

中图分类号: TP391.9 文献标志码: A 文章编号: 1006-8961(2011)06-1070-10

论文索引信息: 袁铨, 张建中, 姜銜, 周珊珊, 曹跃. 动态3维场景中多角色动画的交互式模拟研究 [J]. 中国图象图形学报, 2011, 16(6): 1070-1779

动态3维场景中多角色动画的交互式模拟研究

袁铨¹⁾, 张建中²⁾, 姜銜²⁾, 周珊珊²⁾, 曹跃²⁾

¹⁾(电子科技大学软件学院, 成都 611731) ²⁾(电子科技大学计算机科学与工程学院, 成都 611731)

摘要: 当前角色动画的合成大多是在预设的场景中, 采用导入与角色骨骼模型匹配的运动捕获数据的方法, 这就满足不了多种拓扑结构的骨骼模型和实时变化场景的需要。针对上述问题, 提出重定向运动捕获数据到多个任意骨骼拓扑结构的算法, 通过采用以实时3维动态寻径算法为基础的角色智能寻径模型, 结合语音用户接口代替图形用户接口的方法, 实现虚拟角色在动态3维场景中的真实感运动。实验结果表明, 本方法可以合成流畅而逼真的, 与环境实时交互的角色动画, 提高了数据重用性, 降低了动画合成成本, 满足不同动态3维场景中动画合成的需要。

关键词: 运动捕获; 运动重定向; 智能寻径; 动画与场景交互; 语音识别

Interactive simulator research of multi-character animation in dynamic three dimensional scene

Yuan Quan¹⁾, Zhang Jianzhong²⁾, Jiang Xian²⁾, Zhou Shanshan²⁾, Cao Yue²⁾

¹⁾(School of Software, University of Electronic Science and Technology of China, Chengdu 611731 China)

²⁾(School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu 611731 China)

Abstract: Import motion capture data which have the matching skeleton with character is often used in preset scene in character animation synthesis now. However, this method could not afford the demand of multiple topological skeleton structures and the changing scene in real time. In view of the above questions, this paper proposes a new algorithm of retargeting motion capture data to skeleton model with multiple arbitrary topologies and adopts intelligent character model based on the real time three dimensional path-planning algorithm, integrating with the method of using voice user interface instead of graphical user interface to implement verity movement of virtue character in a dynamical three dimensional scene. The result demonstrates that slick and interactive animations can be synthesized by the proposed method. The method downs the cost of animation synthesis by improving the reuse rate of motion data and can be used widely in human-computer interaction domain by satisfying the need of synthesis of character animation in varied three dimensional virtual scenes.

Keywords: motion capture; motion retargeting; intelligent path-planning; interaction between animation and scene; speech recognition

0 引言

现有的角色动画一般只支持导入与角色骨骼模

型匹配的运动捕获文件来生成动画, 而无法导入骨骼拓扑结构各异的运动捕获文件。同时, 合成的动画无法与场景实时交互。因此, 既能不受骨骼拓扑结构的影响, 又能实时响应动态场景变化的动画合

收稿日期: 2009-09-01; 修回日期: 2010-05-09

基金项目: 国家高技术研究发展计划(863)项目(2007AA010407); 中央高校基本科研业务费专项资金项目(ZYGX2009J065); 四川省重点实验室开发基金项目(2008KJ01)。

第一作者简介: 袁铨(1986—), 男。软件工程专业硕士研究生, 主要研究方向为角色动画。E-mail: yuanlquan7@yahoo.cn。

成方法就显得日益迫切。

利用运动捕获的数据驱动人体模型的运动,是运动捕获数据应用的一个重要方面。但是,运动捕获数据仅支持唯一的骨骼模型,无法适用于角色模型可变环境下,各式各样的骨骼模型。为此,出现了运动重定向技术。

目前研究运动重定向的方法主要有3种:1)基于反向运动学,例如通过反向率^[1]控制的方法,通过研究骨干长度比例结合反向运动学的方法^[2],但它们是针对骨骼长度差异,没有针对骨骼拓扑结构不同的重定向;2)基于时空约束^[3-4]的方法,时空约束解决重定向问题运算量大,收敛速度慢;3)基于物理的方法^[5],即在重定向中添加了物理约束,可以生成符合物理条件的运动,计算比较复杂。

大多数运动重定向算法关注于角色大小的变化及重定向后的运动是否具有原始运动的一些约束特性。对基于骨骼拓扑结构变化的研究并不多,但这方面的研究却可以使运动捕获数据来源更加广泛,动画建模不再需要参照运动捕获骨骼模型,具有较强的实用价值。Monzani^[6]提出了建立中间骨骼的方法重定向具有任意拓扑结构的人体骨骼模型,该方法需要额外的存储空间保存中间骨骼的运动数据,会造成系统开销过大,特别是对大小在1MB以上的运动捕获数据重定向时,程序效率显著下降。

合成角色动画,如果角色与场景无法交互,那么场景改变了,就需要重新制作动画。可以与场景交互的动画角色将在很大程度上节省动画合成的时间,拓展角色动画的功能。角色与场景的交互,涉及角色在场景中智能寻径的问题。在目前研究机器人智能寻径的算法中,distance-propagating^[7]算法与其他大多数寻径的算法相比,具有高效和不依赖地面网格全局信息的优势。但是如果障碍物的边界没有落在网格点上,如何处理障碍物占据的网格点,该算法没有对此作出说明和处理;为使机器人能够在运动过程中,与障碍物始终保持一个安全距离,该算法提出使用penalty function^[8]来解决这个问题,但是该方法复杂,增加了额外的开销;作为碰撞避免的寻径算法,它也只是针对2维空间中机器人寻径。

