

2

Técnicas de Codificação de Vídeo

2.1

Princípios de Codificação de Vídeo

Um sistema que prevê a redução da taxa de bits para representar um sinal digitalizado é chamado de sistema de compressão. E, para conseguir tal objetivo, este sistema trabalha removendo a informação redundante de um sinal antes da transmissão ou do armazenamento. Em sinais de vídeo, existem três tipos de redundância:

- Redundância Espacial: Onde os valores dos pixels são correlacionados com os seus vizinhos dentro de um mesmo quadro, ou seja, não são independentes e, dentro de alguns limites, o valor de um pixel pode ser predito a partir dos valores dos pixels vizinhos.
- Redundância Temporal: Os valores dos pixels são correlacionados com os seus vizinhos entre quadros sucessivos de uma seqüência de vídeo. Da mesma forma, ainda dentro de certos limites, o valor de um pixel pode ser predito a partir dos valores dos pixels de quadros passados e até futuros.
- Redundância Psico-Visual: Refere-se à resposta limitada que o olho humano tem para fins detalhes na imagem (limites de resolução espacial), bem como os limites na habilidade de acompanhar imagens que se movem rapidamente (limites de resolução temporal). Estes limites são utilizados para que seja descartado aquele subconjunto de informação de vídeo que os ultrapassam, já que o sistema visual humano não é capaz de perceber esse tipo de informação.

Através da exploração dessa informação redundante conseguimos comprimir imagens e vídeos sem que nossa visão perceba.

Existem dois tipos de sistemas de compressão: com perdas e sem perdas. Ou seja, quando a informação pode ser totalmente recuperada, após

a compressão, dizemos que o sistema é sem perdas. Porém, se a informação reconstruída for diferente da original, dizemos que o sistema é com perdas. Veja alguns exemplos:

- Compressão sem perdas: Também chamado de compactação, utiliza-se de técnicas para comprimir programas, documentos e imagens médicas. No caso de documentos, podemos citar programas como o WinZip.
- Compressão com perdas: Normalmente utilizadas em sinais que possuem elevada taxa de bits e que não necessitem ser reconstruídos de maneira idêntica. Podemos citar algoritmos como o MPEG, para vídeo e o JPEG, para imagens estáticas.

2.2

Codificação Espacial

2.2.1

Codificação Intra-quadros - DCT

O primeiro passo durante este tipo de codificação se faz através da análise das frequências espaciais usando-se uma transformada. Uma transformada é apenas uma maneira de se expressar um sinal em um domínio diferente, no nosso caso, no domínio da frequência. Uma das transformadas mais usadas é a DCT (*Discrete Cosine Transform*), que tem o propósito de remover a redundância espacial concentrando a energia do sinal em poucos coeficientes.

A DCT bi-dimensional é realizada (já que o sinal de vídeo possui duas dimensões) em pequenos blocos de $N \times N$ pixels da imagem. Primeiramente realiza-se a DCT uni-dimensional nas colunas de cada bloco e posteriormente nas linhas de cada bloco (ver Figura 2.1).

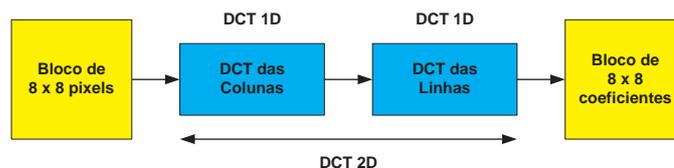


Figura 2.1: Processo realizado pela DCT 2D

A saída de uma DCT 2D é um conjunto de $N \times N$ coeficientes que representam o bloco original da imagem no domínio da transformada e estes

coeficientes são considerados pesos das funções base da DCT. Estas funções (Figura 2.2 [23]), são compostas por combinações das funções cosseno verticais e horizontais. Através desse método, qualquer bloco de imagem pode ser reconstruído combinando-se todas as $N \times N$ funções base, com cada função sendo multiplicada por um peso (coeficiente) apropriado.

