



SCC0661 – MULTIMÍDIA E HIPERMÍDIA

AULA 07: VÍDEO DIGITAL

Prof. Dr. Marcelo G. Manzato
(mmanzato@icmc.usp.br)

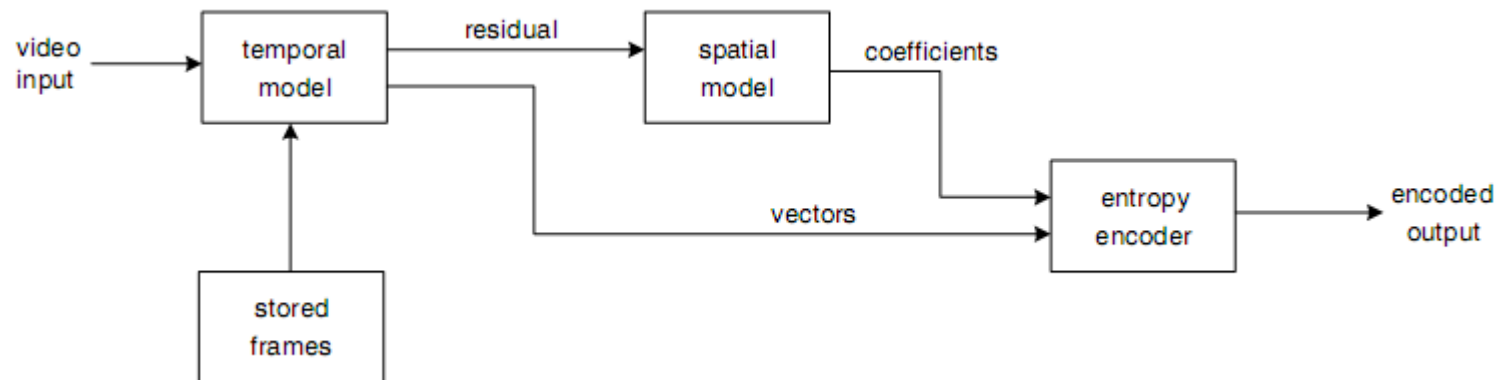
Instituto de Ciências Matemáticas e de Computação – Sala 3-111

1. PRINCÍPIOS DE COMPRESSÃO DE VÍDEO

- Compressão : eliminação de redundâncias
 - Estatística (JPEG-LS)
 - Estatística + Espacial (MJPEG)
 - Estatística + Espacial + Temporal (MPEG)



1. PRINCÍPIOS DE COMPRESSÃO DE VÍDEO



○ Modelo

- Representação dos dados codificados que pode ser usada para reconstruir o vídeo original
- Idealmente deve utilizar poucos bits e recompor os dados com alta fidelidade

1. PRINCÍPIOS DE COMPRESSÃO DE VÍDEO

- Redundância entre quadros adjacentes (redundância temporal)
 - Técnicas para remoção:
 - Prever (“predizer”) o conteúdo de quadros sucessivos.
 - Apenas as diferenças são codificadas.
- Acuidade da predição
 - Quanto bem o movimento é estimado.
 - Operação é chamada de **Estimativa de Movimento**.
 - Predição não é perfeita.
 - **Compensação de Movimento**.

1.1 TIPOS DE QUADROS

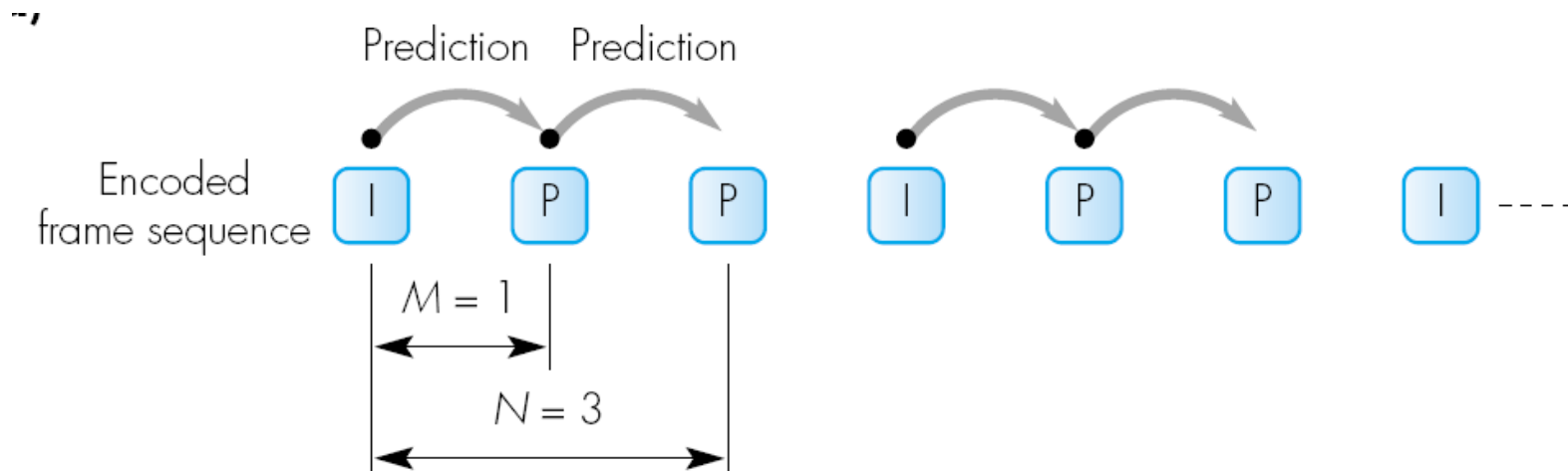
- Para aplicar a técnica de remoção de redundância temporal, é necessário saber primeiro os tipos de quadros existentes
- Três tipos básicos de quadros:
 - Quadros codificados independentemente
 - *Intracoded frames* ou *I-frames* ou quadros I.
 - Quadros “preditos” (*predicted frames*)
 - *Predictive* ou *P-frames* ou quadros P.
 - *Bidirectional* ou *B-frames* ou quadros B.
 - Quadros de apoio
 - *D-frames* ou *DC-pictures* ou quadros D (não tão usados).

