

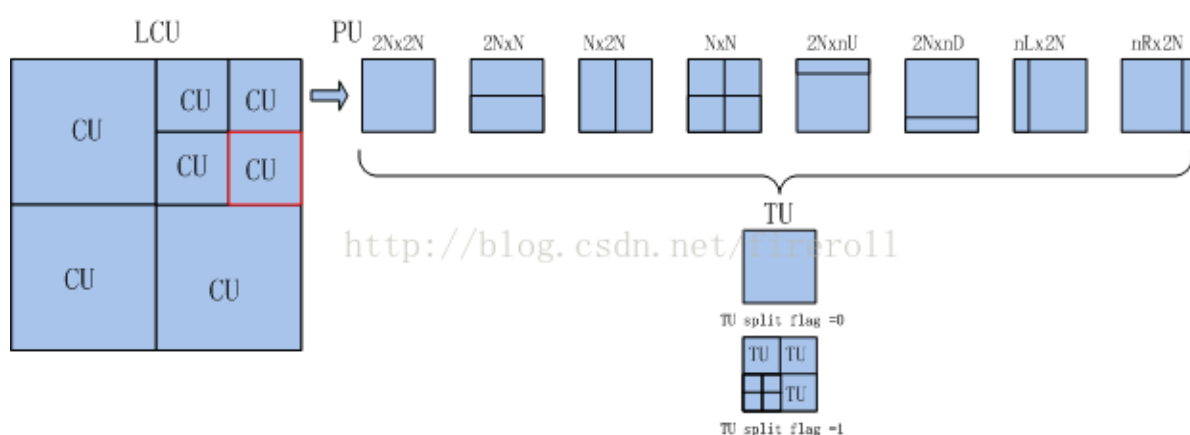
35中预测模式是在PU的基础上进行定义的，但是在具体的帧内预测过程中是以TU为单位的，标准规定PU可以四叉树的形式划分为TU，并且同一个PU内的TU共享一种预测模式

在实际的预测中，每一个TU自己预测自己的，自己参考自己周围的像素点

所以说：PU只是定义预测的方式，而真正的和预测像素和重构的过程都是通过TU进行处理的。

暂时不太了解CU的递归划分方式。等仔细阅读compresscu部分的代码再来补充。

有一个问题：H265为什么要分为PU和TU。在CU的基础上可以划分PU块和TU块。参考<https://blog.csdn.net/fireroll/article/details/77827156>介绍了H264和H265的区别部分介绍到了划分的原因。原谅小白由于水平有限还不能完全搞明白，先当定理记忆。



其中:

