

# **MPEG-4**

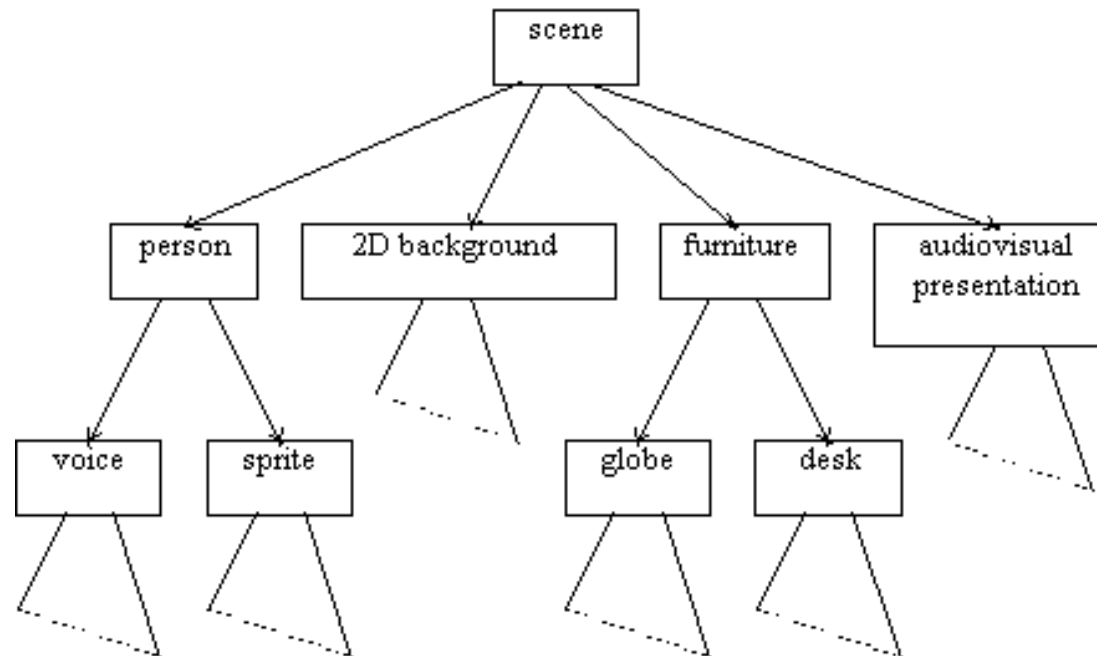
**Profa. Débora Christina Muchaluat Saade**  
**debora@midia.com.uff.br**

# MPEG-4

- **Objetos de mídia podem ser de origem natural ou sintética**
  - *Áudio (voz ou música), vídeo, objetos 2D e 3D*
- **Composições de objetos formam cenas audiovisuais**
- **Interatividade com o usuário**
- **Transmissão sobre qualquer tipo de rede**

# BIFS

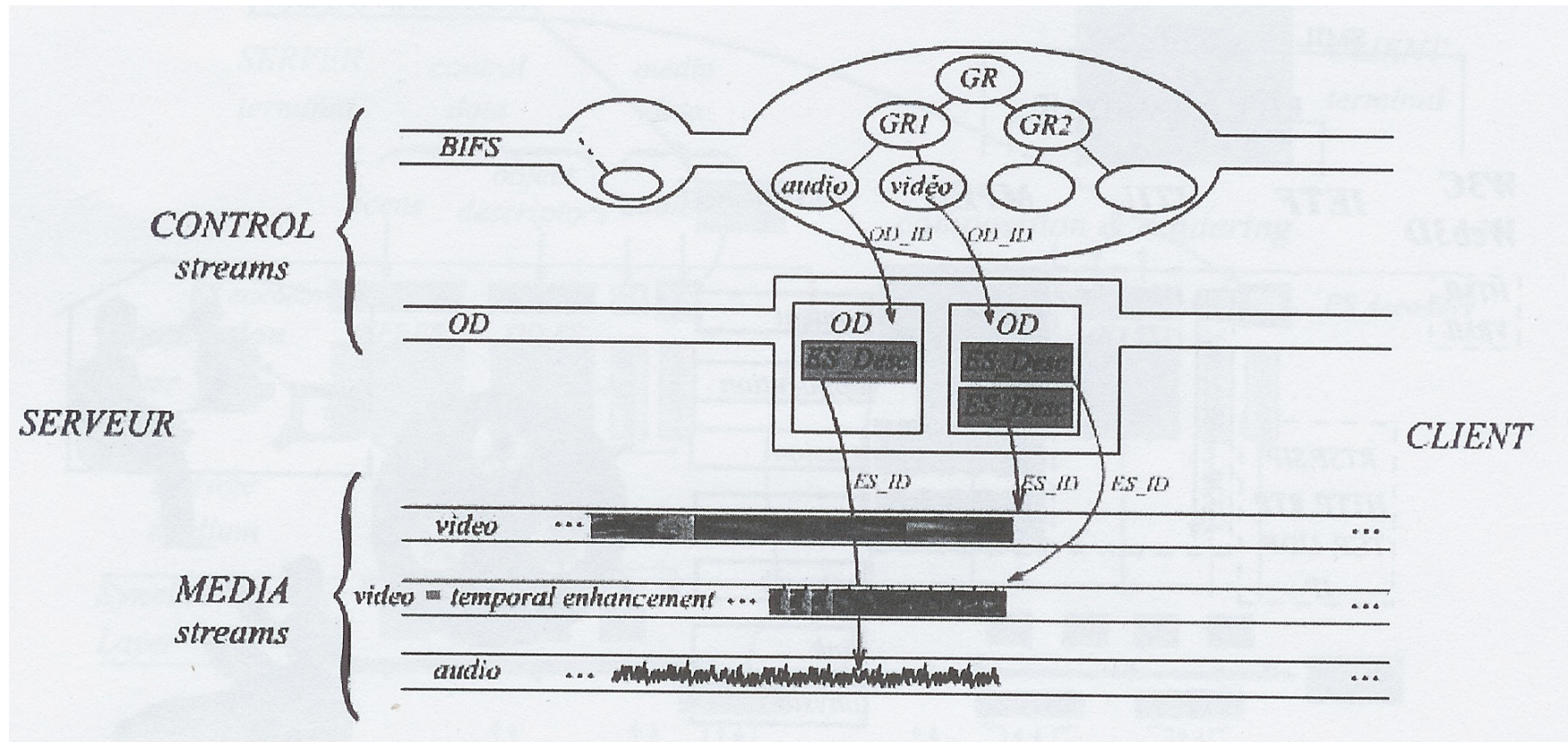
- **Binary Format for Scenes**
  - *Linguagem binária usada para descrição de cenas*
  - *Derivada da VRML (Virtual Reality Modeling Language) – padrão ISO*



