

Compressão Video Digital

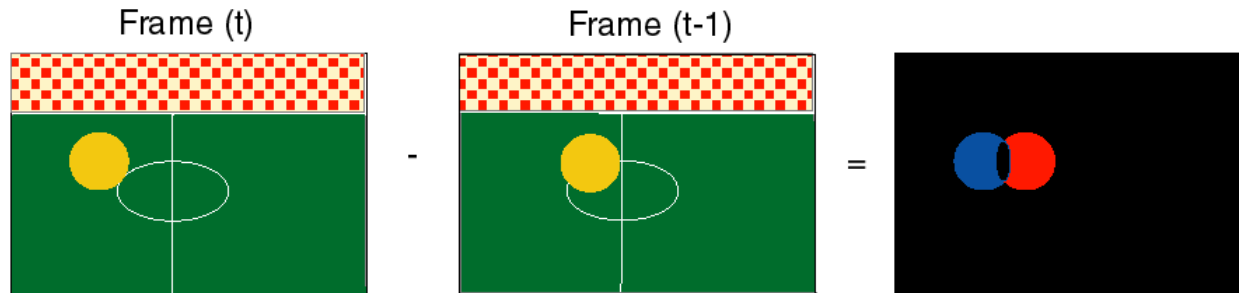
- Transmissão HDTV
 - 1920x1080 pixel
 - 30 frames/segundo
 - 8 bit/canal
 - 1,4 Gbit/seg.
 - Problemas no armazenamento e transmissão;
- Métodos de compressão sem perdas não são suficientes;
- Compressão remove redundâncias:
 - Temporal (redundância entre frames consecutivas)
 - Espacial, estrutural (ex: videotelefonia)
 - Estatística
 - Psicovisual (Sensibilidade do SVH à cor / luminância e objecto parado / movimento)

Compressão de video

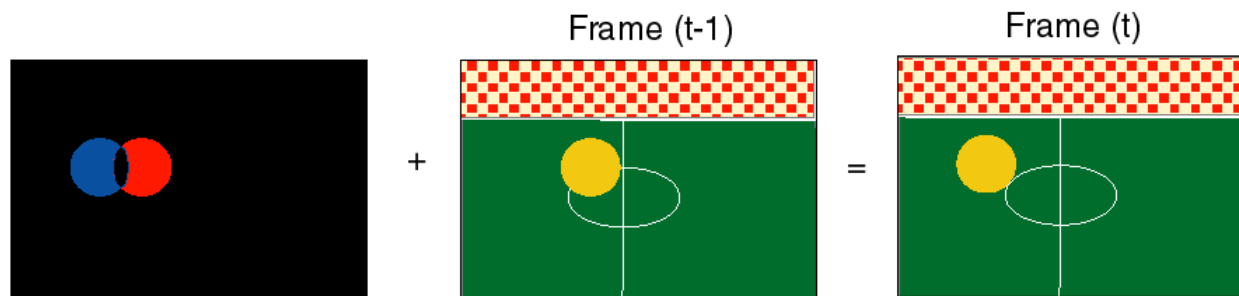
- Redução do Bit rate entre 30 e 50%
 - Conversão RGB – YCbCr
 - Chroma-subsampling (4:2:2, 4:1:1, 4:2:0)
- Codificação **intraframe**
 - Remoção de redundância espacial
(compressão semelhante ao JPEG – DCT, quantificação, VLC)
- Codificação **interframe**
 - Remoção da redundância temporal
 - Diferença entre frames
 - Estimação / Compensação de movimento
 - Numa sequência uma frame é muito semelhante às frames anteriores e posteriores, as diferenças existentes são devidas ao movimento de objectos ou da câmara ou então a uma mudança de cena
(frame rate > 24 frames/seg.)

Codificação Interframe (sem compensação de movimento)

- É codificada a diferença entre frames.
- Codificador



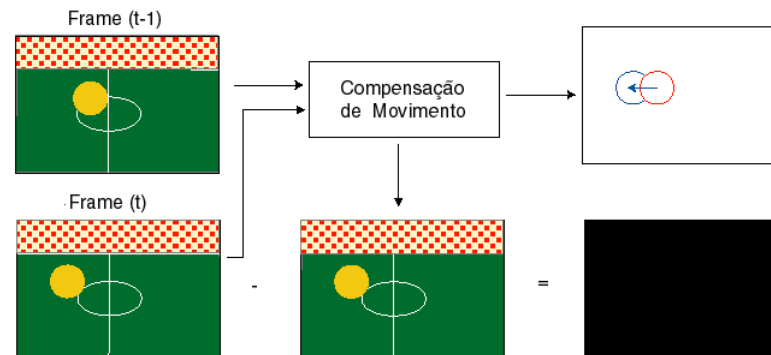
- Decodificador



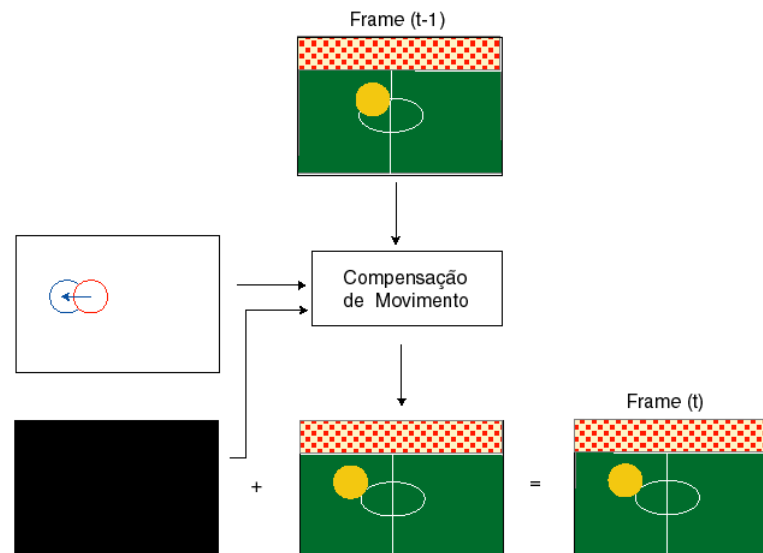
Codificação Interframe com Compensação de Movimento

- Estima-se qual o movimento;
- É codificado o vector de movimento e a diferença entre a frame no instante actual e a frame estimada para o mesmo instante;

- Codificador:



- Decodificador:



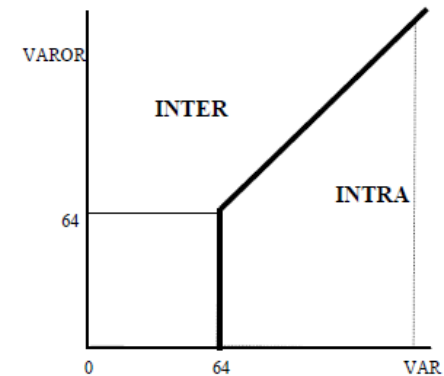
Compensação de movimento

- Divide-se em:
 - Estimar o movimento (determinar o vector de movimento)
 - Compensar o movimento
 - Determinar o erro entre a frame e a predição actual
- Vantagem:
 - Permite reduzir o bitrate (se a frame estimação for muito boa o erro a transmitir é quase nulo e com baixa entropia logo a sua codificação é mais eficiente)
 - Compensação de movimento reduz o bit rate em mais de 30% quando comparado com a codificação baseada apenas na diferença entre frames.
- Desvantagens:
 - Introduce atraso
 - Aumenta a complexidade computacional
 - Necessita de armazenar em memória as frames anteriores e/ou posteriores;