编码单元类似于H.264/AVC中的宏块的概念，用于编码的过程。

预测单元是进行预测的基本单元，

变换单元是进行变换和量化的基本单元。

这三个单元的分离，使得变换、预测和编码各个处理环节更加灵活，

也有利于各环节的划分更加符合视频图像的纹理特征，

有利于各个单元更优化的完成各自的功能。

目录：

- 1.编解码框架差异
- 2.压缩性能比较
- 3.各模块技术差异汇总
- 4.块划分结构
- 5.帧内预测
- 6.帧间预测

- 7.去块滤波
- 8.SAO滤波
- 9.Tile
- 10.WPP
- 11.Dependent slice
- 12.其他技术

1. H.264与H.265的主要差异

H.265仍然采用混合编解码，编解码结构域H.264基本一致，

主要的不同在于：

Ø 编码块划分结构：采用CU (CodingUnit)、PU(PredictionUnit)和TU(TransformUnit)的递归结构。

Ø 基本细节：各功能块的内部细节有很多差异

Ø 并行工具：增加了Tile以及WPP等并行工具集以提高编码速度

Ø 滤波器：在去块滤波之后增加了SAO (sample adaptive offset) 滤波模块

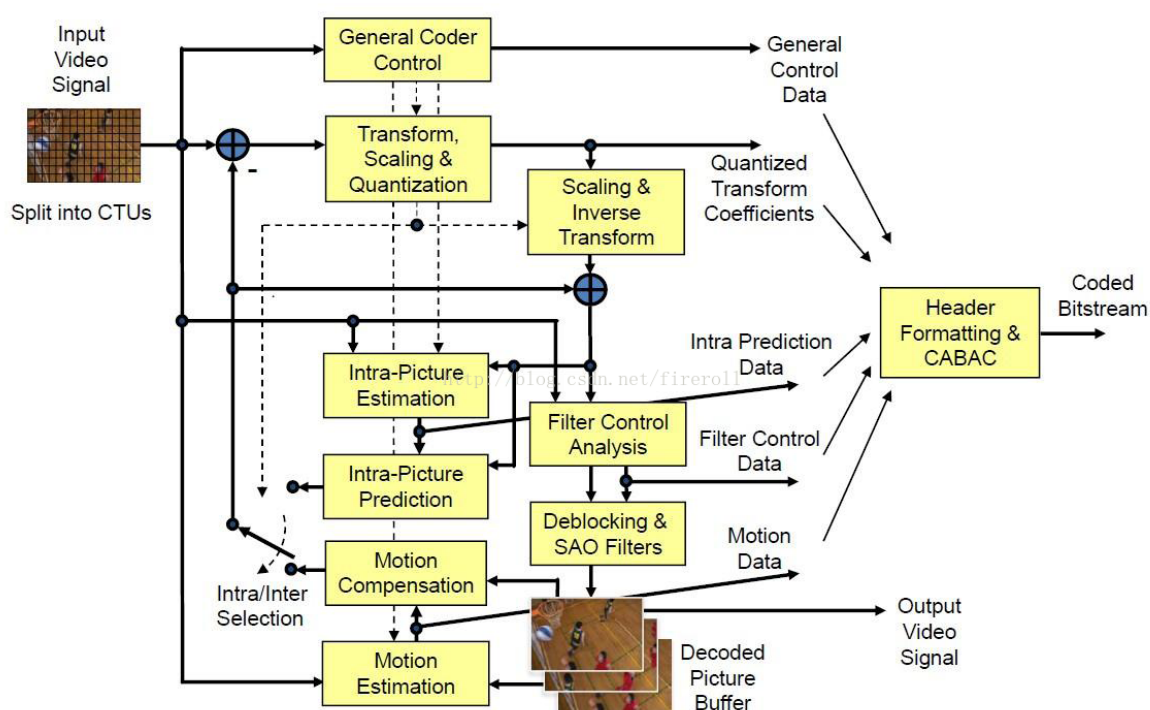


Fig.H.265的框架图

2. 压缩性能比较

PSNR计算方式

$$PSNR_{avg} = (6 \times PSNR_Y + PSNR_U + PSNR_V) / 8$$

H.265/HEVC HM-9.0 和H.264 JM-18.4 的BD-rate 比较:

AllIntra case: 22%

RandomAccess case: 34%

LowDelay case: 37%

3. 各模块技术差异汇总

	H.264	H.265
MB/CU大小	4×4~16×16	4×4~64×64
Inter插值	Luma-为6抽头系数插值 Chroma双线性插值	Luma-1/2像素采用8抽头插值滤波器 Luma-1/4像素采用7抽头插值滤波器 Chroma所有分数像素点采用4抽头系数插值
Inter MVP预测方法	空域中值MVP预测	空域+时域MVP预测候选列表（AMVP） 空域+时域的Merge/Skip的候选列表
Intra预测	亮度4x4块：9种模式 亮度8x8块：9种预测模式 亮度16x16块：4种预测模式 色度：4种预测模式	亮度所有尺寸的CU块：总共35种预测模式 色度所有尺寸的CU块：5种预测模式
变换	DCT 4×4/8×8	DCT 4×4/8×8/16×16/32×32 DST 4×4
滤波器	4×4和8×8边界去块滤波 http://blog.csdn.net/fireroll	●8x8及以上的CU、PU、TU边界去块滤波 ●SAO滤波器(Sample Adaptive Offset)
熵编解码技术	CAVLC及CABAC	CABAC
其他技术	FMO映射关系等	Tile、WPP以及dependent Slice

