

天文数据可视化动画技术研究

席萌

北京天文馆

摘要 本人从参加工作起长期从事特种电影动画技术工作，参与《迷离的星际》、《天上的宫殿》、《宇宙少年侦探团》、《大鸟探险记》、《小强的故事》、《奇妙的星空》、《星空音乐会》、《UFO与外星人》、《黄道十二宫》等十余部特种电影的制作，均担任技术负责人职务。本文通过个人十余年的动画制作经验，结合多年来的特种科普影片制作历程，介绍天文可视化动画技术的演进、现状、新的突破及预期发展方向。我团队从技术流程到代码、素材等，均坚持自主研发，取得了软件、专利等成果，在近5年制作完成的特种电影《撞击与生命》、《奔向月球》、《走近黑洞》、《宇宙大爆炸》均已得到应用。本文重点以海量天文数据可视化的典型应用项目《宇宙大爆炸》为案例背景，介绍动画制作中遭遇的超大数据量可视化建模和动态过程插值等诸多难点，介绍解决思路 and 解决过程，并简介利用虚幻引擎4开发GPU动画渲染软件方面取得的新突破，基于GPU算法的引擎动画技术可将天文可视化动画渲染加速上百倍，新技术的突破为未来实现更大规模、更多类型的天文数据可视化奠定了基础。

关键词 天文 数据 动画 粒子 GPU 虚幻引擎

一、技术背景

本人长期从事特种电影行业，包括球幕、环幕、巨幕以及立体（3D）、动感（4D）等多种影片类型，为我馆以及国内外众多场馆提供特种片源。其中球幕是天文馆、科技馆行业的标配和核心场馆。用球幕展示天象在世界上已有数十年传统，进入本世纪，天文馆从机械的天象仪步入了数字电影时代，数字三维动画渐成主流，天象影片的表现内容也从简单的坐地观星进入不受时空限制的广阔太空。动画技术在为球幕天象影片带来更丰富的表现力的同时，也迎来了更多的挑战。

创造真实可信的宇宙场景，对天文科普影片来说异常重要。我们的宇宙，不仅仅是夜空中闪烁着或明或暗的光点，还有时刻精准的日出日落、月盈月亏，还有各种复杂轨道上运行的深空天体，有围绕银心旋转的银河系悬臂，以及由星系团构成的更宏伟壮丽的宇宙大尺度结构。

以上种种场景，都是由科学数据勾画出来的，所有天体的运行都由数据驱动，不容出错。每一帧动画，都要经过科学顾问的指导和学术委员会的审定，保证了内容的科学性，最终才能推向观众。对于天文科普影片来说，数据是最重要的素材来源，也是必须攻克的难关，天文科普影片的表现力，在于如何更好的实现天文数据可视化。

正因为其技术难度，天文科普影片的现状并不容乐观，国内大多数球幕场馆运行的都是美国 Skyskan 或 E&S 的天象演示软件，电视和网络上天文类纪录片中壮观的宇宙演化模拟基本全都出自国外团队之手，连国家天文台面向公众的科普工具也主要利用微软的 WWT 平台……虽然这种领域不太可能出现“卡脖子”状况，但若数据和技术均不操之于我手，制作过程就会受到技术路径、素材资源、软件平台、知识产权等多重掣肘和依赖。作为国内特种影片的先行者，我馆的数字工作室团队长期以来始终贯彻并坚持着自主开发的路线。

二、技术演进

1. 可视化动画技术流程

对于天文场景，通常以粒子模拟远距离发光天体，用流体模拟星云、星系、星际物质等，但通用技术并不完全适用于天文场景。

通用的粒子动画主要依靠三维软件的动力学解算机制，动画师无需手动操控每个粒子的运行轨迹和形态，而是通过合理设置动力学参数、添加力场或约束，利用软件内置的解算器及其插值算法创造细节。但天文场景的所有数据都是从科学观测或计算而来，每颗粒子，从初始化数据到动态变化过程，都来自真实数据，解算器仅能近似模拟，而无法保证科学性。

从十余年前制作第一部影片《迷离的星际》起，为保证科学数据的精度，我们舍弃了 Maya 软件内便利的动力学工具，采取了全程以程序控制粒子的做法。此后逐渐形成了工作方法与流程，沿用至今，并在制作新的影片时逐年优化和扩充。

以下是我们制作天文可视化动画的主要流程：

- 1、采集世界各大天文科学机构定期更新的星表数据，导入 Maya；
- 2、根据每个项目需求，扩展开发相应的天文计算函数库，如天文坐标系转换、椭圆/抛物线/近抛物线轨道计算、拟合曲线计算、摄动计算、相对论计算等等；少数算法需迭代计算，为提高计算效率则用 C++ 开发外部函数库；
- 3、创建粒子系统，把天文星表数据写入每粒子（PerParticle）属性；创建表达式（Expression）调用天文函数库，按照科学计算驱动粒子坐标及外观属性等；
- 4、初始化天文函数库和资源库的一系列环境变量，将科学模型引用到主场景，定制动画，提交集群渲染；
- 5、项目结题后，将影片所创建的天文相关的科学模型、素材、代码归纳整理，统一素材格式和编码规范，扩充天文资源库。

