



# Digital Asset Management

数字媒体资源管理

## 2. Introduction to Digital Media Format



任课老师：张宏鑫

2019-09-19

# Outline

- Image format and coding methods
- Audio format and coding methods
- Video format and coding methods
- Introduction to HTML and XML
- Graphics format and coding methods

# LZW Encode

"**X X X D Z D D D Z D Z**"

where the root values are X, D, Z. The following table follows the above [encoding algorithm](#), showing the values of P, S and P + S (the pattern added to the dictionary) at the end of each iteration of the while loop, as well as the dictionary that is built up:

	P	S	P + S	Output	Dictionary	
					Code	Pattern
					0	X
					1	D
					2	Z
	X	X				
X	X	XX	0	3	XX	
X	X	XX				
XX	D	XXD	3	4	XXD	
D	Z	DZ	1	5	DZ	
Z	D	ZD	2	6	ZD	
D	D	DD	1	7	DD	
D	D	DD				
DD	Z	DDZ	7	8	DDZ	
Z	D	ZD				
ZD	Z	ZDZ	6	9	ZDZ	
Z		Z	2			

So the final output is:

"**0 3 1 2 1 7 6 2**"

And the input has been compressed from 11 symbols down to 8.

# LZW Decode

"0 3 1 2 1 7 6 2"

The following follows the [decoding algorithm](#), showing the necessary values for producing the decoded output:

	P	C	LOOK_UP(P)	LOOK_UP(C)	Output	Dictionary	
						Code	Pattern
						0	X
						1	D
						2	Z
	0			X	X		
0	3	X		XX	XX	3	XX
3	1	XX		D	D	4	XXD
1	2	D		Z	Z	5	DZ
2	1	Z		D	D	6	ZD
1	7	D		DD	DD	7	DD
7	6	DD		ZD	ZD	8	DDZ
6	2	ZD		Z	Z	9	ZDZ

So the decoded output is:

"X X X D Z D D D Z D Z"

The grey rows show the situation where LOOK\_UP(C) is output as soon as it is added to the dictionary, and it is added to the dictionary as

LOOK\_UP(P) + 1st character of LOOK\_UP(P)

This situation occurs when the algorithm comes across a code for C that is not in the dictionary. It is then deduced entirely from P as above, added to the dictionary, and then output. This case is caused in the encoding when the encoder outputs a code just after it was added to the dictionary.

