本文欢迎转载，但是请注明出处：<https://github.com/lygyue/Books>

认识显卡

大概2000年以前，显卡基本上只是一些2D显卡，我第一块3D显卡，是TNT2，16M显存。

国内3D的萌芽，应该在wow火了之后，在那之前，国内清一色的2d游戏，早期的时候，国内自主研发的网游，主要是大话西游，梦话西游等。当时我在深圳网域，自主研发了一款2d网游，叫《华夏》，底层用的是云风的《风魂》，已经是国内翘楚，后来腾讯收购了网域，后续推出了《华夏2》、《QQ华夏》，我在《华夏1》、《华夏2》都待过，那时候我还不知道什么是3D，正宗的菜逼一个。

WOW推出之后，国内很多人都觉得3D时代真正来临，当时网域的大佬们，一致认为我们该开发我们的3D游戏了，然后组建了当时公司最有能力的人，组建了全新的3d游戏团队，全力公关。

马后炮一下，当时国内压根没有3D的土壤，无论是程序还是美术。所以，这个团队最终失败，人员离职。这波人，后来重组了一个公司，叫《冰川网络》，现在是一家上市公司，混得还不错。我算毒舌的，看到不顺眼的都喜欢喷一下。但是我从来没有吐槽过《冰川网络》，原因无他，都是老同事，总归还是要有所顾忌的，冰川是个好公司：）。

在2000年以后，显卡一度被认为跟声卡、网卡一样，都是没什么大前途的。原因无他，大部分的显卡，其实无论是办公娱乐看片子，都足够了。所以，很多电脑的声卡、显卡、网卡都是集成到主板的，成了附属。

我相信那段时刻，黄教主一定很焦虑，一定觉得前途黑暗。后来，众所周知，黄教主押宝游戏，全力做游戏显卡。

再后来，ATI死了，NV成神了。

但是，即使在NV一统江湖的年代，整个公司的市值，也就是区区几十亿美金。这个市值，跟老黄的江湖地位完全不匹配。不要说什么几千亿的谷歌，苹果，连英特尔市值的1/10都不到。

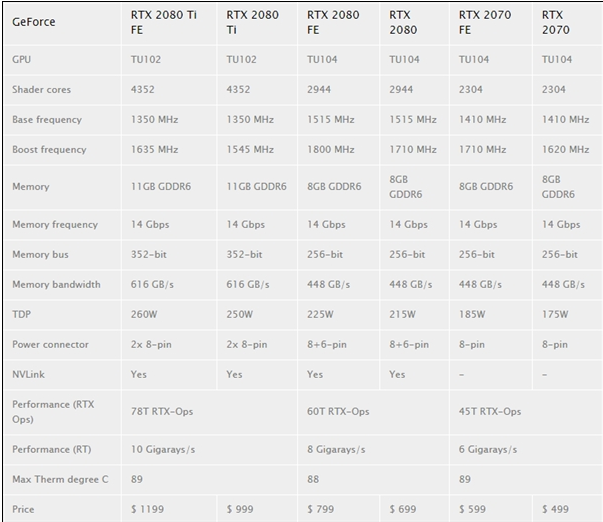
直到AI的崛起，英伟达终于扶摇直上，有了一线公司的底气。这是后话。

你所不知道的是：英伟达不光是一家硬件公司，更是一家全球顶尖的图形学公司。你们以为DX11，DX12是MS推出的，就以为MS很牛逼？你们以为UE4是全球顶尖的3D引擎，就以为Epic是全球顶尖的图形学公司？你们太天真。我估计，DX11也好，DX12也罢，都是MS跟NV联合一起搞的，谁出力更多还不好说。而坊间传言，UE4一缺钱，就找老黄要，还是有一定道理的。整个生态而言，UE4是NV很重要的一环。君不见，只要NV推出了什么新东西，UE4肯定没几天就出来支持了。不是一起搞的，能那么快？

想当初，DX12刚出来的时候，文档还没出多少，UE直接就实现了DX12的渲染，不是一起搞的，能那么快？所以说，整个图形学生态，从高到低，全在帝国主义手里。各位，千万别打算怼天怼地怼宇宙，你怼不起。

但凡看到国内吹牛逼说自主引擎，多牛逼多牛逼的，我都是冷冷一笑，不屑一顾。

好了，吹完了历史，该吹干货了。



看图。

一些烂大街的介绍，参数，这里就不打算细说了。例如工艺，例如显存，例如GPU频率。

从简单到复杂，先说一下GPU架构。这些年，看着NV一代一代的更换GPU架构，用了一个又一个科学大神的名字，很多人容易看花眼，也不知道该看什么。

其实很简单，看你需要什么。如果你是一个图形学菜鸟，就做做一般的开发，那么这东西跟你毫无意义。随便上一张显卡，基本都能满足你的需求。但是，如果你在做一些比较新潮的东西，就要小心了，必须要有硬件支持，没有硬件支持，等死吧。

