Universidade Estadual de Campinas
Faculdade de Engenharia Elétrica e de Computação
Departamento de Comunicações

Transformada Seno Discreta com Rotação de Eixos Bidimensional (DSTr-2D): Aplicações na Codificação e Interpolação de Imagens para Redução de Efeito de Blocos

Evaldo Gonçalves Pelaes

Orientador:
Prof. Dr. Yuzo Iano

Tese apresentada à Faculdade de Engenharia Elétrica e de Computação da Universidade Estadual de Campinas como parte dos requisitos exigidos para a obtenção do título de Doutor em Engenharia Elétrica.

Campinas, SP - Brasil
1998

Este exemplar corresponde a redação final da tese defendida por Evaldo Gonçalves Pelaes e aprovada pela Comissão Julgada em 29 Outubro 1998
Orientador

UNICAMP BIBLIOTECA CENTRAL

384195

CM-00119566-0

UNIDADE BC
M.º CHAMADA:
T/UNICAMP
P361t
V. Ex.
TOMBO BC/36048
PROC. 395/98
C [ ] D [x]
PREÇO R$11,00
DATA 18/12/98
N.º CPD

FICHA CATALOGRÁFICA ELABORADA PELA
BIBLIOTECA DA ÁREA DE ENGENHARIA - BAE - UNICAMP

P361t

Pelaes, Evaldo Gonçalves

Transformada seno discreta com rotação de eixos bidimensional (DSTr-2D): aplicações na codificação e interpolação de imagens para redução de efeito de blocos. / Evaldo Gonçalves Pelaes.--Campinas, SP: [s.n.], 1998.

Orientador: Yuzo Iano.

Tese (doutorado) - Universidade Estadual de Campinas, Faculdade de Engenharia Elétrica e de Computação.

1. Processamento de imagens. 2. Interpolação. I. Iano, Yuzo. II. Universidade Estadual de Campinas. Faculdade de Engenharia Elétrica e de Computação. III. Título.

OFERECIMENTO

"Deus de nossos pais, e Senhor de misericórdia, que todas as coisas criastes pela vossa palavra, e que por vossa sabedoria, formastes o homem para ser o senhor de todas as vossas criaturas, governar o mundo na santidade e na justiça, e proferir seu julgamento na retidão de sua alma, dai-me a sabedoria que partilha do vosso trono, e não me rejeiteis como indigno de ser um de vossos filhos.

Sou, com efeito, vosso servo e filho de vossa serva, um homem fraco, cuja existência é breve, incapaz de compreender vosso julgamento e vossas leis; porque qualquer homem, mesmo perfeito, entre os homens não será nada, se lhe faltar a sabedoria que vem de vós." (Sab. 9, 1-6).

"Mas, ao lado de vós, está a sabedoria que conhece vossas obras; ela estava presente quando fizestes o mundo, ela sabe o que vos é agradável, e o que se conforma às vossas ordens.

Fazei-a, pois, descer de vosso santo céu, e enviai-a do trono de vossa glória, para que, junto de mim, tome parte em meus trabalhos, e para que eu saiba o que vos agrada.

Com efeito, ela sabe e conhece todas as coisas, prudentemente guiará meus passos e me protegerá no brilho de sua glória." (Sab. 9, 9-11).

Ofereço este trabalho em honra e louvor a DEUS, pois sem a sua graça nada teria conseguido.

Senhor, eu vos ofereço todos os momentos de minha vida dedicados a este trabalho, porque senti a vossa presença me conduzindo a cada dia, nas preocupações e alegrias, nos erros e acertos. Obrigado por cada pessoa que colocastes em meu caminho e que, de alguma forma, me ajudaram.

Esta realização não é só minha, mas primeiramente vossa, que derramastes vossa sabedoria e me iluminastes até a conclusão deste trabalho.

GLÓRIA A DEUS!

AGRADECIMENTOS

Ao atingir um objetivo tão almejado, que foi a realização deste trabalho, gostaria de externar meus agradecimentos a todas as pessoas que me acompanharam e apoiaram no decorrer desse período importante de minha vida, assim como agradecer ainda a todas as instituições que me proporcionaram os meios para chegar até aqui.

Um agradecimento especial ao Prof . Dr. Yuzo Iano, meu orientador, pelo modo como conduziu-me, pela confiança, incentivo e amizade a mim dedicados.

Ao amigo Antonio Marcos de Lima Araújo, pela ajuda, sugestões e a presença amiga.

Gostaria de agradecer ao amigo Ayres Marden Almeida do Nascimento e a todos os que estiveram ao meu lado e contribuíram de alguma forma no desenvolvimento deste trabalho.

Aos colegas do departamento de Engenharia Elétrica da Universidade Federal do Pará pelo apoio e confiança que sempre depositaram em mim.

A Universidade Estadual de Campinas (UNICAMP) e a Universidade Federal do Pará (UFPA) por todo o apoio recebido dessas instituições durante a realização do meu curso de doutorado. Ressalto agradecimentos especiais a CAPES e a FAPESP.

A minha esposa Marise e meus filhos Thiago, Tatiana e Thaís, que sempre estiveram ao meu lado dando-me todo apoio, amor e carinho.

Aos meus pais Armando e Rosalé Pelaes que sempre me incentivaram.

Evaldo Gonçalves Pelaes

AGRADECIMENTOS ESPECIAIS AOS ÓRGÃOS DE APOIO E PESQUISA

Agradeço a **CAPES** - Coordenação de Aperfeiçoamento de Pessoal de Nível Superior que através da bolsa **CAPES/PICD,** viabilizou a minha estadia na UNICAMP durante o período de realização do programa de doutoramento.

Agradeço a **FAPESP** – Fundação de Amparo à Pesquisa do Estado de São Paulo que, através do financiamento do **Projeto "Compressão de Informação Visual Aplicada em Processamento Digital de Imagens no Contexto TV/ATV/JPEG/MPEG" , Processo FAPESP no. 93/3488-9,** possibilitou a aquisição de recursos e a existência de suportes computacionais necessários para a realização do presente trabalho.

RESUMO

Este trabalho apresenta uma nova ferramenta para codificação e/ou interpolação de imagens, que é a transformada seno discreta com rotação de eixos bidimensional (DSTr-2D), cuja principal característica é reduzir o efeito de bloqueamento, apresentando ainda uma baixa complexidade computacional. A DSTr-2D é aplicada para interpolação de imagem e seu desempenho é comparado com o obtido pela DCT. É ainda apresentado um outro método de interpolação usando DCT com eliminação de extrapolações. A DSTr-2D apresenta um melhor desempenho com relação aos outros métodos de interpolação, e essa melhoria cresce com o aumento da taxa de amostragem. A DSTr-2D é também aplicada na codificação de imagens, apresentando uma característica especial : a de reduzir o efeito de bloqueamento. Os resultados da codificação com a DSTr-2D são comparados com os obtidos pelos métodos de codificação por transformada como DCT e LOT. Nos processos de codificação são utilizadas quantização escalar uniforme, gaussiana e laplaciana de Lloid-Max. A DSTr-2D é também utilizada para realização simultânea de codificação e interpolação de imagens com reduzido efeito de bloco.

ABSTRACT

This work presents a new tool for image coding and image interpolation based on the two-dimensional discrete sine transform with axis rotation (DSTr-2D). The main characteristics of this transform are the few blocking effects found in the reconstructed image and its computational simplicity. In this work, the DSTr-2D is applied to image interpolation and its performance is compared with results obtained from DCT. A method of interpolation using DCT without extrapolation is also presented. The DSTr-2D achieves a performance better than the other interpolation methods used for comparison purposes. Indeed the better performance is obtained when the sample rate is increased. The DSTr-2D is also applied to the image coding, and it was verified that its advantage relative to DCT in terms of blocking effects in the reconstructed image. The results obtained in image coding using DSTr-2D are compared with results obtained from DCT and LOT. The images were coded using scalar, gaussian, laplacian and Lloid-Max quantization . Finally, the DSTr-2D is applied to performs the image coding and the image interpolation, simultaneously.

SUMÁRIO

Capítulo 1 Introdução 1

1.1. Conversão da Taxa de Amostragem, Decimação e Interpolação 1

1.2. Codificação Digital de Imagens 4

1.3. Contribuições e Organização deste Trabalho 6

Capítulo 2 Transformação por Blocos. 9

2.1. Introdução 9

2.2. Transformada de Fourier Discreta 10

2.3. Transformada Cosseno Discreta 11

2.4. Transformada Seno Discreta 19

2.5. LOT ("Lapped Orthogonal Transform") 24

2.6. Transformada Seno Discreta com pré e pós-Rotação de Eixos 29

2.7. Transformada Seno Discreta com Rotação de Eixos Bidimensional 34

Capítulo 3 Interpolação de Imagens por Transformadas 39

3.1. Amostragem Periódica de Sinais 39

3.2. Mudança da Taxa de Amostragem Usando Processamento Discreto 45

3.2.1. Redução da Taxa de Amostragem: Decimação 45

3.2.2. Aumento da Taxa de Amostragem: Interpolação 47

3.3. Amostragem de Sinais Bidimensionais 49

3.4. Interpolação de Sinais Bidimensionais 52

3.5. Interpolação de Sinais por Transformada 53

3.6. Implementação e Comparação dos Métodos de Interpolação por Transformada em Sinais Unidimensionais 57

3.7. Interpolação de Imagens por Transformada 62

3.8. Aplicação dos Métodos de Interpolação de Imagens por Transformada 65

Capítulo 4 Codificação de Imagens por Transformadas 72

4.1. Transformadas Usadas na Codificação de Imagens 73

4.2. Transformadas de Imagens por Blocos 76
4.3. A Quantização 81
4.4. Alocação de Bits e Quantização dos Coeficientes Transformados 88
4.5. Efeito de Bloqueamento em Imagens Codificadas e Reconstruídas 94

Capítulo 5 Codificação de Imagens por Transformadas Usando DCT, LOT e DSTr: Resultados Obtidos 96
5.1. Aplicação dos Métodos de Codificação a Imagem Lena 97
5.2. Aplicação dos Métodos de Codificação a Outras Imagens. 108
5.3. Codificação de Imagens Usando Decimação e Interpolação por DSTr. 115

Capítulo 6 Conclusões 122

Apêndice A 129
Apêndice B 135
Referências Bibliográficas 141

LISTA DE FIGURAS

1.1. Diagrama de blocos de um codificador de imagens por transformadas. 5

2.1. a) Exemplo de uma seqüência discreta $x[n]$ de comprimento $N = 5$ e b) Seqüência $y[n] = x[n]+x[2N-1-n]$ usada como passo intermediário para a definição da transformada coseno discreta de $x[n]$. 12

2.2. Seqüências periódicas a) $\tilde{x}[n]$ obtida pela repetição de $x[n]$ a cada N pontos; b) $\tilde{y}[n]$ obtida pela repetição da seqüência $y[n]$ a cada $2N$ pontos. 13

2.3. Funções bases para DCT para $N = 8$. 16

2.4. a) Exemplo de uma seqüência $x[n_1, n_2]$ e b) a correspondente seqüência $y[n_1, n_2] = x[n_1, n_2] + x[2N_1-1-n_1, n_2] + y[n_1, 2N_2-1-n_2] + x[2N_1-1-n_1, 2N_2-1-n_2]$, usada como passo intermediário para a obtenção da DCT-2D de $x[n_1, n_2]$. 18

2.5. a) Seqüência periódica $\tilde{x}[n_1,n_2]$ obtida pela repetição de $x[n_1, n_2]$; b) Seqüência periódica $\tilde{y}[n_1,n_2]$ obtida de $y[n_1, n_2]$. 18

2.6. a) Exemplo de uma seqüência discreta $x[n]$ com N = 5 e b) A seqüência $y[n]$ usada como passo intermediário para a definição da transformada seno discreta de $x[n]$. . . 20

2.7. Seqüência periódica $\tilde{x}[n]$ obtida pela repetição de $x[n]$ a cada N pontos. 21

2.8. Seqüência periódica $\tilde{y}[n]$ obtida pela repetição de $y[n]$ a cada $2N+2$ pontos. 21

2.9. Funções bases para DST com $N = 8$. 23

2.10. Sobreposição das funções bases na LOT. 25

2.11. Diagrama de blocos para processamento de sinais com a LOT para L = 2M. 26

2.12. Funções bases para LOT quando $M = 8$. 28

2.13. Seqüência original $x[n]$ e a reta $r[n]$ traçada entre os pontos $x[0]$ e $x[7]$ 30

2.14. Seqüência $xr[n] = x[n]-r[n]$ obtida pela pré-rotação de $x[n]$. 30

2.15. Seqüência estendida $y[n]$, usada como passo intermediário para determinar a DST de $xr[n]$. 30

2.16. Seqüência periódica $\tilde{y}[n]$, formada pela repetição da seqüência $y[n]$. 30

2.17. Exemplo de uma seqüência discreta longa $x[n]$ para aplicação da DSTr. 32

2.18. Seqüência discreta $x[n]$ agrupadas em blocos e as retas $r_1[n]$, $r_2[n]$, ... usadas no

procedimento de pré-rotação de eixos. 33
2.19. Seqüência discreta *xr[n]* gerada após o procedimento de pré-rotação de *x[n];* Observa-se que em geral, as amostras vão reduzindo seus valores ao se aproximarem das extremidades de cada bloco. 33
2.20. Seqüência estendida após a realização do procedimento de pré-rotação de eixos. . . 33
2.21. Exemplo de uma seqüência genérica *x[m,n]* agrupada em blocos de tamanho 4 x 4, usada para determinar a DSTr-2D. 36
2.22. A seqüência *x[m,n]* agrupada em blocos de tamanho 4 x 4, após o procedimento de pré-rotação. 37
2.23. Grade com as amostras originais resultante do procedimento de pré-rotação, sobre esta grade será aplicado também a pré-rotação. 37
2.24. Seqüência final após a junção da grade e os blocos processados pela pre-rotação, sobre a qual será aplicada a DST. 37
2.25. Aplicação da transformada seno discreta sobre um bloco obtido após os procedimentos de pré-rotação. 38
2.26. Funções bases usadas para transformação de blocos de tamanho 8 x 8. 38
3.1. Sistema de amostragem. a) Diagrama de blocos com dois estágios; b) trem de impulsos periódicos modulados por $x_c(t)$; c) seqüência discreta obtida na saída do sistema. 41
3.2. a) Espectro do sinal contínuo no tempo $x_c(t)$, com largura de faixa w; b) Espectro de um trem de impulsos; c) Espectro do sinal amostrado $x_a(t)$, com freqüência de amostragem $f_s>2w$. 42
3.3. Filtro de reconstituição; a) No domínio da freqüência; b) No domínio do tempo. . . 44
3.4. Diagrama de blocos de um sistema genérico usado para reduzir a taxa de amostragem por um fator M. 46
3.5. a) Espectro de um sinal $x_c(t)$ de faixa limitada; b) Espectro do sinal amostrado; c) Espectro do sinal decimado, que eqüivale ao sinal original amostrado com uma nova taxa de amostragem $f_s = 1/MT_s$. 47
3.6. Diagrama de blocos de um sistema genérico para aumentara a taxa de amostragem. .49
3.7. Interpolação no domínio da freqüência. a) Espectro do sinal contínuo original; b) Espectro do sinal amostrado com uma taxa T_s; c) Espectro de sinal expandido;

d) Espectro do sinal interpolado para taxa $Lf_s = L/T_s$. 49
3.8. Trem de impulsos periódicos bidimensional, espaçados de T_{s1} na direção x_1 e de T_{s2} na direção x_2. 51
3.9. Sinal de teste usado para interpolação por DSTr, DCT-I e DCT/FFT, com taxa de interpolação igual a 2. 58
3.10. Sinal de teste com 9 amostras 60
3.11. Gráfico da relação Sinal-Ruído de interpolação para o sinal de teste para taxas de amostragem f_N, $2f_N$, $4f_N$, $8f_N$ e $16f_N$, usando DSTr, DCT/FFT e DCT-I 61
3.12. Segundo sinal de teste usado para interpolação, com 16 amostras 62
3.13. Imagem de teste gerada por amostragem do sinal $s(x_1,x_2)$, com 256x256 amostras. . 66
3.14. Imagens obtidas por interpolação da imagem de teste de 128x128 para 255x255 amostras, usando a) DSTr-2D, b) DCTd, c) DCT/FFT, d) DCT-I. 67
3.15. Imagens de teste originais obtidas por amostragem a) 64x64 b) 505x505. 68
3.16. Imagens obtidas por interpolação da imagem de teste de 64x64 para 505x505 amostras, usando a) DSTr-2D, b) DCTd, c) DCT/FFT, d) DCT-I. 68
3.17. Imagem Lena original de tamanho 128x128. 70
3.18. Detalhe da imagem Lena original (128x128) 70
3.19. Detalhe da imagem interpolada por DSTr de (128x128) para (1017x1017). 71
3.20. Detalhe da imagem interpolada por DCTd de (128x128) para (1017x1017). 71
3.21. Detalhe da imagem interpolada por DCT/FFT de (128x128) para (1017x1017) . . . 71
3.22. Detalhe da imagem interpolada por DCT-I de (128x128) para (1017x1017). 71
4.1. Diagrama de blocos de um codificador de imagens por transformadas. 74
4.2. a) Imagem Lena original de tamanho 512x512 80
b) Leitura em zig-zag da variância dos coeficientes transformados da imagem Lena. . 80
4.3. Gráfico da variância dos coeficientes AC para a imagem Lena transformada (DCT) . 80
4.4. Gráfico da variância dos coeficientes AC para a imagem Lena transformados (LOT) . 80
4.5. Gráfico da variância dos coeficientes AC para a imagem Lena transformados (DSTr) 80
4.6. Gráfico da variância dos coeficientes AC para a imagem Lena transformados (DST) . 80
4.7. Ilustração da operação de quantização 82
4.8. Função de entrada-saída para um quantizador uniforme de 8 níveis 85
4.9. Erro de quantização para o quantizador de oito níveis 85

4.10. Função de entrada-saída para um quantizador ótimo de Lloid-Max de 8 níveis com distribuição laplaciana. 88
4.11. Procedimento de agrupamento dos coeficientes transformados 89
4.12. Tabelas de alocação de bits para se codificar a imagem Lena com uma taxa média de 1 bit/pixel para a) DCT; b) LOT; c) DSTr. 91
4.13. Imagem codificada com DCT e taxa de 1 bpp 92
4.14. Imagem codificada com LOT e taxa de 1 bpp 92
4.15. Imagem codificada com DCT e taxa de 1 bpp 92
4.16. Imagem codificada com DCT, alocação adaptativa e taxa de 1 bpp 94
4.17. Imagem codificada com LOT, alocação adaptativa e taxa de 1 bpp 94
4.18. Imagem codificada com DSTr, alocação adaptativa e taxa de 1 bpp 94
4.19. Detalhes da imagem original e codificada 95
5.1. Gráfico da Relação Sinal-RuídoxTaxa para a imagem Lena codificada com DCT, LOT e DSTr e quantização uniforme. 99
5.2. Gráfico da Relação Sinal-Ruído x Taxa para a imagem Lena codificada com DCT, LOT e DSTr e quantização laplaciana. 99
5.3. Gráfico da Relação Sinal-Ruído x Taxa para a imagem Lena codificada com DCT, LOT e DSTr e quantização gaussiana. 99
5.4. Imagens Lena original e codificadas, com taxa de 1 bpp, usando-se DCT, LOT e DSTr para quantização uniforme e blocos (8x8). 100
5.5. Imagens Lena original e codificadas, com taxa de 1 bpp, usando-se DCT, LOT e DSTr para quantização laplaciana e blocos (8x8). 101
5.6. Imagens Lena original e codificadas, com taxa de 1 bpp, usando-se DCT, LOT e DSTr para quantização laplaciana e blocos (8x8). 102
5.7. Detalhes (I e II) da imagem Lena original usados para comparar efeitos de blocos. . . 103
5.8. Detalhe I para as imagens codificadas com taxa 1 bpp usando-se DCT, LOT e DSTr e quantização uniforme para comparação de efeito de blocos 104
5.9. Detalhe I para as imagens codificadas com taxa 1 bpp usando-se DCT, LOT e DSTr e quantização laplaciana: comparação de efeito de blocos 104
5.10. Detalhe I para as imagens codificadas com taxa 1 bpp usando-se DCT, LOT e DSTr e quantização gaussiana para comparação de efeito de blocos 104

5.11. Detalhe II para as imagens codificadas com taxa 1 bpp usando-se DCT, LOT e DSTr e quantização uniforme para comparação de efeito de blocos 105
5.12. Detalhe II para as imagens codificadas com taxa 1 bpp usando-se DCT, LOT e DSTr e quantização laplaciana para comparação de efeito de blocos 105
5.13. Detalhe II para as imagens codificadas com taxa 1 bpp usando-se DCT, LOT e DSTr e quantização laplaciana para comparação de efeito de blocos 105
5.14. Detalhes da imagem Lena codificada com 0.2 bpp com a) DCT, b) LOT e DSTr. . 106
5.15. Detalhes das imagens codificadas com DCT, LOT e DSTr, com taxa de 0,2 bpp, com quantizador gaussiano e alocação de bits adaptativa com 4 classes de energia. . . .107
5.16. Detalhes das imagens codificadas com DCT, LOT e DSTr, com taxa de 0,5 bpp. . 107
5.17. Gráfico da SNRxTaxa para Lena codificada com DCT, LOT e DSTr com blocos de tamanho 16x16 e quantizador de Lloid-Max laplaciano. 108
5.18. a) Imagem Pahka original; b) gráfico da SNRxTaxa para a imagem codificada com DCT, LOT e DSTr, quantização laplaciana e blocos (8x8). 109
5.19. Imagem Maine original de tamanho 432x640 e detalhes da imagem original 110
5.20. Imagem Maine codificada com 1 bpp e quantização uniforme usando-se a) DCT, LOT e c) DSTr. 111
5.21. Imagem Maine codificada com 1 bpp e quantização laplaciana usando-se a) DCT, LOT e c) DSTr. 112
5.22. Imagem Maine codificada com 1 bpp e quantização gaussiana usando-se a) DCT, LOT e c) DSTr. 113
5.23. Imagem original "Fachada" (512x512) 114
5.24. Imagem original "Peppers" (512x512). 115
5.25. Diagrama de blocos de blocos de um sistema usando codificação e interpolação. . .116
5.26. Imagens Lena: a) original; b) codificada por decimação interpolação com taxa equivalente a 0,5 bpp e c) imagem codificada com DSTr com taxa de 0,5 bpp. . . .118
5.27. Detalhes das imagens a) original; a) codificadas por decimação e interpolação com DSTr e c) codificada com DSTr a 0,5 bpp, quantização laplaciana com 4 classes e blocos de 8x8. 119
5.28. Imagens: a) original; b) codificada por decimação interpolação com taxa equivalente a 0,25 bpp e c) imagem codificada com DSTr com taxa de 0,25 bpp. . 120

5.29. Detalhes das imagens originais, codificada por decimação interpolação e codificadas com DSTr, para taxa de 0,25 bpp, quantização laplaciana e blocos de tamanho 8x8. 121

LISTA DE TABELAS

3.1. Relação Sinal-Ruído e relação Sinal-Ruído de pico para os três métodos de interpolação de sinais unidimensionais usando DSTr, DCT/FFT e DCT-I, com taxa de interpolação igual a 2. 59

3.2. Relação Sinal-Ruído e relação Sinal-Ruído de pico para os três métodos de interpolação de sinais unidimensionais usando DSTr, DCT/FFT e DCT-I, com taxa de interpolação igual a 16. 60

3.3. Relação Sinal-Ruído e relação Sinal-Ruído de pico para os três métodos de interpolação de sinais unidimensionais usando DSTr, DCT/FFT e DCT-I, com taxa de interpolação igual a 16. 62

3.4. Comparação entre os métodos de interpolação para a imagem de teste para interpolar imagens de tamanho 128x128 para 255x255 e 256x256 para 511x511, usando DSTr, DCTd, DCT/FFT e DCT-I. 67

3.5. Comparação entre os métodos de interpolação, usando DSTr, DCTd, DCT/FFT e DCT-I para interpolar a imagem de teste de 64x64 para 505x505. 69

3.6. Relação Sinal-Ruído $(SNR)_{dB}$ entre a imagem Lena(512x512) original e interpolada usando DSTr, DCTd, DCT/FFT e DCT-I. 70

3.7. Relação Sinal-Ruído de pico $(PSNR)_{dB}$ entre a imagem Lena(512x512) original e interpolada usando DSTr, DCTd, DCT/FFT e DCT-I. 70

5.1. Relação Sinal-Ruído $(SNR)_{dB}$ e Relação Sinal Ruído de pico $(PSNR)_{dB}$ entre a imagem Lena512 original e codificada com taxas de 1 a 8 bpp, usando-se DCT, LOT e DSTr e blocos de tamanho 8x8. 98

5.2. SNR e PSNR para a imagem Lena512 codificada com 1 bpp usando-se DCT, LOT e DSTr, blocos de 8x8, com alocação adaptativa com 4 classes e quantização de

Lloi-Max laplaciana. 107
5.3. SNR e PSNR para a imagem Maine codificada com 1 bpp usando-se DCT, LOT e DSTr, com blocos de 8x8 e quantização uniforme , laplaciana e gaussiana. 110
5.4. SNR e PSNR para a imagem Maine codificada com 1 bpp usando-se DCT, LOT e DSTr, com blocos de 16x16 e quantização uniforme , laplaciana e gaussiana. . . . 110
5.5. SNR e PSNR para Imagem "fachada" codificada com 1 bpp usando-se DCT, LOT e DSTr, com blocos de 8x8 laplaciana. 114
5.6. SNR e PSNR para Imagem "peppers" codificada com 1bpp usando-se DCT, LOT e DSTr, com blocos de 8x8 laplaciana. 115
5.7. SNR e PSNR entre a imagem Lena (512x512) original e codificada por decimação interpolação (DSTr/DSTr) e codificada com DSTr para taxa de 0,5 bpp. 117
5.8. SNR e PSNR entre a imagem Lena (512x512) original e codificada por decimação interpolação (DSTr/DSTr) e codificada com DSTr para taxa de 0,25 bpp. 119

ABREVIATURAS

FFT: Transformada Rápida de Fourier ('Fast Fourier Transform").

DFT: Transformada de Fourier Discreta ("Discrete Fourier Transform").

DCT: Transformada Cosseno Discreta ("Discrete Cosine Transform").

DST: Transformada Seno Discreta ("Discrete Sine Transform").

LOT: "Lapped Orthogonal Transform".

DSTr: Transformada Seno Discreta com Rotação de Eixos.

SNR: Relação Sinal-Ruído.

PSNR: Relação Sinal-Ruído de Pico.

CAPÍTULO 1

INTRODUÇÃO

O termo processamento digital de imagens refere-se, geralmente ao tratamento, inclusive por computador, de imagens bidimensionais digitalizadas, que são matrizes de números reais ou imaginários representados por um número finito de bits. O processamento digital de imagens tem sido usado em diversas áreas comerciais e de pesquisas tais como, sensoriamento remoto via satélite que envolve a análise de imagens com a finalidade de se fazer mapeamento geográfico, verificar crescimento urbano e demográfico, desmatamento, queimadas, etc. Outras aplicações usuais se encontram na transmissão e armazenamento de imagens com aplicações comerciais, tais como televisão digital, teleconferência, fac-símile, redes de comunicação de computadores, digitalização de arquivos históricos, bibliotecas, processamento de imagens médicas, radar, sonar, robótica, inspeção industrial automatizada, enfim, tem-se uma infinidade de aplicações [1,4].

Nos últimos anos, devido, principalmente a sua grande aplicabilidade, tem crescido significativamente o aparecimento de novas técnicas e novos algoritmos nas diversas áreas que envolvem processamento de imagens, tais como, representação e modelamento de imagens, análise, restauração e textura de imagens, codificação, compressão e interpolação de imagens [1, 4, 20]. Esse trabalho se concentrará basicamente, em duas áreas que se referem a interpolação e codificação digital de imagens. Na codificação de imagens, o objetivo principal é procurar uma maneira de representar uma imagem com o menor número possível de bits, para armazenar ou transmitir, preservando um determinado nível de qualidade da imagem reconstituída. Em geral esses procedimentos são realizados em um domínio transformado. A interpolação de sinais e imagens está relacionada com a conversão da taxa de amostragem.

1.1. Conversão da Taxa de Amostragem, Decimação e Interpolação

A amostragem de sinais contínuos é um procedimento fundamental em processamento digital de sinais, que se constitui em representar e recuperar um sinal através de suas amostras discretas, seguindo determinados critérios [2, 3]. Existem vários tipos de amostragem discutidos na literatura [1,2,3], tais como amostragem uniforme e não uniforme, amostragem

uniforme com múltiplas funções, porém o tipo mais usado é a amostragem uniforme (periódica), na qual as amostras são uniformemente espaçadas de um período **T**, denominado de período de amostragem; seu inverso, **f =1/T**, é por definição a freqüência de amostragem . No entanto, para que uma função contínua $\mathbf{x_c(t)}$ seja inteiramente recuperada do correspondente sinal amostrado $\mathbf{x_a(t)}$, o período de amostragem **T** deve ser escolhido de modo que satisfaça os critérios estabelecidos por Nyquist, isto é, a freqüência de amostragem **f** deve ser maior ou igual a duas vezes a largura de banda do sinal a ser amostrado [1, 3]. No caso de imagens bidimensionais , a amostragem é feita tanto no espaço como em amplitude, correspondendo portanto a imagens discretas tanto no espaço quanto em amplitude. As coordenadas (x_1, x_2) representam o espaço amostrado e as amplitudes amostradas representam os níveis de cinza. Devido a largura faixa de uma imagem ser muito grande e de natureza bidimensional, a freqüência de amostragem (f_1, f_2), com $f_1 = 1/T_1$ e $f_2 = 1/T_2$, T_1 na abcissa ou horizontal e T_2 na ordenada ou vertical, torna-se bastante elevada uma vez que deve ser maior que duas vezes a largura de banda. Isso implica em uma grande quantidade de dados a ser manipulada quando se deseja armazenar uma imagem digitalizada. Essa é uma das razões porque se tem procurado técnicas que reduzam a quantidade de dados a serem armazenados.

Em alguns casos, um sinal ou uma imagem de entrada pode ainda ser amostrado segundo algum período de amostragem pré-determinado **T**, e em seguida convertido para um sinal amostrado com um novo período de amostragem **T'**, de modo que o sinal resultante corresponda à mesma função analógica. Em outros casos, pode ser mais eficiente ou conveniente trabalhar com diferentes freqüências de amostragem para um dado algoritmo de processamento de sinais, de modo que a taxa de amostragem do sinal deve ser convertida para diferentes taxas ao longo do processamento[2, 3].

O processo para se realizar a conversão da freqüência de amostragem $\mathbf{f_a = 1/T}$ para uma nova freqüência de amostragem $\mathbf{f_a' = 1/T'}$, é denominado de interpolação e decimação. O processo de conversão é chamado de interpolação quando a nova freqüência de amostragem é maior que a taxa original, isto é:

$$\mathbf{f_a' > f_a \quad ou \quad T' < T}$$

Isso corresponde a criação de novas amostras a partir de um conjunto reduzido de amostras. Quando a nova freqüência é menor do que a taxa original, isto é,

$$f_a' < f_a \quad \text{ou} \quad T' > T$$

Então o processo de conversão de amostragem é denominado de decimação. Nesse caso ocorre a eliminação de amostras.

Tanto o processo de decimação como o de interpolação podem ser implementados através de operações de filtragens lineares. Os procedimentos de decimação e interpolação encontram aplicabilidade nas áreas de sistemas de comunicações, sistemas de radar, processamento digital de sinais de voz, processamento digital de imagens, em aplicações biológicas como, por exemplo, a análise detalhada de algum trecho de uma imagem, ampliada através de algum método de interpolação.

Existem vários métodos de interpolação usados para imagens, tais como interpolação bilinear, cúbica e spline. Outros métodos utilizam transformadas discretas, os quais vem sendo continuamente aprimorados através de diversos estudos [2, 3, 6, 14, 15, 16, 17, 18, 19].

A implementação da transformada discreta de Fourier (**DFT**) através de algoritmos computacionalmente eficientes, como a transformada rápida de Fourier (**FFT**), tem sugerido a utilização dessa poderosa ferramenta na aplicação de interpolação. Além disso é cada vez maior o número de aplicações onde é viável utilizar processadores digitais de sinais (**DSP**) ou "chips" dedicados ao cálculo de freqüência [14].

Em 1973, **Schafer e Rabiner** [2], consideram a utilização da **DFT (FFT)** na interpolação de sinais. **Crochiere e Rabiner** [15] fazem em 1981 uma síntese dos algoritmos desenvolvidos. Em 1986, **Prasad e Satyanarayana** [3] aplicam esse método a seqüências de duração finitas. **Fraser** [14] sugere sutis modificações e observa melhorias no desempenho da interpolação através da **FFT**, em relação a linear e a cúbica, no tocante ao erro de interpolação. Além disso, **Fraser** verifica que o erro de interpolação é crescente do centro para os extremos da seqüência de amostra. Visando reduzir o erro médio quadrático na interpolação, **Fraser** propôs o janelamento do sinal e a eliminação das amostras dos extremos da seqüência. **Araujo** [21], em 1994, com o mesmo objetivo propõe a utilização de pré e pós-rotação de eixo na seqüência. Essa rotação, de pequena complexidade, produziu melhorias na interpolação com filtragem pela FFT. Em 1996, **Araujo** [22], apresentou um novo método para interpolação usando pré e pós-rotação de eixos com filtragem pela transformada seno

(**DSTr**). Este método trouxe melhorias em relação aos anteriores, no que se refere a complexidade computacional e ao desempenho. **Pelaes** em 97 [23], apresenta dois métodos de interpolação de imagens com eliminação de extrapolações, usando a transformada coseno discreta (DCT) e a transformada seno discreta com rotação de eixos (**DSTr**). Esses métodos são aplicados na interpolação de imagens neste trabalho e os resultados são comparados com imagens interpoladas por outros métodos, tais como bilinear e cúbica. Os procedimentos da transformada seno discreta com rotação de eixos (**DSTr**) são também utilizados neste trabalho como uma ferramenta para codificação de imagens em diferentes taxas. O algoritmo é implementado e os resultados de simulações são comparados com imagens codificadas com as transformadas **DCT** e **LOT**(Lapped Orthogonal Transform) [5, 12, 13].

1.2. Codificação Digital de Imagens

A codificação digital de imagens é um procedimento de representação de uma imagem como uma seqüência de bits para armazená-la ou transmiti-la através de um canal, com o objetivo principal de minimizar o número de bits necessários para representar a imagem com um certo critério de fidelidade, preservando os níveis de qualidade e inteligibilidade necessária para uma dada aplicação [1, 4, 20]. Isso se deve principalmente ao custo de armazenamento ou transmissão, que crescerá inevitavelmente com o aumento do número de bits. Existem muitas técnicas de codificação de imagens [4, 9].

Na transmissão, a principal meta é reduzir a largura da banda necessária para transmitir imagens digitalizadas, cujas aplicações incluem, televisão digital, televisão de alta definição (HDTV - High Definition Television), teleconferência, fac-símile, redes de computadores, etc. Em áreas que envolvem armazenamento de imagens, o objetivo da aplicação de codificação está relacionado principalmente com a redução do espaço de sinais ocupado por bancos de dados que guardam imagens digitalizadas, tais como imagens médicas, acervos históricos de bibliotecas, imagens de programas espaciais, imagens obtidas por satélite, etc. Os níveis de qualidade e inteligibilidade dependem da aplicação. Por exemplo, em aplicações que envolvem a digitalização de imagens espaciais e de acervos históricos, a imagem recuperada deve estar o mais próximo possível da imagem original. Já em aplicações como televisão digital, a alta qualidade da imagem é um fator importante. No entanto, muitas informações,

não perceptíveis ao olho humano, podem ser descartadas para o usuário final do receptor de vídeo, sem comprometer a qualidade da imagem final reconstruída. Portanto na codificação de imagens pode-se reduzir o número de bits para representá-las sem comprometer a qualidade da imagem reconstruída. A figura 1.1 mostra o diagrama de blocos de um sistema típico de codificação de imagens por transformadas.

Como é mostrado na figura 1.1, a imagem original é primeiramente transformada para um domínio mais apropriado a fim de se explorar suas características, tal como a correlação entre os pixels. Esse procedimento é realizado através de transformadas conhecidas na literatura, tais como transformada coseno discreta (DCT), transformada seno discreta (DST), lapped orthogonal transform (LOT), transformada Slan, transformada Karhunen-Loeve (KLT), transformada wavelet, etc [1, 4, 5, 6, 7, 9]. Em geral, os coeficientes transformados concentram grande parte da energia em alguns coeficientes. Isto significa que se pode recuperar uma imagem com uma boa qualidade, a partir de um número reduzido de coeficientes.

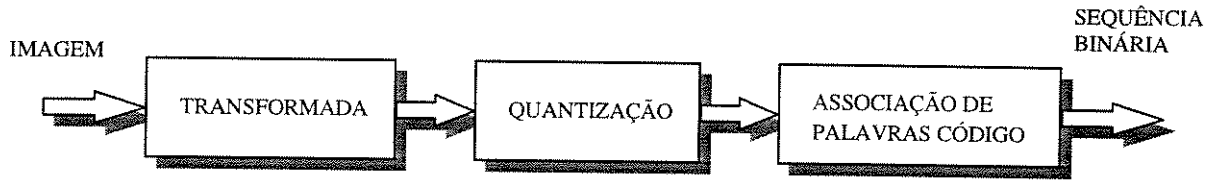


Figura 1.1. Diagrama de blocos de um codificador de imagens por transformadas.

O segundo bloco do diagrama da figura 1.1 realiza a quantização, que é um das operações básicas da codificação. A transformada extrai as redundâncias existentes entre os pixels vizinhos dentro de um bloco, cujo tamanho foi escolhido a priori por exemplo 8x8, gerando coeficientes transformados com diferentes variâncias. Pode-se, portanto, quantizar com um número menor de bits os coeficientes com menor variância, que é o ponto chave da codificação por transformadas. Com base nessas informações e com uma taxa de codificação previamente estabelecida, uma tabela de alocação de bits é gerada e os coeficientes transformados são quantizados de acordo com essa tabela. A quantização dos coeficientes pode ser feita usando-se quantização escalar ou vetorial. Na quantização escalar, cada

coeficiente transformado é independentemente representado pelo número de bits de acordo com a tabela de alocação. Na quantização vetorial, dois ou mais coeficientes são conjuntamente representados por um escalar pertencente a um conjunto de seqüências previamente estabelecidas denominado de dicionário.

O terceiro bloco do codificador de imagens da figura 1.1 faz o mapeamento do conjunto de coeficientes transformados e quantizados, a um conjunto de palavras-código de tamanho fixo ou variável. Quando é utilizada a codificação de entropia, o mapeamento dos coeficientes quantizados é feito utilizando-se, por exemplo, códigos de Huffman, Lempel-Ziv, Aritméticos, etc. Nesse caso, o número médio de bits por pixel, necessários para representar os coeficientes da imagem transformada, é reduzido e se aproxima da entropia.

Para reconstituir a imagem, é realizado o processo inverso, isto é, os coeficientes quantizados são reconstituídos a partir da seqüência de bits e a transformada inversa é aplicada sobre esses coeficientes, obtendo-se a imagem reconstruída, que é uma réplica da original e apresentando portanto algum erro devido a quantização e a codificação.

Na codificação de imagens por transformadas, as imagens são processadas em blocos. Devido a esse processamento independente, alguns erros de codificação produzem descontinuidades entre os blocos da imagem reconstruída, uma vez que as últimas amostras de um bloco não casam perfeitamente com as primeiras amostras dos blocos adjacentes. Esses erros aparecerão como efeitos de bloqueamento (ou efeito de blocos) visível na imagem reconstituída. Com a finalidade de reduzir o efeito de bloqueamento, vários trabalhos foram desenvolvidos, envolvendo pré e pós-filtragem [5], sendo que o mais importante foi desenvolvido por Malvar, a LOT ("Lapped Orthogonal Transform") [5, 12, 13]. A LOT utiliza o procedimento de sobreposição de blocos para minimizar o efeito de bloqueamento.

1.3. Contribuições e Organização deste Trabalho

Pretende-se neste trabalho apresentar dois métodos de interpolação de imagens por transformada com eliminação de extrapolações [3]. O primeiro método utiliza uma pré-rotação de eixos seguido da interpolação pela transformada seno discreta (DST) e posteriormente uma pós-rotação de eixos. Esse método é uma extensão bidimensional do método de interpolação desenvolvido por Araújo[22]. O outro método faz a interpolação

utilizando a transformada cosseno discreta (DCT) com uma simples modificação no processamento, utilizando princípios da DST. Os dois métodos são implementados e simulações são realizadas e comparadas com outros métodos de interpolação, tais como o método clássico usando DCT implementado através da FFT, e o método de interpolação pela DCT-I proposto por Wang [18, 19]. Todos os métodos de interpolação são implementados por blocos e é observado que, no método de interpolação clássico usando DCT/FFT, há o aparecimento de efeito de bloqueamento visível na imagem interpolada, enquanto nos outros dois métodos, DSTr e DCT com descarte, tal efeito não é observado. Esses dois métodos apresentam melhores desempenhos no tocante à relação sinal-ruído, quando comparados com os métodos de interpolação clássico por DCT e DCT-I. O fato da imagem interpolada por DSTr não apresentar efeito de bloqueamento visível, levou a usar, neste trabalho, os procedimentos da transformada seno discreta com pré e pós-rotação de eixos, na codificação de imagens. Os resultados obtidos por simulação mostram que a DSTr tem um bom desempenho para codificação de imagens, comparável ao desempenho da DCT, com a vantagem de não apresentar efeitos de bloqueamento visível em imagens reconstituídas em baixas taxas. Seu desempenho é também comparado com a LOT (Lapped Orthogonal Transform), verificando-se que, em muitos casos a DSTr apresenta desempenho superior à LOT.

Finalmente, tem-se um método de interpolação por DSTr que apresentou desempenho superior em relação aos outros métodos de interpolação de imagens por transformadas, tais como método implementado por DCT-I e o clássico método por DCT/FFT. Além disso, os mesmos procedimentos da DSTr são usados na codificação de imagens. Portanto, tem-se um mesmo algoritmo que apresenta um bom desempenho tanto para interpolação como para codificação de imagens, além de não apresentar efeitos de bloqueamento visíveis. Aproveitando essas vantagens, o algoritmo pode ser implementado de maneira integrada usando interpolação e codificação de imagens. Esse método integrado é implementado neste trabalho e os resultados de simulações apresentam bons resultados, obtendo-se ganhos em relação ao método de codificação que utiliza somente a DSTr. Além disso, as imagens reconstituídas apresentam-se, subjetivamente, com melhor qualidade.

Este trabalho apresenta no capítulo 2, uma descrição das Transformadas e suas propriedades, que serão utilizadas no decorrer deste trabalho, tais como a transformada

discreta de Fourier (DFT), transformada coseno discreta (DCT), transformada seno discreta (DST), lapped orthogonal transform (LOT) e a transformada seno discreta com rotação de eixo (DSTr). No capítulo 3, são apresentados os métodos de interpolação de imagens bilinear, cúbica, DCT clássico, e os métodos de interpolação com eliminação de extrapolações usando DCT e DSTr, além de resultados obtidos por simulação para diversas imagens. No capítulo 4, é analisada a aplicação das transformadas DCT, LOT e DSTr em imagens, suas propriedades, quantização e alocação de bits. No capítulo 5 são apresentados os resultados da codificação de diversas imagens usando DCT, LOT e DSTr. Apresenta ainda um algoritmo integrado usando codificação e interpolação por DSTr. Finalmente, são apresentadas no capítulo 6, as conclusões deste trabalho e as sugestões para o desenvolvimento de novas pesquisas na área de codificação de imagens usando a DSTr.

CAPÍTULO 2

Transformação por Blocos

2.1 Introdução

As transformadas têm sido intensamente usadas em processamento digital de sinais e processamento digitais de imagens. A aplicabilidade das mesmas tem crescido consideravelmente nos últimos anos e esse fato tem incentivado o aparecimento de novas técnicas de transformação [1, 5, 7, 10]. Em processamento de imagens, as imagens são geralmente divididas em blocos e a transformação é aplicada independentemente em cada bloco, bem como a quantização dos coeficientes transformados além de outros processamentos que venham a ser realizados no domínio transformado. Quando se trabalha com baixas taxas, esses processamentos independentes dão origem a efeitos de bloqueamento, que são descontinuidades geradas devido aos erros de quantização, que contribuem para que um bloco não fique perfeitamente casado com os blocos adjacentes [5].

A transformada por bloco de uma seqüência discreta $x[n]$, $n = 0, 1, 2, 3, \ldots, N-1$, de comprimento $N = KM$, K inteiro, é obtida agrupando-se a seqüência em blocos de comprimento M, e aplicando-se a transformada sobre cada bloco, independentemente, como explanado a seguir.

Definindo-se o vetor **x** como um bloco de comprimento M, tem-se

$$\mathbf{x} = [x(kM)\ x(kM+1) \ldots x(kM+M-1)]^T, \qquad k=0, 1, 2, \ldots, K-1 \tag{2.1}$$

onde T, indica a matriz transposta, uma vez que **x** está sendo definido como um vetor coluna e k representa a ordem do bloco.

