

DOI:10.3969/j.issn.1003-5060.2023.01.006

面向类人角色动画的骨骼运动数据生成算法

王磊, 李书杰, 谢文军, 贾伟

(合肥工业大学 计算机与信息学院, 安徽 合肥 230601)

摘要:为降低获取与给定类人角色模型相匹配的骨骼运动数据的成本,提高运动数据的可重用性,文章提出一种仅使用较少种类的动作捕捉数据获取多种类角色模型的骨骼运动数据生成算法。该算法首先利用动作捕捉数据训练运动数据自编码器,然后在自编码器生成的隐变量空间中通过迭代约束求解的方式获得多种不同类人角色模型的骨骼运动数据。在 Adobe Mixamo 数据库上的实验表明,该文提出的算法具有较好的表现性能,能够有效生成多种类的骨骼运动数据,可以作为一种扩展骨骼运动数据库的方法,能够为影视、动画和游戏等领域降低动画制作的成本。

关键词:骨骼动画;类人角色动画;骨骼运动数据;运动数据生成;动作捕捉

中图分类号:TP391.414 **文献标志码:**A **文章编号:**1003-5060(2023)01-0036-07

Generation algorithm of skeletal motion data for humanoid character animation

WANG Lei, LI Shujie, XIE Wenjun, JIA Wei

(School of Computer Science and Information Engineering, Hefei University of Technology, Hefei 230601, China)

Abstract: In order to reduce the cost of obtaining skeletal motion data matching a given humanoid character and improve the reusability of the skeletal motion data, this paper proposes an algorithm for generating skeletal motion data using only a few types of motion capture data to obtain multiple types of humanoid characters. The algorithm trains an autoencoder using a small amount of motion capture data, and then obtains skeletal motion data for many different humanoid characters by iterative solving in the latent space generated by autoencoder. The results of experiments on the Adobe Mixamo database show that the algorithm proposed in the paper has better performance and can effectively generate various types of skeletal motion data. It can be used as a method to expand the skeletal motion database and reduce the cost of movies, animations, games and other fields.

Key words: skeletal animation; humanoid character animation; skeletal motion data; motion data generation; motion capture

0 引言

目前,角色动画被广泛应用于虚拟现实、游戏和电影制作中。由骨骼运动数据生成角色动画是目前主流动画制作软件(如 3Dmax、Maya、Blender 等)采用的方式,即用已有骨骼运动数据驱动 3D 角色模型。早期骨骼运动数据的获取方式是

由动画制作师逐帧制作,人力成本和时间成本都较高。随着动作捕捉仪的出现,骨骼运动数据可以通过可穿戴或光学动作捕捉仪获取,但是由于动作捕捉仪穿戴不便,并且只能在特定环境下使用,其只能采集少量表演者的运动数据。然而 3D 类人角色模型丰富多样,具有多种多样的骨骼长度和比例,因此若将动捕数据直接用于驱动不同

收稿日期:2021-12-22;修回日期:2022-02-08

基金项目:国家自然科学基金资助项目(61877016)

作者简介:王磊(1997—),男,安徽砀山人,合肥工业大学硕士生;

李书杰(1983—),女,安徽合肥人,博士,合肥工业大学讲师,硕士生导师,通信作者,E-mail:lisjhfut@hfut.edu.cn.

骨骼长度和比例的角色,则会出现穿地、自相交等不合理现象。

从国内外研究现状来看,使用动作捕捉仪采集表演者 A 的运动数据驱动类人动画角色 B,这一问题的常用解决方案是运动重定向技术,即将表演者 A 的运动数据经过运动重定向算法生成类人动画角色 B 的数据,如图 1 所示。



图 1 动捕采集表演者 A 的运动数据驱动类人动画角色 B

目前,3D 人体骨骼运动重定向方法可以分为 2 种:一种是基于约束的优化求解算法;另一种是基于深度学习的算法。基于约束的优化求解方法的典型代表是逆向运动学方法,即通过角色 A 的末端效应器位置约束角色 B 的末端效应器位置,然后再反向求解角色 B 的其余骨骼节点的坐标。由于约束优化类方法基于人为定义的约束条件求解角色 B 的运动,其中人为定义的约束往往会造成运动失真的现象。随着深度学习的发展,大量研究人员探索用深度神经网络实现运动重定向。文献[1]提出的运动神经网络(neural kinematic networks,NKN)以源角色 A 的运动数据作为输入,输出为目标角色 B 的运动;文献[2]提出的姿态运动网络(pose movement networks,PMnet)在 NKN 的基础上进一步提高了输出运动的精度。

