

自动索引多组件字符

Gaurav Bharaj, Thorsten Thormählen, Hans-Peter Seidel, Christian Theobalt

MPI Informatik, 德国



图1: 我们的方法为可以映射到输入动画框架(最左边)的多组件网格创建装备。

摘要

即使对于有经验的动画制作者来说, 通过创建动画骨架来索引任意3D角色也是一个耗时的过程。在本文中, 我们提出了一种算法, 可自动为多组分3D模型创建动画装备, 因为它们通常可在联机形状数据库中找到。我们的算法将输入的多分量模型和输入动画框架与相关的运动数据一起输入。然后为输入模型创建一个目标骨架, 计算刚性蒙皮权重以及目标骨架和输入动画骨架的关节之间的映射。自动方法不需要额外的语义信息, 例如组件标签或用户提供的对应关系, 并且在组件数量显著不同的广泛模型上成功。它隐式处理输入和目标骨架之间的大规模和比例差异, 并且可以处理某些形态差异, 例如, 如果输入和目标具有不同数量的肢体。我们算法的输出可以直接用于重定向系统中, 以创建合理的动画角色。

类别和主题描述符(根据ACM CCS): I. 3. 7 [计算机图形]: 三维图形和现实主义 - 动画

1. 介绍

计算机动画从能够制作不可能的动画中吸引了很多吸引力。通常情况下, 人造物体是动画的。著名的例子是Pixar的Luxo Jr. (1986) 中的两个灯, 或者故事片中的多部分人物, 比如玩具总动员 (1995), 机器人 (2005), 变形金刚 (2007) 等。创建这样的动画是技术上具有挑战性的过程, 甚至需要有经验的艺术家的日子, 如果不是数周的工作。在一个常见的场景中, 艺术家以输入动画的形式提供

通过动作捕捉或艺术家关键帧捕捉到的运动骨架。该动作必须转移到目标角色, 该角色仅作为静态形状表示形式给出, 但没有动画骨架。

在一个称为索具的过程中, 艺术家手动为目标模型定义一个骨架, 并通过表面蒙皮将几何片段附加到它上面。目标骨架不仅需要符合输入运动骨架的结构, 还需要包含放置在物理适当位置的关节。最后, 动画师需要定义输入和输入之间的对应关系



目标骨架关节, 然后可以将目标钻机和输入运动传递给重定向系统以合成最终动画。如果输入和输出框架的比例非常不同, 或者它们在形态上不同, 例如目标模型的肢体比输入动画框架更多, 则会出现非常普遍的挑战。

在本文中, 我们提出了一种新的算法, 该算法可以自动创建用于多组件角色模型动画的骨骼装备, 因为它们通常在联机形状库中找到。多组件模型由数十到数百个单独的闭合三角形网格组成, 但没有组件标签。我们的算法期望作为输入多组分模型和移动动画骨架。输出是模型的目标骨架, 包括刚体蒙皮权重, 以及输入动画骨架和目标骨架之间的联合对应关系。

该算法存在以下核心贡献:

- 一种用于骨架生成的新方法, 从组件之间的接触图开始, 并通过图聚类来简化这种初始表示, 从而导出具有刚性蒙皮权重的有意义的目标骨架。
- 基于动态规划的输入与目标骨架之间的多组件多对一联合映射方案。

如果模型中的组件数量与输入动画中的关节数量的数量级不同, 并且它对骨架尺寸的差异是强健的, 我们的方法也会成功。它固有地处理输入骨架和目标之间的某些形态差异, 例如不同数量的肢体。该方法提出了合理的骨架位置, 尽管在大多数多组件模型中关节通常没有明确建模。输入和目标骨架之间的联合对应以这样的方式进行计算, 即反映输入骨架的结构, 并且因此输出动画保留输入的特征。此外, 我们的算法对变化的输入网格分辨率具有鲁棒性, 即使对由数千个组件构成的输入形状也可以可靠地找到钻台。该方法是通用的, 并且在没有用户干预的情况下成功, 并且不需要任何语义拉普拉斯像在[HRE*08]或逐点对应, 如[ZXTD10]。

2. 相关工作

在本节中, 我们将回顾我们在方法中处理的有关子问题的重要相关工作。

自动索具。许多相关论文试图为机械组件的集合制作动画, 假设组件之间的相互作用已经以物理上合理的方式进行建模, 即通过明确建模的关节几何形状或接触表面上合理定义的相互作用进行建模。哈恩的做法[Hah88]通过研究它们的物理特性来动画刚体组件

摩擦, 质量和惯性矩。系统不会像我们的目标那样尝试制作复杂的多组件模型。模拟人生[Sim94]提出了使用基于人工神经网络的学习方法来演变形态变化的简单生物的运动的技术。

Xu等人[XWY*09]动画3D (CAD) 模型, 但期望正确模拟真实接头的几何形状, 包括例如旋转, 棱柱或球窝接头。其他方法, 如Baran等人提出的方法[BP07], 自动钻取单组分可变形模型。Hecker等人[HRE*08]描述了一种钻取用户创建的虚拟方法字符。然而, 他们的方法需要明确的用户输入, 例如用户给出的用于绑定肢体的语义标签, 例如手臂或腿。另一个动画范例是两个已知表面对应模型之间基于网格的变形传递[ZXTD10, SP04]。相比之下, 我们的系统通过使用全自动管线系统为数百个小型部件组成的3D模型进行动画和动画处理。

