

CODIFICAÇÃO DE VÍDEO EM H.264

Aluno: Thiago Vaz de A. C. Dias

Orientador: Abraham Alcaim

1. Introdução

Ouvimos com frequência nos dias de hoje a palavra televisão digital. Mas o que exatamente é isso? Qual a diferença e as vantagens desta perante o atual sistema televisivo?

A principal diferença entre o sistema utilizado atualmente para a transmissão de imagens no Brasil e este novo sistema é que o sinal de transmissão é digitalizado. Isso significa que as imagens (e o som) que chegam em nossas casas foram digitalizados a partir de imagens convencionais para então serem transmitidos.

Este sistema seria inviável a poucos anos atrás sem os avanços das técnicas de codificação de imagens. Sem estas técnicas, um sinal de vídeo digital na mesma resolução utilizada pelo sistema atual de televisão (SDTV – Standard Definition TV) necessitaria de aproximadamente 270 Mbits/s (270 Mbps) no padrão ITU-R-BT-601 [1], o que é impossível de se transmitir em um único canal de 6 MHz utilizado para a transmissão destes sinais. Assim, seriam necessários mais de treze canais para a transmissão de um único canal de TV. No caso da HDTV (High Definition TV), a televisão de alta-definição que está chegando agora no Brasil, seria ainda mais inviável, já que seriam necessários aproximadamente 1,5 Gbits/s, se utilizado o padrão ITU-R-BT-709 [2].

Este novo advento só se tornou possível com o lançamento do codec (codificador) MPEG-2 [3] do grupo MPEG (que também é o padrão H.262 [4] do VCEG) em 1994 e se torna cada vez mais viável e com aplicações mais diversificadas, como vídeos para telefones móveis e câmeras de segurança. O objeto de estudo desta iniciação é o H.264 AVC (ou simplesmente H.264, também é o padrão MPEG-4 Parte 10, do grupo MPEG), um dos codificadores mais modernos e com maior poder de compressão.

2. Codificação

Uma imagem digitalizada é dividida em vários pixels. Cada um destes pixels é representado por número extraído de um conjunto finito predefinido e cada número destes é

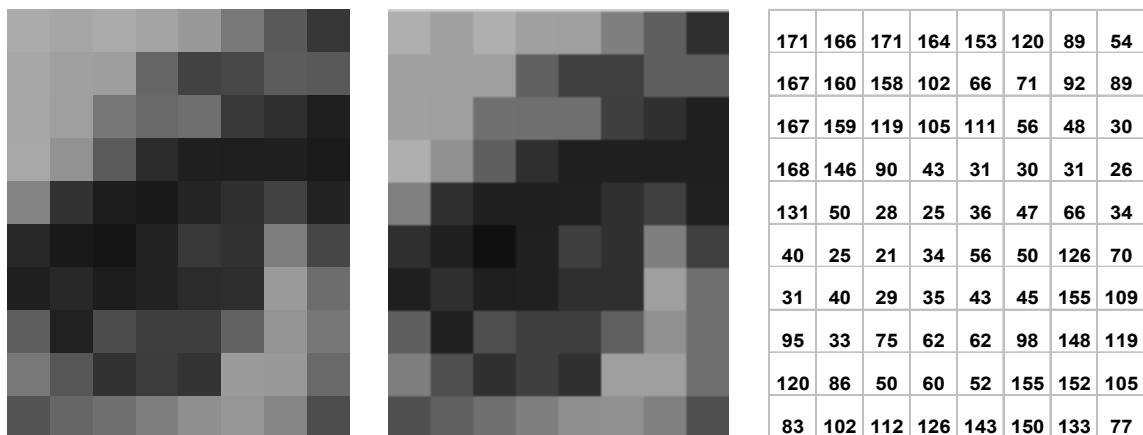


Figura 2.1: Representação de uma imagem (olho humano) sendo digitalizada com 8 bits por pixel. Repare que os valores estão entre 0 e 255 (variam de 21 a 171) [5].

constituído por diferentes seqüências de 1's e 0's. Quanto maior for a quantidade de pixels de uma imagem maior será a quantidade de números, maior será a quantidade de 1's e 0's e consequentemente maior será a quantidade de bits necessários para enviá-la ou armazená-la.

Um vídeo nada mais é do que uma seqüência de imagens estáticas. Agora, se cada pixel de uma imagem e cada imagem de um vídeo for digitalizada desta maneira será gerado um imenso arquivo. A tela de uma televisão comum geralmente tem 640x480 pixels, ou seja, 307200 pixels. No caso de um simples vídeo em preto e branco, com 256 níveis de cinza (o que equivale a dizer que cada pixel pode assumir um valor entre 0 e 255, ou seja, cada pixel tem 8 bits ou 1 byte), com 10 segundos de duração, a uma taxa de exibição de 30 fps (quadros por segundo), teria 737280000 bits, ou seja, mais de 737Mbits, que por sua vez é mais de 92 Mbytes. Um codificador busca exatamente reduzir este tamanho, chegando a fazê-lo em mais de 100 vezes. Por isto o estudo em técnicas de compressão vem sendo intensificado cada vez mais.

A taxa de bits é o principal parâmetro relacionado à qualidade do som e da imagem do arquivo codificado. É ele que define o tamanho do arquivo final. Numa seqüência de vídeo, por exemplo, a taxa de 1 kbit/s (um kilobit por segundo) significa que cada segundo do arquivo codificado vai ter 1000 bits. Ou seja, se um vídeo tiver 64 segundos, o arquivo vai ter 64 kbits, ou 8 kbytes. Quanto maior for a *bitrate*, maior será a qualidade do vídeo, mas também será maior o seu tamanho. O MPEG-2 Layer 3 (ou MP3, como é mais conhecido) é um codificador de áudio bem popular e geralmente é usada a taxa de 128 kbits/s para a codificação de músicas, que é a menor taxa para que o arquivo final fique com boa qualidade.

2.1. Tipos de codificadores

Codificador (ou *codec*) é um aparelho (hardware) ou programa de computador (software) capaz de codificar e/ou decodificar sinais digitalizados ou arquivos armazenados (em computadores, DVDs, etc). A palavra codec pode ser entendida como o acrônimo de “Codificador-Decodificador”.

Os codecs são capazes de codificar seqüências de áudio ou vídeo para armazenagem ou transmissão e depois podem decodificá-las para exibição ou edição, sendo comumente utilizados nas transmissões digitais de televisão, em vídeo conferências, mídias portáteis, etc.

Uma filmadora, por exemplo, capta sons e imagens (sinais analógicos), faz a digitalização destes e posteriormente faz a codificação do áudio e do vídeo, ocupando menos espaço, possibilitando assim, um maior tempo de gravação. Na exibição este sinal é decodificado, e o vídeo pode então ser assistido.

O MP3 é um outro exemplo de codificador de áudio. Graças a ele é possível reduzir um arquivo de áudio mantendo uma boa qualidade em mais de 10 vezes. Hoje, já existem outros codificadores de áudio, capazes de codificar sons utilizando a metade do espaço que o MP3 utiliza com a mesma qualidade.

Os codificadores podem ser divididos em dois grandes grupos, os sem perdas (*lossless*) e os com perdas (*lossy*). Os codificadores do primeiro grupo diminuem o tamanho de um arquivo sem alterar a qualidade do som e da imagem. Esse processo é chamado de compressão. Quando feita a descompressão, processo contrário a compressão, será obtido um arquivo exatamente igual ao original. Geralmente os arquivos comprimidos são de duas a três vezes menores que o original, sendo muito utilizados na radiodifusão e por emissoras de televisão.

O segundo grupo é formado por codecs que podem diminuir ainda mais o tamanho dos arquivos originais, chegando a gerar arquivos até 300 vezes menores, no caso de vídeos e 13 vezes menores no caso de arquivos de áudio [6]. Neste caso, o arquivo codificado apresenta uma perda na qualidade do som e da imagem e não é possível obter exatamente o mesmo arquivo quando feita a decodificação.

A taxa de compressão é escolhida a fim de controlar o tamanho do arquivo final de acordo com a qualidade que se deseja. A transmissão esportiva como a de uma partida de futebol, por exemplo, deve codificar bem os movimentos, mas não precisa se preocupar muito com as cores ou texturas das imagens, já um leilão de jóias precisaria se preocupar muito mais com estes últimos aspectos do que com a estimação ou predição de movimentos.

Aplicações como televisão digital ou os discos de alta definição que substituirão os atuais discos de DVD têm seus arquivos codificados bem menores que os originais apresentando uma perda imperceptível aos olhos humanos.

2.1.1. Codificadores sem perda (compactação)

A codificação com a utilização de codificadores lossless, ou sem perdas, também é chamada de compactação. Este tipo de codificação recebe estes nomes pois, depois de feita a descompressão do arquivo é obtida uma seqüência exatamente igual à original, como se não tivesse ocorrido a codificação. Dentro deste grupo os codificadores mais importantes são a Codificação de Huffman e a Codificação por Comprimento de Corridas, mas também temos outras como a Codificação Aritmética.

2.1.1.1. Código de Huffman

Dentre os códigos de bloco instantâneos, o código de Huffman é o ótimo. A idéia principal deste código é destinar aos símbolos de maior probabilidade de ocorrência menos bits e aos símbolos de menor ocorrência mais bits. Podemos obter as palavras-código geradas por este codificador montando uma árvore seguindo o processo descrito abaixo:

- (1) Os símbolos são ordenados em ordem de probabilidade de ocorrência.
- (2) Aos símbolos de menor probabilidade são associados arbitrariamente os bits 0 e 1.
- (3) Agrupa-se então estes dois símbolos em um grupo chamado s_1 de probabilidade igual a soma das probabilidades dos dois símbolos.
- (4) Repete-se então a etapa (2) tomando os bits de s_1 como se fossem um só e o restante dos bits, criando-se um novo conjunto de símbolos.
- (5) O grupo s_2 é obtido com a repetição da etapa (3).
- (6) Repete-se então este processo até que todos os símbolos sejam utilizados.
- (7) A palavra-código é então obtida agrupando-se os bits a partir da raiz da árvore.

Exemplo: Considere um conjunto de seis símbolos com probabilidades de ocorrência 0,20, 0,15, 0,05, 0,15, 0,35 e 0,10 para os quais deseja-se determinar um código de Huffman. A ordenação dos símbolos e o procedimento de codificação de Huffman são ilustrados na figura a seguir.

Os passos a serem seguidos nesse exemplo são:

- (1) Aos símbolos com menor probabilidade s_5 e s_6 são associados os bits 0 e 1.
- (2) s_5 e s_6 são agrupados como $s^{(1)}$ de probabilidade 0,15, isto é, $0,10 + 0,05$.
- (3) Dos três símbolos de menor probabilidade – s_3 , s_4 e $s^{(1)}$ – uma vez que os três apresentam a mesma probabilidade escolhe-se arbitrariamente, os símbolos s_4 e $s^{(1)}$, associando-se a eles os bits 0 e 1, respectivamente.
- (4) s_4 e $s^{(1)}$ são agrupados como $s^{(2)}$ de probabilidade 0,30, isto é, $0,15 + 0,15$.
- (5) Agora os símbolos com menor probabilidade são s_2 e s_3 , aos quais são associados os bits 0 e 1, respectivamente.
- (6) s_2 e s_3 são agrupados como $s^{(3)}$ de probabilidade 0,35, isto é, $0,20 + 0,15$.

(7) Agora o símbolo com menor probabilidade é $s^{(2)}$ (0,30) e em seguida s_1 ou $s^{(3)}$ (0,35). No exemplo escolheu-se, arbitrariamente, $s^{(2)}$ e $s^{(3)}$, associando-se a eles um 1 e um 0, respectivamente.

(8) $s^{(2)}$ e $s^{(3)}$ são agrupados como $s^{(4)}$ de probabilidade 0,65 (0,30 + 0,35).

(9) Finalmente, restam apenas dois símbolos $s^{(4)}$ e s_1 , aos quais são associados os bits 1 e 0, respectivamente. Depois de completado o procedimento acima, as palavras-código são então determinadas, seguindo-se os ramos no sentido esquerda-direita, a partir da base da árvore ilustrada na figura 2.2.

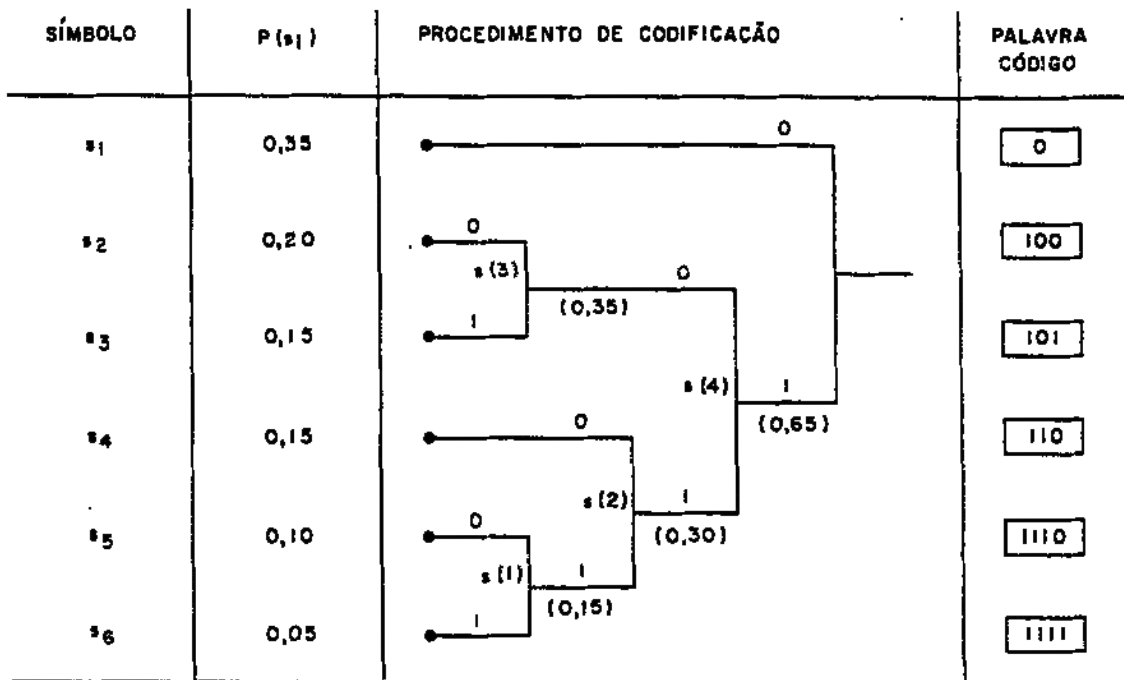


Figura 2.2: Procedimento de Codificação de Huffman. [7]

Uma observação importante é que este código não é único. Por exemplo, no passo (7) do exemplo poderia ter sido escolhido $s^{(2)}$ e s_1 ao invés de $s^{(2)}$ e $s^{(3)}$, alterando o código final.

