文献阅读报告

Artemis: Articulated Neural Pets with Appearance and Motion Synthesis

姓名：杨溟豪

学号：19211311

班级：通信 1901

目录

0. 写在前面

0.1. 文献信息

- 论文名称：Artemis: Articulated Neural Pets with Appearance and Motion Synthesis
- 论文作者：Haimin Luo, Teng Xu, Yuheng Jiang, Chenglin Zhou, QIwei Qiu, Yingliang Zhang, Wei Yang, Lan Xu, Jingyi Yu
- 论文年份：2022.2.11
- 论文类型：AI 领域国际会议 2021-2022 年论文

0.2. 选择该篇论文的原因

图 1 系统概念图

看了很多篇论文，有自然语言处理方面、计算机视觉、人工智能方面的，有些论文太过于学术性，充斥大量学术名词，让我这种通信专业学生感到很迷茫。但是这篇论文不同，它主要介绍了一个可以生成虚拟神经动物的系统，同时通过 VR 还可以与用户进行交互，之前在 b 站看过一期游戏建模的视频，其中 BOSS 的建模就是通过对小动物动作的捕捉训练而成的，与这篇论文中的方法很相似，这让我对这篇论文产生了浓厚的兴趣，也正是这种兴趣让我坚持读完了这篇长达 19 页的论文。

这是当时 b 站游戏建模视频链接：

《阶段成果》：游戏科学虎年贺岁小短片

https://www.bilibili.com/video/BV1844y1s7Nk?spm_id_from=333.337.search-card.all.click

PS：看完这个视频再看论文会有不一样的感受！

0.3. 全文脉络总结

由于本篇论文较长，因此我制作了一张图来进行全文的脉络总结，如下图所示。

另外每一章节的关键内容我都在章节标题后第一段进行了总结，写了自己的理解和思考，并对关键内容进行了突出显示，方便阅读。

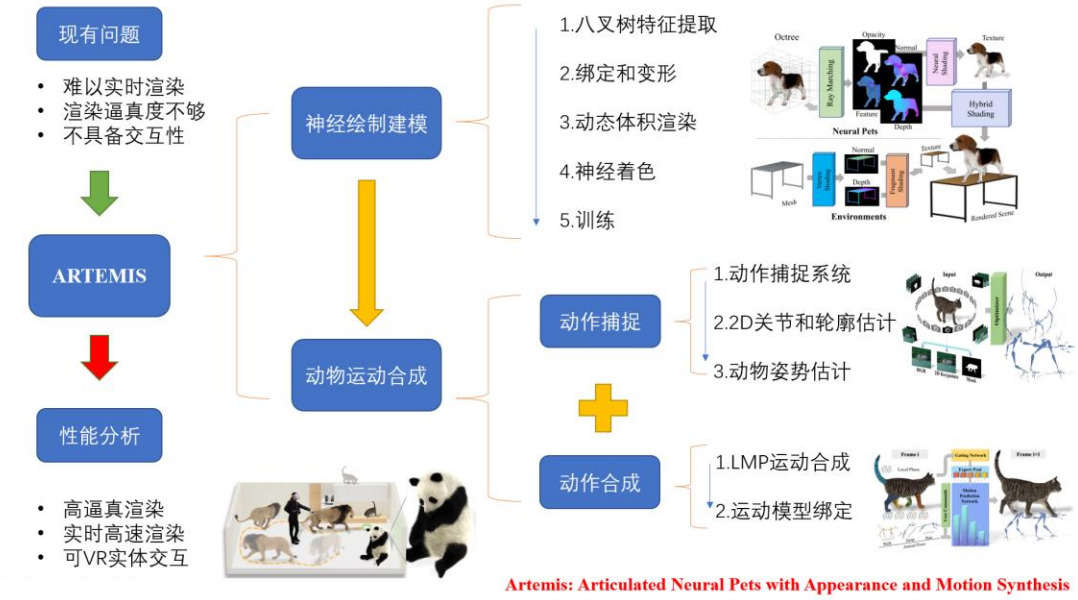

图 2 全文脉络总结

0.4. 感受、收获、体会

之前从来没敢想过能读完长达 19 页的英文文献。在最开始选择这篇文献时内心仍然存在畏难情绪，但是在观看了老师的 b 站讲解视频后，掌握了读英文文献的窍门，阅读速度明显加快，我能够快速的抓住文章的中心脉络，找到每个章节的主题主旨句，对全文的内容有了更深刻的理解。在这个过程中，第一遍略读主题句，第二遍细读，分析数据，查找资料，第三遍快速回忆全文，总结内容。三遍下来，掌握了论文的主要内容，而且不费时费力，效率十分的高。

虽然这篇论文多达 19 页，也具有较多的学术词汇和公式，但是在不断查阅资料的过程中，也有了许多收获，了解了关于动画建模方面的学术前沿，对该领域的学术术语有了更多的了解，如消融研究，RGBA 等术语。

同时我学到了很多计算机图形学的相关知识，对于目前动物动画建模的现状有了了解，其主要存在占用资源高，实时性差的问题，作者在这里开创性的采用八叉树来存储不透明度特征，以解决实时性的问题。另外我对动画建模的流程有了更深刻的理解，首先需要提取图片特征，然后绑定到目标骨骼，在之后进行动态渲染和训练得到一个初步模型，之后通过动作捕捉训练集捕捉动作再进行运动合成预测控制已经得到的模型就完成了虚拟神经动物建模和交互。

综合来讲，这篇论文虽然长，虽然难度不小，但是坚持读下来之后有了不小的收获，对我也一种精神鼓舞，增强了我的信心。

1. 引言

这里引用论文中的一句话： **Our love for animals is a great demonstration of humanity. 我们对动物的热爱是人性的伟大体现。**（这句话真的是很有深意的一句话）目前人类正在进入虚拟时代，

人们更希望能将动物带到虚拟世界中作为伴侣。然而，计算机生成 (CGI) 毛茸茸的动物受到繁琐的离线渲染的限制，很难做到交互式运动控制。

在这篇论文中，作者针对下列的现有问题，提出了一个新的系统-ARTEMIS 一种新颖的神经建模和渲染管道，用于生成具有绒毛外观和运动合成的神经宠物。该系统支持交互式运动控制、实时动画和毛茸茸动物的逼真渲染，进一步将 ARTEMIS 集成到支持 VR 耳机的现有引擎中，提供前所未有的身临其境的体验，用户可以通过生动的动作和逼真的外观与各种虚拟动物亲密互动。

