个人资料

yapingmcu

访问：197323次
积分：2095
等级：BLOG 5
排名：第13651名

原创：2篇 转载：119篇
译文：0篇 评论：8条

YUV格式详解

标签：yuv 格式 采样 YUYV YUY2

2015-11-25 10:59 202人阅读 评论(0) 收藏 举报

分类： camera（15）

目录(?) [+]

YUV是指亮度参量和色度参量分开表示的像素格式，而这样分开的好处就是不但可以避免相互干扰，还可以降低色度的采样率而不会对图像质量影响太大。YUV是一个比较笼统地说法，针对它的具体排列方式，可以分为很多种具体的格式。转载一篇对yuv格式解释的比较清楚地文章，也可以直接参考微软的那篇文章。

对于YUV格式，比较原始的讲解是MPEG-2 VIDEO部分的解释，当然后来微软有一个比较经典的解释，中文的大多是翻译这篇文章的。文章来源：http://msdn.microsoft.com/en-us/library/aa904813(VS.80).aspx

这里转载有人已经翻译过的，个人认为已经翻译的很不错了，遂放弃翻译。

http://hondrif82q.spaces.live.com/blog/cns!776E82726DE60562!177.entry

http://hondrif82q.spaces.live.com/blog/cns!776E82726DE60562!178.entry

YUV格式解析1（播放器——project2）

根据板卡api设计实现yuv420格式的视频播放器

打开*.mp4;*.264类型的文件，实现其播放。

使用的视频格式是YUV420格式

YUV格式通常有两大类：打包（packed）格式和平面（planar）格式。前者将YUV分量存放在同一个数组中，通常是几个相邻的像素组成一个宏像素（macro-pixel）；而后者使用三个数组分开存放YUV三个分量，就像是一个三维平面一样。表2.3中的YUY2到Y211都是打包格式，而IF09到YVU9都是平面格式。（注意：在介绍各种具体格式时，YUV各分量都会带有下标，如Y0、U0、V0表示第一个像素的YUV分量，Y1、U1、V1表示第二个像素的YUV分量，以此类推。）

MEDIASUBTYPE_YUY2 **YUY2**格式，以4:2:2方式打包

MEDIASUBTYPE_YUYV **YUYV**格式（实际格式与YUY2相同）

MEDIASUBTYPE_YVYU **YVYU**格式，以4:2:2方式打包

MEDIASUBTYPE_UYVY **UYVY**格式，以4:2:2方式打包

MEDIASUBTYPE_AYUV 带Alpha通道的4:4:4 **YUV**格式

MEDIASUBTYPE_Y41P **Y41P**格式，以4:1:1方式打包

MEDIASUBTYPE_Y411 **Y411**格式（实际格式与Y41P相同）

MEDIASUBTYPE_Y211 **Y211**格式

MEDIASUBTYPE_IF09 IF09格式

MEDIASUBTYPE_IYUV IYUV格式

MEDIASUBTYPE_YV12 YV12格式

MEDIASUBTYPE_YVU9 YVU9格式

表2.3

YUV 采样

(13688)
手机的AP和BP是什么?那
(11845)
针对WIN7系统装上驱动后 (8865)
RGB/HSV/YUV颜色空间 (8726)
c,c++里面，头文件里面的 (7853)
TRACE32调试技巧 (6159)
指令STMFD和LDMFD分 (5058)
高通Android智能平台环境 (4811)
Ubuntu下代替dos2unix命 (3947)

评论排行

裸机开发学习心得 (3)
高通Android智能平台环境 (1)
MIPI概述 (1)
针对WIN7系统装上驱动后 (1)
camera调试工具 (1)
Android USB VID PID 及 (1)
jpg图片的Exif及gps信息读 (0)
指令STMFD和LDMFD分 (0)
arm堆栈知识 (0)
S3C2440-启动分析 (0)

推荐文章

* 2016 年最受欢迎的编程语言是什么？
* Chromium扩展(Extension)的页面(Page)加载过程分析
* Android Studio 2.2 来啦
* 手把手教你做音乐播放器(二)技术原理与框架设计
* JVM 性能调优实战之：使用阿里开源工具 TProfiler 在海量业务代码中精确定位性能代码

最新评论

MIPI概述
滔滔江水: 讲的比较详细，受教了！！
Android USB VID PID 及 ADB (
A_XCODE_TEACHER: Great
camera调试工具
Brady的博客: camera驱动调试、技术交流、摄影爱好、兴趣爱好群:330777216
针对WIN7系统装上驱动后，设备
上课讲话讲话: 驱动在哪里？
高通Android智能平台环境搭建_
sypzz: good things
裸机开发学习心得
浙妞: 有些不赞同，但总体说得挺好。mark
裸机开发学习心得
zhang835705223: mark
裸机开发学习心得
借我你的一生_yikai: 赞...