例如，你要做一个视频播放器，专业支持8K的视频播放，那么，首先，你必须要直到哪些显卡是支持8K的。如图：



随便上NV官网查一查，你就会知道，只有1050以上的显卡，才能支持8k。当然了，除了显卡，还需要操作系统的支持，这个必须win10，你可以从其他渠道查得到。

再如，你打算做最潮流的RTX开发，那么，你必须查找哪款显卡，哪个架构，是硬件上支持的，否则很容易坑逼。

一句话，这个架构只是一个名字，这个名字背后支持了什么技术，随便查查即可，不需要太上心。

很简单？没意思？好吧，这只是开胃菜，下面需要隆重介绍一下，显存。

先别喷，这次我不讲显存的大小，那个大小就只是一个大小，没太大研究价值，**重点是这三个：显存频率，显存位宽，显存带宽**。



在我刚开始搞图形学的时候，我完全没有研究过这块，也完全不了解这块的意义。

在我搞了图形学好几年之后，我已经会独立完成一个延迟渲染了，我以为我已经入门了，我才开始研究这一块，才知道了显卡的很多不为人知的东东。

先说一些结论：

**显存带宽 = 显存频率 \* 显存位宽 / 8**

看上图，616Gb/s = 14GB/s \* 352 / 8。除以8很好理解，一个字节8位。

以前菜逼看显卡参数的时候，一直不知道这个有什么用。这里解释一下。

首先，你的显存是需要保存数据的。例如模型的顶点数据，索引数据，纹理数据，这些数据，都是要从内存传到显存的。这个的意思是：你理论上每秒钟可以传入616G的数据数据到显存。

是不是很大？我就算达不到理论值，我砍掉一半，是不是也有300G，好牛逼！

我以前也是这么以为的。有个几十G，不是够了吗，为什么越做越大，我没有那么多数据需要传过去啊。

太天真。

有以下几个问题（这部分全是我个人理解，没有任何资料佐证，不保证绝对正确）：

1. 显存带宽不仅仅是贴图，顶点等等的传入，还包括其他很多，例如骨骼动画的数据，比你想象的更大。例如**渲染到纹理。专业术语叫redner to texture。简称RTT。**
2. 显存操作不是瞬时操作，而是耗时操作。例如你一秒钟传几十G数据，这个时候，你其他什么也不用干了。不是说你一边传了几十G数据，一边还在进行大量的渲染计算。一句话，这个显存的读写，是个耗时操作。
3. 别人我不知道，我以前的理解，潜意识里 认为显存的写入，是像流水一样，不断的写。后来才发现，其实不是流水，是打气筒。一次一次的打气，你要么一次打的气更多，要么你打气的频率更快。这就是显存位宽，显存频率的意思。

所以，为什么说延迟渲染的一个弊端，是占用了大量的显存带宽？随便做一下计算。假设你渲染一个1920 \* 1080的全屏，你的GBuffer使用了3个Render Target（最少都要2个，3个很克制了，我听过7个的）。那么一帧的GBuffer需要用到的带宽是：1920 \* 1080 \* 4 \* 3 = 23M

。你以为这就够了？假设你一秒钟渲染60帧，还要乘以60.那么就是23 \* 60 = 1.4G。

看到了吗，看着几百G，但是就一个最简单的延迟渲染，就去掉了你1.4G。如果你还有几个实时阴影图，又是要增加几倍。所以，在老一点几十G显存带宽的显卡里，瞬间就摇摇欲坠了。

除了RTT，还有一个非常占用显存带宽的东东，那就是矩阵。一个矩阵是4 \* 4，16个浮点数，64字节。看起来不多对吗？但是架不住量大。

先算个比较一般的Draw Call，我就算3000.那么这里就有3000个矩阵需要传进去。假设是一个大型游戏，同屏有100人，一个人的骨骼算64块。那么这里的矩阵数是100 \* 64 = 6400.

好吧，你还得算每秒钟60帧吧。30帧的游戏，太low了点啊。什么，你一次只传一个矩阵就够了，其他的不用传？不现实啊。其他参数总是要传一点的啊。那么我算翻一倍，算128字节。那么带宽的消耗就变成了：

128 \* (6400 + 3000) \* 60 = 68M。

看起来不多啊，还能接受啊。

没那么简单。

首先，有一个叫**硬件蒙皮**的概念。因为涉及到大量的计算，不使用硬件蒙皮的话（实质就是用硬件来计算），效率低。但是，一块骨头，很可能不仅仅是只受到一块骨头的影响，是会叠加的。例如你的身体动了，手肯定也跟着动。你手臂动了，手指肯定也跟着动。