- 现有问题

  目前有关动物模型的制作不是易事，虽然在动画片方面有了很大的成功，但是通过计算机生成数字动物模型，并对其进行动画制作和高真实渲染一直都是动画制作工作室的“奢侈”。因为在这这些工作中，创建的过程需要不仅建模师极大的艺术创造能力还需要计算机强大的计算能力，十分耗费资源。然而即使具有充足的资源，数字动画动物照片级的真实渲染仍然只能离线工作，不能够进行实时互动。

  现存挑战有很多方面，其中最关键最核心的是实时渲染需求和逼真渲染需求之间的冲突。首先，由于动物身上覆盖着毛皮，采用传统造型过程，需要精湛的艺术和繁重的劳动。另一方面渲染过程也十分耗时，通常采用离线光线跟踪来生成半透明、光散射、体积阴影等的照片真实感，即使使用最先进的图形硬件也远远不能达到实时效果。除了渲染之外，以交互速度和高真实感设置模型的动画仍然具有挑战性。最后，为了让数字动物在虚拟世界中茁壮成长，它们应该响应用户的指令，因此，渲染和动画组件都需要与交互紧密集成。

- 现有方法

  传统的物理建模和渲染方法能够生成高质量的图像，但由于毛发和毛皮的复杂光线反射属性，难以进行实时渲染，因此计算量很高。

  基于图像的方法直接从图像中重建头发模型，生成静态头发模型的效率更高，但在动态场景效果上损失了视图的一致性和较大的代价。目前的神经表示方法在有绒毛对象建模和绘制方面取得了很好的效果，但仍然不能生成实时的动态头发和毛发效果。

- 创新贡献

  1. **作者提出了一种新颖的神经建模和渲染系统 Artemis，该系统支持 2D 屏幕上或 VR 环境中的自然运动控制和用户与虚拟动物的交互。**同时收集了不同尺度动物的新的运动捕捉数据集，并训练了一个新的骨骼检测器。

  2. **作者提出了一种可区分的神经表示法，用于建模具有毛茸茸外观的动态动物。**这种新方法可以有效地将传统的 CGI（计算机生成图像）资源转换为 NGI（神经生成图像）资源，从而实现实时和高质量的渲染。

3. **作者为虚拟现实环境下的各种 NGI 动物提供了可控的运动合成方案。**用户可以像在真实世界里一样向虚拟动物发送命令或与其亲密交互。

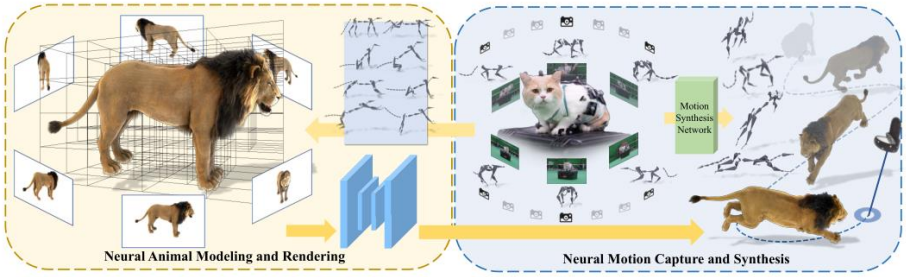

图 3 Artemis 系统结构

2. 动画神经动物

这里首先介绍一个技术：Neural Opacity Radiance Fields（神经不透明辐射场），是一种深度渲染方法。

与物理建模毛发的半透明性不同，作者在这里采用了神经绘制的方法，并将问题转化为辐射场上的视图合成。神经辐射领域的开创性工作从概念上提供了一种自然的毛发渲染的解决方案：NERF 根据光线上每个点上的颜色和密度表示场景，其中密度自然反映该点的不透明度。但是，与自然毛发或头发相比，从 NERF 生成的动画蒙版往往具有噪波且不那么连续。

一种新方法 ConvNeRF 通过在特征空间中处理图像而不是直接在 RGB 颜色中处理图像来解决此问题。这里的关键是使用特征来表示每个空间点的不透明度，而不是直接使用密度。原来的 ConvNeRF 只能处理静态对象。Bruteforce 方法是对动态对象进行逐帧优化，这在通常执行长运动序列的神经动物上显然是不可行的。

作者首先将神经不透明辐射场扩展到动态动物。不仅加速训练，更重要的是实现了实时渲染。在动画场景下。作者将不透明度特征存储在体积八叉树结构中，以用于实时渲染。具体地说，作者设计了一种特征索引方案，用于快速提取和绘制特征。作者还进一步介绍了一种基于骨架的体积变形方案，该方案使用源自原始 CGI 模型的蒙皮权重来桥接标准帧和用于动画的实时帧。此外，作者还设计了一个神经阴影网络，处理动画对象建模，并采用高效的对抗性训练方案进行模型优化。

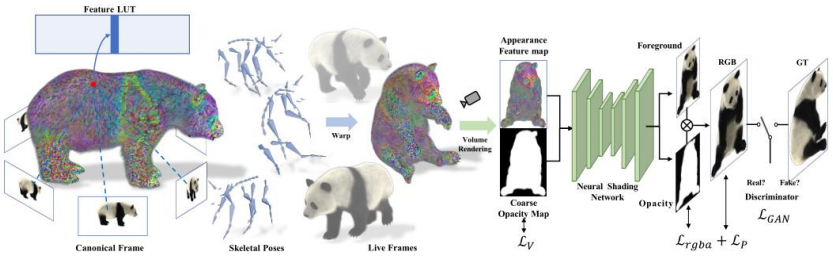

图 4 动画神经动物算法

2.1. 八叉树特征提取

与原始的 NeRF 或 PlenOctree 不同，其中对象的几何形状是未知的，作者将 CGI 动物模型作为输入。因此，作者首先将 CGI 动物角色（例如老虎或狮子）转换为基于八叉树的表示。同样，原始 CGI 模型包含非常详细的毛皮，直接转换为离散体素会导致后续神经建模中出现强烈的混叠和严重错误。如果去除毛皮并仅使用裸模型，体素表示将与实际有显着差异。作者在这里应用一个简单的技巧来解决这个问题：使用来自密集视图集的渲染 alpha matte 作为输入，并进行保守的体积雕刻来构造八叉树：初始化一个统一体积并使用从 alpha matte 扩展的蒙版雕刻它，稍后还=需要渲染的多视图 alpha 遮罩来训练神经不透明度场。生成的八叉树包含一个在 3D 空间中占据的体素数组 P。使用这种体积表示，作者的目标是在每个体素上存储一个与视图相关的特征。因此，可以分配一个称为特征查找表 (FLUT) 的单独数据数组 F 来存储原始 PlenOctree 中的特征和密度值。此处 FLUT 用于有效查询任意 3D 位置的特征，以加速训练和推理。对于体绘制过程中空间中给定的查询点，可以在恒定时间内索引到 FLUT，并为该点分配相应的特征和密度。