但是,此前的深度学习算法都要求源角色 A 和目标角色 B 均具有大量运动数据作为训练集,实际问题却是角色 A 的运动数据由动作捕捉仪获得,往往仅有少量采集数据。并且,若 B 是一个设计师设计的全新角色,则无相关数据可以用于训练网络。针对这一实际问题,本文提出一种深度学习算法,该算法能够使用少量源角色的数据,生成无任何已知运动数据的目标角色的运动数据。另外,从种类上来说,作为目标角色的类人动画角色种类丰富多样,而源

角色往往只是少数几位在动作捕捉环境下的表演者,因此若想驱动丰富多样的类人动画角色,则需要使用少数几位表演者的数据,生成多种不同的目标角色的数据。

本文提出的算法首先利用少量源角色的动捕数据训练一个运动数据自编码器,然后在该自编码器生成的隐变量空间中通过迭代约束求解的方式获得多种不同类人角色模型的骨骼运动数据。该方法在训练阶段仅需要用到源角色的运动数据,无需目标角色的数据作为训练数据。在执行阶段,只需要通过调整迭代约束求解时的参数,即可获得不同目标角色的运动数据。

综上所述,本文的工作包括以下 3 个方面:

(1) 提出了一种深度学习算法,该算法能够基于单一源角色的运动数据生成目标角色的运动数据,且无需目标角色的运动数据作为训练集。

(2) 提出的算法能够使用单一源角色的运动数据训练网络,该网络训练完成后能生成多种不同目标角色的运动数据。

(3) 在 Adobe Mixamo 数据集上的实验表明,当基于单一源角色运动数据生成未知角色的运动数据时,本文提出的算法比现有运动重定向算法具有更高的精度,即本文的算法更适用于面向类人角色动画的骨骼运动数据生成。

1 相关工作

本文讨论的问题是少量动捕采集的表演者的运动数据驱动丰富多样的类人动画角色。目前该问题常用的解决方案是运动重定向,即将源角色 A 的运动数据经过重定向计算得到目标角色 B 的运动数据。因此,本文首先介绍运动重定向相关技术和方法,再介绍本文算法与现有运动重定向算法的区别与联系。

运动重定向算法是指将源角色运动重定向到与其在骨骼长度比例,甚至是拓扑结构不同的目标角色上,即在保持原始运动特征的前提下,根据目标角色的骨架结构对原始数据进行编辑和修改的过程^[3]。从国内外研究现状来看,针对 3D 骨骼运动数据重定向的方法可以分为 2 种类型,包括传统的基于约束求解的算法和基于深度学习的算法。

1.1 传统约束算法

基于约束的优化求解算法的典型代表是逆向运动学方法,即通过源角色的末端效应器位置约束目标角色的末端效应器位置,然后再反向求解

角色 B 的其余骨骼节点的坐标。但是,逆向运动学方法逐帧求解目标角色的运动,使得求解出来的运动存在一定的抖动现象。文献[3]将运动重定向问题看成时空约束问题,在整个运动序列上进行优化求解,从而减轻了抖动问题;文献[4]首先运用逆向运动学方法逐帧求解重定向后的运动,然后再用 B 样条对运动轨迹进行平滑拟合以去除抖动;文献[5]对每帧进行逆向运动学求解,然后通过最小化相邻帧之间末端效应器差别减少运动的抖动;文献[6]提出一种快速重定向算法可以将一些特定指定行为迁移到任意骨架上,但是其结果仍需要额外的平滑处理。

早期的传统运动重定向算法往往依赖人为设置的约束,如末端效应器约束、平滑性约束等,在约束函数的解空间中按照特定规则选取某个最优解。但是需要指出的是,这个解空间是在数量上满足那些人为设置约束的向量构成的空间,但并不一定满足运动学约束。这一特性使得传统的运动重定向方法输出的数据往往会丢失一些运动学特性,使得运动存在抖动现象,而且即使经过平滑运算后,仍然存在不自然、不真实的现象。

1.2 深度学习算法

随着动捕数据的积累,研究者们也探索利用深度学习算法提升运动重定向算法的性能。文献[7]提出一种基于变分自编码器的运动重定向的网络结构,但是该网络需要用配对训练数据训练网络。所谓配对训练数据,是指在训练时需要源角色和目标角色完全同步并一致的动作序列。但是对于重定向问题而言,最难获取的是配对训练数据。动作捕捉仪显然很难同时获得 2 个不同角色完全一致的动作,而动画师能手工制作出来的配对数据量也非常少。

