# Transformada de Discreta de Co senos DCT

O primeiro passo, na maioria dos sistemas de compressão de imagens e vídeo, é identificar a presença de redundância espacial (semelhança entre um pixel e os pixels em sua vizinhança) em cada imagem, campo ou frame do vídeo.

Isto pode ser feito aplicando-se a Transformada Discreta de Cosenos (DCT) ao longo da imagem.  A DCT é um processo sem perda (*lossless*) e reversível.

Matematicamente, corresponde a uma simples multiplicação por uma matriz, que tem a propriedade de converter dados de amplitude espacial (os valores dos pixels) em coeficientes representando freqüências espaciais.

Ao invés de ser feita a transformação da imagem como um todo, com uma única multiplicação por matriz (o que exigiria uma matriz imensa !), o cálculo da DCT é feito geralmente para cada bloco de 8 por 8 amostras da imagem.

Quando a imagem é em tons de cinza (possui apenas a componente de luminância), a DCT é feita usando esses valores (de luminância).

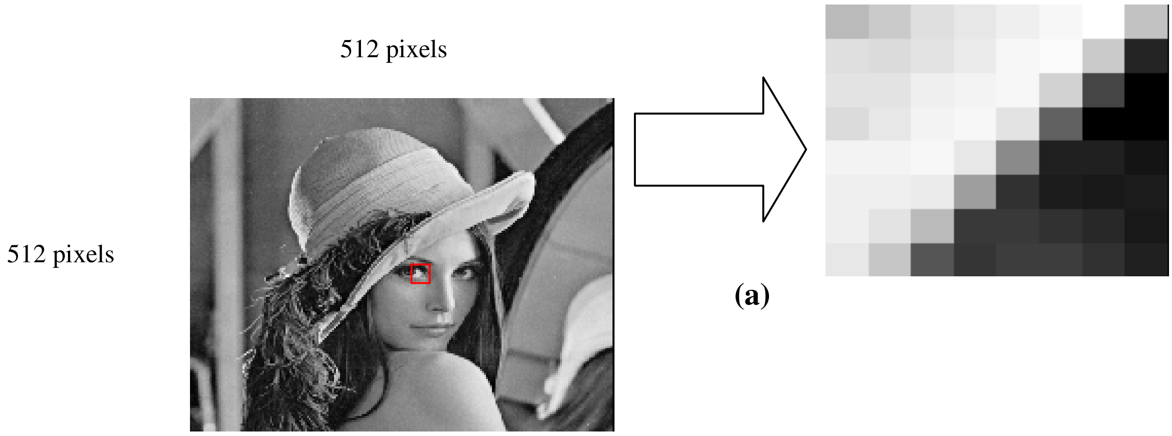
Para imagens coloridas, é calculada a DCT dos blocos correspondentes às amostras de crominância (valores RGB), ou de qualquer outra representação de cores adotada para a imagem.

A transformada DCT é relativamente simples de entender: Para cada dimensão de blocos a ser usada (a mais usada é de 8 x 8 pixels), existe uma matriz de DCT fixa. Realizar a transformação implica simplesmente em em recolher os 64 pixels deste bloco da imagem, fazer o cálculo da DCT para estes valores, obtendo-se novos 64 valores, que são chamados coeficientes da DCT. Este processo é repetido para todos os blocos da imagem.

Vejamos um exemplo para entender melhor a transformação DCT. A **figura 1a** mostra uma imagem (a Lena) em cinza com 512 x 512 pixels. Cada pixel da imagem é representado por 8 bits, onde o valor 0 corresponde ao preto, o valor 255 corresponde ao branco e os valores intermediários fornecem tons de cinza. Essa é uma típica imagem em tons de cinza.

Supõe-se que se calcula a DCT em blocos de 8 x 8 pixels. No exemplo, o bloco em questão localiza-se na região do olho da Lena e está identificado pelo quadrado superposto à figura. Considere o bloco de 8 x 8 pixels mostrado graficamente na **figura 1b**, cujos valores dos pixels são mostrados na **figura 1c**. Observe que o primeiro pixel deste bloco tem o valor 172, que corresponde a um tom de cinza.