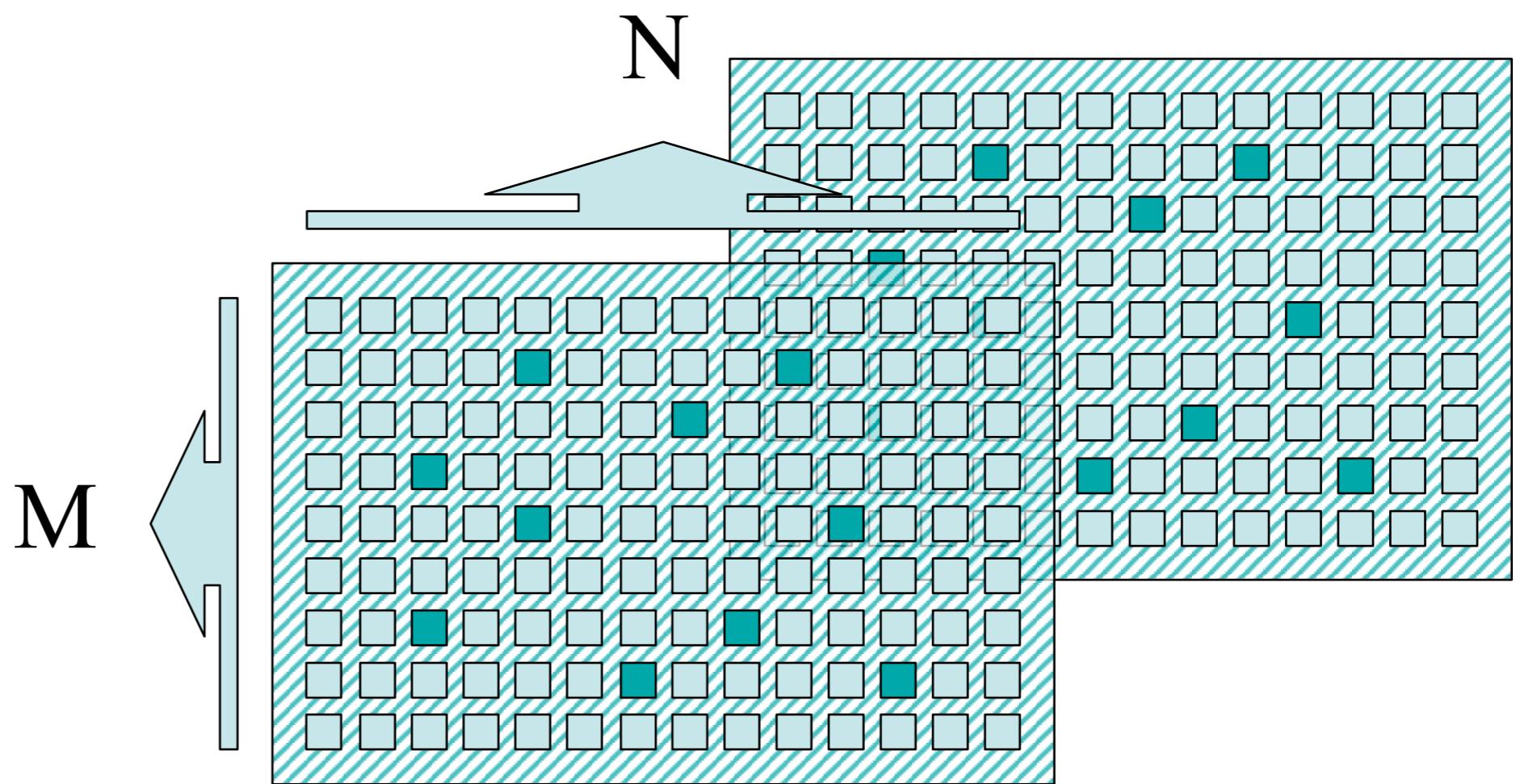


## 2.3. Video formats and coding methods



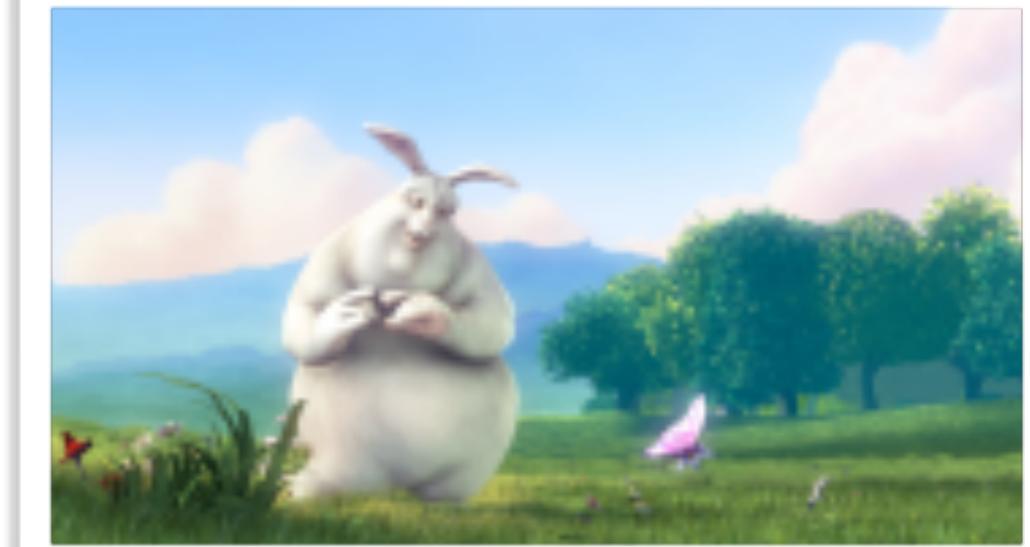
# Representations of video

- Sequence of images ?!?
  - Can be viewed as a 3-dimensional matrix
  - But it is only 50% correct



# Common video formats

- AVI (Microsoft, Divx, ...)
  - avi, wmv, asf
- RM (Realplayer)
  - rm, rmvb
- MOV (Quicktime)
  - mov
- MPEG
  - MPEG-1, MPEG-2, MPEG-4 ...



<http://www.bigbuckbunny.org/index.php/download/>

# Video compression standards

- **MPEG standards**
  - Audio/Video compression, storage and play back standards
  - MPEG-1: VCD
  - MPEG-2: broadcast TV, e.g., DVD、HDTV etc.
  - MPEG-3: replaced by MPEG-2
  - MPEG-4: network video transfer, stream media
  - MPEG-7:
  - MPEG-21:
- **ITU-T H.26x series**

# MPEG-1 Standard ISO/IEC 11172-2 (1991)

## "Coding of moving pictures and associated audio for digital storage media"

- Video

- optimized for bit rates around 1.5 Mbit/s
- originally optimized for SIF picture format,
- but not limited to it:
  - [ **NTSC based** ] : 352x240 pixels at 30 frames/sec
  - [ **PAL based** ] : 352x288 pixels at 25 frames/sec
- progressive frames only
  - no direct provision for interlaced video applications, such as broadcast television

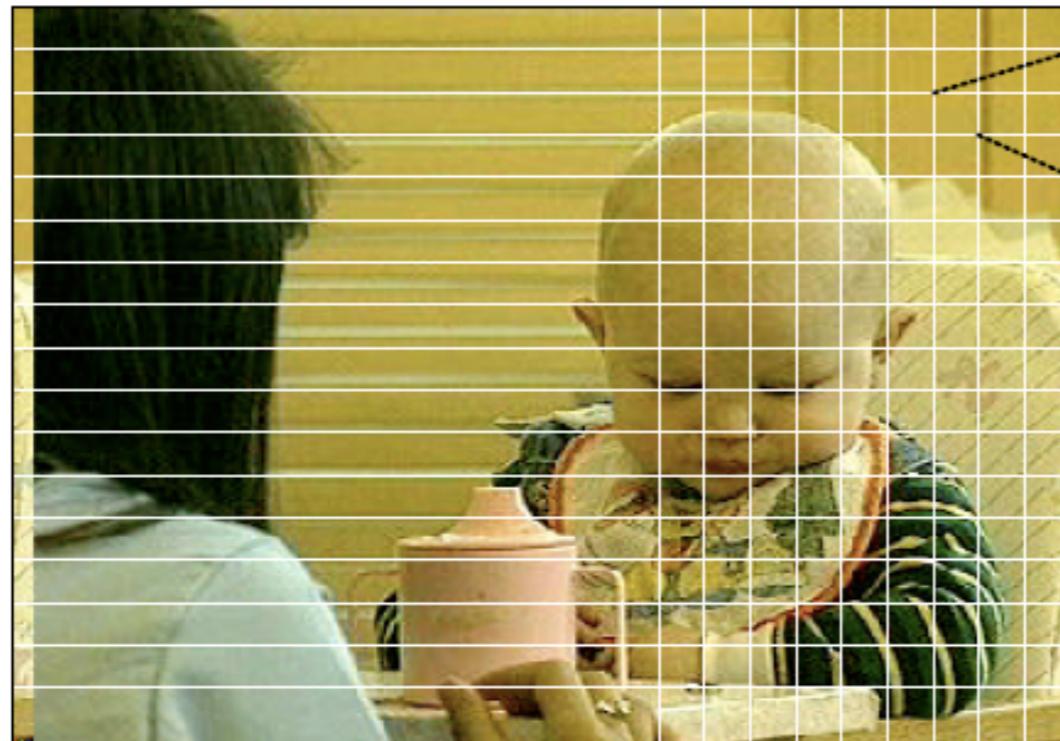
# MPEG-2 Standard ISO/IEC 13818-2 (1994)

- Video
  - 2-15 or 16-80 Mbit/s bit rate ( target bit rate: 4...9 Mbit/sec )
  - TV and HDTV picture formats
  - Supports interlaced material
- MPEG-2 consists of *profiles* (类) and *levels* (级)
  - Main Profile, Main Level (MP@ML)
    - 720x480 resolution video at 30 frames/sec
    - < 15 Mbit/sec (typical ~4 Mbit/sec)
    - for NTSC video
  - Main Profile, High Level (MP@HL)
    - 1920x1152 resolution video at 30 frames/sec
    - < 80 Mbit/sec (typical ~15 Mbit/sec)
    - HDTV

# MPEG-1 v.s. MPEG-2

- MPEG-1 Apps ~
  - CD-I, digital multimedia,
  - video database, e.g. video-on-demand
- MPEG-2 Apps ~
  - satellite, cable, and terrestrial broadcasting,
  - digital networks, and
  - digital VCR

# MPEG compression is based on 8 x 8 pixel **block processing**



8 pixels

8 pixels

- 8 x 8 pixel block can be numerically manipulated by fast signal processor in real time
- Motion estimation is based on comparing the blocks between series of pictures