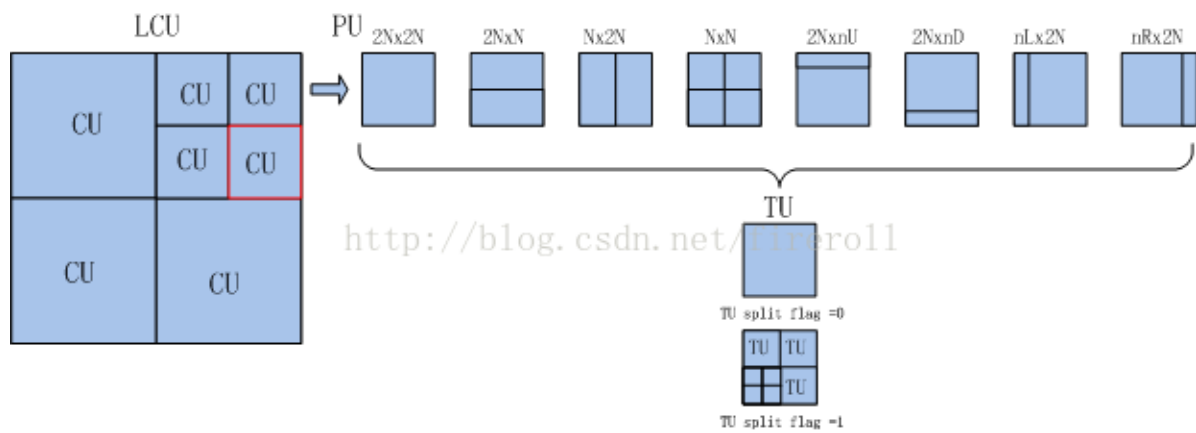
4. 块划分结构

在H.265中，将宏块的大小从H.264的16×16扩展到了64×64，以便于高分辨率视频的压缩。

同时，采用了更加灵活的编码结构来提高编码效率，

包括编码单元（CodingUnit）、预测单元（PredictUnit）和变换单元（TransformUnit）。

如下图所示：



其中:

编码单元类似于H.264/AVC中的宏块的概念, 用于编码的过程。

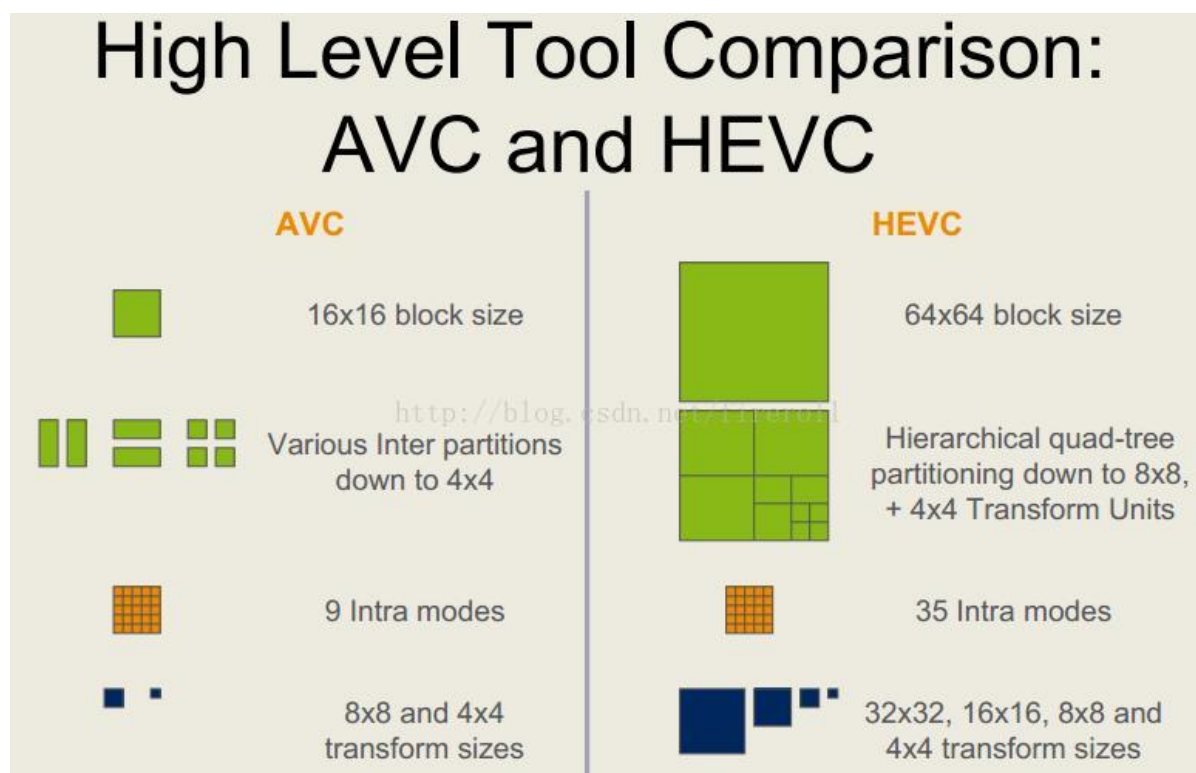
预测单元是进行预测的基本单元,

变换单元是进行变换和量化的基本单元。

这三个单元的分离, 使得变换、预测和编码各个处理环节更加灵活,

也有利于各环节的划分更加符合视频图像的纹理特征,

有利于各个单元更优化的完成各自的功能。



RQT是一种自适应的变换技术, 这种思想是对H.264/AVC中ABT (AdaptiveBlock-size Transform) 技术的延伸和扩展。

对于帧间编码来说, 它允许变换块的大小根据运动补偿块的大小进行自适应的调整;

