Modeling and Simulation 建模与仿真, 2019, 8(2), 45-52
Published Online May 2019 in Hans. http://www.hanspub.org/journal/mos
https://doi.org/10.12677/mos.2019.82007

Hans汉斯

Research on Application of VEGA PRIME-Based MD5 3D Animation

Guotao Zhu, Xiuzhi Zhou, Wenting Hu

Naval Aviation University, Qingdao Branch, Qingdao Shandong
Email: zealotsparc@163.com

Received: Apr. 25th, 2019; accepted: May 13th, 2019; published: May 20th, 2019

Abstract

This article introduces the technical classification and characteristics of 3D animation, and analyses the file format and implementation principle of MD5 3D skeletal skin animation. The skeleton movement and skin calculation process are realized by C++ programming, a plugin for integrating and using MD5 3D animation in VEGA PRIME visual simulation application is designed and implemented by using VEGA PRIME real-time simulation engine and its plugin design specification, which solves the problem that VEGA PRIME does not support mainstream 3D animation.

Keywords

Virtual Reality, Visual Simulation, 3D Animation

基于VEGA PRIME的MD5三维动画应用研究

朱国涛，周秀芝，胡文婷

海军航空大学青岛校区，山东 青岛
Email: zealotsparc@163.com

收稿日期：2019年4月25日；录用日期：2019年5月13日；发布日期：2019年5月20日

摘　要

本文介绍了三维动画的技术分类和特点，在分析MD5骨骼蒙皮动画文件格式和动画原理的基础上，通过C++编程实现了骨骼运动以及蒙皮计算过程，应用VEGA PRIME实时仿真引擎，结合其扩展插件设计规范，设计和实现了在VEGA PRIME视景仿真应用程序中集成和使用MD5三维动画的插件，解决了VEGA PRIME不支持主流三维动画的问题。

文章引用: 朱国涛, 周秀芝, 胡文婷. 基于 VEGA PRIME 的 MD5 三维动画应用研究[J]. 建模与仿真, 2019, 8(2): 45-52.
DOI: 10.12677/mos.2019.82007

关键词

虚拟现实，视景仿真，三维动画

1. 引言

三维动画是计算机图形技术领域中一个重要的研究方向，它可以清晰、直观、逼真地展现角色的动作和行为过程，广泛应用于医学、教育、军事、娱乐等领域。VEGA PRIME 是 MULTIGEN-PARADIGM 公司(后被 PRESAGIS 公司收购)开发的实时视景仿真软件，它集成了海洋、大气、星空、地形、特效等各种通用模块，并且可以通过扩展插件支持虚拟仪表、红外成像、雷达成像等专业模块，广泛应用于城市规划、海洋仿真、建筑设计、飞行仿真等各个领域，是目前最流行的视景仿真工具之一[1] [2]。

VEGA PIRME 本身并不支持目前主流的三维动画模型格式，而类似的扩展插件基本上都是商业发布的，其价格居高不下，限制了 VEGA PRIME 的推广应用。如果能够通过插件扩展的方式使得 VEGA PRIME 能够实时的加载、渲染三维动画，那么在飞行仿真、舰船仿真训练时，就可以以三维动画的形式全方位地展示仿真过程中虚拟角色的动作和行为，从而提高 VEGA PRIME 视景仿真引擎的适用性并增强仿真训练效果。

2. 三维动画技术分类简介

三维动画的基本原理是让模型中各个顶点的位置随时间变化，其主要技术有三种：变形动画、关节动画和蒙皮动画。

1) 变形动画中的角色模型由一系列的渐变网格模型构成，变形动画要达到模型细致的效果，必然需要大量的帧数，最理想的情况是渲染程序中的每一帧都对应于模型中一帧，但这样模型数据量会非常大，因此只能存储关键时刻模型顶点的位置，其过度时刻的顶点数据由两个相邻的关键时刻插值获得，这使得变形动画灵活性差，存储开销大；

2) 关节动画的角色模型由若干独立的关节点组成。每一关节点是一个独立的网格模型，对应于人体的一个关节，不同的关节按照角色的特征组织成一个层次结构，通过改变不同关节的位置和旋转，就可以实现各种所需的动画效果，关节动画的优点是存储空间小，缺点是在关节的连接处会有明显接缝，影响真实感；