# MPEG: only compress moving parts

new picture



previous

picture



difference



Encoder

Decoder

difference



previous picture

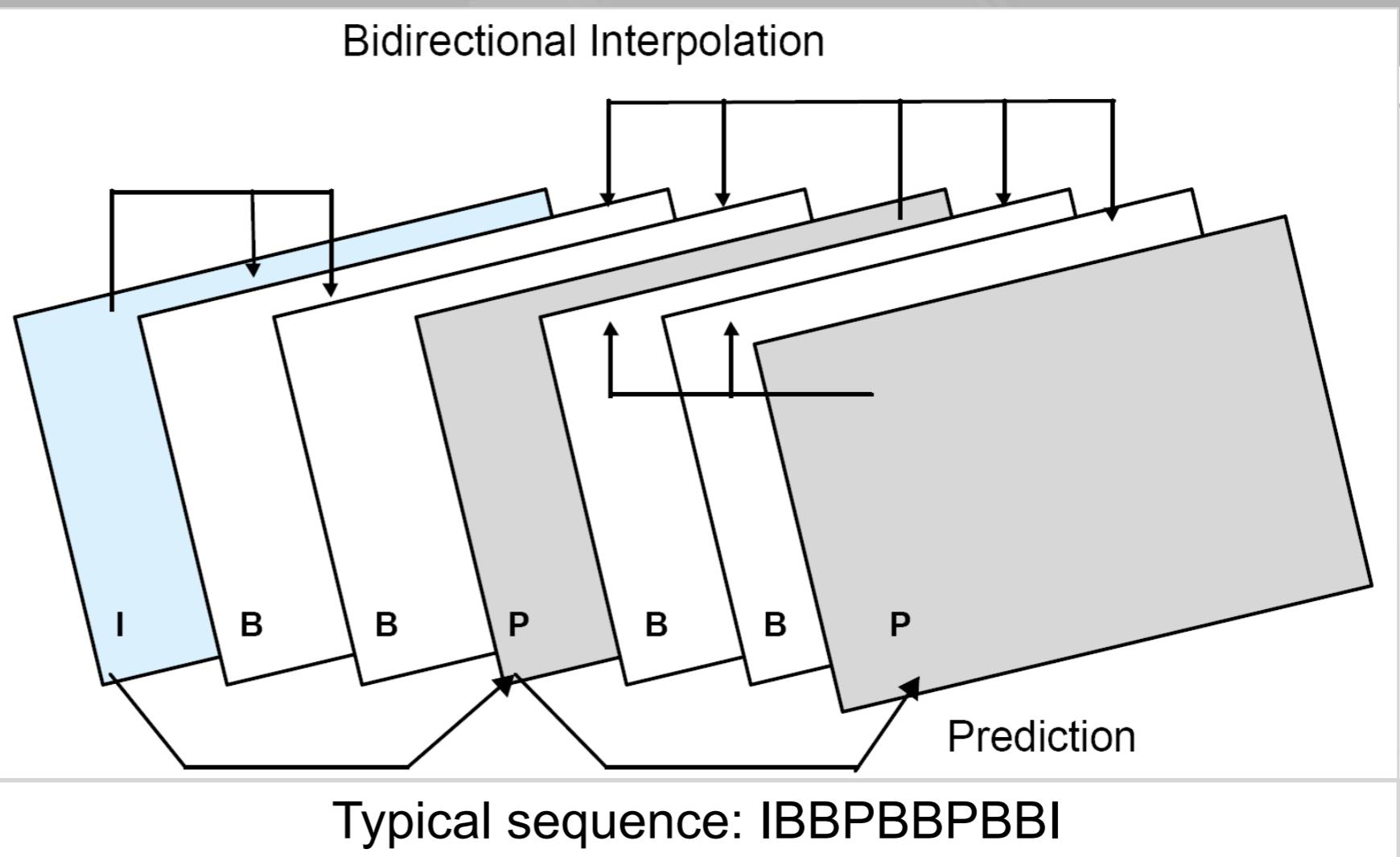


new picture



# MPEG: motion compression

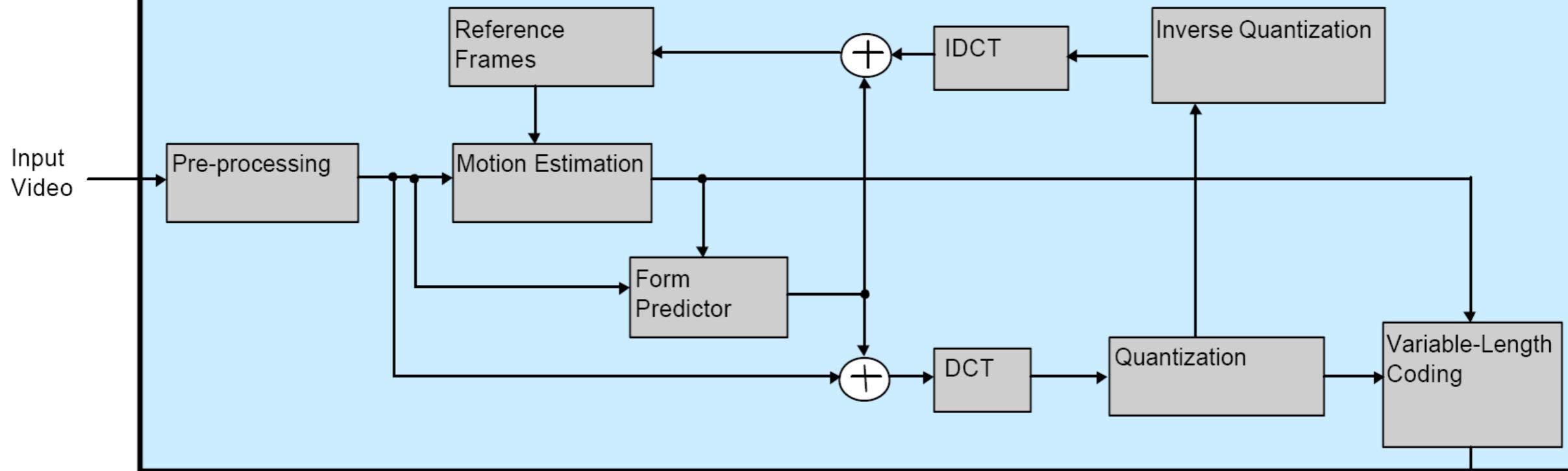
I = Intra-Frame  
P = Predicted frame  
B = Bi-directionally interpolated frame



