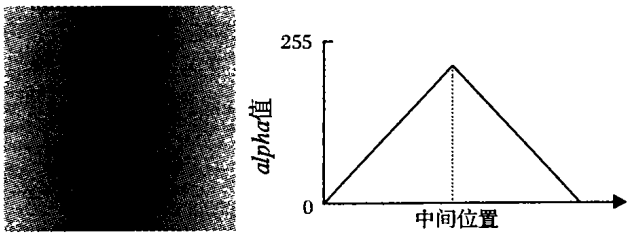
图 1 纹理映射

在上面两步的基础上,对花瓣矩形网格模型的各个顶点按照逆时针方向、逐个三角形面建立索引,这是为了在随后使用多个三角形面拼接模型的时候,通过顶点索引来依次绘制各个三角形面,达到顶点数据复用的目的,减少模型对内存的占用。

2.2 茎的模拟

在三维场景中直接绘制自由曲线十分繁琐,所以最直观的花茎的模拟方式是采用逐段圆柱体拼接的方法,但其缺点显而易见:花茎是光滑的且从根部开始逐渐变细,难以将拼接点处理得很光滑;而且实时性和 CPU 及内存低占用的要求也限制了大计算量的工作。鉴于此,本文采用模型和纹理相结合的方法来减少计算量,保证实时性;花茎的主体采用贝塞尔曲线生成,通过在一定参数范围内随机产生两个控制点的坐标,使得生成的花茎在总体向上的条件下,又具有较好的曲线美。但是,当把纹理映射到花茎上后,产生了较明显的锯齿效果,同时有些地方有断裂出现。因此,采用如下三个步骤解决走样问题。

1) 把花茎的纹理按照从中间到两边线性递减的方法设置 *alpha* 值[5],这样,在最终的贴图效果中,通过纹理亮度的逐渐变弱在视觉上大大降低了走样程度(图 2);



(a) 花茎纹理 (b) 花茎纹理 *alpha* 变化

图 2 花茎纹理及其 *alpha* 变化

2) 采用花茎越长点越密的方案,增加构成花茎的顶点的个数,避免了花茎断裂的情况;

3) 在花茎纹理坐标映射过程中,没有采用直接映射的方法,而是首先计算花茎在当前点的切线的斜率,得到切线仰角,根据此角度,对该顶点的纹理坐标进行旋转。这样做的目的是使得最终的纹理可以按照花茎生长的方向映射,从图 3 可以看出,如此减小走样程度效果十分明显。

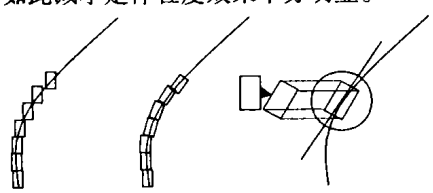


图 3 纹理映射

通过上面三步的组合反走样方法,很好地实现了花茎的模拟,取得了满意的效果。

2.3 涟漪的模拟

涟漪作为整个动画场景的一部分,模拟的结果要能真实表现涟漪的生成、扩散、消失的过程。涟漪的三维模型用一个个椭圆表示,以椭圆两个焦点的中心为椭圆的中心,逆时针方向在椭圆周上等角度产生若干个点,点的密度决定了计算复杂度及最终涟漪的精细度,然后顺次选择椭圆周上的两个点与中心构成三角形面,与各自的纹理坐标建立映射关系。而涟漪的扩散通过逐渐增大椭圆的长短轴长度同时减小各个顶点纹理的 *alpha* 值,就可以使得涟漪在扩散的过程中逐渐融合到环境中去,实现涟漪逐渐晕开的效果。

(a) 建立索引方法 (b) 长短轴 *alpha* 变化趋势 (c) 涟猗纹理

图 4 涟漪模型及长短轴与 *alpha* 值变化图示

2.4 金鱼的模拟

为了使动画场景内容更加丰富,增加了受音乐控制的金鱼。它们也同样运用前述方法产生水彩画效果。金鱼以随机产生的贝塞尔曲线作为游动的路线,鱼体则采用骨骼动画的思想逐段拼接而成,然后计算各自的纹理坐标加以映射。这样做主要是考虑到模型简单,而且当控制鱼运动时,由于鱼体的这种特点,可以使鱼曲线游动时更加光滑自然,不会有僵硬的感觉。