3) 蒙皮动画是在骨骼系统基础上发展而来的，其基本原理为：在骨骼系统的控制下，模型网格的顶点可以被几块骨骼所影响，每块骨头对顶点影响的权重不同，骨骼对顶点的作用之和就是该顶点的最终效果。蒙皮动画数据内容包括骨骼层次数据、模型网格数据、骨骼蒙皮数据和动画关键帧数据，其中骨骼的运动由动画关键帧数据驱动，蒙皮动画同时兼有关节动画的灵活和渐变动画的逼真，现代主流的三维动画格式均采用了蒙皮动画技术[3] [4]。

3. MD5 模型动画原理及文件格式

MD5 三维动画模型格式是由 ID SOFTWARE 公司推出的世界上首款真正意义上的蒙皮动画格式，在

海拔2004年随着DOOM3一起面世，经过几个版本的更新，现在在蒙皮动画格式中依然有着其重要地位。MD5是在骨骼动画基础上发展来的，其中骨骼的概念与人体骨骼类似，可以把人体看作一堆骨骼，然后外面蒙上一层肌肉、皮肤，在骨骼运动的过程中这些肌肉、皮肤就跟随骨骼的运动而变化。在三维动画中，骨骼与骨骼之间通过关节连接，一根骨骼的一端或两端连着两个关节，而一个关节可能连着数条骨骼，骨骼模型的描述也分以骨骼为主和以关节为主，其中MD5格式属于后者。在MD5格式中，关节的集合可以用一个树型的数据结构描述，首先定义一个总的关节根节点，其下连着一个或者多个子关节，这些子关节本身也作为父关节连着一个或多个子关节。在骨骼变化时，父关节的移动首先传递到子关节上，再叠加上子关节本身的移动，于是就可以建立一个前向的驱动模式，每个关节的运动信息可以抽象成一个三维变换矩阵M，这样这个驱动模型可以看作是每个时刻给予每个节点一个变换矩阵，变换节点的位置和旋转以驱动整个骨骼模型的运转。

MD5蒙皮动画模型由两个文件组成：MD5MESH文件和MD5ANIM文件，前者描述了关节(即骨骼)和拓扑，组成网格的顶点的权重、纹理等信息，后者描述了关节变化的关键帧信息[5] [6]。

3.1. MD5MESH文件结构分析

其中，MD5MESH文件的结构和格式如图1所示。

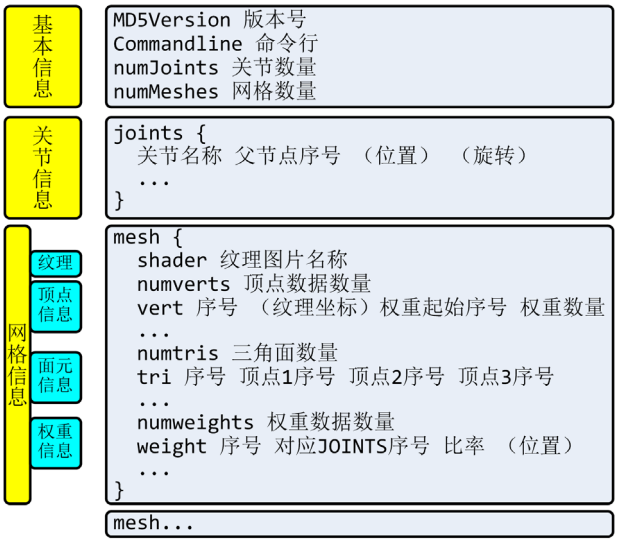


Figure 1. Structure of MD5MESH file
图1. MD5MESH文件的结构

MD5MESH文件由基本信息块、关节信息块和多个网格信息块三部分组成，它是一个文本文件，其中每一行都有特定的格式。

1) 基本信息块中描述了三维动画模型文件的版本号、命令行、关节数量、网格数量等信息；

2) 关节信息块以joints关键字开头，在一个封闭的大括号中描述了组成骨骼系统的各个关节信息，其中的每一行数据都描述一个关节，典型的关节信息示例如下；

“origin” −1 (0.48 0.0 0.0) (0.7 0.0 0.0)
“bip01” 0 (9.52 0.0 0.35) (0.0 0.0 0.0)
…

