

## 虚拟现实一种基于运动混合的实时同步算法

王忆源<sup>1</sup>, 陈福民<sup>2</sup>

(同济大学 计算机科学与技术系, 上海 200092)

(tjyuan@hotmai.com)

**摘要:**在虚拟现实中,采用运动捕获系统建立基本运动库,然后通过运动编辑技术对基本运动进行处理。运动混合技术是编辑技术中最为实用也最为复杂的一种。提出了基于运动混合的实时同步算法,以便更好地动态混合运动,避免产生未预期的效果,以创建复杂的虚拟动画。

**关键词:**虚拟现实;运动混合;群体动画;时间偏差;优先级

**中图分类号:** TP391.9 **文献标识码:** A

## Real-time synchronization algorithm based on blending of motions in VR

WANG Yi-yuan<sup>1</sup>, CHEN Fu-min<sup>2</sup>

(Department of Computer Science and Technology, Tongji University, Shanghai 200092, China)

**Abstract:** In virtual reality, a fundamental motion library is built by using motion capture system. Then appropriate fundamental motions are selected from the library, and desired motions are synthesized from these fundamental ones. One of these motion editing techniques, motion blending, is the most practical and complex one. A real-time synchronization algorithm was presented based on blending module. The basic idea is to make the data set more flexible and to dynamically blend motions to avoid unexpected ones. As a result, complex virtual animations are created from above.

**Key words:** VR(Virtual Reality); motion blending; group animation; time warping; priority

### 1 运动混合技术中的同步问题

当今的虚拟现实系统中,运动捕获技术已经作为一种强大的制作手段被广泛运用于视频游戏和电影制作等相关行业中。当然,这项技术遵循某种动态规则,而且通过运动编辑技术对捕获的运动进行处理,使得虚拟动画更为逼真。在众多编辑技术中,运动混合技术使得我们可以从捕获的简单运动中创造出复杂的动画效果。但目前的混合算法还不成熟,往往带来超出预料的运动错误描述。比如,当双脚都接触地面,将左脚的运动与右脚的运动混合,作为混合结果的运动将无法正确反映原始双脚的状态,而是在地面上整体滑动。因此,为了避免类似错误的出现,运动的动态同步混合就显得相当重要。

在实时运动的动态环境中,动画所采样的基本运动不仅是交互自适应的,而且受到外部条件的制约,随着外部环境的变化而变化。单个角色已经如此,群体动画中不同形态的角色动画就更为复杂。基于运动混合的实时同步算法提出,就是为了改善多角色多运动之间的协调,尽可能的减少运算的复杂度。此算法采用了基于运动相位的时间偏差技术<sup>[1]</sup>,同时通过为基本运动设置优先级,实现在任何时刻可以使用开/关命令激活/解除运动的有效性。在运动混合技术和群体动画领域显得相当重要。

### 2 一种基于运动混合的实时同步算法

#### 2.1 相关工作

之前对同步运动的研究工作主要通过两种途径:识别运

收稿日期:2005-02-05;修订日期:2005-04-26

作者简介:王忆源(1980-),男,江苏人,硕士研究生,主要研究方向:网络多媒体; 陈福民(1945-),男,上海人,博士生导师,主要研究方向:网络多媒体。

#### 3.7 备份与恢复备份

**备份:**主机和备机都要进行备份工作,将所有接收到的网络数据都写入备份日志;当接收到正确心跳数据时,就将该时间点写入备份日志,该时间点是用于备份恢复的。

**恢复备份:**按照日志内容,从最后一次收到正确心跳数据的时间点开始恢复备份。

在大多数双机容错系统中都应用到了备份与恢复备份。由于篇幅有限,在这里就不再详述备份和恢复备份的内容。

### 4 结语

使用相同IP的双机容错系统,减小了IP切换的时延,实时性更强。在双机双工工作模式下,通过修改Linux内核的网络,使备机不发送数据,且对上层透明,备机丢失网络数据的可能性也得到了降低。该系统相对于通过主机转发网络数