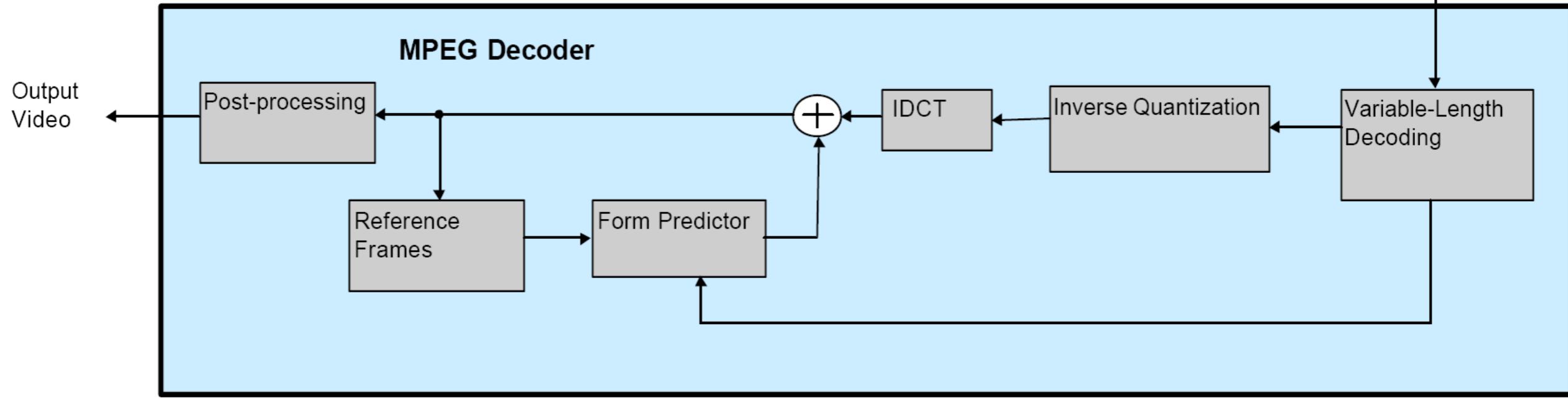
**Video signal: stream of picture, it is not necessary to send every picture**

- Whole picture is needed only when all the content is changed!
- Several pictures has to be buffered to memory to make prediction forward and backward

## MPEG Encoder



## MPEG Decoder



# MPEG: other issues

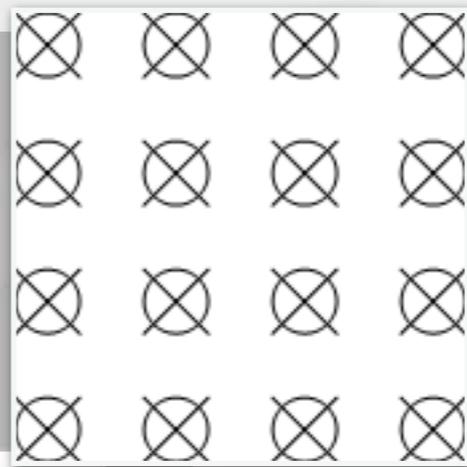
- Motion compensating
- Intra-frame transfer order

# Color video coding

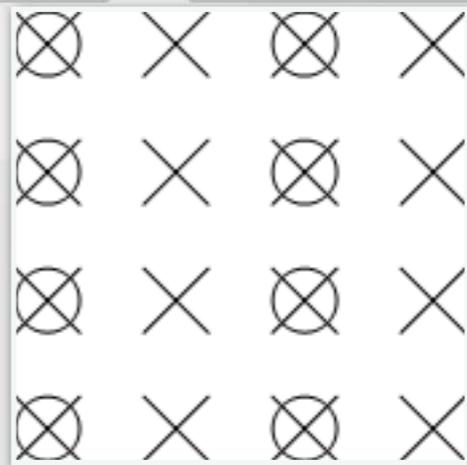
- 抽样和编码整个模拟（彩色）视频信号
  - 例如，复合编码
- 对亮度和色度分别编码
  - 例如，组件编码
  - 亮度比色度更重要，可根据应用场合采用4:2:2, 4:2:0, 4:4:4等不同的编码比率

# YUV 420等格式的说明

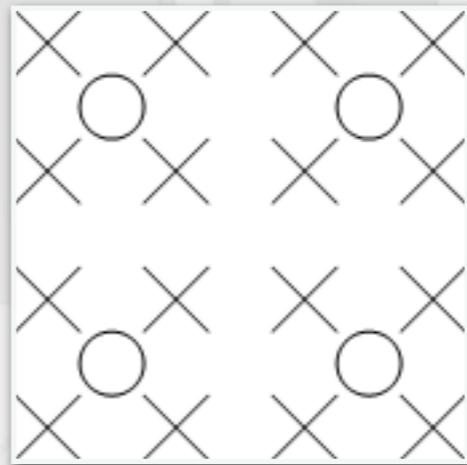
- YUV 的优点是色度通道的采样率可比 Y 亮度通道的低，同时不会明显降低视觉质量
- A:B:C表示法可用来描述 YUV 的采样频率比例
  - 4:4:4 每像素32位 表示色度频道没有下采样
  - 4:2:2 每像素16位 表示 2:1 的水平下采样，没有垂直下采样。对于每两个 U 样例或 V 样例，每个扫描行都包含四个 Y 样例
  - 4:2:0 每像素16位或12位 表示 2:1 的水平下采样， 2:1 的垂直下采样
  - 4:1:1 表示 4:1 的水平下采样，没有垂直下采样。对于每个 U 样例或 V 样例，每个扫描行都包含四个 Y 样例，不太常用