角色动画的制作一直离不开对图形用户接口(GUI)面板的依赖,例如著名的角色动画软件:MotionBuilder、MAYA、Endorphin,都需要动画师熟练地使用GUI提供的各种工具,凭借丰富的经验和熟

练的技能才能合成理想的角色动画。语音识别引擎的出现使语音用户界面(VUI)代替图形用户界面成为可能,让角色动画的合成变得简单和快捷,同时拓展了人机交互的方式。

在结合语音引擎的角色动画研究中,主要有两方面的应用:语音驱动人脸动画和语音驱动角色动画。在第二方面的研究中,Wang^[9]提出了语音控制动画角色和摄像头运动的方法,以运动捕获数据为基础,综合运用语音和鼠标输入合成复杂而细致的动画。但是,该文献的人体模型直接以运动捕获数据为基础建模,并没有考虑当动画角色模型的骨架结构与运动捕获模型的骨架结构不一致时,无法使用运动捕获数据的情况,即动画模型改变将会导致运动捕获数据无法驱动角色运动或者动作严重失真,所以该文献中角色的运动就受运动捕获文件库大小的限制,也没有考虑3维虚拟场景中需要实时添加障碍物,角色动态避障寻径的问题,使合成的角色动画无法与场景实时交互,场景一旦改变就需要重新制作角色动画;Ozan^[10]结合Extensible 3D(X3D)、Virtual Reality Modeling Language(VRML)、Humanoid Animation(H-Anim) Standard和Java Speech API,基于网络设计和开发了VUI来映射一组行为到虚拟人模型,控制角色的运动,该文献提出了在网络环境中应用语音控制角色行为的方法,但是没有考虑角色与虚拟环境交互的问题。

针对上述基于语音引擎控制角色动画合成方法中存在的问题,提出基于运动重定向和角色智能寻径模型,结合语音控制驱动角色运动的真实感动画合成方法。本方法解决了角色骨骼与运动捕获骨骼的不一致问题,实现了在动态场景中角色动画的实时合成,动画合成也不再依赖于对GUI面板的熟练操作。

1 系统流程图

图1简要给出了动态3维场景中多角色动画的交互式模拟研究的系统流程。该图介绍了这样的概念:以重定向技术和3维动态寻径算法为基础,以语音驱动为辅助,角色动画不再受运动捕获文件骨骼拓扑结构的限制,实时的响应虚拟3维场景的变化。

本系统分三大主要部分:

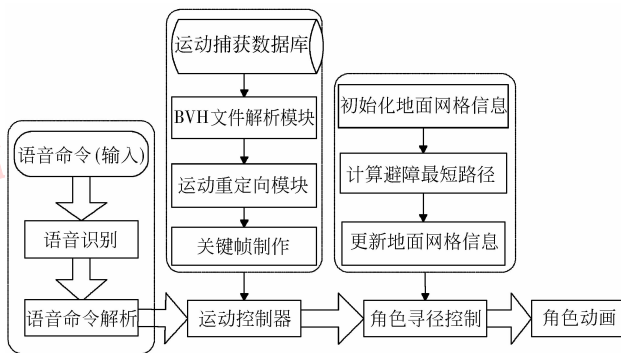


图 1 系统流程

Fig. 1 System flowchart

1) 运动重定向处理

(1) 映射运动捕获骨骼的关节数据到系统定义的动画角色骨骼对应关节, 结合反向运动学修正映射后的运动数据;

(2) 为角色加载经过重定向后的运动捕获动画。

2) 角色智能寻径

(1) 实时检测虚拟场景地面网格的状态信息, 计算从角色位置到目标位置的最短路径;

(2) 协调多个角色在场景中的动态避障和路径规划。

3) 语音控制角色运动

(1) 语音引擎识别并保存符合系统定义的语音命令, 这些语音命令在 Microsoft Speech API 中的 XML 文件中通过上下文无关语法预先定义;

(2) 解析语音命令的关键词, 在运动捕获数据库中找到对应的动作, 为角色实时加载运动数据。

2 运动重定向

提出直接数据对位结合循环坐标下降 (CCD) 的方法, 与其他重定向的方法相比, 本方法针对任意人体骨骼拓扑结构的运动重定向, 而且运算量小, 具有较高的适用性。

2.1 人体骨骼建模

为了简化人体骨骼模型, 使建模和模拟简单化, 又能不影响运动的真实性, 在最大限度上保持人体运动的真实性, 本系统采用了 16 个关节的人体动画骨骼模型, 图 2 是动画角色模型的虚拟骨架示例图。

骨骼模型中六自由度的关节只有根关节 Hip, 它有 3 个平移自由度和 3 个旋转自由度; 关节 Left_

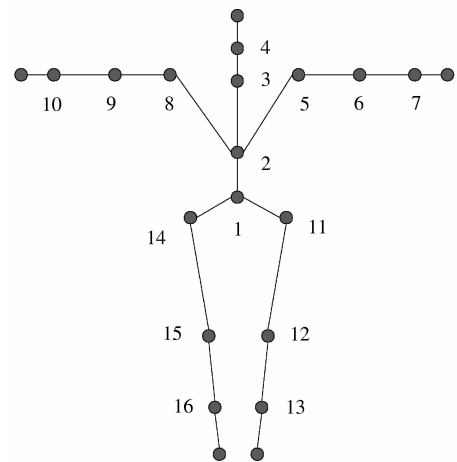


图 2 动画角色骨骼模型

Fig. 2 Animation skeleton model

shoulder、Right_shoulder、Trunk_comp、Left_thigh、Right_thigh 有 3 个旋转自由度; 其余各关节均只有 2 个自由度。