形状分析。最近, 多组分CAD模型的形状分析已经取得了长足的进步。一些作品探索组件如何与每个组件进行机械交互

其他在物理上是正确的。有基于滑点分析的方法[GG04, XWY*09]试图以前, 根据建模的关节几何结构处理正确的数学关节类型。最近, Mitra等人[MY*10]探讨了如何稳步发展, 正确地建模的机械组件会移动; 但是需要具有详细建模几何体的CAD模型。在我们探索的在线数据库模型中, 大多数模型都是为了艺术目的而创建的, 而不是工业CAD。直觉上, 我们知道艺术模型的组成部分应该如何相互作用。然而, 仔细检查表明艺术家通常不会明确地模拟接触表面或关节的实际几何形状。因此, 滑动分析不适用于大多数这些模型。

骨架的创作, 装修和剥皮。Katz和Tal [KT03]和Au等。[ATC*08]提出方法

提取单组分可变形模型的骨架。Baran等人[BP07]提出了一种将给定骨架嵌入到单表面两足动物角色和计算中的方法

剥皮权重。Hecker等人[HRE*08]模型char-通过使用用户提供的语义组件标签将组件与已知骨架相结合。其他ap-

步骤, 例如[XWY*09], 使用多组件CAD模型并将网格转换为个体素网格来为其生成动画。这种方法并不直接符合行业中已建立的动画管道, 因此, 需要一种与用于动画的骨架重定向完全不同的方法。

我们的方法为任意划分的刚体装配生成动画装备。因此, 对于单壳表面计算蒙皮的方法, 例如[BP07]和[KSO10], 不能直接应用。

寻找重新定向的对应关系。在骨架 -

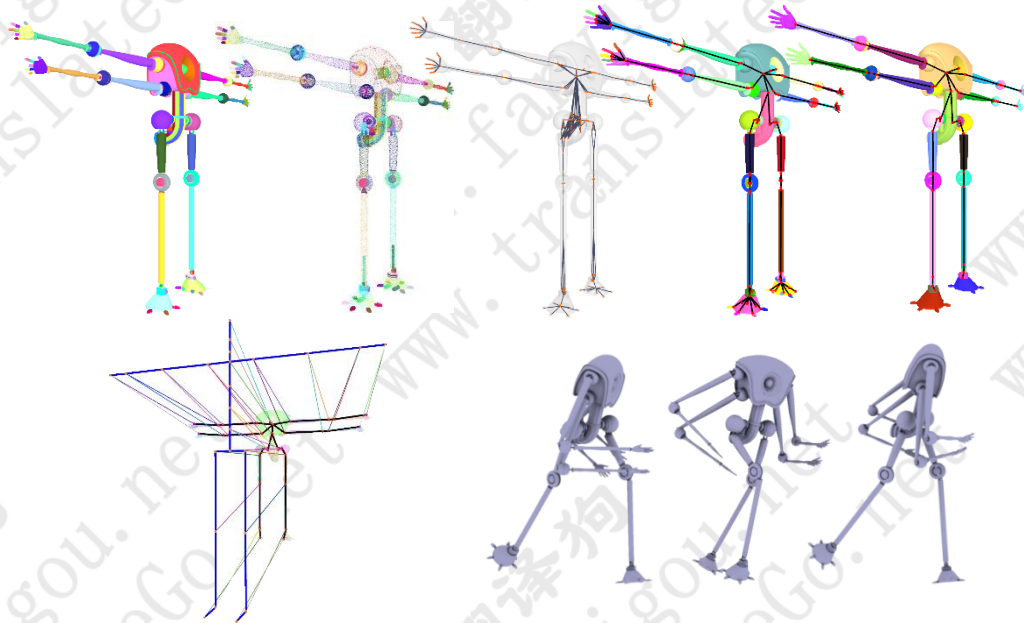


图2: 我们的方法概述 (从左上到右下): 分段输入网格; 基于点的表示; 提取接触图; 接触图的高分辨率聚类 (定义输出骨架); 低分辨率聚类; 输入动画骨架和联合映射到输出框架; 使用联合映射生成的不同绑定输入网格姿态。

术语重定向指的是将关节角度数据从输入骨架映射到目标骨架的过程, 从而满足某些合理运动的约束条件 [Gle01, CK99]。或者, 在无骨架的域中, 通过采取基于网格的动画传输隐式地可以获得重定向 [ZXTD10, SP04]。工作 [YAH10], 作者在角色间转移程式化的动作。所有的方法都需要输入和输出框架或表面之间的语义对应, 通常由用户提供。相反, 我们的方法自动计算这种对应关系。

(高分辨率和低分辨率) 接触图和输入动画骨架的关节。这导致了一个装配好的输入网格, 即包括骨骼分层结构和互连关节的多组件模型的输出骨架, 以及将每个组件与适当的骨骼相关联的刚体蒙皮权重。通过将输入骨架的动画数据重定向到生成的装备, 可以动画模型。

在下面的章节中, 我们将详细介绍我们的自动索具和动画管道的每一步。

3. 自动索具和联合测绘

图1中显示了所提出的用于多部件字符的自动索具和联合映射的方法的概述。2. 我们管道的输入是一个由许多独立组件组成的3D网格。每个组件的表面被采样为点云。基于点云数据, 执行组件的接触分析。这产生了一个图形, 其中每个组件都由一个节点表示, 并且其中如果相应的组件接触, 则在两个节点之间存在边缘。这个接触图通常非常复杂, 并且不直接适用于输入模型的绑定。然后简化接触图并通过聚类节点将其转换为树。从而生成图表的高分辨率和低分辨率聚类。我们管道的第二个输入是输入动画框架。对应关系是在聚类 -