A transformação direta de **x** é definida por:

$$\mathbf{X} = \mathbf{A}^T \mathbf{x} \tag{2.2}$$

onde vetor **X** é transformada de **x** e **A** é a matriz transformação, cujas colunas são formadas pelos vetores que compõem a base da transformada. Se os elementos de **A** são complexos, T

indica a transposta da matriz conjugada. Representando-se na forma de somatório:

$$X[k] = \sum_{n=0}^{M-1} a[n,k]x[n], \qquad k = 1, 2, 3, \ldots, M \tag{2.3}$$

onde os coeficientes *a[n,k]* representam as funções bases da transformada direta e *X[k]* a transformada da seqüência discreta *x[n]*. A transformação inversa é dada por:

$$\mathbf{x} = [\mathbf{A}^{T}]^{-1}\,\mathbf{X} \tag{2.4}$$

Essa equação é válida para qualquer matriz de transformação inversível. Se a transformação é ortogonal, que é o caso das transformadas tratadas ao longo deste trabalho, então

$$\mathbf{A}^{T} = \mathbf{A}^{-1} \tag{2.5}$$

Na forma de somatório, a transformada inversa é escrita como:

$$x[n] = \sum_{k=0}^{M-1} X[k]a^{*}[n,k], \qquad n = 1, 2, 3, \ldots, M. \tag{2.6}$$

2.2. Transformada de Fourier Discreta

A transformada de Fourier discreta (DFT) é uma representação, no domínio da freqüência, de uma seqüência discreta finita *x[n]* $n = 1, 2, 3, \ldots, N$, de comprimento *N*, é definida [1,4, 9, 10] por:

$$X[k] = \sum_{n=0}^{N-1} x[n]\exp(-j2kn\pi / N), \qquad k = 0, 1, 2, \ldots, N-1 \tag{2.7}$$

a transformada de Fourier discreta inversa é dada por:

$$x[n] = \frac{1}{N}\sum_{k=0}^{N-1} X[k]\exp(j2nk\pi / N), \qquad n = 0, 1, 2, ..., N\text{-}1 \tag{2.8}$$

A definição da DFT tem duas interpretações [4], a primeira é que ela corresponde à discretização da transformada de Fourier de *x[n]*, a segunda interpretação é que a DFT representa a seqüência *x[n]* como uma combinação linear de senóides complexas harmonicamente relacionadas.

A DFT é uma transformada separável e sua versão bidimensional corresponde à aplicação da transformada nas linhas e colunas, como é mostrado a seguir.

Considere uma seqüência discreta bidimensional $x[n_1,n_2]$, com $0 \le n_1 \le N_1 - 1$ e $0 \le n_2 \le N_2 - 1$. Sua transformada de Fourier discreta bidimensional (DFT-2D) é dada por:

$$X[k_1,k_2] = \sum_{n_1=0}^{N_1-1}\sum_{n_2=0}^{N_2-1} x[n_1,n_2]\exp(-j2k_1n_1\pi / N_1)\exp(-j2k_2n_2\pi / N_2), \tag{2.9}$$

onde, $k_1 = 0, 1, 2, ..., N_1\text{-}1$ $\qquad k_2 = 0, 1, 2, ..., N_2\text{-}1$

A transformada de Fourier discreta inversa é dada por:

$$x[n_1,n_2] = \frac{1}{N_1N_2}\sum_{k_1=0}^{N_1-1}\sum_{k_2=0}^{N_2-1} X[k_1,k_2]\exp(j2k_1n_1\pi / N_1)\exp(j2k_2n_2\pi / N_2) \tag{2.10}$$

onde, $n_1 = 0, 1, 2, ..., N_1\text{-}1$, $\qquad n_2 = 0, 1, 2, ..., N_2\text{-}1$

A partir da DFT, pode-se definir outras transformadas como a DCT e DST.

2.3. Transformada Cosseno Discreta

A DCT é a transformada mais utilizada em processamento de imagens que envolvem codificação por transformadas, tal como no caso dos padrões JPEG (Joint Photographic Experts Group) e MPEG (Moving Photographic Experts Group). Isso se deve, principalmente, a excelente compactação de energia para dados altamente correlacionados. É a transformada que mais se aproxima assintoticamente da transformada KLT (Karhunen-Loève Transform),

que é a transformada ótima do ponto de vista estatístico [4, 8, 10]. Pode ainda ser aplicada independentemente sobre os blocos de uma seqüência. A transformada coseno discreta (DCT) de uma seqüência unidimensional *x[n]*, pode ser derivada da DFT de uma seqüência discreta [1, 4, 10], obtida pela extensão da seqüência *x[n]*, por simetria par, como será apresentado a seguir.

Dada uma seqüência discreta *x[n]* de comprimento finito N, define-se uma seqüência *y[n]* de comprimento *2N*, com simetria par, através da relação [4]:

$$y[n] = x[n] + x[2N-1-n] = \begin{cases} x[n], & 0 \le n \le N-1 \\ x[2N-1-n], & N \le n \le 2N-1 \end{cases} \qquad (2.11)$$

A figura 2.1 mostra exemplos das seqüências *x[n]* e *y[n]* para o caso em que $N = 5$. A seqüência *y[n]* apresenta simetria em relação ao ponto $n = N - 0{,}5$. Quando uma seqüência periódica $\tilde{x}[n]$ é formada pela repetição de *x[n]* a cada *N* pontos, $\tilde{x}[n]$ apresenta descontinuidades artificiais, visto que o início e o final da seqüência *x[n]* se juntam no processo de repetição. Quando a seqüência periódica $\tilde{y}[n]$ é formada pela repetição da seqüência *y[n]* de comprimento *2N*, essas descontinuidades não ocorrem, como pode ser visto na figura 2.2. A eliminação dessas descontinuidades artificiais contribui para uma importante propriedade, que é muito explorada na codificação de imagens, a compactação de energia [4]. Esse fato contribui para que a DCT seja uma das transformadas mais utilizadas nos métodos de codificação e compressão de imagens.

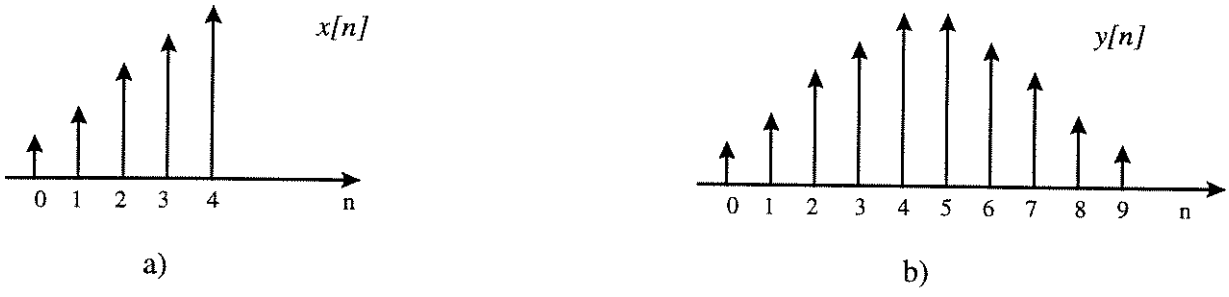
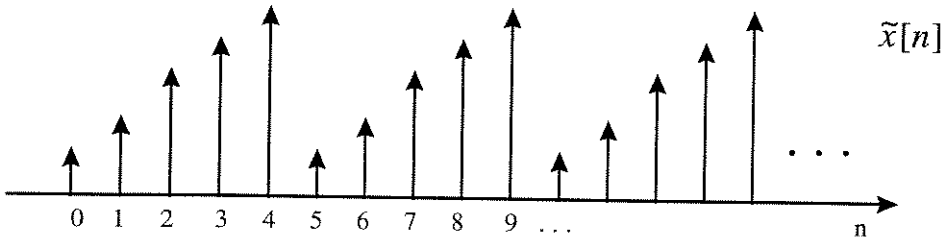
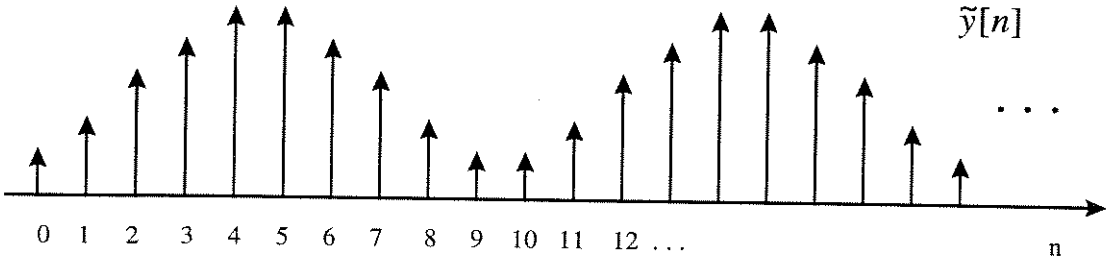


Figura 2.1. a) Exemplo de uma seqüência discreta *x[n]* de comprimento $N = 5$ e b) Seqüência *y[n] = x[n]+x[2N-1-n]* usada como passo intermediário para a definição da transformada coseno discreta de *x[n]*.



a)



b)

Figura 2.2. Seqüências periódicas a) $\tilde{x}[n]$ obtida pela repetição de $x[n]$ a cada N pontos; b) $\tilde{y}[n]$ obtida pela repetição da seqüência $y[n]$ a cada $2N$ pontos.

A DFT $Y[k]$ da seqüência estendida $y[n]$ é dada por:

$$Y[k] = \sum_{n=0}^{2N-1} y[n] W_{2N}^{kn}, \qquad 0 \le k \le 2N-1 \tag{2.12}$$

$$W_{2N} = \exp(-j2\pi / 2N) \tag{2.13}$$

Substituindo 2.11 em 2.12, obtém-se

$$Y[k] = \sum_{n=0}^{N-1} x[n] W_{2N}^{kn} + \sum_{n=N}^{2N-1} x(2N-1-n) W_{2N}^{nk}, \qquad 0 \le k \le 2N-1 \tag{2.14}$$

Fazendo-se a troca de variáveis e algumas operações algébricas, obtém-se:

$$Y[k] = W_{2N}^{-k/2} \sum_{n=0}^{N-1} 2x[n] \cos \frac{\pi}{2N} k(2n+1), \qquad 0 \le k \le 2N-1 \tag{2.15}$$

A DCT de N pontos de *x(n)*, é obtida de *Y[k]* por:

$$X[k] = \begin{cases} W_{2N}^{k/2} Y[k], & 0 \le k \le N-1 \\ 0, & para \quad outros \quad valores \end{cases} \tag{2.16}$$

Portanto, a DCT de *x[n]* é definida por:

$$X[k] = \sum_{n=0}^{N-1} 2x[n] \cos \frac{\pi}{2N} k(2n+1), \qquad 0 \le k \le N-1 \tag{2.17}$$

Para derivar a DCT inversa deve-se relacionar *X[k]* a *Y[k]*, *Y[k]* a *y[n]*, e finalmente, *y[n]* a *x[n]*, como será mostrado a seguir.

Apesar de *Y[k]* ser uma seqüência de *2N* pontos e *X[k]* uma seqüência de *N* pontos, redundâncias em *Y[k]* devido a simetria em *y[n]* nos permitem determinar *Y[k]* de *X[k]*. De (2.15) pode-se escrever:

$$Y[k] = \begin{cases} W_{2N}^{-k/2} Y[2N-k], & 0 \le k \le 2N-1 \\ 0, & k = N \end{cases} \tag{2.18}$$

De (2.16) e (2.18) obtém-se:

$$Y[k] = \begin{cases} W_{2N}^{-k/2} X[k], & 0 \le k \le N-1 \\ 0, & k = N \\ -W_{2N}^{-k/2} X[2N-k], & N+1 \le k \le 2N-1 \end{cases} \tag{2.19}$$

A seqüência *Y[k]* é relacionada a *y[n]* através da DFT inversa de *2N* pontos, isto é:

$$y[n] = \frac{1}{2N} \sum_{k=0}^{2N-1} Y[k] W_{2N}^{nk}, \qquad 0 \le n \le 2N-1 \tag{2.20}$$

Da definição de $y[n]$ em (2.11), pode-se recuperar $x[n]$ de $y[n]$ por:

$$x[n] = \begin{cases} y[n], & 0 \le n \le N-1 \\ 0, & \textit{para outros valores} \end{cases} \tag{2.21}$$

De (2.19), (2.20) e (2.21) obtém-se a transformada coseno discreta inversa:

$$x[n] = \begin{cases} \frac{1}{N}\left[\frac{X[0]}{2} + \sum_{k=1}^{N-1} X[k] \cos\frac{\pi}{2N} k(2n+1)\right], & 0 \le n \le N-1 \\ 0, & \textit{para outros valores} \end{cases} \tag{2.22}$$

A Equação 2.22 pode ainda ser expressa como:

$$x[n] = \sum_{k=0}^{N-1} w[k] X[k] \cos\frac{\pi}{2N} k(2n+1), \qquad 0 \le n \le N-1 \tag{2.23a}$$

onde:

$$w[k] = \begin{cases} \frac{1}{2}, & k = 0 \\ 1, & 1 \le k \le N-1 \end{cases} \tag{2.23b}$$

Tomando-se a equação (2.17), dividindo-a por $\sqrt{2N}$ e multiplicando (2.23) pelo mesmo fator $\sqrt{2N}$ obtém-se o par da transformada coseno discreta direta e inversa:

$$X[k] = \sqrt{\frac{2}{N}} \sum_{n=0}^{N-1} x[n] \cos\frac{\pi}{N} k\left(n + \frac{1}{2}\right), \qquad 0 \le k \le N-1 \tag{2.24a}$$

$$x[n] = \sqrt{\frac{2}{N}} \sum_{k=0}^{N-1} w[k] X[k] \cos\frac{\pi}{N} k\left(n + \frac{1}{2}\right), \qquad 0 \le n \le N-1 \tag{2.24b}$$

onde: $$w[k] = \begin{cases} \frac{1}{\sqrt{2}}, & k = 0 \\ 1, & 1 \le k \le N-1 \end{cases} \qquad (2.24c)$$

O par de transformada coseno discreta como definida em (2.24) é referida como DCT do tipo II em [4, 5, 6]. Essa é uma das formas mais utilizadas na prática [2]. A figura 2.3 mostra as funções bases para a DCT como definida em 2.24, para $N = 8$. Existem outras formas de apresentação da DCT as quais apresentam um deslocamento de fase igual para todas as funções da base. São definidas como DCT dos tipos I, III e IV [4, 5, 6].

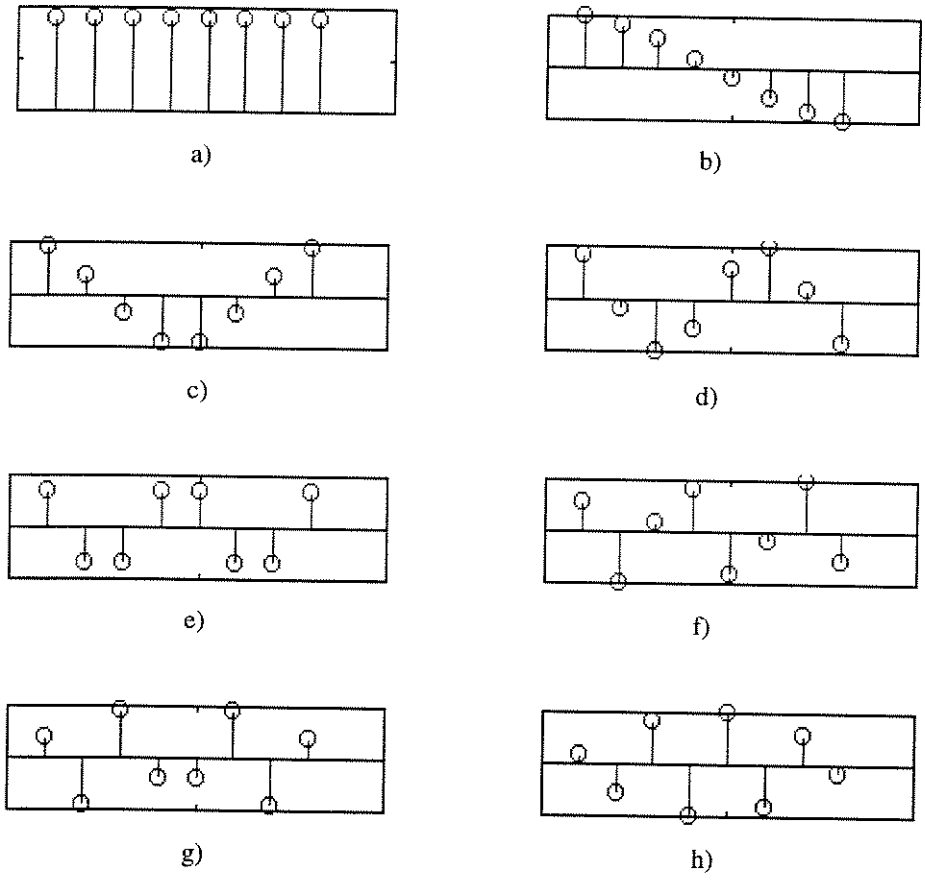


Figura 2.3. Funções bases para DCT para $N = 8$.

Para o caso bidimensional a transformada coseno discreta (DCT-2D) pode ser derivada diretamente da DCT unidimensional, como é mostrado a seguir:

Dada uma seqüência bidimensional $x[n_1,n_2]$, com $0 \le n_1 \le N_1 - 1$, $0 \le n_2 \le N_2 - 1$, deriva-se uma nova seqüência auxiliar $y[n_1, n_2]$ de tamanho $2N_1x2N_2$ e calcula-se $Y[k_1, k_2]$, a DFT-2D de $y[n_1, n_2]$ de tamanho $2N_1x2N_2$. Em seguida obtém-se a DCT-2D de $x[n_1, n_2]$, $X[k_1, k_2]$, a partir de $Y[k_1, k_2]$, com N_1xN_2 pontos.

$$\begin{array}{ccccccc} N_1xN_2 & & 2N_1x2N_2 & & 2N_1x2N_2 & & N_1xN_2 \\ x[n_1, n_2] & \Leftrightarrow & y[n_1, n_2] & \Leftrightarrow & Y[k_1, k_2] & \Leftrightarrow & X[k_1, k_2] \end{array}$$

A seqüência $x(n_1,n_2)$ está relacionada com a seqüência $y(n_1, n_2)$ através da equação:

$$y[n_1, n_2] = x[n_1, n_2] + x[2N_1 - 1 - n_1, n_2] + y[n_1, 2N_2 - 1 - n_2] + x[2N_1 - 1 - n_1, 2N_2 - 1 - n_2]. \quad (2.25)$$

A figura 2.4a mostra o exemplo de uma seqüência bidimensional $x[n_1, n_2]$, quando $N_1=3$ e $N_2 = 4$, e a figura 2.4b mostra a correspondente seqüência $y[n_1, n_2]$, usada como passo intermediário para a obtenção da transformada coseno discreta bidimensional (DCT-2D) da seqüência $x[n_1, n_2]$.

Uma seqüência periódica $\tilde{x}[n_1,n_2]$ com período N_1xN_2 pode ser obtida pela repetição da seqüência $x[n_1, n_2]$ como é mostrado na figura 2.5a. A seqüência periódica $\tilde{y}[n_1,n_2]$ com período $2N_1x2N_2$, é mostrada na figuras 2.5b. Pode-se observar que as descontinuidades geradas pela repetição de $x[n_1, n_2]$ para construir a seqüência periódica $\tilde{x}[n_1,n_2]$ não aparecem em $\tilde{y}[n_1,n_2]$. A eliminação dessas descontinuidades contribuirão para a importante propriedade de compactação de energia da DCT.

A DCT-2D da seqüência $x[n_1, n_2]$ é obtida como segue.

Aplica-se a DFT, sobre as linhas e colunas da seqüência bidimensional $y[n_1, n_2]$, para se obter a DFT-2D $Y[k_1, k_2]$, isto é:

$$Y[k_1, k_2] = DFT\{y[n_1, n_2]\} \quad (2.26)$$

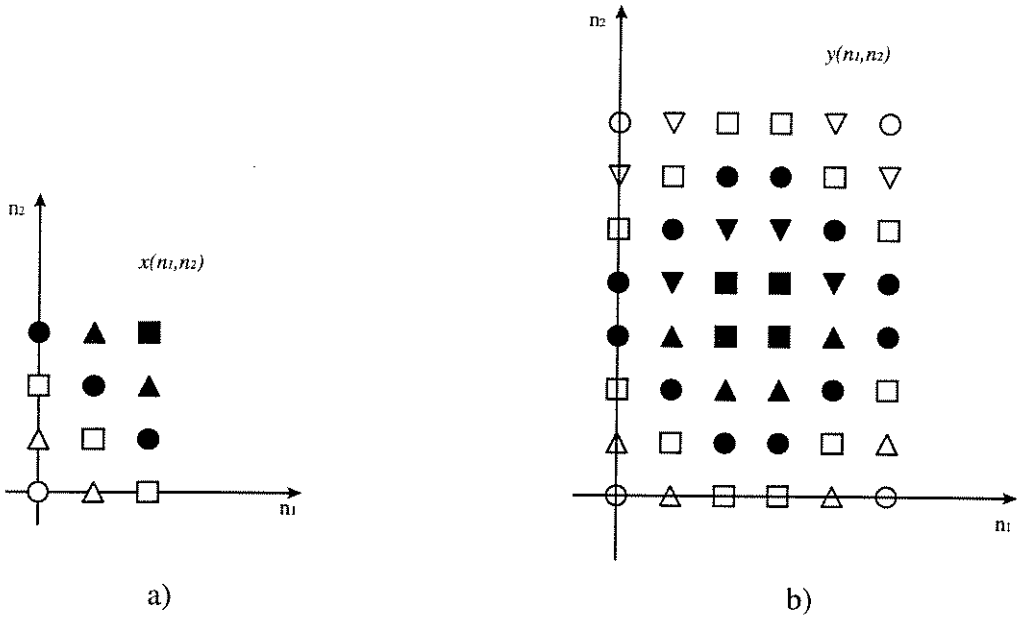


Figura 2.4. a) Exemplo de uma seqüência $x[n_1, n_2]$ e b) a correspondente seqüência $y[n_1, n_2] = x[n_1, n_2] + x[2N_1-1-n_1, n_2] + y[n_1, 2N_2-1-n_2] + x[2N_1-1-n_1, 2N_2-1-n_2]$, usada como passo intermediário para a obtenção da DCT-2D de $x[n_1, n_2]$.

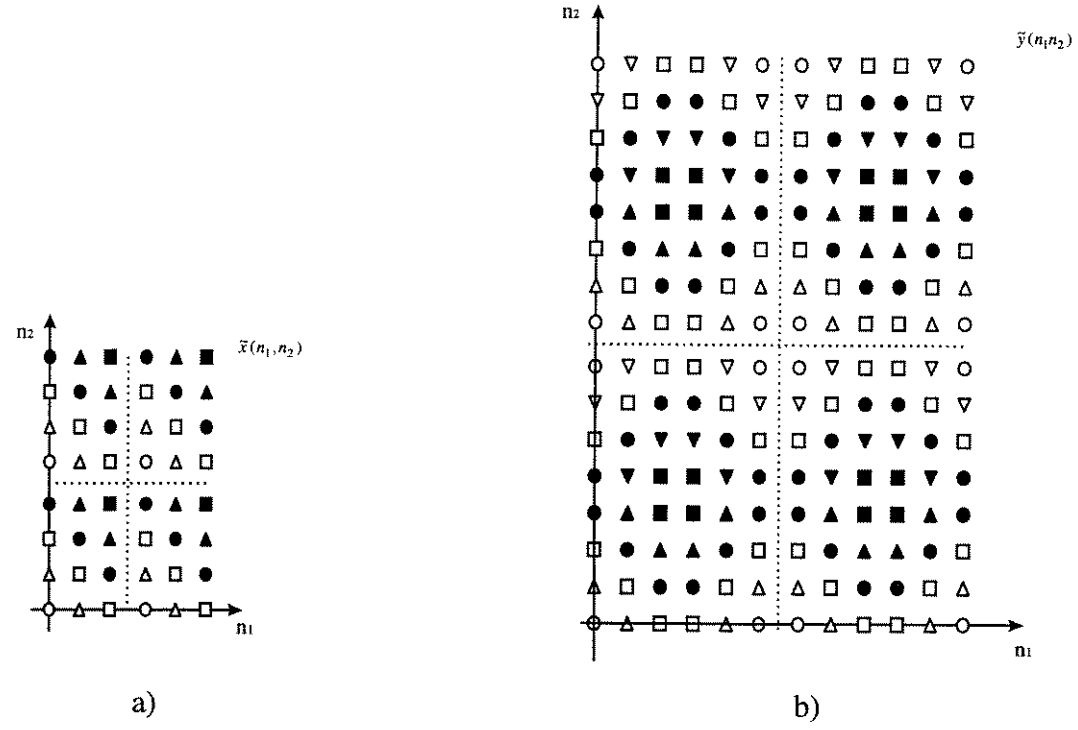


Figura 2.5. a) Seqüência periódica $\tilde{x}[n_1,n_2]$ obtida pela repetição de $x[n_1, n_2]$;

b) Seqüência periódica $\tilde{y}[n_1,n_2]$ obtida de $y[n_1, n_2]$.

A DCT-2D da seqüência $x[n_1, n_2]$ é obtida tomando-se N_1xN_2 pontos de $Y[k_1, k_2]$, isto é:

$$X[k_1,k_2]=\begin{cases} W_{2N_1}^{-k_1/2}W_{2N_2}^{-k_2/2}Y[k_1,k_2], & 0\le k_1\le N_1-1,\quad 0\le k_2\le N_2-1 \\ 0, & \textit{para outros valores} \end{cases} \tag{2.27}$$

Seguindo o mesmo procedimento para o caso unidimensional, obtém-se o par de transformadas para a DCT-2D para a seqüência $x(n_1, n_2)$:

$$X[k_1,k_2]=\frac{2}{\sqrt{N_1N_2}}\sum_{n_1=0}^{N_1-1}\sum_{n_2=0}^{N2-1}x[n_1,n_2]\cos\frac{\pi}{N_1}k_1(n_1+\frac{1}{2})\cos\frac{\pi}{N_2}k_2(n_2+\frac{1}{2}),$$

$$0\le k_1\le N_1-1,\ \ 0\le k_2\le N_2-1 \tag{2.28a}$$

$$x[n_1,n_2]=\frac{2}{\sqrt{N_1N_2}}\sum_{k_1=0}^{N_1-1}\sum_{k_2=0}^{N2-1}w_1(k_1)w_2(k_2)X[k_1,k_2]\cos\frac{\pi}{N_1}k_1(n_1+\frac{1}{2})\cos\frac{\pi}{N_2}k_2(n_2+\frac{1}{2}),$$

$$0\le n_1\le N_1-1,\ \ 0\le n_2\le N_2-1 \tag{2.28b}$$

onde:
$$w_1[k_1]=\begin{cases}\frac{1}{\sqrt{2}}, & k_1=0\\ 1, & 1\le k_1\le N_1-1\end{cases} \tag{2.28c}$$

$$w_2[k_2]=\begin{cases}\frac{1}{\sqrt{2}}, & k_2=0\\ 1, & 1\le k_2\le N_2-1\end{cases} \tag{2.28d}$$

2.4 . Transformada Seno Discreta

A transformada seno discreta (DST) é derivada da transformada discreta de Fourier. Os coeficientes transformados de uma seqüência discreta finita $x[n]$ são reais e estão relacionados com a DFT pela extensão antissimétrica de $x[n]$ [1]. Possui excelente propriedade de

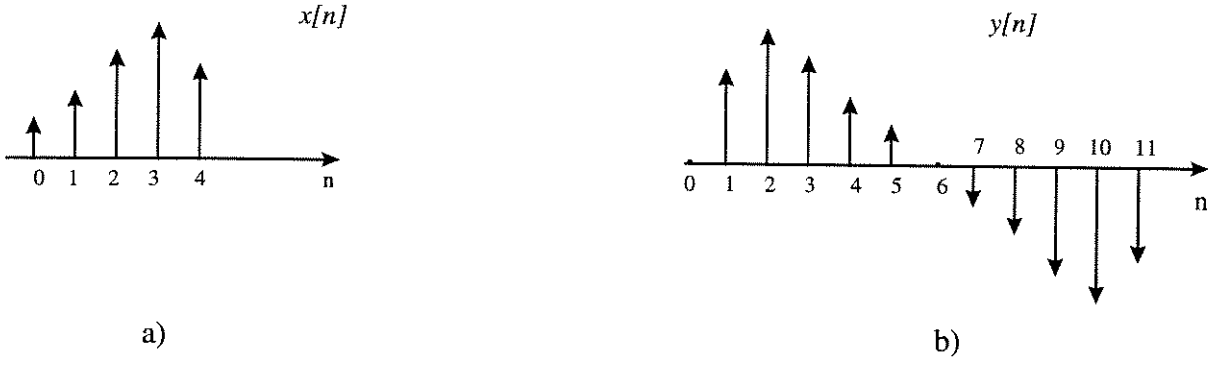
compactação de energia para codificação de imagens[1]. Pode também ser aplicada independentemente sobre os blocos de uma seqüência.

A DST de uma seqüência discreta unidimensional $x[n]$, figura 2.6a, pode portanto, ser derivada utilizando a DFT de uma seqüência intermediária com simetria ímpar, como é mostrado na figura 2.6b e sugerido em **Jain** [1]. A figura 2.7 mostra uma seqüência periódica $\tilde{x}[n]$ formada pela repetição de $x[n]$ a cada N pontos, e pode ser visto que $\tilde{x}[n]$ apresenta descontinuidades artificiais. Quando a seqüência periódica $\tilde{y}[n]$ é formada pela repetição da seqüência $y[n]$ de comprimento $2N$, essas descontinuidades são suavizadas, de acordo com a figura 2.8.

Dada uma seqüência discreta unidimensional $x[n] = (x_0\, x_1\, x_2 \ldots x_{N-1})$ de comprimento N, sua DST pode ser calculada com o auxílio da seqüência intermediária $y[n]$, $n = 1, 2, 3, \ldots, 2N+1$, de comprimento $2N+1$, como mostrado na figura 2.6b, e definida [1] por :

$$y[n]=(0 \; -x_{N-1} \ldots -x_2 \; -x_1 \; -x_0 \; 0 \; x_0 \; x_1 \; x_2 \ldots x_{N-1}). \tag{2.29}$$

$$y[n] = \begin{cases} 0 & , \quad n = 0 \\ -x[N-n] & , \quad 1 \le n \le N \\ 0 & , \quad n = N+1 \\ x[n-N-2] & , \quad N+2 \le n \le 2N+1 \end{cases} \tag{2.30}$$



a) b)

Figura 2.6. a) Exemplo de uma seqüência discreta $x[n]$ com N = 5 e b) A seqüência $y[n]$ usada como passo intermediário para a definição da transformada seno discreta de $x[n]$.

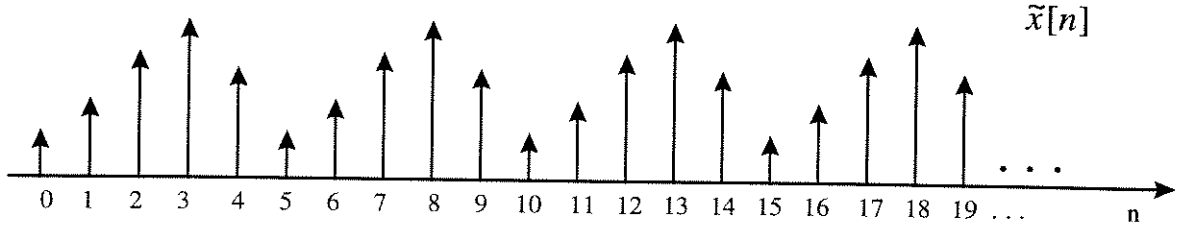


Figura 2.7. Seqüências periódicas $\tilde{x}[n]$ obtida pela repetição de $x[n]$ a cada N pontos.

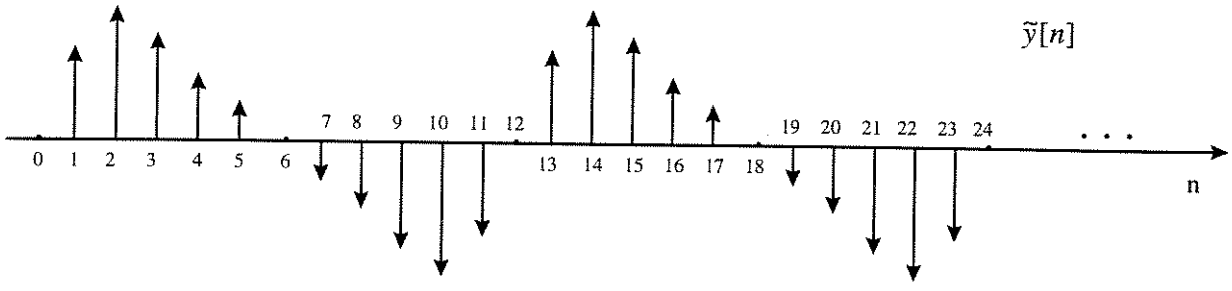


Figura 2.8. Seqüência periódica $\tilde{y}[n]$ obtida pela repetição de $y[n]$ a cada $2N+2$ pontos.

Tomando-se a DFT de $y[n]$, tem-se:

$$Y[k] = DFT(y[n]) = \sum_{n=0}^{2N+1} y[n] W_{2N+2}^{nk}, \qquad 0 \le k \le 2N+1 \tag{2.31}$$

$$Y[k] = \sum_{n=1}^{N} - x[N-n] W_{2N+2}^{nk} + \sum_{n=N+2}^{2N+1} x[n-N-2] W_{2N+2}^{nk}, \qquad 0 \le k \le 2N+1. \tag{2.32}$$

Após a troca de variáveis e algumas operações algébricas obtém-se:

$$Y[k] = \sum_{n=0}^{N-1} 2x[n] \operatorname{sen} \frac{\pi}{N+1} k(n+1), \qquad 0 \le k \le 2N+1 \tag{2.33}$$

A DST de *x[n]* é obtida, tomando-se *N* pontos de *Y[k]*, da forma:

$$X[k] = \begin{cases} Y[k] \ , & 1 \le k \le N \\ 0 \ , & \textit{para outros valores} \end{cases} \tag{2.34}$$

Portanto, a transformada seno discreta de uma seqüência *x[n]* pode ser calculada usando a equação:

$$X[k] = \sum_{n=0}^{N-1} 2x[n] \operatorname{sen} \frac{\pi}{N+1} (k+1)(n+1), \qquad 0 \le k \le N-1. \tag{2.35}$$

Para derivar a DST inversa deve-se relacionar *X[k]* a *Y[k]*, *Y[k]* a *y[n]*, e finalmente determina-se a transformada seno discreta inversa *x[n]* a partir de y*[n]*.
A transformada seno discreta inversa de uma seqüência *X[n]* é dada por:

$$x[n] = \frac{1}{N} \sum_{k=0}^{N-1} X[k] \operatorname{sen} \frac{\pi}{N+1} (k+1)(n+1), \qquad 0 \le n \le N-1. \tag{2.36}$$

Dividindo-se a equação 2.35 pelo fator $\sqrt{2N}$ e multiplicando-se a equação 2.36 pelo mesmo fator, obtém-se o par de transformadas DST para uma seqüência *x[n]* é dada por:

$$X[k] = \sqrt{\frac{2}{N}} \sum_{n=0}^{N-1} x[n] \operatorname{sen} \frac{\pi}{N+1} (k+1)(n+1), \qquad 0 \le k \le N-1. \tag{2.37a}$$

$$x[n] = \sqrt{\frac{2}{N}} \sum_{k=0}^{N-1} X[k] \operatorname{sen} \frac{\pi}{N+1} (k+1)(n+1), \qquad 0 \le n \le N-1. \tag{2.37b}$$

A figura 2.9. mostra as funções bases para a DST quando $N = 8$.

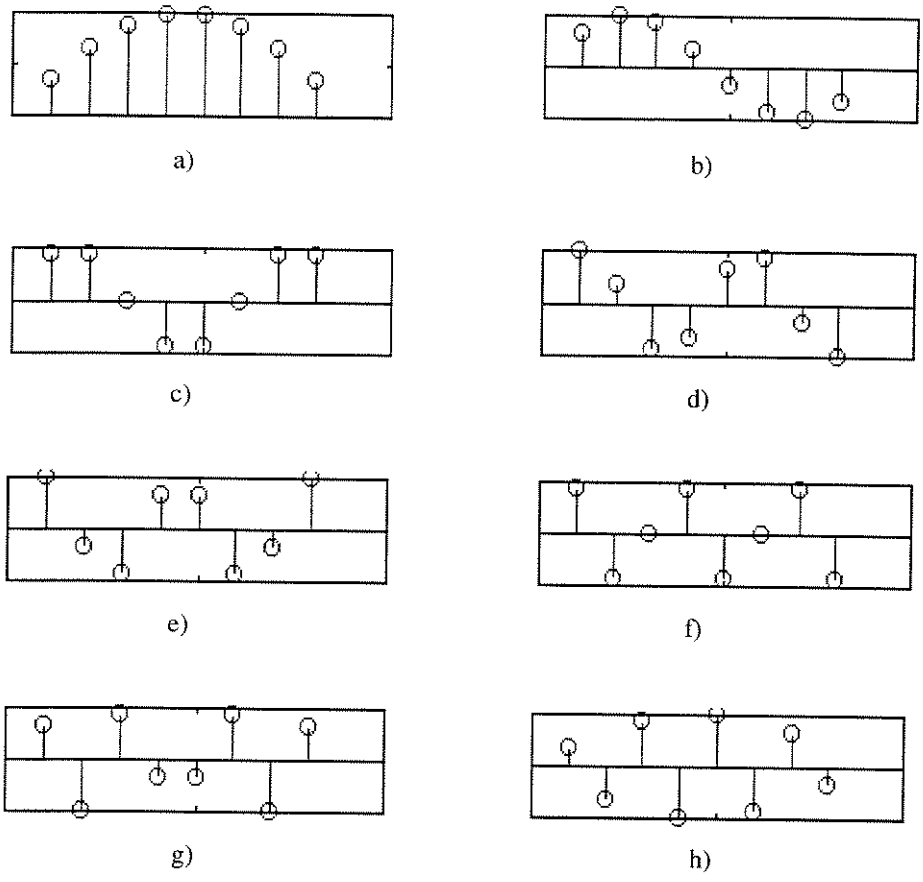


Figura 2.9. Funções bases para DST com $N = 8$.

Para o caso de uma seqüências bidimensional $x(m, n)$, com $0 \le n_1 \le N_1$ e $0 \le n_2 \le N_2$, sua transformada seno discreta é calculada aplicando-se o mesmo procedimento da DST unidimensional sobre as linhas e colunas, obtendo-se o par de transformadas:

$$X[k_1,k_2] = \frac{2}{N_1 N_2} \sum_{n_1=0}^{N_1-1} \sum_{n_2=0}^{N_2-1} 2x[n_1,n_2] \operatorname{sen} \frac{\pi(k_1+1)(n_1+1)}{N_1+1} \operatorname{sen} \frac{\pi(k_2+1)(n_2+1)}{N_2+1},$$

$$0 \le k_1 \le N_1 - 1, \quad 0 \le n_2 \le N_2 - 1 \qquad (2.38a)$$

$$x[n_1,n_2]=\frac{2}{N_1N_2}\sum_{k_1=0}^{N_1-1}\sum_{k_2=0}^{N_2-1}X[k_2,n_2]\,\text{sen}\frac{\pi(k_1+1)(n_1+1)}{N_1+1}\,\text{sen}\frac{\pi(k_2+1)(n_2+1)}{N_2+1},$$

$$0\le n_1\le N_1-1,\ \ 0\le n_2\le N_2-1 \qquad (2.38b)$$

2.5 . LOT ("Lapped Orthogonal Transform")

Na codificação de imagens por transformadas as imagens são processadas em blocos. Devido a esse processamento independente, alguns erros de codificação produzem descontinuidades entre os blocos, pois as últimas amostras de um bloco não casam perfeitamente com as primeiras amostras dos blocos adjacentes. Esses erros aparecerão como efeitos de bloqueamento visível na imagem reconstituída. Com a finalidade de reduzir o efeito de bloqueamento foi desenvolvida a transformada LOT ("Lapped Orthogonal Transform") [5, 12, 13], que utiliza o princípio da superposição de blocos para reduzir o efeito de bloqueamento na imagem reconstituída. Por essa razão a LOT não pode ser aplicada em uma seqüência discreta *x[n]* constituída de um único bloco, sendo necessário que a seqüência seja dividida em *m* blocos ($m \geq 2$) para que seja aplicada a LOT.

O desenvolvimento da LOT começou questionando se as funções da base de uma transformada poderiam ser maiores que o comprimento da transformada. Nesse caso as bases deveriam ter transições suavizadas que se aproximariam de zero nas extremidades. Dessa forma, se o número de coeficientes transformados deve ser *M*, as funções da base teriam comprimento maior que *M*, e assim essas funções da base seriam projetadas também sobre os blocos vizinhos. Assim, as funções da base de um bloco e de seus vizinhos, de ambos os lados, devem se sobrepor, como pode ser visto na figura 2.10. Para que sejam usadas as mesmas funções bases para análise (transformada direta) e síntese (transformada inversa), a porção da região de sobreposição com o bloco vizinho deve ser ortogonal.

Para obter um bloco transformado com *M* coeficientes, a LOT utiliza blocos de entrada de comprimento igual a *2M*, usando um procedimento de sobreposição de blocos. Assim, cada amostra do bloco de entrada é utilizada duas vezes pela transformada.

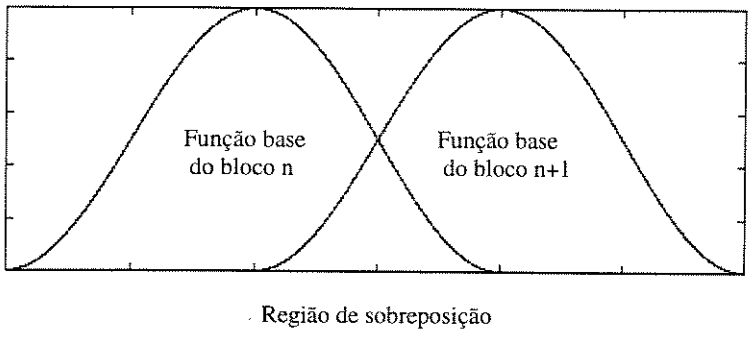


Figura 2.10. Sobreposição das funções bases na LOT.

A LOT **X** de um bloco **x** é dada por:

$$\mathbf{X} = \mathbf{P}^{T}\,\mathbf{x} \tag{2.39}$$

onde **P** é a matriz LOT ($2M \times M$), cujas colunas são formadas pelas funções bases e **x** é um bloco estendido de comprimento $2M$, de uma seqüência $x[n]$, dado por:

$$\mathbf{x} = [\, x(mM - 2M + 1)\;\; x(mM - 2M + 2)\;\; \ldots\;\; x(mM - 1)\;\; x(mM)\,]^{T} \tag{2.40}$$

onde m é o índice do bloco. Assim a LOT mapeia um bloco com $L = 2M$ amostras em um bloco de M coeficientes transformados, onde L é o comprimento das funções bases. Como a taxa de amostragem original deve ser mantida, a seqüência de entrada $x[n]$ deve ser deslocada de M amostras a cada bloco transformado, de modo que a mesma amostra será usada em dois blocos, havendo, portanto, uma sobreposição de L-M amostras para cada bloco. Se os coeficientes permanecerem inalterados a transformada inversa é dada por:

$$\mathbf{x} = \mathbf{P}\,\mathbf{X} \tag{2.41}$$

onde **P** é o operador transformada inversa. Agora a transformada inversa mapeia M coeficientes transformados em L amostras reconstituídas, que certamente serão diferentes das amostras originais, pois somente após a sobreposição dos blocos adjacentes, que serão

adicionados na região de sobreposição, é que o bloco original será recuperado. Por essa razão é que não é possível usar a LOT para transformações com um único bloco.

Esses procedimentos podem ser visualizados na figura 2.11, onde é mostrada a obtenção da transformada LOT direta e inversa de uma seqüência de entrada agrupada em blocos. As bases LOT são as colunas da matriz **P** de dimensão *LxM*. $\mathbf{P}^T$ é o operador transformada direta e **P** é o operador transformada inversa. A figura mostra a transformada LOT direta sendo aplicada sobre um bloco de tamanho $L = 2M$ para produzir um bloco de M coeficientes transformados. A transformada LOT inversa reconstitui um bloco de tamanho L a partir de M coeficientes transformados, com $L - M$ amostras sobrepostas para cada lado. Pode ser visto ainda que a seqüência reconstituída é formada pela sobreposição de blocos adjacentes. Para se obter uma reconstrução perfeita com a transformação LOT , é necessário que a seqüência original de entrada seja infinita. A LOT pode ser também implementada com $L > 2M$.

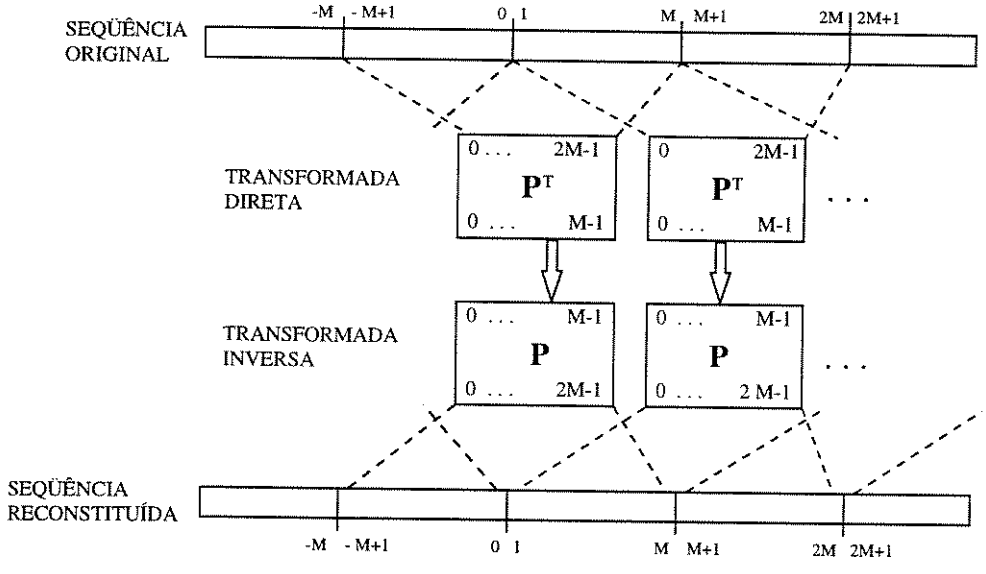


Figura 2.11. Processamento para a LOT com L = 2M.