2.2. 绑定和变形

为了能够使八叉树适配动画动物，作者只需将前一节提取的八叉树特征绑定到目标骨骼 S 中。要在骨骼运动下装配八叉树，蛮力的方法是根据与骨骼的距离来装配所有体素。在线性混合蒙皮 (LBS) 之后，改为使用 CGI 模型提供的蒙皮网格将蒙皮权重应用于体素。具体来说，给定网格顶点和相应的蒙皮权重，通过混合最近顶点而不是仅一个点的权重来生成每个体素蒙皮权重。具体公式如下：

$$w(\mathbf{p}_i) = \sum_{j=1}^{m} \alpha_j w_j,\ \alpha_j = e^{\delta_j} / \Sigma_k^m e^{\delta_k}$$

其中 w_j 是点 v_j 的蒙皮权重，δ_j 是点 v_j 和体素（立体像素）位置 p_i 之间的欧式距离。

有了以上的结果后，作者可以通过一个转换矩阵来将规范骨骼 S`转换成目标骨骼 S。

$$\mathbf{p}_i^t = \sum_{j=1}^{J} w_j(\mathbf{v}_i) M_j^t (M_j^c)^{-1} \mathbf{p}_i^c$$

其中 M 为转换矩阵。

2.3. 动态体积渲染

到此，作者已经完成了八叉树特征提取并装配绑定到目标骨骼。下一步就是动态体积渲染和端到端训练。

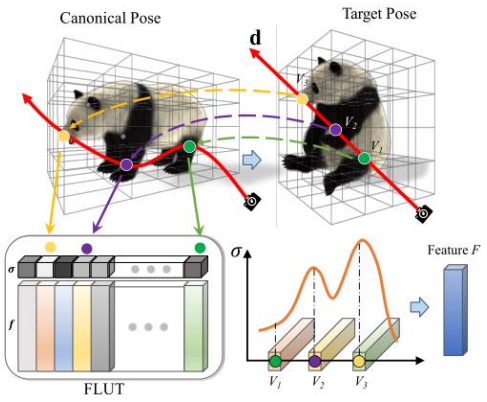

图 5 可微分体积积分方案

作者采用了一种可微分体积积分方案，如图 5 所示。给定一条光线 $r_p^t = (o_p^t, d_p^t)$， o_p^t 为起点， d_p^t 为终点，对应像素 p 和姿势 S_t 。我们可以计算视图相关特征为：

$$F_p^t = \sum_i^m \alpha_i \mathbf{S}(f_i, \mathbf{d}_p^c)$$

$$\alpha_i = T_i(1 - \exp(-\sigma_i \delta_i)), \quad T_i = \exp(-\sum_{j=1}^{i} \sigma_j \delta_j)$$

为了进一步保持视图相关效果的一致性，使用旋转矩阵将光线方向映射到规范空间。最后，通过对每个像素应用该公式来生成与视图相关的特征图 F，并通过沿光线累积来生成粗略的不透明度图 A。

2.4. 神经着色

到目前为止，作者已经将体积光栅化为目标姿势和视点处的神经外观特征图 F 和不透明度 A。最后一步是将光栅化的体积转换为具有类似经典着色器的相应不透明度贴图的彩色图像。

为了保留毛发的高频细节，必须考虑最终渲染图像中的空间内容。作者在 ConvNeRF 之后采用了额外的 U-Net 架构执行图像渲染。请注意，与 ConvNeRF 的基于补丁的策略相比，作者基于光线行进的采样策略支持全图像渲染。准确地说，神经着色网络由分别用于 RGB 和 alpha 通道的两个编码器-解码器分支组成。RGB 分支将 F 转换为具有丰富毛皮细节的纹理图像。Alpha 分支细化粗略的不透明度图 A 以形成超分辨率的不透明度图 A。

最后结合训练，就完成了神经渲染引擎的构建，如下图所示。

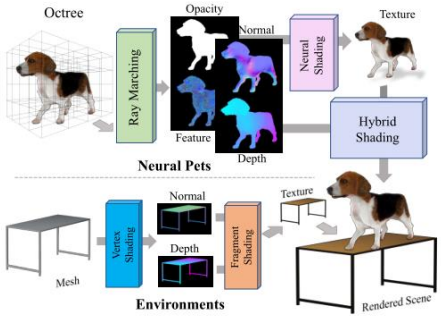

图 6 神经渲染引擎

3.神经动物运动合成

为了让 ARTEMIS 具有运动控制功能，例如，引导动物从一个位置行走或跳跃到另一个位置，必须在动物移动时合成逼真的中间运动。对于运动，常用的方法是数据驱动。因此第一步需要获取一个新的动物动作捕捉数据集。数据采集方面，作者在这里构建了两种类型的动物动作捕捉系统，第一种由一组用于大型动物的 RGB 相机组成，第二种结合了 RGB 相机和 Vicon 相机，用于温顺和小型宠物，如图 7 所示。