Decisão inter/intra

- Modo Intra: Energia elevada
 - mudanças de cena
 - Objectos com oclusões
- Modo inter: Energia pequena
 - Pouco movimento

• Intra

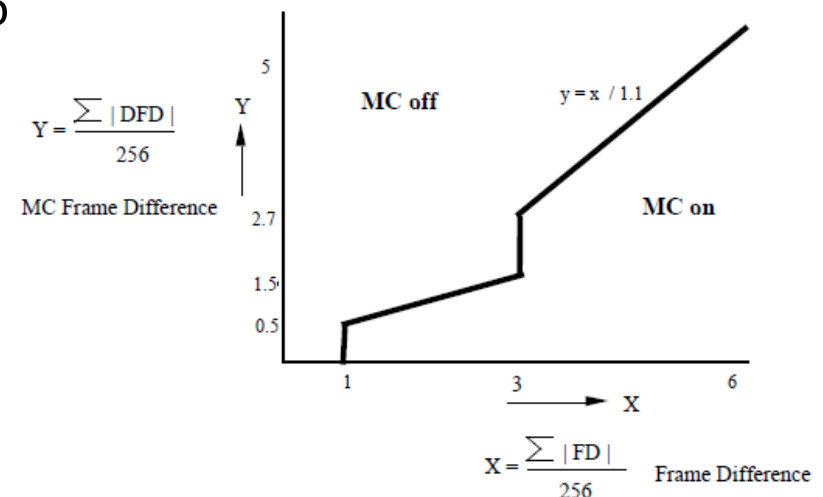


$$\text{Inter} \quad \text{VAR} = \left(\sum (\text{pel} - \text{mc_pel})^2 \right) / 256$$

$$\text{VAROR} = \left(\sum (\text{pel} - \text{blk_avg})^2 \right) / 256$$

Decisão com/sem Compensação de movimento

- Usa compensação de movimento quando a energia é substancialmente inferior



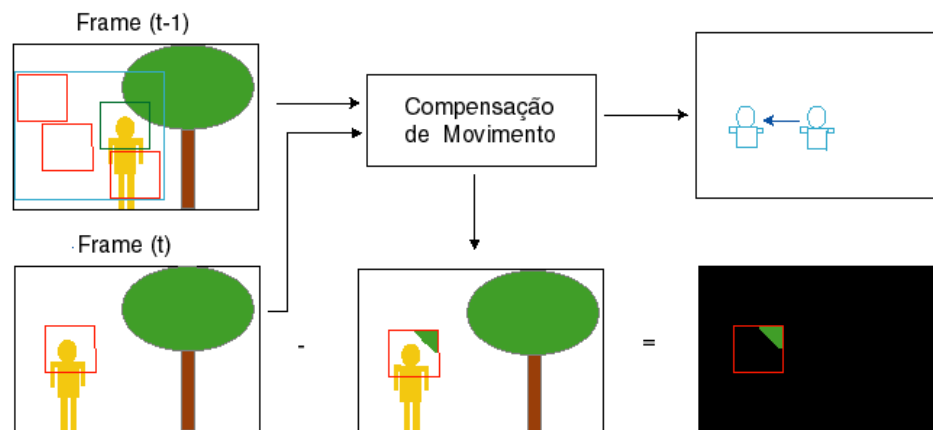
Estimação de Movimento

- Não se pode comparar pixel a pixel!
- O objectivo do MPEG4 é comparar objectos.
- A maior parte das técnicas baseia-se na comparação de blocos (block-matching), assumindo que os objectos são rígidos e que se movem numa direcção.
- Problema: Zoom ou rotações do objecto.
- Funciona bem se:
 - os blocos forem muito mais pequenos que o objecto e o frame rate for suficientemente elevado.
 - Os movimentos dos objectos ou a alteração das condições da câmara (posição, ângulo de visão, focagem, etc) não mudam rapidamente;
(assumindo um frame rate ajustado à situação)

Estimação de Movimento

- Funcionamento:

- Divide-se cada frame em blocos não sobrepostos (**Macro-blocos**);
O tamanho por defeito é 16 x 16 para a luminância e 8x8 para a crominância (admitindo subsampling de 4:2:0)
- Compara-se cada bloco com outros blocos da frame de referência dentro de uma janela de pesquisa (-15 a 15 por defeito);
- Determina-se qual o vector de movimento óptimo, ou seja, aquele com que se obtém menor valor para o critério adoptado.



Compensação de movimento

- Diferença entre vários algoritmos:
 - Critério de semelhança entre blocos
 - Erro absoluto médio (MAE)

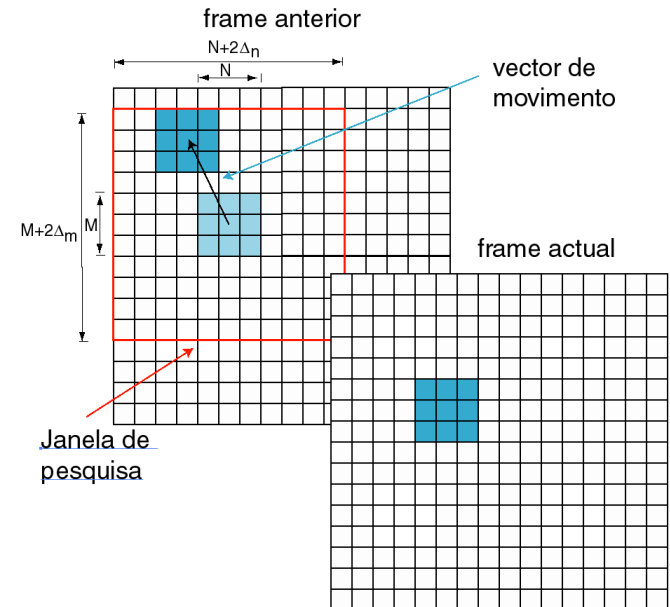
$$(d_m, d_n) = \arg \min_{d_m, d_n} \frac{1}{MN} \sum_{m=1}^M \sum_{n=1}^N |X^t(m, n) - X^{t-1}(m - d_m, n - d_n)|$$

- Erro quadrático médio (MSE)

$$(d_m, d_n) = \arg \min_{d_m, d_n} \frac{1}{MN} \sum_{m=1}^M \sum_{n=1}^N (X^t(m, n) - X^{t-1}(m - d_m, n - d_n))^2$$

$$d_m = -\Delta_m \dots \Delta_m, d_n = -\Delta_n \dots \Delta_n$$

- Estratégia de procura
- Determinação da dimensão dos blocos e da janela.



