

Hélio Pedrini

William Robson Schwartz



Análise de Imagens Digitais

Princípios, Algoritmos
e Aplicações



3

TRANSFORMADAS DE IMAGENS

Neste capítulo são descritas as transformadas de imagens, abordagens que apresentam uma vasta gama de aplicações no escopo de processamento e análise de imagens, por exemplo, utilizadas na extração de características, compressão de dados, segmentação e filtragem de imagens.

Uma das mais conhecidas transformadas de imagens é a transformada de Fourier (DFT, do inglês, *Discrete Fourier Transform*), utilizada para descrever funções por meio de senóides complexas. A versão discreta, utilizada inicialmente em processamento de sinais como ferramenta para análise de funções unidimensionais, apresenta extensão para duas dimensões, a qual recebe a atenção da comunidade de processamento e análise de imagens.

A aplicação de uma transformada resulta na alteração da representação inicial da imagem, de modo a prover informações que permitam uma análise mais acurada dos dados. Por exemplo, a representação resultante da aplicação da transformada de Fourier pode ser utilizada como indicativo das variações presentes nos tons dos pixels, pois provê uma descrição quantitativa das freqüências presentes na imagem.

Enquanto o conceito de freqüência em sinais unidimensionais, como ondas eletromagnéticas, está relacionado com o número de vibrações em um determinado intervalo de tempo, em imagens, esse conceito se relaciona à quantidade de variações na intensidade dos pixels, considerando regiões com tamanhos específicos. Para tal, utiliza-se o termo freqüência espacial. Dessa maneira, com a extensão do conceito para sinais unidimensionais, regiões da imagem que apresentam homogeneidade nos tons de cinza apresentam freqüência espacial baixa, ou simplesmente, freqüência baixa.

O foco deste capítulo está na descrição dos principais conceitos das transformadas bidimensionais contidas na classe das transformações de coordenadas, as quais efetuam uma transformação linear nos dados de entrada. Caso implementadas diretamente, a partir da definição, apresentam alto custo computacional, mesmo para imagens de tamanho moderado.

O restante deste capítulo está dividido como segue. A seção 3.1 apresenta os conceitos gerais das transformações de coordenadas. Na seção 3.2 são descritos os principais conceitos e propriedades da transformada discreta de Fourier, bem como é apresentada a transformada rápida de Fourier, algoritmo utilizado para executar a DFT em tempo aceitável, mesmo para imagens que apresentem maior dimensionalidade. As transformadas discretas de Hartley, do cosseno, do seno e a transformada de Fourier-Mellin são descritas nas seções 3.3 a 3.6. A seção 3.7 apresenta os conceitos da transformada de Karhunen-Loève. Embora não seja

classificada como uma transformação de coordenadas, a transformada wavelet é descrita na seção 3.8.

3.1 Transformações de Coordenadas

Esta seção apresenta os conceitos de uma classe das transformadas de imagens formada pelas transformações de coordenadas, as quais compõem um subconjunto das transformações lineares, cujo resultado depende da combinação linear dos dados da entrada com os coeficientes definidos para cada transformação. Os principais resultados necessários para a definição dessa classe de transformações são descritos no apêndice A.

As transformações de coordenadas, denotadas a partir de agora apenas como transformadas, são classificadas como unidimensionais quando a entrada é um vetor, e bidimensionais quando a entrada é uma matriz. Torna-se importante ressaltar que o termo unidimensional ou bidimensional está relacionado com a dimensionalidade do elemento de entrada, vetor ou matriz, entretanto, a transformação linear é executada de $\mathbb{C}^N \rightarrow \mathbb{C}^N$, na qual N representa o número de componentes do elemento de entrada.

A tabela 3.1 apresenta duas representações para as transformadas, a vetorial ou por meio de somatórios, em que X e x contêm os dados de entrada e Y e y contêm o resultado obtido com sua aplicação, cada um de seus componentes é denominado coeficiente da transformada. A matriz A e a função matricial $B(u, v)$ contêm o núcleo da transformada.

Tabela 3.1: Representações para as transformadas unidimensionais e bidimensionais.

	<i>Unidimensional</i>	<i>Bidimensional</i>
<i>Representação Vetorial</i>	$y^T = Ax^T$	$Y = B(u, v)X$
<i>Representação com Somatórios</i>	$y_u = \sum_{n=0}^{N-1} a_{u,n}x_n$	$y_{u,v} = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} b^{(u,v)}(m, n)x_{m,n}$

Percebe-se pela tabela 3.1, principalmente na representação com somatórios, que cada coeficiente da saída da transformada resulta da soma dos coeficientes de entrada ponderados pelos componentes do núcleo da transformada. Como a maioria das transformadas apresentadas neste capítulo é executada por meio dessas somas ponderadas, métodos gerais, mostrados na seção 3.1.3, podem ser desenvolvidos para a implementação dessas transformadas.

3.1.1 Núcleo da Transformada

O núcleo de uma transformada é composto pelas matrizes S ou S^{-1} da transformação de coordenadas, mostradas pelas equações A.19 e A.20, respectivamente, do apêndice A. Dessa maneira, as transformadas alteram as coordenadas de um vetor em relação à base $B_{\{n\}}$ para coordenadas desse vetor em relação à base $B_{\{n\}}$, utilizando a equação A.18 ou A.21.

Na seção 3.1.2 será descrito que os elementos de uma imagem podem ser considerados como coordenadas de um ponto em relação à base canônica, portanto, não existe a necessidade de se utilizar diretamente a equação A.19 para determinação do núcleo de uma transformada. O núcleo da transformada será a matriz da base de destino ou sua inversa.

O núcleo de uma transformada unidimensional é representado por uma matriz \mathbf{A} , $N \times N$, contendo em suas linhas os vetores $\mathbf{a}_u = [a_{u,0} \ a_{u,1} \ \dots \ a_{u,N-1}]$, em que $u = 0, 1, \dots, N-1$. Por exemplo, a transformada discreta do cosseno, descrita na seção 3.4, apresenta os vetores do núcleo definidos conforme a equação 3.1, onde $n = 0, 1, \dots, N-1$ e $k(n) = 1/\sqrt{2}$ se $n = 0$ e $k(n) = 1$, caso contrário.

$$a_{u,n} = k(n) \sqrt{\frac{2}{N}} \cos\left(\frac{\pi n(u+1/2)}{N}\right) \quad (3.1)$$

Considerando os vetores definidos na equação 3.1, o núcleo da transformada discreta do cosseno para um vetor de entrada com 8 componentes é representado pela matriz \mathbf{A} , mostrada em 3.2.

$$\mathbf{A} = \frac{1}{\sqrt{8}} \begin{bmatrix} 1 & 1 & 1 & \dots & 1 \\ 1 & a_{1,1} & a_{1,2} & \dots & a_{1,7} \\ 1 & a_{2,1} & a_{2,2} & \dots & a_{2,7} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & a_{7,1} & a_{7,2} & \dots & a_{7,7} \end{bmatrix} \quad (3.2)$$

A matriz \mathbf{A} pode ser representada graficamente por meio de N planos cartesianos, um para cada vetor do núcleo, nos quais os pontos denotam os componentes do u -ésimo vetor que compõe o núcleo. A figura 3.1 mostra a representação gráfica do núcleo da transformada discreta do cosseno para uma função de entrada composta de 8 elementos, ou seja, um vetor com oito componentes.

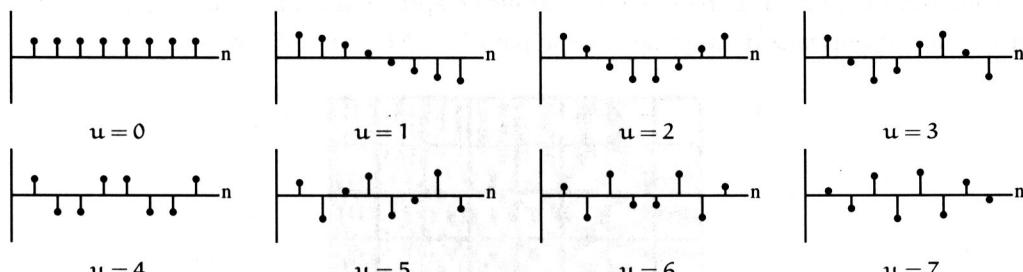


Figura 3.1: Representação gráfica do núcleo da transformada discreta do cosseno para um vetor de entrada composto de oito elementos.

De maneira direta, pode-se encontrar um algoritmo que determina o núcleo de uma transformada unidimensional, pois cada elemento é obtido por meio de uma função bidimensional $f(u, n)$ e seu resultado é armazenado na u -ésima linha e n -ésima coluna da matriz do núcleo. As etapas necessárias para determinação do núcleo de uma transformada unidimensional são apresentadas no algoritmo 3.1.

O núcleo para uma transformada bidimensional é composto por uma função matricial que depende das coordenadas do ponto para o qual a transformada está sendo executada a cada instante. Utiliza-se a função matricial $\mathbf{B}(u, v)$, apresentada em 3.3, para determinação do núcleo, onde $u, v = 0, 1, \dots, N-1$ representam as coordenadas do ponto em que está sendo executada a transformada e cada componente $b^{(u,v)}(m, n)$ depende da transformada em questão.

Núcleo para a transformada unidimensional

Algoritmo para obtenção do núcleo de uma transformada unidimensional, em que o elemento $a_{u,n}$ compõe a u -ésima linha e n -ésima coluna da matriz do núcleo e a função $f(u, n)$ depende da transformada a ser executada. Por exemplo, para transformada discreta do cosseno utiliza-se a equação 3.1 como $f(u, n)$.

```

1 para u = 0 até u < N faça
2   para n = 0 até n < N faça
3     au,n ← f(u, n)

```

Algoritmo 3.1: Núcleo para a transformada unidimensional.

$$\mathbf{B}(u, v) = \begin{bmatrix} b^{(u,v)}(0,0) & b^{(u,v)}(0,1) & \dots & b^{(u,v)}(0,N-1) \\ b^{(u,v)}(1,0) & b^{(u,v)}(1,1) & \dots & b^{(u,v)}(1,N-1) \\ b^{(u,v)}(2,0) & b^{(u,v)}(2,1) & \dots & b^{(u,v)}(2,N-1) \\ \vdots & \vdots & \ddots & \vdots \\ b^{(u,v)}(M-1,0) & b^{(u,v)}(M-1,1) & \dots & b^{(u,v)}(M-1,N-1) \end{bmatrix} \quad (3.3)$$

Uma representação gráfica da função matricial $\mathbf{B}(u, v)$ pode ser obtida por meio de uma imagem em tons de cinza composta de $U \times V$ blocos, cada um composto por $M \times N$ pixels, apresentando intensidade proporcional aos valores obtidos pela instanciação da função $\mathbf{B}(u, v)$. Um exemplo dessa representação gráfica é apresentado na figura 3.2, que mostra o núcleo da transformada discreta do cosseno para $N = M = U = V = 8$.

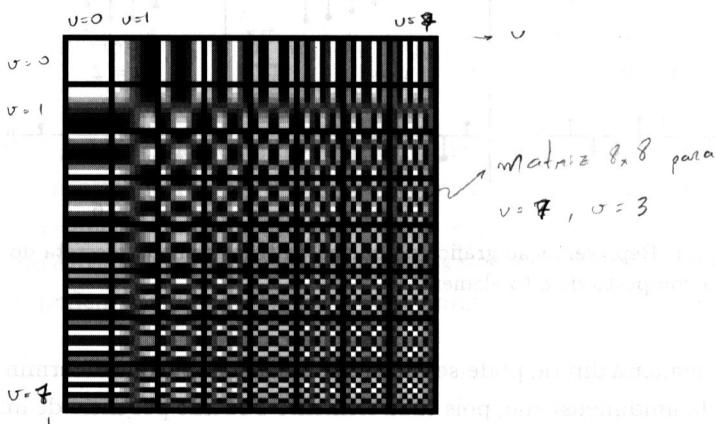


Figura 3.2: Representação gráfica do núcleo da transformada discreta do cosseno para uma matriz de entrada com ordem 8.

O algoritmo 3.2 apresenta as etapas necessárias para a determinação do núcleo de uma transformada bidimensional a ser executada sobre uma matriz de entrada com ordem N . Esse algoritmo não gera todas as matrizes, pois seria necessário o armazenamento de N^4 coeficientes, apenas determina a função matricial para cada ponto da transformada. Dessa maneira, torna-se necessária a execução da transformada para o ponto (u, v) tão logo seja determinada a função $\mathbf{B}(u, v)$ para valores específicos de u e v .

Núcleo para a transformada bidimensional

Algoritmo para obtenção do núcleo de uma transformada bidimensional, em que $f(m, n, u, v)$ depende da transformada a ser executada. Retorna a matriz $B(u, v)$, para $u, v = 0, 1, \dots, N - 1$. Dessa maneira, este método deve ser executado para cada novo par de coordenadas (u, v) .

```

1  para m = 0 até m < N faça
2    para n = 0 até n < N faça
3      b(u,v)(m, n) ← f(m, n, u, v)

```

Algoritmo 3.2: Núcleo para a transformada bidimensional.

O núcleo de algumas transformadas bidimensionais pode ser decomposto em duas matrizes **C** e **D**, permitindo que sua execução seja efetuada em duas etapas, inicialmente sobre as linhas da matriz de entrada e então sobre as colunas da matriz resultante da primeira aplicação. As transformadas que apresentam essa propriedade são denominadas *transformadas separáveis*.

Exemplo 3.1 O núcleo da transformada bidimensional de Fourier é obtido por meio da função $b^{(u,v)}(m, n) = \exp(-2\pi i(mu + nv)/N)$, que pode ser decomposto em exponenciais que dependem apenas de n, u e m, v , conforme mostra a equação 3.4.

$$b^{(u,v)}(m, n) = \exp(-2\pi i n v / N) \exp(-2\pi i m u / N) \quad (3.4)$$

Na seção 3.1.3 é descrito como as transformadas separáveis, como a mostrada neste exemplo, podem ser executadas com custo computacional menor que as transformadas que não apresentam tal propriedade. ■

3.1.2 Definição de Imagem

Conforme descrito na seção anterior, o núcleo de uma transformada é a matriz de mudança de base, obtida por meio da equação A.19. Entretanto, a utilização direta dessa equação é desnecessária para determinação do núcleo, pois os elementos de uma imagem podem ser considerados componentes de um vetor com coordenadas representadas na base canônica. Dessa maneira, a determinação do núcleo da transformada dá-se apenas pela inversão da matriz da base de destino. Explica-se a seguir a representação da imagem apresentando coordenadas em relação à base canônica.

Inicialmente, define-se a composição da base. Para isso, considera-se que os elementos de uma imagem composta de $T = M \times N$ pixels sejam acrescentados ao vetor f e define-se T vetores da base $B_{\{T\}}$ conforme a equação 3.5, em que $n = 0, 1, \dots, T - 1$ representa o n -ésimo vetor de $B_{\{T\}}$ e $n' = 0, 1, \dots, T - 1$ denota o índice de cada componente desse vetor.

$$n' B_{\{T\}}_n = \begin{cases} 1, & \text{se } n = n' \\ 0, & \text{caso contrário} \end{cases} \quad (3.5)$$

Utilizando a equação 3.5, obtém-se a base $B_{\{T\}} = \{\mathbf{b}_0, \mathbf{b}_1, \dots, \mathbf{b}_{T-1}\}$, com sua matriz mostrada em 3.6, em que cada vetor \mathbf{b}_n é composto de T componentes. $B_{\{T\}}$ é ortonormal, pois $\langle \mathbf{b}_i, \mathbf{b}_j \rangle = 0$ quando $i \neq j$, $|\mathbf{b}_i| = 1$ e os vetores de $B_{\{T\}}$ são linearmente independentes.

$$\mathbf{B} = \begin{bmatrix} 1 & 0 & 0 & \dots & 0 \\ 0 & 1 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 1 \end{bmatrix} \quad (3.6)$$

A equação 3.7 mostra a combinação linear do vetor f com os vetores da base $B_{\{T\}}$, resultando no vetor i , que representa as coordenadas do vetor f em relação à base $B_{\{T\}}$. Dessa maneira, uma imagem pode ser considerada um ponto em um espaço vetorial gerado pela base $B_{\{T\}}$, denominada base canônica.

$$i_m = \sum_{n=0}^T f_n b_{m,n}, \quad m = 0, 1, \dots, T-1 \quad (3.7)$$

Como as imagens são representadas por matrizes, a transformação de um vetor com $T = M \times N$ componentes para uma matriz X , $M \times N$ pode ser efetuada pela decomposição de i em M vetores com N componentes, resultando na matriz mostrada em 3.8.

$$X = \begin{bmatrix} i_0 & i_1 & \dots & i_{N-1} \\ i_N & i_{N+1} & \dots & i_{2N-1} \\ \vdots & \vdots & \ddots & \vdots \\ i_{(M-1)N} & i_{(M-1)N+1} & \dots & i_{(M \times N)-1} \end{bmatrix} \quad (3.8)$$

3.1.3 Representação das Transformadas de Imagens

Uma transformação de coordenadas para imagens pode ser escrita na forma mostrada na equação 3.9, em que a imagem de entrada é representada pela matriz X , $N \times N$, o núcleo é definido pela função matricial $B(u, v)$ e os coeficientes resultantes são armazenados na matriz Y . A inversa da transformada é apresentada na equação 3.10, na qual a função matricial $A(m, n)$, obtida com a equação A.20, contém o núcleo.

$$y_{u,v} = \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} x_{m,n} b^{(u,v)}(m, n)$$

$$Y = B(u, v)X \quad (3.9)$$

$$x_{m,n} = \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} y_{u,v} a^{(m,n)}(u, v)$$

$$X = A(m, n)Y \quad (3.10)$$

As equações 3.9 e 3.10 são diretamente implementáveis, a primeira é efetuada por meio do algoritmo 3.3, enquanto a implementação da sua inversa (equação 3.10) é obtida de maneira análoga. A desvantagem de se utilizar o algoritmo 3.3 está no elevado custo computacional

de $O(N^4)$, característica que torna sua utilização inviável, mesmo para valores moderados de N.

Transformadas de imagens

Algoritmo para execução de transformadas de imagens. A matriz X contém a imagem de entrada, Y contém os coeficientes resultantes da aplicação transformada, ambas com ordem N. A função matricial B(u, v) representa o núcleo da transformada.

```

1  para u = 0 até u < N faça
2    para v = 0 até v < N faça
3      yu,v ← 0
4      para m = 0 até m < N faça
5        para n = 0 até n < N faça
6          yu,v ← yu,v + xm,nb(u,v)(m, n)

```

Algoritmo 3.3: Transformadas de imagens.

Utilizando o conceito de transformadas separáveis descrito no final da seção 3.1.1, será apresentado um algoritmo com ordem de complexidade $O(N^3)$ para execução das transformadas de imagem. Nesse caso, devido às características apresentadas pelo núcleo, a transformada pode ser executada inicialmente sobre as linhas da matriz de entrada e, posteriormente, sobre as colunas da matriz resultante da primeira aplicação.

As transformadas que apresentam o núcleo separável podem ser representadas como mostra a equação 3.11, em que $c_{m,u}$ e $d_{n,v}$ dependem especificamente da transformada. Por exemplo, na transformada discreta de Fourier, $c_{m,u} = \exp(-2\pi i mu/N)$ e $d_{n,v} = \exp(-2\pi i nv/N)$.

$$y_{u,v} = \sum_{m=0}^{N-1} c_{m,u} \sum_{n=0}^{N-1} x_{m,n} d_{n,v} \quad (3.11)$$

Com a decomposição do núcleo, as transformadas bidimensionais são representadas de maneira distinta daquela mostrada na tabela 3.1, pois passam a apresentar a forma $Y = CXD$, em que C e D denotam o novo núcleo da transformada e X a matriz que contém a imagem de entrada, todas com ordem N. O algoritmo 3.4 mostra um método para determinação dos coeficientes de uma transformada que apresenta o núcleo separável.

O algoritmo 3.4 utiliza $2N^3$ multiplicações, além da matriz auxiliar E para a determinação dos coeficientes da transformada, apresentando ordem de complexidade $O(N^3)$. Apesar da redução da complexidade em relação ao algoritmo 3.3, ainda não é suficiente para viabilizar a utilização de transformadas em imagens que apresentam valores elevados de N.

3.2 Transformada Discreta de Fourier

A transformada discreta de Fourier, capaz de descrever funções a partir de exponenciais complexas, provém de uma especificação da série discreta de Fourier (DFS, do inglês, *Discrete Fourier Series*). Inicialmente, a transformada discreta de Fourier foi utilizada em processamento de sinais para determinar as freqüências presentes em uma dada função unidimensional. Posteriormente, sua versão bidimensional, a qual provê informações a respeito

Transformadas separáveis

Algoritmo para execução de transformadas que apresentam o núcleo separável. A execução desse algoritmo é dada em duas etapas: primeiro, sobre as linhas da imagem de entrada e, posteriormente, sobre as colunas da matriz auxiliar E. Decompõe-se o núcleo original da transformada nas matrizes C e D.

```

1 // executa a transformada sobre as linhas da entrada
2 para u = 0 até u < N faça
3     para v = 0 até v < N faça
4         eu,v ← 0
5         para i = 0 até i < N faça
6             eu,v ← eu,v + xu,i.di,v
7
8 // executa a transformada sobre as colunas de E
9 para u = 0 até u < N faça
10    para v = 0 até v < N faça
11        yu,v ← 0
12        para i = 0 até i < N faça
13            yu,v ← yu,v + cu,i.ei,v

```

Algoritmo 3.4: Transformadas separáveis.

das variações na intensidade de cinza, passou a ser utilizada pela comunidade de processamento e análise de imagens.

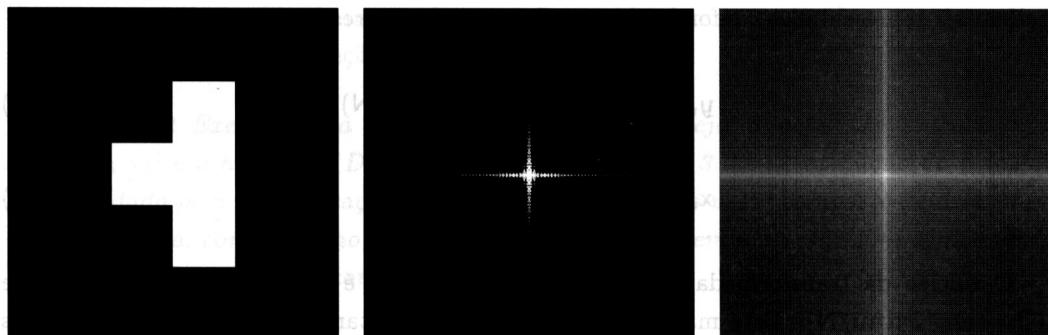
A DFT é uma transformação de coordenadas que resulta em componentes pertencentes ao conjunto dos números complexos, em que cada coeficiente é obtido pela combinação linear dos elementos da entrada com o núcleo da transformada. Sua aplicação é efetuada para um vetor contendo N elementos, obtendo-se como resultado um vetor que apresenta o mesmo número de componentes. Os elementos do vetor de entrada podem ser obtidos com a aplicação da transformada inversa sobre os coeficientes resultantes.

