

基于动态规划的关键帧提取算法

费广正¹, 张焰², 韩红雷¹, 黄赞榛²

(¹中国传媒大学动画学院, 北京 100024; ²中国传媒大学计算机与软件学院, 北京 100024)

摘要: 在传统的帧动画中, 动画师对关键帧进行编辑并通过插值生成一段连续的动画。运动捕捉数据的每一帧均可看作关键帧, 但是对其直接进行编辑非常繁琐, 我们需要从运动中提取具有代表性的关键帧, 简化运动编辑操作。已有的关键帧抽取方法主要基于一些局部误差度量策略, 算法的全局误差难以得到保证。本文提出了一种基于动态规划的运动捕捉数据的关键帧提取算法, 该算法通过曲线拟合技术对密集采样的运动捕捉数据进行减帧, 在生成指定数目关键帧的同时保证误差达到全局最小。此外, 用户可以利用传统的关键帧技术对减帧后的运动进行编辑。

关键词: 运动编辑, 关键帧, 动态规划

Dynamic Programming Based Key-frame Extraction Algorithm

Feiguangzheng¹, Zhangyan², Hanhonglei¹, Huangzanzhen²

(¹Animation School, Communication University of China, Beijing, 100024)

(²Computer and Software School, Communication University of China, Beijing, 100024)

Abstract: In traditional key-frame animation, animators edit the key frames to generate a continuous animation sequence by interpolation. However, each frame of motion capture data can be considered as a key frame, and it is too tedious to perform editing task to such data. We need to extract the representative key frames from motion to simplify the motion editing operation. Existing key frame extraction algorithms mainly consider local error metrics, thus do not guarantee a global error tolerance. In this paper, we propose a dynamic programming approach to key reducing for motion capture data, which can reduce key frames of intensive sampled motion capture data through a curve fitting algorithm, and this method promises a minimized global error with a given number of key frames. In addition, users can edit key-reduced motion capture data using traditional technologies of key-frame.

Keyword: Motion editing, Key-frame, Dynamic programming

1 引言

真人动作是游戏和电影等媒体中的重要部分, 而运动捕捉技术是可以获得这些动作的可靠手段。它不但能够捕捉真实的人体运动, 与传统动画制作技术相比, 还具有质量高、代价小、制作速度快、易于实现动作特效、易于重适应等许多独特的优越性, 因此该技术在影视制作、计算机游戏等领域得到了广泛的应用。

然而运动捕捉技术本质上只是对真实世界运动的复制, 不能满足各种应用对运动的需求, 在许多场合下需要对运动捕捉数进行处理, 提高运动捕捉技术的利用率。目前, 人们已经提出了一系列有效的运动编辑与合成方法[1][2][3][4][5][6], 这些方法在运动控制的灵活性和实现运动对环境或角色本身的适应性方面取得了巨大的成功。然

而, 许多动画师仍然习惯于利用传统的关键帧技术对角色运动进行编辑。

虽然运动捕捉的手段很多, 但是运动捕捉的成本通常较高。为了连续地记录表演者的动作, 要求运动捕捉设备有很高的采样速率, 一些光学运动捕捉设备的采样速率甚至可达每秒 1000 帧以上。由于运动捕捉数据缺乏帧间相关性, 运动的每一帧均可看作是一个关键帧, 对如此庞大的数据直接进行存储和关键帧修改显然是不现实的。一种简单的解决方法是对原始数据进行等间隔重采样生成关键帧, 但捕捉的动作节奏或频率通常并不是恒定的, 如一段武术表演, 时而舒缓, 时而奔放。等间隔重采样必然在缓慢处过采样, 造成数据冗余, 而在激烈处欠采样, 从而导致动作细节丢失。

一方面, 在传统的帧动画中, 动画师编辑关键帧并通过插值生成动画; 另一方面, 关键帧提取在基于内容

的视频检索中有着很广泛的应用[12][13]。受这些启发,可从运动捕捉数据中提取一些具有代表性的帧,运动编辑可集中在这些帧上。由于这些方法都试图通过减少运动数据的帧数来达到简化编辑操作的目的,所以,这种方法也称为运动数据减帧。