Estratégia de Procura

- Avaliada segundo:

- Eficiência da compensação de movimento

$$\frac{\text{Energia do bloco}}{\text{Energia do resíduo}}$$

- Complexidade computacional

Número de operações aritméticas efectuadas por cada bloco;
Notar que implementações em hardware podem paralelizar o algoritmo.

- Pesquisa exaustiva (FSM)

- Percorre todos os blocos

- Número de cálculos = $(2p + 1)^2$, $p = \Delta_m = \Delta_n$

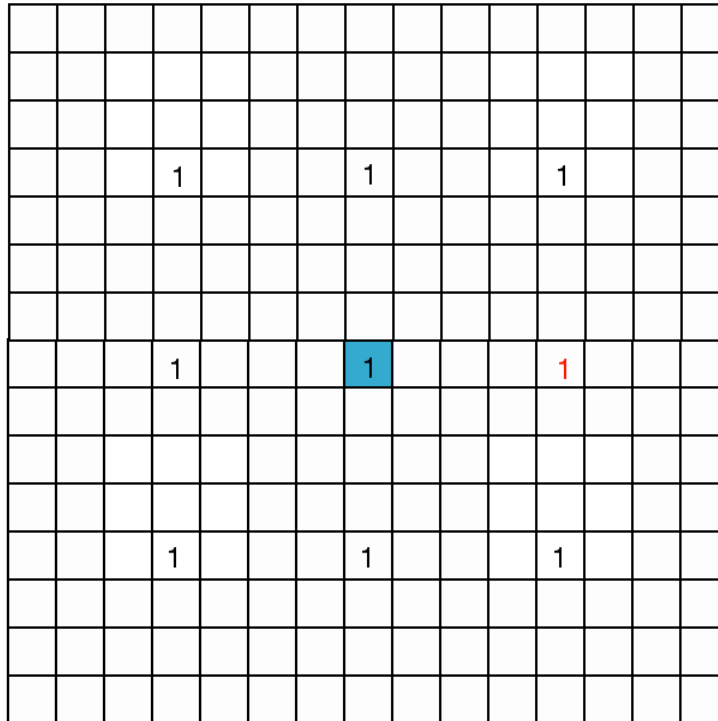
- Cada cálculo = $2(M \times N)$

- Pesquisa sub-óptimas:

- logaritmica 2D, em cruz, em três passos, em quatro passos, binária, em espiral, Hierarquica, etc

Pesquisa em três passos

- Pesquisa nove blocos incluindo o bloco central;
- Centra-se a janela de pesquisa no bloco com valor da MAE mais baixo, reduz o offset para metade;



Pesquisa em três passos

- Pesquisa nove blocos incluindo o bloco central;
- Centra-se a janela de pesquisa no bloco com valor da MAE mais baixo, reduz o offset para metade;

[illegible]

Pesquisa em três passos

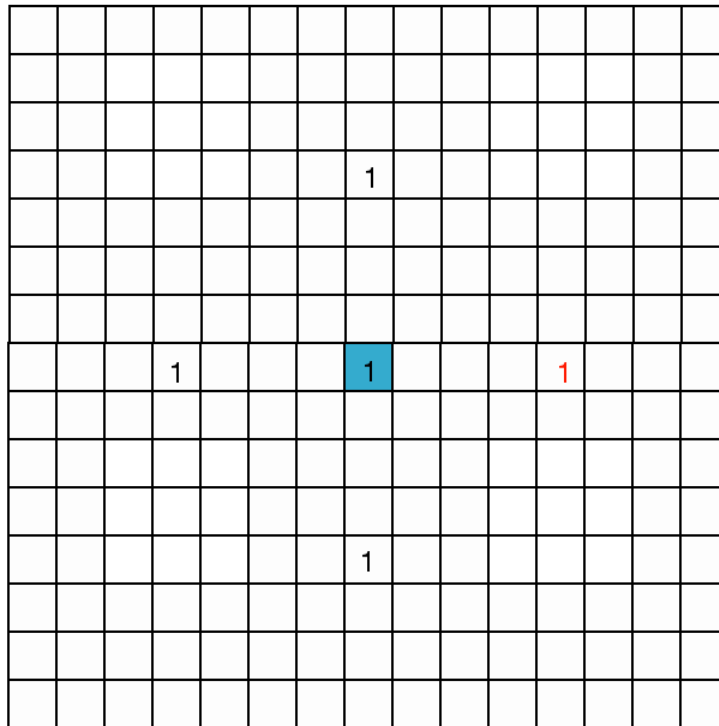
- Pesquisa nove blocos incluindo o bloco central;
- Centra-se a janela de pesquisa no bloco com valor da MAE mais baixo, reduz o offset para metade;

			1			1			1					
								2	2		2			
			1			1	2		1		2			
										3	3	3		
								2	2	3	2	3		
										3	3	3		
			1			1			1					

- Número de cálculos = $1 + 8 \log_2 p$

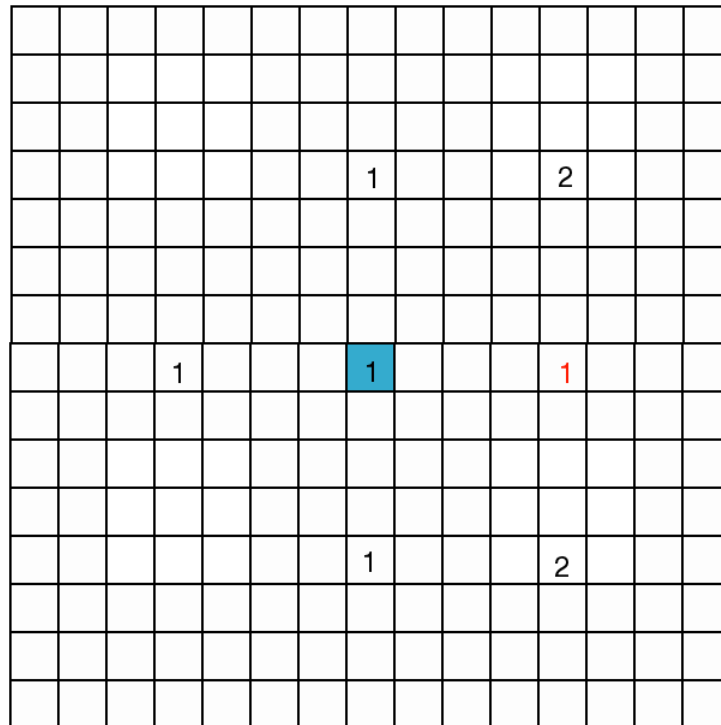
Pesquisa logaritmica 2D

- Pesquisa o bloco central e quatro blocos na direcção horizontal e vertical;
- Se o mínimo for encontrado num extremo, centra-se a janela de pesquisa nesse bloco;
- Se fôr no centro diminui-se a janela de pesquisa