近年来涌现了许多利用非配对数据的研究。文献[10]受非配对的图像翻译(unpaired image-to-image translation)任务^[8-10]的启发,结合生成对抗网络(generative adversarial nets, GAN)^[11]提出运动重定向网络 NKN,运用 GAN 来解决运动重定向问题,不需要使用配对数据来训练网络。此后,产生了一系列效仿它解决运动重定向的工作。例如,文献[2]提出 PMnet 网络,采用四元数格式的隐变量,使得该网络能够较好地表达运动数据的特征,提升了结果的精度。文献[12]提出一种基于骨骼感知的神经网络(skeleton-aware networks, SAN),该方法通过池化操作提取源角色和目标角色的公共骨架,使得网络不仅能够同一种骨架拓扑结构

下完成重定向任务,而且在不同骨架拓扑结构间也有不错的表现。但是这些方法仅对训练数据集中的角色能够较好地实现重定向,而当训练集中不包括目标角色的数据时,生成的目标角色的骨骼运动数据效果欠佳。

1.3 本文算法与重定向算法的区别

本文算法与现有重定向算法的相同点是算法输入均为源角色的运动序列和目标角色的骨架信息,生成目标角色的运动;不同点在于现有重定向算法在训练时需要目标角色的运动序列作为训练数据,而本文提出的算法在训练时可以没有任何目标角色的运动数据,因此本文算法的意义在于可以生成多种没有原始数据的目标角色的骨骼运动序列,可以作为骨骼运动数据库的增强算法。

为验证本文算法的特点,在实验中本文用仅包含 1 位表演者动捕数据的训练数据集对网络进行训练,并实现了依据该表演者的数据生成多种不同目标骨骼的运动序列,如图 2 所示。

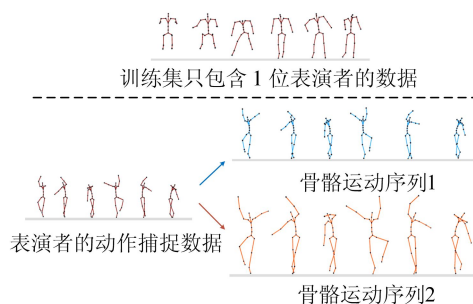


图 2 由 1 位表演者动作捕捉数据生成多种目标骨骼运动序列

2 骨骼运动数据生成算法

2.1 整体思路

本文算法分为训练阶段和运行阶段 2 个阶段。在训练阶段,依据已有的骨骼运动数据训练 1 个运动自编码器,训练时该网络的输入、输出均为训练集中已有的运动数据,训练损失函数为重建设均方误差;在运行阶段,利用已训练好的自编码器,输入表演者的运动序列以及目标骨骼的参数,然后迭代生成所需的骨骼运动序列。本文所采用的网络结构如图 3 所示。

为验证本文所提算法能够仅用少量动捕设备采集的运动数据生成多种不同的骨骼运动数据,本文从 Adobe Mixamo 数据库中仅选取了 1 个角色的运动数据来训练网络,即训练集中仅包含 1 种骨架结构。

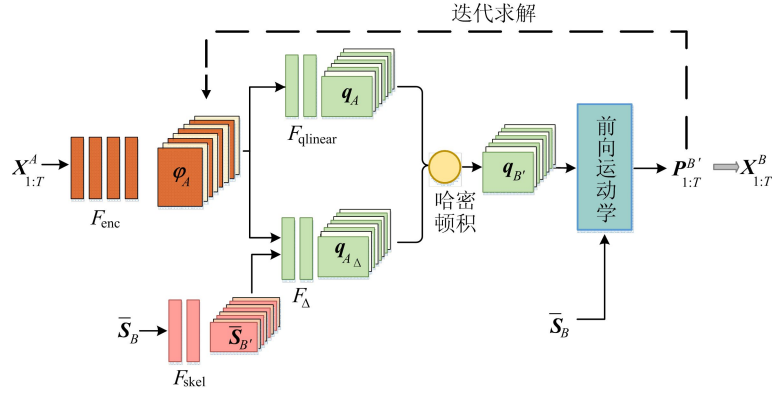


图 3 本文采用的网络结构图

然后用训练好的自编码器生成不同的角色运动序列并与文献[2, 12]的结果进行对比。其中, Adobe Mixamo 数据库中不同角色的数据具有不同的骨骼节点数量和动画帧率, 在训练网络之前需要对运动数据进行预处理, 包括: 将运动数据降采样为 60 帧/s; 保留角色的 22 个骨骼关节, 如图 4 所示; 将训练数据分割为 120 帧的运动片段。