# MPEG-4

- **Cada objeto tem um descritor**
  - *Object descriptor*
    - Fornece as informações sobre o objeto (tipo de codificação, ...)
- **Fluxo MPEG4 possui vários fluxos elementares:**
  - *Fluxos de mídia*
    - Cada objeto tem os seus fluxos (1 ou mais)
  - *Fluxos de controle*
    - Descrição em BIFS
    - Descritores de objetos

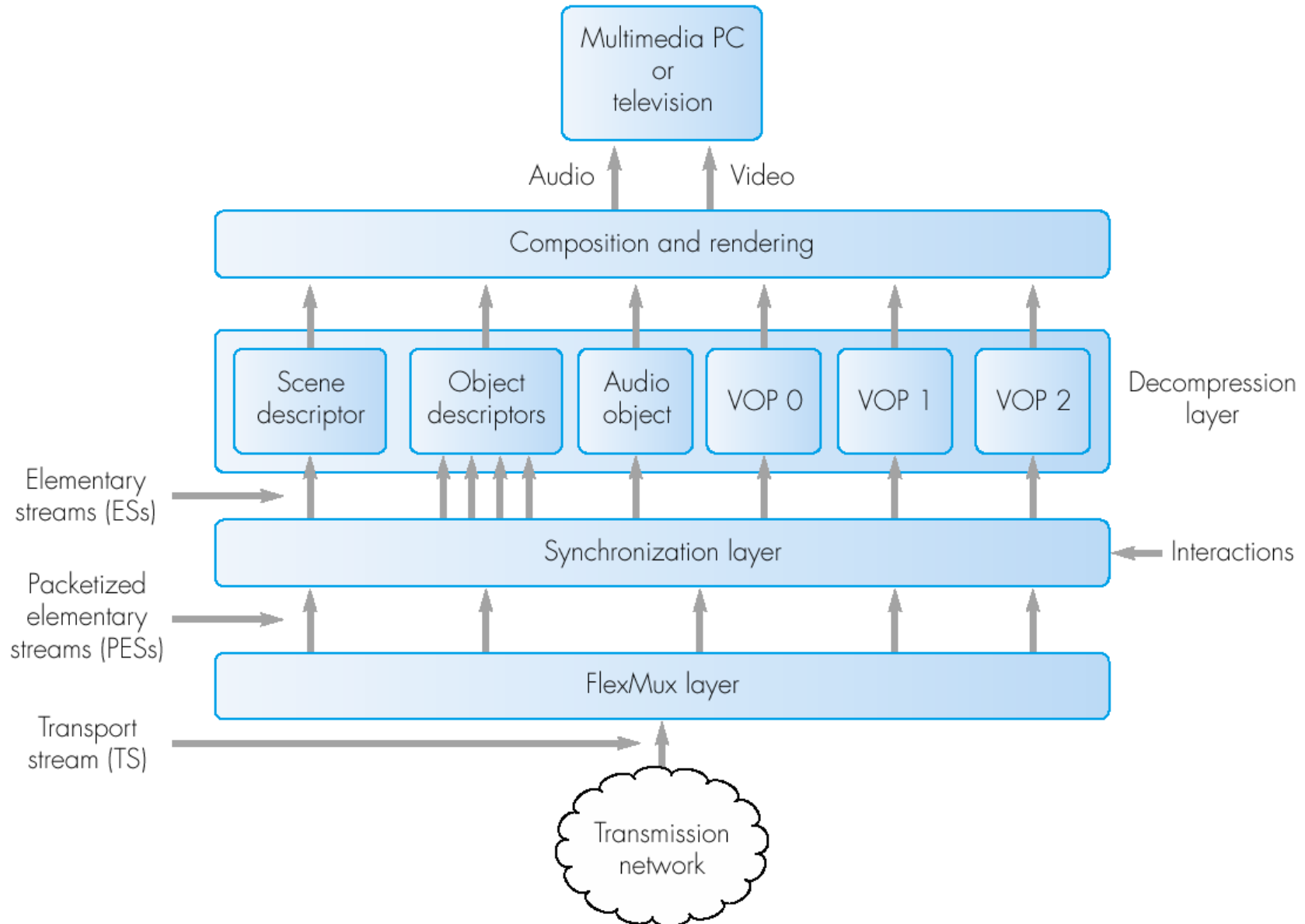
# Integração entre os fluxos elementares



