

# COMPRESSÃO DE IMAGENS EM MOVIMENTO: PADRÕES MPEG, H.264, DIRAC

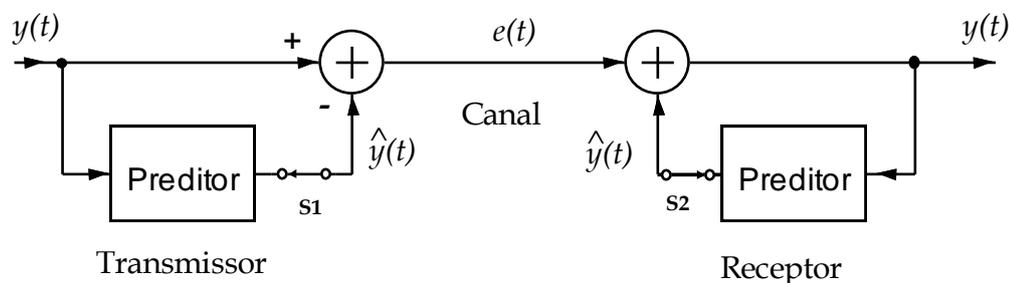
## PTC3547 - CODIFICAÇÃO E TRANSMISSÃO MULTIMÍDIA

Guido Stolfi – EPUSP – 05 / 2018

### 1. Codificação com Preditor

Todo processo de compressão de dados busca de alguma forma reduzir a redundância presente na informação que é fornecida pela fonte de sinal. Um recurso bastante útil para isto é o uso de um codificador com **preditor**. Este tipo de codificador (vide figura 1.1) procura determinar uma estimativa  $\hat{y}(t)$  do sinal a ser transmitido, baseando-se nas entradas anteriores  $y(t-1)$ ,  $y(t-2)$ .... Desta forma, basta transmitir apenas o erro de predição  $e(t)$ , ou seja, a diferença entre a estimativa  $\hat{y}(t)$  e o sinal verdadeiro  $y(t)$ . Se o preditor for eficiente, o erro de predição terá baixa auto-correlação e irá assumir preferencialmente valores próximos de zero, reduzindo assim a quantidade de informação a ser transmitida.

Não havendo erros de transmissão, o receptor da figura 1.1 reconstrói integralmente o sinal original, desde que os dois preditores sejam iniciados simultaneamente. Na prática, para evitar propagação de erros, deve-se periodicamente desconectar o elo de predição (ou seja, abrir as chaves S1 e S2) e reiniciar os preditores, transmitindo a informação completa  $y(t)$ .

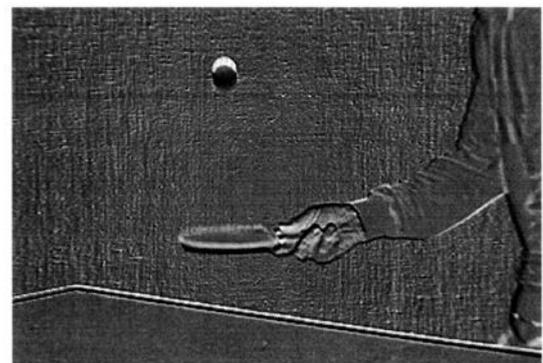
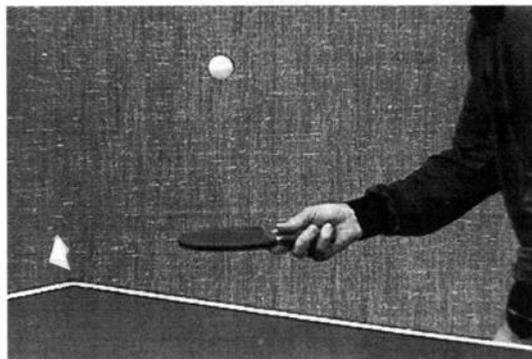
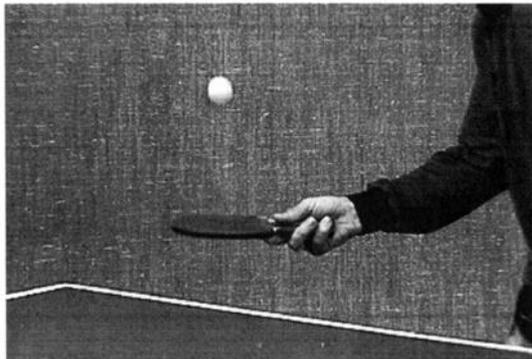


**Fig. 1.1 - Codificador com Preditor**

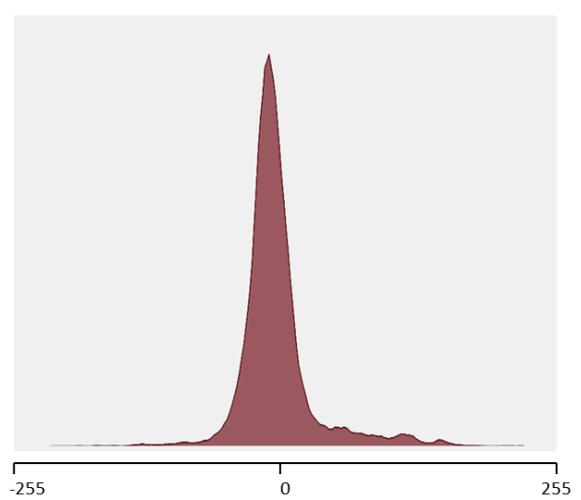
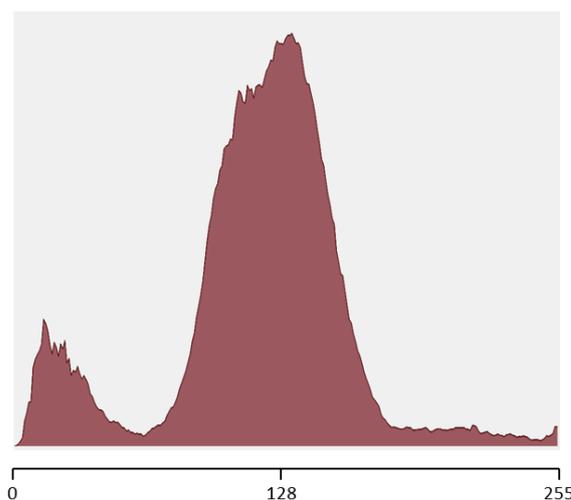
Um preditor simples para aplicações em vídeo seria o "*frame store*", que explora o fato de que um quadro transmitido é geralmente muito semelhante ao quadro anterior. Um preditor deste tipo consiste de uma memória que armazena o quadro anterior, sendo então transmitidas apenas as diferenças *pixel a pixel* entre dois quadros consecutivos (Fig. 1.2). É claro que este preditor falha quando há um corte de cena, ou movimentos rápidos de objetos ou da câmera.

## 2. Preditor para Compensação de Movimento

Um preditor mais eficiente, para aplicação em vídeo, deve procurar compensar os movimentos dos objetos em relação ao campo de visão da câmera. Neste processo, o codificador detecta o deslocamento relativo de partes da imagem entre dois quadros consecutivos, transmitindo essa informação na forma de *vetores de movimento* (deslocamentos com componentes no sentido horizontal e vertical). O preditor, de posse dessas informações, monta uma estimativa da imagem atual baseada em fragmentos tomados a partir de uma imagem de referência, deslocados conforme os vetores de movimento (Figuras 2.1 e 2.4). Normalmente são considerados apenas movimentos de translação.



**Fig. 1.2 – Diferença entre dois quadros sucessivos de uma sequência**



**Histogramas das Imagens e da Diferença**

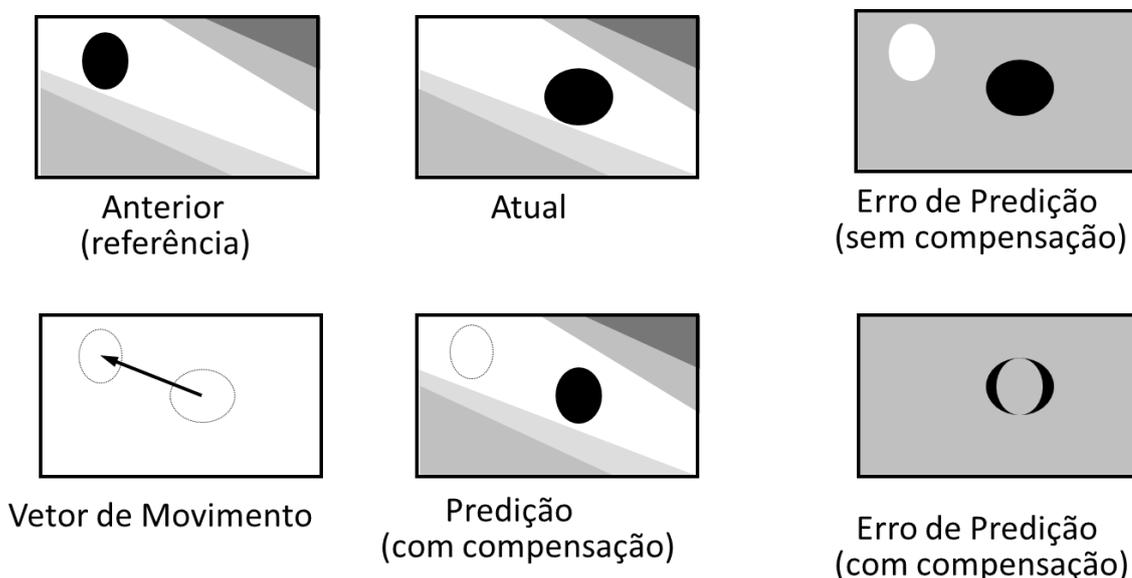
Uma vez que é impraticável determinar quantos objetos com movimentos independentes existem em uma imagem, e qual a sua forma exata, cada fotograma pode ser simplesmente subdividido em vários blocos regulares, sendo então determinados os vetores de movimento para cada bloco.

Ou seja, o que chamamos de “compensação de movimento” é na realidade uma “compensação por translação de blocos semelhantes”.

O *detector de movimento* é a unidade funcional responsável pela determinação dos vetores. Para cada bloco da imagem atual a ser transmitida, o detector faz uma varredura sobre a imagem de referência, buscando o ponto onde ocorre maior semelhança com o bloco a ser analisado. A semelhança pode ser medida pelo erro quadrático médio ou, mais comumente, pela distorção média absoluta (DMA), expressa por

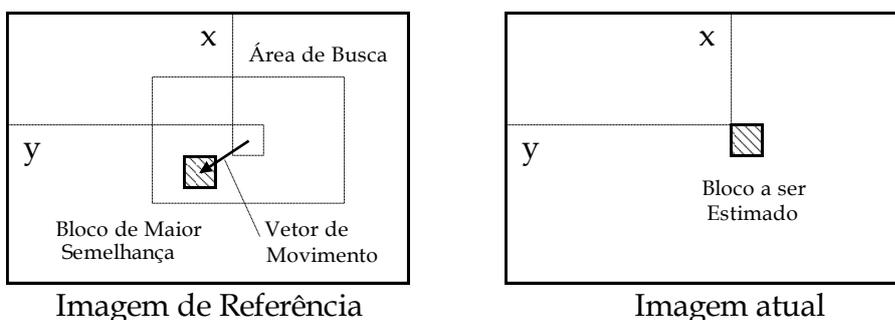
$$DMA(x, y) = \frac{1}{N} \sum_{i,j} |f(x+i, y+j) - ref(x+i+dx, y+j+dy)|$$

onde  $(dx, dy)$  é o deslocamento entre a imagem de referência  $ref(x, y)$  e o bloco da imagem de entrada  $f(x, y)$ . O valor de  $(dx, dy)$  para o qual a DMA é mínima é adotado como sendo o vetor de movimento do bloco de coordenadas  $(x, y)$ .

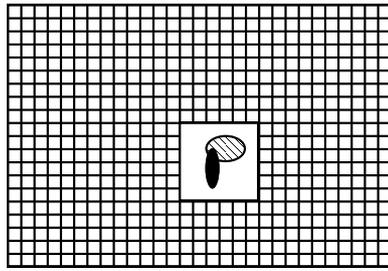


**Fig. 2.1 - Predição com Compensação de Movimento**

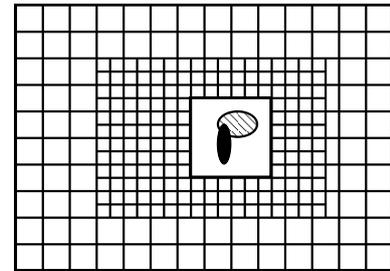
Normalmente a busca é feita dentro de uma área restrita (Fig. 2.2), e/ou utilizando processos hierárquicos (Fig. 2-3), uma vez que uma busca exaustiva sobre toda a imagem representa um esforço computacional elevado. Além disso, a busca pode ser feita com resolução menor do que um pixel, exigindo interpolação da imagem de referência.



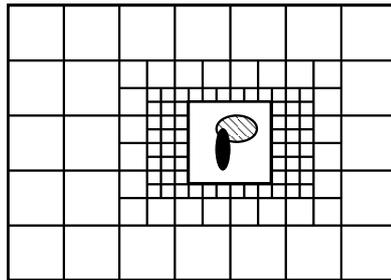
**Fig. 2.2 - Estimador de Movimento**



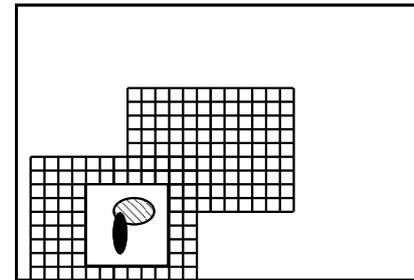
**Busca Exaustiva:** Varre toda a região de busca com resolução de 1/2 pixel.



