

一种3维动画中间帧非线性插值算法

关东东¹⁾ 关华勇¹⁾ 傅颖²⁾

¹⁾(山东大学数学与系统科学学院, 济南 250100) ²⁾(山东省广电网络中心, 济南 250062)

摘要 关键帧动画是3维动画制作的常用模式,中间帧插值又是关键帧动画中最为繁琐的环节。提出了一种针对无骨架3维网格模型的非线性插值算法,用于对网格模型的关键帧姿态进行插补,自动生成中间帧动画序列。该算法首先计算动画角色的网格模型各三角片在相邻关键帧中的仿射变换,并据此生成变形梯度向量,作为3维网格模型的形变信息,这种表示形式体现了变形过程中网格顶点之间的局部互相关关系。随后将各仿射变换分解为旋转成分与拉伸缩放成分,对拉伸缩放成分进行线性插值,而对旋转成分采用四元数插值算法进行非线性插值,合成中间帧姿态的变形梯度向量,并据此计算出中间帧网格模型。当相邻关键帧姿态存在较大差异时,该算法也能快速生成平稳而生动的中间帧动画,可有效减少关键帧的数量,提高动画制作效率。

关键词 3维动画 关键帧 中间帧 四元数

中图法分类号: TP391.41 文献标识码: A 文章编号: 1006-8961(2006)12-1820-07

A Non-linear Interpolation Algorithm for Tweenframes of 3D Animation

GUAN Dong-dong¹⁾, GUAN Hua-yong¹⁾, FU Ying²⁾

¹⁾(School of Mathematics and System Science, ShanDong University, Jinan 250100)

²⁾(Shan Dong Network Center of Radio and Television, Jinan 250062)

Abstract Key-frame animation is the common mode for making 3D animation, and tween-frame interpolation is the most important part. This paper presents a non-linear interpolation algorithm aiming at the non-skeleton 3D mesh mode. This algorithm is used to carry on auto-interpolation to key-frame posture and to form tween-frame animation serial. This algorithm firstly calculates the affine transformation of each triangle of mesh mode in adjacent key-frame and then creates deformation gradient vector. The deformation gradient vector is regarded as deforming information of mesh mode which reveals the local correlation between mesh vertices in deformation. In the following, each affine transformation is decomposed to rotation component and stretch scale component. The latter is made linear interpolation, while the former is made non-linear interpolation using quaternion interpolation algorithm. By this strategy, the deformation gradient vector of tween-frame posture is composed, according to which the tween-frame mesh mode is calculated. This algorithm can make stable and vivid key-frame animation when there is great difference in posture between adjacent key-frame. This algorithm can reduce the number of key-frame and improve the efficiency of animation making.

Keywords 3D animation, key-frame, tweenframe, quaternion

1 引言

关键帧动画是3维动画制作的基本方式,所谓关键帧是指反映动画角色的重要姿态、动作与表情的画面,一般由资深动画师手工制作。单凭关键帧不能产生连续的动画效果,必须在相邻关键帧之间进行插补,生成数量更多的中间帧画面,使动画角色的运动姿态

在关键帧之间平滑过渡,方可形成完整与生动的动画序列。中间帧的制作水平与效率直接影响着动画制作的质量,本文提出了一种针对无骨架3维网格模型的中间帧动画自动插值算法,可方便快捷地生成3维动画角色的中间帧序列。图1所示的人体行走动画序列中,左1姿态与左6姿态为手工制作的关键帧,中间2~5姿态为本文算法给出的中间帧姿态。

现有的3维角色动画制作常采用骨架驱动动画

收稿日期 2006-01-09 改回日期 2006-04-19

第一作者简介:关东东(1975~),男,现在山东大学艺术学院从事计算机动画教学。主要研究方向为计算机图形学、3维动画。E-mail: ddguan2005@yahoo.com.cn