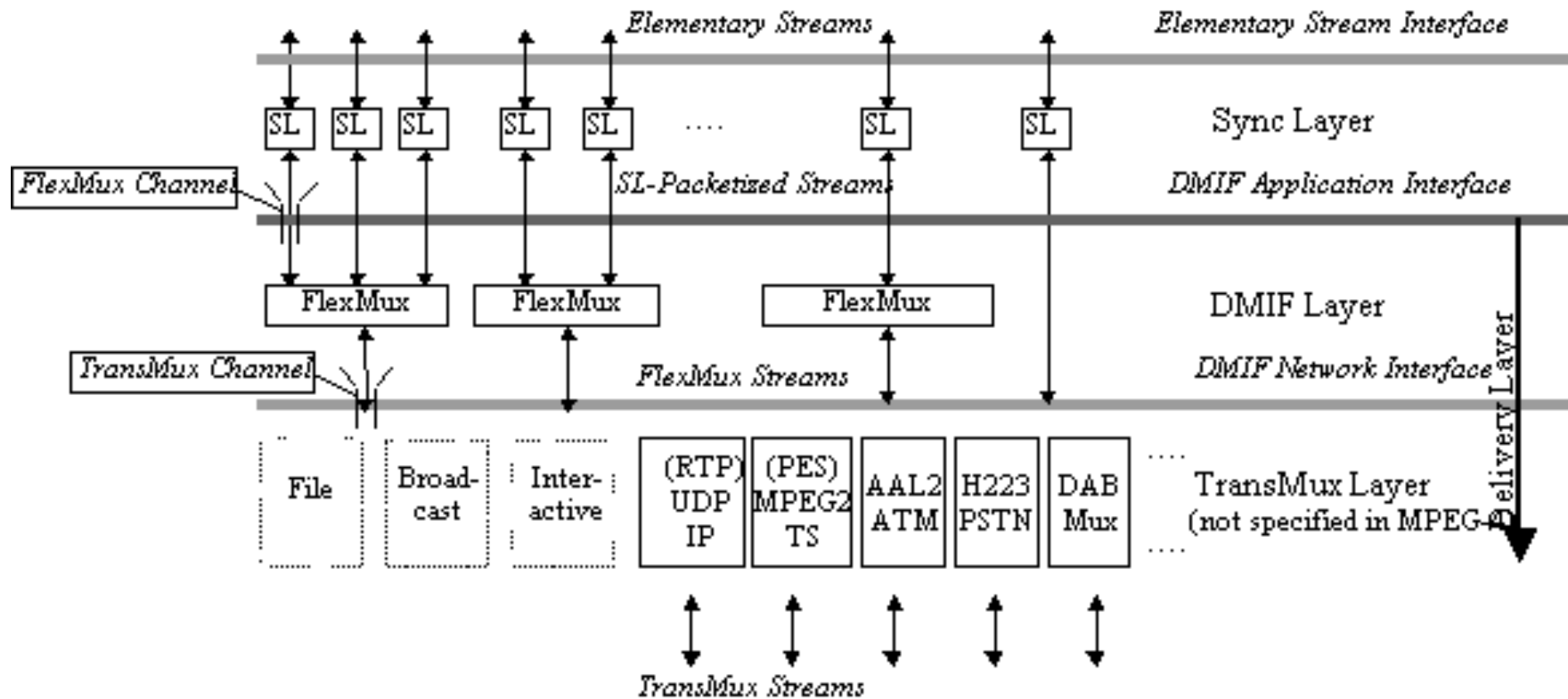
# MPEG-4 Systems

- Fluxo MPEG-4 é transportado na rede como um fluxo de transporte (*transport stream - TS*)
  - *Consistindo de um fluxo multiplexando vários fluxos elementares empacotados (packetized elementary streams - PES)*
- A informação de cada objeto audiovisual é chamada de fluxo elementar (*elementary stream - ES*)
  - *Carregada no payload de um pacote PES*
- Descritores – *elementary stream descriptor* – ESD – são usados pela camada de sincronização para entregar o fluxo elementar ao decodificador apropriado.