**Busca Escalonada:** Varre toda a região com resolução de 1 pixel; usa 1/2 pixel em torno do melhor resultado obtido.

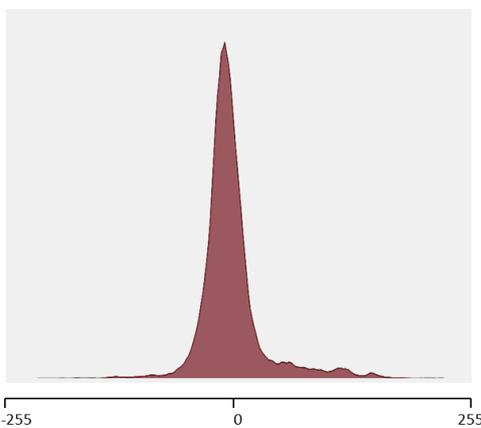
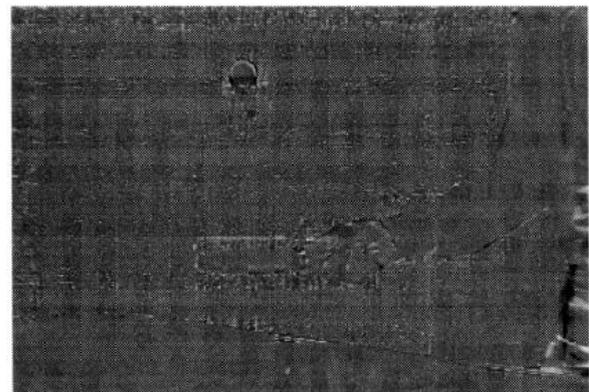
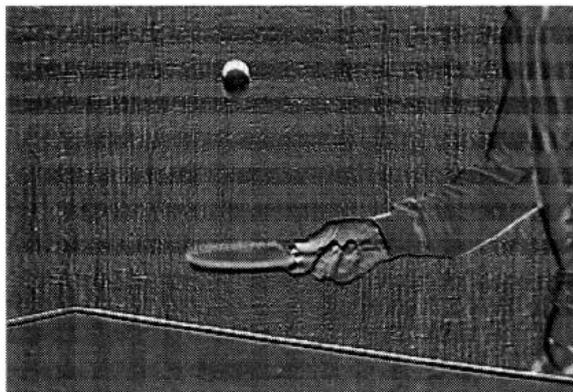


**Busca Hierárquica:** Varre a região com resolução decrescente à medida que a correlação aumenta.

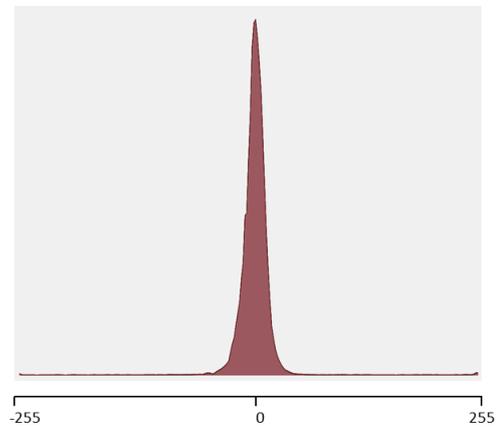


**Busca Recursiva:** Varre sub-regiões em torno do zero e do ponto determinado pelo vetor de movimento correspondente no quadro anterior.

**Fig. 2-3 - Exemplos de Algoritmos de Busca para Detecção de Movimento**



**Sem Compensação de Movimento**



**Com Compensação de Movimento**

**Fig. 2.4 – Erro de Predição e Histogramas Utilizando Compensação de Movimento**

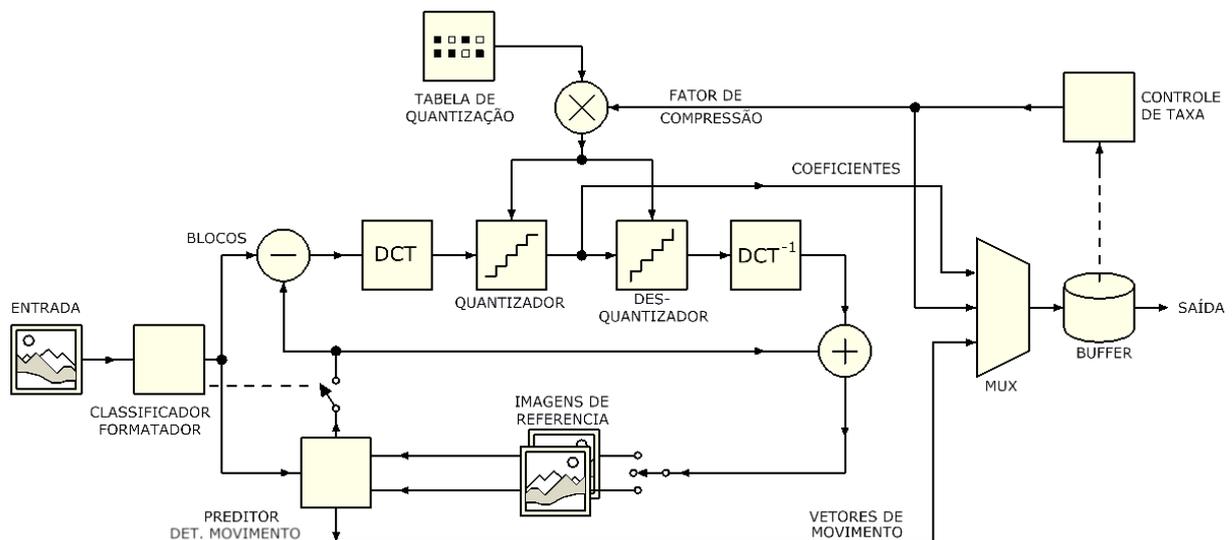
### 3. Codificação com Perdas: MPEG-1

A compensação de movimento reduz consideravelmente a redundância temporal de seqüências de imagens de vídeo. Aplicando ao erro de predição resultante um processo de compressão baseado na percepção visual, semelhante ao JPEG, chegamos a uma arquitetura conveniente de codificador, que remove também as redundâncias espaciais. Isto o torna adequado inclusive para cenas com cortes ou mudanças bruscas de conteúdo.

Processos de compressão sem perdas (compactação de dados), aplicados a sinais de vídeo, conseguem em geral taxas de compressão até 3:1; por outro lado, combinando a compensação de movimento com transformada DCT e quantização, chega-se a taxas de 50:1 ou mais, com baixa visibilidade dos erros de reconstrução. Entre os padrões de compressão que utilizam esta configuração, temos os seguintes:

- **MPEG-1** (*Moving Pictures Experts Group* - ISO/IEC 11172a): destinado à compressão de imagens não entrelaçadas, objetivando taxas de informação até 1,5 Mb/s (aplicações: vídeo-conferência sobre troncos E1 / T1; multimídia em CD-ROM);
- **MPEG-2** (ISO13818): destinado à compressão de imagens de TV entrelaçadas; admite vários níveis de desempenho para taxas de 1,5 a 100 Mb/s (aplicações: TV convencional digital, HDTV);
- **H-261** e **H-263** (ITU): vídeo-conferência através de ISDN (múltiplos de 64 kb/s);
- **H-324** (ITU): vídeo-conferência através de Modem Telefônico;
- **H.264** (ITU): genérico (vídeo-conferência, TV convencional e HDTV);
- **VC-1** (Microsoft / SMPTE): semelhante ao H.264.

A configuração do codificador MPEG-1, embora não especificada explicitamente no padrão, segue a figura 3.1. De modo geral, aplica-se aos demais codificadores citados acima.

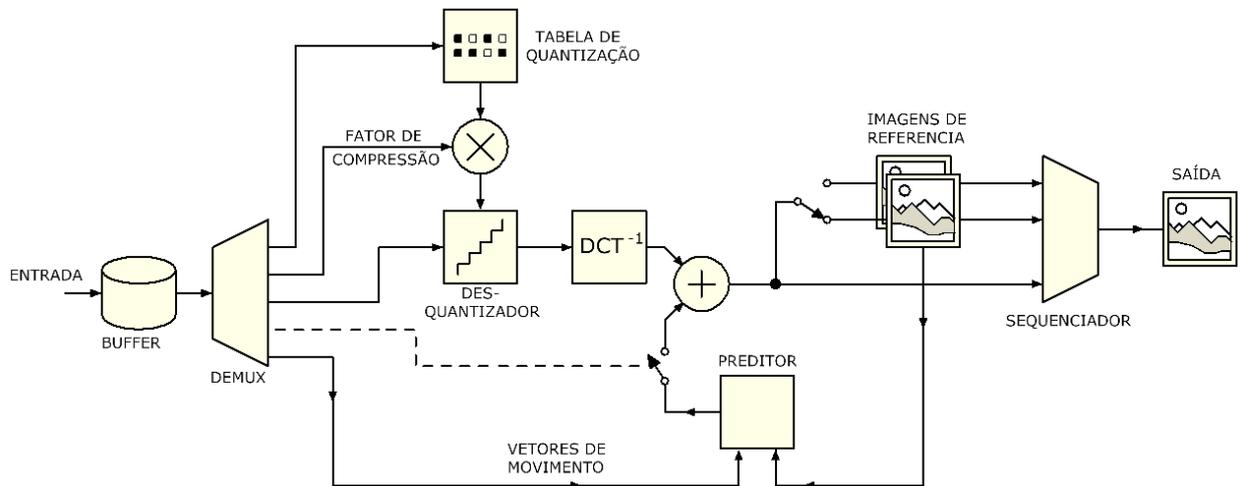


**Fig. 3.1 - Codificador MPEG-1**

Inicialmente, a imagem é formatada (no caso do MPEG-1, é convertida em quadros não-entrelaçados, com resolução típica de 320×240 pixels de luminância e 160×120 pixels de componentes de croma); a seguir, cada plano da imagem (Luminância e componentes de Croma) é subdividido em blocos de 8 x 8 pixels. Os blocos são agrupados em macroblocos, para os quais é feita detecção de movimento, gerando os vetores que serão

transmitidos para o receptor. Os mesmos vetores são fornecidos ao preditor / compensador de movimento, que produz uma estimativa dos blocos a serem codificados, a partir das imagens de referência. Sobre o erro de predição resultante, é aplicada uma transformada DCT, seguida de quantização dos coeficientes. Finalmente, é feita uma compactação dos dados, através de RLE (*Run-Length Encoding*) e codificação Huffman, usando códigos de comprimento variável.

A etapa de quantização é responsável pelas perdas do processo, ou seja, pelos erros de reconstrução. Atuando no fator de escala do quantizador, pode-se controlar a taxa de compressão e a qualidade final da imagem.



**Fig. 3.2 - Decodificador MPEG-1**

### 3.1 Tipos de Imagens

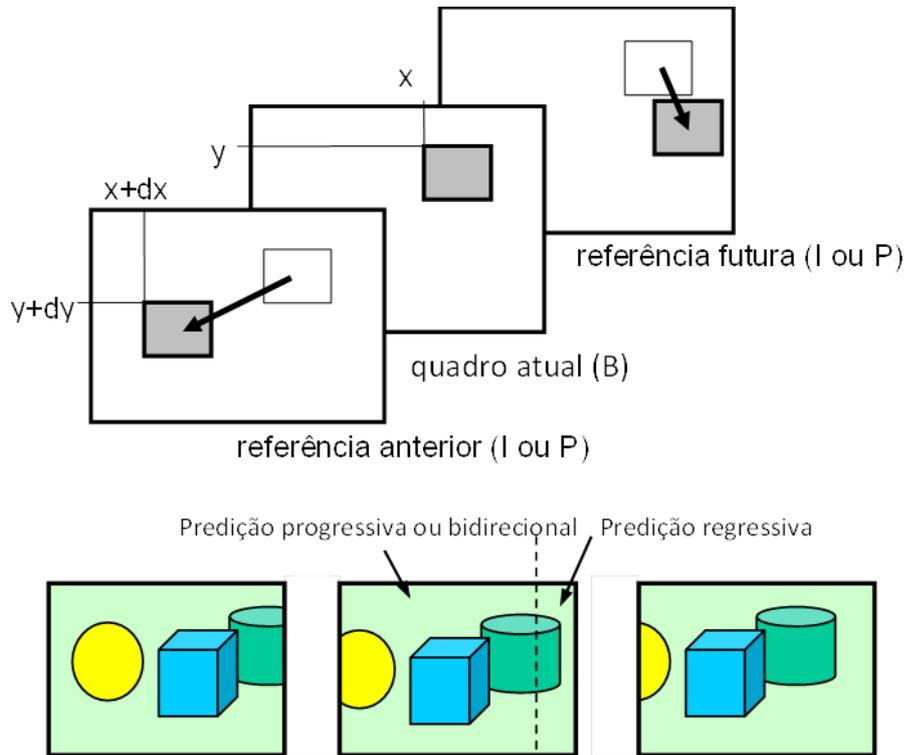
O padrão MPEG-1 define 3 tipos de imagem, conforme o processo usado na compensação de movimento:

**Imagens I** (Independentes, ou *Intra-frame*): são codificadas sem predição de movimento, ou seja, apenas por DCT, quantização e compactação, de forma análoga ao padrão JPEG. São usadas no preditor como imagens de referência para compensação de movimento de quadros futuros. Apresentam menor taxa de compressão efetiva, mas são necessárias para evitar propagação de erros, para inicializar os preditores ou quando há cortes de cenas;

**Imagens P** (com predição progressiva de movimento): são reconstruídas através de predição de movimento, baseando-se em imagens de referência anteriormente codificadas, que podem ser imagens tipo **I** ou tipo **P**. Apresentam taxa de compressão elevada;

**Imagens B** (com predição bidirecional, fig. 3.3): o preditor baseia-se em duas imagens de referência (anterior e posterior, tipo **I** ou **P**); admitem até dois conjuntos de vetores de movimento (progressivos e regressivos) para cada bloco, sendo que neste caso a estimativa adotada é a média das estimativas individuais. Apresentam a maior taxa de compressão dentre os 3 tipos de imagem.

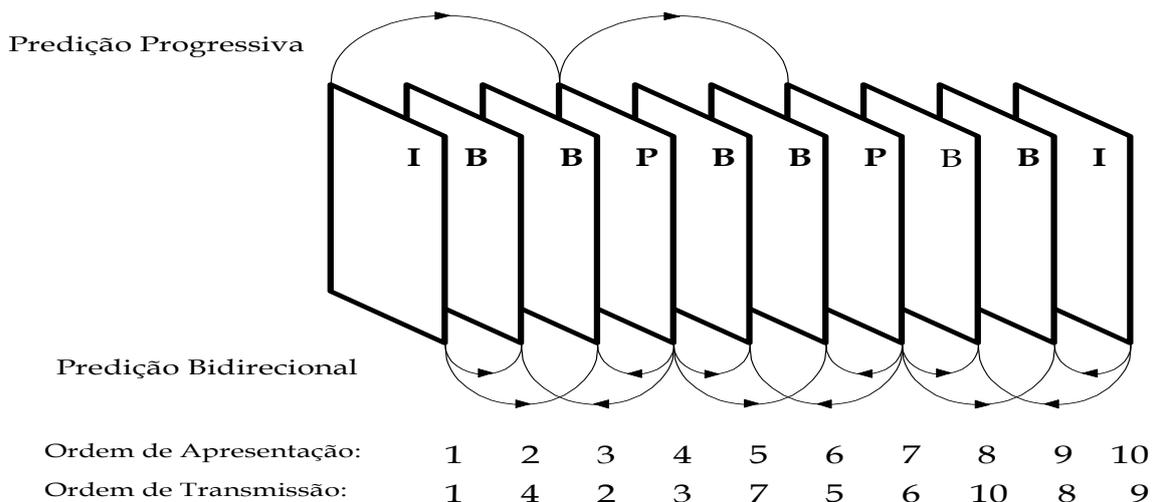
Além desses tipos, o padrão MPEG prevê ainda Imagens **D**, compostas apenas pelos coeficientes DC das transformadas DCT de cada bloco. Estas imagens (raramente usadas) possuem baixa resolução e podem ser usadas como "mostruário" num eventual mapa de programas ou como imagens "Picture in Picture" (PIP).



