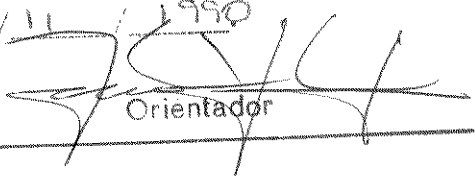
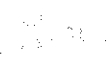


UNIVERSIDADE ESTADUAL DE CAMPINAS
FACULDADE DE ENGENHARIA ELÉTRICA
DEPARTAMENTO DE COMUNICAÇÕES

Este exemplar corresponde à redação final da tese
defendida por Ricardo Lopes de
Queiroz e aprovada pela Comissão
Julgadora em 02/11/1990

Orientador

SISTEMAS DE MULTIRESOLUÇÃO PARA
EXTRAÇÃO DE REDUNDÂNCIAS E TRANSMISSÃO
PROGRESSIVA NA CODIFICAÇÃO DE IMAGENS

Ricardo Lopes de Queiroz 

Orientador: Dr. João Baptista Tadanobu Yabu-uti 

Tese apresentada à Faculdade de Engenharia Elé-
trica da Universidade Estadual de Campinas como
parte dos requisitos exigidos para a obtenção
do título de MESTRE EM ENGENHARIA ELÉTRICA.

Novembro/90

00/9100736

BANCA EXAMINADORA

PROF. DR. JOÃO BATISTA TADANOBU YABU-UTI
DEPARTAMENTO DE COMUNICAÇÕES-FACULDADE DE ENGENHARIA ELÉTRICA
UNIVERSIDADE ESTADUAL DE CAMPINAS

PROF. DR. HENRIQUE SARMENTO MALVAR
DEPARTAMENTO DE ENGENHARIA ELÉTRICA-FACULDADE DE TECNOLOGIA
UNIVERSIDADE DE BRASÍLIA

PROF. DR. JOÃO MARCOS TRAVASSOS ROMANO
DEPARTAMENTO DE COMUNICAÇÕES-FACULDADE DE ENGENHARIA ELÉTRICA
UNIVERSIDADE ESTADUAL DE CAMPINAS

Agradecimentos :

Aos meus pais, pelo suporte e incentivo, aos quais dedico este trabalho.

Ao meu orientador pela amizade e prestatividade.

Ao Prof. Dr. Henrique Malvar, da Universidade de Brasília, pelas discussões, críticas e sugestões em alguns dos temas tratados.

Aqueles, colegas e funcionários da FEE, que direta ou indiretamente colaboraram.

A Ana Maria por sua paciência.

RESUMO :

Este trabalho visa o desenvolvimento de técnicas que explorem múltiplas resoluções de uma imagem para sua codificação. Baseando-nos em técnicas de interpolação e codificação piramidal, foram estudados procedimentos para formação de pirâmides sem prefiltragem. Estas ferramentas serviram de base para a elaboração do DPCM em pirâmide que combina várias taxas de amostragem ao DPCM convencional, formando um sistema híbrido extra-interpolativo. Este novo sistema pode reduzir ainda mais a complexidade do DPCM, sendo mais flexível e eficiente. Além disto, novos métodos de codificação para transmissão progressiva foram abordados, motivando o desenvolvimento da Pirâmide Quincunx sem Prefiltragem e de um método bastante eficiente para tal, que utiliza a Lapped Orthogonal Transform em combinação com estratégias de transmissão sequencial de coeficientes e planos-de-bit.

ABSTRACT :

This work is concerned about the development of techniques to explore multiple resolutions in image coding. Based on the knowledge of interpolation techniques, as well as pyramid coding strategies, non-prefiltered pyramids were studied. These tools were the base to develop the Pyramid DPCM, a scheme that combines multirate processing and regular DPCM, resulting in a hybrid extra-interpolative system. This new approach can reduce DPCM's complexity, and still be more efficient. Furthermore, new coding strategies for progressive transmission were discussed through the development of the Non-prefiltered Quincunx Pyramid and by the efficient use of the Lapped Orthogonal Transform combined with sequential schemes for transmitting coefficients and bit-planes.

SISTEMAS DE MULTIRESOLUÇÃO PARA EXTRAÇÃO DE REDUNDÂNCIAS E
TRANSMISSÃO PROGRESSIVA NA CODIFICAÇÃO DE IMAGENS

ÍNDICE

	PÁGINA
CAPÍTULO 1 - INTRODUÇÃO.....	1
1.1-PADRÕES DE DIGITALIZAÇÃO DAS IMAGENS DE TESTE.....	2
1.2-IMAGENS DE TESTE.....	5
1.3-TRANSMISSÃO PROGRESSIVA COM COMPRESSÃO DE DADOS.....	5
1.4-ABORDAGEM.....	6
1.5-REFERÊNCIAS.....	7
CAPÍTULO 2 - INTERPOLAÇÃO E DIZIMAÇÃO DE SINAIS DISCRETOS....	8
2.1-DIZIMAÇÃO E INTERPOLAÇÃO COM FILTROS IDEAIS.....	8
2.1.1-Dizimação por fator inteiro e o prefiltro.....	10
2.1.2-Interpolação por fator inteiro e o posfiltro.....	11
2.1.3-Estruturas polifásicas.....	12
2.2-PROJETO DE UM INTERPOLADOR.....	13
2.2.1-Aproximações polinomiais.....	16
2.2.2-Otimização de interpoladores.....	20
2.2.3-Otimização conjunta do posfiltro e do prefiltro..	24
2.3-INTERPOLAÇÃO NO CENTRO DO INTERVALO DE AMOSTRAGEM.....	27
2.4-REFERÊNCIAS.....	32
CAPÍTULO 3 - CODIFICAÇÃO PIRAMIDAL.....	34
3.1-PRINCÍPIOS DE CONSTRUÇÃO E DECODIFICAÇÃO.....	34
3.2-PREFILTROS E POSFILTROS.....	38
3.3-PIRÂMIDES REDUZIDAS.....	41
3.3.1-Pirâmide de diferenças reduzida.....	43
3.3.2-Pirâmides sem prefiltragem.....	44
3.4-PIRÂMIDES QUINCUNX SEM PREFILTRAGEM.....	46
3.4.1-Estrutura Polifásica Bidimensional na Dizimação Retangular.....	46
3.4.2-Dizimação quincunx/retangular e o interpolador...	50
3.4.3-Interpoladores práticos.....	53
3.4.4-Resultados comparativos.....	54
3.5-COMENTÁRIOS ADICIONAIS.....	55
3.5.1-Relação com bancos de filtros.....	55
3.5.2-Adaptatividade.....	59
3.5.3-Dimensões da imagem para apresentação ao usuário.	60
3.6-REFERÊNCIAS.....	61

CAPÍTULO 4 - DPCM EM PIRÂMIDE : UM SISTEMA HÍBRIDO.....	63
4.1-DEFINIÇÃO DO MÉTODO.....	64
4.2-COMPARAÇÕES COM SISTEMA CONVENCIONAL.....	68
4.2.1-Vantagens e desvantagens.....	68
4.2.2-Preditores e interpoladores simples.....	69
4.2.3-Entropias de primeira ordem.....	73
4.3-TRANSMISSÃO CONDICIONAL DE BLOCOS.....	76
4.4-SIMULAÇÕES.....	77
4.5-CONSIDERAÇÕES ADICIONAIS.....	88
4.6-REFERÊNCIAS.....	89
 CAPÍTULO 5 - HIERARQUIA UTILIZANDO TRANSFORMADAS	
SOBREPOSTAS (LT's).....	90
5.1-BANCOS DE FILTROS E LT's.....	90
5.2-HIERARQUIA EM ÁRVORE OU DECOMPOSIÇÃO DIRETA?.....	96
5.3-TRANSMISSÃO PROGRESSIVA COM DECOMPOSIÇÃO DIRETA.....	99
5.3.1-Alocação de quantizadores.....	99
5.3.2-Critérios de Transmissão progressiva.....	102
5.3.3-Comentários Adicionais.....	104
5.5-REFERÊNCIAS.....	105
 CAPÍTULO 6 - CONCLUSÃO.....	107
 APÊNDICE - IMAGENS ORIGINAIS E PROCESSADAS	

CAPÍTULO 1

INTRODUÇÃO

A representação digital de sinais analógicos vem recebendo grande atenção nas últimas décadas devido ao aumento de complexidade e velocidade de processamento que os equipamentos digitais vem sofrendo. Esta maior complexidade é acompanhada de redução de custos e aumento de flexibilidade, permitindo aplicações não antes imaginadas. Dentro do escopo de sistemas digitais, o Processamento Digital de Sinais vem, nos últimos anos, ganhando destaque devido ao crescente número de aplicações para tal ferramenta.

Sinais digitalizados têm comportamento razoavelmente imune às condições do meio, como temperatura e umidade, e não sofrem a ação do tempo. Além disso, o processamento de sinais, já digitalizados, é realizado através de sistemas digitais, cujo comportamento é completamente previsível (salvo erros de programação ou falhas de equipamento). Com isto podemos realizar, exatamente, determinado processamento aplicado ao sinal. A transmissão de sinais digitalizados também oferece vantagens dada a sua maior robustez contra erros de canal, como ruídos espúrios, interferências e distorções inerentes. Sinais comumente digitalizáveis englobam, entre outras, as seguintes classes: Áudio, Voz e Imagens. Aqui trataremos de sinais correspondendo a Imagens. Tais sinais podem corresponder a cenas estáticas, como em uma fotografia, ou sequências de cenas formando um sinal de imagens em movimento. Típicos exemplos são os de fotografias, sinais analógicos de vídeo e televisão, películas de cinema, aerofotos, imagens de satélite, etc... Todos estes sinais são devidamente digitalizados para um processamento adequado.

Um grande problema surge quando pensamos em digitalizar imagens. Devido à resolução geralmente exigida, tipicamente requer-se uma imagem amostrada em um mosaico de 256×256 , 512×512 ou mais amostras. Supondo que estejamos tratando imagens

estáticas (em apenas uma de uma sequência de movimentos) e que digitalizemos o sinal com uma quantização uniforme de 256 níveis, teríamos 8 bits por amostra e, por exemplo, $512 \times 512 \times 8$ bits para representá-la totalmente. Isto significa que mais de 2 milhões de bits seriam necessários para representar uma única imagem monocromática, já que nem mencionamos estratégias para representação das cores. Apenas pensamos na intensidade de luz emitida ou refletida pelo ponto e incidente ao observador. Para a redução do montante de informação binária na representação de uma imagem, técnicas de compressão de dados são fundamentais.

Esta compressão se baseia na extração da redundância intrínseca à imagem caracterizada. Tal redundância pode ser estatística ou subjetiva. Redundância estatística pode ser eliminada sem qualquer alteração à imagem. Isto é possível com uma codificação mais eficiente dos símbolos (binários ou não). Por outro lado, a extração de redundância subjetiva deturpa a imagem original procurando explorar ao máximo as deficiências do sistema visual humano. Almeja-se, contudo, que o observador não consiga distinguir ou perceber os erros introduzidos pela codificação. Detalhes sobre técnicas de codificação, bem como aplicações e conceitos do Processamento Digital de Imagens podem ser encontrados em [1] e [2].

O principal objetivo deste trabalho é desenvolver um sistema simples e eficiente, explorando a correlação entre múltiplas resoluções de uma mesma imagem, para codificação de imagens com baixa complexidade e passível de ser aplicado em CODECs de TV. Este é o DPCM em Pirâmide que será apresentado. Também é nosso objetivo estudar sistemas simplificados de codificação em pirâmide, bem como outros sistemas eficientes, para transmissão progressiva. Através deste trabalho teremos como preocupação a codificação de imagens estáticas digitalizadas. Para algumas seções, aplicaremos os métodos descritos em simulações de codificação, principalmente, em imagens monocromáticas.

1.1) PADRÕES DE DIGITALIZAÇÃO DAS IMAGENS DE TESTE

Procurou-se, tanto quanto possível, aproximar os padrões de digitalização aos dispostos na Recomendação 601 do CCIR [3]. Porém a limitação dos recursos disponíveis nos fez segui-las em princípio e em parte, mas nada que altere as características destas imagens para o propósito de simulação. O importante é que as imagens disponíveis obedeceram a um padrão apropriado, umas com maior e outras com menor qualidade de digitalização. É imaginada uma imagem como uma matriz retangular de pontos amostrados a partir de um plano contínuo de intensidade luminosa (cena). Esta matriz tem dimensões finitas e é sub-parte de uma matriz infinita que representa a cena inteira. Tal matriz finita é dita mosaico de amostras e cada elemento (amostra ou ponto) é denominado *pixel*, do inglês *picture element*, e abreviado por *pel*. Cada pixel possui informação sobre a intensidade (luminância) e coloração (crominância) da luminosidade daquele ponto da cena. Para que possamos representá-los digitalmente, precisamos separar estas informações e discretizar as amplitudes dos sinais.

Representando cada pixel em bases tricromáticas, pode-se reproduzir quase todos as cores e iluminações possíveis [1] [2]. Com os sinais ditos RGB (vermelho, verde e azul) podemos representar adequadamente a informação deste pixel. A partir dos três sinais: R, G e B, precisamos digitalizá-los, discretizando a amplitude, às custas de uma quantização uniforme com 256 níveis. Estes níveis podem ser, por sua vez, representados por oito dígitos binários, ou 8 bits, ou um *byte*. Desta forma os sinais RGB assumiriam valores discretos variando de 0 a 255. Em resumo, representa-se a imagem através de três mosaicos de amostras, cada um digitalizado com 8 bits/pixel, somando 24 bits/pixel no total.

Para transformar os sinais das bases tricromáticas em sinais de luminância e crominância, devemos proceder como em [3].

Sendo R,G,B sinais discretos, os quais podem variar de 0 a 255 com amplitudes discretas, e abandonando índices temporais

ou espaciais, temos para cada pixel que encontrar os sinais R' , G' e B' que são sinais escalonados:

$$R' = \text{INT} \left(\frac{219}{255} R \right) + 16$$

$$G' = \text{INT} \left(\frac{219}{255} G \right) + 16$$

$$B' = \text{INT} \left(\frac{219}{255} B \right) + 16$$

sendo $\text{INT}(x)$ o inteiro mais próximo de x , por arredondamento.

Os sinais de luminância (Y) e crominância (C_R e C_B) serão dados por

$$Y = \frac{77}{256} R' + \frac{150}{256} G' + \frac{29}{256} B'$$

$$C_R = \frac{131}{256} R' - \frac{110}{256} G' - \frac{21}{256} B' + 128$$

$$C_B = -\frac{44}{256} R' - \frac{87}{256} G' + \frac{131}{256} B' + 128$$

Seguindo as operações acima descritas, a partir dos bancos da imagem colorida, podemos encontrar outras bases para as mesmas cores, ou então encontrar a imagem monocromática correspondente (Y).

As imagens disponíveis para testes foram digitalizadas de formas distintas, mas, geralmente, em três bancos (R, G, B) para sinais coloridos. Das seis imagens disponíveis, quatro são coloridas (ou seja; dispomos dos sinais RGB) e foram digitalizadas com uma frequência de amostragem de 10MHz, utilizando filtros ópticos para separação das componentes através da exposição de diapositivos a câmeras acopladas eletronicamente a conversores analógico/digital [4]. Desta forma, as imagens são formadas pela amostragem ao longo das linhas de varredura de uma câmera. Todavia, campos entrelaçados são combinados de forma a montar um quadro fixo como proposto: uma matriz retangular de pixels ortogonalmente alinhados.

1.2) IMAGENS DE TESTE

As imagens disponíveis são as da Tabela 1.1 , que podem ser visualizadas no Apêndice. Na tabela são indicados: referência das figuras no Apêndice; "apelido" dado à imagem; as resoluções; disponibilidade dos formatos RGB ou Y.

Tabela 1.1 - Imagens de Teste

	NOME	RESOLUÇÃO	FORMATO
(a)	ZELDA	512x512	RGB
(b)	GIRL	512x512	Y
(c)	PRAIA	512x512	RGB
(d)	COZINHA	512x512	RGB
(e)	CAMERAMAN	512x512	Y
(f)	SALA	512x512	RGB

1.3) TRANSMISSÃO PROGRESSIVA COM COMPRESSÃO DE DADOS

A necessidade de extração de redundâncias se torna evidente devido ao montante de dados requeridos para a representação da imagem. A compressão é vital para economia de banda no canal (transmissão) ou de memória (armazenagem). Neste trabalho iremos tratar de sistemas de multiresolução para compressão de dados que permitam transmissão progressiva.

Cabe agora ressaltar a importância e a necessidade de sistemas que permitam transmissão progressiva. Se a transmissão for muito lenta, seria desejável que o receptor pudesse receber, o mais rapidamente possível, o máximo de informação a respeito do conteúdo da imagem, de forma que o usuário pudesse abortar a

transmissão, antes de ser transmitida a maior parte da imagem, caso não seja esta a figura que esteja procurando. Em outras palavras, o receptor deve reconhecer rapidamente o que está contido na imagem e decidir se prossegue a transmissão para reconstrução total ou se a interrompe para rapidamente fazer uma nova consulta. Isto pode ser muito útil à consulta de bancos de imagens através de canais de baixa capacidade nos quais procuramos uma determinada imagem entre várias. Em sistemas convencionais de varredura progressiva, se a apresentação for bastante lenta, teremos que esperar tediosamente até que por volta de 70% (ou mais) da transmissão seja completada, para identificar o conteúdo. Cabe, ainda, ressaltar que é desejável que se possa reconhecer o conteúdo da imagem com a menor taxa de transmissão, mas compressão de dados também é requerida para que se possa formar um sistema eficiente.

1.4) ABORDAGEM

No Capítulo 2 ir-se-á abordar conceitos que servem de fundamento para os demais capítulos, no que tange o conhecimento de teoria de processamento digital de sinais a multitasas e serão apresentadas técnicas de interpolação através de filtros FIR. No Capítulo 3, a codificação em pirâmide é discutida com ênfase a esquemas simples, mas que impliquem em compressão de dados. Pontos de destaque são as pirâmides reduzidas, a generalização do processo de construção e esquemas sem prefiltragem. O Capítulo 4 apresenta um novo sistema, bastante simples, baseado na pirâmide sem prefiltragem de 3 níveis, para ser adicionado ao sistema DPCM convencional. Esta adição visa um aumento de eficiência com diminuição da velocidade de processamento dos CODECS e tem como objetivo aplicações em sinais de altas taxas de amostragem como CODECS de TV. Por tal, já que cada quadro é apresentado em um intervalo imperceptível de tempo, perde-se o sentido da aplicação de transmissão progressiva. No Capítulo 5 abordaremos brevemente a utilização de transformadas sobrepostas na transmissão progressiva que é um método com capacidade de alta compressão. Nossa meta é a aplicação deste método à codificação de imagens

estáticas. Por fim, no Capítulo 6, estão contidas as conclusões deste trabalho e comentários adicionais.

1.5) REFERÊNCIAS

- [1] W.K. Pratt, *Digital Image Processing*, John Wiley & Sons, 1978.
 - [2] R.Gonzales, P.Wintz, *Digital Image Processing*, Addison Wesley Publishing Co., 2nd ed., 1987.
 - [3] CCIR, Rec. 601-1, vol. XI, 1982-1986, *Encoding Parameters of Digital Television For Studios*.
 - [4] Y.Iano et al, *Imagens Digitalizadas da SMPTE Armazenadas em Fitas Magnéticas*, Contrato 208/87, RT 153, UNICAMP/TELEBRÁS, 1987.
-

CAPÍTULO 2

INTERPOLAÇÃO E DIZIMAÇÃO DE SINAIS DISCRETOS

Neste capítulo serão discutidos alguns conceitos básicos de sistemas com várias taxas de amostragem e interpolação de sinais unidimensionais que pertençam ao escopo deste trabalho.

2.1) DIZIMAÇÃO E INTERPOLAÇÃO COM FILTROS IDEAIS

Basicamente, um processo de dizimação consiste em reamostrar uma determinada sequência temporal em uma frequência menor, descartando $N-1$ a cada N amostras. Esta reamostragem deve ser periódica, com período inteiro em relação a sequência original, pelo fato de não dispormos das amostras no meio de intervalos entre instantes de amostragem. Com a dizimação haverá *aliasing*, ou interferência entre as bandas replicadas do sinal discreto [1] [2], se o sinal não for devidamente limitado em frequência. Esta limitação pode ser conseguida com a utilização de um Prefiltro.

O processo de Interpolação, por sua vez, é basicamente o oposto. São inseridos $N-1$ amostras nulas entre duas amostras vizinhas do sinal original, equivalendo a superamostrar este, supondo nulas as amostras não coincidentes com os instantes de amostragem iniciais. O sinal resultante conterá N réplicas, equiespaçadamente deslocadas, da banda básica do sinal original em seu espectro [1] [2]. Estas réplicas (*imagens*) podem ser atenuadas através do Posfiltro. A seguir serão apresentados alguns resultados que serão tomados como referência no transcorrer do trabalho.