据给备机的双机容错系统,具有更高的可靠性。本系统可广泛应用于数据量不太大的网络服务环境,为服务器提供持续、稳定的服务。

#### 参考文献:

- [1] Droms R. Dynamic Host Configuration Protocol, RFC 2131 [S]. 1997.
- [2] 徐立云, 邵惠鹤. 双机容错系统的一种实现途径[J]. 计算机工程, 2000, 26(9).
- [3] Comer DE. 用TCP/IP进行网际互联 第一卷: 原理、协议与结构(第四版)[M]. 林瑶, 蒋慧, 杜蔚轩, 等译. 北京: 电子工业出版社, 2001.
- [4] 骆耀祖. Linux操作系统分析教程[M]. 北京: 清华大学出版社, 2004.
- [5] 王存祥, 樊婷婷. 校园网IP地址冲突的分析及解决方案[J]. 中国电化教育, 2004, (7).

假设运动相位序列的前  $k + nk - 1$  项是同步的,则有:

$$\forall 1 \leq j \leq k + nk - 1, \bigoplus_{i \in [1, nA]} S_i(j) \neq Err$$

如果运动序列第  $k + nk$  项不同步,即:

$$\bigoplus_{i \in [1, nA]} S_i(k + nk) = Err$$

同步算法扩展序列元素的目的就是为了满足:

$$\bigoplus_{i \in [1, nA]} S_i(k + nk) \neq Err$$

这种解决办法可以扩展至所有  $nA$  个最后同步元素:

$$\bigoplus_{i \in [1, nA]} S_i(k + nk) = \bigoplus_{i \in [1, nA]} S_i(k + nk - 1) \neq Err$$

令子集  $A$  表示第  $k + nk$  步,  $nA$  个动作中可以与前  $k + nk - 1$

1 项同步的最大子集,即:

$$\left( \bigoplus_{i \in [1, nA]} S_i(k + nk - 1) \right) \oplus \left( \bigoplus_{i \in A} S_i(k + nk) \right) \neq Err$$

这个子集不可能为空,因为不同步问题必定是  $LS$  和  $RS$  的矛盾所致。矛盾出现的同时,必定有其他元素与  $LS(RS)$  相协调。

针对非空子集的出现,有以下两种情况:

$\bigoplus_{i \in [1, nA]} S_i(k + nk - 1) = \Phi$ : 根据  $\Phi$  元素的特点,只需延长某一跳跃 ( $\Phi$ ) 动作即可解决同步问题;

$\bigoplus_{i \in [1, nA]} S_i(k + nk - 1) \neq \Phi$ :  $\Phi$  同时也不存在于最大子集  $A$  中,这是因为运动序列第  $k + nk$  项的不同步(代数关系  $\oplus$  结果为  $Err$ )。因此,有:

$$\left( \bigoplus_{i \in [1, nA]} S_i(k + nk - 1) \right) \oplus \left( \bigoplus_{i \in A} S_i(k + nk) \right) \neq \Phi$$

由于子集  $A$  表示可以与前  $k + nk - 1$  项同步的最大子集,所以  $\{i \notin A \wedge i \in [1, nA]\}$  包含为了同步需要扩展的动作索引。能够找出问题所在并解决之,此方案确实可行。

既然方案并不唯一,引入优先级的概念就可以达到优化的效果。为每个动作设置优先级,从高到低,当出现不同步问题,保持高优先级动作不变形,扩展低优先级动作,可以提高运动混合的逼真度和动作的连续性。

#### 2.4.2 序列元素持续时间调整

运动相位序列元素重组同步之后,得到运动混合序列  $\{\forall k, \bigoplus_{i \in [1, nA]} S_i(k)\}$ , 其中元素(动作)的持续时间(有效时间)和优先级有关。根据动作优先级  $P_i$ , 为每个动作  $A_i$  设置规范化加权值  $w_i$ 。  $\Delta S_i(k)$  代表混合之前单独动作的持续时间,  $\Delta S(k)$  则是混合以后的值。计算公式如下:

$$\forall k, \Delta S(k) = \sum_{i \in [1, nA]} w_i \times \Delta S_i(k)$$

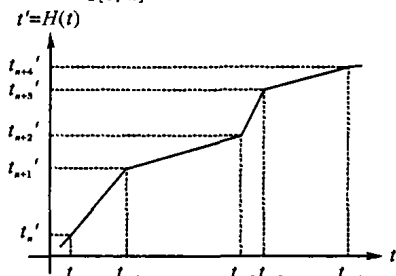


图4 初始时间偏差

#### 2.4.3 时间偏差

调整运动混合序列元素的持续时间,势必造成动画时间与真实时间之间的时间偏差,即动作有效时间的缩小或扩大。这种偏差可以用函数  $t' = H(t)$  表示,它会造成速度与加速度的不连续性(图4)。为了解决这个问题,引入另一个函数来抵消偏差的负效应,使之连续。