我们希望在保持原始运动真实性的前提下,对运动捕捉数据进行压缩,并且赋予用户控制角色的能力,这就需要在原始运动中挑选出具有代表性的关键帧。这篇文章提出了一个基于曲线拟合算法的从运动捕捉数据中提取关键帧的动态规划方法,提取出的关键帧不仅从视觉上概括了原始动画的内容,而且通过参数曲线可以重建原始运动。更为重要的是,这是一种全局最优的减帧策略,降低了局部噪声对选择关键帧过程的影响,而且用户可以控制最后生成的关键帧数目。另外,通过反向动力学可以直接对拟合后的运动进行编辑和控制,在关键帧处对动作的修改可以很方便的传递到整个运动序列。

本文将首先介绍运动捕捉数据关键帧提取算法的数学基础——四元数;然后介绍利用曲线拟合的方法来对运动捕捉数据进行减帧,最后介绍如何通过反向运动学和 *Hermite* 插值来实现运动编辑。

2 相关工作

到目前为止,已经有很多关于运动编辑的研究取得了极大的成功。运动变形[4]可以平滑地对原始运动曲线进行细致的修改。运动重定位[2]技术可以将一个角色的运动映射到另一个结构相同但身体各部分比例不同的角色上,在实现运动数据重用的同时保持了原始运动的重要约束,比如避免滑步的产生。运动信号处理[3]将图像和信号处理领域的技术应用于运动捕捉数据,把运动看作采样后的信号,通过多分辨率分析和滤波对运动信号进行处理。这些技术都是直接对运动进行修改,而我们的方法还可以对运动进行减帧。

为了产生一个参数化运动的连续变化空间,运动合成技术可以在一系列的样本之间进行多目标混合。插值合成技术[5]通过使用线性插值创建了一个以不同方向到达不同目标的行走运动。而多维的运动插值通过不同的风格来表达同一个运动,并且使用径向基函数在运动片段之间进行混合。这些研究和我们的方法有类似的地方:都关注于生成一个原始运动数据的参数化表达方式。

Kinetix's Character Studio[14]是一个用来对运动捕捉数据进行减帧,使其更容易进行编辑的商业软件。这个软件有一种高端的减帧算法,可以判定运动中的特定属性,比如步点,并利用这些关键点来得到更加容易编辑的运动数据形式。沈军行等[7]提出了一种关键帧提取方法,提取出的关键帧能够概括运动特征,方便了运动的存储、检索、

传输和编辑,最后利用简单的四元数插值可以重建运动。但是,由于该方法提取运动数据关键帧的过程是基于局部信息的,所以对噪声比较敏感,而且,提取出的关键帧数目完全和运动捕捉数据本身有关,用户无法对其进行控制。

而运动位移映射[3][4]方法中将运动分解为低频和高频部分,本文中的曲线拟合算法相当于运动位移映射中分解出的低频部分,而该方法中用作位移映射的高频部分则类似于拟合曲线与原始曲线的参数化误差。因此,从某种程度上说,本文算法可作为运动位移映射方法中运动信号分解的一种优化算法,亦可运用类似于运动位移映射的方法方便地进行关键帧编辑。同时,关键帧的提取使得我们只关心这些点的信息,极大地压缩了原始运动数据,节省了存储空间。

3 运动捕捉数据表示

人体是一种关节链结构,其状态可用根关节的平移和所有关节的旋转表示,平移一般用三维向量表示。旋转可用旋转矩阵、欧拉角或四元数等形式来表示。由于四元数在插值方面的优越性,目前被广泛应用于计算机动画中。在数学上,我们常把运动数据表示为一个离散时间向量函数:

$$M(t)=[p(t),q_1(t),q_2(t),\dots,q_n(t)],\quad 1\leq t\leq T$$

其中, $p(t)\in R^3$ 表示根关节的平移运动, $q_i(t)\in S^4$, ($i=1,2,\dots,n$) 为一个四元数,表示关节 i 的旋转运动。

3.1 四元数表示旋转

欧拉角是使用最为广泛的表示旋转的方法,但是难以用简单的数学方法表示两个欧拉角之间的运算,同时还会带来万向节死锁问题(Gimbal lock)。一种替代欧拉角表示的最常用的方法是四元数,该方法最早由 Shoemake[8]引入到图形学应用中。