# Fluxos MPEG-4



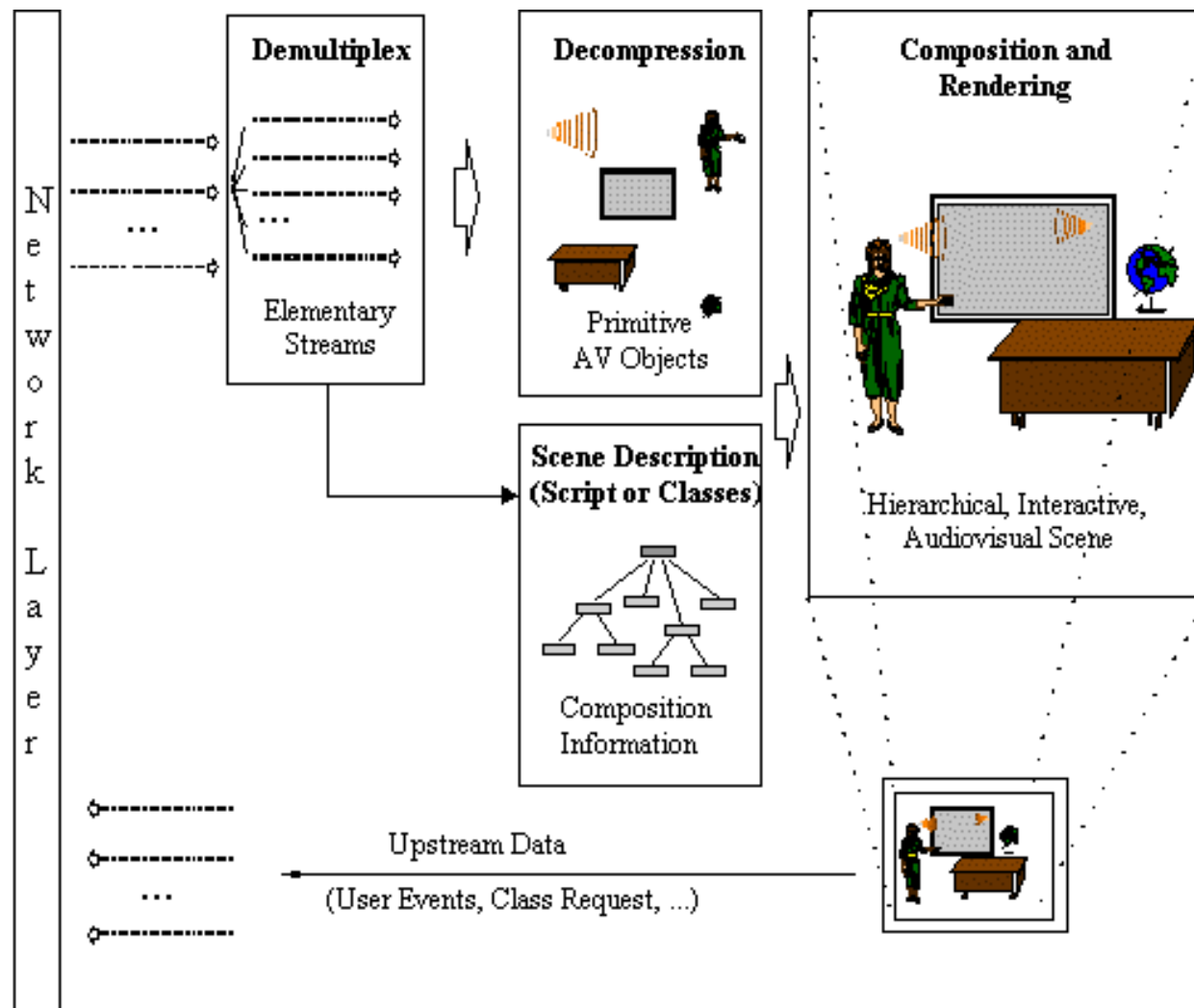
# MPEG-4 Systems/DMIF



**DMIF – Delivery Multimedia Integration Framework**



# Receptor MPEG-4



# MPEG-4 Vídeo

- **Dois padrões para codificação de vídeo MPEG-4**
  - *MPEG-4 Parte 2 – 1994 a 1999*
  - *MPEG-4/H.264 Parte 10 – 1999 a 2005*
    - Marpe, Wiegand, Sullivan. The H.264/MPEG4 Advanced Video Coding Standard and its Applications, IEEE Communications Magazine, Agosto de 2006.
    - Wiegand, Sullivan, Bjontegaard, Luthra. Overview of the H.264 Video Coding Standard, IEEE Transactions of Circuits and Systems for Video Technology, Vol.13, No.7, Julho de 2003.
    - Richardson. H.264 and MPEG-4 Video Compression – Video Coding for Next-generation Multimedia, John Wiley & Sons Inc., 2003.

# H.264/MPEG-4 Parte 10

- **AVC – Advanced Video Coding**
- **Esforço conjunto do grupos:**
  - *ITU-T Video Coding Experts Group (VCEG)*
  - *ISO/IEC Moving Pictures Expert Group (MPEG)*
- **Padrão inicial e conjunto de extensões FRExt (fidelity range extensions) - 2005**

# H.264/MPEG-4 AVC

- Principais novidades na codificação de vídeo em relação a padrões anteriores considerando estimativa e compensação de movimento:
  - *Mais flexibilidade no tamanho do bloco para realizar compensação de movimento variando de 16x16 a 4x4*
  - *Possibilidade de uso de um quarto de pixel (quarter sample) para realizar a compensação de movimento*
  - *Vetores de movimento podem ultrapassar os limites de um quadro (como no H.263)*
  - *Compensação de movimento usando múltiplos quadros de referência (para P e B)*

# H.264/MPEG-4 AVC

- Principais novidades na codificação de vídeo em relação a padrões anteriores considerando estimativa e compensação de movimento:
  - *Desacoplamento da ordem de referência da ordem de apresentação*
  - *Possibilidade de uso de quadros B como referência para codificação de outros quadros*
  - *Predição de quadros de maneira ponderada pelo codificador (cenas com fades)*
  - *Filtragem anti-bloco (deblocking filtering) – evita efeitos de visualização da imagem em blocos (blocking artifacts)*

# H.264/MPEG-4 AVC

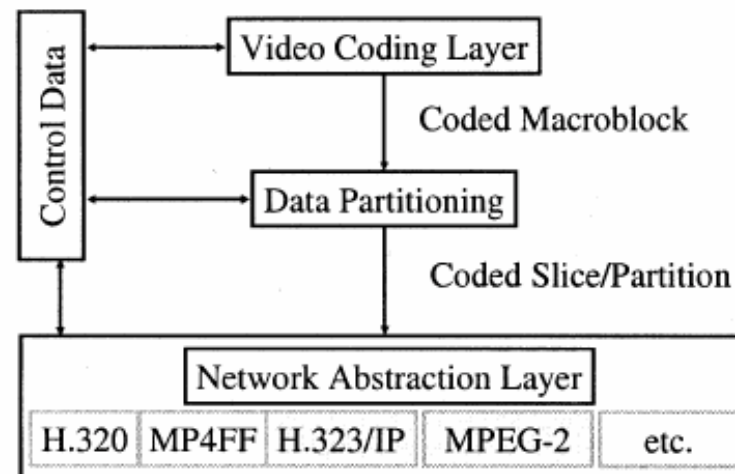
- **Principais novidades adicionais na codificação de vídeo em relação a padrões anteriores que ajudam a aumentar a eficiência na codificação:**
  - *Menor tamanho de bloco para aplicação da transformada (8x8 ou 4x4)*
  - *Transformada hierárquica*
  - *Transformada com palavras de tamanho pequeno – padrões anteriores usavam processamento de 32 bits, H.264/AVC usa aritmética de 16 bits*