Os componentes complexos resultantes da DFT normalmente não são utilizados na forma $z = a + bi$. Entretanto, utiliza-se a fase, representada pelo argumento de z, mostrado na equação 3.12, contendo informações essenciais sobre a estrutura da imagem e o espectro de Fourier, o módulo de z mostrado na equação 3.13, que descreve informações sobre as freqüências espaciais contidas na imagem, no entanto, sem apresentar referências espaciais.

$$\arg z = \arctan \left(\frac{b}{a} \right) \quad (3.12)$$

$$\mod z = \sqrt{a^2 + b^2} \quad (3.13)$$

Dado que os coeficientes do espectro de Fourier apresentam valores com magnitudes distintas, apenas a visualização dos elementos mais claros é possível quando representa-se o espectro de Fourier na forma de imagem em tons de cinza, aplica-se a função $\log[1 + (a^2 + b^2)^{\frac{1}{2}}]$. A diferença entre os resultados obtidos sem e com a utilização do logaritmo pode ser percebida nas figuras 3.3(b) e 3.3(c), respectivamente.



(a) Imagem original; (b) resultado da transformada de Fourier; (c) resultado depois de aplicar a função $\log[1 + (a^2 + b^2)^{\frac{1}{2}}]$.

Figura 3.3: Representação do espectro de Fourier para uma imagem monocromática. (a) imagem original; (b) espectro de Fourier resultante; (c) resultado após a aplicação da função $\log[1 + (a^2 + b^2)^{\frac{1}{2}}]$.

3.2.1 Transformada Unidimensional de Fourier

De grande utilidade em processamento de sinais, a transformada unidimensional de Fourier provém da série discreta de Fourier. A DFS para uma seqüência periódica com período N , ou seja, $\bar{x}_n = \bar{x}_{n+N}$, tem sua representação mostrada na equação 3.14, na qual \bar{x}_n denota o n -ésimo elemento do vetor que contém os elementos da seqüência e \bar{y}_u é denominado coeficiente de Fourier. A equação 3.15 mostra como são obtidos os coeficientes \bar{y}_u , em que $u = 0, 1, \dots, N - 1$.

$$\bar{x}_n = \sum_{u=0}^{N-1} \bar{y}_u \exp(2\pi i n u / N) \quad (3.14)$$

$$\bar{y}_u = \frac{1}{N} \sum_{n=0}^{N-1} \bar{x}_n \exp(-2\pi i n u / N) \quad (3.15)$$

Se uma seqüência não periódica de tamanho finito for representada pela série discreta de Fourier, esta será chamada de transformada discreta de Fourier. A equação 3.16 mostra a representação de uma seqüência finita composta de N componentes por meio da DFS. Os coeficientes de Fourier, y_u , são relacionados com a série de Fourier de maneira análoga, conforme mostra a equação 3.17.

$$x_n = \begin{cases} \bar{x}_n, & 0 \leq n \leq N - 1 \\ 0, & \text{caso contrário} \end{cases} \quad (3.16)$$

$$y_u = \begin{cases} \bar{y}_u, & 0 \leq u \leq N - 1 \\ 0, & \text{caso contrário} \end{cases} \quad (3.17)$$

No caso de imagens, geralmente se desconsidera a existência de periodicidade, ou seja, $f(x, 0) \neq f(x + N, 0)$, para uma imagem $N \times N$. Dessa maneira, como se tem interesse apenas no intervalo $0 \leq x \leq N - 1$ e $0 \leq y \leq N - 1$, a DFT pode ser utilizada para análise de f . A partir das equações 3.16 e 3.17, tem-se que DFT e sua inversa para um vetor de entrada x

podem ser representadas conforme as equações 3.18 e 3.19, respectivamente.

$$y_u = \frac{1}{N} \sum_{n=0}^{N-1} x_n \exp(-2\pi i n u / N) \quad (3.18)$$

$$x_n = \sum_{u=0}^{N-1} y_u \exp(2\pi i n u / N) \quad (3.19)$$

O núcleo da transformada unidimensional de Fourier é mostrado em 3.20, em que $\omega_u^n = \exp(2\pi i n u / N)$. Esta matriz pode ser utilizada diretamente para determinação dos coeficientes de Fourier de uma função discreta unidimensional. Tais coeficientes descrevem o comportamento desta função por meio de exponenciais complexas.

$$B = \frac{1}{N} \begin{bmatrix} 1 & \omega_1^{-1} & \omega_2^{-2} & \dots & \omega_{N-1}^{-(N-1)} \\ 1 & \omega_2^{-1} & \omega_2^{-2} & \dots & \omega_{N-1}^{-(N-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \omega_{N-1}^{-1} & \omega_{N-1}^{-2} & \dots & \omega_{N-1}^{-(N-1)} \end{bmatrix} \quad (3.20)$$

As figuras 3.4 e 3.5 mostram uma representação gráfica para o núcleo da transformada unidimensional de Fourier para um vetor composto de oito elementos. A primeira representa os componentes reais e a segunda os componentes complexos, pois a expressão $\exp(2\pi i n u / N)$ pode ser decomposta em $\cos(2\pi n u / N) + i \sin(2\pi n u / N)$.

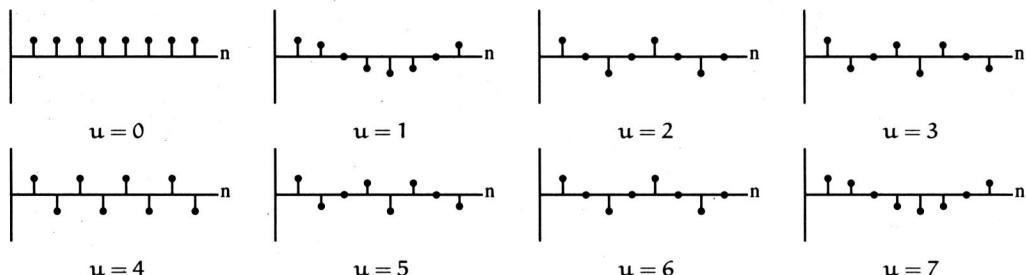


Figura 3.4: Representação gráfica dos componentes reais do núcleo da DFT unidimensional para um vetor de entrada composto de oito elementos.

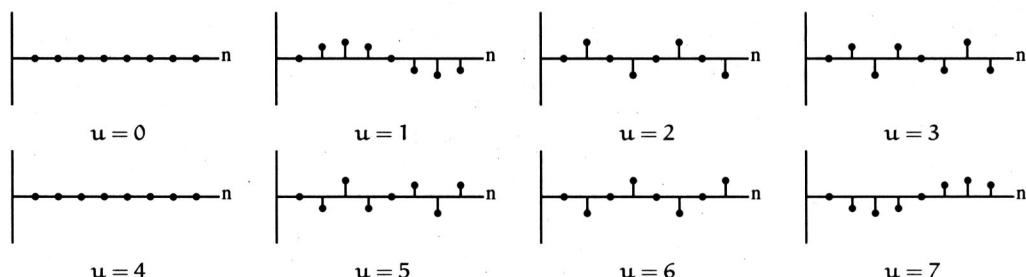


Figura 3.5: Representação gráfica dos componentes complexos do núcleo da DFT unidimensional para um vetor de entrada composto de oito elementos.

O exemplo 3.2 apresenta a execução da DFT unidimensional sobre um vetor de entrada composto por quatro elementos. Percebe-se que os elementos localizados na metade inferior do vetor resultante diferem dos elementos iniciais apenas pelo sinal do termo intermediário.

Essa propriedade é utilizada para o desenvolvimento de algoritmos rápidos para determinar os coeficientes de Fourier (seção 3.2.4).

Exemplo 3.2 Execução da DFT unidimensional. Seja o vetor de entrada $\mathbf{x} = [x_1 \ x_2 \ x_3 \ x_4]$ e o núcleo da DFT mostrada pela equação 3.20. A execução da transformada é dada pela multiplicação da matriz do núcleo pelo vetor composto pelos elementos de entrada, conforme mostrado em 3.21, resultando em um vetor com componentes no domínio dos números complexos.

$$\mathbf{v}^T = \frac{1}{4} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -i & -1 & i \\ 1 & -1 & 1 & -1 \\ 1 & i & -1 & -i \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \frac{1}{4} \begin{bmatrix} x_1 + x_3 + (x_2 + x_4) \\ x_1 - x_3 + i(x_4 - x_2) \\ x_1 + x_3 - (x_2 + x_4) \\ x_1 - x_3 - i(x_4 - x_2) \end{bmatrix} \quad (3.21)$$

Em processamento de imagens, o objetivo normalmente está na obtenção de informações sobre quais freqüências espaciais presentes na imagem ou em qual orientação ocorrem as maiores variações de intensidade de tons de cinza. Após a determinação desses dados, pode-se, por exemplo, aplicar um filtro no domínio de freqüência visando eliminar os componentes que apresentam freqüências indesejadas ao experimento efetuado.

O resultado da DFT unidimensional expressa quais freqüências estão presentes na função de entrada, de modo que quanto maior a freqüência apresentada pela função de entrada, o espectro de Fourier apresentará componentes em regiões mais distantes da origem. A figura 3.6 contém duas funções compostas de freqüências distintas. O espectro de Fourier da função mostrada em (b) apresenta componentes distantes da origem, indicando a presença de componentes de alta freqüência na função de entrada.

Uma representação abstrata da relação espaço-freqüência para a DFT pode ser dada por um plano apresentando dois eixos ortogonais, n e u , correspondendo, respectivamente, ao domínio espacial e ao domínio de freqüência, como mostra a figura 3.7. A aplicação da DFT em uma função de entrada $f(n)$ altera sua representação do eixo n para o eixo u , correspondendo a uma rotação de $\alpha = \pi/2$ radianos, efetuada nesse plano.

Aplicações sucessivas da DFT sobre a função representada por $f(n)$ ocasionam rotações em ângulos múltiplos de $\pi/2$ radianos. Por exemplo, se executada $\mathcal{FF}\{f(n)\}$, sua representação estará novamente sobre o eixo de espaço, entretanto, apresentada como $f(-n)$, onde \mathcal{F} denota a aplicação da DFT sobre uma função de entrada. A figura 3.8 mostra o resultado da aplicação de $\mathcal{FF}\{f(n)\}$ sobre a função $f(n) = \sin(x) + \sin(4x)$, nota-se que $\mathcal{FF}\{f(n)\}$ é representada no domínio espacial, no entanto, $\mathcal{FF}\{f(n)\} = f(-n)$.

Com a utilização da DFT, torna-se impossível obter representações que não estejam em ângulos múltiplos de $\pi/2$, ou seja, apenas são possíveis representações de $f(n)$ que estejam sobre os eixos n e u da figura 3.7. Para permitir rotações em ângulos diferentes, torna-se necessária a determinação de um operador linear que possibilite representações que não estejam sobre esses eixos.

Com o intuito de generalizar o ângulo de rotação possível para representação de $f(n)$ foi proposta a transformada fracional de Fourier (FRFT, do inglês, *Fractional Fourier Transform*), que apresenta o operador linear \mathcal{R}^α , onde α é o ângulo em que será representada

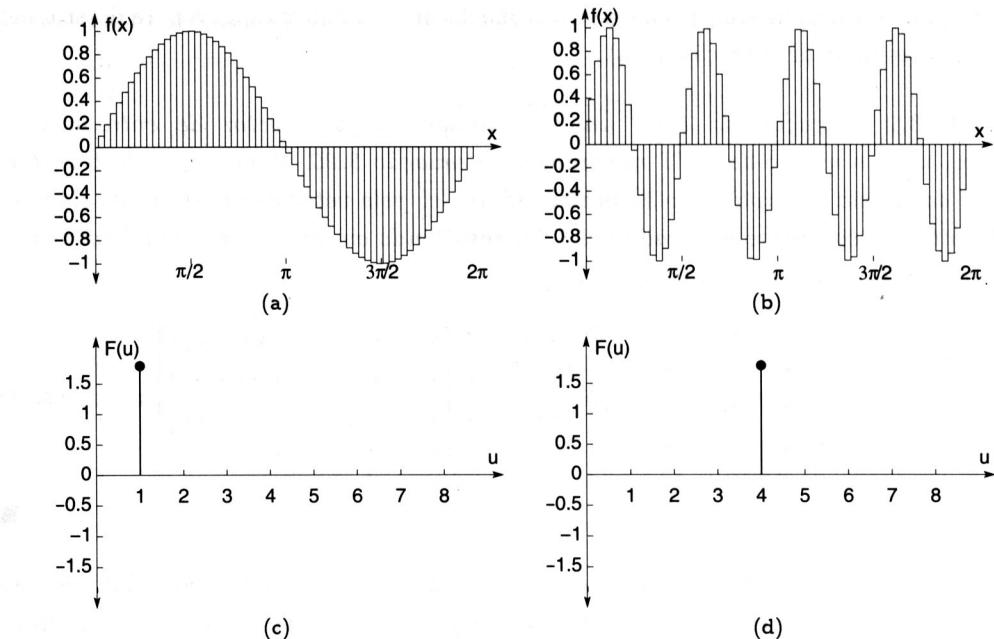


Figura 3.6: Aplicação da DFT e representação do espectro de Fourier. (a) função de entrada apresen- tando componentes de baixa freqüência; (b) função apresentando componentes de alta freqüência; (c) espectro de Fourier resultante da aplicação da DFT à função mostrada em (a); (d) espectro de Fourier resultante da aplicação da DFT à função apresentada em (b).

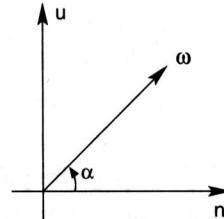


Figura 3.7: Plano representando espaço-freqüência por meio dos eixos ortogonais n e u , em que ω denota a função de entrada após ter sua representação alterada.

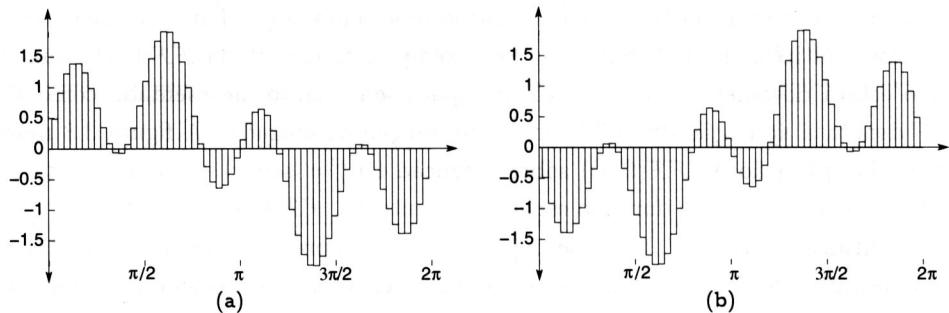


Figura 3.8: Múltiplas aplicações da DFT unidimensional. (a) função de entrada $f(n) = \sin(x) + \sin(4x)$; (b) resultado da aplicação de $\mathcal{F}\{f(n)\}$.

a função $f(n)$. Quando se utiliza $\mathcal{R}^{\pi/2}$, obtém-se o mesmo resultado de uma aplicação da DFT, o que a torna um caso específico da FRFT.

Essa generalização da transformada de Fourier surgiu durante a década de 1980 (Namias 1980, McBride e Kerr 1987), recebendo atenção da comunidade de processamento de sinal (Almeida 1994). As definições e a implementação da versão discreta da FRFT são descritas nos trabalhos de Bultheel e Sulbaran (2004) e Ozaktas et al. (1996).

3.2.2 Transformada Bidimensional de Fourier

Extensão da versão unidimensional, a transformada bidimensional de Fourier pode ser aplicada diretamente em imagens sem que haja a necessidade de transformar os dados de entrada em um vetor antes da obtenção dos coeficientes de Fourier. As equações 3.22 e 3.23 mostram a transformada de Fourier e a sua inversa, respectivamente, em que $x_{m,n}$ denota o pixel localizado na m -ésima linha e n -ésima coluna da matriz de entrada X e $y_{u,v}$ representa o coeficiente de Fourier situado na u -ésima linha e v -ésima coluna de Y , matriz resultante da transformada.

$$y_{u,v} = \frac{1}{N^2} \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} x_{m,n} \exp(-2\pi i(mu + nv)/N) \quad (3.22)$$

$$x_{m,n} = \sum_{v=0}^{N-1} \sum_{u=0}^{N-1} y_{u,v} \exp(2\pi i(mu + nv)/N) \quad (3.23)$$

Assim como na versão unidimensional, a expressão $\exp(-2\pi i(mu + nv)/N)$, que compõe o núcleo da DFT 2D, pode ser decomposta na soma $\cos(-2\pi(mu + nv)/N) + i \sin(-2\pi(mu + nv)/N)$. Dessa maneira, os componentes do núcleo podem ser representados separadamente por meio de duas imagens monocromáticas. A figura 3.9 mostra essa representação para uma matriz de entrada com ordem 8.

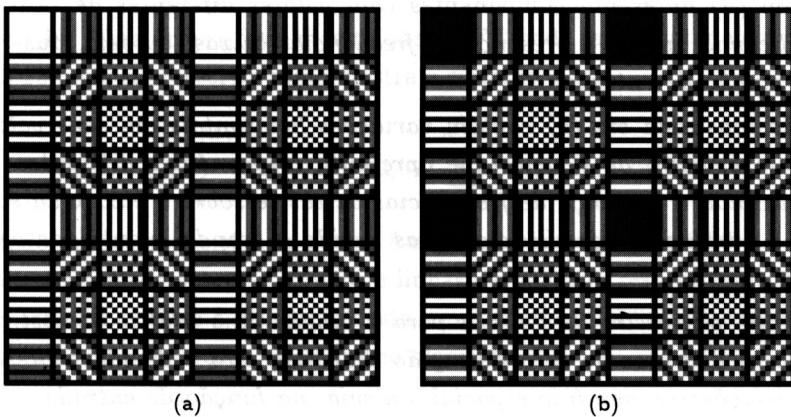


Figura 3.9: Representação gráfica do núcleo da transformada bidimensional de Fourier para uma matriz de entrada com ordem 8. (a) componente real do núcleo; (b) componente complexo do núcleo.

O espectro de Fourier obtido a partir da DFT bidimensional, além de prover informações sobre a orientação das estruturas presentes na imagem de entrada, efetua um mapeamento das variações nos tons de cinza dos pixels. Um número pequeno de variações de intensidade de cinza em um determinado espaço indica a presença de regiões de baixa freqüência espacial,

enquanto um número maior de variações indica a presença de regiões de freqüência espacial alta na imagem.

Pode-se resumir a interpretação do espectro de Fourier para uma função bidimensional por meio do plano mostrado na figura 3.10. A presença de componentes em regiões mais claras indica a existência de freqüência espacial baixa na imagem de entrada, enquanto a presença de componentes em regiões mais escuras indica a presença de freqüência espacial alta.

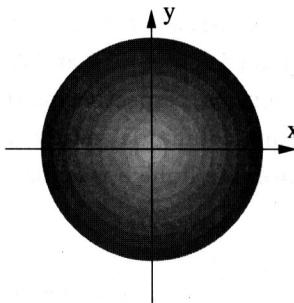


Figura 3.10: Interpretação do espectro de Fourier resultante da aplicação da DFT bidimensional.

O exemplo 3.3 apresenta a interpretação do espectro de Fourier para as funções bidimensionais cujos gráficos são mostrados na figura 3.11. Percebe-se a possibilidade de se obter informações sobre a orientação das estruturas presentes nas funções de entrada.

Exemplo 3.3 A figura 3.11 representa funções bidimensionais (a, b, c) e seus respectivos espectros de Fourier (d, e, f). O espectro de Fourier para a função homogênea mostrada em (a) apresenta componentes com valores diferentes de zero apenas na origem, indicando apenas a presença de freqüências baixas, ou seja, ausência de variações na função de entrada.

A função mostrada em (b) contém variações horizontais em seus valores, resultando em um espectro de Fourier que apresenta componentes diferentes de zero na origem, indicando a presença de freqüências baixas, e sobre o eixo x da superfície, o que indica a presença de freqüências altas devido à grande quantidade de variações horizontais.

Finalmente, o espectro de Fourier para a função cujo gráfico é mostrado em (c) contém valores distintos de zero na região diagonal; isto é devido à presença dessa classe de variações no domínio espacial, ou seja, na função de entrada. ■

A figura 3.12 ilustra a aplicação da DFT bidimensional em uma imagem partitionada em blocos compostos de 64×64 pixels. As figuras 3.12(d) e 3.12(f) apresentam o resultados da transformada inversa quando se mantêm apenas os coeficientes mais significativos, ou seja, aqueles mais próximos da origem, como mostrado nas figuras 3.12(c) e 3.12(e). Devido à manutenção de muitos componentes de baixa freqüência na figura 3.12(e), a reconstrução da imagem, mostrada na figura 3.12(f), apresenta melhor qualidade se comparada com a figura 3.12(d).

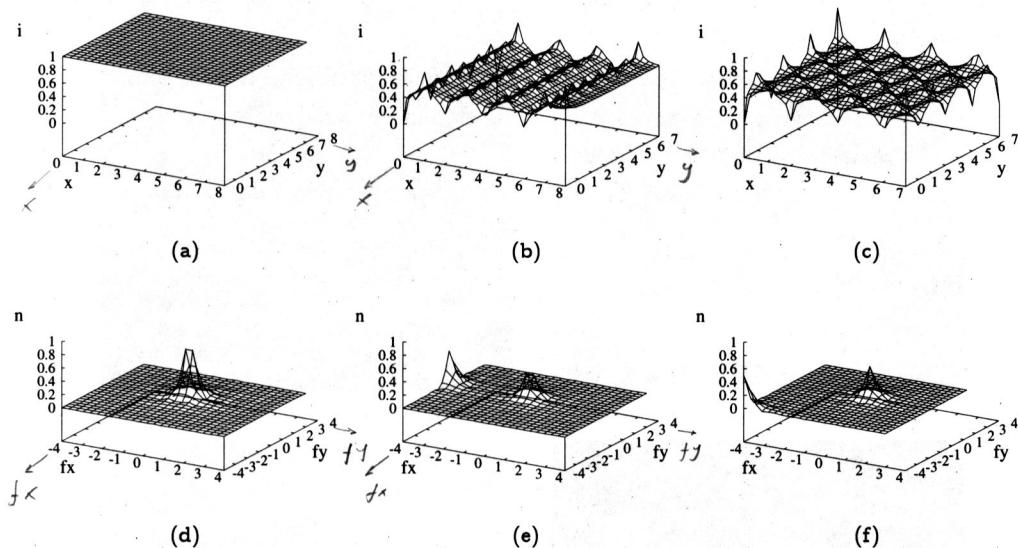


Figura 3.11: Resultados da transformada bidimensional de Fourier. (a) função homogênea; (b) função com variações horizontais; (c) função que apresenta variações diagonais; (d)-(f) apresentam o espectro de Fourier das funções mostradas em (a)-(c).