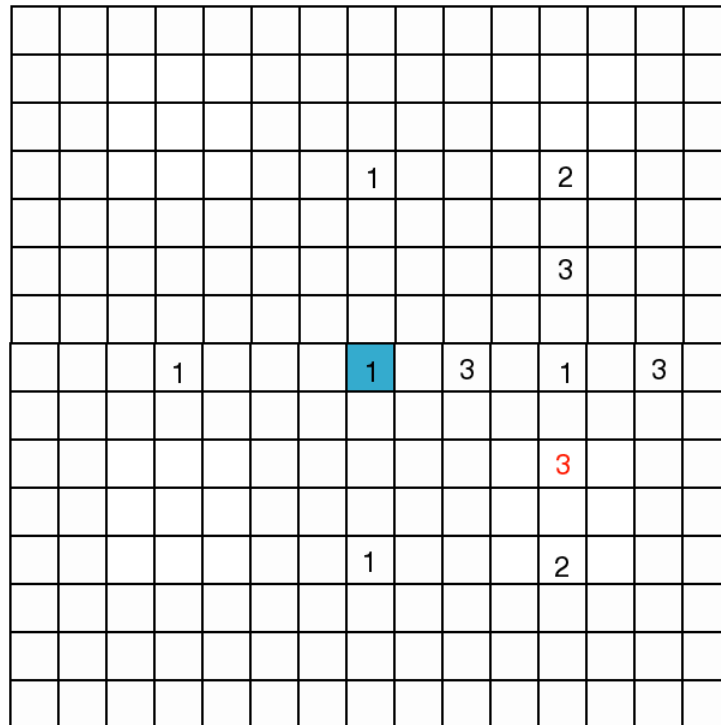
Pesquisa logaritmica 2D

- Pesquisa o bloco central e quatro blocos na direcção horizontal e vertical;
- Se o mínimo for encontrado num extremo, centra-se a janela de pesquisa nesse bloco;
- Se fôr no centro diminui-se a janela de pesquisa



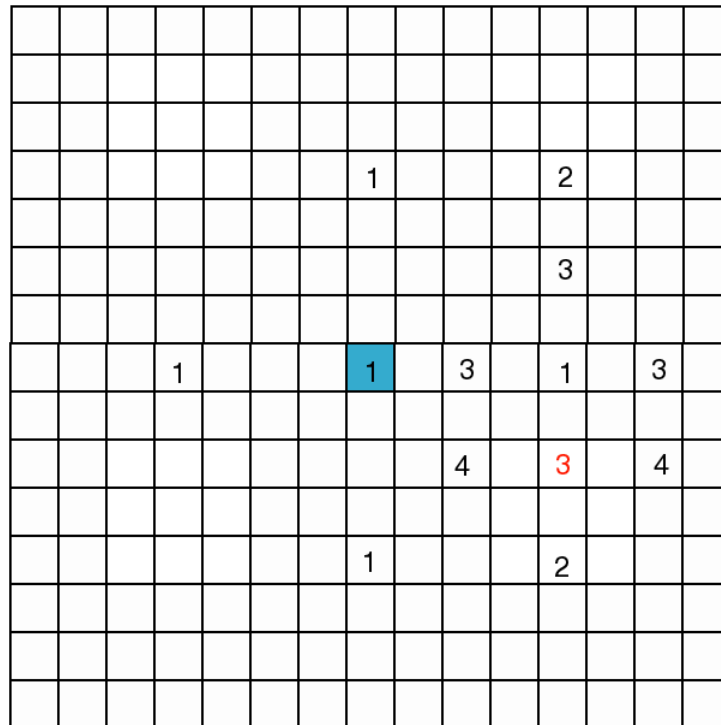
Pesquisa logaritmica 2D

- Pesquisa o bloco central e quatro blocos na direcção horizontal e vertical;
- Se o mínimo for encontrado num extremo, centra-se a janela de pesquisa nesse bloco;
- Se fôr no centro diminui-se a janela de pesquisa



Pesquisa logaritmica 2D

- Pesquisa o bloco central e quatro blocos na direcção horizontal e vertical;
- Se o mínimo for encontrado num extremo, centra-se a janela de pesquisa nesse bloco;
- Se fôr no centro diminui-se a janela de pesquisa



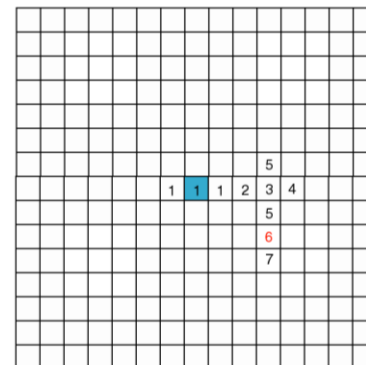
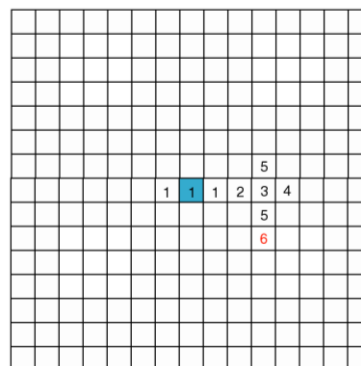
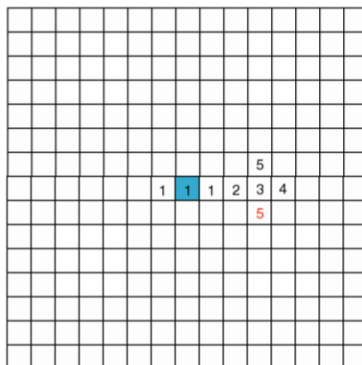
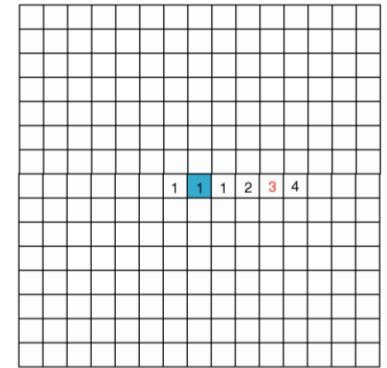
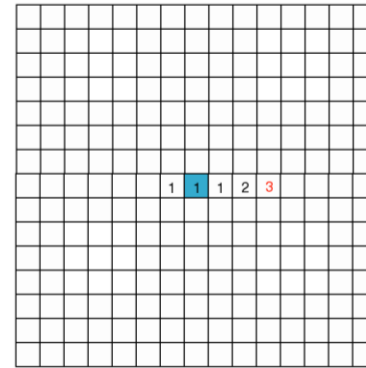
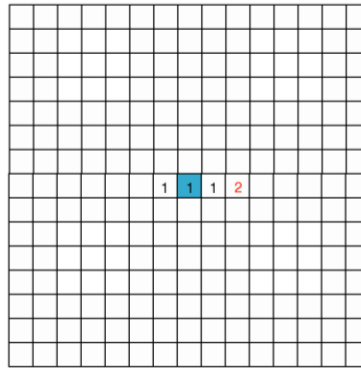
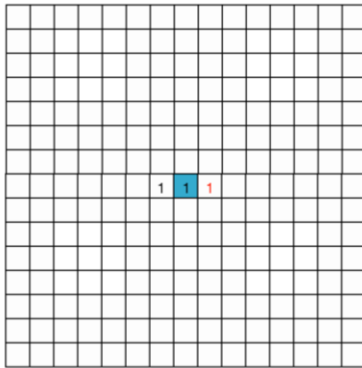
Pesquisa logaritmica 2D