YUV 的优点之一是，色度频道的采样率可比 Y 频道低，同时不会明显降低视觉质量。有一种表示法可用来描述 U 和 V 与 Y 的采样频率比例，这个表示法称为 A:B:C 表示法：

- 4:4:4 表示色度频道没有下采样。
- 4:2:2 表示 2:1 的水平下采样，没有垂直下采样。对于每两个 U 样例或 V 样例，每个扫描行都包含四个 Y 样例。
- 4:2:0 表示 2:1 的水平下采样，2:1 的垂直下采样。
- 4:1:1 表示 4:1 的水平下采样，没有垂直下采样。对于每个 U 样例或 V 样例，每个扫描行都包含四个 Y 样例。与其他格式相比，4:1:1 采样不太常用，本文不对其进行详细讨论。

图 1 显示了 4:4:4 图片中使用的采样网格。灯光样例用叉来表示，色度样例则用圈表示。

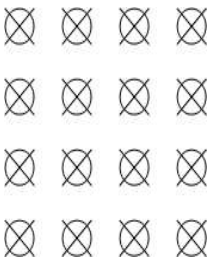


图 1. YUV 4:4:4 样例位置

4:2:2 采样的这种主要形式在 ITU-R Recommendation BT.601 中进行了定义。图 2 显示了此标准定义的采样网格。

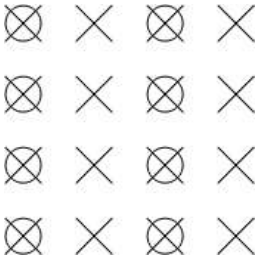


图 2. YUV 4:2:2 样例位置

4:2:0 采样有两种常见的变化形式。其中一种形式用于 MPEG-2 视频，另一种形式用于 MPEG-1 以及 ITU-T recommendations H.261 和 H.263。图 3 显示了 MPEG-1 方案中使用的采样网格，图 4 显示了 MPEG-2 方案中使用的采样网格。

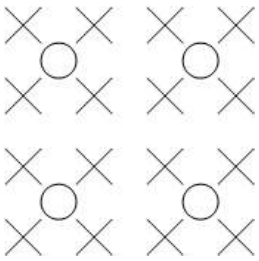


图 3. YUV 4:2:0 样例位置（MPEG-1 方案）

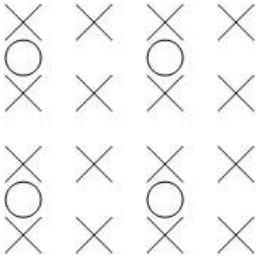


图 4. YUV 4:2:0 样例位置
（MPEG-2 方案）

与 MPEG-1 方案相比，在 MPEG-2 方案与为 4:2:2 和 4:4:4 格式定义的采样网格之间进行转换更简单一些。因此，在 Windows 中首选 MPEG-2 方案，应该考虑将其作为 4:2:0 格式的默认转换方案。

表面定义

本节讲述推荐用于视频呈现的 8 位 YUV 格式。这些格式可以分为几个类别：

- 4:4:4 格式，每像素 32 位
- 4:2:2 格式，每像素 16 位
- 4:2:0 格式，每像素 16 位
- 4:2:0 格式，每像素 12 位

首先，您应该理解下列概念，这样才能理解接下来的内容：

- 表面原点。对于本文讲述的 YUV 格式，原点 (0,0) 总是位于表面的左上角。
- 跨距。表面的跨距，有时也称为间距，指的是表面的宽度，以字节数表示。对于一个表面原点位于左上角的表面来说，跨距总是正数。
- 对齐。表面的对齐是根据图形显示驱动程序的不同而定的。表面始终应该 DWORD 对齐，就是说，表面中的各个行肯定都是从 32 位 (DWORD) 边界开始的。对齐可以大于 32 位，但具体取决于硬件的需求。
- 打包格式与平面格式。YUV 格式可以分为打包 格式和平面 格式。在打包格式中，Y、U 和 V 组件存储在一个数组中。像素被组织到了一些巨像素组中，巨像素组的布局取决于格式。在平面格式中，Y、U 和 V 组件作为三个单独的平面进行存储。