1.1 TIPOS DE QUADROS

○ Quadros I.

- São codificados sem nenhuma referência a outros quadros.
- Cada quadro é tratado como uma imagem independente sendo Y, Cb e Cr codificados usando o algoritmo JPEG.
- Aparecem no fluxo de saída em intervalos regulares.
 - N = GOP (*group of pictures*)
 - *span*: número de quadros (3 a 12) até chegar a um quadro I sucessivo.

1.1 TIPOS DE QUADROS

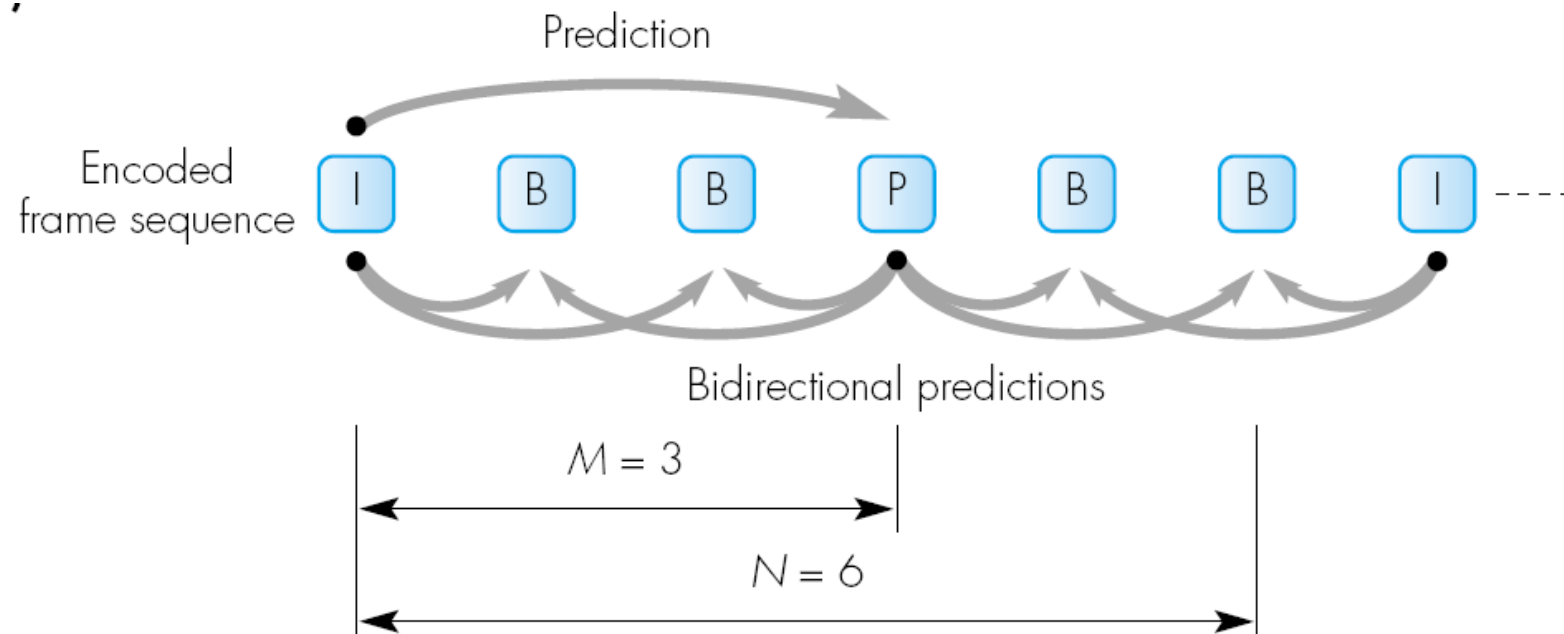


1.1 TIPOS DE QUADROS

○ Quadros P

- São codificados em relação ao conteúdo de um quadro I ou de um quadro P anterior.
- Usam combinação de estimativa e compensação de movimento
 - Alcançam maiores taxas de compressão do que quadros I.
- **Propagam erros** - número de quadros P entre quadros I é limitado.
- $M = \textit{prediction span}$
 - número de quadros do quadro P até um quadro I ou P imediatamente anterior.
- Desempenho: taxa de compressão entre 20:1 e 30:1.