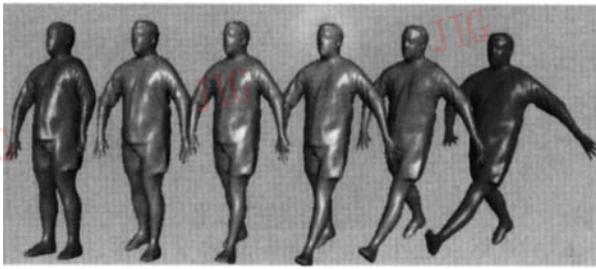


图 1 人体行走动画插值效果

Fig. 1 The interpolation result of walking animation

模式^[1],即首先制作动画角色的刚体骨架,由动画师手工调整骨架中各关节的姿态,形成骨架关键帧,再由动画软件平台自动生成骨架中间帧,最后将表示动画角色皮肤与衣物的 3 维网格模型绑定到骨架上,使网格模型与骨架随动,生成 3 维角色的动画效果。该方法存在的问题是当骨架姿态出现显著变化时,简单的随动常易造成皮肤及衣物的网格模型发生自交、扭曲及断裂的所谓绑定失真现象,必须由动画师反复进行手工修正,影响了动画制作的效率。对刚性物体或由刚体骨骼组成的骨架模型的处理,以采用参数插值方法为多,凡传统的插值方法都可以应用到关键帧参数插值中,如文献 2][3]中提出了一类适于关键帧系统的三次插值样条,把关键帧处的切矢量分为入矢量与出矢量两部分,并引入张量、连续量和偏移量 3 个参数对样条进行控制,由于关键帧插值不考虑物体的物理属性以及参数间的相互关系,其产生的运动不一定是合理的。

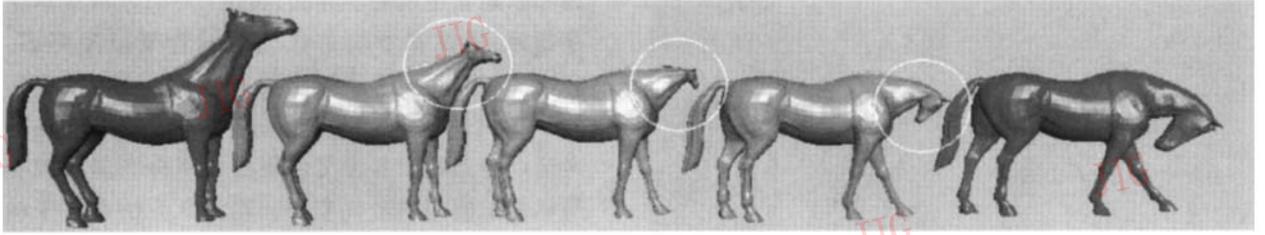
在骨架驱动动画模式中,骨架结构是依照人体或动物身体结构构造的,由关节连接的若干刚性骨骼组成了层次结构。对骨架姿态进行调整以产生动画效果的手段包括运动学学(kinematics)与动力学(dynamics)方法。动力学分为前向运动学(forward kinematics)与逆向运动学学(inverse kinematics),前向运动学方法由动画师指定骨架中各个关节的旋转角度与位移,据此计算动画角色中各节点的空间位置,形成关键帧姿态。逆向运动学只需要动画师指定一个关节位置,即可通过解算器自动求出其上层骨架结构中所有关节的位置,对于简单运动可采用机器人学中的经典方法 Denavit-Hartenberg 解算器求解,为实现更为复杂逼真的运动姿态编辑,目前的逆向运动学研究集中于引入对运动捕捉数据的“学习”机制,以真实运动姿态为参考生成满足动画师指定约束条件的骨架姿态,文献 4]通过骨架姿态的相似度

指标以建立逆运动学算法,文献 5]引入能量消耗最小指标,根据参考姿态优化目标骨架模型姿态,文献 6]构造参考运动姿态间的似然函数,建立具有“学习”机制的逆运动学算法。文献 7]通过构建参考运动姿态间中各关节位置的概率分布函数以确定目标骨架模型的最优姿态。动力学方法是基于动力学方程,通过指定肢体的受力与力矩控制动画角色的运动,文献 8]中提出了一种模拟人体行走的动画混合方法,利用共轭梯度法求解约束方程,获得了较好的效果。基于运动学与动力学的骨架驱动动画虽有了很大的发展,但在皮肤绑定的过程中的绑定失真现象仍是不可避免的。

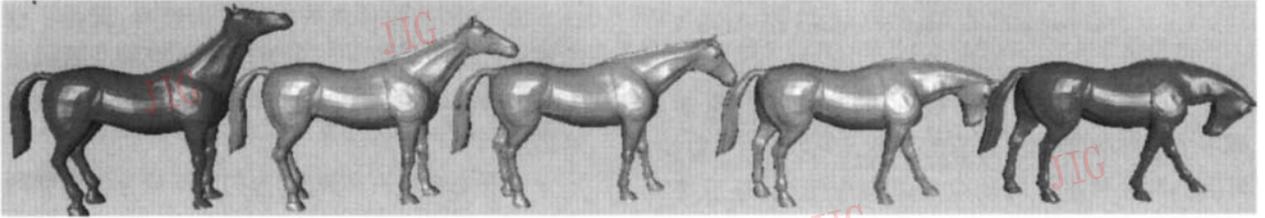
为避免绑定失真现象,人们转而研究无骨架的网格模型的中间帧插值算法,尝试直接对无骨架网格模型的关键帧姿态进行自动插补,生成无绑定失真的动画序列。BlendShape 方法是目前最常用的制作无骨架网格模型中间帧的方法,其原理是直接对网格模型中的顶点位置进行线性插值,但当相邻关键帧中动画角色的姿态相差较大时,线性插值易造成网格模型的畸变,如图 2 第 1 栏所示的马头缩小的畸变现象。因此采用线性插值算法时,动画师必须增加手工制作关键帧的数量,以避免相邻关键帧的姿态相差过大,因而制约了计算机自动生成动画中间帧的效率。