É aplicada a DCT aos valores mostrados na **figura 1c**, obtendo-se assim os coeficientes da DCT, mostrados numericamente na **figura 1d**. A **figura 1e** é uma visualização gráfica dos coeficientes da DCT deste bloco. Observe que os coeficientes possuem valores **positivos e negativos. O** primeiro coeficiente (canto superior esquerdo) possui o maior valor, enquanto os últimos coeficientes (próximos ao canto inferior direito) possuem valores pequenos. Estes valores pequenos normalmente são desprezados (o que corresponde a substituí-los por zero), sem que a imagem sofra grandes deformações. A possibilidade de desprezar os coeficientes menos significativos sem muita perda da qualidade da imagem é justamente a razão principal de se usar a DCT na codificação de imagens (e vídeo), pois permite economia de bits.



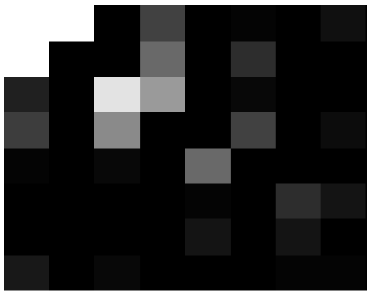

**(a)** **(b)**

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 172 | 179 | 188 | 191 | 196 | 200 | 204 | 174 |
| 188 | 187 | 190 | 193 | 199 | 201 | 178 | 101 |
| 189 | 189 | 196 | 197 | 199 | 183 | 117 | 84 |
| 186 | 192 | 197 | 199 | 189 | 130 | 85 | 85 |
| 198 | 197 | 199 | 192 | 149 | 100 | 100 | 95 |
| 195 | 195 | 193 | 158 | 108 | 98 | 96 | 98 |
| 195 | 189 | 171 | 111 | 111 | 108 | 104 | 96 |
| 192 | 177 | 124 | 110 | 113 | 113 | 108 | 100 |

**(c)**

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 1256,4 | 228,6 | -50,0 | 17,7 | -15,6 | 2 | -2,7 | 5,8 |
| 154,8 | -80 | -93,2 | 27 | -6,5 | 12,3 | 2 | 0,7 |
| 9,7 | -92,3 | 57,3 | 39,3 | -29 | 3,4 | 6,3 | 1,5 |
| 16,3 | -12,7 | 35,4 | -47,6 | -6,9 | 17,8 | -2,1 | 4,4 |
| 2,1 | -18,2 | 4 | -14,4 | 27,6 | -5,7 | -12,9 | -1,4 |
| -3 | -3,9 | 0,6 | -9,3 | 2,5 | -17,8 | 12,3 | 6,1 |
| -1,2 | -5,4 | 1,9 | -7,2 | 6,2 | -1,5 | 6,2 | -11,8 |
| 7,1 | -2,9 | 3,8 | 0,9 | -1,4 | 0 | 2 | 2,9 |

**(d)**



**(e)**

**Figura 1: (a) Imagem de 512x512 pixels; (b) Visualização do bloco 8x8 com os pixels da imagem representados em tons de cinza; (c) Valores dos pixels correspondentes a região com um quadrado vermelho na imagem (a); (d) Valores dos coeficientes DCT do bloco; (e) Visualização da DCT do bloco.**

Verifica-se que o primeiro valor da matriz DCT é mais importante, ele é chamado de valor médio ou valor DC, os outros coeficientes são chamados de valores AC.

A propriedade importante da DCT é que ela transforma os pixels de um domínio onde "todos são iguais", para um novo domínio, onde há hierarquia. O primeiro coeficiente da DCT ( o DC ) é mais importante que o 64º ( coeficiente AC de mais alta freqüência ). A figura 2 mostra a distribuição de freqüência em uma DCT de duas dimensões de 16 x 16 pixels.

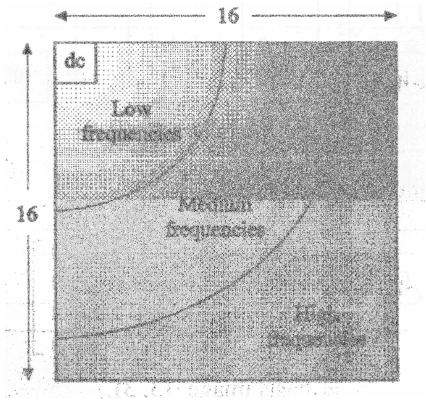



**Figura 2 Distribuição de freqüências**

Os coeficientes da DCT-2D podem ser *escaneados* em uma maneira predeterminada. Um modelo padrão chamado de scan zig-zag mostrado na figura 3 depende da distribuição de frequências.