**Fig. 3.3 – Predição Bidirecional (Imagem tipo B)**

### 3.2 Grupos de Imagens (GOP)

Uma sequência de vídeo é composta por **Grupos de Imagens** (GOP's - Groups of Pictures). Um GOP (que pode conter imagens tipo **I**, **P** e/ou **B** em várias proporções) é **fechado** se as previsões de movimento das suas imagens são efetuadas sem necessitar de quadros de referência externos ao Grupo; desta forma, sequências de vídeo podem ser editadas (montadas) tomando-se como pontos de corte os inícios destes GOPs.



**Fig. 3.4 - Grupo de Imagens em MPEG-1**

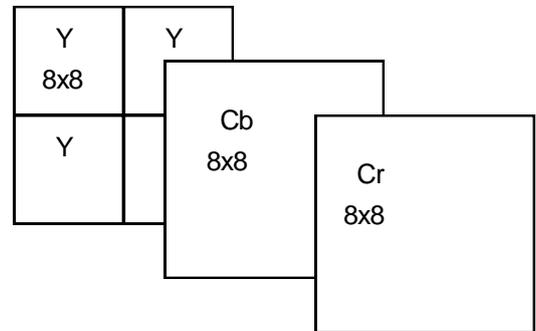
A estrutura do GOP pode ser caracterizada pelos parâmetros **M** (distância entre imagens tipo **I**) e **N** (distância entre imagens **P**). Na figura 3.3 temos um GOP aberto, numa configuração típica **IBBPBBPBB** com **M=9**, **N=3**. Note que a sequência efetiva de transmissão é diferente da

sequência de exibição das imagens, uma vez que o receptor necessita das imagens **I** e **P** (1 e 4) para reconstruir as imagens **B** intermediárias (2 e 3).

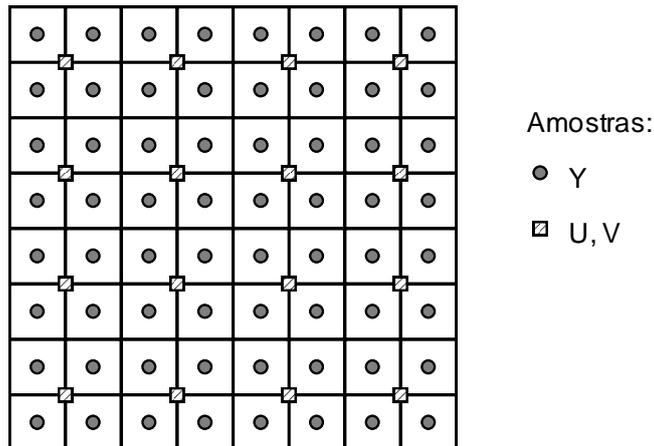
O tamanho do GOP influencia também o tempo de aquisição de uma sequência, uma vez que o receptor tem que aguardar uma imagem **I** para poder decodificar as demais imagens; portanto, a escolha da proporção de imagens **I**, **B** e **P** é um compromisso entre taxa de bits disponível, qualidade final da imagem, tempo de aquisição e recuperação de erros de transmissão. É comum adotar-se uma imagem **I** a cada 15 quadros (1/2 segundo).

### 3.3 Macroblocos

O elemento básico de predição de movimento é o macrobloco, que consiste de um conjunto de 4 blocos de 8×8 amostras de luminância, e 2 blocos de crominância correspondentes (figura 3.5). As componentes de crominância **Cb** (B-Y) e **Cr** (R-Y) são sub-amostradas na vertical e na horizontal conforme a estrutura da figura 3.5, num padrão conhecido como "4:2:0", de modo que cada bloco de crominância cobre a mesma área que os 4 blocos de luminância.



**Fig. 3.5 - Estrutura do Macrobloco MPEG**



**Fig.3.6 - Estrutura de Amostragem "4:2:0"**

Dentro de uma imagem **P** ou **B**, cada macrobloco tem ainda a possibilidade de ser codificado independentemente (ou seja, sem usar predição de movimento).

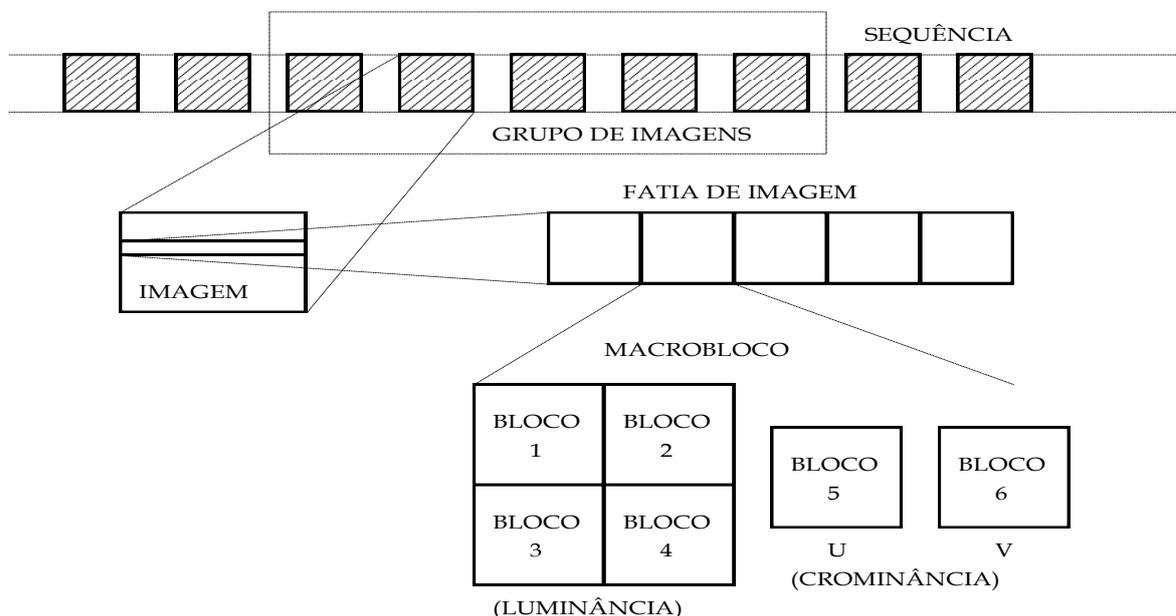
## 4. Sintaxe MPEG-1 Vídeo

O padrão MPEG não especifica detalhadamente as arquiteturas do codificador e do decodificador; apenas define a sintaxe do fluxo de dados e as ações que o decodificador deve tomar conforme os dados recebidos. Isso deixa margem para diferentes implementações do codificador, com diferentes níveis de desempenho.

Os dados MPEG referentes ao vídeo são estruturados numa hierarquia, conforme a figura 4.1, composta dos seguintes elementos:

- **Bloco:** conjunto de 8×8 amostras (pixels);

- **Macrobloco:** conjunto de 4 blocos de Luminância e 2 blocos de Crominância; estes blocos compartilham os mesmos vetores de movimento, tipo de codificação e fator de escala alternativo do quantizador;
- **Fatia (slice):** agrupamento horizontal de macroblocos; estes macroblocos compartilham o mesmo fator de quantização básico. Como são codificadas independentemente e possuem códigos de início facilmente identificáveis, limitam a propagação de erros de transmissão;
- **Imagem:** quadro completo (**I**, **P** ou **B**), composto de Fatias consecutivas; os blocos dentro de uma imagem compartilham a mesma resolução dos vetores de movimento e o mesmo tipo de imagem;
- **Grupo de Imagens (GOP):** composto de imagens **I**, **P** e/ou **B**: esta é a célula básica de predição de movimento e edição de sequências;
- **Sequência:** um trecho de vídeo (programa, segmento de imagens); composto de GOP's que compartilham a mesma resolução espacial/temporal e eventuais matrizes alternativas de quantização.

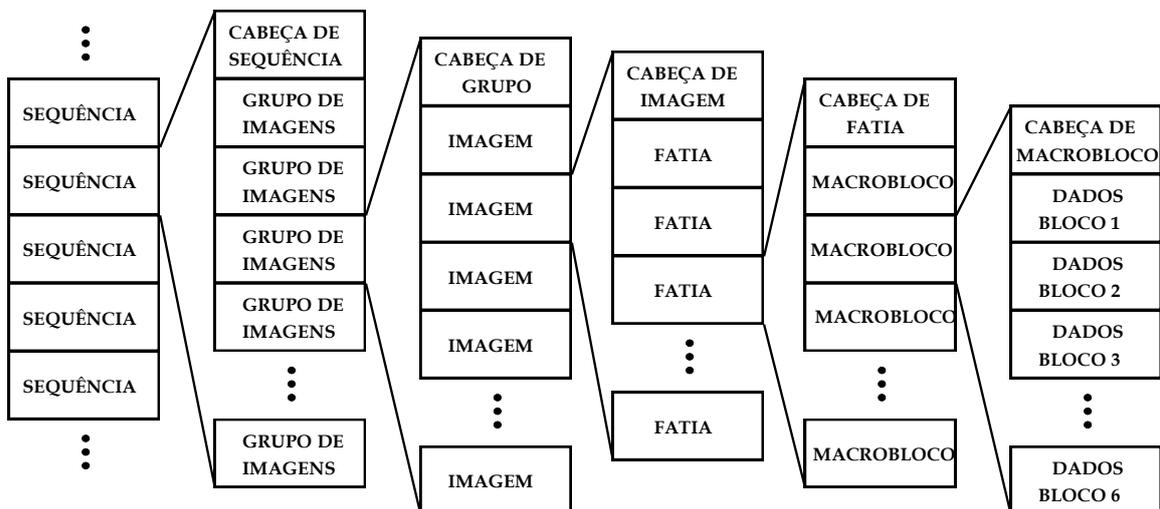


**Fig. 4.1 - Hierarquia dos Elementos do Vídeo MPEG**

Na sintaxe do fluxo de dados MPEG, os vários níveis são delimitados por cabeçalhos (*Headers*). Os códigos de identificação (prefixos, ver tabela 4.1) dos cabeçalhos contêm sequências de pelo menos 23 bits "zero", alinhadas a nível de bytes. Estas sequências nunca são confundidas com possíveis sequências de dados, o que permite sua identificação mesmo na ausência de outras formas de sincronismo do processo de decodificação.

A figura 4.2 esquematiza a estrutura do fluxo de dados MPEG-1 (bit stream) referente à informação de vídeo digital.

A seguir serão descritos os cabeçalhos e os elementos da sintaxe na codificação MPEG-1 para vídeo. As tabelas de codificações referidas abaixo se encontram no Apêndice I.



**Fig. 4.2 - Estrutura do Bit Stream MPEG-1**

Prefixo:	Código Hexadecimal:
Início de Sequência	00 00 01 B3
Início de Grupo de Imagens	00 00 01 B8
Início de Imagem	00 00 01 00
Início de Fatia (linha 1 até 175)	00 00 01 01 até 00 00 01 AF
Início de Extensão	00 00 01 B5
Fim de Sequência	00 00 01 B7
Início de Área de Dados do Usuário	00 00 01 B2

**Tabela 4.1 - Códigos de Prefixo dos Cabeçalhos MPEG**

#### 4.1 Sequência

##### Cabeçalho de Sequência:

Código de Início de Sequência (00 00 01 B3 h)
Resolução Horizontal (12 bits)
Resolução Vertical (12 bits)
Relação de Aspecto (4 bits - Tabela 1)
Taxa de Frames por segundo (4 bits - Tabela 2)
Taxa de Bits por segundo (18 bits, múltiplo de 400 bps)
Tamanho de Buffer necessário (10 bits, mult. de 2 kBytes)
Matriz de Quantização (Opcional) (Tabela 3 )
Matriz de Quantização Intra-Frame (64 valores)
Matriz de Quantização Inter-Frame (64 valores)
Código de Início de Extensões (Futuras) (00 00 01 B5 h)
Código de Área de Dados (Opcionais) (00 00 01 B2 h)

A sequência inicia com informações sobre resolução, relação de aspecto do pixel, tamanho do buffer necessário para decodificação, e eventualmente matrizes de quantização opcionais. Caso a **Taxa de Bits** seja variável, é codificada como 03 FF FFh.

O **código de Extensão** é usado em sequências MPEG-2.

---

## 4.2 Grupo de Imagens (GOP)

### Cabeçalho de Grupo:

Código de Início de Grupo de Imagens (00 00 01 B8 h)
Time Code (25 bits - Tabela 4)
Flag de Grupo Aberto / Fechado
Flag de Sequência Interrompida
Código de Extensões (Futuras) (00 00 01 B5 h)
Código de Área de Dados (Opcionais) (00 00 01 B2 h)

Para cada GOP, temos o **Time Code** (horas, minutos, segundos e quadros) correspondente à primeira imagem a ser exibida, contado a partir do início da sequência (GOP com **Time Code** igual a 0). Desta forma o decodificador pode aguardar o instante exato em que deve iniciar a apresentação de cada imagem. O GOP é fechado se pode ser decodificado independentemente de imagens **I** ou **P** pertencentes a outro GOP. O flag de **Sequência Interrompida** indica que houve edição (corte) da sequência e algumas imagens **B** seguintes podem não ser decodificáveis por dependerem de uma imagem de referência não disponível.

## 4.3 Imagem

### Cabeçalho de Imagem:

Código de Início de Imagem (00 00 01 00 h)
Referência Temporal (10 bits)
Tipo de Imagem (3 bits - Tabela 5)
Tempo Previsto para Encher Buffer (16 bits)
* Flag de Precisão dos Vetores de Movimento (1 ou 0,5 pixels)
* Código dos Vetores de Movimento (3 bits, Tabela 6)
** Flag de Precisão dos Vetores Regressivos (1 ou 0,5 pixels)
** Código dos Vetores Regressivos (3 bits, Tabela 6)
Código de Extensões (Futuras) (00 00 01 B5 h)
Código de Área de Dados (Opcionais) (00 00 01 B2 h)
* - Apenas para Imagens P ou B
** - Apenas para Imagens B

No início de cada imagem do GOP há um campo que apresenta uma contagem incremental (**referência temporal**), que é usada para reordenar as imagens **I**, **P** e **B** para exibição na sequência correta. Para imagens **P** e **B**, são especificadas as precisões dos vetores de movimento (1 ou 1/2 pixel) e o respectivo alcance.