2.1.1.2. Comprimento de corridas

A utilização deste codificador é mais eficiente em imagens com 2 níveis de cores (preto e branco), como num fax, por exemplo, onde ocorrem grandes repetições de 0's juntas (corridas ou rajadas). É exatamente esta característica explorada por este codificador.

A codificação por comprimento de corridas não é eficiente somente para símbolos binários, ela é muito útil para qualquer entrada em que existam grandes corridas ou rajadas de símbolos iguais. No padrão MPEG-2, por exemplo, as imagens são separadas em grupos de 64 pixels sendo transformados em 64 coeficientes ordenados em ordem crescente de frequência. Usualmente grande parte destes coeficientes são nulos, criando corridas de 0's. Assim, o MPEG-2 cria símbolos formados pelo par (comprimento da corrida de 0's, valor do elemento não-nulo que sucede a corrida). Geralmente ainda é feita uma compactação com o Código de Huffman desta.

Exemplo: Considere que temos a seguinte seqüência que representa um bloco de 64 pixels de uma imagem:

maioria dos computadores e tocadores de DVD atuais. O MP3, por exemplo, utiliza a mesma codificação de áudio encontrada no MPEG-1.

Este codificador tem cinco partes e é referenciado como ISO/IEC 11172.

2.1.2.3. H.262 ou MPEG-2

O MPEG-2, como é mais conhecido o H.262, foi desenvolvido em parceria pelos grupos ITU-T (VCEG) e ISO/IEC JTC1 (MPEG). Este padrão é o mais difundido atualmente e é encontrado desde transmissões digitais de televisão até nos discos de SVCD (Super Video CD) e DVD utilizados por qualquer um de nós. Este também é o padrão utilizado na maioria dos sistemas HDTV (High Definition TV) e ainda será utilizado pelo HD DVD e pelo Blu-ray, sistemas que substituirão os atuais DVDs.

A codificação de vídeo do MPEG-2 é semelhante a usada no MPEG-1, sendo acrescido o suporte à vídeo entrelaçado (diferentemente do anterior, que só podia ser utilizado o vídeo progressivo). Este padrão não é eficiente para baixas bitrates, mas é muito superior aos antecessores em taxas maiores que 3 Mbps.

Embora seja considerado ultrapassado, ainda será encontrado no mercado por muito tempo. Uma outra vantagem do MPEG-2 é que qualquer aparelho feito dentro dos seus padrões é capaz de reproduzir mídias codificadas pelo MPEG-1.

O nome formal deste codificador é ISO/IEC 13818.

2.1.2.4. H.263

O H.263 foi desenvolvido pelo grupo ITU-T (VCEG) e é otimizado para baixas bitrates, sendo utilizado em videoconferências. Sua versão original foi finalizada em 1995, mas outras versões foram lançadas em 1998 e 2000, sendo conhecidas por H.263v2 (ou H.263+) e H.263v3 (ou H.263++), respectivamente.

Este codec foi desenvolvido com base no H.261, no MPEG-1 e no MPEG-2 e pode substituir o primeiro em qualquer bitrate. Além disto, serviu de base para o H.264.

2.1.2.5. MPEG-4 Parte 2

O MPEG-4 tem mais de 20 partes, mas são as partes 2 e 10 responsáveis pela codificação de vídeo. O MPEG-4 parte 10 também é conhecido como H.264 AVC e foi desenvolvido em conjunto com o grupo ITU-T, sendo melhor especificado no próximo item deste texto.

A parte 2 do MPEG-4 ainda é dividida em diferentes profiles, capazes de adaptar o codificador às mais diversas aplicações. Alguns destes especificam algumas poucas características para que um decodificador seja capaz de exibir seqüências de vídeo codificadas com diferentes códigos. Suas aplicações vão desde vídeos para celulares (com imagens de baixa qualidade – bitrates e resoluções baixas) até transmissões de alta definição (HDTV).

Este padrão é referenciado como ISO/IEC 14496-2.

2.2. Aplicações

No início, sons e vídeos eram armazenados analogicamente em fitas magnéticas, como o K7. Com a introdução do CD (*compact disk*), que armazena dados digitais, as fitas magnéticas foram substituídas e os sinais analógicos passaram a ser digitalizados para poderem ser armazenados. Assim começou a pesquisa por métodos que fizessem o tamanho dos dados gerados serem reduzidos, balanceando a qualidade do som ou vídeo, a complexidade dos algoritmos que fazem esta redução e outros fatores.

Hoje, nos deparamos com codificadores o tempo inteiro e nem percebemos. As transmissões digitais do sinal de TV, seja via cabo ou satélite, os discos de DVD que compramos nas lojas, os arquivos de música ou filme nos computadores e até quando

gravamos um DVD, seja no gravador acoplado à TV ou no computador, são exemplos da utilização de diversos codecs.

2.3. Funcionamento básico dos codecs padrões

A maioria dos codificadores de vídeo tem a mesma estrutura básica, sendo diferenciados por alguns recursos e implementações extras, possíveis devido aos avanços tecnológicos, que permitiu a utilização de mais esforço computacional.

A primeira etapa de um codificador começa com a conversão do sinal de RGB para YCbCr. Essa conversão facilita a codificação, pois descorrelaciona as cores, separando a luminância da crominância e, na maioria dos casos, ainda faz uma sub-amostragem da segunda (nos sinais 4:2:2 e 4:2:0). Como a luminância é muito mais importante, por ser mais perceptível ao olho humano, pode ser separado mais espaço para ela e menos para a crominância. Esse sinal é então amostrado e é aplicada uma transformada (a mais comum é a 8×8 DCT – *Discrete Cosine Transform*). Depois, o sinal é compactado por entropia, tendo como resultado uma seqüência binária que representa o arquivo de vídeo.

Para fazer a decodificação deste sinal é aplicado o inverso de cada etapa na ordem inversa do processo de codificação. Exceto pela quantização que, por não poder ser um processo inversível, é feita uma aproximação desta.

Os decodificadores são padronizados para que diferentes empresas possam criar seus codificadores. Desta forma, existem diversas versões de codificadores que produzem diferentes resultados, mas todas seqüências codificadas podem ser exibidas no mesmo decodificador.

2.4. Principais blocos

Existem muitos blocos em cada codificador, mas geralmente são utilizados os mesmos encontrados no H.261, sendo algumas características alteradas e outras removidas. Com o passar do tempo e o avanço tecnológico, se tornou possível utilizar técnicas muito mais pesadas (que exigem muito mais recurso computacional) que as utilizadas pelo H.261, sendo ainda acrescentadas aos codificadores técnicas que haviam sido retiradas nesse meio tempo.

2.4.1. Transformada cosseno discreta (DCT)

Uma transformada muda a nossa maneira de ver o sinal alterando o seu domínio. É como se estivéssemos olhando uma pessoa de lado ao invés de olhámos de frente: é a mesma pessoa, vista apenas por uma outra perspectiva. A mesma coisa acontece com os sinais. Mudar a “forma de olhá-los” não os altera.

A transformada cosseno discreta (*Discrete Cosin Transform – DCT*) é uma das mais utilizadas por mais se aproximar da transformada ótima, a Transformada de Karhunen-Loève (KLT) [8]. Estas transformadas modificam o domínio do sinal do tempo para frequência, concentrando a energia deste em poucos coeficientes e removendo a redundância espacial.

No caso de imagens, é aplicada uma transformada bi-direcional, já que o sinal tem duas dimensões. Qualquer que seja a transformada, ela não é aplicada diretamente sobre a imagem inteira. Primeiramente, a imagem é dividida em blocos menores. É então aplicada uma DCT uni-direcional nas colunas destes blocos e posteriormente nas linhas.

O resultado deste processo é um conjunto de $N \times N$ coeficientes que representam a imagem original no domínio da frequência. Esses coeficientes são chamados de peso das funções base da DCT. O primeiro coeficiente de cada bloco, de coordenadas horizontal e vertical nulas, é chamado de DC e representa o brilho médio deste. Os outros coeficientes são chamados de AC e muitos deles são nulos.

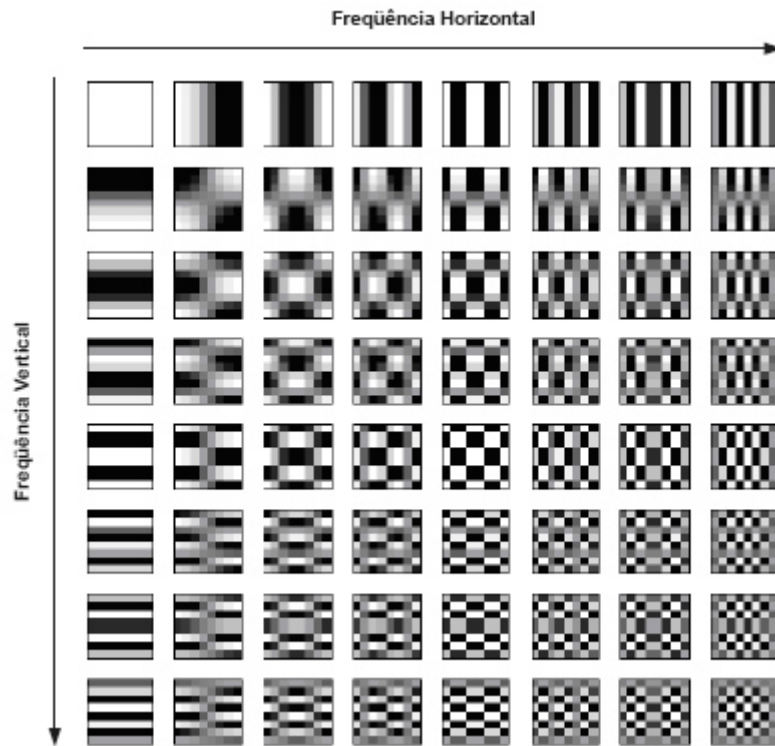


Figura 2.3: Funções Base da DCT 8x8. [5]

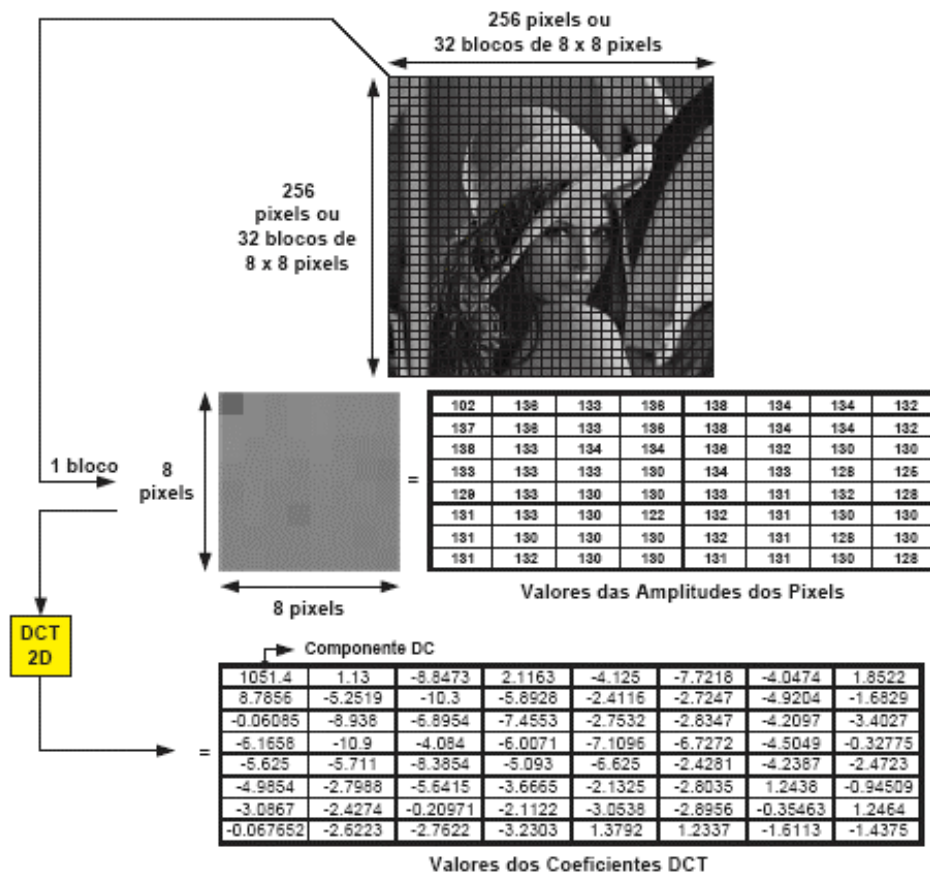


Figura 2.4: Transformada de um macrobloco da imagem LENA, monocromática, de 256x256 pixels. [5]

Posteriormente pode-se reconstruir o sinal combinando-se as funções base da DCT (figura 2.3), com cada função da DCT sendo multiplicada pelo seu peso, ou coeficiente, correspondente.

A DCT não reduz a quantidade de bits necessária para representar cada pixel. Na verdade ela até usa mais bits para representar cada coeficiente. A redução da quantidade de bits necessários para representar cada bloco vem do fato que a transformada concentra energia de baixa frequência e os coeficientes de alta frequência tendem a valores próximos de zero. Assim, não são transmitidos os coeficientes com valores próximos a zero e os outros coeficientes são quantizados e codificados. Isso é fruto da correlação espacial existente entre os pixels próximos num mesmo quadro.

A figura 2.4 ilustra este processo para a imagem LENA, monocromática, de 256x256 pixels.

2.4.2. Quantização

Depois de aplicada a transformada é então feito o processo de quantização. Mais uma vez existem diversos métodos, uns mais eficientes e outros com uma qualidade do sinal de saída melhor. O objetivo da quantização é enviar menos informações (bits) para ser codificada e posteriormente decodificada. Este processo reduz a qualidade da imagem e é o único que é irreversível, então deve-se tomar muito cuidado para não alterar demasiadamente as características originais da imagem.

O erro de quantização é definido como a diferença entre o sinal que entra e o sinal que sai do quantizador. Uma quantização menos precisa é mais rápida, e reduz o número de bits a serem enviados, porém aumentam o erro de quantização, aumentando também a degradação da imagem.



Figura 2.5: Ilustração de uma amostragem degradando a qualidade de uma imagem.

Uma quantização menos delicada é mais tolerável nos coeficientes de alta frequência, pois o ruído gerado é menos perceptível ao olho humano. O erro de quantização é ainda menos perceptível nas componentes de crominancia do que de luminancia.

Tirando proveito da limitação do olho humano, alguns codificadores utilizam uma matriz de peso. Os coeficientes passam então por um critério, usualmente linear, para quantizar os coeficientes de maneiras diferentes.