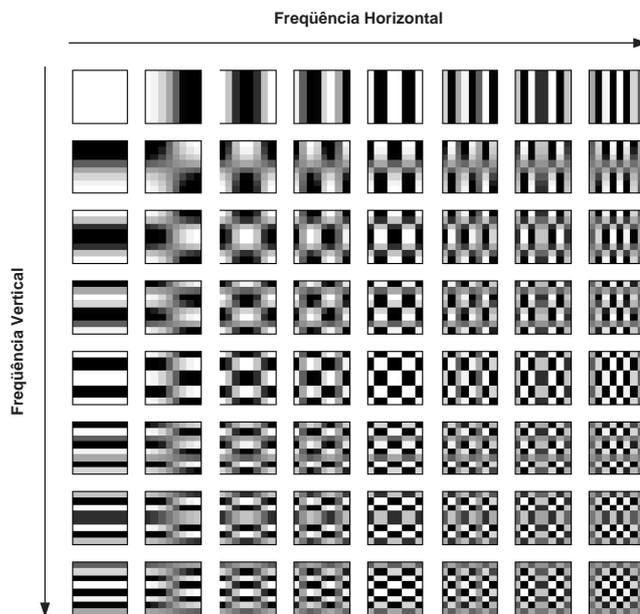


Figura 2.2: Funções base da DCT 8 x 8

O coeficiente correspondente à frequência horizontal e vertical nulas é chamado de coeficiente DC e representa o nível de brilho médio do bloco e os outros coeficientes são chamados de coeficientes AC.

A DCT não reduz diretamente o número de bits necessários para representar um bloco. Na verdade, para um bloco de 8×8 pixels com 8 bits por pixel, a DCT produz um bloco 8×8 coeficientes de 11 bits cada, ou seja, o alcance dos valores dos coeficientes é maior que dos valores dos pixels. A redução em termos de bit ocorre através da observação que, para típicos blocos de imagens naturais a distribuição dos coeficientes é não uniforme, ou seja, a transformada tende a concentrar energia em coeficientes de baixa frequência e os outros coeficientes tendem a ter valores perto de zero.

A redução de taxa é alcançada a partir da não transmissão dos coeficientes com valores perto de zero e quantizando e codificando os outros coeficientes. A distribuição não-uniforme dos coeficientes é resultado da redundância espacial presente no bloco de imagem original.

Na Figura 2.3 mostramos um exemplo de transformação através da DCT bi-dimensional em cima da imagem monocromática LENA, de 256×256 pixels.

A DCT 2D pode ser denotada matematicamente por:

$$Y = AXA^T \tag{2-1}$$

E a IDCT ou DCT inversa por:

$$X = A^T Y A \tag{2-2}$$

onde X representa o bloco original, Y o bloco de coeficientes, A é a matriz transformação de dimensão $N \times N$, i representa as linhas de cada bloco e j as colunas de cada bloco. Os elementos de A são:

$$A_{ij} = C_i \cos\left(\frac{(2j+1)i\pi}{2N}\right) \text{ onde } C_i = \sqrt{\frac{1}{N}} \ (i = 0), \ C_i = \sqrt{\frac{2}{N}} \ (i > 0) \tag{2-3}$$

