DOI:10.20033/j.1003-7241.(2025)10-0106-04

GPU 区域着色器的同顶点检测技术

张淮声[1]，余 莉[2]，卞仁玉[1]

（1.格兰菲智能科技有限公司，上海 201203；2.上海建桥学院信息技术学院，上海 201306）

摘 要：GPU 芯片中的曲面细分模块可以生成大量的顶点，并构造线段或者三角形来表现三维模型的细节信息，但是顶点数量很大，给后续的计算和存储带来很大的压力。考虑到这些几何图元彼此相邻且共享多个顶点，提出了一种同顶点检测的硬件架构实现策略，在构造区域着色器线程之前，添加一个域值匹配单元来比较输入顶点的域值，跳过具有相同域值的顶点，仅对不匹配的顶点进行空间分配并执行区域着色器线程。实验表明，该方法将区域着色器中涉及的顶点数量减少了 50%左右，整体性能也得到提高。

关键词：曲面细分；区域着色器；线程；匹配失配

中图分类号：TP277 **文献标志码**：A **文章编号**：1003-7241(2025)10-0106-04

Same vertex detection technique for domain shader in GPU

ZHANG Huaisheng[1], YU Li[2], BIAN Renyu[1]

(1. Glenfly Technologies INC., Ltd., Shanghai 201203, China;
2. Department of Information Technology, Shanghai Jianqiao University, Shanghai 201306, China)

Abstract: Tessellation module in GPU chip can generate a large number of vertices and construct line or triangle lists to represent the details of the 3D model, but the number of vertices is large, which brings great pressure to the subsequent calculation and storage. Considering that those geometry primitives are adjacent to each other and share many vertices, it proposes a same vertex detection mechanism. Before the domain shader wave is constructed, one domain matching unit (DMU) is added to compare the domain values of the input vertices so that the vertices with the same domain values could be skipped, and only the mismatched vertices are allocated and executed by DS wave. Experiments show that the method reduces the number of vertices involved in domain shader by about 50%, and the overall performance is improved, too.

Keywords: tessellation; domain shader; wave; hit-miss

0 引言

Tessellation 称为曲面细分，多年来一直是计算机图形学的重要研究内容。该技术通过输入细分块和细分参数，生成大量的线段或者三角面片，镶嵌拼接为连续的几何形态，使得模型表面变得或者细腻顺滑（例如面部），或者高低不平（例如地形、海面），从而增强了绘制真实感。早期的曲面细分技术获得了良好的细分效果，但是计算量很大，难以通过硬件实时计算。文献[1]提出了一种迭代技术，考虑到细分块之间的邻接性生成连续的网格，并且对于硬件实现友好，使得通过硬件加速曲面细分成为可能。为了便于用户编写实时的曲面细分渲染程序，微软公司在 2010 年提出了 Direct3D11 绘制平台[2]，该平台参考了文献[1]的方法，引入了硬件实现的曲面细分功能。这一平台易于程序员的实现，实时绘制效果也不错。其它的绘制平台，例如 OpenGL、Vulkan 等[3]也先后采用了该方法，随着 GPU 的图形绘制流水线引入了曲面细分功能，很多研究和产品开发工作也大量采用该技术[4-6]。

该技术增加了两个新的着色器，外壳着色器（hull shader，HS）、区域着色器（domain shader，DS）和一个曲面细分硬件单元（tessellator，TESS）。其中 HS 进行细分块的预处理功能，把细分参数送入到硬件加速模块 TESS，TESS 输出大量顶点到 DS，DS 中计算这些顶点的空间位置和颜色等信息，使得渲染结果栩栩如生。但大量的顶点给后续的计算带来一定的压力。本文在 GPU 架构中引入了一个域值匹配检测单元，在顶点进入 DS 之前，先进行匹配检测，对于相同区域值的顶点，仅返回已有的匹配顶点索引号，不进行 DS 计算；只对不匹配的顶点执行 DS。通过这种匹配检测技术，减少了相同域值的重复计算，降低了属性数据缓存等资源的消耗，提高了整个区域着色器的执行效率。实验表明，通过增加域值匹配单元，对同顶点进行过滤后，在 DS 中参与计算的顶点数目可以减少 50%左右，提高了 DS 的执行效率。