使用四元数最大的优点是:利用球面线性插值(一般称为 SLERP)所得到的结果即有很好的数学理论基础,并且插值的效果也很好。当前,已经有研究者发明了四元数空间构造样条曲线的方法,比如 Kim 和 Shin[10],或者 Johnstone 和 Williams[11]的有理公式。同时,四元数和其他表示旋转表示方法之间可以进行互相转换。

本文中,由于涉及大量的编辑操作和插值运算,利用四元数表示可以得到自然平滑的运动数据,所以我们将关节的旋转都转换为四元数的形式。

3.2 四元数之间距离的计算

首先计算四元数之间的距离。设参考向量 $p_0 \in R^3$, 单位四元数 q_1 将 p_0 旋转到 p_1 (如图 1 所示), 即有:

$$[0, p_1] = q_1 [0, p_0] q_1^{-1}, \text{ 和 } [0, p_2] = q_2 [0, p_0] q_2^{-1}.$$

设单位四元数: $q_2 q_1^{-1} = [w, x, y, z] = [\cos \theta, \sin \theta(a, b, c)]$, 现将其作用于向量 p_1 :

$$\begin{aligned} & (q_2 q_1^{-1}) [0, p_1] (q_2 q_1^{-1})^{-1} \\ &= (q_2 q_1^{-1}) (q_1 [0, p_0] q_1^{-1}) (q_2 q_1^{-1})^{-1} = (q_2 q_1^{-1}) (q_1 [0, p_0] q_1^{-1}) (q_1 q_2^{-1}) \\ &= q_2 (q_1^{-1} q_1) [0, p_0] (q_1^{-1} q_1) q_2^{-1} = q_2 [0, p_0] q_2^{-1} = [0, p_2] \end{aligned}$$

由上式可见, $q_2 q_1^{-1}$ 将 p_1 转到了 p_2 (即 p_1 绕轴 (a, b, c) 旋转 2θ 角到 p_2), 这就说明为了将 q_1 产生的旋转变换到 q_2 产生的旋转须通过 $q_2 q_1^{-1}$ 做一额外旋转。因此 θ 自然就可用来衡量 q_1, q_2 之间的距离, 同时表示 q_1, q_2 所产生的旋转之间的差异(图 1)

定义: q_1, q_2 为两个单位四元数, 若 $q_2 q_1^{-1} = [w, x, y, z]$, 则 q_1, q_2 之间的距离定义为: $Dist(q_1, q_2) = \arccos(w)$

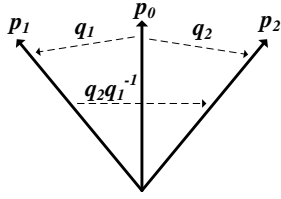


图 1: 四元数之间的距离

下面引入帧间距用来衡量两帧之间人体姿势的差异, 将来被应用于下一章节讨论的误差计算。

定义: 设 $m(t_1), m(t_2)$ 分别为不同时刻 t_1, t_2 上的两帧, 它们之间的距离定义为:

$$Dist(t_1, t_2) = \|m(t_1) - m(t_2)\| = \sum_{i=1}^n \omega_i Dist(R_i(t_1), Ri(t_2))$$

其中 $\omega_i (i=1, 2, \dots, n)$ 为加权系数, 表示各关节对运动影响的重要度, $Dist(R_i(t_1), Ri(t_2))$ 根据四元数之间的距离公式求得。因为人体结构是一种树状的关节链, 父关节的运动 would 带动子关节一起运动, 所以从运动上来讲父关节显得更加重要。

为每个节点赋予不同的权重, 权重说明了该关节在选取关键帧的过程中的重要程度。一般将躯干部分的关节赋以大的权重, 四肢关节比较小的权重。因为躯干部位的变化直接影响了运动的真实程度。

