****

硕 士 研 究 生 读 书 报 告



题目 习得的运动匹配

作者姓名 赵云路

作者学号 22051182

指导教师 李启雷

学科专业 软件工程

所在学院 软件学院

提交日期 二○二○年十二月

Learned Motion Matching

A Dissertation Submitted to

Zhejiang University

in partial fulfillment of the requirements for

the degree of

Master of Engineering

Major Subject: Software Engineering

Advisor: Qilei Li

By

YunLu Zhao

Zhejiang University, P.R. China

2020

摘要

该论文提出了一种运动匹配算法的替代方法，该方法保留了运动匹配的积极特性，同时还实现了基于神经网络的生成模型的可扩展性。尽管基于神经网络的角色动画生成模型能够从大量的动画数据中学习具有表现力的、紧凑的控制器，但由于其灵活性、可预测性、较低的预处理时间和视觉质量，运动匹配等方法仍然是游戏业的一种流行选择。这些特性有时很难用基于神经网络的方法来实现。然而，不同于神经网络，这类方法的内存使用量随使用的数据量呈线性增加，这导致必须在动画制作的多样性和现实世界的制作预算之间进行权衡。此项工作结合了这两种方法的优点，通过将运动匹配算法分解为单独的步骤，介绍了如何使用习得的、可扩展的替代方法依次替换运动匹配的每个组件。最终模型不需要在内存中存储动画数据或其他的匹配元数据。同时，模拟运动匹配的行为，保留了质量、控制和快速的迭代时间，这在业界是非常重要的。

**关键词**：运动匹配， 生成模型，神经网络， 角色动画， 动画

Abstract

This paper proposed a learned alternative to the Motion Matching algorithm which retains the positive properties of Motion Matching but additionally achieves the scalability of neural-network-based generative models. Although neural-network-based generative models for character animation are capable of learning expressive, compact controllers from vast amounts of animation data, methods such as Motion Matching still remain a popular choice in the games industry due to their flexibility, predictability, low preprocessing time, and visual quality - all properties which can sometimes be difficult to achieve with neural-network-based methods. Yet, unlike neural networks, the memory usage of such methods generally scales linearly with the amount of data used, resulting in a constant trade-off between the diversity of animation which can be produced and real world production budgets. This work combined the benefits of both approaches and, by breaking down the Motion Matching algorithm into its individual steps, showed how learned, scalable alternatives can be used to replace each operation in turn. The final model has no need to store animation data or additional matching meta-data in memory, meaning it scales as well as existing generative models. At the same time, this method preserved the behavior of Motion Matching, retaining the quality, control, and quick iteration time which are so important in the industry.

**Keywords：**motion matching, generative models, neural networks, character animation, animation

1引言

游戏等交互式应用程序，对更大、更具沉浸感和动态性的世界的需求不断增加，使得在需求的场景呈指数级增长的情况下，生成能够真实自然地作出反应的角色变得更加困难。3A游戏现在通常包含数以万计的动画片段，所有这些动画都必须在正确的上下文中触发。

运动匹配（Motion Matching）是一种搜索大型动画数据库的方法，能够根据给定的上下文找到最合适的动画。具有灵活性、可预测性、需要很少的处理时间、视觉质量以及快速迭代等优点。相比于用状态机来指定细粒度的动画逻辑，运动匹配允许动画师指定动画的属性，最合适的匹配会通过最邻近搜索自动完成。当有大量数据的时候，运动匹配提供了一种简单而高效的方法处理3A游戏中需求的大量的动画过渡和交互。运动匹配最主要的限制在于内存使用量（以及某种程度上的运行时性能）会随数据量和需要匹配的特征的数量呈线性增长，这使得开发者不能将其与强大的数据处理方法，比如自动化数据增强（Automatic Data Augmentation）相结合。

同时学术界开始关注一种越来越吸引人的基于神经网络的生成模型，具有很低的内存开销, 数据方面的可扩展性，以及快速的运行时评估时间。最近的一些研究表明基于神经网络的模型可以在一些高难度的情况（比如在崎岖的地面探索[Holden et al. 2017] 四足动物运动[Zhang et al. 2018]）下高效的用于生成真实性动作，并且与环境或者其他角色交互。 但是这些模型很难去控制，行为不可预测，需要很长的训练时间，而且可能生成比原本训练集质量更差的动画。