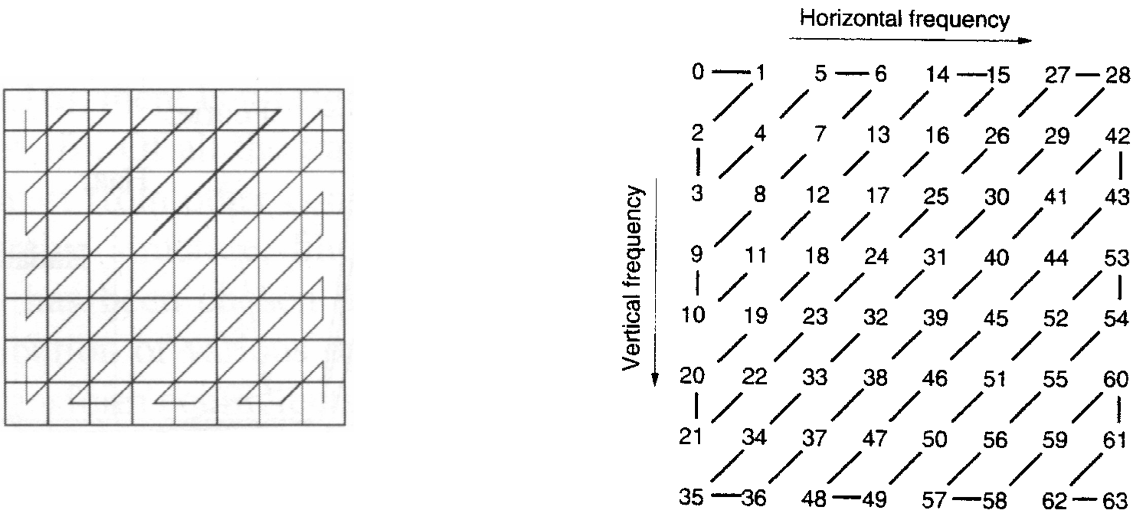
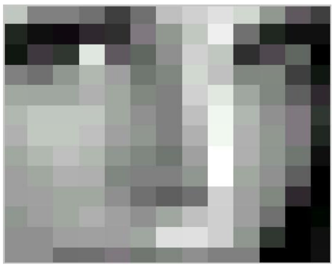
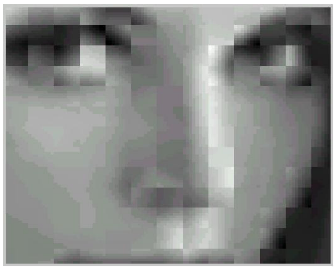



**Figura 3 Scan Zig-Zag**

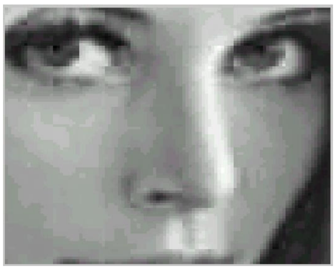
São mostradas abaixo quatro situações distintas, onde se pode notar sensíveis mudanças em relação à qualidade de imagem ( a imagem original considerada para este exemplo é uma partição da figura Lena512 ). Em todos os casos foram realizadas DCT's em blocos 8x8 pixels. No primeiro caso, figura (a), apenas a componente DC da imagem foi utilizada e os outros 63 coeficientes foram considerados iguais a zero. No segundo caso, figura (b), considerou-se a componente DC mais dois componentes AC, melhorando a definição da imagem, até que na última figura (figura (d) ) são usados todos os coeficientes. Observa-se com isso a cópia fiel em relação à figura original, não havendo perdas de informações.



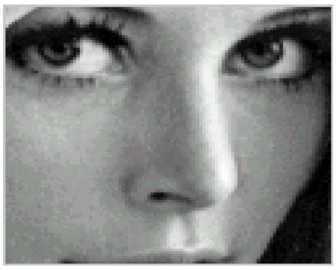
**(a) somente componente DC**



**(b) componente DC +2 AC mais baixos**



**(c) componente DC + 9 AC mais baixos**



**(d) todos os 64 coeficientes**

**Figura 4 Exemplo de DCT's utilizando coeficientes pre-determinados**

## *Separação da imagem dentro dos blocos:*

Ao se usar DCT em codificação de imagens, não se costuma calcular uma única DCT para a imagem toda, pois isso exigiria um número muito grande de cálculos. A alternativa adotada é segmentar a imagem em blocos e calcular a DCT para cada bloco.