1.1 TIPOS DE QUADROS



M = prediction span N = group of pictures (GOP) span

1.1 TIPOS DE QUADROS

○ Quadros B

- São codificados em relação ao conteúdo de um quadro I ou de um quadro P anterior e/ou de um posterior.
- Envolve o processamento de 3 quadros: o quadro I ou P anterior, o quadro atual e o quadro I ou P posterior. Todos não codificados.
- Aumento no tempo (*delay*) para codificação e decodificação. É o tempo de esperar o próximo quadro I ou P.
- Provêem alta taxa de compressão: entre 30:1 e 50:1.
- Não propagam erros. Por quê?

1.1 TIPOS DE QUADROS

○ Quadros B

- Decodificação:

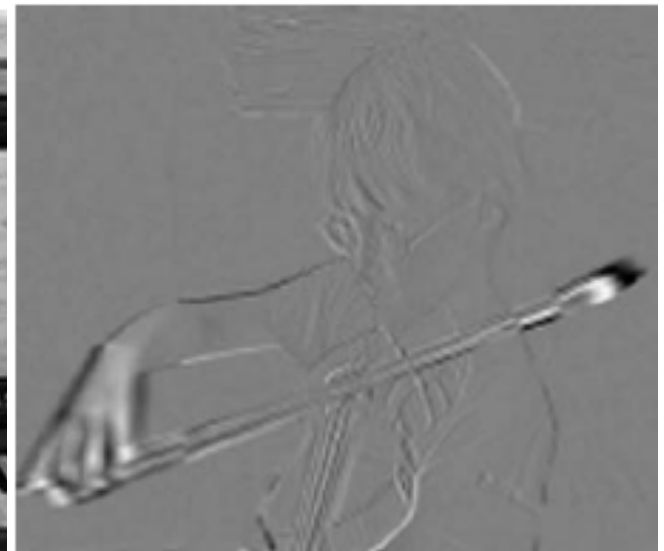
$I_1 B_2 B_3 P_4 B_5 B_6 P_7 B_8 B_9 I_{10} \dots$

- Codificação:

$I_1 P_4 B_2 B_3 P_7 B_5 B_6 I_{10} B_8 B_9 \dots$

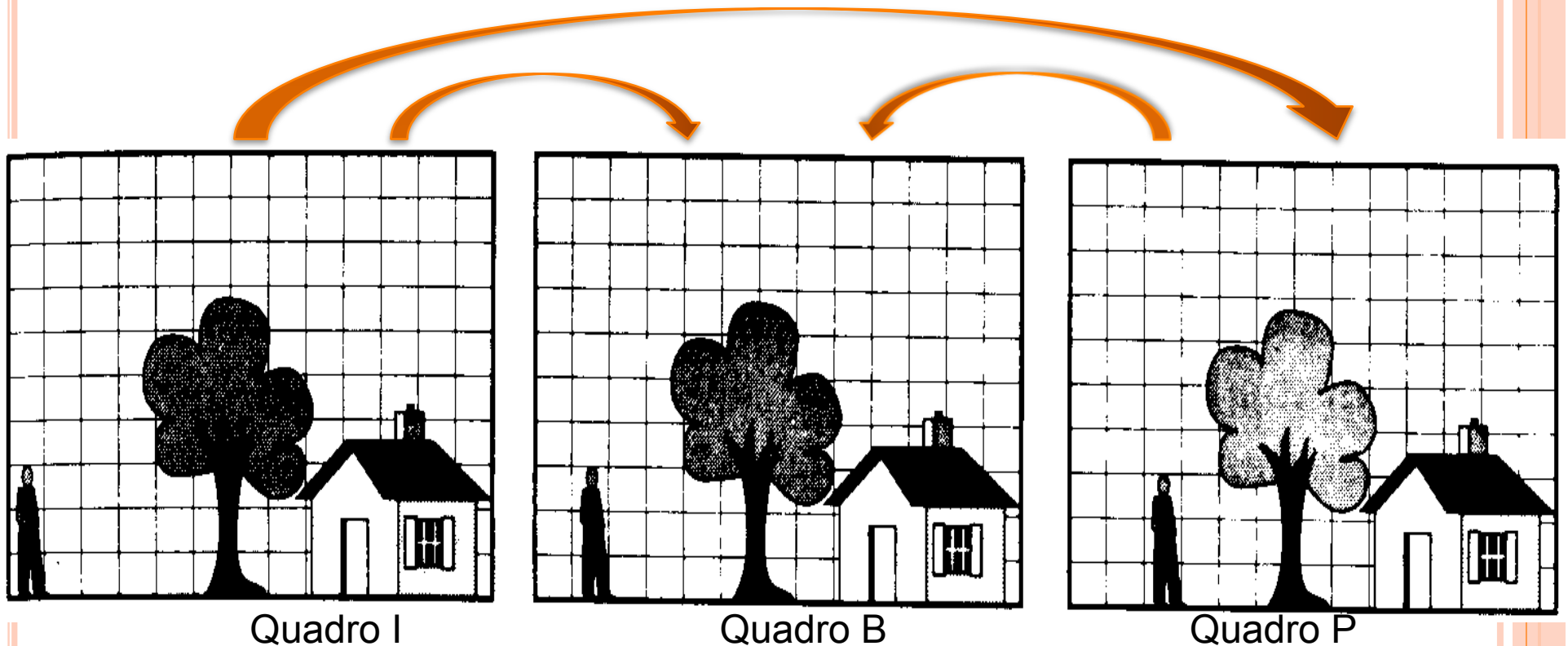


1.2 ESTIMATIVA E COMPENSAÇÃO DE MOVIMENTO



1.2 ESTIMATIVA E COMPENSAÇÃO DE MOVIMENTO

- Estimativa e Compensação baseada em Bloco
 - Consiste em achar regiões na imagem que podem ser encontradas nas imagens seguintes.