其中“bip01”是关节名称，其后的数字是父关节的索引序号，−1表示没有父关节，0指向名称为“origin”

的关节，紧跟的一个小括号中的三个值是 bindpose 状态(即骨骼的初始姿态)下关节的位置(p_x, p_y, p_z)，最后一个小括号中的三个值是 bindpose 状态下关节的旋转，用四元数表示，文件中给出了四元数三个分量(q_x, q_y, q_z)，需要按公式 1 自行计算第四个分量 q_w。

获取关节的位置和旋转信息后，可以构造表示关节位置和旋转的局部矩阵 M_L，然后叠加父关节的位置和旋转，即可以得到关节的全局矩阵 M_G，对应的计算公式如公式 2 所示。

$$t = 1 - q_x^2 - q_y^2 - q_z^2$$
$$q_w = \begin{cases} 0 \mid_{t\le 0} \\ -\sqrt{t} \mid_{t>0} \end{cases} \quad \text{(公式 1)}$$

$$M_L = \begin{bmatrix} 1-2q_y^2-2q_z^2 & 2q_xq_y+2q_wq_z & 2q_xq_z-2q_wq_y & 0 \\ 2q_xq_y-2q_wq_z & 1-2q_x^2-2q_z^2 & 2q_yq_z+2q_wq_x & 0 \\ 2q_xq_z+2q_wq_y & 2q_yq_z-2q_wq_x & 1-2q_x^2-2q_y^2 & 0 \\ p_x & p_y & p_z & 1 \end{bmatrix}$$

$$M_G = M_L \times M_{G-\text{Parent}} \quad \text{(公式 3)}$$

公式 2 中 $M_{G\text{-Parent}}$ 是父关节的全局矩阵，如果父关节仍然存在父节点，则可以继续通过上述公式迭代计算。

3) 网格信息块以 mesh 关键字开头，其中包含四类数据：纹理信息、顶点信息、图元信息和权重信息。其中，shader 后的字符串表示网格对应的纹理图片路径；numverts 表示此网格包含的顶点数量；vert 表示顶点，顶点并不是直接通过坐标的形式给出，而是给出了顶点对应权重信息的索引起始序号和权重数量，此外为了实现纹理贴图还给出了顶点对应的纹理坐标；numtris 表示网格所包含的三角面片的数量；tri 开头的行表示三角面片，其后的四个数字分别表示面片序号和组成三角面片的三个顶点的索引；numweights 表示权重信息的数量；weight 开头的行表示权重信息，每个权重信息都给出了序号、对应的关节序号、比率值(即权重值)和位置等信息。

上述信息读取完毕后，就可以进行蒙皮计算，也就是计算顶点的实际位置，计算公式如下：

$$V = \sum_{i=s}^{s+n} \left(P_i \times M_{Gi} \right) \cdot b_i \quad \text{(公式 3)}$$

其中，s 是指顶点对应的权重信息的索引起始序号，n 表示顶点对应的权重数据的数量，P_i 指的是第 i 条权重数据对应的位置，M_{Gi} 表示第 i 条权重数据对应的关节的全局矩阵，用行向量形式的齐次坐标[p_x, p_y, p_z, 1]表示，b_i 指的是第 i 条权重数据对应的比率值(即权重值)，计算结果 V 就是顶点的坐标。

3.2. MD5ANIM 文件格式分析

MD5ANIM 文件也是一个文本文件，如图 2 所示，它由基本信息块、节点集成信息块、包围盒信息块、基础帧信息块和多个关键帧信息块组成。

1) 基本信息块中描述了三维动画模型文件版本号、命令行、关键帧数量、关节数量、播放帧率和骨骼动画变化数据数量等信息。

2) 节点继承信息块以 hierarchy 关键字开头，大括号中的每一行都包括了关节名称，父关节序号，关节信息更新掩码 FLAG 和关节信息更新起始位置，其中关节名称和父关节序号等内容与 MD5MESH 文件中关节信息块的对应内容完全一致。