A matriz **P** tem dimensão L x M e suas colunas são formadas pelas funções bases LOT as quais são obtidas através de algoritmos de otimização recursivos. A seguir é apresentado

um dos algoritmos para obtenção da matriz **P** de tamanho *LxM*, com $L = 2M$ e ganho de codificação máximo conforme é descrito em [4].

Passo1. Defina uma matriz LOT $\mathbf{P}_0$ por:

$$\mathbf{P}_0 = \frac{1}{2}\begin{bmatrix} \mathbf{D}_e - \mathbf{D}_0 & \mathbf{D}_e - \mathbf{D}_0 \\ \mathbf{J}(\mathbf{D}_e - \mathbf{D}_0) & -\mathbf{J}(\mathbf{D}_e - \mathbf{D}_0) \end{bmatrix} \tag{2.42}$$

onde $\mathbf{D}_e$ e $\mathbf{D}_0$ são matrizes de dimensão $M \times M/2$, contendo respectivamente, funções bases DCT com simetrias pares e ímpares e **J** é a matriz de dimensão *MxM*:

$$\mathbf{J} = \begin{bmatrix} 0 & \dots & 0 & 1 \\ 0 & \dots & 1 & 0 \\ \dots & \dots & \dots & \dots \\ 1 & 0 & 0 & 0 \end{bmatrix} \tag{2.43}$$

Passo 2. Considere a matriz de covariância $\mathbf{R}_{xx}$ de um dado sinal e defina uma nova matriz $\mathbf{R}_0$ definida por:

$$\mathbf{R}_0 \equiv \mathbf{P}_0^T \mathbf{R}_{xx} \mathbf{P}_0 \tag{2.44}$$

Como a LOT nesse ponto, depende de um suposto espectro do sinal a ser transformado, pois necessita do conhecimento da matriz de covariância $\mathbf{R}_{xx}$, tem sido usado então como modelo, um processo de Markov de primeira ordem com coeficiente de correlação $\rho = 0{,}95$, que é um bom modelo para imagem [4]. No entanto, o algoritmo não é muito sensível ao espectro desse suposto sinal.

Passo 3. Determine os autovetores de $\mathbf{R}_0$ e defina uma matriz **Z**, cujas colunas são os autovetores de $\mathbf{R}_0$. A matriz LOT é então, dada por:

$$\mathbf{P} = \mathbf{P}_0 \mathbf{Z} \tag{2.45}$$

A figura 2.12 mostra as funções bases LOT, que são as colunas da matriz **P**, para $M = 8$, obtidas pelo algoritmo acima descrito. Observe que as funções bases tem cada uma 16 amostras, e ainda, cada função base tem uma característica comum, isto é, apresentam as extremidades amortecidas.

Nas aplicações que envolvem transformações de seqüências bidimensionais e de comprimento finito, como em processamento digital de imagens, a LOT é aplicada nas linhas e colunas da imagem, observando-se que neste caso, por se tratar de seqüências de comprimento finito, uma modificação deve ser introduzida para a obtenção da LOT do primeiro e do último bloco de cada linha ou coluna, devido ao procedimento de sobreposição de blocos. Assim as funções bases para esses blocos serão menores, como será visto no capítulo 4, onde será apresentada a codificação de imagem por transformadas.

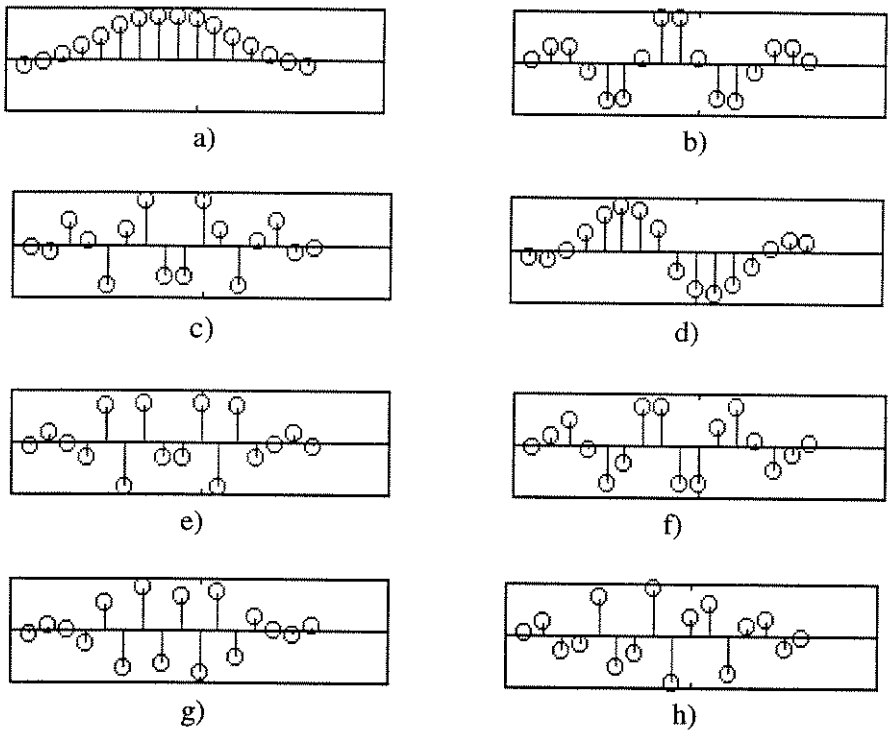


Figura 2.12. Funções bases para LOT quando $M = 8$.

2.6. Transformada Seno Discreta com pré e pós-Rotação de Eixos

Neste item será apresentada uma variação da transformada seno discreta; esta variação consiste em realizar um procedimento de pré-rotação de eixo na seqüência antes de ser aplicada a transformada seno discreta. Um procedimento de pós-rotação é realizado na seqüência após a aplicação da transformada inversa, para se recuperar a seqüência original. O procedimento de pré-rotação é realizado da forma a seguir:

Dada uma seqüência discreta $x[n] = [x_0\ x_1\ x_2\ x_3\ \ldots\ x_{N-1}\ ]$ de comprimento N, o procedimento de pré-rotação é definido como a diferença entre a seqüência $x[n]$ e a reta $r[n]$ que passa pelos pontos $x[0] = x_0$ e $x[N-1] = x_{N-1}$ da seqüência $x[n]$.

Definindo $xr[n] = [xr_0\ xr_1\ xr_2\ \ldots\ xr_{N-1}\ ]$, $n = 1, 2, 3, \ldots, N$, como a seqüência obtida pela pré-rotação de eixos, tem-se:

$$xr[n] = x[n] - r[n]. \tag{2.46}$$

onde,

$$r[n] = \frac{x[N-1] - x[0]}{N-1}\ n + x[0]. \tag{2.47}$$

A figura 2.13 mostra o exemplo de uma seqüência discreta de comprimento $N = 8$ e a reta traçada entre as amostras $x[0]$ e $x[7]$. A figura 2.14 mostra a seqüência $xr[n]$ obtida pela pré-rotação de eixo da seqüência original $x[n]$. Observa-se que os valores de $xr[0]$ e $xr[7]$ na seqüência $xr[n]$ da figura 2.14 são nulos nessas posições. Os correspondentes valores originais, isto é, os valores das amostras $x[0]$ e $x[7]$ são armazenados para serem utilizados no procedimento de pós-rotação, que é o processo inverso, usado para se recuperar a seqüência original $x[n]$. O procedimento de pré-rotação cria um amortecimento nas extremidades da seqüência até se anular. Esse amortecimento, juntamente com a aplicação da transformada seno discreta, irá contribuir para a redução de efeitos de bloqueamento quando aplicados na codificação de imagens , como será visto no capítulo 4. A transformada seno discreta é portanto, aplicada sobre a seqüência $xr[n]$, e será apresentada a seguir.

Tomando a seqüência $xr[n]$, define-se uma seqüência estendida $y[n]$ como mostrada na figura 2.15, que servirá de passo intermediário para a obtenção da transformada seno discreta

de *xr[n]*. A figura 2.16 mostra uma seqüência periódica formada pela repetição dessa seqüência. A seqüência *y[n]* é definida por:

$$y[n] = \begin{cases} xr[n], & 0 \le n \le N-2 \\ -xr[2N-2-n], & N-1 \le n \le 2N-3 \end{cases} \qquad (2.48)$$

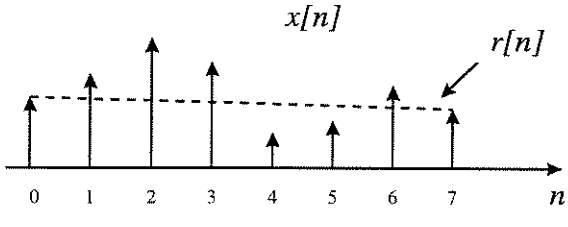


Figura 2.13. Seqüência original *x[n]* e a reta *r[n]* traçada entre os pontos *x[0]* e *x[7]*.

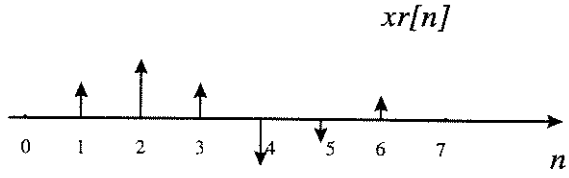


Figura 2.14. Seqüência *xr[n] = x[n] - r[n]*, obtida pela pré-rotação da seqüência *x[n]*.

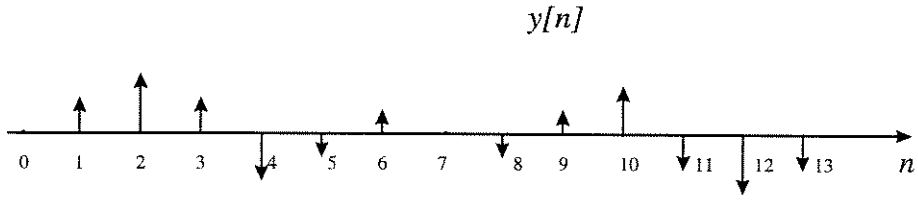


Figura 2.15. Seqüência estendida y*[n]*, usada como passo intermediário para determinar a DST de *xr[n]*.

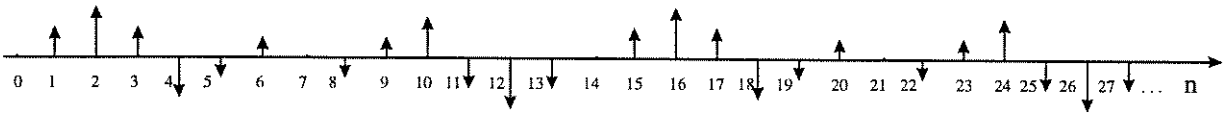


Figura 2.16. Seqüência periódica $\tilde{y}[n]$, formada pela repetição da seqüência *y[n]*.

Aplicando-se a DFT sobre a seqüência *y[n]*, obtém-se:

$$Y[k] = \sum_{n=0}^{2N-3} y[n] W_{2N-2}^{nk}, \qquad 0 \le k \le 2N-3 \tag{2.49}$$

$$Y[k] = \sum_{n=0}^{N-2} xr[n] W_{2N-2}^{nk} + \sum_{n=N-1}^{2N-3} - xr[2N-2-n] W_{2N-2}^{nk}, \qquad 0 \le k \le 2N-3 \tag{2.50}$$

Após a troca de variáveis e algumas operações algébricas obtém-se:

$$Y[k] = \sum_{n=0}^{N-2} 2x[n] \operatorname{sen} \frac{\pi}{N-1} kn, \qquad 0 \le k \le 2N-3 \tag{2.51}$$

A DST de *xr[n]* é obtida, tomando-se *N-1* pontos de *Y[k]*, da forma:

$$XR[k] = \begin{cases} Y[k] & , \quad 0 \le k \le N-1 \\ 0 & , \quad para \quad outros \quad valores \end{cases} \tag{2.52}$$

Considerando ainda que o coeficiente transformado *XR[0]* é sempre nulo, a transformada seno discreta da seqüência *xr[n]* pode ser calculada usando-se a equação:

$$XR[k] = \sum_{n=1}^{N-2} 2xr[n] \operatorname{sen} \frac{\pi}{N-1} kn, \qquad 1 \le k \le N-2. \tag{2.53}$$

A transformada seno discreta é portanto aplicada sobre os *N-2* pontos da seqüência *xr[n]*, uma vez que os pontos *xr[0]* e *xr[N-1]*, sendo nulos, produzirão coeficientes nulos nestas posições . Para cada um desses pontos, existe uma amostra na forma original, que deve ser transmitida ou armazenada nessas posições, pois ela será usada no procedimento de pós- rotação para recuperar a seqüência original. O par de transformadas DST para a seqüência *xr[n]*, é portanto dado por:

$$XR[k] = \sqrt{\frac{2}{N-1}} \sum_{n=1}^{N-2} xr[n] \operatorname{sen} \frac{\pi}{N-1} kn, \qquad 1 \le k \le N-2. \tag{2.54a}$$

$$xr[n] = \sqrt{\frac{2}{N-1}} \sum_{k=1}^{N-2} XR[k] \operatorname{sen} \frac{\pi}{N-1} kn, \qquad 1 \le n \le N-2 \tag{2.54b}$$

Quando a seqüência discreta *x[n]* é muito longa, ela é dividida em blocos, e os procedimentos de pré-rotação e a transformada seno discreta são aplicados em cada bloco. A figura 2.17 mostra um exemplo de uma longa seqüência discreta *x[n]*. Essa seqüência é dividida em blocos de tamanho $M = 8$, e uma reta *r[n]* com $M+1$ pontos, é traçada entre a primeira amostra de um bloco qualquer e a primeira amostra do bloco seguinte. A figura 2.18 mostra a seqüência *x[n]* dividida em blocos e as retas $r_1[n]$, $r_2[n]$, ... usadas no procedimento de pré-rotação de eixos. A figura 2.19 mostra a seqüência *xr[n]* gerada após o procedimento de pré-rotação de eixos. Note que nas transições entre blocos as amostras são sempre nulas e, em geral, as amostras vão se aproximando de zero nas proximidades dos extremos de cada bloco, que é uma conseqüência do procedimento de pré-rotação de eixos. Observa-se ainda que cada amostra localizada na extremidades de um bloco é usada duas vêzes no procedimento de pré-rotação. Isso contribui para a redução de efeitos de bloqueamento quando esses procedimentos de pré rotação de eixos, juntamente com a aplicação da transformada seno discreta, são usados na codificação de imagens, como será visto nos capítulos 4 e 5. A figura 2.20 mostra uma seqüência periódica formada pela repetição de blocos após a pré-rotação.

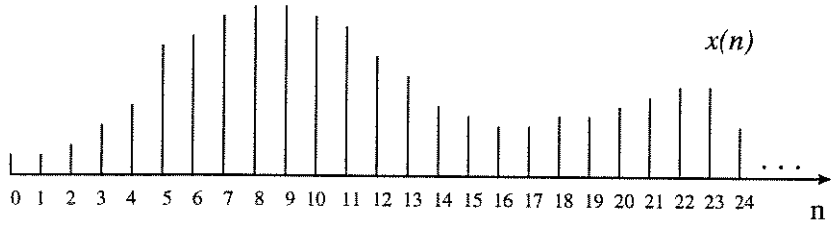


Figura 2.17. Exemplo de uma seqüência discreta *x[n]*.

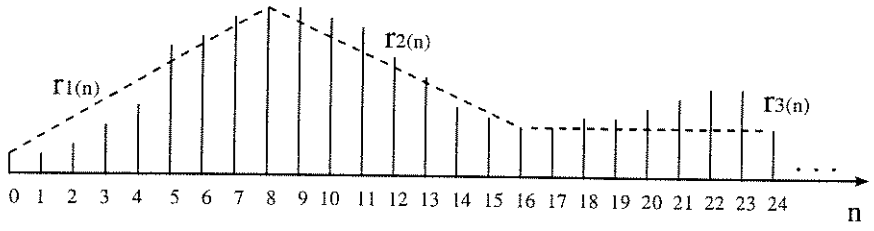


Figura 2.18. Seqüência discreta *x[n]* agrupadas em blocos e as retas $r_1[n]$, $r_2[n]$, ... usadas na pré-rotação de eixos.

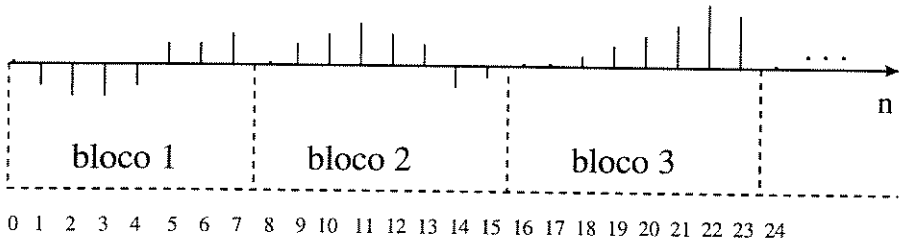


Figura 2.19. Seqüência discreta *xr[n]* gerada após o procedimento de pré-rotação de *x[n]*. Observe que em geral, as amostras vão reduzindo seus valores ao se aproximarem das extremidades de cada bloco.

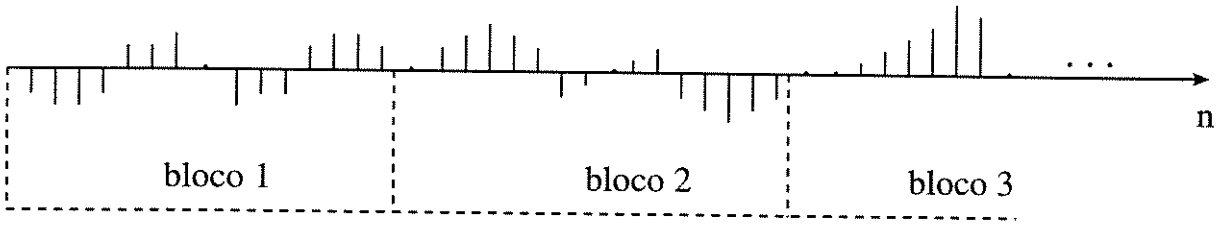


Figura 2.20. Seqüência estendida após a realização do procedimento de pré-rotação de eixos.

2.7 . Transformada Seno Discreta com Rotação de Eixos Bidimensional

A transformada seno discreta com rotação de eixos bidimensional (DSTr-2D), que faz parte da contribuição deste trabalho, pode ser obtida de sua versão unidimensional, bastando para isso obedecer a algumas particularidades que serão descritas a seguir.

Os procedimentos de rotação de eixos pode ser aplicados inicialmente sobre as linhas e colunas da seqüência bidimensional, acrescidos de alguns procedimentos adicionais que serão apresentado a seguir.

- Dada uma seqüência bidimensional $x[m,n]$, de tamanho NxN, deve-se dividi-la em blocos de tamanho MxM, $N \leq M$ e considerar que $x[m,n]$ é periódica.
- Tomar um bloco de tamanho $(M+1) \times (M+1)$, de modo que a última linha e a última coluna correspondam, respectivamente, à primeira linha e primeira coluna de blocos adjacentes, e aplicar os procedimento de pré-rotação de eixos em cada linha e em cada coluna do bloco.
- Aplicar o procedimento de pré-rotação de eixos na primeira linha e na primeira coluna do bloco original, substituindo a primeira linha e primeira coluna do bloco obtido no item anterior pelas obtidas nesse ítem.
- Descartar a última linha e a última coluna do bloco, para obter um bloco de tamanho MxM.
- Repetir o procedimento para os demais blocos da seqüência bidimensional $x[m,n]$, e considerar que a seqüência $x[m,n]$ é periódica quando se tratar de blocos que fazem limites com a última linha ou última coluna.

Para ilustrar o procedimento de pré-rotação de eixos de uma seqüência discreta bidimensional, a figura 2.21 mostra o exemplo de uma seqüência genérica agrupada em blocos de tamanho 4x4, sobre os quais serão aplicados a rotação de eixos. O procedimento de pré-rotação de eixos é aplicado em cada linha, de cada bloco, isto é, entre a primeira amostra de cada linha do bloco que está sendo processado e a primeira amostra de cada linha do bloco seguinte. Em seguida o mesmo procedimento é aplicado nas colunas, gerando blocos onde a primeira e a última linha são nulas, assim como a primeira e a última coluna. A última linha e a última coluna são eliminadas, para se obter nesse caso blocos de tamanho 4x4, permanecendo as primeiras linhas e colunas nulas, como mostra a figura 2.22. Para essas

linhas e colunas nulas existem as correspondente linhas e colunas originais que serão utilizadas no procedimento de pós-rotação para se recuperar o bloco original. Essas linhas e colunas originais formam uma grade, como mostra a figura 2.23. No entanto, essa grade não deve permanecer com todas as amostras originais, pois não seria apropriado para aplicações em processamento de imagens. Então, sobre essa grade é também aplicado o procedimento de pré-rotação unidimensional, e dessa forma substituirão as primeiras linhas e colunas nulas do bloco processado, como pode ser visto na figura 2.24. Por fim, somente uma amostra em cada bloco, permanecerá com o valor original. Sobre esses blocos processados pela rotação de eixos, será aplicada a transformada seno discreta, daí a razão pela qual se denominou de transformada seno discreta com rotação de eixos (DSTr). A transformada seno discreta é aplicada em cada bloco de acordo como mostrado na figura 2.25, isto é, após os procedimentos de pré-rotação de um bloco de tamanho *MxM*, a transformada seno discreta unidimensional é aplicada sobre a primeira linha e a primeira coluna do respectivo bloco, enquanto que no restante do bloco, portanto sobre um sub-bloco de tamanho *(M-1)x(M-1)*, aplica-se a transformada seno discreta bidimensional. Nota-se que mesmo após a aplicação da DST, a amostra b(1,1) de cada bloco transformado continuará com o valor original, pois ela será armazenada ou transmitida nessa forma, para ser utilizada no procedimento de pós rotação, aplicado após a transformada seno discreta inversa. Observe que o comprimento das funções bases são menores que o comprimento do bloco, sendo igual ao comprimento do bloco menos um. Por exemplo, na figura 2.25, onde o bloco tem dimensão 4x4, o comprimento da transformada é 3. Para blocos de tamanho 8, a transformada seno discreta tem comprimento igual a 7. Esse procedimento é diferente do usado na transformada LOT que utiliza funções bases de comprimento 16 para a obtenção de um blocos transformados de tamanho 8x8.

Outro ponto que deve ser observado é quando os procedimentos de pré-rotação é aplicado sobre uma imagem, visto que as linhas e as colunas têm comprimento finito. Uma vez que a pré-rotação é feita utilizando blocos de tamanho *(M+1)x(M+1)*, e o tamanho da imagem é em geral múltiplo de *M*, deve-se neste caso, considerar a imagem como periódica, de modo que a amostra de ordem *(M+1)* de uma linha (ou coluna) do último bloco, corresponde à primeira amostra da correspondente linha (ou coluna) do primeiro bloco. Procedendo-se dessa forma, somente uma amostra em cada bloco permanecerá com o seu

valor original, e em todos os blocos serão igualmente aplicada a DST, como é mostrado na figura 2.25. Apresenta-se a seguir a aplicação da DST sobre os blocos de uma imagem *xr[m,m]*, agrupada em blocos de tamanho *MxM*, e obtida pelo procedimento de pré-rotação como descrito anteriormente.

Seja *xbr[m,n]* um bloco de tamanho *MxM*. Aplica-se a DST-2D no sub-bloco de tamanho *(M-1)x(M-1)* a partir da segunda linha e segunda coluna, obtendo-se o sub-bloco transformado, isto é:

$$XBR[k_1,k_2]=\frac{2}{M}\sum_{n_1=1}^{M-1}\sum_{n_2=1}^{M-1}xbr[n_1,n_2]\,\mathrm{sen}\frac{\pi}{M}k_1n_1\,\mathrm{sen}\frac{\pi}{M}k_2n_2\,,\quad 1\le k_1\le M-1,\quad 1\le k_2\le M-1 \tag{2.55}$$

sobre a primeira linha e a primeira coluna de cada bloco aplica-se:

$$XR[k]=\sqrt{\frac{2}{M}}\sum_{n=1}^{M-1}xr[n]\,\mathrm{sen}\frac{\pi}{M}kn\,,\qquad 1\le k\le M-1\,. \tag{2.56}$$

Assim, a primeira amostra de cada bloco não sofre nenhuma transformação, permanecendo inalterada, para então ser utilizada no procedimento de pós-rotação.

A figura 2.26 mostra as funções bases usadas na transformação de blocos de dimensão 8 x 8, usando a DSTr, aplicados sobre as linhas e as colunas. As funções tem comprimento igual a 7, isto é, o comprimento do bloco menos um.

X00	X01	X02	X03	X04	X05	X06	X07	X08	X09 . . .
X10	X11	X12	X13	X14	X15	X16	X17	X18	X19 . . .
X20	X21	X22	X23	X24	X25	X26	X27	X28	X29 . . .
X30	X31	X32	X33	X34	X35	X36	X37	X38	X39 . . .
X40	X41	X42	X43	X44	X45	X46	X47	X48	X49 . . .
X50	X51	X52	X53	X54	X55	X56	X57	X58	X59 . . .
X60	X61	X62	X63	X64	X65	X66	X67	X68	X69 . . .
X70	X71	X72	X73	X74	X75	X76	X77	X78	X79 . . .
X80	X81	X82	X83	X84	X85	X86	X87	X88	X89 . . .
X90	X91	X92	X93	X94	X95	X96	X97	X98	X99 . . .
.	.	.	.	.	.	.	.	.	. . .

Figura 2.21. Exemplo de uma seqüência genérica *x[m,n]* agrupada em blocos de tamanho 4 x 4.

0	0	0	0	0	0	0	0	0	0 ...
0	x_{r11}	x_{r12}	x_{r13}	0	x_{r15}	x_{r16}	x_{r17}	0	x_{r19} ...
0	x_{r21}	x_{r22}	x_{r23}	0	x_{r25}	x_{r26}	x_{r27}	0	x_{r29} ...
0	x_{r31}	x_{r32}	x_{r33}	0	x_{r35}	x_{r36}	x_{r37}	0	x_{r39} ...
0	0	0	0	0	0	0	0	0	0
0	x_{r51}	x_{r52}	x_{r53}	0	x_{r55}	x_{r56}	x_{r57}	0	x_{r59} ...
0	x_{r61}	x_{r62}	x_{r63}	0	x_{r65}	x_{r66}	x_{r67}	0	x_{r69} ...
0	x_{r71}	x_{r72}	x_{r73}	0	x_{r75}	x_{r76}	x_{r77}	0	x_{r79} ...
0	0	0	0	0	0	0	0	0	0 ...
.	.	.	.	.	.	.	.	.	. ...

Figura 2.22. A seqüência *x[m,n]* agrupada em blocos de tamanho 4 x 4, após o procedimento de pré-rotação.

x_{00}	x_{01}	x_{02}	x_{03}	x_{04}	x_{05}	x_{06}	x_{07}	x_{08}	x_{09} ...
x_{10}	0	0	0	x_{14}	0	0	0	x_{18}	0 ...
x_{20}	0	0	0	x_{24}	0	0	0	x_{28}	0 ...
x_{30}	0	0	0	x_{34}	0	0	0	x_{38}	0 ...
x_{40}	x_{41}	x_{42}	x_{43}	x_{44}	x_{45}	x_{46}	x_{47}	x_{48}	x_{49} ...
x_{50}	0	0	0	x_{54}	0	0	0	x_{58}	0 ...
x_{60}	0	0	0	x_{64}	0	0	0	x_{68}	0 ...
x_{70}	0	0	0	x_{74}	0	0	0	x_{78}	0 ...
x_{80}	x_{81}	x_{82}	x_{83}	x_{84}	x_{85}	x_{86}	x_{87}	x_{88}	x_{89} ...
x_{90}	0	0	0	x_{94}	0	0	0	x_{98}	0 ...
.	.	.	.	.	.	.	.	.	. ...

Figura 2.23. A grade com as amostras originais resultante do procedimento de pré-rotação, sobre esta grade será aplicado também a pré-rotação.

x_{00}	x_{r01}	x_{r02}	x_{r03}	x_{04}	x_{r05}	x_{r06}	x_{r07}	x_{08}	x_{r09} ...
x_{r10}	x_{r11}	x_{r12}	x_{r13}	x_{r14}	x_{r15}	x_{r16}	x_{r17}	x_{r18}	x_{r19} ...
x_{r20}	x_{r21}	x_{r22}	x_{r23}	x_{r24}	x_{r25}	x_{r26}	x_{r27}	x_{r28}	x_{r29} ...
x_{r30}	x_{r31}	x_{r32}	x_{r33}	x_{r34}	x_{r35}	x_{r36}	x_{r37}	x_{r38}	x_{r39} ...
x_{40}	x_{r41}	x_{r42}	x_{r43}	x_{44}	x_{r45}	x_{r46}	x_{r47}	x_{48}	x_{r49} ...
x_{r50}	x_{r51}	x_{r52}	x_{r53}	x_{r54}	x_{r55}	x_{r56}	x_{r57}	x_{r58}	x_{r59} ...
x_{r60}	x_{r61}	x_{r62}	x_{r63}	x_{r64}	x_{r65}	x_{r66}	x_{r67}	x_{r68}	x_{r69} ...
x_{r70}	x_{r71}	x_{r72}	x_{r73}	x_{r74}	x_{r75}	x_{r76}	x_{r77}	x_{r78}	x_{r79} ...
x_{80}	x_{r81}	x_{r82}	x_{r83}	x_{84}	x_{r85}	x_{r86}	x_{r87}	x_{88}	x_{r89} ...
x_{r90}	x_{r91}	x_{r92}	x_{r93}	x_{r94}	x_{r95}	x_{r96}	x_{r97}	x_{r98}	x_{r99} ...
.	.	.	.	.	.	.	.	.	. ...

Figura 2.24. Seqüência final após a junção da grade e os blocos processados pela pré-rotação. Sobre esta seqüência será aplicada a DST.

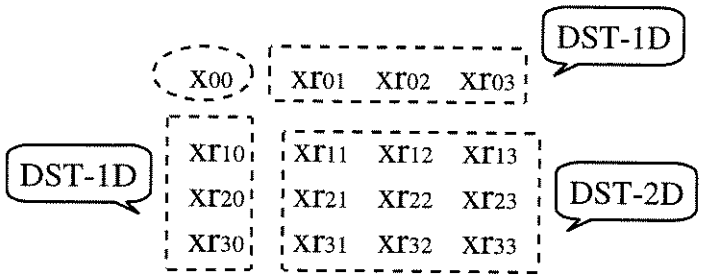


Figura 2.25. Aplicação da transformada seno discreta sobre um bloco após os procedimentos de pré-rotação.

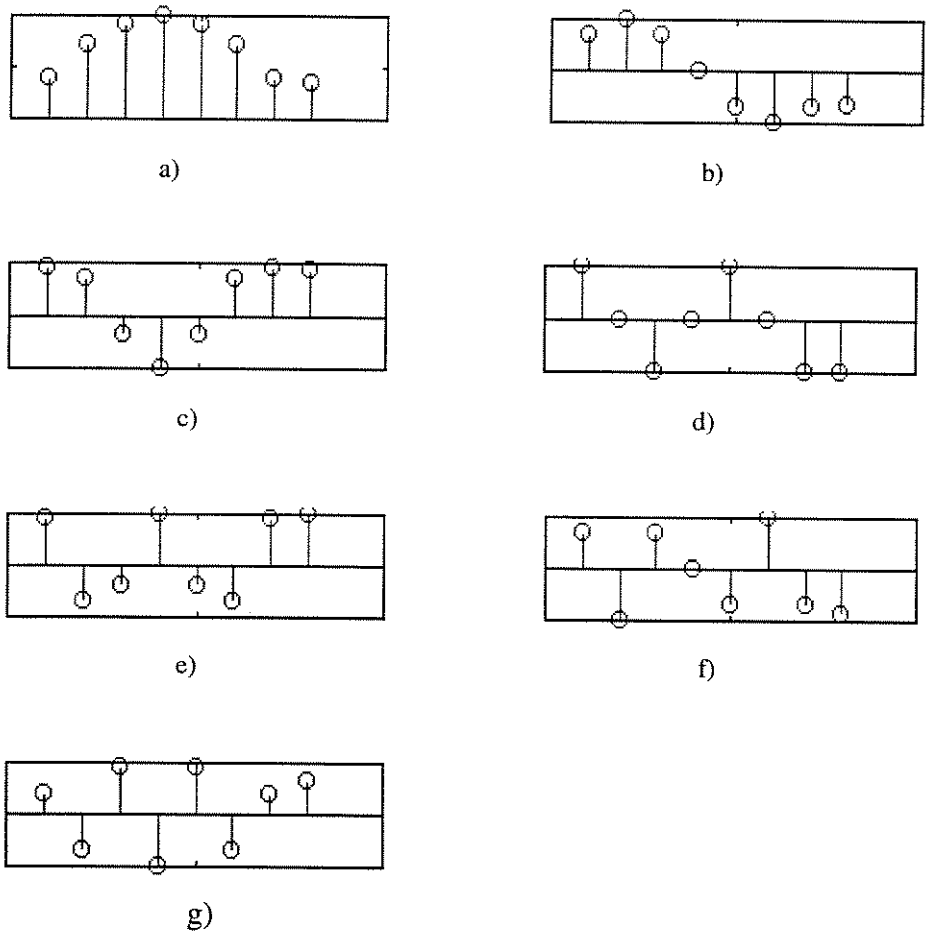


Figura 2.26. Funções bases usadas para transformação de blocos de tamanho 8 x 8.

CAPÍTULO 3

Interpolação de Imagens por Transformadas

A transformada discreta de Fourier (**DFT**) ou a sua implementação através de algoritmos eficientes, como a transformada rápida de Fourier (**FFT**), vem sendo usada há bastante tempo na interpolação de valores intermediários em seqüências finitas [2]. No entanto só recentemente esta técnica de interpolação no domínio transformado vem sendo aperfeiçoada e investigada com mais detalhes, as quais incluem interpolação usando **FFT** [14] com o janelamento da seqüência e eliminação de amostras. **Wang** [18, 19] propôs o uso da **DCT** na interpolação de seqüências unidimensionais, mostrando que seus resultados são superiores aos obtidos pela interpolação com **FFT**. **Araujo** [21], em 1994, com o objetivo de melhorar o desempenho de seqüências interpoladas propôs a utilização de pré e pós-rotação de eixo na seqüência. Essa rotação, de pequena complexidade, produziu melhorias na interpolação com filtragem pela FFT. **Araujo** [22], apresentou outro método de interpolação usando pré e pós-rotação de eixos com filtragem pela transformada seno, o que trouxe melhorias em relação aos anteriores, no que se refere a complexidade computacional e ao desempenho. **Pelaes** em 97 [23], apresentou dois métodos de interpolação de imagens com eliminação de extrapolações, usando a transformada coseno discreta (DCT) e a transformada seno discreta com rotação de eixos. Esses dois métodos são aplicados na interpolação de imagens neste trabalho e os resultados são comparados com imagens obtidas por outros métodos de interpolação por transformadas.

3.1. Amostragem Periódica de Sinais.

Existem vários tipos de amostragem de sinais, no entanto a amostragem periódica é a mais simples e a mais usada para representação de sinais contínuos através de suas amostras periódicas, onde uma seqüência de amostras $x[n]$ é obtida de um sinal contínuo $x_c(t)$ de acordo com a relação:

$$x[n] = x_c(nT_s), \quad -\infty < n < +\infty \tag{3.1}$$

onde T_s é o período de amostragem e o inverso $f_s = 1 / T_s$ é a freqüência (ou taxa) de amostragem, em amostras por segundo.

Um sistema que realiza a operação de amostragem é chamado de um conversor contínuo discreto ideal. Na prática, no entanto, a operação de amostragem é feita por um conversor analógico digital, (A/D), tais sistemas podem ser vistos como uma aproximação de um conversor contínuo discreto. Na implementação de um conversor A/D deve ser levado em conta o tipo de quantização das amostras de saída, linearidade, limitações da taxa de amostragem, etc. É conveniente representar matematicamente o processo de amostragem em dois estágios: o primeiro consiste em modular um trem de impulsos e o segundo, de converter o trem de impulsos em uma seqüência discreta no tempo. A figura 3.1a mostra o diagrama de blocos de um sistema de amostragem com dois estágios, onde $s(t)$ representa um trem de impulsos, $x_c(t)$ é o sinal contínuo no tempo a ser amostrado, $x_a(t)$ representa o trem de impulsos modulado e $x[n]$ a seqüência discreta obtida. A figura 3.1.b mostra um trem de pulsos periódico com período T_s, modulado por um sinal contínuo $x_c(t)$, e a figura 3.1.c mostra a seqüência discreta $x[n]$, obtida pela conversão a partir do trem de impulsos modulado. A relação entrada saída, no domínio do tempo e da freqüência, para o conversor C/D (contínuo/discreto) é a seguir apresentada.

Considere um sinal $g(t)$ formado por um trem de impulsos, dado por:

$$g(t) = \sum_{n=-\infty}^{\infty} \delta(t - nT_s) \tag{3.2}$$

onde $\delta(t)$ é um impulso unitário ou função delta de Dirac. Então, como mostrado na figura 3.2, o sinal modulado $x_a(t)$ é dado por:

$$x_a(t) = x_c(t)g(t) = x_c(t)\sum_{n=-\infty}^{\infty} \delta(t - nT_s) \tag{3.3}$$

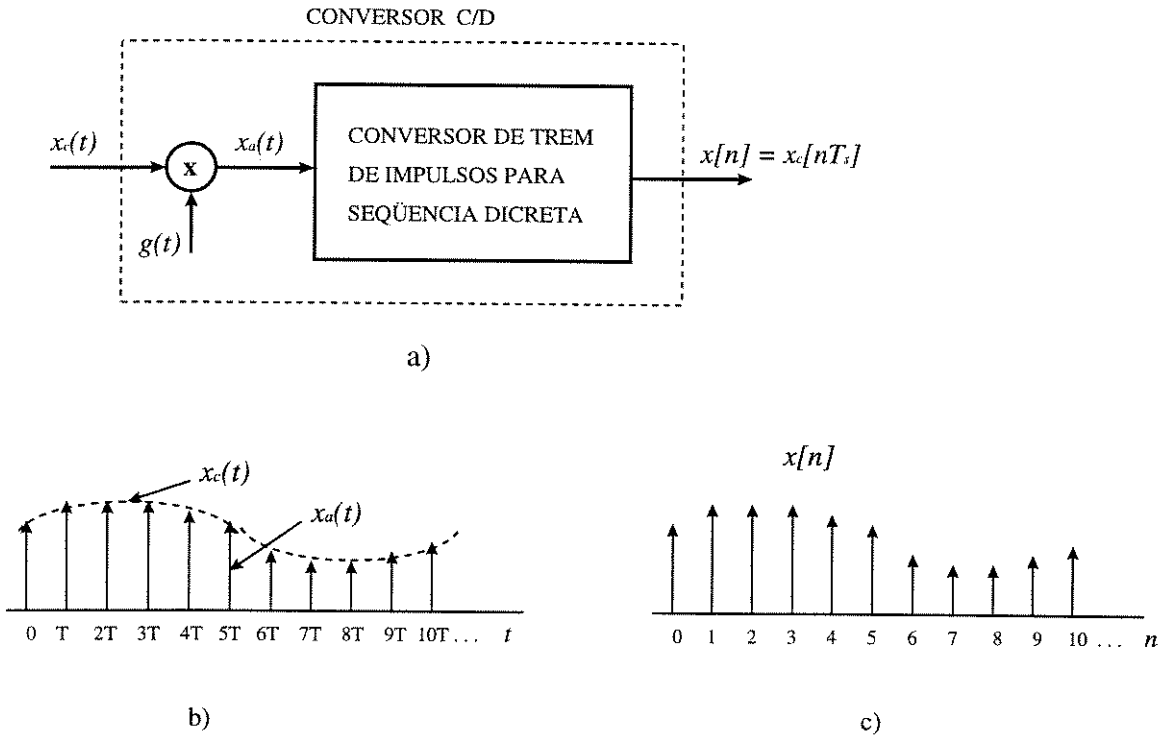


Figura 3.1. Sistema de amostragem. a) Diagrama de blocos com dois estágios; b) trem de impulsos periódicos modulados por $x_c(t)$; c) seqüência discreta obtida na saída do sistema.

que pode ser expressa na forma:

$$x_a(t) = \sum_{n=-\infty}^{\infty} x_c(nT_s)\delta(t - nT_s) \tag{3.4}$$

Para se obter a transformada de Fourier do sinal $x_a(t)$, calcula-se a transformada de Fourier do trem de impulsos $g(t)$ dado por:

$$G(f) = \frac{1}{T_s} \sum_{k=-\infty}^{\infty} \delta(f - \frac{k}{T_s}) \tag{3.5}$$

Aplicando-se a propriedade da transformada de Fourier em (3.3), da qual um produto no tempo corresponde uma convolução na freqüência, obtém-se:

$$X_a(f) = X_c(f) * G(f) \tag{3.6}$$

onde " * " significa a operação de convolução. Substituindo-se (3.5) em (3.6), obtém-se, finalmente:

$$X_a(f) = \frac{1}{T_s} \sum_{k=-\infty}^{\infty} X_c(f - \frac{k}{T_s}) \qquad (3.7)$$

A equação (3.7) mostra que a transformada de Fourier do sinal amostrado $x_a(t)$ é formado pela repetição da transformada de Fourier de $x_c(t)$, com período f_s. A figura 3.2 mostra através de gráficos os procedimentos para a obtenção do espectro do sinal amostrado, onde a freqüência de amostragem $f_s > 2w$, e w é a largura de faixa do sinal $x_c(t)$, em hertz.

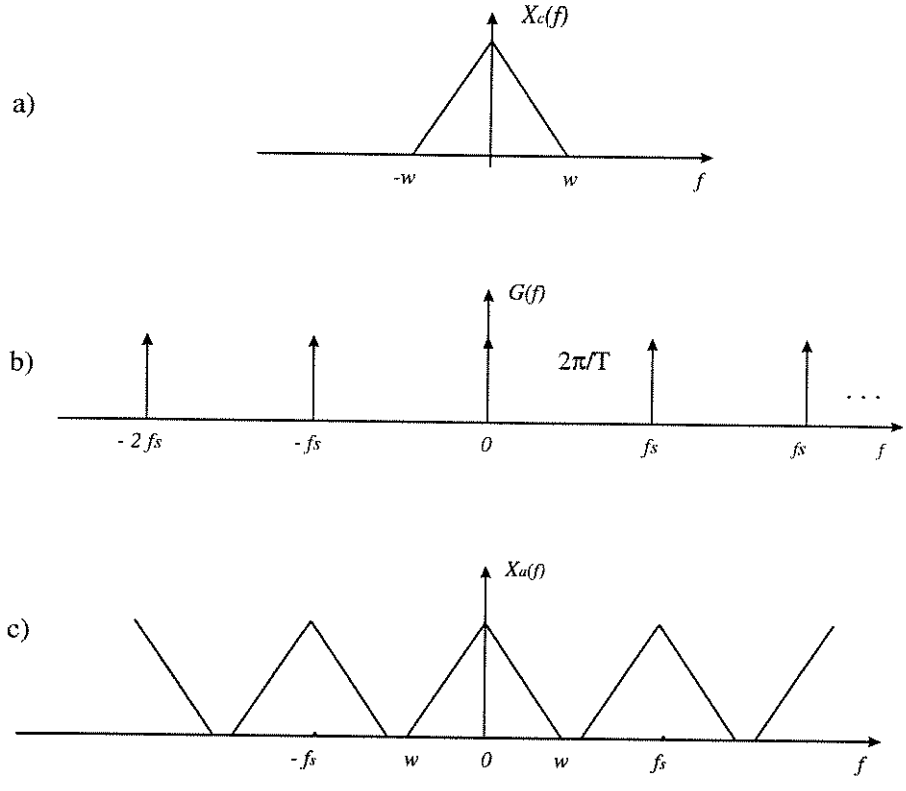


Figura 3.2. a) Espectro do sinal contínuo no tempo $x_c(t)$, com largura de faixa w; b) Espectro do trem de impulsos; c) Espectro do sinal amostrado $x_a(t)$, com freqüência de amostragem $f_s > 2w$.