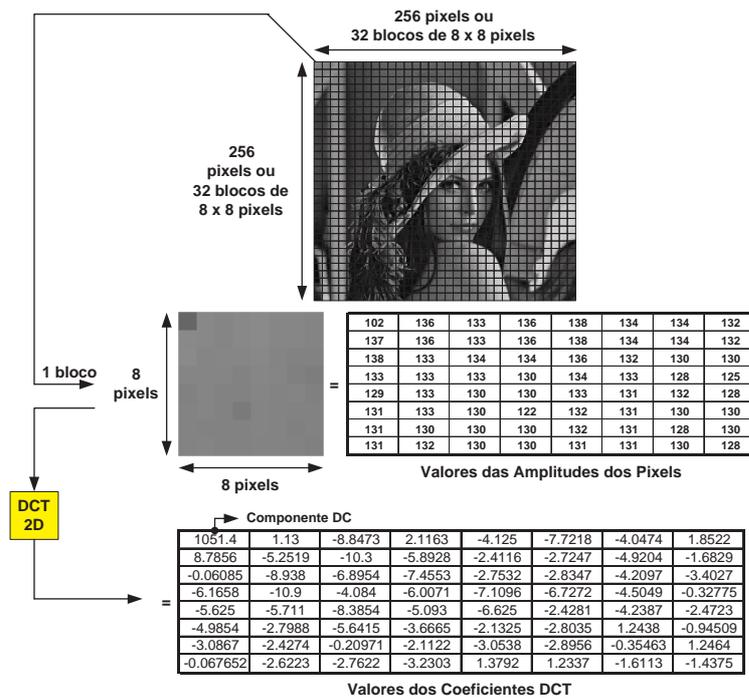


Figura 2.3: Exemplo de Transformação pela DCT

2.2.2 Quantização

Depois que a DCT realizou a transformação do bloco da imagem original para um domínio diferente, este bloco de coeficientes é quantizado. Um quantizador mapeia um sinal que possui uma faixa de valores X em faixas reduzidas de valores Y e cada um desses valores (coeficientes) é quantizado de uma maneira diferente, de acordo com as frequências espaciais que ele representa. O objetivo é minimizar o número de bits a serem transmitidos para o decodificador, de uma maneira eficiente, a fim de que este possa proceder com a operação inversa e reconstruir a imagem: uma quantização menos precisa reduz o número de bits que precisam ser transmitidos porém, aumentam a degradação da imagem devido ao aumento do erro de quantização, que é definido como a diferença entre o sinal de entrada e o sinal de saída de um quantizador (ver Figura 2.4).



Figura 2.4: Erro de Quantização

Na Figura 2.4 S_i é o sinal de entrada que pertence aos números reais e Z_i o sinal de saída, que pertence aos números inteiros. Dessa forma, o erro ou ruído de quantização é dado por:

$$q_i = S_i - Z_i \quad (2-4)$$

Note que este ruído de quantização, que é introduzido pelo quantizador, não pode ser revertido pelo decodificador e, sendo assim, o processo de compressão torna-se *lossy* ou com perdas.

Um número maior de erros na quantização pode ser tolerado nos coeficientes de alta frequência, pois o ruído de alta frequência é menos perceptível ao olho humano. E ainda: o ruído de quantização é menos visível nas componentes de crominância do que na componente de luminância. Em decorrência das particularidades do sistema visual humano, o padrão MPEG recomenda matrizes de peso para definir a relativa adequação da quantização dos diferentes coeficientes. Os coeficientes ponderados pelas matrizes de peso são, então, passados através de um critério fixo de quantização, o qual é usualmente um critério linear.

Entretanto, para alguns modos de predição (2.3.1), há um aumento da zona de entrada perto de zero, chamada zona morta, do inglês *dead zone*. Este método tem por objetivo maximizar o número de coeficientes que serão quantizados com valores nulos e, além disso, é sabido que pequenas variações perto de zero são causadas por ruído, o que significa que suprimindo esses valores, podemos obter uma aparente melhoria na qualidade subjetiva da imagem.

Existem vários tipos de quantizadores, como quantizadores escalares e vetoriais, que mapeiam uma amostra do sinal de entrada para um valor de saída quantizado ou mapeiam um grupo de amostras do sinal de entrada para um grupo de valores quantizados, respectivamente. Além disso, esses quantizadores podem ser uniformes, que seguem o mesmo passo de quantização para todos os níveis de decisão e de reconstrução e não-uniformes que variam o passo entre os diferentes níveis de decisão e reconstrução.

Um exemplo simples de quantização escalar pode ser obtido através do processo de aproximação de um número fracionário pelo seu inteiro mais próximo, ou seja:

$$NQ = \lfloor \frac{X}{\Delta P} \rfloor \quad (2-5)$$

$$Y = NQ \cdot \Delta P \quad (2-6)$$

onde ΔP é o passo de quantização, NQ é o inteiro mais próximo da operação em questão, X é o valor de entrada e Y o valor quantizado de saída.

