多媒体编码及其信息安全应用

Multimedia Coding and Its Application to Information Security

第五讲

视频编码基础及H.264概述

授课时间：2022年3月21日

内 容 提 纲 （3 节 课 内 容）

1. 视频编码基础

2. H.264/AVC总体概述

3. H.264/AVC编码标准特性

4. 小结

内 容 提 纲 （3 节 课 内 容）

1. 视频编码基础

2. H.264/AVC总体概述

3. H.264/AVC编码标准特性

4. 小结

1.1 预备知识

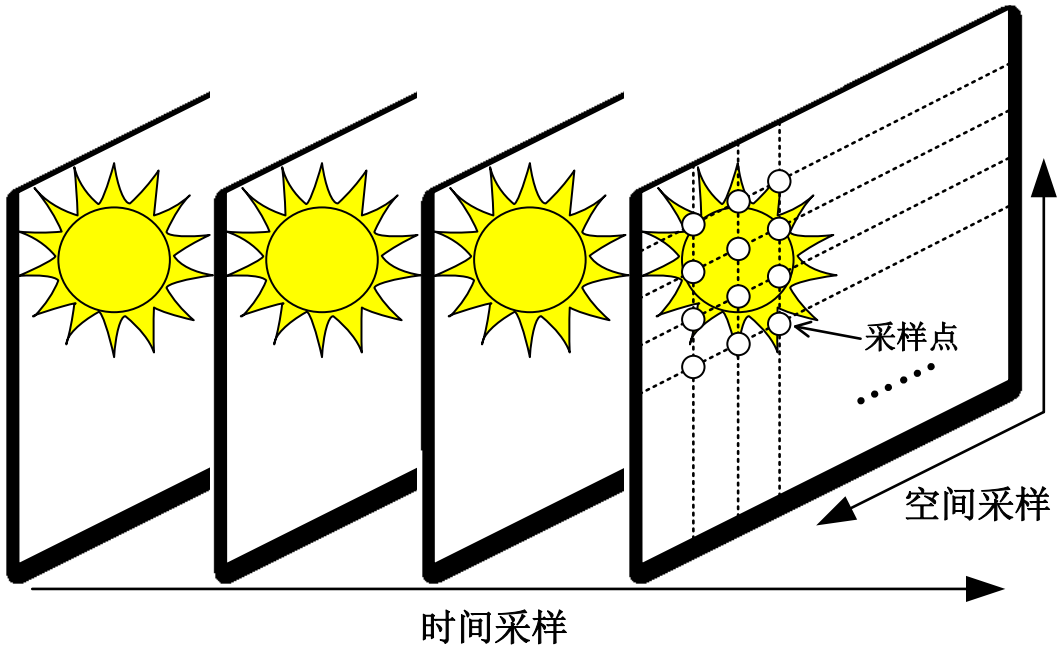
- 数字视频——当今影响力最大的传播媒介
  - **Youtube** 每月有超过 20 亿登录用户访问 YouTube，观看视频总时长超过 10 亿小时，每天的观看次数高达数十亿次。
  - **Vimeo** 每月观看人数1亿7千万人，视频观看次数超过7.15亿次。
  - **抖音** 日活跃用户数超4亿。
  - **Facebook** 视频的参与度/互动量保持在每周两亿五千万次左右。

1.1 预备知识

数字视频采集

- 真实世界的自然场景在空间和时间上都是连续的，将它们表示成数字形式需要进行空间采样（Spatial Sampling）和时间采样（Temporal Sampling）。

- 数字视频是指经过（空间和时间）采样的实际场景以数字形式的表示。采样点也称为像素（Pixel）。每个像素采用一个或一组数字表示，用于描述相应采样点的亮度（Luminance）和色彩。



1.1 预备知识

色彩空间

- RGB色彩空间基于三基色（白光的三种主要合成颜色）原理，通过红（R）、绿（G）、蓝（B）这三种基色表达颜色信息。具体地，在RGB色彩空间中，一个像素由三个数字表示，分别代表红、绿、蓝这三种基色的相对比例。由于这三种基色同等重要，故通常采用相同精度表示它们的色度值。

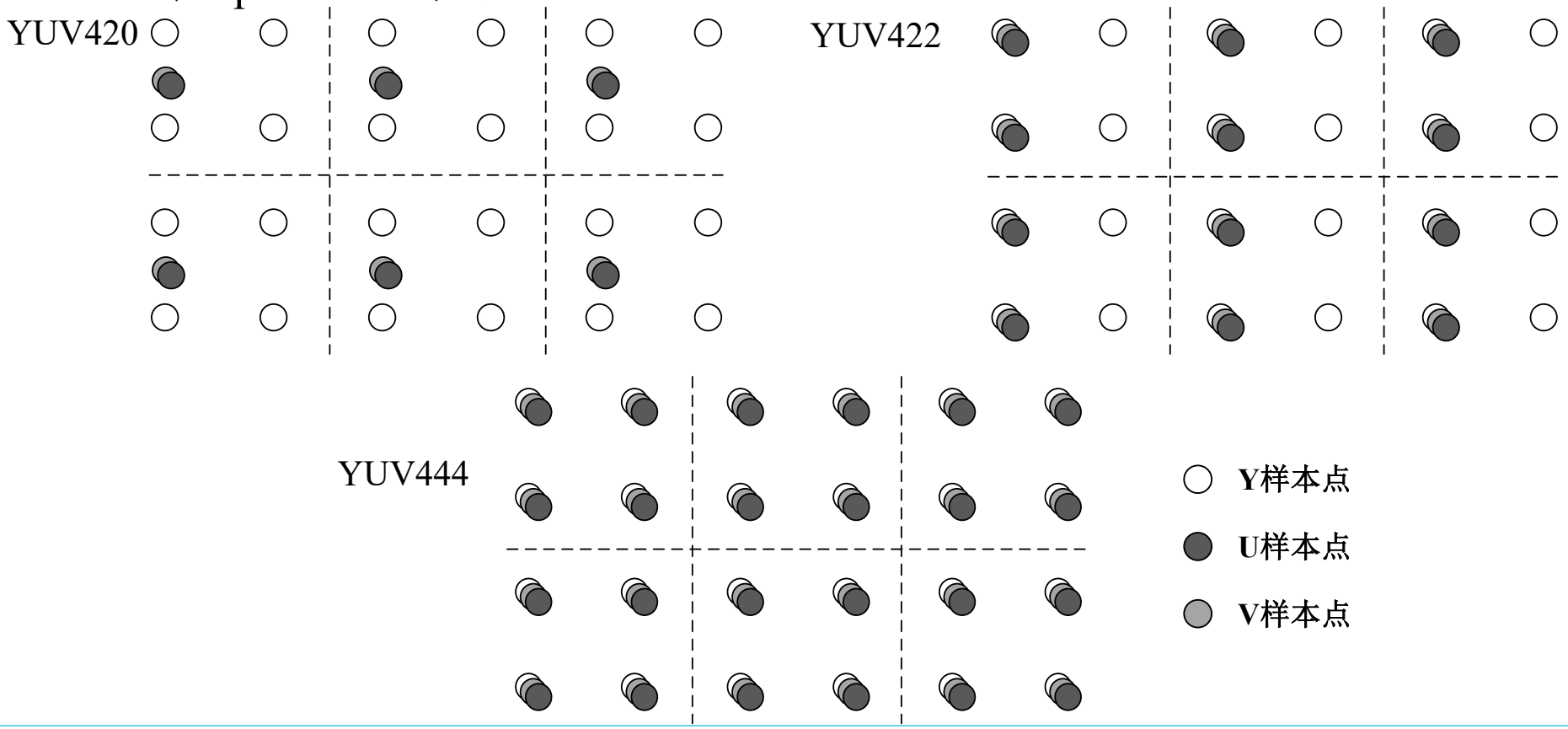
- 人类视觉系统HVS（Human Visual System）对亮度的敏感程度高于对色度（Chrominance）的敏感程度。基于HVS 的这一特性，研究者提出了YUV色彩空间，通过亮度（Y）和色度（U、V）这两个基本成分表达颜色信息，并且对亮度成分采用比色度成分更高精度的采样。相比RGB色彩空间，采用YUV色彩空间描述视频的颜色信息，能够有效降低所需存储或处理的视频数据量，并且不会对视频的视觉质量产生明显影响。




1.1 预备知识

色彩空间

- 根据对亮度分量（Y）和两个色度分量（U、V）的采样比例的不同，YUV采样格式可分为YUV420、YUV422 和YUV444 三种。其中，YUV420格式最为常用，广泛用于视频会议、数字电视、DVD（Digital Video Disc）等消费应用领域；YUV422格式通常用于高质量色彩再现（Reproduction）。



1.1 预备知识

视频质量评价

- 对视频数据只进行无损（Lossless）压缩仅能达到有限的压缩效果，故绝大多数视频编码技术是基于有损（Lossy）压缩的，这使得原始视频数据和重建（Reconstructed）视频数据存在差异，当压缩率达到一定程度时，难免对视频的视觉质量（简称视频质量）造成负面影响。因此，需要建立一套针对视频质量的评价准则。

- 视频质量本身是一个主观概念，会受到许多主观因素（如观看心情、观看重点）的影响，使得难以在主观层面对其做出完全精确的量化。因此，在实际应用中通常采用客观质量评价准则衡量视频质量。需要说明的是，虽然客观质量评价准则能够产生精确、可重复的质量评价结果，但其无法完全反映或再现观测者观看视频时的主观视觉体验。目前常见的客观质量评价准则主要是峰值信噪比（Peak Signal to Noise Ratio，PSNR）和结构相似度（Structural SIMilarity index，SSIM）。峰值信噪比PSNR是最为常用的客观质量评价准则。

1.1 预备知识

视频质量评价

- 峰值信噪比（Peak Signal-to-Noise Ratio，PSNR)

$$\mathrm{PSNR} = 10 \times log_{10}\left[\frac{(2^n-1)^2}{MSE}\right]$$

$$\mathrm{MSE} = \frac{1}{mn}\sum_{i=0}^{m}\sum_{j=0}^{n}\|I(i,j)-K(i,j)\|^2$$

原图

Y：40.632022
U：44.596545
V：45.759277

视频质量评价

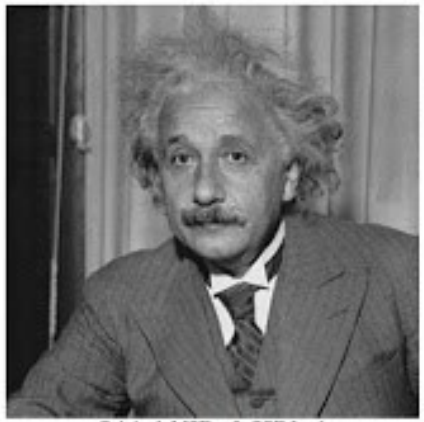
- 结构相似性指标（Structural Similarity index，SSIM）

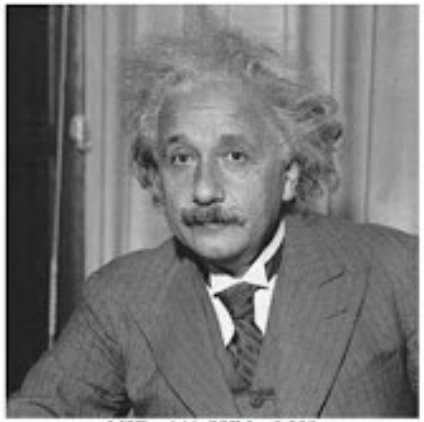
$$\mathrm{SSIM}(\mathbf{X},\mathbf{Y}) = [L(\mathbf{X},\mathbf{Y})]^{\alpha}[C(\mathbf{X},\mathbf{Y})]^{\beta}[S(\mathbf{X},\mathbf{Y})]^{\gamma}$$

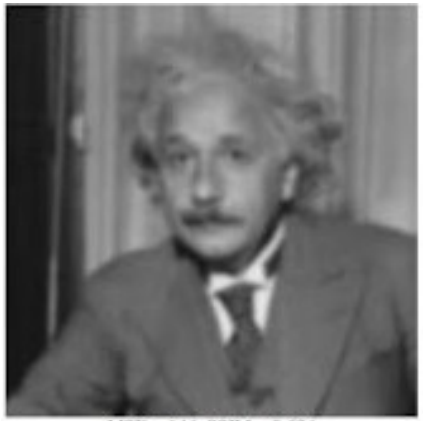
$$L(\mathbf{X},\mathbf{Y}) = \frac{2\mu_{\mathbf{X}}\mu_{\mathbf{Y}} + c_1}{\mu_{\mathbf{X}}^2 + \mu_{\mathbf{Y}}^2 + c_1}$$

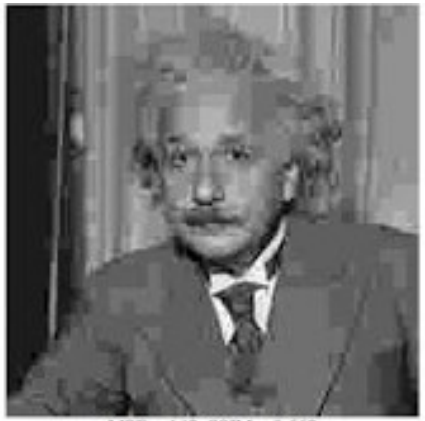
$$C(\mathbf{X},\mathbf{Y}) = \frac{2\sigma_{\mathbf{X}}\sigma_{\mathbf{Y}} + c_2}{\sigma_{\mathbf{X}}^2 + \sigma_{\mathbf{Y}}^2 + c_2}$$

$$S(\mathbf{X},\mathbf{Y}) = \frac{\sigma_{\mathbf{XY}} + c_3}{\sigma_{\mathbf{X}}\sigma_{\mathbf{Y}} + c_3}$$


