**Real-Time Hair Simulation with Neural Interpolation**

**IEEE TRANSACTIONS ON VISUALIZATION AND COMPUTER GRAPHICS 2020**

# 基于神经插值的实时头发仿真

1. **论文背景介绍**

发型是时尚的重要组成部分，它反映了现实生活中人类的个性。随着AR/VR应用的日益普及，头发的数字化成为3D化身建模的一项重要工作。通过3D合成技术的最新进展，各种各样的发型可以从单个图像忠实地捕捉，但仍然满足不了人们对头发模型的可交互性的需求。为了追求头发动画的真实感，许多基于物理的方法被采用。尽管这些方法可以生成高度逼真的细节，但由于计算成本高，它们不能直接应用于交互式应用程序。为了简化头发模拟的计算，人们常常采用启发式模型简化复杂的头发与头发之间的交互作用。最近Chai等人提出了一种数据驱动的方法，成功以交互式的速度生成了视觉上逼真的头发。他们将先通过图划分确定引导头发，然后确定插值的权重集，这导致每个新的发型都要重新训练一组权重集。作者通过实时计算引导发的状态，并将此作为参数输入一个卷积神经网络，来预测全体头发的状态。此外，为了增强头发细节的真实感，又添加了一个生成式对抗神经网络用来生成头发精细尺度的位移，并将其叠加到主干网上，最后输出最终结果。

该文章的创新点在于，第一个使用神经插值框架对不同类型的头发进行差值仿真，相比计算昂贵的方法，该文章将生成速度提高到可交互的速率。除此之外，能够动态预测插值的权重，同时保证精度的鲁棒性。最后，还应用了一个位移生成器，用来产生头发的细节。

1. **方法和实验**

**头发状态参数表示**

作者将每根头发视作绑在一个头皮曲面的n个粒子，每根头发都有一个根粒子分布在头皮曲面的上。如果用UV图来表示这个曲面，那么每根头发的根位置都是一个二维坐标，每根头发有n个粒子，在这篇文章里作者将n取为25，且每个粒子存储他们的三维欧式坐标。至此，每根头发都可以表示成一个四维张量，而整个头发的状态可以表示为在一个三维网格的函数。图四展示了上述表示方法的基本思想。

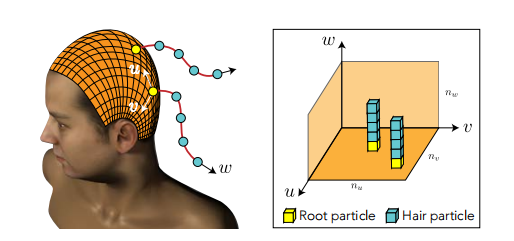


图 1 头发状态的参数化表示方法

**神经插值**

利用前一步的张量的毛发状态表示，利用机器学习研究的最新进展，作者提出了一种神经毛发状态插值器。作者在这一步，选择了网格上均匀分布的一些点作为引导发，实时计算他们的状态，再通过他们的状态进行神经插值，计算得到普通头发的状态，如图2所示。每根普通头发选取它周围相邻的16根引导发作为插值的数据。在途中，蓝色点为待插值的头发点，红色的16个点是该插值点的根据的引导发。

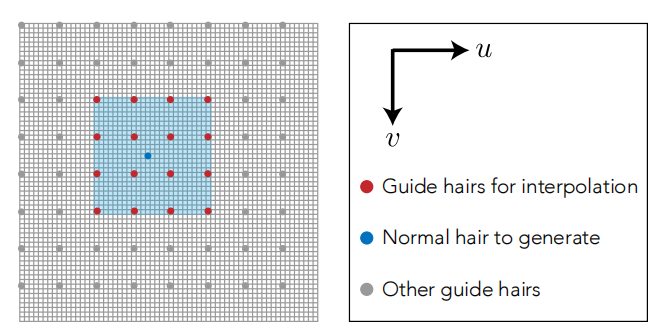


图 2 引导头发插值方法

**网络架构**

这篇文章里所使用的网络架构大概分为两个部分。一是一个多层的特征提取网络，用来生成插值的权重，二是一个线性插值器。当经过第一部分生成插值的权重后，第二部分使用这些权重进行插值，如图3所示。在第一层的结束我们生成了一个16层的三维张量。这个十六层对应了与周围16个引导之间的权重。再输入引导发的网络实时状态值，得到最后的输出。