A Figura 2.5 mostra dois exemplos de quantizadores escalares, um uniforme (o qual mapeia linearmente os valores entre a entrada e a saída) e um não-uniforme (que possui uma zona morta perto do valor zero, no qual pequenos valores são mapeados para zero).

A Tabela 2.1 mostra um exemplo de quantização dos coeficientes da DCT, com três passos de quantização. Esses coeficientes são relativos à primeira linha da matriz de coeficientes transformados da imagem LENA da Figura 2.3:

Note que conforme aumentamos o passo de quantização, mais coeficientes são levados para zero, reduzindo assim o número de bits necessários à transmissão. Veja na Figura 2.6 todos os coeficientes quantizados com os dois primeiros passos:

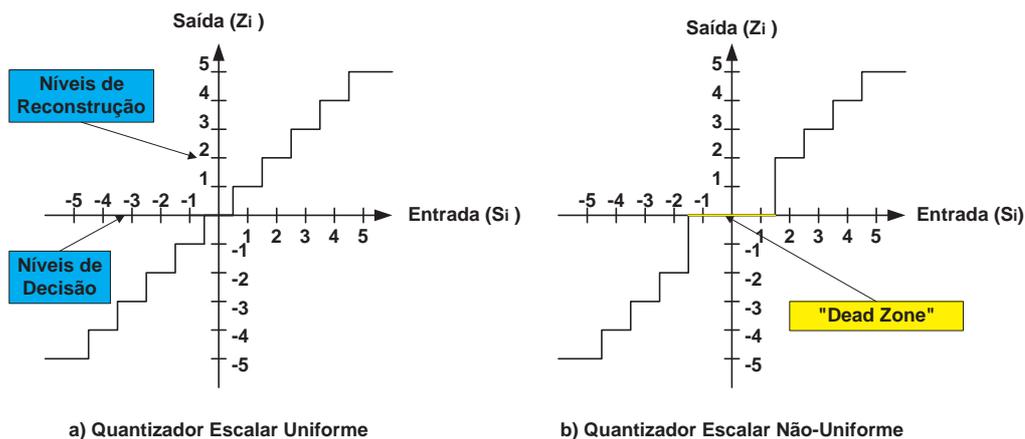


Figura 2.5: Quantizadores Escalares

Tabela 2.1: Exemplo de Quantização de Coeficientes DCT com três Passos de Quantização diferentes.

Coeficientes DCT	Quantizados com Passo=4	Quantizados com Passo=16	Quantizados com Passo=32
1051.4	1048	1040	1024
1.13	0	0	0
-8.8473	-8	0	0
2.1163	4	0	0
-4.125	-4	0	0
-7.7218	-4	0	0
-4.0474	-4	0	0
1.8522	0	0	0

2.2.3 Varredura ZIG-ZAG

Após o processo de quantização, os coeficientes DCT de um bloco são re-ordenados a fim de agrupar os coeficientes não-nulos, permitindo assim uma representação eficiente para aqueles coeficientes quantizados com valores nulos. A maneira de se *varrer* cada bloco de coeficientes varia de acordo com o padrão de vídeo que está sendo codificado, ou seja, se este vídeo trabalha com varredura progressiva ou entrelaçada. A Figura 2.7 apresenta esses dois modos de varredura:

Essa varredura começa no coeficiente DC e produz uma seqüência serial dos valores dos coeficientes quantizados, devidamente ordenados de acordo com o padrão de varredura.