3 音乐控制方法

3.1 控制参数的选择

在 media player 播放音频文件的同时,可以实时得到当前时刻左右两声道、各 1024 个采样点的波形及频率数据,这些原始数据描述了当前音乐的特征。音乐可视化的目的是用视觉效果表现听觉感受,有实时性、表现性的要求。因此参数的选择既要能体现音乐在当前时刻的特征,又要兼顾实时性的要求,降低计算复杂度。故折中考虑,取当前时刻得到的数据的平均值作为控制参数是一个理想的选择。

统计分析和实验结果表明,对于同一首歌,频率平均值变化幅度要大于波形平均值的变化幅度。而可视化场景要能充分体现出每首歌节奏的差别,在视觉上有直观的表现。如果选用变化平缓的参数,需要对这些参数进行放大来增大这种差别的影响,以体现节拍之间的差别;当选用起伏较大的参数时,差别已经比较明显,再经过适当调整就可以满足控制的要求,所以频率平均值是个合适的选择。

3.2 场景元素的控制

场景中主要的元素有花茎、花朵、金鱼等,音乐的节奏变化就体现在对这三种元素的控制上。根据得到的频率平均值,控制各元素的动作行为,并针对具体元素的不同加以适当变化,使得被调整元素体现出自身的特点,同时达到与音乐节奏的完美配合。

1) 花茎的控制:利用频率平均值控制花茎产生的数目和花茎的生长速度。频率值与花茎数目、生长速度符合线性关系。

2) 花的控制:利用频率平均值控制花开的速度,速度值整体符合开口向上的二次函数的下降支,由于函数的这支具有凹函数的性质,变化率的绝对值逐渐减小,使得花开得效果表现为开放速度逐渐变慢,更符合真实情况,每次用频率平均值修正速度增长的步长即可;当花凋谢时,落下的花瓣使得场景下部产生涟漪,涟漪自然消散。

3) 金鱼的控制:利用频率平均值控制鱼的游速,该值变化产生的影响沿着鱼体逐段传递,每段鱼体都计算自己与它直接连接的前段之间的偏向角和移动后的距离,通过方向角调整自己的朝向,通过距离调整自己的速度,这样,整条鱼就随着音乐游动起来。

通过上述控制方法,在音乐播放时,节奏明快的音乐将配合生成动感明显的效果,节奏舒缓的音乐则生成相对轻盈的效果。

4　性能优化

由于动画场景是对音乐播放的补充,因此,不能由于动画生成占用了大量的 CPU 处理时间而使得音乐的播放断断续续;或者,由于 CPU 同时处理音乐播放使得播放的动画有明显的跳跃感觉。这两种情况都是无法接受的。实现中采用的性能优化方法如下:

1) 锁定帧频。严格控制每一帧的生成时间,同时在每一帧生成之后延长固定的时间。由于当前计算机处理能力大大增强,导致帧生成时间与延时相比仅占很小的比率,这样就可以达到锁定帧频的目的,而固定的帧频对于控制整个动画场景的行为有很大的帮助。

2) 内存利用率优化。通过简化物理模型,为模型的顶点建立索引,大大降低了内存的占用率。

3) 其他优化方式。例如全局变量和内联函数的使用,以及纹理的大小采用 2 的方幂,使得很多乘/除运算可以通过左/右移位完成等,都有助于改善程序执行的性能,减少系统调用及运算复杂性。

实验结果表明,对于性能同 Dell Dimension™ 4600n 的计算机,通常情况下,CPU 使用率维持在 8% 以下(包含 media player 其他方面的使用),内存占用率在 15 兆以下(包含 media player 自身对内存的使用)。

5　结语

通过综合运用各种算法和技术,最终在保证 CPU 低占用、内存低消耗、实时性要求下取得了令人满意的音乐可视化效果,对生成更复杂的音乐特效,以及虚拟现实、场景动画等方面都有着一定的借鉴意义。

图 5　最终音乐可视化效果

参考文献:

[1] 陈彦云,孙汉秋,郭百宁,等. 自然雪景的构造和绘制[J]. 计算机学报, 2002, 25 (9): 475 - 480.