3.2.3 Propriedades da Transformada de Fourier

Esta seção descreve as principais propriedades da transformada discreta de Fourier. Considera-se que a DFT é aplicada a uma matriz de entrada \mathbf{X} e o seu resultado é representado pela matriz \mathbf{Y} . Ambas as matrizes apresentam ordem N , sendo compostas pelos elementos $x_{m,n}$ e $y_{u,v}$, respectivamente.

Separabilidade Esta propriedade mostra que a DFT pode ser executada em duas etapas. A equação 3.22 pode ser reescrita como mostra a equação 3.24.

$$y_{u,v} = \frac{1}{N^2} \sum_{m=0}^{N-1} \exp(-2\pi i mu/N) \sum_{n=0}^{N-1} x_{m,n} \exp(-2\pi i nv/N) \quad (3.24)$$

O resultado da decomposição do núcleo é mostrado nas equações 3.25 e 3.26, as quais executam a transformada sobre as colunas e linhas da imagem, respectivamente, conforme descrito na seção 3.1.1. Com a utilização desta propriedade, a DFT bidimensional pode ser executada com $O(N^3)$ operações, descritas no algoritmo 3.4.

$$e_{m,v} = \sum_{n=0}^{N-1} x_{m,n} \exp(-2\pi i nv/N) \quad (3.25)$$

$$y_{u,v} = \frac{1}{N^2} \sum_{m=0}^{N-1} e_{m,v} \exp(-2\pi i mu/N) \quad (3.26)$$

Simetria Conjugada A propriedade de simetria conjugada mostra que a magnitude da transformada está centrada na origem. Quando se considera o caso unidimensional, o es-

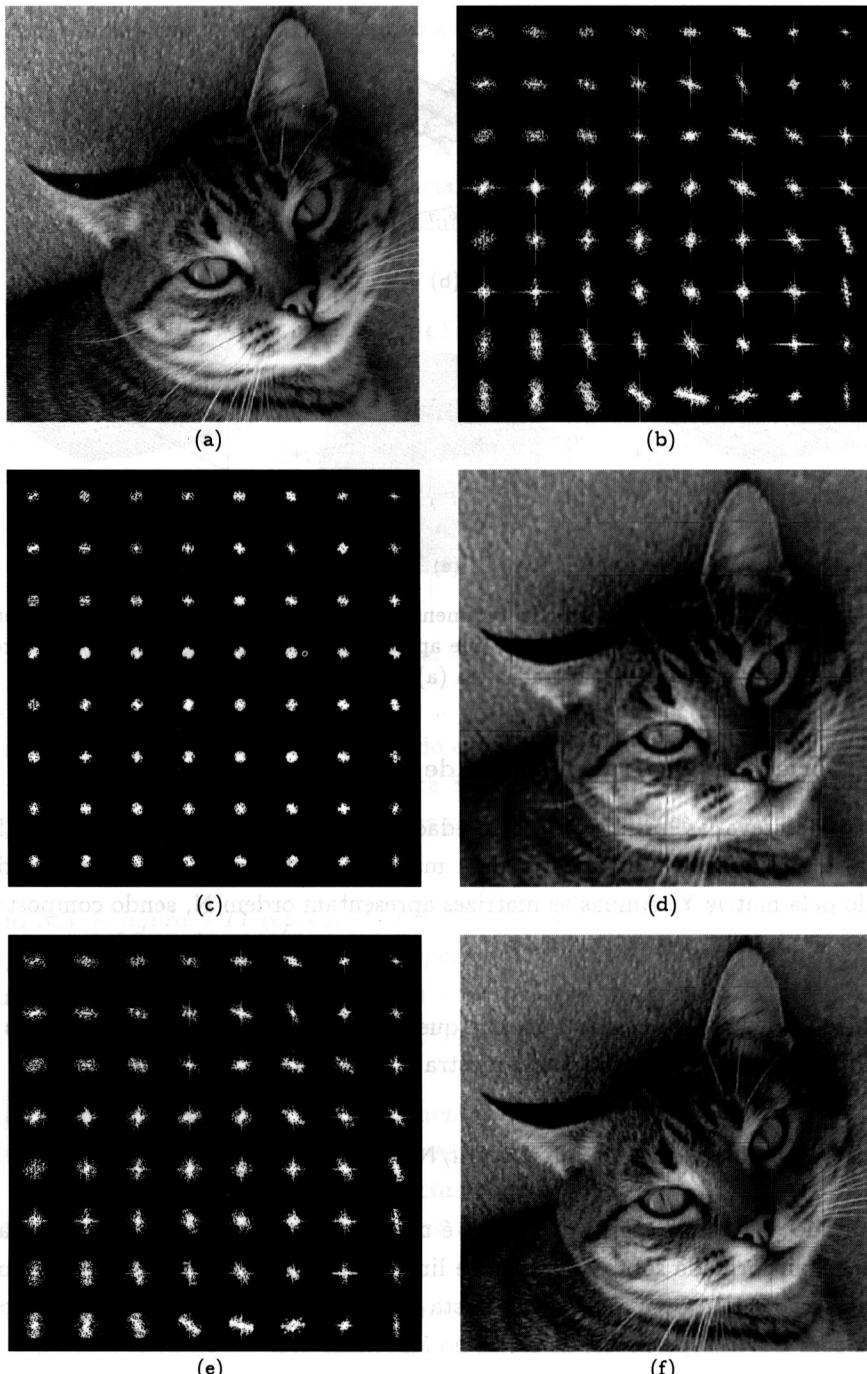


Figura 3.12: Aplicação da DFT em blocos compostos de 64×64 pixels. (a) imagem original; (b) resultado da DFT; (c) espectro mantendo apenas os coeficientes em um raio de 8 pixels da origem de cada bloco; (d) transformada inversa de (c); (e) espectro mantendo os coeficientes no raio de 16 pixels da origem; (f) transformada inversa de (e).

pectro de Fourier mostrado na figura 3.13 apresenta sua magnitude refletida em relação ao eixo u , ou seja, está centrada na origem do plano cartesiano.

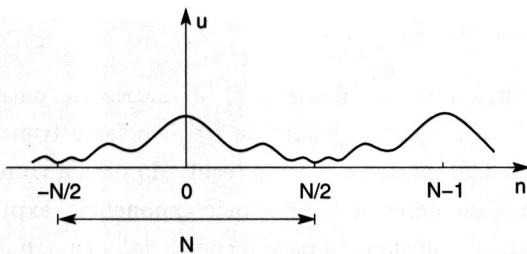


Figura 3.13: Representação da DFT unidimensional no plano cartesiano. A magnitude é centrada na origem e os valores à esquerda do eixo u são reflexões dos valores à direita desse eixo.

Algoritmos rápidos para execução da DFT utilizam essa propriedade para reduzir o custo computacional necessário para se determinar os coeficientes de Fourier, como o algoritmo proposto por Sorensen et al. (1987), o qual determina a DFT para funções de entrada compostas apenas por números reais.

De maneira mais precisa, se a matriz de entrada \mathbf{X} contém apenas componentes reais, o resultado da transformada discreta de Fourier apresenta a propriedade da simetria conjugada, ou seja, $y_{u,-v} = y_{-u,-v}^*$, conforme mostrado na equação 3.27. Tal propriedade pode não ser válida para entradas que apresentam componentes no domínio dos números complexos devido à necessidade de se considerar explicitamente o conjugado complexo de $x_{m,n}$.

$$\begin{aligned} y_{-u,-v}^* &= \frac{1}{N^2} \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} x_{m,n} \exp(2i\pi(-ux - vy)/N) \\ &= \frac{1}{N^2} \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} x_{m,n} \exp(-2i\pi(ux + vy)/N) \\ &= y_{u,v} \end{aligned} \quad (3.27)$$

Translação A propriedade da translação mostra que, ao multiplicar a representação no domínio espacial por um exponencial, a origem do plano de freqüências é transladada. O mesmo ocorre com a origem do plano espacial quando se multiplica todos os coeficientes de Fourier por um exponencial. Essas propriedades são mostradas nas equações 3.28 e 3.29, onde $\rho = \exp(2i\pi(u_0 m + v_0 n)/N)$ e $\theta = \exp(-2i\pi(m_0 u + n_0 v)/N)$.

$$\begin{aligned} \mathcal{F}(\rho x_{m,n}) &= \frac{1}{N^2} \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} x_{m,n} \exp(2i\pi(u_0 m + v_0 n)/N) \exp(-2i\pi(mu + nv)/N) \\ &= \frac{1}{N^2} \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} x_{m,n} \exp(-2i\pi[m(u - u_0) + n(v - v_0)]/N) \\ &= y_{(u-u_0),(v-v_0)} \end{aligned} \quad (3.28)$$

$$\begin{aligned} \mathcal{F}^{-1}(\theta y_{u,v}) &= \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} y_{u,v} \exp(-2i\pi(m_0 u + n_0 v)/N) \exp(2i\pi(mu + nv)/N) \\ &= \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} y_{u,v} \exp(2i\pi[u(m - m_0) + v(n - n_0)]/N) \end{aligned} \quad (3.29)$$

$$= x_{(m-m_0), (n-n_0)}$$

A equação 3.28 mostra que, ao multiplicar a matriz de entrada pelo exponencial $\exp(i2\pi(u_0m + v_0n)/N)$, a origem do plano de freqüências é transladada para a coordenada (u_0, v_0) . A equação 3.29 mostra o mesmo resultado para a transformada inversa, que, com a multiplicação dos coeficientes de Fourier pelo exponencial $\exp(-i2\pi(m_0u + n_0v)/N)$, a origem do plano espacial é transladada para a coordenada (m_0, n_0) .

Aplicando-se diretamente a transformada de Fourier em uma imagem composta de elementos reais, a origem do plano de freqüências coincidirá com a origem da imagem resultante, o que dificulta a interpretação. Com o objetivo de facilitar a interpretação da DFT, a origem do plano de freqüências pode ser transladada para o centro da imagem, para isso, substitui-se $u_0 = v_0 = N/2$ na equação 3.28, o que resulta na equação 3.30.

$$\begin{aligned} \mathcal{F}(x_{m,n} \exp(2i\pi[(N/2)m + (N/2)n]/N)) &= \mathcal{F}(x_{m,n} \exp(i\pi(m+n))) \\ &= \mathcal{F}(x_{m,n} (-1)^{m+n}) \end{aligned} \quad (3.30)$$

Conforme a equação 3.30, para que a origem do plano de freqüências seja transladada para o centro da imagem, a DFT deve ser aplicada após a multiplicação de cada elemento da matriz de entrada por $(-1)^{m+n}$. Um exemplo do resultado da aplicação da transformada utilizando a translação da origem é mostrado na figura 3.14(c), que indica a presença de freqüência espacial alta nas direções horizontal e vertical. Efetuando-se a translação da origem, torna-se mais simples a determinação das freqüências e orientações apresentadas pelas estruturas contidas na imagem de entrada.

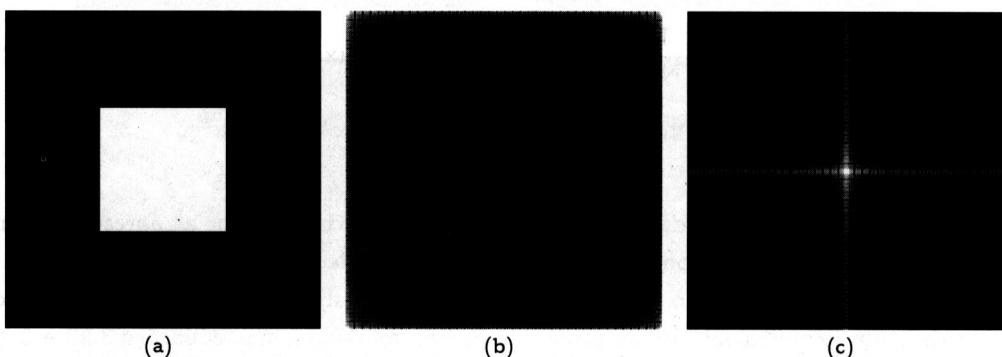


Figura 3.14: Transformada de Fourier com e sem a utilização da translação da origem. (a) imagem original; (b) transformada sem utilizar a translação da origem; (c) DFT utilizando a translação da origem para a coordenada $(N/2, N/2)$ por meio da multiplicação dos coeficientes de entrada por $(-1)^{m+n}$

Um resultado importante da propriedade da translação da origem está no fato do espectro de Fourier ser invariante em relação à translação, conforme mostra a equação 3.31. Essa propriedade pode ser exemplificada pelas figuras 3.15(b) e 3.15(c), as quais são obtidas a partir de translações efetuadas na imagem mostrada na figura 3.15(a). Embora efetuada a translação, o espectro de Fourier resultante das três imagens não apresenta alterações.

$$|\theta y_{u,v}| = |y_{u,v}| |\cos(2\pi(um_0 + vn_0)/N) - i \sin((um_0 + vn_0)/N)| = |y_{u,v}| \quad (3.31)$$

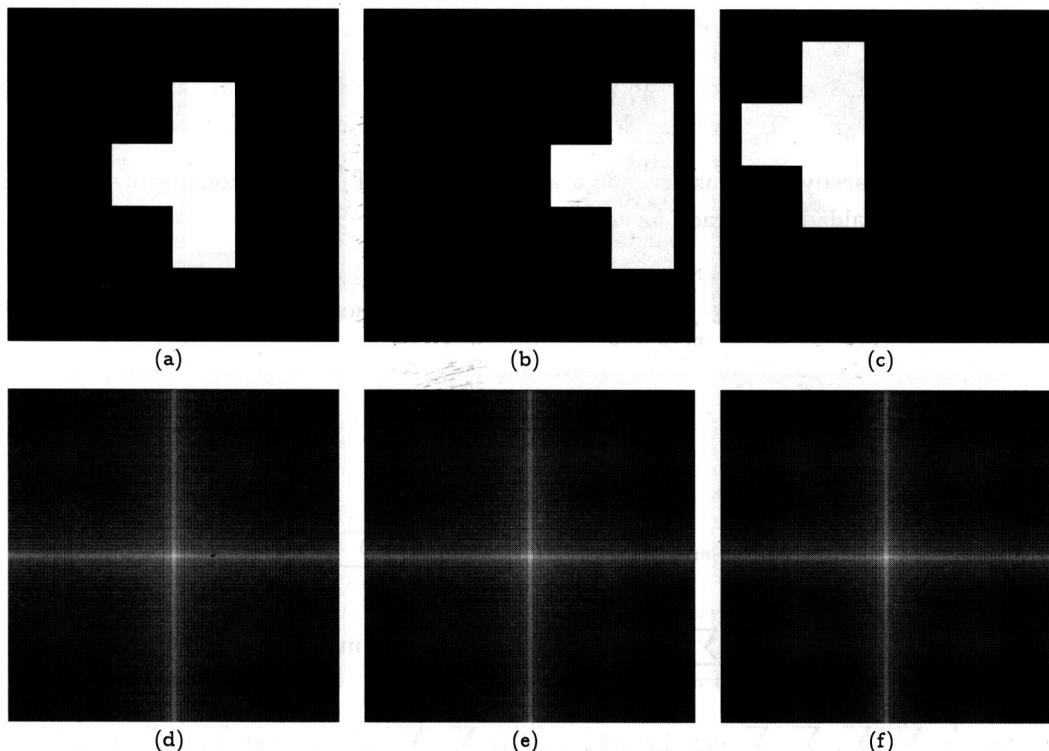


Figura 3.15: Transformada de Fourier considerando a translação na imagem de entrada. (a) imagem original; (b)-(c) translações do objeto contido na imagem; (d)-(f) espectros de Fourier resultantes.

Valor Médio O coeficiente $y_{0,0}$ resultante da transformada de Fourier contém o valor médio dos N^2 elementos da matriz de entrada, conforme mostra a equação 3.32, em que \bar{X} denota o valor médio da imagem de entrada.

$$y_{0,0} = \frac{1}{N^2} \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} x_{m,n} = \bar{X} \quad (3.32)$$

Mudança de Escala A multiplicação da matriz de entrada pela constante α implica a multiplicação do resultado da transformada pela mesma constante, como mostrado na equação 3.33.

$$\begin{aligned} \mathcal{F}(\alpha x_{m,n}) &= \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} \alpha x_{m,n} \exp(-i2\pi(mu + nv)/N) \\ &= \alpha \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} x_{m,n} \exp(-i2\pi(mu + nv)/N) \\ &= \alpha y_{u,v} \end{aligned} \quad (3.33)$$

A multiplicação das coordenadas (m, n) pelas constantes α e β , respectivamente, implica a divisão das coordenadas por essas constantes e a divisão do número complexo resultante

por $|\alpha\beta|$, como mostra a equação 3.34.

$$\mathcal{F}(x_{\alpha m, \beta n}) = \frac{1}{|\alpha\beta|} y_{u/\alpha, v/\beta} \quad (3.34)$$

Teorema de Parseval Considerando a aplicação da DFT bidimensional sobre a matriz \mathbf{X} , é válida a igualdade mostrada na equação 3.35.

$$\frac{1}{N^2} \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} |x_{m,n}|^2 = \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} |y_{u,v}|^2 \quad (3.35)$$

$$\begin{aligned} & \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} |y_{u,v}|^2 = \\ &= \frac{1}{N^4} \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} x_{m,n} \exp(-2\pi i(mu + nv)/N) \sum_{j=0}^{N-1} \sum_{k=0}^{N-1} x_{j,k}^* \exp(2\pi i(ju + kv)/N) \\ &= \frac{1}{N^4} \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} \sum_{j=0}^{N-1} \sum_{k=0}^{N-1} x_{m,n} x_{j,k}^* \exp(2\pi iu(j-m)/N) \exp(2\pi iv(k-n)/N) \\ &= \frac{1}{N^4} \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} \sum_{m=0}^{N-1} \sum_{j=0}^{N-1} \exp(2\pi iu(j-m)/N) \sum_{n=0}^{N-1} \sum_{k=0}^{N-1} x_{m,n} x_{j,k}^* \exp(2\pi iv(k-n)/N) \\ &= \frac{1}{N^4} \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} \sum_{m=0}^{N-1} \sum_{j=0}^{N-1} \exp(2\pi iu(j-m)/N) x_{m,n} x_{j,n}^* N \\ &= \frac{1}{N^4} \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} x_{m,n} x_{m,n}^* N^2 \\ &= \frac{1}{N^2} \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} |x_{m,n}|^2 \end{aligned} \quad (3.36)$$

Rotação Esta propriedade mostra que, se a imagem de entrada sofrer uma rotação de um ângulo θ_0 , o espectro de Fourier também será afetado pela rotação. Portanto, a invariância obtida com respeito à translação não ocorre no caso da rotação. As figuras 3.16 e 3.17 apresentam o espectro de Fourier obtido a partir de imagens que sofreram rotações.

Linearidade Sejam as matrizes \mathbf{X} e \mathbf{W} de entrada para a transformada de Fourier. A aplicação da DFT sobre $\mathbf{X} + \alpha\mathbf{W}$ é igual a aplicação de $\mathcal{F}\{\mathbf{X}\} + \alpha\mathcal{F}\{\mathbf{W}\}$, conforme mostrado na equação 3.37, em que α representa um escalar.

$$\begin{aligned} \mathcal{F}\{\mathbf{X} + \alpha\mathbf{W}\} &= \frac{1}{N^2} \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} (x_{m,n} + \alpha w_{m,n}) \exp(-i2\pi(mu + nv)/N) \\ &= \frac{1}{N^2} \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} x_{m,n} \exp(-i2\pi(mu + nv)/N) \\ &\quad + \alpha \frac{1}{N^2} \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} w_{m,n} \exp(-i2\pi(mu + nv)/N) \end{aligned} \quad (3.37)$$

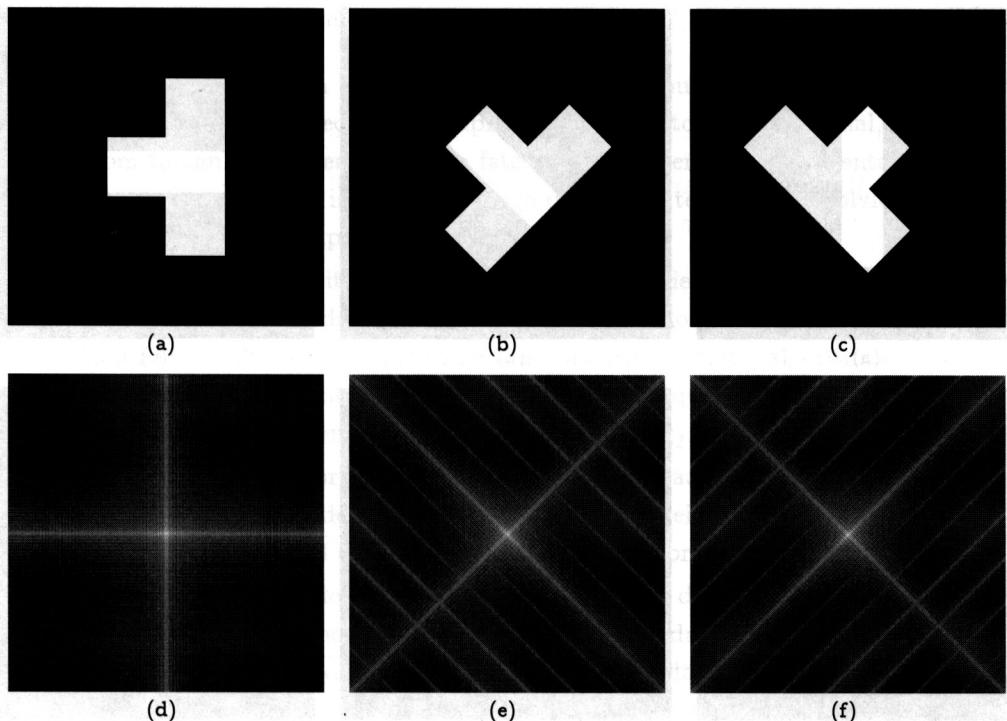


Figura 3.16: Espectro de Fourier obtido a partir da rotação de uma imagem. (a) imagem original; (b) rotação de 45° ; (c) rotação de 135° ; (d)-(f) respectivos espectros de Fourier.

$$= \mathcal{F}\{\mathbf{X}\} + \alpha \mathcal{F}\{\mathbf{W}\}$$

Convolução Uma das propriedades mais difundidas da transformada da Fourier é a da *convolução*. Essa propriedade mostra que a transformada de Fourier, aplicada após a operação de convolução entre duas funções no domínio espacial (definida pela equação 3.38), equivale à multiplicação dessas funções no domínio de freqüência, conforme mostrado na equação 3.39, na qual se efetua a mudança de variáveis $t = n - m$.

$$\mathbf{f} * \mathbf{g} = \sum_{n=0}^{N-1} f_n g_{p-n}, \quad p = 0, 1, \dots, N-1 \quad (3.38)$$

$$\begin{aligned} \mathcal{F}\{\mathbf{f} * \mathbf{g}\} &= \sum_{m=0}^{N-1} f_m \sum_{n=0}^{N-1} g_{n-m} \exp(-2\pi i nk/N) \\ &= \sum_{m=0}^{N-1} f_m \sum_{t=-m}^{N-1} g_t \exp(-2\pi i(t+m)k/N) \\ &= \sum_{m=0}^{N-1} f_m \exp(-2\pi imk/N) \sum_{t=0}^{N-1} g_t \exp(-2\pi itk/N) \\ &= \mathcal{F}\{\mathbf{f}\} \mathcal{F}\{\mathbf{g}\} \end{aligned} \quad (3.39)$$

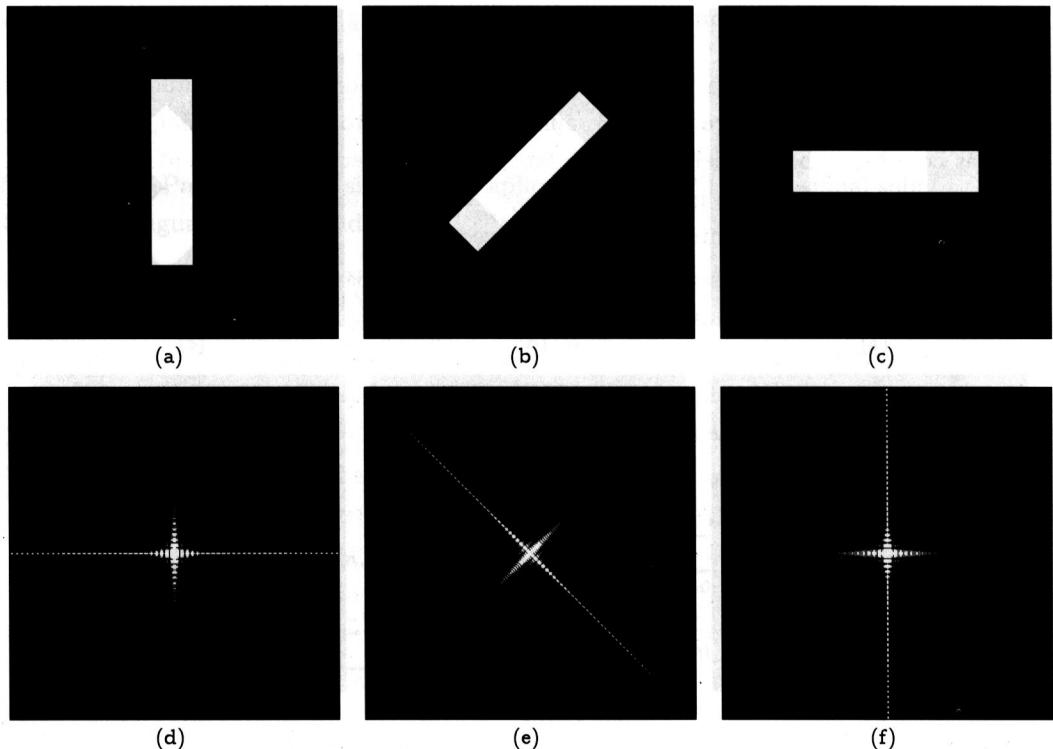


Figura 3.17: Espectro de Fourier obtido a partir da rotação de uma imagem sem aplicar a função $\log[1 + (a^2 + b^2)^{\frac{1}{2}}]$. (a) imagem original; (b) rotação de 45° ; (c) rotação de 90° ; (d)-(f) respectivos espectros de Fourier.