对于帧内编码来说，它允许变换块的大小根据帧内预测残差的特性进行自适应的调整。

大块的变换相对于小块的变换，一方面能够提供更好的能量集中效果，并能在量化后保存更多的图像细节，但是另一方面在量化后却会带来更多的振铃效应。

因此，根据当前块信号的特性，自适应的选择变换块大小，如下图所示，可以得到能量集中、细节保留程度以及图像的振铃效应三者最优的折中。

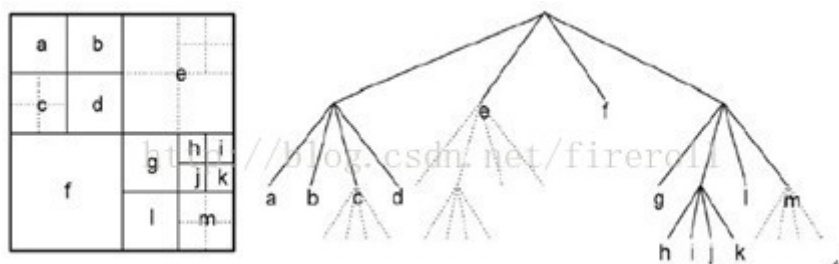


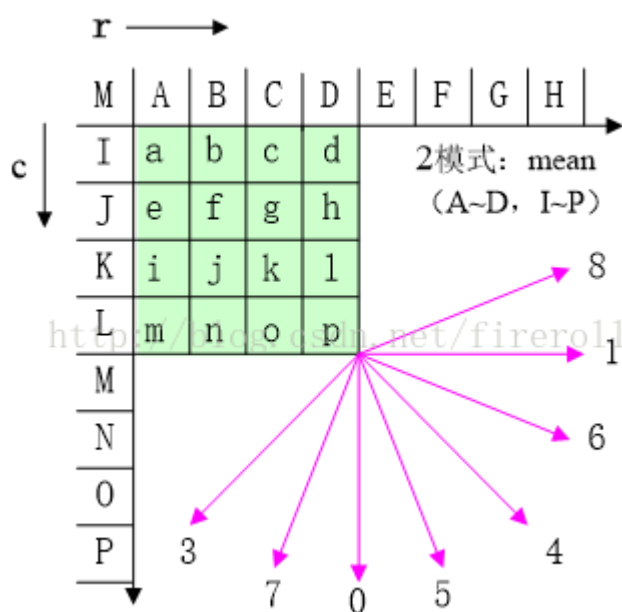
Fig. 灵活的块结构示意图

5. 帧内预测模式

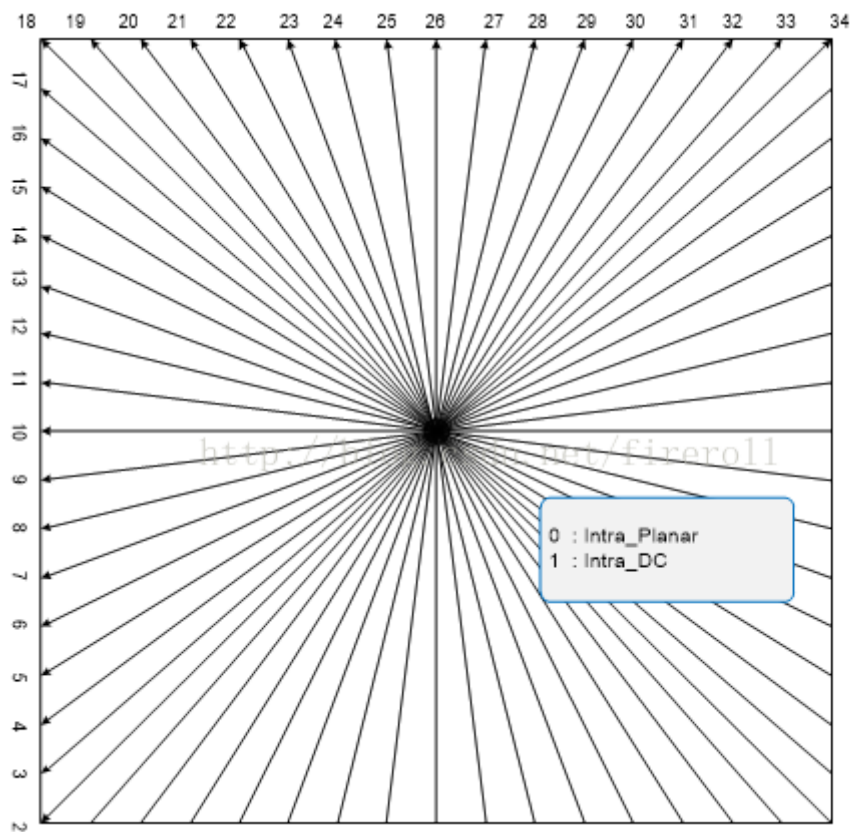
本质上H.265是在H.264的预测方向基础上增加了更多的预测方向

H.265: 所有尺寸的CU块，亮度有35种预测方向，色度有5种预测方向

H.264: 亮度 4x4块9个方向，8x8块9个方向，16x16块4种方向，色度4种方向



H.264的帧内预测方向:



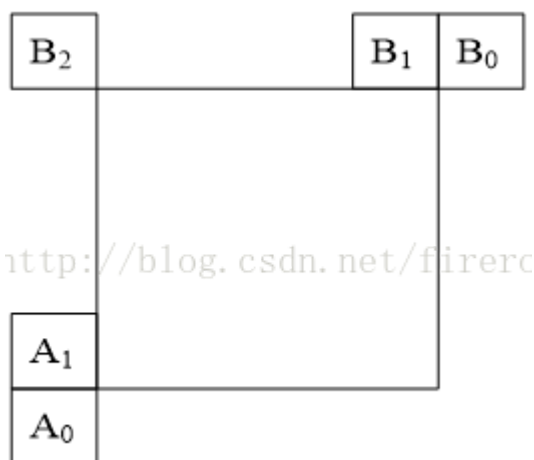
H.265的帧内预测方向：

6. 帧间预测

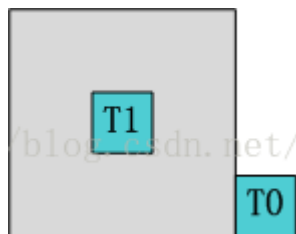
本质上H.265是在H.264基础上增加插值的抽头系数个数，改变抽头系数值以及增加运动矢量预测值的候选个数，以达到减少预测残差的目的。