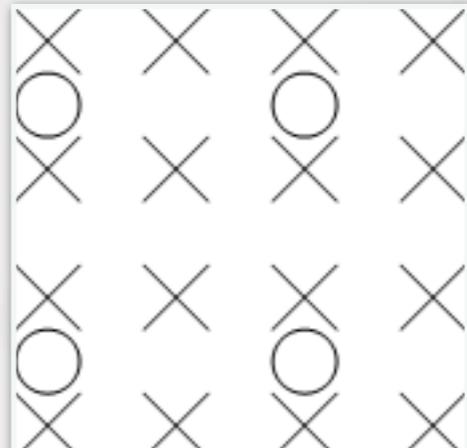
**YUV444**



**YUV422**

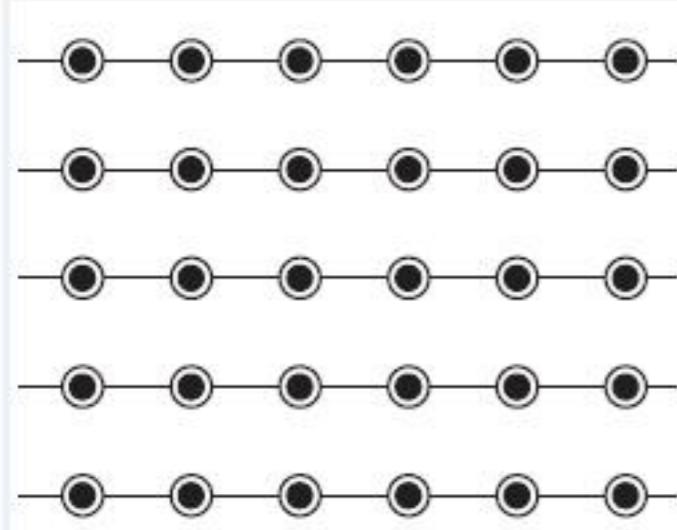


**YUV420  
MPEG-1**

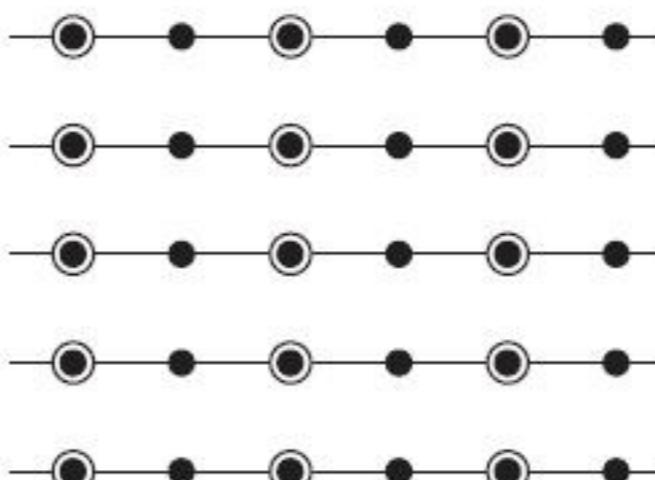


**YUV420  
MPEG-2**

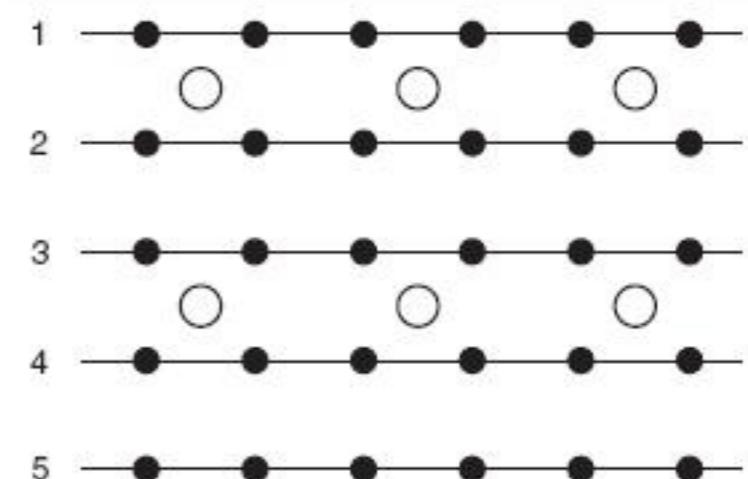
YUV4:4:4采样



YUV4:2:2采样



YUV4:2:0采样



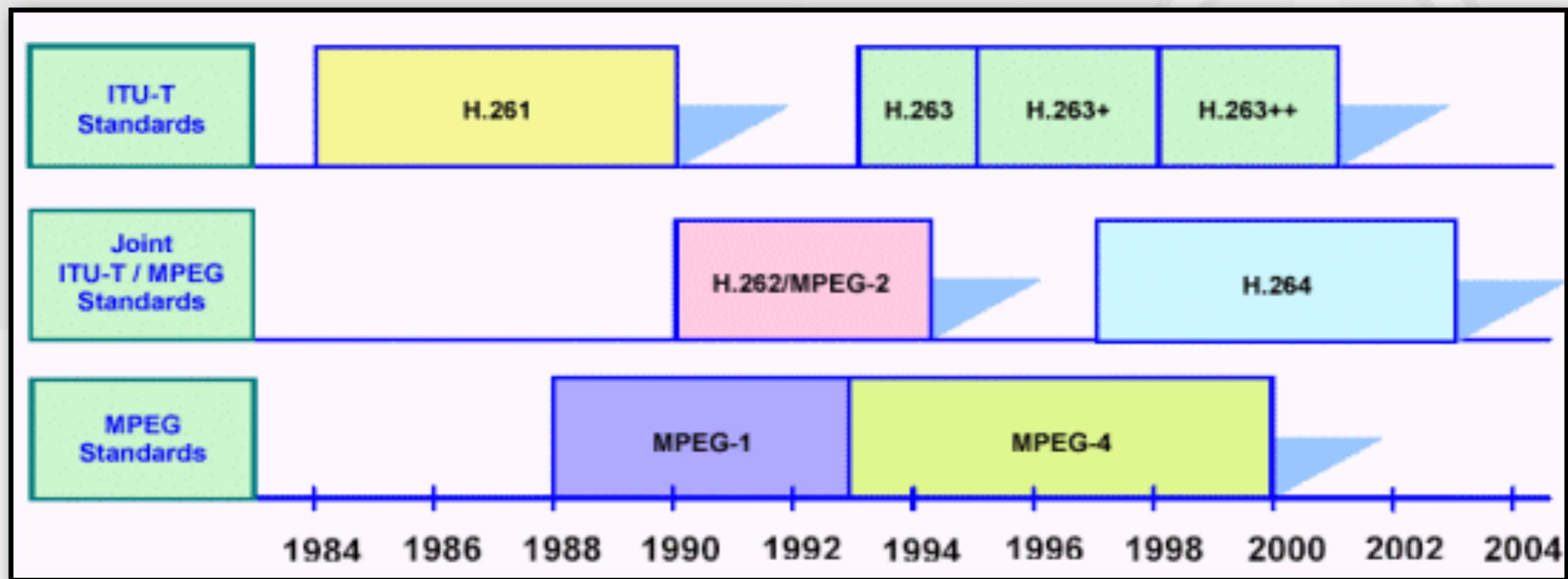
黑点表示采样该像素点的Y分量，以空心圆圈表示采用该像素点的UV分量

# Overview of H.264

- JVT (Joint Video Team)
  - founded on December 2001, Pattaya Thailand.
  - video coding specialists from ITU-T and ISO, the two international standards organizations
  - **goal:** define a new video coding standards to achieve high compression rate, high image quality, good network adaptive coding frame.
- H.264: A new video compression standard
  - accepted by ITU-T
  - accepted by ISO
    - called AVC (Advanced Video Coding) standard
    - as the 10th part of MPEG-4