本文提出的中间帧插值算法,是一种针对无骨架网格模型的非线性插值算法,首先分别计算网格模型中各三角片在相邻关键帧中的仿射变换,作为相邻关键帧之间的形变信息,然后,将仿射变换分解为旋转成分与拉伸缩放成分,对拉伸缩放成分仍采用线性插值,而对旋转成分则利用四元数结构进行非线性插值,最后反求出中间帧网格模型的顶点位置。由于该算法引入了基于四元数的非线性插值,当相邻关键帧的动画角色姿态存在较大的旋转扭曲时,也能生成平稳而生动的中间帧动画,并避免了畸变现象,因此采用本文算法可有效降低关键帧的数量,使动画师只需专注于少量反映动画角色关键转折姿态的关键帧即可,有效提高了动画制作的效率。此外该算法还可利用四元数优良的插值特性,对相邻关键帧中的动画角色进行有效的外推,获得生动而夸张的动画效果。图 2 中第 1 栏是 3 维马网格模型利用 BlendShape 算法进行线性插值时出现的畸变现象,而第 2 栏是由本文算法给的中间帧动画插值效果,网格姿态平滑过渡而无畸变发生。

本文主要解决的问题是如何在给定的 3 维模型



(a) BlendShape 算法



(b) 本文算法

图 2 BlendShape 算法与本文算法的插值结果比较
Fig.2 The interpolation results of BlendShape and our algorithm

关键帧之间进行光滑的中间帧插值,而 3 维模型的关键帧姿态是通过 3 维网格模型的空间变形(mesh deformation)算法进行设计的,3 维网格模型空间变形算法是目前动画研究的热点,文献 [9]中提出了利用 Laplace 坐标表示网格模型的局部几何形状,将网格空间变形问题转换为含约束条件的线性最优化问题,文献 [10],[11]针对 Laplace 坐标表示对旋

转变敏感的缺点进行了优化改进,取得了良好的效果。通过各种网格变形算法,动画师可直接对已有的无骨架的角色网格模型进行姿态调整,生成预想的关键帧(图 3)。本文提出的基于无骨架网格模型的中间帧动画插值算法与网格变形算法相结合,形成了一种较完整的方便的 3 维关键帧动画的制作手段,是对骨架驱动动画的一种有力补充。

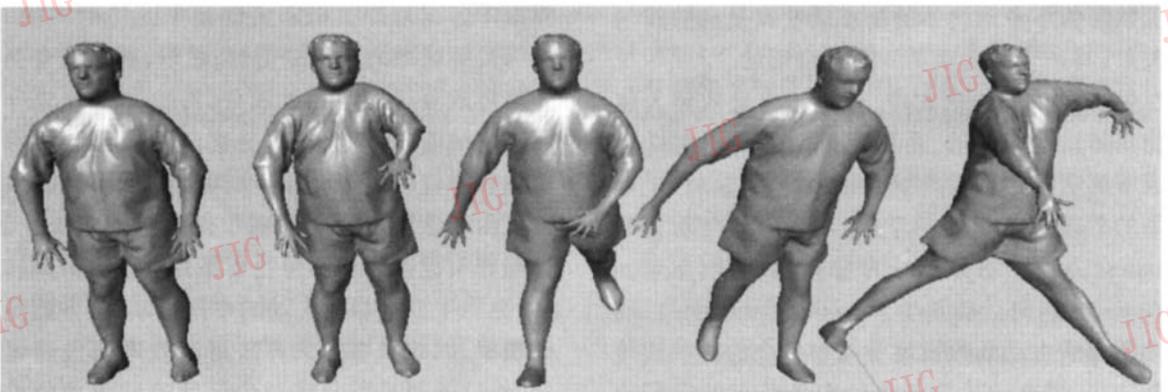


图 3 利用网格空间变形算法对无骨架网格模型进行姿态调整生成关键帧
Fig.3 Key frame acquired by mesh deformation

2 四元数及四元数插值

R^3 空间中的旋转可以用旋转矩阵、欧拉角及四元数 3 种形式表示^[12],其中旋转矩阵的元素含有非

线性的正/余弦成分,不适于进行插值运算。欧拉角表示可能导致空间自由度的丧失,即 gimbal lock 现象,而且等量的欧拉角变化并不一定对应等量的旋转。而四元数模型由于对空间旋转具有优越的插值性质,被广泛用于航空器与机器人的位姿控制。