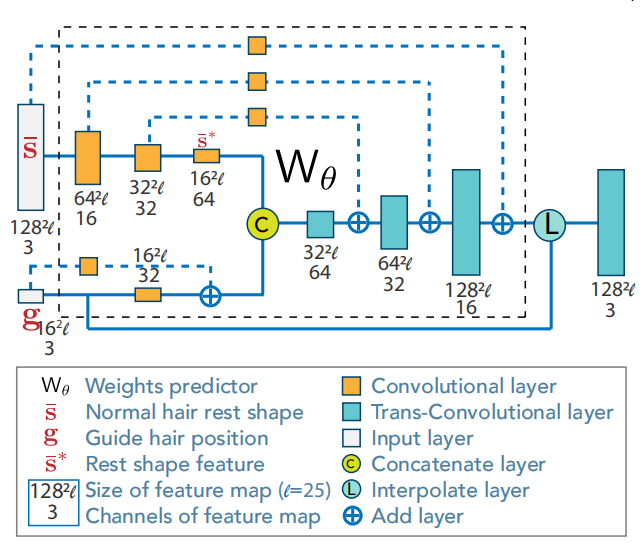


图 3 神经网络结构图

此外，我们的第一部分网络是一个轻量级的残差网络，增强网络学习的稳定性。我们在第一层输入普通和引导头发的状态数据，然后进行降采样操作，降到一定程度时且当普通和引导发的分辨率相同时，将他们连接起来，然后再做上采样操作，知道恢复到原来的分辨率。在这个网络中LeakyReLU被用作隐藏层的激活函数。网络末端使用softmax对插值权值进行归一化。使其满足权值的约束，这样的网络结构可以对各种不同空间的发型进行训练。

**损失函数**

为了训练网络参数至最优的状态，我们采用了如下的损失函数：

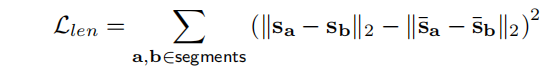
**1638618025(1)**

他由以下四部分组成，分别是原始数据回归项Ldata、头发长度保持项Llen、引导头发模式捕捉项Lsnap和时间相干项Ltempo，接下来我将从四部分来介绍损失函数。

对于每一帧，我们惩罚基真值状态ˆs与神经插值器生成的正常毛发状态s之间的偏差的L1范数，这保证了足够接近的数据重构，公式如下所示：



由于一根头发在身体上很容易弯曲，但几乎不可拉伸，我们惩罚每一根正常头发的长度变化相对于上一状态，头发长度惩罚项如下所示,其中a和b分别是发丝段的起始点和结束点:



如果我们假设，头发插值完全基于重心坐标插值，那么会出现一些穿模、伪影的问题，如图4所示。

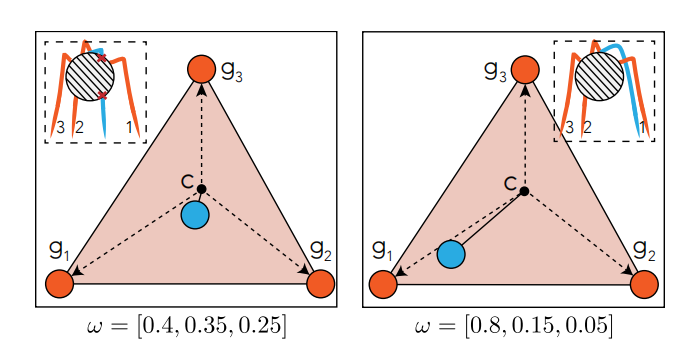
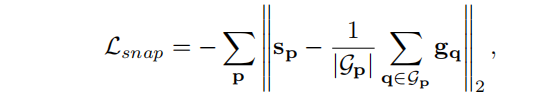
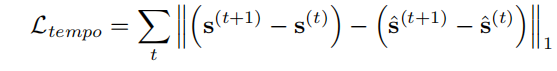


图 4 左图为重心坐标插值，右图是我们希望的插值结果

所以我们希望避免出现这种问题，头发的状态应该是偏向他周围引导发的某一种形式，而不是所有的引导发做平均，所以作者在损失函数里引入了一项惩罚项，如下所示。如果Sp的值越接近与之相邻的插值头发的位置那么该项的值就越大。



为了避免残影，作者还引入一项时域上变化不一致的惩罚项，如下所示，计算前后两个状态的变化的差值，使神经插值的估计在时间的变化上与真实值保持尽可能地一致：



**GAN增强细节**

上述神经插值器对中等分辨率的毛发产生令人满意的结果。然而，当正常头发的分辨率上升到128\*128时，由于我们一直使用16\*16引导头发，头发细节逐渐恶化。在这种情况下，每根导丝的支撑区域会变大，这必然会导致正常头发更平滑的插补。因此，我们引入了一个精细尺度的位移生成器来增强神经插补器输出的真实细节。根据我们的观察，只有当分辨率高于128\*128时，该生成器才会在视觉上产生相当数量的毛发细节。