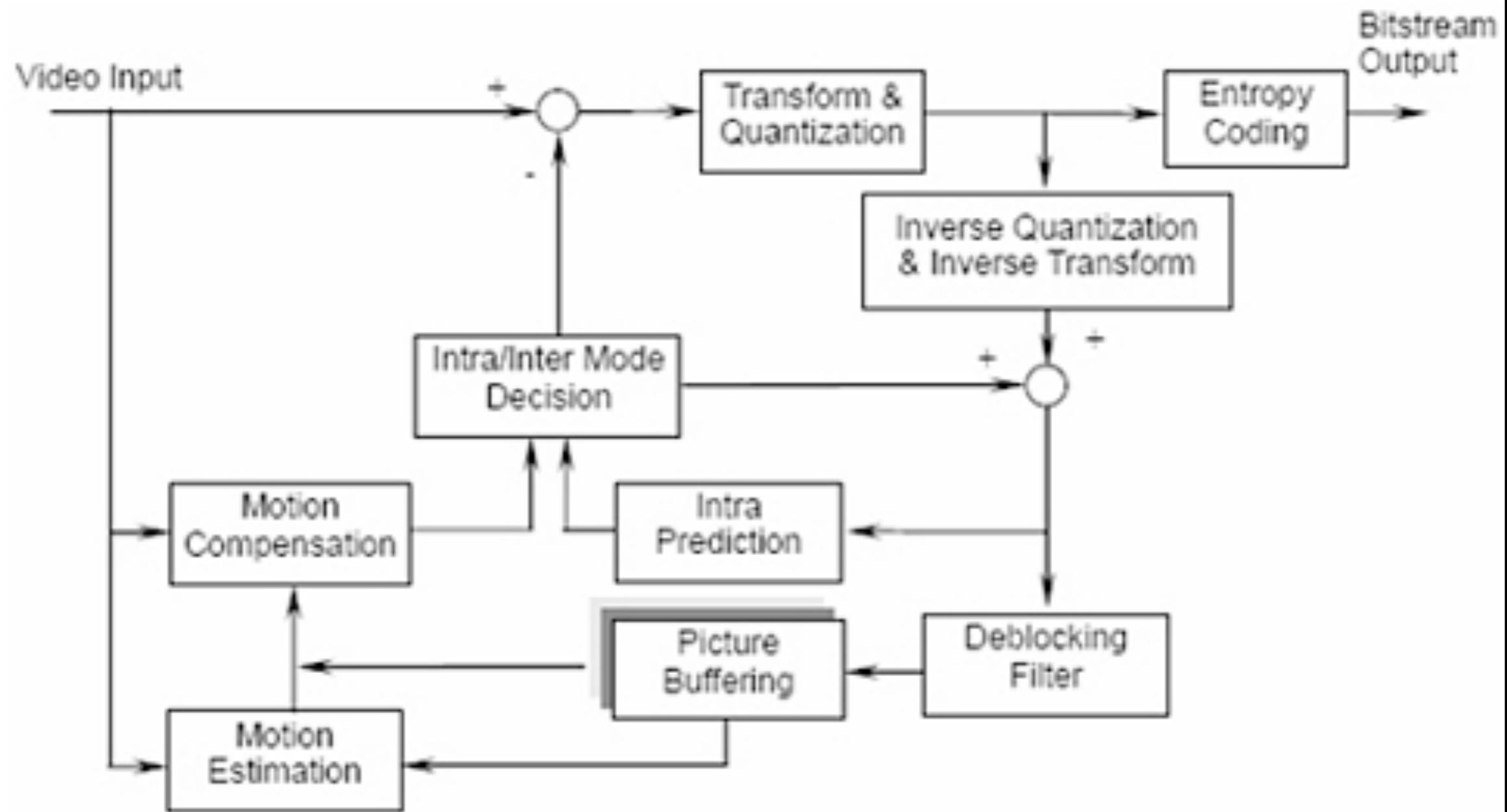


# Major history of digital video standard



# H.264 coding principle

ASTRI H.264 Baseline Profile Encoder Function Diagram



# H.264的主要技术特点

1. 4类DCT整数变换以及相应的量化方法
2. 7种宏块预测模式
  - $16 \times 16, 16 \times 8, 8 \times 16, 8 \times 8, 8 \times 4, 4 \times 8, 4 \times 4$
  - 运动估计和补偿更加精确
3. 多参考帧
4. 帧内预测
5. 改进的去块效应滤波器（Deblocking filter）
6. 增强的熵编码方法
  - UVLC (Universal VLC) 、CAVLC (Context adaptive VLC) 和 CABAC
7. 1/4像素插值
8. 宏块级逐行、隔行自适应编码MBAFF

# Advantages and shortages of H.264

## High compression rate

- In the same image quality, H.264 can be compressed as size of
  - 36% of MPEG-2, 61% of MPEG-4 , 51% of H.263
- Low bit stream, high quality

## High error correctness rate

- H.264 provides necessary tools to solve the error coding problem in unstable network environments

## Network adaptation

- H.264 provides Network Adaptation Layer so as to make files of H.264c can be easily transferred in different network environments.

## High computation price

- In the same image quality, H.264 is twice of MPEG-2 in computation complexity.