## 4.4 Fatia ("Slice")

### Cabeçalho de Fatia:

Código de Início de Fatia de Imagem (00 00 01 01 até 00 00 01 AF h)
Fator de Escala do Quantizador "q" (5 bits)
Flag de Informação Extra
Informação Extra (Opcional) (bytes)

O último Byte do prefixo de Início de Fatia indica a posição vertical, ou seja, em qual linha de macroblocos que o primeiro macrobloco da fatia está posicionado. A seguir consta o fator de escala  $q$  do quantizador (fator multiplicativo da tabela de quantização, válido apenas para os coeficientes AC da DCT), utilizado na quantização dos coeficientes AC da fatia. Os valores  $F_{x,y}$  reconstruídos no receptor são obtidos por:

$$F_{x,y} = \begin{cases} DC \times 8 & \text{para } x = 0; \quad y = 0 \\ \frac{AC_{x,y} \times q \times Q_{x,y}}{8} & \text{demais casos} \end{cases}$$

Além disso, os vetores de movimento são zerados no início de cada fatia. Dentro de cada fatia, os vetores de movimento são calculados de forma diferencial, em relação aos vetores do macrobloco  $(n-1)$  anterior:

$$Mov(n) = Mov(n-1) + (BASE \times FATOR \pm RESIDUAL) \times k$$

onde **BASE** e **RESIDUAL** constam do Cabeçalho de Macrobloco; **FATOR** (alcance dos vetores) e **k** (precisão) constam do Cabeçalho de Imagem.

#### 4.5 Macrobloco

##### Cabeçalho de Macrobloco:

Enchimento (11 bits = 000 0000 1111 b)
Incremento de Endereço de Macrobloco ( 1 a 11 bits - Tabela 7)
Tipo de Macrobloco (Tabela 8)
* Fator de Escala de Quantização Alternativo (5 bits)
* Vetores de Movimento Progressivos:
Horizontal (Base e Residual)
Vertical (Base e Residual)
* Vetores de Movimento Regressivos:
Horizontal (Base e Residual)
Vertical (Base e Residual)
* Padrão de Codificação de Blocos (Tabela 10)
Coeficientes DC e AC dos blocos codificados
* Presentes de Acordo com os Flags Correspondentes do Tipo de Macrobloco (Tabela 8)

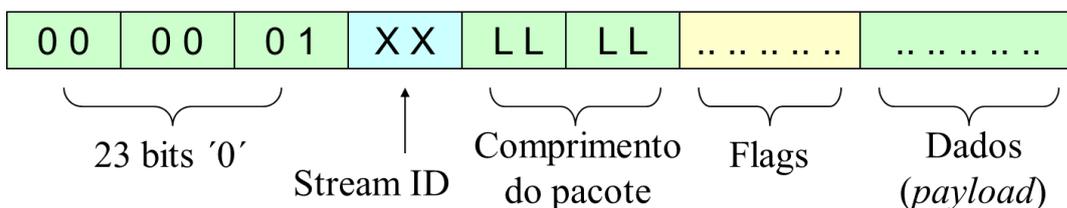
O campo de **Incremento de Endereço** especifica a posição relativa do macrobloco em relação ao último macrobloco codificado. Por exemplo, quando todos os coeficientes de um macrobloco de uma imagem **P** ou **B** forem zero, este macrobloco não precisa ser codificado e pode ser saltado. No entanto, o primeiro e o último macroblocos de uma fatia devem sempre ser codificados. No caso de imagens tipo **I**, todos os macroblocos são sempre codificados.

O **tipo de macrobloco** é um campo de comprimento variável que especifica se a codificação (a nível de macrobloco) é Intra (sem predição de movimento), ou com predição progressiva, regressiva ou bidirecional; se utiliza um fator de escala de quantização alternativo; e se alguns blocos individuais são descartados segundo um padrão de codificação. Em seguida vêm os vetores de movimento para o macrobloco, e os coeficientes da DCT codificados por RLE e Huffman (apenas para os blocos habilitados pelo padrão de codificação).

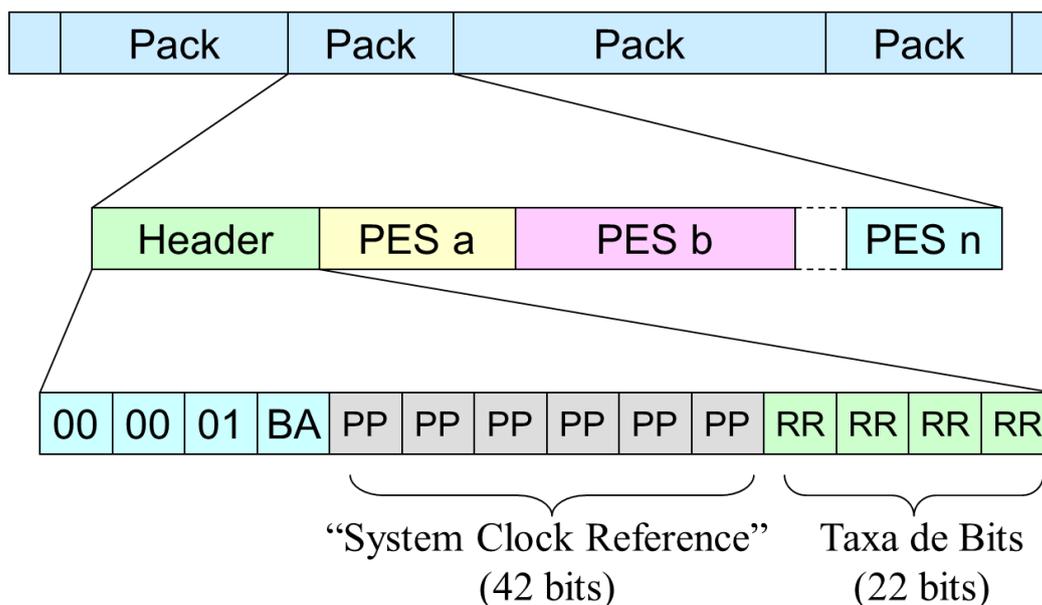
## 5. Transporte de Fluxos de Dados MPEG

O padrão MPEG prevê métodos para combinar (multiplexar) informações codificadas de áudio e vídeo, de modo que os respectivos decodificadores possam reconstruir os sinais de forma sincronizada, em tempo real. Essa multiplexação, descrita na norma ISO/IEC 11172 - Parte 1, baseia-se na subdivisão dos fluxos de dados de áudio e vídeo em pacotes, da seguinte forma:

1. Os codificadores de áudio e vídeo produzem sequências próprias de dados, denominadas **ES** (*Elementary Streams*, ou fluxos elementares), que podem ter taxas de informação fixas ou variáveis;
2. Os fluxos ES são subdivididos em pacotes de comprimento fixo ou variável, formando fluxos que são denominados de **PES** (*Packetized Elementary Streams*). Estes pacotes, que em geral ocupam 2 kBytes a 64 kBytes, são delimitados por códigos de início (*Start Code ID*). A figura 4.3 apresenta a estrutura de um pacote **PES**; os códigos de início estão listados na Tabela 4.2. Os bits de “Flags” incluídos no cabeçalho dos pacotes incluem informações referentes a criptografia, taxa de bits, CRC e sincronismo temporal. Este sincronismo é dado na forma de “*Time Stamps*”, ou referências temporais, que indicam ao decodificador qual é o instante relativo em que a informação (vídeo ou áudio) deve ser decodificada.
3. Pacotes **PES** de áudio, vídeo e outros tipos de dados relacionados são combinados em um fluxo de dados denominado **PS** (*Program Stream*). O **PS** consiste de uma sequência de “Packs” (figura 4.4), que por sua vez são sequências de pacotes **PES** de áudio, vídeo etc. Cada “Pack” é precedido de um cabeçalho que traz uma referência temporal (**SCR** – *System Clock Reference*) e um campo que indica a taxa de bits do **PS**. O **SCR** consiste de um contador de 33 bits, incrementado à taxa de 90 kHz, sendo que o decodificador deverá implementar um contador correspondente, que deve ser mantido em sincronismo com o **SCR**. A taxa de bits do **PS** é dada em múltiplos de 50 bps.



**Fig. 4.3 – Estrutura de um Pacote PES**



**Fig. 4.4 – Estrutura do “Program Stream” (PS)**

XX	Tipo de Fluxo de Dados
B9	Fim de Packet Stream
BA	Pack Header
BB	Header de Sistema
BC	Tabela de Mapa de Programas (PS-PMT)
BD, BF	Fluxo de Dados Privado
BE	Enchimento
C0 ~ DF	Áudio MPEG-1 ou MPEG-2
E0 ~ EF	Vídeo MPEG-1 ou MPEG-2

**Tabela 4.2 – Códigos de Início “Start Code ID” MPEG**

## 6. Codificação MPEG-2: Níveis e Perfis de Desempenho

O padrão MPEG-2, definido em 1995/96 conjuntamente pela ITU (H.222 / H.262) e pela ISO/IEC (13818), tem como objetivo a codificação de sinais de vídeo para aplicações genéricas, desde multimídia e vídeo-conferência, até TV de Alta Definição (HDTV) para aplicações em radiodifusão ou estúdio. Especificamente, inclui recursos para codificação eficiente de imagens entrelaçadas.

Dentro desse contexto, o padrão define 4 "Perfis", relacionados com os recursos de processamento e complexidade dos codificadores, e 4 "Níveis", relacionados com o formato de vídeo e taxa de bits resultante:

Perfil	Recursos Adicionais
Simples (SP)	Nenhum (Sistema Mínimo)
Principal (MP)	Predição bidirecional ( Quadros tipo B)
Escalável em SNR (SNRP)	Codificação Hierárquica com níveis diferentes de prioridade para imagem básica e detalhes
Escalável Espacial (SSP)	Codificação Hierárquica com níveis diferentes de prioridade para imagem 4 × 3 e 16 × 9
Alto (HP)	Todos os recursos e codificação 4:2:2 (Dobro de amostras de Crominância)

Nível	Formato de Vídeo	Taxa de Bits
Baixo (LL)	240 Linhas × 360 Pontos ( qualidade VHS)	~1,5 Mb/s
Principal (ML)	480 Linhas × 720 Pontos ( CCIR-601 - Qualidade Estúdio p/ TV Convencional)	4 ~ 6 Mb/s
Alto-1440 (H14L)	1080 Linhas × 1440 Pontos (Formato HDTV)	20 ~ 60 Mb/s
Alto (HL)	1080 Linhas × 1920 Pontos (Formato HDTV - Qualidade Estúdio)	20 ~ 100 Mb/s

**Tabela 5.1 – Níveis e Perfis MPEG-2**

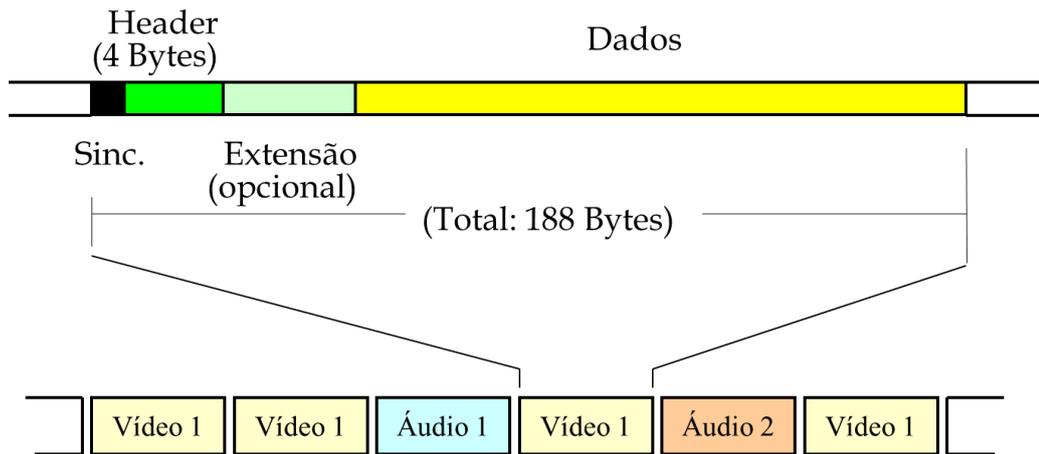
Por exemplo, a combinação **SP@LL** corresponde a um codificador com desempenho equivalente ao MPEG-1 para multimídia; já **MP@ML** é um formato adequado para TV convencional entrelaçada, atualmente usado em sistemas como Sky, DirecTV, DVB e na codificação de vídeo no DVD (“Digital Versatile Disc”). O formato **MP@HL** é adequado para radiodifusão de HDTV, usado no Japão e EUA.

Outro requisito atendido pelo MPEG-2 foi contemplar a transmissão através de canais sujeitos a erros; para isso, definiu-se um Fluxo de Transporte na forma de Pacotes de Transporte (**TS – Transport Stream**) de comprimento fixo, igual a 188 bytes. Os **TS** carregam fragmentos de áudio, vídeo ou dados, possuindo um cabeçalho (figura 5.1) que traz informações de sincronismo e criptografia, além de um campo de identificação de programa (**PID**). O tamanho do pacote é adequado para proteção eficiente através de códigos corretores de erros, e admite formatação em células ATM (pacotes de 48 bytes). Na falta de informação a ser transmitida, o Fluxo de Transporte é preenchido com *pacotes nulos*, que são identificados pelo PID = 1 1111 1111 1111 (0x1FFF em hexadecimal).

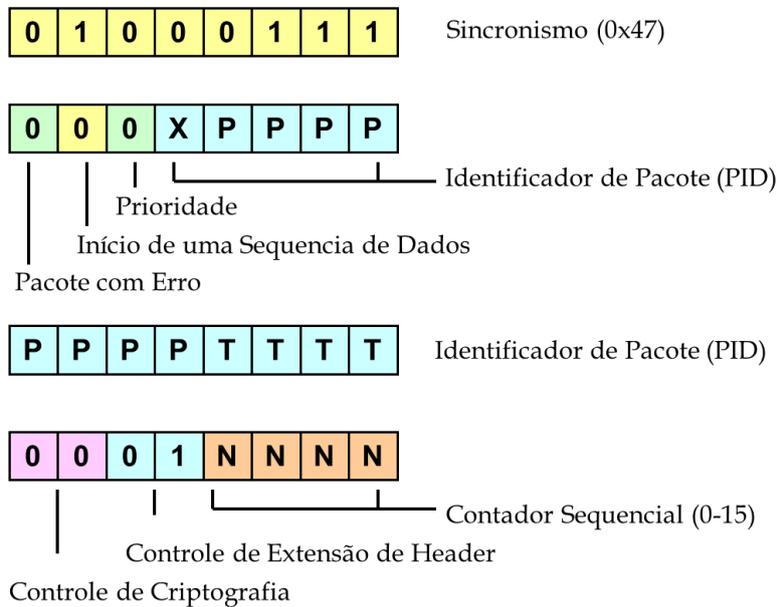
A formatação em **TS** provê compatibilidade de meios de transmissão e compatibilidade de serviços (vídeo, áudio multi-lingual, legendas, dados etc).

O fluxo de transporte pode assim carregar vários programas independentes, que compartilham apenas o mesmo canal de comunicação. A figura 5.2 mostra um esquema de multiplexação de pacotes, no qual informações de áudio, vídeo e dados (legendas, informações auxiliares, etc) são reunidas em um fluxo de programa. Vários fluxos de programa são a seguir reunidos em um fluxo de transporte, que inclui ainda pacotes de Mapa de Programação e de Informações de Sistema (**SI - System Information**). Estes últimos trazem tabelas referentes aos sistemas de transmissão e modulação usados, bem como mapas de canais virtuais e outras informações relevantes à rede de distribuição.