# H.264/MPEG-4 AVC

- Principais novidades adicionais na codificação de vídeo em relação a padrões anteriores que ajudam a aumentar a eficiência na codificação:
  - *Transformada inversa exata, permitindo obter a mesma qualidade de decodificação em todos os decodificadores*
  - *Codificação por entropia usando codificação aritmética (CABAC – context-adaptive binary arithmetic coding)*
  - *Codificação por entropia adaptativa ao contexto: ambos os métodos CAVLC e CABAC utilizam codificação adaptativa ao contexto para melhorar o desempenho do codificador*

# H.264/MPEG-4 AVC

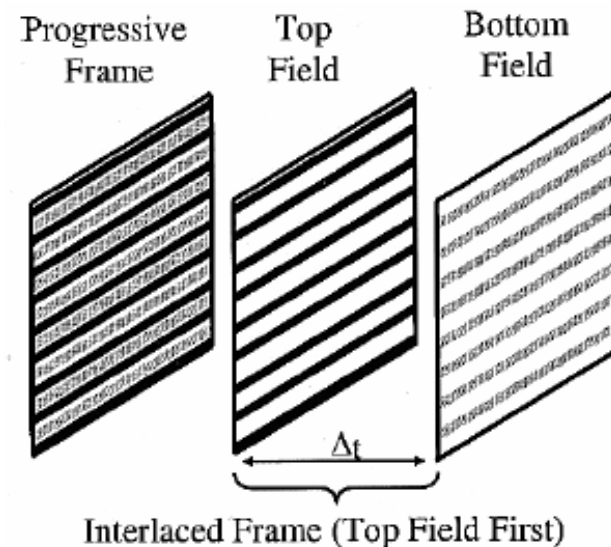
- **Estrutura do codificador de vídeo H.264/AVC**





# H.264/MPEG-4 AVC

- VCL (Video Coding Layer)
- Sequência de vídeo codificada
  - *Quadros codificados*
    - Quadro inteiro
      - *Quadro progressivo*
    - 1/2 quadro (campo)
      - *Quadro entrelaçado*

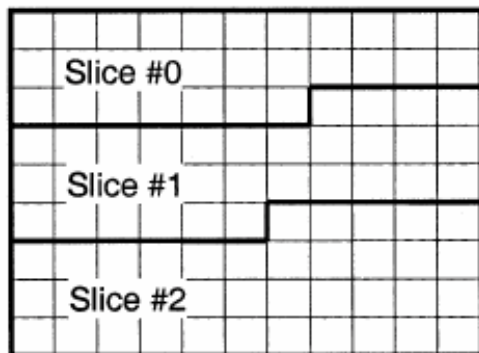


# H.264/MPEG-4 AVC

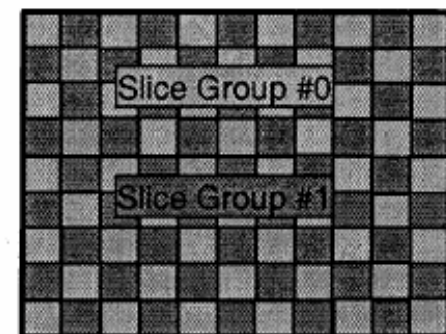
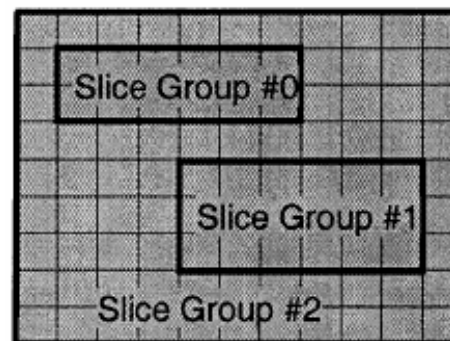
- **Representação de cor YCrCb**
- **Formato mais comum**
  - *4:2:0 e 8 bits por componente*
- **Pode chegar a**
  - *4:2:2 com 10 bits por componente*
- **Cada quadro é dividido em macroblocos de**
  - *16x16 para Y*
  - *8x8 para Cr e Cb*

# H.264/MPEG-4 AVC

- **FMO – Flexible macroblock ordering**
  - *Conceito de grupos de fatias*
    - Existe um mapa relacionando macroblocos com grupo de fatias
      - *Cada macrobloco contém um número de identificação de grupo de fatias*
- **ASO – Arbitrary Slice Ordering**
  - *slices podem ser enviados em qualquer ordem*