这些细节，以后可能在骨骼动画这块会再细讲。现在先不讲。

总之，硬件蒙皮，据我所知，是大量消耗显存带宽的东西。我听说有自己在shader里面封装四元数的计算，传入四元数的。我没有这么干过。按道理是可以的。

回到显存。

熟悉了显存频率，显存带宽，显存位宽，知道了这些数据的重要性。那么，时候熟悉一下硬件了。

介绍一下常见的显存类型，目前主流的显存类型，主要有DDR5，DDR6，HBM，HBM2，HBM3之流。

我们的目的，是需要最快的显存。那么，根据公式，要么提升显存的频率，要么提升显存的位宽。

可想而知，提升显存的频率，终究会到瓶颈。因此，一大堆公司开始研究新的东西，致力于提升显存的位宽。这就是HBM的初衷。

**一般的DDR显存，位宽都是128，256，而HBM的位宽，貌似可以轻易去到1024.那么同等频率下，显存带宽一下子提升了几倍。但是，事实上，HBM的频率，并没有那么高。所以这个提升，其实要小一些。**

网上早几年就吹嘘HBM是大势所趋，而ATI，NV的各种高端卡，都上了HBM2，给人一种感觉，DDR完蛋了。然而，最新的GTX2080之流，居然全是DDR6。真的没看懂，难道又有了什么我不知道的PY交易了吗？又或者DDR有什么法子起死回生？有知道的大佬们可以科普一下，这个我真的不知道。

好了，显存的科普到此为止，讲讲GPU。

CPU跟GPU有什么区别？正确来说，有什么优劣？什么时候用GPU，而什么时候用CPU？

先假设你对CPU已经比较了解，我尽量精简来讲，免得被喷水。

简单来说，CPU就是一个高速执行各种指令的一个高速计算器。你一边看电影，一边写文档 ，还一边看网页，一边下载，其实是CPU不断在不同的进程之间切换（假设是最简单的单核CPU）。你感觉不出来这个切换，那是因为CPU速度太快了。

所以从设计上看，CPU就不是往多线程的路子走的，因为线程的同步，其实开销很大，还容易出错。

而GPU是反过来的。GPU设计上，就是为了多线程。那么线程同步呢？当然是尽量不要同步的，真要同步了，也有办法，但是会降低效率。

所以在开局的CS5（compute shader5）上，单个线程组，就最高支持1024个线程。假设我现在要在屏幕上画几十万个三角形，你说是一个for循环，循环几十万次，里面一个一个画三角形快，还是我直接1024个线程，一次画1024个三角形更快？结果不言而喻。

所以，从这个解释，可以轻易明白了：GPU只适用于那些大量重复的，一样的计算。例如你要画个三角形，然后打几个字，然后放段音乐，再打开个网页玩玩，再画下一个三角形。这种需求，用GPU那就是坑逼了。

什么？你以为这就是真的坑逼了？真没有。老黄表示不服。

老黄表示，我可以在硬件层实现一些固定需求，然后再开放一些接口，让你们来实现定制需求。固定需求有哪些？例如视频解码啊，我现在可以直接硬件解码8K，你用CPU解码给我看看？光是一个NV12转RGB就算死你，小样！

所以，我们一方面看到，GPU集成了越来越多的固定需求，然后开放了各种接口，让大伙实现自己的需求。最早的AI训练，用CPU欲仙欲死，然后就有人开始用了GPU，然后NV的股价短短时间上涨了十几倍。

GPU集成了的需求，常见的有：

视频编码，解码。你不要以为你会用个FFMEPG就牛逼了。老实说，FFMPEG真的不牛逼，黄教主才是真牛逼。

**MSAA抗锯齿**。这个后续讲到抗锯齿的时候再专门讲，这里不详细讲。

**纹理硬解。你可以这样理解，一般的windows程序，你写个3d渲染，用什么jpg，png，就是占用了硬盘空间小了。在显存里，还是会直接解压成位图，一点也不小。但是，如果你用了DDS格式，那么，这个在显存里还是占用那么小！因此，一切不用DDS作为端游的纹理格式，都是值得怀疑的，除非你真的有其他特殊需要。这块会在纹理模块详细再讲。**

物理加速。很多年前，物理引擎各路开花，有什么haveok，有什么牛顿，有什么physX，还有一大堆小型的其他引擎。到了今天，大家慢慢发现，所有跟老黄作对的，基本不死也残了。因为老黄把physX集成进了硬件，你其他的，就算用使用CUDA之类的进行硬件加速，也很难干得过别人老黄的啊，况且还有UE4大力加持。说多了都是泪。

海飞丝系统（hairworks）。说穿了，一大堆游戏公司说，现在人物的头发渲染不行啊，效果不行啊，效率也低，怎么破？老黄说，让我来，我从底层实现一套系统，给大家用，不好用我不收钱。然后，ATI悲剧了。据说当初巫师3头发渲染，ATI没支持，然后被喷成狗？我不是游戏硬核玩家，只是听说，别打我。不过由此可见NV已经要成了游戏的标配。如果不是反垄断的要求，我估计Intel，NV早就可以干死AMD了，现在都是留着遛狗一样的感觉，我是AMD我都心塞。

除了这些，估计还有其他一堆。例如深度学习相关的就集成了一大堆。这块我不熟，不评价，不过以老黄的风格，不集成不现实，我蒙都能蒙对：）。

好了，显卡相关的先讲到这里，以后想到其他的，再补充。我尽量精简，不打算大篇幅的水。