

RGB是按三基色加光系统的原理来描述颜色

RGB（红绿蓝）色彩模式，R 代表 Red（红色），G 代表 Green（绿色），B 代表 Blue（蓝色）

科学家发现 这三种颜色（不同比例）可以合成其它任何一种颜色，因此 使用RGB 色彩模式来表示图像

图像 可以使用点阵的方式 来显示！每个点 就是一个像素点！

分辨率 就是点阵的规模，1280 x 720 就是长 1280，宽 720 的点阵

每个像素点 都由 RGB 合成，

保存成文件 就是图片，多张图片不断切换就是视频

BMP文件，是图像原始数据，除了BMP 头，就是原始数据

JPG等，都进行了压缩处理

RGB 888，表示 R,G,B 分别使用 8位表示（就是一个字节），一起 就是 24位（24位宽），3个字节，一副 1280 x 720 的图像的原始数据 就是 1280x720x 3

RGB565，表示 R 使用 5位表示，G 使用6位表示，B使用 5位表示，一起就是 16位（16位宽），2个字节，

其它 RGB555 类似

ARGB：A 表示 透明的意思

ARGB1555 表示 A 使用 1位（是否透明）表示，RGB 分别使用 5位表示

数据格式：

24bit RGB888

24ibt RGB888 R7 R6 R5 R4 R3 R2 R1 R0 G7 G6 G5 G4 G3 G2 G1 G0 B7 B6 B5 B4 B3 B2 B1 B0

16bit ARGB1555

16ibt RGB888 A0 R4 R3 R2 R1 R0 G4 G3 G2 G1 G0 B4 B3 B2 B1 B0

按照 ARGB 的顺序来的

RGB 之间转换

RGB888 转 RGB565

RGB 565 使用 5 位来 表示 RGB888 8位的 R（红色），使用 6 位 表示 原本 8位的 G 这里就需要 量化压缩 把 8位数据 压缩成 5位

规则：取高位（只需要 5位，就将 原始数据 8位的 R，取前 5位）其他类似

将获取到的 RGB 数据，按照 RGB565 的顺序排列好就行了  
量化上做了压缩，却损失了精度。

RGB565 转 RGB888

RGB888 使用 8位来表示 RGB565 的 R（红色）

量化补偿的方法：

1. 将原数据填充至高位
2. 对于低位，用原始数据的低位进行补偿
3. 如果仍然有未填充的位，继续使用原始数据的低位进行循环补偿

量化补偿，举例：

16bit RGB565 -> 24bit RGB888 的转换

16bit RGB565 R4 R3 R2 R1 R0 G5 G4 G3 G2 G1 G0 B4 B3 B2 B1 B0

24bit RGB888 R4 R3 R2 R1 R0 0 0 0 G5 G4 G3 G2 G1 G0 0 0 B4 B3 B2 B1 B0 0 0 0

24bit RGB888 R4 R3 R2 R1 R0 R2 R1 R0 G5 G4 G3 G2 G1 G0 G1 G0 B4 B3 B2 B1 B0 B2  
B1 B0