[2] MAJUMDER A, GOPI M. Hardware Accelerated Real Time Charcoal Rendering [A]. Proceedings of the second international symposium on non-photorealistic animation and rendering[C], 2002. 59 - 66.

[3] 石永鑫,孙济洲,张海江,等. 基于粒子系统的中国水墨画仿真算法[J]. 计算机辅助设计与图形学学报, 2003, 15(6): 72 - 78.

[4] 彭群生,鲍虎军,金小刚. 计算机真实感图形的算法基础[M]. 北京: 科学出版社, 1999. 165 - 220.

[5] 孙家广,杨长贵. 计算机图形学[M]. 北京: 清华大学出版社, 1996. 210 - 214.

(上接第 728 页)

在自然中,河水主要分水上和水下;在本系统中,由于漫游视线的限制,河水主要是水上的模拟,不涉及到水下。基本思想就是用多边形网格来表示水面,用正旋曲线计算出网格顶点位置,最后用纹理贴图进行渲染[4]。但在大规模的随机河流场景中,由于网格数目非常大,如果采用上述办法,将耗费大量时间计算各个网格的正旋波动,严重影响到场景中主要物体的绘制。由于河流不是漫游系统的主要表现物,一般位于视线远角,因此只要在一定距离上,能够产生流动的效果即可。

2.4　运动方程

运动方程,就是表现场景中物体如何运动的方程。一个完善的运动方程可以使漫游系统中的物体运动接近于现实中物体的运动。但是一个完善的运动方程必然耗费系统大量的运算时间;在本系统中,为了提高实时性,考虑真实感要求,对运动方程作了一定的简化。

汽车的运动方程涉及到两个关键的运动计算:第一,运行时运动状态计算;第二,与场景中物体的碰撞后的状态响应。

汽车运动时,需要响应用户换档、控制油门、刹车、倒车、点火消息,根据当前位置和姿态计算下一个位置和姿态。基本原理就是矢量方向上运动分解,主要利用汽车的模拟回调函数 GV_obi_set_sim_callback 来控制私有数据实现。在此基础上,将汽车运动方程扩展更为复杂的运动:比如点火时的抖动效果、加减速以及发生碰撞时的惯性效果等。

雨滴的运动轨迹在考虑重力、风向的影响下可以类似实现。

3　辅助技术

3.1　建模要求

在漫游系统中,汽车的方向盘、主动轮、被动轮都需要单独控制,其中一部分运动将带动其他部分一起运动。为了实现这些对象之间各种精确的相对运动,建模时就必须采用分层数据结构。同时,汽车与地形发生碰撞时,需要确认发生碰撞的是地形中的哪个物体,如果不采用分层结构也无法实现。因此,建模时,我们利用 MultiGen Creator 的 DOF 技术,来配合 OpenGVS 的对象工具实现。

3.2　DirectX

目前的 OpenGVS 版本已经不再支持 MultiGen Creator 的声音节点,因此,为了得到漫游中比较逼真的音效效果,比如点火、运行、碰撞时的音效,必须结合 DirectX[5] 的 3DSound 技术实现。由于 OpenGVS 对于底层硬件支持比较滞后,需要我们利用新版本的 DirectX 来实现对新硬件的支持。

参考文献:

[1] OpenGVS Programming Guide V4.3 [Z]. Quantum3D, Inc, 1999.

[2] HEARN D, BAKER MP. 计算机图形学(第 2 版)[M]. 蔡士杰,吴春镕,孙正兴,等译. 北京: 电子工业出版社, 2002.

[3] BLYTHE D. Natural Phenomena[A]. 1997 SIGGRAPH Conference [C], 1997.

[4] GRANTHAM B. Simulating Natural Phenomena [A]. 1999 SIGGRAPH Conference [C], 1999.

[5] DirectX 9.0 Programmer's Reference [Z]. Microsoft Corporation, 2003.