以上流程基本应用于每一部天文科普影片，最终也内置于我们自主开发的软件工具中。

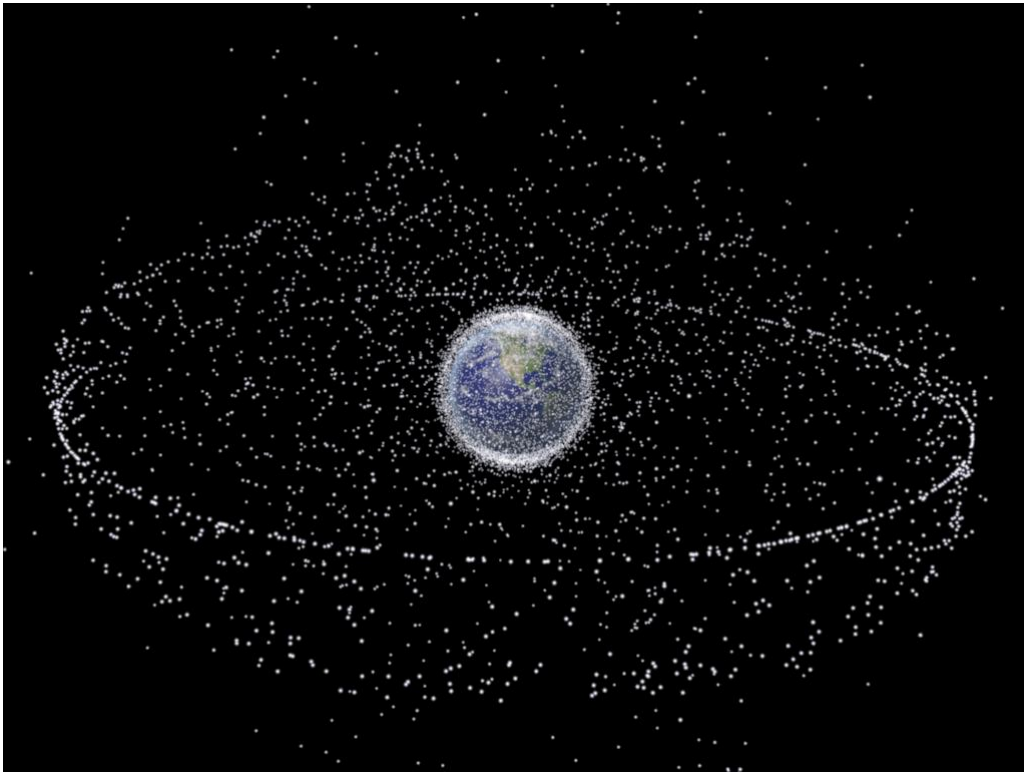


图 1 程序化创建的粒子系统示例：地球人造卫星

2. 发展历程

作为小型动画团队的技术负责人，天文数据、素材和技术开发都要亲力亲为，无法一蹴而就，都是在解决影片制作问题的长期实践中逐渐积累而成。

以下是前十年的技术积累历程：

时间节点	影片	技术开发内容
2006	《迷离的星际》	利用依巴谷星表创建恒星和西方星座系统； 解决球幕多通道合成渲染技术；
2009	《天上的宫殿》 《宇宙少年侦探团》	创建中国古天文星官体系； 开发特种电影集群渲染与合成技术；
2011	《奇妙的星空》	创建太阳系行星、卫星轨道系统
2013	《UFO与外星人》	创建太阳系小天体轨道系统； 包括小行星、彗星、人造卫星等；
2012-2015	《星空音乐会》 《每月观星指南》 《黄道十二宫》	开发基于Maya Mel的通用天文计算函数库； 开发动画渲染命令集。

图 2 技术开发

2016 年将以上可视化资源和技术汇总为软件《太微数字宇宙系统》，渲染技术封装为《太微云渲染管理软件》，并继续扩展。

到 2018 年初已形成包含数十万天体的通用天文科普资源库，足以搭建成为功能完整、自成一体的虚拟数字天象仪，支撑以后的天文影片开发，并形成 2 个专利：用于可视化建模的《一种天球映射方法及装置》和用于球幕渲染的《一种虚拟球幕的创建方法和装置》。

某些天文场景不需要完全写实，但仍需要科学性参考。由于非天文专业的动画师对天文物理均缺乏概念，需要三维模型作为更直观的参照，以上资源能够提供很多帮助。例如我在 2018 年的球幕电影《走近黑洞》中，通过修改已有资源，快速定制虚拟小行星带示例、银心附近虚拟天体轨道示例等模型，为动画团队制作虚拟星系的天体运动场景提供了参考。

从 2018 年起，为提升渲染效率，我开始研究“虚幻引擎 4”（Unreal Engine 4），并将以上天文可视化资源逐渐移植到虚幻 4 平台，移植工作持续至今。

虚幻 4 在辅助制作方面已发挥重要作用，例如已利用虚拟球幕专利技术为《走近黑洞》完成球幕制式转换，开发和测试引力透镜着色器材质特效等；为《撞击与生命》和《奔向月球》进行了 VR 环境镜头测试；在《宇宙大爆炸》项目中，使用虚幻 4 的 Niagara 粒子系统进行快速颜色映射模拟，加速了材质开发进度；今年正在进行部分可视化镜头的改良重制。虚幻 4 已开始成为重要的生产力工具。