- Pesquisa o bloco central e quatro blocos na direcção horizontal e vertical;
- Se o mínimo for encontrado num extremo, centra-se a janela de pesquisa nesse bloco;
- Se fôr no centro diminui-se a janela de pesquisa
- Número de cálculos = $2 + 7 \log_2 p$

[illegible]

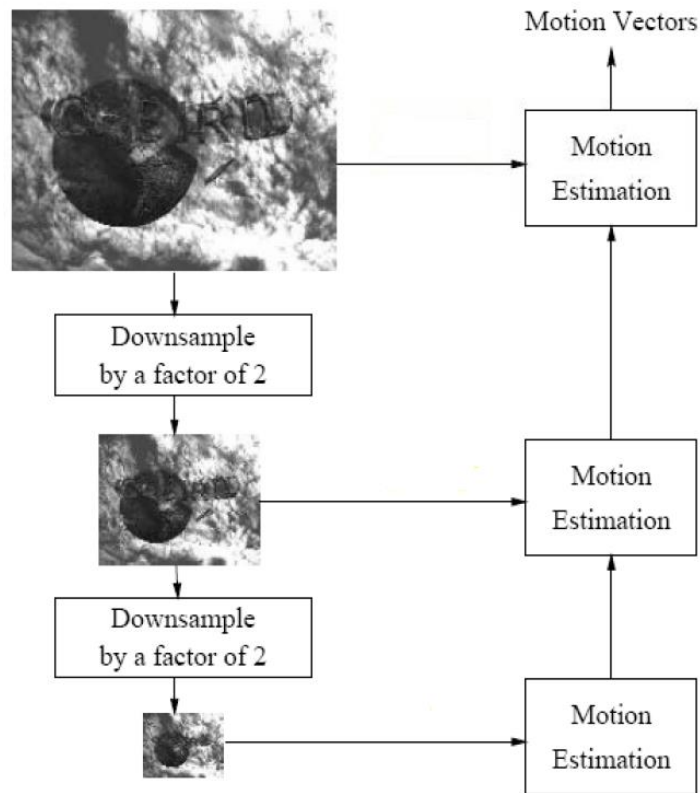
Pesquisa conjugada

- Pesquisa uma direcção à procura do mínimo e depois pesquisa a direcção ortogonal.
- Número de cálculos: $2p + 3$



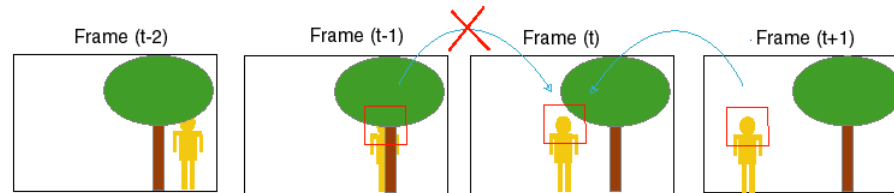
Pesquisa Hierarquica

- É uma pesquisa multiresolução:
 - É feita uma estimação inicial do vector de movimento numa resolução mais baixa;
 - Nas resoluções seguintes o vector de movimento é refinado.



Compensação de Movimento

- Existem algoritmos que pesquisam em fracções do pixel para estimar o vector de movimento;
- Estimação bidireccional:
 - Tem melhor desempenho quando há oclusões ou mudanças de cenas;
 - Obriga a reordenar as frames;

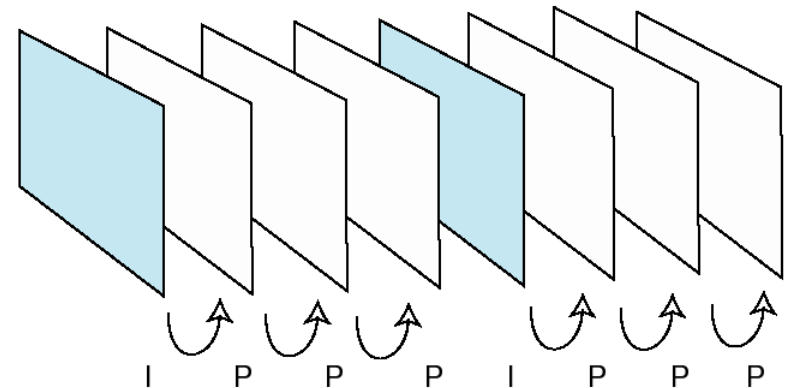


- A maior parte das técnicas faz estimação de movimento com a luminância e os vectores são também aplicados às duas componentes de crominância.

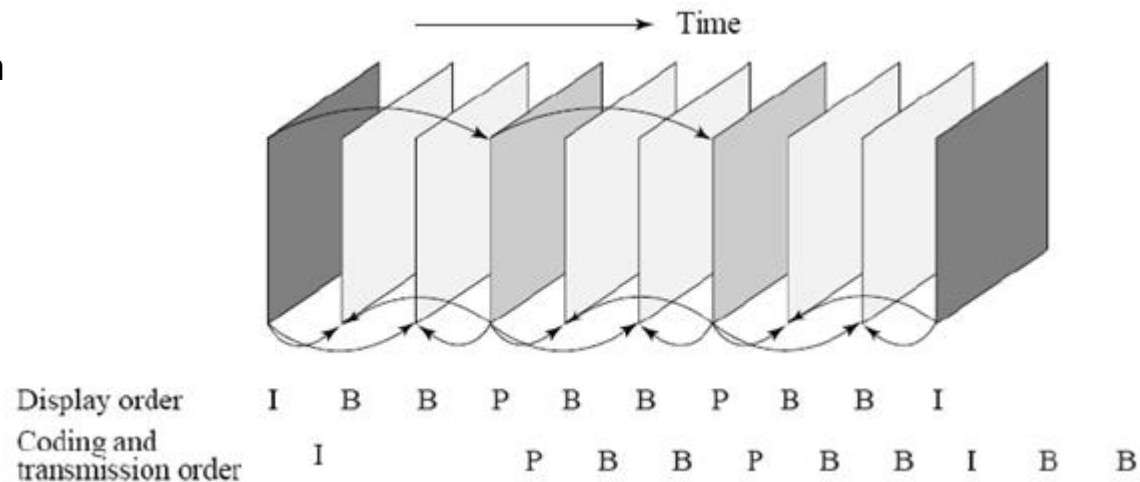
Sequência de frames

- Uma sequência é composta por frames:

- I-frames (intraframes)
- P-frames (interframes)