该论文提出了一种既保留了运动匹配的优点属性同时实现了基于神经网络生成模型的可扩展性的方法。该方法把运动匹配算法分解为独立的阶段，然后用神经网络替代。在动画质量、运行时性能以及内存使用量方面能够得到很好的结果。

**2 相关工作**

**2.1 数据驱动的动画合成**

数据驱动的动画合成方法具有很长历史，包括基于图的方法、线性方法、基于核的方法以及最近基于神经网络的方法。由于动作匹配中有“匹配损失”这一概念，动作匹配可以被视为一种基于图的搜索算法。此外动作图方法还有许多扩展：动作语法、参数化动作图等。

**2.1 动作匹配**

动作匹配第一次提出是在2015年，作为Motion Fields的一种贪心的近似算法由Büttner和Clavet提出。这个动画系统是专门为荣耀战魂这个游戏开发的，后来由Clavet在2016年框架化为对动画数据库的搜索，以找到与当前姿势和用户轨迹最匹配的帧。Zadziuk提出了数据采集和控制的其他方法。后来，Harrower展示了动作匹配可以用于角色的互动，而Holden展示了运动匹配作为使用机器学习进行动画合成的一个特例，其中最近邻回归用于从用户的控制映射到动画数据。

**3 研究内容**

**2.1 基础动作匹配**

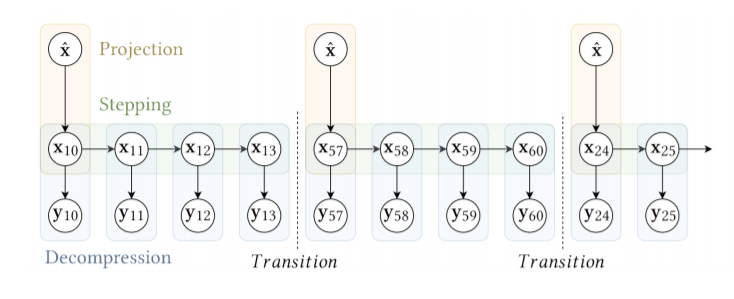


图 1. 基础动作匹配算法概览。

首先，定义一个特征向量 ，该向量代表了每一帧中要去匹配的特征。不同的行为可以通过定义不同的特征向量来实现。在locomotion的场景下，我们将特征向量定义为 ， 为未来（60帧率下）20帧、40帧、60帧的角色信息，其中为轨迹位置， 为轨迹的朝向， 为两只脚的关节的方向（相对于角色的本地坐标系）， 为两只脚的关节的速度（相对于角色的本地坐标系）， 为髋关节速度（相对于角色的本地坐标系）。

然后，再定义一个姿势向量 ，其包含了动画在一帧中所需的所有的姿势信息， ，其中分别表示关节在本地坐标系下的位移和旋转，表示关节在本地坐标系下的位移和旋转的速度，分别表示角色根节点的位移和旋转的速度，为其他额外任务的特定输出，比如脚的接触信息、其他角色在场景中的位置和轨迹。

对于Mocap数据中的每一帧，计算其特征向量和姿势向量，并分别存储到两个大的矩阵中：特征数据库 和动画数据库 。

在运行时，每隔N帧或者当用户输入变化巨大的时候，构造一个查询向量 ，代表了希望得到的特征向量，然后在特征数据库中根据查询向量进行最邻近搜索，使得

一旦找到之后，动画将从第 帧开始播放，姿势向量可以简单的取该帧对应的姿势，然后每一帧从当前帧开始推进一帧。

最邻近搜索可以通过加速结构，比如KD-树、基于体素的查找表或者簇进行加速

最后，在特征数据库上可以使用PCA进行降维，来减少内存使用量以及减少性能开销，同时会造成一些精确度上的损失，但是特征数量很大或者相关的、冗余的特征很多时，这通常是值得的。