三、技术现状

当通用天文可视化资源建立完善之时，我们在 2019 年启动的新球幕影片《宇宙大爆炸》中又遇到极限挑战。影片涉及大量星系尺度甚至宇宙大尺度结构的动态演化场景，国内动画行业没有可供参考的经验，国内合作过的动画公司也都无法提供解决方案。

为此我们聘请了武向平院士为总顾问，聘请中科院国家天文台计算宇宙学研究团组的王杰研究员、景英杰博士进行数据支持和指导，获得了基于国际宇宙学模拟项目“室女座联盟”子项目 AurigaProject 的海量数据，该项目在超算集群中通过大型高分辨率磁流体动力学算法生成数据，模拟数以万计的银河系大小的星系演化，暗物质与气体在引力作用下逐渐聚集、碰撞融合为星系团，最终勾勒出大尺度结构的宇宙“骨架”。时间上从宇宙大爆炸 30 万年后直至今天，跨越 138 亿年，空间尺度达到 3 亿光年。

这批数据包含 128 个时序数据快照（Snapshot）文件，平均每两个快照间隔约 1 亿多年。每个快照记录宇宙的一个瞬时状态，包含数千万条粒子数据，有三种粒子类型：暗物质、气体、恒星。每条数据记录了该粒子在此快照时刻的三维坐标、速度矢量、密度、内能、体积等参数。所有快照的数据总计达到数十亿条。

对本次可视化动画任务来说，超大数据量建模和动态过程重建是两大难点，动态重建尤

为困难，相当于把 128 张拍摄于各个时期的宇宙的照片还原成连续的动画，需要大量的数学推算。

我把问题拆解开来逐步推进：

第 1 阶段：数据分析

利用 Python 脚本在集群上分析数据，计算每个参数的数量级、极值、均值、方差等，生成分布图；利用 MATLAB 进行曲线拟合，将数量级差异过大的数据映射到 Maya 能够处理的数量级范围内；对数据进行分组和排序，等等；

在得到部分数据分析结果的同时，后续可视化建模也已同步展开。此阶段耗时超过 2 个月；

第 2 阶段：局部建模

从其中一个快照提取中心部位约二百万条数据。此步骤仍按照先前总结的可视化技术流程完成，分别创建了暗物质和物质的局部静态模型。

最终得到的暗物质模型为团块聚集形态；气体和恒星模型为类似银河系形态，范围约数十万光年。经王杰老师认可，确认了静态模型的科学性。此模型已用于渲染宣传海报：



图 3 暗物质（左）与中心星系（右）局部静态模型

仅验证中心区域是不够的，另外还提取了宇宙多个区域，降采样局部建模进行科学性验证，限于篇幅不一一贴图。

第 3 阶段：动态插值

每两个快照之间需要通过插值算法推算任意时刻的中间状态，以模拟粒子的动态演变。但由于快照时间间隔较大，进行简单的线性插值或贝塞尔插值之后，边缘区域尚能保持正常状态，但大量中心区域天体被甩飞，有时甚至会出现明显错误的空洞，这说明单一的插值方法不能普适于所有区域的粒子：

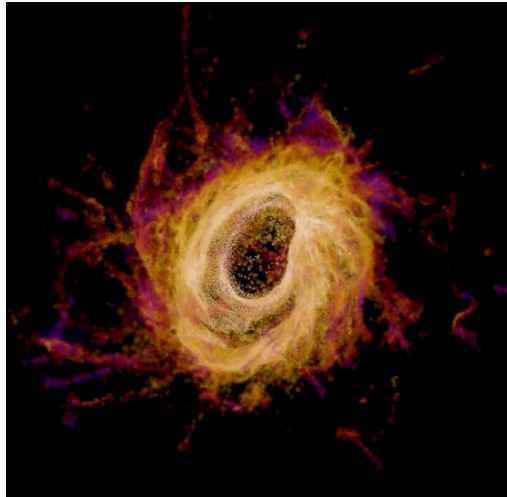


图 4 错误运动状态截图

经计算粒子相对引力中心的瞬时角速度，发现在两个快照长达约 1 亿年的间隔之内，天体绕转的周期远不止一周，越接近引力中心，旋转越快。我们仅能把前后两个快照作为起点和终点，其实中间的运动过程远比预想的更复杂，这是导致上述错误的原因。

经过数十次采样、建模、渲染、修正的反复试验，确定了从内到外渐变过渡的组合插值算法：

- 1、中心区域的粒子轨道在 1 亿年内绕转多个周期，采用自定义算法的螺旋曲线插值；
- 2、星系盘上的大部分粒子绕转少于 1 周，但角速度变化较大，采用 Hermite 曲线插值；
- 3、星系外侧的粒子角速度变化较小，但仍有一定的曲率，采用贝塞尔插值；
- 4、距离引力中心更远的粒子采用最简的线性插值，以节省运算量。