Fluxo de Transporte:



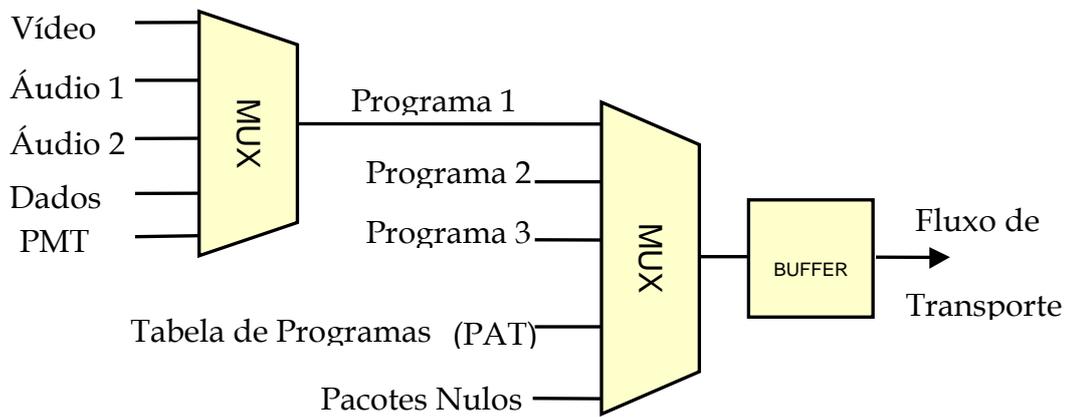
Estrutura do Header (4 bytes):



Descrição do PID:

x	Programa TV / Outros
pppppppp	Número do Programa (1 - 255)
t t t t	Tipo de Pacote: 0h = Mapa de Programa 1h = Vídeo            4h = Áudio Principal 5h = Áudio Secundário    Ah = Dados

**Figura 5.1 - Pacote de Transporte MPEG-2 (TS)**



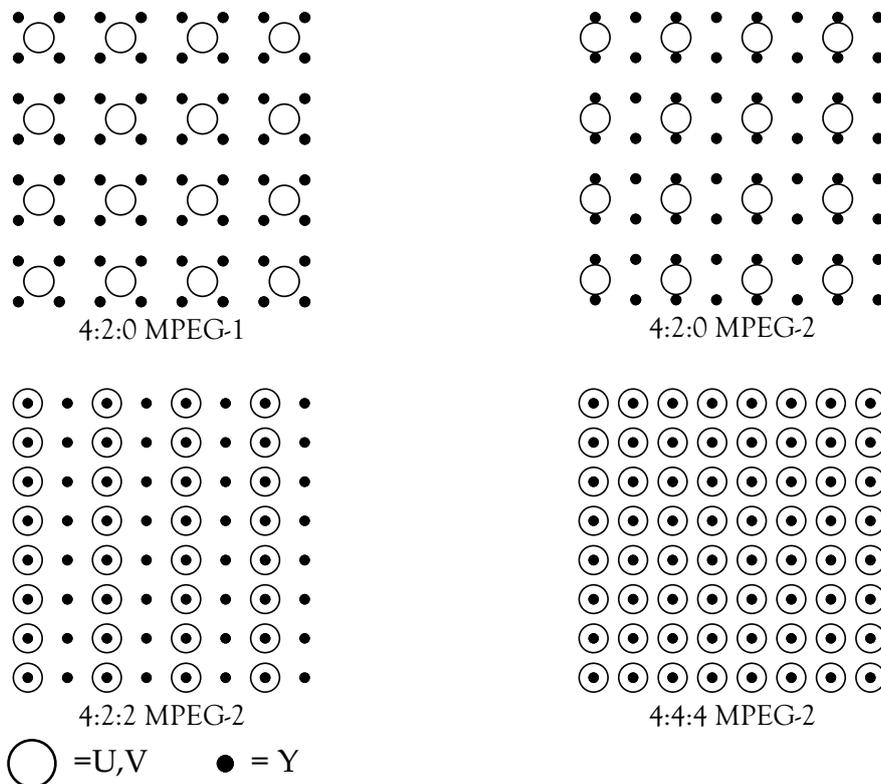
**Fig. 5.2 - Multiplexação de Programas – Fluxo de Transporte MPEG-2**

Para o caso de transmissão por um canal de taxa fixa, o buffer de saída amortece as flutuações estatísticas de taxa de bits de cada programa, e o ajuste dos quantizadores nos codificadores de vídeo pode ser feito em função do grau de ocupação deste buffer. Eventualmente, pacotes nulos são acrescentados ao TS para manter a taxa especificada.

A seguir serão descritas as principais diferenças entre a codificação MPEG-2 e MPEG-1, no que diz respeito à compressão de vídeo.

### 6.1 Estruturas de Amostragem:

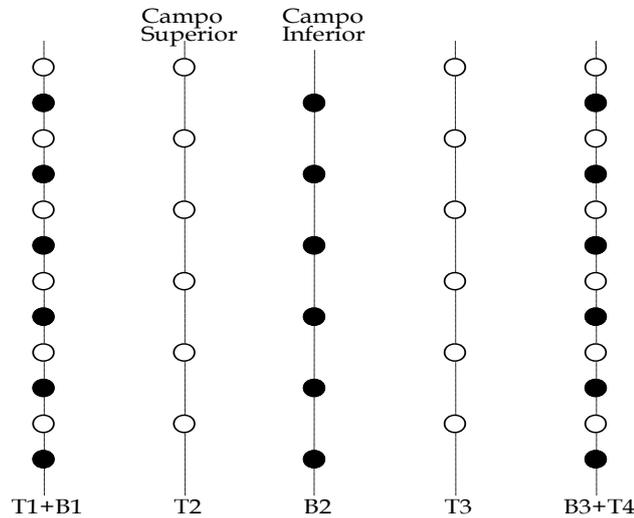
O padrão MPEG-2 admite amostragem de componentes nos formatos "4:2:2" (similar ao CCIR-601) e "4:4:4" (para aplicações especiais). Além disso, no formato "4:2:0" o alinhamento das amostras de crominância é diferente do adotado no MPEG-1 (figura 5.3). Para a estrutura 4:2:2 os macroblocos contêm 4 blocos de luminância e 2+2 blocos de crominância.



**Fig. 5.3 - Estruturas de Amostragem MPEG-1 e MPEG-2**

## 6.2 Imagens de Campo e de Quadro:

Em MPEG-2, as imagens podem ser codificadas como Imagens de Quadro (não entrelaçadas) ou Imagens de Campo (entrelaçadas); os Campos são identificados como Superior (Top) e Inferior (Bottom), sendo que uma imagem de Quadro é composta de dois campos consecutivos, independentemente da ordem (T1+B1 ou B1+T2). Assim sendo, os tipos de imagens codificáveis passam a ser **IQ**, **IC**, **PQ**, **PC**, **BQ** e **BC**. A figura 5.4 mostra uma sequência de duas imagens de Quadro, com 3 imagens de Campo interpostas.



**Fig. 5.4 - Imagens de Quadro e de Campo**

## 6.3 Predição de Movimento

Havendo entrelaçamento, as linhas adjacentes de um quadro completo procedem de campos diferentes, amostrados em instantes diferentes. Neste caso, a existência de movimento causa redução de correlação entre os pixels pertencentes a campos opostos de um mesmo quadro, mesmo em linhas adjacentes. Considerando este fato, além de usar predição de imagens de Quadro para imagens de Quadro (similar à predição de movimento do MPEG-1), o MPEG-2 admite outros modos de predição, tomando como referência imagens **IQ**, **PQ** ou pares de imagens **IC** ou **PC** :

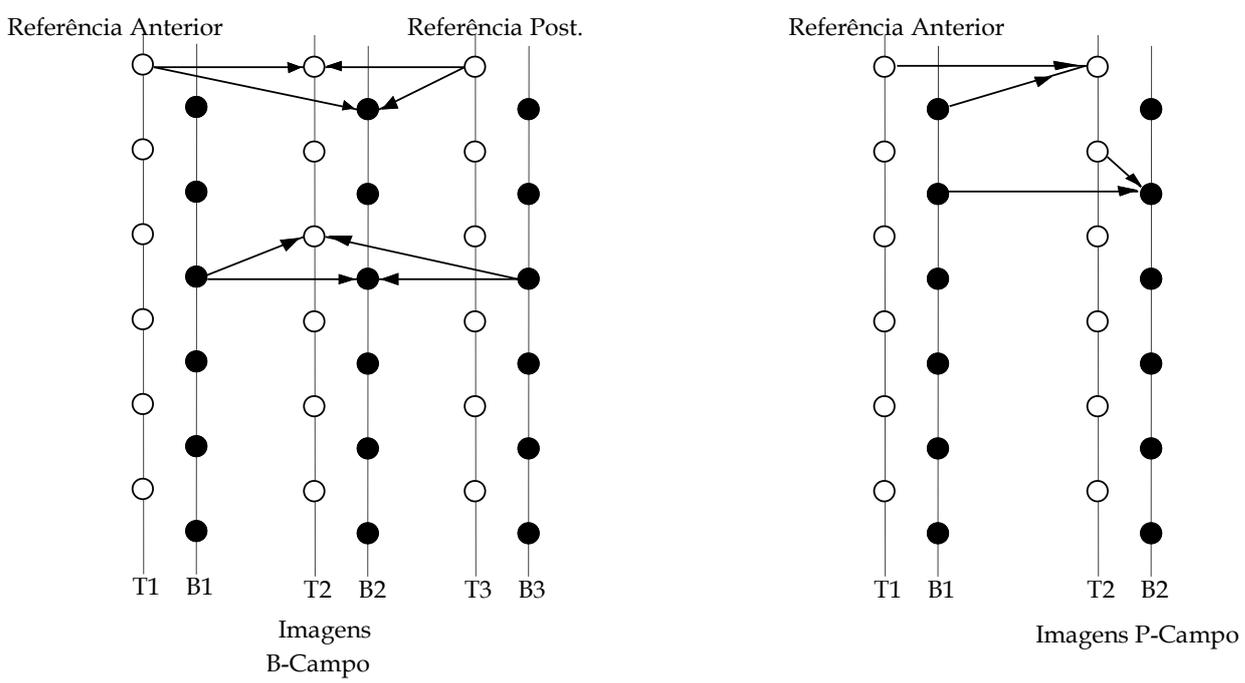
- Predição de Imagens de Campo para Imagens de Campo (figura 5.5): Os vetores de movimento (1 para imagens **PC** e 2 para imagens **BC**) podem ser provenientes de campos com mesma paridade ou opostos ao campo sendo codificado. Isso significa que a busca para detecção de movimento deve em princípio ser efetuada para os dois campos de referência (4 no caso de imagens **BC**) e então escolhido o campo que proporcionar menor erro de predição. Nas imagens **PC** a predição do segundo campo pode ser feita tomando como referência o primeiro campo da mesma imagem.
- Predição de Imagens de Campo para Imagens de Quadro (figura 5.6): neste processo, uma imagem tipo **PQ** é prevista a partir de 2 campos de referência anteriores, e uma imagem **BQ** a partir de 4 campos de referência (imagens **IC** e/ou **PC**).

Neste caso os macroblocos da imagem a ser codificada são subdivididos em dois sub-macroblocos  $16 \times 8$ , sendo que para cada um será enviado um vetor de movimento independente (2 no caso de imagens **BQ**), que pode originar-se de qualquer um dos campos da imagem de referência.

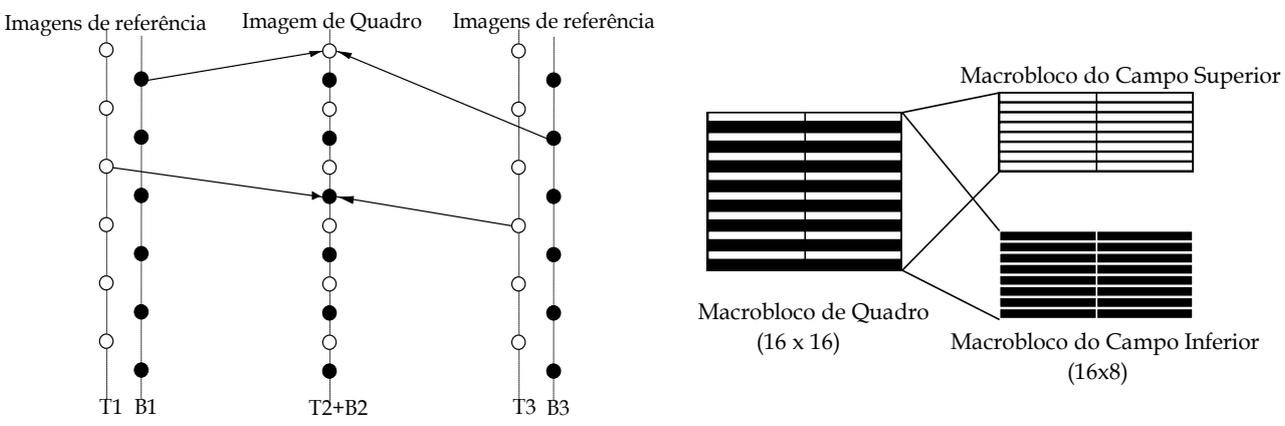
- Predição "Dual-Prime" para imagens **P** (Figura 5.7): este método transmite apenas um vetor de movimento por macrobloco, sendo que este vetor é adotado para os dois campos.

Para cada um dos campos são feitas duas previsões: uma usando o vetor transmitido, baseando-se no campo de referência de mesma paridade, e outra extrapolando ou interpolando o vetor de movimento sobre o campo de polaridade oposta. Os vetores assim calculados são corrigidos somando-se um vetor de movimento diferencial, transmitido juntamente com o vetor de movimento principal, porém limitado em  $\pm 1/2$  pixel. É feita então a média das duas previsões obtidas. Este método proporciona alta taxa de compressão para imagens entrelaçadas com movimento moderado.