H.265与H.264一样插值精度都是亮度到1/4，色度到1/8精度，但插值滤波器抽头长度和系数不同。

H.265的增加了运动矢量预测值候选的个数，而H.264预测值只有一个



H.265的空域候选项：



H.265时域共同位置候选项

7. 去块滤波

本质上H.265的去块滤波与H.264的去块滤波及流程是一致的，做了如下最显著的改变：

Ø 滤波边界： H.264最小到4x4边界滤波；而H.265适应最新的CU、PU和TU划分结构的滤波边缘，最小滤波边界为8x8，

Ø 滤波顺序： H264先宏块内采用垂直边界，再当前宏块内水平边界；而H.265先整帧的垂直边界，再整帧的水平边界

ALF在编解码环路内，位于Deblock和SAO之后，

用于恢复重建图像以达到重建图像与原始图像之间的均方差（MSE）最小。

ALF的系数是在帧级计算和传输的，可以整帧应用ALF，

也可以对于基于块或基于量化树（quadtree）的部分区域进行ALF，

如果是基于部分区域的ALF，还必须传递指示区域信息的附加信息。

8. 采样点自适应偏移（Sample AdaptiveOffset）滤波

SAO(sample adaptive offset)滤波其实就是对去块滤波后的重建像素按照不同的模板进行分类，并对每一种分类像素进行补偿，分类模板分为BO(Band offset)和EO(Edge offset)。