**2.2 习得的动作匹配**

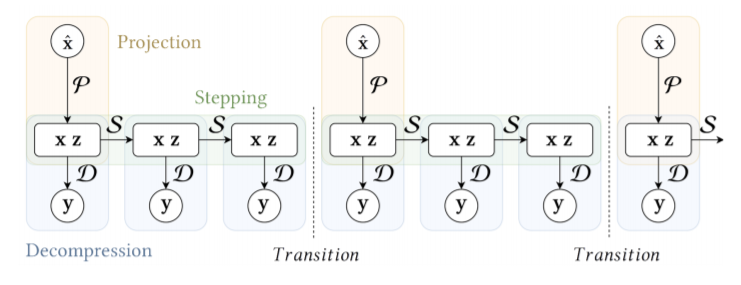


图 2. 习得的运动匹配算法概览。

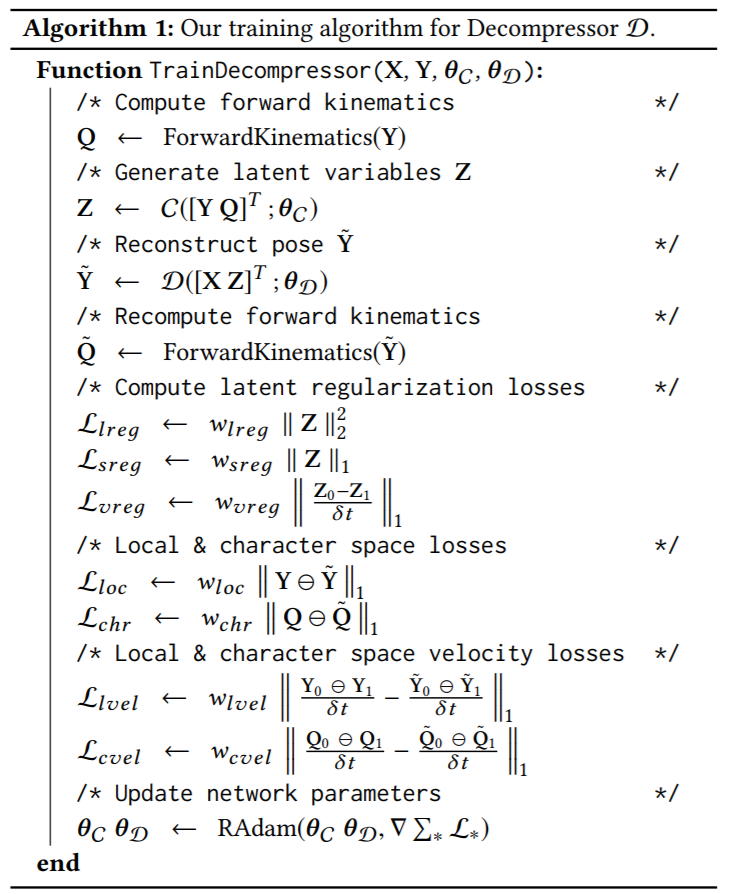
通过观察图2，我们可以发现运动匹配算法由三个关键的阶段组成：Projection：在特征数据库中通过最邻近算法查找与查询向量最匹配的特征向量，Stepping：推进帧索引在特征数据库中增加，Decompression：根据特征数据库中当前帧在动画数据库中查找其对应的姿势。

这三个阶段可以简单的进行每N帧循环一次，然后通过插入混合来消除动作的不连续性。就扩展性而言，这个系统的核心问题是对X和Y的依赖，这会导致大量内存开销，并且内存开销会随着我们添加额外的动画、姿势特征或匹配特征而增长。

习得的运动匹配算法如图2所示。在用户输入上使用Projector网络将其映射到特征数据库中最近的特征向量和相关的隐变量上。然后使用Stepper将这组特征向量和隐变量及时向前移动，然后使用Decompressor生成完整的角色姿态。与基本运动匹配一样，这个过程只需每隔N帧重复一次，然后插入一个混合来消除不连续性。