4:4:4 格式，每像素 32 位

推荐一个 4:4:4 格式，FOURCC 码为 AYUV。这是一个打包格式，其中每个像素都被编码为四个连续字节，其组织顺序如下所示。

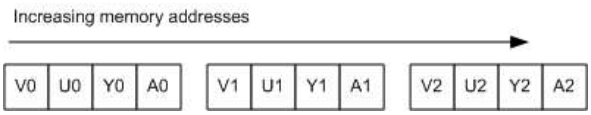


图 5. AYUV 内存布局

标记了 A 的字节包含 alpha 的值。

4:2:2 格式，每像素 16 位

支持两个 4:2:2 格式，FOURCC 码如下：

- YUY2
- UYVY

两个都是打包格式，其中每个巨像素都是编码为四个连续字节的两个像素。这样会使得色度水平下采样乘以系数 2。

YUY2

在 YUY2 格式中，数据可被视为一个不带正负号的 **char** 值组成的数组，其中第一个字节包含第一个 Y 样例，第二个字节包含第一个 U (Cb) 样例，第三个字节包含第二个 Y 样例，第四个字节包含第一个 V (Cr) 样例，如图 6 所示。

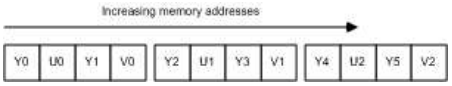


图 6. YUY2 内存布局

如果该图像被看作由两个 little-endian **WORD** 值组成的数组，则第一个 **WORD** 在最低有效位 (LSB) 中包含 Y0，在最高有效位 (MSB) 中包含 U。第二个 **WORD** 在 LSB 中包含 Y1，在 MSB 中包含 V。

YUY2 是用于 Microsoft DirectX® Video Acceleration (DirectX VA) 的首选 4:2:2 像素格式。预期它会成为支持 4:2:2 视频的 DirectX VA 加速器的中期要求。

UYVY

此格式与 YUY2 相同，只是字节顺序是与之相反的 — 就是说，色度字节和灯光字节是翻转的（图 7）。如果该图像被看作由两个 little-endian **WORD** 值组成的数组，则第一个 **WORD** 在 LSB 中包含 U，在 MSB 中包含 Y0，第二个 **WORD** 在 LSB 中包含 V，在 MSB 中包含 Y1。

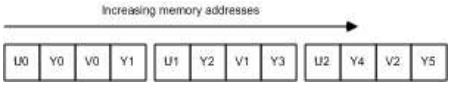


图 7. UYVY 内存布局

4:2:0 格式，每像素 **16** 位

推荐两个 4:2:0 每像素 16 位格式，FOURCC 码如下：

- IMC1
- IMC3

两个 FOURCC 码都是平面格式。色度频道在水平方向和垂直方向上都要以系数 2 来进行再次采样。

IMC1

所有 Y 样例都会作为不带正负号的 **char** 值组成的数组首先显示在内存中。后面跟着所有 V (Cr) 样例，然后是所有 U (Cb) 样例。V 和 U 平面与 Y 平面具有相同的跨距，从而生成如图 8 所示的内存的未使用区域。