BO总共有32中类型，只取从sao_band_position开始的连续四个类别作为当前SAO类别



BO分类：



EO分类模块：

SAO在编解码环路内，位于Deblock之后，通过对重建图像的分类，对每一类图像像素值加减一个偏移，达到减少失真的目的，从而提高压缩率，减少码流。

采用SAO后，平均可以减少2%~6%的码流,而编码器和解码器的性能消耗仅仅增加了约2%。

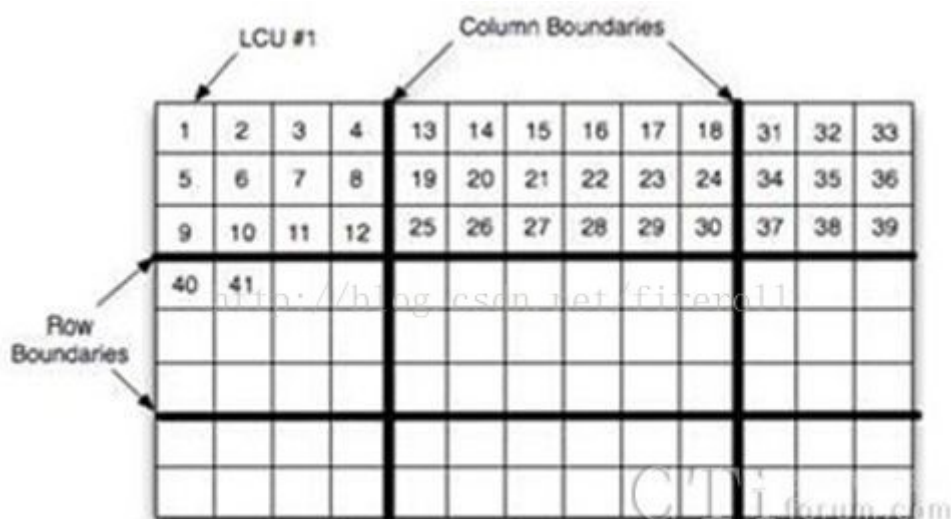
9. Tile

Tile：将图像分割为矩形区域。

其主要目的是增强并行处理性能。

每个tile区域相当于一幅子图像，可独立的以LCU块为单位进行编解码。

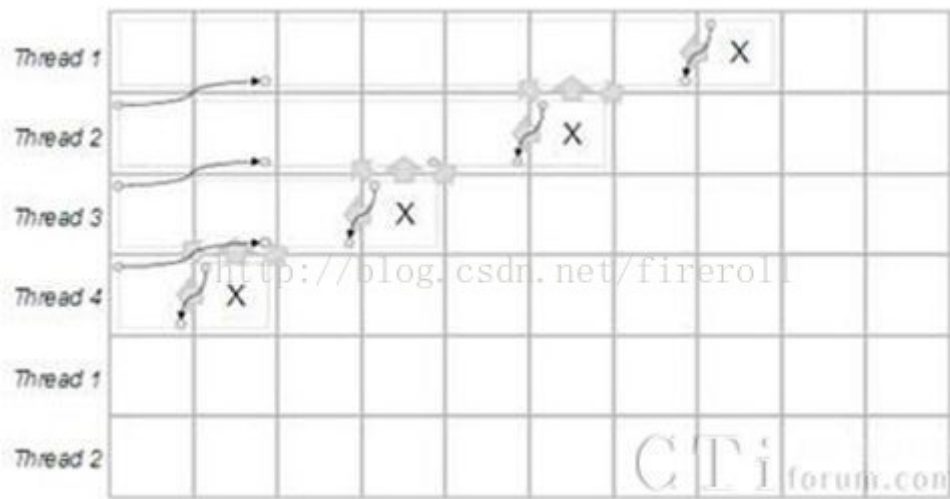
一个Tile块为基本的并行单元，每个Tile为一个子码流



\10. WPP

WPP: 全称为wavefront parallel process，以LCU行为基本的编码单位。

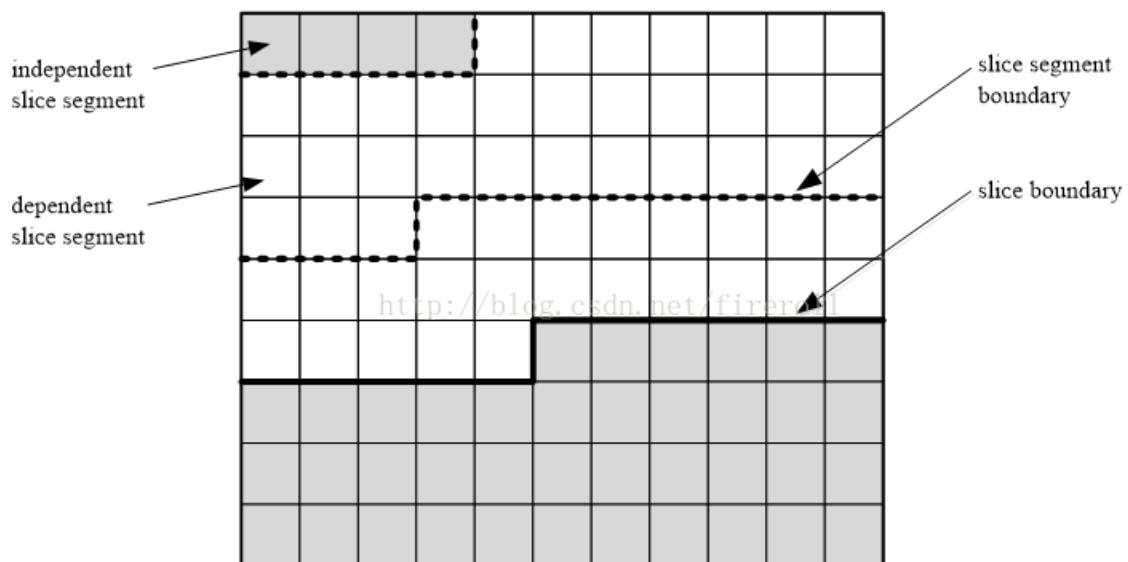
以一行LCU块为基本的并行单元，每一行LCU为一个子码流



\11. Dependentslice

Dependent slice: 该技术可以理解为对原先Slice NALU的数据划分，使其可以适合更加灵活的打包方式。

Slice 和dependent slice 的示意图如下



\12. 其他相关技术

Ø Transform_skip模式: transform_skip_flag, 该模式不进行变换，但是要进行量化，该模式对文本桌面视频有较好效果

Ø 内部比特深度增加: 为了保证中间预测、变换以及量化过程中的内部比特精度，以达到更好的压缩性能