函数的构造过程如下: $t$  和  $v(t)$  分别表示当前时刻和偏差向量。运动混合序列元素时间调整后,  $t_{n+1}$  也相应发生变

化,如图5所示。

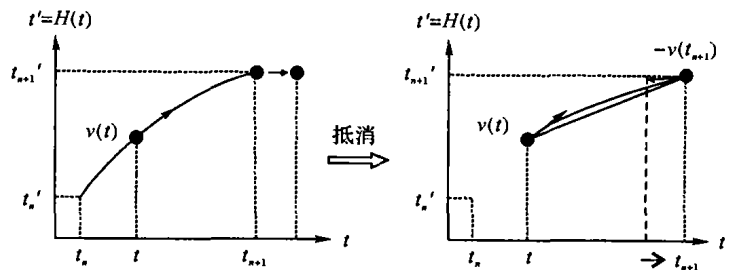


图5 单调抵消函数构造示意图

利用对称性规则,取偏差向量  $v(t)$  在  $t_{n+1}$  时刻的对称向量  $-v(t_{n+1})$ , 尽可能抵消偏差的负效应。综合考虑时刻两端(比如  $t_n$  与  $t_{n+1}$  时刻),可以得到每个时域的单调抵消函数(图6)。

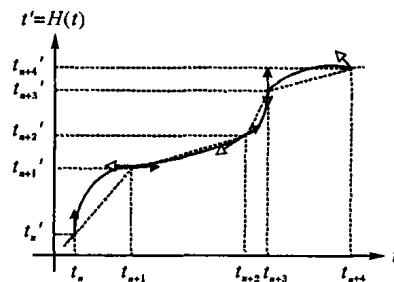


图6 最终时间偏差

### 3 结语

本文提出了一种基于运动相位时间偏差等技术的运动混合实时同步算法。引入代数关系  $\oplus$  检测若干动作能否混合,依据 MKM 动画引擎的实时性与交互性,在任意时刻可以激活/解除运动的有效性,实现运动相位序列的重组同步;根据动作优先级,为每个动作设置加权参数,调整运动混合序列中元素的有效时间,并消除由此造成的时间偏差负效应,保证动作的一致连续性。这种算法不仅适用于单角色动作,更能协调群体动画中不同形态的角色动作。运动学方面的应用不仅限于本文研究的角色行走、跳跃,对于角色的其他动作(脚部的跳舞、奔跑,手部的射击、摇摆等)同样适用。该算法的运算复杂度可以忽略不计,但同时受到一定的限制,比如运动混合偶然不符合动力学理论等。因此,算法还可以完善,实现方式也可以更多样,为相关方面进一步的研究奠定了基础。

#### 参考文献:

- [1] MENARDAIS S, MULTON F, KULPA R, et al. Motion blending for real-time animation while accounting for the environment [A]. In Computer Graphics International [C]. 2004.
- [2] UNUMA M, ANJYO K, TAKEUCHI R. Fourier principles for emotion-based human figure animation [A]. Proceedings of the 22nd annual conference on Computer graphics and interactive techniques [C]. ACM Press, 1995. 91 - 96.
- [3] BRUDERLIN A, WILLIAMS L. Motion signal processing [A]. Proceedings of the 22nd annual conference on Computer graphics and interactive techniques [C]. ACM Press, 1995. 97 - 104.
- [4] LEE J, CHAI J, REITSMA P, et al. Interactive control of avatars animated with human motion data [A]. Proceedings of the 29th annual conference on Computer graphics and interactive techniques [C]. ACM Press, 2002. 491 - 500.
- [5] ASHRAF G, WONG K. Constrained frame-space interpolation [J]. Computer Animation, 2001. 61 - 72.