Arquivos com Formato .BMP

É um formato onde a maior parte das informações descritas no cabeçalho (tamanho 14 bits) está relacionada a paleta de cores RGB, resolução e a informações de largura/comprimento da imagem.

```
typedef struct tagBITMAPINFOHEADER
{
DWORD biSize;
LONG biWidth;
LONG biHeigth;
WORD biPlanes;
WORD biBicount;
DWORD biCompression;
DWORD biSizeImage;
LONG biXPelsPerMeter;
LONG biYPelsPerMeter;
DWORD biClrUsed;
DWORD biClrImportant;
}  BITMAPPINFOHEADER;
```

THE COLOR TABLE

Os Arquivos .BMP, no windows, vem com cinco formatos: 2 cores (1 bit por pixel), 16 cores (4 bits por pixel), 256 cores (8 bits por pixel), 65.000 (16 bits por pixel) cores e 16 milhões de cores (24 bits por pixel). O número de bits por pixel, pode ser determinado no cabeçalho pelas sintaxes: biBitCount ou bcBitCount.

QUALIDADE

Em geral, a qualidade destas imagens está relacionada a resolução das mesmas; ou seja a quantidade de pontos na vertical/horizontal.

Arquivos com Formato .JPG

1 O que é JPEG?

Desde junho de 1982, WG8 (Grupo de trabalho 8) da ISO (Organização Internacional de Padronização) tem trabalhado na padronização de compressão e descompressão de imagens [11Y588]. Em junho de 1987, dez técnicas diferentes para imagens coloridas de tons de cinza foram apresentadas. Estas técnicas foram comparadas, e foram analisados três delas mais adiante. Uma transformação adaptável da técnica de codificação baseada no DCT alcançou o melhor resultado e, então, foi adotado para JPEG [LMY88, WVP88]. JPEG é um projeto em comum de ISO/IECJTC1/5C2/WG1O e a comissão Q.16 de CCITT SGVIII. Em 1992, JPEG se tornou um Padrão ISO Internacional (IS) [0rg93].

JPEG é aplicado para imagens coloridas e de tons de cinza [L0W91, MP91, Wal9l]. Uma codificação e decodificação rápida de imagens também é usada para seqüências de vídeo conhecida como *Motion JPEG.* Hoje, partes do JPEG já estão disponíveis como pacotes de software.

JPEG cumpre as seguintes exigências para garantir futuras distribuições e aplicações [Wal9l]:

- A implementação do JPEG deve ser independente do tamanho da imagem.
- A implementação do JPEG deve ser aplicável a qualquer imagem.
- A representação de cor deve ser independente de implementação especial.
- O conteúdo da imagem pode ser de qualquer complexidade, com qualquer característica estatística.
- A especificação padrão de JPEG deve ser estado-de-arte (ou próximo) relativo ao fator de compressão e qualidade de imagem alcançada.
- A complexidade do processo têm que permitir uma solução de software para rodar em tantos processadores padrões disponíveis quanto possível. Adicionalmente, o uso de hardware especializado deve aumentar qualidade de imagem substancialmente.
- Decodificação seqüencial (linha-por-linha) e decodificação progressiva (refinamento da imagem inteira) deve ser possível. Um lossless, codificação

hierárquica da mesma imagem com resoluções diferentes semelhante a imagens de Photo-CD deve ser apoiado.

O usuário pode selecionar a qualidade da imagem reproduzida, o tempo de processamento para a compressão e o tamanho da imagem comprimida escolhendo parâmetros individuais apropriados.

Aplicações não precisam incluir um codificador e um decodificador ao mesmo tempo. Em muitas aplicações só é preciso um único deles. O fluxo de dados codificado tem um formato de intercâmbio fixo que inclui dados de imagem codificados, como também os parâmetros escolhidos e tabelas do processo de codificação. Se o processo de compressão e descompressão concordam em um conjunto comum de tabelas de codificação a ser usado, por exemplo, os dados das tabelas respectivas deles não precisam ser incluídos no fluxo de dados. Existe aí um contexto comum entre codificar e decodificar. O formato de intercâmbio pode ter um formato abreviado que não garante a inclusão das tabelas necessárias. O formato de intercâmbio em modo regular (i.e., o formato não abreviado) inclui toda a informação necessária para decodificar sem qualquer conhecimento prévio do processo de codificação.