4 运动曲线拟合

事实上, 每一个关节的运动均可看作一条运动曲线, 运动捕获数据则可看成上述运动曲线的集合。因此, 提取关键帧的过程可以看作是对密集样本点选取关键点然后

构造一组特定参数化曲线的过程。使得曲线与原始数据满足某种最小化误差的问题在数学上即为曲线拟合问题, 因此运动数据关键帧抽取问题可以理解为一组曲线拟合问题。

在运动曲线拟合中需要明确以下几个问题: 1. 拟合曲线采用何种参数形式; 2. 如何定义拟合曲线与原始数据的误差; 3. 如何确定参数曲线的控制点。我们将对上述问题分别进行阐述。

4.1 拟合曲线的参数形式

在进行曲线拟合时, 通常需要事先确定一种曲线形式, 然后试图求解曲线的各参数使之尽可能准确地拟合数据。由于参数形式简单以及容易交互的特点, 本文选择用分段三次 *Hermite* 样条曲线来拟合运动数据, 各关键点作为每段 *Hermite* 曲线的控制点。至少首尾两帧是关键帧, 中间每增加一个关键帧, 就增加一段 *Hermite* 曲线。

在计算 *Hermite* 曲线的表达式时, 需要用到两个端点处的切向量, 算法中通过对该点与前后两个相邻点构成的向量求平均近似得到 (如图 2 中的 P 点), 首尾端点处的切向量则分别取与后一点或前一点构成的向量。由于距离较远的点对曲线形状的影响比较小, 因此在实际应用中这种方法是可行的。

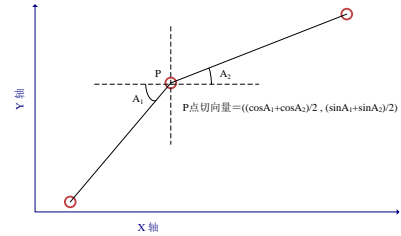


图 2: 数据点处切向量的计算方法

从上述定义中可以看出, 分段 *Hermite* 曲线节点位置 (即关键帧位置) 的选择将直接决定曲线的表达式, 从而影响曲线的最终拟合效果。Plass 等[9]运用动态规划方法来寻找一系列最佳节点 (knot) 使得拟合误差最小, 本文中采用了类似的算法。

4.2 误差度量准则

我们将拟合曲线引起的误差定义为样本点到曲线的距离平方和再加上 τ , τ 是调节参数, 代表多选择一个控制点的代价; 而样本点到曲线的距离可以定义如下: 样本数据点可以表示为方程 $F(P, T)$, 其中 P 表示数据点, T 表示该数据点到起始数据点之间的线段长度在总数据点线段长度中所占的比例; 拟合曲线可以表示为方程 $G(P', T)$, 其中 P' 表示曲线上的数据点, T 表示其对应的参数。设某个样本数据点 p 对应的参数 $T = t$, 则其在拟合曲线上

的对应点是满足 $T=t$ 的点 p' ， p 和 p' 之间的距离就表示 p 到曲线的距离。

假设 $h(t)$ 表示节点 i 和节点 j 决定的一段 *Hermite* 曲线， $g(t)$ 为原始运动曲线（此处仅需考虑其在各采样点处的值），则误差的计算公式为：

$$e_{ij} = \sum_k \|h(t_k) - g(t_k)\|^2 + \tau$$

其中， k 为 i 和 j 之间的任意节点， t_k 为其对应的参数，分别为两条曲线对应的弧长（或弦长）参数， τ 为一正值平衡常数，它的大小决定了拟合曲线逼近原始样本点集的程度： τ 越大，误差越大，拟合曲线控制点越少； τ 越小，误差越小，控制点越多；当 $\tau=0$ 的时候，该算法将会将所有样本数据点作为控制点。对于 τ 的值，可以由程序员经验给出，也可以动态的让用户自己调节。

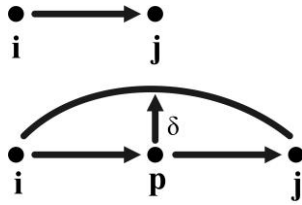


图3：误差 e_{ij} 的计算

我们先考虑只有两个节点的情况，由于 i 和 j 之间没有其他节点（图3的上半部分），显然有 $e_{i2} = \tau$ ；如果在 i 和 j 之间存在另一个节点 p （图3的下半部分），必须计算新节点带来的误差 δ ，即 $h(t)$ 和 $g(t)$ 之间距离的平方。如果 p 引起的误差 δ 小于减少一个节点增加的 τ 值，则 p 被选中为拟合曲线的控制点，反之则丢弃 p 点。