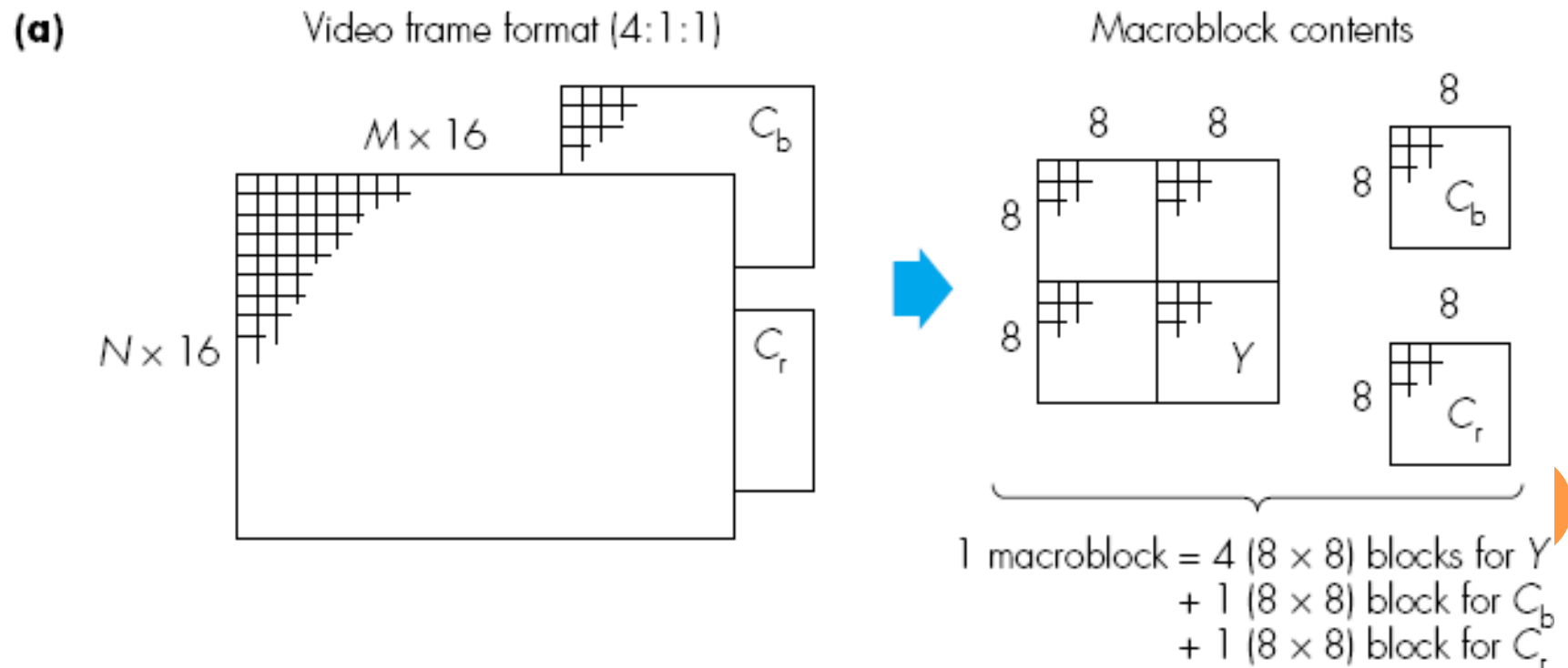


1.2 ESTIMATIVA E COMPENSAÇÃO DE MOVIMENTO

- Estimativa e Compensação baseada em Bloco
 - Buscar na imagem de referência uma região que melhor se assemelha à região do quadro atual (**estimativa de movimento**)
 - A região candidata torna-se o previsor do bloco do quadro atual, sendo subtraída do bloco atual para formar um resíduo (**compensação de movimento**)
 - O resíduo é codificado juntamente com o *offset* entre o bloco atual e a posição da região candidata (**vetor de movimento**)

1.2 ESTIMATIVA E COMPENSAÇÃO DE MOVIMENTO

- A imagem é dividida em macroblocos.
 - Y, Cr e Cb = matrizes de 16x16, 8x8 e 8x8 pixels (formato 4:1:1).