3.1. 输入

我们的方法通常需要两个输入: 首先, 用户提供3D网格, 其次输入动画骨架。

我们期望3D网格由许多单独的组件组成, 但三角形的组件标签不可用, 即该模型是先验的, 只是一组非集群三角形补丁。对于诸如机器人, 船舶, 汽车, 家用电器等人造物体通常是这种情况。特别是, 我们的方法不适用于由单个壳体制成的网格, 因为通常情况下人类或动物的3D模型。输入3D网格可以在任何3D建模软件包中手动建模, 或者可以在互联网模型数据库 (如Dosh Design, Turbosquid或GoogleWarehouse) 中找到。这种互联网数据库中的大多数3D网格都没有用于动画的骨架。第二个输入是一个动画骨架。这种骨骼 -

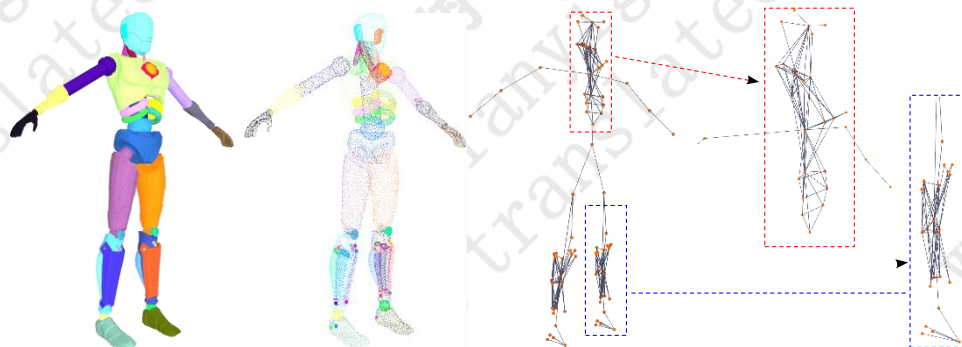


图3: 接触图的生成:(从左到右)多组分网格, 点云, 计算出的接触图, 高频成分, 即过度密集连接的接触图分量: 胸部放大率, 放大率足部区域

吨是由骨骼连接的树木层次结构给出的。还提供了定义联合转换的运动数据。这些动画骨架既可以手动建模, 也可以在运动捕获数据库中找到, 例如CMU Mocap数据库或文档Mocap数据库(HDM05)。

3.2. 接触图

为了将动画骨架的运动转移到输入3D网格, 我们需要建立3D网格的所有组件和动画骨架的关节之间的对应关系。这是一项颇具挑战性的任务。首先, 组件和关节之间没有一对一的映射。相反, 许多组件可以与同一个关节相关联。其次, 输入网格的结构可能与动画数据的结构明显不同, 例如, 可能有不同数量的手臂或腿。因此, 为了解决这个具有挑战性的对应问题, 我们需要从与动画骨架兼容的输入网格中提取输出骨架结构。

为多分量输入网格提取初始骨架结构的任务与为单个外壳创建的网格生成骨架非常不同。不幸的是, 尽管有几何比较, 但精确确定合适的接头位置并不是微不足道的, 这将使模型以合理的方式表达出来。原因在于, 在大多数多组件模型中, 关节或接触表面没有被建模为真正的几何实体, 这大大减少了建模时间, 但是去除了可以在骨架提取中利用的有价值的信息。因此, 我们必须制定一项策略来确定潜在的关节位置, 尽管缺少几何信息。对于多组件网格, 组件之间的连接已经是一个重要的提示, 并且这是一个有效的假设, 即关节位于组件之间的接触点附近。因此, 我们处理流程的第一步是建立联系图。

在接触图 $\alpha(N, E)$ 中, 每个节点 N_i 是网格分量, 并且如果分量 N_i 和 N_j 接触, 则存在边缘 $E_{i, j}$ 。

通过搜索连接的多边形, 将输入网格分割成组件。组件是输入模型的一组多边形。将多边形分配给组件的过程是一个简单的重复算法。首先创建一个新组件。然后, 我们从任何多边形开始, 在多边形网格的边上执行区域生长。通过区域生长可达到的每个多边形属于同一个组件, 并将其添加到此组件的多边形集合中。一旦区域增长停止, 我们通过创建下一个组件并在尚未分配给该组件的所有多边形上执行区域生长来重复该过程。一旦所有多边形都分配给组件, 此过程将停止。

我们现在要确定哪些组件与之联系。为了避免非均匀网格分辨率的问题, 我们首先使用Hammersley点采样将所有组件表面重采样为点云, 如[WLH97]。我们还计算每个组件的定向边界框。然后, 对于任何两个给定的组件, 我们检查它们的定向边界框是否碰撞。如果是这种情况, 我们检查组件的任何两点是否小于距离阈值 ϵ 。我们凭经验选择 ϵ

整个网格尺寸的0.01%。对于任何两个组件, 在两点足够接近的情况下, 我们生成接触点, 即边 $E_{i, j}$ 。图中显示了分段输入网格和生成的接触图的示例 3。

3.3. 图聚类

接触图已经是输入网格骨架结构的一个很好的表示。理论上, 人们可以考虑使用接触图和输入动画骨架之间的图形匹配来导出对应关系。然而, 接触图和动画骨架的分辨率和结构是非常不同的,