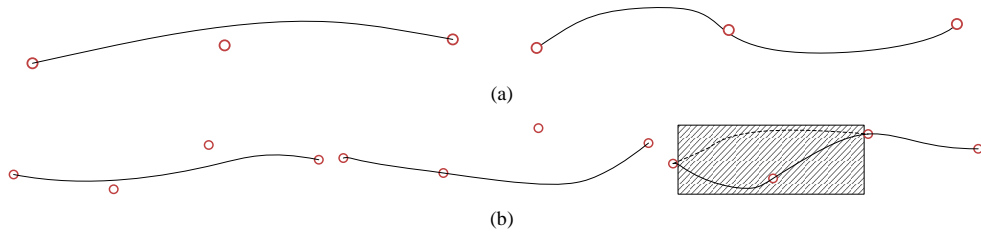


图4：利用动态规划寻找拟合曲线的节点

4.3 拟合曲线控制点选取

在计算误差 e_{ij} 的同时必须为每个节点维护一个从开始点到该节点最小的 e_{ij} ，每次增加一个新节点的时候，都要重新计算之前保存的所有 e_{ij} ，这是一个动态规划的过程，类似于图论中的 Dijkstra 算法。我们假设 E_{ij} 表示节点 i 和节点 j 在可能有其他节点情况下的最小拟合误差，则有：

$$E_{ij} = \min(E_{ik} + e_{kj}), \quad i < k < j$$

显然公式被递归定义，并假设已经将所有 $i < k < j-1$ 的 E_{ik} 计算出来并且予以保存。通过计算之前曲线拟合区域内的所有节点 k 到样本点 j 之间距离的方式，我们可以得到所有的 e_{kj} 值。

那么，计算数据点的最优拟合曲线也就是计算 E_{1j} ($2 < j \leq n$)，并且将对应的 k 依次予以保存。最后，所有 k 就是使得拟合最优的节点，利用这些 k 为控制点的三次 *Hermite* 曲线对原始数据点的拟合，就是误差最小的拟合。

图4显示了曲线控制点选取的过程。图4(a)表示拟合3个数据点选择节点的两种方法；图4(b)表示拟合4个点的时候，可以利用前3个点保存的结果，将比较次数由4次减少为3次。

考虑到运动数据是由多个关节的运动曲线组成，我们将对每一个自由度曲线应用上述的曲线拟合算法。

5 基于关键帧的运动编辑

运用上述算法抽取出关键帧之后，我们可以很方便地计算生成每一个关节的分段三次 *Hermite* 曲线，同时我们也可以很容易地计算出运动数据在每一个原始帧与拟合曲线相应帧之间的参数化误差。记录下上述抽取出的关键帧位置，并将运动中每一帧的参数化误差向量存储到一个列表中，用 *Hermite* 曲线连接这些关键点，便形成了只包含几个关键帧信息的原始运动的新表示。

利用 *Hermite* 曲线易于交互编辑的特性，我们可以对关键帧进行直观的编辑操作，由于 *Hermite* 曲线是自动生成的，编辑的结果可以很容易地传递到运动中的其他帧，并保持较好的效果。这样便提供了一个非常方便和快捷的交互工具，通过传统的关键帧技术对角色运动进行编辑。

在我们的系统中，由于每个关节的运动通过减帧算法得到的关键帧并不相同，可以直接对需要调整的单个关节进行编辑。通过反向动力学在提取的关键帧处对角色的姿态进行修改，用鼠标拖拽角色的关节点（作为末端效应器），骨架根据反向动力学求解方案，自动计算出与之相

邻的父节点的位置，同时修改被传递到所有帧，运动被调整为动画师想要的风格。当然，编辑的前提是必须保持原始运动的重要约束，以免修改后的运动失去物理真实感。

6 实验结果

我们用 VC++ 6.0 和 OpenGL 实现了一个运动数据减帧系统，该系统中对本文提出的算法进行了实现和验证，既保持了原始运动的特征，又一定限度地压缩了数据。

