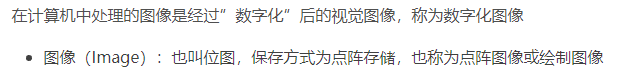
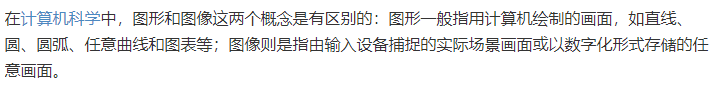
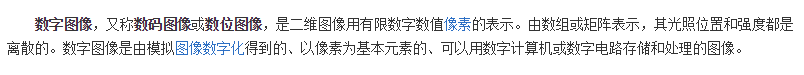
图像基础

什么是计算机图像？这是个好问题。我看过无数的定义，下面随便抄几个出来：







菜逼们，看到这几个解释，明白了吗？清楚了吗？不清楚？背下来，考试的时候这就是标准答案！

可能是我天赋不大够，老实说，我的图形图像之路充满了坎坷。看到以上这类解释，对我而言，就是用一些我不懂的术语，去解释另外一个我不懂的东西。就像第一个解释，刚入门的菜鸟，是不是得再去深入理解什么是“点阵”？点阵储存是个什么鬼？

第二个解释，“以数字化的形式储存”，怎么以数字化的形式储存？

第三个解释，是百科上面的解释，别人能不能理解我不知道。如果在我刚入门的时候去解释这个东西，我只能是一头雾水。

这里，我决定用我自己的语言和脉络，来解释一下什么是图像。解释图像之前，我来先对现代科技的一些基本的概念做一些我自己的阐述，不保证正确。

我认为，现代科技的一个重要基础，就是量化/数字化。什么是量化，而为什么需要量化？举例：在没有量化的时候，今天冷不冷？冷！多冷？好冷！

而量化了的时候呢？今天冷不冷？冷，多冷？-30度。

现代科技的基础，是把距离、重量、力量、能量、温度、角度、气压、电流……一切你能想到的东西，做了量化/数字化。只有数字化以后，才能参与计算。

图像的实质，是颜色的量化！

颜色的量化有好几种方式，最常见的是RGB，其他较常用的还有HSL/HSV，这个主要是给美术用的。RGB是纯数字，美术看不懂，但是美术能看懂色度，饱和度，亮度，所以增加了一个这东东，直接套用公式转换即可。还有视频里面常见的YUV，诸如此类的东西一大堆，这里不做详细的介绍，这里主要讲RGB。

RGB的意思是：任何颜色可以分解成红、绿、蓝的混合。这就是色光三原色原理。那么问题来了，颜色是如何保存的？这里，我大概介绍一下颜色的发展历程。

大概在1996年，我开始见到第一台286电脑。那会显示器是黑白显示器，颜色只有黑白两种颜色，编程的时候，颜色的描述只有两个数字：0和1. 0是黑色，1是白色。

接下来，在386的年代，用Qbasic编程，颜色就多了一些，我最早的时候，用过16色编程，0是黑色，1是蓝色，4是红色，15是白色（年代久远，有可能记错了）。我第一个比较大型的网络游戏，那是1997年用Qbasic写过一个中国象棋，基于Novell网络。

接下来的几年，个人计算机迅猛发展，短短几年时间，从16色就进入了16位色，32位色的年代。

这里，只讲解一下 16位色，然后估计很容易久能想明白32位色，8位色。回到之前讲过的C++基础，16位就是2字节。那么颜色是怎么保存的？可以是565，5551，655，556，如果是自己做存取的话，你想怎么搞怎么搞。通用的话我用最多的是565，5551.后者带一个alpha通道。

01111 110011 11001

红 绿 蓝

所以，16位色理论上能描述的颜色数是：

2^5 \* 2^6 \* 2^5 = 2^16，也就是65535种颜色。

那么，32位能描述多少颜色？是2^32那么多吗？不是的，能描述的是2^24那么多。所以说，24位颜色跟32位颜色其实是一样的，只是32位颜色多了一个8位的alpha通道。

这里，讲讲Alpha（透明度）。我刚入门的时候，那还是十几年前。美术讲什么alpha，我一头雾水，只知道这东西是透明值。只是听说是0-255之间。至于透明度是如何起作用的，完全不知道。

在现在各种资料满天飞的年代，理解这类东西容易了太多。

首先，alpha是透明度的量化！可以是0，1两个数字，可以是2位（8位的颜色图，就有2222的格式，用2位描述透明度），可以是8位。

透明度只有在混合的时候才有意义。计算公式非常简单：假设一个8位的alpha，透明度为100，当前图片颜色为Col1，背景颜色为Cold

，公式为：

Col = Col1 \* (float(100) / float(255)) + Col2 \* (float(255 - 100) / float(255));

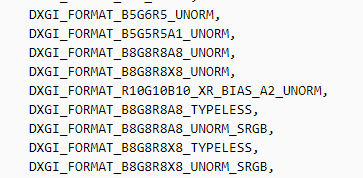
什么？这么简单的公式还复杂？那么我们分开两步：

Float alpha = 100 .0f/ 255.0f;

Col = Col1 \* alpha + Col2 \* (1.0f - alpha);

如果这么简单的公式都看不明白，放弃吧，图形学绝对跟你无缘。

看明白了这样简单的数据储存，很多看起来牛逼哄哄高大上的东西，就会发现也就那么回事。



看图，上面两个就是16位颜色图，一个是565，一个是5551.下面的都是各种32位标准颜色图。至于后面的什么typeless之类的，无非就是存储的格式罢了。例如：你创建了一张贴图，等于申请了一块显存，那么你存储的时候，可以是以整数储存，可以是无符号整数，也可以是无类型，直接memcpy，后续再指定类型。一般情况下，你用不到那么多。

DXGI定义了上百种颜色，其实一般人用到的不多，后续讲到纹理、贴图相关内容的时候再详细描述，这里还是回到图像的基础原理。