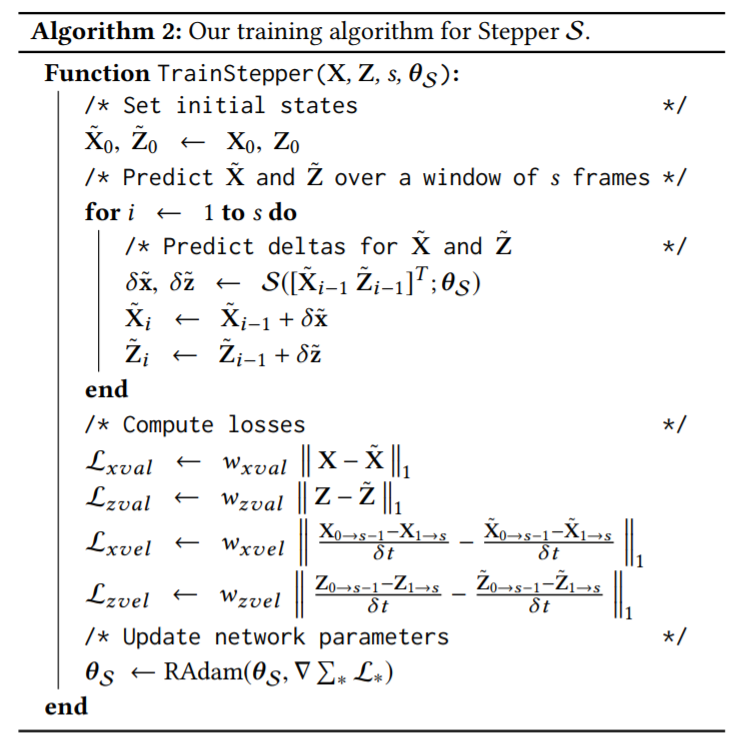
此方法将训练三个神经网络分别称为Projector、Stepper、Decompressor，然后将这三个网络分别替代运动匹配算法中对应的Projection、Stepping、Decompression阶段。本节的其余部分将展示训练这三个神经网络来创建替代的解决方案，以此来消除在内存中存储任何数据库的需要。

Decompressor 的目标是避免在内存中存储动画数据库，根据某一帧的特征向量，直接生成该帧对应的姿势向量，尽管特征向量包含了一些对应姿势的关键信息，比如脚的位置和速度，但通常包含的信息不足以映射到对应姿势的信息。因此，引进一个额外的隐变量，将特征向量缺失的信息编码到中。Decompressor的训练算法如下：



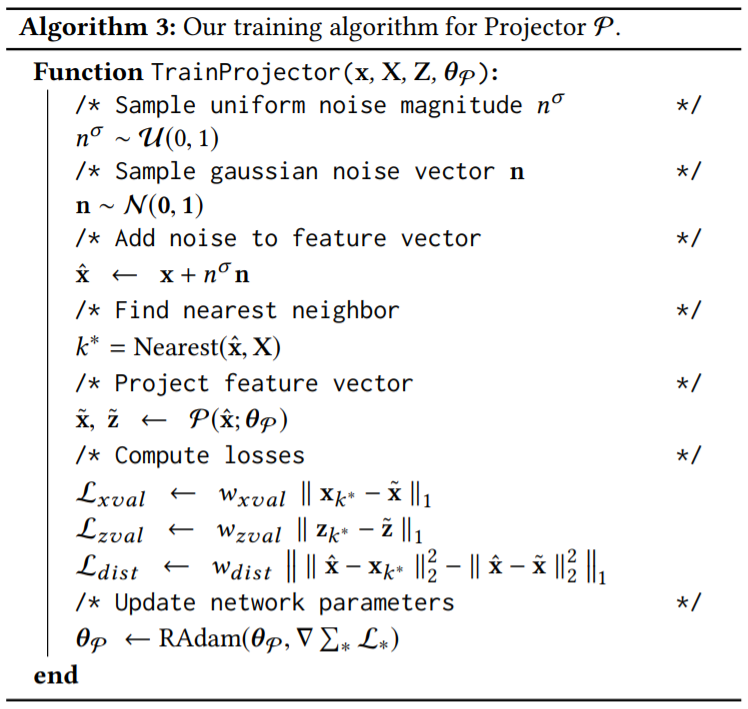
Stepper的任务是根据初始的匹配特征向量和隐特征向量，每一帧推进一个时间步，然后得到下一个连续的。在基础运动匹配算法中，这一步骤非常简单，通过增加帧索引i然后查找对应的X, Z。但是，如果为了避免将X, Z存储在内存中，这一简单的方案将不再可行。