Seja o sinal contínuo $x_a(t)$ e seu correspondente sinal discreto $x(n)$, obtido pela amostragem periódica de $x_a(t)$, de forma que $x(n) = x_a(nT)$, sendo T o período de amostragem. Assumimos que o sinal contínuo tem transformada de Fourier dada por

$$X_a(\omega) = \int_{-\infty}^{\infty} x_a(t) e^{-j\omega t} dt \quad (E 2.1)$$

e que desejamos obter a sequência $y(n)=x_a(n\tau)$, sendo τ um novo período de amostragem que é um fator ou múltiplo inteiro de T .

$$T/\tau = \text{inteiro} \quad \text{ou} \quad \tau/T = \text{inteiro} \quad (E 2.2)$$

Definindo $X(z)$, como a Transformada Z de $x(n)$, os resultados a seguir são facilmente encontrados em [1].

$$X(z) = \sum_n x(n) z^{-n} \quad (E 2.3)$$

$$X(e^{j\omega T}) = X(z) \Big|_{z=e^{j\omega T}} = \sum_n x(n) e^{-j\omega n T} = \frac{1}{T} \sum_n X_a(\omega + n 2\pi/T) \quad (E 2.4)$$

$$X(e^{j\omega T}) = \frac{1}{T} X_a(\omega) \quad |\omega| \leq \frac{\pi}{T} \quad (E 2.5)$$

$$x_a(t) = \sum_n x(n) \frac{\sin(\frac{\pi}{T}(t-nT))}{\frac{\pi}{T}(t-nT)} \quad (E 2.6)$$

Definiremos, agora, os símbolos representativos de superamostradores (expansores de frequência de amostragem) e subamostradores (compressores de frequência de amostragem), de acordo com [2].

$x(n) \rightarrow \boxed{\downarrow N} \rightarrow y(m)$

Subamostrador

 $y(m) = x(Nm)$

(E 2.7)

$x(n) \rightarrow \boxed{\uparrow N} \rightarrow y(m)$

Superamostrador

$$y(m) \begin{cases} = x(m/N) & m = 0, \pm N, \pm 2N, \dots \\ = 0 & m \neq 0, \pm N, \pm 2N, \dots \end{cases}$$

(E 2.8)

2.1.1) Dizimação por Fator Inteiro e o Prefiltro

Como já dito, a Dizimação consiste na reamostragem do sinal, mantendo uma a cada conjunto de amostras originais, e descartando as restantes. Se $Q = \tau/T$ (Q inteiro), temos:

$$y(n) = x(Qn) \quad (\text{E 2.9})$$

Pode ser demonstrado [3] que, neste caso:

$$Y(e^{j\omega\tau}) = \frac{1}{Q} \sum_{i=0}^{Q-1} X(e^{j(\frac{\omega\tau - 2\pi i}{Q})}) = \frac{1}{Q} \sum_{i=0}^{Q-1} X(e^{j(\omega T - 2\pi i/Q)}) \quad (\text{E 2.10})$$

Se o sinal original for limitado em banda a Ω , tal que $\Omega < \pi/\tau$, de (2.10) acha-se:

$$Y(e^{j\omega\tau}) = \frac{1}{Q} X(e^{j\omega T}) = \frac{1}{QT} X_a(\omega) = \frac{1}{\tau} X_a(\omega) ; |\omega| < \pi/\tau \quad (\text{E 2.11})$$

Se, por outro lado, $\Omega > \pi/\tau$, haverá *aliasing* entre faixas adjacentes do espectro periódico.

O sistema para Dizimação pode, então, ser ilustrado pela Figura 2.1, composto de um filtro, dito PREFILTRO (para limitar a frequência e evitar *aliasing*) e do subamostrador.

Com $f_s = 1/T$ e $\omega = 2\pi f$, as características do Prefiltro ideal devem ser:

$$H(f) \begin{cases} = 1 & \Rightarrow |f/f_s| \leq 1/2Q \\ = 0 & \Rightarrow 1/2Q < |f/f_s| \leq 1/2 \end{cases} \quad (\text{E 2.12})$$



Figura 2.1 - Esquema de um Dizimador, com o Prefiltro e o subamostrador.

2.1.2) Interpolação por Fator Inteiro e o Posfiltro

Neste caso, o inverso, a nova frequência de amostragem é maior. Como já comentado, um superamostrador irá inserir amostras de valor zero entre as amostras originais. Com uma multiplicação na frequência original de Q , teremos $Q = T/\tau$. Sendo Q um inteiro, T o período de amostragem inicial e τ o novo período. Reapresentando a equação (E 2.8) :

$$x_1(n) = x(n/Q) \quad n = k Q \quad (k \text{ inteiro qualquer})$$

$$x_1(n) = 0 \quad n \neq k Q \quad (\text{E 2.13})$$

São inseridas $Q-1$ amostras em cada intervalo de amostragem. Pode ser mostrado [3] que :

$$X_1(e^{j\omega\tau}) = X(e^{j\omega T}) \quad (\text{E 2.14})$$

Da equação acima, conclui-se que $X_1(e^{j\omega\tau})$ é periódico com período π/T ao invés de $\pi/\tau = Q\pi/T$. Isto nos informa que irão aparecer repetições, réplicas ou imagens, do espectro original, se superpondo com espaçamento de π/T . Deve-se filtrar $x_1(n)$ com intuito de retirar estas componentes indesejáveis. Este filtro não possui ganho unitário, já que o sinal interpolado $y(n)$ deve obedecer a (E 2.11).

$$Y(e^{j\omega\tau}) = \frac{1}{\tau} X_a(\omega) = \frac{Q}{T} X_a(\omega) \quad |\omega| < \pi/\tau \quad (\text{E 2.15})$$

Para obter $y(n)$ a partir de $x_1(n)$, devemos passar o último através de um filtro $H(e^{j\omega\tau})$, passa-baixas com corte em $\pi/Q\tau$. Este filtro trabalha com uma frequência $f_s=1/\tau$ e é chamado de POSFILTRO.

$$Y(e^{j\omega\tau}) = H(e^{j\omega\tau}) X_1(e^{j\omega\tau}) = H(e^{j\omega\tau}) X(e^{j\omega T})$$

$$Y(e^{j\omega\tau}) = H(e^{j\omega\tau}) \frac{1}{T} X_a(\omega) \quad |\omega| < \pi/\tau \quad (E 2.16)$$

Comparando (E 2.15) e (E 2.16) e levando em conta que $\omega = 2\pi f$ e $f_s = 1/\tau = Q/T$, as características deste filtro serão:

$$H(f) = \begin{cases} Q & |f/f_s| \leq 1/2Q \\ 0 & 1/2Q < |f/f_s| \leq 1/2 \end{cases} \quad (E 2.17)$$

A Figura 2.2 mostra o diagrama de um interpolador, composto de um superamostrador e de um Posfiltro.

2.1.3) Representação Polifásica do Interpolador

Na dizimação e interpolação pode-se notar que os filtros aplicados sempre estão localizados na terminação que trabalha com a maior taxa de amostragem. É possível escolher outra representação para as Figuras 2.2 e 2.4 de forma a levar em conta amostras nulas ou que serão descartadas. Como veremos posteriormente, existem vantagens, em se decompor um sinal em suas fases, para o projeto dos interpoladores. Para apresentar as estruturas polifásicas vamos desenvolver a expressão de convolução dada pela filtragem do sinal superamostrado $x(m)$ (obtido a partir de $x(n)$) fornecendo $y(m)$, como na Figura 2.2.

Da forma apresentada, e levando em conta que $x(n)$ tem apenas uma amostra não nula a cada Q amostras, temos que:

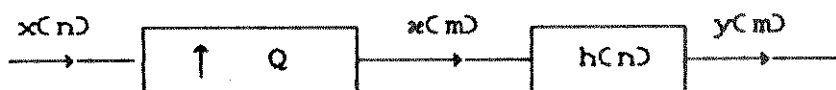


Figura 2.2 - Esquema de um Interpolador, com o Posfiltro e o superamostrador.

$$y(m) = \sum_n x(n)h(m-n) = \sum_n x(nQ)h(m-nQ) \quad (\text{E 2.18})$$

$$y(kQ+i) = \sum_n x(nQ)h(kQ+i-nQ) = \sum_n x(nQ)h((k-n)Q+i) \quad (\text{E 2.19})$$

$$\text{Sejam : } y_1(n) = y(nQ+i) \quad (\text{E 2.20})$$

$$h_1(n) = h(nQ+i) \quad (\text{E 2.21})$$

E lembrando que $x(n) = x(nQ)$, concluímos que:

$$y_1(m) = \sum_n x(n) h_1(m-n) \quad (\text{E 2.22})$$

2.2) PROJETO DE UM INTERPOLADOR

Para aproveitar uma das maiores propriedades de filtros digitais FIR, assumimos que os filtros aqui tratados devem possuir fase linear, o que implicaria em atraso de grupo constante ou nulo. Ao último nos referimos como de fase nula. Os conceitos de fase nula ou linear se misturam, já que dependem da implementação. O primeiro é não causal e requer fisicamente uma retenção de algumas amostras, equivalendo exatamente ao atraso provocado pela implementação de forma causal com fase linear. São, portanto, o mesmo sistema com visualização matemática diferente. Os filtros aqui tratados terão resposta impulsiva simétrica ao redor da origem e, por consequência, atraso de grupo nulo e comprimento N ímpar. Por simplificação, de agora em diante consideraremos $T=1$ (intervalo de amostragem normalizado).

$$h(n) = h(-n) \quad \text{e} \quad h(n) = 0 \text{ para } |n| > (N-1)/2 \quad (\text{E } 2.23)$$

Para o Prefiltro, $Q-1$ amostras devem ser descartadas, e aproveitando a simetria de $h(n)$

$$y(n) = h(Qn) * x(Qn) = \sum_{k=-(N-1)/2}^{(N-1)/2} h(k) x(Qn-k)$$

$$y(n) = h(0)x(Qn) + \sum_{k=1}^{(N-1)/2} h(k)(x(Qn-k) + x(Qn+k)) \quad (\text{E } 2.24)$$

Para o Posfiltro interpolador, fazendo considerações semelhantes, temos :

$$y(n) = \sum_{k=-L}^{k=L} x(k) h(n-Qk) \quad (\text{E } 2.25)$$

$$L = \text{trunc} \left[\frac{n - \frac{(N-1)}{2}}{Q} \right] \quad \text{trunc}(x) = \text{parte inteira de } x$$

Pensemos, agora, em um sistema no qual devemos interpolar um sinal que não necessariamente foi previamente filtrado. Neste enfoque as amostras na entrada do interpolador devem pertencer ao sinal interpolado, sendo idênticas às amostras originais do sinal suposta e previamente dizimado, por isso devem ser mantidas, sem serem deturpadas por qualquer processamento ou filtragem. Então $y(nQ) = x(n)$, logo :

$$y(nQ) = x(n) = \sum_{k=-L}^{k=L} x(k) h(nQ-Qk)$$

Sem perda de generalidade, podemos supor $n=0$ e

$$y(0) = x(0) = \sum_{k=-L}^{k=L} x(k) h(-Qk)$$

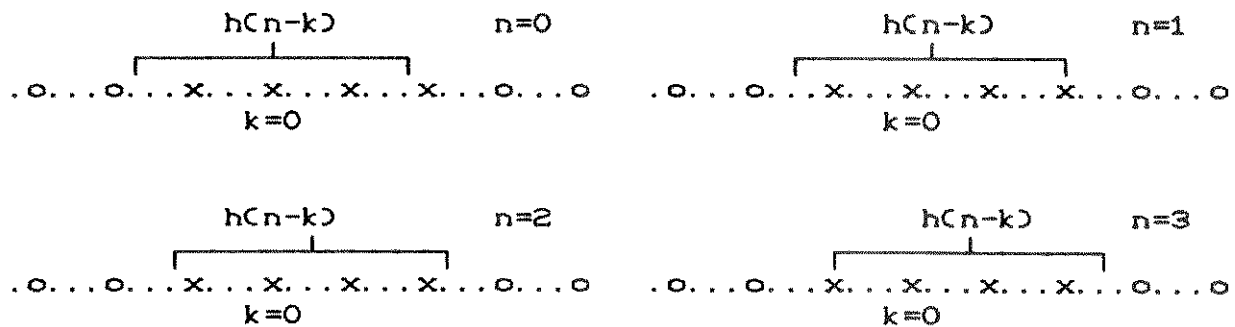
$$\text{Daí : } h(kQ) = 1 \text{ p/}k=0 \text{ e } h(kQ)=0 \text{ p/}k \neq 0 \quad (\text{E 2.26})$$

Por fim, devemos analisar o comprimento da resposta impulsiva em função do número de amostras do processo original envolvidas. Para tal, na Figura 2.3, é mostrada uma sequência equiespaçada de amostras que $h(n)$ deve envolver no processo de filtragem. O centro de simetria do núcleo de convolução ($h(0)$) estará sobre a amostra desconhecida a ser interpolada. Fica claro que a amostra a ser interpolada deve pertencer ao intervalo central entre as amostras originais envolvidas pela convolução. Caso contrário, $h(n) \neq h(-n)$. Para que exista um intervalo central, é óbvio que a interpolação deva envolver um número par de amostras originais. Com isto em mente, para envolver M amostras do processo original, $N=QM-1$. Juntando tal resultado com (E 2.23) e (E 2.26), obtemos um conjunto de condições para realização do interpolador

$$\begin{aligned} h(n) &= h(-n) \\ h(n) &= 0 \text{ para } |n| > (N-1)/2 \\ M &\text{ par} \\ N &= QM-1 \\ h(kQ) &= 1 \text{ p/}k=0 \text{ e } h(kQ)=0 \text{ p/}k \neq 0 \end{aligned}$$

(E 2.27)

O interpolador deve ser uma aproximação de um passa-baixas ideal que pode ser obtida como em quaisquer dos métodos descritos em [1] ou em [4-7]. Porém tais métodos podem levar a resultados que não satisfaçam (E 2.27). Além disto, nestes métodos não possuímos controle sobre os parâmetros de otimização no que diz respeito a melhor ou pior qualidade de interpolação em relação a erros de interpolação. Para burlar tal adversidade podemos lançar mão de alternativas nas quais temos um controle mais justo. Estas incluem aproximações polinomiais (controle no domínio do tempo) e otimizadas (minimização de erros de interpolação).



- o - Amostras do processo original não envolvidas na convolução
- x - Amostras do processo original envolvidas na convolução
- . - Amostras nulas a serem interpoladas

Figura 2.3 - Ilustração da convolução do filtro interpolador para $M=4$ e $Q=4$. ($y(n) = \sum x(k)h(n-k)$)

2.2.1) Aproximações Polinomiais

Este método visa uma aproximação no domínio temporal e se baseia na reconstrução do sinal contínuo. Esta reconstrução deve ser da forma:

$$y_a(t) = \sum_n x(n) \xi_n(t) \tag{E 2.28}$$

$$y(m) = \sum_n x(n) \xi_n(m) = \sum_n x(n) h(m-n) \tag{E 2.29}$$

Sendo ξ uma base de convolução para interpolação contínua, $x(n)$ o sinal discreto superamostrado e $y_a(t)$ uma aproximação de $x_a(t)$. Reamostrando este sinal de (E 2.28), teremos (E 2.29) e ξ será uma base de convolução discreta. Ao invés de recuperarmos o sinal contínuo equivalente ao discreto, como em (E 2.6), podemos fazer aproximações baseadas em interpoladores polinomiais clássicos ou em considerações geométricas sobre o sinal. Estas aproximações se baseiam no fato de que M pontos no plano bidimensional, podem ser ajustados por

um polinômio de grau $M-1$, de forma que tais pontos pertençam à curva descrita por este polinômio. Devem ser levadas em consideração as restrições de (E 2.27), para o desenvolvimento de um filtro FIR que perfaça uma interpolação deste tipo. Desta forma, o sinal a ser interpolado entre as amostras $x(k)$ e $x(k+1)$ deve envolver as amostras $x(k-L+1) \dots x(k-1), x(k), x(k+1) \dots x(k+L)$ do processo original, com $M=2L$.

Podemos considerar alguns tipos mais comuns de interpoladores, como B-Splines [9] e Lagrangeanos. A primeira forma não permite projeto de filtros FIR [15] e apenas o caso trivial de ordem-zero pode ser considerado. Este se refere à interpolação por máxima proximidade, no qual as amostras são aproximadas pela amostra original mais próxima. A interpolação de Lagrange, por sua vez, aproxima M amostras por um polinômio de grau $M-1$.

A interpolação linear se baseia na simples aproximação do sinal contínuo, correspondente à sequência $x(n)$, por retas unindo amostras adjacentes. Ou seja: interpolação linear envolve duas amostras originais consecutivas, apenas, ($M=2$) e as amostras interpoladas recaem sobre uma linha reta que une aquelas. Novamente, este é um caso particular tanto da interpolação Lagrangeana (polinômio de grau 1) como das B-Splines (B-Spline de ordem 1). Desta forma, $h(n) = 1 - |n|/Q$ para $|n| < Q$.

Outro método interessante de interpolação se baseia no trabalho de Robert G. Keys [8] que, por sua vez, se baseou no trabalho de Huo e Andrews [9]. Foi procurada uma base que retivesse algumas propriedades das B-Splines, como continuidade e continuidade de derivadas, mas que utilizasse diretamente as amostras do sinal original e que o sinal contínuo recuperado se ajustasse aos pontos originais. Para tal, a base de interpolação deveria ser nula para instantes de amostragem originais, ou seja, uma versão contínua de (E 2.26).

O sinal $x_a(t)$ é amostrado com período T , gerando a sequência $x(n)$. O sinal $y_a(t)$, função interpolada, que aproximaria o sinal original pode ser expresso pela equação (E 2.28). Reapresentando as bases interpolativas nesta equação, temos :

$$y_a(t) = \sum_n x(n) \xi\left(\frac{t-kT}{T}\right) \quad (\text{E 2.30})$$

Define-se a base cúbica para interpolação [8] como:

$$\xi(s) = \begin{cases} 3/2 |s|^3 - 5/2 |s|^2 + 1 & 0 < |s| < 1 \\ -1/2 |s|^3 + 5/2 |s|^2 - 4 |s| + 2 & 1 < |s| < 2 \\ 0 & |s| \geq 2 ; |s|=1 \\ 1 & s = 0 \end{cases} \quad (\text{E 2.31})$$

As condições de contorno para esta base levam em conta que se possua um conjunto de $N+1$ amostras : $x(n)$; $0 \leq n \leq N$.

$$\begin{aligned} x(-1) &= x(2) - 3 x(1) + 3 x(0) \\ x(N+1) &= x(N-2) - 3 x(N-1) + 3 x(N) \end{aligned} \quad (\text{E 2.32})$$

A função de interpolação por convolução cúbica é única e apresenta convergência com taxa $O(T^3)$. Uma base de convolução, satisfazendo as mesmas condições de continuidade e restrições, terá convergência de terceira ordem se, e somente se, for como (E 2.31). Poder-se-ia, achar funções com convergência de quarta ordem de forma análoga ao aqui apresentado. Uma base envolvendo mais amostras originais e com convergência com $O(T^4)$ é dada por:

$$\xi(s) = \begin{cases} 4/3 |s|^3 - 7/3 |s|^2 + 1 & 0 < |s| < 1 \\ -7/12 |s|^3 + 3 |s|^2 - 59/12 |s| + 15/6 & 1 < |s| < 2 \\ 1/12 |s|^3 + 2/3 |s|^2 + 21/12 |s| - 3/2 & 2 < |s| < 3 \\ 0 & |s| \geq 3 ; |s|=1 ; |s|=2 \\ 1 & s = 0 \end{cases} \quad (E 2.33)$$

Para encontrar o filtro FIR que execute uma interpolação pelo método aqui descrito, façamos T o período da sequência original $x(n)$ e $\tau = 1s$ o período da sequência superamostrada correspondente $x(n)$. Com isto, para um fator de superamostragem inteiro Q, teremos $T=Q\tau=Q$ e :

$$h(n) = \begin{cases} 3/2Q^3 |n|^3 - 5/2Q^2 |n|^2 + 1 & 0 < |n| < Q \\ -1/2Q^3 |n|^3 + 5/2Q^2 |n|^2 - 4/Q |n| + 2 & Q < |n| < 2Q \\ 0 & |n| \geq 2Q ; |n|=Q \\ 1 & n = 0 \end{cases} \quad (E 2.34)$$

valendo as mesmas condições de contorno de (E 2.32), com o cuidado de substituir $x(k)$ por $x(kQ)$. Para esta base de convolução, é interessante salientar que esta está de acordo com (E 2.29), abrangendo 4 amostras do processo dizimado e tendo comprimento $4Q-1$. Por fim, similarmente, da equação 2.33, podemos obter

$$h(n) = \begin{cases} 4/3Q^3 |n|^3 - 7/3Q^2 |n|^2 + 1 & 0 < |n| < Q \\ -7/12Q^3 |n|^3 + 3/Q^2 |n|^2 - 59/12Q |n| + 15/6 & Q < |n| < 2Q \\ 1/12Q^3 |n|^3 + 2/3Q^2 |n|^2 + 21/12Q |n| - 3/2 & 2Q < |n| < 3Q \\ 0 & |n| \geq 3Q ; |n|=Q ; |n|=2Q \\ 1 & n = 0 \end{cases} \quad (E 2.35)$$

2.2.2) Otimização de Interpoladores

Nas seções precedentes, foram tratados Posfiltros, ou filtros interpoladores, no domínio espacial (ou temporal). Os filtros são fixos e independentes do sinal a ser filtrado. O projeto é feito uma vez e aplicado a diferentes sinais que podem ser ou não estacionários. Um outro enfoque pode ser dado, na tentativa de minimizar um critério de erro entre o sinal original e o interpolado após dizimação.