A figura 3.18 apresenta um exemplo da aplicação da operação de convolução entre as funções $f = [1 \ 1]$ e $g = [0 \ 1 \ 1 \ 1]$, obtendo-se como resultado $f * g = [0 \ 1 \ 2 \ 2 \ 1 \ 0]$. A função g é refletida na origem e deslocada sobre função f . A cada novo deslocamento, calcula-se a multiplicação ponto a ponto das duas funções e o resultado é armazenado.

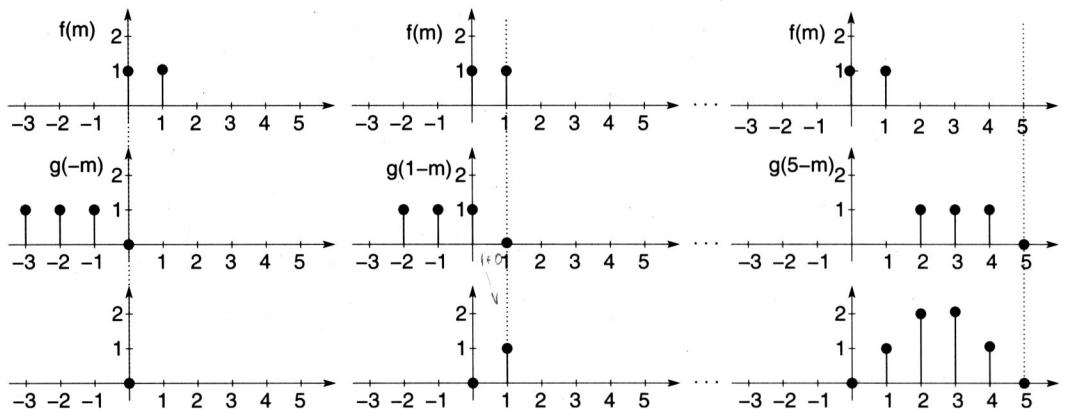


Figura 3.18: Execução da operação de convolução entre as funções f e g .

3.2.4 Transformada Rápida de Fourier

O cálculo dos coeficientes da transformada discreta de Fourier, se executado com o uso dos métodos mostrados na seção 3.1.3, apresenta alto custo computacional, mesmo para entradas com tamanhos moderados. Pelo fato de a DFT ser uma ferramenta útil na área de processamento de sinais e imagens, ao longo do tempo, tem-se desenvolvido algoritmos capazes de executá-la em tempo computacional reduzido.

Os algoritmos com propósito de calcular de maneira eficiente a transformada de Fourier são denominados transformada rápida de Fourier (FFT, do inglês, *Fast Fourier Transform*). Algumas abordagens têm sido utilizadas para criação desses algoritmos rápidos, tais como a baseada no mapeamento de índices, a qual permite que a DFT para uma função de entrada composta de N elementos seja calcula com $O(N \log_2 N)$ operações.

Esta seção descreve os algoritmos rápidos de Fourier baseados no mapeamento de índices, mais especificamente os métodos $\text{radix-}2^k$ baseados no conceito de decimação, processo que decompõe a computação total da DFT no cálculo de transformadas menores.

O processo de decimação decompõe a computação da transformada em cálculos de DFT sucessivamente menores, explorando a periodicidade da exponencial complexa $\exp(-i2kn\pi/N)$. Com essa decomposição, os resultados das subtransformadas são utilizados mais que uma vez, permitindo, dessa maneira, a redução no custo computacional requerido.

Devido ao compartilhamento dos coeficientes resultantes das DFT menores, a FFT normalmente é representada a partir de diagramas de blocos que indicam o relacionamento entre os valores intermediários. Tais diagramas são compostos de somadores, subtratores e multiplicadores, cuja representação é mostrada na figura 3.19.

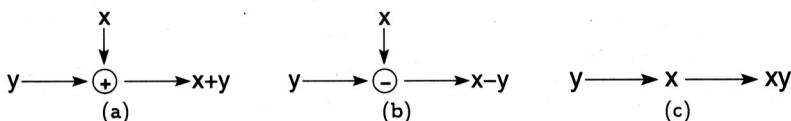


Figura 3.19: Componentes do diagrama de blocos. (a) somador para dois elementos de entrada, x e y ; (b) subtrator para as variáveis x e y ; (c) multiplicador para as variáveis x e y .

Algoritmos que decompõem o vetor de entrada x em vetores sucessivamente menores são denominados algoritmos de decimação no tempo (DIT, do inglês, *Decimation in Time*), enquanto aqueles que decompõem o vetor de saída y em vetores menores são denotados como algoritmos de decimação na freqüência (DIF, do inglês, *Decimation in Frequency*). Os diagramas da figura 3.20 ilustram os dois processos de decimação. No primeiro, o vetor de entrada é particionado em dois vetores, de modo a calcular a transformada para 8 componentes, posteriormente, os resultados são utilizados para o cômputo da transformada total, ou seja, para os 16 elementos de entrada.

O mapeamento de índices especifica uma relação entre os índices de um vetor $N = LK$ para os elementos de uma matriz que apresenta dimensão $L \times K$ (Burrus 1977, Sorensen et al. 1985). Tal relacionamento é obtido por meio da substituição dos índices de freqüência e tempo n e u da equação 3.18 pelos valores apresentados nas equações 3.40 e 3.41, em que $KL = N$ e as variáveis n_1, n_2, u_1 e u_2 denotam os novos índices da transformada, agora

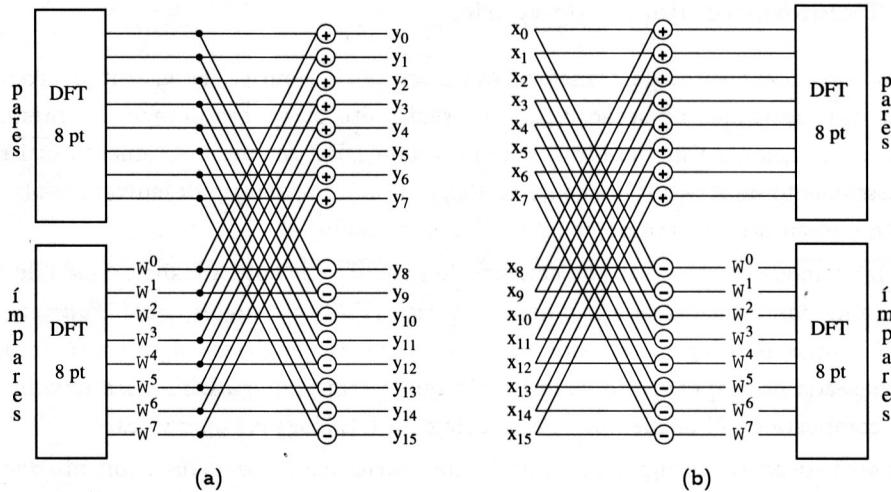


Figura 3.20: Métodos de decimação. (a) decimação no tempo; (b) decimação na freqüência.

apresentando-se na forma bidimensional.

$$n \equiv n_1 + L n_2 \quad (3.40)$$

$$u \equiv u_1 + K u_2 \quad (3.41)$$

A escolha dos valores de L e K nas equações 3.40 e 3.41 determinam se o algoritmo rápido de Fourier pertence à classe de decimação no tempo ou na freqüência. Por exemplo, um método para calcular a FFT que utiliza $L = 2$ e $K = N/2$ executa decimação no tempo pelo fato de particionar o vetor de entrada em dois novos vetores.

Dentre os algoritmos mais comuns na classe radix-2^k estão o *radix-2*, *radix-4* e *radix-8* (Bouguezel et al. 2004), conforme os valores escolhidos para L e K , tanto na decimação no tempo quanto na decimação na freqüência. A utilização conjunta de métodos radix-2^k origina os métodos *split-radix* (Duhamel 1986, Sorensen et al. 1986), os quais aplicam o *radix-2* e o *radix-4* nos componentes do vetor de entrada que apresentam índices pares e ímpares, respectivamente.

Para apresentar maior clareza no desenvolvimento das equações subsequentes, utiliza-se a definição mostrada na equação 3.42; dessa maneira, a DFT passa a apresentar a forma mostrada pela equação 3.43.

$$W = \exp(-2\pi i / N) \quad (3.42)$$

$$y_u = \frac{1}{N} \sum_{n=0}^{N-1} x_n W^{nu} \quad (3.43)$$

A aplicação do mapeamento de índices na equação 3.43 resulta na equação 3.44, a qual pode ser considerada uma transformada bidimensional. O termo N provém da igualdade $LK = N$, consequentemente $W^{N n_2 u_2} = 1$.

$$y_{u_1+Ku_2} = \frac{1}{N} \sum_{n_1=0}^{L-1} \sum_{n_2=0}^{K-1} x_{n_1+Ln_2} W^{(n_1+Ln_2)(u_1+Ku_2)} \quad (3.44)$$

$$\begin{aligned}
 &= \frac{1}{N} \sum_{n_1=0}^{L-1} \sum_{n_2=0}^{K-1} x_{n_1+Ln_2} W^{n_1 u_1} W^{Kn_1 u_2} W^{Ln_2 u_1} W^{Nn_2 u_2} \\
 &= \frac{1}{N} \sum_{n_1=0}^{L-1} \sum_{n_2=0}^{K-1} x_{n_1+Ln_2} W^{n_1 u_1} W^{Kn_1 u_2} W^{Ln_2 u_1}
 \end{aligned} \quad (3.44)$$

Exemplo 3.4 Algoritmo radix-4 com decimação no tempo. Com a utilização de $L = 4$ e $K = N/4$, obtém-se o algoritmo radix-4; substituindo os valores de L e K na equação 3.44 resulta em quatro transformadas para $N/4$ elementos cada, conforme mostrado na equação 3.45.

$$\begin{aligned}
 y_u = \frac{1}{N} \left(\sum_{n_2=0}^{N/4-1} x_{4n_2} W^{4n_2 u_1} + W^{u_1} W^{u_2 N/4} \sum_{n_2=0}^{N/4-1} x_{4n_2+1} W^{4n_2 u_1} + \right. \\
 \left. W^{2u_1} W^{2u_2 N/4} \sum_{n_2=0}^{N/4-1} x_{4n_2+2} W^{4n_2 u_1} + W^{3u_1} W^{3u_2 N/4} \sum_{n_2=0}^{N/4-1} x_{4n_2+3} W^{4n_2 u_1} \right) \quad (3.45)
 \end{aligned}$$

Em geral, os algoritmos rápidos para determinar a DFT radix- 2^k particionam sucessivamente os somatórios mostrados na equação 3.45 e utilizam valores calculados para somatórios menores visando à determinação dos coeficientes da transformada de maneira eficiente. ■

Com o intuito de expor os conceitos dos algoritmos FFT baseados no mapeamento de índices, será descrita a derivação do algoritmo clássico de decimação no tempo denominado *radix-2*, proposto por Cooley e Tukey (1965). Ao contrário do *radix-4*, que partitiona o vetor de entrada em quatro, o *radix-2* partitiona apenas em dois vetores, compostos pelos elementos com coeficientes pares e ímpares, respectivamente. Para efetuar o referido particionamento, substitui-se $L = 2$ e $K = N/2$ na equação 3.44, obtendo-se como resultado a equação 3.46.

$$\begin{aligned}
 y_{u_1+u_2 N/2} &= \frac{1}{N} \left\{ \sum_{n_2=0}^{N/2-1} x_{2n_2} W^{2n_2 u_1} + \sum_{n_2=0}^{N/2-1} x_{2n_2+1} W^{u_1} W^{Nu_2/2} W^{2n_2 u_1} \right\} \\
 &= \frac{1}{N} \left\{ \sum_{n_2=0}^{N/2-1} x_{2n_2} W^{2n_2 u_1} + W^{u_1+u_2 N/2} \sum_{n_2=0}^{N/2-1} x_{2n_2+1} W^{2n_2 u_1} \right\} \quad (3.46)
 \end{aligned}$$

Percebe-se que os somatórios da equação 3.46 são duas transformadas de Fourier, a primeira executada sobre os coeficientes pares e a segunda executada sobre os coeficientes ímpares. Utilizando como indexador de freqüências u em vez de $u_1 + u_2 N/2$ resulta na equação 3.47.

$$y_u = \frac{1}{N} \left\{ \sum_{n=0}^{N/2-1} x_{2n} W^{2n u} + W^u \sum_{n=0}^{N/2-1} x_{2n+1} W^{2n u} \right\} \quad (3.47)$$

Após a decomposição da DFT em dois somatórios, deve-se determinar os termos das transformadas até $N/2 - 1$ e a partir de $N/2$, ou seja, serão determinados os coeficientes y_u e $y_{u+N/2}$, para $u = 0, 1, \dots, N/2 - 1$. Considerando a divisão dos termos em dois conjuntos, para a segunda metade dos termos, aqueles com índice maior ou igual a $N/2$, a DFT resulta

na equação 3.48.

$$y_{u+N/2} = \frac{1}{N} \left\{ \sum_{n=0}^{N/2-1} x_{2n} W^{2n(u+N/2)} + W^{(u+N/2)} \sum_{n=0}^{N/2-1} x_{2n+1} W^{2n(u+N/2)} \right\} \quad (3.48)$$

Desenvolvendo os termos da equação 3.48, obtém-se

$$\begin{aligned} y_{u+N/2} = & \frac{1}{N} \left\{ \sum_{n=0}^{N/2-1} x_{2n} \exp(-4\pi i u/N) \exp(-2\pi i n) + \right. \\ & \left. + \exp(-2\pi i u/N) \exp(-\pi i) \sum_{n=0}^{N/2-1} x_{2n+1} \exp(-4\pi i u/N) \exp(-2\pi i n) \right\} \end{aligned} \quad (3.49)$$

A aplicação da relação $\exp(i\theta) = \cos(\theta) + i \sin(\theta)$ na equação 3.49 resulta em

$$\begin{aligned} y_{u+N/2} = & \frac{1}{N} \left\{ \sum_{n=0}^{N/2-1} x_{2n} W^{2nu} (\cos(2\pi n) + i \sin(-2\pi n)) + \right. \\ & \left. W^u (\cos(\pi) + i \sin(\pi)) \sum_{n=0}^{N/2-1} x_{2n+1} W^{2nu} (\cos(2\pi n) + i \sin(-2\pi n)) \right\} \end{aligned} \quad (3.50)$$

Finalmente, obtém-se a equação 3.51 para o cálculo da transformada para os termos com índices maiores ou igual a $N/2$.

$$y_{u+N/2} = \frac{1}{N} \left\{ \sum_{n=0}^{N/2-1} x_{2n} W^{2nu} - W^u \sum_{n=0}^{N/2-1} x_{2n+1} W^{2nu} \right\} \quad (3.51)$$

Uma notação compacta para os somatórios compostos pelos termos pares e ímpares é definida pelas equações 3.52 e 3.53, respectivamente. Nota-se que o dígito acrescentado ao expoente de H representa o índice dos coeficientes da função a serem calculados em cada somatório. Por exemplo, H_u^0 representa que os coeficientes de x_n utilizados por esse somatório serão os que apresentam índices pares ou possuem a terminação 0 quando representados na base binária.

$$H_u^0 = \sum_{n=0}^{N/2-1} x_{2n} W^{2nu} \quad (3.52)$$

$$H_u^1 = \sum_{n=0}^{N/2-1} x_{2n+1} W^{2nu} \quad (3.53)$$

Utilizando o resultado obtido na equação 3.51 e a notação apresentada na equação 3.52, tem-se que a transformada discreta de Fourier pode ser escrita em duas partes, conforme mostrado nas equações 3.54 e 3.55, onde $u = 0, 1, \dots, N/2-1$. Percebe-se que a representação da transformada apresenta redundância; por exemplo, os valores de H_u^0 e H_u^1 aparecem nas duas equações. Dessa maneira, pode-se obter uma redução no número de operações se estas

forem calculadas apenas uma vez.

$$y_u = \frac{1}{N} \{ H_u^0 + W^u H_u^1 \} \quad (3.54)$$

$$y_{u+N/2} = \frac{1}{N} \{ H_u^0 - W^u H_u^1 \} \quad (3.55)$$

Exemplo 3.5 Cálculo da transformada rápida de Fourier para uma função de entrada composta por quatro elementos $x = [x_0, x_1, x_2, x_3]$. Utilizando a equação 3.54, tem-se que

$$y_0 = \frac{1}{N} \{ H_0^0 + W^0 H_0^1 \} \quad (3.56)$$

$$y_1 = \frac{1}{N} \{ H_1^0 + W^1 H_1^1 \} \quad (3.57)$$

$$y_2 = \frac{1}{N} \{ H_0^0 - W^0 H_0^1 \} \quad (3.58)$$

$$y_3 = \frac{1}{N} \{ H_1^0 - W^1 H_1^1 \} \quad (3.59)$$

Percebe-se a redundância nos dados calculados, os coeficientes H_0^0 , H_0^1 , H_1^0 , e H_1^1 podem ser calculados apenas uma vez e utilizados posteriormente para determinar os valores de y_n , conforme mostra o diagrama da figura 3.21, em que as transformadas são calculadas para os coeficientes com índices pares e ímpares, respectivamente. A determinação dos coeficientes y_n depende da avaliação dos valores dos somatórios H , que podem ser obtidos recursivamente, repetindo o mesmo processo descrito para a decomposição da transformada de Fourier original (equação 3.47).

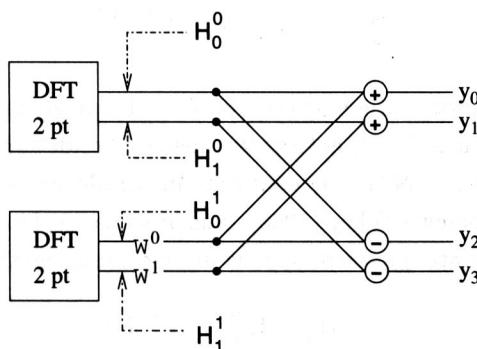


Figura 3.21: Diagrama para o cálculo dos coeficientes de Fourier, evitando o cálculo repetido dos valores de H_0^0 , H_1^0 , H_0^1 e H_1^1 .