我们选取了一个由 111 帧构成的运动捕捉数据进行演示。由于难以对四元数进行可视化展示，不失一般性，这里我们用欧拉角表示了运动的旋转，并绘制了某关节关于 X, Y, Z 坐标轴的旋转角随时间变化的曲线（图 5）。当用户指定不同的 τ 时，得到了控制点数目不同的拟合方案， τ 的值选择越大，最优参数化曲线的控制点越少，但是，拟合误差也越大。

从实验结果可以看出， $\tau \leq 3$ 时，可以很好地用参数化曲线对原始运动捕捉数据进行拟合，并且在保证信息损失较低的情况下，最少可以将数据压缩到原始运动的 80% 左右。

如果 τ 达到 10 小于 70 的话，拟合效果同样可以接受，这时候甚至可以将帧数压缩到 0.8% 左右。也就是说用很少的帧就可以比较好地表示原始运动，我们的编辑操作可以集中到提取出的这些关键帧上进行。

如果选择 $\tau = 0.5$ ，用 46 个关键帧可以基本上还原原始数据，如图 6 所示；如果选择 $\tau = 10$ ，那么用 17 个关键帧就可以基本上还原原始数据，如图 7 所示。

图 8 是原始运动与本文抽取出关键帧之后拟合得到的新运动之间的比较。图中给出了原始运动与新运动对应的第 1、20、40、60、80 和 100 帧。从图中可以看出，本文的关键帧抽取算法很好地保持了原始运动的姿态。

图 9 为原始运动和风格化编辑后的运动的比较。图中给出了原始运动和编辑后运动的第 1、2、4、10、12 和 18 帧。原始运动是一段跑步的动画序列，我们在第 4 帧处将角色调整为跳跃的姿态，编辑后的运动就变成了一个在奔

跑中跳跃，然后落地继续奔跑的动画序列。注意第 1 帧和最后一帧的姿态是相同的，这也符合 *Hermite* 曲线插值的效果。

7 结论和讨论

本文运用曲线拟合的原理，结合动态规划的节点搜寻策略，在四元数空间对运动捕捉数据进行了曲线拟合。对于曲线拟合而言，选择出的节点是曲线的控制点；对于运动数据而言，选择出的节点是动画的关键帧。通过选中的关键帧可以重建原始运动，从而达到了减帧的目的。该算法可以在保证原始运动质量的前提下，对运动捕捉数据进行关键帧提取，用较少的关键帧来实现对原始数据很好的拟合。

一旦提取到关键帧，动画师就可以利用传统的关键帧编辑对提取出的帧进行编辑，编辑操作可以影响到整个运动序列。由于本文提出的关键帧抽取算法是全局最优抽取方法，相比于局部优化的方法来说，该方法一方面可以让用户控制全局拟合误差，另一方面可以防止噪声对关键帧抽取过程的影响。

现在要对未来的工作有一个总结性的讨论。首先， τ 值是由用户随机指定的，按照某些值拟合后的运动是无意义的，而我们只关心那些有意义的经验值。其次，没有在算法中加入物理和动力学约束。当提取的关键帧数目较少时，结果会显得不太自然，部分关节的运动不符合动力学规律，我们将在系统中加入该约束。最后，拟合后的运动缺乏对环境的适应性。比如一个人走路并跨过箱子的运动片断，在拟合后的运动中角色可能会与箱子发生碰撞，如何保持拟合前后的运动与环境的一致性，也成为我们未来研究的一个重点。