Essa seqüência serial produzida pela varredura zig-zag é codificada contando-se o número de coeficientes nulos que precedem um coeficiente

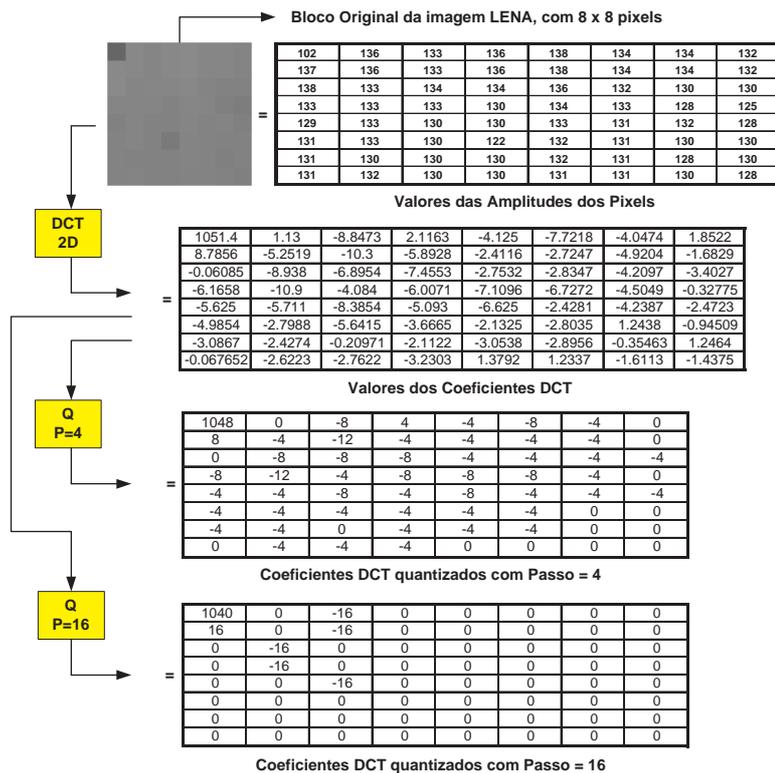


Figura 2.6: Coeficientes DCT quantizados com dois passos de quantização diferentes

não-nulo. A esta técnica chamamos de *Run-Length Coding* ou codificação por corrida de zeros, a qual veremos a seguir.

2.2.4 Codificação RLC

O saída do processo de re-ordenação é uma coleção que normalmente contém um ou mais grupos de coeficientes não-nulos perto do começo do bloco, seguidos por seqüências de coeficientes nulos. O grande número de valores iguais a zero podem ser codificados de uma maneira mais compacta, por exemplo representando-os por uma coleção de séries de pares (*run*, *level*), onde *run* indica o número de zeros que precedem um valor não-nulo e *level* indica a magnitude do coeficiente não-nulo.

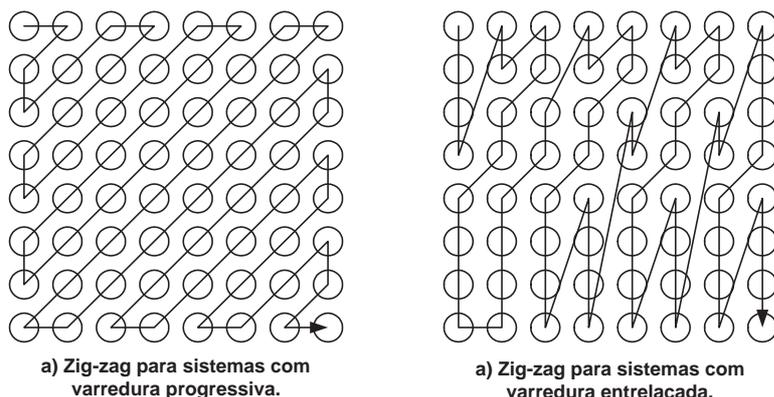


Figura 2.7: Varredura Zig-Zag

A Figura 2.8 apresenta o processo de varredura (zig-zag) e codificação RLC (*Run-Length Coding*) em cima do nosso exemplo com a imagem LENA. Utilizamos o bloco de coeficientes quantizados com passo de quantização igual a 16.