Original, MSE = 0; SSIM = 1


MSE = 144, SSIM = 0.988


MSE = 144, SSIM = 0.694


MSE = 142, SSIM = 0.662

1.1 预备知识

视频数据冗余类型

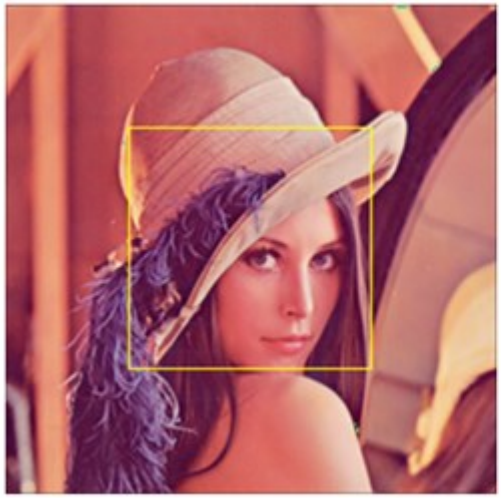
- 空间冗余：视频帧中空间位置相近的采样点的采样值通常存在一定相关性，例如，背景区域的相邻像素在亮度和色度上十分接近。这种空间相关性称为空间冗余。

- 时间冗余：视频序列中相邻视频帧的时间间隔极短，它们在内容上通常存在较强相关性，主要表现在，相邻帧画面的背景和主体基本相同，只是位置和形态可能略微发生变化。这种时间相关性称为时间冗余。

1.1 预备知识

视频数据冗余类型

- 视觉冗余：人类视觉系统对图像场的敏感性是非均匀和非线性的，例如，对亮度的感知比对色度的感知更加敏感；对平坦区域变化的敏感程度高于对边缘和纹理复杂区域变化的敏感程度。然而，在采集视频时，通常假设视觉系统是均匀和线性的，未利用人类视觉系统的特性对敏感区域和非敏感区域的视频数据进行区别对待和处理，从而导致额外的数据开销。



1.1 预备知识

视频数据冗余类型

- 结构冗余：视频帧中可能重复出现在纹理结构或分布模式上具有较强相似性的区域，例如蜂窝、草席、方格状的地板图案。这种结构分布的相关性称为结构冗余。

- 信息熵冗余：根据信息论，表示视频像素时，可以按照其信息熵分配相应的比特数。然而，在视频采集过程中，难以获取每个像素的信息熵，因此一般采用相同比特数表示每个像素。这种非最优编码状态称为信息熵冗余或编码冗余。

- 知识冗余：视频帧中可能存在某些区域，和人类的先验知识紧密相关，例如，人脸五官位置的固有分布。这些规律性的结构或模式可以通过先验知识进行推导和重建，此类冗余称为知识冗余。

1.2 数字视频编码基础

视频编解码器（Codec）简介

- 视频编解码器是通过软件或硬件应用程序完成的视频压缩标准。每个编解码器包括用于压缩视频的编码器和用于重新创建视频的近似值以用于回放的解码器。编解码器实际上来自于将Encoder和Decoder两个概念合并为一个单词。

- 常见视频编解码器包括H.264，VP8，RV40以及这些编解码器的许多其他标准或更高版本，例如VP9。虽然这些标准与视频流有关，但视频通常与音频流捆绑在一起，而音频流可以有自己的压缩标准。音频压缩标准的示例（通常称为音频编解码器）包括LAME/MP3，AAC，FLAC等。

- 不应将这些编解码器与用于封装所有内容的容器混淆。常见容器有MKV（Matroska视频），MOV（Movie的缩写），AVI（音频视频交错）等。这些容器没有定义如何编码和解码视频数据。相反，它们以兼容应用程序可以回放内容的方式存储来自编解码器的字节。此外，这些容器不仅存储视频和音频信息，还存储元数据。

1.2 数字视频编码基础

视频编解码常用基本名词

- **视频序列（Video Sequence）。**编码视频比特流的最高语法结构。它包含一系列一个或多个编码帧。
- **图像组（Group of Pictures，GOP）。**一个或多个编码图像组成的序列
- **帧（Frame）。**一帧包含一个亮度样点矩阵和对应的两个色度样点矩阵。
- **场（Field）。**一帧的交替行组成的两个集合分别为顶场和底场。
- **宏块（Macroblock）。**在H.264标准中，通常指一个16 × 16的亮度样点矩阵和对应的两个色度样点矩阵。
- **码率（Bitrate）。**单位时间内传送的数据位数，通常单位为Kbps。
- **帧率（Frame rate）。**以帧为单位的位图图像连续出现在显示器上的频率。

1.2 数字视频编码基础

视频编解码常用基本名词

- **I帧（I-frame）** 仅使用自身信息编码（即帧内预测编码）的帧。“I”意为“Intra-coded”。I帧图像旨在帮助随机访问视频序列。需要随机访问、快进快退播放的视频可能会相对频繁地使用I帧图像。I帧还可以用于视频中场景切换等其他情况。可在I帧前使用图像组头部（Group of Pictures Header），向解码器指示随机访问时是否可以正确重建该图像组的图像。I帧的编码（压缩）效率相比P帧、B帧较低。

- **P帧（P-frame）** 使用过去（Past）参考场或帧进行运动补偿预测编码的帧。“P”意为“Predictive-coded”。包含前向预测编码图像的帧称为P帧，P帧的编码效率比I帧高、比B帧低。

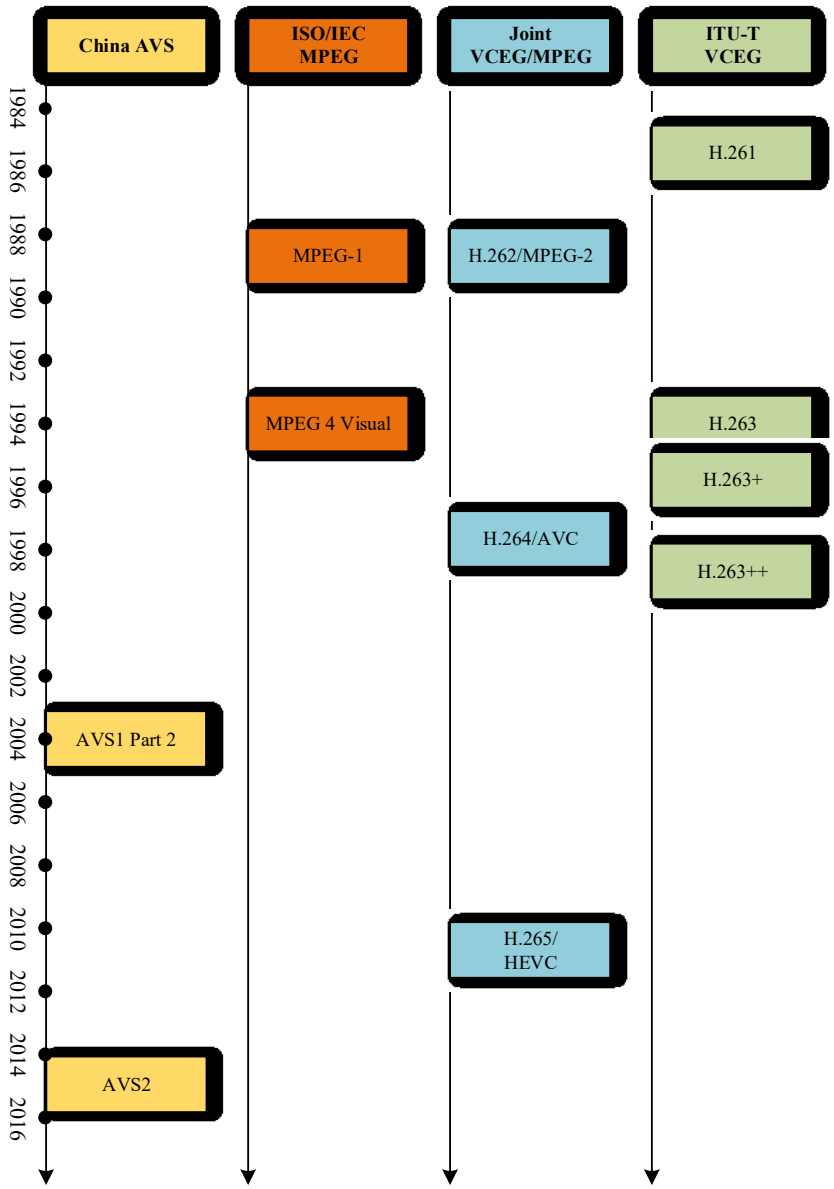
- **B帧（B-frame）** 使用过去和/或未来（Future）参考场或帧进行运动补偿预测编码的帧。“B”意为“Bidirectionally Predictive-coded”。包含双向预测编码图像的帧称为B帧，B帧的编码效率相比I帧、P帧较高。

1.3 视频编码标准的发展

视频编码标准分类

视频编码技术的更新换代，有力推动了压缩编码性能的提升。按照制定时间和编码效率（Coding Efficiency），主流视频编码标准可以分为三代：

- H.261、MPEG1、MPEG2/H.262、MPEG-4 Visual、H.263 等为第一代编码标准；
- H.264/AVC、VC1（Microsoft 制定）、VP8（Google 制定）、AVS1 等为第二代编码标准；
- H.265/HEVC、VP9（Google 制定）、AVS2等为第三代编码标准。



1.3 视频编码标准的发展

MPEG-1视频编码标准

- MPEG-1标准。MPEG-1是MPEG组织指定的第一个视频和音频有损压缩标准。视频压缩算法于1990年定义完成。1992年底，MPEG-1正式被批准成为国际标准，它用于传输1.5Mbps数据传输率的数字存储媒体运动图像及其伴音的编码。MPEG-1采用了块方式的运动补偿、离散余弦变换（DCT）、量化等技术。

- 主要用于在CD-ROM存储运动视频图像，它针对标准分辨率(NTSC制为352×240；PAL制为352×288)的图像进行压缩，每秒30帧画面，具备CD音质。使用MPEG-1的压缩算法，可将一部120分钟长的电影压缩到1.2GB左右。因此，它被广泛地应用于VCD制作。

- MPEG-1与后续视频编码标准不同的是，其帧类型除I帧、P帧以及B帧外，还有一类称为D帧。D帧仅使用DC变换系数进行了编码（编码D帧时删除了AC系数），因此质量很低。D帧仅用于视频的快速预览，例如在高速搜索视频时。

1.3 视频编码标准的发展

MPEG-2视频编码标准

- MPEG-2标准。 MPEG-2是MPEG工作组于1994年发布的视频和音频压缩国际标准。 MPEG-2通常用来为广播信号提供视频和音频编码，包括卫星电视、有线电视等。 MPEG-2经过少量修改后，也成为DVD产品的核心技术。 MPEG-2的另一特点是，其可提供一个较广的范围改变压缩比，以适应不同画面质量、存储容量以及带宽要求。

- MPEG-2的第二部分即视频部分和MPEG-1类似，但是它提供对隔行扫描视频显示模式的支持（隔行扫描广泛应用在广播电视领域）。MPEG-2视频并没有对低比特率（小于1Mbps）进行优化，但是在3Mbps及以上比特率情况下，MPEG-2明显优于MPEG-1。MPEG-2向后兼容，即所有符合标准的MPEG-2解码器也能够正常播放MPEG-1视频流。

- MPEG-2的变换过程为8×8的DCT变换，量化时提供了31个量化参数，熵编码过程则利用哈夫曼编码完成。

1.3 视频编码标准的发展

MPEG-4视频编码标准

- MPEG-4标准。 MPEG-4是为在国际互联网或移动通信设备上实时传输音/视频信号而制定的MPEG标准。 MPEG-4采用图层（layer）方式，可根据图像内容，将其中的对象分离出来分别进行压缩，使图像文件容量大幅缩减，压缩倍数为450倍（静态图像可达800倍）。分辨率输入可从CIF到4K。

- MPEG-4标准是面向对象的压缩方式，能够根据图像的内容， 其中的对象（物体、人物、背景）分离出来，分别进行帧内、帧间编码，并允许在不同的对象之间灵活分配码率，对重要的对象分配较多的字节，对次要的对象分配 较少的字节，从而大大提高了压缩比，在较低的码率下获得较好的效果。

- MPEG-4标准目前分为27个部分。第二部分视频定义了对各类视觉信息（包括自然视频、静止纹理、计算机合成图形等等）的编解码器；第三部分音频定义了对各种音频信号进行编码的编解码器的集合；第十部分高级视频编码（Advanced Video Coding，AVC）定义了更高级的视频编解码器。此外，它和ITU-T H.264标准是一致的，故又称为H.264。