O teorema de Nyquist estabelece que, para se reconstituir um sinal original a partir de sua amostras é necessário que a taxa de amostragem f_s seja maior ou igual a duas vezes a largura deve faixa do sinal original, isto é, se $x_c(t)$ é um sinal de banda limitada com,

$$X_c(f) = 0, \text{ para } |f| > w \tag{3.8}$$

Então o sinal contínuo, $x_c(t)$, é unicamente determinado a partir de suas amostras $x[n] = x_c[nT_s]$, para $n = 0, \pm 1, \pm 2, \ldots$ se

$$f_s > 2w. \tag{3.9}$$

Para se reconstituir o sinal original de banda limitada a partir de suas amostras como descritas acima, é necessário passar o sinal amostrado por um filtro passa baixas ideal com freqüência de corte entre w (a largura de faixa do sinal $x_c(t)$) e f_s - w. Se $h_r(t)$ é a resposta impulso do filtro de reconstituição, então o sinal reconstruído $x_r(t)$ é dado por:

$$x_r(t) = \sum_{n=-\infty}^{\infty} x[n] h_r(t - nT_s) \tag{3.10}$$

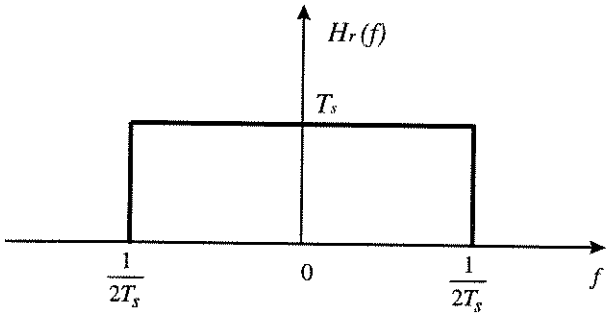
Escolhendo a largura do filtro de reconstituição $B = f_s/2 = 1/2T_s$, tem-se, no domínio da freqüência:

$$H_r(f) = \begin{cases} T_s, & |f| < \dfrac{1}{2T_s} \\ 0, & \text{para outros valores} \end{cases} \tag{3.11}$$

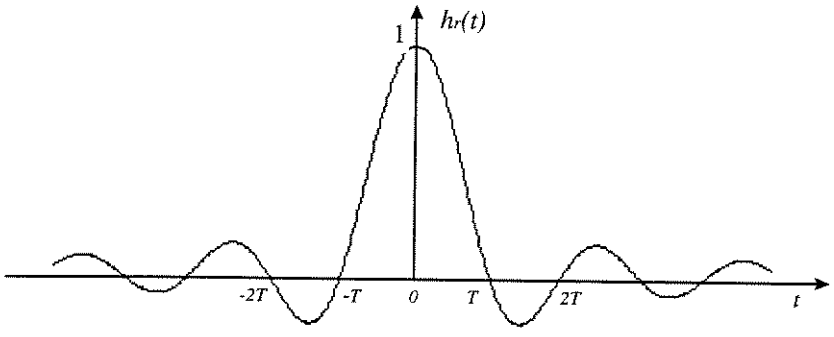
ou no domínio do tempo:

$$h_r(t) = sinc(t/T_s) = \frac{sen(2\pi t / T_s)}{2\pi t / T_s} \tag{3.12}$$

As figuras 3.3.a e 3.3.b mostram $H_r(f)$ e $h_r(t)$, o filtro de reconstrução no domínio da freqüência e do tempo, respectivamente.



a)



b)

Figura 3.3. Filtro de reconstituição; a) No domínio da freqüência; b) No domínio do tempo.

Substituindo (3.12) em (3.10), obtém-se:

$$x_r(t) = \sum_{n=-\infty}^{\infty} x[n] \frac{sen(2\pi(t-nT)/T]}{2\pi(t-nT)/T} \qquad (3.13)$$

Assim, se $x[n] = x_c(nT)$, então [10],

$$x_r(mT) = x_c(mT)\,. \qquad (3.14)$$

3.2. Mudança da Taxa de Amostragem Usando Processamento Discreto.

Como foi visto na seção anterior um sinal contínuo no tempo $x_c(t)$, pode ser representado por um sinal discreto consistindo de uma seqüência de amostras:

$$x[n] = x_c(nT) \tag{3.15}$$

Algumas vezes torna-se necessário trocar a taxa de amostragem de um sinal discreto, isto é, obter uma nova representação do sinal da forma:

$$x'[n] = x_c(nT') \tag{3.16}$$

onde, $T' \neq T$. Obter $x'[n]$ de $x[n]$ significa reconstruir $x_c(t)$ a partir $x[n]$, usando a equação (3.13). Em seguida amostrar-se-ia novamente $x_c(t)$ com o novo período de amostragem T', obtendo-se finalmente o novo sinal discreto $x'[n]$. No entanto este não é um procedimento muito adequado, devido principalmente, às características não ideais dos filtros de reconstituição. Por esse motivo são usados, em geral, métodos que envolvem somente processamentos digital de sinais. Esses métodos são utilizados para reduzir a taxa de amostragem, denominados de **decimação**, ou para aumentar a taxa de amostragem, chamados de **interpolação**.

3.2.1. Redução da Taxa de Amostragem: Decimação.

O processo de converter digitalmente a taxa de amostragem de um sinal de uma dada taxa de amostragem $f_s = 1/T_s$ para uma nova taxa de amostragem $f'_s = 1/(MT_s)$, com M inteiro e $M>1$, é chamado de decimação. A nova seqüência obtida por esse processo, é definida por:

$$x_d[n] = x[nM] = x_c(nMT) \tag{3.17}$$

A seqüência discreta $x_d[n]$ é uma representação exata de $x_c(t)$, se $1/(MT_s) > 2w$, onde w é a largura de faixa do sinal original $x_c(t)$, isto é, $X_c(f) = 0$, se $|f| > w$. Isso significa que a taxa de amostragem pode ser reduzida por um fator M, sem a superposição de espectro ("aliasing"), se a taxa de amostragem original for pelo menos M vezes a taxa de Nyquist ou se a largura de faixa da seqüência original for inicialmente reduzida por uma fator M por um procedimento de pré-filtragem [10]. A figura 3.4 mostra o diagrama de blocos de um sistema genérico usado para reduzir a taxa de amostragem de $f_s = 1/T_s$ para $f_s' = 1/MT_s$.

É interessante obter uma relação entre o sinal decimado e o sinal original no domínio da freqüência. Para tanto, toma-se a equação (3.7), e faz-se $T = MT_s$, para obter-se:

$$X_d(f) = \frac{1}{MT_s} \sum_{k=-\infty}^{\infty} X_c\left(f - \frac{k}{MT_s}\right) \qquad (3.18)$$

A figura 3.5.a mostra o espectro de sinal de faixa limitada em [$-w$, w] e em 3.5.b o espectro do sinal amostrado com uma taxa $f_s = 1/T_s$; a figura 3.5.c mostra o espectro do sinal decimado $x_d[n]$, obtido pela mudança da taxa de amostragem de f_s para f_s/M, onde $M = 2$. Observe que neste caso não há superposição de espectro ("aliasing").

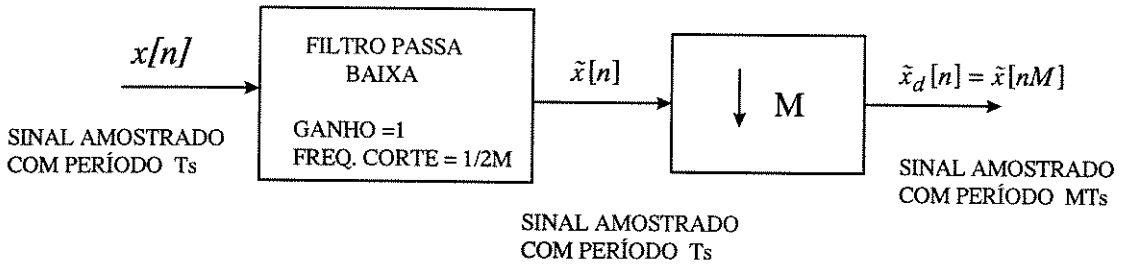


Figura 3.4. Diagrama de blocos de um sistema genérico usado para reduzir a taxa de amostragem por um fator M.

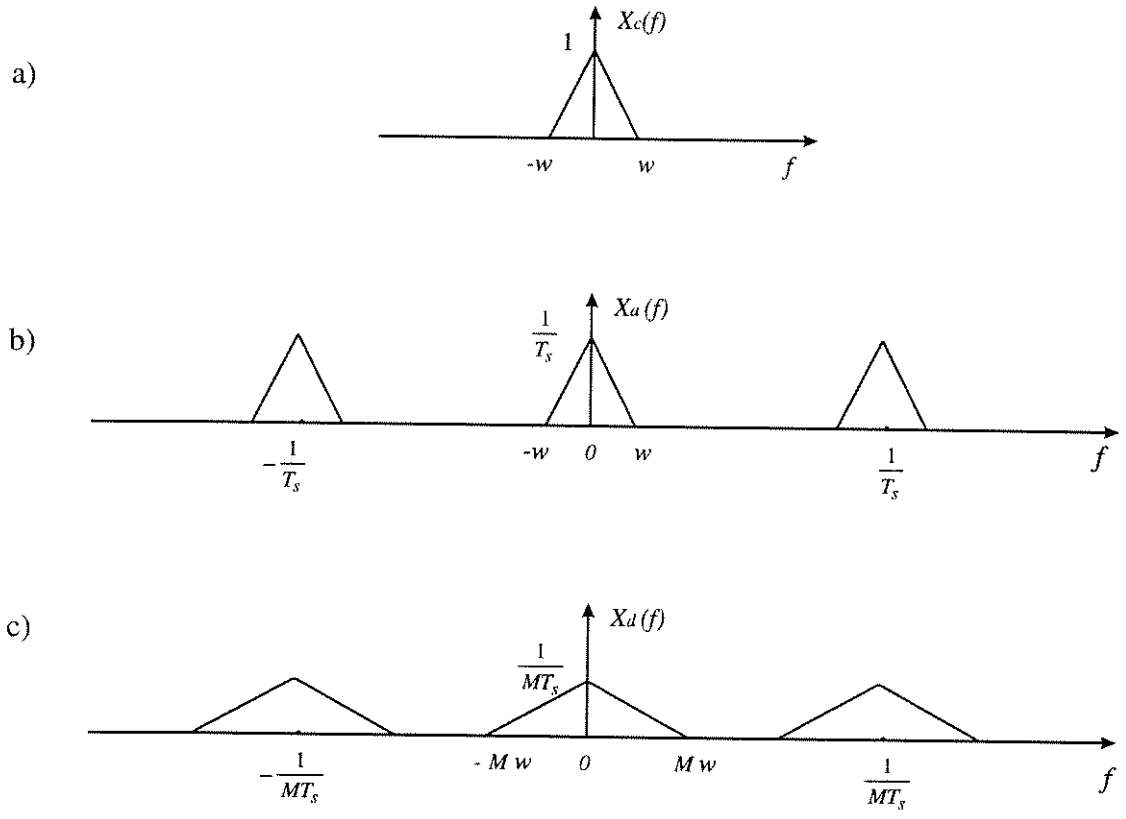


Figura 3.5 a) Espectro de um sinal $x_c(t)$ de faixa limitada. b) Espectro do sinal amostrado. c) Espectro do sinal decimado, que eqüivale ao sinal original amostrado com uma nova taxa de amostragem $f_s = 1/MT_s$.

3.2.2. Aumento da Taxa de Amostragem : Interpolação.

O processo cujo objetivo é aumentar a taxa de amostragem de um sinal é chamado de interpolação, uma vez que são criadas novas amostras do sinal original a partir de um conjunto reduzido de amostras. Historicamente, o processo de interpolação tem recebido uma grande atenção, principalmente de matemáticos, apesar de que em outras áreas, tais como processamento de sinais e de imagens, muitos trabalhos vem sendo freqüentemente desenvolvidos. Os algoritmos são implementados usando-se FFT, DCT, DST, além do uso de filtros digitais FIR e IIR [2, 3, 14, 15].

A interpolação consiste em se obter um novo sinal com uma taxa de amostragem superior à taxa do sinal originalmente amostrado, isto é, o sinal interpolado $x_i[n]$ é tal que:

$$x_i[n] = x_c[nT_s'] = x_c[\, nT_s / L].$$ (3.19)

A figura 3.6 mostra o diagrama de blocos de um interpelador genérico usado para a obtenção de uma seqüência discreta interpolada $x_i[n]$ a partir de $x[n]$. O primeiro bloco da figura 3.6 é chamado de expansor da taxa de amostragem, ou simplesmente expansor. O segundo bloco é um filtro passa baixas com freqüência de corte igual a 1 / 2L e ganho L. Assim, o sistema gera um trem de impulsos discreto $x_e[n]$ que é passada através de um filtro passa baixas para reconstruir a seqüência. A saída do expansor $x_e[n]$ é dada por:

$$x_e[n] = \begin{cases} x[n / L], & n = 0, \pm L, \pm 2L... \\ 0, & para \quad outros \quad valores \end{cases}$$ (3.20)

ou equivalentemente:

$$x_e[n] = \sum_{k=-\infty}^{\infty} x[k]\delta[n - kL]$$ (3.21)

Tomando-se a transformada de Fourier de (3.21), obtém-se:

$$X_e[f] = \sum_{n=-\infty}^{\infty} \left(\sum_{k=-\infty}^{\infty} x[k]\delta[n - kL] \right) e^{-j2\pi fn} = \sum_{k=-\infty}^{\infty} x[k] e^{-j2\pi Lfk} = X(Lf)$$ (3.22)

A figura 3.7 mostra a obtenção de um sinal interpolado no domínio da freqüência. A figura 3.7a mostra o espectro de um sinal contínuo $x_c(t)$ e em 3.7b o espectro do sinal amostrado com uma taxa $f_s = 1/T_s$. Na figura 3.7b é mostrado o espectro do sinal expandido de acordo com a equação (3.22). Finalmente na figura 3.7d é mostrado o espectro do sinal interpolado para uma nova taxa $Lf_s = L/T_s$, obtido pela filtragem do sinal expandido por um filtro passa baixas ideal, com largura de faixa $B = w/L$.

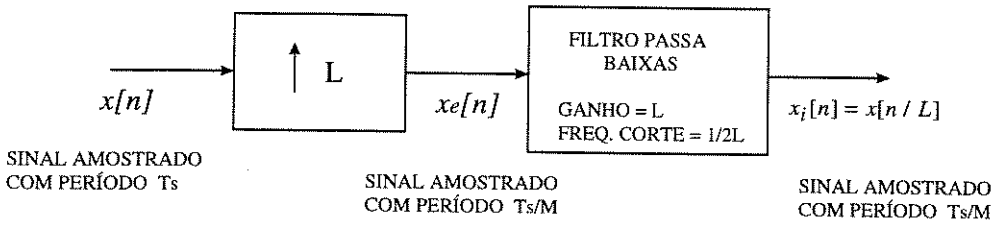


Figura 3.6. Diagrama de blocos de um sistema genérico para aumentara a taxa de amostragem.

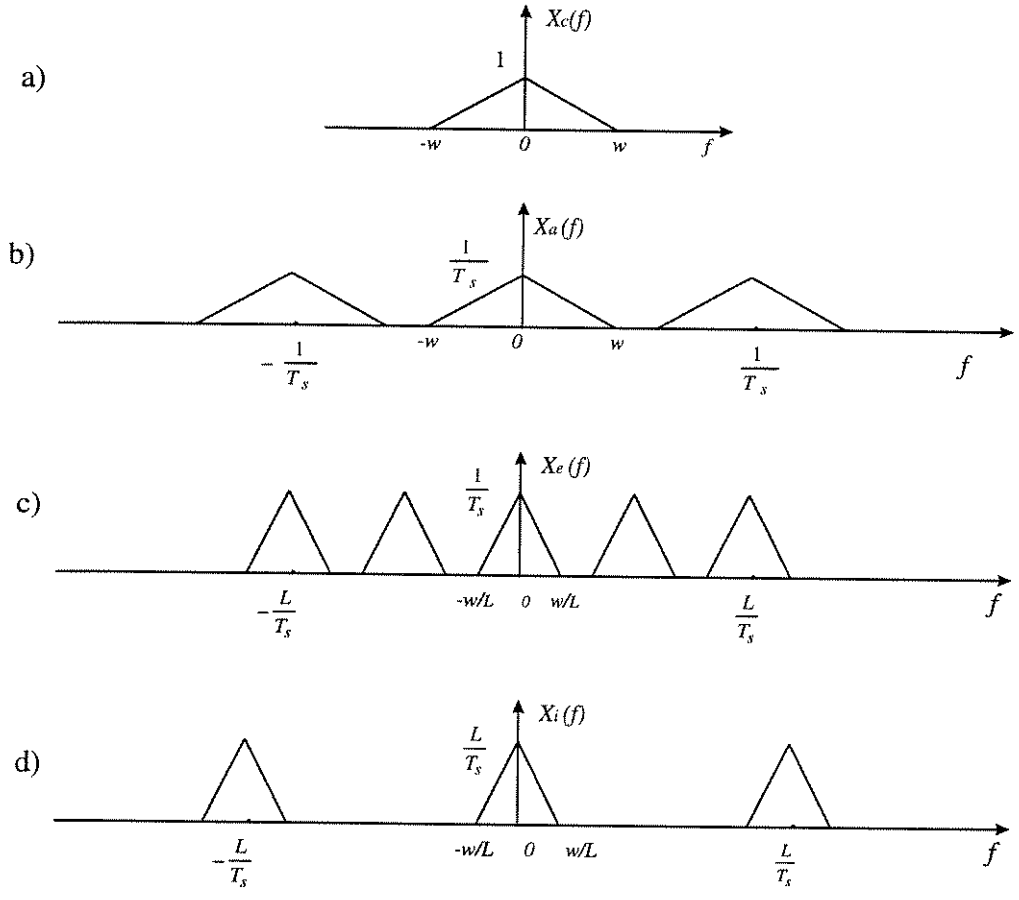


Figura 3.7. Interpolação no domínio da freqüência. A) Espectro do sinal contínuo original; b) Espectro do sinal amostrado com uma taxa T_s; c) Espectro de sinal expandido; d) Espectro do sinal interpolado para taxa $Lf_s = L/T_s$.

3.3. Amostragem de Sinais Bidimensionais.

A amostragem de sinais bidimensionais é feita tanto no espaço como em amplitude, isto é, quando uma imagem contínua $x_c(x_1,x_2)$ é amostrada, as coordenadas (x_1,x_2) representam a posição no espaço e a amplitudes amostradas representam os níveis de cinza. Para ilustrar a

amostragem de sinais dimensionais, considere uma imagem bidimensional de faixa limitada, isto é, sua transformada de Fourier $X(f_1,f_2)$ é tal que:

$$X(f_1,f_2) = 0\,, \quad f_1 > w_1 \;\; e \;\; f_2 > w_2\,. \tag{3.23}$$

onde w_1 e w_2 são as larguras de faixa da imagem bidimensional nas direções f_1 e f_2, respectivamente. Considere-se agora um trem de impulsos em duas dimensões, $\delta_s(x_1,x_2)$, sobre uma grade retangular espaçada de T_{s1} e T_{s2}, ou seja,

$$\delta_s(x_1, y_2) = \sum_{n_1}^{\infty}\sum_{n_2}^{\infty}\delta(x_1 - n_1T_{s1}, x_2 - n_2T_{s2})\,. \tag{3.24}$$

Uma imagem amostrada $x_s(n_1,n_2)$ é definida como:

$$x_s(x_1, x_2) = x_s(x_1, x_2).\delta_s(x_1, y_2) \tag{3.25}$$

$$x_s(x_1, x_2) = \sum_{n_1}^{\infty}\sum_{n_2}^{\infty} x_c(n_1T_{s1}, n_2T_{s2}).\delta(x_1 - n_1T_{s1}, x_1 - n_2T_{s2}) \tag{3.26}$$

Calculando-se a transformada de Fourier de $x_s(x_1, x_2)$, obtém-se :

$$X_s(f_1, f_2) = f_{s1}f_{s2}\sum_{n_1=-\infty}^{\infty}\sum_{n_2=-\infty}^{\infty} X_c(f_1 - n_1f_{s1}, f_1 - n_2f_{s2}) \tag{3.27}$$

ou em função dos períodos de amostragem T_{s1} e T_{s2} ,

$$X_s(f_1, f_2) = \frac{1}{T_{s1}T_{s2}}\sum_{n_1=-\infty}^{\infty}\sum_{n_2=-\infty}^{\infty} X_c(f_1 - \frac{n_1}{T_{s1}}, f_1 - \frac{n_2}{T_{s2}})\,. \tag{3.28}$$

onde $f_{s1} = 1/T_{s2}$ e $f_{s2} = 1/T_{s2}$, e T_{s1} é o período de amostragem na abcissa ou horizontal e T_{s2} na ordenada ou vertical.

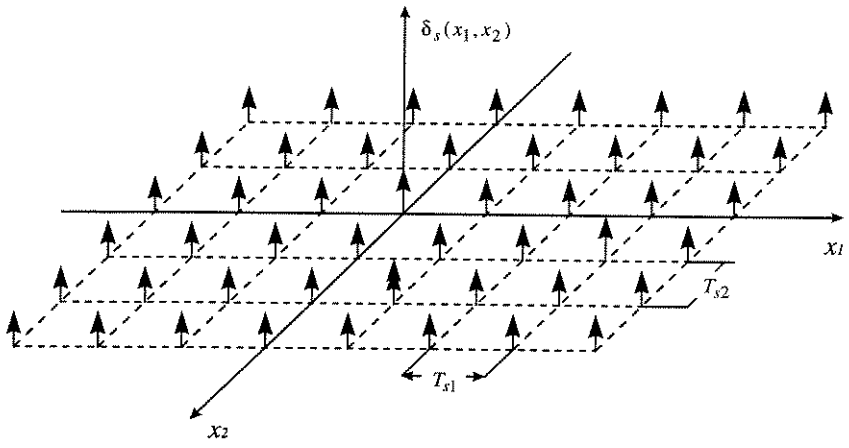


Figura 3.8. Trem de impulsos periódicos bidimensional, espaçados de T_{s1} na direção x_1 e de T_{s2} na direção x_2.

O teorema da amostragem de Nyquist para sinais bidimensionais estabelece que, para se reconstituir uma imagem a partir de suas amostras é necessário, primeiro que a imagem não tenha nenhuma componente diferente de zero acima de w_1 e w_2 (as larguras de faixa da imagem), conforme a equação (3.23); segundo, que as taxas de amostragens f_{s1} e f_{s2} obedeçam aos seguintes critérios:

$$f_{s1} \geq 2w_1 \quad \text{e} \quad f_{s2} \geq 2w_2 \tag{3.29}$$

A reconstrução é feita usando-se um filtro passa baixas ideal, $Hr(f_1,f_2)$, com freqüência de corte w_1, w_2, ou seja,

$$H_r(f_1,f_2) = \begin{cases} T_{s1}T_{s2}, & |f_1| \leq w_1,\ |f_2| \leq w_2 \\ 0, & \textit{para outros valores} \end{cases} \tag{3.30}$$

Assim, a imagem reconstituída $X_r(f_1,f_2)$ é obtida, passando-se a imagem amostrada por $H_r(f_1,f_2)$. A imagem reconstituída é representada pela equação:

$$X_r(f_1, f_2) = X_s(f_1, f_2).H_r(f_1, f_2) \tag{3.31}$$

No domínio espacial tem-se:

$$x_r(x_1,x_2)=\sum_{n_1=-\infty}^{\infty}\sum_{n_2=-\infty}^{\infty}x(n_1,n_2)\frac{\operatorname{sen}\frac{\pi}{T_1}(t_1-n_1T_1)\operatorname{sen}\frac{\pi}{T_2}(t_2-n_2T_2)}{\frac{\pi}{T_1}(t_1-n_1T_1)\frac{\pi}{T_2}(t_2-n_2T_2)} \tag{3.32}$$

A imagem recuperada $x_r(x_1, x_2)$ deve ser idêntica à imagem original $x_c(x_1, x_2)$, se a taxa de amostragem obedece aos critérios de Nyquist para sinais bidimensionais, caso contrário $x_r(x_1,x_2)$ será uma versão de $x_c(x_1, x_2)$ sobreposição de espectro ("aliasing").

3.4. Interpolação de Sinais Bidimensionais.

Considere uma seqüência bidimensional $x[n_1, n_2]$ obtida por amostragem de uma seqüência analógica $x_c[x_1, x_2]$, isto é:

$$x[n_1,n_2]=x_c[x_1,x_2]\Big|_{x_1=n_1T_1,x_2=n_2T_2}. \tag{3.33}$$

Se $x_c[x_1, x_2]$ é de faixa limitada e as freqüências de amostragem $1/T_1$ e $1/T_2$ são maiores que a freqüência de Nyquist, então o sinal bidimensional pode ser reconstruído passando-se o sinal amostrado $x[n_1, n_2]$, por um filtro de reconstituição $h_r(x_1, x_2)$, ideal passa baixas, isto é:

$$x_r(x_1,x_2)=\sum_{n_1=-\infty}^{\infty}\sum_{n_2=-\infty}^{\infty}x(n_1,n_2)h_r(x_1-n_1T_1,x_2-n_2T_2) \tag{3.34}$$

onde:

$$h_r(x_1,x_2)=\frac{\operatorname{sen}(\frac{\pi}{T_1}t_1)\operatorname{sen}(\frac{\pi}{T_2}t_2)}{(\frac{\pi}{T_1}x_1)(\frac{\pi}{T_1}x_2)}. \tag{3.35}$$

O uso da expressão mostrada em 3.34 não é apropriado, para o uso na prática, de interpolação de imagens uma vez que o filtro de reconstituição não é estritamente de faixa limitada; então certamente ocorrerá superposição de espectro quando $x_c[x_1, x_2]$ for amostrado. Além disso, o

filtro de reconstituição $h_r(x_1, x_2)$ é uma função de comprimento teoricamente infinita [4]. Em conseqüência disso, outros métodos de interpolação foram propostos na literatura. Alguns são implementados no domínio espacial, como bilinear e cúbica, outros são implementados no domínio transformado usando FFT, DCT e DST [2, 3, 14, 15, 16, 19]. Neste trabalho será dada atenção especial aos métodos de interpolação no domínio transformado. A seguir serão apresentados alguns métodos de interpolação por transformadas usados neste trabalho .

3.5. Interpolação de Sinais por Transformada.

Nesta seção serão apresentados alguns métodos de interpolação de sinais unidimensionais que são implementados através de transformadas. Nas seções seguintes esses métodos serão estendidos para aplicações em sinais bidimensionais e aplicados para interpolação de imagens, e os resultados serão então comparados.

Método 1: Interpolação usando transformada coseno discreta.

Este primeiro método de interpolação foi desenvolvido por Wang [18, 19], e utiliza a transformada coseno discreta do tipo I, com uma pequena modificação. A descrição do algoritmo é apresentado a seguir:

O procedimento a seguir interpola uma seqüência $x[n]$, $n = 0, 1, 2, ..., N$, de comprimento $N+1$, para gerar uma nova seqüência interpolada de comprimento $PN+1$, usando a DCT-I modificada.

- Compute a seqüência de coeficientes DCT-I modificados $X[k]$ para a seqüência $x[n]$, isto é:

$$X(k) = \frac{2}{N} c_k \sum_{n=0}^{N} c_n x(n) \cos\left(\frac{nk\pi}{N}\right) , \quad k = 0, 1, 2, \ldots, N. \tag{3.36}$$

onde:

$$c_j = \begin{cases} 1/2, & j = 0 \quad j = N \\ 0, & \textit{para outros valores} \end{cases} \tag{3.37}$$

- Criar uma nova seqüência de comprimento *PN+1*, acrescentando *(P-1)N* Zeros no final da seqüência *X(m)*:

$$Y(k) = \begin{cases} X(k), & k = 0,\ 1,\ 2, \ldots, N \\ 0, & N \le k \le PN+1 \end{cases} \tag{3.38}$$

- Computar a transformada coseno discreta inversa modificada da seqüência modificada *Y(k)*, para obter a seqüência interpolada de comprimento *PN+1*, dada por:

$$y(n) = \sum_{k=0}^{PN} Y(k)\cos(\frac{nk\pi}{PN}), \qquad n = 0,\ 1,\ 2,\ 3,\ \ldots,PN. \tag{3.39}$$

y(n) é a seqüência interpolada desejada e é relacionada com a seqüência original *x(n)*, por:

$$y(Pn) = x(n). \qquad n = 0,\ 1,\ 2,\ \ldots,\ N \tag{3.40}$$

Método 2: Interpolação usando DCT/FFT.

Nesse método de interpolação uma seqüência *x[n], n = 0, 1, 2, ..., N-1*, de comprimento *N*, será interpolada para se obter uma seqüência de comprimento *PN* [23]. Devido a periodicidade da transformada discreta de Fourier, as últimas *(P-1)* amostras do sinal interpolado são geradas entre a última e a primeira amostra do sinal original. Essas *(P-1)* amostras são denominadas de extrapolações. O algoritmo para esse método é descrito a seguir:

- Dada a seqüência unidimensional *x[n], n = 0, 1, 2, ..., N-1*, de comprimento *N*, obter uma seqüência estendida *y[n], n = 0, 1, 2, ..., 2N-1*, de comprimento *2N*, obtida por extensão de simetria par, isto é:

$$y[n] = \begin{cases} x[n], & 0 \le n \le N-1 \\ x[2N-1-n], & N \le n \le 2N-1 \end{cases} \tag{3.41}$$

- Calcular a transformada de Fourier discreta (DFT) *Y[k]*, da seqüência *y[n]*, isto é:

$$Y[k] = \sum_{n=0}^{PN-1} x[n] W_{2N}^{-nk} \quad , \qquad k = 0, 1, 2, ..., PN-1. \tag{3.42}$$

$$\text{onde:} \quad W_{2N}^{-nk} = \exp(-\frac{2\pi nk}{2N}) \tag{3.43}$$

- Definir uma nova seqüência *Y1[k], k = 0, 1, 2, ... 2PN*, de comprimento *2PN*, a partir de *Y[k]* pela inserção de *2(P-1)N* zeros da forma.

$$Y_1[k] = \begin{cases} y[k], & 0 \le k \le N-1 \\ 0, & N \le k \le (2P-1)N-1 \\ y[k-2PN+2N], & (2P-1)N \le k \le 2PN-1 \end{cases} \tag{3.44}$$

- Calcular a transformada de Fourier discreta inversa (IDFT) da seqüência $Y_1[k]$, usando-se a equação:

$$y_1[n] = \sum_{k=0}^{2PN-1} Y_1[k] W_{2PN}^{nk} \quad , \qquad n = 0, 1, 2, ..., 2PN-1. \tag{3.45}$$

$$\text{onde:} \quad W_{2PN}^{nk} = \exp(\frac{2\pi nk}{2PN}) \tag{3.46}$$

- Tomar a parte real de $y_1[n]$ multiplicando por um fator *P*. Deve-se observar neste ponto que a seqüência $y_1[n]$ é real. Define-se:

$$y_0[n] = parte\ real(\ P\ y_1\ [n]\ )\ , \qquad n = 0, 1, 2, ..., 2PN-1 \tag{3.47}$$

- A seqüência interpolada $x_i[n]$, de comprimento *PN* é obtida dos *PN* primeiros termos da seqüência $y_0[n]$, isto é:

$$x_i[n] = y_0[n], \qquad n = 0, 1, 2, \ldots, PN-1. \tag{3.48}$$

- As últimas $(P-1)$ amostras de $x_i[n]$ são extrapolações e podem ser descartadas, obtendo-se uma seqüência de comprimento $P(N-1)+1$, que contém apenas interpolações, ou mais precisamente,

$$x_i[n] = y_0[n], \qquad n = 0, 1, 2, \ldots, P(N-1)+1. \tag{3.49}$$

Método 3: Interpolação usando Transformada seno discreta com rotação de eixos.

Neste método de interpolação uma seqüência $x[n]$, $n = 0, 1, 2,\ldots, N$, de comprimento $N+1$, é interpolada para obter uma seqüência de comprimento $PN+1$, usando uma pré-rotação de eixos seguido da interpolação usando a transformada seno discreta [22]. Um procedimento de pós-rotação de eixos é aplicado para a obtenção da seqüência interpolada. O algoritmo é descrito a seguir.

- Considerar uma seqüência discreta $x[n]$, $n = 0, 1, 2,\ldots, N$, de comprimento $N+1$.
- Aplicar o procedimento de pré-rotação de eixos, como descrito na seção 2.4, isto é:

$$xr[n] = x[n] - r[n], \qquad n = 0, 1, 2,\ldots, N \tag{3.50}$$

onde,

$$r[n] = \frac{x[N]-x[0]}{N}\, n + x[0]., \qquad n = 0, 1, 2,\ldots, N \tag{3.51}$$

é a reta que passa pelos pontos $x[0]$ e $x[N]$. Além disso $xr[n]$ é seqüência obtida por rotação de eixos.

- Aplicar a transformada seno discreta sobre a seqüência $xr[n]$, para se obter a seqüência transformada $XR[k]$, $k = 0, 1, 2, \ldots, N$, isto é:

$$XR[k] = \sqrt{\frac{2}{N}} \sum_{n=0}^{N} xr[n] \operatorname{sen} \frac{\pi}{N} kn, \qquad 0 \le k \le N\,. \tag{3.52}$$

- Gerar uma nova seqüência *Y[k], k = 0, 1, 2, ..., PN,* de comprimento *PN+1*, obtida pela inserção de *(P-1)* zeros no final seqüência *XR[k].*

$$Y[k] = \begin{cases} XR[k], & 0 \le k \le N \\ 0, & N+1 \le k \le PN \end{cases} \tag{3.53}$$

- Calcular a transformada seno discreta inversa da seqüência *Y[k]*, usando-se a expressão:

$$y[n] = \sqrt{\frac{2}{N}} \sum_{k=0}^{PN} Y[k] \operatorname{sen} \frac{\pi}{PN} kn, \qquad 0 \le k \le PN \tag{3.54}$$

- Realizar o procedimento de pós-rotação para se obter finalmente a seqüência interpolada, $x_i[n]$, de comprimento *PN+1*, dada por:

$$x_i[n] = y[n] + r_i[n], \qquad n = 0, 1, 2, ..., PN. \tag{3.55}$$

onde:

$$r_i[n] = \frac{x[N] - x[0]}{PN} \; n + x[0], \qquad n = 0, 1, 2, ..., PN. \tag{3.56}$$

é a reta interpolada.

3.6. Implementação e Comparação dos Métodos de Interpolação por Transformada em Sinais unidimensionais.

Nesta seção serão apresentados os resultados obtidos pela implementação dos três métodos de interpolação de sinais unidimensionais, apresentados na seção anterior. Os sinais de teste são gerados para várias taxas de amostragem, interpolados pelos três métodos de interpolação e os resultados são comparados.

Seja o sinal de teste *x[n]*, constituído de uma soma de funções senoidais, definido por:

$$x[n] = sen(0{,}2n\pi) + sen(0{,}05n\pi) + sen(0{,}02n\pi) + sen(0{,}1n\pi), \quad n = 0, 2, 3, \ldots, N\text{-}1. \tag{3.57}$$

A figura 3.9 mostra o gráfico de um trecho do sinal de teste, que será utilizado. Foram tomadas inicialmente 32 amostras desse sinal, as quais estão assinaladas no gráfico da figura 3.9. Foram, então, aplicados os três métodos de interpolação, DSTr, DCT-I e DCT/FFT, no sinal amostrado, constituído de 32 amostras, para se obter um sinal interpolado com 63 amostras, o que significa

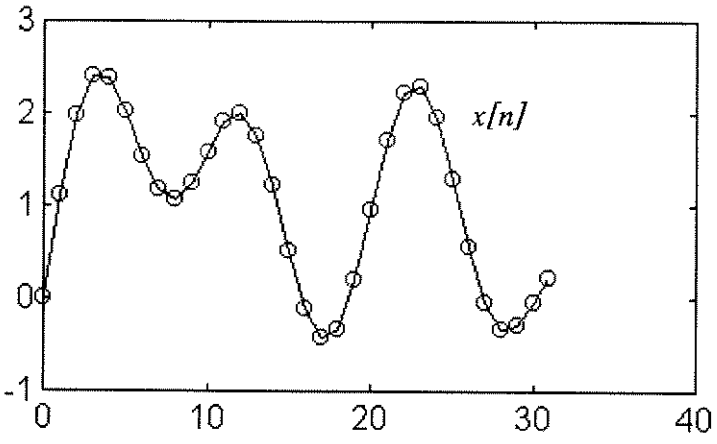


Figura 3.9. Sinal de teste usado para interpolação por DSTr, DCT-I e DCT/FFT, com taxa de interpolação igual a 2.

que foi criada uma amostra entre cada par de amostras originais consecutivas, ou seja, uma interpolação com taxa igual a 2. Os parâmetros usados para efeito de comparação são a relação sinal-ruído (SNR) e a relação sinal-ruído de pico (PSNR). O cálculo da SNR e PSNR é feito entre o sinal interpolado $x_i[n]$ e o sinal $xs[n]$, que é o sinal original $x[n]$, re-amostrado com uma taxa igual a duas vezes a taxa inicialmente amostrada.

A relação sinal-ruído (SNR) é calculada usando-se a expressão [20]:

$$SNR_{dB} = 10\log_{10}\left(\frac{\sum_{n=0}^{N-1} xs[n]^2}{\sum_{n=0}^{N-1}(xs[n]-x_i[n])^2}\right) \tag{3.58}$$

A relação sinal-ruído de pico (PSNR) é calculada entre o sinal re-amostrado $xs[n]$ e o sinal interpolado $x_i[n]$, e usando-se a expressão:

$$PSNR_{dB} = 10\log_{10}\left(\frac{255^2}{\sum_{n=0}^{N-1}(xs[n]-x_i[n])^2}\right) \tag{3.59}$$

Os resultados comparativos para esse primeiro teste são mostrados na tabela 3.1. Como pode ser visto na tabela, a relação sinal-ruído de interpolação usando a DSTr foi superior ao método com DCT-I de Wang em 31,83 dB e 18,42 dB em relação ao método DCT/FFT, para uma taxa de interpolação igual a 2, quando 32 amostras do sinal de teste foram interpoladas para se obter um sinal com 63 amostras.

TABELA 3.1. Comparação entre os métodos de interpolação para o sinal de teste usando os três métodos de interpolação DSTr, DCT/FFT e DCT-I para uma taxa de interpolação igual a 2.

MÉTODO DE INTERPOLAÇÃO	SNR(dB)	PSNR(dB)
DSTr	64,22	109,39
DCT/FFT	45,80	90,96
DCT-I	32,39	77,55

O segundo teste será realizado agora tomando-se 16 amostras do sinal $x[n]$, conservando sempre o mesmo intervalo do sinal de teste. Em seguida o sinal amostrado é interpolado pelos três métodos, DSTr, DCT/FFT e DCT-I, usando-se uma taxa de interpolação igual a 16, isto é, 15 amostras são interpoladas entre cada par de amostras originais consecutivas, para se obter um sinal interpolado $x_i[n]$, de comprimento igual a 241.

A tabela 3.2 mostra a relação sinal-ruído e a relação sinal-ruído de pico para esse teste. Como pode ser visto nessa tabela o método usando DSTr é muito superior ao método da DCT-I, onde a diferença da relação sinal-ruído entre os dois métodos é em torno de 45 dB.

TABELA 3.2. Comparação entre os métodos de interpolação para o sinal de teste usando os três métodos de interpolação DSTr, DCT/FFT e DCT-I para uma taxa de interpolação igual a 16.

MÉTODO DE INTERPOLAÇÃO	SNR(dB)	PSNR(dB)
DSTr	67,47	112,44
DCT/FFT	35,32	80,29
DCT-I	22,96	67,97

O terceiro teste consiste em se tomar o sinal de teste e realizar várias interpolações em diferentes taxas de amostragens, utilizando sempre uma taxa de interpolação igual a 2. Dessa forma, serão tomados inicialmente 9 amostras do sinal de teste, no mesmo intervalo do teste anterior, que corresponde a uma taxa de amostragem aproximadamente igual a taxa de Nyquist. As interpolações foram feitas com o sinal de teste amostrado para as taxas aproximadamente iguais a f_N , $2f_N$, $4f_N$, $8f_N$ *e* $16f_N$, onde f_N é a taxa de amostragem de Nyquist. O gráfico da figura 3.10 mostra o sinal de teste amostrado, para a taxa de Nyquist.

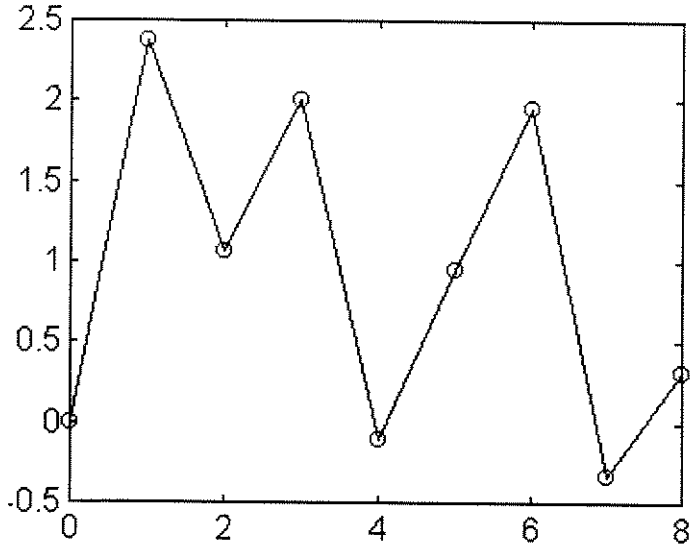


Figura 3.10. Sinal de teste com 9 amostras.

O gráfico da figura 3.11 mostra a relação sinal-ruído de interpolação para diferentes taxas de amostragem. Como pode ser visto, o método da DSTr torna-se mais eficiente à medida que a taxa de amostragem do sinal a ser interpolado cresce. Todas as interpolações, que estão representadas nesse gráfico, foram feitas com taxa de interpolação igual a 2.

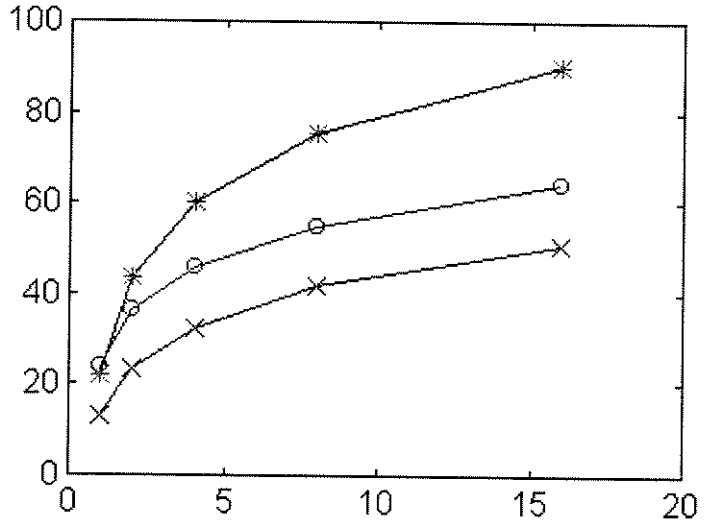


Figura 3.11. Relação sinal-ruído de interpolação para o sinal de teste para taxas de amostragem f_N, $2f_N$, $4f_N$, $8f_N$ e $16f_N$, usando DSTr(*), DCT/FFT(o) e DCT-I(x).

Um segundo sinal de teste sinal, constituído pela soma de funções senoidais, é definido por:

$$x[n] = sen(\ 0{,}2n\pi\ ) + cos(\ 0{,}05n\pi\ ) + sen(\ 0{,}02n\pi\ ) + cos(\ 0{,}1n\pi\ ), \quad n = 0,\ 2,\ 3,\ldots,\ N\text{-}1. \tag{3.60}$$

Esse sinal foi utilizado para interpolação, tomando-se 16 amostras que foram interpoladas para se obter um sinal de comprimento igual a 241, de modo que 15 amostras foram criadas entre cada par de amostras consecutivas. A figura 3.12 mostra o trecho do sinal de teste, onde estão assinaladas as amostras utilizadas. A tabela 3.3 mostra a relação sinal-ruído e a relação sinal-ruído de pico obtidas para os três métodos de interpolação, DSTr, DCT/FFT e DCT-I.

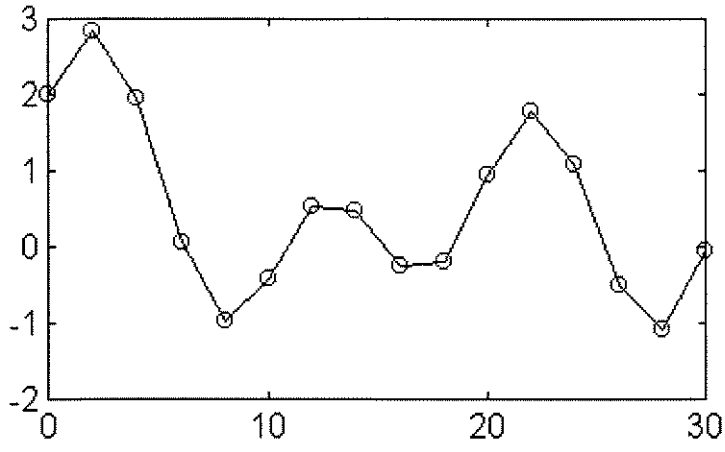


Figura 3.12. Segundo sinal de teste, com 16 amostras.

TABELA 3.3. Comparação entre os métodos de interpolação para o segundo sinal de teste, usando os três métodos de interpolação DSTr, DCT/FFT e DCT-I para uma taxa de interpolação igual a 16.

MÉTODO DE INTERPOLAÇÃO	SNR(dB)	PSNR(dB)
DSTr	48,68	94,99
DCT/FFT	34,51	80,82
DCT-I	23,00	69,31

3.7. Interpolação de Imagens por Transformada.

Nesta seção serão apresentados quatro métodos de interpolação de imagens. O primeiro é uma extensão do método de Wang, usando a DCT-I; o segundo método faz a interpolação usando a DCT/FFT, apresentado na seção anterior, o terceiro método usa a transformada seno discreta com rotação de eixos[22, 23]. O quarto método é uma extensão do primeiro método com a eliminação de extrapolações[23]. A interpolação de imagens é baseada nos métodos de interpolação apresentados nas seções anteriores para sinais unidimensionais. Esses métodos serão aplicados nas linhas e colunas com algumas particularidades, a fim de se tornarem apropriados para interpolação de imagens. A interpolação é feita em blocos de tamanhos 8x8, 16x16, etc. A seguir serão apresentados os métodos de interpolação de imagens.