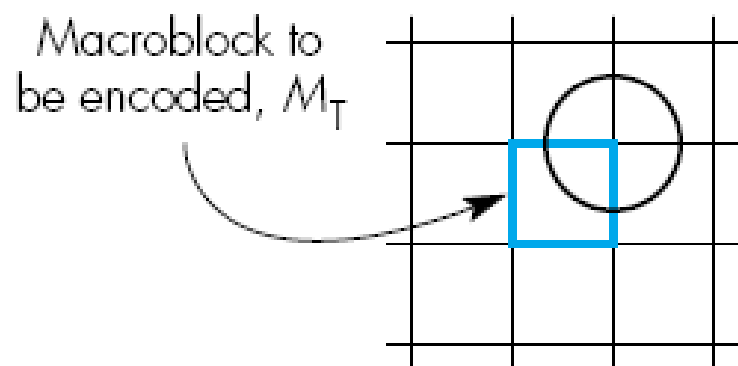


1.2 ESTIMATIVA E COMPENSAÇÃO DE MOVIMENTO

- Para codificar **quadros P**, cada macrobloco do quadro-alvo é comparado, pixel a pixel, com o macrobloco correspondente do quadro-referência (I ou P anterior).
 - Se o conteúdo combina (*match*), apenas o *offset* do macrobloco e o erro de predição são codificados. Senão, estende-se a busca para macroblocos vizinhos.
 - Normalmente utiliza-se apenas a componente Y.

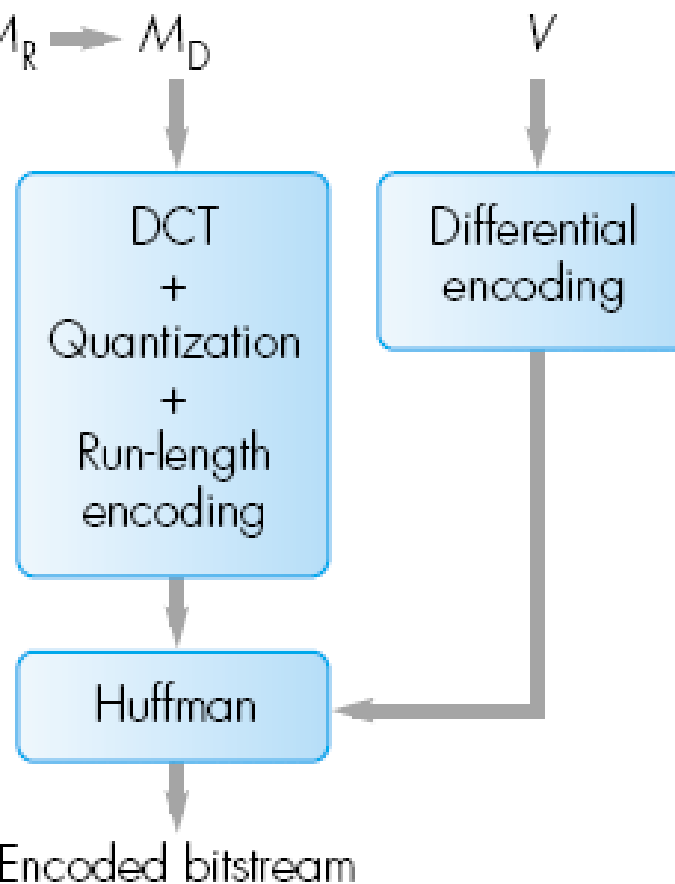
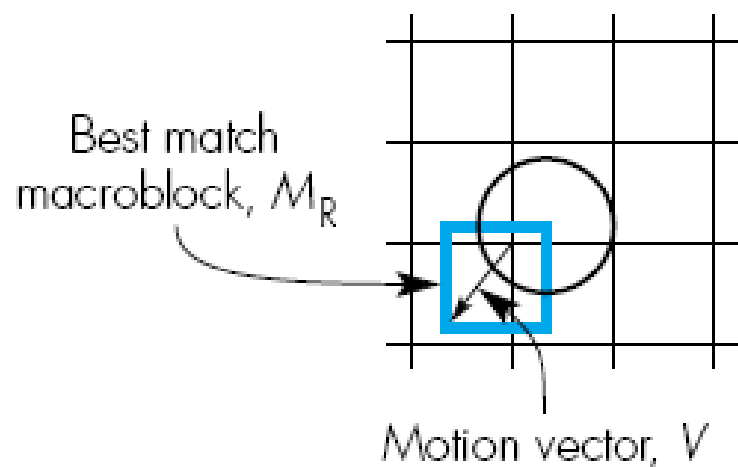
(b)

Search region in target frame:



$$M_T - M_R \rightarrow M_D$$

Same search region in preceding (I or P) reference frame:



1.2 ESTIMATIVA E COMPENSAÇÃO DE MOVIMENTO

- Parâmetros codificados:
 - Um vetor de movimento (*motion vector*).
 - Indica o deslocamento (*offset*) do macrobloco.
 - Erro de predição
 - Três matrizes – uma para cada componente (Y, Cr e Cb) – contendo as diferenças de valores entre os pixels do macrobloco-alvo e os pixels da área de busca.
 - É necessário pois a estimativa de movimento não é um método exato.