3) 包围盒信息块以 bounds 关键字开头，每一行对应一个关键帧，描述了在对应的关键帧下模型对象的最小轴对齐包围盒，由一个最小点的坐标和最大点坐标组成。

Figure 2. Structure of MD5ANIM file
图 2. MD5ANIM 文件的结构

4) 基础帧信息块由 baseframe 关键字开头，每一行都表示对应关节位置和旋转，其中前一个括号中的三个值是关节的默认位置，后一个括号中的三个值是关节的默认旋转(四元数表示，需自行计算 q_w)。

5) 关键帧信息块由 frame 关键字+帧序号开头，大括号中包含描述骨骼位置和姿态的数值，其中每一行对应一个关节，一行数据最多包含六个值，至少包含一个值，这些数据构成一个 Animated Components 数组，其长度等于基本信息块中定义的 num Animated Components 的值，关节的实际的位置和姿态需要根据关节的更新掩码 FLAG 和更新起始位置，使用关键帧数据替换对应的基础帧数据得到，其替换规则如图 3 所示。

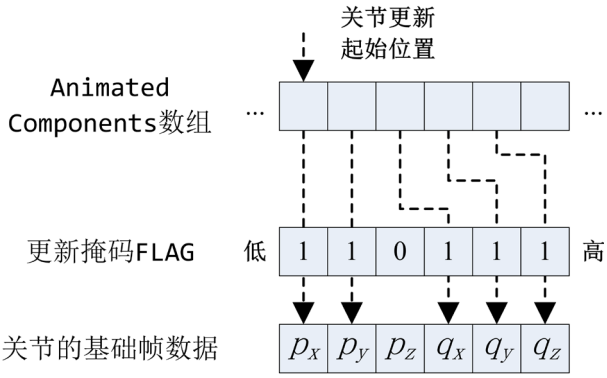


Figure 3. Rules for replacing base frame data
图 3. 替换基础帧数据的规则

更新掩码 FLAG 的值从低到高六个比特分别表示基础帧数据中位置的三个值(p_x, p_y, p_z)和旋转(q_x, q_y, q_z)是否应该被替换，执行替换时首先得到关节对应的基础帧数据和关节的更新起始位置，然后从 Animated Components 数组的关节更新起始位置开始，从低到高依次检查更新掩码 FLAG，如果对应比特位为 1 则使用关键帧数据替换对应的基础帧数据，如对应图中的更新掩码值，则 p_x、p_y、q_x、q_y、q_z 将被 Animated

Components 数组中由关节信息更新起始位置开始标记的 5 个数据依次替换。替换完成后即可得到关键帧状态下此关节的实际位置和旋转，然后就可以根据公式 1、2、3 计算对应的骨骼和蒙皮。

4. VEGA PRIME 动画插件的设计

按照 VEGA PRIME 的插件开发规范，一个完整的 VEGA PRIME 插件包含三部分内容：GCF 文件、XML SCHEMA 文件、可执行文件和库(包括头文件、动态库和链接库) [7] (图 4)。

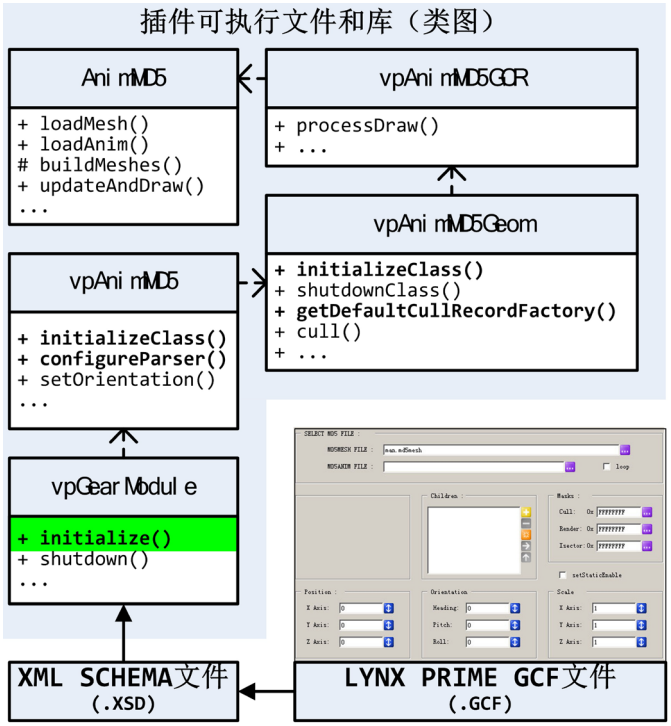


Figure 4. Composition of VEGA PRIME 3D animation plug-in
图 4. VEGA PRIME 三维动画插件组成