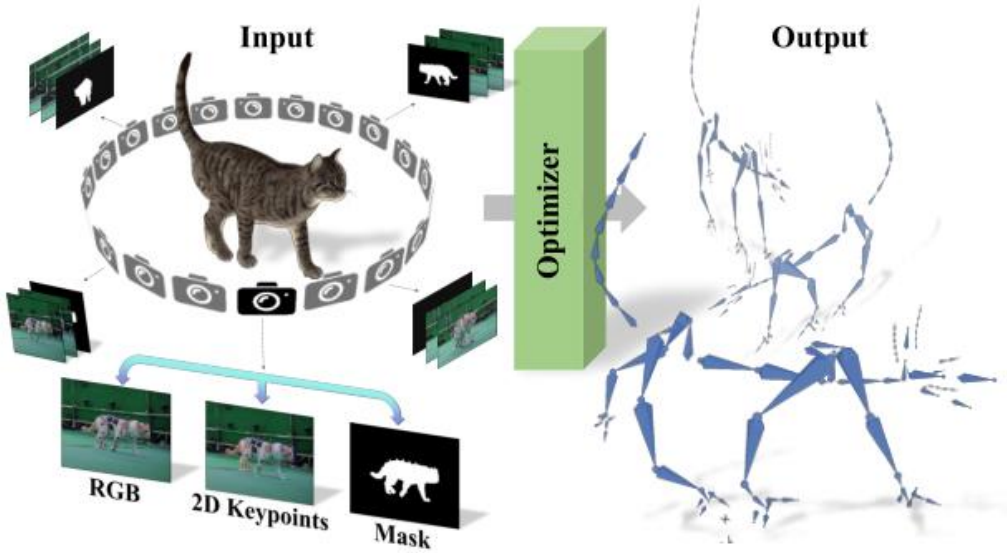

图 7 动作捕捉系统

3.1.动物动作捕捉

3.1.1. 动作捕捉过程

在这个过程中，作者发现尽管四足动物在不同物种之间具有相似的骨骼结构，但它们的形状和规模却截然不同。捕获适用于所有类型四足动物的动作捕捉数据集是根本不可能的。因此，作者开始从温顺的小型宠物中学习运动先验，并将先验转移到老虎和狼等大型动物身上。对于大型动物，作者还与动物园和马戏团合作，构建多视图的 RGB，以进一步提高预测精度。

图 8 动作捕捉系统

图 8 显示了作者团队搭建的的狗和猫等小型动物的动作捕捉系统，其中作者使用 12 台 Vicon Vantage V16 相机均匀分布在动物主体周围，每个相机以 120 Hz 的频率捕捉移动的红外反射标记。作者添加了额外的 22 个 Z-CAM 电影摄影机，它们与 Vicon 交错并以 1920 × 1080 的分辨率以 30 fps 的速度拍摄并在实际运动捕捉之前进行交叉校准和同步。

对于小动物，作者将混合捕捉系统放置在更靠近地面的位置，以便它们能够清楚地获得肢体运动，同时解决腿部之间的遮挡问题。作者团队还聘请了专业的教练指导这些小动物进行不同的动作，包括跳跃和行走等。 Vicon 产生高度准确的运动估计，用作基于 RGB 相机的估计的先验。

对于大型动物，在马、老虎、大象等大型动物上放置标记是不可行的。作者与多家动物园和马戏团合作部署了 RGB 圆顶系统，我们使用 22~60 个摄像头并调整它们的高度和观看方向。同时一些马戏团还向作者团队提供了动物的监控录像，团队手动选择可用的动物并标记它们的姿势。

3.1.2. 2D 关节和轮廓估计

上个部分作者已经获取到了动物动作捕捉的数据集，下一步作者需要自动提取多视图 RGB 图像中的骨骼来进行动物运动推理估计，因此作者需要先估计图像中的 2D 关节和轮廓。

这个部分比较简单，作者运用了已成熟的训练估计模型。对于关节，作者结合了 DeepLabCut 和 SMAL 用于处理四足动物图像数据的模型，并增加了轻量级的手动注释。作者在前 10% 的帧上注释 SMAL 模型中定义的关节，然后允许 DeepLabCut 跟踪和估计剩余帧的 2D 姿势。对于轮廓提取，作者直接使用现成的 DeepLab2 及其预训练模型。

3.1.3. 动物姿势估计

在这个部分作者采用现有的参数化的 SMAL 动物姿势模型。下面根据我的理解，分析一下 SMAL 模型。SMAL 模型可用于解决基于单幅图像的三维动物模型自动生成问题。上一部分已经获取了 2D 关节和轮廓，在这里 SMAL 模型的作用就是利用 2D 关节和轮廓恢复出动物的运

动序列θ_i和γ_i。在这个过程中，为了确保模型的准确性，作者采用了多种办法给模型添加约束，满足约束条件的β,θ,γ被恢复出来，结合所有帧，按照时间顺序排列，就得到了运动序列θ_i和γ_i。

具体分析：SMAL 表示为函数$M(\beta,\theta,\gamma)$，其中β是形状参数，θ是姿势，而γ是平移。$\Pi(x,C_j)$ 表示 3D 点 x 在第 j 个相机上的投影，而 $\Pi(M,C_j)$ 表示 SMAL 模型到相机 j 的投影。作者的目标是从观察到的 2D 关节和轮廓中恢复 SMAL 参数β,θ,γ。作者同时扩展了关键点重投影误差E_{kp}，轮廓误差E_s以确保准确性。这里关键的一步是作者假设已知的动物物种，因此附加了E_β约束β以匹配预训练的物种类型，这样可以确保结果不会偏差太多。对于多视图设置，作者还进一步添加了一个 3D 关键点约束E_{3d}和一个动作捕捉约束E_m为：

$$E_{3d}(\Theta;V)=\sum_k \left\|V_k-\mathbf{Tr}(v_k^i)\right\|_2$$

$$E_m(\Theta;M)=\sum_k \|M_k-\frac{1}{|\Omega|}\sum_{i\in\Omega_k}\mathcal{T}_k^i(\mathbf{Tr}(v_k^i)\|_2$$

其中 Tr(·)是三角测量算子。

对于人体姿态估计，SMPL-X 人体模型引入了先验使用变分自动编码器来惩罚不可能的姿势。在这里作者采用了类似的想法：使用相同的框架在数据集中使用所有恢复的四足动物骨骼姿势对动物运动进行预先训练，然后使用E_{prior}来惩罚与先前的偏差。因此，可以根据优化问题制定动物姿态估计器：

$$\beta,\theta,\gamma\leftarrow\arg\min(E_{kp}+E_s+E_{3d}+E_m+E_\beta+E_{prior})$$

