# 对于大规模CAD模型的实时计算和渲染

## 背景介绍

1. 背景描述

非均匀有理B样条曲线和曲面（NURBS）已经被确认为几何建模的重要行业标准。其充分体现了在计算机图形学中对于精确表达和生成流畅曲线和曲面的需求，因而受到了广泛应用。鉴于NURBS在灵活性和精确性上的优势，国际标准化组织（ISO）特别将其纳入了工业产品（如CAD/CAE/CAM等）的数据表达和数据交换标准STEP中，赋予其唯一并卓越的地位，为定义工业产品的几何形式提供了最佳选择。

在多数高精尖端的工业行业中，如模具制造、汽车设计、航空制造、船舶工程以及建筑设计等领域，大规模的NURBS曲线/曲面构建的CAD模型得到了普遍应用。这些模型的构成规模常达到数十万，甚至是数百万级别，其庞大的规模和精细的结构为我们的现代工业贡献了重要力量。因此，非均匀有理B样条曲线和曲面的运用不仅改变了几何建模的局面，同时也推动了工业产品设计和制造的进步，持续推动着我们的工业社会向着更高效、更精确的方向发展。

1. 存在的问题

目前版本的商业CAD系统通过预先嵌入NURBS模型，并以离线方式将其离散为LOD（细节级别）三角形网格以高效地处理大规模NURBS模型的渲染。然而，预嵌入的特点导致了以下三个问题：

* 1. 预先将NURBS曲面离散为三角网格不可避免地导致了精度的不可恢复的损失。因为三角形网格的分辨率在其被生成之后就固定了，因此在某些视角下，可能会感觉到不平滑的分面着色效果和多边形轮廓的粗糙，这可能会影响CAD设计师在高精度曲面设计场景下的视觉效果和设计准确性。
  2. 在渲染时，通常需要借助缝合算法保证不同LOD之间的平滑过渡，但相邻斑块/LOD水平之间的裂缝很难处理，从而导致特写镜头显示效果跳变等缺陷，进而影响产品设计质量。
  3. 为了达到理想的渲染质量，精细级别三角网格将占用大量的存储空间。良好的渲染质量将导致文件大小和内存使用量大幅增加，给I/O带宽带来沉重负担，并最终影响渲染性能。

1. 面临的挑战

本文的工作不再在预处理步骤中进行镶嵌，而是采用实时自适应镶嵌框架来实时渲染大型NURBS模型。这项任务的挑战如下：

* 1. 至今还没有工作的精确测量采样误差的度量适用于运算计算量巨大的NURBS。
  2. 对于曲面的离散三角网格间的裂纹问题，可以通过防止可能对框架设置限制的裂纹，或者通过设计裂纹检测和消除算法来解决。
  3. 在实时渲染性能上，在达到足够渲染进度要求的同时，尽可能降低模型尺寸规模。

## 相关工作

1. 建模
   1. 近似误差控制。研究人员提出了各种方法对NURBS曲面进行自适应镶嵌，以提高渲染效率和精度，但无法用于NURBS曲线曲面。
   2. 镶嵌引起的裂纹。由于相邻面片的采样率不一致，具有均匀采样网格的自适应镶嵌可能会沿面片边界产生裂纹，但是在性能上会有一定影响。
   3. 非均匀采样，但这种方法不适合GPU并行化，基于双三次曲面具有线性二阶导数的事实，使得它不能应用于有理曲面或更高阶曲面。此外，即使原始样本数量减少，该方法也不能优于均匀采样方法。
2. 评估

NURBS曲面可以直接以精确的NURBS形式进行评估，首先将NURBS面片转换为计算强度较小且精度更高的Bézier面片。在镶嵌过程中，每个Bézier面片的采样率可以单独计算，从而缓解了在使用均匀采样网格时，由于跨度的曲率变化很大而导致不必要的多余基元的问题。使用节点插入算法将NURBS转换为Bézier通常使用双精度在CPU上执行，而渲染使用单精度在GPU上执行，因此在边界渲染期间不存在由转换引入的浮点误差。