由此将静态的快照数据插值为连续的动态粒子场景，粒子运动更加平滑自然，星系演化过程和汇聚碰撞的动态过程都获得专家确认符合科学性。见下图：

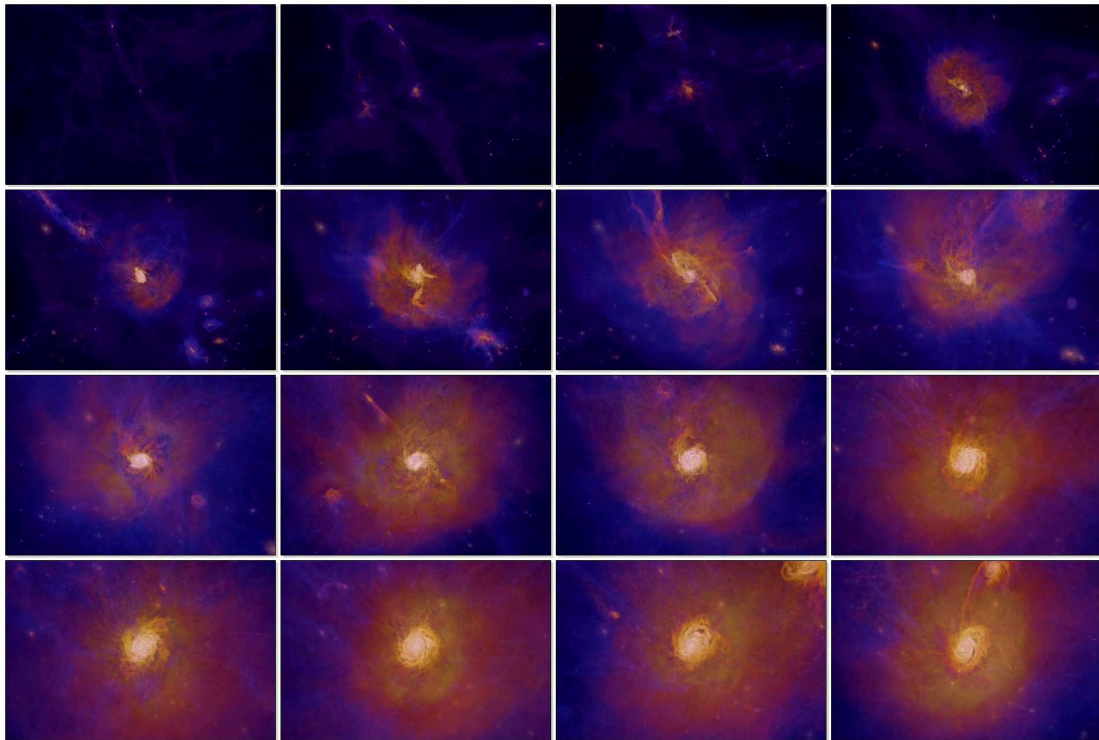


图 5 中心星系动态演化模型

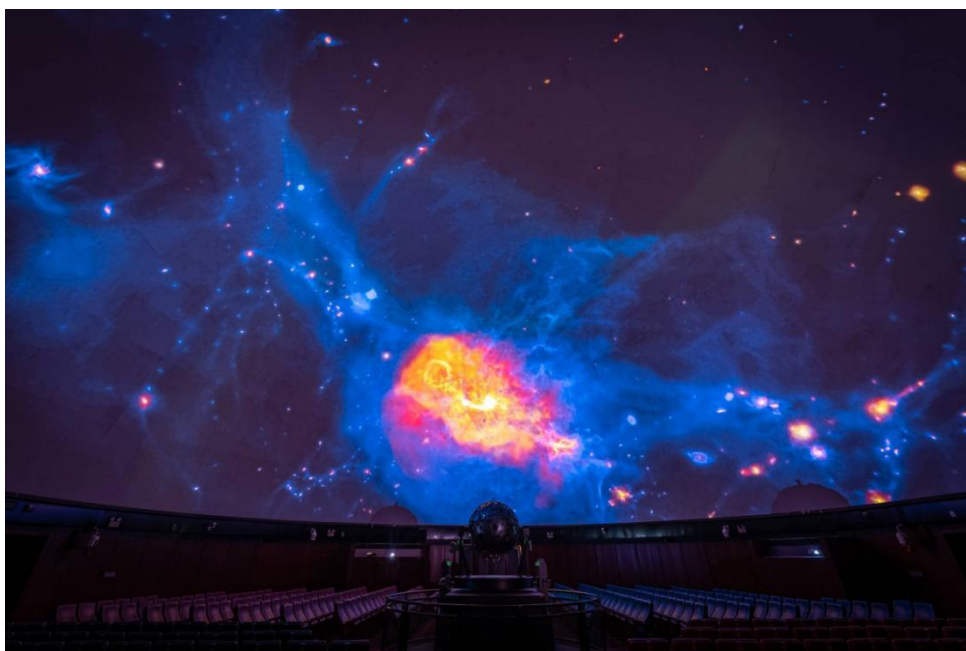


图6 球幕剧场实拍：星系汇聚碰撞过程

第4阶段：全局建模

验证完成局部动态模型后，接着要创建整个3亿光年的静态模型。然而当花费了2天、创建的粒子数达到约4、5百万时，Maya已无法操作，继续操作则直接崩溃。

过滤掉极低密度、极小质量的接近不可见的粒子，仍有一千多万个。由于天体运动的不确定性，无法按粒子位置拆分为多个渲染层，只能将所有粒子作为一个整体创建。

研究Maya的底层机制后，最后放弃了Mel这种低效率的解释性语言，转用较激进的方式：通过调用Maya底层API寻址粒子数组指针，直接批量写入内存区域块来更新PerParticle属性，避免了旷日持久的串行赋值，成功创建了约1200多万个粒子的宇宙大尺度结构静态模型，这个数字已大大超出了Maya2018能够承载的nParticle粒子上限。