本系统采用的运动捕获文件 (BVH 格式) 来自于 CMU Graphics Lab Motion Capture Database, 当然本方法同样针对其他来源的运动捕获文件, 采用 CMU 运动捕获数据库的原因是该数据库有 1.15 GB, 是目前动作最全, 文件最多的运动捕获数据库。通过对运动捕获文件中描述的骨骼建模得到如图 3 所示的骨骼模型。

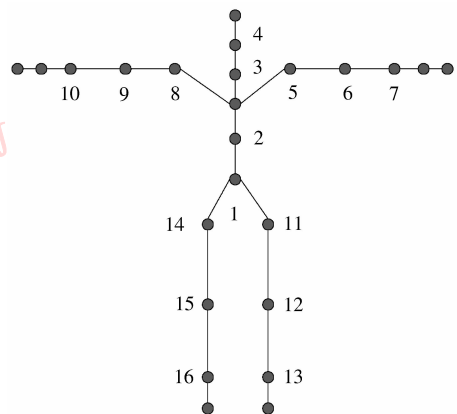


图 3 运动捕获骨骼模型

Fig. 3 Motion capture skeleton model

由图 3 可知, 运动捕获骨骼模型具有 31 个关节 (图中只能看见 24 个关节, 部分关节被遮挡未画出)。采用的动画骨骼模型和运动捕获骨骼模型有以下差异:

1) 人体模型的骨骼长度不一致。

2) 人体模型的关节层次结构不一致,图 4 是系统定义的动画角色骨骼模型,图 5 是运动捕获骨骼层次结构模型。

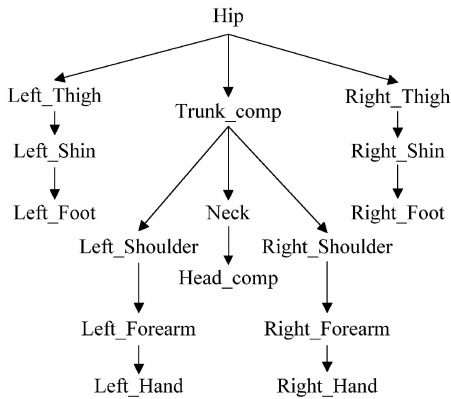


图 4 动画角色骨骼层次结构图

Fig. 4 Hierarchical chart of animation skeleton

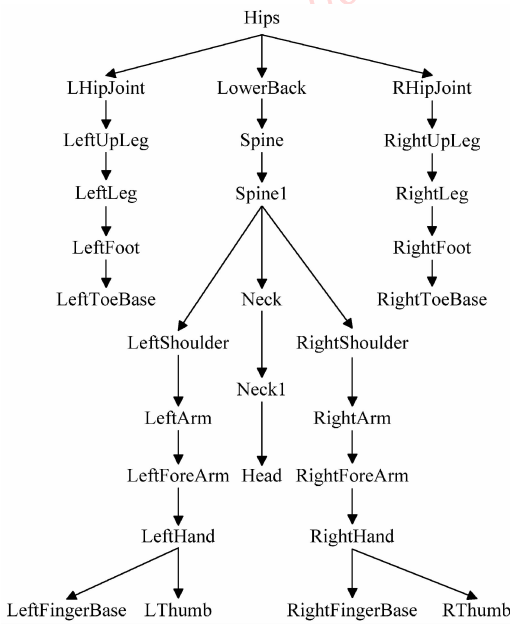


图 5 运动捕获骨骼层次结构图

Fig. 5 Hierarchical chart of motion capture skeleton

3) 关节自由度不一致:运动捕获骨骼模型的根关节具有 3 个平移自由度和 3 个旋转自由度,其他各个关节均只具有 3 个旋转自由度,动画角色骨骼模型各个关节具有不同的自由度个数。

4) 关节绕轴旋转顺序不一致:运动捕获骨骼模型的所有关节是 ZYX 的旋转顺序,动画角色骨骼模型关节分别具有 XYZ、YX、YZ、XZ、X 5 种旋转顺序。

这些差别给运动捕获数据的重用带来很大困难。

2.2 建立关节映射关系

运动重定向需要解决的一个重要问题是在两个骨骼模型之间建立映射关系,映射关系建立的正确与否直接影响重定向结果的真实性。因为在重定向中两种骨骼的关节数目和各关节自由度不一致,造成有些关节找不到对应关系的关节,这些多余的关节如果对运动的效果影响不大,可以忽略。要使重定向后的运动与原运动保持较高的相似性,动画骨骼模型的自由度必须少于运动捕获模型^[6]。

建立两个骨骼关节之间的映射关系,如图 2—3 所示,使编号相同的关节一一对应,对应关系如表 1 所示。

表 1 关节对应关系表

Tab. 1 Corresponding relation of joints

关节标号	动画角色骨骼关节名称	运动捕获骨骼关节名称
1	Hip	Hips
2	Trunk_comp	Spine
3	Neck	Neck1
4	Head_comp	Head
5	LeftShoulder	LeftArm
6	Left_Forearm	LeftForeArm
7	Left_Hand	LeftHand
8	Right_Shoulder	RightArm
9	Right_Forearm	RightForeArm
10	Right_Hand	RightHand
11	Left_Thigh	LeftUpLeg
12	Left_Shin	LeftLeg
13	Left_Foot	LeftFoot
14	Right_Thigh	RightUpLeg
15	Right_Shin	RightLeg
16	Right_Foot	RightFoot