As seguintes considerações são importantes:

- Um tamanho de bloco maior conduz à maior eficiência de codificação, mas requer maior poder computacional.
- Tipicamente são usados blocos de 8x8 ou 16x16 pixels. Blocos de 8x8 é um bom compromisso (*tradeoff*) entre a eficiência de compressão e a complexidade computacional.
- Uma melhor eficiência de compressão pode ser alcançada pelo uso de blocos de diferentes dimensões, entretanto isto aumenta a complexidade computacional.

Um diagrama esquemático de um codificador de imagem por transformada é mostrada na figura 5.

No transmissor, a imagem $f(n_1, n_2)$ é transformada e os coeficientes da transformada $T_f = (k_1, k_2)$ são quantizados. Os coeficientes da transformada quantizados são então codificados: $\hat{T}_f = (k_1, k_2)$ .

No receptor, as palavras-códgo são decodificadas e o resultado dos coeficientes quantizados $\hat{T}_f = (k_1, k_2)$ são transformados inversamente para obter a imagem reconstruída $\hat{f}(n_1, n_2)$ .

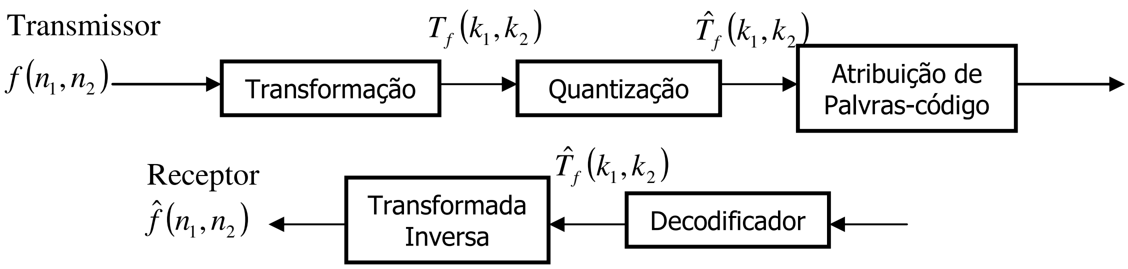



**Figura 5. Codificador de Imagem por transformada**

Em sistemas de codificação por transformada, os pixels são agrupados dentro de blocos. Um bloco de pixel é transformado dentro de outro domínio para produzir um conjunto de coeficientes que então são codificados e transmitidos.

As transformadas utilizadas para codificação de imagem são transformações lineares que podem ser expressas como:

$$T_f(k_1, k_2) = \sum_{n_1=0}^{N_1-1} \sum_{n_2=0}^{N_2-1} f(n_1, n_2) a(n_1, n_2; k_1, k_2)$$

$$f(n_1, n_2) = \sum_{k_1=0}^{N_1-1} \sum_{k_2=0}^{N_2-1} T_f(k_1, k_2) b(n_1, n_2; k_1, k_2)$$

onde $f(n_1, n_2)$ é uma seqüência de $N_1$ x $N_2$ pontos, $T_f(k_1, k_2)$ são os coeficientes da transformada com $N_1$ x $N_2$ pontos e $a(n_1, n_2; k_1, k_2)$ e $b(n_1, n_2; k_1, k_2)$ são chamados de função base. Observando essas equações das transformadas nota-se que $f(n_1, n_2)$ é uma combinação linear das funções base e que os coeficientes da transformada são as amplitudes das funções base na combinação linear.

## Transformada Discreta do Cosseno

A transformada discreta do cosseno (DCT) é a transformada mais largamente usada nos sistemas de codificação de imagem.

A seguir mostra-se como calcular a DCT. Pode-se entender este cálculo como sendo uma multiplicação por uma dada matriz (a matriz DCT), ou ainda o uso de uma dada fórmula (a fórmula da DCT). Ambas as maneiras são equivalentes.

A discussão inicia pelo uso da fórmula, depois discute-se a abordagem matricial, e mostra-se a equivalência entre as duas notações. Ao invés de se iniciar a discussão pela transformação de um bloco de pixels, usa-se um vetor. Entendendo-se o caso unidimensional, fica bem mais fácil entender o bidimensional.