1.3 视频编码标准的发展

H.26x编码标准

- H.261标准。H.261是ITU-T在1990年指定的一个视频编码标准。H.261标准是第一个实用的数字视频编码标准，设计的目的是能够在综合业务数字网上传输质量可接受的视频信号。能够对CIF和QCIF分辨率的视频进行编码。H.261包括了带运动补偿的帧间预测、DCT变换、量化、熵编码以及固定速率的信道匹配的码率控制部分。

- H.263标准。 H.263标准是实在1996年推出的，它是ITU-T为低于64kbps的窄带通信信道指定的替代H.261的视频编解码标准。与H.261相比， H.263的运动补偿精度为1/2像素，不包含环路滤波器，支持超越图像边界的运动向量，基于上下文的算术编码，前向和后向帧间预测。

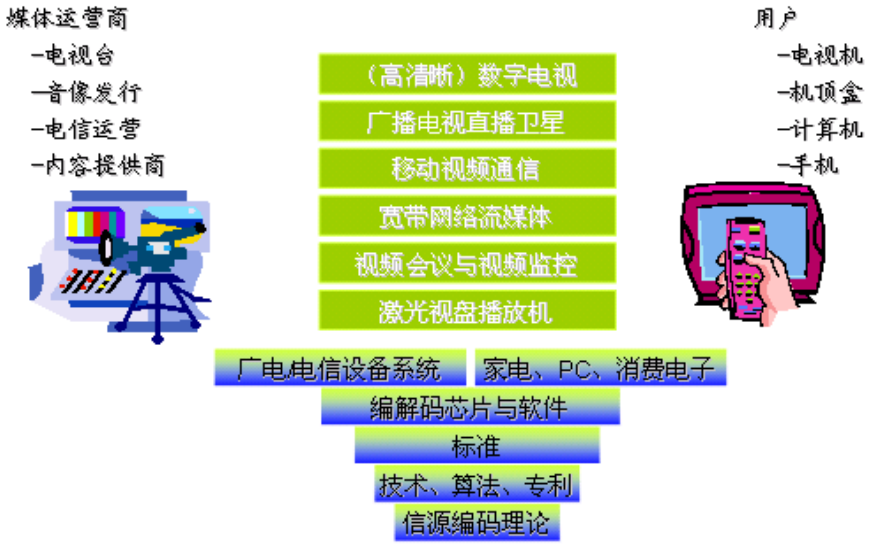
- H.264标准。 H.264是ITU-T的VCEG和ISO/IEC的MPEG的联合视频组开发的数字视频编码标准。其加强了对各种信道的适应能力，在技术上，使用了统一的VLC符号编码，高精度、多模块的唯一估计，基于4×4的整数变换，分层的编码语法等。在相同的重建图像质量下，能够比H.263节约50%的码率。H.264的码流结构网络适应性强，增强了差错恢复能力，能够很好地适应IP和无线网络的应用。

1.3 视频编码标准的发展

AVS China编码标准

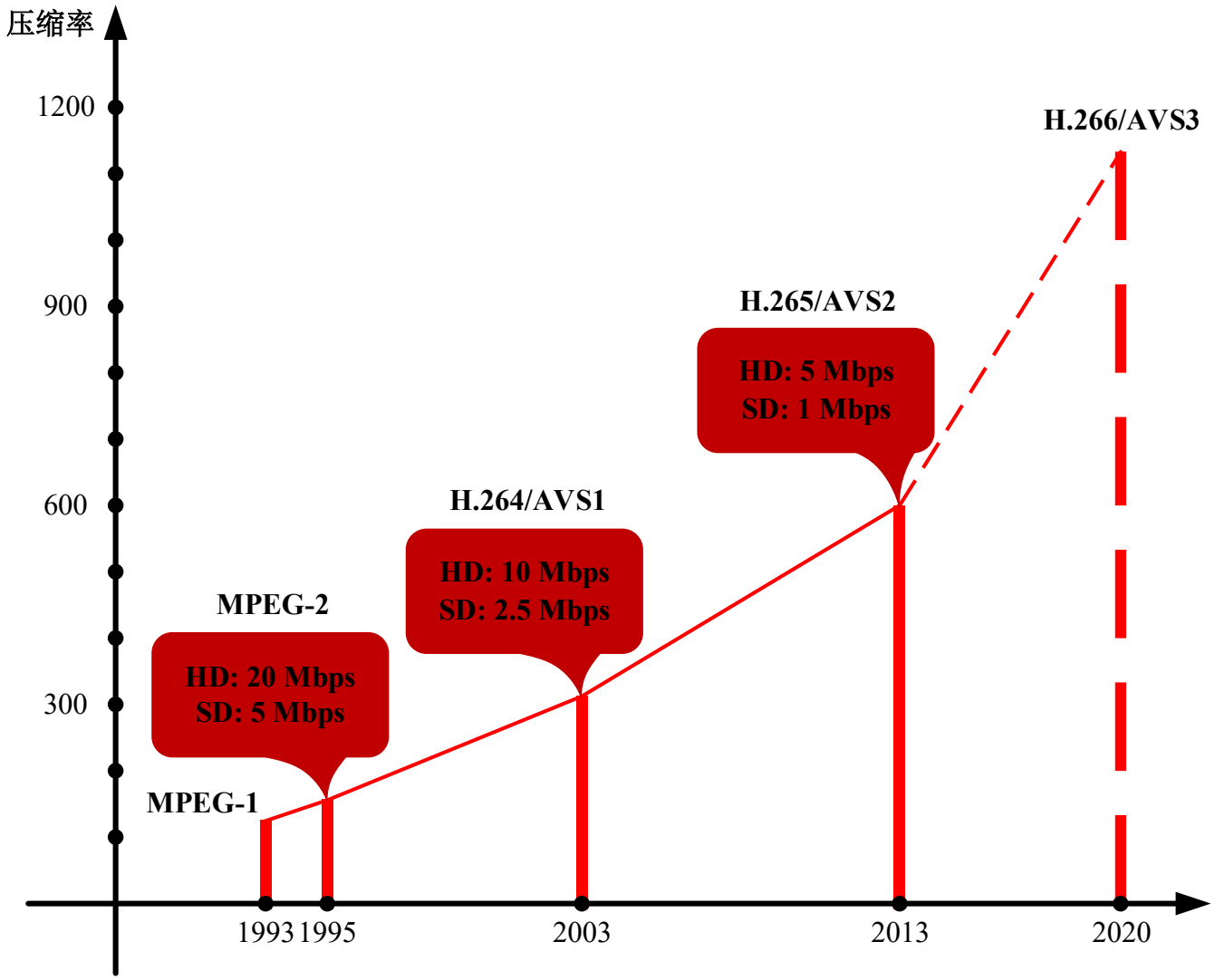
- AVS China是我国具备自主知识产权的第二代信源编码标准。采用了可变块大小的运动补偿预测技术、多方向空间预测技术、多参考帧、改进的运动向量预测编码等先进技术。其压缩性能与H.264相当，比H.261、H.263、MPEG-1、MPEG-2等压缩算法具有明显优势。

- AVS标准主要针对标清（Standard Definition，SD）和高清（High Definition，HD）视频压缩。主要应用于数字视频电视、数字视频光盘（DVD和高清光盘）以及宽带网络多媒体应用如视频会议、视频点播以及IPTV等。



1.3 视频编码标准的发展

- 主流视频编码技术的压缩性能对比



内 容 提 纲 （3 节 课 内 容）

1. 视频编码基础

2. H.264/AVC总体概述

3. H.264/AVC编码标准特性

4. 小结

2.1 H.264/AVC视频编码特性

H.264/AVC视频编码标准优越性

- **低码率。**在同等的还原图像质量条件下，采用H.264/AVC压缩后的视频数据码率只有MPEG-4 Visual (Part-2)的1/2。

- **应用目标范围较宽。**H.264/AVC视频编解码标准可满足从可视电话应用到广播应用下的不同速率、不同解析度、不同传输信道与存储场合的需求。

- **容错能力强。**H.264/AVC视频编解码标准加强了对各种信道的适应能力，能够较好地解决误码和丢包错误， H.264/AVC容错能力比MPEG-4提高1倍。

- **极佳的图像传输质量。**H.264/AVC视频编解码标准在1Mbps码率下就可以达到DVD画质。

- **网络适应性强。**H.264/AVC视频编解码标准提供了网络抽象层，使得H.264/AVC数据文件能够容易地在不同网络上高效传输。

2.2 H.264/AVC视频编码标准常见术语

H.264/AVC视频编解码标准基本名词

- **帧内预测（Intra Prediction）** 使用当前图像中已经解码的样值对当前待编码样值进行预测的过程。
- **帧间预测（Inter Prediction）** 不根据当前图像，而根据已解码的参考图像进行预测。
- **帧内条带/I条带（Intra Slice/I Slice）** 在解码时仅使用同一个条带内部样点进行预测的条带。
- **预测条带/P条带（Predictive Slice/P Slice）** 可根据同一条带内部样点进行帧内预测，或根据已解码的参考图像进行帧间预测的条带。
- **双向预测条带/B条带（Bi-predictive Slice/B Slice）** 可根据同一条带内部样点进行帧内预测，或根据已解码的参考图像进行帧间预测的条带。
- **残差（Residual）** 样点或其他数据元素预测值与解码值之间的差值。

2.3 H.264/AVC视频编码标准常见术语

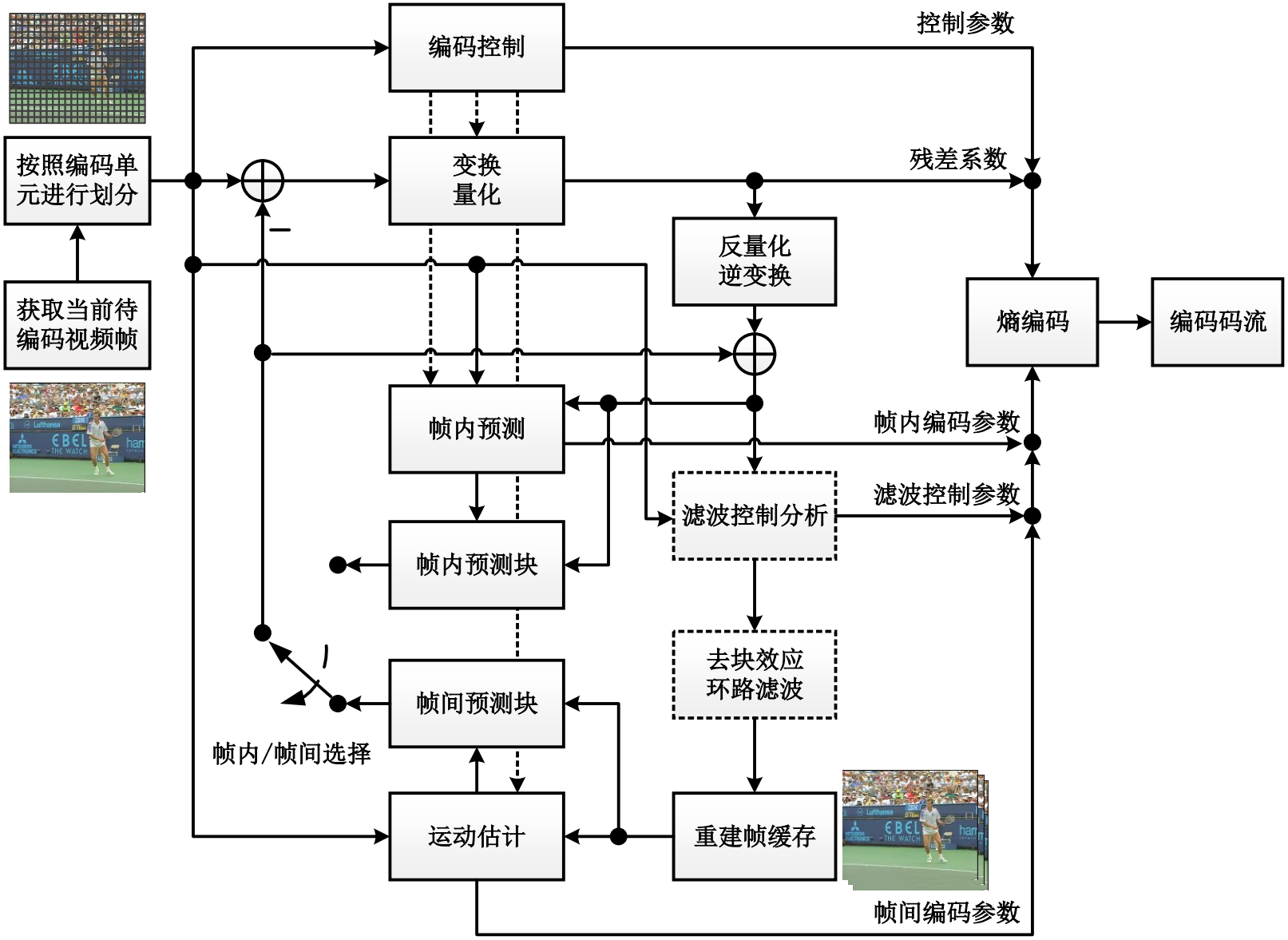
- 档次（Profile）和级（Level）
  - 三个档次共有的编码工具。利用I slice和P slice支持帧内和帧间编码；支持利用基于上下文的自适应的变长编码进行的熵编码；去块效应滤波器；zigzag扫描；1/4像素精度运动估计；4:2:0的亮色度采样。
  - 基本（Baseline）档次。支持条带组、灵活宏块排序和任意条带顺序；支持冗余条带以增加抗误码性；主要应用于可视电话、会议电视以及无线通信等实时视频通信。
  - 主要（Main）档次。支持双向预测；支持加权的帧内预测；支持上下文自适应二进制算数编码器；主要应用于数字广播与数字视频存储。
  - 高级（High）档次。支持基本档次的全部以及主要档次的前两项特性；支持SP/SI条带，用于流间切换、拼接和随机接入等功能；支持数据分割。
  - H.264对所有档次规定了一组相同级别。不同级别用于指定帧大小，处理速率（每秒可以解码的帧或块数）和解码视频序列所需的工作内存上限。