GCF 文件用于描述在 LYNX PRIME 环境下插件的图形交互配置界面，它通过一组 LYNX PRIME 内置控件将插件位置、姿态、缩放等配置参数以页面的形式显示在 LYNX PRIME 程序中；XML SCHEMA 文件用于连接 GCF 文件和插件可执行程序和库，将图形界面下的参数调整行为转化成对应的插件 API 调用，以便在预览时可以直观的观察到调整的效果；插件的可执行程序和库采用编程实现插件的功能并提供 API 接口，其功能实现主要分成两部分。

4.1. 模型的加载和绘制

模型的加载和绘制在 AnimMD5 类中实现，为实现模型文件读取、解析加载、骨骼计算、蒙皮计算的相关功能，需要首先与 MD5 蒙皮动画中的概念相对应，分别定义 Vertex、Triangle、Weight、Joint、Frame、Hierarchy 等结构体，分别用于描述顶点、三角面片、权重、关节、帧(包含基础帧和关键帧)，节点继承关系等，然后定义 Mesh 类和 Anim 类分别表示 MD5MESH 文件和 MD5ANIM 文件的内容，最后在 loadMesh 和 loadAnim 函数中实现文件的加载功能，其过程为：逐行读取文件，根据文件格式解析内容并构造 Mesh 对象或者 Anim 对象，过程中出现错误则加载失败。成功则依次进行关节全局矩阵的计算和蒙皮计算。

蒙皮动画的更新、渲染在 updateAndDraw 中实现，其流程如图 5 所示。

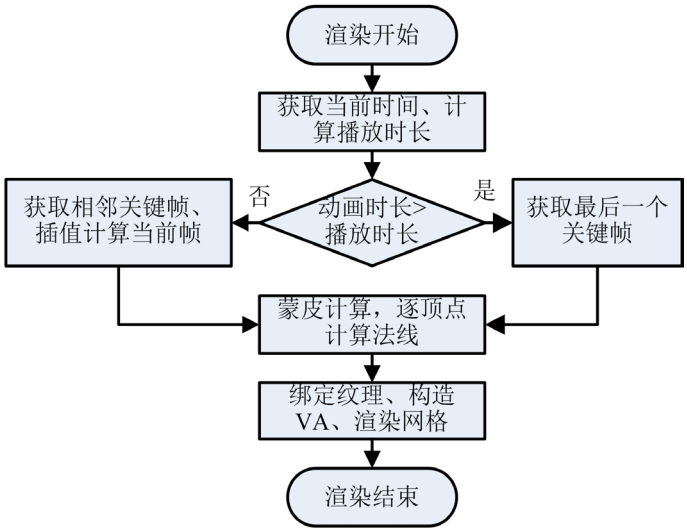


Figure 5. Updating and rendering process of animation
图 5. 动画的更新和渲染流程

1) 在播放开始时记录动画开始播放的时刻，然后在每一帧的更新渲染时获取当前时刻，并且计算动画的播放时长。

2) 比较播放时长和动画时长(即帧率 x 帧数−1)判断动画是否结束，如果已经结束则获取最后一个关键帧。

3) 如果没有结束，则根据帧率、播放时长找到两个相邻的关键帧，然后依次对关键帧各个关节的位置进行线性插值，选择进行球面线性插值，得到当前帧。

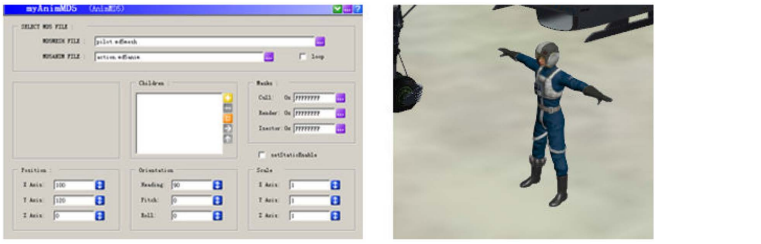
4) 利用步骤 2 或 3 获取的帧信息进行蒙皮计算各个网格的顶点，然后逐顶点计算法线。