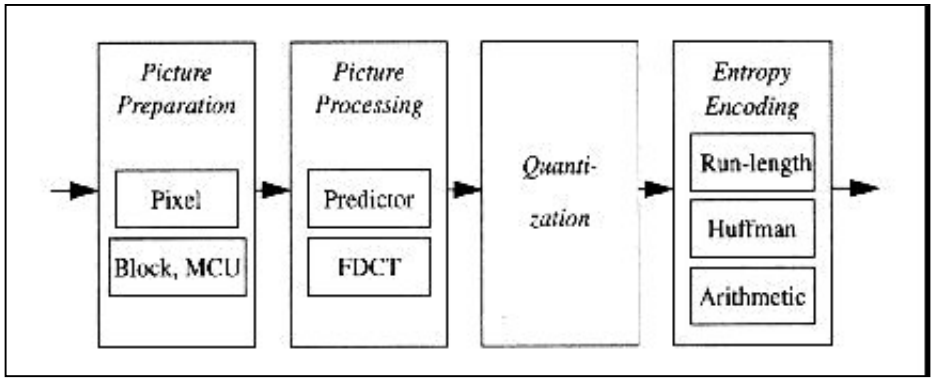


Figura 1: Passos do processo de compressão JPEG.

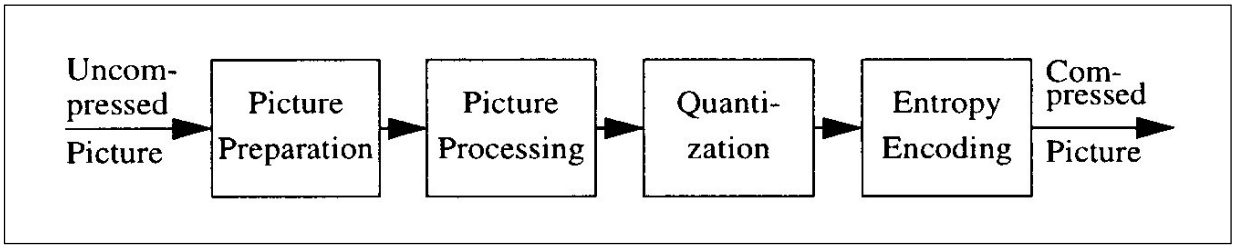


Figura 1.1

A *figura 1* esboça os passos de compressão de JPEG conforme o esquema global mostrado na *Figura 1.1* Quatro variantes diferentes de compressão de imagem podem ser determinadas para guiar os quatro modos. Cada modo inclui combinações adicionais:

- A perda seqüencial do modo baseado em DCT (processo de baseline) deve ser apoiada por toda implementação de JPEG.

- A perda em expansão do modo baseado em DCT provê um conjunto de encarecimentos adicionais para o processo de baseline.

- O modo lossless (menos perda) tem uma baixa taxa de compressão que permite reconstrução perfeita da imagem original.

- Modo hierárquico acomoda imagens de resoluções diferentes e seleciona seus algoritmos dos três modos definidos acima.

O processo baseline possui as seguintes técnicas: Block, MCU, FD CT, Run-lenght e Huffman que são explicados com os outros modos em mais detalhe nesta seção. Na próxima seção, é apresentada preparação de imagem para todos os modos; os passos restantes de processamento de imagem, quantização e codificação entropica.

2 Preparação de imagem

Para o primeiro passo da preparação de imagem, JPEG especifica um modelo de imagem geral. Com este modelo é possível descrever a maioria das representações de imagem de duas dimensões. Por exemplo, o modelo não está baseado em três componentes de imagem com o código YUV de 9-bits e um número fixo de linhas e colunas. Ainda não foi traçado um mapa dos valores de codificação da crominancia. Isto cumpre a demanda de independência dos parâmetros de imagem, como tamanho da imagem, imagem e aspecto dos pixels.

Uma imagem fonte consiste em pelo menos um e no máximo 255 componentes ou planos, como mostrado no lado esquerdo da figura 2. Cada componente C_i pode ter um número diferente de pixels nas componentes horizontal (X_i) e vertical (Y_i). Note, que o índice denota o número da componente ou do plano. Estas componentes podem ser representadas pelos três sinais de cores RGB, YIQ ou YUV, por exemplo.

A Figura 3 mostra as três componentes de uma imagem, cada uma com a mesma resolução, e cada uma tem uma ordem retangular C_i de X_i x Y_i pixels. Os três valores X_i e o três valores de Yi são iguais.

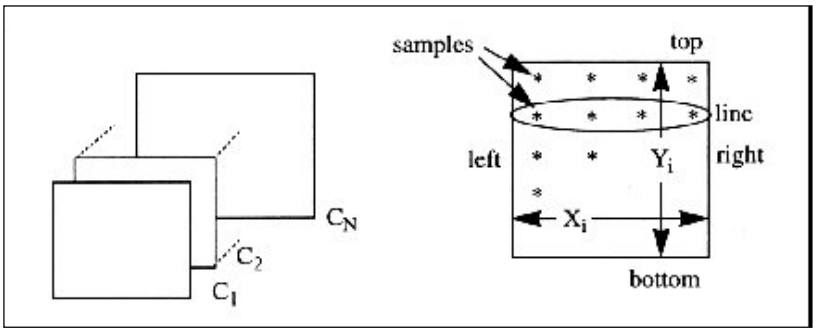


Figura 2: Imagem digital descomprimida com a definição dos respetivos componentes da imagem de acordo com o padrão JPEG.