A partir do diagrama da figura 3.21, tem-se que $y_4 = H_0^0 - W^0 H_0^1$, o que equivale à execução da DFT para quatro elementos com índices pares (equação 3.52), subtraindo o resultado da execução da DFT para os elementos com coeficientes ímpares (equação 3.53) multiplicado pela constante W^0 . ■

Com o objetivo de determinar H_u^0 e H_u^1 , efetua-se novamente a divisão de cada um desses somatórios em dois novos somatórios, agora, cada um composto de $N/4$ elementos. As equações 3.60 e 3.61 apresentam essa decomposição para H_u^0 e H_u^1 , respectivamente.

$$\begin{aligned}
 H_u^0 &= \sum_{n=0}^{N/4-1} x_{4n} W^{4nu} + \sum_{n=0}^{N/4-1} x_{2(2n+1)} W^{2(2n+1)u} \\
 &= \sum_{n=0}^{N/4-1} x_{4n} W^{4nu} + W^{2u} \sum_{n=0}^{N/4-1} x_{4n+2} W^{4nu}
 \end{aligned} \tag{3.60}$$

$$\begin{aligned}
 H_u^1 &= \sum_{n=0}^{N/4-1} x_{2(2n)+1} W^{2(2n)u} + \sum_{n=0}^{N/4-1} x_{2(2n+1)+1} W^{2(2n+1)u} \\
 &= \sum_{n=0}^{N/4-1} x_{4n+1} + W^{2u} \sum_{n=0}^{N/4-1} x_{4n+3} W^{4nu}
 \end{aligned} \tag{3.61}$$

Novamente, para se obter uma representação compacta, define-se H_u^{00} , H_u^{10} , H_u^{01} e H_u^{11} como mostrado nas equações 3.62 a 3.65. Percebe-se que o índice de x_n , quando representado na base binária, apresenta a terminação mostrada no expoente de H .

$$H_u^{00} = \sum_{n=0}^{N/4-1} x_{4n} W^{4nu} \tag{3.62}$$

$$H_u^{10} = \sum_{n=0}^{N/4-1} x_{4n+2} W^{4nu} \tag{3.63}$$

$$H_u^{01} = \sum_{n=0}^{N/4-1} x_{4n+1} W^{4nu} \tag{3.64}$$

$$H_u^{11} = \sum_{n=0}^{N/4-1} x_{4n+3} W^{4nu} \tag{3.65}$$

Os somatórios mostrados nas equações 3.60 e 3.61 apresentam a mesma propriedade obtida para $y_{u+N/2}$ (equação 3.51), entretanto, nesse caso, determina-se H_u^0 , $H_{u+N/4}^0$, H_u^1 e $H_{u+N/4}^1$, para $u = 0, 1, \dots, N/4 - 1$, conforme mostrado nas equações 3.66 a 3.69. No algoritmo proposto por Cooley e Tukey (1965), esse processo de decomposição dos somatórios é efetuado recursivamente até que reste apenas um elemento dentro do somatório.

$$H_u^0 = H_u^{00} + W^{2u} H_u^{10} \tag{3.66}$$

$$H_{u+N/4}^0 = H_u^{00} - W^{2u} H_u^{10} \tag{3.67}$$

$$H_u^1 = H_u^{01} + W^{2u} H_u^{11} \tag{3.68}$$

$$H_{u+N/4}^1 = H_u^{01} - W^{2u} H_u^{11} \tag{3.69}$$

Exemplo 3.6 Continuação do exemplo 3.5. Para determinar os quatro coeficientes de Fourier, deve-se calcular os somatórios H_0^0 , H_1^0 , H_0^1 e H_1^1 . Utilizando os resultados anteriores, obtém-se

$$H_0^0 = H_0^{00} + W^0 H_0^{10}, \tag{3.70}$$

$$H_1^0 = H_0^{00} - W^0 H_0^{10}, \tag{3.71}$$

$$H_0^1 = H_0^{01} + W^0 H_0^{11} \text{ e } H_1^1 = H_0^{01} - W^0 H_0^{11} \tag{3.72}$$

$$H_1^1 = H_0^{01} - W^0 H_0^{11}. \quad (3.73)$$

A representação por meio de diagrama é mostrada pela figura 3.22.

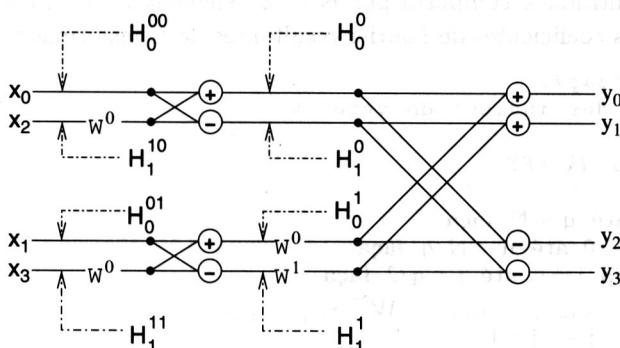


Figura 3.22: Diagrama para o cálculo dos coeficientes de Fourier para um vetor de entrada composto de 4 elementos.

Conforme descrito até então, o algoritmo de Cooley e Tukey calcula a DFT por meio da decomposição do vetor de entrada em subvetores. Essa decomposição permite que os resultados previamente obtidos sejam utilizados, proporcionando a redução na quantidade de computação requerida. Para permitir as decomposições recursivas dos somatórios, esse algoritmo deve receber como entrada uma função composta por $N = 2^R$ elementos (algoritmo 3.5).

A primeira linha do algoritmo chama uma rotina para permutar os índices dos coeficientes da função de entrada x . Isso pode ser feito com a utilização do algoritmo de Buneman (algoritmo 3.6), que obtém iterativamente um vetor com os índices dos coeficientes que devem ser utilizados como entrada para FFT. A cada ciclo, os elementos do vetor, com M componentes, são dobrados e acrescidos de 1, o resultado é concatenado, gerando um vetor composto de $2M$ elementos.

A necessidade da permutação dos índices pode ser percebida pelos índices $4n$, $4n + 2$, $4n + 1$ e $4n + 3$ utilizados pelas equações 3.62 a 3.65. Para uma DFT composta de oito pontos, a ordem de entrada dos coeficientes do vetor x é 0, 4, 2, 6, 1, 5, 3, 7, exatamente a permutação obtida como resultado da execução de dois ciclos do algoritmo 3.6, conforme mostrado na figura 3.23.

Em processamento de imagens, normalmente utiliza-se a DFT bidimensional, no entanto, se calculada utilizando a equação 3.22, seu custo computacional é extremamente alto, $O(N^4)$ para uma matriz de entrada composta por N linhas e N colunas. Com o intuito de reduzir esse custo, diversos métodos têm sido propostos, tais como o método *linha-coluna*, *split-radix 2D* (Bi e Chen 1997) e *vector split-radix* (Pei e Wu 1987, Sundararajan e Ahmad 1996, Wu e Paolini 1989).

O método linha-coluna, para determinar de maneira eficiente a DFT, baseia-se na aplicação da FFT unidimensional sobre as linhas da matriz de entrada, armazenando o resultado em uma matriz intermediária e, em seguida, calcula-se a FFT sobre as colunas da matriz resultante. A equação 3.74 mostra a decomposição do núcleo da DFT que divide

Transformada rápida de Fourier (FFT)

Algoritmo de decimação no tempo radix-2, proposto por Cooley e Tukey (1965), para uma função de entrada x composta por $N = 2^R$ elementos. Ao final da execução, o vetor x conterá os coeficientes de Fourier resultantes da transformada.

```

1 // inicialização
2 permutação dos índices do vetor x
3
4 // execução da FFT
5 e ← lg N - 1
6 para q = 2 até q ≤ N faça
7   para n = 0 até n < N/q faça
8     para j = 0 até j < q/2 faça
9        $x_{nq+j} \leftarrow x_{nq+j} + W^{2^e j} x_{nq+j+q/2}$ 
10      j ← j + 1
11    k ← 0
12    para j = q/2 até j < q faça
13       $x_{nq+j} \leftarrow x_{nq+k} - W^{2^e k} x_{nq+k+q/2}$ 
14      k ← k + 1
15      j ← j + 1
16    n ← n + 1
17  q ← 2q
18  e ← e - 1

```

Algoritmo 3.5: Transformada rápida de Fourier (FFT).

Permutação dos índices de um vetor

Algoritmo de Buneman utilizado para obter a permutação necessária para os índices de um vetor v composto por N componentes. Algoritmos mais eficientes utilizados para permutação dos índices de vetores são propostos por Evans (1987, 1989).

```

1 // inicialização
2 v0 ← 0
3 v1 ← 1
4
5 // permutação dos elementos do vetor de índices
6 para q = 2 até q < N faça
7   para i = 0 até i < q faça
8     vi ← 2vi
9     vi+q ← vi + 1
10    i ← i + 1
11  q ← 2q

```

Algoritmo 3.6: Permutação dos índices de um vetor.

sua execução em duas etapas, similar àquela utilizada para demonstrar a propriedade da separabilidade do núcleo da DFT (seção 3.2.3).

$$y_{u,v} = \frac{1}{N^2} \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} x_{n,m} \exp(-2\pi i(nv + mu)/N)$$

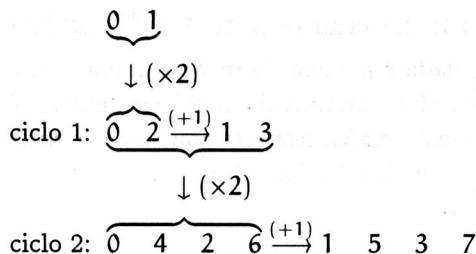


Figura 3.23: Obtenção do vetor de permutação dos índices para execução da FFT para oito pontos. Os elementos do vetor de entrada são reordenados conforme o resultado da execução da permutação dos índices.

$$= \underbrace{\frac{1}{N} \sum_{n=0}^{N-1} \exp(-2\pi i nv/N)}_{\text{DFT colunas}} \underbrace{\frac{1}{N} \sum_{m=0}^{N-1} x_{n,m} \exp(-2\pi imu/N)}_{\text{DFT linhas}} \quad (3.74)$$

O diagrama da figura 3.24 ilustra o resultado da aplicação da FFT sobre as linhas e colunas, o resultado intermediário é armazenado em uma matriz de ordem N. Após a execução da transformada rápida sobre as colunas da matriz intermediária são obtidos os coeficientes da DFT bidimensional. O algoritmo 3.7 descreve as etapas utilizadas pelo método linha-coluna.

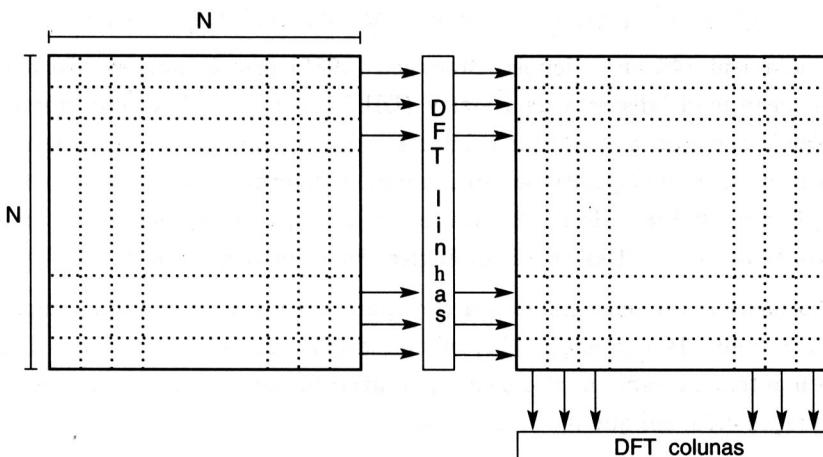


Figura 3.24: Diagrama do método linha-coluna para calcular a DFT bidimensional.

A determinação da transformada bidimensional utilizando o método mostrado pelo algoritmo 3.7 apresenta ordem de complexidade $O(N^2 \log_2 N)$, $O(N)$ para executar a FFT unidimensional sobre as linhas ou colunas e $O(N \log_2 N)$ para cada FFT. Entretanto, esse método requer computação adicional para as operações de transposição e memória extra para o armazenamento de N coeficientes complexos.

Método linha-coluna para a FFT bidimensional

FFT bidimensional calculada por meio do método linha-coluna. A matriz L , $N \times N$, armazena os resultados da FFT unidimensional sobre as linhas, as matrizes X e Y contêm a entrada e a saída da transformada, respectivamente. L_i representa a i -ésima linha da matriz L e a função $\text{FFT1D}(M, i)$ aplica a FFT unidimensional sobre os elementos da i -ésima linha da matriz M .

```

1 // FFT unidimensional sobre as linhas
2 para n = 0 até n < N faça
3    $L_n \leftarrow \text{FFT1D}(X, n)$ 
4
5 // FFT unidimensional sobre as colunas
6 para n = 0 até n < N faça
7    $Y_n \leftarrow \text{FFT1D}(L^T, n)$ 
8
9 // transposição do resultado
10  $Y \leftarrow Y^T$ 
```

Algoritmo 3.7: Método linha-coluna para a FFT bidimensional.

3.3 Transformada Discreta de Hartley

A transformada discreta de Hartley, proposta inicialmente no trabalho de Hartley (1942), possui propriedades semelhantes àquelas da transformada de Fourier. No entanto, necessita apenas de computação real para determinar os coeficientes da transformada e, devido a sua estrutura, o seu núcleo não precisa ser alterado para que se obtenha a transformada inversa.

Aproximadamente 40 anos depois, Bracewell (1983) propôs uma versão discreta, denominada transformada discreta de Hartley (DHT, do inglês, *Discrete Hartley Transform*). A DHT começou a despertar o interesse da comunidade de processamento de sinal apóis a proposição de algoritmos rápidos para a determinação de seus coeficientes (Bi et al. 1998, Bracewell 1983, Hou 1987, Liu et al. 2003), tais algoritmos são chamados de transformada rápida de Hartley (FHT, do inglês, *Fast Hartley Transform*).

A transformada unidimensional de Hartley, para um vetor de entrada composto por N elementos, é apresentada na equação 3.75, em que $\text{cas}(x) = \cos(x) + \sin(x)$ e $u = 1, 2, \dots, N-1$. Sua inversa mantém o mesmo núcleo e efetua a inversão dos coeficientes y_u e x_n , conforme mostra a equação 3.76, em que $n = 1, 2, \dots, N-1$.

$$y_u = \frac{1}{N} \sum_{n=0}^{N-1} x_n \text{cas}(2\pi un/N) \quad (3.75)$$

$$x_n = \sum_{u=0}^{N-1} y_u \text{cas}(2\pi un/N) \quad (3.76)$$

Na versão bidimensional, a transformada de Hartley apresenta a forma mostrada na equação 3.77, onde $u = 0, 1, \dots, N-1$ e $v = 0, 1, \dots, N-1$. Essa transformada não apresenta o núcleo separável como a DFT bidimensional, por esse motivo, muitas vezes utiliza-se a

formulação apresentada na equação 3.78, denominada transformada *cas-cas*.

$$y_{u,v} = \frac{1}{N^2} \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} x_{m,n} \text{cas}(2\pi(mu + nv)/N) \quad (3.77)$$

$$y_{u,v} = \frac{1}{N^2} \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} x_{m,n} \text{cas}(2\pi mu/N) \text{cas}(2\pi nv/N) \quad (3.78)$$

Na figura 3.25 é mostrada a representação gráfica da base da DHT bidimensional, para uma matriz de entrada com ordem 8. Pode-se perceber pela figura 3.25 que na região superior esquerda são detectadas as freqüências baixas, pela ausência de variações, enquanto na região inferior direita são detectadas as freqüências altas.

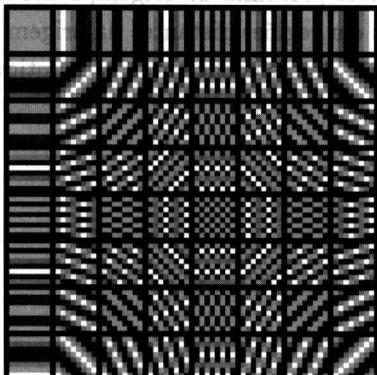


Figura 3.25: Representação gráfica do núcleo da transformada discreta de Hartley (DHT) para uma matriz de entrada com ordem 8.

As mesmas classes de algoritmos rápidos aplicados à DFT são utilizadas para DHT. Tais algoritmos são aplicados à transformada unidimensional (Evans 2004, Sorensen et al. 1985), bidimensional (Grigoryan e Agaian 2000, Shunji et al. 1991) e alguns em transformadas de ordem superior (Bortfeld e Dinter 1996, Boussakta et al. 2001).

3.4 Transformada Discreta do Cosseno

Desde que foi apresentada por Ahmed et al. (1974), a transformada discreta do cosseno (DCT, do inglês, *Discrete Cosine Transform*) tem sido amplamente explorada pela comunidade de processamento de imagens, principalmente nas áreas de compressão, filtragem e extração de características. Esta seção descreve as principais propriedades da DCT e apresenta brevemente as principais classes de algoritmos utilizados para o seu cálculo.

O grande interesse pela DCT está no fato de ela apresentar a propriedade de compactação da energia em regiões próximas da origem, aproximando-se do desempenho obtido pela transformada de Karhunen-Loève, descrita na seção 3.7. Com a compactação da energia, após a execução da transformada, muitos coeficientes apresentarão valores próximos a zero e, portanto, podem ser eliminados sem que haja perda significativa de informações contidas na imagem.

O conceito de energia está relacionado com o tamanho dos vetores de entrada e saída da transformada, sua definição é mostrada na equação 3.79. Diz-se que uma transformada

conserva a energia quando $\|x\|^2 = \|y\|^2$, para uma transformação da forma $y^T = Ax^T$. Com a DCT, os coeficientes que apresentam valores mais significativos se localizam nas áreas próximas da origem, facilitando, dessa maneira, a determinação das regiões que podem ser excluídas quando o objetivo é a compressão de dados.

$$\|v\|^2 = \sum_{k=0}^{N-1} |v_k|^2 \quad (3.79)$$

A propriedade de compactação de energia é ilustrada na figura 3.26. Quando a função de entrada apresenta valores altamente correlacionados, os coeficientes com valores significativos concentram-se em regiões próximas da origem, como mostra a figura 3.26(b). Dado que os coeficientes localizados em regiões distantes da origem apresentam valores próximos a zero, mesmo com a eliminação desses coeficientes, a quantidade de energia não será significativamente reduzida.

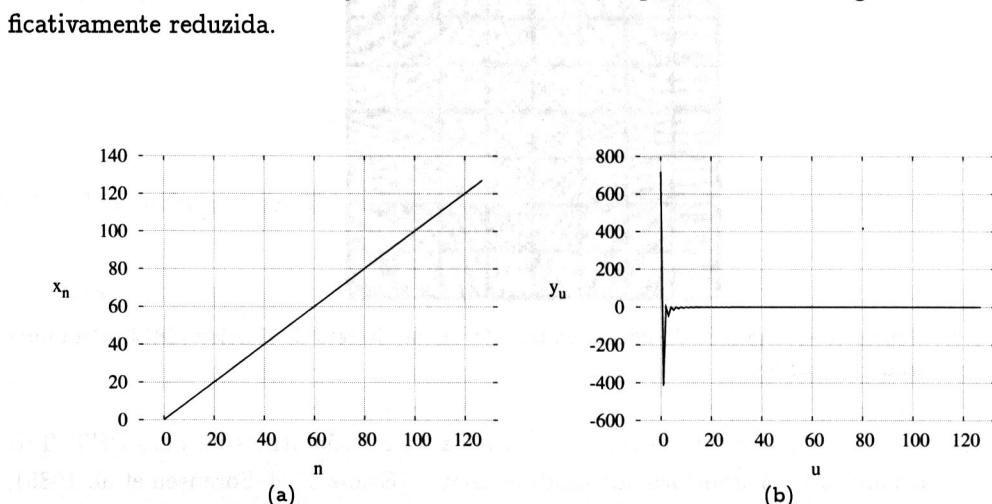


Figura 3.26: Aplicação da transformada do cosseno unidimensional sobre vetores com 128 componentes. (a) função de entrada apresentando alta correlação; (b) resultado da transformada, com grande concentração de energia em regiões próximas à origem.

Ahmed et al. (1974) derivaram o núcleo da transformada unidimensional do cosseno. A equação 3.80 mostra o núcleo para uma função de entrada composta por N elementos, em que $n, u = 0, 1, \dots, N - 1$ e $k(\alpha) = 1/\sqrt{2}$ se $\alpha = 0$. A representação gráfica do núcleo é mostrada na figura 3.27, em que cada plano representa o u -ésimo vetor da base.

$$b_{u,n} = k(n) \sqrt{\frac{2}{N}} \cos \left(\frac{\pi n (u + \frac{1}{2})}{N} \right) \quad (3.80)$$

O núcleo da DCT apresentado por Ahmed et al. pertence a um conjunto de representações obtidas a partir dos autovetores extraídos de variações da matriz mostrada em 3.81, descrita no trabalho de Strang (1999). Após a instanciação dos valores marcados com o símbolo \otimes , são obtidas oito matrizes, a partir das quais calculam-se autovetores, utilizados

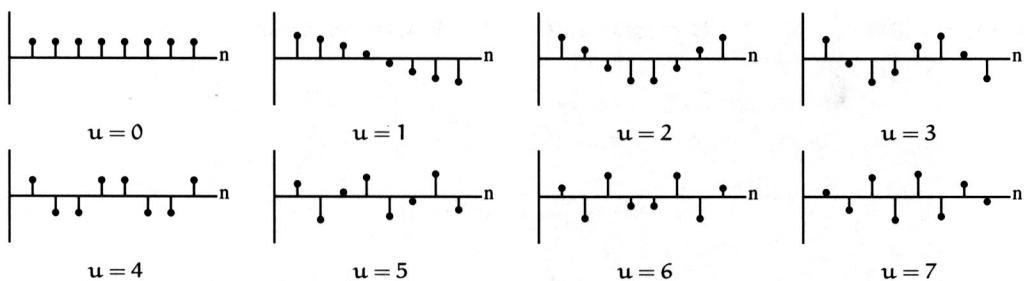


Figura 3.27: Representação gráfica do núcleo da transformada discreta do cosseno (DCT) para um vetor de entrada com oito componentes.

como núcleo para a transformada discreta do cosseno.

$$A = \begin{bmatrix} \otimes & \otimes & & & \\ -1 & 2 & -1 & & \\ & -1 & 2 & -1 & \\ & & \ddots & & \\ & & & -1 & 2 & -1 \\ & & & & \otimes & \otimes \end{bmatrix} \quad (3.81)$$

A partir da matriz mostrada em 3.82, obtém-se a DCT proposta por Ahmed et al., denominada DCT-2 e amplamente utilizada em processamento e análise de imagens. Outros três núcleos são apresentados nas equações 3.83 a 3.85, para o mesmo $k(\alpha)$ definido anteriormente. Uma discussão mais detalhada sobre a obtenção dessas matrizes e os respectivos núcleos pode ser encontrada nos trabalhos de Sanchez et al. (1995) e Strang (1999).

$$A = \begin{bmatrix} 1 & -1 & & & \\ -1 & 2 & -1 & & \\ & -1 & 2 & -1 & \\ & & \ddots & & \\ & & & -1 & 2 & -1 \\ & & & & -1 & 1 \end{bmatrix} \quad (3.82)$$

$$\text{DCT-1: } b_{u,n} = k(u)k(n)\sqrt{\frac{2}{N}} \cos\left(\frac{\pi n u}{N}\right) \quad (3.83)$$

$$\text{DCT-3: } b_{u,n} = k(u)\sqrt{\frac{2}{N}} \cos\left(\frac{\pi u(n + \frac{1}{2})}{N}\right) \quad (3.84)$$

$$\text{DCT-4: } b_{u,n} = \sqrt{\frac{2}{N}} \cos\left(\frac{\pi(n + \frac{1}{2})(u + \frac{1}{2})}{N}\right) \quad (3.85)$$

A transformada unidimensional do cosseno e sua inversa, para um vetor de entrada composto por N elementos, são mostradas nas equações 3.86 e 3.87, respectivamente, em

que $n, u = 0, 1, \dots, N - 1$ e B representa o núcleo da transformada.

$$y_u = \sum_{n=0}^{N-1} x_n b_{u,n} \quad (3.86)$$

$$x_n = \sum_{u=0}^{N-1} y_u b_{n,u} \quad (3.87)$$

Diversos algoritmos rápidos para determinar a DCT, denominados FDCT (do inglês, *Fast Discrete Cosine Transform*), têm sido propostos na literatura. Estes utilizam duas abordagens principais, computação indireta baseada na transformada rápida de Fourier (Chan e Ho 1992, Makhoul 1980, Narashimha e Peterson 1978) e computação direta por meio da decomposição do núcleo da DCT (Cvetkovic e Popovic 1992, Yip e Rao 1984).