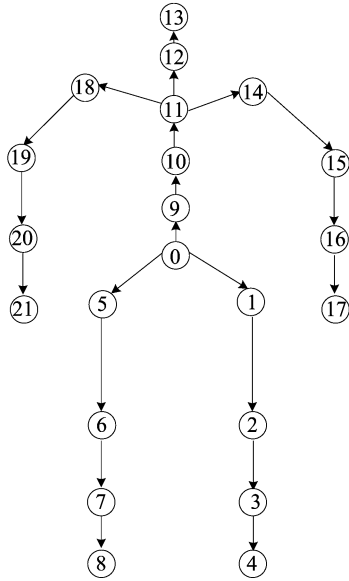


图 4 本文所采用类人角色骨架示意图

2.2 骨骼运动数据生成算法

本文中, 设 $\mathbf{X}_{1:T}^A = [\mathbf{p}_1 \cdots \mathbf{p}_t \cdots \mathbf{p}_T]'$ 为角色 A 的一个骨骼运动序列数据, 其中: $\mathbf{p}_t = [x_{t,1} \ y_{t,1} \ z_{t,1} \ \cdots \ x_{t,J} \ y_{t,J} \ z_{t,J} \ v_t]$ 为序列的第 t 帧, $J=22$ 为关节个数; $v_t \in \mathbf{R}^4$ 为第 t 帧的根关节运动。

首先将运动片段 $\mathbf{X}_{1:T}^A$ 输入编码器 F_{enc} 编码得到隐变量:

$$\varphi_A = F_{enc}(\mathbf{P}_{1:T}^A, \mathbf{W}_{enc}) \quad (1)$$

其中: \mathbf{W}_{enc} 为编码器 F_{enc} 的参数; φ_A 为 $\mathbf{X}_{1:T}^A$ 的隐变

量。得到隐变量后, 利用全连接层 $F_{qlinear}$ 将隐变量 φ_A 做一次正则化:

$$\mathbf{q}_A = \frac{F_{qlinear}(\varphi_A, \mathbf{W}_{qlinear})}{\|F_{qlinear}(\varphi_A, \mathbf{W}_{qlinear})\|_2} \quad (2)$$

在(2)式中, 正则化的作用是让 \mathbf{q}_A 成为单位向量。

然后用全连接层 F_{skel} 来对骨架信息进行编码。令 $\bar{\mathbf{p}}^j$ 表示角色骨架在 T 姿态下的关节坐标, 其根关节坐标为 0。 $\bar{\mathbf{p}}^{\text{parent}(j)}$ 为第 j 个节点的父亲节点, 令:

$$\bar{\mathbf{s}}^j = \bar{\mathbf{p}}^j - \bar{\mathbf{p}}^{\text{parent}(j)} \quad (3)$$

其中, $\bar{\mathbf{S}}_A = [\bar{\mathbf{s}}_A^1 \ \bar{\mathbf{s}}_A^2 \ \cdots \ \bar{\mathbf{s}}_A^J] \in \mathbf{R}^{3 \times J}$ 为角色 A 的骨架信息, 令:

$$\bar{\mathbf{S}}_{A'} = \frac{F_{skel}(\bar{\mathbf{S}}_A, \mathbf{W}_{skel})}{\|F_{skel}(\bar{\mathbf{S}}_A, \mathbf{W}_{skel})\|_2} \quad (4)$$

(4)式也是正则化操作。

其次, 将骨架编码信息和正则化后的隐变量连接起来, 再经过全连接层 F_{Δ} , 得到:

$$\mathbf{q}_{A_{\Delta}} = \frac{F_{\Delta}(\text{Concat}(\varphi_A, \bar{\mathbf{S}}_{A'}), \mathbf{W}_{\Delta})}{\|F_{\Delta}(\text{Concat}(\varphi_A, \bar{\mathbf{S}}_{A'}), \mathbf{W}_{\Delta})\|_2} \quad (5)$$

接着令:

$$\mathbf{q}_A = \text{Hamiton}(\mathbf{q}_A, \mathbf{q}_{A_{\Delta}}) \quad (6)$$

$$\mathbf{p}_{1:T}^{A'} = \text{FK}(\mathbf{q}_A, \bar{\mathbf{S}}_A) \quad (7)$$