A investigação levada a cabo em [11] se baseou na otimização do Posfiltro para interpolação de um sinal com fator de superamostragem inteiro. O circuito imaginado é o da Figura 2.4, no qual $h(n)$ é projetado de forma a minimizar o erro entre $x(n)$ e $y(n)$, a partir das estatísticas de $x(n)$. Na Figura 2.4, a chave é fechada para todo n tal que $n=\lambda Q$, sendo λ um inteiro. Neste citado trabalho os resultados foram obtidos pela minimização de uma norma das sequências de erro, mas aqui chegaremos a resultados semelhantes minimizando o erro médio quadrático e aplicando noções de filtragem de Wiener.

O filtro deve ter comprimento $N=MQ-1$ de forma a envolver M amostras não nulas de $x(n)$. Além disto podemos apresentar $h(n)$ em sua estrutura polifásica. Podemos até apresentar uma estrutura polifásica alternativa para a Figura 2.4, como na Figura 2.5. Com esta figura e estes dados, convém esclarecer a estrutura polifásica, em relação a frequências de amostragem. As fases do sinal (resposta impulsiva do filtro, por exemplo) eram definidas como $h_i(n)=h(nQ+i)$. Isto implica numa dizimação de $h(n)$ para $h_i(n)$. O último teria comprimento $MQ-1$ contra M amostras dos subfiltros. Por outro lado, poderíamos considerar os subfiltros como : $h_i(n)=h(n)$ para $n=kQ+i$ e $h_i(n)=0$ para $n \neq kQ+i$ (k inteiro). Isto manteria as características dos subfiltros, mas respeitaria uma frequência de amostragem comum para todos os sinais envolvidos no circuito e evitaria atrasos para recompor o sinal original. Valem, então, as seguintes relações :

$$h(n) = \sum_{k=0}^{k=Q} h_i(n) \quad (\text{E 2.36a})$$

$$x(n) = \sum_{k=0}^{k=Q} x_i(n) \quad (\text{E 2.36b})$$

$$y(n) = \sum_{k=0}^{k=Q} y_i(n) \quad (\text{E 2.36c})$$

$$e(n) = \sum_{k=0}^{k=Q} e_i(n) \quad (\text{E 2.36d})$$

Divida o sinal original em suas fases, faça o mesmo com a resposta impulsiva e com a saída do filtro interpolador, como nas equações 2.36. Daí, a partir de Figura 2.4, pode-se encontrar a Figura 2.5, com $x(n)=x_0(n)$. Nesta figura deixe de lado a entrada e o comutador, considerando, apenas, as fases $x_i(n)$. Agora considere isoladamente cada filtro polifásico. Para cada $h_i(n)$ a entrada é a mesma ($x(n)$) e sua saída é comparada com a i -ésima fase de $x(n)$. As fases aqui representadas mantêm a mesma frequência de amostragem original $x(n)$, desta forma possuem $Q-1$ amostra nulas entre duas amostras úteis ou $x_i(n) = x(n)$ para $n=mQ+i$ e zero caso contrário. O mesmo vale para as fases de $h(n)$, $y(n)$ e $e(n)$. Sub-amostrando estas fases e descartando as amostras inúteis, simplificaremos a notação, ou seja: $x_i(m)=x(mQ+i)$. Manteremos, por conveniência, o índice temporal n . A otimização de cada subfiltro pode ser esquematizada na Figura 2.5b. Como em [16], seja R_x a matriz de autocorrelação do sinal de entrada, $d(n)$ o sinal de referência e P_{xd} o vetor de correlação cruzada entre a entrada e $d(n)$. A solução do problema é dada por:

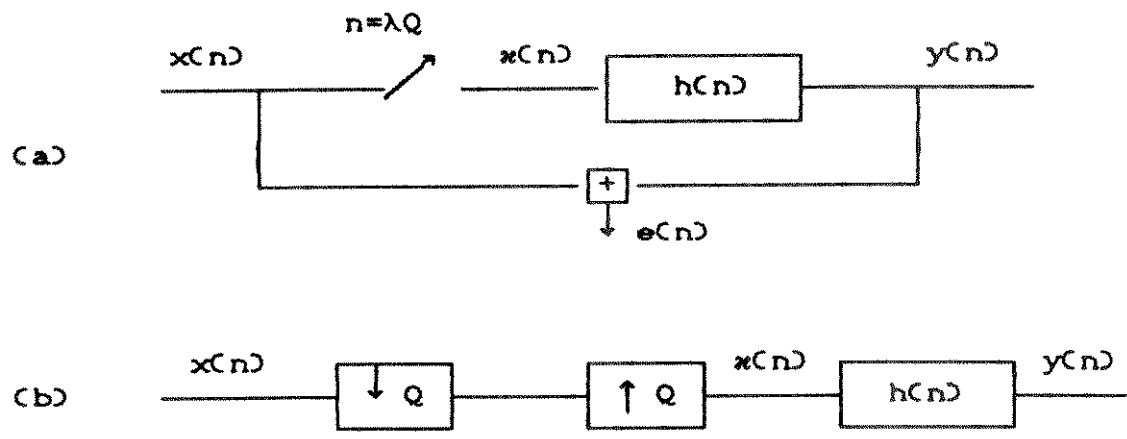


Figura 2.4 - a)Esquema proposto para interpolação sem o Prefiltro b)Diagrama equivalente para o processo de filtragem

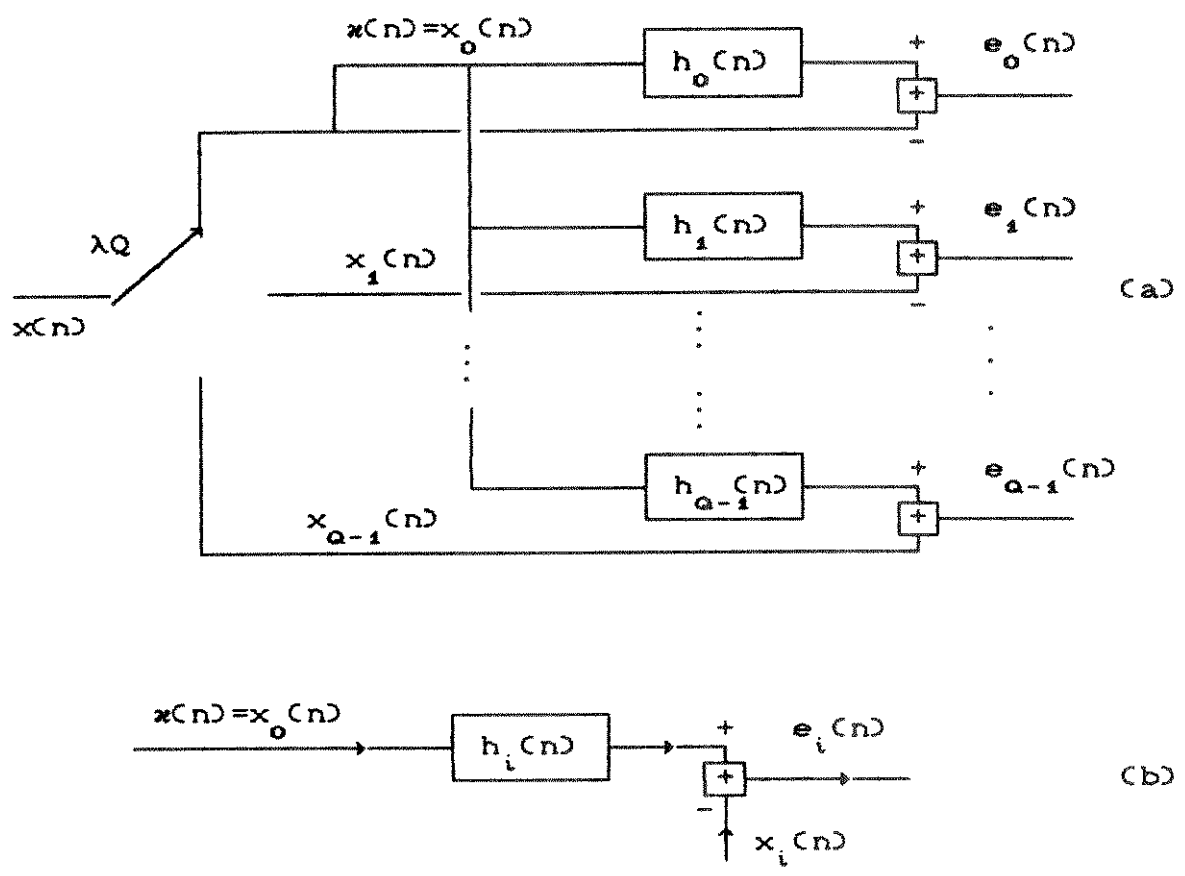


Figura 2.5 - Representação Polifásica para o sistema proposto na Figura 2.4

$$H^T = R_x^{-1} P_{xd} \quad (E 2.37)$$

$$P_{xd}^T = [g(-R) \quad g(-R+1) \quad \dots \quad g(-1) \quad g(0) \quad g(1) \quad \dots \quad g(R-1)] \quad (E 2.38)$$

$$g(k) = E[x(m) d(m-k)] = E[x_0(m) x_1(m-k)] = E[x(mQ) x((m-k)Q+i)] \\ = r(kQ-i) \quad k=-R, \dots, R-1 \quad i=0, 1, \dots, Q-1$$

É encontrada a solução para cada subfiltro h_i . Para h_0 , a solução é o impulso discreto unitário. Para cada uma das $Q-1$ fases restantes, somos levados a um sistema de equações simultâneas. Sendo $N=MQ-1=2RQ-1$ teremos $M=2R$ equações para cada fase. A solução de cada sistema pode ter sua representação matricial alternativamente dada por :

$$A H_i = B_i \quad \text{ou} \quad H_i = A^{-1} B_i \quad i=1, 2, \dots, Q-1 \quad (E 2.39)$$

$$B_i^T = [r(-RQ+i), r(-(R-1)Q+i), \dots, r(-Q+i), r(1), r(Q+i), \dots, \\ r((R-1)Q+i)] \quad (E 2.40)$$

$$H_i^T = [h(-RQ+i), h(-(R-1)Q+i), \dots, h(-Q+i), h(1), h(Q+i), \dots, \\ h((R-1)Q+i)] \quad (E 2.41)$$

$$A = \begin{bmatrix} r(0) & r(Q) & r(2Q) & \dots & r((M-1)Q) \\ r(Q) & r(0) & r(Q) & & \\ r(2Q) & r(Q) & r(0) & & \\ \vdots & & & \ddots & \\ r((M-1)Q) & \dots & & & r(0) \end{bmatrix} \quad (E 2.42)$$

$$H_i^T = [h_i(-R), \dots, h_i(-1), h_i(0), h_i(1), \dots, h_i((R-1))] \quad (E 2.43)$$

Como já dito, h_0 é o impulso unitário e

$$H_0^T = [h_0(-R), \dots, h_0(-1), h_0(0), h_0(1), \dots, h_0((R-1))] = \\ = [0, \dots, 0, 1, 0, \dots, 0] \quad (E 2.44)$$

Outro fato interessante é o de que A é uma matriz Toeplitz e pode ser mais facilmente invertida [14]. Além disto A é a mesma para todas as $Q-1$ fases (independe de i). Com a resolução de (E 2.39) encontramos o subfiltro ótimo para dado sinal estacionário $x(n)$. A partir destes subfiltros monta-se $h(n)$, como em (E 2.36a). Porém como $r(n)=r(-n)$, temos que $B_{Q-i}=B_i$. Daí, sendo A constante para todos subfiltros :

$$H_{Q-i} = A^{-1}B_{Q-i} = A^{-1}B_i = H_i$$

Este resultado nos indica que precisamos inverter uma matriz $M \times M$ e fazer $(Q-1)/2$ multiplicações matriciais, já que necessitamos apenas conhecer metade dos subfiltros ótimos. Além disto, podemos ratificar a linearidade de fase do filtro ótimo encontrado, pois $h_i(n)=h_{Q-i}(n)=h_i(-n)$ (Vide Eq. 2.41).

2.2.3) Otimização Conjunta do Posfiltro e do Prefiltro

Os resultados desta seção são encontrados em [12] e são uma generalização de outros trabalhos na área. É suposto um meio ruidoso com um Prefiltro sempre ativo e o erro é filtrado por uma função de peso de observação. O sistema proposto foi o da Figura 2.6, sendo $F(\omega)$ e $G(\omega)$ as transformadas de Fourier das respostas impulsivas do Prefiltro e do Posfiltro ($f(n)$ e $g(n)$) respectivamente.

É adicionado ruído à leitura dos dados originais e à saída do Prefiltro (canal) e estes ruídos são representados por $u(n)$ e $v(n)$. O erro de interpolação é filtrado por uma janela que deve representar a sensibilidade do observador a tal erro. Os sistemas das figuras 2.6a e 2.6b são equivalentes com :

$$\delta_a(n) = Q \sum_m \delta(n-mQ) = \sum_{m=0}^{Q-1} e^{j(2\pi mn/Q)} \quad (E 2.45)$$

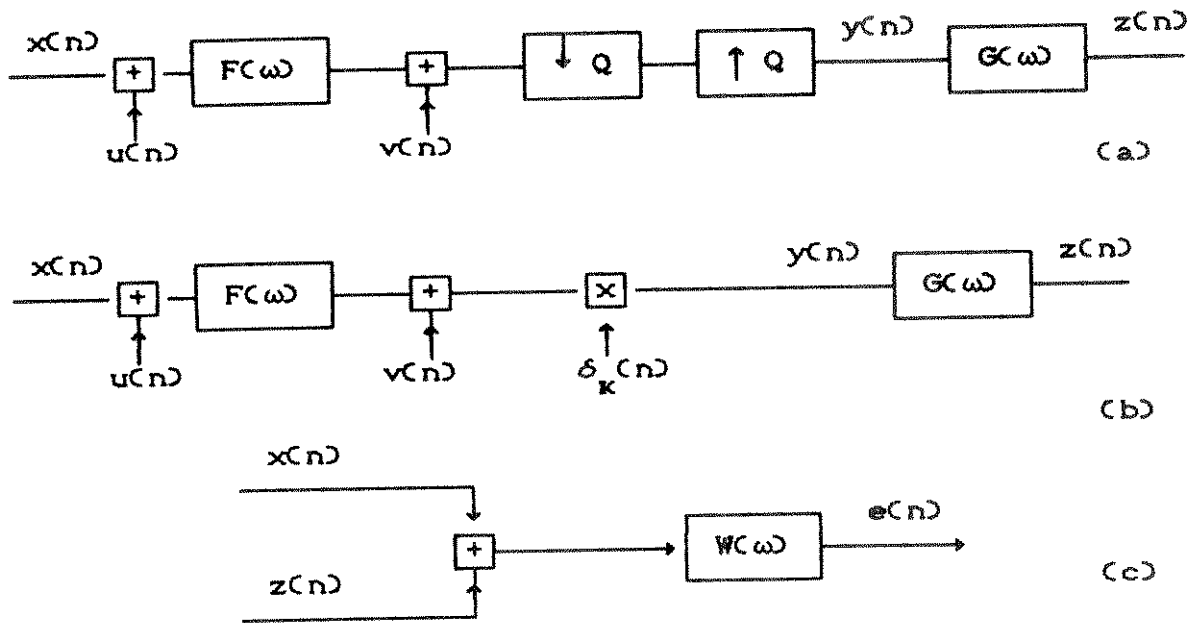


Figura 2.6 - (a) Diagrama básico para interpolador em meio ruidoso com prefiltragem. (b) Representação Equivalente (c) Obtenção do erro.

Neste caso nem todas as restrições de (E 2.27) são exigidas, deixando a cargo do algoritmo, minimizar ao máximo o erro. Porém foi respeitada a simetria de $f(n)$ e $g(n)$. Para simplificar, adotaremos, nesta seção, algumas notações em relação a um sinal qualquer $s(n)$ do circuito: $S(\omega)$ será sua transformada de Fourier, $R_s(n)$ sua sequência de autocorrelação e $\Phi_s(\omega)$ será sua densidade espectral de potência.

Note que a multiplicação pelo trem de impulsos introduz não estacionaridade ao sinal $y(n)$. Como tratado em [13], estes processos podem ser encarados como cicloestacionários e o erro a ser minimizado será computado em um período.

$$\xi = \frac{1}{Q} \sum_{n=0}^{Q-1} E[e^2(n)] = \frac{1}{2\pi} \int_{-\pi}^{\pi} \Phi_e(\omega) d\omega \quad (\text{E 2.46})$$

O erro é uma função quártica dos coeficientes de $f(n)$ e $g(n)$. Pontos com derivada nula não necessariamente serão mínimos globais, por isso não é possível encontrar uma solução fechada. Se fixarmos um dos filtros, as equações serão quadráticas com soluções facilmente encontráveis. Para $f(n)$ fixo, o posfiltro ótimo será dado pela solução do seguinte sistema Toeplitz :

$$AC(\omega) = |WC(\omega)|^2 \Phi_y(\omega)$$

$$BC(\omega) = |WC(\omega)|^2 F(\omega) [\Phi_x(\omega) + \Phi_u(\omega)]$$

$$\sum_{m=-M}^{m=M} g(m) a(1-m) = b(1) \quad 1 = -M, -M+1, \dots, 0, \dots, M \quad (E 2.47)$$

Para a solução do Prefiltro ótimo, fixamos $g(n)$ e otimizamos $f(n)$ com a restrição de que a potência de saída do Prefiltro permaneça abaixo de determinado valor. Tal valor pode ser tomado, sem perda de generalidade, como a unidade. Desta forma, sendo $\omega_Q = 2\pi/Q$, temos :

$$CC(\omega) = [\Phi_x(\omega) + \Phi_u(\omega)] \sum_{m=0}^{Q-1} |WC(\omega + m\omega_Q)|^2 G^2(\omega + m\omega_Q)$$

$$DC(\omega) = |WC(\omega)|^2 G(\omega) \Phi_x(\omega)$$

$$P = \sum_{m=-L}^L \sum_{l=-L}^L f(1)f(m) [R_x(1-m) + R_u(1-m)] \leq 1 \quad (E 2.48)$$

$$\sum_{m=-L}^{m=L} f(m) [c(1-m) + \lambda [R_x(1-m) + R_u(1-m)]] = d(1) \quad -L \leq 1 \leq L \quad (E 2.49)$$

Este sistema apresenta uma forma Toeplitz simétrica, porém não conhecemos λ a priori. Escolhe-se $\lambda=0$, inicialmente, e resolve-se (E 2.49). Se $P < 1$, o filtro ótimo é o obtido. Caso contrário, resolve-se (E 2.49) repetidamente com adaptação de λ ,

de acordo com [12]. Esta repetição é levada a cabo até encontrarmos $P \approx 1$. A solução garantidamente converge para um Prefiltro globalmente ótimo.

Para otimização conjunta do Posfiltro e do Prefiltro, nenhuma solução fechada foi encontrada, por isto um método iterativo é sugerido. O método pertence à classe de algoritmos conhecidos como *coordinate descent*. O algoritmo para tal otimização conjunta é :

Inicialização - Fazer $i=0$ e $f_0(n) = \alpha \delta(n)$

Recorrência

- 1) Para $f(n)=f_i(n)$, encontrar $g(n)$ e fazer $g_i(n)=g(n)$
- 2) Para $g(n)=g_i(n)$, encontrar $f(n)$ e fazer $f_{i+1}(n)=f(n)$
- 3) Calcular $\Delta = \max |f_{i+1}(n)-f_i(n)| \quad -L \leq n \leq L$

Parada - Para Δ menor que algum valor especificado.

Deve-se salientar que tal método não converge garantidamente para o valor ótimo, irá depender em muito de W , da escolha inicial de λ e do sinal de entrada. O algoritmo pode apontar para soluções em pontos de sela ou mínimos locais, mas se inicialmente $\lambda=0$, na maioria das vezes $f(n)$ e $g(n)$ convergirão para seus valores ótimos.

2.3) INTERPOLAÇÃO NO CENTRO DO INTERVALO DE AMOSTRAGEM

Esta seção se dedica a particularizar os resultados deste capítulo para o caso $Q=2$ que possui grande importância na codificação em Pirâmide, que será tratada em capítulos posteriores. Iremos comparar o método polinomial da convolução cúbica, os métodos otimizados (embora dependentes do sinal) e o que utiliza blocos segmentados.

Para um interpolador atender às condições impostas na Seção 2.2 e supondo uma interpolação no centro do intervalo de amostragem, resta-nos calcular apenas um filtro polifásico. Isto devido ao fato que $h(mQ)=0$ para m inteiro $\neq 0$ e que $h(0)=1$, ou

seja $h_0(m) = \delta(m)$ e basta-nos calcular $h_1(n)$. Adotando esta notação, daqui em diante iremos utilizar a notação causal para $h_1(n)$, tal que seja diferente de zero apenas para $n \geq 0$ e que cause um atraso de R amostras e possua $M=2R$ coeficientes no total. Este atraso deve ser compensado em $x_1(n)$ e, como as fases por definição possuem um atraso relativo de meia amostra (uma amostra de $x(n)$), este atraso também deve ser de R amostras em $x_1(n)$. Além disto, por só necessitarmos encontrar um filtro, deixaremos de lado o índice de $h_1(n)$ e nos referiremos a este como puramente $h(n)$. A Figura 2.15 ilustra este processo de filtragem, já admitindo a separação das fases do sinal $x(n)$. O erro de interpolação será representado pelo sinal $e(n)$ e os critérios de comparação se basearão na sua potência (variância).