运动捕获骨骼中没有标号的关节,即找不到对应关系的关节作忽略处理,在这里忽略了运动捕获骨骼模型的 15 个关节,冗余关节并不是主干关节,所以重定向后的运动几乎不会受到影响。

映射关系建立好之后,并不能直接从源骨骼模型(运动捕获骨骼模型)拷贝数据到目标骨骼模型(动画骨骼模型),因为这两种骨骼模型的关节具有不同的局部坐标系。

2.3 数据映射

在拷贝源骨骼的关节数据到目标骨骼的关节时,需要计算旋转矩阵模型以执行数据的映射,把 ZYX 的局部坐标系变换为 XYZ、YX、YZ、XZ、X 表示的局部坐标系。根据源骨骼的关节旋转顺序 ZYX,构建坐标系矩阵 R ,给定目标骨骼的关节旋转顺序,

就可以反求出目标骨骼关节绕轴旋转的欧拉角。运动捕获文件(BVH 格式)中,关节的旋转分量是以欧拉角的形式记录的,用欧拉角表达方位存在或给定方位的表达方式不唯一,因此直接通过欧拉角计算关节的旋转矩阵,重定向的结果会产生违背人体生理规律的运动,例如空翻和不停地旋转。在这里,引入四元数解决此问题。

提取关节的欧拉角旋转分量,转换成四元数,以下是从惯性到物体坐标系的数学表达式:

$$q = q_{roll} \cdot q_{yaw} \cdot q_{pitch} \quad (1)$$

$$q_{roll} = [\cos(r/2), (0, 0, -\sin(r/2))] \quad (2)$$

$$q_{yaw} = [\cos(y/2), (0, -\sin(y/2), 0)] \quad (3)$$

$$q_{pitch} = [\cos(p/2), (-\sin(p/2), 0, 0)] \quad (4)$$

其中, q_{roll} 、 q_{yaw} 、 q_{pitch} 分别为绕 Z 、 Y 和 X 轴旋转的四元数; r 、 y 和 p 分别为绕 Z 、 Y 和 X 旋转的欧拉角。通过求得的四元数 $q = (w, x, y, z)$ 计算关节的旋转变换矩阵 R , 数学表达式为

$$R = \begin{bmatrix} 1 - 2y^2 - 2z^2 & 2xy + 2wz & 2xz - 2wy \\ 2xy - 2wz & 1 - 2x^2 - 2z^2 & 2yz + 2wx \\ 2xz + 2wy & 2yz - 2wx & 1 - 2x^2 - 2y^2 \end{bmatrix} \quad (5)$$

根据已求得的旋转矩阵 R 和目标骨骼关节的旋转顺序作为旋转轴参数,从矩阵 R 中提取符合条件的欧拉角。如果目标骨骼的关节旋转顺序是 XYZ , 关节绕 3 维坐标轴旋转的欧拉角分别设为 θ 、 δ 和 η 。则该旋转矩阵的计算如下:

$$R' = R_x(\theta) \cdot R_y(\delta) \cdot R_z(\eta) \quad (6)$$

虽然关节的局部坐标系不一样,但是它们要具有相同的动作就必然有相同的旋转矩阵,局部坐标系只是一个表示的手段。令

$$R = R' \quad (7)$$

可以求出 θ 、 δ 和 η 。因为两种骨骼的关节自由度个数不一致,数据映射的结果会造成自由度的丢失,直接的后果就是重定向的运动与原运动有差异,这一问题必须通过调整关节的位置来解决。

迭代计算运动捕获骨骼中每一个关节与其父亲节点连线构成的向量 I_1 和动画骨骼中对应关节与其父亲节点连线构成的向量 I_2 所形成的夹角 δ , 把 I_2 绕向量 I_3 旋转 δ , 得到修正后的骨骼位置,同时修改关节的旋转欧拉角分量。其中 I_3 的计算如下:

$$I_3 = I_1 \times I_2 \quad (8)$$

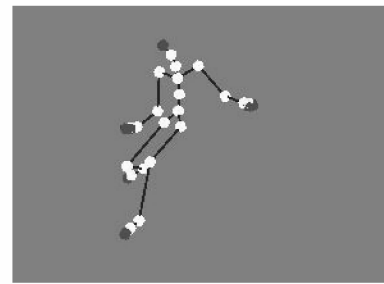
2.4 反向运动学修正关节位置

经过上述处理之后,依然会存在部分运动数据

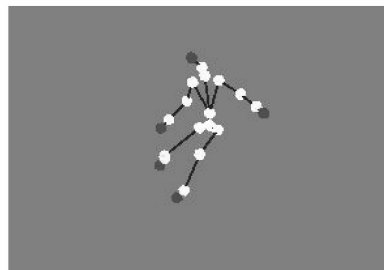
的重定向结果失真,例如产生双腿扭曲和滑步。考虑到反向运动学中 CCD(循环坐标下降)算法解算速度快,满足实时性的要求,适合各种自由度的关节,本系统选择了 CCD 算法来添加约束并修正运动重定向后部分运动失真的问题。

上述过程处理后,运动重定向就有了比较满意的效果了。对 CMU Graphics Lab Motion Capture Database 编号为 01 ~ 09 的共 109 个文件作测试,结果不满意的文件仅有 08_11。对于大小在 1 MB 以上的运动捕获文件,该方法比建立中间骨骼的方法^[8]快 5 ~ 8 s。重定向的结果可以为该系统的人体模型提供丰富的动作数据文件支持。