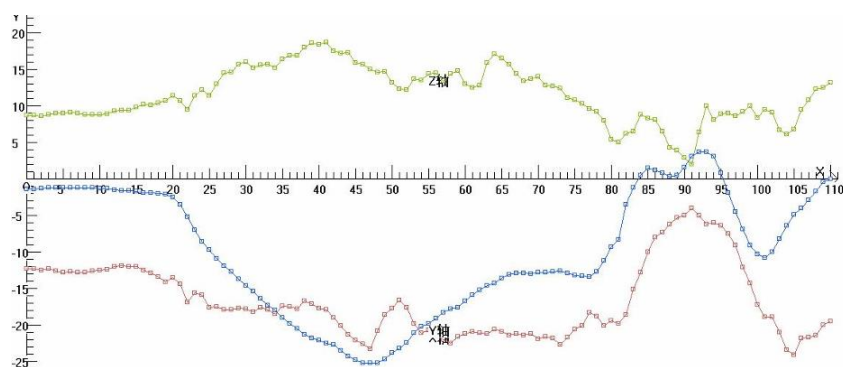


图 5：一段原始运动曲线

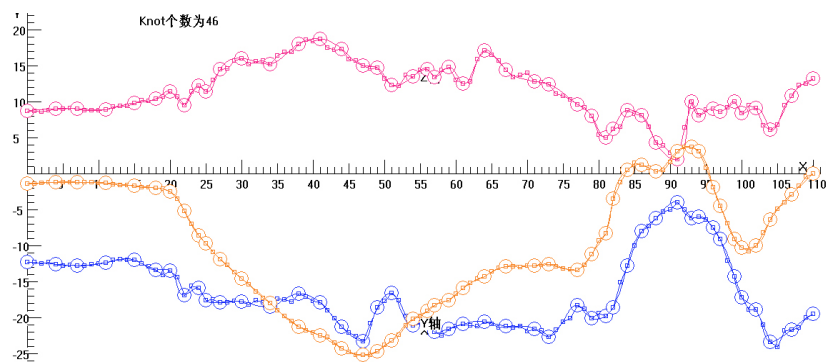


图 6: $\tau = 0.5$ 时的拟合效果

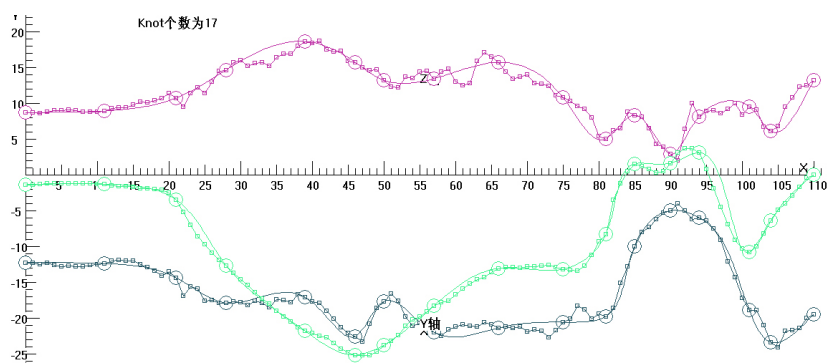


图 7: $\tau = 10$ 时的拟合效果

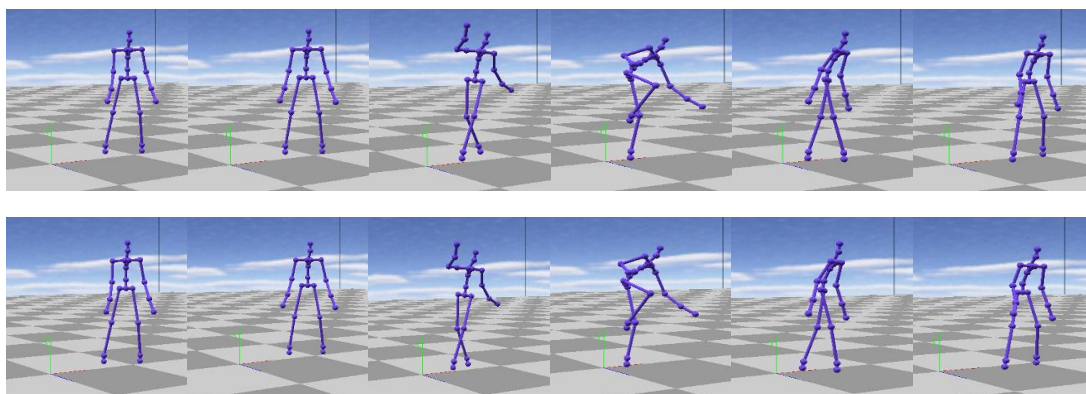


图 8: 原始运动与拟合运动之比较（上图为原始运动帧，下图为对应的拟合运动帧）

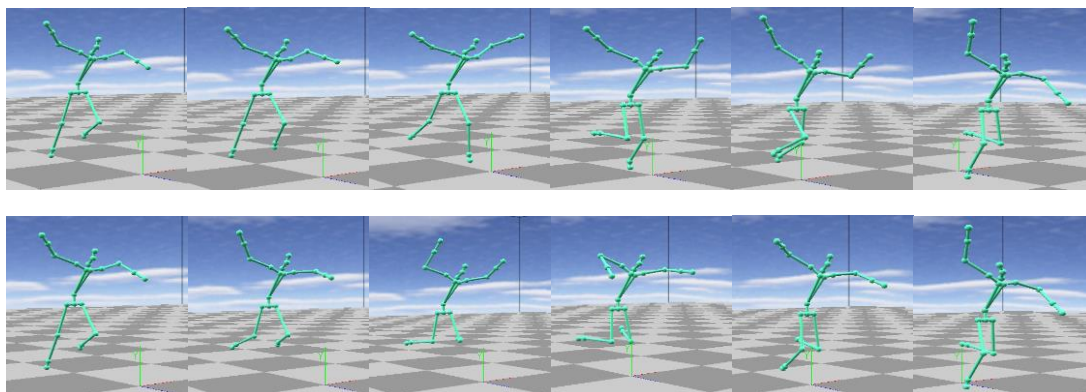


图 9: 原始运动和风格化编辑后的运动

参考文献

- [1] Gleicher M., Motion Editing with Spacetime Constraints, [A]
In: Proceedings of Symposium on Interactive 3D Graphics,
Providence, Rhode Island, 1997, 139~148.
- [2] Gleicher M., Retargeting Motion to New Characters, [A] In :
Computer Graphics Proceedings , Annual Conference Series,
ACM SIGGRAPH, Orlando, Florida, 1998, 33~42.
- [3] Bruderlin A., Williams L., Motion Signal Processing, [A]
In: Computer Graphics Proceedings, Annual Conference Series,
ACM SIGGRAPH, Los Angeles, California, 1996, 97~115.
- [4] Witkin A., Popovic Z., Motion Warping [A] In: Computer
Graphics Proceedings, Annual Conference Series, ACM SIGGRAPH,
Los Angeles, California, 1995, 105~108.
- [5] Arikan O, Forsyth D, O'Brien J. Motion Synthesis from
Annotations, [A]. In: Computer Graphics Proceedings, Annual
Conference Series, ACM SIGGRAPH, San Diego, 2003,
pp. 402~408,
- [6] Li Y, Wang T, Shum H. Motion texture: A Two-level Statistical
Model for Character Motion Synthesis [A]. In: Computer