Substituindo $Q=2$ nas equações 2.40 e 2.41, obtemos as bases de convolução cúbica para o caso em questão

$$\begin{aligned} h(0) &= h(3) = -1/16 ; \\ h(1) &= h(2) = 9/16 \end{aligned} \quad (M=4, R=2, N=7) \quad (E 2.50)$$

$$\begin{aligned} h(0) &= h(5) = 56/96 \\ h(1) &= h(4) = -9/96 \\ h(2) &= h(3) = 1/96 \end{aligned} \quad (M=6, R=3, N=11) \quad (E 2.51)$$

A interpolação linear, por sua vez, é dada por

$$h(0)=h(1)=1/2 \text{ e } h(n)=0 \text{ para } n>1 \quad (E 2.52)$$

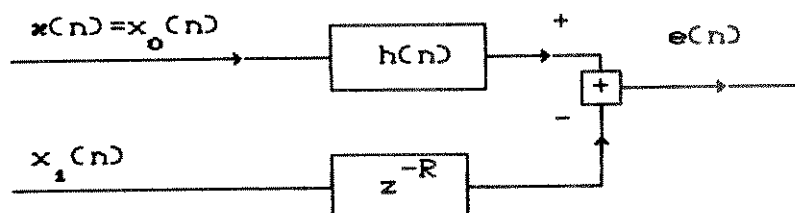


Figura 2.7 - Interpolação no centro do intervalo de amostragem através do uso de um filtro polifásico.

Para interpolação otimizada ([10] e Seção 2.2.2), iremos supor o sinal de entrada como sendo um processo de Gauss-Markov, ou seja, um processo autoregressivo de ordem 1 AR(1) com função de autocorrelação dada por $r(n) = r(0)\rho^n$. Como na equação 2.39, ambos lados contém matriz ou vetor com todos elementos sendo amostras de $r(n)$, podemos cancelar os termos $r(0)$ e obter

$$\begin{pmatrix} 1 & \rho^2 & \rho^4 & \rho^6 & \dots & \rho^{2(M-1)} \\ \rho^2 & 1 & \rho^2 & \rho^4 & & \\ \rho^4 & \rho^2 & 1 & \rho^2 & & \\ \vdots & & & \ddots & & \\ \rho^{2(M-1)} & & & & & 1 \end{pmatrix} \begin{pmatrix} h(0) \\ h(1) \\ h(2) \\ \vdots \\ h(M-1) \end{pmatrix} = \begin{pmatrix} \rho^{M-1} \\ \rho^1 \\ \rho^1 \\ \vdots \\ \rho^{M-1} \end{pmatrix}$$

Para $\rho=0.95$, qualquer que seja M , teremos $h(N/2+1) = h(N/2) \cong 0,5$ e todos os outros coeficientes se anulam. Ou seja, para um processo deste tipo a melhor interpolação seria tomar a média das amostras adjacentes, como em uma interpolação linear. Este critério se revela, aparentemente, como insuficiente para imagens. O principal problema é que, ao modelarmos a imagem como um processo AR(1), não levamos em conta as bordas que são zonas de transições abruptas. Em média, o interpolador linear terá bom comportamento, mas terá regiões de grande precisão de interpolação e outras de baixo rendimento. Isto partindo do princípio que imagens podem ser modeladas da forma descrita, mas considerando que este modelamento é simplista e toma a média do comportamento de uma imagem típica. Em realidade uma imagem é formada por diversas regiões com particulares comportamentos estatísticos. Para verificar o bom comportamento médio de interpoladores lineares, algumas simulações foram realizadas, mas, antes de apresentar resultados, iremos explicar um último método de interpolação que será utilizado em comparações.

Também poderíamos interpolar um sinal se estimássemos seu espectro por uma transformada rápida. Poderia ser, por exemplo, a FFT [1], obtendo N pontos do espectro de um sinal de N amostras. Preencheríamos estes N pontos com N amostras nulas, obtendo $2N$ pontos do mesmo espectro, já que a DFT assume sinais com repetições periódicas [1]. Desta forma, a transformada inversa de $2N$ pontos irá fornecer $2N$ amostras de um sinal cuja DFT é dada por tais pontos. Com isto, podemos fazer uma interpolação eficiente, que independe de projetos de interpoladores, dependendo apenas do tamanho do bloco de amostras e da transformada escolhida, que poderia ser mesmo a transformada discreta de cossenos (DCT) [14]. Dividindo o sinal em blocos, podemos fazer uma estimação local do espectro e é de se esperar que a performance desta interpolação melhore com o aumento do tamanho dos blocos.

Para comparação dos métodos de interpolação, dizimou-se a imagem "Girl", sem prefiltragem, e retendo uma a cada par de amostras do sinal unidimensional formado por suas linhas. Partindo deste sinal dizimado, interpolou-se as amostras faltantes através dos métodos propostos neste capítulo. À cada resultado avaliou-se a relação sinal/ruído (SNR) entre a imagem original e a interpolada, bem como a porcentagem de blocos de 16×16 pixels que possuísse SNR acima de determinado limiar (por exemplo 35dB). Utilizou-se os métodos de projeto de filtros FIR através da janela de Hanning (com filtros de comprimento 7 e 11), da aproximação linear, da convolução cúbica e através do método utilizando a DCT (em lugar da DFT) com diferentes tamanhos de blocos. A Tabela 2.1 mostra resultados obtidos com esta imagem e os métodos citados.

Nesta tabela, os comprimentos se referem ao tamanho dos blocos e, para os filtros FIR, o primeiro número é o comprimento total e o segundo indica o comprimento da fase $h_1(n)$ do filtro.

Tabela 2.1 - Comparação do desempenho de interpoladores

MÉTODO	COMPRIMENTO	SNR (dB)	BLOCOS >35dB (%)
HANNING	7 - 4	25.7	0.0
HANNING	11 - 6	28.4	44.9
LINEAR	3 - 2	27.7	43.9
CONV. CÚBICA	7 - 4	28.4	46.6
BLOCOS(DCT)	4	23.8	22.6
BLOCOS(DCT)	8	24.0	21.7
BLOCOS(DCT)	16	24.1	22.6
BLOCOS(DCT)	32	24.1	21.4

Os resultados indicam que o melhor comportamento fica por conta do filtro baseado na convolução cúbica e que interpolação com blocos não sobrepostos não leva a resultados superiores. O método comum de projeto através da janela de Hanning, necessita de ordens superiores para igualar-se ao desempenho da convolução cúbica e um filtro polifásico de 6 elementos daquele projeto tem desempenho equivalente a um deste que possui apenas 4 elementos. Além do mais, vê-se que em um comportamento global, a interpolação linear é a mais simples e se aproxima bastante dos melhores resultados da tabela acima. Isto não é surpreendente se levarmos em conta que, como discutido, tal interpolação é aproximadamente ótima para o modelamento mais comum de imagens.

2.4) REFERÊNCIAS

- [1] A.V.Oppenheim, R.W.Schafer, *Digital Signal Processing*, Englewoods Cliffs, NJ, Prentice-Hall Inc. 1975.
 - [2] R.E. Crochiere, L.R. Rabiner, *Multirate Digital Signal Processing*, Englewoods Cliffs, NJ, Prentice-Hall Inc. 1983.
 - [3] R. W. Schafer, L.R. Rabiner, *A Digital Signal Processing Approach to Interpolation*, Proc. IEEE, vol 61, June 1973.
 - [4] T.W.Parks, J.H.McClellan, *Chebyshev Aproximation for Nonrecursive Digital Filters with Linear Phase*, IEEE Trans. on Circuit Theory, vol CT-19, No.2, March 1972.
 - [5] L.R.Rabiner, *Design of FIR Digital Filters with Linear Programming Techniques*, Bell Sys. Tech J., vol 51, No.6, July 1972.
 - [6] E. Hoffsteter, A.V.Oppenheim, *A New Technique for the Design of Nonrecursive Digital Filters*, Proc. 5th. Annual Princeton Conf. on Inf. Sci. Syst., pp 64-72, 1971.
 - [7] J.H.McClellan, T.W.Parks, L.R.Rabiner, *A Computer Program for Designing Optimum FIR Linear Phase Digital Filters*, IEEE Trans. on Audio and Electroacustics, vol AU-21, No.6, December 1973.
 - [8] R.G. Keys, *Cubic Convolution Interpolation for Digital Image Processing*, IEEE Trans. on ASSP, vol ASSP-29, December 1981.
 - [9] H.S.Huo, H.C.Andrews, *Cubic Splines for Image Interpolation and Digital Filtering*, IEEE Trans. on ASSP, vol ASSP-26, No.6, December 1978.
 - [10] A.D.Polydoros, E.N.Protonodarius, *Digital Interpolation of Stochastic Signals*, IEEE Trans. on Circuits and Systems, vol CAS, No.11, November 1979.
-

-
- [11] G.Oetken, T.W.Parks, H.W.Schuessler, *New Results in the Design of Digital Interpolators*, IEEE Trans on ASSP, vol ASSP-23, No.3, June 1975.
- [12] H.S.Malvar,D.H.Stealin, *Optimal FIR Pre- and Postfilters for Decimation and Interpolation of Random Signals*, IEEE Trans. on Comm., vol 36, No.1, January 1988.
- [13] W.A.Gardner, L.E.Franks, *Characterization of Cyclostationary Random Signal Processes*, IEEE Trans. on Inform. Theory, vol IT-21, January 1975.
- [14] R.E.Blahut, *Fast Algorithms for Digital Signal Processing*, Reading MA , Addison-Wesley, 1985.
- [15] R.L.Queiroz,J.B.Yabu-uti, *Técnicas de Projeto de Filtros FIR para Dizimação e Interpolação de Sinais Discretos*, RT-178, Contrato 208/87, UNICAMP/TELEBRÁS, Setembro 1989.
- [16] S.Haykin, *Adaptive Filter Theory*, Englewood Clifs, NJ, Prentice-Hall, 1986.
-

CAPÍTULO 3

CODIFICAÇÃO PIRAMIDAL

Neste capítulo iremos abordar um método para transmissão progressiva, bem como compressão de dados, conhecido como Codificação em Pirâmide ou Piramidal. Neste método é possível apresentar progressivamente a imagem, transmitindo primeiramente a informação principal para depois adicionar detalhes de alta frequência, como refinamentos adicionais. Aqui trataremos a codificação piramidal acoplada a processamento sequencial com uma abordagem espacial do processamento envolvido, mas evitaremos interpretações espectrais mais profundas.

A idéia de transmissão progressiva por refinamentos, como vamos apresentar, remonta de trabalhos como, por exemplo, [1], mas em [2] Burt e Adelson estabeleceram formulações significativas, batizando-as de Pirâmides Laplacianas e Gaussianas. Aqui apresentaremos princípios gerais, dos quais [2] é um caso particular. Em seguida apresentaremos os princípios gerais destacando as Pirâmides Laplacianas, Quadtree de Wang e pirâmides reduzidas, bem como alguns métodos recentemente propostos.

3.1) PRINCÍPIOS DE CONSTRUÇÃO E DECODIFICAÇÃO

É suposto que a imagem considerada para transmissão é formada por uma matriz finita de pixels extraída de uma matriz infinita (cena). A matriz é suposta quadrada contendo $N \times N$ elementos, com N suposto grande o suficiente para que as bordas não sejam muito influentes em um processo de filtragem, de forma que estas perturbações não se propaguem por sobre partes significativas. Com isto em mente, pensemos em filtrar a imagem original de duas formas : uni e bidimensionalmente. Para o primeiro caso seja $x(n)$ a imagem original descrita por suas linhas horizontais e, para o segundo, seja $x(n)$ a matriz de pixels representando a imagem.

Se filtrarmos $x(n)$ com um filtro passa-baixas rejeitando mais do que a metade da banda original, este novo sinal filtrado poderia ser representado com metade da frequência de amostragem. Se, novamente, filtrarmos este sinal com outro filtro digital de mesmas características, obteríamos uma terceira imagem com menor resolução ainda. Prosseguindo o processo e insistindo no método iremos obter várias imagens com resoluções decrescentes. Estas imagens, se as imaginarmos colocadas umas sobre as outras como na Figura 3.1, formariam uma pirâmide que aqui trataremos de Pirâmide de Imagens (PI) cujo último nível poderia ser um único pixel correspondendo ao valor médio da imagem.

Se, por outro lado, a cada vez que obtivéssemos uma imagem de menor resolução interpolássemos esta imagem obtendo uma aproximação *borrada* daquela imagem. Por exemplo, teríamos $x(n)$ e uma versão *borrada* (filtrada, sub-amostrada e interpolada) de $x(n)$. Uma forma interessante de codificação seria aquela na qual tomássemos as diferenças entre estas imagens de forma que soubéssemos o erro introduzido pela filtragem. Com isto poderíamos recuperar totalmente a imagem original a partir destes erros. Matematicamente, para o caso unidimensional em primeiro lugar, temos

$$x(n) = x_0(n) = \text{imagem original}$$

$$x_0^F(n) = x_0(n) * h(n)$$

$$x_1(n) = x_0^F(Mn)$$

$$x_1(n) \xrightarrow{\text{interpolação}} x'_0(n) \approx x_0^F(n)$$

sendo $h(n)$ a resposta impulsiva do filtro passa-baixas e M o fator pelo qual a resolução é diminuída em cada nível. Geralmente M é tomado como 2. Repetindo o processo K vezes, teríamos, em cada passo,

$$x_{k-1}^F(Mn) = x_k(n) \Rightarrow x'_{k-1}(n) \quad k=1, \dots, K \quad (\text{E 3.1})$$

com $x_k(n)$, no caso unidimensional, possuindo N^2/M^k amostras.

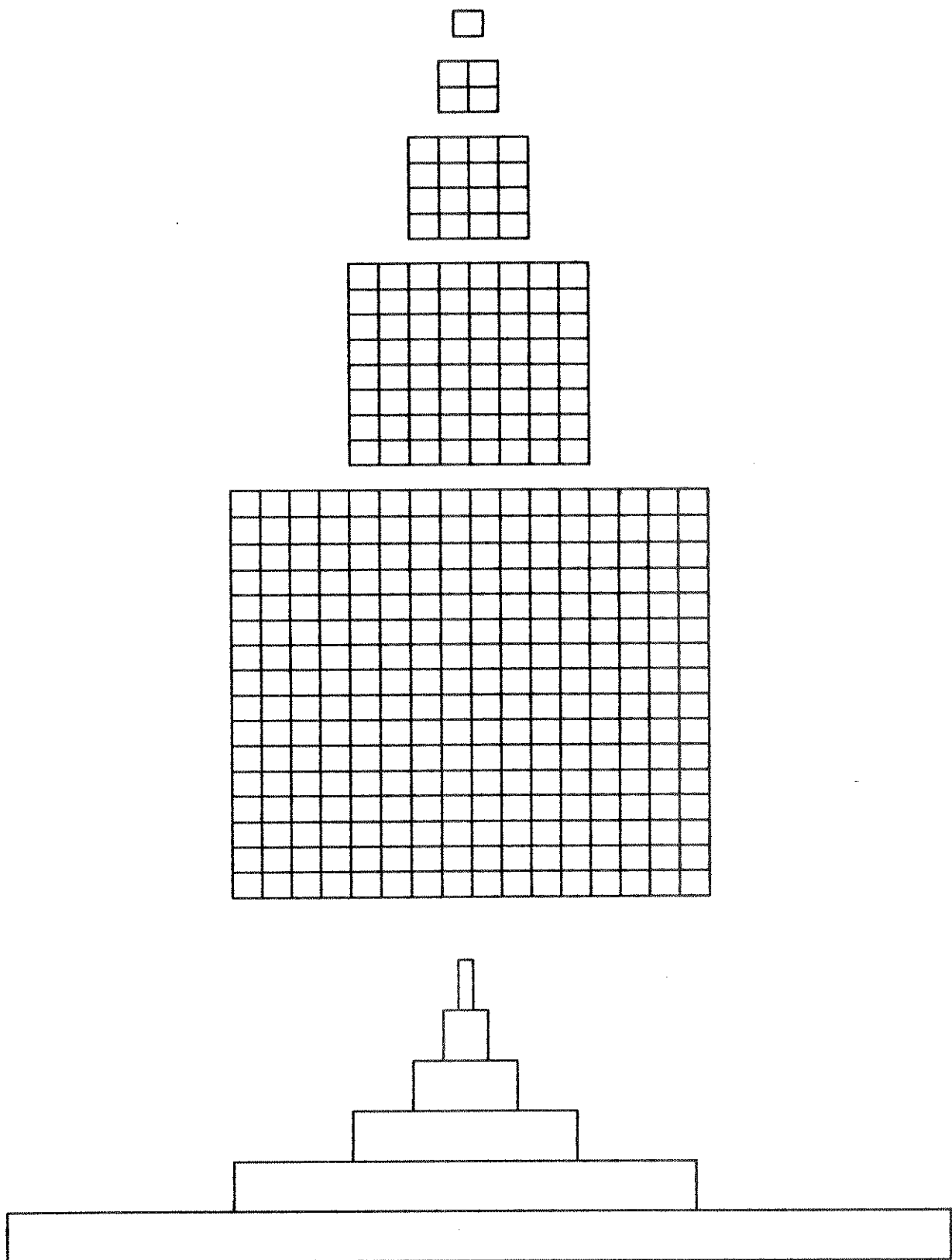


Figura 1 - Pirâmide de Imagens (Ilustração)

À sequência de imagens x_k denominaremos de Pirâmide de Imagens (PI), mas é encontrada na literatura também como Pirâmide Gaussiana ou de Médias. A denominação Pirâmide é fruto da abstração de colocar as imagens superpostas umas às outras como na Figura 3.1. Cada imagem que compõe esta pirâmide será aqui chamada de nível, sendo o nível superior composto pela imagem de menor resolução ($x_k(n)$). Poderíamos, então, codificar e transmitir a pirâmide de cima para baixo de forma que a decodificação seja feita obtendo imagens que seriam progressivamente refinadas. Uma melhor forma de representar estes níveis poderia ser alcançada se codificarmos o erro entre determinado nível e sua aproximação feita a partir do nível próximo superior.

$$L_k(n) = x_k(n) - x'_k(n) \quad k=0, \dots, K-1 \quad (E 3.2)$$

À pirâmide formada, similarmente, pelos níveis $L_k(n)$ denominaremos por Pirâmide de Diferenças (PD), podendo ser encontrada na literatura como Pirâmide Laplaciana. Se fizermos $x_k(n) = L_k(n)$, é fácil ver que apenas necessitamos dos $K+1$ níveis da PD para representar a imagem original descrita por $x(n)$, já que

$$x_k(n) = x'_k(n) + L_k(n) \quad (E 3.3a)$$

$$x_{k+1}(n) \xrightarrow{\text{interpolação}} x_k(n) \quad (E 3.3b)$$

O que realmente é codificado é a PD, mas, na decodificação, é a PI que é apresentada no terminal de vídeo. A razão principal disto advém da redução da entropia das amostras de $L_k(n)$, quando comparada às amostras de $x_k(n)$. Além disto quando o receptor houver formado na tela a imagem $x'_k(n)$ só será preciso somar uma camada de refinamento à imagem apresentada.

Para generalizarmos os resultados para o caso multidimensional poderíamos proceder de forma semelhante, mas tomando sinais e operações também multidimensionais. Daí, considerando notações familiares a [4], teremos

$$n = [n_1, n_2, n_3, \dots, n_R]^T$$

$$x_0(n) = x(n) = \text{imagem R-dimensional original}$$

$$x_k^F(n) = x_k(n) * h(n) \quad (E\ 3.4a)$$

$$x_{k+1}(n) = x_k^F(M\ n) \quad (E\ 3.4b)$$

$$x_{k+1}(n) \xrightarrow{\text{interpolação}} x'_k(n) \quad (E\ 3.5)$$

$$L_k(n) = x_k(n) - x'_k(n) ; L_k(n) = x_k(n) \quad (E\ 3.6)$$

$$k = 0, 1, \dots, K-1$$

As equações 3.1-3.6 foram formuladas genericamente, mas cabe, agora, ressaltar que os fatores de redução de uma imagem para outra são geralmente tomados como 2 em cada direção e a frequência de corte dos filtros seria $f_c = \pi/2T_s$. Este número foi escolhido por comodidade e facilidade para implementação e também porque é o mais adotado na literatura. Com isto em mente e supondo que aplicaremos o método a imagens estáticas, teremos

$$R=2 \text{ e } n = [n_1, n_2]^T$$

$$M = \begin{bmatrix} M_1 & 0 \\ 0 & M_2 \end{bmatrix} = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}$$

3.2) PREFILTROS E POSFILTROS

Como visto no Capítulo 2, o prefiltro é o filtro que limita o sinal antes da dizimação (evita *aliasing*) e o posfiltro é o filtro interpolador que atenua as imagens que aparecem no espectro do sinal após inserção de amostras nulas. Em realidade $h(n)$ em E3.4a é o prefiltro e o posfiltro será utilizado na interpolação indicada em E3.5.