2.1 四元数的定义

定义四元数为 $q = (w \ x \ y \ z) = (w, V)$ 其中 $V = (x \ y \ z) \in \mathbf{R}^3$, 定义四元数的逆为 $Q^{-1} = (w, -V) = (w, -x, -y, -z)$ 。定义四元数的乘法为

$$q_0 q_1 = (w_0, V_0) \chi (w_1, V_1) = (w_0 w_1 - V_0 \cdot V_1, (w_0 V_1 - w_1 V_0 + V_0 \times V_1))$$

四元数的乘法只满足结合律, 而不满足交换率。当 $w^2 + x^2 + y^2 + z^2 = 1$ 时, 称 q 为单位四元数。两个单位四元数的乘积仍是单位四元数, 所有单位四元数在乘法运算的定义下构成的不可交换群记为 S^3 , 是 \mathbf{R}^4 空间中的单位超球面。

一个单位四元数 q 对应唯一的旋转矩阵 R , 二者之间的转换公式可参见文献 [12] 对 \mathbf{R}^3 中的向量 p 的四元数旋转公式为

$$Rp = q(0 \ p)q^{-1} \quad p \in \mathbf{R}^3 \quad (1)$$

2.2 log 映射与 exp 映射

为方便表示四元数插值算法, 现引入由 S^3 到 \mathbf{R}^3 的 log 映射及由 \mathbf{R}^3 到 S^3 的 exp 映射^[13]。

已知 $q = (w \ x \ y \ z) = (w, V)$, 则四元数旋转公式(1)等效为将向量 $p \in \mathbf{R}^3$ 绕单位旋转轴 $\hat{V} \in \mathbf{R}^3$ 旋转 θ 角, 其中 $\theta = 2 \arccos(w)$, $\hat{V} = V / \sin(\arccos(w))$ 。据此定义由 S^3 到 \mathbf{R}^3 的 log 映射:

$$\log(q) = \theta \hat{V} \in \mathbf{R}^3 \quad (2)$$

当已知单位旋转轴 $\hat{V} \in \mathbf{R}^3$ 与旋转角 θ 时, 也可通过由 \mathbf{R}^3 到 S^3 的 log 映射得到对应的四元数:

$$\exp(\theta \hat{V}) = \left(\cos\left(\frac{\theta}{2}\right), \sin\left(\frac{\theta}{2}\right) \hat{V} \right) = (w, V) = q \in S^3 \quad (3)$$

由式(3)可知, 当旋转角度 $\theta = 0$ 时对应的四元数为 $q = (1 \ 0 \ 0 \ 0)$, 而对应的旋转矩阵为 3×3 单位矩阵。

2.3 四元数的球面线性插值算法

设 $q_0, q_1 \in S^3$ 是两个单位四元数, q_0 与 q_1 之间的球面线性插值 (slerp) 曲线为^[13]

$$q(t) = q_0 (q_0^{-1} q_1)^t \quad t \in [0, 1]$$

$q(t)$ 满足 $q(0) = q_0$ 及 $q(1) = q_1$, 并且以恒定的角速度均匀地由 q_0 变化至 q_1 。 $q(t)$ 是 \mathbf{R}^4 空间中单位超球面上由 q_0 至 q_1 的测地线, 即连接两点的大圆弧。

具体可利用 log 映射与 exp 映射方便地计算四元数球面线性插值曲线:

$$q(t) = q_0 (q_0^{-1} q_1)^t = q_0 \exp(t \cdot \log(q_0^{-1} q_1)) \quad (4)$$

3 非线性中间帧自动生成算法

由于 3 维边形网格模型可转换为 3 维三角形网

格, 不失一般性, 可取关键帧 P_0 和终止关键帧 P_1 皆为三角网格模型, P_0 与 P_1 具有相同的顶点数目及拓扑连接关系, 中间帧是介于 P_0 与 P_1 之间自动生成一系列平滑的过渡网格模型 $P(t), t \in [0, 1]$, 使得 $P(0) = P_0, P(1) = P_1$ 。为产生相邻两幅动画关键帧间的平滑过渡的中间帧, 有两个要点, 一是如何正确提取出能反映或抽象出动画关键帧间变化的本质性信息, 即能建立描绘变形的一个良好模型, 二是如何对本质性信息进行合理的插值或者说均分, 使得到的中间形式能导致生成自然而无畸变的中间帧网格。反映形变本质性信息的模型存在多种的选择, 每种都体现了对形变机制的不同理解和解释, 但不同模型的效果却截然不同。例如直接利用相邻关键帧中对应顶点的位置表示变形信息, 似乎是最合理的也是最直观的一种精确描绘, 但它却丝毫没有反映出在变形中各顶点间的互相关关系, 因而对其进行线性插值的过程中, 导致中间帧的严重畸变也并不为怪(图 2(a))。为此本文采用文献 [14] 中提出的变形梯度向量 (deformation gradient vector) 模型, 变形梯度向量描述了网格模型中各三角片在相邻关键帧之间的仿射变换关系, 该模型同时考虑了网格顶点及其邻近顶点的变形信息, 体现了变形中的局部互相关关系, 是对变形信息的一种本质性描述, 利用矩阵极化分解算法将各三角片的变换矩阵分解为非线性的旋转成分及现行的拉伸缩放成分, 并分别进行插值, 可生成光滑合理的中间帧动画效果。