A fórmula da DCT direta, que transforma um vetor x[n] em coeficientes X[k], com k variando de 0 a N-1 é:

$$X[k] = \alpha[k] \sum_{n=0}^{N-1} x[n] \cos\left(\frac{\pi (2n+1) k}{2N}\right)$$

A fórmula da DCT inversa, que obtém x[n], com n variando de 0 a N-1, a partir dos coeficientes DCT X[k] é:

$$x[n] = \sum_{k=0}^{N-1} \alpha[k] \; X[k] \cos\left(\frac{\pi (2n+1) k}{2N}\right)$$

onde em ambos os casos:

$$\alpha[k] = \begin{cases} \sqrt{\frac{1}{N}} & \text{for } k = 0 \\ \sqrt{\frac{2}{N}} & \text{for } k = 1, 2, \ldots, N-1 \end{cases}$$

Por exemplo, supondo que se deseja calcular a DCT do vetor
A=[3 –4 2 1].

Neste caso, N=4 e k=0 representam a coluna e a linha respectivamente. E tem-se: DCT = 1.0000 -0.3170 3.0000 4.4609. Multiplicando a matriz função base inversa pelo vetor DCT, que permite reconstruir o vetor A

---

Por curiosidade, note que a transformada discreta de Fourier (DFT) é calculada a partir das seguintes fórmulas (onde N é o número de pontos da DFT, x[n] é o sinal no tempo e X[k] é o sinal na freqüência):

Fórmula direta (análise) da DFT: $X[k] = \sum_{n=0}^{N-1} x[n] e^{\frac{-j2p\,nk}{N}}$ , calculada para k=0, 1, ..., N-1

Fórmula inversa (síntese) da DFT: $x[n] = \frac{1}{N} \sum_{k=0}^{N-1} X[k] e^{\frac{j2p\,nk}{N}}$ ,calculada para n=0, 1,...,N-1

Para lembrar, a identidade de Euler indica que $e^{j\theta} = \cos(\theta) + j\sin(\theta)$ , sendo j a raiz quadrada de -1. Como essa exponencial complexa aparece na fórmula da DFT, os coeficientes da DFT são, em geral, números complexos. Por outro lado, os coeficientes da DCT são sempre números reais.

---

Ao se usar DCT em imagens, ou seja, DCT 2-D, não se costuma calcular uma única DCT para a imagem toda, pois isto exigiria um número muito grande de cálculos. A alternativa adotada é segmentar a imagem em blocos e calcular a DCT para cada bloco. Tipicamente usam-se blocos de 8x8. A figura 6 ilustra isso.