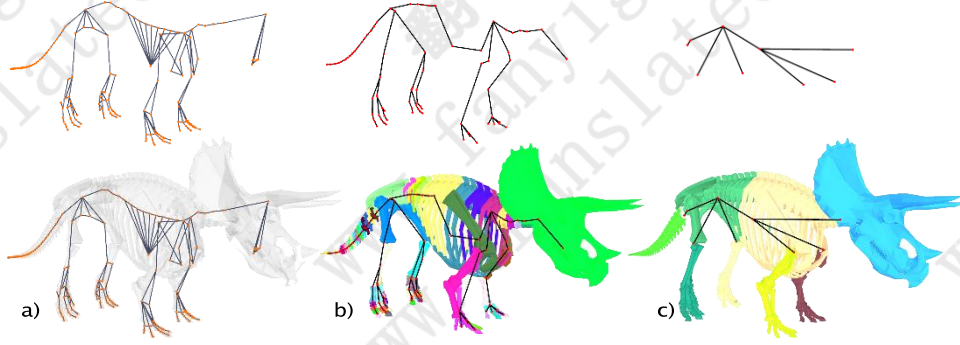


图4: 图聚类结果: a) 接触图和输入网格, b) 高分辨率聚类 (属于相同聚类中心的网格的那些分量由相同颜色表示), c) 低分辨率图

因此, 图形匹配技术, 例如[KA09, LH05, GR96], 不会导致合理的映射。此外, 如果动画骨架和输入网格在形态上不同, 例如, 具有不同数量的肢体, 则通用图匹配方法不处理多对一映射的问题。因此, 我们需要为输入网格生成更简单的骨架结构, 以便更适合与动画数据进行比较。

特别是, 由于动画骨架本质上是一个树形层次结构, 我们还需要从我们的接触图中提取一个合适的树。另外, 接触图特征是过于密集连接的部分的子集, 换句话说, 高频部分太详细以至于不能形成较接部件的合理集合, cp 。图 3。我们想要简化和聚类这些高频成分, 以建立关于输入网格的粗糙语义上更可信的结构的信息。因此, 我们通过算法模仿人类动画师的工作, 他将从更高层次的角度来看待输入, 寻找功能合理的结构。接触图 $\alpha(N, E)$ 的这种粗略版本可以通过对其一些节点 N_i 进行聚类而生成。为此我们采用二次误差度量 (QEM) 方法

在[GH97], 其中应用程序是网格简化。Au等人之前已经证明了这一点。[ATC*08]这个方法也可以应用于图聚类。从概念上讲, 图的边 E_{ij} 被视为三角形的边, 节点 N_i 相当于几何网格的顶点。**聚类算法。迭代地聚类算法**

折叠图形的边缘, 直到没有更多边缘折叠。令 3 -矢量 $n_i = (n_x, n_y, n_z)^T$ 为中心相关组件的点云的质量

到节点 N_i 。如果边 E_{ij} 崩溃, 则最终折叠的节点中心 n 总是被计算为两个涉及的节点中心 n_i 和 n_j 的几何平均值。我们定义一个边 E_{ij} 如果满足标准 A 是可折叠的, 要求

A.1 节点 N_i 或 N_j 的程度大于 2, 或

A.2 连接节点 N_i 和 N_j 的路径不止一条

A.3 组件的表面积分配给 N_i 或 N_j 小于给定阈值, 或

A.4 边 E_{ij} 的长度小于给定的阈值。

这种可折叠性标准正式形成了上面直观概述的图聚类策略。输出框架不应该包含太密集连接的太小部件的部分, 这些部分可能属于同一个实体, 即同一个骨骼。另外, 如果较小的部件之间存在很多链接, 则要识别与主骨链相对应的主要路径。为了确定在每次迭代中哪个边要折叠, 每个可折叠边被分配边缘成本 $C(E_{ij})$ 。在每次迭代中, 具有最小边缘成本的边被折叠。边缘成本定义为

$$C(E_{ij}) = w_1 C_{\text{形状}}(E_{ij}) + w_2 C_{\text{采样}}(E_{ij}), \quad (1)$$

其中 $C_{\text{形状}}$ 是形状成本, $C_{\text{采样}}$ 是取样成本, 它们的权重分别选择为 $w_1 = 1.0$ 和 $w_2 = 0.1$ 。形状成本有助于保持图形的整体形状, 而采样成本有助于防止产生长边。形状成本由下式给出

$$C_{\text{形状}}(E_{ij}) = \varepsilon(n_i) + \varepsilon(n_j) \quad (2)$$

其中误差度量 $\varepsilon(n_i)$ 和 $\varepsilon(n_j)$ 分别是折叠节点中心 n 到 n_i 和 n_j 所有相邻边缘的距离的总和, 其可以通过

$$\varepsilon(p) = p^T \sum_{(i,j) \in E_i} K_{ij} K_{ij} p \tilde{p}^T q_i \tilde{q}_i, \quad (3)$$

其中 $\tilde{p} = (p_x, p_y, p_z, 1)$ 是 3 向量 p 的齐次表示, E_i 是连接到节点 N_i 的边集, 3×4 矩阵 K_{ij} 由

$$\begin{bmatrix} 0 & -a_z & a_y & -b_x \\ -a_z & 0 & -a_x & -b_y \\ a_y & -a_x & 0 & -b_z \end{bmatrix}$$

$$K_{ij} = \begin{bmatrix} -\hat{a}_z & 0 & -a_x & -b_y \\ -a_z & -\hat{a}_x & 0 & -b_z \end{bmatrix}$$

用 $a = (n_i - n_j) / (n_i + n_j)$ 和 $b = a n_i$ 抽样成本惩罚可能产生长边的边缘崩溃, 并由其给出

$$C_{\text{折叠}}(E_i, j) = \ln i - n_j \sum_{(l, k) \in E_i} \ln i - nk l_o \quad (4)$$