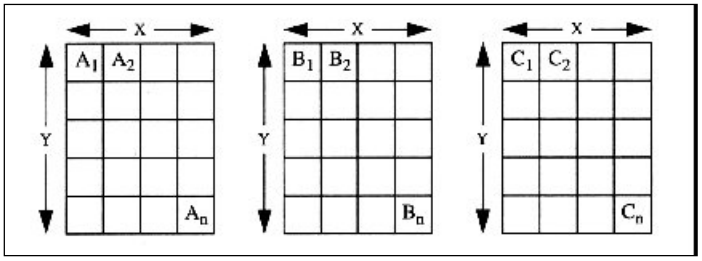


Figura 3: Exemplo de preparação de imagem JPEG com três componentes que têm a mesma resolução.

A resolução das componentes individuais podem ser diferentes. *Figura 4* mostra uma imagem com a metade do número de colunas (i.e., metade do número de amostras horizontais) no segundo e terceiro planos comparados com o primeiro plano: Y1= Y2 = Y3, e X1 = 2X2 = 2X3.

Uma imagem de tons de cinza será, na maioria dos casos, consistida de um único componente. Uma representação de cor RGB tem três componentes com resolução igual (i.e., o mesmo número de linhas Y1 = Y2 = Y3, e mesmo número de colunas X1= X2 = X3). Para JPEG, o processo de coloração de imagem YUV usa-se Y1 = 4Y2 = 4Y3 e X1 =4X2 =4X3.

Cada pixel é representado através de p bits com valores que vão de 0 a 2^p - 1. Todos os pixels de todos os componentes dentro da mesma imagem é codificado com o mesmo número de bits. Os modos lossy (mais perdas) de JPEG usam uma precisão de ou 8 ou 12 bits por pixel. Modos de Lossless (menos perdas) usam uma precisão de 2 até 12 bits por pixel. Se uma aplicação de JPEG faz uso de qualquer outro número de bits, a própria aplicação tem que executar uma transformação de imagem satisfatória para o número definido de bits no padrão de JPEG.

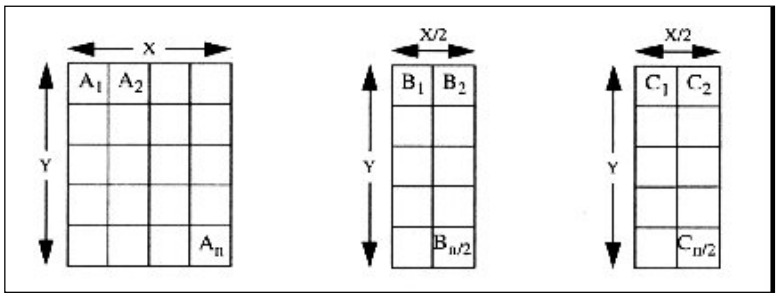


Figura 4: *Exemplo de preparação da imagem JPEG com três componentes que têm resolução diferente.*

As dimensões da imagem comprimida são definidas por um novo valor de X (o máximo de todo o X_i), Y (o máximo de todo o Y_i), H_i e V_i. H_i e V_i são as proporções de amostra relativa a horizontal e vertical especificada para cada componente *i.* H_i e V_i devem ser valores inteiros entre 1 e 4. Esta definição é necessária para o entrelaçamento de componentes que será descrito mais tarde.

Considerando o exemplo seguinte, também mostrado em [0rg93]. Uma figura é determinada com a máxima resolução horizontal e vertical de 512 pixels e os seguintes fatores de amostra:

Nível 0: H_0 =4, V_0 = 1

Nível 1: H_1 = 2, V_1 = 2

Nível 2: H_2 = 1, V_2 = 1

Assumindo X = 512, Y = 512, H**max =** 4 e V**max =** 2 isto conduz para

Nível 0: X_0 =512, Y_0 **=** 256

Nível 1: X_1 = 256, Y_1 = 512

Nível 2: X_2 = 128, Y_2 = 256

Com funções de teto dadas, X_i e Y_i são calculados como segue: para o uso de compressão, a imagem é dividida em unidades de dados. O modo de lossless usa um pixel como uma unidade de dados. O modo de lossy usa blocos de 8 x 8 pixels. Esta definição de unidades de dados é um resultado de DCT que sempre transforma blocos conectados. Na maioria dos casos, as unidades de dados são