使用所有帧恢复的β,θ,γ，就可以构建动物运动序列θ_i和γ_i。

3.2 动作合成

作者的最终目标是生成虚拟宠物，并且可以与用户进行交互，根据命令做出相应的动作。因此为了实现用户和虚拟动物之间的交互，根据用户命令合成动物动作是至关重要的一部分。

在动作合成部分，作者利用了数据驱动方法和人体运动动画的最新研究-局部运动阶段(LMP)用于动物运动合成，从不同身体部位的异步行为中学习运动。算法可以自动提取 LMP 特征，然后对在动物身上收集的非结构化运动数据进行训练。

3.2.1. LMP 受控运动合成

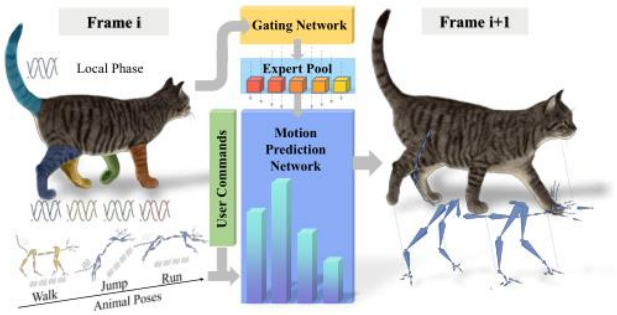

图 9 神经动物控制器

在这里说下我的理解：作者基于局部运动阶段的思想，构建了一个由门控网络和运动预测网络组成的神经动物控制器。门控网络输入 Local Phase 并计算一系列系数。这些系数用于生成运动预测网络。然后，来自前一帧的用户给定的控制信号和运动信息被发送到运动预测网络，运动预测网络使用它们来预测下一帧的运动信息。此外，作者定义控制信号集，通过分析各个状态的速度来判断各个状态。

具体过程：门控网络将当前帧和过去帧的运动状态（关节位置、旋转和速度）作为输入并计算专家系数，然后将其与运动状态动态混合并发送到运动预测网络。从用户那里获取专家权重系数和控制信号，运动预测网络将能够计算未来帧的运动状态。该过程可以表示为：

$$M_i = \phi(M_{i-1}, c)$$

其中 M_i 是帧 i 的运动状态，c 是用户控制信号。

作者将控制信号集定义为 {‘Idle’、‘Move’、‘Jump’、‘Sit’、‘Rest’}。在这些控制信号下，根据动物的脚或臀部与地面的接触状态相应地提取 LMP 特征。通过计算末端执行器和环境对撞机之间的位置和速度差异，还可以自动提取接触标签。我们通过计算根速度并将它们映射到 0 到 1 的值来计算空闲和移动状态。通过联合检查位置、速度和接触信息并将它们映射到 0 之间的值来检测 Jump、Sit 和 Rest，具体来说，Jump 标签没有接触和 y 轴速度，而 Sit 和 Rest 标签没有速度。

3.2.2. 运动转移模型

至此已经完成了动作合成，可以根据用户指令做出相应的动作，下一步需要将 LMP 生成的运动状态绑定到动作捕捉系统捕获的骨架上。需要将 LMP 产生的动作转移到创建的动物模型上，同时保持模型的形状。对于每种类型的虚拟动物，手动应用偏移来约束静止姿势中的旋转和平移分量。然后，应用正向运动学和逆向运动学来计算目标运动状态，并使用变换限制来微调不可能的状态。通过转换后的运动数据，可以实现将运动状态从一个转移到另一个，并驱动神经动物自由移动。

4.沉浸式环境中的神经动物

在前面的部分中，我们已经描述了如何训练我们的动画神经体积表示以在任意姿势和视图下合成动物，以及混合渲染管道。我们的神经运动控制和合成，由我们的动物动作捕捉数据支持，提供了方便的界面来指导动物的运动。在本节中，我们将所有这些模块在 VR 设置下连贯地组装成终极 ARTEMIS 系统，用户可以在其中与动物互动

4.1.互动需求设置

现有运动合成模块通过明确指向目标位置并提供动作类型来引导虚拟动物运动。这里作者进一步探索了高级控制模式，模拟宠物主人常用的一组命令。

• 陪伴：用户可以在虚拟空间中自由移动，虚拟动物将跟随用户。

• 前往：用户指向虚拟空间中的 3D 位置，动物自动到达目标目的地。在存在障碍物的复杂环境中，作者使用 A-Star 算法来寻找动物要遵循的路径。用户还可以控制运动的速度，即动物可以步行或奔跑/跳跃到目的地。

• 绕圈：用户指定一个位置，动物将到达该位置并继续绕圈。用户甚至可以将自己指向目的地并最终被单个或多个动物包围。

• 大小和速度调整：由于我们的神经动物表示，作为一个连续的隐式函数，可以支持无限分辨率，用户可以通过相应地调整其八叉树来调整动物的大小。用户还可以调整移动速度，减慢或加快速度，用户还可以近距离观察动物。

• 自由模式：在没有命令的情况下，动物可以进行任何合理的动作，例如自己探索虚拟世界。实现方法为在动物一个接一个到达目的地的时间间隔内随机设置目的地。

4.2.实验配置细节

在 VR 设置下产生神经动物的整个控制流程如下：

1. 从 OpenVR 中获取 VR 耳机的姿势以及控制器状态。

2. 使用 LMP 控制信号和当前状态来生成估计的运动参数。基于这些参数，使用 LBS 装配规范模型，然后为每一帧构建一个八叉树。

3. 神经渲染管道追踪到八叉树以生成特征图并随后进行渲染，其中背景环境、3D 环境中的障碍物等使用标准图形管道渲染并与神经渲染结果融合。

在将训练有素的神经动物部署到系统时，作者还采用 LookinGood 以自我监督的学习方式微调训练后的模型，以获得更好的左右眼视图一致性，避免 VR 眼睛之间视线不一致可能会导致恶心和头晕，考虑十分周到细节。

5.结果展示

在本节中，作者将 ARTEMIS 在各种具有挑战性的场景下实验以评估系统的性能。作者将神经渲染方案与当前最先进的 (SOTA) 方法进行了比较，用于动态外观渲染和静态不透明度生成，作者的方法更好地保留了高频细节。作者还对我们的渲染方法进行了详细的运行时性能分析，并为我们的动作捕捉方案提供了额外的评估。同时进一步使用消费级 VR 耳机提供 ARTEMIS 的 VR 应用程序，用户可以在其中与各种虚拟动物亲密互动。最后，讨论方案的目前局限。