2.3 H.264/AVC视频编码标准常见术语

- 档次（Profile）和级（Level）

2.4 H.264/AVC视频编解码框架

H.264编码器框架



2.4 H.264/AVC视频编解码框架

H.264编码器框架

- **前向编码分支。**首先从当前输入的视频图像中取一个待编码的宏块M，该宏块以帧内或帧间的模式进行编码，生成一个预测宏块P。在帧内预测模式下，P由当前图像中已经编码、解码、重构且未进行去块滤波的宏块经帧内预测得到；在帧间编码模式下，P由一个或多个参考图像进行运动补偿预测得到。当前宏块M减去预测宏块P得到残差块D，对残差块D进行整数变换、量化后得到一组残差系数X，对系数X再进行重排序和熵编码后即完成了一个宏块的编码过程。

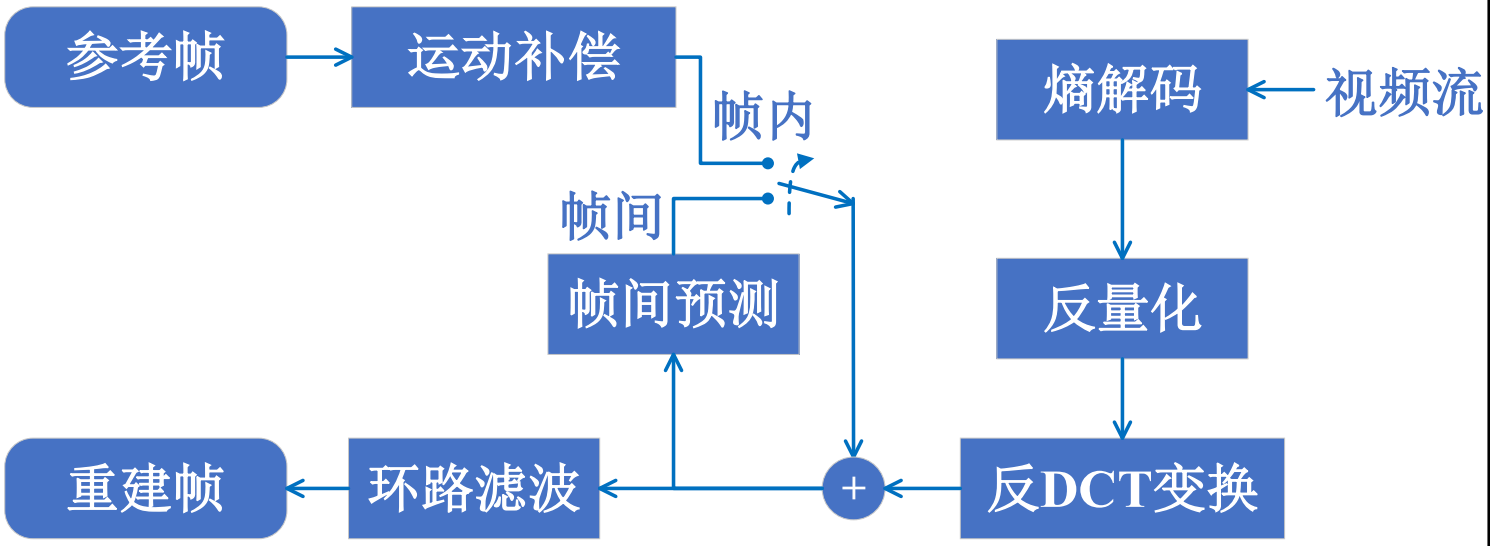
- **后向重建分支。**在后向重建分支中，对量化的宏块系数X进行解码，以得到重建宏块，对后续宏块进行编码需要从一重建的宏块中寻找参考块。具体过程如下：宏块系数X经过反量化和反变换后，得到残差块D的近似值D'（量化过程是有损的），预测块P加上D'得到未滤波的重构宏块，再做环路滤波来减少块效应，即得到了最终的重构宏块。

2.4 H.264/AVC视频编解码框架

H.264解码器框架

- 输入的码流经过熵解码和重排序，得到量化后残差系数X，再经过逆量化、逆变换生成D'，这一过程和编码器重建码流生成的D'过程一致。使用码流中解出的预测块信息，解码器从参考帧中得到预测宏块P，它和编码器形成的预测宏块P相同。P与D'相加得到未滤波的重构宏块，最后对重建图像进行滤波，去除块效应后可得到解码宏块

- 同编码器一样，当前重建图像将用于后续的解码参考，由于上述码流到生成未滤波的重构宏块的过程中与编码端的后向重建分支等同，去块效应滤波器也与编码端的完全一致，这就保证了编解码端使用的参考图像的一致性。



内 容 提 纲 （3 节 课 内 容）

1. 视频编码基础

2. H.264/AVC总体概述

3. H.264/AVC编码标准特性

4. 小结

3.1 帧内预测

帧内预测定义

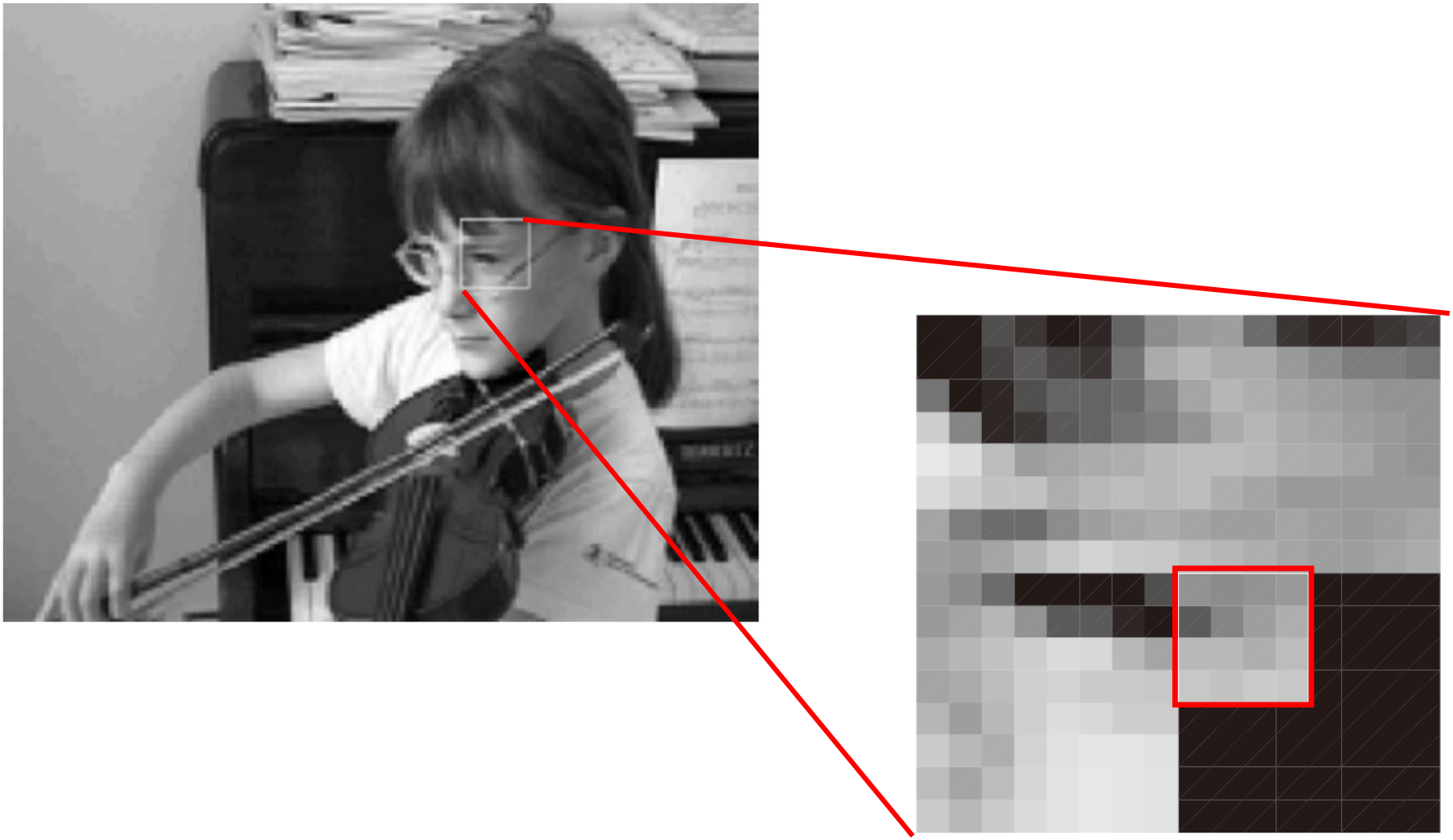
- 帧内预测用于消除视频的空间冗余。视频的空间冗余来源于视频帧内二维像素阵列的空间相关性，即每一个像素大多类似或取决于相邻像素。一个典型的例子如图所示，该视频帧每条水平线的像素是相同的，若直接记录所有像素，将产生冗余信息。

帧内预测技术发展

- MPEG-2等视频压缩标准，对帧内空域像素进行DCT变换转换到频域，只对直流（DC）系数进行差分预测编码。
- H.263、MPEG-4 Visual等视频压缩标准，利用相邻块的频域相关性，利用相邻块直流（DC）/交流（AC）系数预测待编码块的DC/AC 系数。
- H.264/MPEG-4 AVC视频压缩标准进一步挖掘了帧内图像的空间相关性，引入了基于多尺寸分块的空域像素帧内预测技术。编码单元是16 × 16的宏块，分别对亮度和色度进行预测。最佳亮度帧内预测模式和最佳色度帧内预测模式的选择是独立的，分别取决于不同的代价算法。

3.1 帧内预测

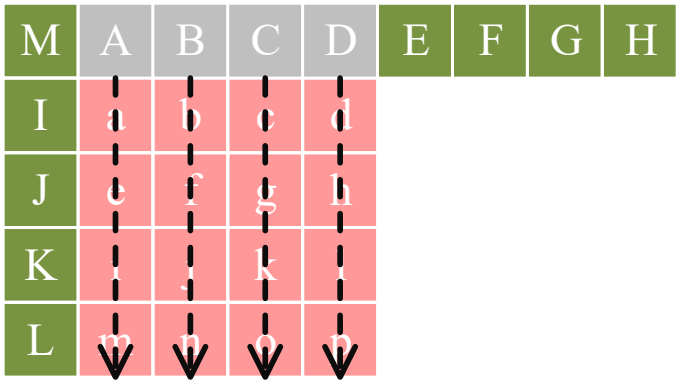
- 4×4块亮度预测

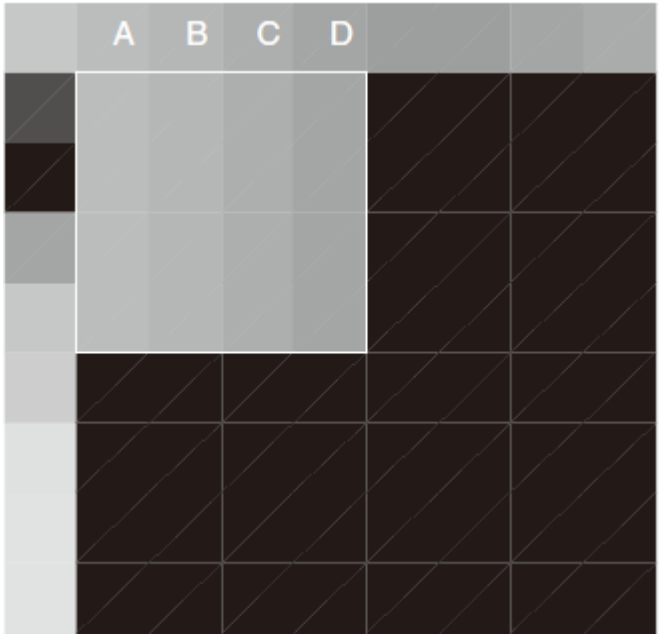


3.1 帧内预测

4×4块亮度预测

- 模式0，垂直预测（vertical prediction）：只有当A，B，C，D可用时，才可使用，预测方式如下：
  - a, e, i, m由A预测（即用A的值当作预测值）
  - b, f, j, n由B预测
  - c, g, k, o由C预测
  - d, h, l, p由D预测