Existem diferentes tipos de quantizadores. Eles podem ser classificados como escalares ou vetoriais. Os do primeiro grupo mapeiam uma amostra do sinal de entrada para

um valor de saída quantizado. Os do segundo mapeiam um grupo de coeficientes do sinal de entrada para um grupo de coeficientes de saída. Além disto, os quantizadores podem ser separados em uniformes e não-uniformes. Os quantizadores uniformes se diferenciam dos outros por seguirem o mesmo passo para todos os níveis de decisão e de reconstrução.

Na figura 2.6 podemos ver dois exemplos de quantizadores escalares, um uniforme e outro não-uniforme. E figura 2.7 é um exemplo de quantização de coeficientes originados de uma DCT com diferentes passos de quantização. Conforme é aumentado o passo da quantização mais coeficientes vão a zero, reduzindo o número de bits a serem transmitidos.

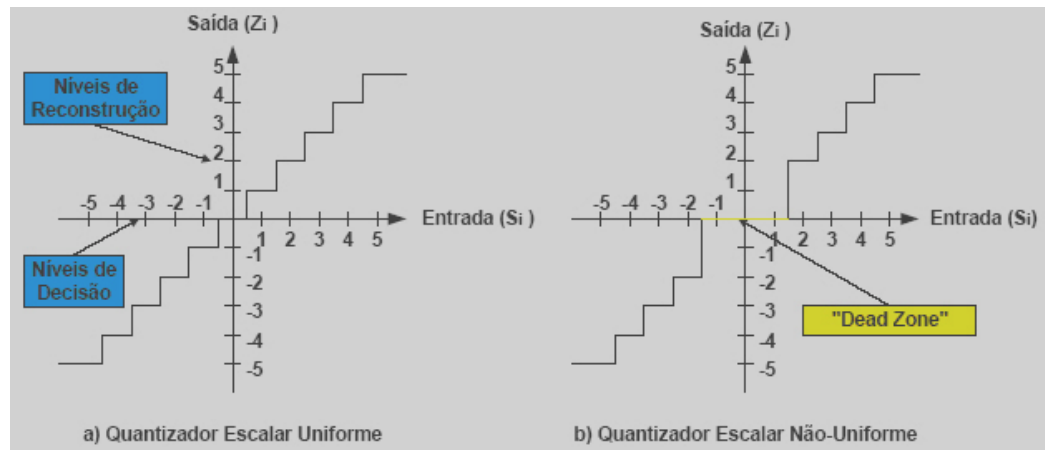


Figura 2.6: Quantizadores escalares. [5]

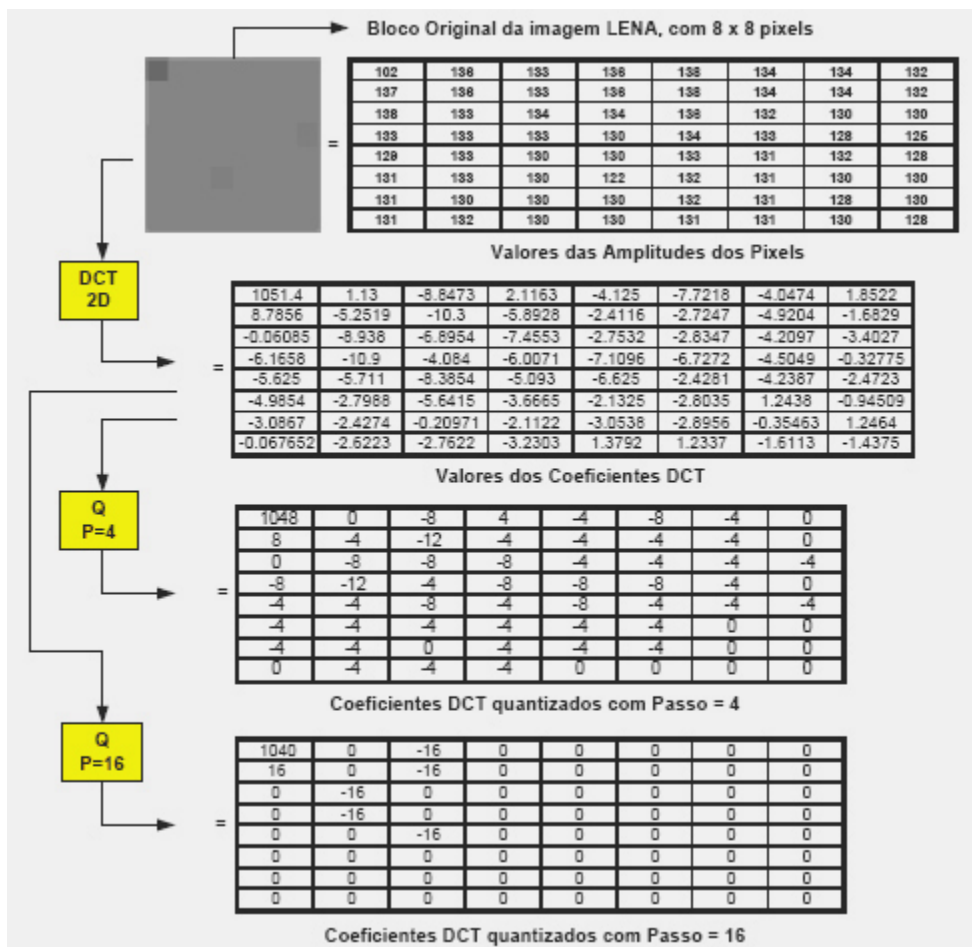


Figura 2.7: Coeficientes de quantização com diferentes

2.4.3 Varredura zig-zag

Existem dois tipos de varreduras zig-zag: a progressiva e a entrelaçada. As duas têm o mesmo objetivo: reordenar os coeficientes depois de aplicada a transformada e feita a quantização a fim de agrupar os coeficientes não-nulos, permitindo uma representação mais eficiente para os coeficientes de valores nulos. A figura 2.8 ilustra os dois tipos de varredura.

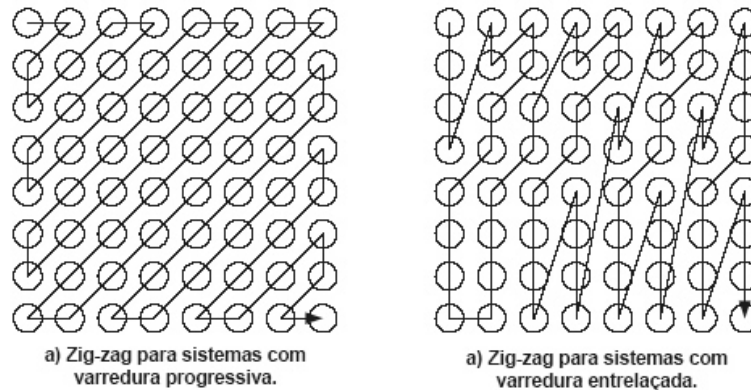


Figura 2.8: Diferentes tipos de varredura. [5]

Exemplo: A reordenação do primeiro quadro da imagem LENA (utilizando a quantização com passo 16) seria:

1040, 0, 16, 0, 0, -16, 0, -16, -16, 0, 0, -16, 0, 0, 0, 0, 0, 0, 0, 0, 0, -16, ...

Após a reordenação, aplicamos à seqüência uma compactação por Corrida de Zeros. Neste caso, a seqüência com a Codificação por Corridas sendo aplicada, os coeficientes passam a ser:

(0,1040) (1,16) (2,-16) (1,-16) (0,-16) (2,-16) (11,-16) (EOB)

2.4.4. Predição temporal (inter quadros)

Esta predição trata da redundância temporal existente entre os quadros, aproveitando as informações contidas em um quadro de referência para a codificação de outros, passados e até futuros. Esse procedimento não pode ser realizado na imagem original: primeiro porque as imagens originais não estarão disponíveis no decodificador, e segundo porque as imagens, depois de quantizadas e realizado todo esse processo até agora, são diferentes das originais.

Na predição temporal existem três tipos quadros (no H.264 ainda existem mais dois):

- Quadros I (intracodificados): Nestes quadros não é feita a predição temporal, apenas a espacial (ou intra).
- Quadros P (predito): Estes quadros são preditos com base em quadros passados, sejam eles do tipo I ou outros quadros do tipo P e ainda são utilizados para a predição de outros quadros, futuros, do tipo P ou B.
- Quadros B (bipredito): Estes quadros fazem sua predição utilizando quadros I e P passados e ainda quadros futuros, oferecendo uma maior compressão.

O H.264 ainda utiliza dois tipos novos de quadros, os SP (*Switching P*) e os SI (*Switching I*). (Mais informações na seção 3.3 deste mesmo texto.)

Nenhum macrobloco de um quadro I usa codificação inter, todos eles são intracodificados. Nos quadros P, pode-se ter macroblocos preditos ou intracodificados. Neste último caso, a predição é nula. E, nos quadros B, pode existir macroblocos interpolados (bipreditos, utilizando quadros de referência passados e futuros), preditos ou intracodificados. Neste último caso, mais uma vez os vetores de movimento são nulos ou iguais ao do último macrobloco, sendo feita a reconstrução no decodificador com base nos quadros de referência sem a predição de movimentos.

No codificador existem parâmetros que definem os grupos chamados de GOP's (*Group of Pictures*). Cada GOP é um conjunto de quadros que possibilitam o acesso randômico. Geralmente são compostos de 12 a 15 quadros (essa quantidade proporciona uma melhor relação entre tamanho e recurso computacional vs. qualidade da imagem codificada) [5]. Além de definirmos o número de quadros em cada GOP, também escolhemos a quantidade de quadros P's e B's. Quanto maior for a quantidade de quadros B's menor é o tamanho final da seqüência de vídeo, mas também tem-se uma demanda por recurso computacional. Quanto menos quadros B's cada GOP tiver, melhor será a qualidade das imagens. Uma boa relação entre quadros B's e P's é de três para um [5]. Cada GOP contém apenas um quadro I.

Para que seja possível a codificação de quadros B, que utilizam quadros de referência passados e futuros, é necessário fazer uma reordenação de cada GOP da seqüência de vídeo. Antes de serem codificados, os quadros estão na ordem temporal, que é a mesma ordem de apresentação. Na codificação, estes quadros são reorganizados para que o quadro I de cada GOP fique antes de todos os quadros P's e B's da mesma e para que os quadros P's se posicionem antes de todos os quadros B's que o utilizam como referência.

Na decodificação é feita novamente uma reordenação dos quadros para que estes voltem às suas posições originais, a de exibição. A figura a seguir pode ilustrar melhor esta reordenação.

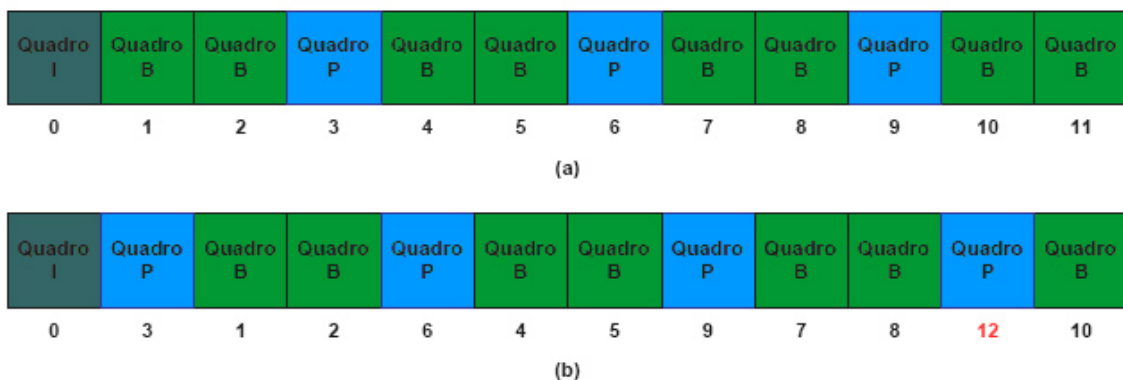


Figura 2.9: (a) quadros dispostos na ordem original, temporal; e (b) quadros dispostos na ordem de codificação. [5]

3. H.264 AVC / MPEG-4 Parte 10

O padrão de codificação de vídeo H.264 AVC (ou MPEG-4 parte 10) foi aprovado em outubro de 2003 pela *Joint Video Team* (JVT), grupo formado por especialistas do *ITU-T Video Coding Experts Group* (VCEG) e *ISO/IEC Moving Picture Experts Group* (MPEG). O JVT tinha como objetivo dobrar a eficiência de codificação em relação aos melhores padrões existentes na época (como o MPEG-2, padrão largamente difundido e utilizado nos DVDs atuais, na maioria das transmissões digitais de televisão, nos computadores domésticos e etc), com vídeo de alta qualidade em todas as taxas e resoluções, adaptação tanto a aplicações com baixo ou alto retardo, robustez a erros, além de outras vantagens, podendo ser aplicado tanto

em mídias portáteis (como no caso do Blu-Ray e do HD DVD, formatos que substituirão os atuais DVDs) quanto em transmissões digitais de televisão (tanto via cabo quanto via satélite), podendo ainda ser utilizado em vídeo conferências e em outras situações.

É de interesse um estudo detalhado do H.264 AVC para sua utilização no SBTVD (Sistema Brasileiro de Televisão Digital), pois oferece, em comparação ao MPEG-2, a possibilidade de se colocar o dobro de canais em uma mesma faixa de transmissão de dados, pois tem, em média, um ganho de mais de 50%, em termos de PSNR (*Peak Signal-to-noise ratio*). Por outro lado, o H.264/AVC exige um recurso computacional muito maior, tanto nos codificadores quanto nos decodificadores, o que representa um custo elevado para os consumidores finais, ainda mais considerando que é uma tecnologia nova e seus aparelhos ainda não são produzidos em larga escala.

3.1. Princípios de codificação de vídeo

Se seguidas as definições da norma ITU-R-BT-601 [1] seriam necessários 270 Mbps para a transmissão de um único canal de TV em resolução normal (SDTV), se tornando impossível fazê-la em um único canal de 6 MHz utilizado para este fim. Para esta transmissão seriam necessários mais de uma dezena destes canais, já que, com a atual tecnologia, só é possível colocar neles até 20 Mbps. No caso da televisão de alta definição (HDTV), se seguida a norma ITU-R-BT-709 [2], seria ainda mais inviável fazer esta transmissão, já que seria necessária a utilização da taxa de 1,5 Gbps.