下图是宇宙透视图，前一阶段创建的局部星系模型在此尺度下已缩小为中心部位的微小光点：

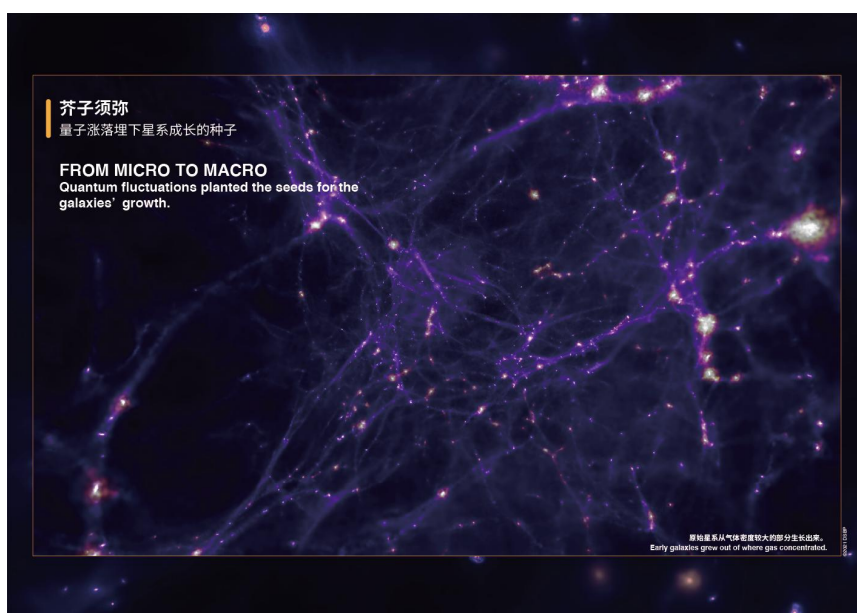


图7 宇宙大尺度结构静态模型（局部）

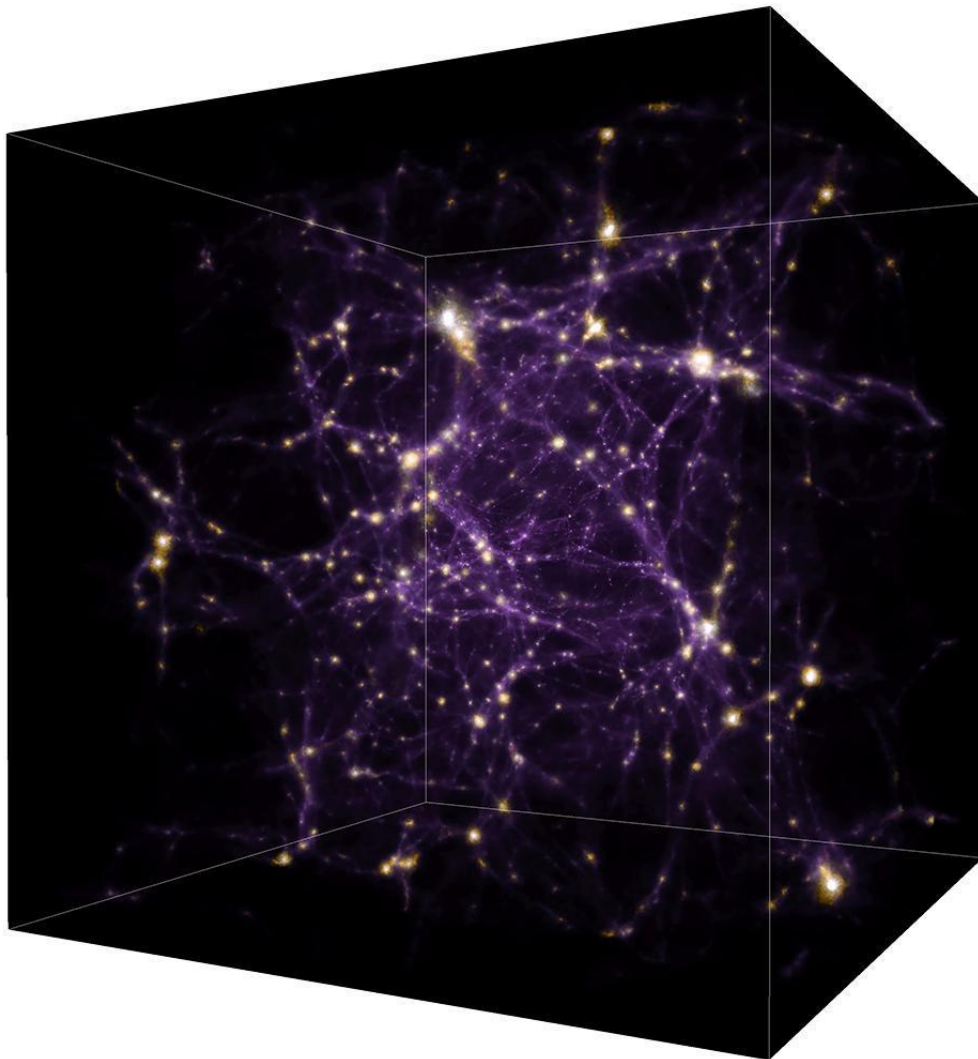


图 8 宇宙大尺度结构静态模型（全局）

直写内存的方式在突破软件极限的同时，也有其巨大缺陷：此模型过于庞大，几乎无法进行任何修改操作，无法写入表达式驱动其属性数据，也无法用 `Software` 或 `Arnold` 等渲染器，仅可用硬件渲染器渲染单帧，若批量渲染序列帧则会导致软件崩溃。