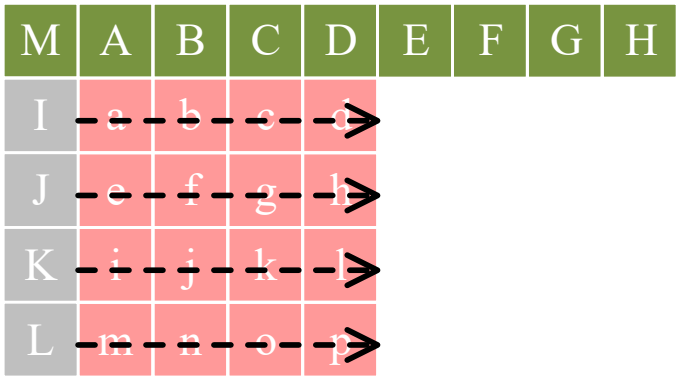


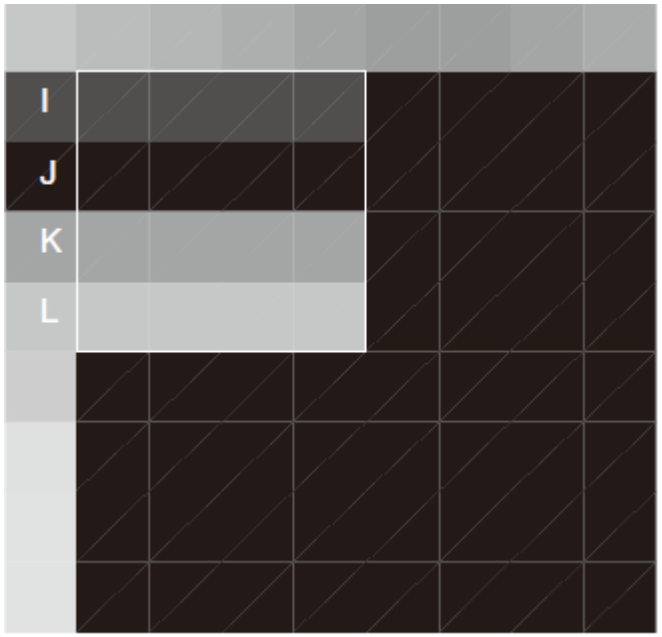
SAD = 317

3.1 帧内预测

4×4块亮度预测

- 模式1，水平预测（horizontal prediction）：只有当I，J，K，L可用时，才可使用，预测方式如下：
  - a, b, c, d由I预测
  - e, f, g, h由J预测
  - i, j, k, l由K预测
  - m, n, o, p由L预测



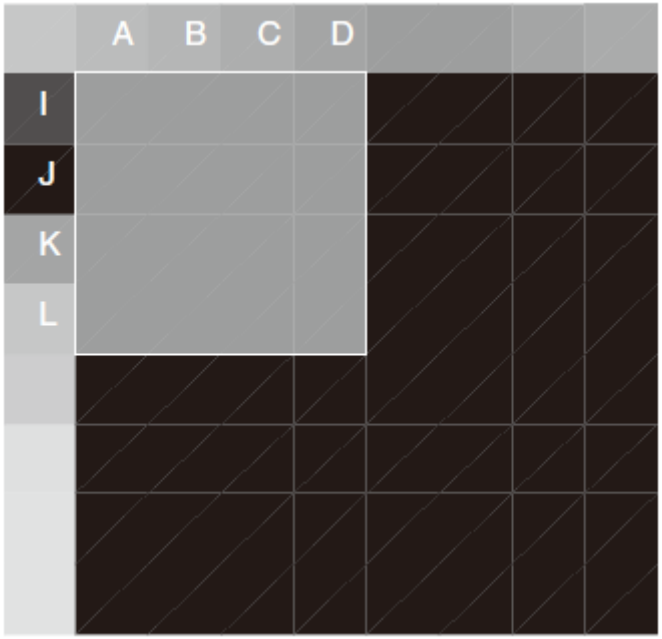


SAD = 407

3.1 帧内预测

4×4块亮度预测

- 模式2，DC预测（DC prediction）：DC模式即平均值模式
  - 当A~L均可用时，a~p预测值均等于(A+B+C+D+I+J+K+L+4) >> 3
  - 当仅I~L可用时，a~p预测值均等于(I+J+K+L+2) >> 2
  - 当仅A~D可用时，a~p预测值均等于(A+B+C+D+2) >> 2
  - 当A~L均不可用时， a~p预测值均等于128，即1 << (BitDepth$_Y$ - 1)

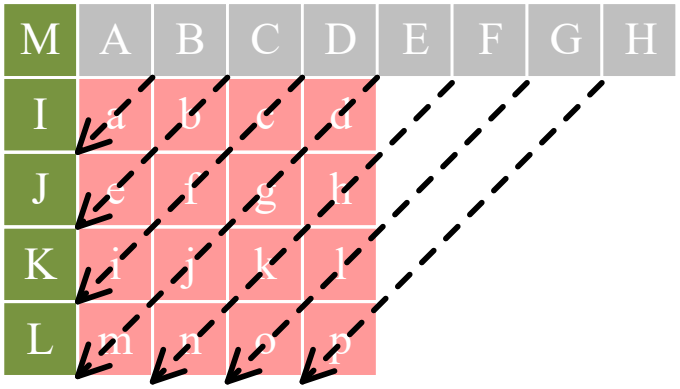


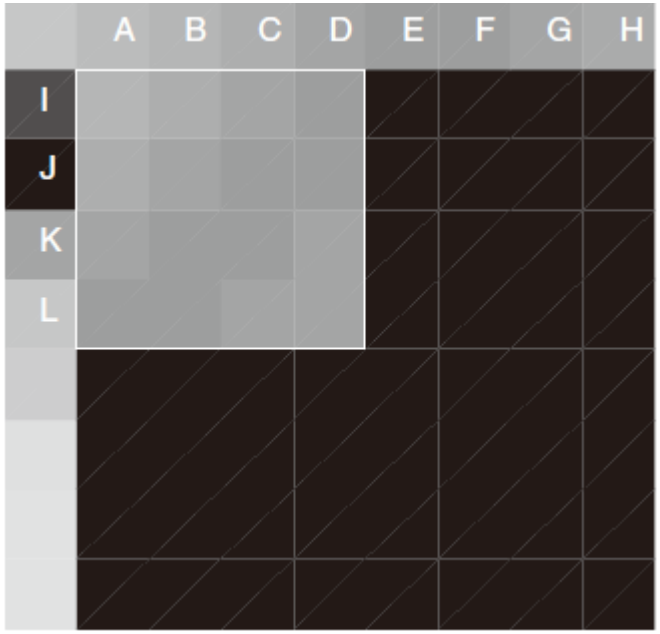
SAD = 317

3.1 帧内预测

4×4块亮度预测

- 模式3，下左对角线预测（diagonal down/left prediction）
  - a的预测值为(A + 2B + C + 2) >> 2
  - b，e的预测值为(B + 2C + D + 2) >> 2
  - c，f，i的预测值为(C + 2D + E + 2) >> 2
  - d，g，j，m的预测值为(D + 2E + F + 2) >> 2
  - h，k，n的预测值为(E + 2F + G + 2) >> 2
  - l，o的预测值为(F + 2G + H + 2) >> 2
  - p的预测值为(G + 3H + 2) >> 2



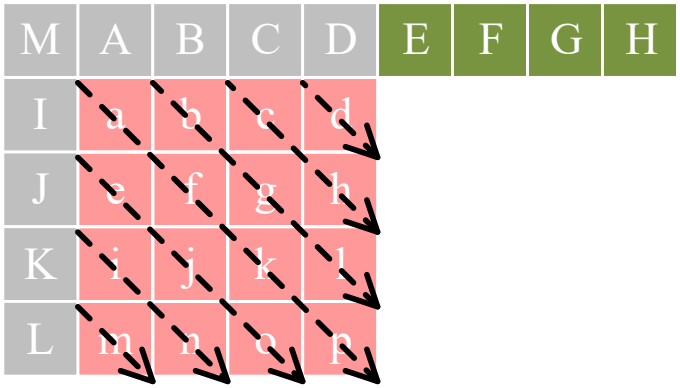


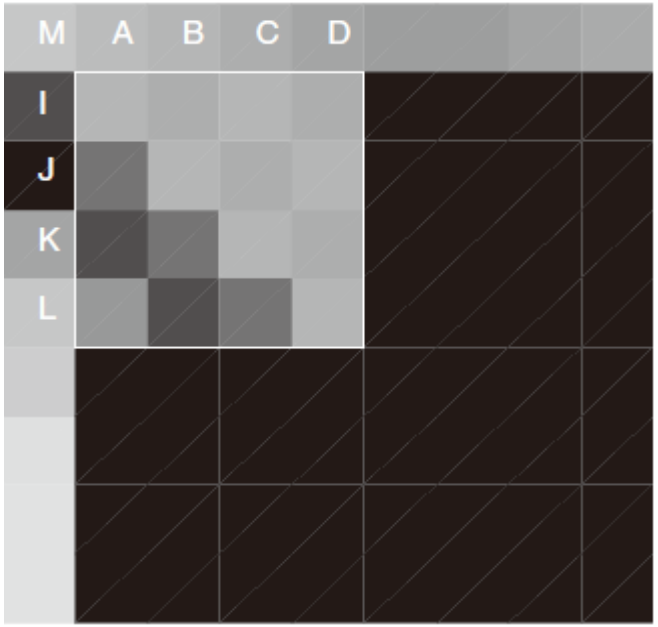
SAD = 350

3.1 帧内预测

4×4块亮度预测

- 模式4，下右对角线预测（diagonal down/right prediction）
  - b, g, l的预测值为(M + 2A + B + 2) >> 2
  - c, h的预测值为(A + 2B + C + 2) >> 2
  - d的预测值为(B + 2C + D + 2) >> 2
  - e, j, o的预测值为(M + 2I + J + 2) >> 2
  - i, n的预测值为(I + 2J + K + 2) >> 2
  - m的预测值为(J + 2K + L + 2) >> 2
  - a, f, k, p的预测值为(A + 2M + I + 2)>>2



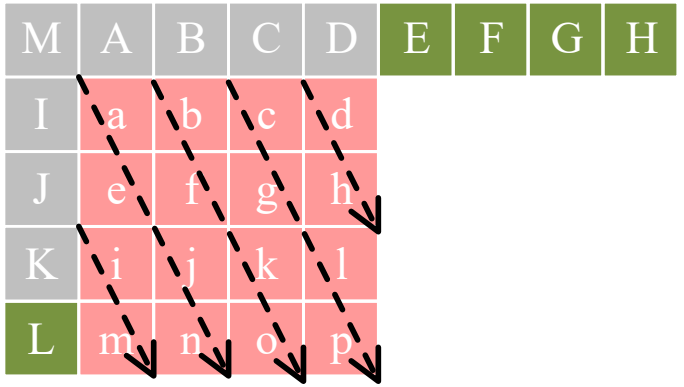


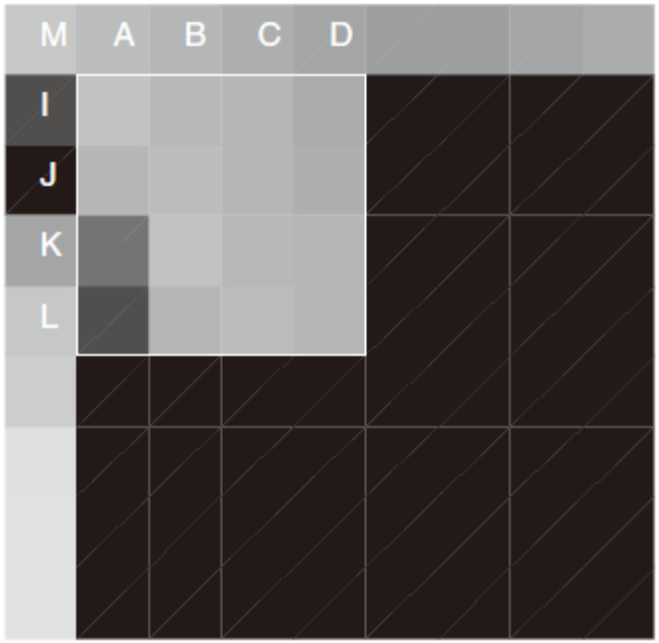
SAD = 466

3.1 帧内预测

4×4块亮度预测

- 模式5，右垂直预测（vertical-right prediction）
  - a，j的预测值为(M + A + 1) >> 1
  - b，k的预测值为(A + B + 1) >> 1
  - c，l的预测值为(B + C + 1) >> 1
  - d的预测值为(C + D + 1) >> 1
  - f, o的预测值为(M + 2A + B + 2) >> 2
  - g, p的预测值为(A + 2B + C + 2) >> 2
  - h的预测值为(B + 2C + D + 2) >> 2
  - e, n的预测值为(I + 2M + A + 2) >> 2
  - i的预测值为(J + 2I + M + 2) >> 2
  - m的预测值为(K + 2J + I + 2) >> 2