Sem FMO



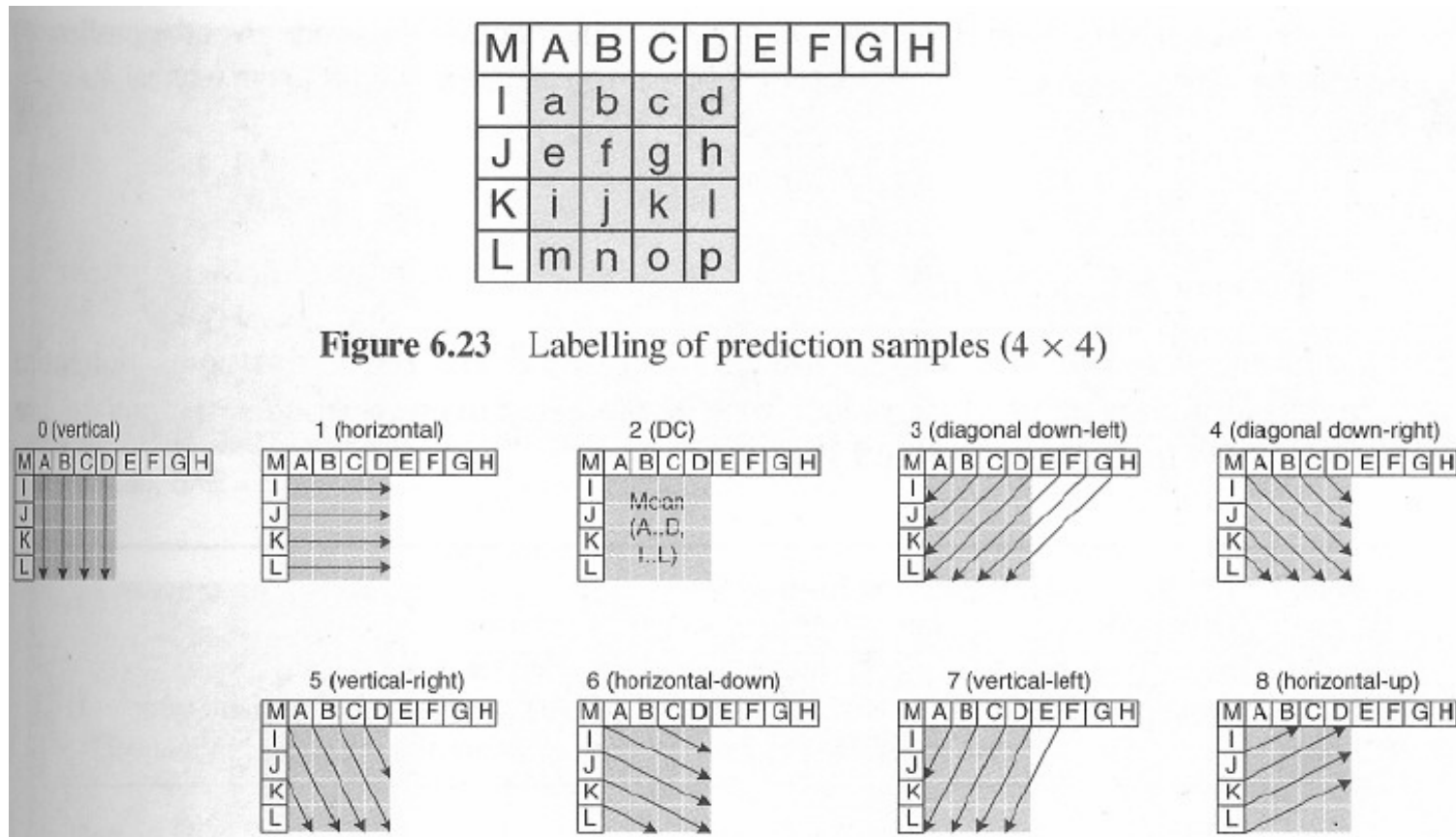
Com FMO (ex.: ROI e videoconferência)

# H.264/MPEG-4 AVC

- **Tipos de quadros:**
  - *Quadros I*
    - Todos os macroblocos em uma fatia (*slice*) podem ser preditos (codificação diferencial) por dados previamente codificados
    - Até em quadros I, amostras são preditas por outras na fatia corrente que já foram codificadas
    - Macroblocos podem variar de 16x16 a 4x4