Desde que a DCT é uma transformada separável, sua versão bidimensional pode ser calculada a partir de aplicações sucessivas da sua forma unidimensional, conforme o diagrama mostrado na figura 3.24. Primeiro, calcula-se para as linhas, armazenando o resultado em uma matriz temporária, em seguida, determina-se os coeficientes da DCT para as colunas dessa matriz, obtendo-se o resultado da transformada.

A figura 3.28 ilustra a aplicação da DCT bidimensional em uma imagem partitionada em blocos compostos de 64×64 pixels. As figuras 3.28(d) e 3.28(f) apresentam os resultados da transformada inversa quando se mantêm apenas os coeficientes mais significativos, ou seja, aqueles mais próximos da origem, conforme mostrado nas figuras 3.28(c) e 3.28(e). Devido à eliminação das freqüências altas, percebe-se, principalmente na figura 3.28(d), a suavização nos contornos presentes na imagem.

3.5 Transformada Discreta do Seno

A transformada discreta do seno (DST, do inglês, *Discrete Sine Transform*), assim como a DCT, apresenta propriedades como a compactação de energia e separabilidade. Para um vetor de entrada composto de N elementos, seu núcleo pode ser obtido pela equação 3.88, em que $n, u = 0, 1, \dots, N - 1$. A representação gráfica do seu núcleo é mostrada na figura 3.29.

$$b_{u,n} = \sqrt{\frac{2}{N+1}} \operatorname{sen}\left(\frac{\pi(u+1)(n+1)}{N+1}\right) \quad (3.88)$$

A transformada unidimensional do seno para um vetor de entrada composto por N elementos e sua inversa são mostradas pelas equações 3.89 e 3.90, respectivamente, em que $u, n = 0, 1, \dots, N - 1$.

$$y_u = \sqrt{\frac{2}{N+1}} \sum_{n=0}^{N-1} x_n \operatorname{sen}\left(\frac{\pi(n+1)(u+1)}{N+1}\right) \quad (3.89)$$

$$x_n = \sqrt{\frac{2}{N+1}} \sum_{u=0}^{N-1} y_u \operatorname{sen}\left(\frac{\pi(n+1)(u+1)}{N+1}\right) \quad (3.90)$$

No caso bidimensional, a transformada do seno e sua inversa são definidas pelas equações 3.91 e 3.92, respectivamente, para uma matriz de entrada com ordem N . A figura 3.30 mostra a representação gráfica do núcleo da DST para uma matriz de entrada

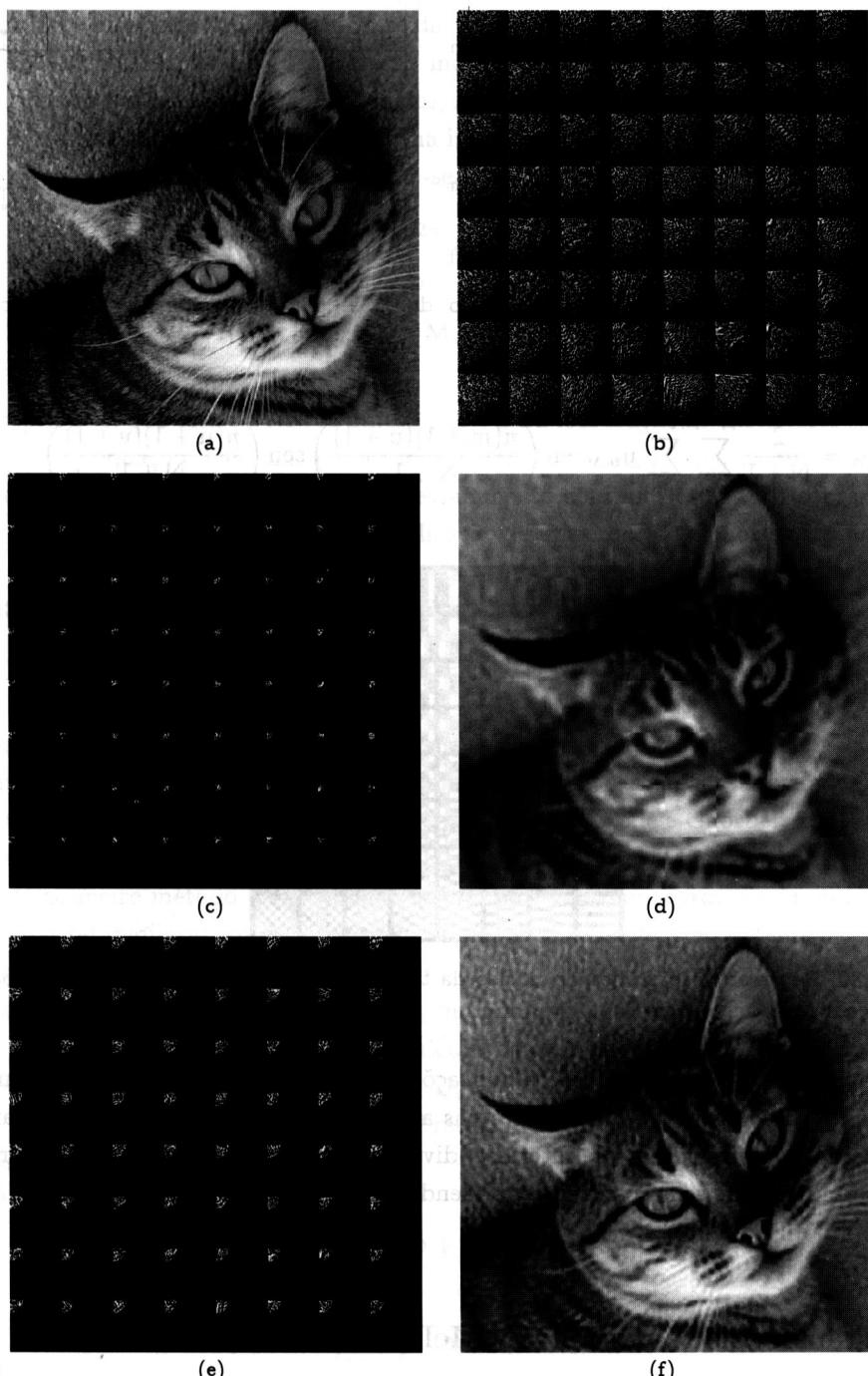


Figura 3.28: Aplicação da DCT em blocos compostos de 64×64 pixels. (a) imagem original; (b) resultado da DCT; (c) espectro mantendo apenas os coeficiente em um raio de 8 pixels da origem; (d) transformada inversa de (c); (e) espectro mantendo os coeficientes no raio de 16 pixels da origem; (f) transformada inversa de (e).

com 8×8 elementos.

$$y_{u,v} = \frac{2}{N+1} \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} x_{m,n} \operatorname{sen}\left(\frac{\pi(m+1)(u+1)}{N+1}\right) \operatorname{sen}\left(\frac{\pi(n+1)(v+1)}{N+1}\right) \quad (3.91)$$

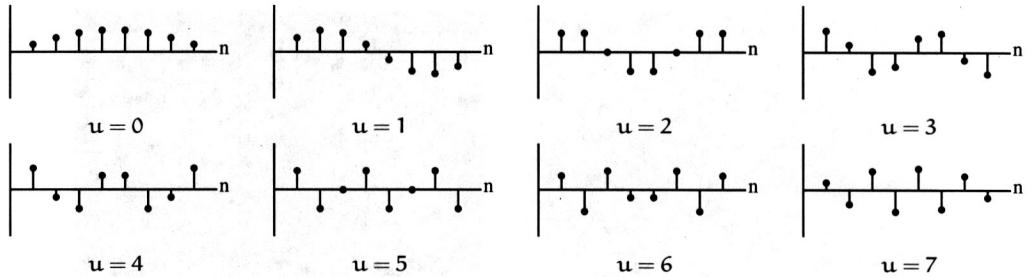


Figura 3.29: Representação gráfica do núcleo da DST para um vetor de entrada com oito componentes.

$$x_{m,n} = \frac{2}{N+1} \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} y_{u,v} \sin\left(\frac{\pi(m+1)(u+1)}{N+1}\right) \sin\left(\frac{\pi(n+1)(v+1)}{N+1}\right) \quad (3.92)$$

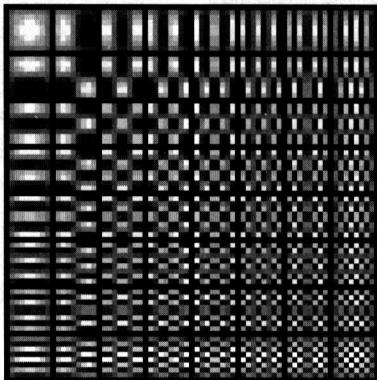


Figura 3.30: Representação gráfica do núcleo da transformada discreta do seno (DST) para uma matriz de entrada com ordem 8.

A transformada do seno apresenta aplicações em processamento e análise de imagens e sinais, como na codificação de imagens e nas aproximações da transformada de Karhunen-Loève (seção 3.7). Devido a esse interesse, diversos algoritmos rápidos têm sido propostos para determinação de seus coeficientes, podendo ser encontrados nos trabalhos de Liu et al. (2000) e Yip e Rao (1980).

3.6 Transformada de Fourier-Mellin

No final da década de 1970, a transformada de Fourier-Mellin (FMT, do inglês, *Fourier-Mellin Transform*) começou a ser utilizada pela comunidade de análise de imagens com os trabalhos de Casasent e Psaltis (1976) e Yatagay et al. (1981). Vários descritores baseados na FMT invariantes à rotação e à escala foram criados, porém, a inexistência de uma versão discreta para FMT tornou-se um problema, pois deve-se estimar a integral mostrada na equação 3.93, em que $f(r, \theta)$ e $M_f(k, v)$ contêm as funções de entrada e saída, respectivamente.

$$M_f(k, v) = \frac{1}{2\pi} \int_0^\infty \int_0^{2\pi} f(r, \theta) r^{-iv} e^{-ik\theta} d\theta \frac{dr}{r} \quad (3.93)$$

A integral mostrada no lado direito da equação 3.93 apresenta problemas quanto à convergência (Ghorbel 1994); por esse motivo, passou-se a utilizar a transformada analítica de Fourier-Mellin (AFMT, do inglês, *Analytical Fourier-Mellin Transform*). As equações 3.94 e 3.95 mostram a AFMT e sua inversa, respectivamente. Nesse caso, em vez de se calcular a integral para $f(r, \theta)$, calcula-se para $f_\sigma(r, \theta) = r^\sigma f(r, \theta)$, em que $\sigma > 0$.

$$M_{f_\sigma}(k, v) = \frac{1}{2\pi} \int_0^\infty \int_0^{2\pi} f(r, \theta) r^{\sigma - iv} e^{-ik\theta} d\theta \frac{dr}{r} \quad (3.94)$$

$$f(r, \theta) = \int_{-\infty}^{+\infty} \sum_{k \in \mathbb{Z}} M_{f_\sigma}(k, v) r^{-\sigma + iv} e^{ik\theta} dv \quad (3.95)$$

A propriedade que torna a AFMT interessante para elaboração de descritores de objetos é mostrada na equação 3.96, em que a função $g(r, \theta) = f(\alpha r, \theta + \beta)$, a qual representa a aplicação de uma rotação com ângulo θ e escalamento com um fator de β sobre f , representada em coordenadas polares. Após a aplicação de um processo de normalização, podem ser extraídos descritores invariantes, diferentemente da transformada de Fourier, que não apresenta invariância a escalamentos e rotações.

$$M_{g_\sigma}(k, v) = \alpha^{-\sigma + iv} e^{ik\beta} M_{f_\sigma}(k, v) \quad (3.96)$$

Apesar da utilização da AFMT resolver os problemas quanto à convergência, ainda há o problema da inexistência de uma versão discreta para determinação dos coeficientes da transformada. Para permitir a utilização de imagens digitais, Derrode e Ghorbel (2001) propõem algumas aproximações discretas. O restante desta seção descreve duas aproximações para a AFMT. O primeiro método utiliza como entrada uma imagem representada no sistema de coordenadas retangulares, enquanto o segundo aplica-se a imagens no sistema de coordenadas polares.

A aproximação da AFMT e sua inversa, quando aplicadas a imagens representadas no sistema de coordenadas retangulares, são calculadas com as equações 3.97 e 3.98, respectivamente. Na primeira equação, P_{\min} , P_{\max} , Q_{\min} e Q_{\max} representam as coordenadas, em relação ao centróide da imagem, do menor retângulo que contém totalmente o objeto de interesse.

$$y_{k,v} = \frac{1}{2\pi} \sum_{q=Q_{\min}}^{Q_{\max}} \sum_{p=P_{\min}}^{P_{\max}} x_{p,q} (p + iq)^{-k} (p^2 + q^2)^{(k-2+\sigma-iv)/2} \quad (3.97)$$

$$x_{p,q} = (p^2 + q^2)^{-\sigma/2} \sum_{v=-V}^V \sum_{k=-K}^K y_{k,v} (p + iq)^k (p^2 + q^2)^{(-k+iv)/2} \quad (3.98)$$

Para as duas equações anteriores, $k \in [-K, K]$ e $v \in [-V, V]$, ambos pertencentes ao conjunto dos inteiros. Considerando esses intervalos, os coeficientes resultantes da aplicação da transformada pertencem à matriz Y , que apresenta dimensão $2K + 1 \times 2V + 1$ e, segundo Derrode e Ghorbel, quanto maior o valor de K e V melhor será a reconstrução da imagem quando aplicada a transformada inversa, ou seja, a equação 3.98.

Antes que a segunda aproximação da AFMT seja efetuada, deve-se reamostrar a imagem de entrada, alterando sua representação para o sistema de coordenadas polares. A figura 3.31 exemplifica essas representações. Um aspecto interessante presente na conversão

do sistema de coordenadas está no fato de que uma rotação em coordenadas retangulares se transforma em uma translação quando se utiliza coordenadas polares, conforme ilustrado na figura 3.31(d).

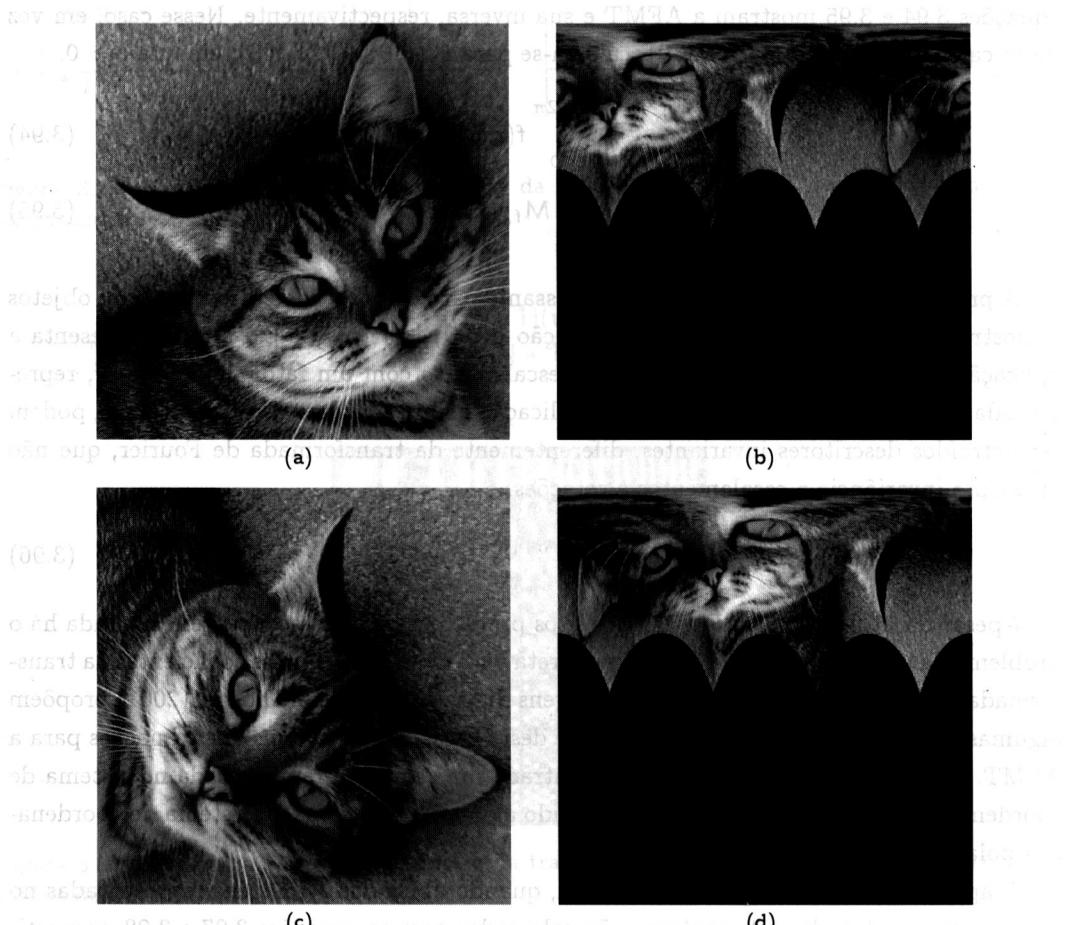


Figura 3.31: Alteração da representação de uma imagem. (a) representação no sistema de coordenadas retangulares; (b) representação no sistema de coordenadas polares; (c) imagem original após rotação de 90° no sentido horário; (d) representação no sistema de coordenadas polares para a imagem após rotação.

As figuras 3.32(a) e (b) mostram a disposição dos pixels de uma imagem nas representações retangular e polar, respectivamente. Para se efetuar a conversão de um sistema de coordenadas para outro, os pixels devem ser reamostrados conforme a grade estabelecida, e os tons de cinza devem ser interpolados.

A grade da amostragem polar é construída pela intersecção entre M círculos concêntricos com N linhas radiais originárias no centro da imagem. Os intervalos de amostragem angular e radial são mostrados pela equação 3.99, em que R denota o raio da menor circunferência capaz de conter toda a imagem quando representada no sistema de coordenadas retangulares.

$$\Delta\theta = \frac{2\pi}{N} \quad \Delta\rho = \frac{R}{M} \quad (3.99)$$

A cada ponto presente na intersecção das linhas radiais e os círculos concêntricos do sistema de coordenadas polares, ou seja, (ρ_m, θ_n) , deve ser amostrado a partir da imagem repre-

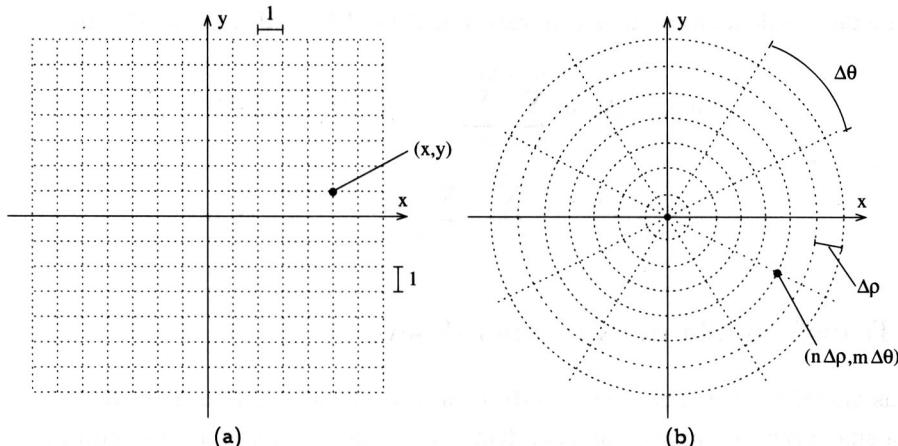


Figura 3.32: Representação de imagens em sistemas de coordenadas distintos. (a) sistema de coordenadas retangulares; (b) sistema de coordenadas polares.

sentada em coordenadas retangulares, conforme ilustra a figura 3.33, em que $\rho_m = m\Delta\rho$ e $\theta_n = n\Delta\theta$.

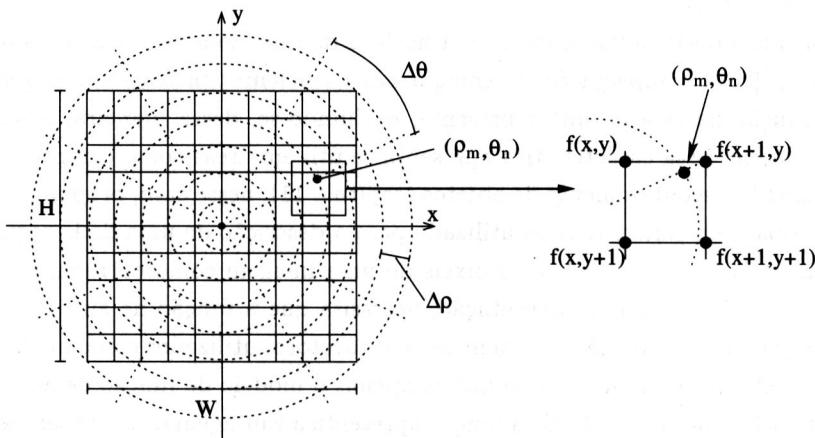


Figura 3.33: Mapeamento de pontos do sistema de coordenadas retangulares para o sistema de coordenadas polares.

A determinação da coordenada (x, y) correspondente a $(m\Delta\rho, n\Delta\theta)$ é obtida por meio das equações 3.100 a 3.101, em que se efetua a translação de $(w/2, h/2)$ para evitar a utilização de coordenadas que apresentem valores negativos na representação das imagens.

$$x = m\Delta\rho \cos(n\Delta\theta) + W/2 \quad (3.100)$$

$$y = m\Delta\rho \sin(n\Delta\theta) + H/2 \quad (3.101)$$

Após a obtenção da representação polar da imagem, a transformada pode ser executada. A aproximação para a AFMT proposta por Derrode e Ghorbel e a sua inversa são mostradas, respectivamente, nas equações 3.102 e 3.103, em que X representa a imagem convertida para

o sistema de coordenadas polares e as variáveis $\Delta\rho$ e $\Delta\theta$ são definidas pela equação 3.99.

$$y_{k,v} = \Delta\rho\Delta\theta \sum_{n=0}^{N-1} \sum_{m=0}^{M-1} x_{n,m} e^{-ikm/M} \rho_n^{\sigma-iv-1} \quad (3.102)$$

$$x_{m,n} = (\rho_n)^{-\sigma} \sum_{v=-V}^V \sum_{k=-K}^K y_{k,v} e^{ikm/M} \rho_n^{iv} \quad (3.103)$$

3.7 Transformada de Karhunen-Loève

Todas as transformadas abordadas neste capítulo possuem núcleo fixo, definido anteriormente a sua execução, ou seja, independente dos dados que terão a representação alterada. No entanto, esta seção descreve os principais aspectos da transformada de Karhunen-Loève¹ (KLT, do inglês, *Karhunen-Loève transform*) que, embora seja uma transformação de coordenadas, seu núcleo é obtido a partir dos dados que compõem a entrada.