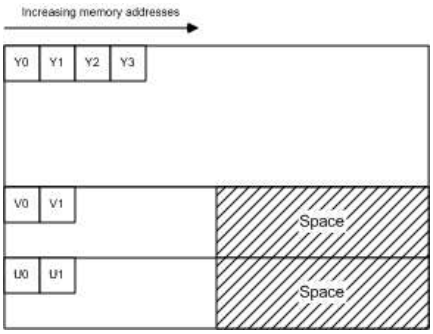


图 8. IMC1 内存布局

IMC3

此格式与 IMC1 相同，只是 U 和 V 平面进行了交换：

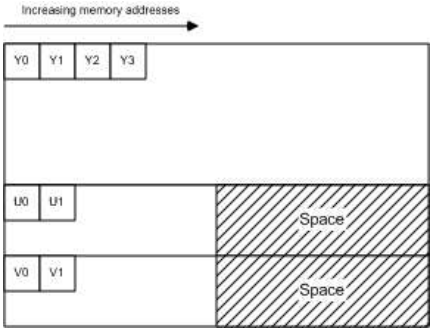


图 9. IMC3 内存布局

4:2:0 格式，每像素 **12** 位

推荐四个 4:2:0 每像素 12 位格式，FOURCC 码如下：

- IMC2
- IMC4
- YV12
- NV12

在所有这些格式中，色度频道在水平方向和垂直方向上都要以系数 2 来进行再次采样。

IMC2

此格式与 IMC1 相同，只是 V (Cr) 和 U (Cb) 行在半跨距边界处进行了交错。换句话说，就是色度区域中的每个完整跨距行都以一行 V 样例开始，然后是一行在下一个半跨距边界处开始的 U 样例（图 10）。此布局与 IMC1 相比，能够更加高效地利用地址空间。它的色度地址空间缩小了一半，因此整体地址空间缩小了 25%。在各个 4:2:0 格式中，IMC2 是第二首选格式，排在 NV12 之后。

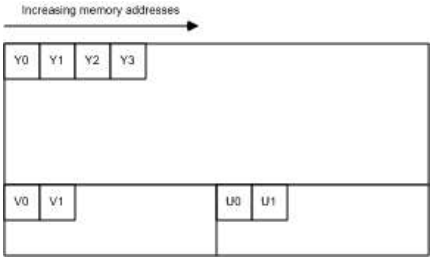


图 10. IMC2 内存布局

IMC4

此格式与 IMC2 相同，只是 U (Cb) 和 V (Cr) 行进行了交换：

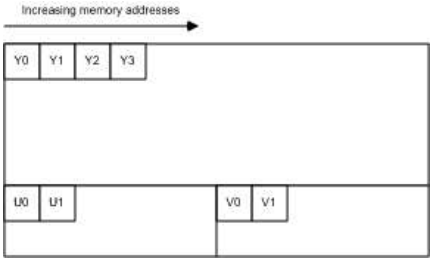


图 11. IMC4 内存布局

YV12

所有 Y 样例都会作为不带正负号的 **char** 值组成的数组首先显示在内存中。此数组后面紧接着所有 V (Cr) 样例。V 平面的跨距为 Y 平面跨距的一半，V 平面包含的行为 Y 平面包含行的一半。V 平面后面紧接着所有 U (Cb) 样例，它的

跨距和行数与 V 平面相同(图 12)。

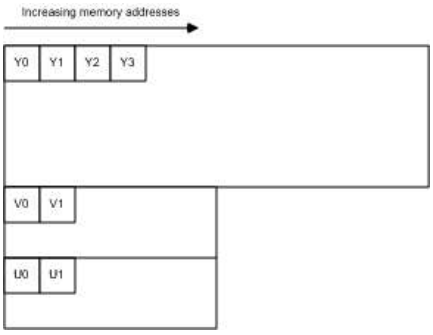
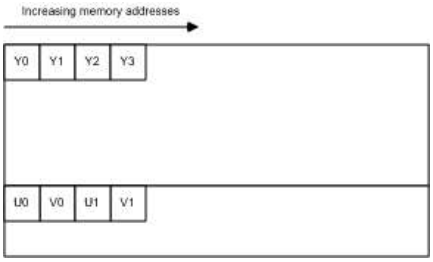


图 12. YV12 内存布局

NV12

所有 Y 样例都会作为由不带正负号的 **char** 值组成的数组首先显示在内存中，并且行数为偶数。Y 平面后面紧接着一个由不带正负号的 **char** 值组成的数组，其中包含了打包的 U (Cb) 和 V (Cr) 样例，如图 13 所示。当组合的 U-V 数组被视为一个由 little-endian **WORD** 值组成的数组时，LSB 包含 U 值，MSB 包含 V 值。NV12 是用于 DirectX VA 的首选 4:2:0 像素格式。预期它会成为支持 4:2:0 视频的 DirectX VA 加速器的中期要求。



YUV格式解析2

又确认了一下H264的视频格式——H264支持4：2：0的连续或隔行视频的编码和解码

YUV(亦称YCrCb)是被欧洲电视系统所采用的一种颜色编码方法(属于PAL)。YUV主要用于优化彩色视频信号的传输，使其向后兼容老式黑白电视。与RGB视频信号传输相比，它最大的优点在于只需占用极少的带宽(RGB要求三个独立的视频信号同时传输)。其中“Y”表示明亮度(Luminance或Luma)，也就是灰阶值；而“U”和“V”表示的则是色度(Chrominance或Chroma)，作用是描述影像色彩及饱和度，用于指定像素的颜色。“亮度”是通过RGB输入信号来创建的，方法是将RGB信号的特定部分叠加到一起。“色度”则定义了颜色的两个方面—色调与饱和度，分别用Cr和CB来表示。其中，Cr反映了GB输入信号红色部分与RGB信号亮度值之间的差异。而CB反映的是RGB输入信号蓝色部分与RGB信号亮度值之同的差异。