processadas componente por componente, como mostrado na figura 1, para os passos de processamento da imagem. Como mostrado na *figura 5*, para um componente, a ordem de processo de unidades de dados é da esquerda para a direita e do topo para baixo, um componente depois do outro. Isto é conhecido como *ordenação de dados não entrelaçada.* Usando o modo não entrelaçado para uma codificação RGB de imagem com resolução muito alta, o display apresentaria só o componente vermelho inicialmente, então, depois, o azul e o verde seriam desenhados resultando na cores da imagem original sendo reconstruída. Adequado para a velocidade de processamento finita do decodificador de JPEG, é então freqüentemente mais razoável unidades de dados entrelaçadas como mostrado na *figura 6*.

Figura 5: *Ordem de unidades de dados não entrelaçados, outro processo de componente de acordo com o padrão JPEG*

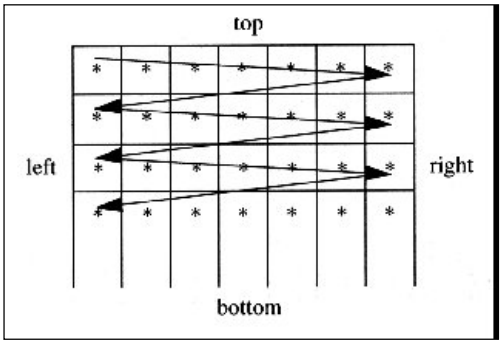
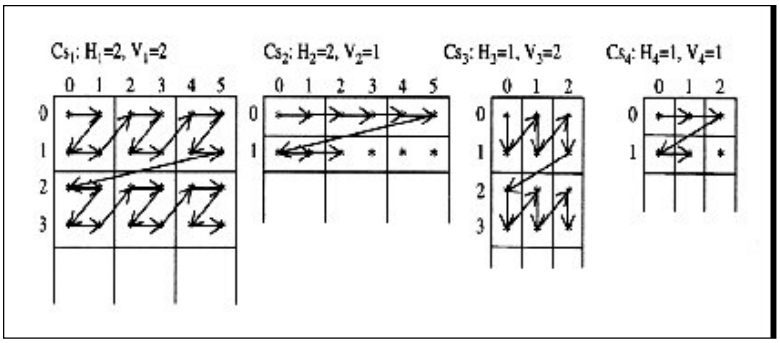


figura 6: *um exemplo com quatro componentes como derivou do padrão de JPEG.*



Unidades da dados entrelaçadas de diferentes componentes são combinadas nas Unidades Mínimas de Código (MCUs). Se todos componentes tem a mesma resolução (X_i x Y_i), um MCU consiste em exatamente uma unidade de dados para cada componente. O decodificador mostra a imagem MCU por MCU. Isto permite apresentação de cor correta, até mesmo em imagens parcialmente decodificadas. No caso de resoluções diferentes para componentes simples, a construção dos MCUs se torna mais complexa (veja *figura 6*). Para cada componente, regiões das

unidades de dados (se necessário, com diferentes números de unidades de dados) são determinadas. Cada componente consiste do mesmo número de regiões. Por exemplo, a *figura 6* mostra seis regiões de cada componente. Um MCU consiste de exatamente uma região em cada componente. Novamente, as unidades de dados com uma região são ordenadas da esquerda para a direita e do topo para baixo.

A *Figura 6* mostra um exemplo com quatro componentes. Os valores de H_i e V_i são providos para cada componente na figura. O primeiro componente tem a resolução mais alta em ambas dimensões e o quarto componente tem a mais baixa resolução. As setas indicam a direção de amostra das unidades de dados para cada componente. O MCUs são construídos na seguinte ordem:

$MCU1 = d^1 00\ d^1 01\ d^1 10\ d^1 11\ d^2 00\ d^2 01\ d^3 00\ d^3 10\ d^4 00$

$MCU2 = d^1 02\ d^1 03\ d^1 12\ d^1 13\ d^2 02\ d^2 03\ d^3 01\ d^3 11\ d^4 01$

$MCU3 = d^1 04\ d^1 05\ d^1 14\ d^1 15\ d^2 04\ d^2 05\ d^3 02\ d^3 12\ d^4 02$

$MCU4 = d^1 20\ d^1 21\ d^1 30\ d^1 31\ d^2 10\ d^2 11\ d^3 20\ d^3 30\ d^4 10$

As unidades de dados do primeiro componente são $C_{s1}: d^1 00. . .d^1 31$

As unidades de dados do segundo componente são $C_{s2}: d^2 00. ..d^2 11$

As unidades de dados do terceiro componente são $C_{s3} :d^3 00.. .d^3 30$

As unidades de dados do quarto componente são $C_{s4}: d^4 00. . .d^4 10$

Até quatro componentes podem ser codificados usando o modo de entrelaçamento de acordo com o padrão de JPEG. Cada MCU consiste em no máximo dez unidades de dados. Dentro de uma imagem, alguns componentes podem ser codificados no modo entrelaçado e outros no não entrelaçado.