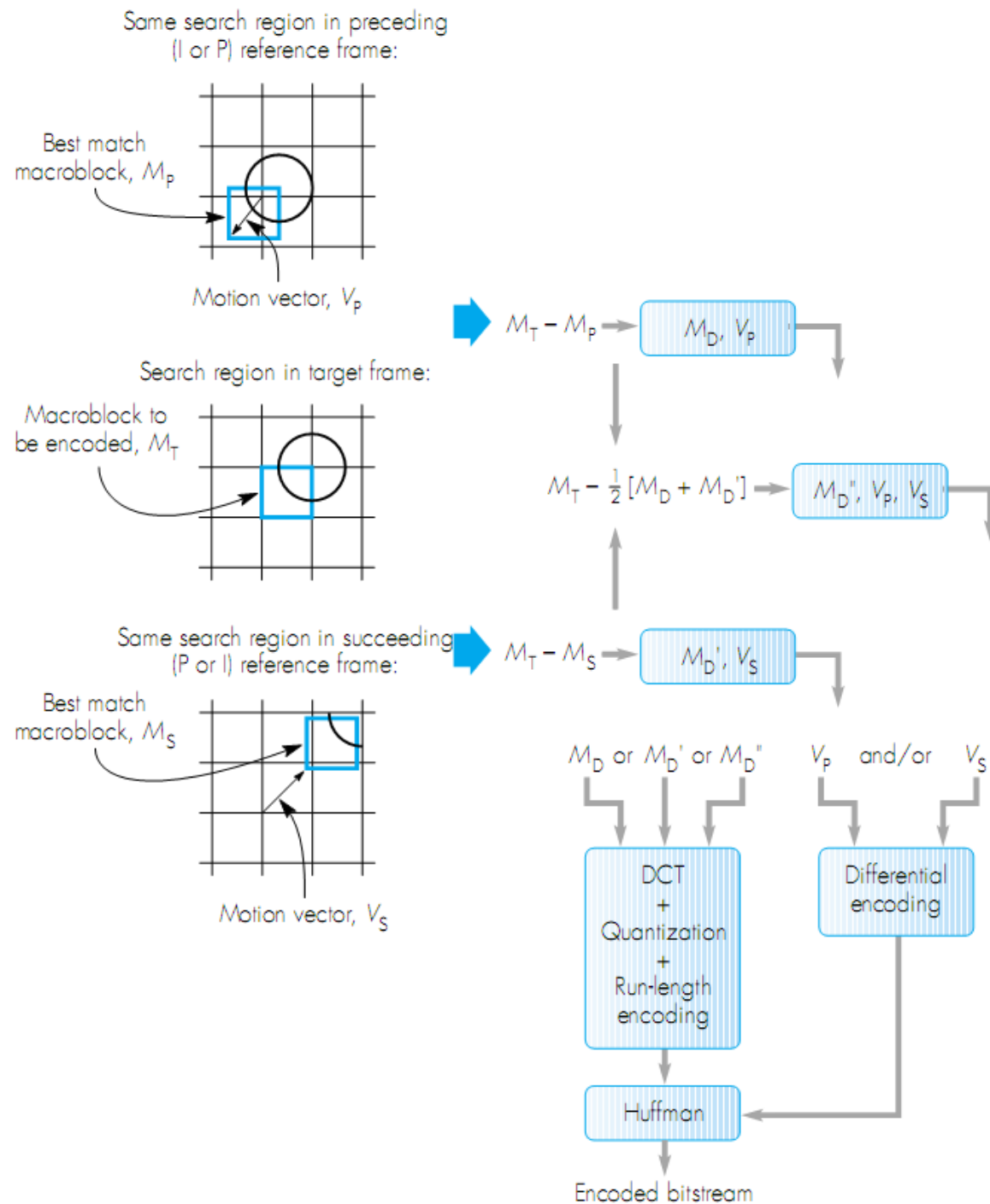
1.2 ESTIMATIVA E COMPENSAÇÃO DE MOVIMENTO

- Codificação dos parâmetros:
 - Vetores de movimento
 - Codificados usando codificação por diferenças
 - Ex. Macroblocos de um mesmo objeto movente
 - Resultado é acoplado na codificação Huffman
 - Erro de predição
 - Codificado como um quadro I
 - Mas as matrizes contém apenas as diferenças entre os macroblocos do quadro alvo e do quadro referência.
- Se um “casamento” (*match*) não é encontrado:
 - Macrobloco é codificado independentemente.
 - Codificação segue os passos de um quadro I:
 - DCT, quantização e codificação por entropia.

1.2 ESTIMATIVA E COMPENSAÇÃO DE MOVIMENTO

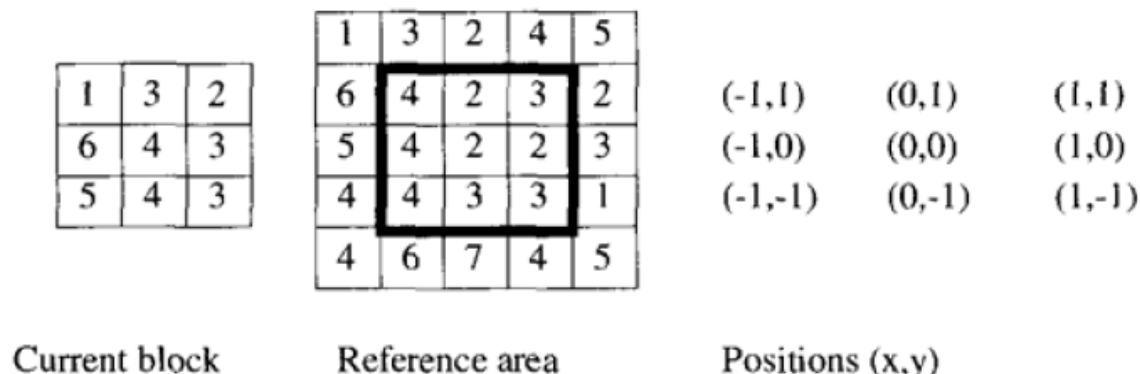
○ Quadros B:

- Estima-se, primeiro, o vetor de movimento e as matrizes de diferenças usando-se o quadro P ou I anterior.
- Depois, estimam-se os mesmos parâmetros usando-se o quadro P ou I posterior.
- Calcula-se um terceiro conjunto de parâmetros usando o macrobloco-alvo e a média dos valores previstos nos dois passos anteriores.
- O conjunto com os menores valores é escolhido para ser codificado – como em um quadro P.