补充一下场的概念——

场的概念不是从DV才开始有的，电视系统已经有了(当然，DV和电视的关系大家都知道)归根结底还是扫描的问题，具体到PAL制式是：

每秒25帧，每帧两场，扫描线(包括电视机的电子束)自上而下先扫描一场，然后再自上而下扫描第二场

之所以引入场的概念，我的理解是主要为了在有限的带宽和成本内使画面运动更加平滑和消除闪烁感。

这两个场的扫描线是一条一条互相间隔开的，比如说对于一个帧来讲，最上面一条线编号为0，紧挨着的是1，再下来是2，3，4，5，6。。。。那么第一场也许是0，2，4，6；也许是1，3，5，7——这就是隔行扫描

在逐行扫描模式下，就是扫描线按照0，1，2，3，4，5的顺序依次扫描，很明显，这时候就不存在场的概念了。

下面区分一下YUV和YCbCr

YUV色彩模型来源于RGB模型，

该模型的特点是将亮度和色度分离开，从而适合于图像处理领域。

应用：模拟领域

Y' = 0.299*R' + 0.587*G' + 0.114*B'

U' = -0.147*R' - 0.289*G' + 0.436*B' = 0.492*(B' - Y')

V' = 0.615*R' - 0.515*G' - 0.100*B' = 0.877*(R' - Y')

R' = Y' + 1.140*V'

G' = Y' - 0.394*U' - 0.581*V'

B' = Y' + 2.032*U'

YCbCr模型来源于YUV模型。YCbCr是 YUV 颜色空间的偏移版本.

应用：数字视频，ITU-R BT.601建议

Y' = 0.257*R' + 0.504*G' + 0.098*B' + 16

Cb' = -0.148*R' - 0.291*G' + 0.439*B' + 128

Cr' = 0.439*R' - 0.368*G' - 0.071*B' + 128

R' = 1.164*(Y'-16) + 1.596*(Cr'-128)

G' = 1.164*(Y'-16) - 0.813*(Cr'-128) - 0.392*(Cb'-128)

B' = 1.164*(Y'-16) + 2.017*(Cb'-128)

PS：上面各个符号都带了一撇，表示该符号在原值基础上进行了伽马校正,伽马校正有助于弥补在抗[illegible]线性分配伽马值所带来的细节损失，使图像细节更加丰富。在没有采用伽马校正的情况下，暗部细节不容易显现出来，而采用了这一图像增强技术以后，图像的层次更加明晰了。

所以说H264里面的YUV应属于YCbCr.

下面再仔细谈谈YUV格式，YUV格式通常有两大类：打包（packed）格式和平面（planar）格式。前者将YUV分量存放在同一个数组中，通常是几个相邻的像素组成一个宏像素（macro-pixel）；而后者使用三个数组分开存放YUV三个分量，就像是一个三维平面一样。

我们常说得YUV420属于planar格式的YUV，颜色比例如下:

Y0U0V0 Y1

Y2U2V2 Y3

Y4 Y5

Y6 Y7

Y8U8V8 Y9

Y10U10V10 Y11

Y12 Y13

Y14 Y15

其他格式YUV可以点这里查看详细内容，而在YUV文件中YUV420又是怎么存储的呢？在常见H264测试的YUV序列中,例如CIF图像大小的YUV序列(352*288),在文件开始并没有文件头,直接就是YUV数据,先存第一帧的Y信息,长度为352*288个byte，然后是第一帧U信息长度是352*288/4个byte，最后是第一帧的V信息,长度是352*288/4个byte，因此可以算出第一帧数据总长度是352*288*1.5,即152064个byte，如果这个序列是300帧的话，那么序列总长度即为152064*300=44550KB,这也就是为什么常见的300帧CIF序列总是44M的原因.