如下两幅图展示了踢球运动的重定向。图 6 (a) 角色的运动是 CMU 的运动捕获文件直接导入产生的动作, (b) 角色的运动是通过本方法重定向后产生的运动,可以看出,该方法成功的重定向了骨骼拓扑结构不同的运动,同时最大限度保持了目标运动与源运动的一致性。



(a) 源运动



(b) 重定向后的目标运动

图 6 踢球运动的重定向

Fig. 6 Motion retargeting of kicking

3 角色智能寻径

为了更好地研究和展现虚拟角色与 3 维场景的智能交互,本系统添加了智能寻径模块,研究虚拟角色在静态和动态的虚拟场景中根据语音命令做出的实时响应。为解决动态 3 维环境中角色与场景交互

的问题,针对 distance-propagating^[7]算法做了以下 3 个方面的改进:

- 1)增加了 3 维场景中随机出现的障碍物占据所有网格点的快速计算方法。
- 2)扩展该算法从 2 维场景到适合多个角色在 3 维虚拟场景中的动态避障。
- 3)简化了原算法关于障碍物安全边界的设定。改进后的方法提高了执行效率,且易于实现。图 7 所示是改进后的算法详细流程图。

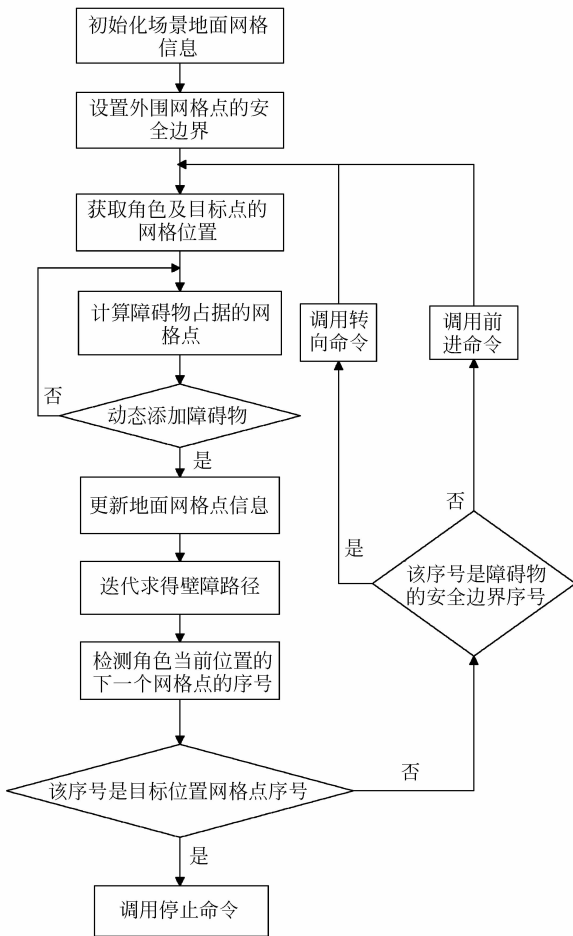


图 7 智能寻径算法流程图

Fig. 7 Intelligence path-planning algorithm flow chart

该算法主要分 3 部分:

- 1)根据地面网格信息求解角色位置到目标位置的最短路径。
- 2)实时更新地面网格信息:检测是否有障碍物出现,目标位置是否发生了移动及角色位置的变化,及时反馈给第一步,从而更新路径信息。
- 3)实时调整动画角色的运动方向:根据上面两步的结果及角色当前运动朝向和路径朝向的夹角,

调整动画角色根关节的位置,从而使角色始终沿着路径运动。

3.1 地面网格初始化

本方法把动画角色所在的场景地面作为一个网格,设计 4 种不同的网格点:障碍物占据的网格点、角色占据的网格点、目标位置占据的网格点和无任何物体的网格点。为场景中的每一个网格点设置一个变量 d_i 来存储每一次迭代中它到最近目标点的距离。 d_i 的初始化赋值如下:

$$d_i(0) = \begin{cases} 0 & i \text{ 是目标网格点} \\ D & i \text{ 是其他网格点} \end{cases} \quad (9)$$

式中, $D > (M - 1) d_{\max}$, M 表示场景中的网格点总数, d_{\max} 表示两个邻接点之间的最大值。

但通过语音驱动的障碍物是随机出现在虚拟场景地面上的,障碍物的底面不一定也不可能刚好落在网格点上,因此这里对障碍物占据的网格点作近似处理,如图 8 所示。

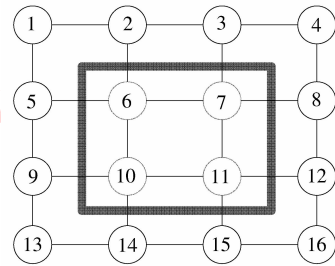


图 8 障碍物占据的网格点

Fig. 8 Mesh points occupied by obstacle

图 8 中黑色粗线条围起来的区域是障碍物区域,可见它的四周并没有落在网格点上,而是仅仅包含序号为 6、7、10、11 的 4 个网格点。根据求得的障碍物中心坐标和障碍物的边长值,求得障碍物底面各个顶点的坐标位置。如图 8 所示,如果已经求得了障碍物左上角顶点位置 (x_j, y_j) ,同时该点落在网格点 1、2、5、6 围成的矩形区域内,根据 (x_j, y_j) 与这 4 个网格点的位置坐标包围关系,求得网格点 6 应该在障碍物区域内,而其他网格点落在障碍物区域外,同理可得网格点 7、10、11 落在障碍物区域内,同时网格点 6、7、10、11 是障碍物所占据的外围网格点,因为网格点的排列是有一定顺序的(该顺序可以根据需要设置),根据这个顺序,障碍物中心位置及障碍物边长,计算包含在外围网格点里的所有障碍物网格点也是很容易的。