一旦找到边缘得分最低的边缘, 该边缘就会折叠。折叠节点中心 n 由算术平均值 $n = 0.5 (n_i + n_j)$ 给出。 N_i 和 N_j 的所有相邻边成为折叠节点 N 的相邻边 (除了重复)。此外, 折叠节点的合并矩阵 Q 由 $Q = Q_i + Q_j$ 给出。在下次迭代中, N_i 和 N_j 被移除, 折叠节点 N 被精确处理为图的任何其他节点 N 。唯一的区别是它的合并矩阵 Q 用在方程 3 而不是通过当前 K_i, j 的和来计算更新的 Q 。重复这个迭代过程直到根据标准 A 没有更多的可折叠边缘。在图形聚类期间, 我们通过记录在每次迭代中合并哪些节点来维护折叠节点的历史。这些信息将在稍后进行讨论。

低和高分辨率的图形。 我们生成两个不同版本的聚类图, 即高分辨率图和低分辨率图。如上所述生成低分辨率图形, 强制执行标准 A 的所有四个子标准。

高分辨率图形仅通过强制执行子标准 A. 1 和 A. 2 生成。通过删除子标准 A. 3 和 A. 4, 我们保留所有只能通过单一路径到达的叶子或节点的节点, 无论它们的表面积或边长是多少。

虽然两种表示都保留了多组分网格的整体形状, 但每种都有其自身的优势。高分辨率图形给出更多细节, 但低分辨率图形更好地表示网格的粗糙结构。图 4 显示示例模型的生成的低分辨率和高分辨率图形。

树提取。 由于子标准 A. 2, 群集图中的每个节点只能通过单个路径到达。如果其中一个节点被选为根节点, 那么这样的无向非循环图可以直接转换为树。我们选择根节点作为中间性最高的原始联系图 $\alpha(N, E)$ 的节点 N_i 。如果节点 N_i 在集群图中不再可用, 我们使用记录的集群历史记录来查找集群图中的代表性节点 (即从其演化的节点), 并将其用作树的根。中间中心考虑了图的局部和全局结构和连通性 [Fre77] 并确定联系图中最重要的节点。我们还尝试了特征, 程度和亲密度的中心性, 发现中间性 - 中心性效果最好。在这个阶段之后, 我们有两个版本的形状模型输出动画框架, 一个是高分辨率, 另一个是低分辨率。

刚性蒙皮。 由于我们在图形聚类过程中保留了折叠节点的历史记录, 因此我们可以确定将接触图中的哪个节点分配给最终聚类树中的节点。我们也知道哪些顶点

多部件网格属于接触图中的哪个节点。这个信息可以用来分配一个向量

对多组分网格的每个顶点的刚性蒙皮权重。此向量中的每个元素表示最终聚集树中的一个节点。在刚性蒙皮中, 向量元素对于该顶点所属的最终聚类树的节点为 1, 对于聚类树的所有其他节点为 0。在图 1 中, 4 可视化了刚性蒙皮。属于聚类树的特定节点的每组顶点以不同的颜色显示。刚性蒙皮的优点是我们的结果与线性混合蒙皮 (LBS) 直接兼容 [MTLT88] 格式, 几乎所有专业 3D 动画包都支持这种格式; 然而, 由于每个顶点被唯一地分配给一个节点, 所以在关节周围不存在变形混合。相反, 每个顶点都会随着转换到其分配的节点而严格移动。这具有预期的效果, 即在动画过程中, 属于聚类树的特定节点的多组分网格的所有组件都将作为刚性聚类移动。

3.4. 联合绘图

重定目标算法, 例如 [Gle01, CK99] 或 Aus Human IK 通常会产生良好效果。Hecker 等人

人。[HRE*08] 提出了一个自动重新定位系统; 怎么样, 曾经, 假设每个肢体的语义标记是已知的。对于来自因特网数据库的 3D 模型, 通常不提供这样的语义信息。因此, 大多数重定向管道需要手动步骤, 即用户手动定义目标骨架的关节与输入动画骨架之间的映射。在本节中, 我们提出一个算法, 它自动计算这个映射。该方法利用了上一节中提取的高分辨率和低分辨率图形。

基本上, 联合映射是一个图 (或树) 匹配问题。同样, 理论上, 应用通用图匹配方法也是可能的 [LH05, GR96, KA09] 找到我们的聚类图和动画骨架之间的映射。这将比直接在联系图和动画框架之间找到映射更好。但是, 它仍然不可靠, 并且如果输入数据不能扩展到包含数百个组件的模型动画骨架比较粗糙。还有一种类似于选举投票的方法 [空中交通管制员*10] 不能用于

我们的设置是因为我们还希望为输入框架和输出框架形态不同的字符映射 (即允许许多对一的分支匹配)。我们还希望找到未在参考姿势中精确建模的角色 (如 T 姿势) 的映射。