因此，引入Stepper网络，使得该神经网路在训练后能够根据输入，得到。一旦训练完成，该神经网络可以替代运动匹配算法中的相应步骤。Stepper的训练算法如下所示：

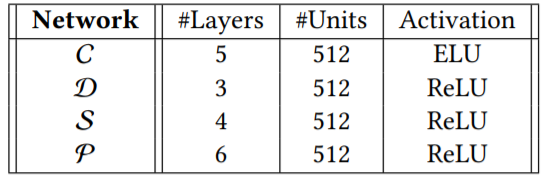


尽管Stepper能够在时间上进行推进从而更新特征向量，但是根据查询向量进行最邻近搜索仍然需要特征数据库。

为了最终不再在内存中存储X和Z，引入了Projector，该神经网络用于模拟最邻近搜索的行为，通过查询向量找到最近的特征向量入口。Projector的训练算法如下：



三个神经网络的结构细节如下：



表格 1. 网络结构细节。

**4 研究结果**

与最近其他的生成模型相比，比如 Phase-Functional Neural Network，LMM动画质量更好并且使用了更少的内存、更快的运行性能。LMM不需要在内存中存储大量的动画数据，几乎能精确模拟 Motion Matching 的行为，同时保留了质量、灵活性、快速的迭代时间。