3.2 3 维寻径

如果角色不是在 2 维平面上寻径, 需要考虑两点: 1) 角色所处的非平面区域要覆盖网格点, 角色的 3 维寻径实际上是沿着网格点在运动; 2) 必须根据实际场景给角色加载相应的动作, 应通过角色的根关节离地高度来判断角色所在的位置是否是 3 维地形, 然后调用相应的动画。

在 3 维环境中, 还必须考虑寻径时虚拟角色的朝向。否则, 会出现角色即使在向路径规定的方向运动, 但角色正面的朝向依旧没有改变。因为虚拟地面用网格点表示, 所以角色朝向的变化是以 45° 步长变化的, 如图 9 所示, 角色在 10 号网格点时, 它可以转向的方向只有如图中黑色粗线条表示的八个方向, 这些方向和它当前前进的方向始终是成 45° 的, 因此, 角色运动过程中路径修正时, 只需使角色根关节转向 45° 便可以。

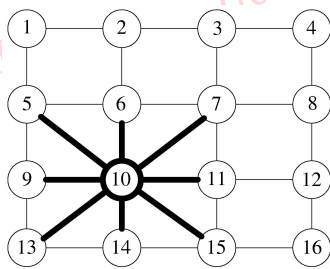


图 9 角色朝向示意图

Fig. 9 Character orientation diagram

因为场景地面不可能被无限细分, 而虚拟角色所在的位置抽象化为一个点, 所以这一点也不一定会落在分好的网格点上, 这时有必要对角色所占据的网格点做一近似, 以离角色最近的网格点为角色所在位置。

3.3 静态避障及路径计算

场景中除了角色运动, 障碍物和目标位置都是静态的。根据 3.1 节提出的地面网格初始化方法和以下的迭代表达式^[7]:

$$d_i(n+1) = \begin{cases} 0 & i \text{ 是目标位置网格点} \\ D & i \text{ 是障碍物网格点} \\ \min(d_{ij} + d_i(n)) & \text{其他} \end{cases} \quad (10)$$

式中, n 是迭代的次数, d_{ij} 表示网格点 i 和网格点 j 通过无障碍物的网格点的最短距离。在一次迭代后, 目标位置的 d_i 将会变为 0, 它的邻接点根据上述表达式计算出到目标位置的实际最短距离; 再经过

一次迭代后, 这些点的邻接点就会计算出到目标位置的实际最短距离, 如此直至迭代到当前角色所在位置的网格点。于是避障的路径便求得。

3.4 动态避障及安全边界设定

动态避障的情况下, 障碍物和目标位置可以随机移动, 地面网格点的状态也就相应地发生了改变, 同时 d_i 所存储的值会随之改变成数学表达式(10)中 3 种值的一种。此时根据值的变化, 再进行迭代, 就可以生成动态避障的路径。

为了避免虚拟角色和障碍物发生碰撞, 每个网格点周围设置一个安全距离范围^[8], 而本文的方法仅仅为障碍物占据的外围网格点设置安全边界距离, 即在求得障碍物占据的外围网格点后, 根据安全边界距离值扩展这些网格点的邻近点的序号为临界点集 L , 实时获取避障路径上角色当前位置的下一网格点序号 A , 如果点 $A \in L$, 则立即调用角色转向命令修正角色寻径路线, 这样就可以避免角色穿透障碍物或者与障碍物碰撞。

4 语音控制角色运动

本系统利用 SAPI^[11] 作为语音识别引擎。SAPI 的命令和控制特征作为上下文无关文法 (context-free grammars) 而实现。上下文无关文法通过 XML 定义了语法结构和语法规则。把系统可能用到的语音命令写入 XML 格式的文件, 系统运行时自动加载该语音命令 XML 文件。当接收到语音命令时, 自动捕获语音命令, 并与 XML 文件中记录的语音命令匹配, 如果该语音命令有效, 则保存该命令, 然后调用对应的动画数据。

语音控制角色运动模块由语音识别和角色动作控制两个子模块组成。

语音识别模块主要是识别语音命令, 分析和保存有效的命令。系统定义的语法规则在这个子模块中被加载, 语音命令被语音引擎捕获后, 系统将该命令与语法规则中的命令进行匹配, 匹配成功, 则解析该命令, 提取关键字保存, 作为下一步调用对应的控制角色动作命令的依据。该子模块在解析语音命令时, 针对的是一条语音命令中的关键字或词。例如语音命令“turn left”, 只要“left”匹配成功, 即视为该命令匹配成功, 并且仅保存关键词“left”。这样就提高了语音引擎的识别率。当然, 复杂的语音命令可以有多个关键词。

角色动作控制模块对避障路线进行规划,对角色朝向进行调整。它接收语音识别模块输入的语音命令,根据该命令调用相应的动作控制接口,例如角色是该寻径还是转身,分析角色当前的运动状态,计算角色当前所在的位置,调整角色运动路线,控制角色的当前动作。图 10 是语音控制角色运动的框架图。