其中: Hamiton 操作的本质是 2 个四元数相乘; (6)式在三维空间中可以理解为用旋转信息 $\mathbf{q}_{A_{\Delta}}$ 微调 \mathbf{q}_A ; 前向运动学(forward kinematics, FK)的作用是将旋转信息赋予给定的骨架 $\bar{\mathbf{S}}_A$ 上。

最后, 本文的网络训练损失函数定义为:

$$L_{\text{recon}}(\mathbf{X}_{1:T}^A, \mathbf{X}_{1:T}^{A'}) = \frac{\lambda_{\text{recon}}}{3TJ} \sum_{t=1}^T \sum_{j=1}^J \|\mathbf{p}_t^{j, \mathbf{X}_{1:T}^A} - \mathbf{p}_t^{j, \mathbf{X}_{1:T}^{A'}}\|_2 \quad (8)$$

其中: 本文中的训练系数 $\lambda_{\text{recon}} = 0.5$; $\mathbf{p}_t^{j, \mathbf{X}_{1:T}^A}$ 为运动片段 $\mathbf{X}_{1:T}^A$ 第 t 帧中第 j 个节点的坐标。

该网络在 NVIDIA Titan V 和 Intel Xeon

Silver 41 142. 20 GHz 配置的机器上训练,训练时长为 2~3 h,利用训练完成的自编码器,通过以下步骤根据给定的 $\mathbf{X}_{1:T}^A$ 和目标角色 B 的骨架信息 $\bar{\mathbf{S}}_B$ 生成骨骼运动序列 $\mathbf{X}_{1:T}^B$ 。

首先利用训练好的编码器 F_{enc} 计算 $\mathbf{X}_{1:T}^A$ 对应的隐变量:

$$\boldsymbol{\varphi}_{B'} = \boldsymbol{\varphi}_A = F_{\text{enc}}(\mathbf{X}_{1:T}^A, \mathbf{W}_{\text{enc}}) \quad (9)$$

其中, $\boldsymbol{\varphi}_A$ 为迭代的初始值。

然后,用类似获得 $\mathbf{P}_{1:T}^A$ 的方式,获取 $\mathbf{P}_{1:T}^{B'}$ 的计算过程为(10)~(14)式。二者的区别在于此时输入网络的骨架为目标角色的骨架 $\bar{\mathbf{S}}_{B'}$:

$$\mathbf{q}_{B'} = \frac{F_{\text{qlinear}}(\boldsymbol{\varphi}_{B'}, \mathbf{W}_{\text{qlinear}})}{\|F_{\text{qlinear}}(\boldsymbol{\varphi}_{B'}, \mathbf{W}_{\text{qlinear}})\|_2} \quad (10)$$

$$\bar{\mathbf{S}}_{B'} = \frac{F_{\text{skel}}(\bar{\mathbf{S}}_B, \mathbf{W}_{\text{skel}})}{\|F_{\text{skel}}(\bar{\mathbf{S}}_B, \mathbf{W}_{\text{skel}})\|_2} \quad (11)$$

$$\mathbf{q}_{B_{\Delta}'} = \frac{F_{\Delta}(\text{Concat}(\boldsymbol{\varphi}_{B'}, \bar{\mathbf{S}}_{B'}), \mathbf{W}_{\Delta})}{\|F_{\Delta}(\text{Concat}(\boldsymbol{\varphi}_{B'}, \bar{\mathbf{S}}_{B'}), \mathbf{W}_{\Delta})\|_2} \quad (12)$$

$$\mathbf{q}_{B'} = \text{Hamiton}(\mathbf{q}_{B'}, \mathbf{q}_{B_{\Delta}'}) \quad (13)$$

$$\mathbf{p}_{1:T}^{B'} = \text{FK}(\mathbf{q}_{B'}, \bar{\mathbf{S}}_B) \quad (14)$$

接下来,利用文献[13]中的 Adam 函数,逐步修正隐变量 $\boldsymbol{\varphi}_{B'}$ 使得如下函数小于一定阈值:

$$L_{\text{iter}}(\mathbf{P}_{1:T}^{B'}) = \lambda_{\text{angle}} L_{\text{angle}}(\mathbf{P}_{1:T}^{B'}) + \lambda_{\text{foot}} L_{\text{foot}}(\mathbf{P}_{1:T}^{B'}) \quad (15)$$