Os codificadores são sistemas que atuam exatamente na redução destas taxas para que se tornem possíveis as transmissões em um único canal de 6 MHz, e em alguns casos, é possível a transmissão de mais de um canal de televisão num único canal deste. Para conseguir fazer isto estes sistemas removem informações redundantes dos sinais de vídeo. Existem três tipos de redundância:

- Redundância espacial: num quadro qualquer de uma seqüência de vídeo, pixels vizinhos geralmente são correlacionados, parecidos. Assim, os codificadores reescrevem as informações do quadro aproveitando as informações de pixels anteriores, gastando menos bits para representar cada pixel. Esse mecanismo reduz um pouco a qualidade da imagem, mas é imperceptível ao olho humano. Essa redundância é explorada na predição intra.
- Redundância temporal: em muitos casos, um pixel não muda muito de posição de um quadro para o seguinte, podendo até não se mover. Imagine um vídeo de telejornal. O fundo da imagem não muda com frequência e mesmo o primeiro plano (o âncora) não se move muito. Assim, um codificador conseguiria reduzir bastante o tamanho do vídeo re-escrevendo cada pixel se baseando nos valores de pixels dos quadros passados e até dos futuros. Essa redundância é explorada na predição.
- Redundância Psico-Visual: a capacidade de visualização do olho humano tem limite para distinção de detalhes e de acompanhar imagens em movimentos rápidos. Os codificadores retiram informações que nem sequer poderíamos ver reduzindo a qualidade do vídeo até o limite do olho humano.

A exploração destes recursos geralmente não afeta a informação de maneira que possamos distinguir os sinais codificados do original.



Figura 3.1: Diferença entre quadros; (em cima) sem compensação de movimentos; e (embaixo) com compensação.

3.2. Principais diferenças em relação aos padrões anteriores

O H.264 reincorporou uma série de recursos que haviam sido retirados do H.261 por necessitarem de muito recurso computacional e ainda inclui outras ferramentas, novas, capazes de aumentar muito o nível de compressão de uma seqüência de vídeo, se tornando muito superior aos codificadores que o antecederam e atualmente o mais eficiente, além de ser aplicável a uma infinidade de aplicações através da seleção dos seus perfis e levels.

Dentre as mudanças e principais recursos encontramos:

- Compensação de movimento muito mais eficiente em comparação com os codificadores que o antecederam, utilizando até 32 quadros de referência. Este recurso oferece um ganho na compressão do vídeo, embora não seja adequado o seu uso em seqüências com movimentos muito rápidos e em seqüências que tenham pouca relação entre os seus quadros.
- O tamanho dos blocos utilizados para compensação de movimento pode ser escolhido, variando desde 4 x 4 até 16 x 16, permitindo uma acuridade surpreendente para a predição de movimentos. A precisão desta ainda foi melhorada para $\frac{1}{4}$ de pixel, tornando-se ainda mais preciso.
- O filtro anti-blocos é outro artifício retirado dos codificadores que agora foi reincorporado, oferecendo uma qualidade muito superior aos vídeos codificados em baixas taxas de bits.
- Um dos métodos de compactação utilizados é a CABAC, que oferece grande compactação ao arquivo de vídeo. Ainda existe a compactação de tamanho variável (Variable-Lenght Coding – VLC) CAVLC, que não tem tanto poder de compactação, mas que é uma alternativa para equipamentos de menor poder de processamento. E, onde não são aplicados nenhum destes dois métodos, é utilizada uma compactação de tamanho variável chamada Exp-Golomb.
- Quadros SI e SP, mas que não são permitidos em todos os perfis.
- Maior proteção contra erros com recursos como reordenação de blocos, o envio de alguns quadros extra (normalmente em taxas mais baixas) entre outros.

Estes e outros recursos dão ao H.264 a possibilidade de ser utilizado para muitos fins, sendo robusto a erros e oferecendo a mesma qualidade de vídeo utilizando taxas de bits muito menores.

3.3. Estrutura

A estrutura do H.264 é muito semelhante a dos seus antecessores, como o MPEG-2, como pode ser visto na figura abaixo, que representa os principais blocos de um codificador H.264.

A primeira etapa deste codificador é a mudança de RGB, modo como as imagens são capturadas, ainda analógicas, para YCbCr. Esta mudança descorrelaciona as cores do vídeo e permite separar mais bits para a codificação de Y, parte relativa à luminância, mais importante por ser mais perceptível ao olho humano. Depois disto, cada quadro é dividido em blocos para então ser codificado. Há também uma divisão da seqüência em grupos de quadros chamados de GOP (*Group Of Pictures*). O primeiro quadro de uma GOP, do tipo I, é intra codificado. Este tipo de quadro só tem codificação espacial e não depende de outros quadros para ser codificado, não existindo compensação e predição de movimentos. Um pixel de um quadro intracodificado utiliza outros pixels da sua vizinhança para ser representado de forma mais “econômica” (gastando menos bits) e depois são codificados.

Os outros quadros (dos tipos P e B) utilizam predições intra e inter. Além de terem como referência os pixels vizinhos para codificar cada um de seus pixels, também utilizam outros quadros como referência. Estes quadros não são codificados, mas são comparados com os quadros originais e a diferença entre eles é codificada. Este método permite economizar

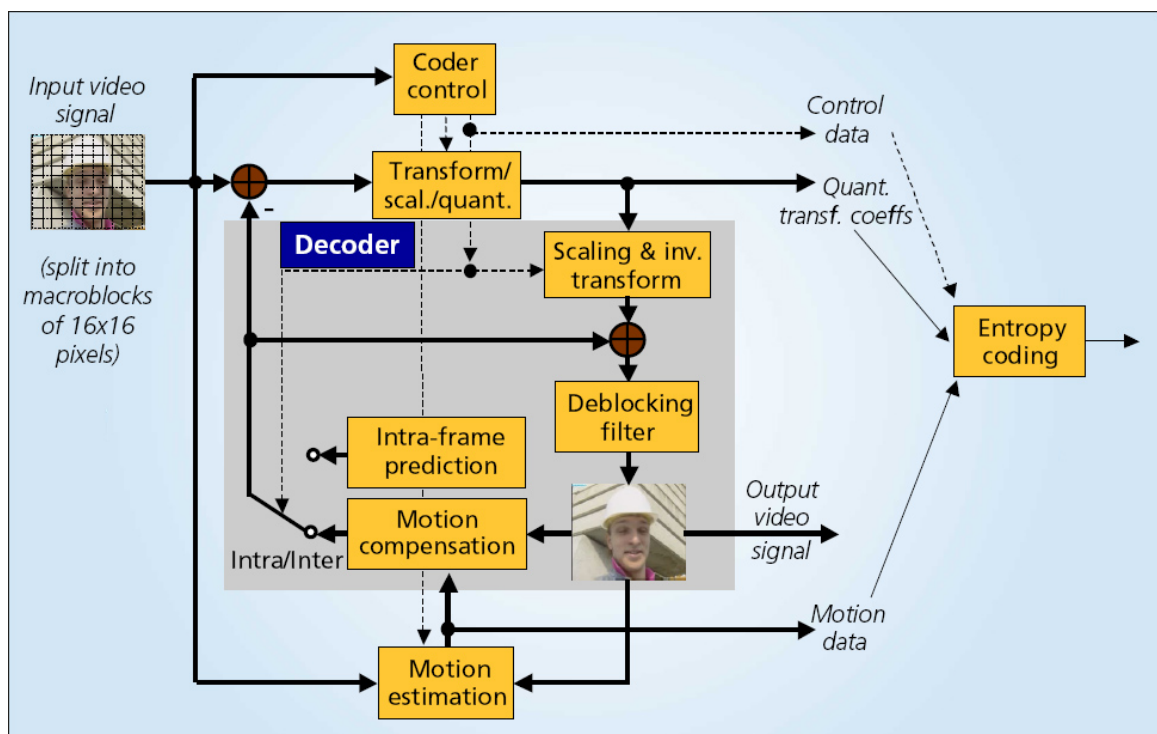


Figura 3.2: Diagrama de blocos para a codificação de um macrobloco. [9]

muitos bits para representar a mesma seqüência de vídeo.

No H.264 ainda foram introduzidos dois novos tipos de quadros: os quadros do tipo SI (*Switching I*) e do tipo SP (*Switching P*). Os quadros do segundo tipo tem a quantização da predição temporal realizada no domínio da freqüência. E os quadros do tipo SI são cópias de

quadros SP em casos onde a predição temporal não pode ser utilizada para evitar erros. Este recurso não está disponível em todos os perfis. [9]

Depois de feitas as predições, todos os quadros sem exceção são quantizados e compactados. Diversas técnicas de compactação são aplicadas, podendo, inclusive, ser utilizada mais de uma e mais de uma vez.

3.3.1. Predição espacial (intra quadros)

A predição espacial no H.264 é feita no domínio do tempo, diferentemente de outros codificadores, que utilizam como referência macroblocos ainda no domínio da frequência. Ao se fazer a predição, o macrobloco que está sendo codificado utiliza macroblocos vizinhos já codificados e decodificados. Para isso, todo codificador tem embutido um decodificador.

Existem dois modos para a codificação intra: INTRA-4x4 e INTRA-16x16. No primeiro modo, cada bloco de 4x4 pixels da luminancia utiliza uma das nove opções possíveis para realizar a predição. No segundo, existem apenas quatro opções. Os quadros de crominancia, por serem menos importantes (menos perceptíveis pelo olho humano), são preditos de maneira semelhante a INTRA-16x16. [6]

3.3.2. Predição temporal (inter quadros)

Podemos escolher dentre quatro opções para realizar a predição temporal nos quadros P's: INTER-16x16, INTER-16x8, INTER-8x16 e INTER-8x8. Estes modos indicam o tamanho dos blocos da luminancia em que serão aplicados a predição. Quando a opção INTER-8x8 for escolhida ainda existem mais quatro opções para sub-dividir cada bloco em sub-blocos. Assim, pode-se dividir cada bloco de 8x8 pixels em até quatro sub-blocos de tamanhos iguais. A figura a seguir (3.3) ilustra melhor estas opções. As componentes da crominancia sempre utilizam interpolação para a codificação inter. [9]

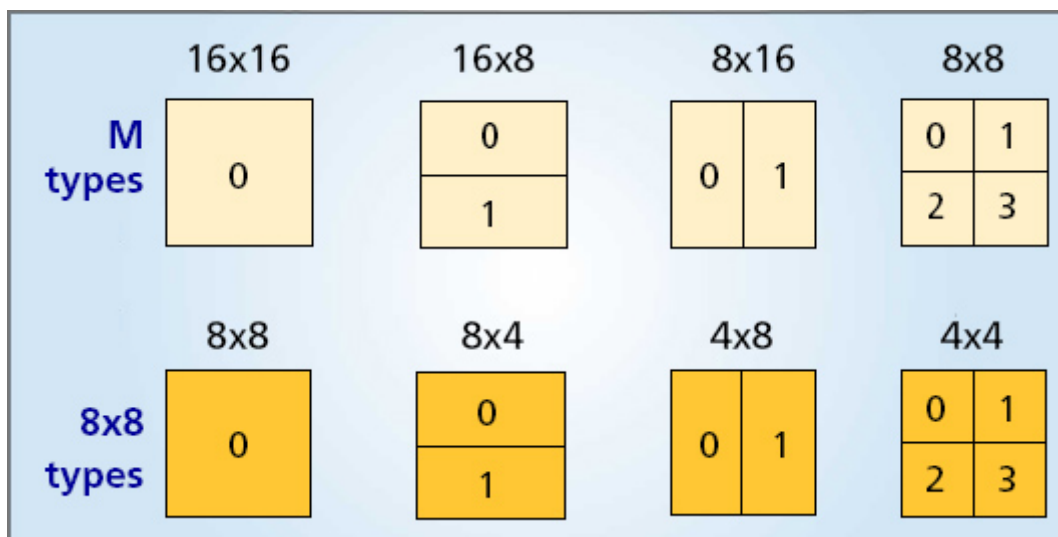


Figura 3.3: Divisões de cada macrobloco para predição temporal. Em cima, as opções de tamanhos para cada macrobloco. Embaixo as divisões de cada sub-bloco quando é escolhida a opção 8x8. [9]

O H.264 ainda oferece suporte a *multi-picture motion-compensated prediction*, que significa que mais de um quadro pode ser utilizado para fazer a predição temporal. Para isso é necessário que tanto o codificador quanto o decodificador mantenham as imagens que servirão de referência armazenadas num buffer.

Nos quadros B's a predição temporal é semelhante à utilizada nos quadros P's. A principal diferença é que este tipo de quadro pode referenciar tanto quadros passados como quadros futuros (utilizando a ordem temporal como base para tal comparação). Para isso é utilizada uma média ponderada de cada referencia. [9]

Para prevenir erros, um recurso chamado de FMO (*Flexible Macroblock Ordering*) é utilizado. O FMO agrupa macroblocos enviando-os separadamente. Se um destes grupos for perdido, é possível utilizar outros macroblocos para tentar amenizar o erro. Caso ocorra um erro como este, existem diversos padrões para recompor a imagem.

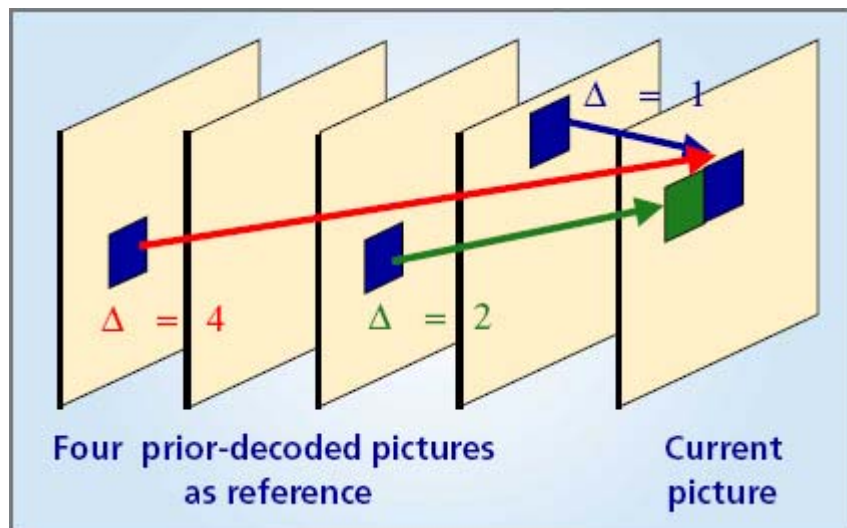


Figura 3.4: predição temporal utilizando mais de um quadro como referência. [9]

3.3.3. Transformada

O H.264 também aplica a transformada nos erros (diferença entre a imagem original e a imagem alterada pelo codificador) ao invés da imagem original, requerendo menos bits para a representação da imagem codificada. Porém é utilizada uma transformada semelhante e com as mesmas características da DCT (Discrete Cosine Transform) em blocos de 4 x 4 pixels. Esta transformada ainda oferece uma função inversa melhor que a DCT. Ainda é aplicada uma transformada extra aos quatro coeficientes de cada componente de cromaticidade. Uma outra transformada ainda é aplicada a cada coeficiente DC da luminância, se foi escolhida a opção INTRA-16x16 para predição espacial.