1. 渲染

CAD模型通常包含许多形状复杂的曲面和小零件，从而在镶嵌后产生大量三角形，因此传统管道很容易达到瓶颈。对于大规模三角形网格的实时渲染，Nanite是最新的GPU驱动实现之一。Nanite是为实时游戏设计的，用于渲染像素级细节和高对象数，但是，Nanite仍然基于预先镶嵌的网格，目前对于NURBS实时渲染解决方案，还没有工作提出。

## 模型架构

本模型由三个核心部分组成，分别为自适应细分、混合光栅化以及裂缝填充。

1. 自适应细分

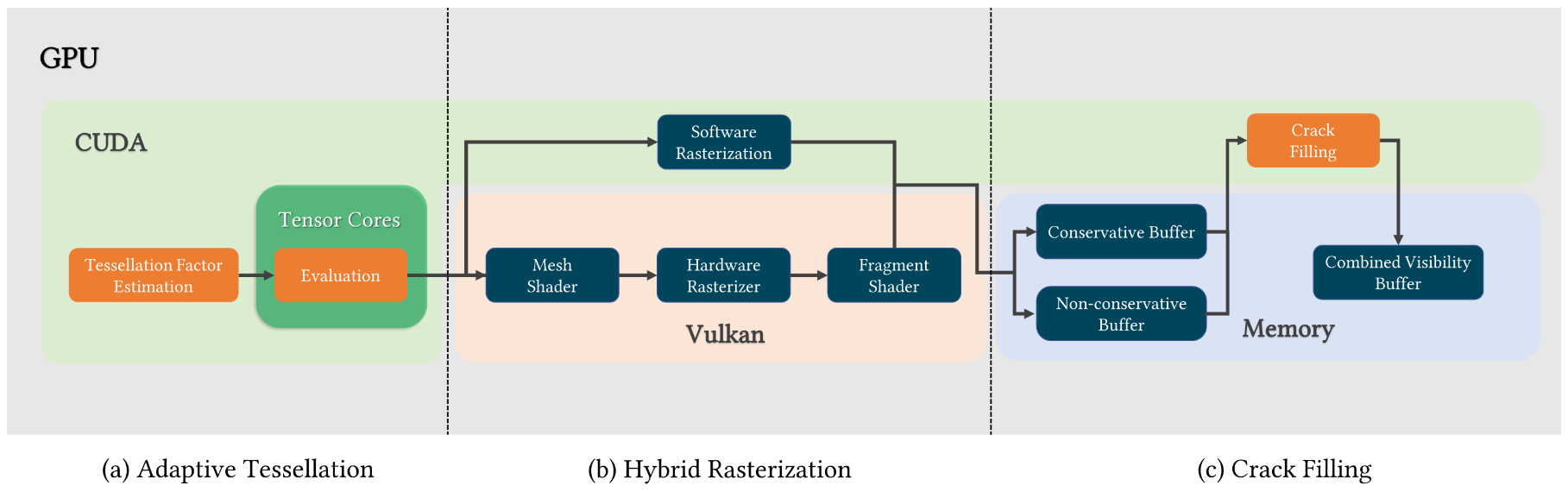
此部分主要通过CUDA来实现镶嵌因子的预测和表面的评估算法。预测后的结果会作为张量核加速矩阵求产品并对评估结果进行累加，将结果完整嵌入到三角形网格中，并分别存储在两个缓冲区中。

1. 混合光栅化

在镶嵌阶段产生的大量三角形网格将被划分为小网格。由小三角形组成的网格会被CUDA所实现的软件光栅处理，其他的网格则通过网格着色器管线进行处理，并将结果写入两个可见性缓冲区。

1. 裂缝填充

本文将采用一项基于像素精确的镶嵌技术，借助保守光栅化和可见性缓冲区来高效填充裂缝。完成裂缝填充后，将生成没有因镶嵌导致裂纹的可见性缓冲区。



## 效率优化

在本节，我们将着重介绍了关于效率优化的内容。所有我们所提出的新算法，包括高度并行的像素精度自适应镶嵌算法、基于CUDA和张量核的NURBS评估加速算法，以及高效的可见性缓冲区和基于光栅化的裂缝填充算法，都采用基于CUDA和张量核的加速机制来加强效率。