Como apresenta o núcleo variável, a execução da KLT divide-se em duas etapas: obtenção do núcleo e execução da transformação de coordenadas propriamente dita. Para que o núcleo seja obtido, deve-se considerar a natureza dos dados, por exemplo, se são pixels que compõem uma imagem ou amostras de um experimento aleatório. Ambas as abordagens são descritas no decorrer desta seção.

A importância dessa transformada está no fato de que, como o núcleo é adaptado aos dados, tem-se que a compactação de energia nos coeficientes próximos à origem é ótimo quanto à redução da relação linear presente, excluindo-se, dessa maneira, a redundância contida nos dados. Essa característica apresenta-se útil em áreas como compressão de imagens (capítulo 6) e reconhecimento de objetos (capítulo 11), nesse caso, proporcionando uma redução no espaço de características utilizado para a descrição de uma dada região.

A execução da KLT que utiliza os pixels de uma imagem como entrada, com objetivo de efetuar alterações em sua representação, necessita que a função de autocorrelação seja calculada. A partir dessa função, extraem-se os autovetores utilizados como núcleo da transformada. Entretanto, normalmente se utiliza apenas o modelo de função de autocorrelação mostrado em 3.104, para $|\rho| < 1$. Essa função apresenta a vantagem de ser separável, além de seus autovetores serem obtidos analiticamente (Rosenfeld e Kak 1982), reduzindo, portanto, o custo computacional requerido para a determinação dos coeficientes da transformada.

$$r^{(u,v)}(m, n) = \rho^{|u-m|} \rho^{|v-n|} \quad (3.104)$$

Quando se aplica a KLT a amostras de um experimento aleatório para alterar sua representação e reduzir a dimensionalidade dos dados, efetua-se uma rotação multidimensional nos eixos do espaço de características, impondo que as variáveis aleatórias presentes no experimento passem a apresentar correlação zero, ou seja, não possa ser estabelecida uma relação linear entre as variáveis.

Exemplo 3.7 Alteração na representação dos dados a partir da KLT quando aplicada a amostras de um experimento aleatório. A figura 3.34 ilustra essa mudança de representação para um espaço de características bidimensional. Quando representados

¹ Esta transformada também é denominada transformada de Hotelling.

nos eixos x e y , pode-se utilizar a função $x \approx y$ para estabelecer uma relação linear entre as variáveis aleatórias.

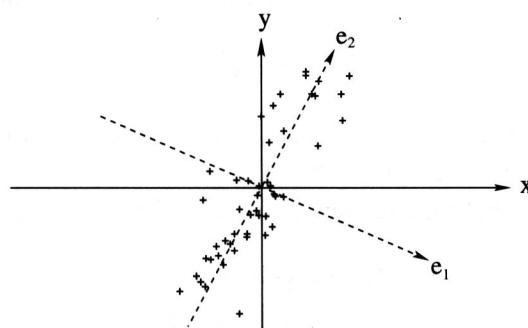


Figura 3.34: Mudança de representação obtida pela transformada de Karhunen-Loève. A representação dos dados, que antes da transformada era nos eixos x e y , passa a ser nos eixos e_1 e e_2 .

Após a execução da KLT, os dados passam a ser representados sobre os eixos e_1 e e_2 , não sendo mais possível estabelecer uma relação linear entre as variáveis aleatórias. Quando isso ocorre, diz-se que as variáveis apresentam correlação baixa ou nula. ■

O grau de dependência linear apresentado pelas variáveis aleatórias é descrito pela matriz de correlação (equação B.17), na qual cada componente apresenta valores entre -1 e 1 . A disposição dos elementos nessa matriz é dada de modo que a posição (m, n) indique a correlação existente entre a m -ésima e a n -ésima variáveis aleatórias. Valores localizados no extremo do intervalo $[-1, 1]$ indicam maior relação linear, a qual é reduzida conforme se aproxima de zero.

Para que não se tenha a presença de redundância nas informações, desde que a matriz de correlação expressa a relação linear entre as variáveis, uma possível abordagem está na transformação dos dados de modo que a nova matriz de correlação seja diagonal. A obtenção dessa matriz é justamente o objetivo da transformada de Karhunen-Loève que, para isso, utiliza a decomposição espectral da matriz de correlação, conforme será descrito nos próximos parágrafos.

Considerando que as variáveis aleatórias que compõem um experimento apresentem média 0 e variância 1 , seja a transformação linear mostrada na equação 3.105, em que os vetores \mathbf{x} e \mathbf{y} contêm, respectivamente, a entrada e o resultado, e a matriz \mathbf{G} denota o núcleo dessa transformação. Para que o resultado apresente apenas variáveis com correlação zero, deve-se impor que a matriz de covariância extraída a partir dos vetores \mathbf{y} seja diagonal.

$$\mathbf{y}^T = \mathbf{Gx}^T \quad (3.105)$$

A definição da matriz de covariância para variáveis aleatórias multivariadas é apresentada na equação 3.106, onde μ_y denota o vetor de médias obtido dos vetores \mathbf{y} .

$$\Sigma_y = \mathbb{E} \{ (\mathbf{y} - \mu_y)^T (\mathbf{y} - \mu_y) \} \quad (3.106)$$

Como a média, ou valor esperado, para o vetor de entrada \mathbf{x} é zero, o valor esperado para \mathbf{y} também será, pois $\mathbb{E}\{\mathbf{y}^T\} = \mathbb{E}\{\mathbf{Gx}^T\} = 0$. Dessa maneira, a definição de covariância apresentada na equação 3.106 pode ser escrita como mostrado na equação 3.107.

$$\Sigma_y = \mathbb{E}\{\mathbf{y}^T \mathbf{y}\} \quad (3.107)$$

Substituindo a equação 3.105 na equação 3.107 e utilizando a propriedade da linearidade da esperança, obtém-se a equação 3.108. Percebe-se que o termo $\mathbf{x}^T \mathbf{x}$ é a matriz de covariância do vetor de entrada, dado que todas as variáveis apresentam média zero.

$$\begin{aligned} \Sigma_y &= \mathbb{E}\{\mathbf{Gx}^T \mathbf{xG}^T\} \\ &= \mathbf{G} \mathbb{E}\{\mathbf{x}^T \mathbf{x}\} \mathbf{G}^T \\ &= \mathbf{G} \Sigma_x \mathbf{G}^T \end{aligned} \quad (3.108)$$

Tem-se como resultado da álgebra linear que uma matriz \mathbf{A} pode ser decomposta em $\mathbf{A} = \mathbf{PDP}^{-1}$, em que \mathbf{P} é a matriz que contém os autovetores de \mathbf{A} em suas colunas e \mathbf{D} é uma matriz diagonal, composta pelos autovalores de \mathbf{A} . Como Σ_x é simétrica, pode-se utilizar a representação $\Sigma_x = \mathbf{PDP}^T$, na qual \mathbf{D} representa a matriz diagonal com os autovalores de Σ_x e \mathbf{P} a matriz com autovetores obtidos a partir de Σ_x .

Alterando a equação 3.108 para a representação mostrada na equação 3.109, tem-se que Σ_y será diagonal quando as linhas do núcleo da transformação contiverem os autovetores obtidos a partir da matriz de covariância Σ_x . Ou seja, o núcleo da transformada que diagonaliza a matriz de correlação dos coeficientes resultantes é obtido com a decomposição da matriz de covariância dos dados de entrada.

$$\Sigma_y = \mathbf{G}^T \Sigma_x \mathbf{G} \quad (3.109)$$

A partir dos elementos presentes na diagonal principal da matriz Σ_y , pode-se utilizar a equação 3.110 para determinar a fração da variância que cada autovetor descreve, no caso desta equação, a fração da variância descrita pelo i -ésimo autovetor. Utiliza-se essa característica para a redução da dimensionalidade presentes nos dados, com a eliminação dos autovetores que explicam uma pequena fração da variância.

$$p_i = \frac{\lambda_i}{\sum_{k=1}^N \lambda_k} \quad (3.110)$$

O algoritmo 3.8 descreve a obtenção do núcleo da transformada de Karhunen-Loève (matriz \mathbf{G}). Após o cálculo de \mathbf{G} , basta efetuar a transformação linear mostrada na equação 3.105 para se obter a mudança do sistema de coordenadas original para o novo sistema, determinado pelos autovetores extraídos da matriz Σ_x .

Como a matriz composta pelos autovetores de Σ_x é ortogonal, sua inversa é igual à transposta (teorema A.3), portanto, sua inversa pode ser obtida por meio da equação 3.111. O seu resultado recupera a representação dos dados em relação ao sistema original de coordenadas.

$$\mathbf{x}^T = \mathbf{G}^T \mathbf{y}^T \quad (3.111)$$

Núcleo da transformada de Karhunen-Loëve

Algoritmo para determinação do núcleo da transformada de Karhunen-Loëve. Considerando m amostras de n variáveis aleatórias. Cada amostra deve ser colocada em uma linha da matriz M , composta por n colunas.

```

1 // subtração da média dos elementos da entrada
2 padronizar as variáveis aleatórias (apêndice B)
3
4 // determinação da matriz de covariância (equação B.16)
5 calcular a matriz de covariância para M
6
7 // autovalores e autovetores (apêndice A)
8 calcular os autovalores e autovetores da matriz de covariância
9 calcular a transposta da matriz dos autovetores

```

Algoritmo 3.8: Núcleo da transformada de Karhunen-Loëve.

Exemplo 3.8 Utilização da KLT para redução da dimensionalidade dos dados utilizados para descrição de regiões de uma imagem. Neste exemplo foram extraídas 44 amostras da imagem mostrada na figura 3.35, em que cada amostra é formada por 32×32 pixels.



Figura 3.35: Imagem aérea de um terreno.

Com o intuito de descrever cada uma das regiões, calcula-se a matriz de covariância (seção 8.1.2) para cada uma das amostras, extraíndo as medidas de contraste e variância, representadas pelas variáveis x_0 e x_1 , respectivamente. Após a padronização, seus valores são apresentados na tabela 3.2.

As matrizes de correlação, de autovalores e do núcleo da transformação são mostradas em 3.112. Percebe-se que a correlação entre as medidas de contraste e variância, obtidas para os dados originais, é alta. Entretanto, após a aplicação da KLT nos vetores aleatórios compostos pelos elementos x_0 e x_1 da tabela 3.2, a correlação entre as duas medidas torna-se nula, conforme a matriz Σ_y .

$$\rho = \begin{bmatrix} 1.000 & 0.635 \\ 0.635 & 1.000 \end{bmatrix} \quad \Sigma_y = \begin{bmatrix} 1.635 & 0 \\ 0 & 0.365 \end{bmatrix} \quad G = \begin{bmatrix} 0.707 & -0.707 \\ 0.707 & 0.707 \end{bmatrix} \quad (3.112)$$

Tabela 3.2: Valores das medidas extraídas a partir da matriz de co-ocorrência de cada amostra.

#	x_0	x_1									
1	0.176	-0.278	12	-0.071	0.826	23	1.086	-0.199	34	1.215	0.317
2	-0.341	-0.595	13	-1.602	-1.477	24	0.162	-0.496	35	0.448	2.957
3	0.683	-0.024	14	-1.313	-0.958	25	0.386	0.290	36	0.768	1.417
4	1.754	1.788	15	-1.266	-0.850	26	-0.869	-0.257	37	-0.019	-0.522
5	1.877	0.771	16	0.340	0.316	27	-0.616	-0.408	38	-0.307	-0.332
6	0.007	-0.463	17	1.464	0.264	28	-1.481	-1.381	39	-1.006	-0.865
7	1.086	0.672	18	-0.353	-0.746	29	-1.557	-1.448	40	-0.377	-0.446
8	0.468	0.081	19	0.878	0.334	30	-1.158	-1.088	41	0.567	0.797
9	0.278	0.971	20	1.640	1.318	31	-1.236	-0.411	42	-0.356	0.876
10	-0.427	1.668	21	1.734	0.453	32	0.208	0.181	43	-1.229	-1.165
11	-0.026	1.294	22	0.038	-0.508	33	0.294	-0.772	44	-1.948	-1.905

A tabela 3.3 apresenta os coeficientes resultantes da aplicação da KLT. Conforme matriz de autovalores, a primeira medida (y_0) tornou-se pouco representativa para descrição de regiões contidas na imagem, pois explica apenas 18% de variância contida nos dados. Dessa maneira, quando se efetua, por exemplo, a classificação de regiões entre solo e vegetação, embora não seja garantida a separabilidade entre as classes, pode-se utilizar apenas a variável y_1 , pois esta descreve uma fração maior da variância.

Tabela 3.3: Valores obtidos após a aplicação da KLT nas variáveis aleatórias que compõem a tabela 3.2.

#	y_0	y_1									
1	-0.321	-0.072	12	0.635	0.534	23	-0.909	0.628	34	-0.635	1.084
2	-0.180	-0.662	13	0.089	-2.177	24	-0.465	-0.237	35	1.774	2.408
3	-0.500	0.466	14	0.251	-1.606	25	-0.068	0.478	36	0.459	1.545
4	0.024	2.505	15	0.294	-1.496	26	0.433	-0.796	37	-0.356	-0.383
5	-0.782	1.872	16	-0.017	0.464	27	0.147	-0.724	38	-0.017	-0.452
6	-0.332	-0.322	17	-0.848	1.222	28	0.071	-2.024	39	0.100	-1.323
7	-0.292	1.243	18	-0.278	-0.777	29	0.077	-2.125	40	-0.048	-0.582
8	-0.274	0.389	19	-0.385	0.857	30	0.049	-1.588	41	0.163	0.965
9	0.490	0.883	20	-0.228	2.092	31	0.583	-1.164	42	0.871	0.367
10	1.481	0.877	21	-0.906	1.547	32	-0.019	0.275	43	0.046	-1.693
11	0.933	0.897	22	-0.386	-0.332	33	-0.753	-0.338	44	0.030	-2.725

A figura 3.36 apresenta os gráficos dos valores listados nas tabelas 3.2 e 3.3. Percebe-se a ausência de correlação no gráfico mostrado em (b), e a maior variância na direção horizontal, podendo ser corroborado pelo primeiro autovalor da matriz Σ_y .

3.8 Transformada Wavelet

A transformada de Fourier permite descrever as diferentes freqüências contidas na imagem e, embora seja de grande importância para algumas aplicações, a localização espacial de tais freqüências não pode ser determinada. Com o intuito de contornar essa limitação apresentada pela transformada de Fourier, utiliza-se a transformada wavelet, a qual, além de informações sobre freqüências, provê informações sobre a imagem.

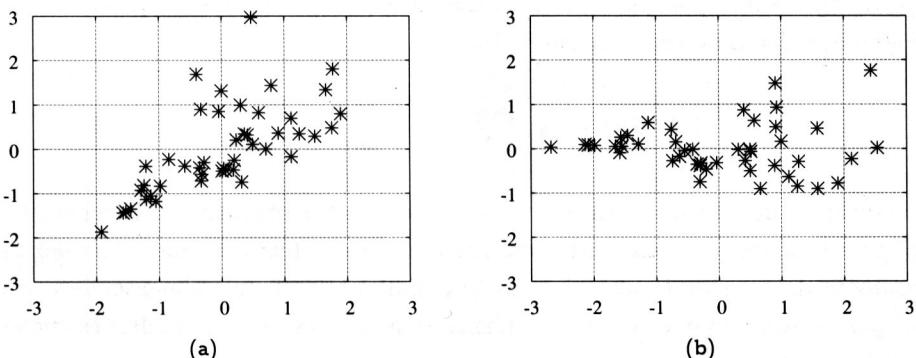


Figura 3.36: Gráficos apresentando as medidas de variância e contraste obtidas a partir da matriz de co-ocorrência. (a) valores originais; (b) resultado da KLT.

Técnicas que aplicam multirresolução em imagens, tais como *wavelets*, objetivam alterar a representação de modo que tanto informações sobre freqüência quanto informações espaciais estejam presentes. O uso da transformada wavelet foi proposta inicialmente para análise de imagens no trabalho de Mallat (1989).

Dado que a transformada wavelet decompõe uma imagem de acordo com a freqüência espacial, cada decomposição representa os atributos visíveis em uma dada faixa de freqüências. Por exemplo, na representação que mostra as freqüências baixas, objetos grandes e com baixas variações em intensidade são enfatizados, enquanto em uma representação que mostra as freqüências altas, detalhes finos, tais como bordas, tornam-se perceptíveis.

A aplicação da transformada wavelet difere das transformadas descritas anteriormente, pois ela utiliza mais de um núcleo para obter os coeficientes da transformada. Tais núcleos são obtidos a partir das funções unidimensionais denominadas *função de escala* e *função wavelet*, mostradas nas equações 3.113 e 3.114, respectivamente, em que k denota a posição onde essa função será aplicada no eixo x , enquanto j provê sua largura.

$$\phi_{j,k}(x) = 2^{j/2} \phi(2^j x - k) \quad (3.113)$$

$$\psi_{j,k}(x) = 2^{j/2} \psi(2^j x - k) \quad (3.114)$$

Pela análise de multirresolução, a transformada wavelet é semelhante a um banco de filtros QMF (do inglês, *quadrature mirror filter*) (Mallat 1989). O QMF é um banco de filtros que divide o sinal de entrada em bandas distintas, neste caso formado pelos filtros passa-baixa e passa-alta, os quais formam as bases para a transformada.

Duas operações importantes para o estudo de filtros são as operações de *decimação* (do inglês, *downsampling*) e *interpolação* (do inglês, *upsampling*). A operação de decimação, definida na equação 3.115, consiste em descartar os termos de uma seqüência cujo índice não é múltiplo de q . Para as transformadas wavelets, costuma-se utilizar $q = 2$, descartando, desse modo, elementos alternados em uma seqüência.

$$(\downarrow q)(\vec{u}) = (\vec{u}_{nq}) \quad (3.115)$$

O operador de interpolação adiciona zeros intercalados em uma seqüência de termos. Esse operador é definido pela equação 3.116.

$$(\uparrow q)(\vec{u})(k) = \begin{cases} \vec{u}(k) & \text{se } n = 2k \\ 0 & \text{se } n = 2k + 1 \end{cases} \quad (3.116)$$

A convolução dos filtros permite a obtenção de coeficientes em um nível seguinte. A relação entre o cálculo de coeficientes wavelets e bancos de filtros foi mostrada por diversos pesquisadores, dentre eles, Daubechies (1992) e Mallat (1989). Inicialmente, deve-se definir uma função de escala ϕ que relaciona a transformada ao aspecto de multirresolução. Essa função é dada pela equação 3.117.

$$\phi(x) = \sqrt{2} \sum_k l_k \phi(2x - k) \quad (3.117)$$

Para que a representação wavelet esteja completa é preciso definir uma segunda função para expandir as diferenças entre as resoluções. Essa função, denotada função wavelet, é mostrada na equação 3.118.

$$\psi(x) = \sqrt{2} \sum_k h_k \phi(2x - k) \quad (3.118)$$

Os coeficientes de l e h , mostrados nas equações 3.117 e 3.118, são dados pelas equações 3.119 e 3.120, respectivamente. Tais coeficientes são utilizados pela transformada rápida wavelet. Essa computação dá-se por meio do cálculo iterativo de convoluções. Como resultado, têm-se os coeficientes da transformada de wavelet para uma dada escala.

$$l_k = \sqrt{2} \int_{-\infty}^{\infty} \phi(x) \phi(2x - k) dt \quad (3.119)$$

$$h_k = \sqrt{2} \int_{-\infty}^{\infty} \psi(x) \phi(2x - k) dt \quad (3.120)$$

Alguns exemplos de filtros são Haar, mostrados em 3.121, e os filtros de Daubechies, mostrados em 3.122.

$$l = \left[\begin{array}{cc} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{array} \right] \quad h = \left[\begin{array}{cc} \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{array} \right] \quad (3.121)$$

$$l = \left[\begin{array}{cccc} \frac{1+\sqrt{3}}{4\sqrt{2}} & \frac{3+\sqrt{3}}{4\sqrt{2}} & \frac{3-\sqrt{3}}{4\sqrt{2}} & \frac{1-\sqrt{3}}{4\sqrt{2}} \end{array} \right] \quad h = \left[\begin{array}{cccc} \frac{1-\sqrt{3}}{4\sqrt{2}} & -\frac{3-\sqrt{3}}{4\sqrt{2}} & \frac{3+\sqrt{3}}{4\sqrt{2}} & -\frac{1+\sqrt{3}}{4\sqrt{2}} \end{array} \right] \quad (3.122)$$

A aplicação dos QMF sobre uma imagem nas direções vertical e horizontal gera um nível de decomposição e produz quatro sub-bandas, LL, LH, HL e HH, como pode ser visto nas figuras 3.37 e 3.38. A decomposição pode ser realizada recursivamente na sub-banda LL (figura 3.38(b)), obtendo níveis adicionais de decomposição.

As sub-bandas LL e HH denotam, respectivamente, as freqüências baixas e altas da imagem, enquanto LH e HL descrevem as freqüências intermediárias presentes na imagem. As sub-bandas LH, HL e HH correspondem às imagens de detalhe, enquanto a sub-banda de baixa freqüência, LL, é a aproximação da imagem em uma resolução menor, estando relacionada à informação espacial.

Há dois métodos para a aplicação dos filtros QMF em imagens, chamados de *decomposição padrão* e *decomposição não-padrão*. Na decomposição padrão, inicia-se calculando

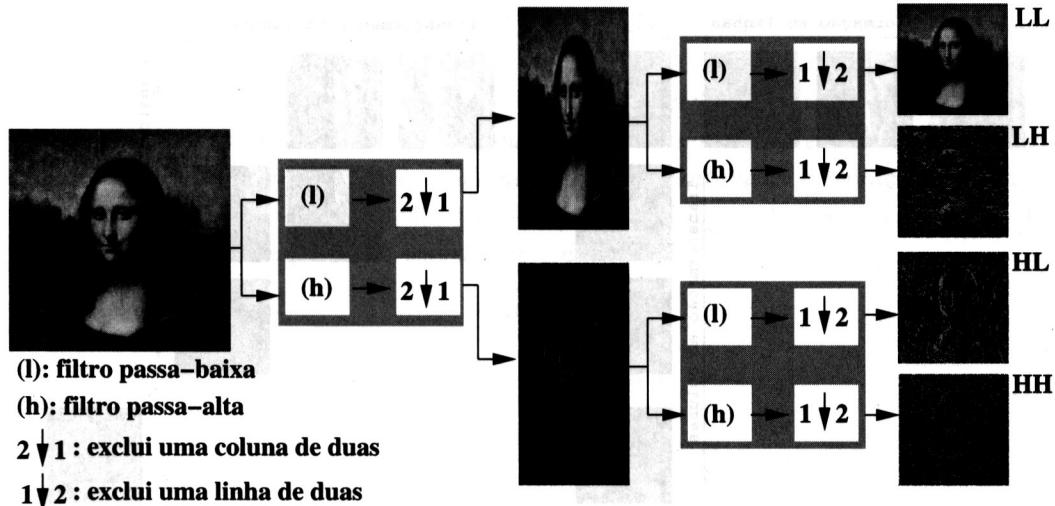


Figura 3.37: Decomposição de um nível da imagem Monalisa utilizando o filtro de Haar, em que as sub-bandas de detalhes foram equalizadas para uma melhor visualização.