# Applications of H.264

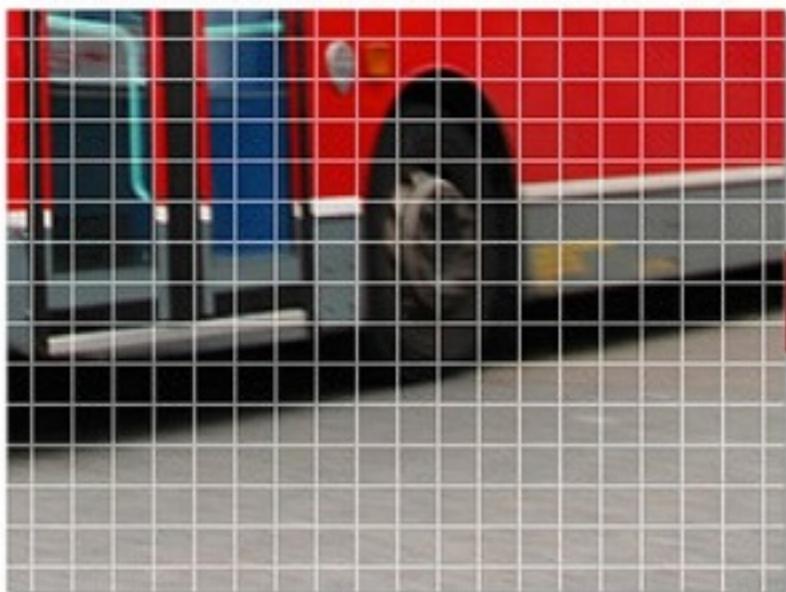
- H.264 standards added a NAL (Network Abstraction Layer)
  - to face the network connection and interface problem in the real applications.
- **video communication**
  - In real-time communication, POLYCOM、TANDBERG、VCON、SONY etc. claimed their own H.264 based TV-meeting products.
- **digital TV broadcasting**
  - MPEG has already finished defining the MPEG-2 compatible standard on H.264 stream coding content
- **video storage-and-play-back**
  - For High resolution DVD (HD DVD) application, H.264/MPEG-4 AVC solution.

# H.265

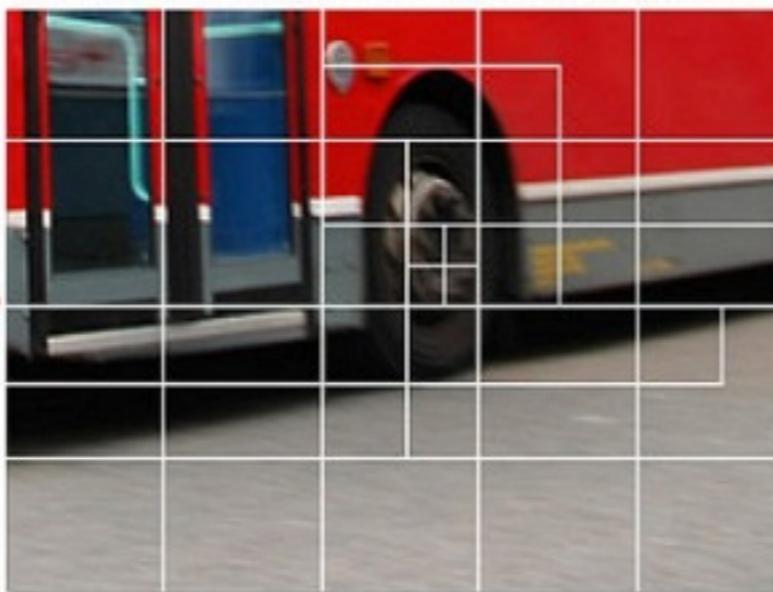
- High Efficiency Video Coding (HEVC), also known as H.265 and MPEG-H Part 2
- 应用于：
  - 高清晰度(Higher Definition)：数字视频的应用格式从720 P向1080 P全面升级，在一些视频应用领域甚至出现了4K x 2K、8K x 4K的数字视频格式；
  - 高帧率(Higher frame rate )：数字视频帧率从30 fps向60fps、120fps甚至240fps的应用场景升级；
  - 高压缩率(Higher Compression rate )：传输带宽和存储空间一直是视频应用中最为关键的资源，因此，在有限的空间和管道中获得最佳的视频体验一直是用户的不懈追求。

# H.265

[Conventional compression format]



[Newly developed compression format]



H.264(JM 16.2), 4112kbps, 33.19dB    HEVC(HM3.0), 1927kbps, 33.14dB



# Summary of video coding

- Resolution
- Coding rate
- Motion coding
- Transfer performance

# VR + image/video/audio

- Image
  - stereo ~
  - 360 Panorama
  - 360 + stereo
- Video
  - stereo ~
  - 360 Panorama
  - 360 + stereo
- Audio
  - stereo
  - surrounded ~



01:42 / 03:09

音量

1080P



# 车猫网 360度立体视频

[http://v.youku.com/v\\_show/id\\_XMTY2NTczOTY1Mg==.html?from=s1.8-1-1.2&spm=a2h0k.8191407.0.0](http://v.youku.com/v_show/id_XMTY2NTczOTY1Mg==.html?from=s1.8-1-1.2&spm=a2h0k.8191407.0.0)



## 2.4. HTML and XML

结构化文档概览



# Overview of HTML

- Hypertext Markup Language
  - Developed by **Tim Berners-Lee**
    - **lightweight** markup language vs. complex **SGML**.
  - Based on pure text format
- Rich abilities to display multimedia information.
  - Later added tags to support image and videos.
- **HTML 3.2 => HTML 4.0 => HTML 5.0**
  - Different browser has their own display effects.



# Overview of all HTML elements

Reference: <http://htmlhelp.com/reference/wilbur/overview.html>

**Head**

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"  
"http://www.w3.org/TR/html4/loose.dtd">  
<html>  
<head>  
    <title>Apple中国</title>  
    <meta http-equiv="content-type" content="text/html;  
charset=gb2312">  
    ...  
</head>  
  
<body>  
<!-- Tag for Activity Group: General, Activity: Apple China -<br/>Homepage -->  
...  
</body>  
</html>
```

**Body**

# Overview of HTML - Head elements

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
    <title>Apple中国</title>
    <meta http-equiv="content-type" content="text/html; charset=gb2312">
    ...
    <link rel="home" href="http://www.apple.com/">
    ...
    <script src="http://images.apple.com/global/scripts/lib/prototype.js" type="text/javascript" charset="utf-8">
    </script>
    ...

    <style type="text/css" media="all">
    ...
    #billboard { width: 1100px; margin: 0 auto 15px; overflow: hidden; position: relative; }
    #ticker { margin-bottom: 15px; }
    #homefooter { margin: 60px auto 50px; }
    ...
    </style>
</head>
```

- **TITLE** - Document title
- **ISINDEX** - Primitive search
- **META** - Meta-information

- **LINK** - Site structure
- **BASE** - Document location
- **SCRIPT** - Inline script
- **STYLE** - Style information



# Overview of HTML - Body elements

```
<html>
<head> ... </head>
<body>
    <H1> Hello, world </H1>
    <P> Digital Asset management is cool! </P>

</body>
</html>
```

- **Block level elements**
  - Headings: H1 => H6
  - Lists: UL, OL, DIR, MENU, LI, DL, DT, DD
  - Text Containers: P, PRE, BLOCKQUOTE, ADDRESS
  - others: DIV, CENTER, FORM, HR, TABLE