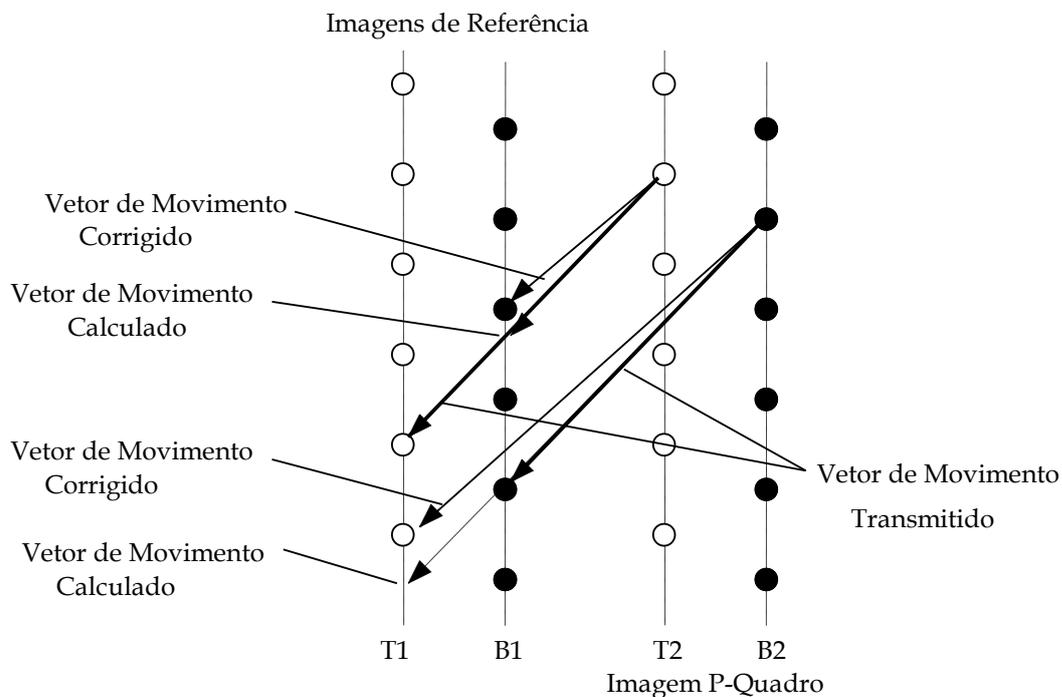
d) **Predição 16x8 para Imagens de Campo:** neste processo cada macrobloco de uma imagem tipo Campo é dividido horizontalmente em dois blocos 16x8, e é determinado um vetor de movimento para cada uma das partes (2 para imagens **BC**). Este método é útil para imagens de Campo com movimento altamente irregular.



**Fig. 5.5 - Predição de Movimento de Campo para Campo**



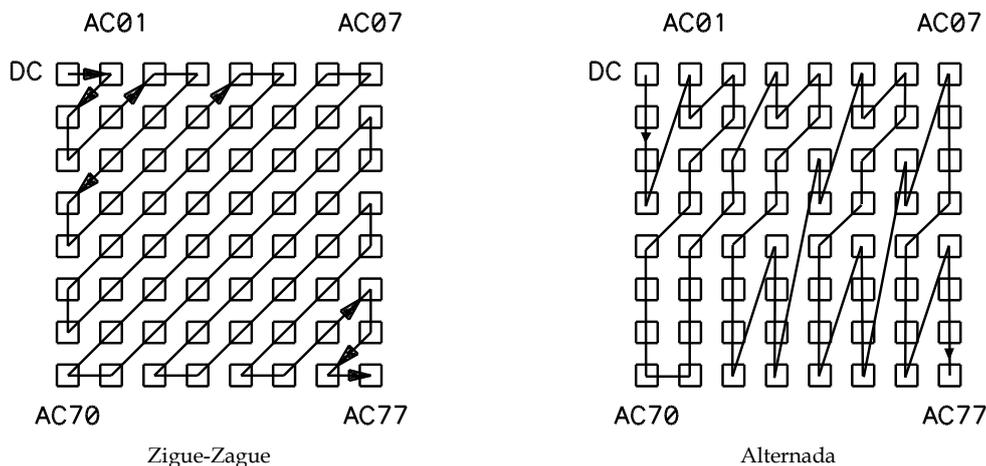
**Fig. 5.6 - Predição de Movimento de Campo para Quadro**



**Fig. 5.7 - Predição "Dual-Prime para imagens P**

#### 6.4 Ordenação dos Coeficientes da DCT:

O padrão MPEG-2 admite um padrão alternado para reordenação dos coeficientes quantizados da DCT, além do padrão zigue-zague convencional (figura 5.8). O padrão alternado é útil para imagens entrelaçadas tipo Quadro, onde pode haver um aumento de energia nas frequências verticais mais elevadas, devido à redução de correlação entre linhas consecutivas.



**Fig. 5.8 - Padrões de Reordenação de Coeficientes da DCT**

#### 6.5 Precisão dos Vetores de Movimento

Em MPEG-1, os vetores de movimento podem ser transmitidos com precisão de 1 ou 1/2 pixel; já em MPEG-2 só são admitidos com precisão de 1/2 pixel.

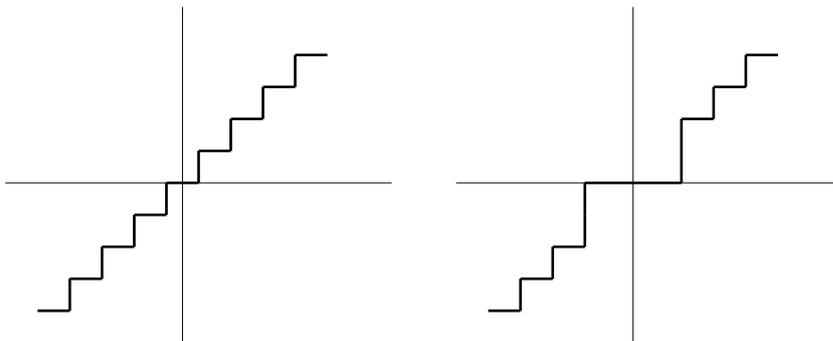
---

## 6.6 Formato das Fatias de Imagem

Na codificação MPEG-2, as Fatias de Imagem (*slices*) não podem ultrapassar as bordas da imagem; ou seja, a borda esquerda da imagem marca sempre um início de Fatia, e a borda direita um final de Fatia. Por outro lado, no MPEG-1, as Fatias podem começar e terminar em qualquer ponto da imagem.

## 6.7 Quantizador não-linear

No MPEG-2, os coeficientes AC da DCT obtidos nos macroblocos com predição de movimento (não-Intra) podem ser quantizados com uma função não-linear, que incorpora uma zona morta (figura 5.9).



**Fig. 5.9 - Quantizador linear (MPEG-1) e não-linear (MPEG-2 não-Intra)**

## 6.8 Dicionário ("Codebook") alternativo

Em MPEG-2 os coeficientes AC dos blocos com predição de movimento (não-Intra) podem ser compactados usando o "Dicionário 0" (igual ao usado pelo MPEG-1) ou um "Dicionário 1" alternativo, baseado em uma distribuição estatística de símbolos diferente.

## 6.9 Imagens I com Vetores de Movimento Opcionais

As imagens tipo **I** podem conter vetores de movimento, o que auxilia o mascaramento de erros de transmissão. Um bloco de referência que não for decodificado devido a erros pode ser aproximado a partir de uma imagem de referência anterior usando esses vetores.

## 6.10 Escalabilidade

O padrão MPEG-2 admite perfis com dois tipos de escalabilidade (ou seja, divisão do fluxo de dados em dois ou mais fluxos independentemente decodificáveis):

- Escalabilidade espacial: um fluxo de transporte carrega a imagem principal, codificada com baixa resolução, enquanto outros fluxos independentes carregam informações para reconstrução da imagem com resolução completa;
- Escalabilidade de Relação Sinal/Ruído: o fluxo principal carrega os coeficientes mais importantes da DCT, permitindo a reconstrução de uma imagem básica com baixa relação S/R, e os fluxos secundários transportam os coeficientes adicionais.

A escalabilidade permite que um mesmo programa seja distribuído por vários meios de comunicação com bandas passantes diferentes, ou para receptores com recursos de processamento diferentes, sendo que cada receptor decodificará a imagem com qualidade

proporcional à sua capacidade. Por exemplo, uma vídeo-conferência pode ser estabelecida entre terminais conectados a linhas telefônicas, redes ISDN ou redes ATM de alta capacidade, sem que a qualidade da imagem tenha que ser "nivelada por baixo".

### 6.11 Efeitos da Compressão de Vídeo

A tabela abaixo apresenta alguns resultados de simulações [Haskell, Puri e Netravali, 1997], usando sequências de vídeo padronizadas, mostrando o ganho de Relação Sinal/Ruído de reconstrução conseguido pela utilização de alguns recursos específicos do padrão MPEG-2. Deve ser considerado que a relação Sinal/Ruído não representa adequadamente a visibilidade dos erros de reconstrução; a imagem deve ser avaliada tendo por base um modelo da visão humana, resultando num índice de qualidade (PQR - Picture Quality Rating).

Recurso Utilizado	Ganho (dB)
Reordenação Alternada × Zig-zag	~ 0,5
Compensação de Movimento Quadro/Campo × MPEG-1	0,5 ~ 1,5
Compensação "Dual Prime"	0,5 ~ 1,0
DCT por Quadro / Campo	~ 0,9
Comp. Movimento + DCT Quadro/Campo × MPEG-1	~ 2,5
"Codebook" 1 × "Codebook" 0 (MPEG-1) (Imagens I)	0,1~ 0,9
Grupo de Imagens =9 × =15	-0,25
Quantizador não-linear	0.0
Imagens de Quadro × Imagens de Campo	~ 0,9
Imagens de Campo × Quadro (Movimentos rápidos)	~ 0,5

**Tabela 5.2 – Ganho de relação S/R para recursos do MPEG-2**

Altas taxas de compressão introduzem vários defeitos na imagem, entre os quais pode-se destacar:

- **Blocagem:** descontinuidades nos contornos dos blocos e macroblocos, causadas por quantização excessiva dos coeficientes da DCT;
- **Perda de detalhes:** texturas suaves são "alisadas", pois a quantização da DCT elimina as componentes de altas frequências e baixas amplitudes;
- **Segmentação de movimento:** em objetos com texturas de baixo contraste, partes da imagem permanecem estáticas quando não deviam, pois os macroblocos correspondentes (e seus vetores de movimento) foram descartados nas imagens **P** e **B**;
- **Perda de detalhes nos movimentos:** objetos em movimento que apresentam texturas detalhadas tornam-se ruidosos ou difusos devido a erros na detecção de movimento;
- **Vazamento de Cromo:** como a crominância possui resolução menor, e é desprezada na detecção de movimento, regiões de cores diferentes com alta saturação podem se misturar na imagem.



## 7. Outros Padrões de Compressão de Vídeo

### 7.1 MPEG-4

Após o início dos trabalhos do comitê MPEG-2, o ITU implantou um outro grupo de pesquisa, cujo objetivo era definir padrões de compressão para TV de Alta Definição. Este grupo, que seria denominado de MPEG-3, acabou sendo desativado, uma vez que as funcionalidades do MPEG-2 cobriam adequadamente os requisitos de HDTV.

Em 1993 o ITU-T iniciou atividades para um novo padrão, destinado à codificação de informação áudio-visual sobre taxas de bits reduzidas. O escopo do que viria a ser o padrão MPEG-4 foi expandido posteriormente, tornando-se um padrão genérico de codificação de eventos multimídia.

Um recurso importante do MPEG-4 é a segmentação de um objeto áudio-visual em objetos semânticos, representados por contornos e texturas. Cada objeto é codificado dentro de planos independentes (*VOP's - Video Object Planes*). O receptor pode manipular separadamente o fluxo de dados proveniente de cada plano, permitindo assim visualização apenas dos objetos selecionados.

Além disso, o decodificador admite a carga de ferramentas (algoritmos e processos programáveis), juntamente com o fluxo de dados, tornando-o aberto a futuros desenvolvimentos de técnicas de compressão e transporte de dados.

### 7.2 Padrão ITU-T H.264

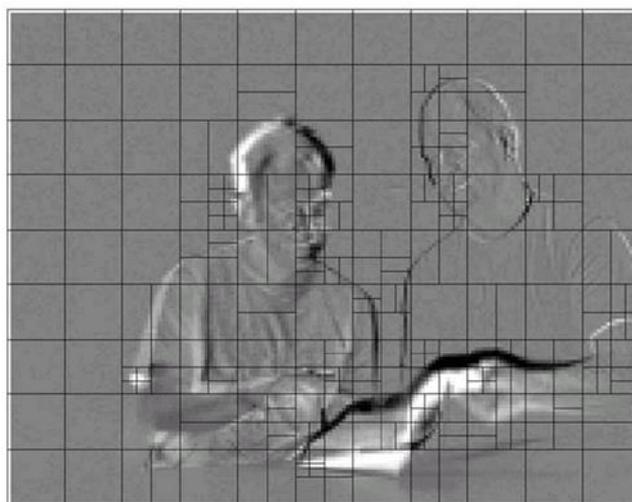
Este padrão, também referido como MPEG-4 parte 10, AVC (*Advanced Video Coding*), foi definido em 2003 com a finalidade de prover técnicas avançadas de compressão para aplicações desde videoconferência até edição de imagens de alta definição em estúdio.

Essa flexibilidade se deve a 4 perfis gerais de implementação: “*baseline*”, *principal*, *estendido* e “*FRExt*” (*Fidelity Range Extension*).

A codificação de vídeo para o Sistema Brasileiro de TV Digital, nas aplicações de TV de alta definição, utiliza este padrão.

De um modo geral, o H.264 utiliza os mesmos princípios do MPEG-2. Entre as ferramentas específicas utilizadas neste novo padrão, estão:

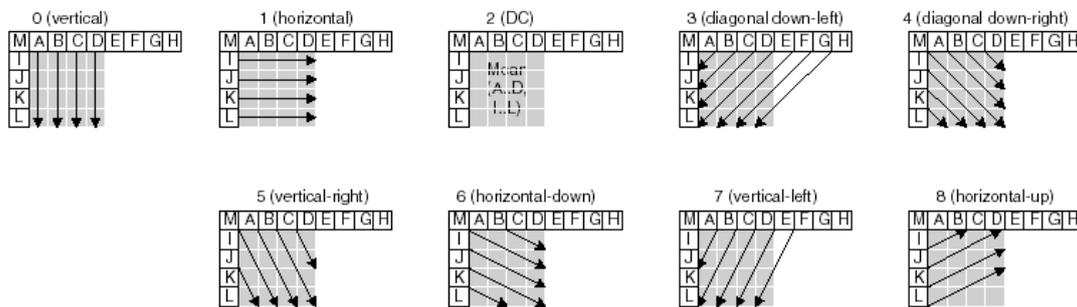
- **Blocos de tamanho variável:** A predição de movimento pode ser efetuada em blocos com dimensão de  $16 \times 16$ ,  $16 \times 8$ ,  $8 \times 8$ ,  $8 \times 4$  e  $4 \times 4$ . Esta possibilidade reduz os efeitos de “blocagem” em contornos de alto contraste da imagem, ao mesmo tempo em que aumenta a eficiência da compressão em regiões mais homogêneas. A figura 6.1 apresenta um exemplo de partição de blocos, superposta a uma imagem obtida através da diferença entre 2 quadros consecutivos. Considerando a partição, em imagens tipo B, um macrobloco de



**Fig. 6.1 – Blocos para Predição de Movimento**

16 × 16 pixels pode ter associado a ele até 32 vetores de movimento (até 16 sub-blocos, com vetores progressivos e regressivos). Adicionalmente, os vetores de movimento admitem precisão de ¼ de pixel.