# H.264/MPEG-4 AVC

- Predição de macroblocos I

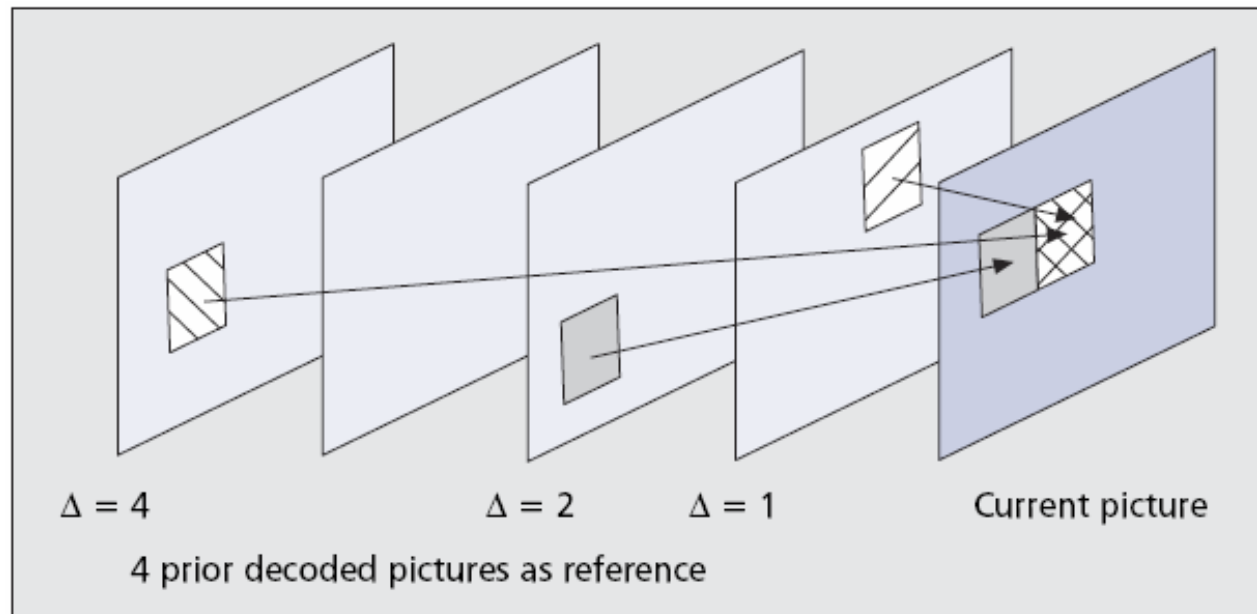


# H.264/MPEG-4 AVC

- **Tipos de quadros:**
  - *Compensação de Movimento*
    - Quadros P
      - *suportam múltiplos quadros de referência*
    - Quadros B
      - *suportam múltiplos quadros de referência*
      - *Também podem ser usados como quadros de referência*
  - *Diferença principal entre quadros P e B é que macroblocos de quadros B podem usar média ponderada de macroblocos de 2 listas distintas de quadros para calcular estimativa e compensação de movimento bidirecional*

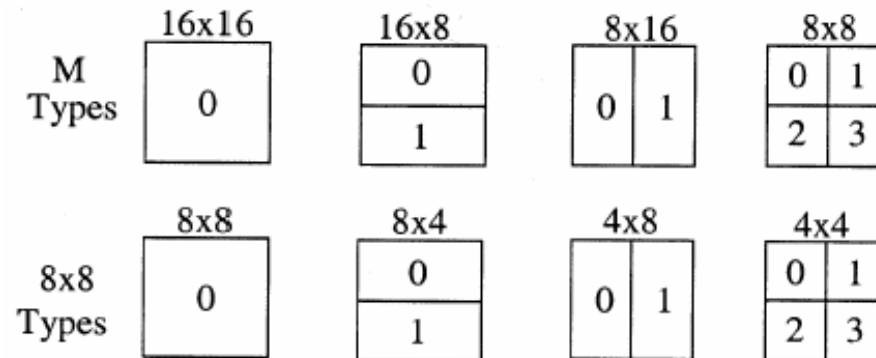
# H.264/MPEG-4 AVC

- **Múltiplos quadros de referência para fazer compensação e estimativa de movimento**



# H.264/MPEG-4 AVC

- **Predição Inter-quadro**
  - *Vários tipos de macroblocos para realizar compensação e estimativa de movimento*
    - 16x16, 16x8, 8x16, 8x8
  - *Se partição 8x8 foi escolhida, ela pode ser particionada novamente em 8x4, 4x8 e 4x4*



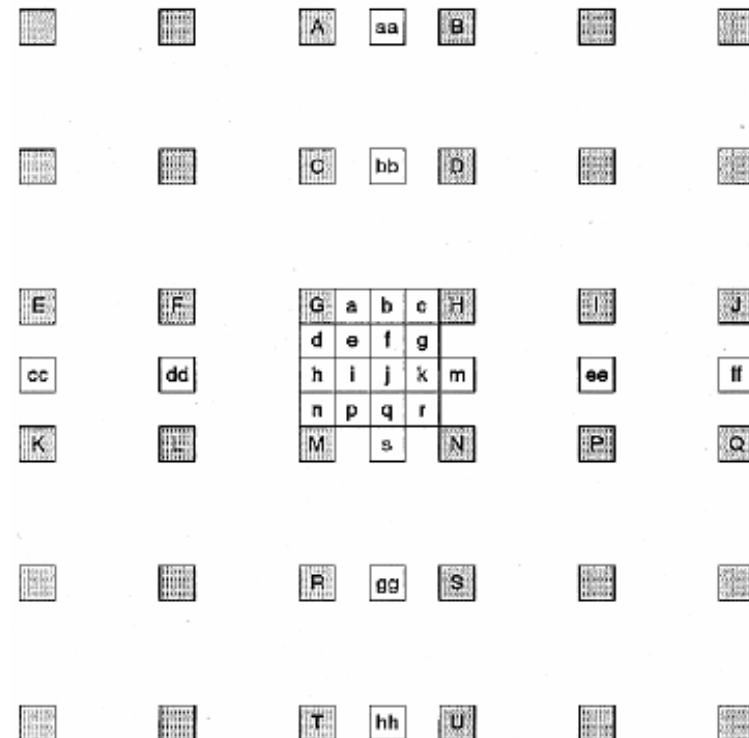


# H.264/MPEG-4 AVC

- **Compensação de movimento (Y)**
  - *precisão de  $\frac{1}{4}$  de pixel*
  - *$\frac{1}{4}$  pixel é calculado com a média do pixel e  $\frac{1}{2}$  pixel*

$$\frac{1}{2} \text{ pixel}$$
$$b = \text{round}((E - 5F + 20G + 20H - 5I + J) / 32)$$

