3. 图形处理单元

> 显示器就是计算机。
> --黄仁勋

从历史上看，图形加速始于在重叠三角形的每个像素扫描线上插入颜色，然后显示这些值。包括访问图像数据的能力允许将纹理应用于表面。添加用于插值和测试z深度的硬件，可以提供内置的可见性检查。由于它们的频繁使用，这些工作被放到专门的硬件以提高性能。渲染管线的更多部分，以及每个部分的更多功能，在连续几代硬件产品中被添加。专用图形硬件相对于CPU的唯一计算优势是速度，但速度至关重要。

在过去的二十年中，图形硬件经历了令人难以置信的转变。第一个包含硬件顶点处理的消费类图形芯片（NVIDIA 的 GeForce256）于1999年发货。NVIDIA创造了图形处理单元(GPU)一词，以将GeForce256与之前可用的仅光栅化芯片区分开来，并且它坚持了下来。在接下来的几年里，GPU从复杂的固定功能管线的可配置实现发展到高度可编程的空白板，开发人员可以在其中实现自己的算法。各种可编程着色器是控制GPU的主要手段。为了提高效率，管线的某些部分仍然是可配置的，而不是可编程的，但趋势是可编程性和灵活性[175]。

GPU通过专注于一组高度并行化的任务而获得了极大的速度。例如，他们拥有专门用于实现z缓冲区、快速访问纹理图像和其他缓冲区以及查找哪些像素被三角形覆盖的定制芯片。第23章介绍了这些元素如何执行它们的功能。但是目前更重要的是，要尽早了解GPU如何实现其可编程着色器的并行性。

第3.3节解释了着色器的工作原理。现在，你需要知道的是，着色器核心是一个小型处理器，它执行一些相对独立的任务，例如将顶点从其在世界中的位置转换为屏幕坐标，或者计算被一个三角形覆盖的像素的颜色。每帧都有数千或数百万个三角形被发送到屏幕上，每秒可能有数十亿次着色器调用，即运行着色器程序的单独实例。

首先，延迟是所有处理器都面临的问题。访问数据需要一些时间。考虑延迟的基本方法是，信息离处理器越远，等待的时间就越长。第23.3节更详细地介绍了延迟。存储在内存芯片中的信息比本地寄存器中的信息需要更长的时间来访问。第18.4.1节更深入地讨论了内存访问。一个关键的问题是等待数据被检索意味着处理器停止，这会降低性能。