进一步来说, 我们的输出骨架的结构可能仍然不完全匹配输入动画骨架的结构, 例如因为某些子链中的关节数量可能

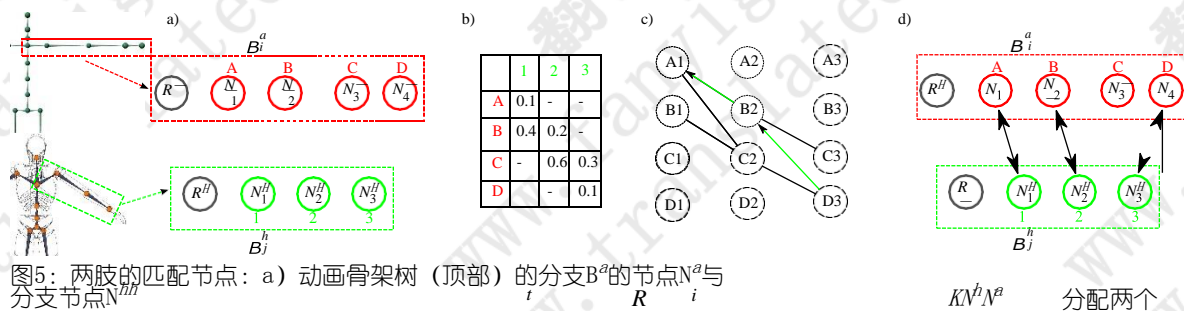


图5: 两肢的匹配节点: a) 动画骨架树 (顶部) 的分支 B_i^a 的节点 N_i^a 与分支节点 N_i^h

节点 N_i^h 的分辨率树 (底部), 表示根节点; b) 成本 (s, t) 节点 c), 因为节点只能按连续顺序匹配, 所以只有某些分配是可能的, 如此 Trellis 图所示。具有最低分支成本 Q (绿色路径) 的分配可以通过动态编程找到; d) 生成具有最低分支成本的映射。

不同。因此, 没有将每个输入骨架关节映射到输出骨架关节的独特方式, 并且没有

独特的离开关节的方式, 即使它们与层次结构中最近的映射关节一起移动。因此, 我们希望开发一种制图方法,

创造出最合理的一套联合通讯 - 因此输入骨架的结构是最好的

在动画树分支中也存在相应的节点:

$$\forall B_i: \exists B_j \text{ 和 } (\forall N_s \in B_i): (\exists N_t \in B_j) \quad (5)$$

此外, 节点的映射必须是连续的, 这意味着如果 N_i^h 映射到 N_i^a , 那么 N_{i+1}^h 只能映射到 N_{i+1}^a 和你 > 0 。此排序约束也

映射到输出装备, 输入动画的本质可以转移到目标模型。

在下文中, 我们提出了满足这些要求的骨骼关节映射公式。联合映射使用角色的高分辨率和低分辨率骨架树, 这是使用图形聚类方法生成的

这里表示为 $\tau(N, E)$ 和 $\tau^l(N^l, E^l)$ 。最终映射输出框架的结构基础是 $\tau^h(N^h, E^h)$, 而 $\tau^l(N^l, E^l)$ 仅支持算法寻找最佳映射。输入的动画骨架也作为树结构给出,

并由 $\tau^a(N^a, E^a)$ 表示。这个想法是确定输入网格中的肢体数量由多少个

输入网格中的肢体数量由多少个在低分辨率树中分支。分支 B 被定义

作为位于从根节点 R 到叶节点的路径上的所有节点 N_i 的集合。例如, 4c 低分辨率树有六个分支。

联合映射算法。实际的联合映射,

动画骨架树的第 j 个分支 B_j^a 与输出框架之间的匹配通过匹配节点上的节点来执行高分率树的分支 B_j^h (仅作为分支

确保没有双重分配节点。这里以及下面的算法描述中, 我们假设分支 B_j^h 中的节点比分支 B_j^a 中的节点少。如果情况并非如此, 使用相同的算法来查找映射, 但是分支 B_j^h 和 B_j^a 的作用被交换。让我成为高分率分支 B_j^h 的数量 i

树和 J 动画中的分支 B_j^a 的数量 j

吨树。为了找到我们的映射问题的一个解决方案, 我们形成了一个映射成本 Q 的完整 $I \times J$ 分支矩阵

$$B = \begin{pmatrix} Q(B^h, B^a) & Q(B^h, B^a) & \dots & 0 & Q(B^h, B^a) \\ 1 & 0 & 1 & 1 & \dots & 1 & J & \dots \\ Q(B^h, B^a) & Q(B^h, B^a) & \dots & Q(B^h, B^a) \end{pmatrix} \quad (6)$$

$$Q(B^h, B^a) \quad Q(B^h, B^a) \quad \dots \quad Q(B^h, B^a)$$

对于分支矩阵 B 的所有元素 $b_{i,j}$, 通过求解优化问题来计算该映射成本 Q 。该过程找到节点 N_i^h 和 N_j^a 使得

B_i 使用节点映射成本 K 分支到节点 N_j^a , 使得整个分支的整体映射成本 Q 被最小化

$$Q(B, B) = K(N, N)$$

的高分辨率树包含足够数量的节点)。通常,高分辨率树的几个分支与来自低分辨率的分支重叠

树,因此,我们知道,我们只需要最后为这些分支之一分配映射。为了找出哪些分支可供选择,我们计算一个映射

为高分辨率树的所有这些分支并保持只有映射成本最低的那个。

更正式地说,对于高分辨率的每个分支 B_i^h

树必须存在一个对应的分支 B_j^l ;
matation 骨架以及此分支的所有节点都必须

$$h_i^a = \sum_{s=1}^n h_{s(i)}^a \quad (7)$$

受上面提到的映射限制
的节点 $s \rightarrow t$ 必须按连续顺序排列。节点映射成本 K
由下式给出:

$$K(N^h, N^l) = \alpha K(N^h, N^l) + \alpha K(N^h, N^l) \quad (8)$$

在这里，术语KPos是世界空间中两个节点之间的位置差异（欧几里得距离），KPerc是节点沿其各个分支的位置差异，以完整分支长度的百分比给出，

$Kdeg(N, N')$ 是节点的度数 $cen = tractity$ ，而KBet和KArea是高分辨率树中节点中心度和归一化面积的归一化值。每个项的加权因子通过实验选择为 $\alpha_1 = 10.0$, $\alpha_2 = 2.0$, $\alpha_3 = 0.2$, $\alpha_4 = 1.0$, $\alpha_5 = 0.5$ 。