Em [2], o núcleo de convolução $h'(n)$ do prefiltro faz o limite das convoluções $h'(n) * h'(n) * \dots * h'(n)$ se aproximar de uma distribuição gaussiana. Escolheu-se o núcleo de convolução contido em uma região de suporte de tamanho 5×5 , gerado a partir de $h'(n)$ e dado por

$$h(n_1, n_2) = h'(n_1) h'(n_2) \quad (E 3.7)$$

$$h'(0) = \alpha ; h'(\pm 1) = 1/4 ; h(\pm 2) = 1/4 - \alpha/2 ; h'(n) = 0 \quad |n| > 2 \quad (E 3.8)$$

Variando-se o parâmetro α pode-se variar a resposta impulsiva do filtro e, portanto, sua resposta em frequência. O nome de Pirâmide Gaussiana segue da escolha deste filtro e, como comentado, este é um caso particular. Se a imagem for modelada como um sinal com distribuição gaussiana, o resultado da filtragem será gaussiano e a imagem diferença entre imagem original e filtrada terá distribuição aproximadamente laplaciana. Por isto foi escolhido o nome de Pirâmide Laplaciana para a PD, neste caso que serviu como base para a maioria dos trabalhos.

Para a interpolação, o filtro é o mesmo, tomando o cuidado de se incorporar um ganho de 4.

$$h_{int}(n_1, n_2) = 2h'(n_1) 2h'(n_2) = 4 h(n_1, n_2)$$

Em [3] o processo é bastante distinto, como ilustrado na Figura 3.2, mas não é necessário uma interpretação das operações como filtragem. Como o fator de compressão é 2 em cada direção, cada quatro pixels formam um novo pixel no nível superior da PI. A diferença deste método para um mais geral é que cada conjunto de quatro pixels (formando uma quádrupla) possui uma relação única com o pixel formado, por exemplo tomando a média destes. Para melhor explicar, sejam os seguintes pixels de um nível k com $N^2/2^{2k}$ elementos:

$$\begin{aligned} a &= x_k(2m, 2n) ; b = x_k(2m+1, 2n) ; c = x_k(2m, 2n+1) ; \\ d &= x_k(2m+1, 2n+1) \quad m, n \in \{0, \dots, N/2^{k+1}-1\} \end{aligned} \quad (E 3.9)$$

E o pixel formado no nível acima da pirâmide será

$$\frac{a + b + c + d}{4} = x_{k+1}(m, n) \quad (E 3.10)$$

e o nível correspondente da PD será

$$L_k(i,j) = x_k(i,j) - x_{k+1}(m,n) \text{ para } i \in [2m, 2m+1] \text{ (E 3.11)} \\ j \in [2n, 2n+1]$$

Para este caso a PD é conhecida como Pirâmide de Diferenças (de onde absorvi a nomenclatura) e a PI como Pirâmide de Médias. Para uma interpretação do processamento com convolução com núcleos, podemos definir o filtro como tendo a resposta impulsiva assimétrica e definida por (Figura 3.2):

$$h(0,0)=h(1,0)=h(0,1)=h(1,1)=1/4 \quad \text{(E 3.12)}$$

$$h(n_1, n_2) = 0 \text{ p/ } n_1, n_2 \notin [0,1]$$

$$h_{int}(n_1, n_2) = 4 h(n_1, n_2) \quad \text{(E 3.13)}$$

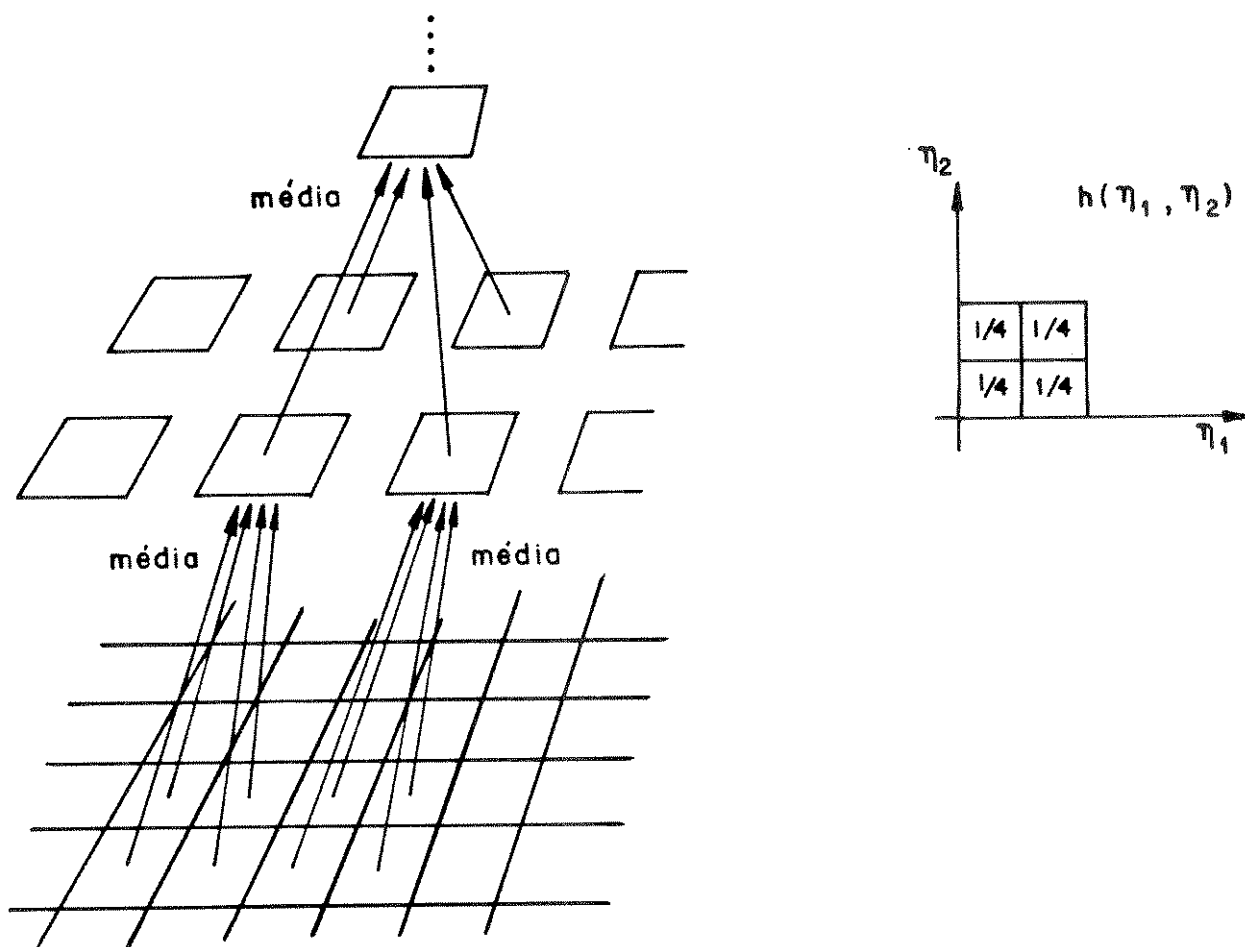


Figura 3.2 - Pirâmide de médias e seu núcleo de convolução.

A principal vantagem deste método é sua simplicidade. Note que nenhuma multiplicação é envolvida, apenas somas e subtrações, mas, por outro lado, sua eficiência poderia ser aumentada se estendêssemos um pouco mais a resposta impulsiva do filtro.

3.3) PIRÂMIDES REDUZIDAS

Embora adequada à transmissão progressiva, a codificação em pirâmides requer a codificação de mais do que N^2 amostras para representar N^2 pixels. Identificando como *nodos* as amostras contidas nos níveis da PD e chamando de N_n o montante total de *nodos* por imagem, poderíamos dizer que N_n/N^2 seria um fator de *overhead* que deve ser levado em conta quando calcularmos a entropia média dos sinais a serem codificados. Por outra ótica, podemos melhor explicar chamando a atenção para que a entropia de codificação será dada em bits/amostra ou bits/nodo e não em bits/pixel. Para calcular a entropia equivalente em bits/pixel N_n deve ser computado. No caso unidimensional, com dizimação por fator 2, temos o total de *nodos* dado por

$$N_n = N^2(1 + 1/2 + \dots + 1/2^K) = N^2 \frac{1 - 2^{-K-1}}{1 - 1/2} = N^2 2 (1 - 2^{-(K+1)})$$

Se K , o número de níveis, for suficientemente grande,

$$N_n \approx 2N^2 \Rightarrow N_n/N^2 = 2 \text{ ou } N_n - N^2 = N^2 \Rightarrow 100\% \text{ de overhead.}$$

Para o caso bidimensional, com o mesmo fator de dizimação em cada dimensão, temos :

$$N_n = N^2 \sum_{n=0}^K (1/4)^n = N^2 \frac{4}{3} (1 - 2^{-2(K+1)}) \approx \frac{4}{3} N^2$$

$$\frac{N_n - N^2}{N^2} = \frac{1}{3} \Rightarrow 33\% \text{ de overhead.}$$

onde a aproximação é válida, novamente, para K grande.

Para cálculo da extração de redundância do método devemos ponderar este fator para correção da entropia dos nodos (H_p) da pirâmide, para que se possa encontrar a entropia equivalente (H_{eq}).

$$H_{eq} = H_p N_n / N^2 \quad (\text{em bits/pixel}) \quad (E 3.14)$$

Existem, na verdade, alguns métodos para construção de pirâmides reduzidas, nas quais $N_n = N^2$ de forma que o número de nodos é exatamente igual ao número de pixels. Nestas, a entropia pode ser computada diretamente e o grau de compressão pode ser diretamente relacionado com outros métodos de compressão de dados. Para mostrar que esta redução é possível, seja um nível k da PI que possua m amostras (m apenas um inteiro qualquer, quadrado de outro), então o nível $k+1$ da PI terá m/M^2 amostras e, obrigatoriamente, o nível k da PD deve conter apenas $m - m/M^2$ amostras e não m como esperado. Note que a soma dos nodos do nível k da PD com os do nível $k+1$ da PI devem ser iguais a soma dos do nível k da PI. Uma forma de se conseguir tal condição poderia ser a utilização de transformadas hierárquicas, sub-bandas com dizimação máxima, etc., mas iremos tratar destas no capítulo 5. Embora o esquema proposto em [3] seja uma destas transformadas, seu desenvolvimento teve enfoque puramente espacial e requer um baixo custo computacional (Pirâmide de Diferenças Reduzida). Outra forma de obter esta pirâmide reduzida surgiria da não prefiltração do sinal de um nível da PI para o outro. Neste caso, continuam valendo as equações 3.1-3.6, mas $h(n)$ seria um impulso unitário, ou seja, sem prefiltração. Este método tem aplicações limitadas devido ao montante de energia espúria advinda do *aliasing* introduzido de um nível para o outro, mas pode ser aplicado a pirâmides de poucos níveis e é competitivo por sua simplicidade.

3.3.1) Pirâmide de Diferenças Reduzida

Voltemos às equações 3.9-3.11. Simplificando a notação podemos reescrevê-las como sendo $\bar{m}$ a média de determinada quádrupla que passa para o nível superior da PI, de forma que

$$\bar{m} = (a + b + c + d)/4$$

e as diferenças contidas na PD seriam L_a, L_b, L_c e L_d , tais que

$$L_a = \bar{m} - a ; L_b = \bar{m} - b ; L_c = \bar{m} - c ; L_d = \bar{m} - d$$

Lembre-se que estas 5 equações descrevem apenas o processo de codificação de uma determinada partição de 2×2 pixels de uma imagem pertencente a PI e $\bar{m}$ é apenas um pixel pertencente a uma imagem de um nível adjacente superior ao em questão. Elas apenas ilustram, por simplificação, o processo geral.

Note que possuímos 5 equações, mas no processo de decodificação o receptor já terá $\bar{m}$ calculado (vindo do nível superior da PI reconstruída) e receberá as amostras L_a, L_b, L_c e L_d para calcular a quádrupla (a, b, c, d) . Com isto seriam apenas quatro incógnitas e poderíamos eliminar uma das equações, o que implica em eliminar a codificação de, por exemplo, L_d . Logo, conhecendo $\bar{m}$, L_a , L_b e L_c , as equações de reconstrução seriam :

$$a = \bar{m} - L_a \quad (E \ 3.15a)$$

$$b = \bar{m} - L_b \quad (E \ 3.15b)$$

$$c = \bar{m} - L_c \quad (E \ 3.15c)$$

$$d = 4\bar{m} - a - b - c = \bar{m} + L_a + L_b + L_c \quad (E \ 3.15d)$$

Com isto as operações embutidas no método continuariam muito simples, mas a pirâmide seria reduzida e a eficiência seria a mesma, já que não se modificou a forma de construção da Pirâmide e sim a sua representação através da PD.

3.3.2) Pirâmides Sem Prefiltragem

Podemos formar uma pirâmide reduzida se eliminarmos a prefiltragem do sinal de um nível para o outro. Desta forma m/M^2 amostras passariam inalteradas de um nível para outro e neste nível correspondente da PD só seria necessário armazenar erros referentes às $m-m/N^2$ amostras restantes. Atende-se, portanto, à condição anteriormente descrita.

A Figura 3.3 é uma ilustração da decomposição de um sinal unidimensional em uma pirâmide de 3 níveis ($K=2$). Nesta figura são representados x_0, x_1, L_0, L_1 e $L_2 = x_2$, mas L_0 e L_1 não têm amplitudes indicadas, apenas a posição relativa das amostras em relação às amostras de $x(n)$.

É cabível, entretanto, tecer alguns comentários sobre esta formulação que tem como idéia básica a troca do excesso de nodos da PD pela introdução de *aliasing* na sub-amostragem. Além da redução de *overhead*, outro ponto positivo é a simplicidade de codificação, já que não há prefiltragem. Quanto a *aliasing*, este será proibitivamente corruptor para níveis superiores de uma pirâmide de muitos níveis, mas tolerável nos níveis inferiores. É desejável buscar aplicações (como a que será discutida no capítulo 4) que requeiram poucos, talvez 3 ou 4, níveis da pirâmide. Por outro lado, poderíamos comutar este método com outro, ou torná-lo adaptativo, mas o ponto importante é que o aumento do erro de interpolação, provocado por *aliasing*, aumenta a entropia e taxa de bits, diminuindo a eficiência de compressão do método. Porém esta perda deve ser menor que o aumento de taxa provocado pelo número extendido de nodos em pirâmides não reduzidas. O método descrito na seção anterior também elimina *overhead* e é mais adequado à transmissão progressiva, contudo, sem prefiltragem, obtemos também um método reduzido, permitindo uma interpolação mais eficiente que a interpolação de ordem zero dentro dos elementos da quádrupla. Novamente, então, recaímos em uma comparação entre dois métodos, através do contrabalanço de *aliasing* e interpoladores mais eficientes e de maior comprimento.

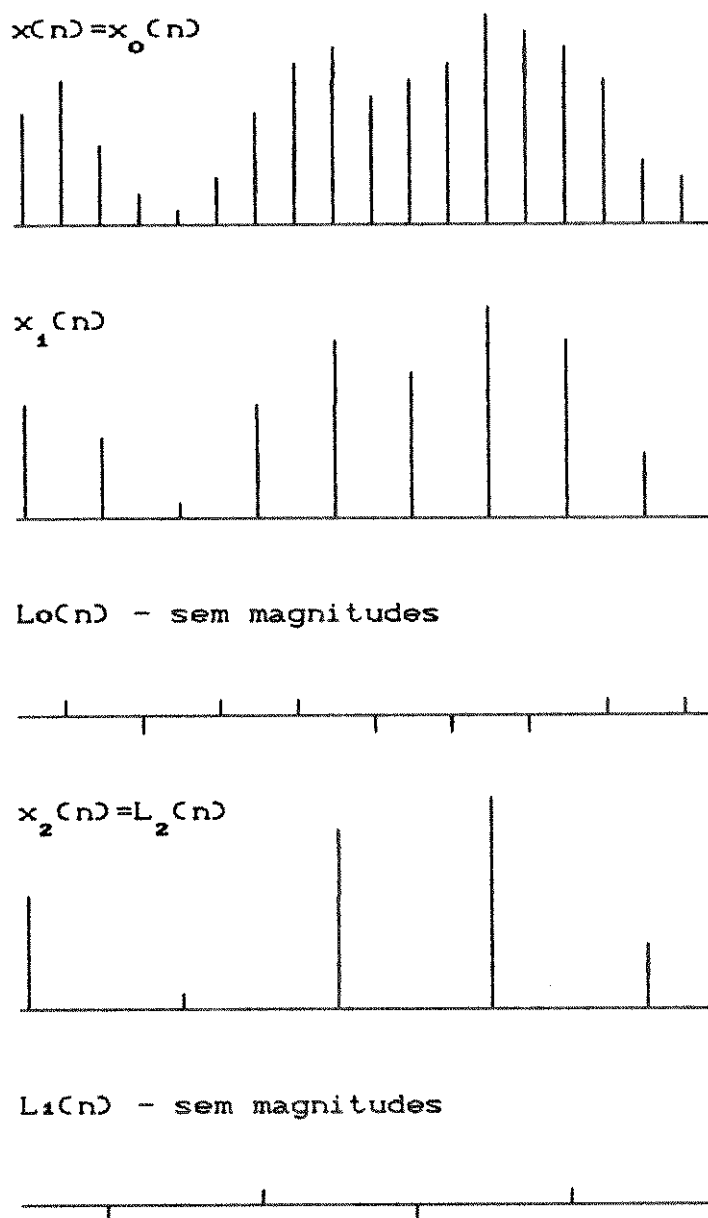


Figura 3.3 - Exemplo ilustrativo de eliminação de prefiltragem.

Por fim, eliminar a prefiltragem irá simplificar a codificação e tornar o método menos atrativo para aplicações em transmissão progressiva. Estes fatores também devem ser pesados quando da escolha da aplicação.

3.4) PIRÂMIDES QUINCUNX SEM PREFILTRAGEM

Como visto, sem prefiltragem, uma imagem da PI é obtida de outra em nível inferior por simples processo de sub-amostragem. Comumente iríamos utilizar dizimações horizontais seguida de verticais, formando uma matriz retangular de pixels. Todavia, iremos utilizar uma combinação de sub-amostragem retangular e hexagonal para formar uma pirâmide quincunx. Tal processo, embora menos eficiente, exhibe propriedades interessantes. Ainda, utilizaremos notações similares a [4] .

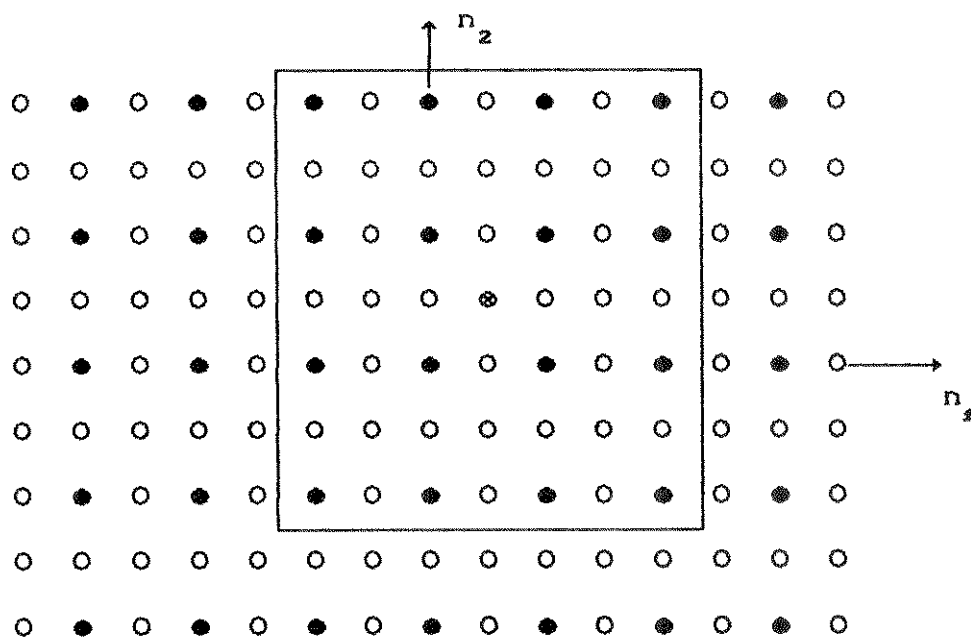


Figura 3.4 - Dizimação Retangular e região de suporte de filtro interpolador (7x7) para interpolar amostra @.

3.4.1) Estrutura Polifásica Bidimensional na Dizimação Retangular

Como na equação 3.4b, a sub-amostragem pode ser descrita por produto matricial. Note que apesar de $n=(n_1, n_2)$, ser escrito na horizontal, por simplificação, devemos encarar este vetor como coluna e se $y(n)=x(M n)$ e

$$M = \begin{bmatrix} M_1 & 0 \\ 0 & M_2 \end{bmatrix} = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix} = 2 I \quad (E 3.16)$$

então $y(n_1, n_2) = x(2n_1, 2n_2)$. A Figura 3.4 ilustra a sub-amostragem retangular e, nesta figura, é mostrada a região de suporte de um filtro FIR de ordens 7×7 que será utilizado para interpolar a amostra assinalada com $\otimes$.

Este filtro, com resposta impulsiva $h(n)$, deve ser um filtro FIR fase-nula, obedecendo às restrições descritas no Capítulo 2 para um interpolador, de forma que as amostras já existentes não sejam alteradas pela filtragem.