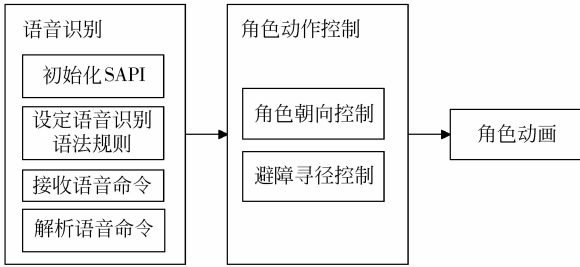


图 10 语音控制角色运动的框架图

Fig. 10 The framework of voice controls character movement

为了角色模型可以实时地响应语音命令的控制,本文设计的是在系统中预先加载好做成的一段动画数据,这样,发出的语音命令被识别和解析后,就调用相关的动画数据,达到实时驱动角色响应语音命令。如果要响应“walk”、“turn left”和“turn right”的命令,就需要加载相应的 3 段动画,依次为行走、左转和右转。

5 实验结果

按照上述方法,在动画合成系统 Anima Studio 平台上用 Visual Studio 2005 与 OpenGL 实现一个语音控制多角色环境下智能动画合成的原型程序,角色模型采用几何体建模^[12],并对动画骨骼模型蒙皮。编辑器界面如图 11 所示,界面右侧是鼠标输入控制工具栏,用于实时显示输入的语音控制命令。

在该编辑器中实现了运动重定向,多角色动态避障寻径和语音控制角色行为的功能。如图 12 所示以重定向后的步行运动为例说明本方法的效果。图 12 显示的是语音控制角色的 3 维寻径过程,动画帧数稳定在 13 ~ 25 帧/s,图中的斜面是提前搭建好的,目标点(白色球体)是实时随机指定的。

图 13 显示的是语音控制双人静态避障寻径,动画帧数稳定在 12 ~ 20 帧/s,虚拟场景中的障碍物是提前搭建好的,图中的白色球体是目标点,在语音控制下,两个虚拟角色共同向目标点运动。

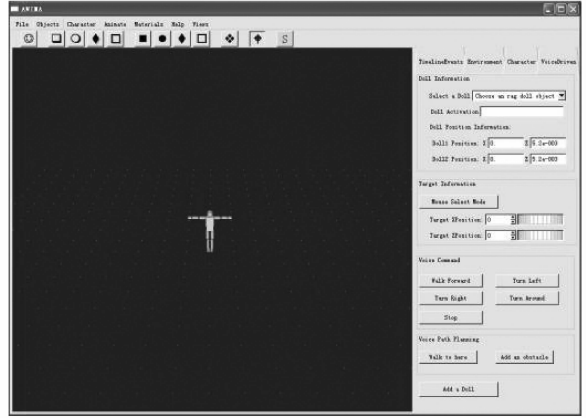


图 11 编辑器界面

Fig. 11 Editor interface

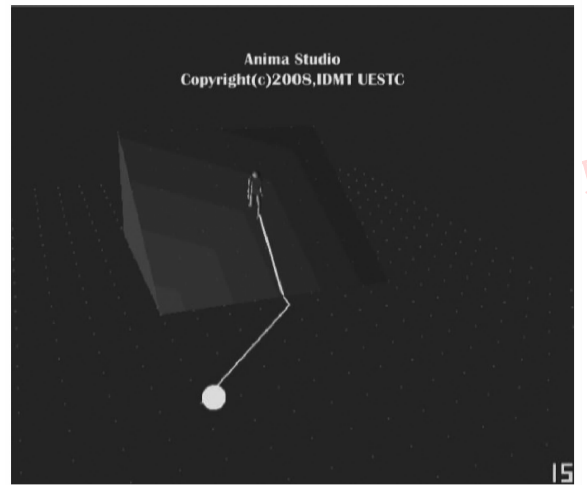


图 12 语音控制 3 维寻径

Fig. 12 Three dimensional path-planning under voice control

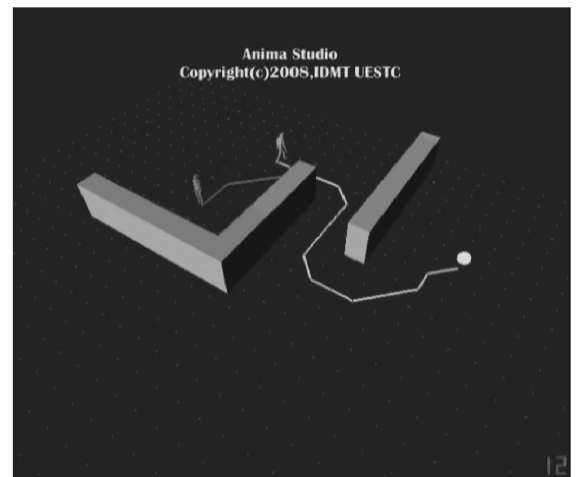


图 13 语音控制双人静态寻径

Fig. 13 Static path-planning of double under voice control

图 14 显示的为双人动态寻径,动画帧数稳定在 13 帧/s 左右,在虚拟人向目标位置运动时,场景中无障碍物,在运动过程中,通过语音命令,先后随机添加了两个几何体作为障碍物,图中深色和浅色的两条线代表了从角色位置到目标位置无障碍物的路径。