通过强力搜索解决这个优化问题具有组合 $O(n^2)$ 的复杂性。但是，这个问题与扫描线上的序列匹配问题类似

立体相机[VMVPVG02]。使用相同的算法，求解优化问题的最优解。7 可以通过动态编程有效地找到（如图所示）5）。

一旦分支矩阵B的所有元素 b_{ij} 都是通用的，我们可以找到一个分支B的最佳映射

选择相应的 B^a

j ，产生最小的

成本 $Q(B^h, B^a)$ 为矩阵B的每一行。结果是我们有高分辨率树中的每个分支找到一个分支映射到动画骨架树中的一个分支以及相应的节点映射。

减少映射的数量。 我们通过分析高分辨率树的哪些分支与来自低分辨率树的分支重叠并仅保留具有最低映射成本Q的最佳映射来减少这组分映射。作为例子，4 高分辨率树有 $I = 17$ 个分支，低分辨率树有 $L = 6$ 个分支，所以我们只保留17个可用分支映射中的6个。为了找出哪些分支重叠，我们查看高分辨率树和低分辨率树起源的原始接触图的节点。由于我们保留了折叠节点的历史记录，因此我们知道来自联系图的哪些节点聚集在一起以形成高分辨率树或低分辨率树的结果节点。高分辨率树的每个分支都被分配到低分辨率树的分支，与其共享最多的联系图节点。结果，我们获得了将高分辨率树的分支分割成L个集合。对于这些集合中的每一个，我们只保留具有此集合中最低分支匹配成本的分配。

处理冲突的映射。 独立的分支可以在重叠的分支段处重叠并共享节点。当我们分别对待每个分支时，可能发生高分辨率骨架中的同一节点被映射到动画骨架的多个节点。在这种情况下，我们应用赢家通吃方法，从节点中删除所有冲突映射，并且只保留针对其产生的成本Q的分支的映射（根据方程式。7）最低。

更新刚性蒙皮。 可能发生节点 N^h 没有从动画骨架节点 N^a 获得映射。在

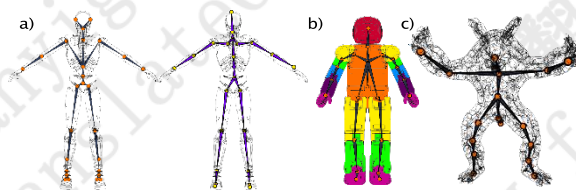


图6：比较：a）我们自动创建的钻机（左）非常类似于艺术家制作的钻机（右）； b）模拟关节几何的Asimo：我们的方法发现的关节正确放置在模拟关节附近； c）我们的方法也成功地应用于高度断裂的多组分物体。

在这种情况下，我们必须更新其刚性蒙皮重量。所有属于节点 N^h 的顶点现在都被分配给该分支中的节点的父节点。重复这个过程，直到到达一个有父节点的映射动画骨架。

4. 结果

在本节中，我们将展示一些使用我们的方法生成的骨架创建，刚性蒙皮/聚类 and 联合映射的结果。其他结果可以在补充视频中找到，我们还会显示动画角色。为了生成这些动画，Autodesk的HumanIK用于运动重定向步骤。我们所有的输入模型都是直接从互联网存储库中取得的，无需修改。

如图1和2所示。1 和 7，所提出的方法能够处理完全不同的形态（例如，两足动物，四足动物和未成形）。此外，所示模型的组件数量也有很大不同（#组件数量为angel bot = 165, t-rex = 183, toilet = 9, 大型机器人 = 220, horse = 152, armadillo = 567）。我们测试了不同的输入动画框架（两足动物，四足动物），它们有时在规模和形态上与多组分模型明显不同。在所有情况下，都会生成非常合理的钻井平台，并将各种输入运动（从步行到复杂的武术移动）重定向到这些钻机，一致地产生可信的动画。对于所有显示的模型，系统需要不到半分钟的时间来执行流水线的所有步骤（单个CPU上的C++代码）。平均来说，联系图计算需要1.9秒，而图群集需要2.0秒，联合映射需要0.784秒。图。6 表明用我们的方法创建的骨骼可以与艺术家制作的骨架相媲美。我们的方法也可以用明确建模的关节几何体进行网格划分。例如，为Asimo模型生成的骨架，如图所示 6b，与所产生的结果相当

通过一种需要显式联合几何进行分析的方法[XWY*09]。此外，使用多组件对象

in [ZXTD10]，符合我们的管道（如图所示）6c），这进一步说明了我们方法的多功能性。



图7: 具有不同形态的五个模型的结果:(从左到右)输入多组分模型; 分割和接触图; 生成高分辨率骨架; 生成的骨架与输入动画骨架的联合映射。

5. 讨论和未来的工作

由于我们管道的输出使用（行业标准）线性混合蒙皮格式，因此计算出的目标骨架和刚性蒙皮权重可以轻松由专业艺术家使用；然而，像我们这样的自动化系统，也使新手用户能够对3D模型进行动画制作，因为他们只需从互联网存储库中选择输入网格和所需的动作即可。大多数模型（如此处和视频中显示的结果）都不需要任何手动干预。尽管如此，我们的自动方法的结果有时会受到估算错误的影响。尤其是，由于我们不使用任何先验或用户给定的语义信息，所产生的联合映射和聚类可能导致模型在动画下不切实际的运作。使用标准的三维动画软件可以很容易地在这种情况下修改自动联合映射，以实现更合理的机械操作。而且，修改该方法将是相对直接的，以便它提出一系列可能的映射，用户从中选择优选映射。