Sendo $h(n)$ um interpolador (unidimensional), então pode-se encontrar os filtros polifásicos [5] que o compõem.

$$h_i(m) = h(mM+i) \quad i=0,1,\dots,M-1$$

ou, mantendo as frequências de amostragem,

$$h_i(n) = \begin{cases} h(n) & \text{se } n=mM+i \\ 0 & \text{se } n \neq mM+i \end{cases}$$

Para obediência da restrição das amostras, $h_0(n) = \delta(n)$, restando, apenas projetar as $M-1$ fases restantes. Para extensão bidimensional definamos $k=(i,j)$ ($0 \leq i \leq M_1-1$, $0 \leq j \leq M_2-1$), M como em E3.16 e $m = (m_1, m_2)$.

$$h_k(m) = h_{ij}(m_1, m_2) = h(Mm + k), \text{ ou} \quad (E 3.17)$$

$$h_k(n) = \begin{cases} h(n) & \text{se } n = Mm \\ 0 & \text{se } n \neq Mm \end{cases} \quad (E 3.18)$$

$$h(n) = \sum_k h_k(n) \quad (E 3.19)$$

E, novamente, pode-se ver que

$$h_{00}(n) = \delta(n) \quad (E 3.20)$$

A Figura 3.5a indica o período de sub-amostragem para $M_1=M_2=2$. Nesta pode-se conferir que o número de filtros polifásicos, bem como o número de amostras de cada período são dados por

$$D = |\det(MD)|$$

No caso, $D=M_1M_2=4$ e, como já conhecemos um destes filtros polifásicos, resta-nos calcular 3, como na Figura 3.5a. Nesta figura, as três amostras do período que necessitam ser interpoladas são indicadas por a,b e c.

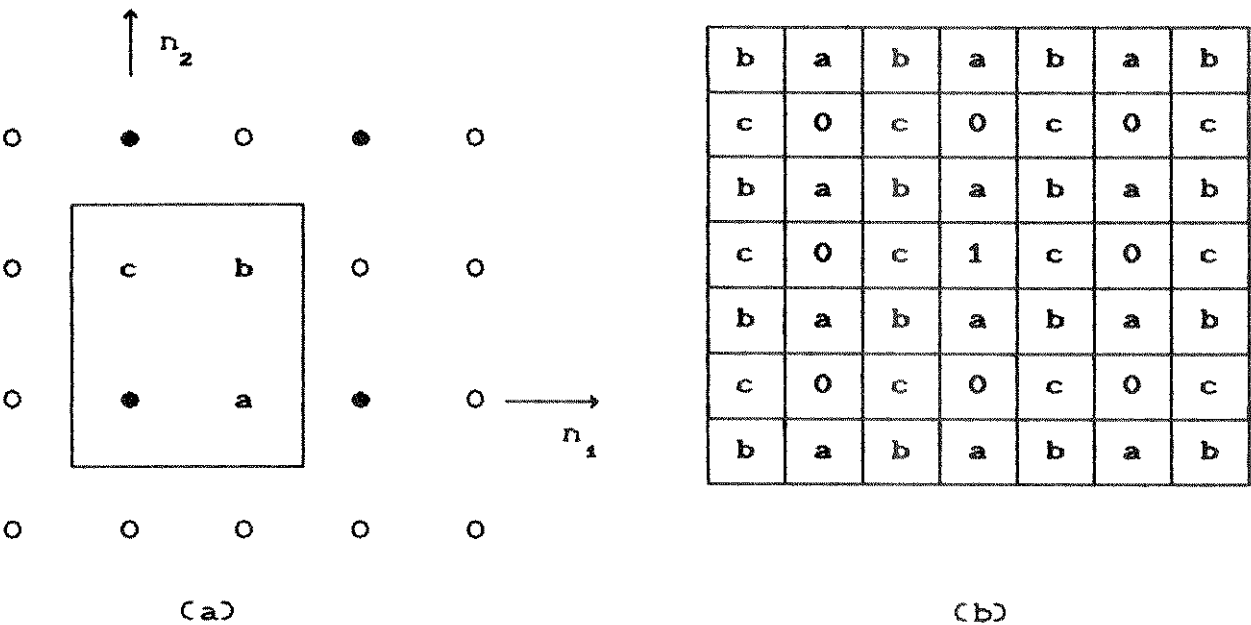


Figura 3.5 - a) Período de Sub-amostragem e amostras a serem interpoladas em cada período. b) Posição das amostras de $h(n)$ correspondentes a cada filtro polifásico.

Supondo, como exemplo, o filtro 7×7 descrito, a Figura 3.5b ilustraria as posições das amostras não nulas das fases de $h(n)$ (E 3.18) correspondentes à interpolação das amostras a,b e c. As amostras com 0's e 1's representam seu valor real e correspondem a $h_{00}(n) = \delta(n)$ por E3.20.

Suponha, ainda, que o filtro interpolador $h(n)$ seja calculado a partir de um núcleo unidimensional $g(n)$. Podemos utilizar, para tal, o método da Transformação de McClellan [4], como em [6], ou o método da Resposta Impulsiva Separável. Seja $g(n)$ (já levando em conta uma das fases) dado por

$$g(n) = [a_2 \ 0 \ a_1 \ 1 \ a_1 \ 0 \ a_2] \quad (E\ 3.21)$$

Se aplicarmos o primeiro método a $g(n)$, E3.20 não será atendido com as bases habituais. Se tentarmos encontrar uma transformação com isopotenciais menos arredondadas iremos nos aproximar do segundo método. Por isso, e tentando evitar o projeto bidimensional direto, simplificamos o problema fazendo

$$h(n_1, n_2) = g(n_1) g(n_2) \quad (E\ 3.22)$$

Com isto,

$$h(n) = \begin{bmatrix} a_2^2 & 0 & a_1 a_2 & a_2 & a_1 a_2 & 0 & a_2^2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ a_1 a_2 & 0 & a_1^2 & a_1 & a_1^2 & 0 & a_1 a_2 \\ a_2 & 0 & a_1 & 1 & a_1 & 0 & a_2 \\ a_1 a_2 & 0 & a_1^2 & a_1 & a_1^2 & 0 & a_1 a_2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ a_2^2 & 0 & a_1 a_2 & a_2 & a_1 a_2 & 0 & a_2^2 \end{bmatrix} \quad (E\ 3.23)$$

Comparando E3.23 com a Figura 3.5b, conclui-se que

$$h_a(n) = h_{10}(n) = \begin{bmatrix} 0 & 0 & 0 & 0 \\ a_2 & a_1 & a_1 & a_2 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (E 3.24)$$

$$h_c(n) = h_{01}(n) = h_a^t(n) \quad (E 3.25)$$

$$h_b(n) = h_{11}(n) = \begin{bmatrix} a_2^2 & a_1 a_2 & a_1 a_2 & a_2^2 \\ a_1 a_2 & a_1^2 & a_1^2 & a_1 a_2 \\ a_1 a_2 & a_1^2 & a_1^2 & a_1 a_2 \\ a_2^2 & a_1 a_2 & a_1 a_2 & a_2^2 \end{bmatrix} \quad (E 3.26)$$

Note que as amostras localizadas em a e c seriam interpoladas ao longo de uma dimensão, apenas.

3.4.2) Dizimação Quincunx/Retangular e o Interpolador

Seja $x(n)$ a imagem original, então

$$x_{k+1}(n) = x_k(V n) = x(V k n) \quad (E 3.27)$$

Se só aplicarmos dizimação retangular, então

$$V = M$$

$$V_0 = I ; V_1 = 2I ; V_2 = 4I \text{ e } V_k = 2^k I \quad (E 3.28)$$

Em $x(V k n)$, a densidade de amostras por amostras originais ou a frequência de sub-amostragem relativa à original (ou o inverso do tamanho do período de sub-amostragem) seria dado por

$$d_k = \frac{1}{D} = 1/2^{2k} \quad (E 3.29)$$

A estrutura a ser tratada aqui emprega dizimação hexagonal alternada com a dizimação retangular. Para a dizimação hexagonal [4],

$$V = \begin{bmatrix} M & M \\ M & -M \end{bmatrix} ; D = |\det(V)| = 2M^2$$

e o esquema progressivo de dizimação seria descrito por :

$$\begin{aligned} V_1 &= \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} & V_2 &= \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix} \\ V_3 &= \begin{bmatrix} 2 & 2 \\ 2 & -2 \end{bmatrix} & V_4 &= \begin{bmatrix} 4 & 0 \\ 0 & 4 \end{bmatrix} \\ &\vdots & &\vdots \\ V_{2k-1} &= \begin{bmatrix} 2^{k-1} & 2^{k-1} \\ 2^{k-1} & -2^{k-1} \end{bmatrix} & V_{2k} &= \begin{bmatrix} 2^k & 0 \\ 0 & 2^k \end{bmatrix} \end{aligned} \quad (E 3.30)$$

$$\text{Podemos, ainda, conferir que } d_k = 2^{-k}. \quad (E 3.31)$$

Há um maior número de passos, mas comparando E3.31 com E3.29 e verificando E3.30, nota-se que a cada dois passos percorremos o mesmo passo da dizimação puramente retangular. A codificação e decodificação irá envolver um maior número de passos, aumentando complexidade, mas há uma redução de entropia de erro de interpolação. Note que, para interpolar $x(a)$ ou $x(c)$ (veja Figura 3.5) partindo de uma subamostragem hexagonal (veja Figura 3.6), necessitaríamos conhecimento prévio do valor reconstruído de $x(b)$. Logo, a entropia do erro de interpolação diminui, dado que conheceríamos o valor reconstruído de cada amostra vizinha a a ou c . Sendo E_a o erro de interpolação de $x(a)$ e $H(E_a)$ a sua entropia, obviamente temos

$$H(E_a | V=V_{2k}) \geq H(E_a | V=V_{2k-1}) = H(E_a | V=V_{2k}; x(b))$$

Por outra ótica, pode-se afirmar que a dizimação hexagonal gera menor erro de reconstrução para sinais

circularmente limitados [4]. O sinal em questão não é limitado, mas assumindo um modelo isotrópico para imagens [7], a intensidade de *aliasing* seria circularmente proporcional, a interferência espectral da banda base (oriunda das repetições periódicas) seria então aproximadamente hexagonal e vale a superioridade da dizimação hexagonal sobre a retangular.

Outro ponto que implicaria em melhor interpolação decorre do filtro utilizado. Para aproximar $x(Vz^{k-1}n)$ a partir de $x(Vz^k n)$, que equivale a aproximar $x(b)$, só necessitamos utilizar $h_b(n)$ que é mais eficiente e interpola em ambas direções. Evitaríamos utilizar $h_a(n)$ e $h_c(n)$ e para interpolar $x(a)$ e $x(c)$, também utilizaríamos $h_b(n)$. Para visualizar como isto seria possível, na Figura 3.6 é mostrada a dizimação hexagonal, indicando as amostras a e c , e é disposto novo sistema de eixos de coordenadas inclinado de 45 graus. Sob estes novos índices $(n'_1, n'_2) = n'^T$, $x_i(n')$ poderia ser encarado como sub-amostrado retangularmente. Com isto basta utilizar a fase $h_b(n')$ de acordo com as direções inclinadas, obtendo interpolações bidimensionais eficientes também para $x(a)$ ou $x(c)$.

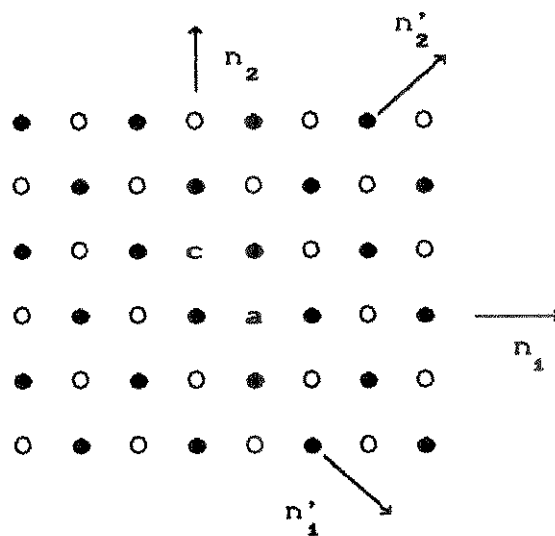


Figura 3.6 - Dizimação hexagonal e eixos inclinados auxiliares.

3.4.3) Interpoladores Práticos

Tomamos como exemplo um filtro 7x7, mas poderíamos encontrar filtros geradores $g(n)$ para $M_1=M_2=2$ com ordem $4k-1$ (k inteiro positivo qualquer). Daí, $g(n)$ forneceria aproximação de ordem $2k-1$ e seria da forma :

$$g(n)=[a_k \ 0 \ \dots \ a_2 \ 0 \ a_1 \ 1 \ a_1 \ 0 \ a_2 \ \dots \ 0 \ a_k]$$

O núcleo escolhido até agora como exemplo utilizou $k=2$ e um bom exemplo seria o da extensão discreta do núcleo de convolução cúbica [8] [9].

$$g_{cc}(n) = [-1 \ 0 \ 9 \ 1 \ 9 \ 0 \ -1] \times 1/16 \quad (E \ 3.32)$$

Se quisermos um núcleo de convolução menor, por exemplo $k=1$, a aproximação seria linear para $g(n)$ e planar para $h_b(n)$. Tal interpolação apenas toma a média das amostras ao redor.

$$g(n) = [1/2 \ 1 \ 1/2] \ ; \ h_b(n) = \begin{bmatrix} 1/4 & 1/4 \\ 1/4 & 1/4 \end{bmatrix} \quad (E \ 3.33)$$

Outra filtragem menos custosa, mas razoavelmente eficiente poderia ser feita se interpolássemos ao longo de duas direções ortogonais, apenas, e tomássemos a média dos resultados. Se utilizássemos o núcleo de E3.32, teríamos

$$h_b(n) = \begin{bmatrix} -1 & 0 & 0 & -1 \\ 0 & 9 & 9 & 0 \\ 0 & 9 & 9 & 0 \\ -1 & 0 & 0 & -1 \end{bmatrix} \times 1/32 \quad (E \ 3.34)$$

3.4.4) Resultados Comparativos

Como comentado na seção 3.3.2, eliminar a prefiltragem deve acarretar uma limitação quanto ao número de níveis da pirâmide. Para uma dizimação retangular, o quinto nível da PI, por exemplo, seria formado por uma sub-amostragem da imagem original com uma frequência 2^{-4} vezes menor que a original em cada direção. Melhor explicando, de cada bloco de 16×16 pixels reteríamos apenas um no canto superior esquerdo. É claro que o *aliasing* introduzido seria proibitivo.

Contudo, tomamos uma imagem monocromática de 256×256 pixels (digitalizada com 8 bits/pixel) e verificamos a entropia dos quatro primeiros níveis da PD para os métodos da Pirâmide de Diferenças Reduzida (PDR) e da Pirâmide Quincunx sem Prefiltragem (PQsp). Para o último, agrupamos dois níveis em um de forma que consideramos E3.30 como um passo apenas. Todavia mantendo-se, para a codificação, o nível intermediário. O resultado final é que, a cada nível da PD, $2/3$ das amostras são calculados de uma forma e $1/3$ com outra. Este agrupamento só se faz necessário por motivos de comparação dos métodos e pode-se considerar o número de níveis nominal como metade do realmente alcançável.

Utilizamos a interpolação mais simples que é dada pela equação 3.33 e os resultados obtidos foram os seguintes :

TABELA 3.1 - Entropia em bits/pel das amostras da PD

	Nível da PD	PDR	PQsp	Resolução	Diferença
H ₁	0	4.14	4.08	256x256	0.06
H ₂	1	4.60	4.87	128x128	-0.27
H ₃	2	5.20	5.62	64x64	-0.42
H ₄	3	7.25	7.25	32x32	0.00
H _{T3}	0-1-3	4.42	4.43	---	-0.005
H _{T4}	0-1-2-3	4.32	4.35	---	-0.025

Imagem utilizada - GIRL (256x256x8bits)

Na tabela anterior H_4 representa a entropia de primeira ordem do nível de resolução mínima (32x32 pixels) e foi tomada como a entropia da imagem original. As entropias totais foram calculadas por

$$H_{T3} = H_1 \frac{3}{4} + H_2 \frac{3}{16} + H_4 \frac{1}{16}$$

$$H_{T4} = H_1 \frac{3}{4} + H_2 \frac{3}{16} + H_3 \frac{3}{64} + H_4 \frac{1}{64}$$

Vê-se pelos resultados que o desempenho dos métodos é comparável para poucos níveis, embora a Pirâmide Quincunx exija uma menor complexidade.

3.5) COMENTÁRIOS ADICIONAIS

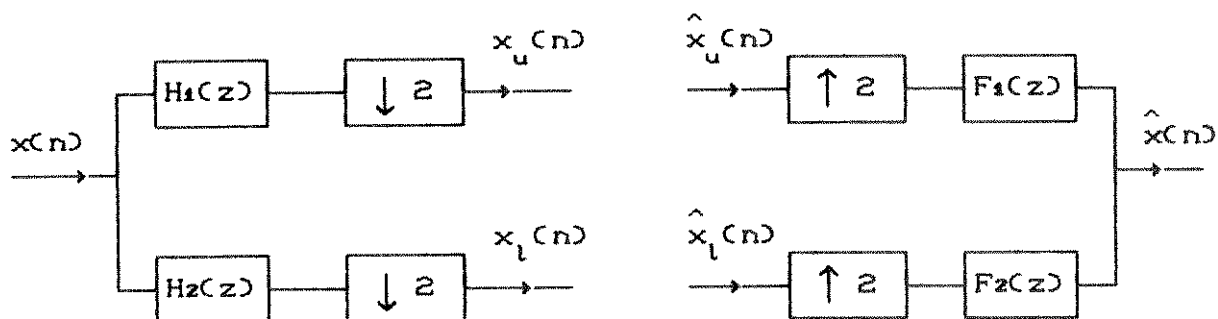
3.5.1) Relação com bancos de filtros

Por simplificação faremos a análise primeiramente sobre sinais unidimensionais.

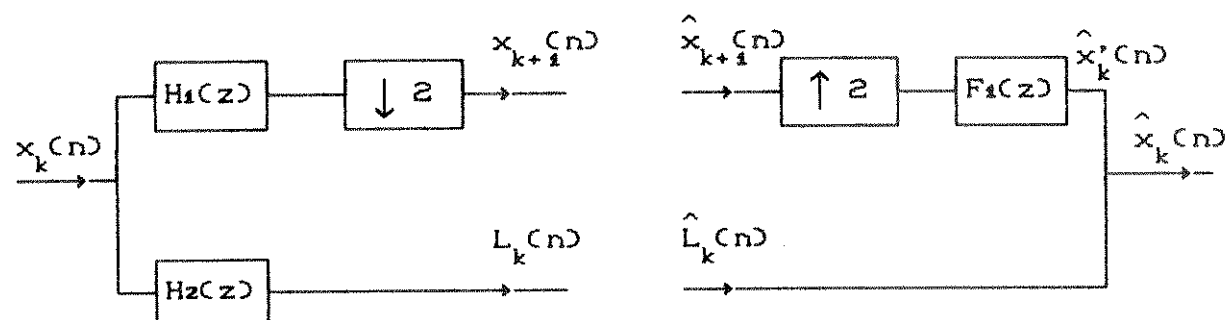
A codificação em pirâmide é encarada como uma estrutura em árvore já que em cada passo uma sequência é transformada em duas outras. Um determinado nível da PI gera outro nível da PI e o respectivo nível da PD. Para se completar o processo repete-se o procedimento para a nova imagem da PI formada.

Por isto, iremos fazer uma comparação com o banco de filtros QMF (Quadrature Mirror Filter) de duas bandas [5]. Este é esquematizado como na Figura 3.7a, com $H_1(z) = H_2(-z) = F_1(z) = -F_2(-z)$. Este banco pode ser conectado em árvore com outros similares de forma que se partilhe como desejado o espectro. Na codificação em pirâmide o mesmo ocorre e sua esquematização é mostrada na Figura 3.7b. Nesta figura, foi levado em conta o modelo descrito na seção 3.2 e notamos que as principais diferenças residem no projeto do passa-altas e na não-dizimação do sinal.

Para o caso bidimensional basta-nos proceder como em [10] utilizando bancos QMF ao longo de cada direção, ou seja aplicando-os às linhas para depois aplicá-los às colunas, ocorrendo o mesmo para pirâmides. Em [11] é proposto que se sub-amostre o ramo passa-altas já que é suposto que as baixas frequências teriam sido atenuadas, mas no nosso entender isto caracterizaria codificação em Transformadas/Sub-Bandas e neste trabalho fazemos distinção entre os métodos. Uma breve análise da hierarquia de sistemas de tal tipo será feita no Capítulo 5. Além disto, seria exigido núcleos mais elaborados para os filtros que compõem os bancos, caso contrário a sub-amostragem poderia provocar *aliasing* insustentável que poderia comprometer a recuperação do sinal, principalmente no caso descrito das pirâmides sem prefiltração. Por outro lado, em [18] é demonstrado que se fizermos o conjunto de filtros do banco QMF atenderem algumas restrições, sempre será possível dizimarmos o ramo passa-altas na codificação em pirâmide, permitindo reconstrução perfeita, mas seguiremos os princípios descritos neste capítulo, já que os filtros expostos não atendem estas condições. É uma questão de opção, deixando uma maior discussão sobre sistemas Sub-bandas para o Capítulo 5. Tal ponto de vista com relação a pirâmides é comumente chamado de Transformada em Pirâmide. Certa discussão de nomenclaturas e comparações são levadas a cabo em [12]. São, todas, formas de se analisar o mesmo princípio, mas neste capítulo nos atemos ao princípio primário como descrito na seção 3.2 e esquematizado na Figura 3.7. Deve ser salientado que os mesmos autores de [3] em outro trabalho [13] se referiram ao termo *Pyramid Transform*, mas não tendo relação estreita com este agora citado. Em [13] apenas tomou-se a transformada discreta de cossenos da imagem e aplicou-se o método da seção 3.3.1 aos coeficientes resultantes. Guardou-se, contudo, uma relação entre coeficientes de mesma hierarquia entre os diversos blocos. Seria, entretanto, uma transformada hierárquica mista, na qual, como veremos no Capítulo 5, transformadas hierárquicas não sobrepostas (como é o caso da pirâmide de diferenças reduzida) se aplicam em uma estrutura em árvore à uma transformada de cossenos que é um banco de filtros particular, ou uma transformada não sobreposta.