3 Modo de perda seqüencial baseado no DCT

3.1 Processamento da imagem

Depois de preparação da imagem, as amostras de imagem descomprimidas se agrupam em unidades de dados de 8 x 8 pixels e são passadas ao encoder; a ordem destas unidades de dados é definida pelo MCUs. Neste modo de baseline, são codificadas amostras simples usando *p* = 8 bits. Cada pixel é um inteiro entre 0 e 255.

O primeiro passo no processamento de imagem no modo baseline (baseline process), como mostrado na *Figura 7*, é uma transformação executada por DCT [ANR74, NP78]. Os valores dos pixels são trocados no alcance [-128,127], com zero como o centro. Estas unidades de dados de 8 x 8 valores de pixel trocados são definidos por S_{yx}, onde x e y estão entre zero e sete. Cada um destes valores são transformados usando forward DCT (FDCT): esta transformação deve ser feita 64 vezes por unidade de dados. O resultado são 64 coeficientes de ***Suv.***

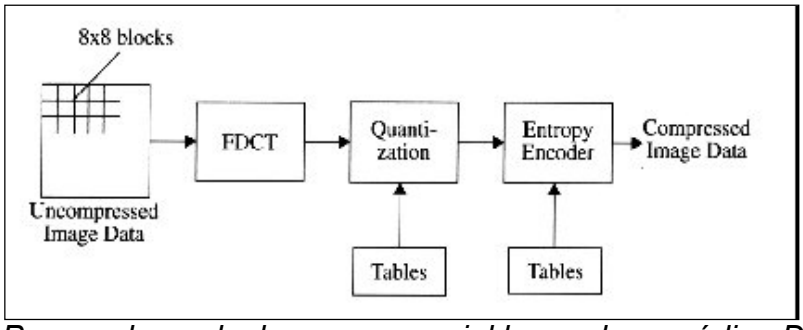


Figura 7*: Passos do modo lossy sequencial baseado no código DCT.*

$$S_{vu} = \frac{1}{4} c_u c_v \sum_{x=0}^{7} \sum_{y=0}^{7} S_{yx} \cos \frac{(2x+1)u\pi}{16} \cos \frac{(2y+1)v\pi}{16}$$

$$\text{where } c_u,\ c_v = \frac{1}{\sqrt{2}} \text{ for u, v} = 0\text{; otherwise } c_u,\ c_v = 1$$

fórmula 1

DCT é semelhante a *Transformação de Fourier Discreta* (DFT); traça os valores do tempo para o domínio da freqüência. Então, cada coeficiente pode ser considerado como uma freqüência bi-dimensional.

O coeficiente **S_{00}** corresponde à mais baixa freqüência em ambas as dimensões. É conhecido como o coeficiente DC, que determina a cor fundamental da unidade de dados de 64 pixels. O coeficiente DC e o coeficiente DCT tem freqüência zero em ambas as dimensões. Os outros coeficientes são chamados coeficientes AC. Coeficientes AC são coeficientes DCT para os quais a freqüência em uma ou ambas dimensões não é zero. Por exemplo, **S70** representa a freqüência mais alta na direção horizontal, e é possível a separação mais íntima de linhas verticais em 8x 8 unidade de dados. **S07** representa a freqüência mais alta na dimensão vertical, i.e., a separação mais íntima de linhas horizontais. **S77** indica a freqüência mais alta que aparece igualmente em ambas as dimensões. O valor absoluto de **S77** é maior se a fonte 8 x 8 unidade de dados consiste em uma matriz cheia, com quantos componentes 1 x 1 for possível. Uma ou ambas dimensões são não zero. Por exemplo, **S44** será maior se o bloco consistir em 16 quadrados de 4 x 4 pixels. Dando uma olhada na *fórmula 1* acima de FDCT, nós reconhecemos que as expressões de co-seno só dependem de x e *u, y* e *v* respectivamente, mas não depende de *Syx*. Então, estas expressões de co-seno representam constantes que não têm que ser calculado inúmeras vezes. Há muitas técnicas efetivas e implementação de DCT. Contribuições importantes podem ser achadas em [DG9O, Fei9O, 11ou88, Lee84, LF91, S1186, VN84, Vet85].