Método 1. Interpolação de imagens usando DCT-I.

Esse método é a extensão bidimensional do método de interpolação com DCT-I, apresentado para sinais unidimensionais. No caso bidimensional, cada bloco de tamanho *(Bl+1)x(Bl+1)* é interpolado para se gerar um bloco de tamanho *(pBl+1)x(pBl+1)*, onde *p* é a taxa de interpolação. Segue a descrição do algoritmo.

- Dividir a imagem *x[m,n], m = 0, 1, 2,...,M-1, n = 0, 1, 2,...,N-1,* de tamanho *MxN*, em blocos de tamanho *BlxBl.*
- Tomar um bloco de tamanho *(Bl+1)x(Bl+1)*, onde a primeira e a última linha pertencem a blocos adjacentes.
- Fazer a interpolação do bloco aplicando-se a DCT-I modificada nas linhas e colunas para se obter um bloco interpolado de tamanho *(pBl+1)x(pBl+1).*
- Eliminar a última linha e última coluna, para se obter um do bloco interpolado de tamanho *pBl x pBl.*
- Repetir o procedimento para os demais blocos.

Método 2. Interpolação de imagens usando DCT/FFT.

Nesse método, cada bloco de tamanho *BlxBl* será interpolado para gerar um bloco de tamanho *pBl x pBl*, onde *p* é a taxa de interpolação. Segue a descrição do algoritmo.

- Dividir a imagem *x[m,n], m = 0, 1, 2,...,M-1, n = 0, 1, 2,...,N-1,* de tamanho *MxN*, em blocos de tamanho *BlxBl.*
- Aplicar a DCT/FFT-2D em cada bloco, que corresponde a aplicação de transformada unidimensional em cada linha e cada coluna do bloco, como definido em (3.41) e (3.42), para se obter o bloco transformado.
- Acrescentar *2(p-1)Bl* zeros em cada linha e coluna de cada bloco transformado, a fim de se obter blocos de tamanho *2pBl x 2pBl*, onde *p* é a taxa de interpolação.
- Calcular a transformada DCT/FFT-2D inversa para cada bloco.
- Tomar os blocos interpolado formado pelas primeiras *pBl x pBl* amostras de cada bloco obtido no ítem anterior, para se obter finalmente a imagem interpolada de tamanho *pM x pN.*

Método 3: Interpolação de imagens usando transformada seno discreta com rotação de eixos.

Nesse método de interpolação uma imagem *x[m,n], m = 0, 1, 2,...,M-1, n = 0, 1, 2,...,N-1,* de tamanho MxN, é dividida em blocos de tamanho *BlxBl*, e a interpolação por transformada seno discreta com rotação de eixo é aplicada sobre as linhas e colunas de cada bloco. O algoritmo é descrito a seguir.

- Dividir a imagem *x[m,n], m = 0, 1, 2,...,M-1, n = 0, 1, 2,...,N-1,* de tamanho MxN, em blocos de tamanho *BlxBl.*
- Tomar um bloco de tamanho *(Bl+1)x(Bl+1)*, onde a primeira e a última linha pertencem a blocos adjacentes.
- Aplicar o procedimento de pré-rotação de eixos nas linhas e colunas do bloco, de modo que as primeiras e últimas linhas e colunas são nulas, e aplicar a transformada seno discreta bidimensional.
- Acrescentar *(p-1)Bl* zeros em cada linha e coluna, onde p é a taxa de interpolação, e calcular a transformada seno discreta inversa.
- Aplicar o procedimento de pré-rotação seguido da interpolação pela transformada seno discreta sobre as primeiras e últimas linhas e colunas do bloco original.
- Aplicar o procedimento de pós-rotação de eixos sobre o bloco interpolado e eliminar a última linha e a última coluna para obter um bloco interpolado tamanho *pBlxpBl.*
- Aplica o mesmo procedimento aos demais blocos.

Método 4. Interpolação de imagens usando DCT/FFT com eliminação de extrapolações.

Nesse método, que será referido como DCTd, no decorrer dese trabalho, a interpolação pela transformada DCT/FFT é aplicada em blocos de tamanho *(Bl+1)x(Bl+1)*, seguido da eliminação das *p* últimas linhas e *p* últimas colunas para se obter um bloco interpolado de tamanho *pBlxpBl.* O algoritmo é descrito a seguir.

- Dividir a imagem *x[m,n], m = 0, 1, 2,...,M-1, n = 0, 1, 2,...,N-1,* de tamanho *MxN*, em blocos de tamanho *BlxBl.*
- Tomar um bloco de tamanho *(Bl+1)x(Bl+1)*, onde a primeira e a última linha pertencem a blocos adjacentes.
- Aplicar os procedimentos de interpolação por DCT/FFT-2D, para se obter um bloco interpolado de tamanho *p(Bl+1) x p(Bl+1),* onde *p* é a taxa de interpolação.

- Eliminar as p últimas linhas e p últimas colunas para obter um bloco interpolado de tamanho $pBl \times pBl$.
- Repetir o procedimento para os demais blocos.

3.8. Aplicação dos Métodos de Interpolação de Imagens por Transformada

Nesta seção os métodos de interpolação de imagens apresentados na seção anterior serão aplicados na interpolação de imagens e os resultados serão, então, comparados. Inicialmente, uma imagem de teste será gerada por amostragem, e os quatro métodos de interpolação serão aplicados nessa imagem para obter imagens interpoladas, as quais serão comparadas com uma outra imagem de teste gerada por amostragem superior. Posteriormente os métodos serão aplicados a outras imagens conhecidas da literatura.

A imagem de teste será gerada por amostragem do sinal bidimensional, de baixa freqüência e de faixa limitada, constituído pela soma de funções senoidais bidimensionais [24], definida por :

$$s(x_1, x_2) = sen[0{,}0005(x_1^2 + x_2^2)] + \cos[0{,}0008(x_1^2 + x_2^2)] + sen(0{,}0002(x_1^2 + x_2^2)] \quad (3.61)$$

Dessa forma uma imagem de tamanho MxN, é gerada pela matriz dos pontos amostrados do sinal bidimensional $s(x_1,x_2)$, para $x_1 = 0, 1, 2,...,M$ e $x_2 = 1, 2,...,N$. A imagem de tamanho 256x256 mostrada na figura 3.13 foi gerada por amostragem do sinal $s(x_1,x_2)$, para x_1 , $x_2 = 0, 1, 2,...,256$. Essa imagem será usada como referência e as demais serão imagens subamostradas, isto é, com intervalo de amostragem maior que um, ou superamostradas, com intervalos menor que um.

Como uma primeira aplicação, os quatro métodos de interpolação de imagens, serão aplicados em duas imagem de teste; uma de tamanho 128x128 e outra de 256x256, para se obter imagens interpoladas de tamanhos 255x255 e 511x511, respectivamente, usando blocos de tamanho 32x32. A figura 3.14 mostra as imagens de tamanho 255x255 obtidas por interpolação. Como pode ser observado, as imagens interpoladas não apresentam diferenças visíveis, no entanto, as diferenças aparecem na Tabela 3.4, que mostra a relação sinal-ruído (SNR) e a relação sinal-ruído de pico (PSNR) de interpolação entre a imagem de teste gerada

por amostragem e a imagem interpolada, para cada método. A Tabela 3.4 mostra que os métodos de interpolação por DSTr e DCTd apresentam melhor desempenho, em relação aos outros dois métodos, para essa situação, principalmente quando cresce a taxa de amostragem, pois a diferença da relação sinal ruído de interpolação, entre o métodos por DSTr e DCT-I cresce de aproximadamente 15 para 18 dB, quando cresce a taxa de amostragem da imagem de teste.

Em uma segunda aplicação, uma imagem teste de tamanho 64x64 será interpolada pelos quatro métodos, DSTr, DCTd, DCT/FFT e DCT-I para se obter uma imagem de tamanho 505x505. A figura 3.15 mostra as imagens originais obtidas por amostragem da imagem de teste, a primeira de tamanho 64x64 e a segunda, de tamanho 505x505. Na figura 3.16 são mostradas as imagens de tamanho 505x505 obtidas por interpolação, na qual pode ser visto que a imagem interpolada por DCT/FFT começa a apresenta efeitos de bloqueamento visíveis, o que não ocorre com as demais. A tabela 3.5 mostra a relação sinal-ruído e a relação sinal-ruído de pico de interpolação. Como pode ser observado, a interpolação por DSTr continua a apresentando um melhor desempenho seguido do método de interpolação por DCTd.

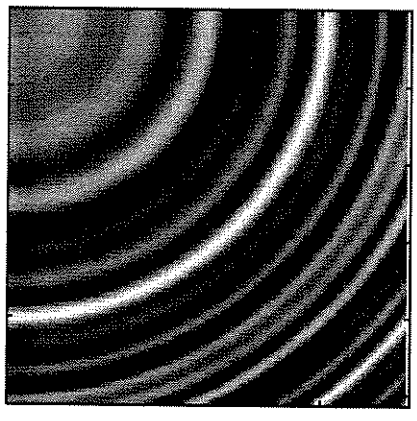
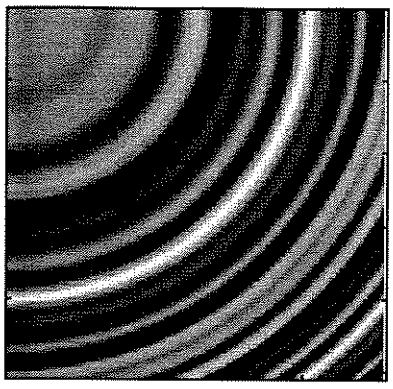
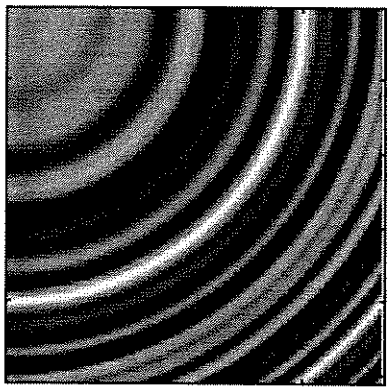


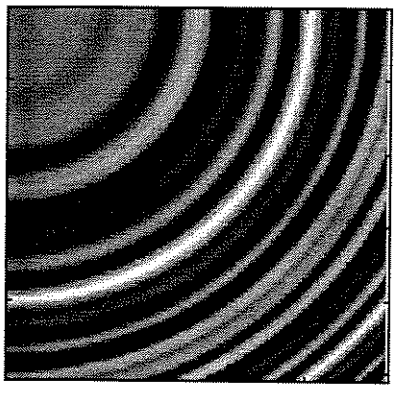
Figura 3.13. Imagem de teste gerada por amostragem do sinal $s(x_1,x_2)$, com 256x256 amostras.



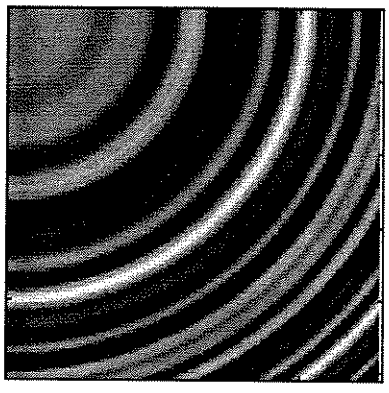
a) Imagem interpolada por DSTr



b) Imagem interpolada por DSTd



c) Imagem interpolada por DCT/FFT

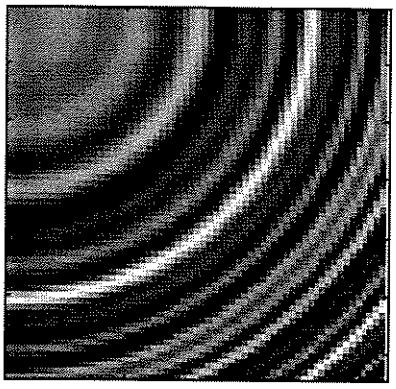


d) Imagem interpolada por DCT-I

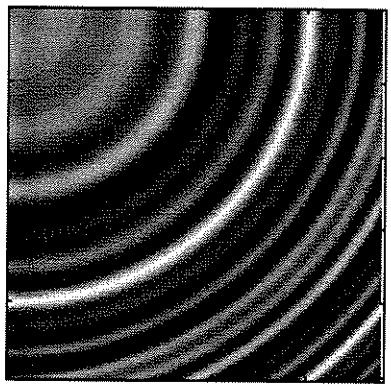
Figura 3.14. Imagens obtidas por interpolação da imagem de teste de 128x128 para 255x255.

TABELA 3.4. Comparação entre os métodos de interpolação para a imagem de teste para interpolar a imagens de 128x128 para 255x255 e de 256x256 para 511x511.

	(128x128) => (255x255)		(256x256) => (511x511)	
MÉTODO DE INTERPOLAÇÃO	**SNR(dB)**	**PSNR(dB)**	**SNR(dB)**	**PSNR(dB**
DSTr	49,27	95,73	58,66	105,12
DCTd	45,41	91,88	52,74	99,20
DCT/FFT	34,90	81,36	40,85	87,31
DCT-I	34,33	80,79	40,71	87,18

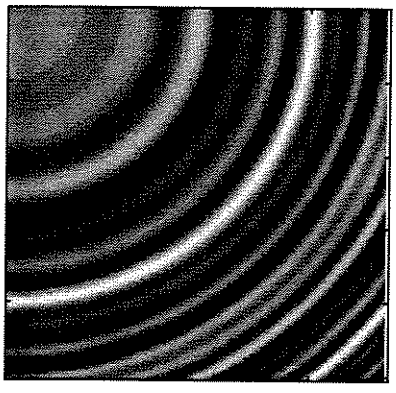


a) Imagem de teste 64x64

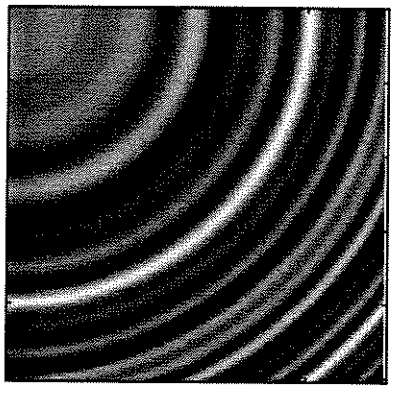


b) Imagem de teste 505x505

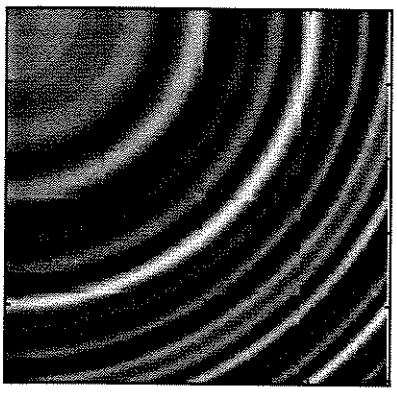
Figura 3.15. Imagens originais obtidas por amostragem.



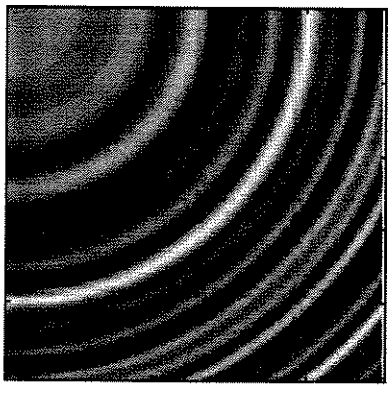
a) Imagem interpolada por DSTr.



b) Imagem interpolada por DCTd.



c) Imagem interpolada por DCT/FFT.



d) Imagem interpolada por DCT-I.

Figura 3.16. Imagens obtidas por interpolação da imagem de teste de 64x64 para 505x505.

TABELA 3.5. Comparação entre os métodos de interpolação, usando DSTr, DCTd, DCT/FFT e DCT-I para interpolar a imagem de teste 64x64 para 505x505.

MÉTODO DE INTERPOLAÇÃO	SNR(dB)	PSNR(dB)
DSTr	37,08	83,54
DCTd	34,27	80,73
DCT/FFT	21,04	67,50
DCT-I	24,40	70,85

Os quatro métodos de interpolação são agora aplicados na imagem Lena de tamanho 256x256 para se obter imagens interpoladas de tamanho 512x512. O tamanho dos blocos utilizados nas interpolações são de 4, 8, 16, 32 e 64. Essas imagens interpoladas são comparadas com uma imagem original. A Tabela 3.5 mostra a relação sinal-ruído (SNR) para cada método obtidas para cada método. A Tabela 3.6 apresenta uma comparação entre a relação sinal-ruído de pico(PSNR). Para se obter a imagem interpolada de tamanho 512x512, foi gerada uma extrapolação da última linha e última coluna para os métodos DCT-I, DCTd e DSTr. Esse procedimento foi implementado repetindo a última linha e a última coluna da imagem Lena256 antes da interpolação, gerando dessa forma uma imagem interpolada de tamanho 513x513, da qual foram eliminadas a última linha e a última coluna para se obter finalmente a imagem Lena512 interpolada. Esse procedimento não afetou o desempenho dos três métodos de interpolação, pois introduziu um erro desprezível, em torno de 0,003 dB.

Os quatro métodos de interpolação são agora aplicados uma subimagem de tamanho 128x128 (Lena128) obtida da imagem Lena512, a qual é mostrada na figura 3.17. Essa imagem é interpolada por uma taxa igual a 8, para se obter uma imagem de tamanho 1017x1017, que é uma imagem 1024x1024 subtraída as extrapolações. Um detalhe da imagem original é mostrado na figura 3.18. As figuras 3.19, 3.20, 3.21 e 3.22 mostram os detalhes das imagens interpoladas pelos quatro métodos, DSTr, DSTd, DCT/FFT e DCT-I, respectivamente. Pode-se observar a presença de novamente o efeito de bloco visível n0 detalhe da imagem interpolada pelo método DCT/FFT.

Tabela 3.6. Relação sinal-ruído (SNR) entre as imagens Lena512 original e Lena512 interpolada.

MÉTODO DE	TAMANHO DO BLOCO				
INTERPOLAÇÃO	4	8	16	32	64
DSTr	29,25	29,69	29,91	30,01	30,06
DCTd	29,45	29,80	29,97	30,04	30,07
DCT/FFT	27,41	28,62	29,34	29,69	29,92
DCT-I	27,51	27,42	27,38	27,35	27,34

Tabela 3.7. Relação sinal-ruído de pico (PSNR) entre as imagens Lena512 original e Lena512 interpolada.

MÉTODO DE	TAMANHO DO BLOCO				
INTERPOLAÇÃO	4	8	16	32	64
DSTr	34,91	35,35	35,57	35,67	35,71
DCTd	35,10	35,45	35,62	35,70	35,73
DCT /FFT	33,07	34,27	34,99	35,35	35,57
DCT-I	33,17	33,08	33,03	33,01	32,99

Figura 3.17. Imagem Lena128 original

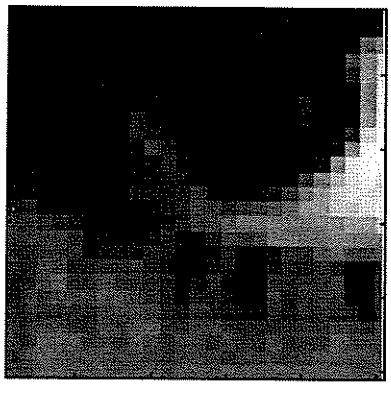


Figura 3.18. Detalhe imagem Lena128 original

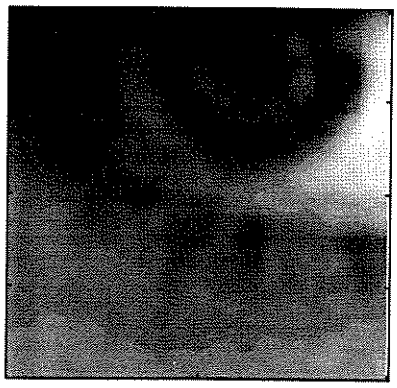
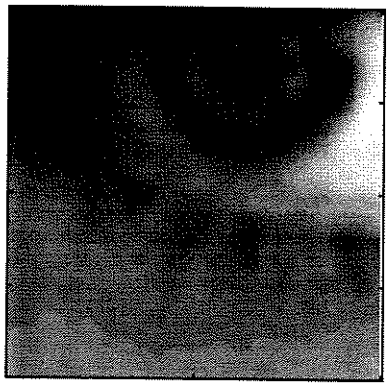



Figura 3.19. Detalhe da imagem interpolada por DSTr (128x128) => (1017x1017).

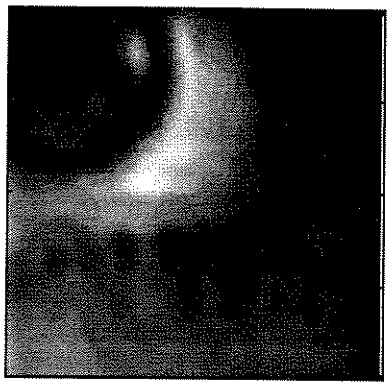


Figura 3. 20. Detalhe da imagem interpolada por DCTd (128x128) => (1017x1017).

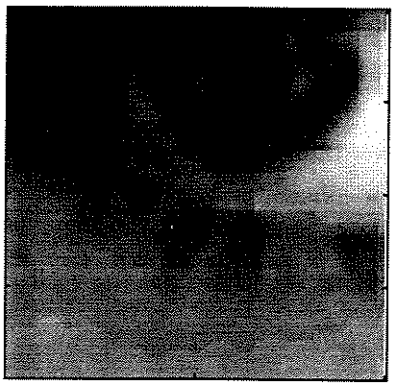


Figura 3.21 Detalhe da imagem interpolada por DCT/FFT (128x128) => (1017x1017).

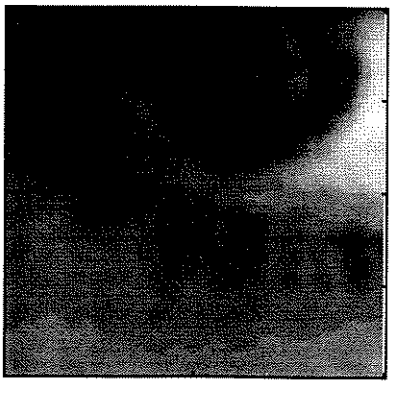
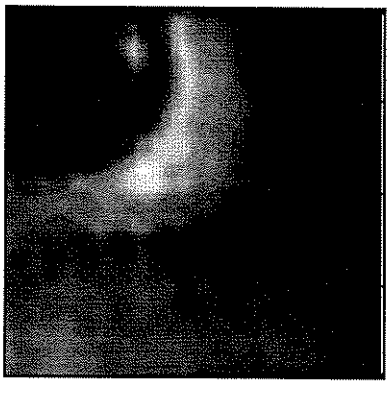



Figura 3.22. Detalhe da imagem interpolada por DCT-I (128x128) => (1017x1017).

CAPÍTULO 4

Codificação de Imagens por Transformadas

A codificação digital de imagens é um procedimento de representação de uma imagem como uma seqüência de bits, com o menor número de bits possível, preservando os níveis de qualidade e inteligibilidade necessárias para uma dada aplicação [1, 4, 20]. Esse fato se deve principalmente ao custo de armazenamento ou transmissão de imagens digitalizadas, que aumentará inevitavelmente com o tamanho da imagem. A codificação de imagens pode ser feita por várias técnicas conhecidas e apresentadas na literatura científica. Os algoritmos que implementam as técnicas de codificação de imagens se classificam, em três diferentes classes, dependendo do aspecto escolhido para se codificar a imagem [4, 20]. A primeira classe de algoritmos inclui os codificadores de formas de ondas; neste caso, a intensidade ou alguma variação da intensidade da imagem é codificada. Esses algoritmos são implementados diretamente no domínio espacial, dos quais fazem parte a modulação por codificação de pulso (PCM), modulação delta (DM), modulação diferencial por codificação de pulso(DPCM), etc. A segunda classe inclui os algoritmos que codificam uma imagem através de modelos. Nessa classe uma imagem ou alguma parte de uma imagem é modelada e os parâmetros do modelo são codificados. A imagem é então sintetizada a partir dos parâmetros codificados desse modelo. A terceira classe de algoritmos incluem os codificadores por transformada. Nessa classe de algoritmos a imagem é transformada para um outro domínio por uma transformada, tal como a transformada de Fourier, transformada coseno, transformada seno, etc. Nesse domínio, os coeficientes transformados são processados e codificados. A codificação por transformada será o objetivo principal deste capítulo e dos capítulos seguintes.

Os codificadores por transformadas são implementados em três diferentes estágios. No primeiro estágio a imagem é mapeada para o domínio da transformada através de uma transformada linear inversível, tal como DFT, DCT, KLT, etc. No segundo estágio os coeficientes transformados são quantizados. A quantização dos coeficientes pode ser feita usando-se quantização escalar ou vetorial. Na quantização escalar cada coeficiente transformado é independentemente representado pelo número de bits de acordo com a tabela de alocação, que pode ser previamente estabelecida, ou determinada com base nas estatísticas dos coeficientes transformados. Na quantização vetorial, dois ou mais coeficientes são

conjuntamente representados por um escalar pertencente a um conjunto de seqüências previamente estabelecidas denominado de dicionário. No terceiro estágio, os níveis de saída do quantizador são representados por palavras-código de comprimento fixo ou variável, sempre procurando minimizar o comprimento médio das palavras-código, que representam os coeficientes quantizados.

4.1. Transformadas Usadas na Codificação de Imagens.

As técnicas de codificação por transformadas tem como objetivo reduzir a correlação que existe entre as intensidades dos pixels da imagem, de modo que as informações redundantes não sejam codificadas. Exploram ainda o fato de que para uma imagem típica, grande parte da energia se concentra em um pequeno número de coeficientes transformados. Essa característica das transformadas usuais para codificação de imagens, é denominada de propriedade de compactação de energia, o que permite recuperar uma imagem de uma pequena fração de coeficientes transformados. Essa é uma das propriedades mais importantes para uma transformada que se presta para codificação de imagens. Daí a razão, pela qual a transformada cosseno discreta (DCT) é uma das transformadas mais utilizadas em processamento e codificação de imagens. Outras transformadas, com suas variações, vem sendo usadas recentemente em trabalhos publicados em congressos e revistas especializadas. São exemplos disto, a transformada wavelet [25, 26], LOT (lapped orthogonal transform), transformada seno discreta (DST) transformada seno discreta com rotação de eixos (DSTr) [26], etc.

O diagrama de blocos da figura 3.1 mostra um codificador de imagens. A imagem $x[n_1,n_2]$ na entrada do codificador é transformada, e seus coeficientes transformados $X[k_1,k_2]$ são quantizados, produzindo uma versão aproximada dos coeficientes $\hat{X}[k_1,k_2]$, que são codificados, sendo atribuído uma palavra código a cada coeficiente. O conjunto de palavras-código é transmitido ou armazenado. Para se recuperar a imagem, o conjunto de palavras-código é decodificado para, então, se obter os coeficiente quantizados $\hat{X}[k_1,k_2]$. A transformada inversa é calculada e uma versão aproximada da imagem $\hat{x}[n_1,n_2]$ é obtida.

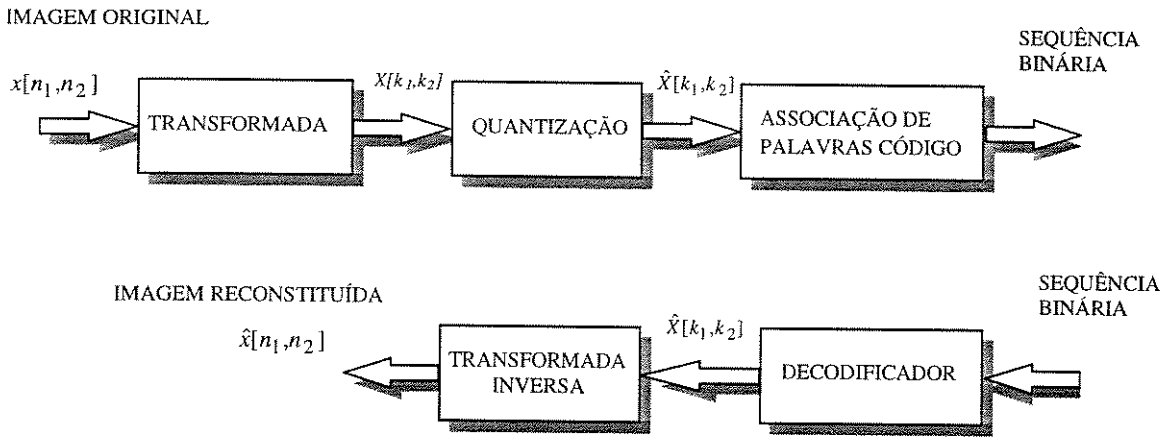


Figura 4.1 . Diagrama de blocos de um codificador de imagens por transformadas.

As transformadas usadas na codificação de imagens devem apresentar algumas propriedades que são importantes para o bom desempenho da transformada. A seguir são apresentadas algumas dessas propriedades.

- As transformadas devem ser lineares, de modo que o par de transformadas (direta e inversa) podem ser expressas como:

$$X[k_1,k_2] = \sum_{n_1=0}^{N_1-1}\sum_{n_2=0}^{N_2-1} x[n_1,n_2]\;\; a[n_1,n_2;k_1,k_2]$$

$$0 \le k_1 \le N_1-1,\;\; 0 \le k_2 \le N_2-1 \qquad (4.1a)$$

$$x[n_1,n_2] = \sum_{k_1=0}^{N_1-1}\sum_{k_2=0}^{N_2-1} X[k_1,k_2]\;\; b[n_1,n_2;k_1,k_2]$$

$$0 \le n_1 \le N_1-1,\;\; 0 \le n_2 \le N_2-1 \qquad (4.1b)$$

onde, $x[n_1,n_2]$ é seqüência bidimensional de tamanho N_1xN_2 que representa uma imagem digitalizada. $X[k_1,k_2]$ é uma seqüência bidimensional de tamanho N_1xN_2, que representa os coeficientes transformados e $a[n_1,n_2\,;\,k_1,k_2]$ e $b[n_1,n_2\,;\,k_1,k_2]$ representam as funções bases da transformada. Assim, $x[n_1,n_2]$ pode ser visto na equação (4.1b), como uma combinação linear das funções bases $b[n_1,n_2\;;\;k_1,k_2]$, cujos coeficientes transformados $X[k_1,k_2]$ representam as amplitudes dessas funções bases.

- As funções bases devem ser separáveis, de modo que a transformada direta e a transformada inversa possam ser aplicadas sobre as linhas e as colunas de uma imagem, como se fossem transformadas unidimensionais. Dessa forma, reduz-se o número de operações aritméticas necessárias para se calcular a transformada de uma imagem, bem como o esforço computacional que certamente será menor. Assim, (4.1a) e (4.1b) podem ser escritas na forma:

$$X[k_1,k_2]=\sum_{n_1=0}^{N_1-1}\sum_{n_2=0}^{N_2-1}x[n_1,n_2]\;a_1[n_1,k_1]a_2[n_2,k_2],$$

$$0\le k_1\le N_1-1,\;\; 0\le k_2\le N_2-1. \qquad (4.2a)$$

$$x[n_1,n_2]=\sum_{k_1=0}^{N_1-1}\sum_{n_2=0}^{N_2-1}X[k_1,k_2]\;\;b_1[n_1,k_1]b_2[n_2,k_2],$$

$$0\le n_1\le N_1-1,\;\; 0\le n_2\le N_2-1 \qquad (4.2b)$$

- A transformada deve ter a propriedade de reduzir a correlação entre os coeficientes transformados.
- A transformada deve apresentar a propriedade de concentrar grande parte da energia em um reduzido número de coeficientes, de modo que se possa recuperar uma imagem a partir de um pequeno número de coeficientes transformados, obedecendo a um determinado critério de fidelidade dependendo da aplicação. A KLT (Karhunem-Loève Transform) é a melhor transformada, entre todas as transformadas lineares, em termos de compactação de energia, mas não possui algoritmos eficientes para sua implementação [4]. No entanto ela serve de referência para outras transformadas.

Neste trabalho serão usadas na codificação de imagens as transformadas cosseno discreta (DCT), LOT (Lapped Orthogonal Transform) e a transformada seno discreta com rotação de eixos (DSTr). Essa última faz parte da contribuição deste trabalho. A DCT foi escolhida para ser usada na codificação por ser a transformada mais usada em processamento de imagens e por apresentar um melhor desempenho em relação a outras transformadas, servindo como um parâmetro de referência. A LOT é uma transformada que além de

BIBLIOTECA

apresentar um ótimo desempenho, tem uma característica especial que é a redução do efeito de bloqueamento. A Codificação de imagens usando a transformada seno discreta com rotação de eixos (DSTr), que está sendo proposto neste trabalho, apresenta um bom desempenho, comparável à DCT e LOT. Tem a particularidade de reduzir o efeito de bloqueamento visível em imagens reconstituídas para baixas taxas. Essas transformadas foram apresentadas no capítulo 2. Algumas propriedades dessas transformadas serão comparadas, com relação aos desempenhos quando aplicadas na codificação de imagens. As três transformadas DCT, LOT e DSTr, satisfazem as propriedades apresentadas anteriormente, isto é, são lineares e suas funções bases são separáveis; possuem uma boa propriedade de compactação de energia, sendo que cada uma apresenta uma particularidade. As transformadas são aplicadas em blocos, assim como seus coeficientes transformados são processados e codificados em blocos.

4.2. Transformadas de Imagens por Blocos

Na codificação de imagens por transformada, uma imagem $x[n_1,n_2]$, de tamanho N_1xN_2 é dividida em blocos de tamanho MxM, que em geral, é um múltiplo inteiro do tamanho da imagem. Cada um desses blocos é codificado separadamente, podendo levar em conta algumas características locais da imagem, como por exemplo, regiões uniformes, regiões com variações rápidas ou com a presença de bordas. A transformação por blocos reduz ainda, o esforço computacional, pois permite a implementação de algoritmos rápidos para calcular as transformadas, tais como DFT, DCT, DST, LOT [5] etc.

Apesar do tamanho do bloco tornar mais eficiente a implementação de algoritmos rápidos para computação dos coeficientes transformados, quando os blocos são menores e ainda de se adaptarem mais facilmente às características locais da imagem, o tamanho dos blocos não pode ser reduzido indefinidamente, pois à medida que o tamanho do bloco diminui, menor será a correlação explorada entre as intensidades dos pixels vizinhos. A correlação entre os pixels vizinhos é uma das principais características exploradas na codificação de imagens. Por conta disso, com a redução do tamanho do bloco, aumenta a correlação entre os blocos vizinhos, como cada bloco da imagem é codificado separadamente, essa correlação entre blocos não é explorada, reduzindo o desempenho da codificação por transformada. O tamanho de blocos típicos usados na codificação de imagens é de 8x8 e

16x16. A seguir são apresentadas as transformadas que serão aplicadas neste trabalho na codificação de imagens.

A transformada cosseno discreta utilizada na transformação de imagens neste trabalho é a do tipo-II (DCT-II) [7], que é a mais utilizada, por apresentar melhor desempenho, em relação às outras versões (DCT-I, DCT-III e DCT-IV), quando são aplicadas na codificação de imagens. A DCT-II é aplicada em blocos de tamanho MxM, que é múltiplo do tamanho da imagem. Assim o par de transformadas (direta e inversa) para a DCT-II de um bloco de tamanho MxM da imagem $x[n_1,n_2]$, é dada por:

$$X[k_1,k_2]=\frac{2}{M}\sum_{n_1=0}^{M-1}\sum_{n_2=0}^{M-1}x[n_1,n_2]\cos\frac{\pi}{M}k_1(n_1+\frac{1}{2})\cos\frac{\pi}{M}k_2(n_2+\frac{1}{2}),$$

$$0\le k_1\le M-1,\ \ 0\le k_2\le M-1 \tag{4.3a}$$

$$x[n_1,n_2]=\frac{2}{M}\sum_{k_1=0}^{M-1}\sum_{k_2=0}^{M-1}w_1(k_1)w_2(k_2)X[k_1,k_2]\cos\frac{\pi}{M}k_1(n_1+\frac{1}{2})\cos\frac{\pi}{M}k_2(n_2+\frac{1}{2}),$$

$$0\le n_1\le M-1,\ \ 0\le n_2\le M-1. \tag{4.3b}$$

onde:
$$w_1[k_1]=\begin{cases}\frac{1}{\sqrt{2}}, & k_1=0\\ 1, & 1\le k_1\le M-1\end{cases} \tag{4.3c}$$

$$w_2[k_2]=\begin{cases}\frac{1}{\sqrt{2}}, & k_2=0\\ 1, & 1\le k_2\le M-1\end{cases} \tag{4.3d}$$

Pode-se ver em (4.3), que as funções bases são separáveis, e portanto a transformada é separável.

A transformada seno discreta com rotação de eixos (DSTr) é aplicada em blocos de tamanho (M+1)x(M+1). Em primeiro lugar é aplicado o procedimento de pré-rotação de eixos sobre as linhas e colunas do bloco de modo que o bloco resultante possui as linhas e colunas nulas nas bordas. Sobre esse bloco é aplicada a transformada seno discreta bidimensional.

Sobre a primeira linha e a primeira coluna do bloco original aplica-se o procedimento de pré-rotação de eixos seguido da transformada seno discreta unidimensional, restando um único elemento do bloco que permanecerá na forma original, que é o referente a primeira linha e a primeira coluna. A última linha e a última coluna do bloco são descartadas, obtendo-se no final um bloco transformado de tamanho MxM. Quando a transformada é aplicada nos últimos blocos, isto é, naqueles que fazem fronteiras com as bordas da imagem, a (M+1)-ésima linha ou coluna desse bloco, será igual à primeira linha ou à primeira coluna dos primeiros blocos da imagem. É como se a imagem fosse periódica. Isso permitirá a implementação do procedimento de pós-rotação de eixos que será aplicado, após a transformação inversa, para se recuperar a imagem. Além disso, esse procedimento permitirá que apenas uma amostra em cada bloco permaneça na forma original, o que corresponde ao valor DC da transformada cosseno discreta. A transformada seno discreta utilizada na transformação é a do tipo I (DST-I) [7]. Portanto, se $xr[n_1,n_2]$ representa um bloco imagem, de tamanho MxM obtida pelo procedimentos de pré-rotação como é descrito na seção 2.5, o par de transformadas para esse bloco é dado em seguida. Sobre o sub-bloco de tamanho (M-1)x(M-1), que se inicia na segunda linha e segunda coluna, aplica-se o par de transformada seno discreta bidimensional, isto é:

$$XR[k_1,k_2]=\frac{2}{M}\sum_{n_1=1}^{M-1}\sum_{n_2=1}^{M-1}xr[n_1,n_2]\,\mathrm{sen}\frac{\pi}{M}k_1n_1\,\mathrm{sen}\frac{\pi}{M}k_2n_2,$$

$$1\le k_1\le M-1,\ \ 1\le k_2\le M-1. \qquad (4.4a)$$

$$xr[n_1,n_2]=\frac{2}{M}\sum_{k_1=1}^{M-1}\sum_{k_2=1}^{M-1}XR[k_1,k_2]\,\mathrm{sen}\frac{\pi}{M}k_1n_1\,\mathrm{sen}\frac{\pi}{M}k_2n_2,$$

$$1\le n_1\le M-1,\ \ 1\le n_2\le M-1. \qquad (4.4b)$$

sobre a primeira linha e a primeira coluna do bloco $xr[n_1,n_2]$ aplica-se o par da transformada seno discreta unidimensional:

$$XR[k]=\sqrt{\frac{2}{M}}\sum_{n=1}^{M-1}xr[n]\,\mathrm{sen}\frac{\pi}{M}kn,\qquad 1\le k\le M-1. \qquad (4.5a)$$

$$xr[n] = \sqrt{\frac{2}{M}} \sum_{k=1}^{M-1} XR[k] \operatorname{sen} \frac{\pi}{M} kn, \qquad 1 \le n \le M-1 \tag{4.5b}$$

A transformada inversa $x[n_1,n_2]$ é obtida após os procedimentos de pós-rotação de eixos, como descrito na seção 2.5, e que é aplicada em primeiro lugar sobre a primeira linha e primeira coluna do bloco, em seguida será aplicada no restante do bloco.

A LOT é, em geral, aplicada em blocos de tamanho 2Mx2M, utilizando o princípio da superposição de blocos para reduzir o efeito de bloqueamento e gerar um bloco transformado de tamanho MxM. Suas funções bases são geradas a partir de funções de bases DCT-II de comprimento igual a 2M, como foi visto no capítulo 2, onde foi descrito um algoritmo para a obtenção da funções bases para a LOT.

Para ilustrar a propriedade de compactação de energia, e a transformação por bloco, a imagem Lena, de tamanho 512x512, mostrada na figura 4.2a, foi transformada usando a DCT, LOT, DSTr e DST, com blocos de tamanho 8x8. Em seguida foi calculada a variância para cada coeficiente transformado (AC) que compõem um bloco de tamanho 8x8, obtendo-se um bloco constituído pelas variâncias. Para construir os gráficos, o bloco contendo as variâncias dos coeficientes, foi lida na forma de zig-zag, de modo que, os maiores coeficientes, em geral, são lidos em primeiro lugar. A figura 4.2b ilustra o procedimento de leitura dos coeficientes. As figuras 4.3 e 4.4 mostram os gráficos referentes às variâncias dos coeficientes transformados (AC), para a imagem Lena512, processada com DCT e LOT, respectivamente. O gráfico da figura 4.5 representa os coeficientes transformados AC, quando a imagem é transformada usando-se DSTr e a figura 4.6, mostra o gráfico para os coeficientes transformados AC, quando a transformada seno discreta é aplicada diretamente na imagem sem o procedimento de pré-rotação. Do gráfico das figuras 4.5 e 4.6, pode-se notar que o procedimento de pré-rotação de eixos, contribui para melhorar a compactação de energia dos coeficientes transformados, quando comparado com a energia dos coeficientes transformados obtidos pela transformada seno discreta sem a pré-rotação de eixos.

Figura 4.2a. Imagem Lena (512x512)

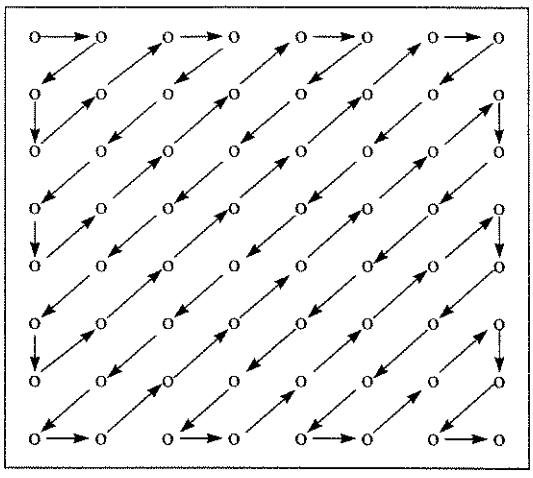


Figura 4.2b. Leitura em zig-zag da variância dos coeficientes.

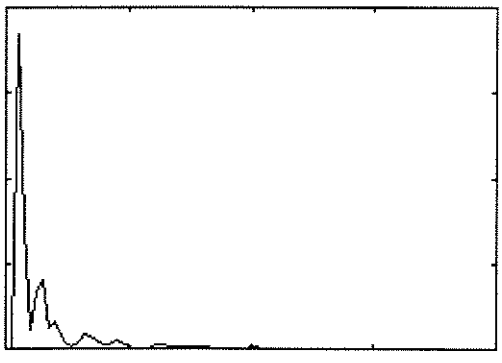


Figura 4.3. Variância dos Coeficientes(AC) DCT

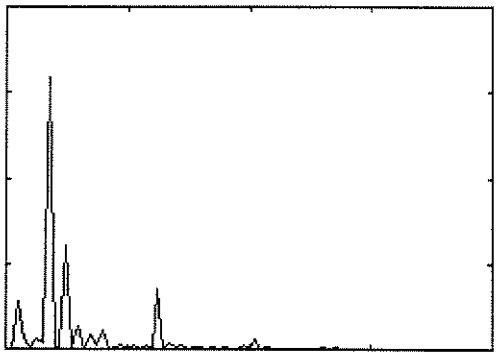


Figura 4.4. Variância dos Coeficientes(AC) LOT

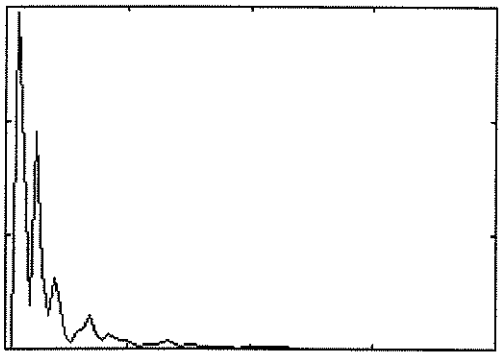


Figura 4.5. Variância dos Coeficientes(AC) DSTr

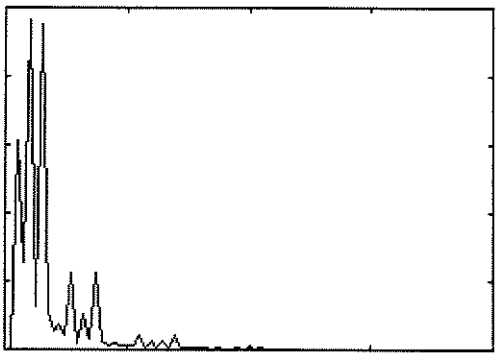


Figura 4.6. Variância dos Coeficientes(AC) DST

4.3. A Quantização

A quantização é um procedimento que visa representar as amplitudes de um sinal que possui um número infinito de diferentes amplitudes dentro de um dado intervalo de variação do sinal, por um número finito de diferentes amplitudes, usando-se um número finito de bits para representá-las. Existem vários métodos de quantização, no entanto eles ser classificados em dois tipos que são: quantização escalar e quantização vetorial.