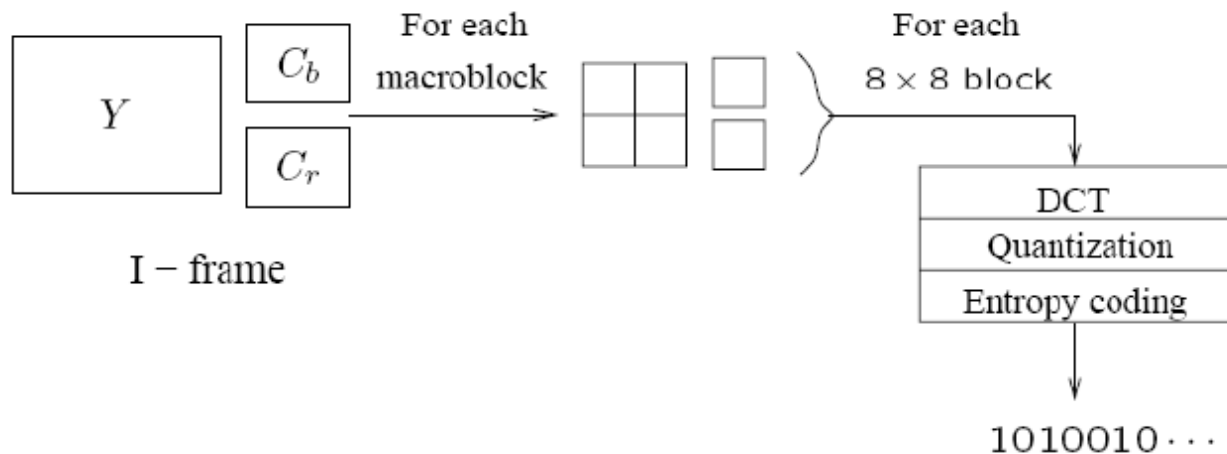


- B-frames (interframe)
- Maior bit rate
- Necessária mais memória
- Maior atraso



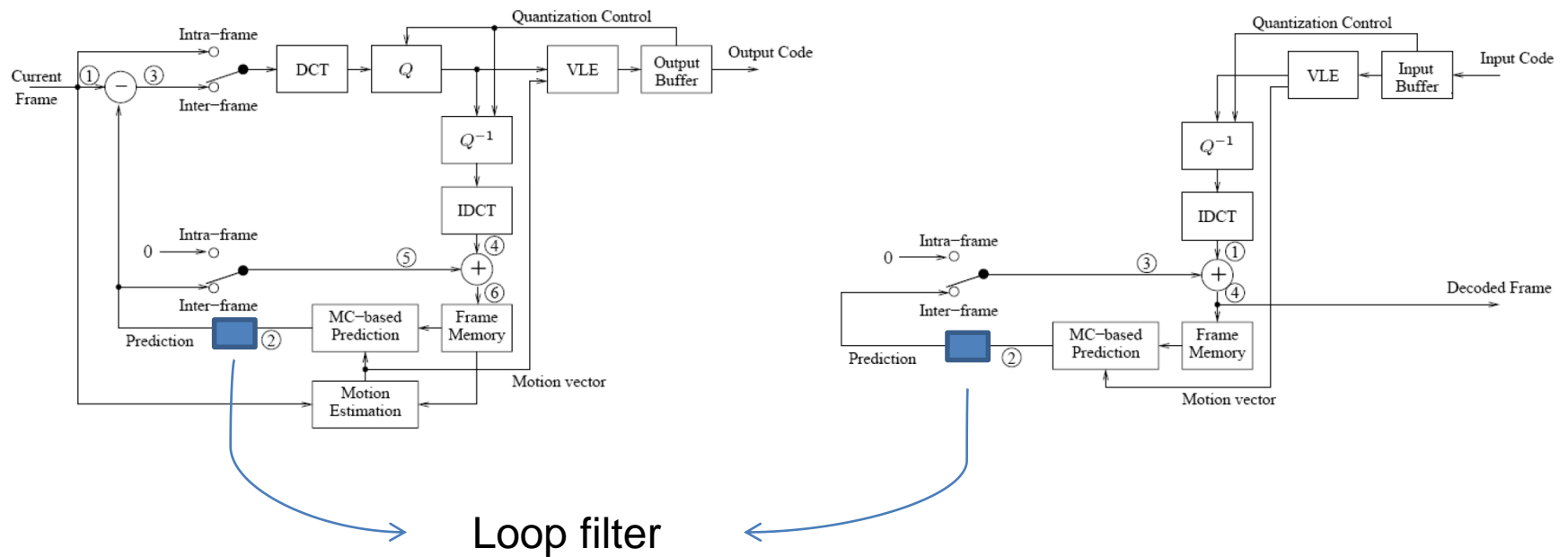
I-Frames

- Remoção de redundância espacial
- Tratadas como imagens independentes;
- DCT para cada bloco de 8x8 do MB;
- Quantificação (usa um coeficiente igual para o MB ou GOB);
- zigzag;
- Compressão por código de Huffman



- Macro-bloco (MB): 4 blocos Y, 1 bloco C_b e um bloco C_r (Derivado do subsampling 4:2:0)

Exemplo de codificador e decodificador (H.261)



Exemplo Quantificação (MPEG1)

- A quantificação dos coeficientes da DCT é diferente para as

- Intra-frames:

$$QDCT[i, j] = \text{round} \left(\frac{8 \times DCT[i, j]}{\text{step_size}[i, j]} \right) = \text{round} \left(\frac{8 \times DCT[i, j]}{Q_1[i, j] * \text{scale}} \right)$$

- Inter-frames:

$$QDCT[i, j] = \left\lfloor \frac{8 \times DCT[i, j]}{\text{step_size}[i, j]} \right\rfloor = \left\lfloor \frac{8 \times DCT[i, j]}{Q_2[i, j] * \text{scale}} \right\rfloor$$

- Tabela (intra) – Q_1

8	16	19	22	26	27	29	34
16	16	22	24	27	29	34	37
19	22	26	27	29	34	34	38
22	22	26	27	29	34	37	40
22	26	27	29	32	35	40	48
26	27	29	32	35	40	48	58
26	27	29	34	38	46	56	69
27	29	35	38	46	56	69	83

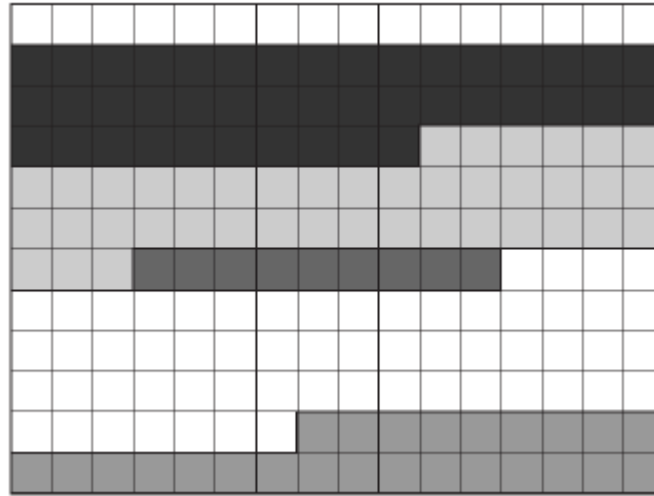
- Tabela (inter) – Q_2

16	16	16	16	16	16	16	16
16	16	16	16	16	16	16	16
16	16	16	16	16	16	16	16
16	16	16	16	16	16	16	16
16	16	16	16	16	16	16	16
16	16	16	16	16	16	16	16
16	16	16	16	16	16	16	16
16	16	16	16	16	16	16	16

- Scale é um inteiro entre 1 e 31

Organização da frame

- Uma frame é dividida em vários SLICES ou GOBs (usados na norma H.261)
- O SLICE começa e acaba numa imagem