Para reconstrução da imagem, o decodificador usa o *DCT Inverso* (IDCT). Os coeficientes que ***SVU*** deve ser usado para o cálculo:

$$S_{yx} = \frac{1}{4} \sum_{v=0}^{7} \sum_{u=0}^{7} c_u c_v S_{vu} \cos \frac{(2x+1)u\pi}{16} \cos \frac{(2y+1)v\pi}{16}$$

$$\text{where } c_u,\ c_v = \frac{1}{\sqrt{2}} \text{ for u, v} = 0; \text{ otherwise } c_u,\ c_v = 1$$

Se o FDCT, como também o IDCT, pudesse ser calculado com total precisão seria possível reproduzir exatamente os 64 pixels da fonte. De um ponto de vista teórico, DCT seria lossless neste caso. Em prática, precisão é restringida e DCT é lossy; porém, o padrão de JPEG não define precisão. Por isto, duas implementações diferentes de um decodificador de JPEG poderiam gerar imagens diferentes como produção da mesma imagem comprimida. JPEG define a máxima tolerância somente.

A maioria das áreas de uma imagem típica consiste em regiões grandes de uma única cor que, depois de aplicar DCT, é representado por muitos coeficientes com valores muito baixos. Porém, as extremidades são transformadas em coeficientes que representam freqüências altas. Imagens de complexidade média

consistem em muitos coeficientes AC com um valor próximo a zero. Então, codificação entropica é usada para alcançar redução de dados considerável.

3.2 Quantização

Seguindo os passos da *Figura 1*, a quantização de todos os coeficientes de DCT é executada. Esta é uma transformação de lossy. Para este passo, a aplicação de JPEG proporciona uma tabela com 64 entradas. Cada entrada será usada para a quantização de um dos 64 coeficientes DCT. Assim, cada um dos 64 coeficientes pode ser ajustado separadamente. A aplicação tem a possibilidade de afetar o significado relativo dos diferentes coeficientes e freqüências específicas podem ser tratadas com mais importância que outras. Estes coeficientes devem ser determinados de acordo com as características da imagem fonte. A possível compressão é influenciada às custas da qualidade de imagem realizável.

Cada entrada da tabela é um valor inteiro de 8 bits chamado $Q_{uv.}$ O processo de quantização fica menos preciso com o aumento de tamanho das entradas da tabela. Quantizatição e desquantização tem que usar as mesmas tabelas. Não há especificação padrão para quantização em JPEG; aplicações podem especificar valores que customizam a qualidade da figura desejada de acordo com as características de imagem particulares.

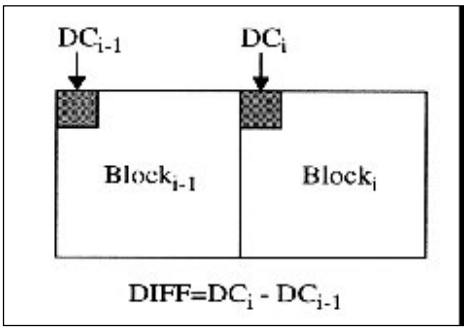


Figura 8: *Preparação de Coeficiente DCs do DCT para codificação entropicar, inclusive o cálculo da diferença entre valores vizinhos.*

3.3 Codificação entropica

Durante o primeiro passo da codificação entropica, os coeficientes DC quantizados são tratados separadamente dos coeficientes AC quantizados. A ordem do processo é especificada por uma *seqüência de zig-zag* como mostrado na *Figura 9*.

- Os coeficientes DC determinam a cor básica das unidades de dados. Entre unidades de dados adjacentes a variação de cor é bastante pequena. Então, um Coeficiente DC é codificado como a diferença entre o Coeficiente DC atual e o prévio. Só as diferenças são processadas subseqüentemente (*veja Figura 8*).

- O DCT que processa ordem dos Coeficientes AC que usam a seqüência de zig-zag ilustra que coeficientes com mais baixas freqüências (tipicamente com valores mais altos) são codificados primeiro, seguido pelas freqüências mais altas (com valores tipicamente pequenos, perto de zero). O resultado é uma sucessão estendida de bytes de dados semelhantes e permite codificação entropica muito eficiente. Nota que a seta entre o Coeficiente DC e o primeiro Coeficiente AC denota que este valor de DC tem a mais baixa freqüência.