第 5 阶段：动画渲染

影片剧本设计有多组镜头需要在动态的宇宙大尺度结构中穿梭漫游，一个无法进行操作的粒子系统模型显然无法用于制作动画。我尝试多种优化方法后均告失败，后来受定格动画启发，找到了解决办法。

本方法比定格动画更为繁复，每帧都要重建整个宇宙：即启动 `Maya` 渲染器时，根据当前帧的宇宙年龄，读取对应的快照数据执行插值计算，生成当前的宇宙数据，再自动创建宇宙模型；渲完单帧之后立即清空场景和当前的粒子缓存，以避免内存累积导致崩溃。一切动画动作都通过预先写好的 `PreRender Frame Mel` 脚本机制调用，全程无手工干预。

得益于自主开发的《太微云渲染》集群渲染软件，我们能够直接修改已有代码，得以快速定制渲染流程。虽然每帧耗时很久，但基本实现了稳定的批量动画渲染。最终可以实现宇宙任意时期的全局动态变化。

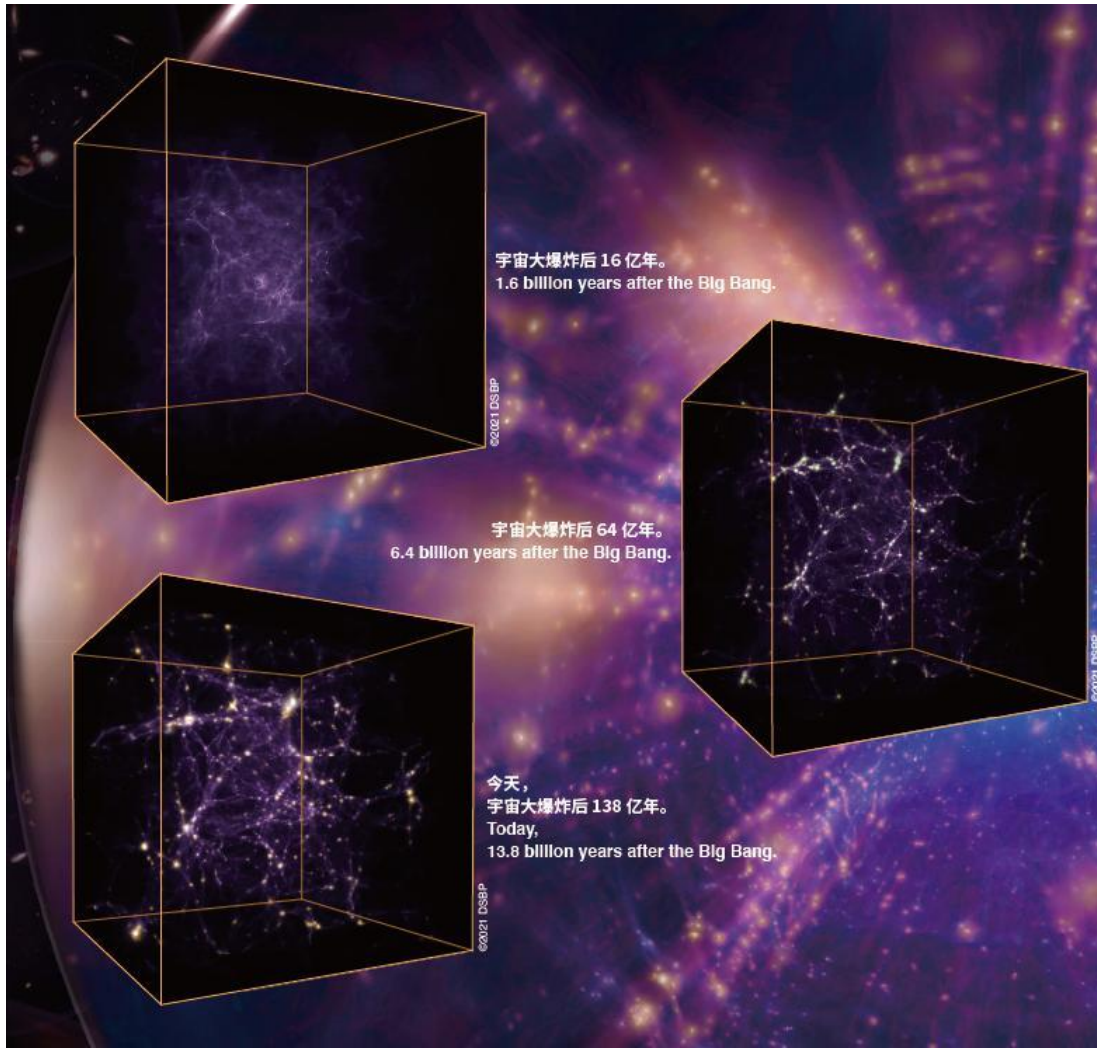


图 9 宇宙演化模型（宇宙年龄：16 亿年、64 亿年、138 亿年）

总结：

以上仅简述了主要步骤，具体细节不一赘述。5 个阶段历时超过 1 年半，开发 Mel 和 Python 代码达到上万行，还包括多次技术路线试错，例如最初曾试用流体模拟网格渲染，因 3D 流体容器分辨率上限过低，细节达不到精度要求而放弃；另外还尝试过 2 套小规模星系碰撞数据、1 套跨 10 亿年的暗物质数据，因模拟的结果不符合影片剧本所需的形态而废弃。