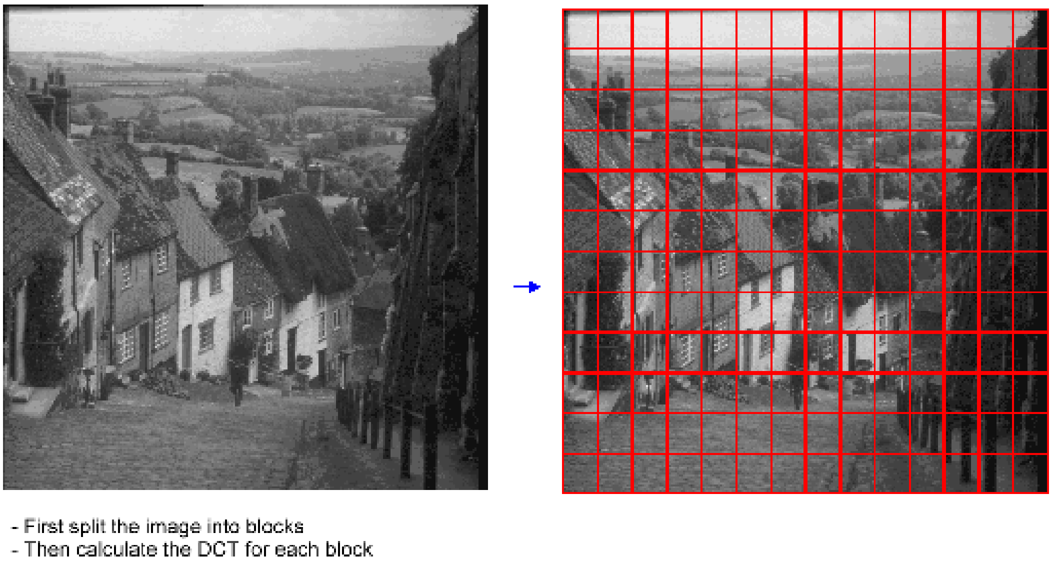
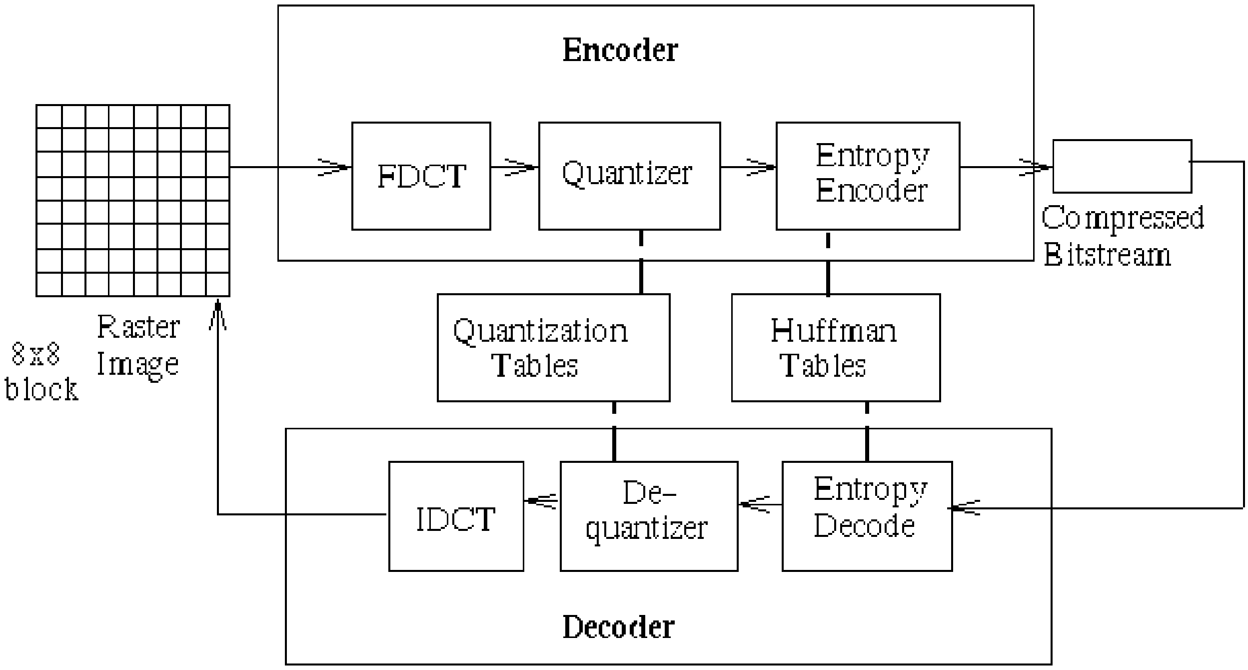



**Figura 6. Divisão da imagem em blocos para cálculo da DCT.**

Os coeficientes representam as componentes de frequência espacial que compõem o bloco original. Cada coeficiente pode ser visto como um peso aplicado a uma função base.

Quando se está navegando na Internet e faz-se o download de uma imagem no formato JPEG (extensão .JPG), que está codificada na forma progressiva do JPEG, a imagem vai se tornando cada vez mais nítida. Isto ocorre pois são enviados primeiro os coeficientes DC das DCT's e depois os coeficientes AC, gradativamente, desde os AC de mais baixa frequência até os de mais alta. O diagrama abaixo ilustra o JPEG.




No caso do JPEG, como as DCTs são processadas em blocos de 8 x 8 pixels, a fórmula da DCT bidimensional às vezes é escrita como:

$$F(u,v) = \frac{C_u}{2}\frac{C_v}{2}\sum_{y=0}^{7}\sum_{x=0}^{7} f(x,y)\cos\left[\frac{(2x+1)u\pi}{16}\right]\cos\left[\frac{(2y+1)v\pi}{16}\right]$$

With:

$$C_u = \begin{cases} \frac{1}{\sqrt{2}} & \text{if u} = 0, \\ 1 & \text{if u} > 0 \end{cases} ; C_v = \begin{cases} \frac{1}{\sqrt{2}} & \text{if v} = 0, \\ 1 & \text{if v} > 0 \end{cases}$$

e a fórmula inversa como:

$$f(x,y) = \sum_{u=0}^{7}\sum_{v=0}^{7} F(u,v)\frac{C_u}{2}\frac{C_v}{2}\cos\left[\frac{(2x+1)u\pi}{16}\right]\cos\left[\frac{(2y+1)v\pi}{16}\right]$$