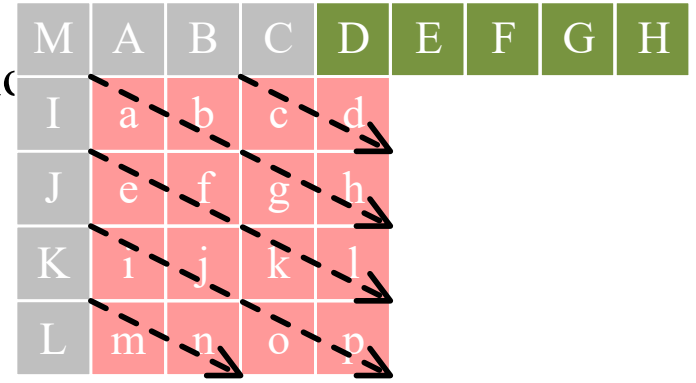


SAD = 419

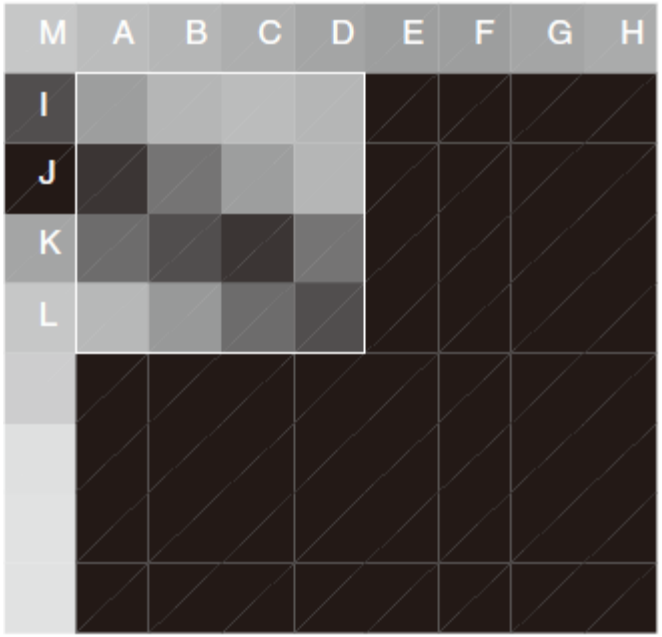
3.1 帧内预测

4×4块亮度预测

- 模式6，下水平预测（horizontal-down predictio



  - a, g的预测值为(M + I + 1) >> 1
  - e, k的预测值为(I + J + 1) >> 1
  - i, o的预测值为(J + K + 1) >> 1
  - m的预测值为(K + L + 1) >> 1
  - f, l的预测值为(M + 2I + J + 2) >> 2
  - j, p的预测值为(I + 2J + K + 2) >> 2
  - n的预测值为(J + 2K + L + 2) >> 2
  - b, h的预测值为(I + 2M + A + 2) >> 2
  - c的预测值为(B + 2A + M + 2) >> 2
  - d的预测值为(C + 2B + A + 2) >> 2

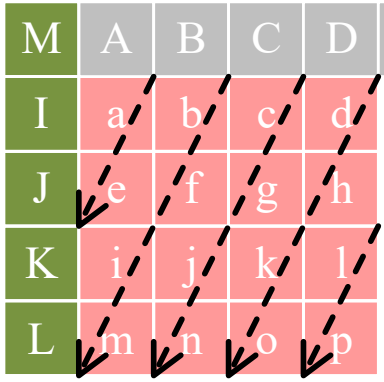


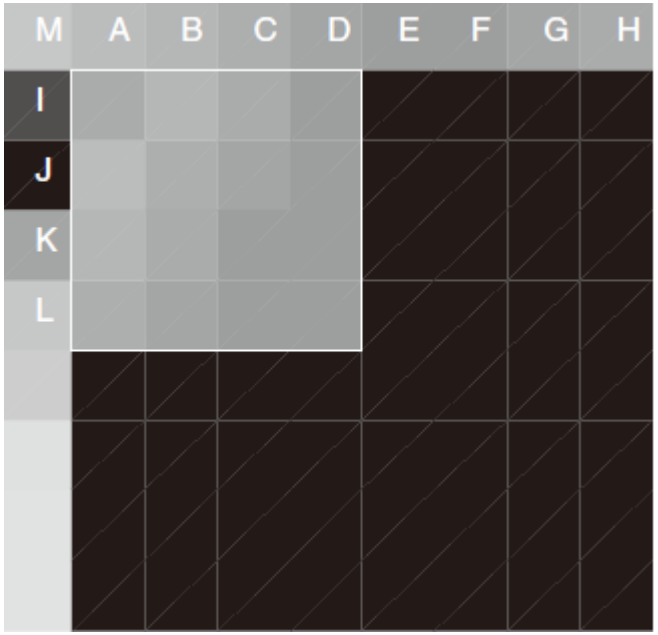
SAD = 530

3.1 帧内预测

4×4块亮度预测

- 模式7，左垂直预测（vertical-left prediction）
  - a的预测值为(A + B + 1) >> 1
  - b, i的预测值为(B + C + 1) >> 1
  - c, j的预测值为(C + D + 1) >> 1
  - d, k的预测值为(D + E + 1) >> 1
  - l的预测值为(E + F + 1) >> 1
  - e的预测值为(A + 2B + C + 2) >> 2
  - f, m的预测值为(B + 2C + D + 2) >> 2
  - g, n的预测值为(C + 2D + E + 2) >> 2
  - h, o的预测值为(D + 2E + F + 2) >> 2
  - p的预测值为(E + 2F + G + 2) >> 2



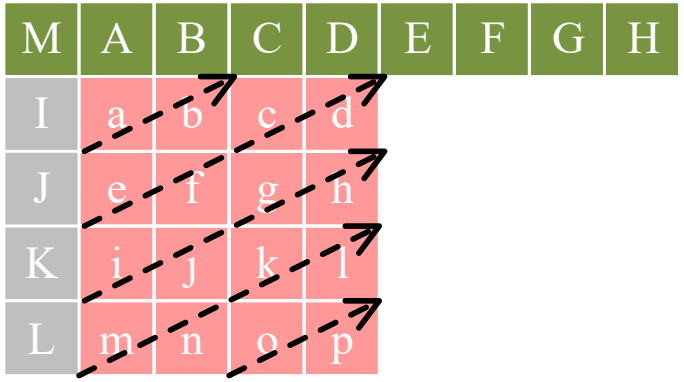


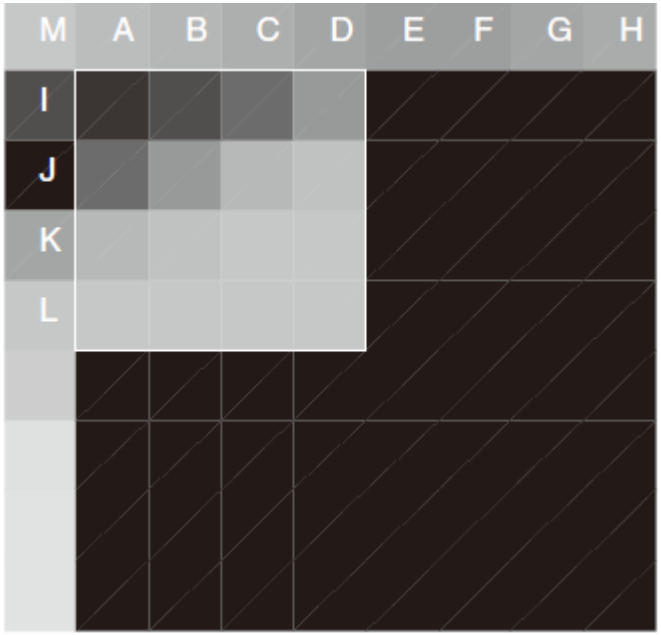
SAD = 351

3.1 帧内预测

4×4块亮度预测

- 模式8，上水平预测（horizontal-up prediction
  - a的预测值为(I + J + 1) >> 1
  - e, c的预测值为(J + K + 1) >> 1
  - i, g的预测值为(K + L + 1) >> 1
  - b的预测值为(I + 2J + K + 2) >> 2
  - f, d的预测值为(J + 2K + L + 2) >> 2
  - j, h的预测值为(K + 3L + 2) >> 2
  - m, k, n, l, o, p的预测值为L



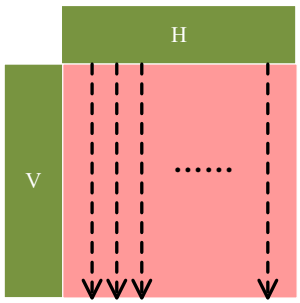


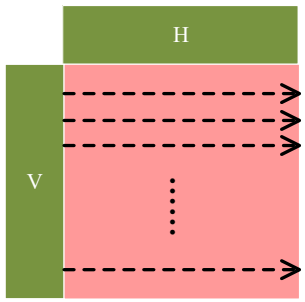
SAD = 203

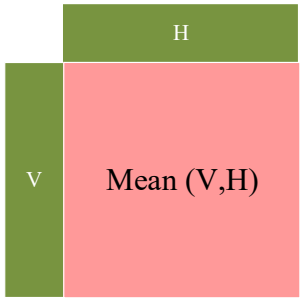
3.1 帧内预测

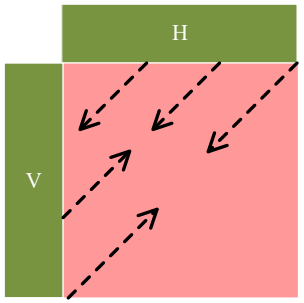
16×16块亮度预测

- 模式0，垂直预测（vertical prediction）
  - 使用上方参考元素垂直预测
- 模式1，水平预测（horizontal prediction）
  - 使用左侧参考元素水平预测
- 模式2，DC预测（DC prediction）
  - 使用上方和左侧参考元素的平均值预测
- 模式3，平面预测（plain prediction）
  - 使用上方和左侧参考元素通过线性函数进行预测









3.1 帧内预测

帧内预测流程（以基本档次为例）

- 步骤1：得到宏块$\mathbf{M}^{16\times16}$的亮度分量$\mathbf{B}$，分别计算16×16预测模式和4×4预测模式的代价。若16×16预测模式代价较小，跳转步骤3执行。

- 步骤2：将$\mathbf{B}$划分为互不重叠的4×4子块，以如图所示4×4亮度块扫描顺序，分别计算每个子块的预测块，合并得$\mathbf{B}$的预测块$\mathbf{P}$，转步骤4执行。

0	**1**	**4**	**5**
2	**3**	**6**	**7**
8	**9**	**12**	**13**
10	**11**	**14**	**15**

- 步骤3：计算$\mathbf{B}$的预测块$\mathbf{P}$。

- 步骤4：计算残差块$\mathbf{R}=\mathbf{B}-\mathbf{P}$，将$R$划分为互不重叠的$4\times4$子块，逐个进行后续变换和编码操作。

3.1 帧内预测

帧内预测模式代价计算

- Intra4×4代价计算公式使用的是率是真优化（Rate Distortion Optimization，RDO）模型

$$J(\boldsymbol{s},\boldsymbol{c},IMODE|QP,\lambda_{\mathrm{MODE}}) = SSD(\boldsymbol{s},\boldsymbol{c},IMODE|QP) + \lambda_{\mathrm{MODE}} \cdot R(\boldsymbol{s},\boldsymbol{c},IMODE|QP)$$

$$SSD(\boldsymbol{s},\boldsymbol{c}) = \sum_{i,j} |\boldsymbol{s}(i,j) - \boldsymbol{c}(i,j)|^2$$

  - QP是量化参数；λ_{MODE}是模式选择的拉格朗日乘子；$IMODE$代表帧内亮度4×4的9种预测模式中的一种；$\boldsymbol{s}$是原始亮度分块；$\boldsymbol{c}$是重构亮度分块；SSD是s和c的平方误差和；$R(\boldsymbol{s},\boldsymbol{c},IMODE|QP)$代表选择$IMODE$模式时所需的编码比特数。

- 基于16×16块的预测的代价计算使用的是绝对变换误差和（Sum of Absolute Transformed Differences，SATD）

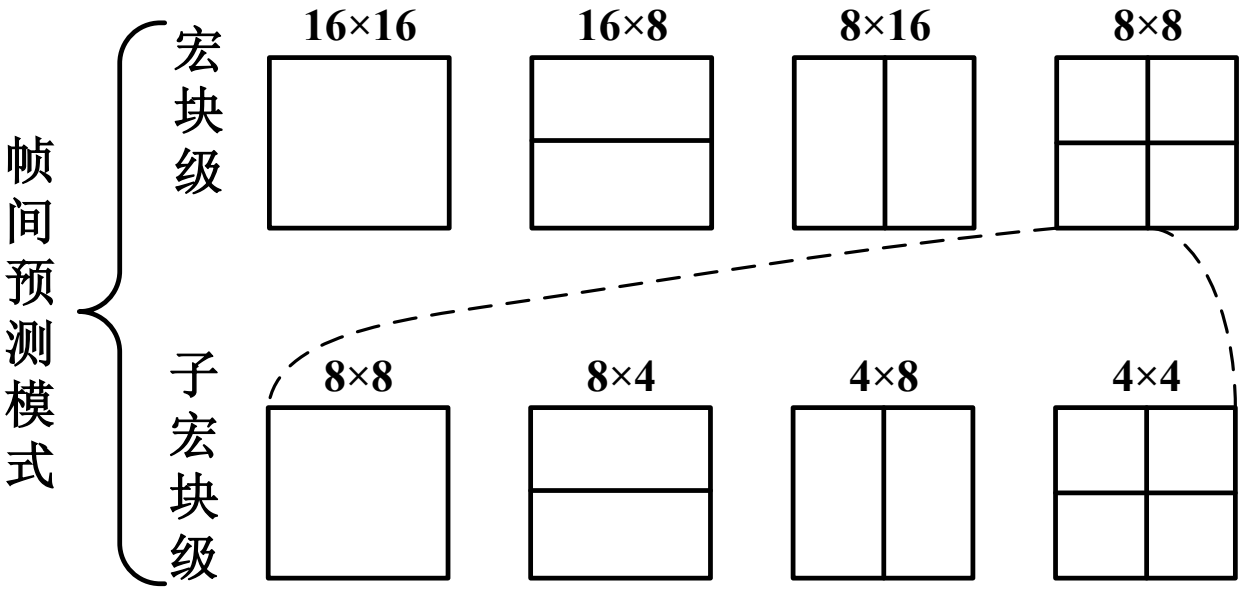
$$\mathrm{SATD} = \sum_{i,j} |T\{s(x,y) - c(x,y)\}|$$