5.1.动态神经渲染的比较

在这里，作者将 ARTEMIS 中的神经渲染管道与最近用于动态场景渲染的 SOTA 方法进行比较。作者比较了两种基于辐射场的方法，包括 NeuralBod 和 AnimatableNeRF，以及基于体积的方法 NeuralVolumes。由于 ARTEMIS 的 NGI 动物需要一个额外的预定义骨骼装备，该骨骼装备具有来自相应 CGI 数字资产的剥皮权重。因此，为了公平比较，其他算法使用与 ARTEMIS 方法相同的实验设置来训练基线模型，并采用 ARTEMIS 的 CGI 动物模型的地面实况网格顶点和蒙皮权重来训练 NeuralBody 和 AnimatableNeRF。

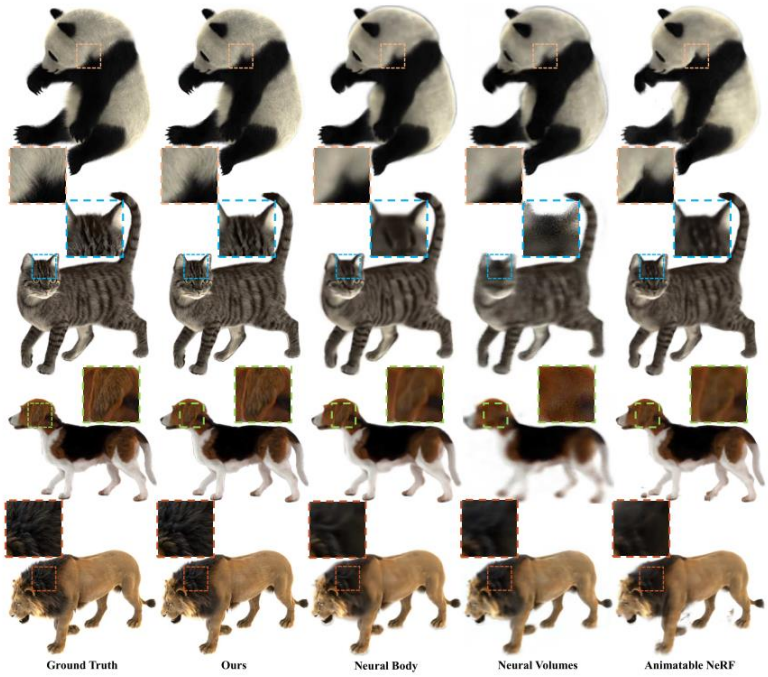

图 10 效果比较

图 10 提供了我们的方法与上述三种方法的比较结果。只有颜色与背景相差很大的毛皮才能被识别为毛皮，例如狮子鬃毛和猫的额头。这三种基线方法会出现毛皮模糊的伪影和外观细节的丢失。具体来说，NeuralVolumes 由于其有限的建模能力而出现严重的噪音和模糊效果，这在猫的额头上最为明显。NeuralBody 只能生成低频毛皮细节作为模糊效果，而 AnimatableNeRF

表现稍好但仍不能真实地恢复毛皮细节。与此形成鲜明对比的是，作者团队的方法很好的地生成了更清晰的细节，特别是对于动物中常见的那些毛茸茸的区域。对于定量评估，我们采用峰值信噪比 (PSNR)、结构相似度指数 (SSIM) 和学习感知图像块相似度 (LPIPS) 作为指标来评估渲染精度。如表中所示。如表 1 所示，作者的方法在上述所有指标上都显着优于其他方法，说明作者方法在保留渲染细节方面的优越性。

表 1 不同渲染方法的定量比较

Method		Neural Body	Neural Volumes	Animatable NeRF	Ours
Panda	↑ PSNR	30.38	30.11	26.51	**33.63**
	↑ SSIM	0.970	0.965	0.957	**0.985**
	↓ LPIPS	0.110	0.116	0.112	**0.031**
Cat	↑ PSNR	30.77	28.14	31.37	**37.54**
	↑ SSIM	0.972	0.951	0.973	**0.989**
	↓ LPIPS	0.067	0.087	0.061	**0.012**
Dog	↑ PSNR	32.37	26.80	31.19	**38.95**
	↑ SSIM	0.978	0.945	0.975	**0.989**
	↓ LPIPS	0.075	0.129	0.074	**0.022**
Lion	↑ PSNR	30.11	29.59	27.87	**33.09**
	↑ SSIM	0.956	0.947	0.944	**0.966**
	↓ LPIPS	0.111	0.123	0.123	**0.035**

5.2.静态 RGBA 渲染的比较

作者将论文中的方法与最近针对静态场景的不透明渲染任务的 SOTA 方法进行比较。作者选择了三个方法作为基准，包括基于显式点云的方法，称为神经不透明度点云 (NOPC)，以及基于辐射场的方法 NeRF 和 ConvNeRF。

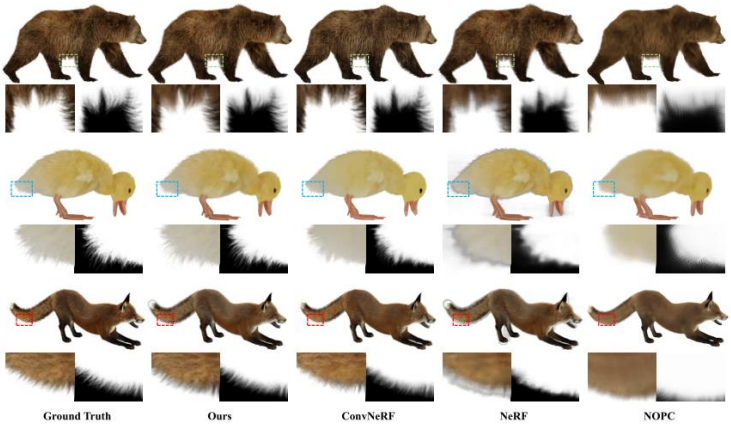

图 11 比较结果

图 11 显示了作者的方法和其他静态方法的几个 RGB 图和 alpha 图的结果。由于基于点云的插值，NOPC 在边界区域存在混叠和重影。由于其 MLP 网络的表示能力有限，特别是高频细节，NeRF 遭受严重的模糊伪影，而 ConvNeRF 改善了 alpha 和 RGB 细节，但由于基于补丁的训练策略在有限的稀疏训练视图上仍然会导致网格状伪影。相比之下，作者的方法实现了最佳性能，作者的方法可以使用来自其他帧的视图来补偿特定帧的缺失视图。