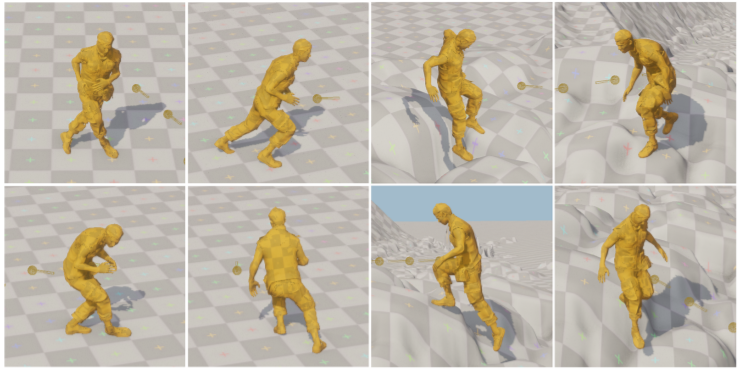


图 3. 该方法运用于经典的角色locomotion场景和穿越崎岖地面。



图 4. 该方法运用于与椅子交互和与其他玩家交互。



图 5. 该方法运用于四足动物运动。



图 6. 该方法与其他基于神经网络的方法的比较。LMM（左），PFNN（右上），MANN（右下）。

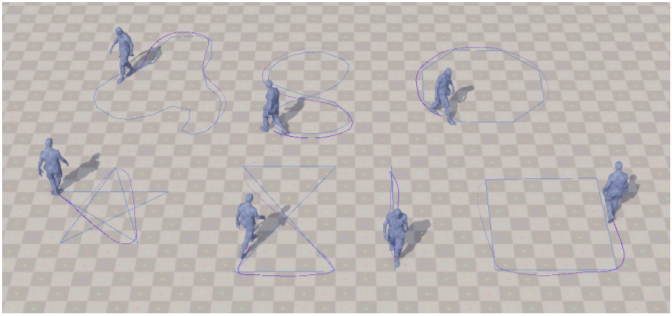
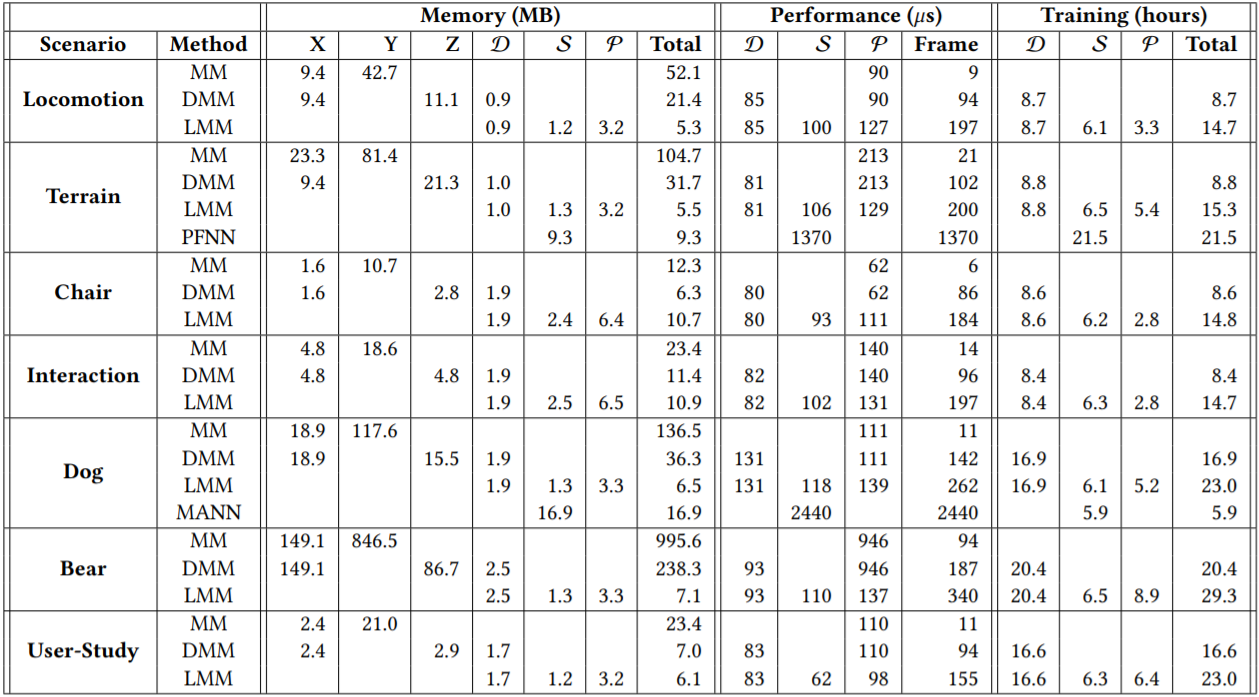


图 7. 该方法与预定义路径之间的接近程度。



图 8. 反向动力学对此方法的影响。



表格 2. 基本运动匹配（MM），带Decompressor的运动匹配（DMM），论文中方法（LMM），相位函数神经网络（PFNN），模式自适应神经网络（MANN）之间的内存、性能以及训练时间的比较。

最后，需要注意的是，尽管与基本运动匹配（无需训练）相比，该方法训练时间较长，但其框架仍然允许艺术家和设计师快速迭代，因为他们可以使用基本运动匹配，直到他们对结果满意为止，然后简单地切换到学习的运动匹配作为后处理。

**4 未来方向**

作者在研究中发现Projector的最终精确度是整体结果的质量的最大影响因素。同时发现如果增加网络隐藏层的数量会使运行时的估计时间增加，从而超过制作的预算。作为未来的一步，探索在不牺牲运行时性能的情况下提高精确度的方法会很有趣，比如使用内存层[Lample et al.2019]。类似地，如果能找到一种方法来使该系统能够进行端到端的训练，而不是在文中介绍的两阶段过程，这也将是很有趣的。

**5 总结**

该论文介绍了运动匹配以及一种运动匹配的替代方法，该替代方法由三个不同的神经网络组成，每个神经网络用来模拟运动匹配算法中特定的组件。通过这种方法可以将现有的基于神经网络的方法的可扩展性与运动匹配的灵活性结合起来，从而在多个领域中实现质量、性能以及内存使用量上的最先进的水平。

习得的运动匹配具有从根本上改变现有的基于运动匹配的动画系统缺陷的可能性，并允许艺术家，设计师和程序员完全释放他们的创造力，可以建立对成千上万种不同情景做出真实而独特的反应的角色，而无需担心对内存或性能产生的影响。

参考文献

[1] Holden, D., Kanoun, O., Perepichka, M., & Popa, T. (2020). Learned motion matching. ACM Transactions on Graphics (TOG), 39(4), 53-1.

[2] Simon Clavet. 2016. Motion Matching and The Road to Next-Gen Animation. In Proc. of GDC 2016.

[3] Kristjan Zadziuk. 2016. Motion Matching, The Future of Games Animation... Today. In Proc. of GDC 2016.

[4] Fabio Zinno. 2019. ML Tutorial Day: From Motion Matching to Motion Synthesis, and All the Hurdles In Between. In Proc. of GDC 2019.