ESTIMATIVA E COMPENSAÇÃO DE MOVIMENTO

- *Block matching*



- Erro quadrático médio – Mean Squared Error (MSE)
 - Posição (0,0)

$$\{(1-4)^2 + (3-2)^2 + (2-3)^2 + (6-4)^2 + (4-2)^2 + (3-2)^2 + (5-4)^2 + (4-3)^2 + (3-3)^2\} / 9 = 2.44$$



ESTIMATIVA E COMPENSAÇÃO DE MOVIMENTO

- *The “best” match* : posição $(-1, 1)$ – Melhor predição (“modelo”) é a região apontada por $(-1, 1)$.
 - Vetor de movimento: $(-1, 1)$

Table 6.1 MSE values for block matching example

Position (x, y)	$(-1, -1)$	$(0, -1)$	$(1, -1)$	$(-1, 0)$	$(0, 0)$	$(1, 0)$	$(-1, 1)$	$(0, 1)$	$(1, 1)$
MSE	4.67	2.89	2.78	3.22	2.44	3.33	0.22	2.56	5.33



ALGORITMO

○ Codificação:

- 1. Calcular a diferença entre o macrobloco atual e um conjunto de regiões de vizinhança no quadro de referência.
- 2. Selecionar a região que fornece o menor erro (“best match”).
- 3. Subtrair a região selecionada do macrobloco atual para produzir um macrobloco de resíduo (erro de predição).
- 4. Codificar e transmitir o macrobloco de resíduo.
- 5. Codificar e transmitir o vetor de movimento, que indica a posição da região selecionada em relação à posição do macrobloco atual (ex. $(-1, 1)$).

○ Decodificação:

- 1. Decodificar o macrobloco de resíduo e vetor de movimento.
- 2. Somar o bloco de resíduo com a região apontada pelo vetor de movimento no quadro de referência.



ESTIMATIVA E COMPENSAÇÃO DE MOVIMENTO

- Critérios de comparação:

$$\text{MSE} = \frac{1}{N^2} \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} (C_{ij} - R_{ij})^2$$

- Mean squared error

$$\text{MAE} = \frac{1}{N^2} \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} |C_{ij} - R_{ij}|$$

- Mean absolute error

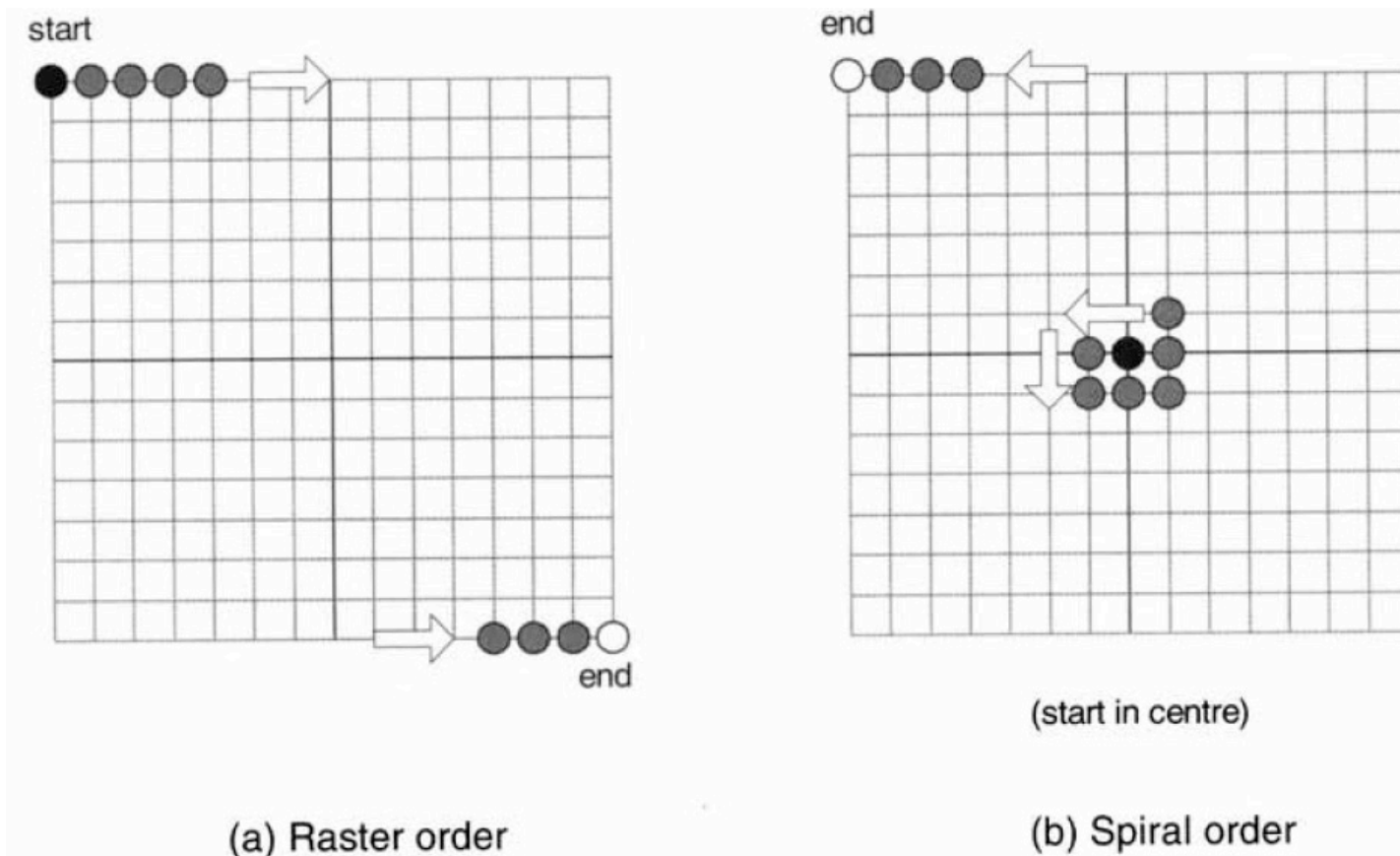
$$\text{SAE} = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} |C_{ij} - R_{ij}|$$