在这里补充一下 RGB 和 alpha 在计算机图形学中的概念：在计算机图形学中，一个 RGB 颜色模型的真彩图形，用由红、绿、蓝三个色彩信息通道合成的，每个通道用了 8 位色彩深度，共计 24 位，包含了所有彩色信息。为实现图形的透明效果，采取在图形文件的处理与存储中附加上另一个 8 位信息的方法，这个附加的代表图形中各个素点透明度的通道信息就被叫做 Alpha 通道。Alpha 通道使用 8 位二进制数，就可以表示 256 级灰度，即 256 级的透明度。白色（值为 255）的 Alpha 像素用以定义不透明的彩色像素，而黑色（值为 0）的 Alpha 通道像素用以定义透明像素，介于黑白之间的灰度（值为 30-255）的 Alpha 像素用以定义不同程度的半透明像素。因而通过一个 32 位总线的图形卡来显示带 Alpha 通道的图形，就可能呈现出透明或半透明的视觉效果。

表 2 定量比较

RGB		NOPC	NeRF	ConvNeRF	Ours
Bear	↑ PSNR	18.43	28.12	**32.34**	30.95
	↑ SSIM	0.886	0.954	0.953	**0.967**
	↓ LPIPS	0.140	0.113	0.063	**0.038**
Duck	↑ PSNR	25.45	30.35	34.31	**37.14**
	↑ SSIM	0.967	0.978	0.985	**0.986**
	↓ LPIPS	0.075	0.091	0.052	**0.026**
Fox	↑ PSNR	17.42	27.53	**33.42**	30.94
	↑ SSIM	0.914	0.966	0.973	**0.976**
	↓ LPIPS	0.106	0.099	0.047	**0.029**
Alpha		NOPC	NeRF	ConvNeRF	Ours
Bear	↑ PSNR	17.89	31.65	36.37	**40.13**
	↑ SSIM	0.918	0.986	0.992	**0.995**
	↓ SAD	144.2	199.2	11.80	**8.072**
Duck	↑ PSNR	19.77	30.09	33.02	**36.81**
	↑ SSIM	0.849	0.923	0.990	**0.994**
	↓ SAD	110.6	36.17	12.76	**8.558**
Fox	↑ PSNR	15.68	23.81	34.90	**36.43**
	↑ SSIM	0.903	0.968	0.993	**0.995**
	↓ SAD	192.4	52.32	11.30	**9.555**

作者的方法生成更逼真的不透明度渲染，甚至实时支持动态场景。相应的定量结果在表 2 中提供。对于 alpha 图的评估，除了 PSNR 和 SSIM 之外，我们进一步采用绝对距离之和（SAD）作为指标。我们的方法优于所有与 alpha 相关的指标的方法，并保持了与 ConvNeRF 的 RGB 纹理渲染相当的性能。由此可见作者的方法在高质量毛皮和不透明细节渲染方面的有效性。

5.3.消融研究

最开始看着一部分懵懵懂懂，不明白这个研究是为了什么，查找资料后有了一定的理解，在这里写出来：“消融研究”这一术语的根源于 20 世纪 60 年代和 70 年代的实验心理学领域，其中动物的大脑部分被移除以研究其对其行为的影响。在机器学习，特别是复杂的深度神经网络的背景下，已经采用“消融研究”来描述去除网络的某些部分的过程，以便更好地理解网络的行为。**说白了就是设立对照组的意思，通过去除某个模块的作用，来证明该模块的必要性，如果消融实验后得到结果不好或者性能大幅下降，说明该模块起到了作用。**

图 12 无 NS、无 GAN 和 无 VRT 模块的消融研究可视化

这部分作者对“熊”数据进行了广泛的消融研究。 w/o VRT 和 w/o GAN 表示在没有 VRT 和 GAN 模块的情况下训练的模型。图 12 显示 GAN 损失有助于保留更多高频细节并合成更清晰的 alpha 图。VRT 在保持时间一致性的同时导致略微平滑的纹理。作者进一步通过体积直接渲染 RGBA 图像训练模型，而无需神经着色网络 (NS) (w/o NS)。由于几何形状不对齐和有限的体积分辨率，它在腿部和腹部周围存在噪音和伪影。表 3 中的定性结果充分展示了每个模块的贡献。

表 3 对无 NS、无 GAN 和无 VRT 模块的消融研究进行定量评估

Models	*RGB*			Alpha		
	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	SAD ↓
(a) w/o NS	29.47	0.939	0.152	29.27	0.981	24.68
(b) w/o GAN	34.29	0.964	0.051	35.84	0.991	12.47
(c) w/o VRT	33.81	0.961	**0.044**	35.74	**0.992**	12.23
(d) ours	**34.43**	**0.965**	0.045	**36.18**	**0.992**	**11.92**

5.4.运行时间性能评估

表 4 运行时间对比

Dynamic	CGI	Neural Body	Neural Volumes	Animatable NeRF	Ours
runtime (ms)	$\sim 5 \times 10^5$	2353	181.7	18142	**34.29**
fps	–	0.425	5.504	0.055	**29.16**
Static	CGI	NOPC	NeRF	ConvNeRF	Ours
runtime (ms)	$\sim 5 \times 10^5$	51.23	18329	2599	**20.38**
fps	–	19.52	0.055	0.385	**49.07**

作者团队的算法在 Nvidia TITAN RTX GPU 上运行。表 4 比较了作者的方法与其他方法运行一帧时间。对于动态场景，将完整运行时间与其他方法进行比较。我们的方法实现了实时新颖的姿势合成和渲染。对于静态场景，比较了自由视图渲染运行时间。显然作者的方法比其他方法快得多（大约 10 倍），尤其是实现了传统 CGI 动物角色的三到四个数量级的渲染速度，这进一步说明 ARTEMIS 系统的优越性。