- Cada slice contém um número variável de MB.
(a codificação de cada slice é independente e podem ter tabelas de quantificação diferentes - scale)
O conceito de slice é importante para a sincronização e a recuperação de erros: é o nível mais baixo na hierarquia que pode ser totalmente recuperado sem decodificar todos os VLC's
- O tamanho e a posição do SLICE é especificado no header

MPEG 2

				PERFIL (Profile)						
				Simple (SP) 4:2:0	Main (MP) 4:2:0	SNR Escal. SNR 4:2:0 (Main + SNR)	SPATIAL Escal.Espac 4:2:0 (SNR + Espacial)	High (HP) 4:2:0 /4:2:2 (+ espacial)	4:2:2 (extende Main com maior Deb. Bin.	
		Resolução	Frames /sec							Aplicações
NIVEL (Level)	Very High (HL)	1920x1152	60		IPB 62.7Mpix/s 80 Mbps			IPB 100Mbps (3 layers)	IPB 300Mbps	produção de filmes
	High (H14)	1440x1152	60		IPB 47 Mpix/s 60 Mbps		IPB 47Mpix/s 60Mbps (3 layers)	IPB 80Mbps (3 layers)		consumidor HDTV
	Main (ML)	720x576	30/25	IP 10.4Mp/s 15Mbps	IPB 10.4 Mpix/s 15 Mbps	IPB 10.4Mp/s 15Mbps (2 layers)		IPB 20Mbps (3 layers)	IPB 50MBps	estudio de TV
	Low (LL)	356x288	30/25		IPB 3.04 Mpix/s 4 Mbps	IPB 3.04Mp/s 4Mbps (2 layers)				consumidor, qual idade VHS

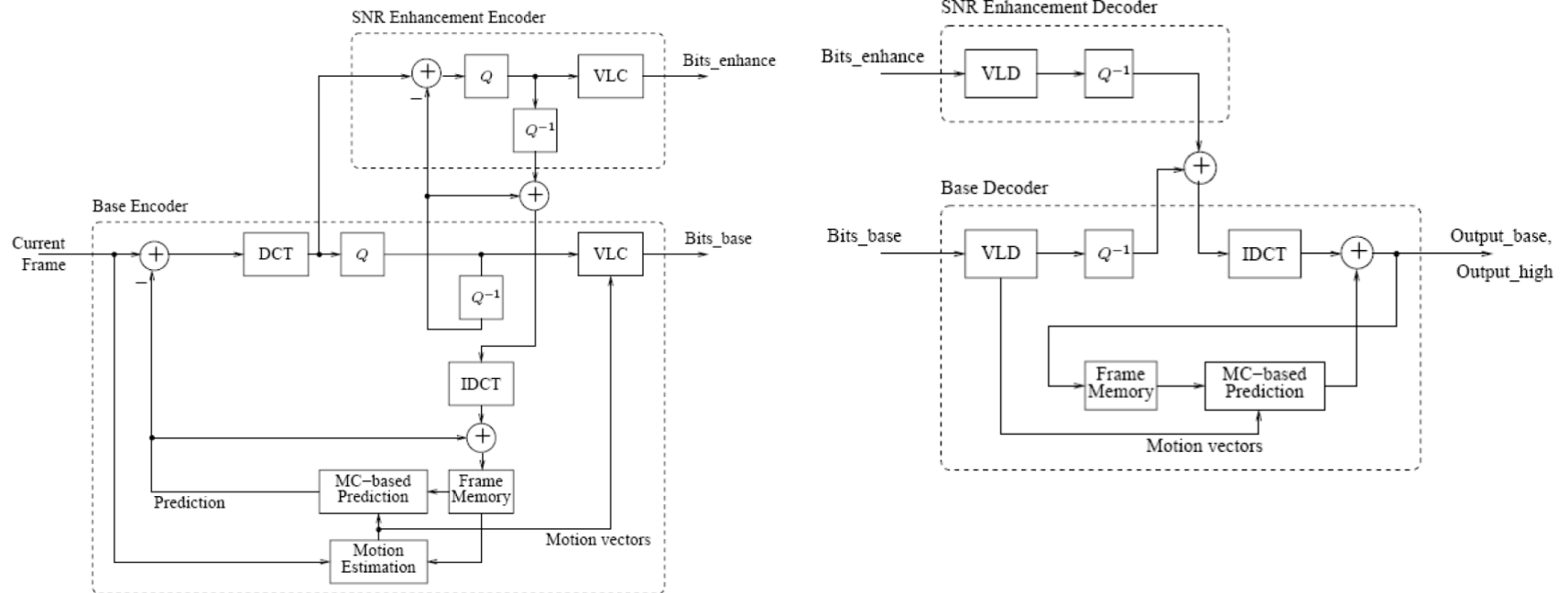
Na norma MPEG2 também existe no nível Main o perfil MVP – multi view profile

MPEG 2

- Modo de codificação escalaveis:
 - Codificação hierarquica do video;
 - As camadas correspondem a Bitstreams que podem ser transmitidos em canais diferentes
- Modos:
 - SNR : base layer e enhancement layer (melhor SNR)
 - Espacial: base layer e enhancement layer (maior resolução espacial)
 - Temporal: base layer e enhancement layer (maior frame rate)
 - Híbrida: combina dois dos três modos anteriores .
 - Partição de dados: depois da quantificação dos coeficientes da DCT este são separados em diferentes partições.

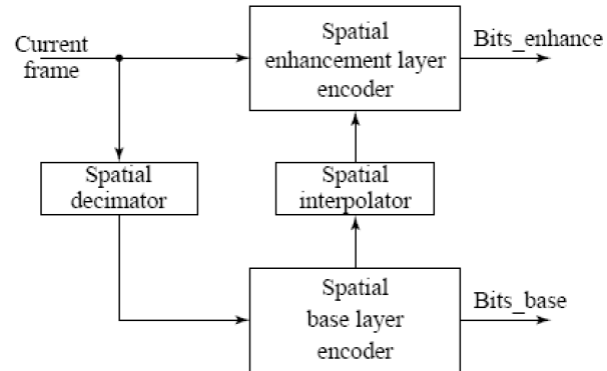
MPEG 2

■ SNR escalável:

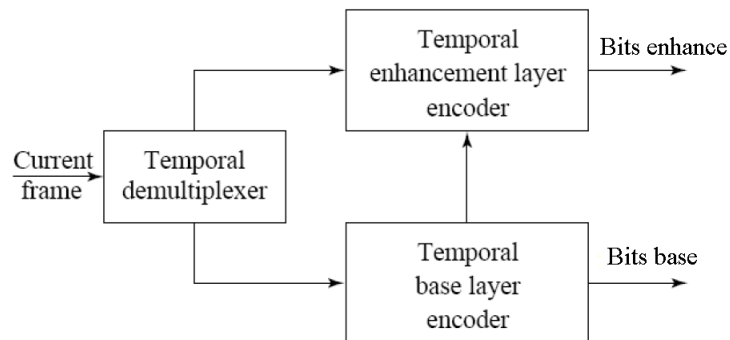


MPEG 2

■ Escalabilidade espacial:



■ Escalabilidade temporal:



MPEG4

- Desenvolvido com base nas tendências:
 - Comunicação sem fios
 - Aplicações interactivas (navegação e retransmissão)
 - Múltiplas aplicações (múltiplos fluxos de dados em simultâneo – vários pontos de vista da mesma cena)
 - Acesso universal (robustez a erros em canais ruidosos)
 - Flexibilidade e extensibilidade (Escalabilidade)
- Criada uma toolbox para diferentes tipos de dados audio-visuais
- Inicialmente pretendia-se transmissões com débito Binário baixo com o mínimo de distorção e baixa complexidade
- Paradigma da codificação de vídeo convencional ou baseada em objectos
- Manipulação e edição do bitstream sem descomprimir-lo

AVC = H.264 / MPEG4 – Part 10

- Codificação de vídeo baseada em frames
- Aumento da eficiência (50% do débito com a mesma qualidade do H.263, MPEG2 e MPEG4-part 2)
- Resiliência a erros (em meios como internet móvel e wireless)
- Características:
 - Componentes de cor com 8 a 14bits
 - Formatos 4:2:0, 4:2:2, 4:4:4
 - Conversão RGB – YCbCr (evita operações com virgula flutuante)
 - Transformada inteira de blocos 4x4 (reduz artefactos)
 - Múltiplas tramas de referência – cada MB pode ter até 8 e cada imagem até 16 imagens de referência (permite maior compressão e maior resiliência a erros)
 - Cada referência tem peso diferente (weighted prediction)
 - Slice tipo I (predição dentro do mesmo slice) tipo P e tipo B
 - B-slice pode servir como referência
 - B-slice podem usar como referência 2 (ambas posteriores ou anteriores)

AVC = H.264 / MPEG4 – Part 10

- Características:
 - Compensação de movimento com estrutura em árvore (16x16 , 8x16, 16x8, 8x8, 4x4)
 - Vectores de movimento com resolução de $\frac{1}{4}$ pixel
 - Deblocking filter in the loop



1) Without filter

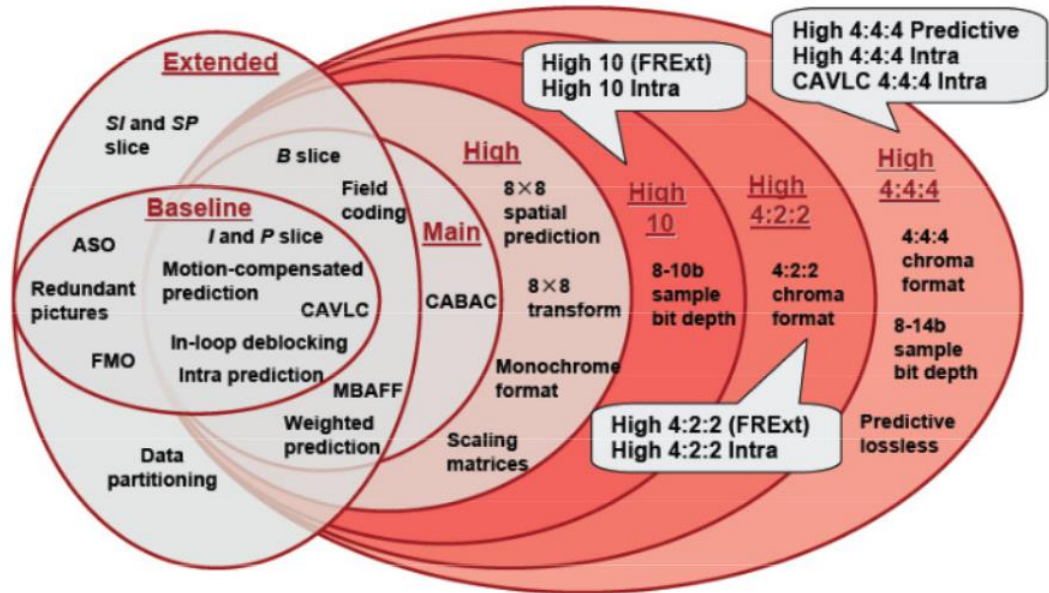
2) With H.264/AVC deblocking

AVC = H.264 / MPEG4 – Part 10

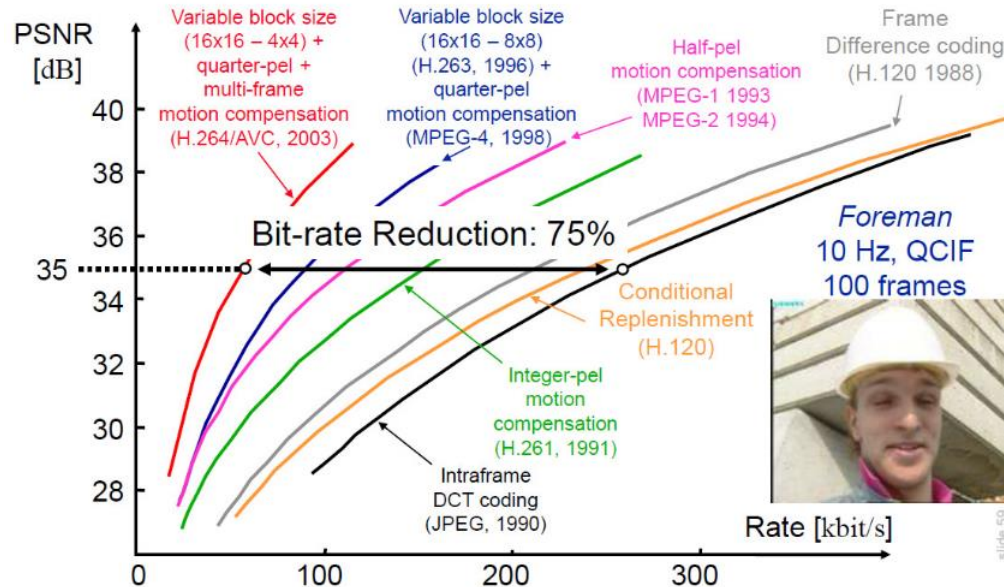
- Características:
 - Codificação entrópica:
 - UVLC – Universal VLC (Usa uma tabela em vez de várias)
 - CAVLC – Context Adaptive VLC (selecciona a tabela com base na informação já processada)
 - CABAC – Context Adaptive Binary Arithmetic Coding (o modelo de probabilidade é actualizado à medida que se processa a informação)
 - Modos flexíveis para evitar erros noemadamente em redes IP
 - FMO – Flexible MB ordering
 - ASO – Arbitrary slice ordering
 - RS – Redundant slices

AVC = H.264 / MPEG4 – Part 10

- Perfis:
 - Main – broadcast and storage (Blue-ray disc)
 - Não inclui FMO, ASO, RS
 - High,
 - High10
 - High 10 intra
 - High 4:2:2
 - High 4:2:2 intra
 - High 4:4:4
 - High 4:4:4 intra



AVC = H.264 / MPEG4 – Part 10



- Extensões
 - SVC – Scalable Video Coding
 - MVC – Multi-view Video Coding
- Próximo Standard – HEVC – High Efficiency Video Coding