3.1 提取相邻关键帧的形变信息 f

设动画角色的三角网格模型中含有 n 个顶点及 m 个三角片, 记网格模型中第 j 个三角片的由起始姿态到终止姿态所发生的仿射变换为 Φ_j , Φ_j 由平移向量 d_j 与 3×3 矩阵 T_j 组成, T_j 包含仿射变换中的旋转、放缩及错切成分, 表示了仿射变换的形变信息。可利用文献 [14] 中的方法, 根据各三角片在始姿态与终止姿态的顶点位置计算出矩阵 T_j 。矩阵序列 $\{T_1, T_2, \dots, T_m\}$ 表示了网格模型中各三角片由起始中间帧到终止中间帧的形变信息。可将形变信息 $\{T_1, T_2, \dots, T_m\}$ 中各矩阵 T_j 中的元素分别按列拉伸, 排成 $9m \times 1$ 向量 f , 称之为变形梯度向量, m 为网格中三角片的数目。形变信息的梯度向量 f 可以表示为终止姿态网格模型的网格顶点 X 位置的线性组合^[14]:

$$f = GX \quad (5)$$

其中 $X = (x_1, \dots, x_p, y_1, \dots, y_p, z_1, \dots, z_p)^T$ 为 $3p \times 1$ 向量, 表示终止姿态网格模型的顶点位置。 $9m \times 3p$

变换矩阵 G 完全由起始姿态网格模型的顶点位置所决定。其中 $p = m + n$, n 是网格顶点总数。矩阵 G 的每行中只含有 4 个非零项, 是一个稀疏矩阵^[14]。由于网格各三角片的相互缝合的关系使一个顶点为多个三角形共享, 顶点序号必然重复, 故致使变换矩阵 G 成为高矩阵, 故式 (5) 是超定方程组, 通过它求解中间帧 X 时, 是求非相容解, 一般借助最小二乘问题方法求解:

$$x(t) = \min_x \|Gx(t) - f(t)\|^2 \quad (6)$$

由于系数矩阵 G 是稀疏矩阵, 故在求解式 (6) 时可采用文献 [15] 中的反射牛顿迭代算法 (reflective Newton method) 进行最小二乘求解, 可大幅降低运算时间。

3.2 利用非线性插值生成中间帧

一个非奇异矩阵可被唯一表示为一个正交矩阵与一个正定对称矩阵的乘积, 称之为矩阵的极化分解 (polar decomposition)^[16]。利用该方法可将各三角片对应的非奇异变换矩阵 T_j 唯一地分解为 $T_j = R_j S_j$, 其中, 正交矩阵 R_j 表示空间旋转变换, 而正定对称矩阵 S_j 可通过合同变换成为正定对角矩阵, 故可视为含有坐标系变换的空间缩放变换, 称之为拉伸缩放变换。对旋转成分可进行四元数球面插值, 而对拉伸缩放成分可直接进行线性矩阵插值, 利用此插值方法可得到中间帧姿态相对于起始姿态的形变信息 $\{T_1(t), T_2(t), \dots, T_m(t)\}$, 最后根据式 (6) 计算出中间帧网格模型的顶点位置。

对 T_j 进行矩阵极化分解的过程如下: 设 $T_j = R_j S_j$, 由于 R_j 是正交矩阵, 故有 $T_j^T T_j = S_j^T (R_j^T R_j) S_j = S_j^T S_j$ 成立, 对正定矩阵 $T_j^T T_j$ 进行对角化分解可得:

$$S_j^T S_j = T_j^T T_j = U^{-1} \begin{bmatrix} \lambda_1 & & \\ & \lambda_2 & \\ & & \lambda_3 \end{bmatrix} U$$