5.5. VR 中的交互式 NGI 动物

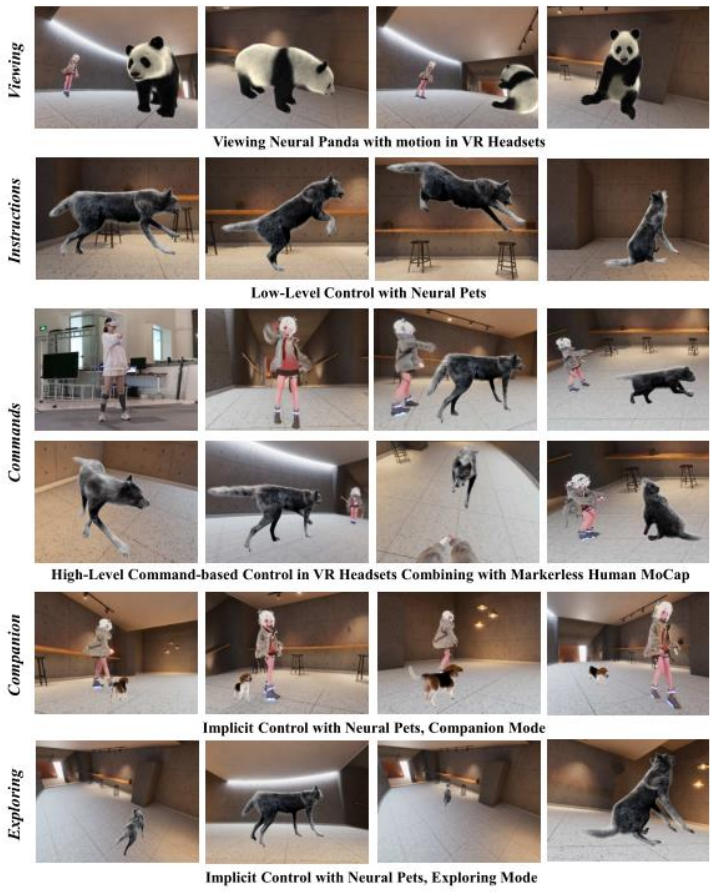

图 13 交互结果

如图 13 所示，展示了在 VR 应用中的渲染结果，包括不同级别的交互和不同的视角。

“viewing”分别显示了从第三个视图和第一个视图（VR 耳机）观察到的栩栩如生的虚拟熊猫。即使在 VR 耳机中，也可以清楚地看到良好的毛皮效果。

“Instructions”显示低级控制动画结果。可以使用“跳跃”、“移动”和“坐下”等控制信号明确地驱动的神经宠物。

“Commands”说明了高级控制模式“Go to”。用户指向虚拟空间中的 3D 位置，狼会自动到达目标目的地，甚至还挥手把狼叫回来，同时移动的速度也由用户控制。

在“Companion”中，虚拟宠物会像真正的宠物一样跟随并陪伴用户。

“Exploring”展示了自由模式，在没有命令的情况下，动物可以做出任何合理的动作，自己探索虚拟世界。图 14 中展示了更多的交互细节。

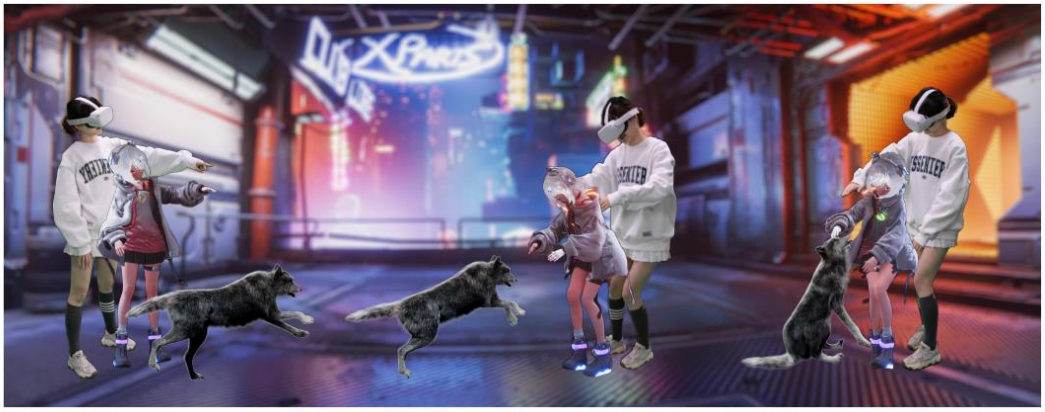

图 14 交互细节

5.6 局限

- 过度依赖于相应 CGI 动物角色的预定义骨骼结构和蒙皮权重

尽管作者的方法在渲染时间加速方面有了相当大的改进，但目前的方法并不能从捕获的数据中全自动生成神经动物。

- 与 NGI 动物的互动严重依赖人类动作捕捉和预先定义的基于规则的策略

目前的 ARTEMIS 设计仅基于几种相对基本的交互模式。这种基本交互模式规则需要预先设置。尽管 ARTEMIS 拥有前所未有的交互体验，但迫切需要一种更先进、基于知识的 NGI 动物与人类之间的交互策略。此外，需要更多的工程努力来为当前的 VR 应用程序提供更身临其境的体验，尤其是支持多种动物的渲染。

6.结论

本文提出了一个名为 ARTEMIS 的神经建模和渲染管道，用于生成具有绒毛外观和运动合成的神经宠物。

ARTEMIS 系统具有交互式运动控制、逼真的动画和毛茸茸动物的高质量渲染，并且具有实时性。ARTEMIS 的核心是神经生成 (NGI) 动物更新了传统 CGI 动物角色的渲染过程，可以生成具有丰富的外观、皮毛和不透明度细节的实时照片级渲染结果，提升了三到四个数量级量级的加速。

此外，混合渲染引擎能够将 ARTEMIS 集成到现有的消费级 VR 耳机平台中，从而为用户提供超现实的沉浸式体验，与虚拟动物进行亲密互动，就像在现实世界中一样。广泛的实验结果和 VR 展示展示了神经动物建模和渲染方法的有效性，支持前所未有的沉浸式交互，更新了人类感知和与虚拟动物互动的方式，提供更加身临其境、逼真和响应迅速的互动体验，在动物数字化和保护或虚拟现实/增强现实、游戏或娱乐。