1 曲面细分模块简介

微软提出的 Direct3D11 绘制平台大幅度地加速了细分曲面的构造。图 1 显示了支持曲面细分的绘制流水线，

*基金项目：上海市“晨光计划”基金项目（AASH1702）；上海市教委项目（JXGG202064）；上海建桥学院科研项目（BBJQ202504）

收稿日期：2024-01-13

虚线框中是新增加的细分模块,包含 HS、DS 和 TESS 三部分。HS 对每个细分块,计算出多个控制点(control points, CPs)和细分参数(patch consts, PCs),将细分参数传送到 TESS 中,可计算出新顶点的域值,使用二维 UV 坐标表示;将控制点传送到 DS 中,DS 根据控制点和 TESS 输出的顶点 UV 域值,插值出最终的三维坐标以及颜色等属性信息。DS 再把生成的新顶点数据,输出到几何着色器(geometry shader, GS)中,进行后期的几何处理。

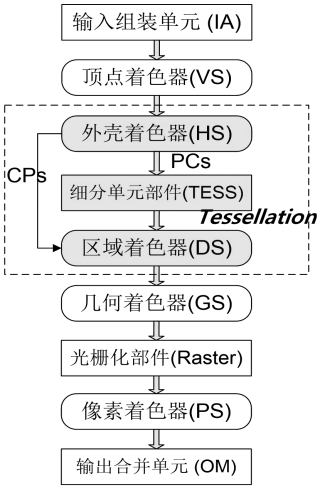

图 1 包含曲面细分模块的图形绘制流水线

用户可以定义三种细分块:平行线块,三角形块和四边形块。平行线块主要用于毛发、草地等的生成,所以主要的输出类型是由多个顶点连接而成的线段。三角形块主要输出大量相互连接的三角形[2],可以表现出物体的凹凸表面、高低起伏的地形、海面等。考虑到三角形之间的邻接关系,一个顶点常常会被多个三角形共用,因此 DS 会收到大量重复的顶点。四边形块的细分结果与三角形块类似,主要也是生成物体的表面细节信息,同样会有很多重复的顶点送入到 DS 中。

TESS 细化出的新顶点的域值用二维坐标 UV 表示,其范围在[0, 1.0]之间。为了便于硬件实现,把浮点的 UV 坐标值用 17 bits 的定点数表示,其中最高的 1 bit 表示整数部分,低 16 bits 表示小数部分。本文提出的同顶点的检测技术,就是根据顶点的 UV 坐标判断是否相同,从而减少相同域的重复计算。

2 区域着色器同顶点的检测技术

2.1 区域着色器线程构造单元

GPU 中的线程称作 wave,组织成单指令多数据(single instruction multiple data, SIMD)的形式,一般为 32 个数据单元(lane)构成一个 wave(称为 SIMD32),在 GPU 的执行单元中并行执行。为了能让区域着色器的多个顶点同时执行,GPU 硬件中引入了区域着色器线程构造单元(domain shader wave constructor, DSWC),用于依次组装 TESS 输出的顶点。

DSWC 主要负责从 TESS 接收域值、控制点、配置参数和命令等,把域值按照 32 个一组构造 DS 线程。由于每个细分块可能会生成很多的三角形,TESS 将输出大量的顶点,容易造成 DS 线程过多,严重影响整体的绘制效率。考虑到 TESS 生成的几何图元之间紧密相连,存在很多相互邻接的顶点,这些位置相同顶点的域值也一样,理论上仅需执行一次区域着色器任务即可,因此,本文在 DSWC 中增加一个域值匹配单元(domain matching unit, DMU),用于检测出相同域值的顶点,从而避免 DS 的重复计算。

2.2 域值匹配单元

2.2.1 DMU 的逻辑结构

图 2 为 DMU 的内部逻辑结构,主要包含如下几个子单元:

▲ TagRAM:DMU 内部的域值缓存,存储 n 行顶点的域值信息,每行记录 32 个顶点的 UV 坐标和对应的 dsid 值,每行对应一个 simd32 的 DS 线程;这里 n 一般取为 2;

▲ 顶点域值的比较单元(domain compare unit, DCMP):将接收 TESS 发来的顶点域值 UV,并与存储在 TagRAM 中的域值进行比较,如果找到相同的,就认为是 hit,如果未找到相同的,就认为是 miss;

▲ CTRL:DMU 的控制单元,接收 DCMP 单元发来的 hit/miss 信息,如果是 hit,将从 TagRAM 读出 dsid 返回给 TESS;如果是 miss,向 DSIDT 发送信息分配一个新的 disd,并把该顶点的 UV 值和 dsid 记录到 TagRAM 中,然后把新的 dsid 返回给 TESS。