另外也总结了较多有用的经验：例如对超大规模数据进行降采样之后再建模，可以快速获得低精度模型轮廓，便于尽早验证其科学性，也可作为 Layout 动画的粗模使用。再如在效果调试过程中，利用密度-颜色映射的 Ramp 材质，可以直观的观察出粒子密度的分布，更便于进一步精调参数，此上色方法可推广到所有参数的可视化调试过程中，如下图：

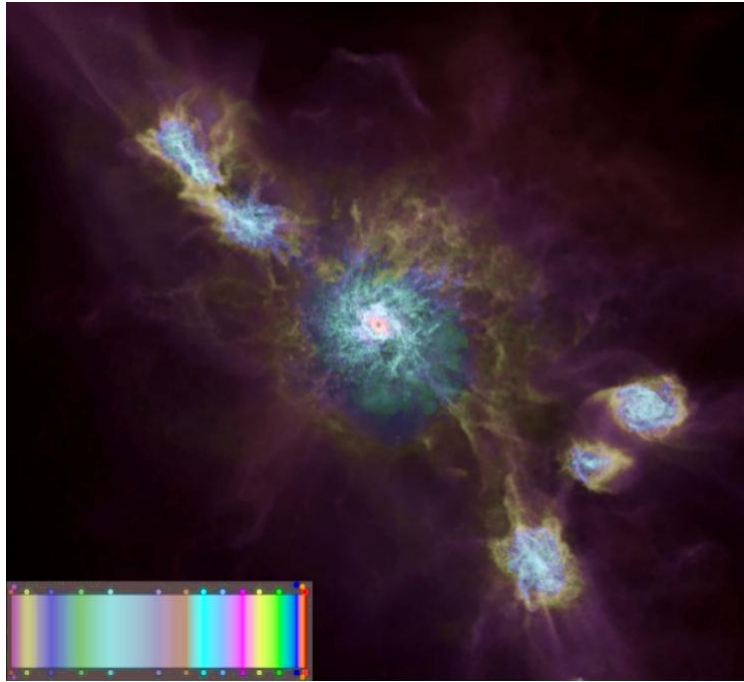


图 10 粒子密度-颜色映射图

至此，常规动画软件流程无法解决的问题，均被我们攻坚克难逐一解决，而我们长期形成的可视化技术流程也基本被颠覆。此项目对未来实现更大规模天文数据可视化具有开创性的意义。

虽然影片今年顺利首映，但两度延期的项目进度同时也提醒我们，未来的项目需要更高效的算法，以及更高效的工具。

四、技术突破

《宇宙大爆炸》影片项目于 2021 年初顺利结题，在 2020 年下半年攻克技术难关之后，我得以重新思考之前的方法路线是否合理。

近两年在向虚幻 4 移植可视化资源过程中，我测试了 Sequencer 的动画批渲功能，以及蓝图脚本、材质着色器等，重点研究了 Niagara 粒子的技术特性，它可以支持更多 GPU 算法，更适用于数据密集型的大规模数据可视化。通过针对 GPU 算法的研发，取得了两点重要突破：

1、利用 RGBA 图像存储天文浮点数据，以较小的资源存储极大的数据密度，将 CPU 的数据压力转移到 GPU 上。经过对浮点的压缩和编码处理，6 帧 4K 图即可解决 Auriga 项目的一个快照数据（气体+恒星）的存储需求，仅占用显存约 300MB（原数据每个快照最小 2GB），对主流 GPU 没有压力；

2、开发像素着色器（Pixel Shader）从输入图像中解析海量天文数据并进行插值计算，再利用材质着色器的世界位置偏移（WorldPositionOffset）属性，可偏移粒子到指定天文坐标——即 GPU 可以在瞬间同时对全部粒子执行大规模的像素偏移。

本人今年已将以上创新申报发明专利《天文数据可视化方法、装置、计算机设备及可读存储介质》，目前处于公开状态，公开号 CN112862928A，专利局网站可查阅全文。技术流程见下图：