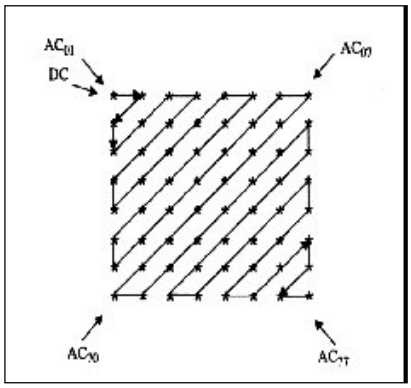


Figura 9: *Preparação dos coeficientes AC do DCT para codificação entropica, na ordem crescente de frequencia.*

JPEG especifica Huffman e codificação aritmética como métodos de codificação entropica. Para o lossy seqüencial baseado no modo DCT, discutido nesta seção, só a codificação de Huffman é permitida. Em ambos os métodos, uma codificação dos valores zero dos Coeficientes AC quantizados é aplicada primeiro. Adicionalmente, Coeficientes Acs não zero, como também os Coeficiente DCs, são transformados em uma representação espectral para comprimir os dados. O número de bits exigidos depende do valor do coeficiente. Um Coeficiente AC não zero será representado usando de 1 a 10 bits. Para a representação de Coeficiente DCs, uma resolução mais alta de 1 bit para um máximo de 11 bits é usada. O resultado é uma representação de acordo com o *ISO* formato que especifica as seguintes informações:

- O número de coeficientes subseqüentes com o zero de valor.

- O número de bits usados para a representação do coeficiente seguinte.

- O valor do coeficiente representado usando o número especificado de bits.

A vantagem principal do Huffman sobre a codificação aritmética é a implementação livre, poia não é protegido por uma patente como a aritmética.

Desvantajoso é o fato que a aplicação tem que prover a tabela de códigos pois JPEG não os predefine. Este modo de baseline permite o uso de diferentes tabelas de Huffman para Coeficientes DC e AC.

No caso de *codificação seqüencial, é* codificada a imagem inteira e é decodificada em uma única passagem. *Figura 10* mostra um exemplo de decodificação com apresentação imediata; o quadro é apresentado de cima para baixo.

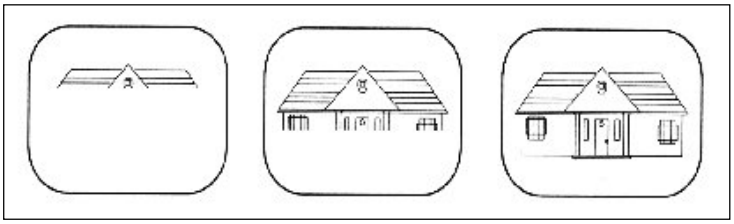


Figura 10

4 Expanção Lossy baseado no modo DCT

Pré-processamento de imagem neste modo difere do modo previamente descrito em termos do número de bits por amostra. Especificamente, uma precisão de amostra de 12 bits por amostra, mas também pode ser usado 8 bits por amostra.

Para o lossy expandido no modo baseado em DCT, JPEG especifica *codificação progressiva* além de codificação seqüencial. Na primeira passagem, aparece uma representação muito áspera da imagem que parece fora de foco e é refinado durante passos sucessivos. Um exemplo esquemático é mostrado na *Figura 11*.

A representação de imagem progressiva é alcançada por uma expansão de quantização. Isto também é conhecido como *codificação de layered.* Para esta expansão, um pára-choque é adicionado à produção do quantizador que temporariamente armazena todos os coeficientes do DCT quantizado. Progressiveness é alcançado de dois modos diferentes:

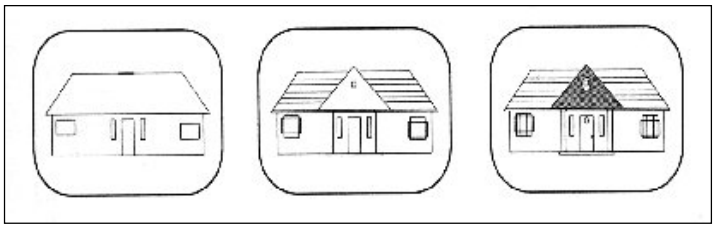


Figura 11.

- Usando uma *seleção espectral,* na primeira passagem, são passados os coeficientes DCT da quantização de freqüências baixas de cada unidade de dados para a codificação entropica. Em corridas seguintes, são processados os coeficientes de freqüências mais altas.

- *Aproximação sucessiva* transfere tudo dos coeficientes quantizados em cada passagem, mas bits simples são diferenciados de acordo com a sua significancia. Os bits muito significantes são codificados primeiro.

Além da codificação de Huffman, pode ser usado codificação aritmética neste modo. A codificação aritmética não requer nenhuma tabela para a aplicação já que é adaptado automaticamente às características estatísticas de uma imagem. De acordo com várias publicações, a compressão alcançada na codificação é de 5% a 10% melhor que a alcançado alcançada por Huffman. Outros autores assumem uma taxa de compressão semelhante. Codificação Aritmética é ligeiramente mais complexo e sua proteção através de patentes deve ser considerada.