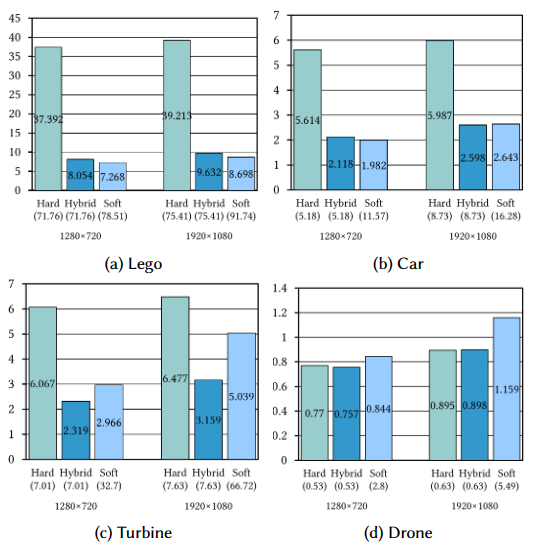
在实际的操作过程中，我们特别采用了可视化缓冲区，把弹性框架ETER融入了基于Mesh Shaders的GPU渲染管线，这是一种融合了软硬件的光栅化渲染方法。通过将ETER框架与此种渲染管线集成，我们得以评估并验证ETER在实时细分（tessellation）性能与渲染效果方面的表现。

借助这种方案，我们成功地提高了实时细分渲染的效率，并且从理论和实际两个层面证实了ETER在实时渲染中的应用价值。这种优化不仅提升了整体的效率，而且也确保我们的渲染结果在质量和速度方面达到了良好的平衡。此部分按计划进行，大大优化了我们的研究过程，同时也为我们的未来研究提供了宝贵的经验。

## 实验效果

我们使用4个trimmed NURBS CAD模型（无人机，乐高，整车，航空发动机）对ETER框架的性能进行评估，结果表明，在单张RTX 3090Ti GPU上，该框架能够实时地对370万个有理Bézier曲面进行像素精度的自适应Tessellation演算，生成并渲染多达2.5亿个三角形。

由于ETER所展现的卓越性能，我们能够前所未有地处理深度复杂度超过100个三角形每像素的场景，这不仅能完全满足且远远超出实际三维CAD渲染场景的需求。因此，相当一部分GPU计算资源可以分配给其他渲染任务，例如光照、材质等等。正如UE5游戏引擎的Nanite技术所展示的数亿三角形规模场景的渲染能力，基于ETER的实时演算渲染方案可以将CAD模型可见部分以相同规模进行渲染。对于大规模CAD模型，如果使用当前主流的预采样轻量化LOD方案，那么很有可能需要数百亿甚至更高规模的预采样LOD三角网格才能达到与ETER接近的渲染精度和质量。



## 总结与展望

1. 总结

在本研究中，我们深入探讨并提出了一种独特的弹性框架，即ETER，它使我们能够实时精确绘制大型的NURBS模型，确保其达到像素级的精度，并且在质量方面没有裂纹。此最终成果是基于我们所设计的三种全新并行算法：高度并行的像素精度自适应镶嵌算法、基于CUDA和张量核的 NURBS评估加速算法和高效的、可见性缓冲区以及保守的光栅化裂缝填充算法。此外，我们利用了GPU来加速这些过程，以实现更优秀的性能。

1. 未来展望

然而，我们的研究并没有到此止步。ETER框架使用具有均匀采样网格的自适应镶嵌，这可能引发裂纹产生，且与其相关联的裂缝填充算法在现实场景中应用存在一定的限制。这导致在实时镶嵌科技的发展过程中解决这些裂纹的问题成为了我们面临的一项严峻挑战。

因此，我们的未来研究方向将会聚焦于如何在保证实时性的前提下，解决镶嵌过程中的裂纹问题。我们期待这一领域的未来发展，并乐观地看待我们能够建立一套改进的解决方案，可以更好地处理这些裂纹，同时还保持渲染质量和绘制速度。

【参考文献】

[1] RUICHENG XIONG, YANG LU, YAJUN ZENG, and LIGANG LIU, ETER: Elastic Tessellation for Real-Time Pixel-Accurate Rendering of Large-Scale NURBS Models

[2] Young In Yeo, Lihan Bin, and Jörg Peters. 2012. Efficient Pixel-Accurate Rendering of Curved Surfaces. In Proceedings of the ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games. Association for Computing Machinery, New York, NY, USA, 165–174.