3.3.4. Quantização

Para a quantização dos coeficientes da transformada o H.264 utiliza, para cada macrobloco, uma quantização escalar escolhida num universo de 52 opções pelo QP (Quantization Parameter – Parâmetro de Quantização).

Depois de quantizados, os coeficientes ainda sofrem uma varredura ZigZag e codificação por entropia.

3.3.5. Codificação por entropia

In H.264/AVC, two methods of entropy coding are supported. The default entropy coding method uses a single infinite-extend codeword set for all syntax elements, except the quantized transform coefficients. Thus, instead of designing a different VLC table for each syntax element, only the mapping to the single codeword table is customized according to the data statistics. The single codeword table chosen is an exp-Golomb code with very simple and regular decoding properties.

For transmitting the quantized transform coefficients, a more sophisticated method called Context-Adaptive Variable Length Coding (CAVLC) is employed. In this scheme, VLC tables for various syntax elements are switched, depending on already-transmitted syntax elements. Since the VLC tables are well designed to match the corresponding conditioned statistics, the entropy coding performance is improved in comparison to schemes using just a single VLC table.

The efficiency of entropy coding can be improved further if Context-Adaptive Binary Arithmetic Coding (CABAC) is used [3]. On the one hand, the use of arithmetic coding allows the assignment of a non-integer number of bits to each symbol of an alphabet, which is extremely beneficial for symbol probabilities much greater than 0.5. On the other hand, the use of adaptive codes permits adaptation to non-stationary symbol statistics.

Another important property of CABAC is its context modelling. The statistics of already-coded syntax elements are used to estimate the conditional probabilities. These conditional probabilities are used for switching several estimated probability models. In H.264/AVC, the arithmetic coding core engine and its associated probability estimation are specified as multiplication-free low-complexity methods, using only shifts and table look-ups. Compared to CAVLC, CABAC typically provides a reduction in bit-rate of between 10 - 15% when coding TV signals at the same quality.

[escrever!]

3.3.6. Filtro anti blocagem (*In-loop Deblocking Filter*)

Geralmente imagens codificadas (principalmente a baixas bitrates) tendem a blocadas (com blocos de pixels aparentes). Esta característica é uma das mais perceptíveis, que degradam mais a qualidade subjetiva das imagens. Com este filtro é possível diminuir a taxa de bits de 5 a 10% obtendo-se uma imagem com mesma qualidade subjetiva.



Figura 3.5: (esquerda) quadro sem a aplicação do filtro anti-blocos; (direita) com a aplicação.

3.3.7. Considerações finais sobre a estrutura do H.264

O H.264 não é só o codificador de vídeo mais moderno, mas também o mais eficiente se comparado com qualquer um dos padrões que o antecederam, podendo ser utilizado em qualquer tipo de aplicação. Combinando seus perfis e levels mais básicos podemos utilizá-lo em vídeo-conferências, com imagens pequenas e uma qualidade de vídeo baixa, comum para este tipo de aplicação. Selecionando configurações um pouco melhores podemos aplica-lo em câmeras de vigilância (inclusive já existem aparelhos que o utilizam

desta forma). Melhorando estas características, colocamos o H.264 na televisão digital com definição normal (SDTV) ou alta (HDTV). Ainda podemos ir mais a frente e configurá-lo para utilizações avançadas (publico *hi-end*), com definições que nem podem ser exibidas pelos equipamentos existentes.

3.4. Profiles

Originalmente o H.264 tinha somente três profiles (baseline, main e extended). Posteriormente foram adicionados mais 4 profiles para codificações de alta qualidade (high, high 10, high 4:2:2, high 4:4:4).

3.4.1. Baseline profile

Este perfil foi desenvolvido para ser usado principalmente por aplicações que requerem menos processamento e de custo mais baixo. Essas aplicações não necessitam de alta qualidade (resoluções e bitrates baixas), como videoconferências e vídeos em celulares.

3.4.2. Main profile

O objetivo principal desde profile era atender a maior parte dos consumidores, balanceado qualidade e recurso computacional, sendo utilizado tanto para armazenagem quanto para transmissões. Mas posteriormente foi desenvolvido o High profile, que se tornou mais adequado para estas aplicações.

3.4.3. Extended profile

Desenvolvido para aplicações de streaming (seqüências de vídeo utilizadas na internet) com alta capacidade de compressão e algumas características para prevenção de erros.

3.4.4. High profile

Este foi o perfil adotado por aplicações de alta definição, em especial pelos sucessores do DVD (o HD DVD e o Blu-Ray) e ainda para as transmissões de televisão de alta qualidade (HD TV).

3.4.5. High 10 profile

Este perfil vai além da capacidade de reprodução dos produtos domésticos existentes hoje no mercado. Baseado no High profile, ainda foi adicionada a capacidade de codificação com suporte a 10 bits por quadro.

3.4.6. High 4:2:2 profile

Desenvolvido para aplicações profissionais que utilizam vídeo entrelaçado. Este perfil foi baseado no High 10 profile, permitindo a utilização do formato de croma 4:2:2.

3.4.7. High 4:4:4 profile

Originalmente desenvolvido para aplicações que exijam extremamente alta qualidade, este profile pode codificar seqüências em formato 4:4:4, ainda tem suporte para codificação a 12 bits por quadro e outras características mais, sendo bem superior aos dois anteriores. Mas atualmente este perfil foi abandonado para o desenvolvimento de um outro com as mesmas qualidades.

Coding Tools	Baseline	Main	Extended
I and P Slices	X	X	X
CAVLC	X	X	X
CABAC		X	
B Slices		X	X
Interlaced Coding (PicAFF, MBAFF)		X	X
Enh. Error Resil. (FMO, ASO, RS)	X		X
Further Enh. Error Resil (DP)			X
SP and SI Slices			X

Figura 3.6: Profiles originais do H.264 e suas características. [10]

Coding Tools	High	High 10	High 4:2:2	High 4:4:4
Main Profile Tools	X	X	X	X
4:2:0 Chroma Format	X	X	X	X
8 Bit Sample Bit Depth	X	X	X	X
8x8 vs. 4x4 Transform Adaptivity	X	X	X	X
Quantization Scaling Matrices	X	X	X	X
Separate Cb and Cr QP control	X	X	X	X
Monochrome video format	X	X	X	X
9 and 10 Bit Sample Bit Depth		X	X	X
4:2:2 Chroma Format			X	X
11 and 12 Bit Sample Bit Depth				X
4:4:4 Chroma Format				X
Residual Color Transform				X
Predictive Lossless Coding				X

Figura 3.7: Profiles avançados do H.264 e suas características. [10]

3.5. Levels

Levels também definem algumas características do codificador. No H.264 AVC, os levels são cinco, com subdivisões. Segue uma tabela com estas restrições.

Level Number	Typical Picture Size	Typical frame rate	Maximum compressed bit rate (for VCL) in Non-FRExt profiles	Maximum number of reference frames for typical picture size
1	QCIF	15	64 kbps	4
1b	QCIF	15	128 kbps	4
1.1	CIF or QCIF	7.5 (CIF) / 30 (QCIF)	192 kbps	2 (CIF) / 9 (QCIF)
1.2	CIF	15	384 kbps	6
1.3	CIF	30	768 kbps	6
2	CIF	30	2 Mbps	6
2.1	HHR (480i or 576i)	30 / 25	4 Mbps	6
2.2	SD	15	4 Mbps	5
3	SD	30 / 25	10 Mbps	5
3.1	1280x720p	30	14 Mbps	5
3.2	1280x720p	60	20 Mbps	4
4	HD Formats (720p or 1080i)	60p / 30i	20 Mbps	4
4.1	HD Formats (720p or 1080i)	60p / 30i	50 Mbps	4
4.2	1920x1080p	60p	50 Mbps	4
5	2kx1k	72	135 Mbps	5
5.1	2kx1k or 4kx2k	120 / 30	240 Mbps	5

Figura 3.8: Levels do H.264 e suas características. [10]

3.6. Comparações

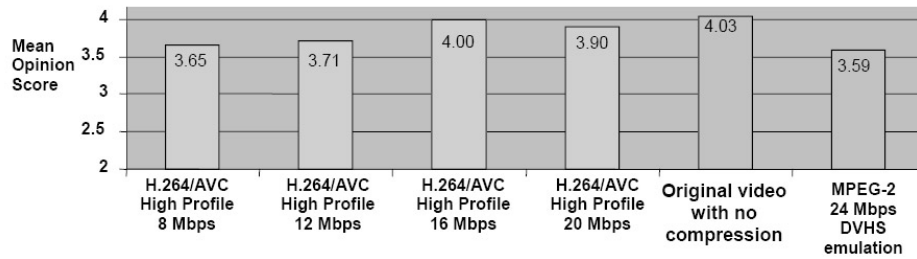


Fig. 8: Perceptual Test of FRExt High Profile Capability by Blu-ray Disc Association [7]

Figura 3.9: Análise subjetiva de uma mesma seqüência codificada de várias formas diferentes, inclusive pelo MPEG-2. Acima de 3 pontos é aceitável a utilização para alta definição. [13]

Coder	Average Bit Rate Savings Relative To:		
	MPEG-4 ASP	H.263 HLP	MPEG-2
H.264/AVC MP	37.44%	47.58%	63.57%
MPEG-4 ASP	-	16.65%	42.95%
H.263 HLP	-	-	30.61%

Coder	Average Bit Rate Savings Relative To:		
	H.263 CHC	MPEG-4 SP	H.263 Base
H.264/AVC BP	27.69%	29.37%	40.59%
H.263 CHC	-	2.04%	17.63%
MPEG-4 SP	-	-	15.69%

Figura 3.10: Comparação das bitrates utilizadas para se obter uma qualidade no H.264 e outros codecs. [11]

Sequence	Coding Gain
IRENE	53%
MOBILE & CALENDAR	58%
PARIS	42%
STUDENTS	45%
TEMPETE	41%
Average Gain	49.2%

Table 3. Coding gain of H.264 over H.263 Baseline Profile for various CIF sequences.

Sequence	Coding Gain
IRENE	29%
MOBILE & CALENDAR	31%
PARIS	22%
STUDENTS	24%
TEMPETE	20%
Average Gain	25.4%

Table 5. Coding gain of H.264 over H.263 Conversational High Compression Profile for various CIF sequences.

Sequence	Coding Gain
AKIYO	51%
CAR PHONE	40%
CLAIRE	48%
COAST GUARD	38%
CONTAINER	41%
FOREMAN	42%
IRENE	50%
MISS AMERICA	44%
MOBILE & CALENDAR	48%
NEWS	56%
PARIS	52%
TEMPETE	37%
AVERAGE GAIN	45.6%

Table 4. Coding gain of H.264 over H.263 Baseline Profile for various QCIF sequences.

Sequence	Coding Gain
AKIYO	25%
CAR PHONE	20%
CLAIRE	34%
COAST GUARD	19%
CONTAINER	22%
FOREMAN	20%
IRENE	26%
MISS AMERICA	25%
MOBILE & CALENDAR	22%
NEWS	28%
PARIS	22%
TEMPETE	17%
AVERAGE GAIN	23.3%

Table 6. Coding gain of H.264 over H.263 Conversational High Compression Profile for various QCIF sequences.

Figura 3.11: Comparação do ganho obtido utilizando o H.264 e o MPEG-2. [12]

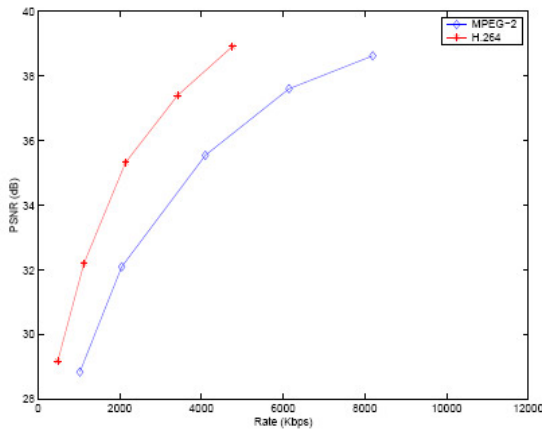


Fig. 1. Rate-PSNR curves of CAROUSEL sequence encoded using H.264 and MPEG-2 encoders.

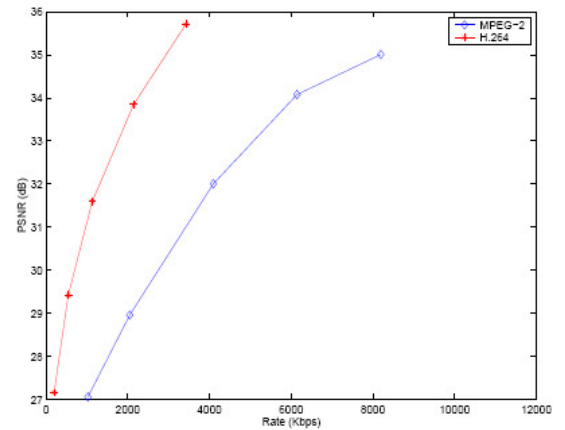


Fig. 2. Rate-PSNR curves of FLOWER GARDEN sequence encoded using H.264 and MPEG-2 encoders.

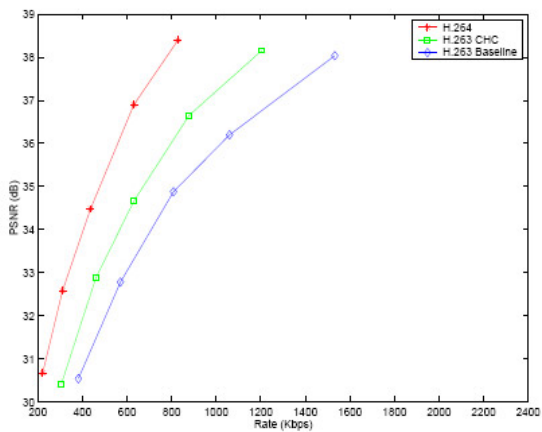


Fig. 3. Rate-PSNR curves of CIF PARIS sequence encoded using H.264, H.263 Baseline and H.263 CHC encoders.