Quatro tabelas de codificação para a transformação de DC - e Coeficientes AC podem ser definidos pela aplicação de JPEG. Em um modo mais simples, faz-se uma escolha de duas tabelas de Huffman uma para coeficientes DC e outra para coeficientes AC. Por isto, podem ser usados doze tipos alternativos de processo neste modo (*veja tabela 1*). O modo de exibição mais extensamente usado é o modo de exibição seqüencial com 8 bits por amostra e codificação de Huffman.

Image Display	Bits per Sample	Entropy Coding
sequential	8	Huffman Coding
sequential	8	Arithmetic Coding
sequential	12	Huffman Coding
sequential	12	Arithmetic Coding
progressive successive	8	Huffman Coding
progressive spectral	8	Huffman Coding
progressive successive	8	Arithmetic Coding
progressive spectral	8	Arithmetic Coding
progressive successive	12	Huffman Coding
progressive spectral	12	Huffman Coding
progressive successive	12	Arithmetic Coding
progressive spectral	12	Arithmetic Coding

Tabela 1.

5 Modo Lossless

O modo lossless mostrado na *Figura 12* usa unidades de dados de pixels únicos para a preparação da imagem. Qualquer precisão entre 2 e 16 bits por pixel pode ser usada.

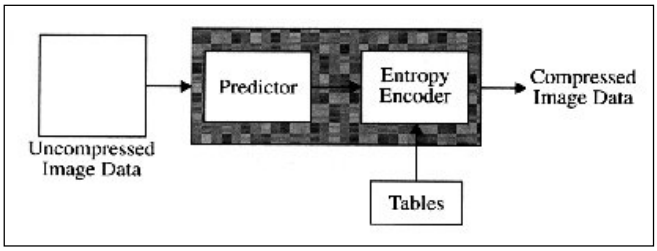


Figura 12: *Modo de Lossless baseado em uma predição.*

Neste modo, o processamento de imagem e quantização usam uma técnica de predictive em vez de uma técnica de transformação. Como mostrado na *Figura 13*, para cada pixel X, um de oito possível predictors é selecionado. O critério de seleção é uma predição que é o melhor possível para o valor X das já conhecidas amostras de *A, B* e *C.* Os predictors especificados são listados na *tabela 2*.

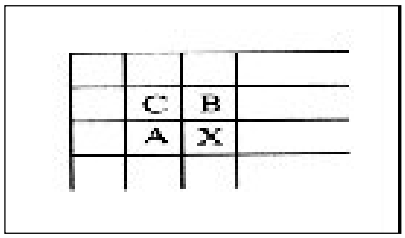


Figura 13: *Princípio da predição no modo de lossless.*

Selection Value	Prediction
0	No Prediction
1	X=A
2	X=B
3	X=C
4	X=A+B-C
5	X=A+(B-C)/2
6	X=B+(A-C)/2
7	X=(A+B)/2

tabela 2.

O número do predictor escolhido, como também a diferença da predição para o valor atual, é passado à codificação de entropia subseqüente. Codificação Entropica pode usar técnicas de Huffman ou aritmética.

6 Modo hierárquico

O modo hierárquico ou usa o algoritmos lossy baseados em DCT descritos acima ou alternativamente a técnica de compressão de lossless. A característica principal deste modo é a codificação de uma imagem em resoluções diferentes, i.e., o dados codificado contém imagens em várias resoluções. A imagem preparada é provada inicialmente a uma mais baixa resolução (reduzida pelo fator 2^n). Subseqüentemente, a resolução é reduzida por um fator 2^{n-1} verticalmente e horizontalmente. Esta imagem comprimida é então o resultado prévio. O processo é repetido até a resolução máxima da imagem estar comprimida.

Codificação hierárquica requer consideravelmente mais capacidade de armazenamento, mas a imagem comprimida está imediatamente disponível em resoluções diferentes. Então, aplicações que trabalham com mais baixas resoluções não têm que decodificar a imagem inteira e subseqüentemente aplicar algoritmos de processamento de imagens para reduzir a resolução - em outro palavra, escalonar fica barato. De acordo com as experiências dos autores com imagens escalonadas no contexto de DVI, qualquer escalonamento executado pela aplicação consome tempo considerável. Leva menos tempo de processamento na CPU para exibir uma imagem com resolução máxima que processar um escalonamento da imagem que exiba um número reduzido de pixels. Ainda, no caso de imagens codificadas de acordo com o modo de JPEG hierárquico, a exibição de uma figura de tamanho reduzido consome menos poder de processamento que qualquer resolução mais alta.

Disciplina: Computação Gráfica
Professor: Marcus Antônio
Assunto: Formatos BMP e JPG
Componentes:

- Enio
- Leonardo
- Marcelo Dantas
- Marcelo Santos

Universidade Federal do Rio Grande do Norte
Centro de Ciências Exatas
Departamento de Informática e Matemática Aplicada

Computação Gráfica

Natal, 25 de novembro de 1998.