其中:迭代系数 $\lambda_{\text{angle}}=0.3$; $\lambda_{\text{foot}}=0.5$; 2 个损失函数分别为角度约束和触地约束。

(1) 角度约束。文献[14]指出,相似的运动之间有相似的关节角。因此,本文通过角度约束让生成的骨骼运动与源运动有相同的关节角度:

$$L_{\text{angle}}(\mathbf{P}_{1:T}^A, \mathbf{P}_{1:T}^{B'}) = \frac{1}{22T} \sum_{t=1}^T \sum_{i=1}^{22} \|\boldsymbol{\theta}_{i^t}^{P^A} - \boldsymbol{\theta}_{i^t}^{P^{B'}}\|_2 \quad (16)$$

其中: $\boldsymbol{\theta}_{i^t}^{P^A}$ 为源运动数据在第 t 帧第 i 个关节角度; $\boldsymbol{\theta}_{i^t}^{P^{B'}}$ 为生成的运动在第 t 帧第 i 个关节角度。

(2) 触地约束。此约束用于避免生成的骨骼运动数据出现穿地的现象。本文通过约束生成运动的左脚、右脚、左脚踝、右脚踝 4 个关节的纵坐标与地面的最小值来避免穿地现象:

$$L_{\text{foot}}(\mathbf{P}_{1:T}^{B'}) = \frac{1}{4T} \sum_{t=1}^T \sum_{i=1}^4 |\min(\mathbf{y}_{i^t}^{P^{B'}}, 0)| \quad (17)$$

迭代完成后,即可认为此时生成的运动 $\mathbf{P}_{1:T}^{B'}$ 为 $\mathbf{X}_{1:T}^B$ 。

3 实验及分析

现有文献中用神经网络生成目标角色的骨骼运动数据的方法主要是通过运动重定向网络,这些运动重定向网络均需要目标角色的运动数据来训练网络。本文算法的主要特色是在没有目标角色数据的情况下生成目标角色的运动。为说明本文算法的有效性,在对比实验中本文从 Adobe Mixamo 数据库中仅选用 1 个角色的运动数据来训练网络,然后分析不同方法在这种情况下生成新角色运动的效果。

在训练阶段,选用 Adobe Mixamo 数据库中的 Malcolm 角色的数据对网络进行训练。在实验过程中,选用 Adobe Mixamo 数据库中 Aj 的运动数据作为输入,产生目标角色 Big_ vegas、Claire 和 Mutant 的数据,如图 5 所示。其中 Aj 的身高为 141.97 cm,而 Big_ vegas、Claire、Mutant 的身高分别为 155.40、139.49、162.75 cm。本文对比的算法包括 PMnet 和 SAN。测试数据共有 200 个不同的运动片段。

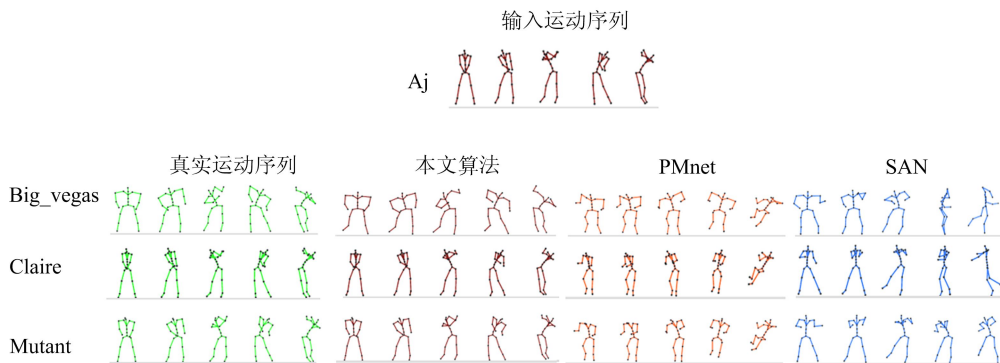


图 5 本文算法与现有算法生成的骨骼运动数据效果对比

根据文献[1]中的方法,本文用均方误差和帧间平滑性 2 个数值指标来衡量生成骨骼运动数据

的效果,见表 1 所列。表 1 中,均方误差是指生成数据关节坐标与真实数据关节坐标之间差值,是

常用的衡量骨骼运动数据差异的方法;帧间平滑性表明了生成运动数据的抖动程度,值越大抖动程度越大。

表 1 不同算法的结果对比

算法	帧间平滑均值	均方误差
SAN	67.80	5.45
PMnet	49.44	2.70
不迭代的本文算法	22.34	1.53
本文算法	15.04	0.90