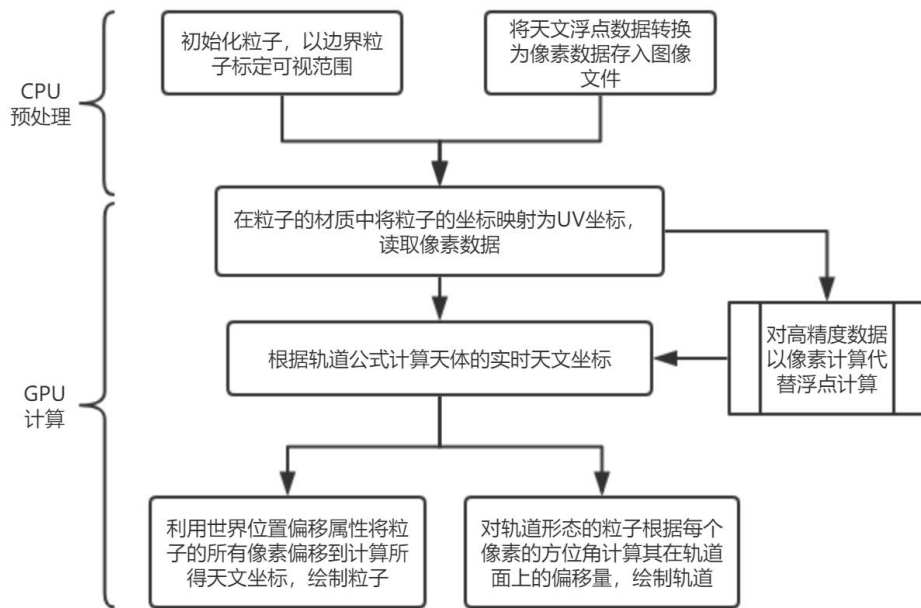


图 11 基于 GPU 的天文数据可视化技术流程图

基于以上技术，能够将极为耗时的海量天文存储、计算和动画渲染环节完全转移到 GPU 的并行计算中，从而极大的加速可视化动画效率。经测试，1200 万粒子的宇宙大尺度结构漫游场景，Maya 从建模到渲染每帧耗时近 2 小时，在虚幻 4 中竟达到每秒 7.3 帧（见下图），充分展现了 GPU 并行计算的威力。

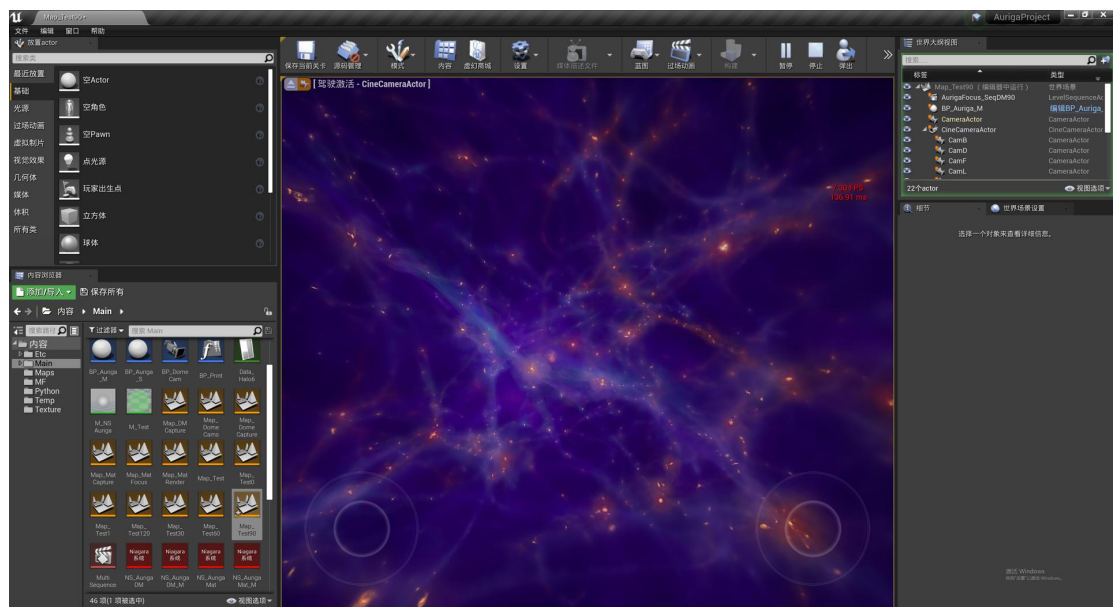


图 12 虚幻 4 粒子实时渲染场景截图

借此技术突破，我利用虚幻 4 引擎开发了一套《宇宙矩阵天文数据可视化软件》，专用于制作超大规模的天文数据可视化动画，借助 Sequencer 渲染机制，动画定制化程度更高。此软件可以为多年来积累的所有天文可视化资源全面赋能提速，目前正对《宇宙大爆炸》的部分镜头改良重制中。

五、技术展望

作为动画项目的技术负责人，日常不仅仅只考虑编程算法等问题，还要考虑动画团队的协作、集群环境的搭建和运营、动画场景的优化渲染，计算资源的分配调度，等等，许多细节问题会占用大量时间，但天文数据可视化这条主线多年来是一直贯穿始终的，必须保持足够的精力和资源投入在这个方向上，这是天文馆动画团队保持行业竞争力的立身之本。

王杰研究员对我们近期的可视化工作表达了赞许，并期待未来在暗物质晕精细结构模拟方面寻找进一步合作的机会。来自科学家的认可使我们深受鼓舞，我们意识到天文可视化工作不仅能够帮助我国科学家将成果展示给大众，也有助于科学研究者从视觉信息中发现其潜藏的模式，进而获得新的科学发现。

未来我们将与科学界进一步深度合作。我国已产生大量的天文大数据，如 LAMOST 巡天数据、FAST 的脉冲星数据等，其价值远未被公众认知。中国的科学数据宝藏，应该在中国的动画团队手中来点亮。虽然这些数据规模远超我们当前能够处理的数据体量，但两年前的我们还不敢想象能够挑战 Auriga 项目数据，如今 GPU 技术更是打开了一扇新的大门。只有不断迎接新的挑战，全力拥抱新技术，才能加速我们的技术创新，制作出更好的特种科普电影。