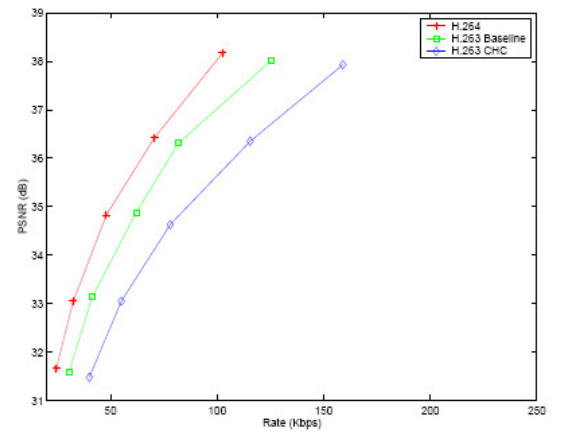


Fig. 4. Rate-PSNR curves of QCIF FOREMAN sequence encoded using H.264, H.263 Baseline and H.263 CHC encoders.

Figura 3.12: Gráficos comparando os ganhos obtidos pela utilização do H.264 no lugar do MPEG-2. [12]

4. Resultados

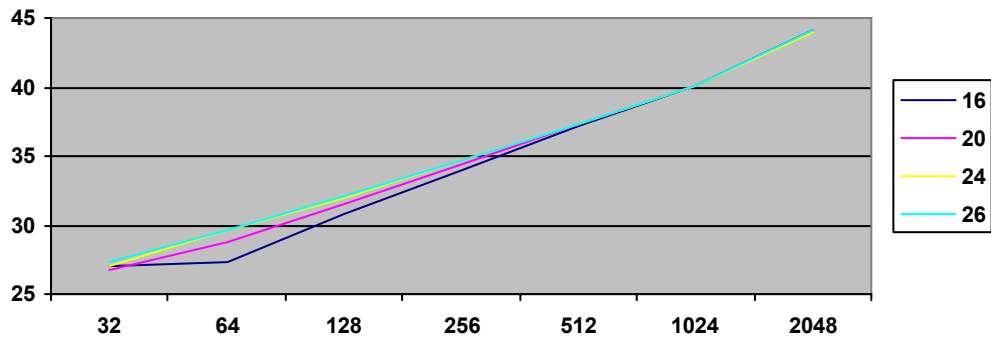
4.1. Testes realizados

Todos os testes foram realizados com o codificador de referência JM11, desenvolvido pelo mesmo grupo que criou o padrão H.264, o JVT (*Joint Video Team*).

Além da taxa de bits (bitrate), também foi feita a variação do parâmetro de quantização (QP), seguindo as instruções do texto VCEG-N81d1, de 27 de setembro de 2001. Neste mesmo texto ainda é possível encontrar outras sugestões sobre as condições em que o codificador deve ser analisado variando outros parâmetros.

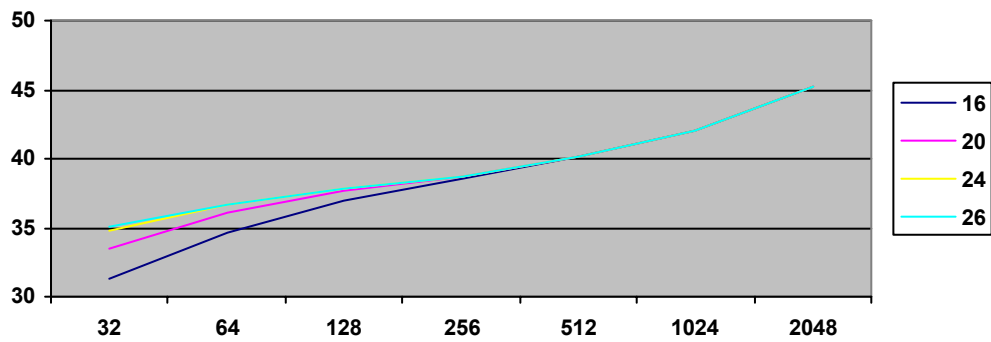
Os resultados obtidos podem ser encontrados nos gráficos e tabelas abaixo.

Seqüências CIF (352 x 288px)
bridge-close.cif



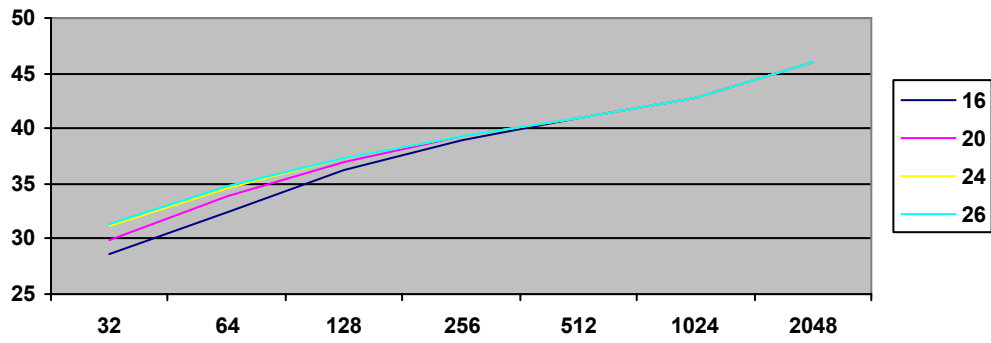
QP/bitrate	32	64	128	256	512	1024	2048
16	26,97	27,39	30,78	34,04	37,11	40,12	44,07
20	26,72	28,76	31,57	34,48	37,29	40,11	44,07
24	27,00	29,57	32,00	34,74	37,29	40,10	44,03
26	27,38	29,71	32,09	34,70	37,28	40,10	44,06

bridge-far.cif



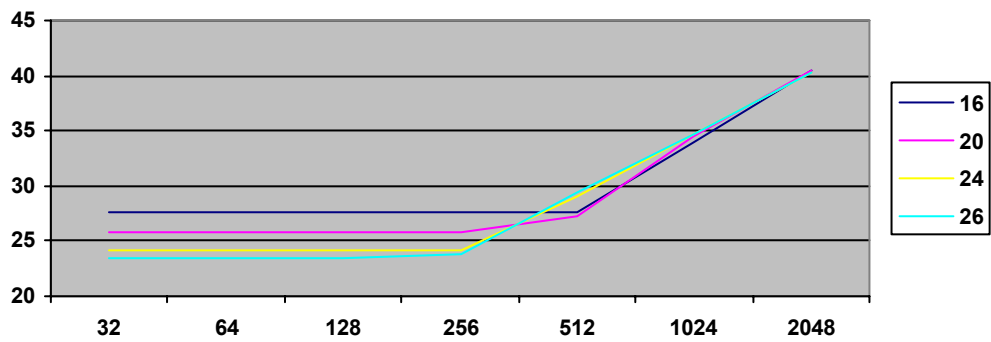
QP/bitrate	32	64	128	256	512	1024	2048
16	31,34	34,67	36,95	38,56	40,09	42,02	45,28
20	33,51	36,06	37,61	38,76	40,10	42,02	45,28
24	34,85	36,63	37,76	38,76	40,10	42,03	45,27
26	35,11	36,71	37,76	38,75	40,11	42,02	45,27

highway.cif



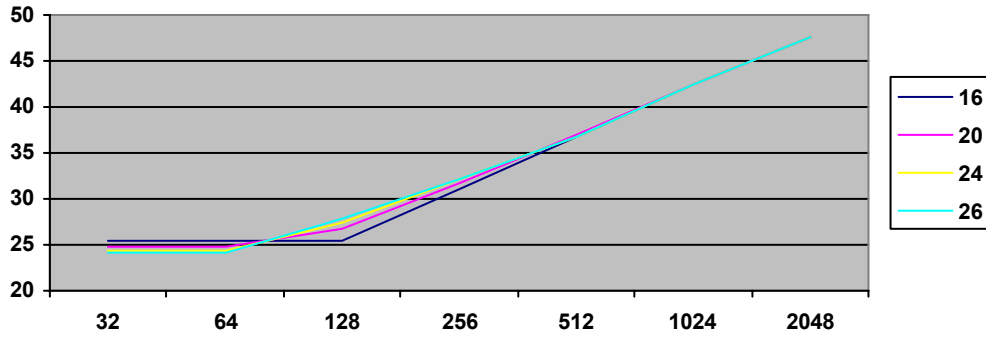
QP/bitrate	32	64	128	256	512	1024	2048
16	28,59	32,37	36,21	38,96	40,88	42,83	45,99
20	29,95	33,96	37,00	39,31	40,89	42,82	45,99
24	31,07	34,56	37,37	39,34	40,89	42,82	45,99
26	31,31	34,74	37,38	39,34	40,89	42,83	45,99

mobile.cif



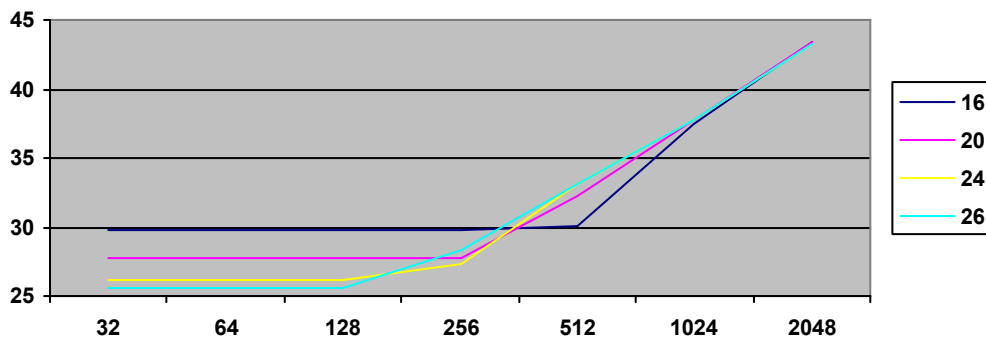
QP/bitrate	32	64	128	256	512	1024	2048
16	27,59	27,59	27,59	27,59	27,59	33,86	40,39
20	25,77	25,77	25,77	25,77	27,23	34,46	40,39
24	24,17	24,17	24,17	24,17	29,13	34,61	40,38
26	23,5	23,5	23,5	23,8	29,48	34,61	40,34

paris.cif



QP/bitrate	32	64	128	256	512	1024	2048
16	25,36	25,36	25,43	31,09	36,83	42,48	47,65
20	24,87	24,87	26,77	31,81	36,92	42,46	47,64
24	24,4	24,4	27,49	32,09	36,83	42,45	47,61
26	24,18	24,18	27,72	32,08	36,79	42,43	47,57

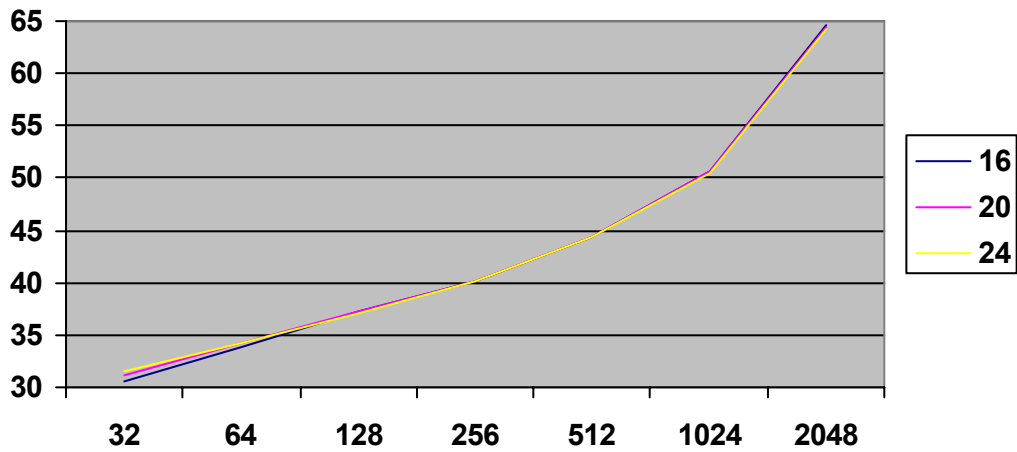
tempete.cif



QP/bitrate	32	64	128	256	512	1024	2048
16	29,73	29,73	29,73	29,73	30,05	37,48	43,40
20	27,82	27,82	27,82	27,82	32,27	37,75	43,38
24	26,22	26,22	26,22	27,38	33,08	37,76	43,32
26	25,53	25,53	25,53	28,39	33,18	37,75	43,26

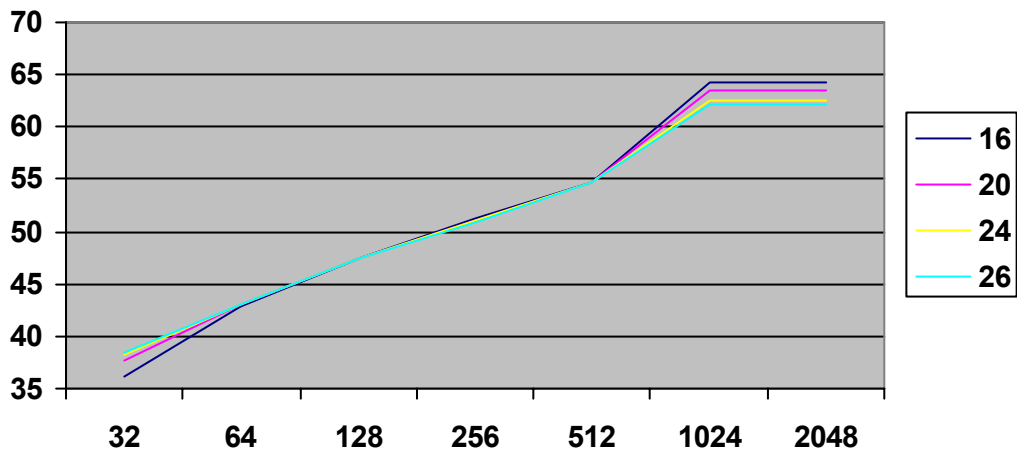
Seqüências QCIF (176 x 144px)

bridge-close.qcif



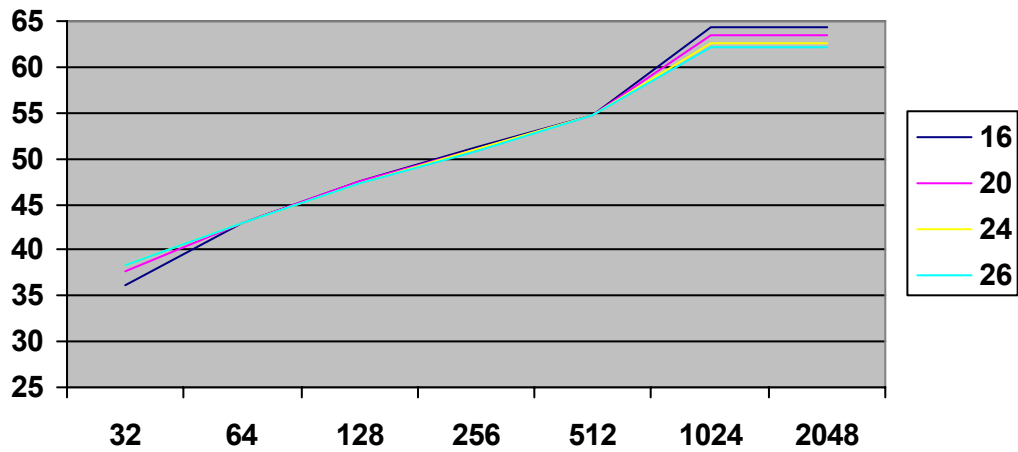
QP/bitrate	32	64	128	256	512	1024	2048
16	30,54	33,77	37,21	40,22	44,29	50,58	64,58
20	31,22	34,18	37,20	40,17	44,29	50,58	64,46
24	31,54	34,28	37,13	40,15	44,28	50,56	64,24