• **Preditor Linear Bidimensional para Blocos Tipo I:** Os blocos tipo I, que não utilizam predição de movimento, podem ser codificados a partir de um preditor linear que utiliza as informações dos pixels adjacentes já codificados (9 pixels acima e 4 pixels à esquerda, para blocos 4 × 4, ou 17 pixels acima e 16 à esquerda para 16 x 16). A codificação por transformada é aplicada apenas sobre o erro de predição, resultando em ganhos consideráveis de compressão para imagens tipo I. A figura 6.2 apresenta os preditores lineares utilizados em blocos 4 × 4. Para blocos 16 x 16, existem 4 tipos de preditores lineares (horizontal, vertical, média e interpolação diagonal).



**Fig. 6.2 – Preditores Lineares para Blocos 4 x 4 tipo I**

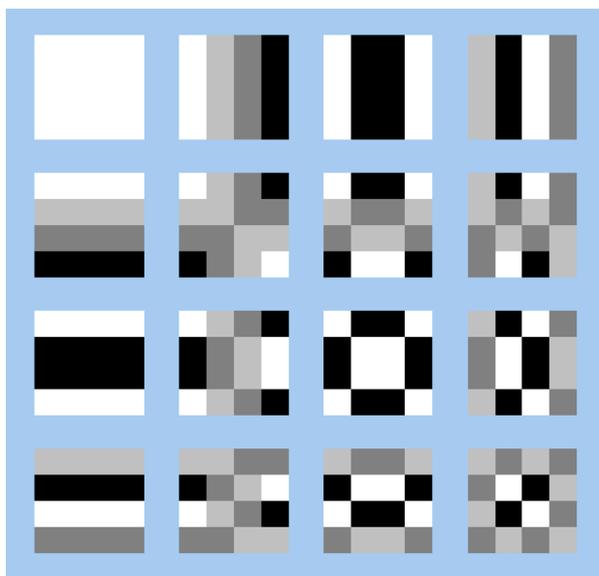
• **Transformada Inteira:** A codificação dos erros de predição é feita por uma transformada similar à DCT, porém utilizando coeficientes inteiros, resultando em maior facilidade de cálculo e menores erros de implementação. Aliás, a implementação do padrão H.264 utiliza apenas aritmética com números inteiros de 16 bits. A transformada, utilizada em blocos 4 × 4, é dada por

$$Y = C \cdot X \cdot C^T \otimes E$$

$$C = \begin{bmatrix} +1 & +1 & +1 & +1 \\ +2 & +1 & -1 & -2 \\ +1 & -1 & -1 & +1 \\ +1 & -2 & +2 & -1 \end{bmatrix} \quad E = \begin{bmatrix} \frac{1}{4} & \frac{1}{\sqrt{40}} & \frac{1}{4} & \frac{1}{\sqrt{40}} \\ \frac{1}{\sqrt{40}} & \frac{1}{10} & \frac{1}{\sqrt{40}} & \frac{1}{10} \\ \frac{1}{4} & \frac{1}{\sqrt{40}} & \frac{1}{4} & \frac{1}{\sqrt{40}} \\ \frac{1}{\sqrt{40}} & \frac{1}{10} & \frac{1}{\sqrt{40}} & \frac{1}{10} \end{bmatrix}$$

Onde  $X$  é um bloco 4 × 4 de pixels a serem processados,  $Y$  é o conjunto resultante de coeficientes,  $C$  é a matriz base da transformação, e  $E$  é uma matriz de correção, cuja aplicação na realidade é associada à etapa posterior de quantização.

A figura 6.3 apresenta as primitivas normalizadas da Transformada Inteira do H.264.



**Fig. 6.3 – Primitivas da Transformada Inteira do H.264**

- **Codificação Entrópica:** o padrão H.264 prevê dois tipos de codificação estatística para os símbolos a serem transmitidos:
  - CAVLC (*Context Adaptive Variable Length Coding*): utiliza tabelas de comprimento variável que se adaptam às características da imagem de acordo com os símbolos transmitidos anteriormente.
  - CABAC (*Context Adaptive Binary Arithmetic Coding*): baseia-se em modelos para predição da probabilidade condicional de ocorrência dos símbolos transmitidos, em função das características da região da imagem sendo codificada.
- **Filtro anti-blocagem:** os contornos dos blocos reconstruídos pelo preditor podem ser amenizados através da aplicação de filtros espaciais, em pontos selecionados pelo codificador, de modo a reduzir efeitos de “blocagem” na imagem recuperada.

Os recursos do H.264 permitem em geral uma redução de 50% na taxa de bits, para uma mesma qualidade de imagem, porém a um custo computacional da ordem de 5 vezes maior, em relação ao MPEG-2. Este codificador é bastante promissor para substituir o MPEG-2 em novas aplicações de TV digital.

#### 7.4 SMPTE VC-1 (Windows Media Vídeo 9)

Este padrão foi inicialmente desenvolvido como codificador proprietário pela Microsoft, e posteriormente submetido como padrão à SMPTE. Apresenta características e recursos similares ao H.264, sendo que algumas diferenças são:

- Uso de quantização com zona morta;
- Transformadas DCT  $8 \times 8$ ,  $8 \times 4$ ,  $4 \times 4$ , além de transformada inteira  $16 \times 16$ ;
- Fator de compensação para cenas com “fade-out” e “fade-in”;
- Tabelas de codificação múltiplas, selecionáveis a nível de quadro
- Complexidade computacional menor que o H.264.

## 7.5 Padrão DIRAC (BBC)

O decodificador “Dirac” foi desenvolvido em 2007 pela BBC (British Broadcasting Corporation), inicialmente para uso interno, e atualmente é distribuído livremente na forma de projeto “Open Source” (código aberto, livre de “Royalties”). Utiliza basicamente as seguintes ferramentas:

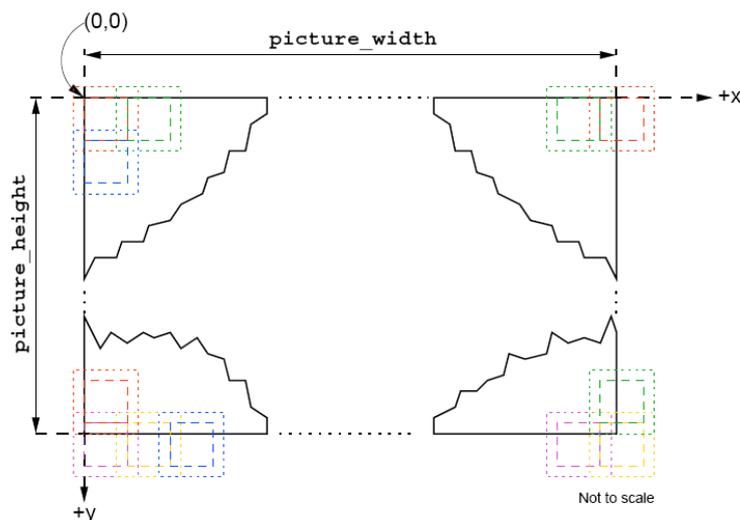
- Compensação de movimento por blocos superpostos
- Transformada por “Wavelets” sobre a imagem inteira
- Blocos de codificação com quantizadores independentes por região
- Codificação de símbolos: *Exp-Golomb* ou Aritmética
- Imagens organizadas como lista duplamente ligada (ponteiros para imagem anterior e posterior)

O codificador Dirac é estruturalmente simples, com poucas ferramentas, porém praticamente sem restrições nos parâmetros de codificação. Por exemplo, foi aplicado com sucesso para codificar sinais em “Super-Hi-Vision” (7680 x 4320 pixels, 60 quadros por segundo, a 70 Mb/s).

### 7.5.1 Compensação de Movimento no padrão Dirac

A compensação de movimento é realizada em 3 níveis:

- Nível global (válido para a imagem inteira): suporta translação em x e y, rotação, cisalhamento e efeitos de perspectiva (trapézio). Desta forma, as deformações devidas aos principais movimentos de câmera (*pan, tilt, zoom, travelling*) são compensadas em uma única operação geométrica;
- Nível de macroblocos: cada conjunto de 4 x 4 blocos da imagem recebe um vetor de movimento;
- Nível de blocos: o macrobloco especifica se cada bloco individual, ou cada grupo de 2 x 2 blocos, possui componentes adicionais de vetores de movimento. A dimensão do bloco é arbitrária (porém constante em toda a imagem), e os blocos podem se superpor horizontal e/ou verticalmente (fig. 6.4). Os planos de luminância e crominância podem ter blocos de tamanhos independentes.



**Fig. 6.4 – Blocos de previsão de movimento no Dirac**

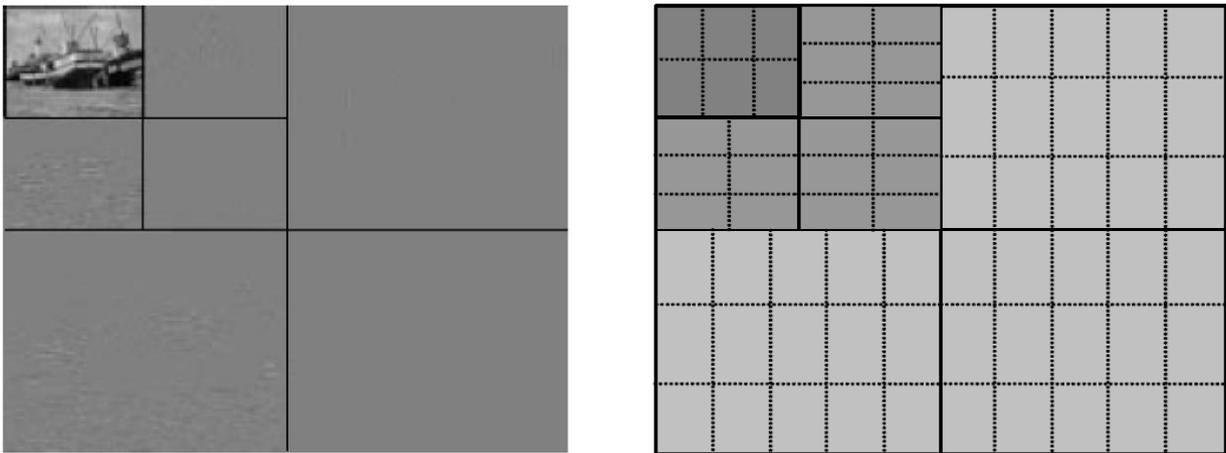
Os vetores de movimento têm precisão de 1 até 1/8 de pixel. Além disso, a compensação de movimento admite pesos independentes para cada imagem de referência, permitindo assim a codificação eficiente de efeitos de “fading” (fusão).

### 7.5.2 Transformada “Wavelet”

A transformada “Wavelet” subdivide sucessivamente o espectro espacial da imagem em duas metades, nos sentidos horizontal e vertical, de forma similar à utilizada no padrão JPEG-2000. O Dirac permite a escolha de 6 tipos de filtros “Wavelet”. O processo de subdivisão e decimação é levado adiante até que a imagem decimada alcance dimensões que não podem ser divididas por 2. Por exemplo, uma imagem 1920 x 1088 será decimada 6 vezes, até atingir 30 x 17 pixels.

A figura 6.5 apresenta um exemplo do processo de filtragem / decimação, com 2 níveis. Cada região de cada nível pode ainda ser subdividida em regiões de codificação retangulares, de dimensões arbitrárias, sendo que em cada região de codificação pode ser usado um fator de quantização diferente.

A aplicação da transformada “Wavelet” sobre a imagem inteira contribui para a ausência de efeitos de “blocagem” na compressão, mesmo em altas taxas.



**Fig. 6.5 – Aplicação de “Wavelets” e decimação; regiões de codificação da imagem**

### 7.5.3 Sintaxe do Fluxo de Dados

O fluxo de dados consiste de concatenações de “Sequências Dirac”, sendo que cada Sequência é formada por “Unidades de Dados”, precedidas de “Cabeçalhos de Classificação (“Parse Headers”). Cada cabeçalho de classificação, por sua vez, consiste dos seguintes elementos:

- **Prefixo de Classificação:** consiste de uma sequência de 4 bytes, com valor 0x42424344 (equivalente aos caracteres “BBCD” em código ASCII);
- **Código de Classificação (“Parse Code”):** consiste de 1 byte, cujos bits definem independentemente o perfil (“Low Delay” ou normal), a codificação (Aritmética ou Exp-Golomb), o tipo de dados contidos na Unidade de Dados (imagem, dados auxiliares, início ou fim de Sequência), o número de imagens de referência usadas pela imagem atual (0, 1 ou 2) e um flag que indica se a imagem atual deve ser salva como referência;
- **Distância até o próximo cabeçalho:** número inteiro com 4 bytes, definindo a distância (em bytes) até a próxima Unidade de Dados. Este campo é usado para permitir ações de “Fast Forward” na sequência de vídeo;

- 
- Distância até o cabeçalho anterior: número inteiro com 4 bytes, definindo a distância até o início da unidade de dados anterior. É usado para permitir ações de “*Rewind*”.

No início de uma Sequência, a Unidade de Dados correspondente identifica a versão, perfil e nível do codificador usado, bem como os parâmetros de vídeo utilizados (resolução, taxa de quadros por segundo, estrutura de amostragem, etc.). Estes parâmetros podem ser padronizados (tabelados) ou arbitrários.

#### 7.5.4 Níveis e Perfis Dirac

O padrão Dirac define 4 perfis de codificação:

- **Low Delay**: não utiliza predição de movimento (apenas transformada “Wavelet”), e a imagem pode ser subdividida em faixas horizontais. Este perfil apresenta baixa latência (menor que o tempo de 1 campo), sendo adequado para aplicações bidirecionais;
- **Simple**: sem predição de movimento e sem codificação aritmética, com transformada “Wavelet” sobre a imagem inteira. Com baixa complexidade computacional, alta qualidade de imagem, com taxa de compressão baixa, destina-se a aplicações profissionais;
- **Main**: sem predição de movimento, mas com codificação aritmética. Maior complexidade computacional com maior eficiência de compressão;
- **Main Long GOP**: com predição de movimento por blocos, e com codificação aritmética. Proporciona maior taxa de compressão, com qualidade superior ao MPEG-2 e H.264.

Quanto aos níveis de complexidade, o Dirac atualmente define apenas 2 níveis:

- **Nível 1**: parâmetros de vídeo (tamanho da imagem, estrutura de amostragem, taxa de quadros, etc.) padronizados, pertencentes a formatos pré-definidos (p. ex., QCIF, SDTV, HDTV etc.). Este nível é válido para os perfis “*Low Delay*”, “*Simple*” e “*Main*”;
- **Nível 128**: parâmetros de vídeo padronizados ou arbitrários, mas com amostras de 8 bits. Este nível é válido para o perfil “*Main Long GOP*”.

Demais níveis poderão ser definidos futuramente.