表 1 中的数据表明,本文算法在没有目标角色的骨骼运动数据情况下,生成的骨骼运动数据效果在均方误差和数据平滑性上均明显优于 PMnet 和 SAN。此外,可以看出,迭代操作对本文的结果有较大的提升。

为了清晰地展示本文算法与其他算法结果的差别,本文对比同一运动序列经过不同算法处理后的结果如图 5 所示。图 5 未提供不迭代操作的结果,主要原因是其不是本文算法的完全体,且不是本文对比的重点算法。由图 5 可知,本文算法生成的骨骼运动数据与真实数据之间具有较高的相似性,而 PMnet 和 SAN 生成的骨骼运动数据均出现了失真的现象。

在训练数据集仅包含一个角色的运动数据的情况下,采用本文的方法可以生成大量不同角色的运动,如图 6 所示。

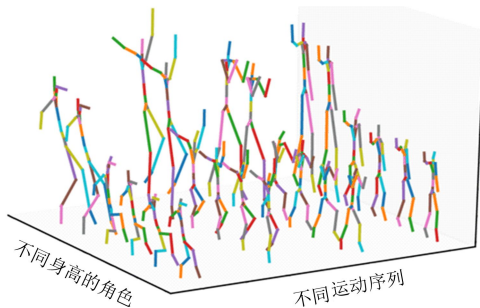


图 6 本文算法生成的骨骼运动数据库

从图 6 可以看出,本文算法可以实现利用 1 个角色的运动数据生成多种不同角色的骨骼运动数据,可以有效实现骨骼运动数据库的数据增强,并且极大地提高了骨骼运动数据的可重用性。

4 结 论

目前,骨骼动画是角色动画软件主要采用的动画制作方式,骨骼运动数据是角色动画制作的重要素材。但是通过动捕设备获得骨骼运动数据

成本高,能获取的数据量少。本文提出了一种骨骼运动数据生成算法,该算法能够基于少量的采集数据生成大量的骨骼运动数据,能够有效对骨骼运动数据库进行增强。

本文实验结果表明本文所提算法的有效性,生成的骨骼运动数据与真实数据具有较高的相似性。当然,本文算法生成的数据必然与真实数据还有一定的差异,但本文算法作为数据增强方法,已经可以较大地减少动画师的手工制作时间。在本文算法的基础上,可以实现交互式的骨骼运动数据生成插件,首先由本文算法生成初始的运动数据,再由动画师做少量修改即可供动画软件使用,可以有效节约动画制作成本。

[参 考 文 献]

- [1] VILLEGAS R, YANG J M, CEYLAN D, et al. Neural kinematic networks for unsupervised motion retargeting[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. [S. l.]: IEEE, 2018: 8639-8648.
- [2] LIM J, CHANG H J, CHOI J Y. Learning of disentangled pose and movement for unsupervised motion retargeting[C]//Proceedings of the British Machine Vision Conference. [S. l. : s. n.], 2019: 1-13.
- [3] GLEICHER M. Retargetting motion to new characters[C]//Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques. [S. l. : s. n.], 1998: 33-42.
- [4] LEE J, SHIN S Y. A hierarchical approach to interactive motion editing for human-like figures[C]//Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques. [S. l.]: ACM, 1999: 39-48.
- [5] CHIO K J, KO H S. Online motion retargeting[J]. The Journal of Visualization and Computer Animation, 2000, 11(5): 223-235.
- [6] FENG A, HUANG Y, XU Y, et al. Automating the transfer of a generic set of behaviors onto a virtual character[C]//International Conference on Motion in Games. [S. l.]: Springer, 2012: 134-145.
- [7] SEONG U K, JANG H, KIM J. A variational U-net for motion retargeting[J]. Computer Animation and Virtual Worlds, 2020, 31(4/5): e1947.
- [8] HUANG X, LIU M Y, BELONGIE S, et al. Multimodal unsupervised image-to-image translation[C]//Proceedings of the European Conference on Computer Vision. [S. l.]: Springer, 2018: 172-189.
- [9] ZHU J Y, PARK T, ISOLA P, et al. Unpaired image-to-image translation using cycle-consistent adversarial networks[C]//Proceedings of the IEEE International Conference on Computer Vision. [S. l.]: IEEE, 2017: 2223-2232.