由此可求出正定对称矩阵 S_j 及旋转矩阵 R_j :

$$S_j = U^{-1} \begin{bmatrix} +\sqrt{\lambda_1} & & \\ & +\sqrt{\lambda_2} & \\ & & +\sqrt{\lambda_3} \end{bmatrix} U$$

$$R_j = T_j S_j^{-1}$$

因此中间帧所对应的局部形变信息可表示为 $\{T_1(t), \dots, T_m(t)\} = \{R_1(t)S_1(t), \dots, R_m(t)S_m(t)\}$, $t \in [0, 1]$ 在 $t=0$ 时, $R_j(0) = S_j(0) = I$, 对应初始姿态, 而 $t=1$ 时 $R_j(1) = R_j$, $S_j(1) = S_j$, 对应终止姿态。在 $0 < t < 1$ 时刻, 中间帧的拉伸缩放矩阵 $S_j(t)$ 可直接通过矩阵线性插值得出:

$$S_j(t) = (1-t)I + tS_j, t \in [0, 1] \quad (7)$$

而中间帧的旋转矩阵 $R_j(t)$ 可利用四元数球面线性插值算法计算, 在 $t=1$ 时刻计算出旋转矩阵 R_j 对应的四元数 q_1 , 而在 $t=0$ 时, 恒等旋转矩阵 $R_j(0)$ 对应的四元数为 $q_0 = (1, 0, 0, 0)$, 则 $0 < t < 1$ 时刻中间帧的旋转矩阵 $R_j(t)$ 对应的四元数 $q(t)$ 可根据四元数球面插值式 (4) 计算:

$$q(t) = q_0 \exp(t \cdot \log(q_0^{-1} q_1)) = \exp(t \cdot \log(q_1)) \quad (8)$$

再将 $q(t)$ 转换为对应的旋转矩阵 $R_j(t)$, 并可得 $T_j(t) = R_j(t)S_j(t)$ 。在根据式 (7)、(8) 计算出 t 时刻的中间帧所对应的局部形变信息 $\{T_1(t), \dots, T_m(t)\}$ 之后, 可根据式 (6) 计算出 t 时刻的中间帧网格模型的顶点位置, 而对于网格模型的整体位移, 可由关键帧中网格模型各顶点的几何平均点的位移计算得出, 并对整体位移进行线性插值, 得出各中间帧网格模型的整体位移, 构造出中间帧网格模型。图 4 ~ 图 8 分别给出了利用本文算法生成的人体、马及人脸网格模型中间帧动画插值结果, 图中左右两端分别为网格关键帧, 中间为自动生成的中间帧。表 1 中给出了在赛扬 1.5G 平台上各网格模型生成

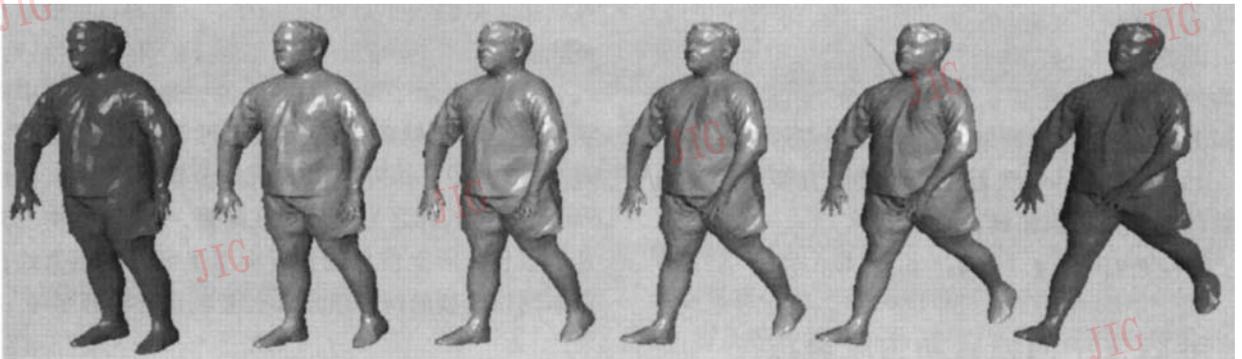


图 4 人体网格模型中间帧动画插值效果(图中左右两端分别为网格关键帧, 中间为自动生成的中间帧)