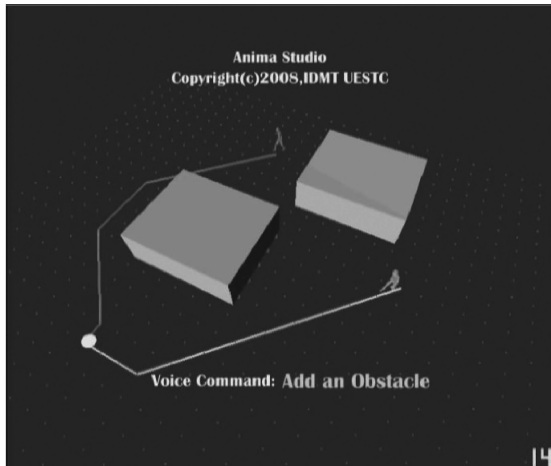


图 14 语音控制双人动态避障

Fig. 14 Dynamic path-planning of double under voice control

对 CMU 的 Motion Capture Database 中编号为 01 ~ 09 共 109 个运动捕获文件作测试,包括的动作涉及 playground、walk、walk on uneven terrain、dance、basketball、run 共 6 类,通过本文的方法可以合成质量较好的角色动画;本系统所使用的语音识别引擎在同一个人经过训练的前提下,语音识别成功即系统作出正常响应的效率能够达到 90% 以上。

6 结 论

提出语音控制多角色环境下智能动画合成的方法,并对相应的算法进行了实验论证。本文提出的方法,与传统的动画制作软件比较,具有以下特点:

1) 实用性 目前主流的角色动画软件的重定向功能对骨骼拓扑结构一致性的要求较高,例如 Endorphin 等,无法对骨骼拓扑结构完全不同的角色动画重定向;而本方法效率与同类算法相比,简便而高效,针对任意骨骼拓扑结构重定向问题,使运动捕获文件来源更加广泛,提高了运动捕获文件的重用性。

2) 易用性 本方法把运动重定向和智能寻径引入角色动画的合成,结合语音驱动控制,用户只需

发出语音命令就可合成与环境交互的角色动画。

3) 高效性 无论该方法中的运动重定向算法,还是智能寻径算法,本方法的开销和执行时间与文中提到的算法相比都要小得多。

但是在研究和实现过程中,存在以下两个方面的难点,也即是未来研究的方向:

1) 通过运动重定向技术重用动作文件,必然会造成动画的在一定程度上失真,而且本文提到的重定向没有考虑物理属性,因此,对于运动重定向技术的改进,可以提高动作的真实感。

2) 实验结果虽然可以添加两个及两个以上的虚拟角色,但随着角色个数的增加,在动画合成过程中,响应速度会下降,效率也会变低,未来的希望能在角色个数增加的同时,效率的下降可以控制在可以接受的范围内。

参考文献 (References)

- [1] Choi K, Ko H. On-line motion retargetting [J]. Journal of Visualization and Computer Animation, 2000, 11(5):223-235.
- [2] Yang Xinian, Zhang Jiaming, Zhao Shibin. Motion retargetting with geometry scaling [J]. Journal of Image and Graphics, 2002, 7(9):871-875. [杨熙年, 张家铭, 赵士宾. 基于骨干长度比例之运动重定目标算法 [J]. 中国图象图形学报, 2002, 7(9):871-875.]
- [3] Gleicher M. Retargetting motion to new characters [C]// SIGGRAPH 98 Conference Proceedings, Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques. Orlando, Florida; ACM Press, 1998:33-42.
- [4] Luo Zhongxiang, Zhuang Yueting, Liu Feng, et al. Space-time constraints based motion editing and motion retargetting [J]. Journal of Computer-Aided Design & Computer Graphics, 2002, 14(12):1146-1151. [罗忠祥, 庄越挺, 刘丰, 等. 基于时空约束的运动编辑和运动重定向 [J]. 计算机辅助设计和计算机图形学学报, 2002, 14(12):1146-1151.]
- [5] Tak S, Ko S. A physically-based motion retargetting filter [J]. ACM Transactions on Graphics, 2005, 24(1):98-117.
- [6] Monzani J S, Baerlocher P, Boulic R, et al. Using an intermediate skeleton and inverse kinematics for motion retargetting [J]. EUROGRAPHICS, 2000, 19(3):245-250.
- [7] Willms A R, Yang S X. An efficient dynamic system for real-time robot path-planning [J]. IEEE Transactions on Systems, Man and Cybernetics, Part B. 2006, 36(4):755-766.
- [8] Willms A R, Yang S X. An extension of the distance-propagating dynamic system for robot path planning to safe obstacle clearance [C]// Proceedings of the IEEE International Conference on

- Robotics and Biomimetics. Washington, USA; IEEE, 2006;1396-1401.
- [9] Wang Z, M. van de Panne. "Walk to here": a voice driven animation system [C] // 2006 ACM SIGGRAPH / Eurographics Symposium on Computer Animation. Vienna, Austria; ACM 243-250.
- [10] Ozan Apaydin. Networked Humanoid Animation Driven by Human Voice Using Extensible 3D (X3D), H-Anim and JAVA Speech Open Standards [D]. Monterey, California; Naval Post Graduate School March, 2002.
- [11] Microsoft Corporation. Microsoft Speech SDK 5.1 Documentation [EB/OL]. (2009-03-03) [2009-06-08]. [http://www.microsoft.com/downloads/details.aspx? FamilyID = 5e86ec97-40a7-453f-b0ee-6583171b4530&displaylang = en#filelist](http://www.microsoft.com/downloads/details.aspx?FamilyID=5e86ec97-40a7-453f-b0ee-6583171b4530&displaylang=en#filelist).
- [12] Shapiro A, Faloutsos P, Ng-Thow-Hing V. Dynamic animation and control environment [C] // Graphics Interface 2005, Victoria, British Columbia, Canada; Canadian Human-Computer Communication Society, 2005;61-70.