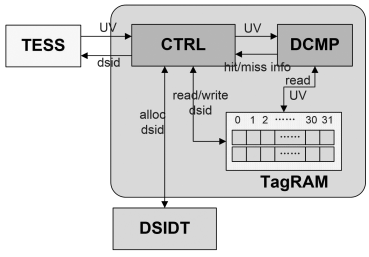

图 2 DMU 的逻辑结构

DMU 需要和 DSIDT 进行交互,用于记录和管理 DS 线程的输出顶点在 VB 中的坐标位置信息,对于 DMU 中失配的顶点,需要从 VB 中分配一个索引地址,称为 dsid,用于记录相应顶点经过 DS 计算后得到的输出属性,包括位置、法向量、颜色等信息。当顶点信息被后序模块使用完毕,还需要负责释放顶点的空间。

对每个输入的顶点,DMU 通过 DCMP 子单元进行匹配操作,将会有匹配(hit)和失配(miss)两种情况。

2.2.2 HIT 的执行机制

输入顶点的 UV 域值在 TagRAM 找到相同域值的情况称为匹配。如图 3 所示,在 TagRAM 中存在两行的缓存 buffer,每行空间可以存放 32 个顶点的 UV 坐标和 dsid;在 DSIDT 中存在一个 64 行的 buffer,buffer 的每行记录一个 DS 线程 wave 的状态,32 个顶点组织成一个 wave,只有

miss 的顶点才会添加到当前的 wave 中,如果是 hit 的顶点仅记录 hit 的个数,不添加到当前 wave 中。所以 TagRAM 中的两行 buffer 分别对应 DSIDT 中最新的两个 wave。

HIT 情况下的执行机制如下:

1) 当 DMU 收到 TESS 送来的一个新顶点的时候,将该顶点与 TagRAM 中存放的所有顶点的 UV 值进行比较;

2) 在 TagRAM 中找到相同 UV 顶点的 dsid 作为新顶点的 dsid,返回给 TESS;

3) 从 DSIDT 中找到该顶点所在的 wave,将该 wave 的 ref_cnt 加 1。

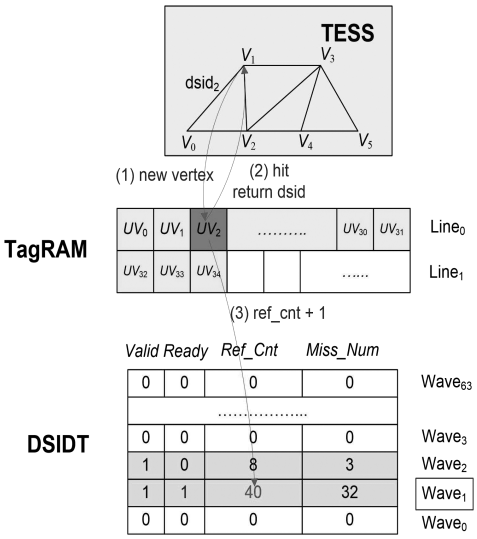

图 3 HIT 的执行过程

2.2.3 MISS 的执行机制

输入顶点的 UV 域值在 TagRAM 找不到相同域值的情况称为失配。如图 4,miss 情况下的执行机制如下:

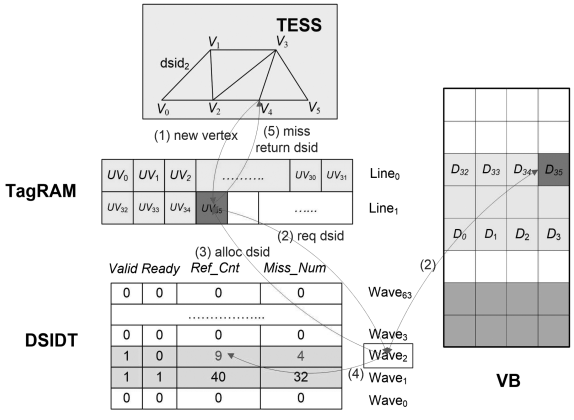

图 4 MISS 的执行过程

1) 当 DMU 收到 TESS 送来的一个新顶点,将该顶点与 TagRAM 中存放的所有顶点的 UV 进行比较,如果所有顶点均比较后,找不到相同 UV 域值的顶点,就进入 MISS 状态;

2) 通过 CTRL 向 DSIDT 申请一个新的 dsid;