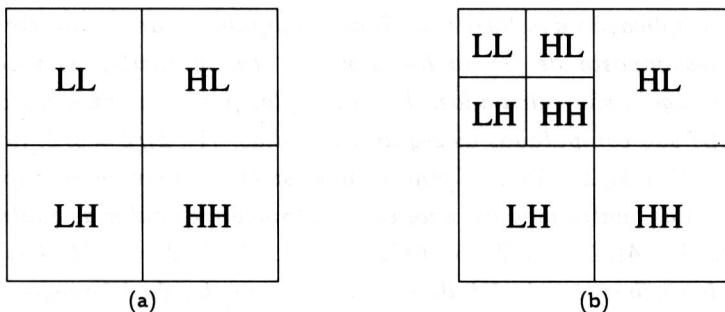


Figura 3.38: Transformadas bidimensionais. (a) decomposição em um nível; (b) decomposição em dois níveis.

a transformada wavelet completa para cada linha da imagem, resultando em uma imagem onde a primeira coluna contém a média de cada linha e as demais colunas contêm as diferenças. O algoritmo então repete o cálculo para cada coluna, resultando assim em uma imagem onde somente o primeiro pixel da primeira linha contém a média e todos os demais contêm as diferenças. Um exemplo desse método é mostrado na figura 3.39(a).

A decomposição não-padrão intercala colunas e linhas a cada iteração da transformada wavelet. O primeiro passo é calcular somente a primeira iteração da transformada para cada linha da imagem, não a transformada completa. Isso gera uma imagem com médias em sua metade esquerda e diferenças em sua metade direita. O segundo passo é repetir o processo para as colunas, resultando em uma imagem onde somente o quadrante esquerdo superior possui as médias. O processo todo é repetido recursivamente nesse quadrante até que reste apenas um pixel contendo as médias e todos os demais contendo apenas diferenças. Um exemplo é mostrado na figura 3.39(b).

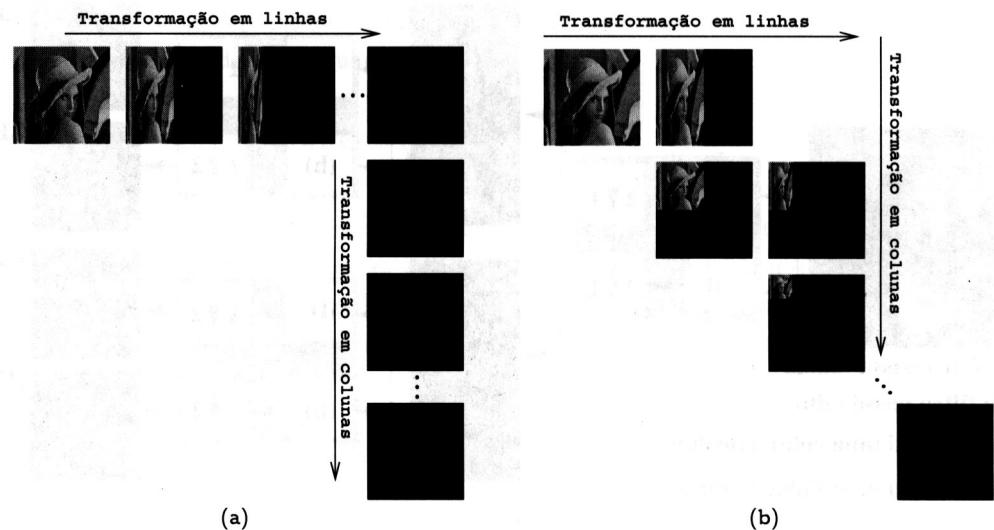


Figura 3.39: Aplicação da transformada wavelet bidimensional. (a) decomposição padrão; (b) decomposição não-padrão.

Exemplo 3.9 A aplicação dos filtros de Haar exemplifica claramente como uma imagem pode ser decomposta de forma hierárquica. Inicialmente, as médias de cada dois pixels da imagem são calculadas. Por exemplo, para um conjunto de oito pixels $(1, 2, 3, 4, 5, 6, 7, 8)$ são computadas as seguintes médias: $(1+2)/2 = 3/2$, $(3+4)/2 = 7/2$, $(5+6)/2 = 11/2$, $(7+8)/2 = 15/2$. Como é impossível reconstruir os oito pixels originais somente com as quatro médias, é necessário também calcular as quatro diferenças: $(1-2)/2 = -1/2$, $(3-4)/2 = -1/2$, $(5-6)/2 = -1/2$, $(7-8)/2 = -1/2$. Essas diferenças são chamadas de coeficientes de detalhes ou, simplesmente, de diferença. A figura 3.40 exemplifica esse processo de decomposição e de recuperação dos coeficientes.

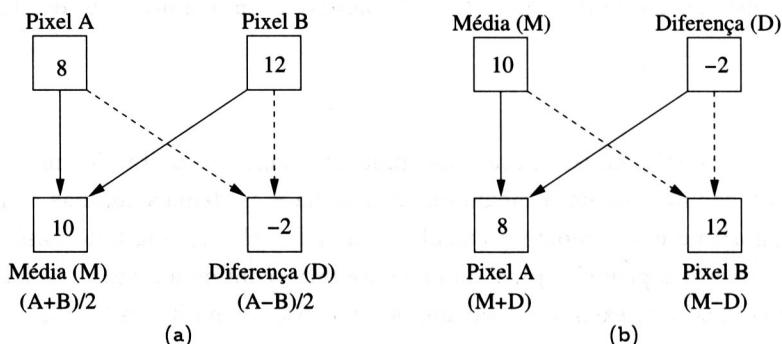


Figura 3.40: Decomposição e recuperação dos coeficientes de Haar.

A média carrega a maior parte da informação e pode ser interpretada como uma aproximação da imagem original, enquanto a diferença carrega apenas os detalhes necessários para aumentar a fidelidade dessa aproximação, reconstituindo a imagem original. Como resultado da aplicação dessa transformação ao conjunto de pixels $(1, 2, 3, 4, 5, 6, 7, 8)$, obtém-se o conjunto de coeficientes

$(3/2, 7/2, 11/2, 15/2, -1/2, -1/2, -1/2, -1/2)$ que possui quatro médias e quatro valores pequenos que representam a diferença. Assim, é possível repetir o processo somente nos quatro primeiros valores e obter então duas médias e seis diferenças. O processo pode se repetir, como visto na figura 3.41, até ter como resultado somente uma média seguida de vários valores pequenos que representam as diferenças. Dessa maneira, tem-se $(36/8, -16/8, -4/4, -4/4, -1/2, -1/2, -1/2, -1/2)$ como o resultado final da aplicação da transformada wavelet de Haar.

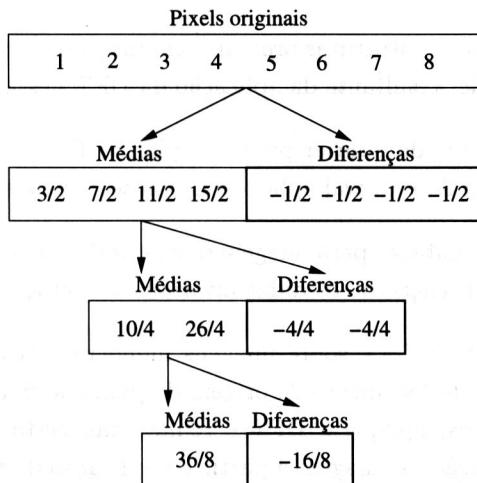


Figura 3.41: Transformada wavelet de Haar.

3.9 Notas Bibliográficas

Dentre as abordagens utilizadas para obtenção de algoritmos rápidos para a transformada de Fourier, além da decimação em tempo ou freqüência descritos na seção 3.2.4, encontram-se métodos baseados em fatores primos (Burrus e Eschenbacher 1981) e o algoritmo de Winograd (Silverman 1977).

Algumas revisões bibliográficas também são encontradas na literatura, tais como o trabalho de Strang (1999), tratando sobre a transformada discreta do cosseno; os dois trabalhos de Blair (1995b, 1995a) apresentam uma descrição da transformada de Fourier e dos algoritmos rápidos utilizados para obtenção dos coeficientes resultantes da transformada.

Algoritmos rápidos para o cálculo da transformada de Hartley podem ser encontrados nos trabalhos de Bi et al. (1998), Bracewell (1983), Hou (1987) e Liu et al. (2003). Os trabalhos propostos por Feig e Winograd (1992) e Kok (1997) apresentam métodos rápidos para a determinação da transformada discreta do cosseno, enquanto Gupta e Rao (1990), Liu et al. (2000) e Yip e Rao (1980) propõem algoritmos rápidos para a transformada discreta do seno.

Em algumas aplicações, a entrada da DFT é composta apenas por números reais. Nesse caso, a propriedade da simetria conjugada (seção 3.2.3) pode ser utilizada para reduzir a quantidade de computação requerida. No trabalho de Sorensen et al. (1987) são descritos

métodos para determinação dos coeficientes da transformada de Fourier para funções de entrada que apresentam talas características.

Descrições adicionais sobre as transformadas wavelets podem ser encontradas em Chui (1992), Erlebacher et al. (1996), Louis et al. (1997), Mallat (1998), Meyer (1993) e Stollnitz et al. (1996).

3.10 Problemas

1. Qual é a relação entre as estruturas presentes em uma imagem e as freqüências obtidas no espectro de Fourier resultante da aplicação da DFT nessa imagem?
2. Calcule a transformada de Fourier para a seqüência $f(x) = [1 \ 2 \ 0 \ 1]$. Mostre que o resultado do teorema de Parseval aplica-se a essa seqüência e sua transformada.
3. Proponha uma representação para imagem que permita que a transformada de Fourier seja utilizada para obtenção de medidas invariantes à rotação.
4. Implemente e execute a DFT sobre imagens monocromáticas e plote o espectro de Fourier com e sem o deslocamento da origem do plano de freqüências. Após a divisão da imagem em blocos, aplique a DFT e remova um certo número de coeficientes. Obtenha a reconstrução da imagem a partir da aplicação da transformada inversa.
5. Considerando os espectros de Fourier resultantes da aplicação da DFT em blocos da imagem, como aqueles mostrados na figura 3.12(b), apresente uma abordagem que possa ser utilizada para descrição das regiões contidas na imagem.
6. Por que a reconstrução mostrada na figura 3.28(d) apresenta-se mais suavizada que a imagem original mostrada na figura 3.28(a)?
7. Como ficaria a reconstrução da imagem se apenas os coeficientes localizados em regiões distantes da origem fossem mantidos na figura 3.28(b)?
8. Qual será a principal diferença entre o espectro de Fourier resultante para as figuras 3.31(b) e (d)?
9. Qual é o comportamento esperado para a compactação de energia obtida pela DCT quando aplicada a uma função com comportamento pseudo-aleatório e a uma função que apresenta correlação alta?
10. Explique por que os coeficientes localizados em regiões distantes da origem do plano de freqüências podem ser removidos sem acarretar grande perda de informações quando se aplica a DCT.
11. Divida uma imagem em blocos e execute a DFT e a DCT. Posteriormente, remova 25%, 50% e 75% dos coeficientes e obtenha a reconstrução da imagem por meio da aplicação das transformadas inversas. Utilize métricas de qualidade de imagens apresentadas na seção 2.12 para comparar os resultados obtidos pela DFT àqueles obtidos por meio da DCT, em relação à imagem original.

12. Apresente um algoritmo para converter uma imagem representada em coordenadas retangulares para coordenadas polares.
13. Por que não é possível obter uma representação gráfica geral, como aquela mostrada na figura 3.30, para o núcleo da transformada de Karhunen-Loève?
14. Considere a existência de um conjunto de amostras descritas por um certo número de medidas. Como a KLT pode ser utilizada para redução da dimensionalidade dos dados?
15. Supondo que o objetivo seja a classificação de amostras entre duas classes, apresente um exemplo ilustrando que a redução da dimensionalidade dos dados utilizando a KLT não garante a separabilidade entre as classes.

Referências Bibliográficas

- Ahmed, N., Natarajan, T. e Rao, K. R. (1974). Discrete Cosine Transform, *IEEE Transactions on Computers* C-23(1): 90–93.
- Almeida, L. B. (1994). The Fractional Fourier Transform and Time-Frequency Representations, *IEEE Transactions on Signal Processing* 42(11): 3084–3091.
- Bi, G. e Chen, Y. (1997). Split-Radix for 2D DFT, *Electronics Letters* 33(3): 203–205.
- Bi, G., Kot, A. C. e Meng, Z. (1998). Computation of 2D Discrete Hartley Transform, *Electronics Letters* 34(11): 1058–1059.
- Blair, G. M. (1995a). A Review of the Discrete Fourier Transform. Part 1: Manipulating the Powers of Two, *Electronics & Communication Engineering Journal* 7(4): 169–177.
- Blair, G. M. (1995b). A Review of the Discrete Fourier Transform. Part 2: Non-Radix Algorithms, Real Transforms and Noise, *Electronics & Communication Engineering Journal* 7(5): 187–194.
- Bortfeld, T. e Dinter, W. (1996). Calculation of Multidimensional Hartley Transforms Using One-Dimensional Fourier Transforms, *IEEE Transactions on Signal Processing* 43(5): 1306–1310.
- Bouguezel, S., Ahmad, M. O. e Swamy, M. N. S. (2004). Improved Radix-4 and Radix-8 FFT Algorithms, *International Circuits and Systems*, Vol. 3, pp. 561–564.
- Boussakta, S., Alshibami, O. H. e Aziz, M. Y. (2001). Radix- $2 \times 2 \times 2$ Algorithm for the 3-D Discrete Hartley Transform, *IEEE Transactions on Signal Processing* 49(12): 3145–3156.
- Bracewell, R. N. (1983). Discrete Hartley Transform, *Journal of the Optical Society of America* 73: 1832–1835.
- Bultheel, A. e Sulbaran, H. E. M. (2004). Computation of the Fractional Fourier Transform, *Applied and Computational Harmonic Analysis* 16(3): 182–202.
- Burrus, C. (1977). Index Mappings for Multidimensional Formulation of the DFT and Convolution, *IEEE Transactions on Acoustics, Speech, and Signal Processing* 25(3): 239–242.
- Burrus, C. e Eschenbacher, P. (1981). An In-place, In-order Prime Factor FFT Algorithm, *IEEE Transactions on Acoustics, Speech, and Signal Processing* 29(4): 806–817.
- Casasent, D. e Psaltis, D. (1976). Position Oriented and Scale Invariant Optical Correlation, *Applied Optics* 15: 1793–1799.
- Chan, S. C. e Ho, K. L. (1992). Fast Algorithms for Computing the Discrete Cosine Transform, *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing* 39(3): 185–190.

- Chui, C. K. (1992). *An Introduction to Wavelets*, Academic Press, San Diego, CA, Estados Unidos.
- Cooley, J. W. e Tukey, J. W. (1965). An Algorithm for the Machine Calculation of Complex Fourier Series, *Mathematics of Computation* 19: 297–301.
- Cvetkovic, Z. e Popovic, M. V. (1992). New Fast Recursive Algorithms for the Computation of Discrete Cosine and Sine Transforms, *IEEE Transactions on Signal Processing* 40(8): 2083–2086.
- Daubechies, I. (1992). *Ten Lectures on Wavelets*, CBMS-NSF Regional Conference Series in Applied Mathematics, SIAM.
- Derrode, S. e Ghorbel, F. (2001). Robust and Efficient Fourier-Mellin Transform Approximations for Gray-level Image Reconstruction and Complete Invariant Description, *Computer Vision and Image Understanding* 83(1).
- Duhamel, P. (1986). Implementation of "Split-Radix" FFT Algorithms for Complex, Real, and Real-Symmetric Data, *IEEE Transactions on Acoustics, Speech, and Signal Processing* 34(2): 285–295.
- Erlebacher, G., Hussaini, M. Y. e Jameson, L. M. (eds.) (1996). *Wavelets: Theory and Applications*, Oxford University Press, Nova York, NY, Estados Unidos.
- Evans, D. M. W. (1987). An Improved Digit-Reversal Permutation Algorithm for the Fast Fourier and Hartley Transforms, *IEEE Transactions on Acoustics, Speech, and Signal Processing* 35(8): 1120–1125.
- Evans, D. M. W. (1989). A Second Improved Digit-Reversal Permutation Algorithm for Fast Transforms, *IEEE Transactions on Acoustics, Speech, and Signal Processing* 37(8): 1288–1291.
- Evans, D. M. W. (2004). A Novel Algorithm for Computing the 1-D Discrete Hartley Transform, *IEEE Signal Processing Letters* 11(2): 156–159.
- Feig, E. e Winograd, S. (1992). Fast Algorithms for the Discrete Cosine Transform, *IEEE Transactions on Signal Processing* 40(9): 2174–2193.
- Ghorbel, F. (1994). A Complete Invariant Description for Gray-Level Images by the Harmonic Analysis Approach, *Pattern Recognition Letters* 15(10): 1043–1051.
- Grigoryan, A. M. e Agaian, S. S. (2000). Three Algorithms for Computing the 2-D Discrete Hartley Transform, *Proceedings of International Conference on Image Processing*, Vol. 2, pp. 359–362.
- Gupta, A. e Rao, K. R. (1990). A Fast Recursive Algorithm for the Discrete Sine Transform, *IEEE Transactions on Acoustics, Speech, and Signal Processing* 38(3): 553–557.
- Hartley, R. V. L. (1942). A More Symmetrical Fourier Analysis Applied to Transmissions Problems, *Proceedings of IRE* 30: 144–150.
- Hou, H. S. (1987). The Fast Hartley Transform Algorithm, *IEEE Transactions on Computers* C-36(2): 147–156.
- Kok, C. W. (1997). Fast Algorithm for Computing Discrete Cosine Transform, *IEEE Transactions on Signal Processing* 45(3): 757–760.
- Liu, J. G., Chan, F. H. Y., Lam, F. K. e Li, H. F. (2000). Moment-Based Fast Discrete Sine Transforms, *IEEE Signal Processing Letters* 7(8): 227–229.
- Liu, J. G., Chan, F. H. Y., Lam, F. K., Li, H. F. e Fung, G. S. (2003). Moment-Based Fast Discrete Hartley Transform, *Signal Processing* 83(8): 1749–1757.
- Louis, A. K., Maaf, P. e Rieder, A. (1997). *Wavelets: Theory and Applications*, John Wiley & Sons, Chichester, Reino Unido.
- Makhoul, J. (1980). A Fast Cosine Transform in One and Two Dimensions, *IEEE Transactions on Acoustics, Speech, and Signal Processing* 28(1): 27–34.
- Mallat, S. (1998). *A Wavelet Tour of Signal Processing*, Academic Press, San Diego, CA, Estados Unidos.

- Mallat, S. G. (1989). A Theory for Multiresolution Signal Decomposition: The Wavelet Representation, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 11(7): 674–693.
- McBride, A. C. e Kerr, F. H. (1987). On Namius Fractional Fourier Transforms, *IMA Journal of Applied Mathematics* 39: 159–175.
- Meyer, Y. (1993). *Wavelets: Algorithms and Applications*, Society for Industrial and Applied Mathematics, Philadelphia, PA, Estados Unidos.
- Namius, V. (1980). The Fractional Order Fourier Transform and its Application to Quantum Mechanics, *J. Inst. Math. Appl.* 25: 241–265.
- Narashimha, M. J. e Peterson, A. M. (1978). On the Computation of the Discrete Cosine Transform, *IEEE Transactions on Communications* 26: 934–946.
- Ozaktas, H. M., Arikan, O., Kutay, M. A. e Bozdagt, G. (1996). Digital Computation of the Fractional Fourier Transform, *IEEE Transactions on Signal Processing* 44(9): 2141–2150.
- Pei, S.-C. e Wu, J.-L. (1987). Split Vector Radix 2-D Fast Fourier Transform, *IEEE Transactions on Circuits and Systems CAS-34*(8): 978–980.
- Rosenfeld, A. e Kak, A. C. (1982). *Digital Picture Processing*, Vol. 1–2, Academic Press, Orlando, FL, Estados Unidos.
- Sanchez, V., Garcia, P., Peinado, A. M., Segura, J. C. e Rubio, A. J. (1995). Diagonalizing Properties of the Discrete Cosine Transforms, *IEEE Transactions on Signal Processing* 43(11): 2631–2641.
- Shunji, H., Jianguo, W. e Hueizhong, Q. (1991). Split Vector Radix Algorithm for Two-Dimensional Hartley Transform, *IEEE Transactions on Aerospace and Electronic Systems* 27(6): 865–868.
- Silverman, H. (1977). An Introduction to Programming the Winograd Fourier Transform Algorithm (WFTA), *IEEE Transactions on Acoustics, Speech, and Signal Processing* 25(2): 152–165.
- Sorensen, H., Heideman, M. e Burrus, C. (1986). On Computing the Split-Radix FFT, *IEEE Transactions on Acoustics, Speech, and Signal Processing* 34(1): 152–156.
- Sorensen, H., Jones, D., Burrus, C. e Heideman, M. (1985). On Computing the Discrete Hartley Transform, *IEEE Transactions on Acoustics, Speech, and Signal Processing* 33(5): 1231–1238.
- Sorensen, H., Jones, D., Meideman, M. e Burrus, C. (1987). Real-Valued Fast Fourier Transform Algorithms, *IEEE Transactions on Acoustics, Speech, and Signal Processing* 35(6): 849–863.
- Stollnitz, E. J., DeRose, T. D. e Salesin, D. H. (1996). *Wavelets for Computer Graphics: Theory and Applications*, Morgan Kaufmann Publishers, San Francisco, CA, Estados Unidos.
- Strang, G. (1999). The Discrete Cosine Transform, *SIAM Review* 41(1): 135–147.
- Sundararajan, D. e Ahmad, M. O. (1996). Vector Split-Radix Algorithm for DFT Computation, *Proceedings of International Symposium on Circuits and Systems*, Vol. 2, pp. 532–535.
- Wu, H. R. e Paolini, F. J. (1989). On the Two-Dimensional Vector Split-Radix FFT Algorithm, *IEEE Transactions on Acoustics, Speech, and Signal Processing* 37(8): 1302–1304.
- Yatagay, T., Choji, K. e Saito, H. (1981). Pattern Classification Using Optical Mellin Transform and Circular Photodiode Array, *Optical Communications* 38(3): 162–165.
- Yip, P. e Rao, K. (1980). A Fast Computational Algorithm for the Discrete Sine Transform, *IEEE Transactions on Communications* 28(2): 304–307.
- Yip, P. e Rao, K. R. (1984). Fast Decimation-in-Time Algorithms for DST's and DCT's, *Circuits, Systems, and Signal Processing* 3: 387–408.