# Overview of HTML - Body elements

```
<html>
<head> ... </head>
<body>
    <H1> Text-level elements </H1>
    <A href="http://www.google.com"> GOOGLE <IMG src=" ... "> </A>

</body>
</html>
```

- **Text-level elements**
  - Logical markup: **EM** ...
  - Special markup: **A**, **IMG**, **APPLET** ...
  - Physical markup: **B**, ...
  - Forms: **INPUT** ...
  - Tables: **CAPTION**, **TR**, **TH**, **TD**



# About CSS

## 叠样式表 (Cascading Style Sheets)

- [http://www.w3school.com.cn/css/css\\_intro.asp](http://www.w3school.com.cn/css/css_intro.asp)
- 样式定义如何显示 HTML 元素
- 样式通常存储在样式表中 (\*.css)
- 把样式添加到 HTML 4.0 中，是为了解决内容与表现分离的问题
- 外部样式表可以极大提高工作效率
- 外部样式表通常存储在 CSS 文件中
- 多个样式定义可层叠为一个



# HTML 5.0

- 学习参考：<http://www.w3school.com.cn/html5/index.asp>
- 实例解释：<http://directguo.com/html5>
  - <http://html5-slide-template.googlecode.com/svn/trunk/html5-slide-template.html#slide1>
- 一套Web富客户端开发的工业标准
  - 许多新特性：内建的视频、音频标记，元素拖放功能
  - 最新的 Safari、Chrome、Firefox 以及 Opera 支持某些 HTML5 特性，Internet Explorer 也已经逐步支持



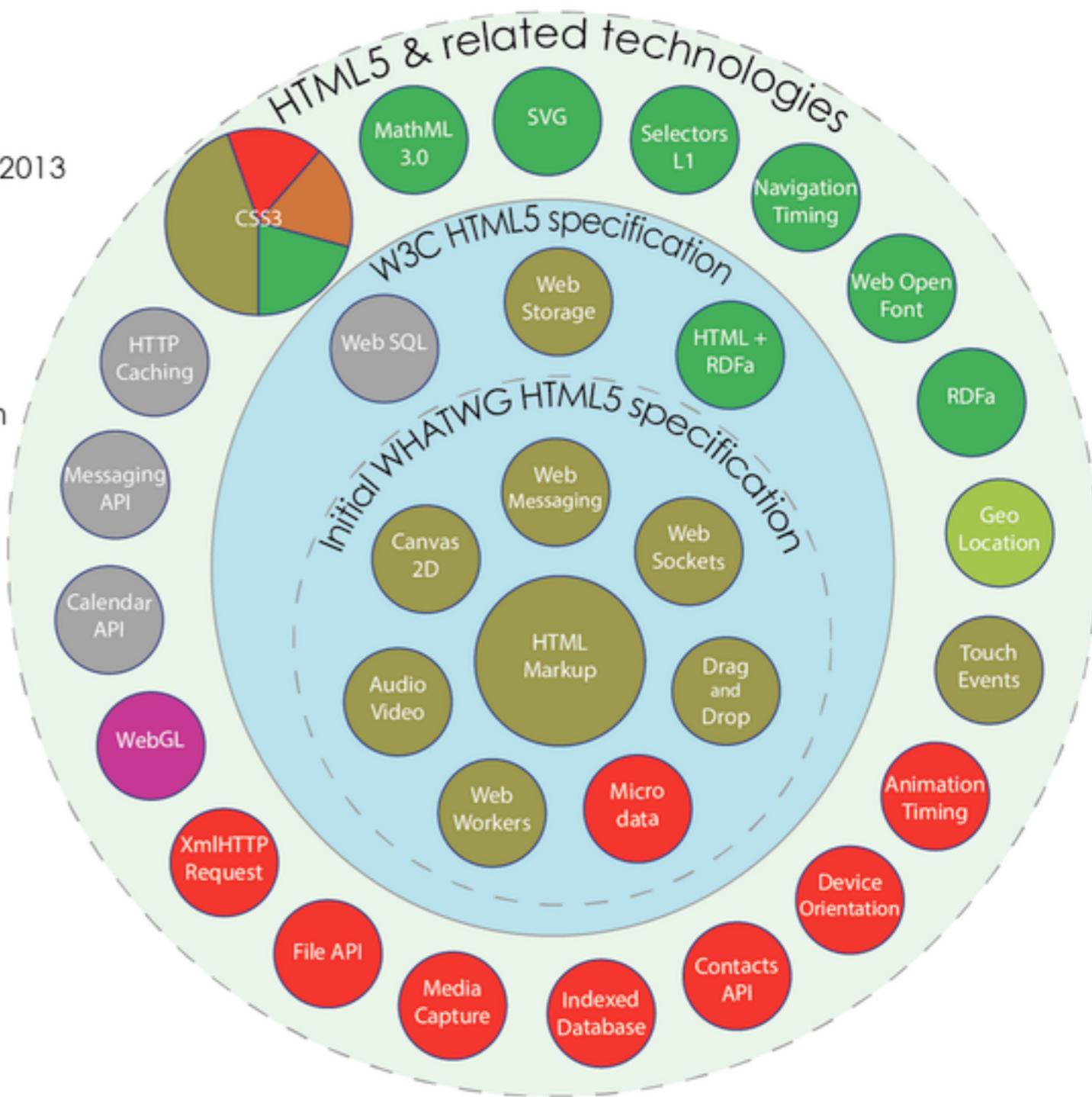
# Interesting demos

- [http://www.cnblogs.com/chu888chu888/  
archive/2012/02/29/2374841.html](http://www.cnblogs.com/chu888chu888/archive/2012/02/29/2374841.html)
- html5 Slides
- Impress (<http://impress.github.io/impress.js>)

# HTML5

Taxonomy & Status on January 20, 2013

- W3C Recommendation
- Proposed Recommendation
- Candidate Recommendation
- Last Call
- Working Draft
- Non-W3C Specifications
- Deprecated



by Sergey Mavrody CC BY · SA

# About JavaScript

- <http://www.cad.zju.edu.cn/home/zhx/DAM/2015/doku.php?id=js>
- JavaScript

# About Python

- <http://www.cad.zju.edu.cn/home/zhx/DAM/2015/doku.php?id=python>
- 网页服务：
  - Flask ...
- 科学计算：
  - numpy and scipy ...
- 图像处理：
  - Image and opencv ...
- 还有很多其他库：
  - virtualenv 构建虚拟环境
  - pip 安装python扩展包
  - ...

# Alternative solutions

- LaTeX
- Markdown
- ...

# 举个栗子

- HTML逆向解析——我的实验微信公众号：



- 欢迎关注、实验