Fig. 4 The interpolation result acquired by our algorithm

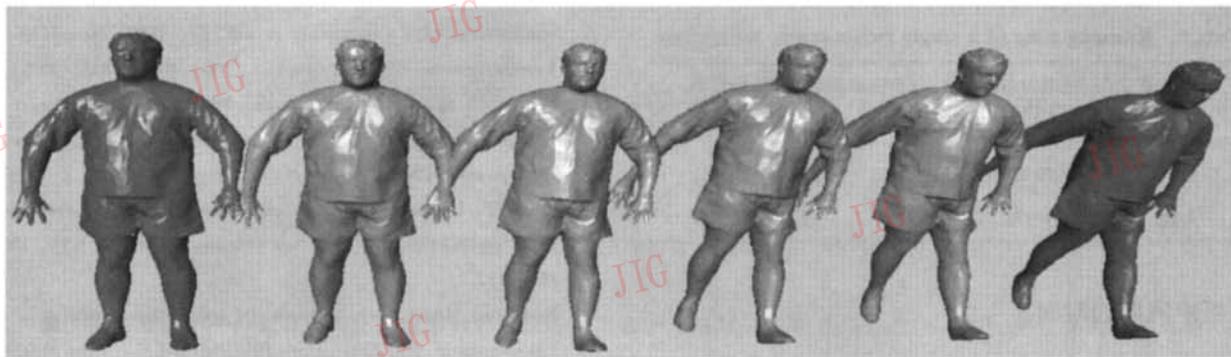


图 5 人体网格模型中间帧动画插值效果(图中左右两端分别为网格关键帧,中间为自动生成的中间帧)

Fig. 5 The interpolation result acquired by our algorithm

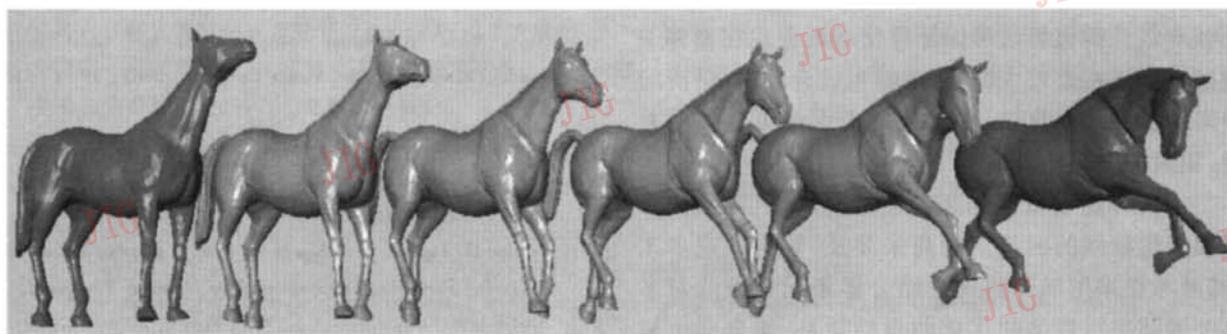


图 6 马网格模型中间帧动画插值效果(图中左右两端分别为网格关键帧,中间为自动生成的中间帧)

Fig. 6 The interpolation result of horse acquired by our algorithm

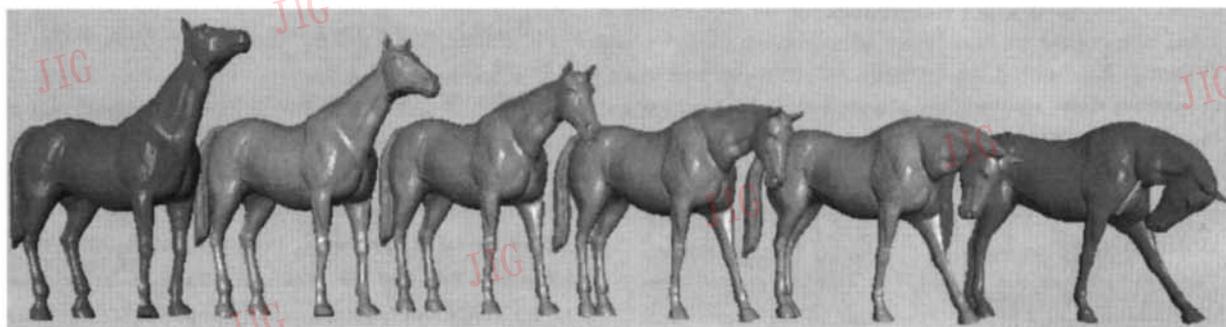


图 7 马网格模型中间帧动画插值效果(图中左右两端分别为网格关键帧,中间为自动生成的中间帧)

Fig. 7 The interpolation result of horse acquired by our algorithm

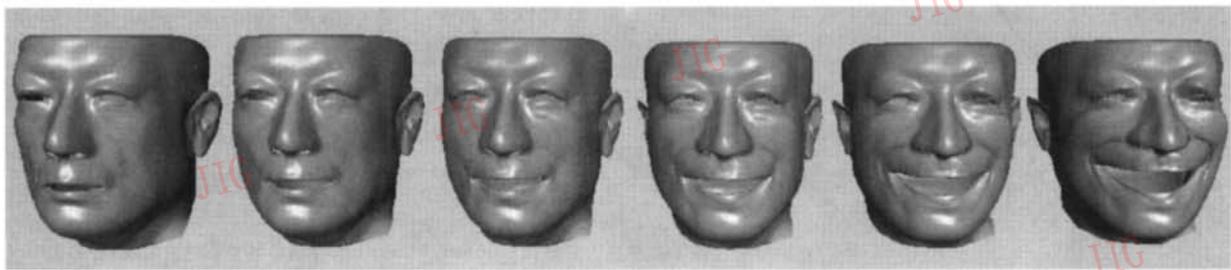


图 8 人脸网格模型中间帧动画插值效果(图中左右两端分别为网格关键帧,中间为自动生成的中间帧)