Na quantização escalar cada amostra do conjunto de dados de entrada é independentemente mapeado para um dado nível de quantização, de acordo com as características do quantizador e em seguida, é representado por um dado número finito de bits previamente estabelecido. Na quantização vetorial, os coeficientes são agrupados em blocos e cada bloco é conjuntamente quantizado, e representado por um escalar pertencente a um conjunto de seqüências previamente estabelecidas denominado de dicionário. A quantização escalar é a mais usada porque é mais simples de ser implementada, apesar da quantização vetorial apresentar um melhor desempenho.

Os níveis de cinza que representam a intensidade dos pixels de uma imagem são, em geral, representados por oito bits, isto é, são inteiros entre 0 e 255. Quando uma imagem é transformada por DCT, LOT ou DST, os coeficientes gerados são números reais, podendo assumir uma infinidade de valores dentro de um dado intervalo. Esses coeficientes precisam ser representados por um número finito de bits, ou seja, precisam ser quantizados. Se cada coeficiente real é mapeado independentemente para algum desses níveis, então a quantização é dita ser escalar. Na quantização escalar, os coeficientes transformados são vistos simplesmente como números escalares que devem ser quantizados. A seguir será apresentada uma descrição de alguns métodos de quantização escalar que serão utilizados na quantização dos coeficientes das imagens transformadas no processo de codificação.

Seja x a entrada de um quantizador, que deve ser representada por um dentre os L níveis de quantização. Então a saída quantizador $\hat{x}$ pode ser expresso por:

$$\hat{x} = Q(x) = r_i, \qquad d_{i-1} < x \leq d_i \tag{4.6}$$

onde, Q(.) representa a operação de quantização, r_i, para $1 \leq i \leq L$, representam os L níveis de reconstituição e d_i, para $1 \leq i \leq L+1$, representam os $L+1$ níveis de decisão. A figura 4.7

mostra a função de entrada-saída para um quantizador genérico. Se a amplitude da entrada cai num intervalo entre e d_{i-1} e d_i será mapeado para um nível de reconstituição r_i. Por essa razão a operação de quantização é considerada como sendo um processo determinístico. Em geral o número L-ário é transmitido ou armazenado na forma binária. Se $L = 2^R$, então a taxa de bits é dada por:

$$R = log_2 L \quad \text{bits/amostra.} \tag{4.7}$$

O erro de quantização ou ruído de quantização é definido por:

$$e_q = x - \hat{x} \tag{4.8}$$

Para calcular a variância do erro de quantização, deve-se supor que os dados de entrada x do quantizador é uma variável aleatória com média zero, variância σ_x^2 e função densidade de probabilidade $p_x(x)$. Se os dados de entrada x não tiverem a média nula, basta subtraí-los da média, e então a entrada terá média nula. Portanto, a variância da variável aleatória x é definida por:

$$\sigma_x^2 = E[x^2] - E[x]^2 = E[x^2], \qquad \text{pois } E[x] = 0. \tag{4.9}$$

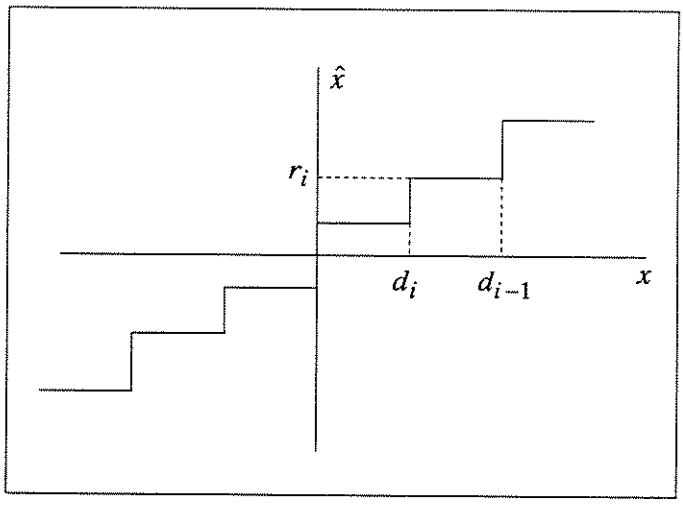


Figura 4.7. Ilustração da operação de quantização

O erro de quantização e_q é uma variável aleatória com variância dada por:

$$\sigma_q{}^2 = E[e_q{}^2] = \int_{-\infty}^{\infty} e_q{}^2 p_q(q)dq \tag{4.10}$$

O erro de quantização $e_q = x - \hat{x}$ é uma função da variável aleatória x, e portanto

$$\sigma_q{}^2 = E[e_q{}^2] = \int_{-\infty}^{\infty} (x-\hat{x})^2 p_x(x)dx \tag{4.11}$$

Mas, a integração é feita em L intervalos diferentes, que correspondem aos L níveis de reconstituição. Portanto a equação (4.11) pode ser escrita da forma:

$$\sigma_q{}^2 = E[e_q{}^2] = \sum_{i=1}^{L} \int_{d_{i-1}}^{d_i} (x-r_i)^2 p_x(x)dx \tag{4.12}$$

A variância do erro de quantização é um dos parâmetros mais importantes para avaliação do desempenho de quantizadores e em geral, é usado na forma dada pela equação (4.12).

Existem vários métodos de quantização [4, 9], porém o mais simples é o quantizador uniforme, no qual os níveis de quantização e de reconstituição são uniformemente espaçados, ou seja:

$$d_i - d_{i-1} = \Delta, \qquad 1 \le i \le L \tag{4.13}$$

$$r_i = \frac{d_i + d_{i-1}}{2}, \qquad 1 \le i \le L \tag{4.14}$$

onde Δ representa o espaçamento entre dois níveis de reconstituição ou entre dois níveis de decisão consecutivos. A figura 4.8 mostra a função que representa a entrada-saída para um quantizador uniforme de 8 níveis, onde é suposto que os coeficientes de entrada foram

normalizados, tais que $|x|$ varia entre 0 e 1. A figura 4.9 mostra a função que representa o erro de quantização para esse quantizador uniforme.

O quantizador uniforme, é extremamente fácil de ser implementado, no entanto, na maioria das vezes não é ótimo. Por exemplo, quando se trata da quantização dos coeficientes transformados de uma imagem, haverá regiões em que um dado nível de reconstrução raramente ocorrerá, enquanto outros que ocorrem com mais freqüência serão quantizados com o mesmo número de níveis. Certamente a quantização será mais eficiente se no processo de quantização forem atribuídos intervalos de decisão mais próximos, para regiões onde os coeficientes maiores ocorrerem com maior freqüência. Em seguida enfatiza-se a variância para o quantizador uniforme.

Para se calcular a variância para o quantizador uniforme supõem-se que a amplitudes dos dados de entrada x variam entre o intervalo $-x_m$ e x_m. E portanto, os intervalos de decisão e de reconstituição serão todos iguais a:

$$\Delta = \frac{2x_m}{L} = \frac{2x_m}{2^R} \tag{4.15}$$

O erro de quantização e_q , como mostrado na figura 4.9 para um caso particular de um quantizador de 8 níveis, varia na faixa:

$$-\frac{\Delta}{2} \leq e_q \leq \frac{\Delta}{2} \tag{4.16}$$

Supondo que erro de quantização é uniformemente distribuído nesse intervalo, a função densidade de probabilidade para o erro de quantização pode ser escrita como:

$$p_q(q) = \begin{cases} \dfrac{1}{\Delta}, & |q| \leq \dfrac{\Delta}{2} \\ 0, & \textit{para outros valores} \end{cases} \tag{4.17}$$

Substituindo (4.17) em (4.10) e resolvendo a integral obtém-se a variância para o erro de quantização de um quantizador uniforme.

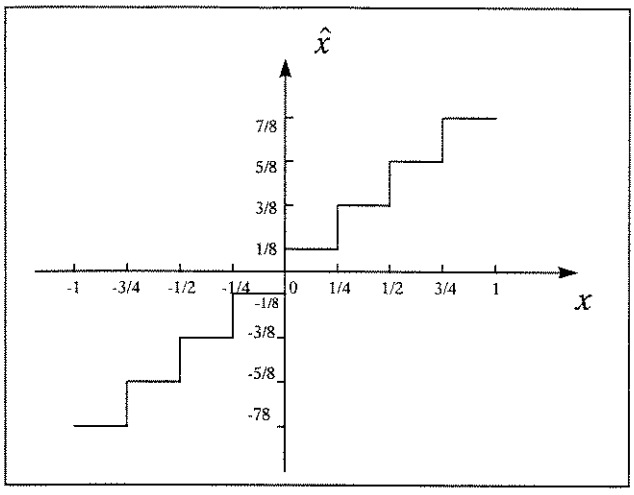


Figura 4.8. Função de entrada-saída para um quantizador uniforme de 8 níveis.

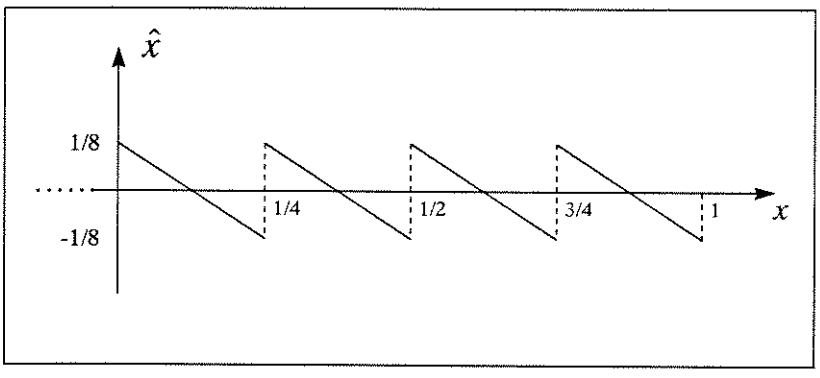


Figura 4.9. Erro de quantização para o quantizador de oito níveis.

$$\sigma_q^{\,2} = \frac{\Delta^2}{12}. \tag{4.18}$$

Substituindo (4.15) em (4.18) obtém-se:

$$\sigma_q^{\,2} = \frac{1}{3} x_m^{\,2}\, 2^{-2R}. \tag{4.19}$$

As equações (4.18) e (4.19) mostram que o desvio padrão σ_q do ruído de quantização de um quantizador uniforme cresce linearmente com o intervalo de reconstituição Δ e

decresce exponencialmente com a taxa R em bit/amostra. Deve-se observar que o conhecimento da função densidade de probabilidade do ruído de quantização, não é muito comum, no entanto existem métodos para se obter versões aproximadas dessa função [9].

Para melhorar o desempenho do quantizador, deve-se reduzir o intervalo de decisão em regiões onde houver maior ocorrência dos dados de entrada do quantizador gerando um número maior de níveis de reconstituição nessas regiões. Nas regiões onde houver menor ocorrência dos dados, os níveis de decisão e de reconstituição serão mias afastados. A quantização feita dessa forma é denominada de quantização não uniforme. Existem vários métodos de quantização não uniforme, e que dependem da aplicação.

No processo de quantização não uniforme os intervalos de decisão entre d_{i-1} e d_i e os níveis de decisão r_i, não são mais igualmente espaçados; eles estão portanto, relacionados com a estatística do conjunto dos dados de entrada a serem quantizados. A determinação dos d_i e r_i ótimos depende do critério de distorção usado. Em geral se usa como critério de distorção o erro médio quadrático mínimo (MMSE). Usando o MMSE determina-se r_k e d_k que minimiza a distorção média D, dada por:

$$D = E\left[d(\hat{x} - x)\right] = E[e_q{}^2] = E\left[(\hat{x} - x)^2\right] \tag{4.20}$$

onde x, a entrada do quantizador é uma variável aleatória com função densidade de probabilidade $p_x(x)$.

De (4.20) pode-se escrever:

$$D = E[e_q{}^2] = \sum_{i=1}^{L} \int_{d_{i-1}}^{d_i} (x - r_i)^2 p_x(x) dx \tag{4.21}$$

Para minimizar D, considera-se $d_0 = -\infty$ e $d_L = \infty$ e calcula-se:

$$\frac{\partial D}{\partial r_k} = 0, \qquad 1 \le k \le L \tag{4.22a}$$

$$\frac{\partial D}{\partial d_k} = 0, \qquad 1 \le k \le L-1 \tag{4.22b}$$

Obtendo-se:

$$r_k = \frac{\int_{d_{k-1}}^{d_k} x p_x(x) dx}{\int_{d_{k-1}}^{d_k} p_x(x) dx} \tag{4.23a}$$

$$d_k = \frac{r_k + r_{k+1}}{2} \tag{4.23b}$$

$$d_0 = -\infty \text{ e } d_L = \infty \tag{4.23c}$$

A equação (4.23a) estabelece que o nível de reconstrução r_k é o centróide da função densidade de probabilidade $p_x(x)$ no intervalo entre d_k e d_{k-1}. Os níveis de decisão d_k são dados pelo ponto médio entre dois níveis de reconstituição r_k e r_{k+1}, exceto d_0 e d_L. Essas equações apresentam a solução para o quantizador uniforme, no entanto a resolução envolve um problema não linear. As soluções dessas equações quando a entrada do quantizador x é uma variável aleatória com funções densidades de probabilidade uniforme, gaussiana ou laplaceana foram apresentadas por Lloid-Max [4]. Os quantizadores não lineares baseados no critério do erro médio quadrático mínimo como parâmetro de referência de distorção são referidos na literatura como quantizadores de Lloid-Max. No apêndice A é mostrada as tabelas para vários quantizadores ótimos de Lloid-Max com taxas de 1, 2, 3, 4, 5, 6, 7, e 8 bit/amostra, quando a entrada do quantizador é uma variável aleatória com funções densidades de probabilidade uniforme, gaussiana e laplaciana, com média zero e variância unitária. A figura 10 mostra a função de entrada-saída para um quantizador ótimo de Lloid-Max de 8 níveis, com distribuição laplaciana.

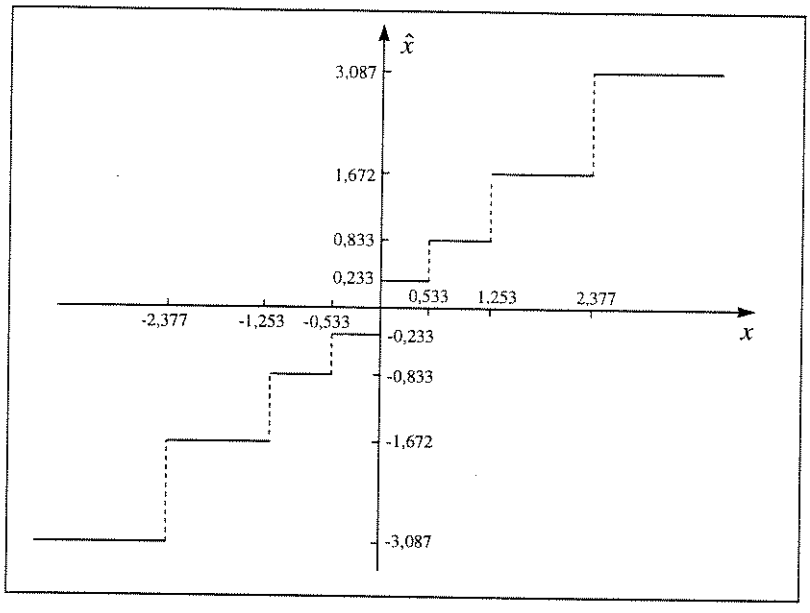


Figura 4.10. Função de entrada-saída para um quantizador ótimo de Lloid-Max de 8 níveis, com distribuição laplaciana.

4.4. Alocação de Bits e Quantização dos Coeficientes Transformados.

A quantização é uma das operações básicas da codificação de imagens. A imagem é dividida em blocos e a aplicação da transformada extrai as redundâncias existentes entre os pixels vizinhos dentro de cada bloco, de tamanho previamente escolhido, gerando coeficientes transformados com diferentes variâncias. Esses coeficientes são quantizados, de modo que, aqueles que tem menor variância serão quantizados com menor número bits e os coeficientes com maior variância serão quantizados com maior número de bits. Esse é um dos pontos básicos para se obter um bom desempenho na codificação de imagens. Os procedimentos para se alocar a taxa de bits, para cada coeficiente de um bloco transformado será descrito a seguir.

De posse dos coeficientes transformados de uma imagem de tamanho NxN, dividida em blocos de tamanho MxM, agrupa-se os coeficientes na mesma posição, obtendo-se um total de M^2 vetores, $cv(i,j)$, $i, j = 0, 1, 2, M-1$, de comprimento $n = (N/M)x(N/M) = N^2/M^2$. Em seguida determina-se as variâncias $\sigma_{i,j}^2$, $i, j = 0, 1, 2,. . .,M-1$, para cada um dos vetores, que corresponde à variância para cada coeficiente transformado $c(i,j)$ do bloco. A figura 4.11 ilustra o procedimento de agrupamento desses vetores.

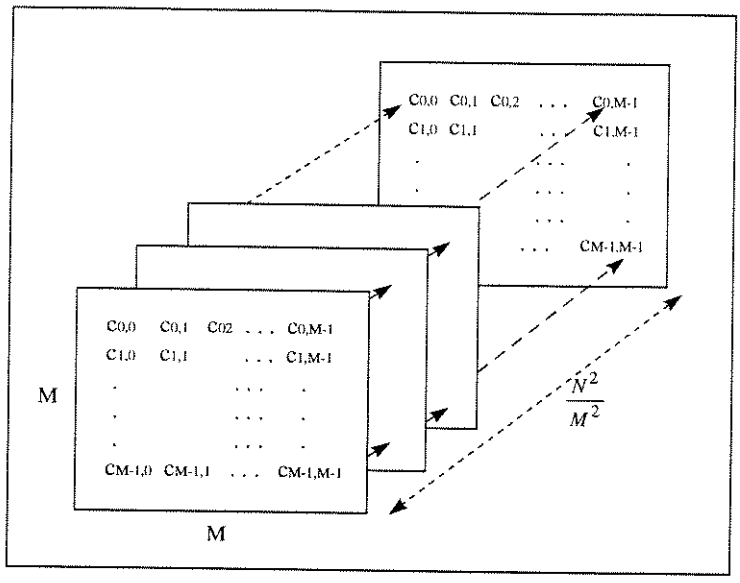


Figura 4.11. Procedimento de agrupamento dos coeficientes transformados.

Como foi mostrado na seção 4.2, devido à propriedade de compactação de energia, alguns coeficientes transformados retêm a maior parte da energia; certamente que esses coeficientes devem ser quantizados com uma taxa de bit/amostra maior que os outros coeficientes que carregam uma menor parcela de energia. A variância de cada coeficiente serve, então como parâmetro para se alocar as taxas de bit/amostra para cada coeficiente. O problema agora é o seguinte: dada uma taxa fixa de R bit/amostra como alocar as taxas para cada coeficiente?

Dado um bloco com $M^2 = MxM$ coeficientes transformados e uma taxa média de R bit/amostra, tem-se um total de M^2R bits por bloco. Deve-se então determinar o número de bits $B(i,j)$, $i,j=0, 1, 2,...,M-1$ para cada coeficiente transformado de modo que satisfaça:

$$\sum_{i=0}^{M-1}\sum_{j=0}^{M-1} B(i,j) = M^2 R \qquad (4.24)$$

A estratégia tipicamente ótima atribui uma maior taxa aos coeficientes transformados com maior variância de modo que o erro médio quadrático seja minimizado. Isso é um problema de otimização cuja solução é conhecida como regra da log-variância [5, 9, 28]. Assim o número de bits $B(i,j)$, $i, j = 0, 1, 2,...,M-1$, para os coeficientes transformados é dado por:

$$B(i,j) = \alpha + \frac{1}{2}\log_2 \frac{\sigma_{i,j}^2}{\sigma_g^2}, \quad i,j = 0, 1, 2,...,M\text{-}1 \tag{4.25}$$

onde $\sigma_{i,j}^2$ é a variância do coeficiente transformado *c(i,j)* , *i, j = 0, 1, 2,...,M-1*, e σ_g^2 é a média geométrica das variâncias dos coeficientes transformados, isto é:

$$\sigma_g^2 = \left(\prod_{i=0}^{M-1} \prod_{j=0}^{M-1} B(i,j) \right)^{1/M^2} \tag{4.26}$$

e α é um multiplicador de Lagrange, tal que satisfaça a equação (4.24). Para taxas relativamente altas, α pode ser aproximado por *R*, a taxa média em bit/amostra. Se *B(i,j)* for negativo para algum coeficiente transformado *c(i,j)*, então *B(i,j)* = 0, isto é, nenhum bit será alocado a esse coeficiente.

Com base na tabela de alocação de bits, é então feita a quantização dos coeficientes. Se a quantização usada é escalar (Lloid-Max), então cada coeficiente é individualmente quantizado de acordo com o número de bits alocados para aquela posição, obtendo-se a imagem transformada e quantizada, para uma taxa de codificação média de *R* bit/amostra. Esses coeficientes transformados e quantizados, são mapeados em um conjunto de palavras-código de tamanho fixo ou variável, para que possam ser transmitidos ou armazenados. Quando é utilizada a codificação de entropia, o conjunto de palavras-código é determinado por um código que extrai as redundâncias, tais como os códigos de Huffman, Lempel-Ziv, Aritméticos, etc, cujo objetivo é reduzir, tanto quanto possível, o número médio de bits por pixel, necessários para se representar a imagem transformada e quantizada.

Para reconstituir a imagem, o processo inverso é realizado, isto é, os coeficientes quantizados são reconstituídos pelos procedimentos inversos de quantização e a transformada inversa é aplicada sobre os coeficientes reconstituídos, obtendo-se uma réplica da imagem original, que certamente apresentará alguma distorção devido aos erros de codificação. Essa distorção é medida, em geral, pela relação sinal-ruído (SNR) e pela relação sinal-ruído de pico (PSNR), entre a imagem original e a imagem reconstituída, que para uma imagem *x(i, j)* de tamanho $N_1 x N_2$, são calculadas pela expressões:

$$SNR_{dB} = 10\log_{10}\left(\frac{\sum_{i=0}^{N_1-1}\sum_{j=0}^{N_2-1} x[i,j]^2}{\sum_{i=0}^{N_1-1}\sum_{j=0}^{N_2-1}(x[i,j]-x_r[i,j])^2}\right) \qquad (4.27)$$

$$PSNR_{dB} = 10\log_{10}\left(\frac{255^2}{\sum_{i=0}^{N_1-1}\sum_{j=0}^{N_2-1}(x[i,j]-x_r[i,j])^2}\right) \qquad (4.28)$$

onde $x_r[i, j]$ representa a imagem reconstituída.

Para ilustrar a aplicação dos procedimentos de codificação, a imagem Lena foi codificada com uma taxa média de 1 bit/pixel, usando DCT, LOT e DSTr, com blocos de tamanho 8x8 e quantização de Lloid-Max, com distribuição laplaciana. As figuras 4.12a, 4.12b e 4.12c, mostram as tabelas de alocação de bits para as imagens codificadas com DCT, LOT e DSTr. Observa-se pelas tabelas que as transformadas retratam a propriedade de concentração de energia dos coeficientes da respectiva transformada. As figuras 4.13, 414 e 4.15, mostram as imagens reconstituídas, a partir das imagens codificadas com DCT, LOT e DSTr respectivamente.

8	5	3	3	2	1	1	0
4	3	3	2	2	1	1	0
2	3	2	2	1	1	0	0
2	2	2	1	1	1	0	0
1	1	1	1	1	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

a) DCT

8	3	2	5	1	3	1	2
2	2	1	2	0	2	0	1
1	1	0	1	0	1	0	0
4	3	1	3	1	2	0	1
1	0	0	1	0	0	0	0
2	1	1	2	0	1	0	0
0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0

b) LOT

8	5	3	2	2	1	1	0
5	5	3	2	2	1	1	1
2	3	2	1	1	1	0	0
2	2	1	1	1	0	0	0
1	1	1	0	0	0	0	0
1	1	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

c) DSTr

Figura 4.12. Tabela de alocação de bits para se codificar a imagem Lena com uma taxa média de 1 bit/pixel; a) Tabela de alocação para DCT; b) Tabela de alocação para LOT; a) Tabela de alocação para DSTr .

Figura 4.13. Imagem codificada com DCT e taxa de 1bpp. SNR=26,88 dB e PSNR=32,54 dB.

Figura 4.14. Imagem codificada com LOT e Taxa de 1 bpp. SNR=27,27 dB e PSNR=32,93.

Figura 4.15. Imagem codificada com DSTr e taxa de 1bpp. SNR=26,53 dB e PSNR=32,23 dB.

Os procedimentos apresentados acima para alocação de bits, determinam uma única tabela onde se encontram especificados os números de bits $B_{i,j}$, $i,j=0, 1, 2,...,M-1$, para cada coeficiente, de uma imagem transformada em blocos de tamanho MxM, estabelecendo, então, o número de níveis de reconstrução $L_{i,j} = 2^{B_{i,j}}$ para cada coeficiente. Esta solução não é ótima, uma vez que todos os coeficientes na mesma posição são quantizados com o mesmo número de níveis, pois existem coeficientes pertencem a regiões totalmente diversificadas na imagem, como regiões uniformes ou com muitas variações. Certamente que alguns desses

coeficientes necessitariam de um número de níveis de reconstituição maior, enquanto outros precisariam de um número de níveis bem menor. Uma solução para melhor alocar os bits, usando uma classificação dos coeficientes, com a finalidade de aumentar o desempenho dos métodos de codificação, foi apresentado por Chen-Smith [29], cujo algoritmo é apresentado a seguir.

- Dada uma imagem de tamanho *NxN*, calcular a sua transformada usando blocos de tamanho *MxM*.
- Calcule a energia AC para cada bloco $x_{i,j}[m,n]$, $m,n=0, 1, 2,...,M-1$ e $i,j=0, 1, 2,...,(N/M)-1$, dada por:

$$E_{AC}(i,j) = \sum_{m=0}^{M-1}\sum_{n=0}^{M-1} x_{i,j}[m,n] - x_{i,j}^2[0,0] \qquad (4.29)$$

- Classificar os blocos em ordem crescente de energia AC, e dividir o número total de blocos em Q classes de modo que cada classe tenha o mesmo número de blocos. Guardar a informação da classe a que pertence cada bloco.
- Calcular a variância de cada coeficiente transformado $\sigma_q^2(m,n)$, $m,n=0, 1, 2,..., M-1$ e $q=0, 1, 2,...Q-1$, para cada classe.
- De posse das variâncias $\sigma_q^2(m,n)$, estimar o número de bits $b_q(m,n)$ para cada classe, gerando dessa forma Q tabelas de alocação de bits.

Esse algoritmo, apesar de gerar uma informação adicional maior, resulta em um desempenho muito maior em relação ao primeiro caso, sem a classificação dos coeficientes. As figuras 4.6, 4.17 e 4.18 apresentam as imagens recuperadas a partir da imagem Lena codificada com taxa de 1 bit/pixel, usando DCT, LOT e DSTr, com blocos de tamanho 8x8, usando o algoritmo de classificação de Chen-Smith para alocação de bits,descrito anteriormente. Foram usadas 4 classes de energia, na classificação dos blocos. Observa-se que houve um ganho em torno de 5 dB na relação Sinal-Ruído (SNR), para cada método de codificação. Houve ainda, uma melhora subjetiva na qualidade das imagens reconstruídas.

Figura 4.16. Imagem codificada por DCT e 1 bpp. SNR=31,11 e PSNR=36,77.

Figura 4.17. Imagem codificada por LOT e 1 bpp. SNR=31,03 e PSNR=36,9

Figura 4.18. Imagem codificada por DSTr e 1 bpp. SNR=31,00 e PSNR=36,56

4.5. Efeito de Bloqueamento em Imagens Codificadas e Reconstruídas

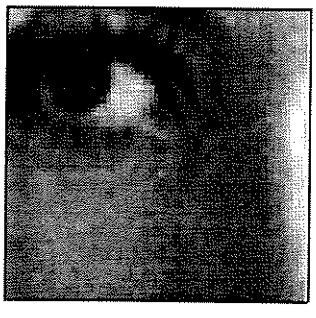
Quando uma imagem é codificada por transformadas, o fato de os blocos serem transformados e quantizados independentemente, principalmente quando se usa baixas taxas, os efeitos do ruído devido a codificação, tendem a aparecer nas bordas dos blocos quando a imagem é reconstituída, uma vez que não há uma continuidade dos valores das intensidades dos pixels da imagem nas fronteiras entre dois blocos. O aparecimento dessas descontinuidades entre os blocos resulta em uma distorção denominada de efeito de bloqueamento. Devido a essa distorção aparecem, na imagem reconstituída, a formação de

vários blocos, que se tornam mais aparentes à medida que decresce a taxa de codificação. Isso é muito comum na DCT. Para reduzir o efeito de bloqueamento, vários trabalhos foram desenvolvidos, envolvendo pré e pós-filtragem [5] ou usando sobreposição de blocos [4]. Uma das técnicas mais conhecidas da literatura para redução do efeito de bloqueamento aplicadas na codificação de imagens é a LOT ("Lapped Orthogonal Transform") [5, 12, 13]. A LOT foi inicialmente desenvolvida com a finalidade de se reduzir o efeito de bloco [5]. Como foi visto no capítulo 2, a LOT utiliza o procedimento de sobreposição de blocos para minimizar o efeito de bloqueamento.

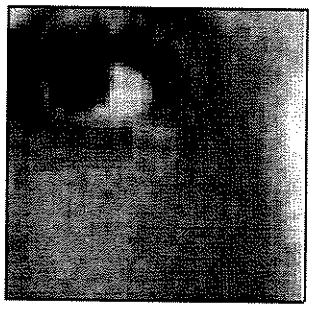
A transformada seno discreta com rotação de eixos, quando aplicada para codificação de imagens tem a propriedade de apresentar um reduzido efeito de bloqueamento. A figura 4.17a apresenta um detalhe da imagem Lena original e as figuras 4.17b, 4.17c e 4.17d apresentam os mesmos detalhes para imagens reconstituídas usando DCT, LOT e DSTr, respectivamente. A taxa usada é de 1 bpp, o tamanho do bloco é de tamanho 8x8 e a quantização é de Lloid-Max com distribuição laplaceana. Como pode ser visto na figura, somente a DCT apresenta efeito de bloco visível. No capitulo 5 serão apresentadas com mais detalhes os resultados referentes aos efeito de bloqueamento.

a) Detalhe da imagem original.

b) Detalhe da imagem codificada por DCT.



c) Detalhe da imagem codificada por LOT.



d) Detalhe da imagem codificada por DSTr.

Figura 4.19. Detalhes da imagem original e codificada.

CAPÍTULO 5

Codificação de Imagens Usando DCT, LOT e DSTr : Resultados Obtidos

Neste capítulo serão apresentados os resultados obtidos por simulação dos métodos de codificação usando-se a transformada cosseno discreta (DCT), a LOT (lapped orthogonal transform) e a transformada seno discreta com rotação de eixos (DSTr). Os programas para simulação foram todos implementados no MATLAB. Os métodos de codificação foram aplicados em várias imagens conhecidas da literatura, tais como a Lena (512x512), Peppers (512x512), Pahka (640x560), Maine (432x640), além de outras imagens, que foram introduzidas. As imagens foram codificadas em várias taxas, usando-se blocos de tamanho 8x8 e 16x16. O tipo de quantizador usado é o de Lloid-Max, com distribuição uniforme, laplaciano e gaussiano. Não foi usada, em nenhum caso a codificação de entropia. Para cada imagem, foram aplicados os três métodos de codificação e os resultados foram então comparados, tomando como parâmetro, a relação Sinal-Ruído (SNR) e a relação Sinal-Ruído de pico (PSNR). Outro ponto de referência usado para comparação foi o de aparecimento de efeitos de bloqueamento visíveis nas imagens reconstituídas, uma vez que a DCT, apesar de ser a transformada mais usada na codificação de imagens, pelo seu desempenho em relação a outras transformadas, apresenta efeito de bloqueamento visível quando uma imagem é codificada em baixas taxas. A LOT tem uma característica marcante que é a redução do efeito de bloqueamento visível nas imagens reconstituídas em baixas taxas. A codificação de imagens usando-se a transformada seno discreta com rotação de eixos (DSTr), que está sendo proposto neste trabalho, tem também a propriedade de reduzir efeito de bloqueamento visível na imagem reconstituída. Dessa forma alguns detalhes da imagem reconstituída serão ampliados, para que se possa perceber o efeito de bloqueamento, obtendo-se então, uma comparação subjetiva entre os três métodos de codificação de imagens no que se refere ao efeito de bloqueamento visível.

5.1. Aplicação dos métodos de codificação a imagem Lena

Nesta primeira aplicação, com o objetivo de comparar o método de codificação usando-se DSTr com os outros dois métodos de codificação usando-se DCT e LOT, a imagem Lena (512x512) foi codificada pelo três métodos para taxas que variam de 1 a 8 bits por pixel, usando-se blocos de tamanho 8x8. Em cada método, foi usada foi usada a regra da log-variância para a alocação de bits, usando-se uma única classe de energia, ou seja uma única tabela de alocação de bits, para cada imagem transformada, conforme foi apresentado capítulo 4. Com base nas tabelas de alocação os coeficientes transformados, para cada método de codificação, foram quantizados usando-se quantizadores de Lloid-Max com distribuição uniforme, laplaciana e gaussiana. Em seguida as imagens forma reconstituída a partir dos coeficientes transformados e quantizados, usando-se os três quantizadores diferentes. A relação Sinal-Ruído (SNR) e a relação Sinal-Ruído de pico (PSNR), foi então calculada para cada caso. A **Tabela 5.1** apresenta a relação Sinal-Ruído (SNR) e a relação Sinal-Ruído de pico (PSNR) entre a imagem Lena original e a reconstituída para taxas que variam de 1 a 8 bit/pixel, usando-se DCT, LOT e DSTr, com blocos de tamanho 8x8. As figuras 5.1, 5.2 e 5.3 apresentam os gráficos para a relação Sinal-Ruído (SNR) em função da taxa de codificação, para os três métodos, DCT, LOT e DSTr, considerando-se, em cada caso, que a distribuição dos coeficientes transformados é uniforme, laplaciana e gaussiana, respectivamente. Pode-se observar nos gráficos dessas figuras que os três métodos de codificação apresentam desempenho bem próximos para o caso de quantização uniforme. Para quantização laplaciana e gaussiana a DSTr apresentou melhor desempenho para taxas mais elevadas, acima de 3 bit/pixel.

A seguir serão mostrados a imagem original as imagens reconstituídas para cada método de codificação, para se fazer uma avaliação subjetiva dos métodos. Serão mostrados ainda, alguns detalhes ampliados das imagens reconstituídas, para que se possa fazer uma avaliação subjetiva com relação ao efeito de bloqueamento, que geralmente aparece quando uma imagem é codificada em baixas taxas (menor ou igual a 1 bit/pixel), visto que o efeito de bloqueamento, muitas vezes, não se torna aparente quando a imagem é vista por inteiro.

Tabela 5.1- Relação sinal ruído (SNR) e Relação de Sinal-Ruído de pico (PSNR) para a imagem Lena codificada com taxas que variam de 1 a 8 bpp usando DCT, LOT e DSTr, com blocos de tamanho 8x8.

		TIPO DE QUANTIZAÇÃO					
		UNIFORME		LAPLACIANA		GAUSSIANA	
TAXA	TRANSFORMADA	SNR	PSNR	SNR	PSNR	SNR	PSNR
1 bpp	DCT	25,85	31,52	26,68	32,54	25,55	31,20
	LOT	26,20	31,85	27,27	32,93	25,97	31,62
	DSTr	25,53	31,17	26,58	32,23	25,43	31,09
2 bpp	DCT	28,99	34,65	30,75	36,41	28,27	33,63
	LOT	29,05	34,71	30,81	36,46	28,54	34,19
	DSTr	28,21	33,87	30,63	36,28	28,63	34,28
3 bpp	DCT	32,26	37,92	34,37	40,02	30,97	36,63
	LOT	32,08	37,74	34,03	39,69	30,96	36,62
	DSTr	30,82	36,47	34,49	40,15	30,81	36,47
4 bpp	DCT	36,27	41,93	37,19	42,85	33,64	39,30
	LOT	35,76	41,41	36,45	42,10	33,63	39,30
	DSTr	34,99	40,65	38,29	43,94	34,00	39,66
5 bpp	DCT	41,47	47,13	38,83	44,48	36,51	42,16
	LOT	40,27	45,93	37,81	43,47	36,00	41,66
	DSTr	40,55	46,21	40,64	46,29	37,22	42,87
6 bpp	DCT	46,79	52,44	39,44	45,10	39,66	45,31
	LOT	45,42	51,08	38,72	44,38	38,47	44,13
	DSTr	46,43	52,08	41,63	47,27	41,14	46,80
7 bpp	DCT	49,95	55,61	39,59	45,25	42,08	47,73
	LOT	49,31	54,97	39,31	44,97	40,44	46,10
	DSTr	50,67	55,72	41,80	47,46	43,96	49,62
8 bpp	DCT	50,69	56,35	39,59	45,25	42,65	48,31
	LOT	50,38	56,04	39,52	45,18	41,23	46,88
	DSTr	51,01	56,66	41,81	47,47	44,73	50,39

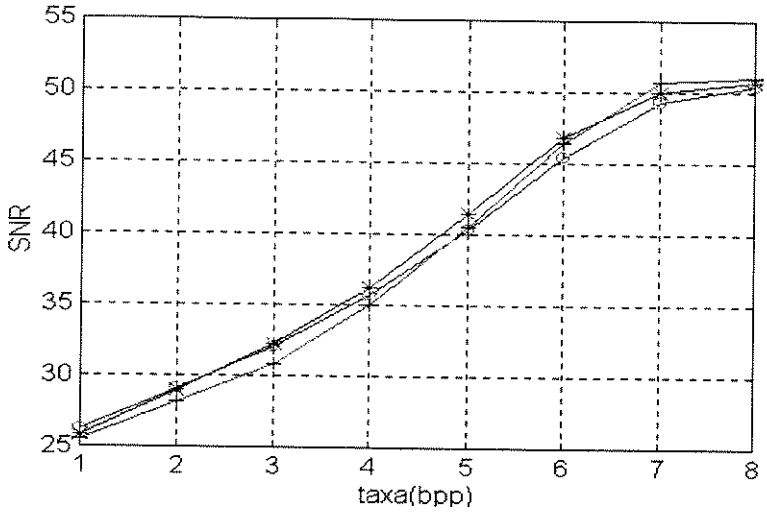


Figura 5.1.Relação Sinal-Rruído (SNR) x Taxa para quantização uniforme: DCT(*), LOT(o) e DSTr.

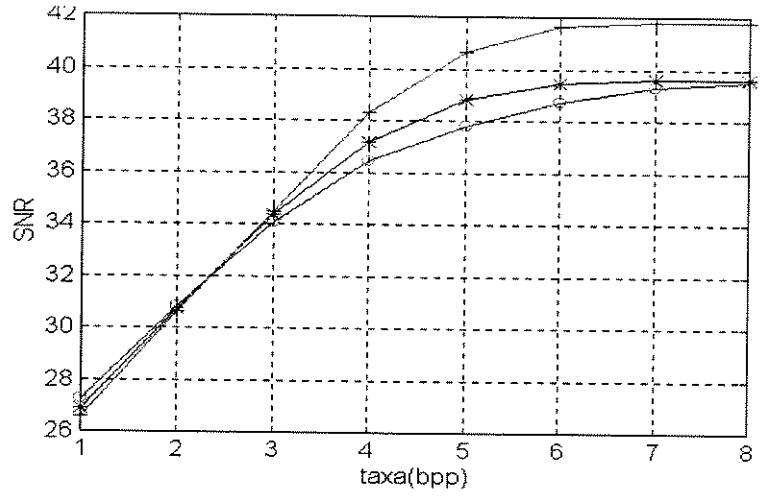


Figura 5.2. Relação Sinal-Ruído (SNR) x Taxa para quantização laplacina: DCT(*), LOT(o) e DSTr(+).

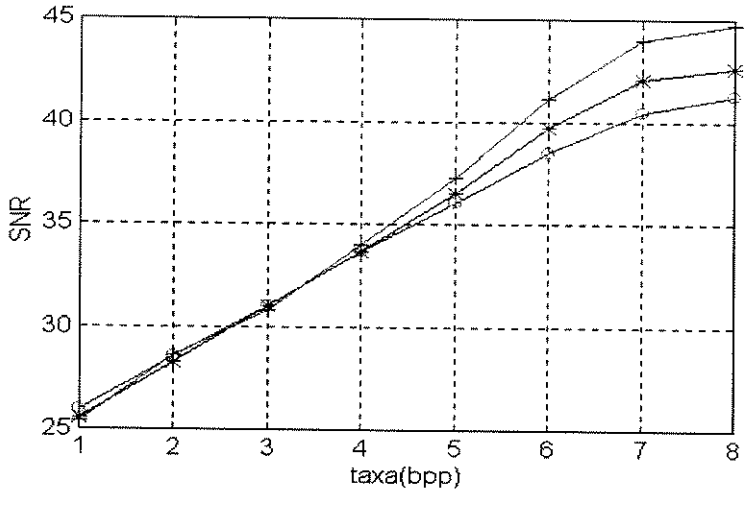


Figura 5.3. Relação Sinal-Truído (SNR) x taxa para quantização gaussiana: DCT(*), LOT(o) e DSTr(+).

A figura 5.4 mostra a imagem Lena original e as imagens codificadas, com taxa de 1 bpp, usando-se DCT, LOT e DSTr, quando os coeficientes são quantizados usando-se quantização uniforme. Subjetivamente não se nota diferenças entre as imagens reconstituídas. Isso poderá ser visto posteriormente quando serão mostrados alguns detalhes dessas imagens.

a) Imagem Lena original.

b) Imagem codificada com DCT.

c) Imagem codificada com LOT.

d) Imagem codificada com DSTr.

Figura. 5.4. Imagens original e codificadas usando DCT, LOT e DSTr com quantização uniforme e blocos (8x8).

A figura 5.5 mostra a imagem Lena original e as imagens codificadas, com taxa de 1 bpp, usando-se DCT, LOT e DSTr, quando os coeficientes são quantizados usando-se quantizador Lloid-Max com distribuição laplaciana. Nota-se ainda que não há diferenças marcantes entre as imagens reconstituídas.

a) Imagem Lena Original.

b) Imagem codificada com DCT.

c) Imagem codificada com LOT.

d) Imagem codificada com DSTr.

Figura. 5.5. Imagens original e codificadas usando DCT, LOT e DSTr com quantização laplaciana e blocos (8x8).

A figura 5.6 mostra a imagem Lena original e as imagens codificadas, com taxa de 1 bpp, usando-se DCT, LOT e DSTr, quando os coeficientes são quantizados usando-se quantizador Lloid-Max com distribuição gaussiana. Pode-se notar uma distorção no ombro da Lena, com a forma de degraus de uma escada, para as imagens codificadas com DCT e LOT, aparecendo com maior intensidade na imagem codificada com DCT.

a) Imagem Lena Original.

b) Imagem codificada com DCT.

c) Imagem codificada com LOT.

d) Imagem codificada com DSTr.

Figura. 5.6. Imagens original e codificadas usando DCT, LOT e DSTr com quantização gaussiana e blocos (8x8).

Serão mostrados agora alguns detalhes da imagem Lena, com o objetivo de se comparar subjetivamente os efeitos de bloqueamento nas imagens codificadas com os três métodos de codificação usando DCT, LOT e DSTr. A figura 5.7 mostra três detalhes da imagem Lena original, dois da face próximo ao olho e outro do ombro, que serão tomados como referência para comparação subjetiva com relação ao aparecimento de efeitos de bloqueamento nas imagens reconstituídas.

Figura 5.7. Detalhes originais da imagem Lena.

A figura 5.8 mostra os detalhes das imagens reconstituídas quando a imagem Lena é codificada com DCT, LOT e DSTr, usando quantização uniforme e taxa de 1 bpp. Nota-se um intenso efeito de bloqueamento em toda a extensão do detalhe da imagem codificada por DCT. É pouco intenso no detalhe da imagem codificada com LOT e pouco visível no detalhe da imagem codificada por DSTr. A figura 5.9 mostra os detalhes quando se usa o quantizador de Lloid-Max com distribuição laplaciana. Nota-se ainda o aparecimento do efeito de bloqueamento, próximo do olho, no detalhe da imagem codificada por DCT. Esse efeito não é visível nos detalhes das imagens codificadas por LOT e DSTr.

A Figura 5.10 mostra os detalhes das imagens codificadas por DCT, LOT e DSTr, quando se usa quantização de Lloid-Max com distribuição gaussiana. Pode-se observar novalmente o aparecimento de efeito de bloqueamento visível no detalhe da imagem codificada por DCT. Não aparecem efeitos de bloqueamento nos detalhes das imagens codificadas por DSTr e LOT. As figuras 5.11, 5.12 e 5.13 mostram outros detalhes (região do ombro) das imagens codificadas, quando se usa quantização uniforme, laplaciana e gaussiana, respectivamente.

a) DCT b) LOT c) DSTr

Figura 5.8. Detalhes das imagens codificadas com DCT, LOT e DSTr, para quantização uniforme.

a) DCT b) LOT c) DSTr

Figura 5.9. Detalhes das imagens codificadas com DCT, LOT e DSTr, para quantização laplaciana.

a) DCT b) LOT c) DSTr

Figura 5.10. Detalhes das imagens codificadas com DCT, LOT e DSTr, para quantização gaussiana.