$$\frac{1}{4} \text{ pixel}$$
$$a = \text{round}((G + b) / 2)$$



# H.264/MPEG-4 AVC

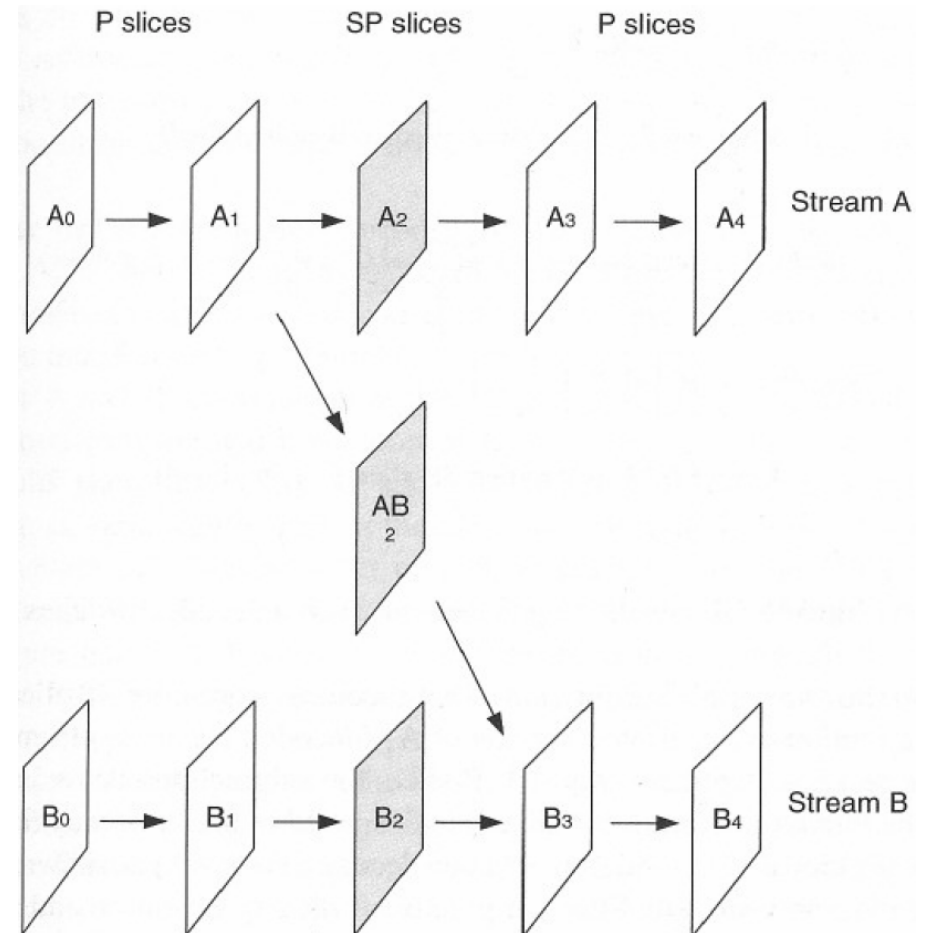
- **Tipos de quadros:**

- *SI (switching I) e SP (switching P) permitem*

- troca eficiente entre fluxos de vídeo

- *por exemplo, o mesmo vídeo codificado em diferentes taxas*

- acesso aleatório eficiente para decodificadores



# H.264/MPEG-4 AVC

- **Transformadas e Quantização**
  - *Ao invés de usar a DCT em blocos 8x8, como nos outros padrões*
  - *MPEG-4/H.264 usa transformadas inteiras de diferentes tamanhos de blocos (4x4)*

$$H = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 2 & 1 & -1 & -2 \\ 1 & -1 & -1 & 1 \\ 1 & -2 & 2 & -1 \end{bmatrix}$$

- *Transformada de Hadamard adicional é aplicada aos coeficientes DC das componentes de crominância (2x2)*

# H.264/MPEG-4 AVC

- **Codificação por Entropia**

- *Exponential Golomb Codes*

- valores de parâmetros que são mais frequentes usam palavras-código de menor tamanho (ex. Macrobloco tipo P 16x16 – code\_num=0)

**Formato:**

[M zeros][1][INFO]

onde INFO tem M bits

$$M = \text{floor}(\log_2[\text{code\_num} + 1])$$

$$\text{INFO} = \text{code\_num} + 1 - 2^M$$

**Decodificação:**

1. Leia M zeros seguidos de 1
2. Leia o campo INFO de M bits
3.  $\text{code\_num} = 2^M + \text{INFO} - 1$

Table 6.9 Exp-Golomb codewords

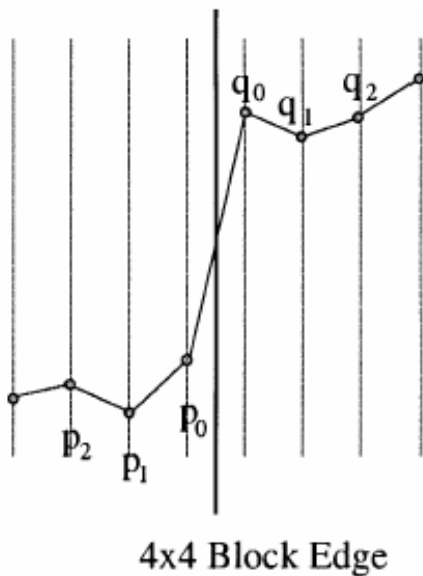
| code_num | Codeword |
|----------|----------|
| 0        | 1        |
| 1        | 010      |
| 2        | 011      |
| 3        | 00100    |
| 4        | 00101    |
| 5        | 00110    |
| 6        | 00111    |
| 7        | 0001000  |
| 8        | 0001001  |
| ...      | ...      |

# H.264/MPEG-4 AVC

- **Codificação por Entropia**
  - *Para transmitir os coeficientes das transformadas quantizados:*
    - CAVLC – *Context-Adaptive Variable Length Coding*
    - CABAC – *Context-Adaptive Binary Arithmetic Coding*
  - *CABAC fornece uma redução típica de 10% a 20% em relação ao CAVLC*

# H.264/MPEG-4 AVC

- *In-loop deblocking filter* (filtragem anti-bloco)
  - *Idéia básica: identificar “blocking artifact”*
    - se existe diferença absoluta grande entre amostras próximas ao limite de um bloco => efeito em blocos
    - *Filtro suaviza a diferença*



sem filtro



com filtro