bridge-far.qcif



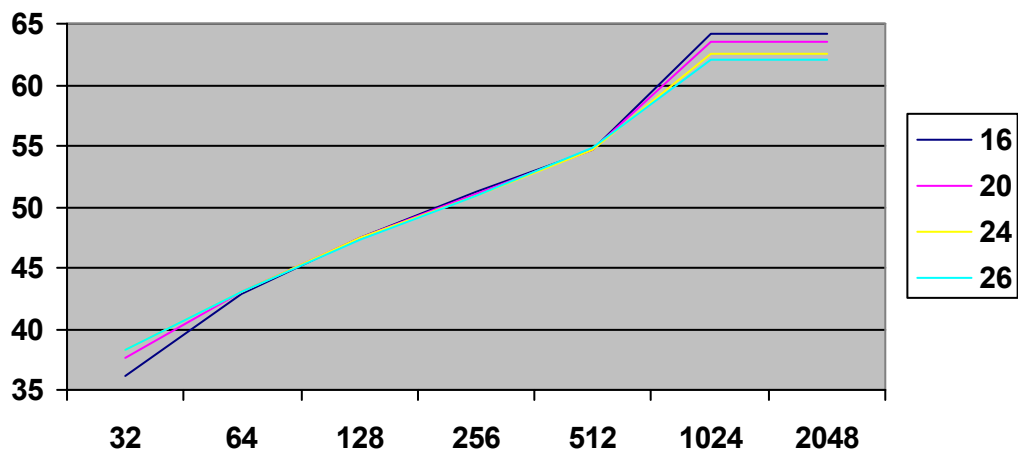
QP/bitrate	32	64	128	256	512	1024	2048
16	c	40,21	41,53	43,21	46,18	51,48	67,21
20	38,96	40,27	41,53	43,21	46,18	51,48	66,99
24	38,97	40,27	41,52	43,21	46,17	51,50	66,72
26	38,91	40,28	41,53	43,21	46,16	51,54	66,58

carphone.qcif



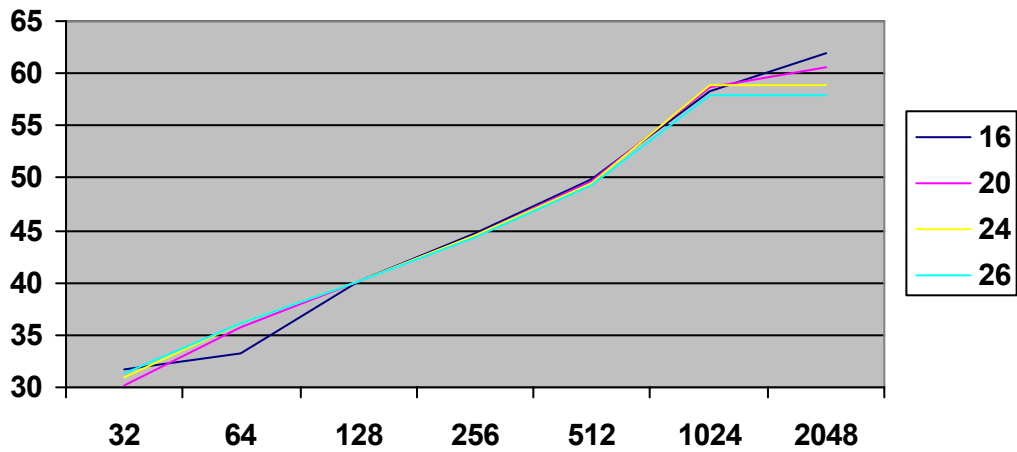
QP/bitrate	32	64	128	256	512	1024	2048
16	c	32,47	38,23	43,17	48,22	55,01	62,89
20	28,31	33,58	38,39	43,15	48,22	55,35	61,82
24	29,43	34,10	38,33	43,08	48,01	55,43	60,58
26	29,70	34,18	38,39	43,01	47,89	55,50	59,91

claire.qcif



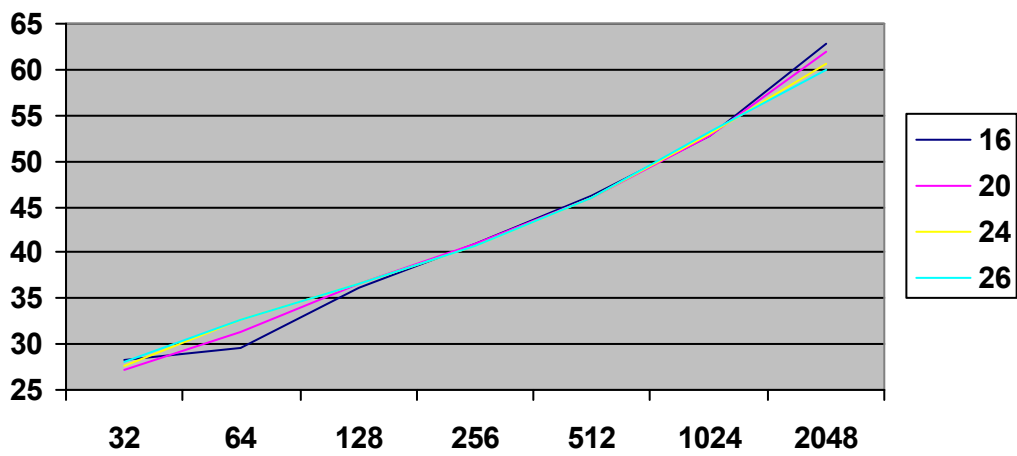
QP/bitrate	32	64	128	256	512	1024	2048
16	36,12	42,82	47,44	51,17	54,70	64,25	64,25
20	37,65	42,97	47,42	51,10	54,70	63,48	63,48
24	38,33	42,98	47,39	50,97	54,75	62,58	62,58
26	38,36	42,99	47,37	50,87	54,78	62,09	62,09

container.qcif



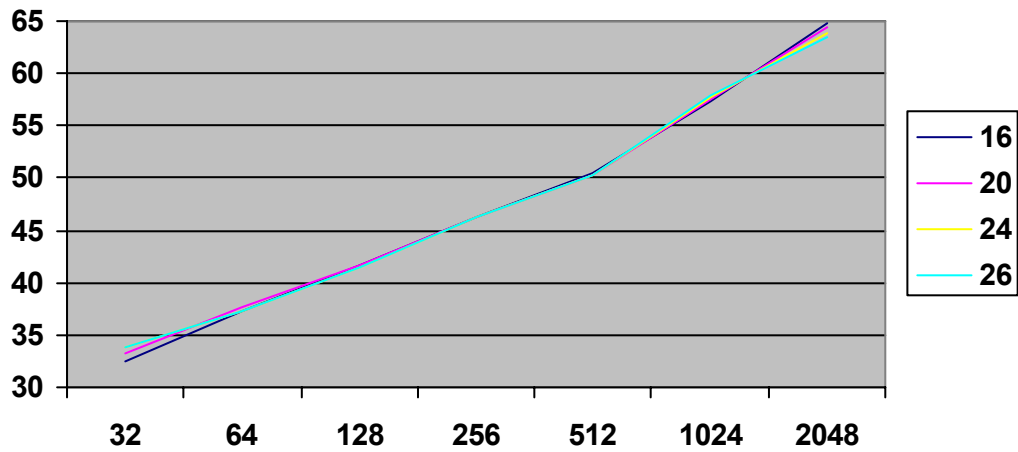
QP/bitrate	32	64	128	256	512	1024	2048
16	31,70	33,21	40,14	44,64	49,86	58,32	61,87
20	30,28	35,76	40,18	44,59	49,70	58,65	60,51
24	30,91	36,17	40,12	44,48	49,45	58,90	58,90
26	31,42	36,11	40,07	44,42	49,29	58,00	58,00

foreman.qcif



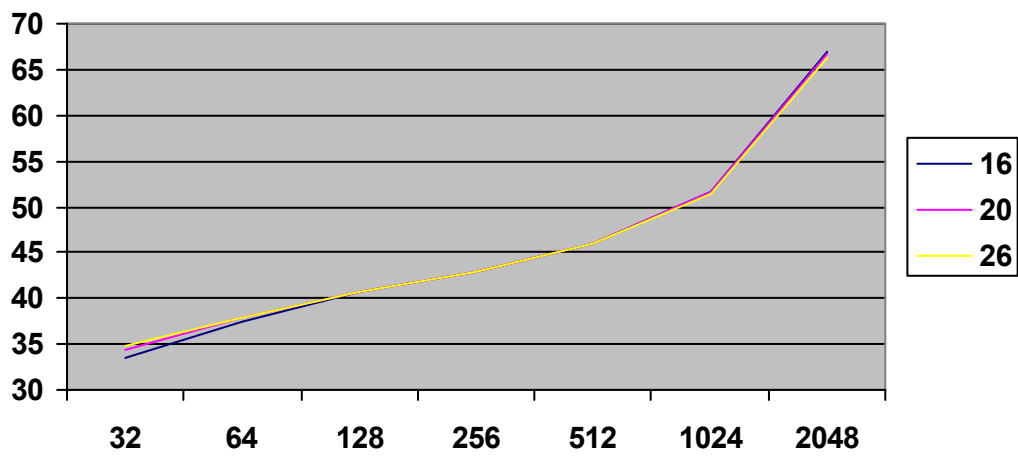
QP/bitrate	32	64	128	256	512	1024	2048
16	28,32	29,63	36,11	40,87	46,16	52,82	62,89
20	27,15	31,39	36,52	40,86	46,09	52,85	61,87
24	27,71	32,57	36,56	40,83	45,97	52,98	60,63
26	28,13	32,69	36,55	40,78	45,88	53,11	59,94

grandma.qcif



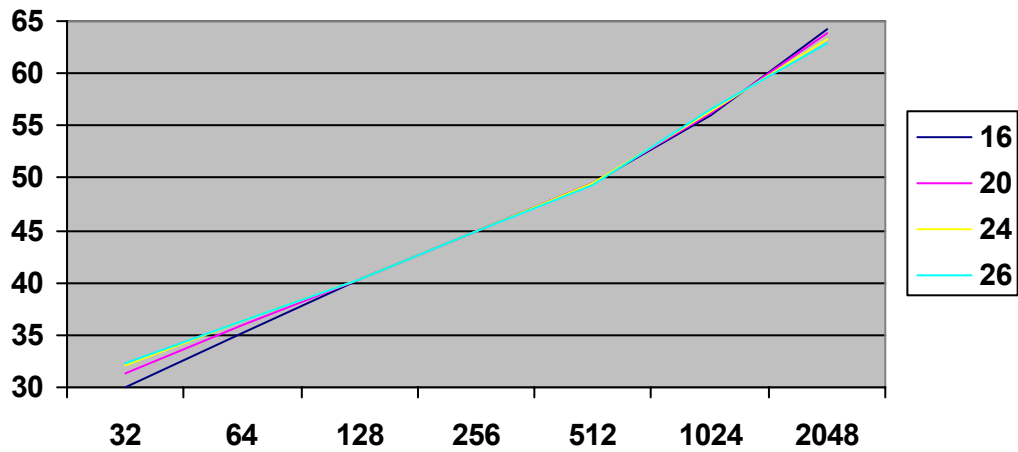
QP/bitrate	32	64	128	256	512	1024	2048
16	32,43	37,33	41,75	46,28	50,39	57,34	64,85
20	33,20	37,56	41,62	46,26	50,34	57,57	64,35
24	33,76	37,33	41,47	46,23	50,26	57,79	63,76
26	33,85	37,31	41,44	46,22	50,19	57,90	63,45

highway.qcif



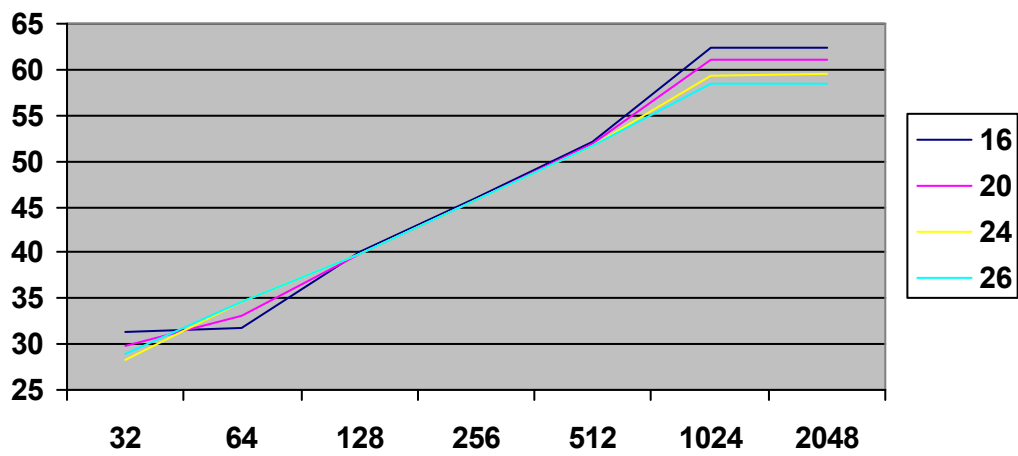
QP/bitrate	32	64	128	256	512	1024	2048
16	33,58	37,39	40,63	42,90	45,91	51,54	66,87
20	34,41	37,89	40,69	42,90	45,91	51,55	66,67
26	34,83	37,94	40,68	42,89	45,89	51,52	66,25