#### 7.6 Padrão H.265 / HEVC

Desenvolvido como sucessor do H.264, este padrão conjunto ISO/IEC e ITU-T acrescenta mais opções às ferramentas de compressão utilizadas, ao mesmo tempo que o simplifica em outras instâncias. O padrão HEVC (*High Efficiency Video Coder*), finalizado em 01/2013, suporta codificação UHDV (8k) com taxa de bits 50% menor que o H.264. Algumas características responsáveis por este desempenho são:

- Codificação com blocos de tamanho até 64 x 64
- Preditores “Intra” com mais opções de direções de interpolação
- Processamento paralelo possibilitado pela divisão da imagem em regiões independentes
- Transformadas inteiras de 4 x 4, 8 x 8, 16 x 16 e 32 x 32
- Uso de codificação CABAC apenas

---

## 8. Bibliografia e Referências

- A. Murat Tekalp - *Digital Video Processing* - Prentice Hall, 1995
- Andrew F. Inglis, Arch C. Luther - *Video Engineering* - Mc Graw-Hill, 1996
- ATSC - Grand Alliance HDTV System Specification – ATSC, 1994
- Barry Haskell et al. - *Digital Video: An Introduction to MPEG-2* - Chapman & Hall, 1997
- C. Poynton – *Digital Vídeo and HDTV: Algorithms and Interfaces* –Morgan Kaufmannn, 2003
- I. E. G. Richardson - *H.264 and MPEG-4 Video Compression* –J. Wiley & Sons, 2003 (figs. 1.2, 2.4, 6.1, 6.2)
- Jason Spielman – *MPEG Video Compression, Decompression Standard* –Motorola Multimedia Device Data Handbook, 1995
- Jerry Whitaker, ed. - *DTV – The Revolution in Electronic Imaging* –Mc Graw-Hill, 1998
- John Mitchell et al. - *MPEG Video Compression Standard* - Chapman & Hall, 1996 (figs. 5.10, 5.11)
- Michael Robin, Michel Poulin - *Digital Television Fundamentals* - Mc Graw-Hill, 1997
- Peter D. Symes - *Video Compression* –Mc Graw-Hill, 1998
- Rafael C. Gonzales, Richard E. Woods - *Digital Image Processing* - Addison Wesley, 1992
- BBC – British Broadcasting Corporation – *The Dirac Video Compression Standard* – BBC, 2008
- Mathias Wien – *High Efficiency Video Coding* – Springer, 2015

## 9. Apêndice – Tabelas de Codificação MPEG-1

**Tabela 1 - Codificação da Relação de Aspecto do Pixel**

CÓDIGO	ALTURA / LARGURA
0000	Proibido
0001	1.00
0010	0.6735
0011	0.7175
0100	0.7615
0101	0.8055
0110	0.8495
0111	0.8935
1000	0.9375
1001	0.9815
1010	1.0255
1011	1.0695
1100	1.1135
1101	1.1575
1110	1.2015
1111	Reservado

**Tabela 2 - Codificação da Taxa de Frames / Segundo**

CÓDIGO	FRAMES / s
0000	Proibido
0001	23.976
0010	24
0011	25
0100	29.97
0101	30
0110	50
0111	59.94
1000	60
1001	Reservado
1010	Reservado
1011	Reservado
1100	Reservado
1101	Reservado
1110	Reservado
1111	Reservado

**Tabela 3 - Matriz de Quantização "Default" (\*)**

**Intra-Frame (Imagens Tipo I)**

8	16	19	22	26	27	29	34
16	16	22	24	27	29	34	37
19	22	26	27	39	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

**Inter-Frame (Imagens Tipo P, B)**

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

**Tabela 4 - Codificação do Time Code**

CAMPO	BITS	VALORES
Flag Drop Frame	1	0 / 1
Horas	5	0 a 23
Minutos	6	0 a 59
(Valor Fixo)	1	1
Segundos	6	0 a 59
Frames	6	0 a 59

**Tabela 5 - Codificação do Tipo de Imagem**

CÓDIGO	TIPO DE IMAGEM
000	Proibido
001	Imagem I
010	Imagem P
011	Imagem B
100	Imagem D
101	Reservado
110	Reservado
111	Reservado

**Tabela 6 - Codificação do Tamanho dos Vetores de Movimento**

CÓDIGO	TAMANHO	FATOR
001	0	1
010	1	2
011	2	4
100	3	8
101	4	16
110	5	32
111	6	64

TAMANHO = Número de bits da Codificação do Resíduo

FATOR = Fator Multiplicativo do Valor Base (Tabela 9)

**Alcance dos Vetores de Movimento**

CÓDIGO	ALCANCE (1/2 pixel)	ALCANCE (1 pixel)	MÓDULO
001	-8 a 7.5	-16 a 15	32
010	-16 a 15.5	-32 a 31	64
011	-32 a 31.5	-64 a 63	128
100	-64 a 63.5	-128 a 127	256
101	-128 a 127.5	-256 a 255	512
110	-256 a 255.5	-512 a 511	1024
111	-512 a 511.5	-1024 a 1023	2048

**Tabela 7 - Códigos de Incremento de Endereço de Macrobloco**

CÓDIGO	INCREM.	CÓDIGO	INCREM.	CÓDIGO	INCREM.
1	1	0000 1000	13	000 0010 0000	25
011	2	0000 0111	14	000 0001 1111	26
010	3	0000 0110	15	000 0001 1110	27
0011	4	00 0001 0111	16	000 0001 1101	28
0010	5	00 0001 0110	17	000 0001 1100	29
0 0011	6	00 0001 0101	18	000 0001 1011	30
0 0010	7	00 0001 0100	19	000 0001 1010	31
000 0111	8	00 0001 0011	20	000 0001 1001	32
000 0110	9	00 0001 0010	21	000 0001 1000	33
0000 1011	10	000 0010 0011	22	000 0000 1000	Escape (33)
0000 1010	11	000 0010 0010	23	000 0000 1111	Enchimento
0000 1001	12	000 0010 0001	24		

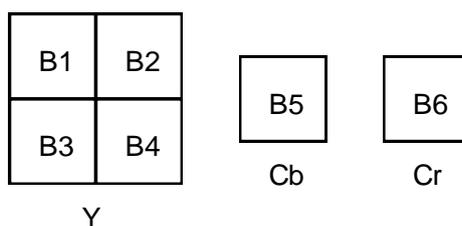
**Tabela 8 - Tipos de Macroblocos**

TIPO DE IMAGEM	CÓDIGO	NOME	FATOR DE ESCALA	MOVIMENTO Progr.	MOVIMENTO REGR.	PADRÃO DE BLOCOS	CODIF. INTRA
Tipo I	1	intra-d	0	0	0	0	1
	01	intra-q	1	0	0	0	1
Tipo P	1	pred-mc	0	1	0	1	0
	01	pred-c	0	0	0	1	0
	001	pred-m	0	1	0	0	0
	0 0011	intra-df	0	0	0	0	1
	0 0010	pred-mcq	1	1	0	1	0
	0 0001	pred-cq	1	0	0	1	0
	00 0001	intra-q	1	0	0	0	1
Tipo B	10	pred-i	0	1	1	0	0
	11	pred-ic	0	1	1	1	0
	010	pred-b	0	0	1	0	0
	011	pred-bc	0	0	1	1	0
	0010	pred-f	0	1	0	0	0
	0011	pred-fc	0	1	0	1	0
	0 0011	intra-d	0	0	0	0	1
	0 0010	pred-icq	1	1	1	1	0
	00 0011	pred-fcq	1	1	0	1	0
	00 0010	pred-bcq	1	0	1	1	0
	00 0001	intra-q	1	0	0	0	1

**Tabela 10 - Padrão de Codificação de Blocos**

Código	B 1	B 2	B 3	B 4	B 5	B 6		Código	B 1	B 2	B 3	B 4	B 5	B 6
111	1	1	1	1	0	0		00011100	1	0	0	0	1	1
1101	0	0	0	1	0	0		00011011	0	0	1	1	0	1
1100	0	0	1	0	0	0		00011010	1	1	0	0	0	1
1011	0	1	0	0	0	0		00011001	0	1	0	1	0	1
1010	1	0	0	0	0	0		00011000	1	0	1	0	0	1
10011	0	0	1	1	0	0		00010111	0	0	1	1	1	0
10010	1	1	0	0	0	0		00010110	1	1	0	0	1	0
10001	0	1	0	1	0	0		00010101	0	1	0	1	1	0
10000	1	0	1	0	0	0		00010100	1	0	1	0	1	0
01111	0	1	1	1	0	0		00010011	0	0	1	1	1	1
01110	1	0	1	1	0	0		00010010	1	1	0	0	1	1
01101	1	1	0	1	0	0		00010001	0	1	0	1	1	1
01100	1	1	1	0	0	0		00010000	1	0	1	0	0	1
01011	0	0	0	0	0	1		00001111	0	1	1	0	0	1
01010	1	1	1	1	0	1		00001110	1	0	0	1	0	1
01001	0	0	0	0	1	0		00001101	0	1	1	0	1	0
01000	1	1	1	1	1	0		00001100	1	0	0	1	1	0
001111	0	1	1	0	0	0		00001011	0	1	1	1	0	1
001110	1	0	0	1	0	0		00001010	1	0	1	1	0	1
001101	0	0	0	0	1	1		00001001	1	1	0	1	0	1
001100	1	1	1	1	1	1		00001000	1	1	1	0	0	1
0010111	0	0	0	1	0	1		00000111	0	1	1	1	1	0
0010110	0	0	1	0	0	1		00000110	1	0	1	1	1	0
0010101	0	1	0	0	0	1		00000101	1	0	0	1	1	0
0010100	1	0	0	0	0	1		00000100	1	1	1	0	1	0
0010011	0	0	0	1	1	0		00000011	0	1	1	1	1	1
0010010	0	0	1	0	1	0		00000010	0	0	1	1	1	1
0010001	0	1	0	0	1	0		000000101	1	1	0	1	1	1
0010000	1	0	0	0	1	0		000000100	1	1	1	0	1	1
0001111	0	0	0	1	1	1		000000011	0	1	1	0	1	1
0001110	0	0	1	0	1	1		000000010	1	0	0	1	1	1
00011101	0	1	0	0	1	1								

B1, B2, B3, B4 - Luminância (Y) B5, B6 - Crominância (Cb e Cr)



**Tabela 9 - Valor Base do Vetor de Movimento**

CÓDIGO	VALOR BASE	CÓDIGO	VALOR BASE
1	0		
011	-1	010	1
0011	-2	0010	2
00011	-3	00010	3
0000111	-4	0000110	4
00001011	-5	00001010	5
00001001	-6	00001000	6
00000111	-7	00000110	7
0000010111	-8	0000010110	8
0000010101	-9	0000010100	9
0000010011	-10	0000010010	10
00000100011	-11	00000100010	11
00000100001	-12	00000100000	12
00000011111	-13	00000011110	13
00000011101	-14	00000011100	14
00000011011	-15	00000011010	15
00000011001	-16	00000011000	16

**Tabela 11 - Codificação da Componente DC**

CÓDIGO DE TAMANHO (LUMINÂNCIA)	CÓDIGO DE TAMANHO (CROMINÂNCIA)	TAMANHO DA AMPLITUDE	CÓDIGO DE VALOR DA AMPLITUDE	VALOR ABSOLUTO RESULTANTE
1111110	1111110	8	00000000...01111111	-128...-255
111110	1111110	7	0000000...0111111	-64...-127
11110	111110	6	000000...011111	-32...-63
1110	11110	5	00000...01111	-16...-31
110	1110	4	0000...0111	-8...-15
101	110	3	000...011	-4...-7
01	10	2	00...01	-2...-3
00	01	1	0	-1
100	00	0	-	0
00	01	1	1	1
01	10	2	10...11	2...3
101	110	3	100...111	4...7
110	1110	4	1000...1111	8...15
1110	11110	5	10000...11111	16...31
11110	111110	6	100000...111111	32...63
111110	1111110	7	1000000...1111111	64...127
1111110	11111110	8	10000000...11111111	128...255

**Tabela 12: Codificação RLE / Huffman para Coeficientes AC Intra e Coeficientes Não-Intra**

Zeros	Amplitude	Código
EOB		10
0	1	1s
0	1	11s
0	2	0100s
0	3	00101s
0	4	0000110s
0	5	00100110s
0	6	00100001s
0	7	0000001010s
0	8	000000011101s
0	9	000000011000s
0	10	000000010011s
0	11	000000010000s
0	12	0000000011010s
0	13	0000000011001s
0	14	0000000011000s
0	15	0000000010111s
0	16	0000000011111s
0	17	0000000011110s
0	18	0000000011101s
0	19	0000000011100s
0	20	0000000011011s
0	21	0000000011010s
0	22	0000000011001s
0	23	0000000011000s
0	24	0000000010111s
0	25	0000000010110s
0	26	0000000010101s
0	27	0000000010100s
0	28	0000000010011s
0	29	0000000010010s
0	30	0000000010001s
0	31	0000000010000s
0	32	00000000011000s
0	33	00000000010111s
0	34	00000000010110s
0	35	00000000010101s
0	36	00000000010100s
0	37	0000000001011s
0	38	00000000010010s
0	39	00000000010001s
0	40	00000000010000s
1	1	011s
1	2	000110s
1	3	00100101s
1	4	0000001100s
1	5	000000011011s
1	6	0000000010110s
1	7	0000000010111s
1	8	00000000011111s
1	9	00000000011110s
1	10	00000000011101s
1	11	00000000011100s
1	12	00000000011011s
1	13	00000000011010s
1	14	00000000011001s
1	15	000000000010011s

Zeros	Amplitude	Código
1	16	0000000000010010s
1	17	0000000000010001s
1	18	0000000000010000s
2	1	0101s
2	2	0000100s
2	3	0000001001s
2	4	000000010100s
2	5	0000000010100s
3	1	00111s
3	2	00100100s
3	3	000000011100s
3	4	0000000010011s
4	1	00110s
4	2	0000001111s
4	3	000000010010s
5	1	000111s
5	2	0000001001s
5	3	000000010010s
6	1	000101s
6	2	000000011110s
6	3	000000000010100s
7	1	000100s
7	2	000000010101s
8	1	0000111s
8	2	000000010001s
9	1	0000101s
9	2	0000000010001s
10	1	00100111s
10	2	0000000010000s
11	1	00100011s
11	2	000000000011010s
12	1	00100010s
12	2	0000000000011001s
13	1	00100000s
13	2	0000000000011000s
14	1	0000001110s
14	2	0000000000010111s
15	1	0000001101s
15	2	0000000000010110s
16	1	0000001000s
16	2	0000000000010101s
17	1	000000001111s
18	1	000000011010s
19	1	000000011001s
20	1	000000010111s
21	1	000000010110s
22	1	0000000011111s
23	1	0000000011110s
24	1	0000000011101s
25	1	0000000011100s
26	1	0000000011011s
27	1	000000000011111s
28	1	0000000000011110s
29	1	0000000000011101s
30	1	0000000000011100s
31	1	0000000000011011s
escape		000001

**Coeficiente positivo: s=0 negativo: s=1**