我们方法的另一个局限是它需要多组分网格和动画骨架的大致初始对齐，即它们应该大致处于相同的姿势（至少在一个框架中）。这是必要的，因为在方程中使用了位置项。⁹ 这并不意味着这两个输入都必须处于完美的T形姿势；但两者都必须直立，面向同一个方向。此外，输入模型和动画框架在被我们的管道处理之前会缩放到默认高度。

在未来的工作中，我们还希望扩展我们的方法，将动画骨架的特定运动考虑在内，以估计给定运动的最佳关节映射。最后，我们计划通过使用显式对称准则来改进方法。实际上，大多数生成的钻井平台实际上是对称的，但目前算法中并未实施。

6. 结论

我们已经提出了一个用于自动索具和联合映射计算的系统，用于没有先验组件标签的多组件模型。通过识别相关的组件并分析它们之间的联系，可以生成一个初始图形，通过图形聚类可以简化为粗糙树。在生成模型的骨架表示后，执行与动画骨架的联合映射。该方法快速并且成功地处理了具有数百个组件的模型，以及输入动画骨架和提取的模型骨架之间的形态差异。

参考

TALC.-L., CHH.-K., COHEN-OZ
D., L作者: T.-Y.: 通过网格收缩进行骨架提取。ACM
Trans. 图形. 27 (2008)。2,5

- *10] AOK-C., TALC.-L., COHEN-OZ, ZHENG
Y., F.H.: 选民投票快速自动形状对应。计算机图形学论
坛29 (2010)。6
- [BP07] BORDAS L., POPOVIC J.: 自动索具和3D角色动画。ACM
TOG 26 (2007)。2
- [CK99] CHEN K.-J., KIM H.-S.: 在线运动重定向。可视化与计
算机动画杂志11 (1999)。3,6
- [Fre77] FRIEDMAN LC: 基于中介性的一组中心性度量。
Sociometry 40 (1977)。6
- [GG04] GELFAND N., GUTAS L.J.: 使用局部滑点分析的形状分割。
在Proc. SGP (2004)。2
- [GH97] GELFAND M., HECKBERT PS: 使用二次误差度量的曲面简化。
在Proc. SIGGRAPH (1997)。5
- [Gle01] GLENN J.: 比较基于约束的运动编辑方法。图形。
模型63 (2001)。3,6
- [GR96] GIL S., RANGARAJA.: 图形匹配的分级分配算法。
IEEE TPAMI 18 (1996)。5,6
- [哈88] HALL JK: 逼真的刚体动画。SIG-
GRAPH计算机。图形. 22 (1988)。2
- *08] HALL C., RANGARAJA, ENSLER RW, DEWESE (伊瑟)
J., AVNARD J., 面包车 PROOIJEN K.: 实时运动重新定位到高度多样
化的用户创建形态。ACM TOG 27 (2008)。2,6
- [KA09] KANG Y., ALGOZ.: 谱图匹配的概率方法。大多数
(2009)，1-7。5,6
- [KS010] KANG Y.-L., SLOAN P.-P., O'SULLIVAN C.: 快和
高效的动画网格蒙皮。COMPUT. 图形. 论坛29 (2010)。
2
- [KT03] KATZ S., TALC.: 使用模糊聚类和切割的分层网格分
解。ACM TOG 22 (2003)。2
- [LH05] LEONARD M., HALL JK.: 使用成对约束的对应问题的谱技术。
在Proceedings ICCV (2005) 中。5,6
- [MTLT88] MAGNENAT-THALMANN N., LAPERRIERE R.,
THALMANN D.: 用于手动和对象抓握的关节局部变形。在图形界
面 (1988)。6
- [MY*10] MITRA NJ, YAN Y.-L., YAN D.-M., LI W.,
AGARWAL M.: 说明机械组件如何工作。
ACM TOG 29 (2010)。2
- [Sim94] SIMON K.: 不断变化的虚拟生物。在Proc. SIG-
GRAPH (1994)。2
- [SP04] SLOAN RW, POPOVIC J.: 三角形网格的变形传递。ACM
Trans. 图形. 23 (2004)。2,3
- [VMVPVG02] VAN MEERBERGEN G., VERGAUWEN M.,
POLLEFEYS M., VAN GELDEREN L.: 采用动态规划的分层对称立体算法。
IJCV 47 (2002)。8
- [WLH97] WILSON T.-T., LEE W.-S., HENG P.-A.: 采样
与hammersley和halton点。J. Graph. 工具2 (1997年)。
4
- 面包车 K., 面包车 K., 面包车 K. 面包车
PANG M., CHEN F., GSH OESB.: 联合意识对可变形模型的操纵。
ACM TOG 28 (2009)。2,8
- [YAH10] YAN K., AN Y., HODGINS J.: 动画非 -
具有人体运动数据的人形字符。在Proc. SCA
(2010)。3
- [ZXTD10] ZHANG K., XU W., TANG Y., DEFORM M.: Deformation
传输到多组件对象。COMPUT. 图形. 论坛29
(2010)。2,3,8