对抗神经网络都分为两部分，本文的生成器能产生精细尺度的位移，并将其添加到神经插值器的输出。相加后的结果被放入鉴别器进行对抗性训练。具体的架构如图5所示。

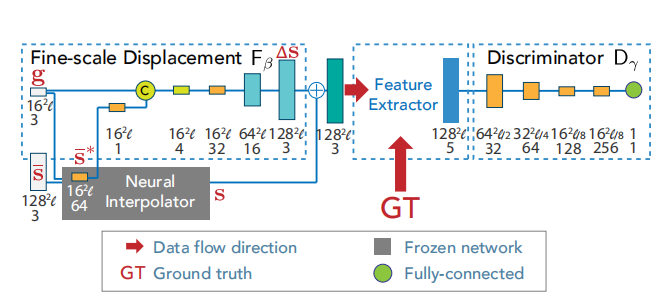


图 5 对抗神经网络架构图

1. **结果与应用前景分析**

作者基于上述方法，训练了一组神经网络参数，并在测试中，算法表现出了优越的结果。首先他们检验了各个损失函数的损失项对于整体算法的效果的影响。如图6所示。

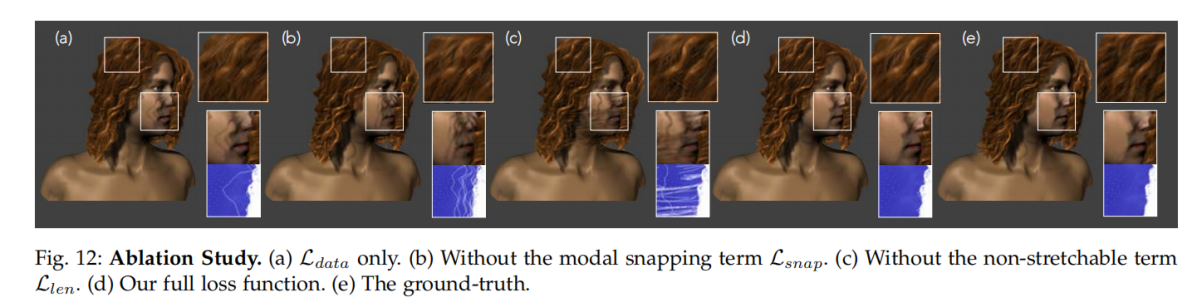


图 6 结果分析图，最左侧只有真实数据回归项，结果呈现了有大量的伪影。第二幅图剔除了snap项，导致出现了大量穿模的伪影。第三幅图剔除了长度的惩罚项，出现了很多形状怪异的头发，第四幅图是作者完整的结果，显示与第五幅图经过长时间计算得到的正确值基本接近。

作者还给出了基于他们的算法的更多结果，如下图7所示。这些结果均是在实时、可交互的速率得到的，如表1所示，质量和速度都展现出了相较以往方法更好的结果。他们所使用的PC设备由 Intel Core i7-4790、CPU and an NVIDIA GTX 1080 GPU组成，他们所使用的头发模拟系统是基于C++开发的，并且用到了一些并行计算模块。



图 7 头发实时模拟结果

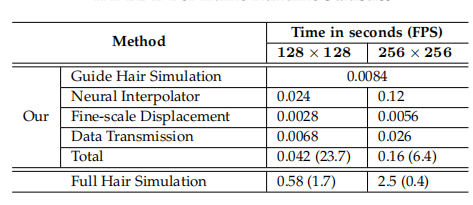


表 1 每一次计算所需要耗费的时间

这篇文章基于一种新的神经插值的框架，能够以实时速率进行头发模拟，同时生成高质量的头发模拟效果。这一成果可以在诸多领域得到使用，例如基于该方法开发DCC软件插件，给设计师设计不同发型时，能够实时的得到发型的运动表现，降低传统美术工作流的成本。此外，还可以用于游戏行业中，解决人体头发实时计算的难题，提高游戏的可交互性和实时效果，提高游戏沉浸感。最后，这样的技术可以运用到当下大火的虚拟数字人、元宇宙中，为用户呈现更加真实的人物形象。综上所示，此研究的运用场景相当之广，人体头发模拟一直以来就是图形学界一个具有高热度的话题，能够解决这一领域的文章除了具备巨大的学术价值外，对于整个图形学产业界都具有深远的意义，且蕴含巨大经济回报。