(a)



(b)

Figura 3.7 - a) Par de filtros para estrutura QMF. O banco pode ser conectado em cascata pela associação em árvore destas estruturas. O ramo superior indica o sinal passa-baixas enquanto o inferior indica o sinal passa-altas. b) Estrutura análoga para codificação em pirâmide sem decodificação local

Como os filtros não são ideais, para o bom funcionamento do método, a figura 3.7b deve ser modificada de forma a incorporar uma decodificação local do sinal passa-baixas e, a partir deste sinal, realizar a filtragem do ramo passa-altas. Esta decodificação é feita a partir do ramo do passa-baixas, tomando o sinal $x'_k(n)$ na figura 3.7b e o resultado seria como na Figura 3.8a. Em 3.8c, o mesmo diagrama é mostrado para o sistema sem prefiltragem. Note, nesta figura, o bloco z^{-1} de atraso, para desfazer o sincronismo dos sub-amostradores.

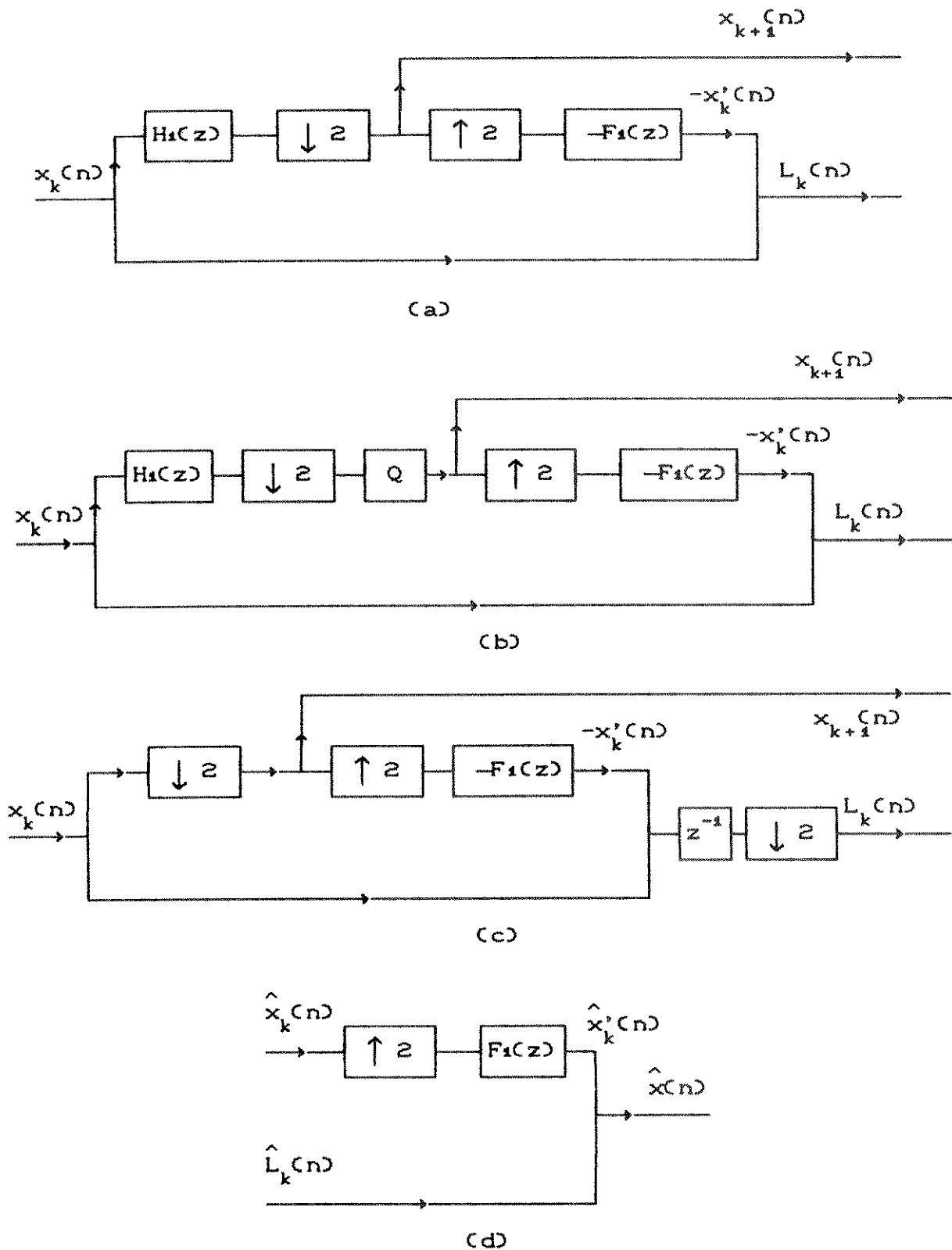


Figura 3.8 - a) Diagrama do módulo de codificação da Figura 3.7, incorporando decodificação local. b) Mesmo módulo que em (a), mas incorporando quantizador. c) Mesmo módulo que em (a), mas eliminando prefiltragem. d) Decodificador.

Sendo $X'_k(z)$ a Transformada Z de $x'_k(n)$, a decodificação local deste sinal implicaria em fazer

$$X'_k(z) = \frac{1}{2} (H_1(z) F_1(z) X(z) + H_1(-z) F_1(z) X(-z))$$

$$L_k(z) = X_k(z) - X'_k(z).$$

Adicionalmente, poderíamos, na decodificação local, incorporar o efeito de quantização. Isto imprimiria maior realismo à Figura 3.8a e é mostrado na Figura 3.8b. Convém ressaltar que as mesmas considerações podem ser aplicadas ao sistema sem prefiltragem se simplesmente fizermos $H_1(z)=1$. Desta forma, o sistema seria como apresentado na Figura 3.8c. Seria possível, igualmente, incorporar o efeito da quantização na decodificação local bastando para tal incorporar o bloco deste quantizador a esta figura. Finalmente em 3.8d é mostrado o decodificador, válido para os dois primeiros casos.

3.5.2) Adaptatividade

Como qualquer outro método, podemos torná-lo adaptativo, mas possuímos muitos variáveis com as quais podemos lidar. Os pontos principais que seriam susceptíveis à adaptação poderiam ser enumerados como:

- i) Quantizadores das amostras dos níveis da PD.
- ii) Forma de codificação do nível superior da PD ($L_k(n)$).
- iii) Direções e Períodos de sub-amostragem (M_1 e M_2).
- iv) Prefiltros e interpoladores.
- v) Número de níveis.
- vi) Tamanho da imagem a ser descortinada ao usuário.

O primeiro caso, adaptação de quantizadores, pode ser realizada através dos métodos comuns encontrados na literatura. Tal quantização pode ser escalar ou vetorial [14] [15]. Em [16] (iii) foi feito adaptativo graças a uma formulação que diz respeito a uma estrutura de dizimação arbitrária. Por último, Troxel em [17] propõe sistema de transmissão progressiva adaptativo levando em conta a resposta visual humana. Com isto, (i), (iii), (v) e (vi) foram feitos adaptativos em um trabalho bastante interessante.

3.5.3) Dimensões da Imagem para Apresentação ao Usuário.

Quando da decodificação da pirâmide através da PD, necessitamos apresentar ao usuário as imagens parcialmente obtidas. Uma controvérsia bastante cabível diz respeito ao tamanho da imagem a ser descortinada em cada passo de decodificação. Duas formas, entre outras, são possíveis : Tamanho reduzido proporcionalmente à resolução da imagem $x_k(n)$ (que no momento é a imagem decodificada da PI); ou tamanho original representando uma versão borrada (interpolada) obtida a partir de $x_k(n)$. Em [17] é dito que o primeiro método torna as imagens melhor reconhecíveis a uma resolução bastante reduzida. Os resultados são baseados em estudos do modelo visual humano e consideram que o observador se posiciona a uma distância fixa do monitor ou tela. Esta é uma questão bastante sujeita a discussões, principalmente porque depende da aplicação, e ainda é estudada.

3.6) REFERÊNCIAS

- [1] S.L. Tanimoto, *Image Transmission With Gross Information First*, Computer Graphics and Image processing, vol.9, pp 72-76, January 1979.
 - [2] P. J. Burt, E.H. Adelson, *The Laplacian Pyramid as a Compact Image Code*, IEEE Trans. on Commun., COM-31, pp 532-540, April 1983.
 - [3] L. Wang, M. Goldberg, *Reduced-difference Pyramid : a Data Structure for Progressive Image Transmission*, Optical Engineering, Vol 28 #7, July 1989.
 - [4] D.E. Dudgeon, R. M. Mersereau, *Multidimensional Digital Signal Processing*, Englewood Cliffs,NJ, Prentice Hall, 1983.
 - [5] R.E. Crochiere, L.R. Rabiner, *Multirate Digital Signal Processing*, Englewood Cliffs, NJ, Prentice-Hall, 1983
 - [6] T. C. Chen, R. J. P. de Figueiredo, *Image Decimation and Interpolation Techniques Based on Frequency Domain Analysis*, IEEE Trans on Commun., COM-32, pp 479-484, April 1984.
 - [7] J.B. O'Neal Jr., T. R. Natarajan, *Coding Isotropic Images*, IEEE Trans. on Information Theory, IT-23, pp 697-707, November 1977.
 - [8] R.G. Keys, *Cubic Convolution Interpolation For Digital Image Processing*, IEEE Trans. on ASSP, ASSP-29, pp 746-749, June 1983.
 - [9] R.L.Queiroz,J.B.Yabu-uti, *Técnicas de Projeto de Filtros FIR para Dizimação e Interpolação de Sinais Discretos*, RT-178, Contrato 208/89, UNICAMP/TELEBRÁS, Setembro 1989.
-

-
- [10] J.W. Woods, S.D. O'Neil, *Sub-band Coding of Images*, IEEE Trans. on ASSP, ASSP-34, pp 1278-1288, October 1986.
 - [11] T. Kronander, *Sampling of Band-Pass Pyramids*, IEEE Trans. on Commun., COM-35, pp 125-127, January 1987.
 - [12] N. Baaziz, C. Labit, *Laplacian Pyramid Versus Wavelet Decomposition for Image Sequence Coding*, Proc. of ICASSP, pp 1965-1968, 1990.
 - [13] L. Wang, M. Goldberg, *Pyramid Transform Coding Using Vector Quantization*, Proc. of ICASSP, pp 812-815, 1988.
 - [14] N.S. Jayant, P. Noll, *Digital Coding of Waveforms*, Englewood Cliffs, NJ, Prentice-Hall, 1984.
 - [15] N.M. Nasrabadi, R.A. King, *Image Coding Using Vector Quantization : A Review*, IEEE Trans. on Commun., vol. 36, no. 8, August 1988.
 - [16] S.S. Ribes, L.T. Urgell, *An Adaptive Pyramid Image Coding System*, Proc. of ICASSP, pp 808-811, 1988.
 - [17] W.D. Hofmann, D.E. Troxel, *Making Progressive Transmission Adaptive*, IEEE Trans. on Commun., COM-34, pp 806-813, August 1986.
 - [18] M. Vetterli, C. Herley, *Wavelets and Filter Banks : Relationships and New Results*, Proc. of ICASSP, pp 1723-1726, 1990.
-

CAPÍTULO 4

DPCM EM PIRÂMIDE : UM SISTEMA HÍBRIDO

Neste capítulo iremos tratar de um método híbrido para redução da velocidade de trabalho de codificadores do tipo DPCM em ambientes de alta taxa de amostragem. Tal método envolve sinais com diferentes resoluções e se baseia na codificação em pirâmide sem prefiltragem, como descrito no capítulo anterior. É, em realidade, uma aplicação do esquema proposto, mas não é propícia para transmissão progressiva. A principal razão para tal decorre do fato que em sistemas de alta amostragem, como CODEC's de vídeo e TV, não há necessidade nem aplicabilidade deste tipo de transmissão. Por outro lado, conceitos de sistemas de multiresolução são utilizados para uma compressão de dados eficiente, com boa extração de redundâncias, permitindo reconstrução de imagens com alta qualidade subjetiva, mas a um custo baixo.

O DPCM em pirâmide é uma alternativa ao sistema DPCM convencional, no sentido que, acoplando circuitos em paralelo a este, podemos reduzir a taxa de amostragem dos componentes do circuito como um todo. Diminuir taxas implica em reduzir velocidade e aumentar o intervalo de processamento entre amostras. Com o aumento do intervalo, dois benefícios são aparentes. Em primeiro lugar, poder-se-ia aumentar o número de operações envolvidas na predição ou em pesquisas de adaptação, tornando-os mais eficientes. Alternativamente, pode-se manter os algoritmos, mas utilizar componentes mais lentos e, conseqüentemente, mais baratos. Em sistemas de codificação de vídeo e TV é comum a taxa de amostragem girar em torno de 10MHz. Com isto, entre cada par de amostras, têm-se 100ns para efetuar todo o processamento requerido. Desta feita, a implementação destes sistemas em países onde a tecnologia de ponta não é rapidamente alcançada, pode se tornar uma tarefa enormemente difícil. Por tal, um aumento do período de processamento pode se tornar uma grande ajuda e servir como instrumento de vital importância para o desenvolvimento destes CODEC's.

Não se quer, aqui, projetar um sistema por completo, mas apresentar os princípios básicos e demonstrar que é possível, ainda, obter maior eficiência em relação a sistemas DPCM convencionais, todavia adicionando circuitos paralelos de baixa complexidade e perfeitamente viáveis.

4.1) DEFINIÇÃO DO MÉTODO

Em primeiro lugar, pensando em tornar paralelas as operações de um DPCM, poder-se-ia tentar uma estrutura polifásica empregando vários sistemas em paralelo. Por exemplo, 4 CODECS DPCM, cada um aplicado a um sinal dizimado por 4. Ou seja, sendo $y_i(n)$ a entrada de cada CODEC, teríamos

$$y_i(n) = x(4n+i) \quad i=0,1,2,3$$

sendo $x(n)$ o sinal original a ser codificado.

As principais desvantagens deste método são claras se levarmos em conta que a complexidade foi quadruplicada, apesar de cada um trabalhar com um quarto da taxa de amostragem. A taxa de bits/segundo (e não bits/amostra, já que cada fase possui menos amostras por segundo que o processo não dizimado) de cada CODEC também deve ser aproximadamente multiplicada por 4 para se encontrar a taxa resultante. Esta taxa resultante deve ter a mesma ordem de grandeza da do DPCM isolado aplicado ao sinal original, não havendo, a princípio, alguma vantagem. Além disto, não haveria compressão sem uma maior precisão dos sistemas individuais, já que as amostras estariam mais espaçadas e a predição seria menos eficiente.

O sistema proposto a seguir combina estas diferentes fases, mas permitindo a redução real da entropia do erro.

O princípio geral se baseia na pirâmide sem prefiltração. Para que se possua um atraso tolerável e baixa complexidade, limitamo-nos a 3 níveis apenas e aplicaremos o

método unidimensionalmente às linhas de uma imagem, imaginando a codificação simples de um sinal de vídeo. O método é descrito na Seção 3.3.2 e ilustrado na Figura 3.3. Iremos codificar os níveis $L_0(n)$, $L_1(n)$ e $L_2(n)$ da PD. O princípio é bastante simples:

- i) Dizime por dois e por quatro o sinal original $x(n)$, obtendo $x_1(n)$ e $x_2(n)$, respectivamente.
- ii) Codifique $x_2(n)$ através do DPCM. Chame os sinais de erro de predição de $L_2(n)$, redefinindo, assim, este nível da pirâmide como sendo um sinal de erro e não o sinal original, como definimos o último nível da PD no capítulo anterior.
- iii) Interpole $x_2(n)$, duplicando sua frequência de amostragem. Codifique o erro entre o sinal resultante e $x_1(n)$. Despreze as amostras de diferença envolvendo as amostras originais de $x_2(n)$, que serão sempre nulas. Este sinal será $L_1(n)$.
- iv) Interpole $x_1(n)$, duplicando sua frequência de amostragem. Codifique o erro entre o sinal resultante e $x(n)$. Despreze as amostras de diferença envolvendo as amostras originais de $x_1(n)$, que serão sempre nulas. Este sinal será $L_0(n)$.

O diagrama do circuito para realizar tais procedimentos é ilustrado na Figura 4.1. Nesta figura, MUX2 e DEMUX2 são dispositivos para agrupar e dividir as fases de um sinal e substituem as chaves das comutações polifásicas da Seção 2.1.3. Para o DEMUX2, suas duas saídas tem metade da amostragem do sinal de entrada e dividem-no deixando cada ramo com as amostras pares ou ímpares da entrada. O inverso ocorre com MUX2, reagrupando um sinal a partir de dois outros que possuam metade da amostragem do primeiro. Os atrasos ξ e η são compensatórios e iguais aos atrasos provocados pelos interpoladores. Adicionalmente $h(n)$ representa a fase da resposta impulsiva do filtro para interpolar as amostras faltantes e o DPCM é tratado por enquanto como uma caixa-preta que se baseia no esquema da Figura 4.2 [11]. Desta forma, não entraremos em detalhes sobre o DPCM, apenas comparando-o com o sistema aqui modificado.

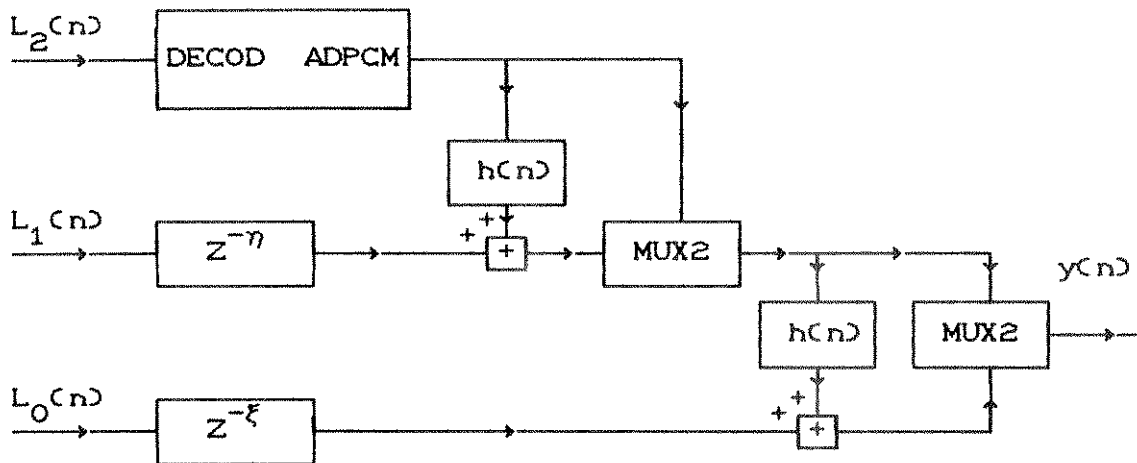
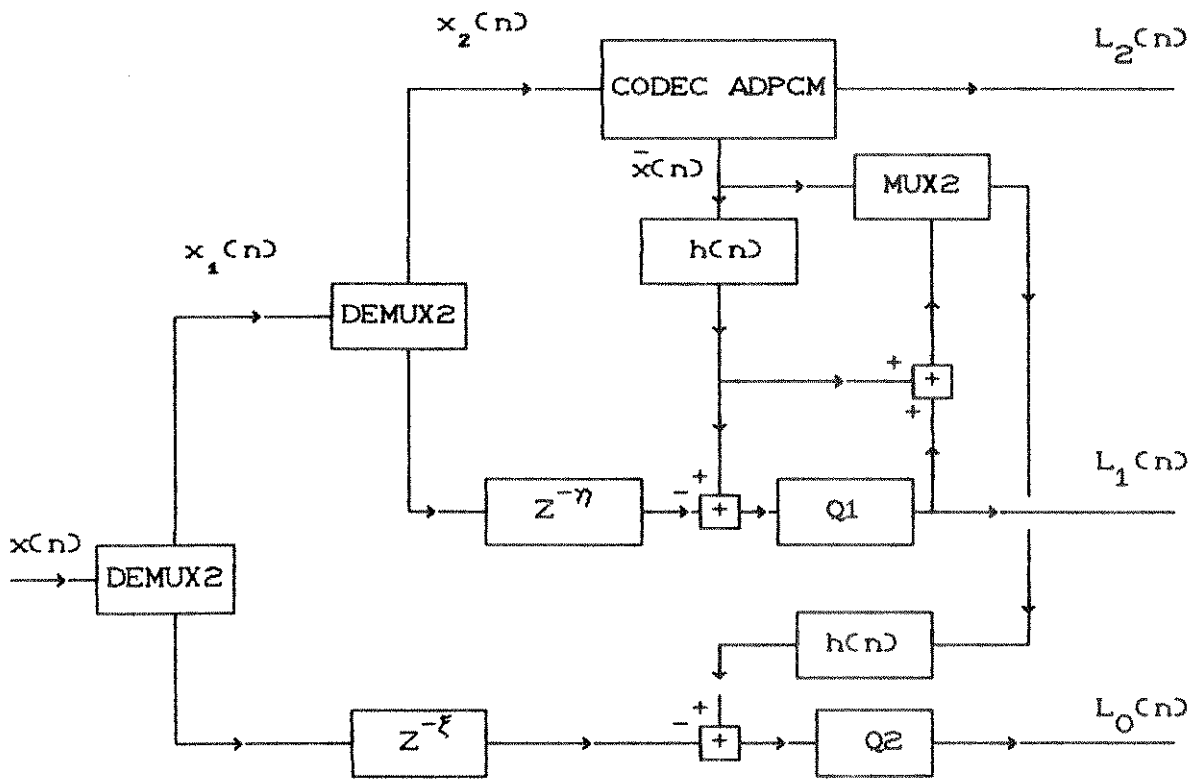


Figura 4.1 - Transmissor e receptor do DPCM em pirâmide.