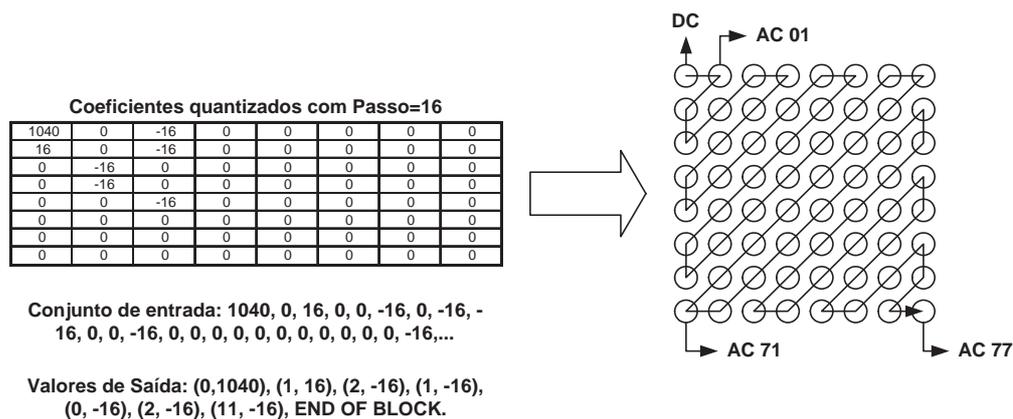


Figura 2.8: *Run-Length Coding*

2.2.5 Codificação VLC

Cada par ($run, level$) é então codificado usando um código de comprimento variável (VLC - *Variable Length Coding*). O VLC gera um código que tem diferentes comprimentos dependendo da frequência esperada de ocorrência de cada par ($run, level$). Combinações com maior probabilidade de ocorrerem usam palavras código com menor número de bits e combinações com menor probabilidade de ocorrerem usam palavras código com maior número de bits.

2.3 Codificação Temporal

Esta técnica explora a redundância temporal através da estimação do quadro que será codificado pelo quadro anterior, que é usado como referência. Esta predição não pode ser realizada na imagem original, já que as imagens originais não estarão disponíveis no decodificador (as imagens decodificadas não são idênticas às imagens originais, pois o processo de redução de taxa introduz pequenas distorções na imagem decodificada). Conseqüentemente, o codificador possui um decodificador, do qual as predições podem ser formadas.

Enquanto a unidade básica de codificação para a redundância espacial no padrão MPEG é baseada em blocos de 8 x 8 pixels, a unidade utilizada nas etapas de estimação de movimento é usualmente baseada em macroblocos de 16 x 16 pixels.

Para se explorar a correlação temporal da seqüência, utilizam-se técnicas DPCM. Basicamente esta técnica consiste em codificar a diferença entre o sinal e a sua predição, ao invés de codificar o próprio sinal.

Na codificação MPEG, a predição inter-quadros (ver Figura 2.9) é combinada com a DCT e com a codificação RLC / VLC. O codificador subtrai a predição da entrada para formar uma *imagem-erro de predição*. A DCT é aplicada ao erro de predição, os coeficientes resultantes são quantizados e esses valores são codificados usando RLC / VLC.