3) DSIDT 分配一个新的 dsid,作为该顶点在 VB 中的输出位置,和 UV 域值一起记录到 TagRAM 的最新位置;

4) 从 DSIDT 中找到该顶点所在的 wave,将该 wave 的 ref_cnt 和 miss_num 均加 1;

5) 返回该顶点的 dsid 给 TESS。

3 实验结果分析

为了验证本方法的有效性,我们采用 VC++,编写了使用域值匹配单元 DMU 的模拟程序,在 CPU 为 Intel Core i7、内存 16GB 的计算机上运行并获得实验数据。分别测试了 TESS 输出为 Line 和 Triangle,在不同细分参数的情况下,使用 DMU 后执行 DS 所减少的顶点比例。

图 5 统计了在细分参数均相同的条件下,一组平行线块的细分情况,细分参数均匀分布,分别为 4、8、12、…、64,TESS 输出的类型为 Line,每条 Line 需要 2 个顶点表示。其中的柱状图分别是没有 DMU(no_DMU_ vtx)和使用 DMU(use_DMU_vtx)得到的需要执行 DS 的顶点数目,约定 use_DMU 的 hit 顶点的比率为 Hit_ratio = 1−use_DMU_vtx /no _DMU_vtx,采用折线段表示。hit 比例越高,说明最终参与 DS 执行的顶点越少,对整体性能提升越大。从图示可知,随着细分参数的逐步增大,no_DMU 和 use_DMU 所得到的顶点数目也会相应增多,但是依然会有大量的顶点由于 hit 而跳过 DS,如图中的 Hit_ratio 所示,当细分参数大于 20 之后,其 hit 的比例接近 50%,也就是说,通过增加 DMU 单元,当 TESS 输出为 Line 的时候,提高了 DS 的性能达到 50%左右。

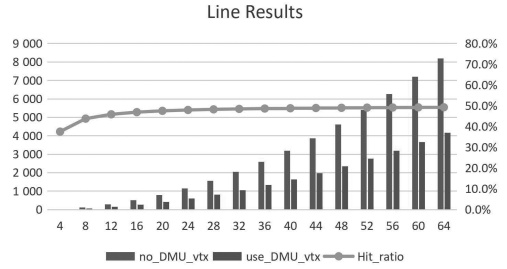

图 5 TESS 输出为 Line,use_DMU 获得的增益效果

图 6 记录了 TESS 在输出 triangle 的时候,use_DMU 获得的增益效果。此时每个 triangle 需要 3 个顶点表示,当细分参数逐步增大的时候,no_DMU 和 use_DMU 所得到的顶点数目均会相应增加,同时 hit 的顶点数目也会增加。当细分参数比较小的时候,其 hit 比例表现得更好:70%~80%,此时每个细分块生成的三角形数量比较少,造成共用的顶点数目相对较多;当细分参数逐步增大,其 hit 比例逐步稳定在 65%左右。这说明,对于 TESS 输出为 triangle 的情况,通过增加 DMU 模块,获得的 DS 性能提升情况能够达到 65%,比 line 的情况效果更佳。

上面两个实验是在理想情况下的测试结果,为了更进一步检测在实际应用中的效果,我们选择了 3DMark11 等

测试平台和应用程序，包括 FireStrike、HeavenV4. 0、Hitman、Crysis3。考虑到存在大量的 draw，每个 draw 也有很多的细分块，我们计算平均每个 draw 的输出顶点的总数（no_DMU_vtx_perDraw）和平均每个 draw 实际执行 DS 的顶点数目（use_DMU_vtx_perDraw），并计算平均每个 draw 的 Hit 比例为 Hit_ratio_perDraw = 1- use_DMU_vtx_perDraw/no_DMU_vtx_perDraw，用来反映使用 DMU 的效率提高情况。

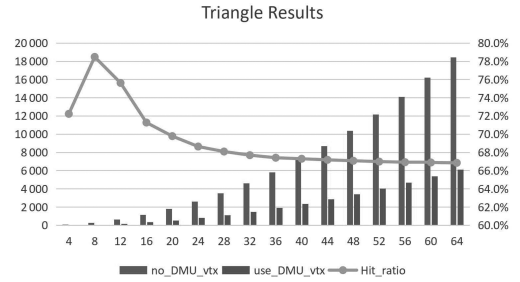

图 6　TESS 输出为 Triangle，use_DMU 获得的增益效果

从图 7 中可以看到，不同应用程序的每个 draw 均会生成大量的顶点，产生顶点数量不同，但其实很多顶点都是重复的，通过引入 DMU 模块，大约有 38%～55%的顶点能够被 hit，从而跳过 DS 计算，也就是说，我们的方法在实际应用中，能够提高应用程序 50%左右的区域着色器的性能。