  - T表示哈达玛变换，哈达玛变换使用的哈达玛矩阵是由$+1$和-1元素构成的正交方阵。

3.2 帧间预测

H.264/AVC树状结构分块

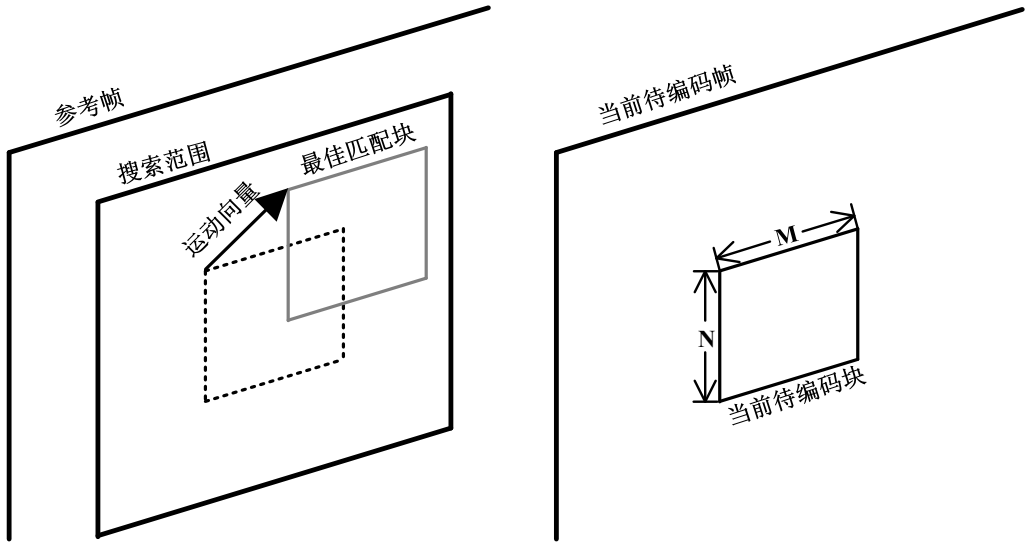
- H.264/AVC标准中，宏块的两级树状结构如图所示，在第一级分割中，分块从大到小可以分为16×16，16×8，8×16或8×8的宏块分割。对于最小的8×8子宏块（Sub-macroblock），可以进一步分割为8×4，4×8或4×4的子宏块，这一级的分割称为子宏块分割。在进行运动估计的时候，每种分块模式通常要被尝试一次，通过运动搜索计算出宏块各种可能的分块方式所能得到的最小代价，选取这些最小代价中最小值对应的分块模式作为该宏块的最佳帧间划分模式。



3.2 帧间预测

基于块的运动估计与运动补偿

- 基于块的运动估计（Motion Estimation，ME）是在参考帧（Reference Frame）中的某个搜索范围内，按照给定的搜索算法和块匹配准则，寻找当前待编码块（如16 × 16 亮度块）的最佳匹配块。
- 参考帧是已经过编码并重建的视频帧，在播放顺序上可以位于当前待编码视频帧之前或之后；最佳匹配块相对于当前待编码块的位置偏移采用运动向量MV表示；块匹配准则通常是关于预测残差（当前待编码块和参考块之间的差异）和运动向量的代价函数。
- 基于块的运动补偿（Motion Compensation）是指从当前待编码块中减去运动估计所得的最佳匹配块从而形成残差块的过程。



3.2 帧间预测

运动估计搜索算法

- 视频编码标准并未规定运动估计所需采用的搜索算法，其可分为两种类型：**全搜索**和**快速搜索**。

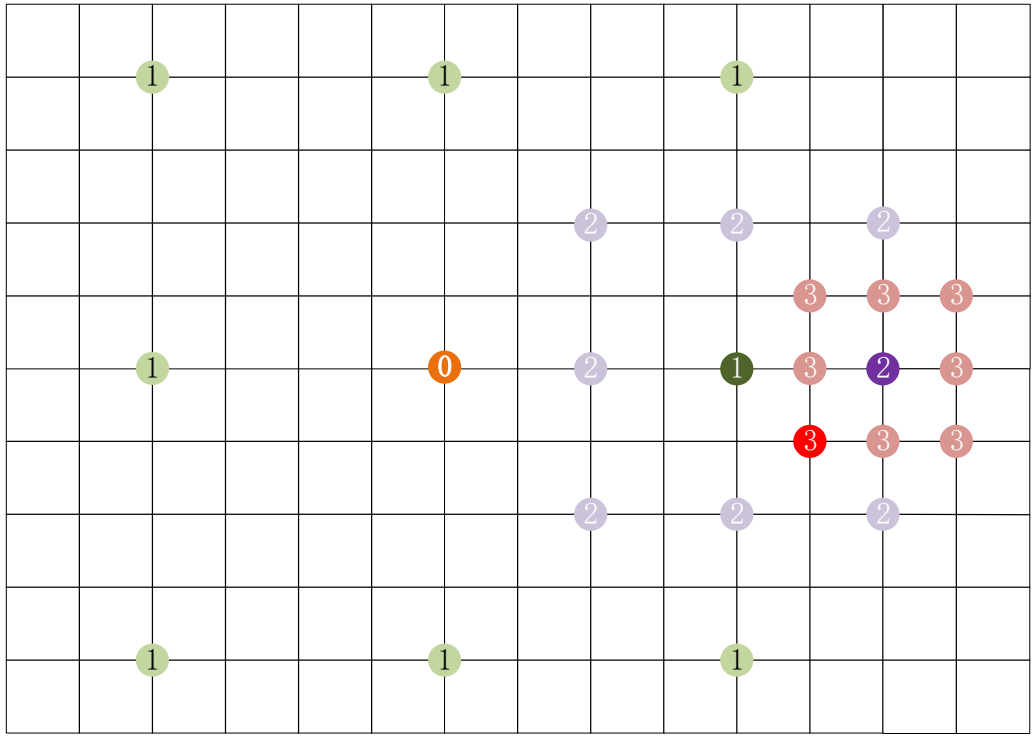
- 全搜索算法需要遍历参考帧中搜索范围内的每个参考块，分别计算它们的代价函数，从中选取具有最小代价函数值的参考块作为最佳匹配块。可以看出，全搜索算法虽然可以保证得到（搜索范围内的）全局最佳匹配块（即代价函数的全局最优解），但是算法时间复杂度较高，在实际应用中较少采用。

- 快速搜索算法的基本思想是通过设计有效的搜索策略或模式，从而以尽可能少的搜索位置确定最佳匹配块。经典的快速搜索算法包括：三步搜索算法、钻石型搜索算法、六边形搜索算法、非对称十字多层六边形格点搜索算法（Unsymmetrical cross Multi-Hexagon-grid search，UMH）、增强预测区域搜索算法（Enhanced Predictive Zonal Search，EPZS）。需要注意的是，某些快速搜索算法只能确保得到局部最佳匹配块（即代价函数的局部最优解），无法保证得到全局最佳匹配块。

3.2 帧间预测

三步搜索算法

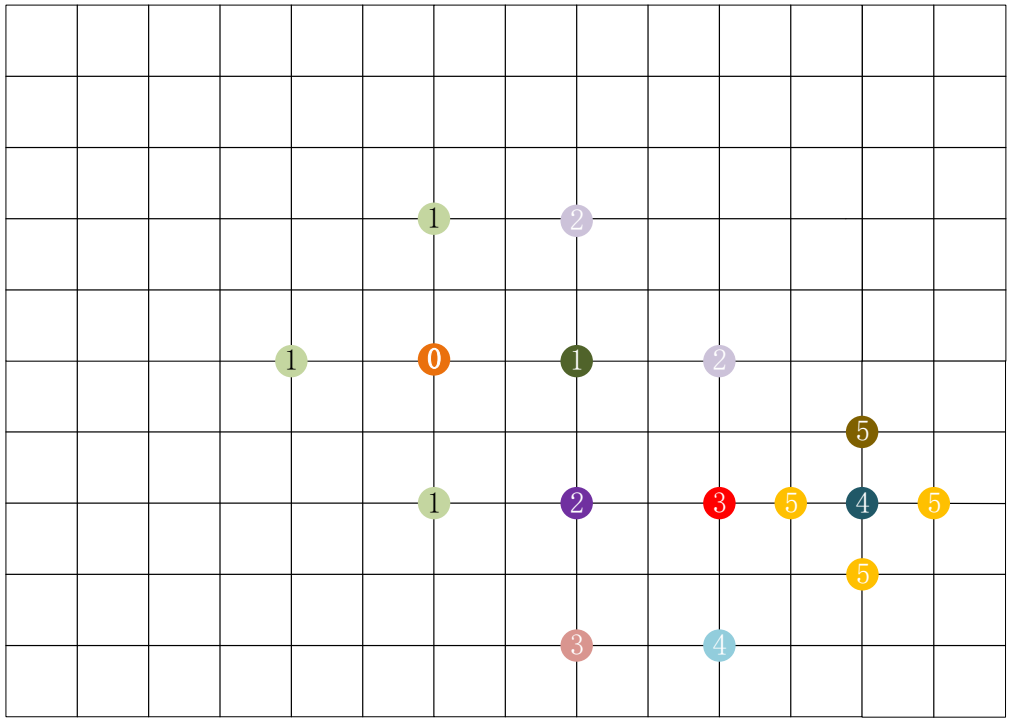
- 从原点0出发，以4为补偿，搜索中心点0和8个标记为1的点，依据准则函数，从这9个点中找到最匹配点，并记为1S。
- 从点1S出发，以2为步长，搜索1S和8个标记为2的点，依据准则函数，从这9个点中找到最佳匹配点，记为2S。
- 从点3S出发，以1为步长，搜索2S和8个标记为3的点，依据准则函数，从这9个点中找到最佳匹配点，作为最终结果输出。



3.2 帧间预测

二维对数搜索算法

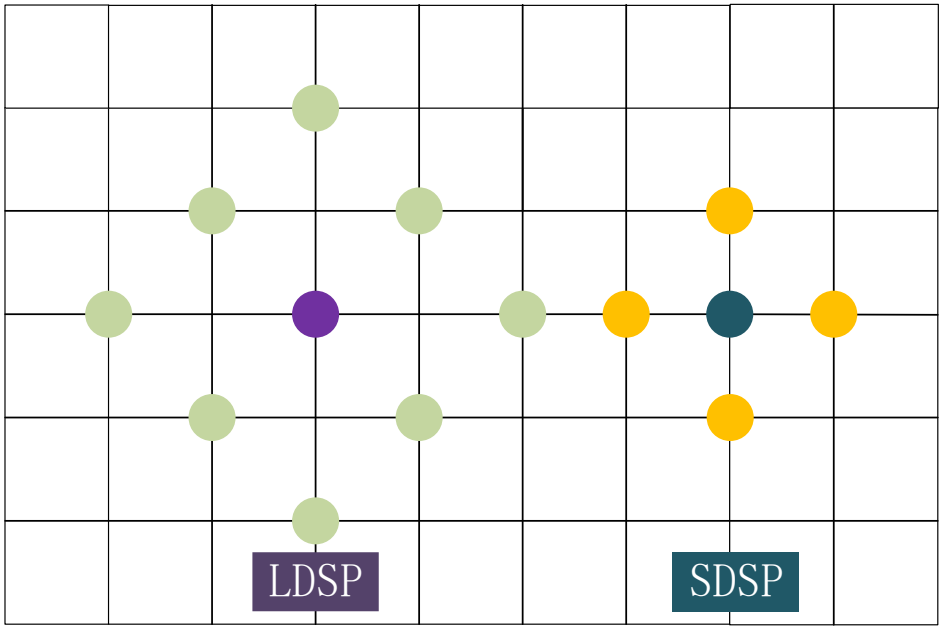
- 从原点0出发，选取一定的步长，搜索以十字形分布的5个点。
- 若最小点在边缘4个点处，则以该点作为中心点保持步长不变，重新搜索十字形分布的5个点。若最小点位于中心点，则保持中心点位置不变，将步长减半，搜索十字形分布的5个点。如果新的十字形点群的中心位于搜索窗的边缘，这时步长也要减半。
- 不断减少步长，直到步长为1。这时在中心点及周围步长为1的点中，找出最佳点即为算法输出。