- Sum of absolute errors/
differences



ESTIMATIVA E COMPENSAÇÃO DE MOVIMENTO

■ Full Search:



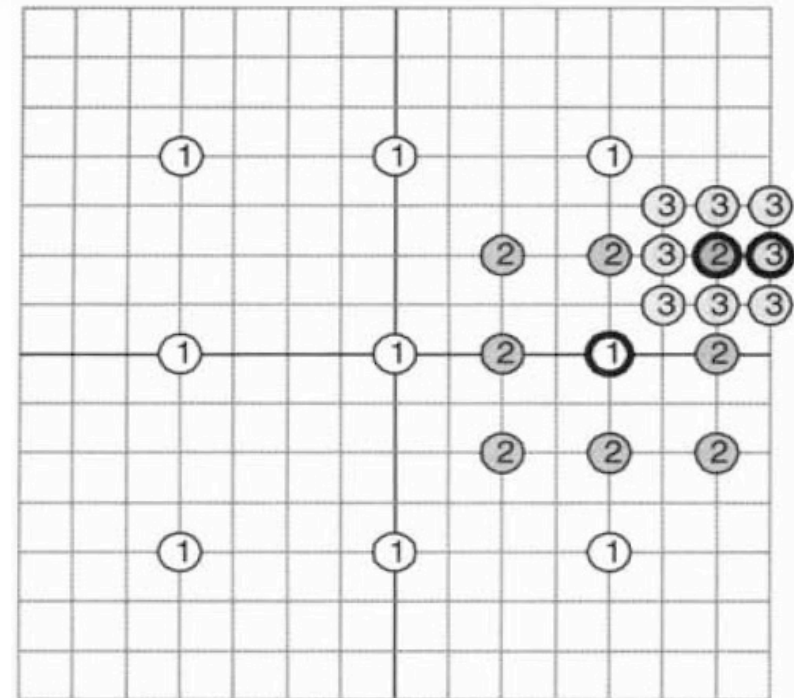
BUSCA RÁPIDA

- Full search normalmente é proibitivo, principalmente em CODECs que devem operar em tempo real.
- Busca rápida (*fast search*) procura reduzir consideravelmente o número de comparações através de amostragem.

BUSCA RÁPIDA

■ Three-Step Search (TSS)

1. Search location (0, 0).
2. Set $S = 2^{N-1}$ (the step size).
3. Search eight locations $\pm S$ pixels around location (0, 0).
4. From the nine locations searched so far, pick the location with the smallest SAE and make this the new search origin.
5. Set $S = S/2$.
6. Repeat stages 3–5 until $S = 1$.



25 comparações usando TSS ($N=3 \rightarrow S=4$)
225 comparações usando full-search

BUSCA RÁPIDA

- Ponto crítico:
 - O fast search consegue encontrar um SAE mínimo “global” ao invés de um SAE mínimo “local”?
 - Normalmente, fast search produz performance de compressão inferior do que o full search.

Table 6.3 Motion estimation algorithm comparison, five frames: search window = ± 15

Algorithm	Total SAE (uncompensated)	Total SAE (compensated)	Number of comparison operations
Full search	1 326 783	897 163	99.1×10^6
Three-step search	...	914 753	3.6×10^6

PARA SABER MAIS

- Luther, A. C. Using Digital Video. AP Professional, 1995. (capítulo 2 e apêndice A).
- Richardson, L. E. G. H.264 and MPEG-4 Video Compression, Wiley, 2003.
- Halsall, F. Multimedia Communications: Applications, Networks, Protocols, and Standards, Addison-Wesley Publishing, 2001. ISBN: 0201398184. Capítulo 3, seção 4.3.
- H.261 e H.263:
 - http://www.compression-links.info/H.261_H.263
- Padrões MPEG:
 - <http://www.chiariglione.org/mpeg/>