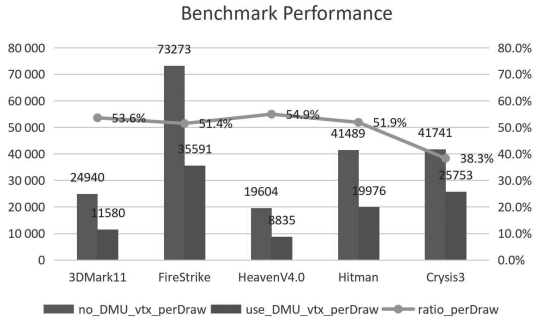

图 7　在应用程序中的性能提升

4　结束语

在目前的图形绘制流水线中，曲面细分技术 Tessellation 已经成为表现三维模型表面细节的重要步骤，目前的图形绘制平台，例如 Direct3D、Vulkan 引入了 TESS 单元，能够实时生成大量的顶点，构造为线段或者三角形等几何图元。这些顶点数量会很大，给后序的区域着色器 DS 的计算和存储带来较大压力。考虑到这些几何图元相互邻接，共享的顶点很多，我们提出了区域着色器的同顶点检测技术，可以减少顶点的重复计算。通过在区域线程构造器中增加一个匹配检测单元 DMU，对 TESS 的输入顶点进行域值的比较，发现有重复的顶点，直接返回该顶点的 dsid；只有 miss 的顶点，才会分配一个新的 dsid，并组装到当前 DS 线程中。实验表明，该方法得到的计算顶点数目，只相当于传统方法的 50%左右，DS 需要计算属性的顶点数目明显减少，总体性能也得到提高。

参考文献

[1] HENRY MORETON. Watertight Tessellation using Forward Differencing[C]. Proceedings of the ACM SIGGRAPH/EUROGRAPHICS workshop on Graphics hardware, ACM SIGGRAPH/Enrographics/ACM Press, 2001:25-32.

[2] FRANK LUNA. Introduction to 3D Game Programming with DirectX 11[M]. USA: Mercury Learning&Information, 2012:864.

[3] The Khronos Vulkan Working Group. Vulkan ® 1.4-A Specification (with all registered Vulkan extensions)[EB/OL]. (2025-08-29)[2025-09-16]. https://registry.khronos.org/vulkan/ specs/latest/html/vkspec.html

[4] 张兵强，张立民，艾祖亮，等. 屏幕空间自适应的地形 Tessellation 绘制[J]. 中国图象图形学报，2012，17(11):1431-1438.

[5] 袁岁维，万人民，刘世海. 某抛物面索网天线有限元模型建模方法研究[J]. 自动化技术与应用，2023，42(4):15-16,24.

[6] 李融，丁欣，等. 基于 GPU 的海量城市管线高效建模与实时绘制[J]. 计算机辅助设计与图形学学报，2015，29(4):597-604.

作者简介：张淮声（1976—），男，博士，高级工程师，研究方向：通用计算，GPU 架构设计与实时绘制。

通信作者：余　莉（1996—），女，副教授，研究方向：计算机图形学。

（上接第 21 页）

[13] 陈东键. 基于视觉的 RGV 自主引导系统设计[D]. 石家庄：石家庄铁道大学，2023.

[14] 王文刚. 基于 PLC 的轮对 RGV 输送车自动控制系统[J]. 自动化技术与应用，2023，42(9)：20-24.

[15] 苗旺龙. 基于 RFID 技术的铁路车轮堆垛机定位控制系统[J]. 自动化技术与应用，2023，42(5)：31-34.

[16] CHANG Lidan，LI Jianguo. Control Strategy of RGV Operation Blockage and Deadlock in Plane Mobile Stereo Garage[J]. IOPConference Series:Earth and Environmental Science，2021，719(2)：022013.

[17] 冯子龙. 一体化电梯门机驱动控制解决方案[J]. 自动化技术与应用，2023，42(9)：37-40.

作者简介：何长江（2000—），男，硕士研究生，研究方向：机械电子工程。

通信作者：周德强（1979—），副教授，硕士生导师，研究方向：机械电子工程。