Fig. 8 The interpolation result of face expression acquired by our algorithm

表 1 网格模型单幅中间帧的生成时间

Tab. 1 Running time of a single tween-frame animation

网格模型	三角片数目	单幅中间帧生成时间(s)
人体	12 004	18.6
马	16 020	24.4
人脸	8 978	12.8

单幅中间帧的时间。

4 结 论

由于引入了四元数非线性插值,本文提出的中间帧非线性插值算法可保证即使动画角色在相邻关键帧中姿态相差较大时,仍能够生成合理的、平滑的中间帧。利用该算法,可有效减少手绘关键帧的数量,提高动画制作效率。

本文讨论仅限于相邻关键帧间的插值问题,对多幅关键帧间的插值问题尚未涉及,作为下一步工作,可考虑采用四元数理论结合样条函数方法对多幅关键帧进行整体的优化的自动插补,使自动生成的动画序列在关键帧处附近的位姿更为光滑,动画的质量进一步提高。

参考文献(References)

- 1 Steketee S N, Badler N I. Parametric key frame interpolation incorporating kinetic adjustment and phrasing control[J]. Computer Graphics, 1985, **19**(3) : 255 ~ 262.
- 2 Kochanek D H U, Bartels R H. Interpolation splines with local tensor continuity and bias control[J]. Computer Graphics, 1984, **18**(3) : 245 ~ 254.
- 3 Yoshizawa S, Belyaev A G, Seidel H P. Free-form skeleton-driven mesh deformations [A]. In : Proceedings of the Eighth ACM Symposium on Solid Modeling and Applications[C], Seattle, USA, 2003 : 247 ~ 253.
- 4 Yamane K, Nakamura Y. Natural motion animation through constraining and deconstraining at will[J]. IEEE Transactions on Visualization and Computer Graphics, 2003, **9**(3) : 352 ~ 360.
- 5 Grassia F S. Believable Automatically Synthesized Motion by Knowledge-Enhanced Motion Transformation [D]. Pittsburgh, Pennsylvania, USA : Carnegie Mellon University, 2000.
- 6 Brand M, Hertzmann A. Style machines[A]. In : Proceedings of ACM SIGGRAPH 2000[C], New Orleans, Louisiana, USA, 2000 : 183 ~ 192.
- 7 Hertzmann Martin Aaron. Style-based inverse kinematics[A]. In : Proceedings of the 2004 ACM SIGGRAPH[C], Los Angeles, California, USA, 2004 : 522 ~ 531.
- 8 Bruderlin Armin, Calvert Tom. Knowledge-driven, interactive animation of human running[A]. In : Proceedings of Graphics Interface 1996[C], Toronto, Ontario, Canada, 1996 : 213 ~ 221.
- 9 Alexa Marc. Differential coordinates for local mesh morphing and deformation[J]. The Visual Computer, 2003, **19**(2) : 105 ~ 114.
- 10 Lipman Y, Levin D. Differential coordinates for interactive mesh editing[A]. In : Proceedings of Shape Modeling International[C], Genova, Italy, 2004 : 181 ~ 190.
- 11 Sorkine O, Lipman Y, Cohen-or D, et al. Laplacian surface editing [A]. In : Proceedings of Symposium on Geometry Processing, ACM SIGGRAPH/Eurographics[C], Nice, France, 2004 : 352 ~ 360.
- 12 Shoemake Ken. Animating rotation with quaternion curves[A]. In : Proceedings of ACM SIGGRAPH 1985[C], New Yorker, USA, 1985 : 245 ~ 254.
- 13 Grassia F S. Practical parameterization of rotations using the exponential map[J]. The Journal of Graphics Tools, 1998, **3**(3) : 29 ~ 48.
- 14 Sumner R W. Deformation transfer for triangle meshes[J]. ACM Transactions on Graphics, 2004, **23**(3) : 399 ~ 405.
- 15 Coleman T F, Li Y. A reflective newton method for minimizing a quadratic function subject to bounds on some of the variables[J]. SIAM Journal on Optimization, 1996, **6**(4) : 1040 ~ 1058.
- 16 Shoemake Ken, Duff Tom. Matrix animation and polar decomposition [A]. In : Proceedings of Graphics Interface 1992[C], Toronto, Ontario Canada, 1992 : 82 ~ 88.