Graphics Proceedings, Annual Conference Series, ACM SIGGRAPH,
San Antonio, 2002, pp. 465~472
- [7] 沈军行, 孙守迁, 潘云鹤, 从运动捕获数据中提取关键帧, 计算机
辅助设计与图形学学报, Vol116 , No15, pp. 719-723.
- [8] Shoemake K. Animating Rotation with Quaternion Curves
[J]. Computer Graphics, 1985, 19 (3): pp. 245~254.
- [9] Michael Plass, Maureen Stone. Curve-Fitting with Piecewise
Parametric Cubics [J]. Computer Graphics, 1983, 17 (3),
pp. 229~239.
- [10] Myoung-Jun Kim and Sung Yong Shin and Myung-Soo Kim. A
General Construction Scheme for Unit Quaternion Curves With
Simple High Order Derivatives [A]. Proceedings of SIGGRAPH
95, Computer Graphics Proceedings, Annual Conference Series,
pp. 369-376.
- [11] John K. Johnstone and James P. Williams. Rational Control
of Orientation for Animation [A]. Graphics Interface ' 95,
pp. 179-186 (May 1995).
- [12] DeMenthon D, Kobla V, Doermann David. Video summarization
by curve simplification [A]. In: Proceedings of the sixth ACM
international conference on Multimedia, Bristol, 1998,
pp. 211~218.
- [13] Zhao Li, Qi Wei, Li Stan Z., et al. Key-frame extraction
and shot retrieval using nearest feature line (NFL) [A]. In:
Proceedings of the 2000 ACM workshops on Multimedia, Los
Angeles, 2000, pp. 217~220.
- [14] Kinetix Division of Autodesk Inc. Character studio.
Computer Program, 1997.