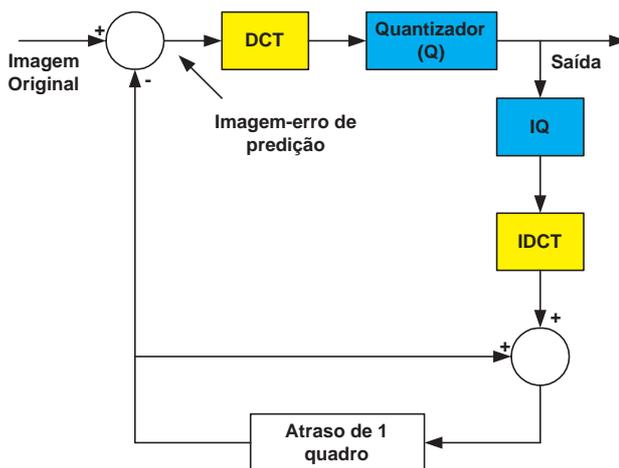


Figura 2.9: Codificador com predição inter-quadros

2.3.1 Modos de Predição

Um macrobloco pode ser predito por predição *forward*, a partir de uma imagem passada, por predição *backward*, a partir de uma imagem futura, ou interpolada através da média entre uma predição *forward* e *backward*. Outra opção é não realizar predição, de tal forma que o próprio bloco da imagem-fonte seja codificado, ao invés de codificar com base no erro de predição. A codificação de tais macroblocos é conhecida como Intra ou do tipo I. Diferentemente, o padrão H.264/AVC utiliza predição nos quadros do tipo I, como será visto no Capítulo 3.

Para cada macrobloco a ser codificado, o codificador escolhe entre os possíveis modos de predição, tentando minimizar as distorções na imagem decodificada, dentro das restrições de disponibilidade da taxa de bits do canal. A informação sobre o modo de predição escolhido é transmitida ao decodificador, juntamente com o erro de predição, de tal forma que o decodificador possa regenerar a predição correta.

2.3.2 Tipos de Quadros

Os quadros podem ser divididos em três tipos, de acordo com a codificação preditiva utilizada:

- Quadros I (intracodificado): Esses quadros são codificados de maneira independente dos outros quadros.
- Quadros P (predito): São codificados utilizando predição por estimação de movimento a partir de uma imagem I ou P passada, e podem ser usados como referência para predição *forward*.
- Quadros B (bidirecional): usam tanto imagens I ou P passadas quanto futuras para estimação de movimento, e oferecem o mais alto grau de compressão.

Todos os macroblocos de um quadro I são intracodificados. Nos quadros P, pode-se ter macroblocos preditos, ou intracodificados, o que equivale a utilizar um macrobloco nulo como predição. Finalmente, nos quadros B pode-se ter macroblocos interpolados, preditos ou intracodificados. Nos quadros B e P, existe a possibilidade de não se codificar determinado macrobloco. Nesse caso, o macrobloco é reconstruído no decodificador utilizando-se somente os quadros de referência e considerando-se nulos os vetores de movimento.

Os quadros a serem codificados geralmente estão dispostos na ordem temporal (ordem de apresentação na tela). Quando os quadros são codificados, eles são incluídos no *bitstream* de acordo com a ordem em que serão decodificados. No decodificador, faz-se um novo ordenamento e os quadros são novamente dispostos em ordem temporal.

Na Figura 2.10 ilustramos o explicado acima.

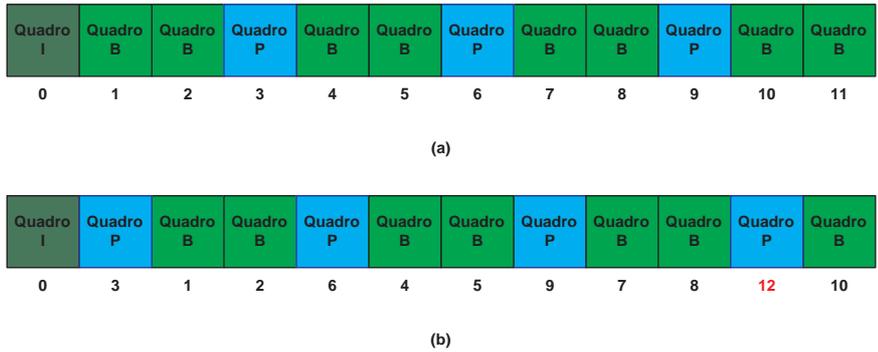


Figura 2.10: (a) Ordem Temporal (b) Ordem de Codificação

Os quadros também são divididos em grupos chamados GOP (*Group of Pictures*). O GOP é um conjunto de quadros que possibilita acesso aleatório. Tipicamente são agrupados entre 12 e 15 quadros em um mesmo GOP e é estabelecida uma ordem de quadros (I, P ou B), que se repete ao longo da seqüência.