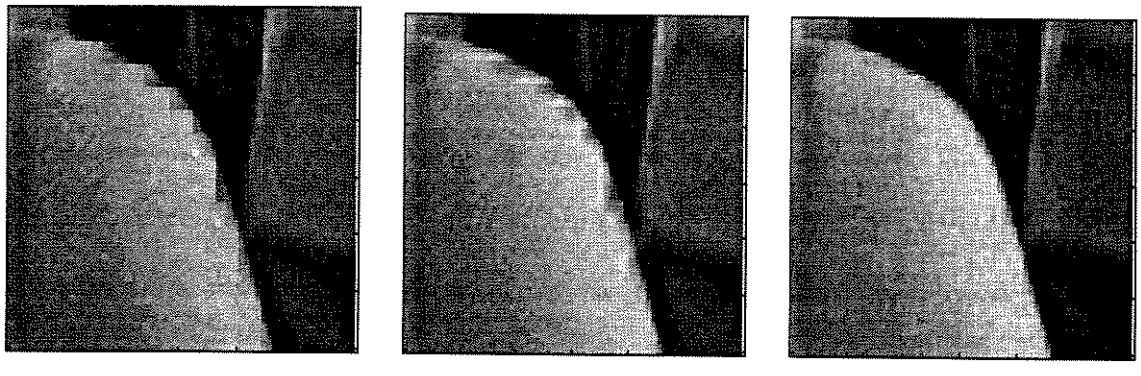
Pode-se observar na figura 5.11 que o detalhe da imagem codificada por DCT, com quantização uniforme apresenta um efeito de bloco mais intenso que a LOT e DSTr. Nas figuras 5.12 e 5.13, que mostram os detalhes das imagens codificadas quando se usa quantizadores laplaciano e gaussiano, o efeito de bloqueamento aparece na forma de escada nas imagens codificadas por DCT e LOT.

a) DCT b) LOT c) DSTr

Figura 5.11. Detalhes das imagens codificadas com DCT, LOT e DSTr, para quantização uniforme.

a) DCT b) LOT c) DSTr

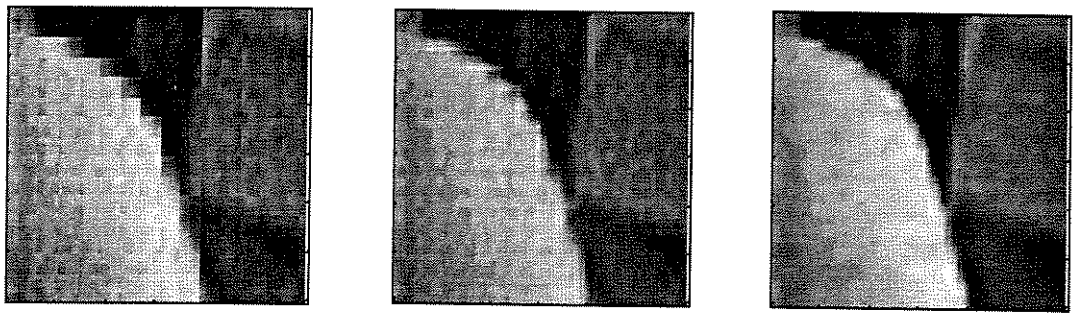
Figura 5.12. Detalhes das imagens codificadas com DCT, LOT e DSTr, para quantização laplaciana.



a) DCT b) LOT c) DSTr

Figura 5.13. Detalhes das imagens codificadas com DCT, LOT e DSTr, para quantização gaussiana.

Para enfatizar ainda o efeito de bloqueamento a imagem Lena(512x512) foi codificada com taxas de 0,2 bpp, usando-se DCT, LOT e DSTr, com blocos de 8x8 e quantizador laplaciano, e usando-se uma única tabela para a alocação de bits. As relações Sinal-Ruído obtidas para DCT, LOT e DSTr foram: 20,70, 21,35 e 21,32 dB, respectivamente. Neste caso as relações Sinal-Ruído estão muito próximas entre si. A figura 5.14 apresenta os detalhes das imagens reconstituídas. Observa-se que a distorção na forma de escada torna-se muito mais forte na imagem codificada por DCT, e menos intenso na imagem codificada por DSTr.



a) b) c)

Figura 5.14. Detalhes da imagem Lena codificada com 0,2 bpp com a) DCT, b) LOT e DSTr.

A imagem Lena foi codificada com taxas de 0,2 bpp, usando alocação adaptativa de bits, com 4 classes de energia, com o restante das condições idênticas aos do procedimento anterior. As relações Sinal-Ruído obtidas para esse caso foram: DCT 23,70, LOT 24,54 e DSTr 22,95dB. A figura 5.15 mostra os detalhes para as imagens codificadas com DCT, LOT e DSTr. Os detalhes mostram que as imagens já estão bastante deterioradas pelo ruído, e o efeito de bloco é mais acentuado na imagem codificada por DCT e menos visível para a DSTr, ainda que a DSTr apresente a menor relação Sinal-Ruído.

A imagem Lena foi ainda codificada com taxas de 0,5, 1, 2, 3 e 4 bpp, usando-se os três métodos de codificação, DCT, LOT e DSTr, blocos de tamanho 8x8, com quantização laplaciana e alocação de bits adaptativa com 4 classes de energia. A Tabela 5.2 mostra a relação Sinal-Ruído (SNR) e relação Sinal-Ruído de pico (PSNR) para cada taxa. Observa-se que as relações Sinal-Ruído estão bem próximas para taxas de 1 e 2 bpp. Para taxas maiores que 2 bpp, a DSTr apresenta a maior relação Sinal-Ruído. A figura 5.16 mostra os detalhes para a imagem Lena codificada com taxa de 0,5 bpp. Observa-se que a DCT continua apresentando um intenso efeito de bloqueamento visível.

Figura 5.15. Detalhes das imagens codificadas com DCT, LOT e DSTr, com taxa de 0,2 bpp, quantizador gaussiano e alocação de bits adaptativa com 4 classes de energia.

Tabela 5.2. SNR e PSNR para a imagem Lena codificada com DCT, LOT e DSTr, e blocos de 8x8, com alocação adaptativa com 4 classes usando quantização de Lloi-Max laplaciana.

	0,5 bpp		1 bpp		2 bpp		3 bpp		4 bpp	
	SNR	PSNR	SNR	PSNR	SNR	PSNR	SNR	PSNR	SNR	PSNR
DCT	28,13	33,79	31,10	36,77	34,77	40,42	37,28	42,93	39,01	44,66
LOT	28,21	33,87	31,03	36,67	34,31	39,97	36,94	42,60	38,81	44,46
DSTr	27,44	33,10	30,96	36,62	34,35	40,01	37,45	43,11	40,06	45,72

DCT LOT DSTr

Figura 5.16. Detalhes das imagens codificadas com DCT, LOT e DSTr, com taxa de 0,5 bpp.

A figura 5.17 mostra o gráfico da relação sinal-ruído (SNR) em função da taxa de codificação em bit/pixel, para os três métodos de codificação DCT, LOT e DSTr, com as taxas variando de 1 a 8 bpp, blocos de tamanho 16x16 e quantização laplaciana de Lloid-Max com uma única tabela para alocação de bits. Observa-se no gráfico que a DCT apresenta um melhor desempenho para taxas menores ou iguais a 3 bpp. Para taxas iguais ou maiores que 4

bpp a DSTr apresenta um melhor desempenho. O quantizador laplaciano de Lloid-Max tem apresentado um melhor desempenho, para taxas até 4 ou 5 bpp, de modo que as imagens reconstituídas tem sido subjetivamente melhores, com menor efeito de bloqueamento e maior relação sinal-ruído. Por esse motivo tem-se usado com mais freqüência os quantizadores de Lloid-Max com distribuição de Laplace.

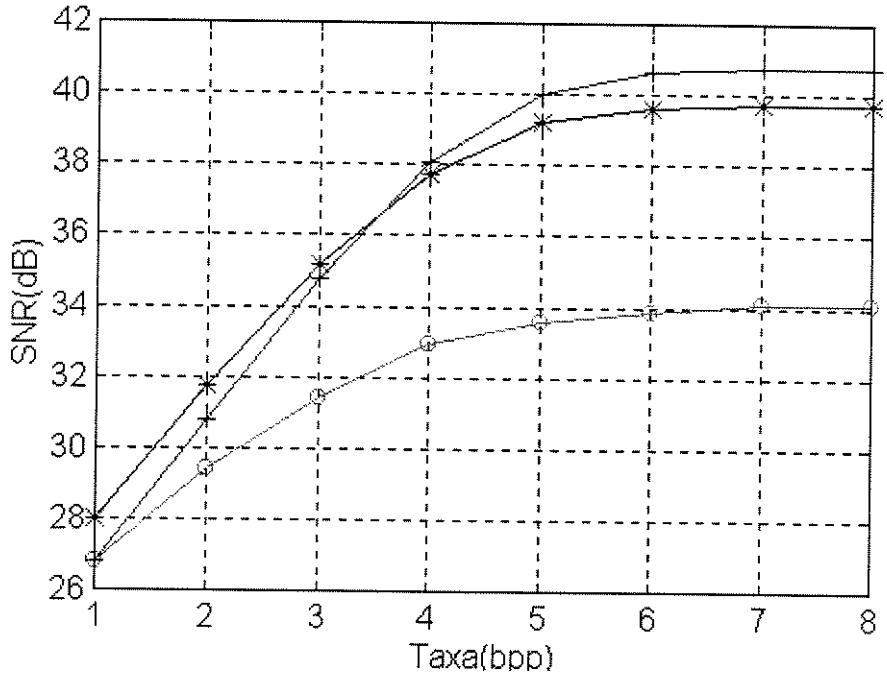


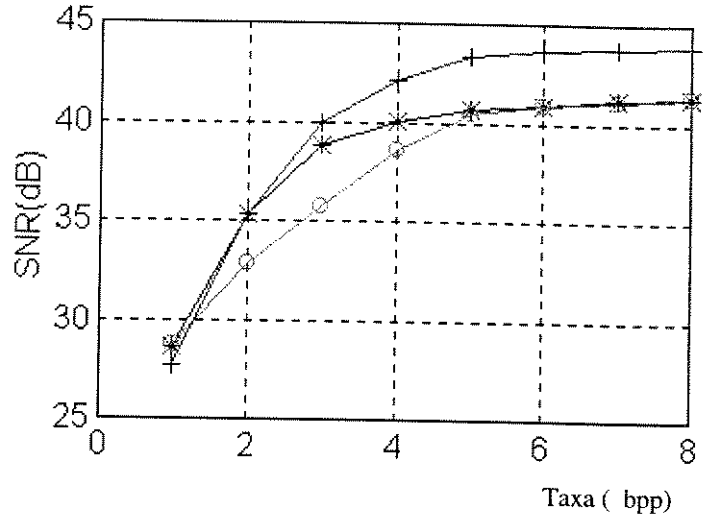
Figura 5.17. Desempenho em termos da SNR para Lena codificada com DCT(*), LOT(o) e DSTr(+) e com taxa de 1 bpp, blocos de tamanho 16x16 e quantizador de Lloid-Max laplaciano.

5.2. Aplicação dos Métodos de Codificação a Outras Imagens

Outras imagens foram codificadas usando-se, DCT, LOT e DSTr. Algumas dessas imagens, com os seus respectivos desempenhos, serão apresentadas a seguir.

A figura 5.18a mostra a imagem "Pahka" original de tamanho 640x560. Essa imagem foi codificada, com taxas de 1 a 8 bpp, usando-se DCT, LOT e DSTr com blocos de tamanho 8x8 e quantizador laplaciano de Lloid-Max, usando-se uma única tabela de alocação de bits. A figura 5.18b mostra o gráfico para a relação Sinal-Ruído em função da taxa de codificação. Nota-se pelo gráfico dessa figura que as imagens codificadas com DSTr e DCT apresentam relações sinal-ruído comparáveis para taxas até 2 bpp, no entanto para taxas maiores que 2 bpp, a DSTr apresentou maior relação Sinal-Ruído.

a) Imagem Pahka original



b) SNR x Taxa para DCT(*), LOT(o) e DSTr(+).

Figura 5.18. Imagem Pahka original e SNRxTaxa para a imagem codificada com DCT, LOT e DSTr.

Outra imagem denominada "Maine" de tamanho 432x640, mostrada na figura 5.19, com os respectivos detalhes, foi codificada, com taxa de 1 bpp usando-se DCT, LOT e DSTr, blocos de tamanhos 8x8 e 16x16 para quantização uniforme, laplaciana e gaussiana de Lloid-Max, usando-se uma única tabela alocação de bits. As relações Sinal-Ruído (SNR) e Sinal-Ruído de pico (PSNR) obtidas para as imagens reconstituídas são mostradas na Tabela 5.3 para blocos de tamanho 8x8 e na Tabela 5.4 para blocos de 16x16. As tabelas mostram que a DSTr apresentou, na maioria dos casos, a menor relação sinal-ruído comparada com a DCT e LOT. No entanto, observando-se os detalhes das imagens reconstituídas verifica-se que a DSTr apresentou subjetivamente o melhor desempenho. A figura 5.20 mostram as imagens codificadas usando-se DCT, LOT e DSTr para quantização uniforme. São mostrados ainda dois detalhes (detalhe I e detalhe II), da imagem reconstituída para cada método de codificação. Observando-se o detalhe I da figura 5.20 pode-se notar o aparecimento de efeitos de bloqueamento para a DCT e LOT, sendo mais aparente para a DCT. A DSTr não apresenta efeito de bloqueamento visível, nesse caso. As figuras 5.21 e 5.22 mostram as imagens codificadas usando-se DCT, LOT e DSTr para quantização laplaciana e gaussiana. Pode-se observar no detalhe II o aparecimento de uma distorção, na forma de uma sombra, nas imagens codificadas com DCT e LOT, quando o quantizador é laplaciano ou gaussiano. Essa distorção não aparece na imagem codificada com DSTr em nenhum dos dois casos. Isso mostra que o procedimento de rotação de eixos também contribui para redução de

descontinuidades, pois naquela região da imagem existe uma grande variação das intensidades dos pixels.

Tabela 5.3. Relação Sinal-Ruído (SNR) e Sinal-Ruído de pico (PSNR) para a imagem Maine codificada com DCT, LOT e DSTr, com blocos de tamanho 8x8.

TRANSFORMADA	TIPO DE QUANTIZAÇÃO					
	UNIFORME		LAPLACIANA		GAUSSIANA	
	SNR	PSNR	SNR	PSNR	SNR	PSNR (dB)
DCT	25,57	30,70	26,53	31,66	25,45	30,58
LOT	25,69	30,82	26,83	31,96	25,86	30,99
DSTr	25,18	30,31	26,25	31,38	24,45	29,58

Tabela 5.4. Relação Sinal-Rído (SNR) e Sinal-Ruído de pico (PSNR) para a imagem Maine codificadas com DCT, LOT e DSTr, com blocos de tamanho 16x16.

TRANSFORMADA	TIPO DE QUANTIZAÇÃO					
	UNIFORME		LAPLACIANA		GAUSSIANA	
	SNR	PSNR	SNR	PSNR	SNR	PSNR (dB)
DCT	27,07	32,20	27,98	33,11	26,03	31,17
LOT	26,65	31,78	26,94	32,07	25,57	30,70
DSTr	25,96	31,09	27,59	32,72	26,02	31,15

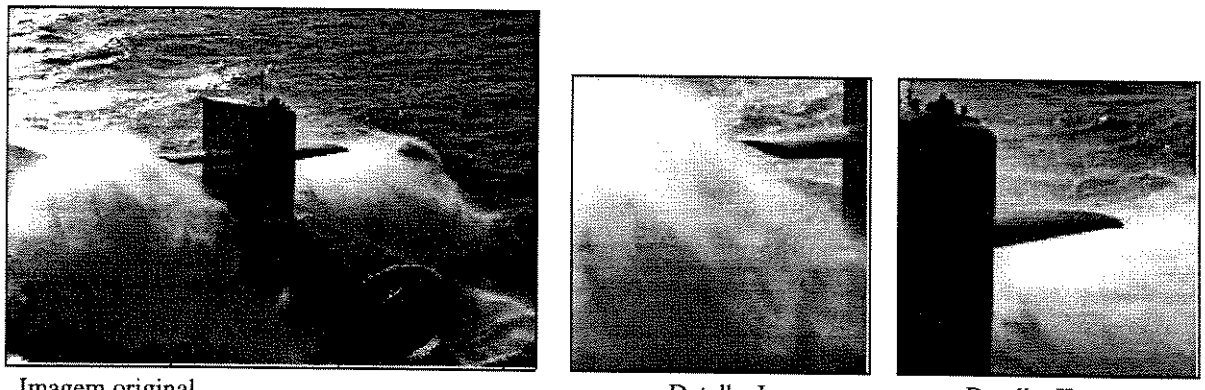
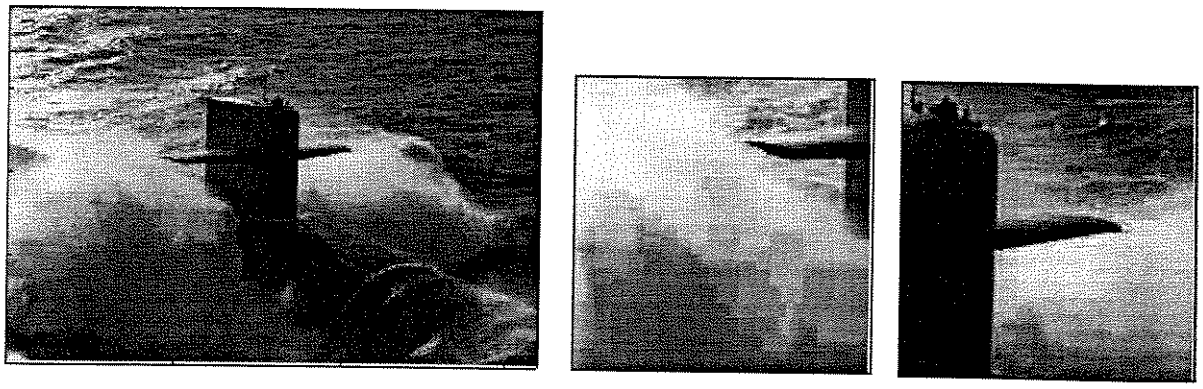
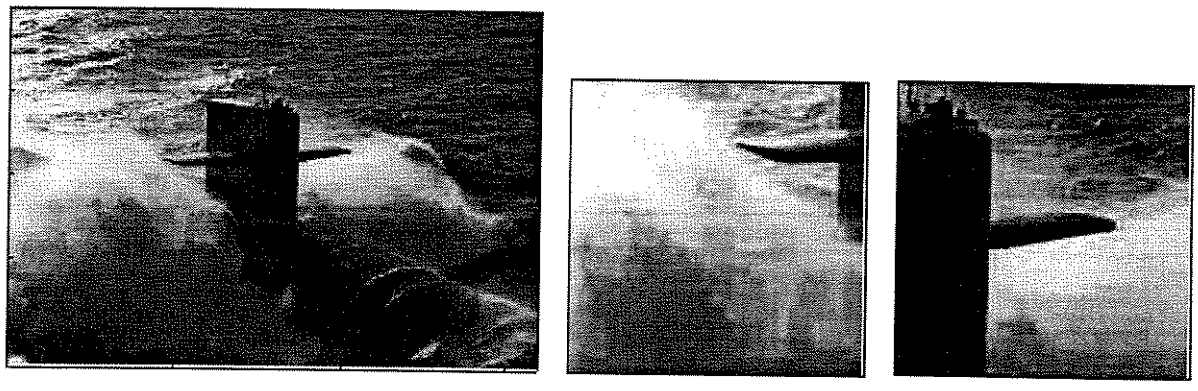


Imagem original Detalhe I Detalhe II

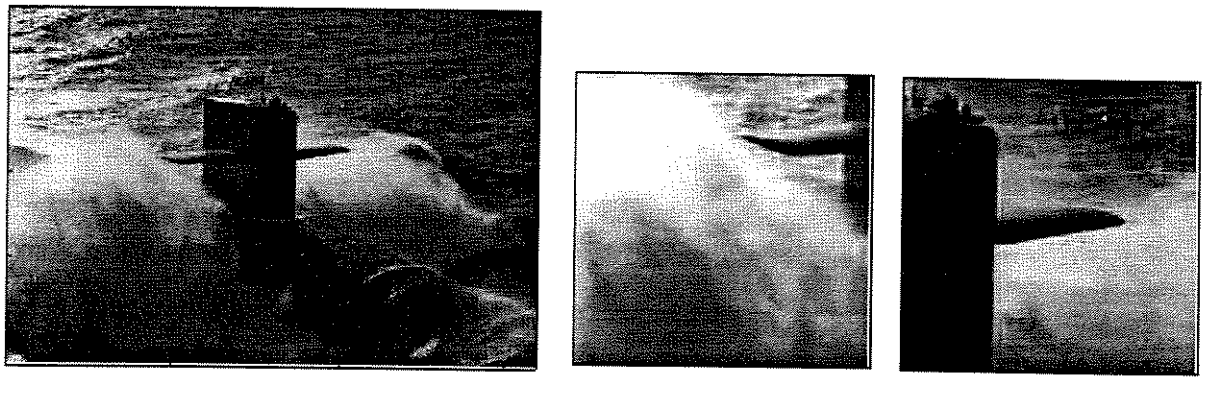
Figura 5.19. Imagem Maine original de tamanho 432x640 e detalhes da imagem original.



a) Imagem codificada com DCT. Detalhe I Detalhe II

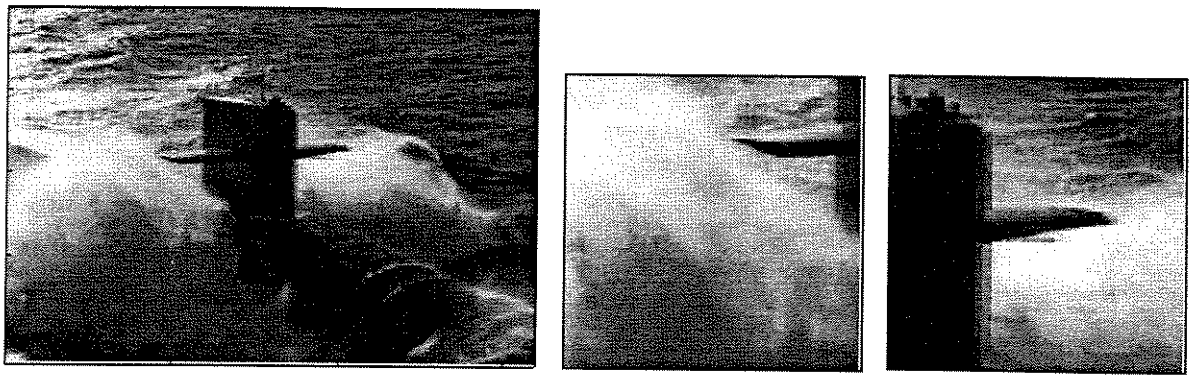


b) Imagem codificada com LOT. Detalhe I Detalhe II

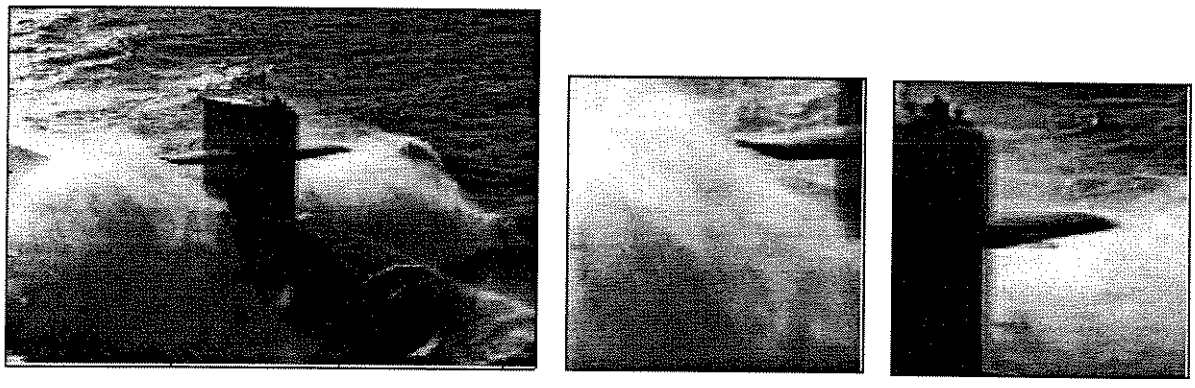


c) Imagem codificada com DSTr Detalhe I Detalhe II

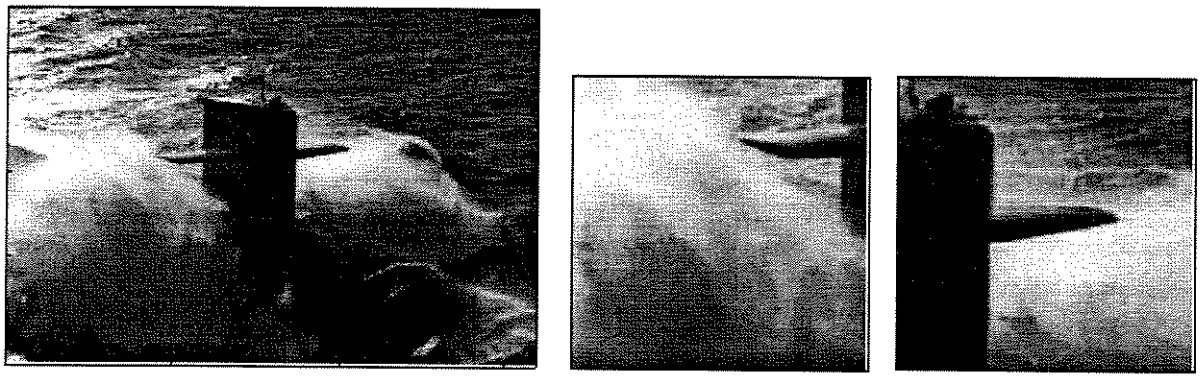
Figura 5.20. Imagem codificada com 1 bpp e quantização uniforme usando-se a) DCT, b) LOT e c) DSTr.



a) Imagem codificada com DCT Detalhe I Detalhe II

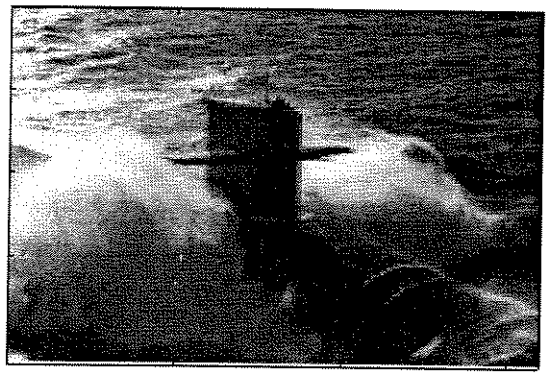


b) Imagem codificada com LOT Detalhe I Detalhe II

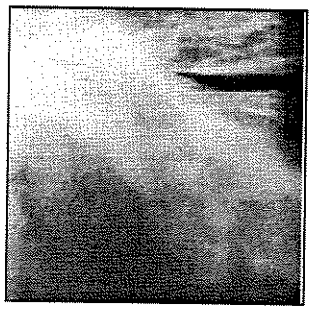


c) Imagem codificada com DSTr Detalhe I Detalhe II

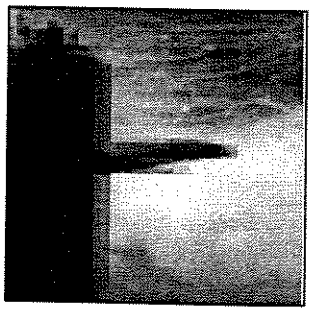
Figura 5.21. Imagem codificada com 1 bpp e quantização laplaciana usando-se a) DCT, b)LOT e c) DSTr.



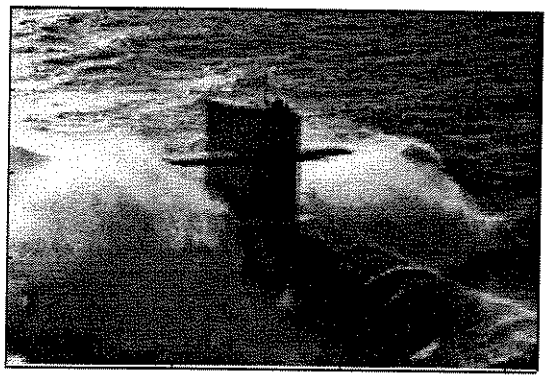
a) Imagem codificada com DCT



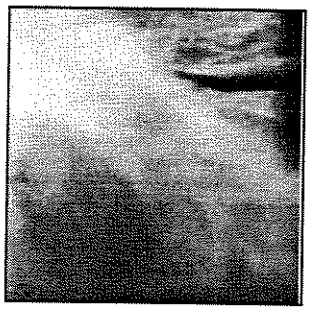
Detalhe I



Detalhe II

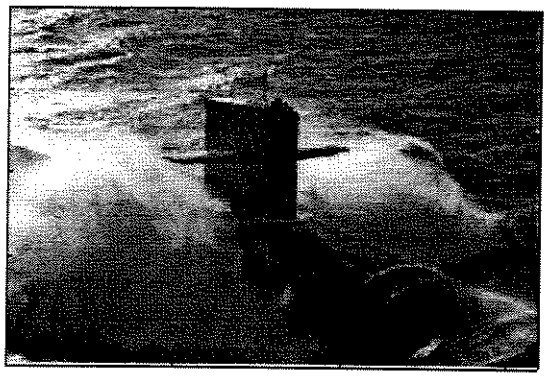


b) Imagem codificada com LOT



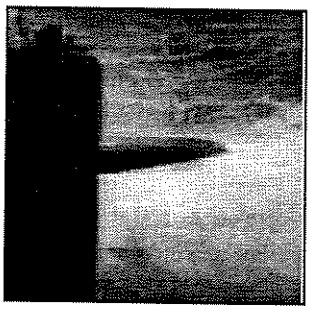
Detalhe I

Detalhe II



a) Imagem codificada com DSTr

Detalhe I



Detalhe II

Figura 5.22. Imagem codificada com 1 bpp e quantização gaussiana usando-se a) DCT, b) LOT e c) DSTr.

Outra imagem usada para comparar o desempenho dos métodos de codificação foi a imagem denominada "fachada", mostrada na figura 5.23 de tamanho 512x512. Essa imagem foi codificada usando-se DCT, LOT e DSTr, com blocos de tamanho 8x8, quantização laplaciana de Lloid-Max, com alocação de bits adaptativa com quatro classes para taxas de codificação de 1, 2, 3 e 4 bpp. Essas taxas foram usadas para se comparar o desempenho dos métodos para taxas de codificação mais altas. A relação Sinal-Ruído (SNR) e a relação Sinal-Ruído de pico (PSNR) obtida entre a imagem original e a imagem reconstituída, para cada método, é mostrada na Tabela 5.5. Para essa imagem, a DSTr apresentou um melhor desempenho em relação a DCT e LOT, para taxas iguais a 2, 3 e 4 bpp. Para a taxa igual a 1 bpp o desempenho apresentado pela DSTr é praticamente igual a DCT.

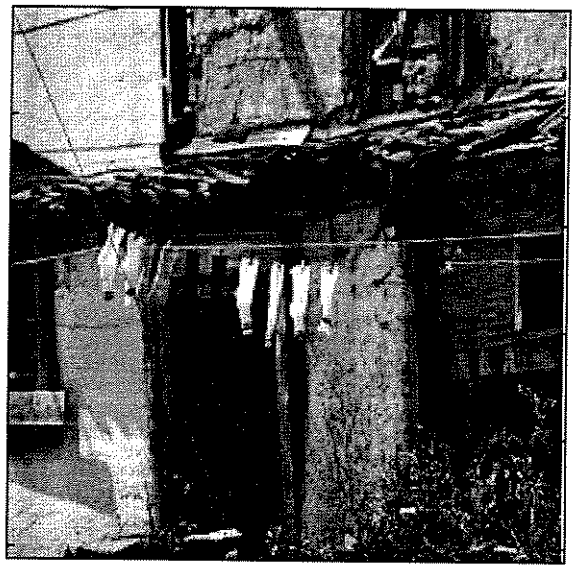


Figura 5.23. Imagem original "fachada" (512x512).

Tabela 5.5. SNR e PSNR para Imagem "fachada" codificada com DCT, LOT e DSTr, blocos de tamanho 8x8 e quantização laplaciana.

	1 bpp		2bpp		3 bpp		4 bpp	
	SNR	PSNR	SNR	PSNR	SNR	PSNR	SNR	PSNR
DCT	27,79	33,39	32,30	37,90	34,29	39,89	35,13	40,73
LOT	27,18	32,78	31,23	36,83	33.35	38,95	34,53	40,12
DSTr	27,71	32,31	33,24	38,84	35,97	41,57	37,24	42,84

Outra imagem denominada de "peppers", mostrada na figura 5.24, foi codificada com taxas de 0,5, 1 e 2 bpp, usando DCT, LOT e DSTr, com blocos de tamanho 8x8, quantização laplaciana de Lloid-Max, com 4 classes de energia. As relações Sinal-Ruído (SNR) e a relação Sinal-Ruído de pico (PSNR) obtida para cada método é mostrado na Tabela 5.6. Para essa imagem, a DSTr apresentou os menores valores para SNR e PSNR, exceto para a taxa de 0,5 bpp.

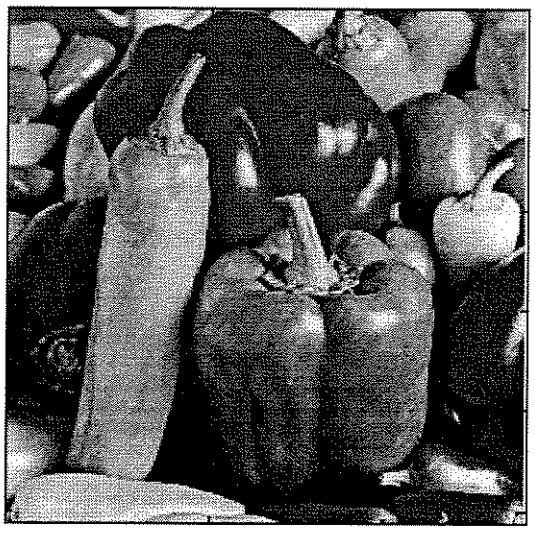


Figura 5.24. Imagem original "peppers" (512x512).

Tabela 5.6. SNR e PSNR para Imagem "peppers" codificada com DCT, LOT e DSTr, blocos de tamanho 8x8 e quantização laplaciana.

	0,5 bpp		1 bpp		2 bpp	
	SNR	PSNR	SNR	PSNR	SNR	PSNR
DCT	26,69	32,76	29,53	35,53	33,42	39,12
LOT	25,84	32,54	28,93	34,63	33.05	38,75
DSTr	26,39	32,54	28,77	34,47	32,14	37,84

5.3. Codificação de Imagens Usando Decimação e Interpolação por DSTr.

A DSTr apresenta uma característica de reduzir o efeito de bloqueamento nas imagens reconstituídas, como foi visto nas seções anteriores, além de apresentar um desempenho

comparável ao da DCT e da LOT. No entanto, a DSTr foi desenvolvida inicialmente com a finalidade de se fazer interpolação de sinais unidimensionais[22]. Posteriormente, foi feita uma extensão bidimensional da DSTr para se fazer interpolação de imagens[23], onde se mostrou razoavelmente eficiente como foi visto no capítulo 2. Nesta seção serão apresentados os resultados da codificação de imagens por decimação e interpolação integrando as duas técnicas que utilizam a DSTr: interpolação e codificação. O objetivo de se usar a DSTr é melhorar a qualidade subjetiva da imagem reconstituída, que deverá ter um reduzido efeito de bloqueamento quando codificada em baixas taxas de codificação, além de se ter um método que utiliza o mesmo algoritmo para codificação e interpolação de imagens.

O método consiste em se fazer inicialmente a decimação da imagem original, por uma taxa de 4:1, isto é, para cada quatro amostras somente uma será mantida. Dessa forma, uma imagem de tamanho MxN será decimada para se obter uma versão da imagem de tamanho (M/2)x(N/2). A decimação é feita para essa taxa porque as imagens que se dispõem, em geral, não são amostradas a taxas muito elevadas, de modo que muitos detalhes seriam perdidos se a imagem original fosse decimada a taxas maiores, por exemplo 8:1. A imagem decimada é então codificada com uma taxa igual a quatro vezes a taxa de codificação desejada, de modo que a quantidade de dados armazenados ou transmitidos são equivalentes. Essa imagem codificada é então reconstituída e interpolada, usando DSTr, para se obter a imagem com o tamanho original com MxN pixels. A figura 5.25 mostra o diagrama de blocos do método de interpolação e codificação proposto. O procedimento de decimação é realizado passando-se inicialmente a imagem original por um filtro FIR ("anti-aliasing") de ordem 40, seguindo-se a redução da taxa de amostragem, como descrito no capítulo 2. Essa combinação de codificação com interpolação deve-se ao fato da DSTr apresentar um reduzido efeito de bloqueamento quando é usada tanto na codificação quanto na interpolação de imagens.

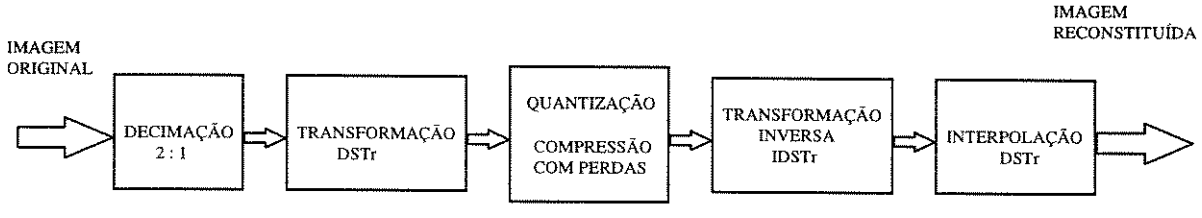


Figura 5.25. Diagrama de blocos de blocos de um sistema usando codificação e interpolação.

Para ilustrar o método de interpolação por decimação e interpolação usando-se a DSTr, a imagem Lena, de tamanho 512x512, foi utilizada como imagem de teste. A imagem original foi inicialmente decimada, obtendo-se uma versão de tamanho 256x256. Essa imagem decimada foi codificada com uma taxa de 2 bpp usando-se a DSTr, com blocos de tamanho 8x8 e quantização laplaciana de Lloid-Max, com alocação de bits adaptativa com 4 classes de energia. A imagem reconstruída foi interpolada por DSTr para se obter a imagem com o tamanho original 512x512. A taxa de codificação é portanto equivalente a 0,5 bpp. Para se comparar o desempenho entre os métodos, a imagem original foi ainda codificada com uma taxa de 0,5 bpp usando-se somente a DSTr, com blocos de tamanho 8x8 e quantização laplaciana com alocação de bits adaptativa com quatro classes de energia. A relação Sinal-Ruído (SNR) e a relação Sinal-Ruído de pico (PSNR) entre a imagem original e a reconstruída foram então calculadas para cada método de codificação e estão mostradas na Tabela 5.7. O método de codificação por decimação e interpolação com DSTr apresentou, para essa imagem, uma melhoria de 0,86 dB em relação ao método que utiliza somente a DSTr. A figura 5.26 mostra as imagens original e reconstituídas pelos dois métodos de codificação, para que se possa fazer uma avaliação subjetiva. Nota-se que as imagens reconstituídas não apresentam diferenças marcantes. No entanto, pelos detalhes mostrados na figura 5.27, pode-se observar que a imagem codificada por decimação e interpolação apresenta uma melhor definição, mais precisamente na região na qual aparece uma parte do chapéu da Lena. Pode-se ainda notar que os dois métodos não apresentam efeitos de bloqueamento visíveis, para essa taxa de codificação (0,5 bpp).

Tabela 5.7. SNR e PSNR entre a imagem Lena (512x512) original e codificada por decimação e interpolação (DSTr/DSTr) e codificada com DSTr para taxa de 0,5 bpp e quantização laplaciana (Lloid-Max).

	DSTr	DSTr/DSTr
SNR (dB)	27,08	27,92
PSNR (dB)	32,73	33,58

A imagem Lena decimada (256x256) foi codificada com uma taxa de 1 bpp, usando-se DSTr, blocos de tamanho 8x8 e quantização de Lloid-Max laplaciana, com alocação de bits adaptativa com 4 classes. Em seguida foi interpolada para se obter a imagem reconstruída de tamanho (512x512) com taxa de codificação equivalente de 0,25 bpp. Para se comparar os

desempenhos, a imagem Lena (512x512) foi também codificada, nas mesmas condições anteriores, usando somente DSTr para uma taxa de 0,25 bpp. A relação Sinal-Ruído (SNR) e a relação Sinal-Ruído de pico (PSNR) para cada método são apresentadas na Tabela 5.8, onde se pode notar uma melhora de 0,41 dB na relação Sinal-Ruído, quando se compara os dois métodos. A figura 5.28 apresenta as imagens codificadas por decimação e interpolação com DSTr e somente por DSTr e a figura 5.29 mostra detalhes das imagens codificadas. Observa-se que as imagens reconstruídas já estão bastante corroídas pelo ruído devido a baixa taxa de codificação, mesmo assim apresentam um reduzido efeito de bloqueamento visível.

a) Lena original (512x512).

b) Imagem codificada e interpolada (DSTr/DSTr) ;

c) Imagem codificada com DSTr;

Figura 5.26. Imagens: a) original; b) codificada por decimação e interpolação com taxa equivalente a 0,5 bpp e c) imagem codificada com DSTr com taxa de 0,5 bpp.

a) Detalhe da imagem original;

b) detalhe da imagem codificada e interpolada (DSTr/DSTr).

c) Detalhe da imagem codificada com DSTr

Figura 5.27. Detalhes das imagens a) original; a) codificadas por decimação e interpolação com DSTr e c) codificada com DSTr a 0,5 bpp, quantização laplaciana com 4 classes e blocos de 8x8.

Tabela 5.8. SNR e PSNR entre a imagem Lena (512x512) original e codificada por decimação e interpolação com DSTr e codificada com DSTr para taxa de 0,25 bpp e quantização laplaciana.

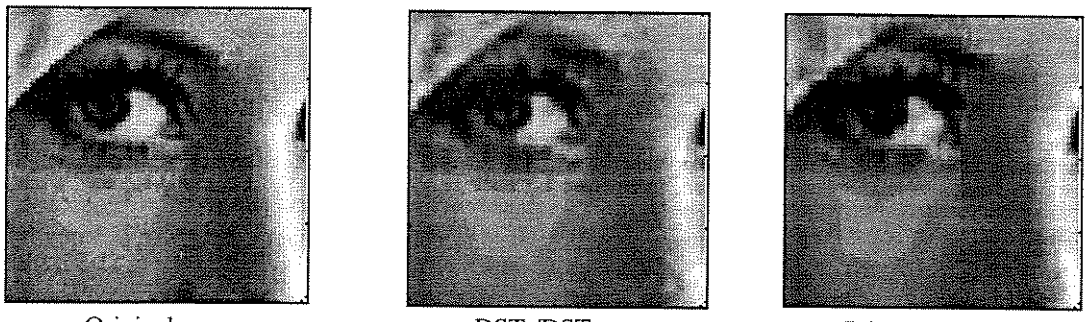
	DSTr	DSTr/DSTr
SNR (dB)	24,45	24,86
PSNR (dB)	30,10	30,51

a) Imagem Lena original (512x512)

b) Imagem codificada por decimação e interpolação (DSTr/DSTr)

c) imagem codificada somente com DSTr.

Figura 5.28. Imagens: a) original; b) codificada por decimação e interpolação com taxa equivalente a 0,25 bpp e c) imagem codificada com DSTr com taxa de 0,25 bpp.



Original DSTr/DSTr DSTr

Figura 5.29. Detalhes das imagens originais, codificada por decimação interpolação (DSTr/DSTr) e codificadas com DSTr, para taxa de 0,25 bpp, quantização laplaciana e blocos de tamanho 8x8.

CAPÍTULO 6

Conclusões

A codificação de imagens por transformadas é um procedimento de representação de uma imagem como um arranjo de bits para armazená-la ou transmiti-la através de um canal, com o objetivo principal de minimizar o número de bits necessários para representar a imagem com um certo critério de fidelidade, preservando os níveis de qualidade e inteligibilidade necessários para uma dada aplicação. Esses procedimentos empregam transformadas cuja finalidade é mapear a imagem para um outro domínio de modo que essa transformação minimize ou até elimine as redundâncias existentes no domínio espacial. A transformada mais usada na codificação de imagens é a transformada cosseno discreta (DCT), uma vez que ela é usada nos padrões JPEG e MPEG. A transformação é comumente feita por pequenos blocos (4x4, 8x8, 16x16). Assim, transformadas com a DCT, apresentam um problema inerente que é o efeito de bloqueamento quando usadas em baixas taxas. Outras transformadas têm sido usadas na codificação de imagens tais como a lapped orthogonal transform (LOT), transformada wavelet, etc. A LOT é uma transformada que apresenta uma característica especial: a redução do efeito de bloqueamento.

Na codificação de imagens, as transformadas são usadas em conjunto com as técnicas de quantização e de alocação de bits, e esse procedimento se constitui na principal ferramenta para obtenção de imagens com baixas taxas e com um certo grau de qualidade. Para se representar os níveis de quantização, podem também ser usados os algoritmos de codificação de entropia que têm por objetivo reduzir ainda mais as redundâncias que permanecem após a quantização dos coeficientes transformados. Neste trabalho não foi usada a codificação de entropia uma vez que o objetivo principal foi o de comparar métodos de codificação por transformadas, principalmente quanto ao aparecimento de efeitos de bloqueamento visíveis nas imagens reconstituídas, quando são codificadas em baixas taxas, usando condições idênticas para todos os esquemas de codificação utilizados.