则 C 是 $G_F(4)$ 上参数为 $[21, 15, 5]$ 的厄米特 LCD 码。

例 3 设 $m=4, \delta=4, C$ 是 $G_F(4)$ 上长为 85 的 ω 常循环码, 其生成多项式为:

$$g(x) = m_1(x)m_7(x)m_{253}(x)m_{241}(x) = x^{16} + x^{15} + \omega x^{14} + x^{12} + \omega x^{11} + \omega x^9 + \omega^2 x^8 + x^7 + x^5 + \omega x^4 + x^2 + \omega x + \omega,$$

则 C 是 $G_F(4)$ 上参数为 $[85, 69, 5]$ 的厄米特 LCD 码。

4 结 论

本文构造了四元厄米特互补对偶常循环码, 并分析它们的参数。确定长度为 $n=(4^m-1)/3$ 的厄米特 LCD 常循环码的维数, 给出了它们最小距离的下界。本文仅分析了长度为 $n=(4^m-1)/3$ 的四元厄米特互补对偶常循环码的参数, 其他长度也有待进一步探讨。

[参 考 文 献]

- [1] 刘春颖. 几类特殊的常循环码[D]. 武汉: 华中师范大学, 2012.
- [2] AYDIN N, SIAP I, RAY-CHAUDHURI D K. The structure of 1-generator quasi-twisted codes and new linear codes[J]. Designs, Codes and Cryptography, 2001, 24(3): 313-326.
- [3] MASSEY J L. Linear codes with complementary duals[J]. Discrete Mathematics, 1992, 106(107): 337-342.
- [4] CARLET C, GUILLEY S. Complementary dual codes for counter-measures to side-channel attacks[C]//The 4th International Castle Meeting Coding Theory and Applications, CIM Series in Mathematical Sciences. [S. l.: s. n.], 2014: 97-105.
- [5] LI C J, DING C S, LI S X. LCD cyclic codes over finite fields[J]. IEEE Transactions on Information Theory, 2017, 63(7): 4344-4356.
- [6] LI S X, LI C J, DING C S, et al. Two families of LCD BCH codes[J]. IEEE Transactions on Information Theory, 2017, 63(9): 5699-5717.
- [7] ZHU S X, PANG B B, SUN Z H. The reversible negacyclic codes over finite fields[J]. Journal of Systems Science and Complexity, 2018, 31(4): 1065-1077.
- [8] GÜNERI C, ÖZKAYA B, SOİÉ P. Quasi-cyclic complementary dual codes[J]. Finite Fields and Their Applications, 2016, 42: 67-80.
- [9] BOONNIYOMA K, JITMAN S. Complementary dual subfield linear codes over finite fields[J]. Thai Journal of Mathematics, 2016(Special Issue ICMSA2015): 133-152.
- [10] CARLET C, MESNAGER S, TANG C, et al. Euclidean and Hermitian LCD MDS codes[J]. Designs Codes and Cryptography, 2018, 86(11): 2605-2618.
- [11] LI C J. Hermitian LCD codes from cyclic codes[J]. Designs Codes and Cryptography, 2018, 86(10): 2261-2278.
- [12] MACWILLIAMS F J, SLOANE N J A. The theory of error-correcting codes[M]. Amsterdam: Publishing Company, 1977.
- [13] ALY S A, KLAPPENECKER A, SARVEPALLI P K. On quantum and classical BCH codes[J]. IEEE Transactions on Information Theory, 2007, 53(3): 1183-1188.
- [10] ABEL G G, WEIJER J, BENGIO Y. Image-to-image translation for cross-domain disentanglement[J]. Advances in Neural Information Processing Systems, 2018, 31: 1287-1298.
- [11] GOODFELLOW I, JEAN P A, MIRZA M, et al. Generative adversarial nets[J]. Advances in Neural Information Processing Systems, 2014, 27: 2672-2680.
- [12] ABERMAN K, LI P, LISCHINSKI D, et al. Skeleton-aware networks for deep motion retargeting[J]. ACM Transactions on Graphics, 2020, 39(4): 62:1-14.
- [13] KINGMA D P, BA J. Adam: a method for stochastic optimization [EB/OL]. [2022-12-14]. <https://arxiv.org/pdf/1412.6980.pdf>.
- [14] TAK S Y, GO H S. Example guided inverse kinematics[J]. Journal of The Korea Computer Graphics Society, 1999, 5(1): 11-17.

(责任编辑 张 镅)

(责任编辑 李 凯)

(上接第 41 页)