mthr_dotr.qcif



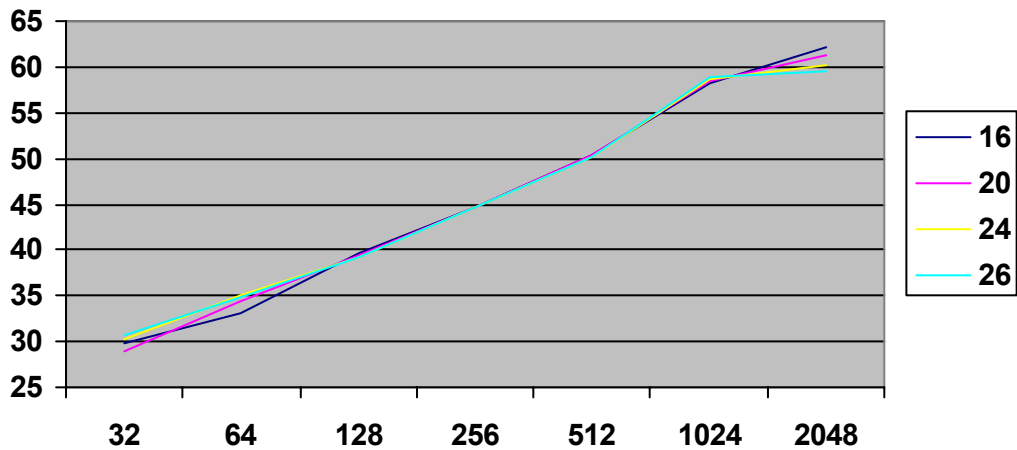
QP/bitrate	32	64	128	256	512	1024	2048
16	30,05	35,18	40,25	44,94	49,56	56,08	64,26
20	31,42	35,97	40,35	44,93	49,53	56,17	63,80
24	32,13	36,23	40,34	44,91	49,46	56,42	63,27
26	32,32	36,25	40,33	44,89	49,40	56,49	62,97

news.qcif



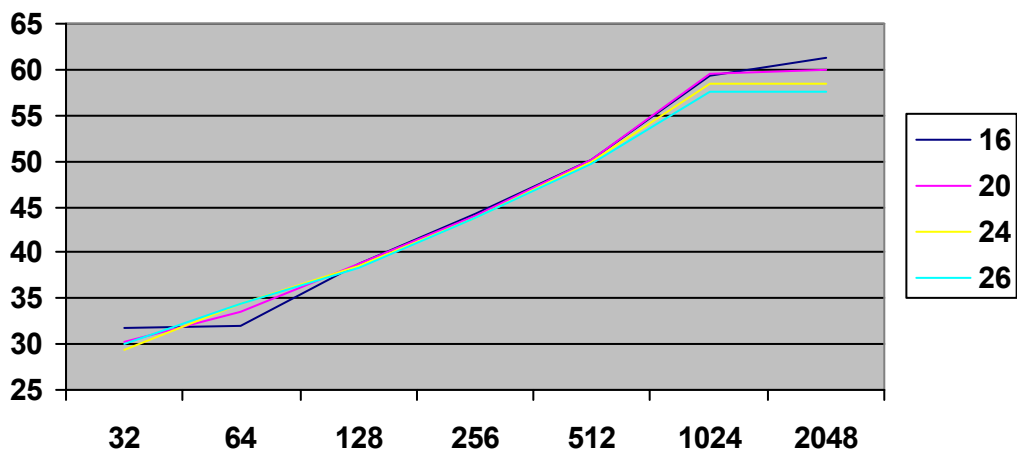
QP/bitrate	32	64	128	256	512	1024	2048
16	31,41	31,81	39,98	45,90	52,05	62,34	62,38
20	29,72	33,17	39,95	45,79	51,95	60,99	61,01
24	28,24	34,60	39,83	45,80	51,74	59,40	59,43
26	28,90	34,68	39,84	45,75	51,61	58,53	58,34

salesman.qcif



QP/bitrate	32	64	128	256	512	1024	2048
16	29,74	33,15	39,74	44,69	50,37	58,14	62,07
20	28,88	34,43	39,48	44,65	50,32	58,39	61,21
24	30,19	35,06	39,13	44,57	50,16	58,74	60,17
26	30,62	34,77	39,11	44,60	50,04	58,90	59,59

silent.qcif



QP/bitrate	32	64	128	256	512	1024	2048
16	31,79	31,97	38,87	44,17	50,23	59,25	61,38
20	30,18	33,44	38,82	43,93	50,10	59,60	60,06
24	29,28	34,43	38,50	43,88	49,83	58,43	58,50
26	30,01	34,36	38,29	43,88	49,72	57,59	57,63

Dos gráficos, conseguimos concluir que o parâmetro de quantização influencia pouco no resultado final (se variado conforme as instruções seguidas). A PSNR em codificações com taxa de bits de 2048 kbps não chega a divergir em 5dB (em alguns casos ainda é igual). Uma diferença como esta é imperceptível ao olho humano. Em alguns casos a PSNR com QP

mais baixo foi melhor, mas isso sugere uma recodificação/reavaliação da seqüência e do log de saída.

Também podemos notar que em algumas fachas de valores os resultados são iguais (como na seqüência mobile.cif, onde os resultados foram iguais com QP=16 e bitrate de 32 a 512kbps - 27,59dB). Aparentemente isso é uma restrição do próprio codificador. Dependendo dos parâmetros a taxa de bits não pode ser tão baixa ou tão alta quanto escolhemos. Não há indício de erros, mas podemos olhar no arquivo de informações da saída que a taxa de bits não foi a escolhida. Algumas vezes esta também é a razão de codificações com taxas mais baixas terem PSNR mais altas (na verdade o codificador não fez a codificação na taxa que escolhemos e então o resultado pode parecer estranho).

4.2. Outras codificações

Também foram feitas algumas codificações com o software SUPER. Este programa, gratuito, permite selecionarmos quase qualquer arquivo de vídeo como entrada e um formato de saída, escolhendo o codec de áudio e vídeo, além do container (chamamos este processo de transcodificação). O programa não permite uma comparação objetiva entre os arquivos de entrada e saída por não criar um arquivo log com as especificações do vídeo de saída, mas é possível fazer comparações subjetivas entre as seqüências originais e as codificadas.

Nestes casos foram utilizados como entrada vídeos já codificados com os codecs DivX e XviD (derivados do MPEG-4 parte 2, com grande poder de compressão e altamente difundidos pela internet, utilizados na maioria dos vídeos que circulam pela rede). Os vídeos são segmentos de aulas gravadas na PUC e o padrão H.264 foi adotado posteriormente por conseguir diminuir consideravelmente o tamanho dos arquivos (com alguma diminuição de qualidade) tornando viável a gravação destes arquivos em discos de DVD para distribuição.

	Entrada	Saída
Container utilizado:	AVI (13,4MB; 10,04%)	AVI (1,87MB; 2,66%)
Codificador de áudio:	Dolby AC3 192 kbps 93,1 MB (69,51%)	MP3 (MPEG-2 Layer 3) 96 kbps 13,7MB (19,44%)
Codificador de vídeo:	DivX 5 1218 kbps 27,4 MB (20,46%)	H.264 384 kbps 55,1 MB (77,90%)

Nesta configuração, um vídeo de 20:02 minutos, que antes tinha 137,178 KB, passou a ter 72,439 K, representando uma redução de 47,20% do seu tamanho (com alguma redução de qualidade). É importante ressaltar que o DivX é um codec recente, baseado no padrão MPEG-4 ASP (MPEG-4 parte 2).

4.3. Futuros testes

Pretende-se realizar exaustivos testes a fim de se obter os parâmetros ótimos para diferentes objetivos. Um filme, para ser projetado numa tela de cinema, por exemplo, deve ser codificado com parâmetros (como taxa de bits, level e profile) diferentes dos filmes exibidos na TV, que por sua vez, serão diferentes nas codificações dos filmes que serão exibidos no Brasil a partir do final de 2007, com a introdução do sistema digital brasileiro e transmissões em alta definição (HDTV). Outras possíveis aplicações são vídeos para dispositivos portáteis e celulares. Já é de interesse de algumas operadoras realizar streaming (exibição de vídeos ao vivo) nestes aparelhos.

Para poder encontrar os parâmetros que se encaixam no quadro da TV digital brasileira pretendo entrar em contato com grupos envolvidos no desenvolvimento de software e/ou

hardware para o SBTV, como o telemídia, grupo responsável pela criação de parte do middleware Ginga, que atua nos codificadores que serão acoplados às TVs comuns para que estas recebam o sinal digital.

Pretendo ainda realizar outros testes sugeridos em documentos de especialistas sobre o assunto com seqüências de diferentes formatos.

5. Conclusão

5.1. Dificuldades Encontradas

A principal dificuldade para utilizar o codificador de referência, única ferramenta encontrada capaz de fornecer os dados necessários para análise e completamente configurável (pode-se escolher parâmetro a parâmetro), é encontrar “vídeos crus” (sem nenhuma codificação), seqüências nos formatos *.yuv ou *.raw, principalmente em formatos grandes, semelhantes ao usado na TV e HDTV.

Outro problema é a demora na codificação. Com a introdução da versão 11 do codificador e uma otimização na escolha dos parâmetros adequados o tempo necessário para a codificação das seqüências CIF e QCIF diminuiu consideravelmente. Seqüências que antes demoravam de 10 a 15 horas para serem codificadas passaram a levar 1 ou 2 horas.

Para facilitar a rotina de testes, uma ferramenta com interface gráfica está sendo criada, integrando-se ao codificador e tornando este processo tão simples que qualquer pessoa seria capaz de fazer codificações com o software de referência.

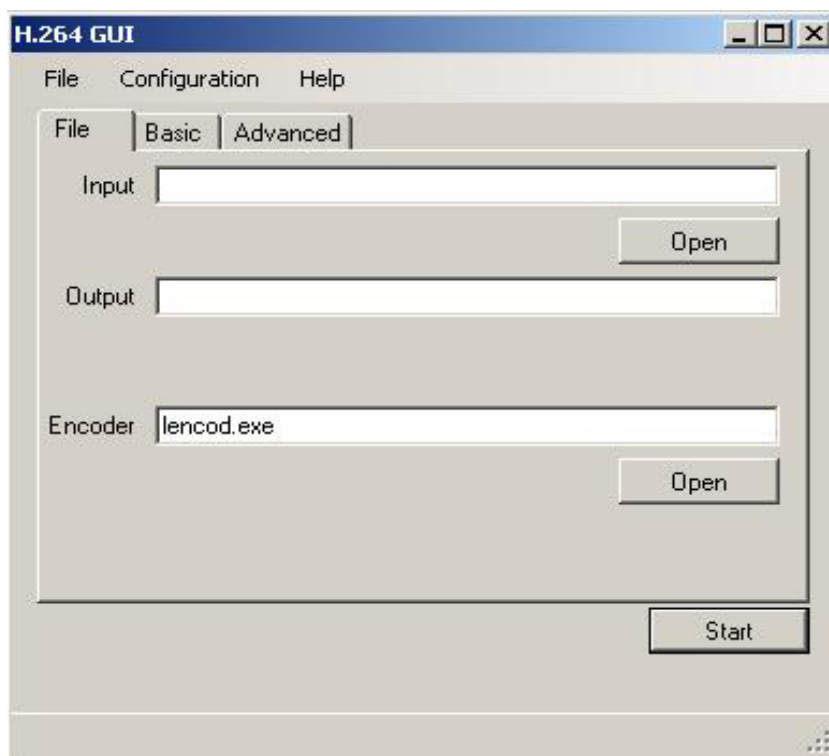


Figura 5.1: Interface gráfica para utilizar o codificador de referência. Facilidade para realizar testes.

5.2. Comentários finais

Todos os resultados encontrados foram de acordo com o esperado. O H.264 realmente provou ser o futuro da codificação de vídeo e não está mais tão longe da realidade. No final do ano o sistema brasileiro de TV digital entrará no ar em São Paulo, permitindo que usuários

comuns tenham acesso à esta nova tecnologia. Este novo sistema não adotou diretamente o H.264, este é uma opção, e não é possível afirmar se ele será usado imediatamente pois, como visto, requer muito recurso computacional, se comparado ao MPEG-2, solução adotada para vídeos em qualidade *standard* (a mesma da TV analógica). Em alguns anos todas as TVs brasileiras já sairão de fábrica preparadas para o sinal digital e o sinal analógico será extinto. Neste futuro próximo os equipamentos também serão mais poderosos, e não sentirão o peso do H.264. Neste mesmo futuro já existirão outros codificadores, bem mais complexos e poderosos que o H.264, capazes de reduzir ainda mais o tamanho dos vídeos, oferecendo mais recursos e qualidade.

6. Referências bibliográficas

- [1] ITU-R. **Studio encoding parameters of digital television for standard 4:3 and wide-screen 16:9 aspect ratios**. Rec. ITU-R-BT 601, outubro de 1995.
- [2] ITU-R. **Parameter values for the HDTV standards for production and international programme exchange**. Rec. ITU-R-BT 709, março de 2002.
- [3] ISO/IEC. **The MPEG-2 international standard**. Ref: ISO/IEC 13818-2, 1996.
- [4] ISO/IEC. **ITU-T Recommendation H.262 - ISO/IEC 13818-2 (MPEG-2): Generic coding of moving pictures and associated audio information**. Video ITU-T and ISO/IEC JTC1, November 1994.
- [5] Eduardo Villares. Tit, trabalho de fim de curso, DEE, PUC-Rio, dezembro 2005.
- [6] www.wikipedia.org
- [7] PURY, Atul; CHEN, Xuemin; LUTHRA, Ajay. **Video Coding Using the H.264/MPEG-4 AVC Compression Standard**. Net. Disponível em: www.sciencedirect.com. Acesso em: julho, 2006.
- [8] ALCAIM, Abraham. Processamento de voz e imagem. Rio de Janeiro, 1999.
- [9] SCHÄFER, Ralf; WIEGAND, Thomas; SCHWARZ, Heiko. **The Emerging H.264/AVC Standard**. 2003.
- [10] SULLIVAN, Gary. J.; TOPIWALA, Pankaj; LUTHRA, Ajay. **The H.264/AVC Advanced Video Coding Standard: Overview and Introduction to the Fidelity Range Extensions**. 2004.
- [11] T. Wiegand, H. Schwarz, A. Joch, and F. Kossentini. **Rate-constrained coder control and comparison of video coding standards**. IEEE Transactions on Circuits and Systems for Video Technology, vol. 13, July 2003.
- [12] KAMACI, Nejat; ALTUNBASAK, Yucel. **Performance comparison of the emerging H.264 video coding standards with the existing standards**. Atlanta, GA, USA: Georgia Institute of Technology.
- [13] T. Wedi and Y. Kashiwagi. **Subjective quality evaluation of H.264/AVC FRExt for HD movie content**. Joint Video Team document JVT-L033, July, 2004.
- [14] OLIVEIRA, Geraldo Cesar de; ALCAIM, Abraham. **Performance of Motion Estimators Over MPEG Coder Considering Frame Delay Analysis**. Rio de Janeiro: CETUC/PUC-Rio.
- [16] RICHARDSON, Iain E. G. **H.264 / MPEG-4 Part 10: Variable Length Coding**. Net. Disponível em: www.vcodex.com. Acesso em: julho, 2006.