5) 绑定纹理，构造顶点数组，使用 OpenGL 渲染当前时刻的模型网格。

4.2. 集成插件设计实现

按照 VEGA PRIME 插件规范，首先必须从 vpModule 派生一个类 vpGearModule 负责插件的初始化、关闭清理等；然后从 vpTransform 派生子类 vpAnimMD5，其实例将作为节点加入到场景图中，并具备调整模型位置、姿态、缩放等功能；vpAnimMD5Geom (从 vsGeometryBase 派生)表示几何体，其实例将作为子节点插入到 vpAnimMD5 中，同时模型的视景裁切在其 cull 函数中实现，过程为：首先获取模型当前时刻的包围盒，然后根据包围盒进行视景裁切计算，如果模型处于当前摄像机的可见范围内，则将对应的 vpAnimMD5GCR 实例加入到筛选结果。动画的更新和渲染在 vpAnimMD5GCR 的 processDraw 函数中完成，在其中调用 AnimMD5: updateAndDraw 即可，代码如下：

```
void vpAnimMD5GCR::processDraw(vrDrawContext *pC)
{
  ::glPushAttrib( GL_ALL_ATTRIB_BITS );
  double nCurTime =
vpKernel::instance()->getSimulationTime();
  _pAnimMD5->updateAndDraw( nCurTime );
  ::glPopAttrib();
}
```



(a) GCF界面 (b) bindpose状态

(c) 行走(0秒) (d) 行走(0.2秒) (e) 行走(0.4秒)

Figure 6. Animation models rendered by plug-ins in VEGA PRIME
图 6. 在 VEGA PRIME 中使用插件渲染的动画模型

5. 结束语

在深入理解 MD5 三维动画模型文件格式和动画原理的前提下，通过 C++编程基于 VEGA PRIME 设计实现了 MD5 三维动画插件，基于该插件的三维动画如图 6 所示，其中图 6(a)是插件 GCF 文件对应的配置界面，图 6(b)是 bindpose 状态下飞行员的模型，图 6(c)~(e)分别是行走状态下飞行员各个时刻的渲染效果。插件具备在仿真程序中控制动画的加载、播放，以及位置、姿态和缩放调整等功能，并且渲染效率能够稳定维持在 60 赫兹以上。实践证明能够满足虚拟角色运动展示的要求。

参考文献

[1] 刘贤梅, 黄静, 刘晓明. 三维动画技术与三维虚拟技术研究[J]. 计算机仿真, 2004, 21(9): 127-130.

[2] 宋子阳. 基于 VEGA PRIME 的飞行器飞行视景仿真平台设计与实现[D]: [硕士学位论文]. 哈尔滨: 哈尔滨工程大学, 2015.

[3] 孔令德, 宋云. 三维建模与动画基础[M]. 北京: 清华大学出版社, 2012.

[4] NXTGO. 骨骼蒙皮动画的实现[EB/OL]. https://www.csdn.net, 2013.

[5] van Oosten, J. (2011) Loading And Animating MD5 Models with OpenGL. https://www.3dgep.com

[6] 袁会杰. 骨骼动画技术的研究与实现[D]: [硕士学位论文]. 成都: 电子科技大学, 2011.

[7] PRESAGIS (2011) VEGA PRIME Programmers Guide. https://www.presagis.com

知网检索的两种方式：

1. 打开知网页面 http://kns.cnki.net/kns/brief/result.aspx?dbPrefix=WWJD
   下拉列表框选择：[ISSN]，输入期刊 ISSN：2324-8696，即可查询
2. 打开知网首页 http://cnki.net/
   左侧“国际文献总库”进入，输入文章标题，即可查询

投稿请点击：http://www.hanspub.org/Submission.aspx
期刊邮箱：mos@hanspub.org