4:4:4采样就是说三种元素Y,Cb,Cr有同样的分辨率,这样的话,在每一个像素点上都对这三种元素进行采样.数字4是指在水平方向上对于各种元素的采样率,比如说,每四个亮度采样点就有四个Cb的Cr采样值.4:4:4采样完整地保留了所有的信息值.4:2:2采样中(有时记为YUY2),色度元素在纵向与亮度值有同样的分辨率,而在横向则是亮度分辨率的一半(4:2:2表示每四个亮度值就有两个Cb和Cr采样.)4:2:2视频用来构造高品质的视频彩色信号.

在流行的4:2:0采样格式中(常记为**YV12**)Cb和Cr在水平和垂直方向上有Y分辨率的一半.4:2:0有些不同，因为它并不是指在实际采样中使用4:2:0，而是在编码史中定义这种编码方法是用来区别于4:4:4和4:2:2方法的).4:2:0采样被广泛地应用于消费应用中，比如视频会议，数字电视和DVD存储中。因为每个颜色差别元素中包含了四分之一的Y采样元素量，那么4:2:0YCbCr视频需要刚好4: 4:4或RGB视频中采样量的一半。

4:2:0采样有时被描述是一个"每像素12位"的方法。这么说的原因可以从对四个像素的采样中看出. 使用4:4:4采样，一共要进行12次采样，对每一个Y,Cb和Cr，就需要12*8=96位，平均下来要96/4=24位。使用4:2:0就需要6*8 =48位，平均每个像素48/4=12位。

在一个4:2:0隔行扫描的视频序列中，对应于一个完整的视频帧的Y,Cb,Cr采样分配到两个场中。可以得到，隔行扫描的总采样数跟渐进式扫描中使用的采样数目是相同的。

对比一下：

Y41P（和Y411）（packed格式）格式为每个像素保留Y分量，而UV分量在水平方向上每4个像素采样一次。一个宏像素为12个字节，实际表示8个像素。图像数据中YUV分量排列顺序如下： U0 Y0 V0 Y1 U4 Y2 V4 Y3 Y4 Y5 Y6 Y8 …

IYUV格式（planar）为每个像素都提取Y分量，而在UV分量的提取时，首先将图像分成若干个2 x 2的宏块，然后每个宏块提取一个U分量和一个V分量。YV12格式与IYUV类似，但仍然是平面模式。

YUV411、YUV420格式多见于DV数据中，前者用于NTSC制，后者用于PAL制。YUV411为每个像素都提取Y分量，而UV分量在水平方向上每4个像素采样一次。YUV420并非V分量采样为0，而是跟YUV411相比，在水平方向上提高一倍色差采样频率，在垂直方向上以U/V间隔的方式减小一半色差采样，如下图所示。
（好像显示不出来突下图像）

各种格式的具体使用位数的需求(使用4:2:0采样, 对于每个元素用8个位大小表示):

格式: Sub-QCIF 亮度分辨率: 128*96 每帧使用的位: 147456
格式: QCIF 亮度分辨率: 176*144 每帧使用的位: 304128
格式: CIF 亮度分辨率: 352*288 每帧使用的位: 1216512
格式: 4CIF 亮度分辨率: 704*576 每帧使用的位: 4866048

上一篇 source insight打开项目时报错
下一篇 Camera 图像处理原理分析

我的同类文章

camera（15）

- jpg图片的Exif及gps信息和... 2016-02-05 阅读 1481
- jpg 格式举例详解 2016-02-05 阅读 227
- 图像算法---白平衡AWB(讲... 2016-02-05 阅读 1208
- MTK Android software Tools... 2015-12-03 阅读 530
- Camera 图像处理原理分析 2015-11-25 阅读 974
- 变焦与对焦(转自csdn) 2014-07-16 阅读 1512
- CMOS Sensor的调试经验分享 2014-07-15 阅读 21640
- ISP与DSP的区别 2014-06-13 阅读 1488
- 摄影基础知识: 曝光补偿完全... 2014-01-07 阅读 897
- 关于光源色温与标准光源的... 2013-07-04 阅读 1592

更多文章

猜你在找

2016软考网络工程师内存存储容量计算强化训练教程
Android之数据存储
360度解析亚马逊AWS数据存储服务
顾荣：开源大数据存储系统Alluxio（原Tachyon）的原
iOS开发高级专题—数据存储
图文详解YUV420数据格式
YUV格式详解
图文详解YUV数据格式
视频与图像RGBYUV格式详解
转YUV格式详解

查看评论

暂无评论

您还没有登录,请[登录]或[注册]

* 以上用户言论只代表其个人观点, 不代表CSDN网站的观点或立场