3.2 帧间预测

菱形搜索算法

- 从原点出发，用LDSP*进行搜索。如果计算得到的最佳点位于中心位置，则转到步骤3，否则转到步骤2。
- 以上一搜索步骤中得到的最佳点为中心，重新构造一个新的LDSP进行搜索。如果得到的新最小块失真点位于中心位置，则转到步骤3，否则重复此步骤，直至检测点到达搜索窗口边缘才终止搜索。
- 以上一搜索步骤中得到的最佳点为中心，用SDSP**进行搜索。本部搜索得到的最佳点即为算法的输出。



*LDSP：大菱形搜索模板，由9个点组成。

**SDSP：小菱形搜索模板，由5个点组成。

3.2 帧间预测

运动估计分块尺寸

- 通常情况下，用于运动估计的分块尺寸越小，运动补偿所得残差块的能量越低。

- 对待编码帧进行运动估计时，若单纯采用较小尺寸的分块，将增加需要进行运动估计的分块的数量，这具有以下两点局限性：
  - 首先，需要更多的最佳匹配块搜索操作，增加了帧间预测的时间复杂度；
  - 其次，生成了更多的运动向量，增加了编码运动向量所需的比特数。这两点局限性在一定程度上可能抵消采用小尺寸分块进行运动估计产生的增益（即运动补偿所得残差块的能量较低）。

- 第二、第三代视频编码技术在进行运动估计时，通常根据待编码视频帧的内容特性，综合采用多种尺寸的分块：在纹理平坦区域采用较大尺寸的分块；在纹理复杂区域和运动剧烈区域采用较小尺寸的分块。

3.2 帧间预测

运动估计搜索精度

- 在参考帧中采用亚像素（如半像素、四分之一像素、八分之一像素）步长进行最佳匹配块搜索，通常能够获得更加理想的运动估计结果。

- 当前主流的视频编码技术通常采用亚像素（Sub Pixel）运动估计，通过搜索参考帧中位于整像素和亚像素位置上的参考块以确定最佳匹配块。

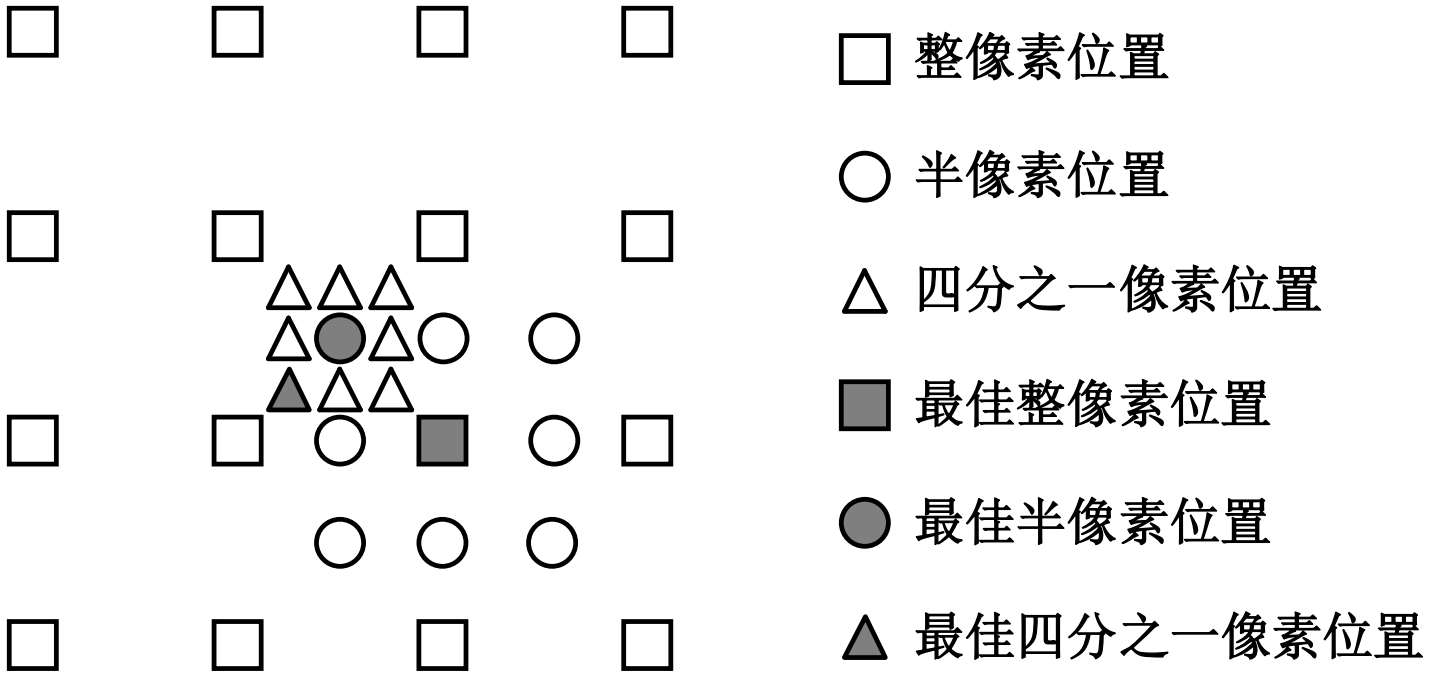
- 由于视频帧只包含整像素点，故亚像素点需要通过插值（Interpolation）计算得到。通常情况下，经过插值的参考帧分辨率越高（即插值越精细），运动估计的结果越精确，运动补偿去除时间冗余的效果越好，从而能够达到更高的压缩编码效率。

- 然而，一方面，插值精度的提升将扩大参考块的搜索空间，提高帧间预测的时间复杂度；另一方面，亚像素运动估计所获得的压缩性能增益将随着插值精度的提升而逐渐降低：半像素运动估计相比整像素运动估计，能够获得最大程度的编码性能提升；四分之一像素运动估计相比半像素运动估计，可获得中等程度的编码性能提升；八分之一像素运动估计相比四分之一像素运动估计，在编码性能的提升程度上将进一步下降，以此类推。

3.2 帧间预测

运动估计搜索精度

- 在进行亚像素运动估计时，综合考虑了时间复杂度和压缩性能增益：先进行整像素运动估计，获得最佳整像素位置；再缩短搜索步长，在此最佳整像素位置周围寻找位于半像素位置的最佳匹配块；此后，在所得最佳匹配块（位于整像素或半像素位置）附近，搜索是否存在位于四分之一像素位置的参考块，使得能够进一步提高运动估计的性能。



3.2 帧间预测

块匹配准则

- 进行运动估计时，通常采用拉格朗日（Lagrangian）代价函数作为块匹配准则，选择具有最小率失真（Rate Distortion）代价的运动向量作为运动估计结果。拉格朗日代价函数的数学表达式为：

$$J_{ME} = D + \lambda R$$

  - D表示当前进行运动估计的分块和其相应运动向量所指向的参考块之间的差异；R表示编码运动估计数据（包括运动向量和参考帧索引）所需的比特数；λ代表拉格朗日乘子（Lagrangian Multiplier）。其和量化步长（Quantization Step Size）相关，用于控制失真D和编码比特数R之间的平衡。较大的λ倾向于以牺牲视觉保真度为代价降低码率，较小的λ侧重于以高码率开销为代价降低视觉失真。

3.2 帧间预测

块匹配准则

- 进行整像素运动估计时，通常采用SAD（Sum of Absolute Differences）指标衡量失真D；进行亚像素运动估计时，D通常采用SATD（Sum of Absolute Transformed Differences）指标进行计算。给定两个矩阵$\mathbf{X}$和$\mathbf{Y}$，则$\mathrm{SAD}(\mathbf{X},\mathbf{Y})$和$\mathrm{SATD}(\mathbf{X},\mathbf{Y})$的数学表达式分别为：

$$\mathrm{SAD}(\mathbf{X},\mathbf{Y}) = \sum_{i,j} |\mathbf{X}(i,j) - \mathbf{Y}(i,j)|$$

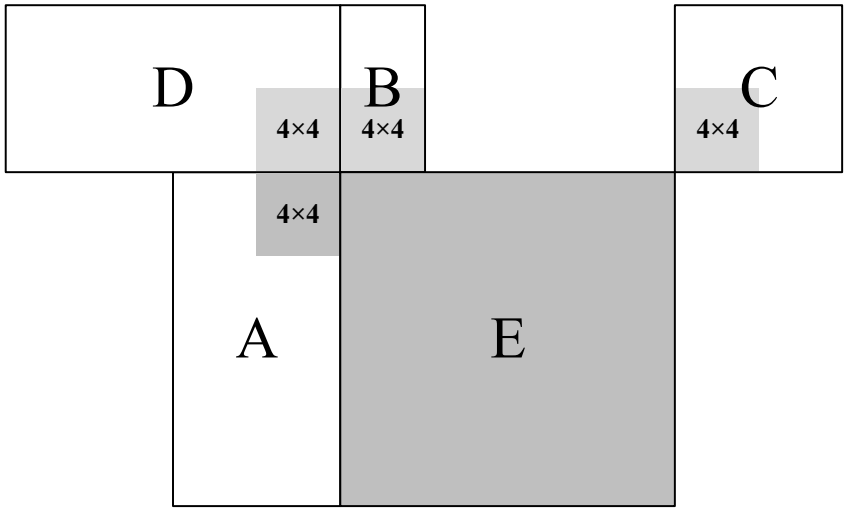
$$\mathrm{SATD}(\mathbf{X},\mathbf{Y}) = \sum_{i,j} (|\mathbf{HT}_{4\times4}(i,j)|)$$

  - $\mathbf{X}(i,j)$和$\mathbf{Y}(i,j)$分别表示$\mathbf{X}$和$\mathbf{Y}$中第(i,j)个元素值；$\mathbf{HT}_{4\times4}$代表$\mathbf{X}$和$\mathbf{Y}$做差后进行4×4哈达玛变换所得的矩阵。

3.2 帧间预测

运动向量预测值（Motion Vector Prediction，MVP）的计算

- 假设E为当前分块，A为E左方最上边4×4邻块所属的分块，B为E上方最左边4×4邻块所属的分块，C为E右上角对角4×4邻块所属分块。E的运动向量预测值MVP可以由邻块A、B和C的运动向量预测得到。
- 对于16×8分割，上方16×8块的MVP根据B预测得到，下方16×8块的MVP根据A预测得到；
- 对于8×16分割，左侧8×16块的MVP根据A预测得到，右侧8×16块的MVP根据C预测得到；
- 对于除16×8和8×16之外的其他分割，MVP取A、B、C块运动向量的中值（对于A、B、C块运动向量的两个分量，分别取中值后组合成MVP）。



3.2 帧间预测

B帧预测

- B帧的预测是双向的，分别参考List0和List1两个参考帧列表进行前向和后向预测。List0为前向参考帧列表，List1 为后向参考帧列表，P帧的帧间预测只用到List0中的参考帧。

- 编码器首先编码一个I帧，然后向前跳过几个帧，将编码过的I帧作为参考帧对该帧进行P帧编码，然后把编码过的I 帧和P帧之间的显示序列中的空隙用B帧填充，参考I帧、P 帧和已经编码过的B帧进行编码。此后，编码器会再次跳过几个帧，使用第一个P帧作为基准帧编码下一个P帧，然后再次跳回，将编码过的两个P帧之间的帧编码成B帧。不断重复此过程，一直编码到帧序列的结尾。

- H.264/AVC中，B宏块的预测模式有四种：直接预测（Direct）模式、双向预测模式、利用List0的单向预测模式和利用List1的单向预测模式。对于不同尺寸的分块，预测模式只能在一定范围内筛选：只有16×16和8×8块能够采用Direct模式；8×8块所选择的预测模式会应用到其中的所有子分块；B 宏块在预测过程中采用双向预测，但实际编码时不一定都有两个参考帧。由于B宏块的单向预测与P宏块类似

3.2 帧间预测

B帧预测模式

- 直接预测模式。
  - 16×16和8×8块可以采用直接预测模式，其特点为：有像素残差、无运动向量残差（MVD）、无参考帧号。解码时，通过直接预测模式计算出前、后向的运动向量，并利用它们得到像素预测值，进而计算像素预测值与残差解码值之和，将其作为像素重构值。直接预测模式可以节省大量比特，因为不需要传输MVD和参考帧号，它们可以通过List0和List1中的已编码帧直接计算出来。

- 双向预测模式
  - 双向预测使用分别位于List0和List1中的两个方向参考帧进行运动补偿。在两个参考帧列表中分别进行运动估计，得到前向和后向运动向量（MV0和MV1）；利用邻近块的同方向运动向量计算该块两个方向的MVP（MVP1 和MVP2），并且用MVP 作为运动估计的搜索起点；通过同方向运动向量和MVP计算出相应的MVD进行编码传输。得到运动向量和使用的参考帧之后，就可以确定两个方向的参考块，进而计算出预测像素值。

内 容 提 纲 （3 节 课 内 容）

1. 视频编码基础

2. H.264/AVC总体概述

3. H.264/AVC编码标准特性

4. 小结

4 小结

- 视频编码标准的发展
  - 数字视频原理
  - 常用视频编码基础
  - 主要视频编解码标准
- H.264/AVC总体概述
  - 编码特性
  - 常用术语
  - 编解码框架
- H.264/AVC编码标准特性
  - 帧内预测
  - 帧间预测

谢 谢

Q&A

欢迎电子邮件、QQ与微信交流问题！