Este trabalho apresentou dois enfoques diferentes: o primeiro discorreu sobre as técnicas de interpolação de imagens por transformadas e o segundo, sobre a codificação de imagens por transformadas. Em ambos os casos foi usada a transformada seno discreta com rotação de

eixos bidimensional (DSTr-2D), que é uma das contribuições deste trabalho. Os resultados foram comparados com técnicas de interpolação e codificação conhecidas na literatura [18, 19].

As contribuições apresentadas neste trabalho se concentram em duas áreas. A primeira contribuição é a transformada seno discreta com rotação de eixos bidimensional (DSTr-2D), que é uma extensão da DSTr-1D, apresentada em [7], acrescida de algumas particularidades para se tornar apropriada às aplicações com sinais em duas dimensões. A contribuição na área de interpolação de imagens apresenta-se na forma de dois algoritmos de interpolação de imagens, sendo que o primeiro utiliza a DSTr-2D e o segundo utiliza a DCT/FFT com descarte de amostras para se eliminar o efeito de blocos nas imagens interpoladas. Finalmente a DSTr-2D é usada na codificação de imagens, onde apresentou um desempenho comparável a DCT e LOT, além de apresentar uma característica especial de redução do efeito de bloqueamento nas imagens reconstituídas.

Para interpolação de imagens, foram apresentados dois novos métodos de interpolação, sendo que o primeiro utiliza a transformada seno discreta com rotação de eixos (DSTr-2D) e o segundo é implementado usando-se a FFT e a definição da DCT (DCTd), com eliminação de extrapolações. Para se comparar o desempenho dos métodos de interpolação, foram usados dois outros métodos de interpolação, sendo o primeiro uma extensão do método de Wang [18,19], usando a DCT-I; o segundo método faz a interpolação usando a DCT/FFT. Em todos os métodos, a interpolação é feita por blocos de tamanho MxM usando-se as transformadas DCT, FFT e DST que são separáveis, de modo que os métodos podem ser implementados aplicando-se a transformada sobre as linhas e colunas de cada bloco. No caso da interpolação de imagens usando-se DSTr-2D, o método utiliza blocos de tamanho (M+1)x(M+1) para realizar o procedimento de pré e pós-rotação de eixos. No final, a última linha e a última coluna de cada bloco são descartadas para se obter um bloco interpolado de tamanho MxM. O método DCTd utiliza também blocos de tamanho (M+1)x(M+1), descartando-se a última linha e a última coluna de cada bloco após a interpolação. Os métodos de interpolação, em todos os casos, foram implementados usando-se blocos de tamanho 8x8, 16x16, 32x32, etc.

Para se comparar o desempenho dos métodos de interpolação, foi inicialmente criada uma imagem a partir de uma equação. Essa imagem foi amostrada em várias taxas, de modo

que, para cada imagem interpolada, tivesse sempre uma imagem original para que se pudesse verificar o desempenho calculando-se a SNR e PSNR. A interpolação por DSTr apresentou a maior relação Sinal-Ruído entre os quatro métodos, seguido da DSTd, DCT/FFT, e DCT-I, em todas as interpolações realizadas usando-se essa imagem. Deve-se destacar que a DSTr apresentou o melhor desempenho quando se aumentou a taxa de amostragem da imagem, pois a relação Sinal-Ruído (SNR) se distancia dos outros métodos à medida que cresce a taxa de amostragem. Essa diferença chegou a 15 dB, em relação ao método DCT-I, quando uma imagem de tamanho 128x128 foi interpolada para se obter uma versão da imagem de tamanho 255x255, ou ainda uma diferença de 20 dB, quando o tamanho da imagem original é de tamanho 256x256 e a imagem interpolada é de tamanho 511x511. Isso significa que a DSTr tem um ótimo desempenho para imagens de alta definição.

Quando a imagem de teste de tamanho 64x64 foi interpolada para se obter uma versão da imagem de tamanho 505x505 (16 amostras interpoladas para cada amostra original), a diferença da SNR chegou a 13 dB em relação ao pior caso (DCT-I).

Outra imagem usada como teste foi a Lena (512x512), que foi decimada para se obter uma imagem de tamanho 256x256. Deve-se resaltar que a imagem original foi passada por um filtro FIR ("anti-aliasing") de ordem 30 antes da decimação. A imagem decimada foi interpolada pelos quatro métodos de interpolação para se obter uma versão da imagem original, usando-se blocos de tamanhos 4x4, 8x8, 16x16, 32x 32 e 64x64. A DSTr apresentou a maior relação Sinal-Ruído em todos os casos, seguido de valores próximos pela DCTd. A maior diferença apresentada foi entre os métodos DSTr e DCT-I, cujos valores da relação Sinal-Ruído variaram entre 2,25 e 2,75 dB. Essas diferenças não são tão grandes quanto no primeiro caso, porque a taxa de amostragem da imagem original não é tão elevada.

Os quatro métodos de interpolação foram ainda aplicados em uma sub-imagem de tamanho 128x128 obtida da imagem Lena (512x512). Essa imagem foi interpolada para se obter uma imagem de tamanho 1017x1017. Uma avaliação subjetiva da imagem foi feita observando-se um detalhe escolhido de cada imagem obtida pelos quatro métodos de interpolação, DSTr, DSTd, DCT/FFT e DCT-I. Pode-se observar a presença de efeito de bloqueamento visível no detalhe selecionado da imagem interpolada por DCT/FFT. Esse fato dá crédito para que a transformada seno discreta com rotação de eixos bidimensional (DSTr-2D) possa ser aplicada na codificação de imagens por transformadas, com a finalidade de

reduzir efeitos de bloqueamento visíveis nas imagens reconstituídas, pois a DSTr-2D havia sido desenvolvida inicialmente para se fazer somente a interpolação de imagens.

Na codificação de imagens por transformadas, procura-se representar uma imagem com o menor número possível de bits, preservando um determinado nível de qualidade da imagem reconstituída. Em primeiro lugar, a imagem é transformada usando-se uma transformada conhecida da literatura, em seguida os coeficientes são quantizados e representados usando-se um código de comprimento fixo ou variável. Devido a grande quantidade de dados que é manipulada quando se trabalha com processamento de imagens, é mais eficiente aplicar a transformada por blocos. Esse fato gera um problema que é inerente à maioria das transformadas, que consiste no efeito de bloqueamento visível. Esse efeito é gerado devido ao aparecimento de descontinuidades entre os blocos vizinhos da imagem reconstituída, decorrente do processamento independente dos blocos. O efeito de blocos é uma distorção que se torna mais aparente na imagem reconstituída à medida que se reduz a taxa de codificação da imagem. A DCT, que é a transformada mais usada na codificação de imagens, apresenta um intenso efeito de bloqueamento visível quando é usada em baixas taxas de codificação. A LOT é uma transformada que reduz o efeito de bloqueamento. A DSTr apresentou um reduzido efeito de bloqueamento quando foi usada na codificação de imagens por transformadas em baixas taxas. Além disso, apresentou um desempenho comparável aos da DCT e da LOT.

Neste trabalho, foi implementado o método de codificação de imagem usando-se DSTr-2D, que foi aplicado em várias imagens, usando-se diferentes taxas de codificação. Para se fazer uma comparação relativa, outros dois métodos foram implementados, a DCT e a LOT. A DCT foi escolhida por ser a transformada mais usada em processamento de imagens e por apresentar um melhor desempenho em relação a outras transformadas e a LOT por ser uma transformada que tem uma característica especial que é a redução do efeito de bloqueamento.

Os métodos de codificação foram implementados usando-se quantização uniforme, laplaciana e gaussiana de Lloid-Max. A alocação de bits foi feita de duas maneiras: a primeira considerou apenas uma tabela de alocação de bits determinada pela regra da log-variância para a imagem inteira; a segunda foi feita seguindo o algoritmo de Chen-Smith, que agrupa os blocos transformados em Q classes de acordo com a energia AC dos coeficientes de cada bloco. Para cada classe é determinada uma tabela de alocação de bits correspondendo a uma

dada taxa de codificação, de modo que a taxa média seja igual à taxa de codificação desejada. Nas simulações em que se usou esse algoritmo foram usadas quatro classes de energia.

As simulações foram feitas usando as imagens Lena (512x512), Maine (640x432), Pahka (640x560), Peppers (512x412), Fachada (512x512). Quando a imagem Lena foi codificada com taxa de 1 bpp, usando-se DCT, LOT e DSTr, com uma única tabela de alocação de bits, a DCT apresentou um intenso efeito de bloqueamento, que foi mais acentuado para a quantização uniforme. A LOT e a DCT apresentaram um efeito de blocos na forma de uma escada na região do ombro da Lena para a quantização laplaciana e gaussiana. A DSTr não apresentou esses efeitos de blocos em nenhum dos casos. Quanto ao desempenho com relação a SNR e PSNR, os três métodos apresentaram valores aproximados para quantização uniforme para taxas que variaram de 1 a 8 bpp. Quando se usou quantização laplaciana e gaussiana, a DSTr apresentou um melhor desempenho comparado ao da DCT e LOT, chegando-se a uma diferença de 3 dB. Para taxas menores que 3 bpp, os valores obtidos de SNR e PSNR foram bem próximos para os três métodos. Em alguns casos, a DSTr apresentou a SNR maior que a da LOT e em outros casos, a situação se inverteu, isto é os valores de SNR foram maiores para a LOT. Outras simulações foram feitas para taxas menores que 1 bpp, confirmando-se sempre a característica da DSTr de reduzir o efeito de bloqueamento. Vários detalhes das imagens reconstruídas foram mostrados para que se fizesse uma avaliação subjetiva do aparecimento de efeitos de bloqueamento.

Outra imagem usada na simulação foi a imagem denominada de Maine (432x640), que possui, como característica especial, uma descontinuidade muito grande em uma dada região da imagem. Nesse caso, tanto a DCT como a LOT apresentaram uma sombra na região de descontinuidade da imagem reconstruída, o que não ocorreu com a DSTr. Para outras como Pahka e Fachada (512x512) a DSTr apresentou uma relação Sinal-Ruído superior a DCT e a LOT para taxas maiores ou iguais a 2 bpp.

Foi ainda implementado um método de codificação por decimação e interpolação com DSTr, o qual integra o bom desempenho da DSTr obtido para interpolação com a característica de redução do efeito de bloqueamento nas imagens reconstituídas quando a codificação é feita em baixas taxas. O método foi aplicado na imagem Lena (512x512) para taxas de 0,25 e 0,5 bpp, pois ele é mais apropriado para aplicações em baixas taxas, uma vez que ele é limitado pelo desempenho obtido para interpolação da imagem. Os resultados da

simulação mostraram uma melhora de 0,41 dB para taxa de 0,25 bpp e de 0,86 bpp para a taxa de 0,5 bpp, além da imagem reconstruída apresentar maior definição e reduzido efeito de bloqueamento.

Enfim, a transformada seno discreta com rotação de eixos (DSTr) é muito simples de ser implementada, além de não exigir muito esforço computacional adicional, em relação a DCT, uma vez que se tem uma pré-rotação que é implementada por subtrações e uma pós-rotação que é implementada por adições, seguida da aplicação da transformada seno discreta sobre a seqüência sem os pontos extremos. No caso da DSTr-2D, a transformada seno discreta bidimensional é aplicada sobre um sub-bloco de tamanho (M-1)x(M-1), e a transformada seno discreta unidimensional é aplicada sobre a primeira linha e primeira coluna de um bloco de tamanho MxM, tendo dessa forma uma redução do número de operações realizadas. Esse fato compensa em parte o número de operações necessárias para se realizar os procedimentos de pré e pós-rotação de eixos. Portanto, um simples procedimento realizado antes e após a aplicação da transformada seno discreta produz uma redução no efeito de blocos que em alguns casos torna-se um problema bastante complexo quando se deseja eliminá-lo ou minimizá-lo.

Sugestões para outros trabalhos:

A DSTr tem uma vasta aplicação em processamento digital de imagens podendo inicialmente ser aplicada em qualquer situação onde se use a DCT, LOT ou qualquer outra transformada implementada em blocos, observando-se suas particularidades, como a presença de uma amostra original em cada bloco transformado e a aplicação da transformada seno discreta unidimensional e bidimensional. Portanto podem ser desenvolvidos trabalhos tais como:

- Codificação de imagens usando DSTr com quantização Vetorial.
- O padrão JPEG usando a transformada seno discreta com rotação de eixos. Esse trabalho já se encontra em andamento na UFPa.
- O padrão MPEG com a transformada seno discreta com rotação eixos.
- Outros trabalhos usando a interpolação de imagens com DSTr.

Assim, foram listadas as principais contribuições e resultados do presente trabalho que espera-se possam auxiliar no desenvolvimento de outras pesquisas relacionadas com o processamento digital de sinais e de imagens.

APÊNDICE A

São apresentadas a seguir as Tabelas contendo os níveis de decisão e os níveis de reconstituição para os quantizadores de Lloid-Max, que foram apresentados na seção 4.3. Na Tabela, r_i representa o nível de reconstituição e d_i representa o nível de decisão para os vários quantizadores com taxas de 1 a 8 bit/amostra, quando se considera que a distribuição do conjunto de amostras de entrada é gaussiana ou laplaciana, com média zero e variância unitária. Para cada quantizador são apresentados somente os valores positivos para que a tabela não se torne muito longa. A parte negativa é igual a parte positiva, exceto que os valores dos níveis de decisão *di* e de reconstituição *ri* são antecedidos pelo sinal negativo.

Na tabela, os níveis de decisão d_i significam que, se o valor da amostra de entrada estiver entre d_k e d_{k+1}, será reconstituída pelo valor r_k. Certamente, os valores do conjunto de amostras de entrada a serem quantizados devem ser normalizados.

O quantizador uniforme não é apresentado por não haver necessidade de se consultar tabelas para se fazer a quantização.

TABELA PARA QUANTIZADORES DE LLOID-MAX

TIPO DE QUANTIZAÇÃO

	GAUSSIANA		**LAPLACIANA**	
No. de Bits	**d_i**	**r_i**	**d_i**	**r_i**
1	0.0000	0,7979	0,0000	0,7071
	∞			∞
2	0.0000	0,4529	0,0000	0,4180
	0.9818	1.5107	1.1202	1.8223
	∞		∞	
3	0.0000	0.2454	0.0000	0.2330
	0.5011	0.7569	0.5320	0.8310
	1.0510	1.3451	1.2492	1.6674
	1.7491	2.1531	2.3699	3.0724
	∞		∞	
4	0.0000	0.1288	0.0000	0.1248
	0.2590	0.3891	0.2663	0.4077
	0.5238	0.6585	0.5707	0.7337
	0.8015	0.9446	0.9259	1.1181
	1.1017	1.2588	1.3523	1.5866
	1.4398	1.6208	1.8867	2.1869
	1.8462	2.0717	2.6060	3.0252
	2.4033	2.7350	3.7286	4.4319
	∞		∞	

TIPO DE QUANTIZAÇÃO (continuação)

	GAUSSIANA		LAPLACIANA	
No. de Bits	d_i	r_i	d_i	r_i
5	0.0000	0.0663	0.0000	0.0645
	0.1328	0.1992	0.1332	0.2019
	0.2663	0.3333	0.2753	0.3488
	0.4013	0.4693	0.4276	0.5064
	0.5388	0.6082	0.5915	0.6766
	0.6797	0.7511	0.7690	0.8614
	0.8250	0.8990	0.9625	1.0636
	0.9763	1.0537	1.1752	1.2867
	1.1354	1.2171	1.4113	1.5358
	1.3045	1.3918	1.6767	1.8176
	1.4869	1.5819	1.9799	2.1421
	1.6874	1.7929	2.3335	2.5249
	1.9136	2.0343	2.7583	2.9918
	2.1787	2.3231	3.2912	3.5906
	2.5096	2.6962	4.0093	4.4279
	2.9807	3.2652	5.1312	5.8345
	∞		∞	
6	0.0000	0.0396	0.0000	0.0389
	0.0793	0.1189	0.0793	0.1197
	0.1586	0.1983	0.1616	0.2035
	0.2381	0.2778	0.2469	0.2904
	0.3177	0.3575	0.3354	0.3805
	0.3975	0.4374	0.4272	0.4739
	0.4775	0.5176	0.5224	0.5709
	0.5579	0.5982	0.6212	0.6715
	0.6388	0.6793	0.7237	0.7759
	0.7201	0.7609	0.8302	0.8844
	0.8020	0.8431	0.9408	0.9972
	0.8846	0.9261	1.0559	1.1146
	0.9681	1.0100	1.1757	1.2368
	1.0525	1.0950	1.3006	1.3644
	1.1381	1.1812	1.4310	1.4977
	1.2251	1.2689	1.5675	1.6373
	1.3136	1.3584	1.7106	1.7839
	1.4041	1.4499	1.8612	1.9384
	1.4969	1.5439	2.0201	2.1017
	1.5923	1.6408	2.1884	2.2751
	1.6910	1.7412	2.3677	2.4603
	1.7936	1.8460	2.5598	2.6593
	1.9010	1.9560	2.7669	2.8746
	2.0142	2.0724	2.9923	3.1100
	2.1347	2.1970	3.2401	3.3702
	2.2646	2.3321	3.5160	3.6619
	2.4066	2.4810	3.8286	3.9953
	2.5649	2.6487	4.1904	4.3856
	2.7462	2.8437	4.6222	4.8587
	2.9623	3.0809	5.1607	5.4626
	3.2372	3.3935	5.8832	6.3037
	3.6376	3.8817	7.0085	7.7132
	∞		∞	

TIPO DE QUANTIZAÇÃO (continuação)

	GAUSSIANA		LAPLACIANA	
No. de Bits	d_i	r_i	d_i	r_i
7	0.0000	0.0376	0.0000	0.0377
	0.0752	0.1128	0.0768	0.1159
	0.1504	0.1880	0.1564	0.1968
	0.2256	0.2632	0.2387	0.2805
	0.3008	0.3384	0.3238	0.3671
	0.3760	0.4135	0.4118	0.4565
	0.4511	0.4887	0.5026	0.5488
	0.5263	0.5639	0.5964	0.6440
	0.6015	0.6391	0.6931	0.7422
	0.6767	0.7143	0.7928	0.8434
	0.7519	0.7895	0.8955	0.9476
	0.8271	0.8647	1.0013	1.0549
	0.9023	0.9399	1.1101	1.1653
	0.9775	1.0151	1.2221	1.2789
	1.0527	1.0903	1.3372	1.3955
	1.1279	1.1655	1.4555	1.5154
	1.2031	1.2407	1.5770	1.6385
	1.2783	1.3159	1.7017	1.7648
	1.3535	1.3911	1.8296	1.8945
	1.4287	1.4664	1.9609	2.0274
	1.5040	1.5416	2.0955	2.1636
	1.5792	1.6168	2.2334	2.3031
	1.6544	1.6920	2.3746	2.4461
	1.7296	1.7672	2.5192	2.5924
	1.8048	1.8425	2.6673	2.7421
	1.8801	1.9177	2.8187	2.8953
	1.9553	1.9929	2.9736	3.0519
	2.0305	2.0682	3.1320	3.2121
	2.1058	2.1434	3.2939	3.3757
	2.1810	2.2187	3.4592	3.5428
	2.2563	2.2939	3.6282	3.7135
	2.3316	2.3692	3.8006	3.8877
	2.4068	2.4445	3.9767	4.0656
	2.4821	2.5198	4.1563	4.2470
	2.5574	2.5951	4.3395	4.4321
	2.6328	2.6704	4.5264	4.6208
	2.7081	2.7458	4.7170	4.8132
	2.7835	2.8212	4.9113	5.0093
	2.8589	2.8966	5.1093	5.2092
	2.9344	2.9721	5.3110	5.4128
	3.0099	3.0476	5.5166	5.6203
	3.0855	3.1233	5.7260	5.8317
	3.1611	3.1990	5.9394	6.0471
	3.2370	3.2749	6.1568	6.2665
	3.3129	3.3509	6.3783	6.4901
	3.3891	3.4272	6.6041	6.7180
	3.4655	3.5038	6.8343	6.9505
	3.5422	3.5807	7.0692	7.1878
	3.6194	3.6581	7.3091	7.4303
	3.6971	3.7361	7.5544	7.6784
	3.7756	3.8150	7.8056	7.9328
	3.8550	3.8949	8.0635	8.1943

No. de Bits	d_i	r_i	d_i	r_i
7	3.9355	3.9762	8.3292	8.4641
	4.0177	4.0593	8.6039	8.7437
	4.1021	4.1448	8.8894	9.0352
	4.1892	4.2336	9.1885	9.3418
	4.2801	4.3266	9.5047	9.6676
	4.3762	4.4257	9.8433	10.0190
	4.4794	4.5331	10.2122	10.4054
	4.5928	4.6525	10.6236	10.8417
	4.7213	4.7902	11.0975	11.3533
	4.8738	4.9575	11.6706	11.9878
	5.0686	5.1797	12.4197	12.8516
	5.3568	5.5339	13.5642	14.2767
	∞		∞	
8	0.0000	0.0347	0.0000	0.0373
	0.0694	0.1041	0.0760	0.1146
	0.1388	0.1735	0.1546	0.1946
	0.2082	0.2429	0.2359	0.2773
	0.2776	0.3123	0.3200	0.3628
	0.3470	0.3817	0.4069	0.4510
	0.4164	0.4511	0.4966	0.5421
	0.4858	0.5205	0.5891	0.6361
	0.5552	0.5899	0.6845	0.7330
	0.6246	0.6593	0.7828	0.8327
	0.6940	0.7287	0.8841	0.9355
	0.7634	0.7981	0.9884	1.0413
	0.8328	0.8675	1.0957	1.1501
	0.9022	0.9369	1.2060	1.2619
	0.9716	1.0063	1.3194	1.3768
	1.0410	1.0757	1.4358	1.4948
	1.1104	1.1451	1.5554	1.6160
	1.1798	1.2145	1.6782	1.7404
	1.2492	1.2839	1.8041	1.8679
	1.3186	1.3533	1.9332	1.9986
	1.3880	1.4227	2.0656	2.1326
	1.4574	1.4921	2.2012	2.2698
	1.5268	1.5615	2.3401	2.4104
	1.5962	1.6309	2.4823	2.5542
	1.6656	1.7004	2.6278	2.7013
	1.7351	1.7698	2.7766	2.8518
	1.8045	1.8392	2.9288	3.0057
	1.8739	1.9086	3.0843	3.1630
	1.9433	1.9780	3.2433	3.3236
	2.0127	2.0474	3.4057	3.4877
	2.0822	2.1169	3.5715	3.6553
	2.1516	2.1863	3.7408	3.8262
	2.2210	2.2557	3.9135	4.0007
	2.2904	2.3252	4.0897	4.1787
	2.3599	2.3946	4.2694	4.3602
	2.4293	2.4640	4.4526	4.5451
	2.4987	2.5335	4.6394	4.7337
	2.5682	2.6029	4.8297	4.9258
	2.6376	2.6723	5.0236	5.1214
	2.7071	2.7418	5.2210	5.3207
	2.7765	2.8112	5.4221	5.5235

No. de Bits	d_i	r_i	d_i	r_i
8	2.8460	2.8807	5.6267	5.7299
	2.9154	2.9501	5.8350	5.9400
	2.9849	3.0196	6.0468	6.1537
	3.0543	3.0891	6.2623	6.3710
	3.1238	3.1585	6.4815	6.5920
	3.1932	3.2280	6.7043	6.8167
	3.2627	3.2974	6.9308	7.0450
	3.3322	3.3669	7.1610	7.2770
	3.4017	3.4364	7.3949	7.5127
	3.4711	3.5059	7.6324	7.7522
	3.5406	3.5753	7.8737	7.9953
	3.6101	3.6448	8.1187	8.2422
	3.6796	3.7143	8.3675	8.4928
	3.7491	3.7838	8.6200	8.7471
	3.8186	3.8533	8.8762	9.0052
	3.8880	3.9228	9.1362	9.2671
	3.9575	3.9923	9.3999	9.5328
	4.0270	4.0618	9.6675	9.8022
	4.0966	4.1313	9.9388	10.0754
	4.1661	4.2008	10.2139	10.3524
	4.2356	4.2703	10.4928	10.6332
	4.3051	4.3399	10.7755	10.9178
	4.3746	4.4094	11.0620	11.2062
	4.4441	4.4789	11.3523	11.4984
	4.5137	4.5484	11.6464	11.7945
	4.5832	4.6180	11.9444	12.0943
	4.6528	4.6875	12.2462	12.3981
	4.7223	4.7571	12.5519	12.7056
	4.7918	4.8266	12.8614	13.0171
	4.8614	4.8962	13.1747	13.3323
	4.9310	4.9657	13.4919	13.6515
	5.0005	5.0353	13.8130	13.9745
	5.0701	5.1049	14.1379	14.3014
	5.1397	5.1744	14.4667	14.6321
	5.2092	5.2440	14.7994	14.9667
	5.2788	5.3136	15.1360	15.3052
	5.3484	5.3832	15.4764	15.6476
	5.4180	5.4528	15.8208	15.9939
	5.4876	5.5224	16.1690	16.3441
	5.5572	5.5920	16.5212	16.6982
	5.6268	5.6616	16.8772	17.0562
	5.6964	5.7312	17.2372	17.4181
	5.7660	5.8008	17.6010	17.7839
	5.8356	5.8705	17.9688	18.1536
	5.9053	5.9401	18.3405	18.5273
	5.9749	6.0097	18.7161	18.9048
	6.0445	6.0794	19.0956	19.2863
	6.1142	6.1490	19.4790	19.6717
	6.1838	6.2187	19.8664	20.0610
	6.2535	6.2883	20.2577	20.4543
	6.3232	6.3580	20.6529	20.8515
	6.3928	6.4277	21.0521	21.2526
	6.4625	6.4973	21.4551	21.6577

No. de Bits	d_i	r_i	d_i	r_i
8				
	6.5322	6.5670	21.8622	22.0667
	6.6019	6.6367	22.2731	22.4796
	6.6716	6.7064	22.6880	22.8965
	6.7413	6.7761	23.1069	23.3173
	6.8110	6.8458	23.5297	23.7420
	6.8807	6.9155	23.9564	24.1707
	6.9504	6.9853	24.3870	24.6034
	7.0201	7.0550	24.8217	25.0399
	7.0899	7.1247	25.2602	25.4805
	7.1596	7.1945	25.7027	25.9249
	7.2294	7.2642	26.1491	26.3733
	7.2991	7.3340	26.5995	26.8257
	7.3689	7.4038	27.0539	27.2820
	7.4387	7.4736	27.5121	27.7423
	7.5085	7.5434	27.9744	28.2065
	7.5783	7.6132	28.4405	28.6746
	7.6481	7.6830	28.9107	29.1467
	7.7180	7.7529	29.3848	29.6229
	7.7879	7.8228	29.8629	30.1030
	7.8578	7.8928	30.3451	30.5872
	7.9279	7.9629	30.8314	31.0756
	7.9981	8.0332	31.3220	31.5684
	8.0685	8.1037	31.8171	32.0657
	8.1392	8.1746	32.3170	32.5682
	8.2103	8.2460	32.8223	33.0765
	8.2822	8.3183	33.3342	33.5920
	8.3551	8.3920	33.8545	34.1169
	8.4298	8.4677	34.3860	34.6551
	8.5072	8.5467	34.9342	35.2132
	8.5888	8.6310	35.5082	35.8032
	8.6776	8.7242	36.1251	36.4470
	8.7788	8.8334	36.8181	37.1892
	8.9037	8.9740	37.6612	38.1333
	9.0848	9.1955	38.8718	39.6104
	∞		∞	

APÊNDICE B

Os programas desenvolvidos para as simulações de interpolação e codificação de imagens foram todos implementados no MATLAB, usando-se principalmente os "Toolboxes" de processamento de sinais e processamento de imagens, além de outras funções que foram desenvolvidas para se realizar as simulações.

As imagens usadas nas simulações, tais como a Lena, Peppers, Pahka, Maine, além de outras são todas de domínio público, e podem ser adquiridas de várias fontes através da rede internet.

Para realizar as simulações, foram implementados em primeiro lugar os programas que calculam a transformada de uma imagem por blocos, usando a DCT, LOT e DSTr-2D. A DCT implementada é referida como DCT-II, que é a mais usada para codificação de imagens. A LOT é implementada de acordo com o algoritmo descrito em [5]. A DSTr-2D tem duas particularidades em sua implementação; em primeiro lugar, é realizado uma pré-rotação de eixos sobre as linhas e colunas de cada bloco. Devido ao procedimento de rotação, uma grade, formada pela primeira linha e primeira coluna de cada bloco, permanece na forma original. Aplica-se, então, o procedimento de pré-rotação sobre essas linhas originais, de modo que somente o primeiro elemento de cada bloco permanece na forma original. A transformada seno discreta é então aplicada de duas formas diferentes sobre cada bloco. Sobre a primeira linha e primeira coluna aplica-se a transformada seno discreta unidimensional e, sobre o restante do bloco, aplica-se a transformada seno discreta bidimensional. É apresentado a seguir o programa fonte que implementa a DSTr-2D, escrito para MATLAB. O programa tem como entrada a imagem no formato do MATLAB e o tamanho do bloco. A imagem na entrada pode ter qualquer tamanho, desde que seja um múltiplo do tamanho do bloco. Os programas aqui apresentados não estão otimizados, com relação ao tempo de processamento, pois o objetivo principal do trabalho é a apresentação de uma técnica usada para codificação de imagem com redução de efeito de blocos através de uma solução simples, com o mínimo de esforço computacional adicional além de não apresentar nenhuma informação paralela para se conseguir o objetivo almejado, quando comparado a outras técnicas de codificação, por exemplo, usando DCT.

Em seguida, são apresentados os programas em MATLAB implementados para se calcular as transformadas de imagens por blocos, usando DSTr, DCT e LOT.

Programa em MATLAB para calcular a transformada seno discreta com rotação de eixos bidimensional (DSTr-2D), direta e inversa, de uma imagem **xx** de tamanho **MxN**, usando blocos de tamanho **bl x bl**. A variável de entrada **t** indica se a transformada a ser realizada é direta (**t=1**) ou inversa (**t≠1**).

```
function y=dstr2 (xx, bl, t)
% function y=dstr2 (xx, bl, t). Calcula a DSTr-2D direta(t=1) e inversa (t≠1)de uma imagem xx usando
% blocos de tamanho blxbl, conforme é descrito no capítulo 2, seção 2.7.

%Cálculo das funções bases para a transformada seno discreta (DST) de comprimento bl-1.
for k=1:bl-1,
 for n=1:bl-1,
  s(n,k)=sin(n*k*pi/bl);
 end
end

fq=sqrt(2/bl);
s=s*fq;
fk=2;
[M,N]=size(xx);
n1=M/bl; n2=N/bl;
xx(M+1,:)=xx(1,:);
xx(:,N+1)=xx(:,1);
for i2=0:n2-1,
 k2=bl*i2+1;
 for i1=0:n1-1,
  k1=bl*i1+1;
  x=xx(k1:k1+bl,k2:k2+bl);

  if t==1
% Realiza a pré-rotação de eixos sobre o bloco x( transformada direta).
    xt=prerota(x);
%  Aplica a DST-2D no sub-bloco de tamanho (bl-1)(bl-1) a partir da segunda
   linha e segunda coluna, a DST-1D na primeira linha  e primeira coluna.
    xt(2:bl,2:bl)=s*(xt(2:bl,2:bl))*s;
    xt(1,2:bl)=xt(1,2:bl)*s*fk;
    xt(2:bl,1)=s*xt(2:bl,1)*fk;
  else

%  Transformada inversa.
     x(2:bl,2:bl)=s*(x(2:bl,2:bl))*s;
    x(2:bl,1)=s*x(2:bl,1)/fk;
    x(2:bl,bl+1)=s*x(2:bl,bl+1)/fk;
    x(1,2:bl)=x(1,2:bl)*s/fk;
    x(bl+1,2:bl)=x(bl+1,2:bl)*s/fk;

%  Procedimento de pós-rotação de eixos do bloco.
    xt=posrota(x);
  end
y(k1:k1+bl-1,k2:k2+bl-1)=xt(1:bl,1:bl);
end
end
```

```
   function y = prerota(xb)
% Function y = prerota (xb) realiza o procedimento de pré-rotação
% de eixos de um bloco xb de tamanho bl x bl.
for jk = 1:2,
  [L,C]=size(xb);
  z = 0;
    for i = 1:L,
      y1 = xb(i,1); y2=xb(i,C);
      a = (y2-y1)/(C-1);
      z(1:C) = xb(i,1:C)-(a*(0:C-1)+y1);
      xb(i,2:C-1) = z(2:C-1);
      end
   xb=xb';
end
y = xb;
```

```
function y = posrota(xb)
% Function y = posrota (xb) realiza o procedimento de pós-rotação
% de eixos de um bloco xb de tamanho bl x bl.

for jk = 1:2,
  [L,C] = size(xb);
  z = 0;
    for i = 1:L,
      y1 = xb(i,1); y2 = xb(i,C);
      a = (y2-y1)/(C-1);
      z(1:C) = xb(i,1:C)+(a*(0:C-1)+y1);
      xb(i,2:C-1) = z(2:C-1);
    end
  xb=xb';
end
y = xb;
```

Programa em MATLAB para se calcular a transformada DCT de uma imagem **xx** de tamanho **MxN**, usando blocos de tamanho **bl x bl**.

```
function yo = dct2 ( xx, bl )

% function yo = dct2(xx, bl)  calcula a DCT-2Dde uma imagem xx usando blocos de tamanho bl x bl.
D = dctbase(bl);
yo = blkproc(X, [bl bl], 'dctmat', D);
```

```
function yo = dctmat(xx, D)
% function yo = dctmat(xx, D) calcula a DCT-2D de uma matriz xx de tamanho MxM, como um único bloco.
% D é uma matriz de dimensão MxM composta pelas funções bases.
yo = D'*xx*D;
```

```
function yo = idct2(xx ,bl)
function yo = idct2(xx, bl)  calcula a DCT-2D inversa de uma imagem xx usando blocos de tamanho bl x bl.
D = dctbase(bl);
out = blkproc(X, [bl bl], 'idctmat', D);



function out = idctmat(xx, D)

% function yo = dctmat(xx, D) calcula a DCT-2D inversa de uma matriz xx de tamanho MxM, como
% um único bloco.
% D é uma matriz de dimensão MxM composta pelas funções bases.

yo = D*xx*D';
```

Programa em MATLAB para calcular a transformada LOT ("Lapped Orthogonal Transform") de uma imagem **xx** de tamanho **MxN**, usando blocos de tamanho **bl x bl**.

```
function yo = LOT(X, bl);
% LOT(X, bl) Calcula a transformada LOT de uma imagem xx, usando blocos de tamanho bl x bl.
P = LOTbase(bl)
[L, M] = size(P);
[L, C] = size(X);
NL = L/M;
NC = C/M;
D = dctbase(M);
for i = 1:M/2
  De(:,i) = D(:, 2*i-1);
  Do(:,i) = D(:, 2*i);
end
J  = ceye(M);
J2 = ceye(M/2);
He = De(1:M/2, :);
P1 = 0.5*[2*He  2*He; J*(De-Do) -J*(De-Do)];
P2 = 0.5*[De-Do De-Do; 2*J2*He -2*J2*He];
y = zeros(L, C);
for i = 1:NL
 if i == 1
  Pa = P1;
 else
  if i == NL
   Pa = P2;
  else
   Pa = P;
  end
 end
 for j = 1:NC
  if j == 1
   Pd = P1;
  else
   if j == NC
    Pd = P2;
```

```
    else
      Pd = P;
    end
  end
  yo((M*(i-1)+1):(M*i), (M*(j-1)+1):(M*j)) = Pa'* X(max(M*(i-1)+1-M/2, 1):min(M*i+M/2, L),
  max(M*(j-1)+1- M/2, 1):min(M*j+M/2, C)) * Pd;
 end
end
```

```
function yo = ILOT(X, bl, fact);
% function yo = ILOT(X, bl, fact). Calcula A LOT inversa de uma imagem X usando blocos de tamanho blxbl.
% fact é usado para normalizar as bases DCT usadas nas bordas. Se não for especificado é considerado ser
% igual a 1.

P=LOTbase ( bl )
[L, M] = size(P);
[L, C] = size(X);
NL = L/M;
NC = C/M;

D = DCTbase(M);
for i = 1:M/2
  De(:,i) = D(:, 2*i-1);
  Do(:,i) = D(:, 2*i);
end
J  = ceye(M);
J2 = ceye(M/2);
He = De(1:M/2, :);
P1 = fact*0.5*[2*He  2*He; J*(De-Do) -J*(De-Do)];
P2 = fact*0.5*[De-Do De-Do; 2*J2*He -2*J2*He];
yo = zeros(L+2*M, C+2*M);
i1 = 1; i2 = 3*M/2;
j1 = 1; j2 = 3*M/2;
for i = 1:NL
 if i == 1
  Pa = P1;
 else
  if i == NL
   Pa = P2;
  else
   Pa = P;
  end
 end
 for j = 1:NC
  if j == 1
   Pd = P1;
  else
   if j == NC
    Pd = P2;
   else
    Pd = P;
   end
  end
  Paux = Pa* X((M*(i-1)+1):(M*i), (M*(j-1)+1):(M*j)) * Pd';
  yo(i1:i2, j1:j2) = yo(i1:i2, j1:j2) + Paux;
```

```
    if j == 1
     j1 = j1 + M/2; j2 = j2 + M;
    else
     if j == (NC-1)
      j1 = j1 + M; j2 = j2 + M/2;
     else
      j1 = j1 + M; j2 = j2 + M;
     end
    end
   end
   j1 = 1; j2 = 3*M/2;
   if i == 1
    i1 = i1 + M/2; i2 = i2 + M;
   else
    if i == (NL-1)
     i1 = i1 + M; i2 = i2 + M/2;
    else
     i1 = i1 + M; i2 = i2 + M;
    end
   end
  end
  yo = yo(1:L,1:C);
```

```
function  yo = LOTbase(N)
% LOTbase(bl) retorna a matriz LOT P baseada na DCT, com máximo ganho de codificação
% conforme é descrito em Malvar, "Signal Processing with Lapped Orthogonal Transform", Artech
% House, 1992.

D = DCTbase(N);
for i = 1:N/2
  De(:,i) = D(:, 2*i-1);
  Do(:,i) = D(:, 2*i);
end
J = ceye(N);
P0 = 0.5*[(De-Do) (De-Do); J*(De-Do)  -J*(De-Do)];
Rxx = Toeplitz(0.95.^(0:(2*N-1)));
Ro = P0'*Rxx*P0;
[Z, D] = eig(Ro);
yo = P0*Z;
```

```
function  D = DCTbase(N)
% DCTbase(N)
% Calcula as bases DCT usadas para o cálculo das funções bases da LOT
c = ones(1, N);
c(1) = 1/sqrt(2);
for k = 1:N
   D(1:N, k) = (c(k)*sqrt(2/N)*cos(((1:N)-0.5)*(k-1)*pi/N))';
end
```

Referências Bibliográficas

1. Jain, Anil K., " Fundamentals of Digital Image Processing ", Prentice-Hall,1987.
2. Schafer, R. W. and L. R. Rabiner, "A Digital Signal Processing Approach to Interpolation", Proceedings of the IEEE, Vol. 61, No. 6, pp. 692 - 702, June 1973.
3. Chochiere, R. E. and Rabiner, L. R., "*Interpolation and Decimation of Digital Signals- a Tutorial Review*", Proc. IEEE, vol 69, pag. 300-331, March 1981.
4. Lim, Jae S., " Two-dimensional Signal Processing and Image Processing ", Prentice-Hall,1990.
5. Malvar, H. S., " Signal Processing with Lapped Transform ", Artech House, 1992.
6. Poularikas, Alexander D., "The Transforms and Applications Handbook ", CRC Press and IEEE Press, 1996.
7. Rao, K. R., and P. Yip, " Discrete Cosine Transform ", Academic Press, Inc, 1990.
8. Max, Joel, "Quantizing for Minimum Distortion ", IRE Trans. Inform. Theory, vol. IT-6, pp. 267-272, March 1960.
9. Jayant, N. S., and P. Noll, "Digital Coding of Waveforms", Prentice-Hall, 1984.
10. Oppenheim, A. V., and R. W. Schafer, " Discrete-Time Signal Processing " Prentice-Hall, 1989.
11. Clarke, R. J., " Relation Between de Karhunen-Loève and Cosine Transforms ", Proc.IEEE, Pt. F, vol. 128, November 1981, pp. 359-360.
12. Malvar, H. S., and D. H. Staelin "Reduction of Blocking Effect in Image Coding with Lapped Orthogonal Transform", IEEE Intl. Conf. on Acoustics, Speech, and Signal Processing, New York, pp. 781-748, April 1988.
13. Malvar, H. S. and D. H. Staelin " The LOT: Transform Coding Without Blocking Effect", IEEE Trans. on Acoustics, Speech, and Signal Processing, vol. 37, April 1989.
14. Fraser, D. "Interpolation by the FFT Revised - An Experimental Investigation", IEEE Trans. on ASSP, vol. 17, No. 5, pp. 665 - 675, May, 1989.
15. Prasad, K. P. and P. Satyanarayana, "Fast Interpolation Algorithm Using FFT", Eletronics Letters , Vol. 22, No. 4, pp. 185 - 187, February 1986.
16. Agbinya J. I., "Interpolation Using the Discrete Cosine Transform", Eletronics Letters , Vol. 28, No. 20, pp. 1927 - 1928, September 1992.

17. Agbinya J. I., "Two-Dimensional Interpolation of Real sequence using the DCT ", Eletronics Letters , Vol. 29, No. 2, pp. 204 - 205, January 1993.

18.Wang, W., "Interpolation Using the Type I Discrete Cosine Transform", Eletronics Letters , Vol. 26, No. 15, pp. 1170 - 1171, 1990.

19. Wang, W., "Interpolation Using the Discrete Cosine Transform : Reconsideration", Eletronics Letters , Vol. 29, No. 2, pp. 198 - 200, January 1993.

20. Gonzalez R. C., "Digital Image Processing", Addison-Wesley, 1992.

21.Araújo, A. M. L. A. Klautau Jr. E V. G. Macedo, "Interpolação usando FFT Associada a Processos de Rotação", Anais do XVII CNMAC, Vol. 2, pag. 638-642, Vitória-ES, Setembro de 1994.

22. Araújo, A. M. L. and A.M. Faria Jr., "Interpolação pela Transformada Discreta em Seno com pré e pós- Rotação", XIV SBT, Curitiba - PR, Setembro de 1996.

23. Pelaes, E. G. , Yuzo Iano e A. M. L. Araújo, "Interpolação de Imagens por Transformadas com Eliminação de Extrapolações e Reduzido Efeito de Bloqueamento", Anais do XV Simpósio Brasileiro de Telecomunicações, Recife-PE Setembro de 1997.

24. Keys, R. G., "Cubic Convolution Interpolation for Digital Image Processing", IEEE Trans. on Acoustics, Speech, and Signal Processing, vol. 6, December 1981.

25. DeVore, R. A. and B. Jawerrh, and B. J. Lucier, "Image Compression Through Wavelet Transform", IEEE Trans. on Inform. Theory, vol. 38, No. 2, pp. 719-746, March 1992.

26.Siriam, P., and M. w. Marcellin, "Image Coding Using Wavelet Transform", IEEE Trans. on Image Processing, vol. 4, No. 6, June 1995.

27. Pelaes, E. G. e Yuzo Iano, "Image Interpolation with low Blocking Effects Using Discrete Sine with Axis Rotation", Proc. of XV International Telecommunication Symposium - ITS98, São Paulo - SP, Brazil, August 1998.

28. Segall, A. "Bit Allocation and Ecoding for Vector Sourse", IRE Trans. on Information Theory, vol. IT-22, March 1976, pp. 162-169.

29. Chen, W. H., and C. H. Smith, "Adaptative Coding of Monochrome and Color Images", IEEE Trans. on Communications, Apr. 1991, pp. 31-44.

30. Pelaes, E. G., Antonio M. L. Araújo, and Yuzo Iano, "Interpolation by Block Transform with Low Block Effect", VII Simposio de Ingenirie Eléctrica, Santa Clara - Cuba, June 1997.

31. Pelaes, E. G., and Yuzo Iano, "Image Coding Using Discrete Sine Transform with Axis Rotation", Submitted to the IEEE Transactions on Consumer Electronics, July 1998.

32. Pelaes, E. G., A. C. P. Veiga, Y. Iano, and E. A. Khalifa, "HDTV Transmission in Brazil: COFDM or 8-VSB?", Proc. of Third International Electronic Engineering Conference - INTERCON 98, Trujillo - Peru, August 1997.

33. Veiga, A. C. P., E. G. Pelaes, Y. Iano, and E. A. Khalifa, "HDTV: The New Perspectives of Television in Brazil", Proc. of Third International Electronic Engineering Conference - INTERCON 96, Trujillo - Peru, August 1996.

34. Wu, P. C, Ling-Gee Chen, and Yeong-Kang Lai, "A Block Shifting Method for Reduction of blocking Effects in Subband/Wavelet Image Coding", IEEE Transactions on Consumer Electronics, Vol. 44, No. 1, February 1998.

35. Wang, Z., and Lieng Wang, "Interpolation Using the Fast Discrete Sine Transform", Signal Processing, vol. 26, 1992, pp. 131-137.

36. Chen, C.-T. and T. Russel Hsing, "Review: Digital Coding for Visual Communicatios", Journal of Visual Communication and Image Representation, Vol. 2, No. 1, March 1991.

37. L., K.-T W.-K Chan, "Development of Simple Orthogonal Transform for Image Compression", IEE.-Vis. Image Signal Processing, Vol. 2, No. 1, February 1995.