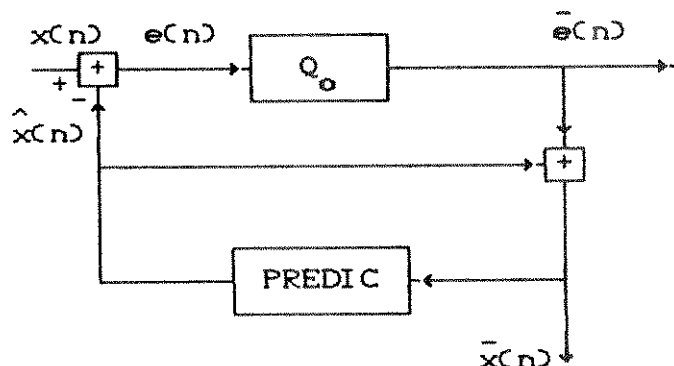


Figura 4.2 - Diagrama básico do sistema DPCM.

Note que $\bar{x}(n)$, na Figura 4.2, representa o sinal decodificado localmente, com o que é possível incorporar os erros de quantização de $x_2(n)$ quando calculamos $L_1(n)$. A mesma decodificação local (incorporando erros de quantização) aparece na Figura 4.1 quando calculamos $L_0(n)$. Assim há sincronização do receptor e transmissor. Não são consideradas as operações de multiplexação nem características do canal, considerando-o como ideal.

Podemos ver a coisa de outra forma. Algumas amostras são erros de predição utilizando predição comum extrapolativa, já que $x_2(n)$ é codificado pelo DPCM. As amostras restantes são erros de interpolação e pode-se encará-las como erro de predição utilizando predição interpolativa. Logo, é um sistema DPCM híbrido inter/extrapolativo. O ganho que pode haver, na codificação, se deve a um melhor desempenho da interpolação sobre a extrapolação. Incorporar interpolação ao DPCM já foi pensado antes no DPCM Interpolativo [1] [2], mas não se explorou um sistema de multiresolução nem conceitos de Processamento Digital de Sinais a Multitaxas. Assim, este DPCM híbrido e o interpolativo são estruturalmente diferentes.

4.2) COMPARAÇÕES COM SISTEMA CONVENCIONAL

Pelo que pode ser inferido na Figura 4.1, vemos que a maior parte do circuito opera sob uma amostragem quatro vezes menor que a original, incluindo o DPCM. O resto opera sob metade da taxa original. Então há uma redução geral da velocidade do circuito que é mais pronunciada no sistema DPCM imbutido. Contudo, às custas de adição de processamento paralelo ao esquema convencional. Note, entretanto, que as contribuições periféricas são bastante simples. São somas e atrasos, havendo maior complexidade apenas na filtragem e na quantização. Os filtros são muito simples (2 ou 4 coeficientes) e podemos adotar quantizações simplistas com bons resultados, fazendo com que o circuito adicional seja de simples implementação.

A taxa final do sistema piramidal é função da taxa de cada sinal. Sendo α, β e γ , as taxas de $L_2(n)$, $L_1(n)$ e $L_0(n)$, respectivamente, a taxa resultante (R) que leva em conta as diferentes frequências de amostragem, mas não inclui nenhuma informação de multiplexação, é dada por :

$$R = \frac{\alpha + \beta + 2\gamma}{4} \quad (\text{E 4.1})$$

4.2.1) Vantagens e Desvantagens

Pode-se, brevemente, apontar as vantagens do DPCM em pirâmide sobre o sistema convencional como

- Redução da velocidade das partes integrantes do sistema e barateamento ou incremento do DPCM imbutido.
- Maior eficiência de predição dada a possível vantagem de predição interpolativa sobre predições extrapolativas.

- Maior compressão, com menor degradação, dada a possibilidade de se adotar três esquemas de quantização diferentes, um para cada nível. Com isto pode-se escolher a forma mais adequada de degradação para a correspondente resolução de cada nível.
- Possibilidade de compressão adicional pela transmissão progressiva e condicional de blocos da imagem, evitando transmitir mais informação de blocos que já tenham atingido um nível de degradação satisfatório sem resolução total.

As principais desvantagens são : i) Necessidade de adição de circuitos paralelos auxiliares; ii) Necessidade de circuitos de multiplexação e sincronismo.

Existe um compromisso entre as estratégias de decodificação do DPCM, isoladamente, e dos níveis $L_1(n)$ e $L_0(n)$. É claro que se o DPCM for demasiadamente sofisticado, com predição e quantização adaptativas e complexas, este, isoladamente, pode se tornar mais eficiente do que o sistema híbrido. Isto é possível se não for incorporado, correspondentemente, sofisticações aos interpoladores e quantizadores do circuito auxiliar. De forma geral, se não houver discrepância acentuada, o DPCM em Pirâmide terá sempre a seu favor as vantagens citadas com a minoração relativa da primeira desvantagem apontada.

4.2.2) Preditores e Interpoladores Simples

Além do quantizador, o sistema DPCM tem seu desempenho determinado pela acuidade do preditor. Este preditor deve ser o mais preciso possível, de forma a reduzir o erro de predição. No sistema piramidal, a mesma consideração é verdadeira, mas não só preditores extrapolativos devam ser projetadas, mas também preditores interpolativos, o que fica a cargo do projeto de um bom interpolador ($h(n)$ na Figura 4.1). Primeiramente, devemos elaborar o projeto do preditor extrapolativo que faz parte do DPCM que codifica o sinal de menor resolução. Projeto de preditores e outras partes, integrantes do DPCM é assunto

suficientemente bem comentado na literatura [1] [4] e a codificação de sinais de TV por DPCM está perto de possuir seu padrão para transmissão a 34Mbit/s [5]. Não iremos detalhar projetos, mas utilizar as técnicas que já existem e que sejam simples, já que queremos apenas ferramentas necessárias às comparações. Quanto ao preditor, iremos adotar predição intra-quadro e deixar de lado a estimação de movimento.

A predição é uma operação de filtragem e pode ser descrita pelo produto interno de dois vetores. Seja X o vetor que contém as amostras envolvidas na predição e P o vetor com os elementos do filtro de predição. Para que haja sincronia entre transmissor e receptor, devemos envolver amostras decodificadas localmente (Figura 4.2), de forma que o vetor de análise X não necessariamente contém amostras originais. Tentando simplificar e eliminar os índices temporais, define-se o valor predito da amostra rotulada como $\hat{X}$ dado por

$$\hat{X} = P^T X = [a_1 \ a_2 \ a_3 \ \dots \ a_n] [\bar{x}_1 \ \bar{x}_2 \ \bar{x}_3 \ \dots \ \bar{x}_n]^T \quad (E\ 4.2)$$

E o sinal a ser codificado será

$$e = X - \hat{X} \quad (E\ 4.3)$$

Na Figura 4.3 são ilustradas algumas linhas de dois campos entrelaçados, como em sinais de TV. Para predição de determinada amostra, rotulada como X , temos as amostras rotuladas como A,B,C... para serem envolvidas na predição. Como já dito, A,B,C... são amostras decodificadas localmente e já incorporam os efeitos de quantização. Em um sistema de diferenças (o mais simples) temos

$$\hat{X} = A \quad P=[1] \quad X=[A] \quad (E\ 4.4)$$

Para predição intracampo [4], temos

$$X^T = [A \ B \ C] \quad P^T = [1 \ -0.75 \ 0.75] \quad (E\ 4.5)$$

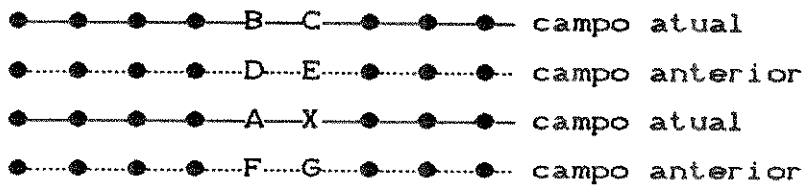


Figura 4.3 - Amostras envolvidas na predição.

Esta é uma aproximação planar e os valores foram aproximados de um valor de mais fácil implementação. Na predição intra-quadro, podemos utilizar sinais dos outros campos em cenas sem movimentos muito bruscos. De [5], temos

$$X^T = [A \ D \ E \ F \ G] \quad P^T = [1 \ -1/2 \ 1/2 \ -1/2 \ 1/2] \quad (E \ 4.6)$$

Esta forma de predição foi descartada das simulações por não termos em disponibilidade cenas em movimento digitalizadas e em imagens estáticas (E 4.6) será uma interpolação. Ainda utilizamos o preditor de (E 4.5) adaptando os coeficientes pelo método do gradiente (LMS) extendido para o caso bidimensional como em [6]. Não entraremos, aqui, em detalhes, mas o passo de adaptação μ foi escolhido de forma a se situar pouco abaixo do limitante de convergência do método ($2/30^2$).

$$P(n) = P(n+1) + \mu (x(n) - \hat{x}(n)) X, \quad \mu = \frac{0.1}{255^2} \quad (E \ 4.7)$$

Para a interpolação, como discutido na Seção 2.4, devemos utilizar filtros fixos e simples. Dois bons candidatos seriam o interpolador linear e a convolução cúbica. Referindo àquela seção, note que os coeficientes do filtro são muito simples sendo frações de potências de 2. Assim, não há erro de quantização dos coeficientes em aritmética de ponto fixo. Também, estes filtros possuem a estrutura polifásica adequada e comprimentos curtos, podendo ser facilmente implementados. De um nível (reconstruído) para outro, o processo de interpolação é descrito por

$$x_{k-1}(2n+1) = (x_k(n)+x_k(n+1)) C_1 - (x_k(n-1)+x_k(n+2)) C_2 \quad (E 4.8)$$

$$e x_{k-1}(2n+1) = \hat{x}_{k-1}(2n+1) + L_k(n) \quad (E 4.9)$$

para $k=1,2$ e $n = 0,1,\dots,N/2^k$ sobre uma linha de N pixels.

A equação 4.8 representa a predição por interpolação, ao passo que a equação 4.9 é o passo final do processo de reconstrução somando os erros de interpolação que devem ter sido previamente quantizados e transmitidos.

Para a convolução cúbica $C_1=9/16$ e $C_2=1/16$ e na interpolação linear $C_1=1/2$ e $C_2=0$.

$$x_{k-1}(2n+1) = (x_k(n)+x_k(n+1)) \frac{9}{16} - (x_k(n-1)+x_k(n+2)) \frac{1}{16} + L_k(n) \quad (E 4.10)$$

$$x_{k-1}(2n+1) = (x_k(n)+x_k(n+1)) 1/2 + L_k(n) \quad (E 4.11)$$

Se quisermos melhorar a interpolação, também podemos utilizar interpoladores bidimensionais, mas aqui o evitaremos, pois se isto for aplicado iria tornar ainda mais complexo o circuito auxiliar, exigindo *buffers* e tudo o mais. Outra razão forte para apenas utilizar a interpolação bidimensional, leva em consideração o tipo de dizimação empregado na pirâmide. Como a interpolação é unidimensional, em um mosaico de amostras ortogonalmente dispostas, a linha anterior do campo está duas linhas acima no quadro e o quadro como um todo está dizimado retangularmente. Isto foi discutido na Seção 3.4, só que aqui necessitamos interpolar a amostra $x(a)$ da Figura 3.5a e, como discutido naquela seção, se supormos um filtro separável, a equação (E 3.4) nos leva à conclusão que o interpolador de qualquer forma seria um filtro unidimensional. Este ponto de vista foi reforçado pelas simulações levadas a cabo quando testando o DPCM em Pirâmide. O desempenho do interpolador bidimensional de médias foi quase idêntico ao da interpolação linear e inferior ao da interpolação utilizando a convolução cúbica.

4.2.3) Entropias de Primeira Ordem

Foi dito que o ganho de predição é dado pelo melhor desempenho de preditores interpolativos. Por outro lado, o DPCM convencionalmente atua sobre amostras consecutivas cuja correlação é alta, permitindo predição extrapolativa eficiente. Já com o sinal dizimado, como em $x_2(n)$, esta correlação cai e diminui a acuidade de predição. O desempenho relativo recai sobre a comparação do ganho pela predição interpolativa contra a perda de precisão da extrapolação em sinais sub-amostrados.

Foram feitas algumas simulações, tendo como medidas as entropias de primeira ordem dos sinais envolvidos.

$$p(x=i) = p_i$$

$$H(x) = - \sum_i p_i \log_2(p_i)$$

Primeiramente mediu-se a entropia do sinal $x(n)$. Para comparar as perdas do DPCM, consideramos os sinais diferença $x(n)-x(n-1)$ em ambos casos (DPCM convencional e híbrido) e avaliamos a entropia dos erros de interpolação, utilizando o núcleo discreto da convolução cúbica.

$$H_0 = H [x(n)]$$

$$H_1 = H [x(n)-x(n-1)]$$

$$H_4 = H [x(n)-x(n-4)]$$

$$H_{L1} = H [L_1(n)]$$

$$H_{L0} = H [L_0(n)]$$

Esta comparação leva em conta preditores como em (E 4.4) e interpoladores como em (E 4.10). Devido a taxa média do DPCM em Pirâmide dada por (E 4.1), temos o ganho entrópico (diminuição da entropia) deste sobre o DPCM convencional dado por

$$G = H_1 - \frac{H_4 + H_{L1} + 2 H_{L0}}{4} \quad (E 4.12)$$

Os resultados estão na Tabela 4.1 para as imagens "Praia", "Cozinha", "Sala" e "Zelda". Vemos, nesta tabela, que há uma redução média (ganho) de entropia da ordem de 0.21 bits/pel.

Tabela 4.1 - Levantamentos do ganho de entropia.

IMAGE	H0	H1	H4	HLo	HL1	G
PRAIA	7.33	4.61	5.97	3.49	4.79	0.18
SALA	6.72	3.84	5.05	2.93	3.99	0.12
ZELDA	6.97	3.40	4.93	2.16	2.81	0.38
COZINHA	6.86	3.56	4.90	2.59	3.56	0.15

Devemos ser cautelosos com estes resultados, dado que há uma grande redução de H0 para H1, indicando que talvez as imagens possuam baixo conteúdo de alta frequência. Isto por sua vez não desmerece o crédito destas simulações, já que este fato afeta tanto às medições de H1 e H4, quanto às de HL1 e HL2. Em outras palavras, não há bases concretas para afirmar se este fato beneficia mais as predições extrapolativas ou interpolativas, mas deve-se examinar estes dados com certa prudência.

Suponha que se deseje repetir o experimento para o preditor de (E 4.5), então só devemos modificar dois cálculos.

$$H_1 = H [x(n) - \hat{x}(n)]$$

$$H_4 = H [x_2(n) - \hat{x}_2(n)]$$

Repetiu-se as simulações e os resultados foram apresentados na Tabela 4.2. O ganho médio caiu para 0.03 bits/pel.

Tabela 4.2 - Levantamentos de G, modificando preditor.

IMAGE	H1	H4	G
PRAIA	4.43	5.67	0.09
SALA	3.54	4.51	-0.04
ZELDA	2.85	3.89	0.10
COZINHA	3.21	4.21	-0.03

A codificação nas simulações é suposta entrópica e, portanto, há reconstrução perfeita. Partindo de tal princípio, os erros de quantização possíveis de $L_1(n)$ não afetam a predição interpolativa de $x(n)$ e $L_0(n)$ não incorpora tais erros. Com isto em mente, o que aconteceria se prefiltrássemos o sinal na obtenção de $x_2(n)$ e $x_1(n)$? Já que a reconstrução é perfeita, $L_0(n)$ se encarregaria de reconstruir perfeitamente metade das amostras e a outra metade estaria irremediável e levemente deturpadas. Devido a esta filtragem, as interpolações seriam mais eficientes (eliminando *aliasing*), mas *a priori* haveria erro de reconstrução, independente de quantização. Por incrível que pareça, este padrão de erro com perda de alta frequência em sequências intercaladas no domínio espacial, não se apresentou degradante, subjetivamente, em testes realizados.

Tabela 4.3 - Levantamentos de G com prefiltragem.

IMAGE	H4	HLo	HL1	G
PRAIA	5.84	2.91	4.45	0.58
SALA	4.94	2.28	2.22	0.91
ZELDA	4.86	1.50	2.42	0.83
COZINHA	4.78	1.98	3.19	0.58

Repetimos o experimento anterior com os sinais prefiltrados com filtros otimizados, como na Seção 2.2.3, fixando o Posfiltro como sendo o núcleo de convolução cúbica discreto e utilizando um Prefiltro ótimo de 11 coeficientes. Note que não precisamos reavaliar H_1 , já que não sofreu alterações. Os resultados são apresentados na Tabela 4.3, na qual se obtém o ganho médio crescendo para 0.73 bits/pel.

Das tabelas, pode-se inferir o melhor desempenho relativo do DPCM piramidal sobre o convencional, no que tange extração de redundâncias estatísticas. Também podemos ver que o ganho decai com a maior eficiência do preditor fixando o interpolador. Outro teste poderia se feito se usássemos o preditor de (E 4.6), mas em imagens estáticas, esta predição seria interpolativa ao longo da direção vertical, e não extrapolativa, e não se poderia fazer comparações adequadas.

4.3) TRANSMISSÃO CONDICIONAL DE BLOCOS

Os três níveis podem ser vistos como : $L_2(n)$ traz informação básica da imagem, $L_1(n)$ é o primeiro refinamento e $L_0(n)$ é o refinamento conclusivo, reconstruindo a imagem. Em muitos casos, não é necessário se enviar refinamentos para todas as partes. Isto porque imagens são geralmente compostas por regiões de baixa e alta frequência (bordas e regiões uniformes etc.). Seria interessante medir a possibilidade de se economizar a transmissão destes níveis em algumas regiões nas quais eles fossem redundantes. Por tal motivo, dividiu-se a imagem em blocos de 16×16 pixels e no caso de imagens de 512×512 pixels encontrou-se 32×32 blocos. Em cada um destes poder-se-ia medir a relação sinal-ruído (SNR) da reconstrução de cada bloco, apenas com $L_2(n)$, e notar que $(1-a) \times 1024$ blocos não necessitariam de refinamento adicional, pois já apresentariam SNR acima de determinado valor especificado. Nos $1024 \times a$ blocos restantes somaríamos $L_1(n)$ e re-interpolaríamos. Poderíamos achar que $(1-b) \times a \times 1024$ blocos não necessitariam de maior refinamento, por possuírem SNR acima de determinado valor.

Procedendo da forma descrita, teríamos a taxa média de bits modificando (E 4.1) e incorporando as frequências de blocos que não precisariam de refinamento adicional. Além disto, devemos incluir uma taxa de transmissão lateral (S) para informar ao receptor quais blocos que irão receber quais refinamentos. Esta taxa não seria superior a 2 bits por bloco, o que implica em uma

taxa próxima de 2kb/quadro e não é muito significativa. A taxa média será, então, dada por

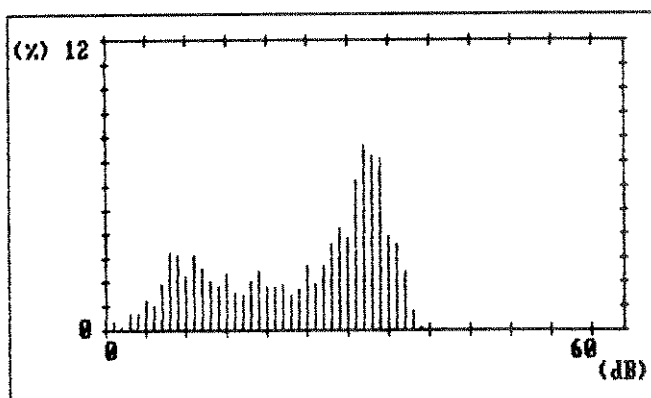
$$R = \frac{\alpha + a \beta + 2 a b \gamma}{4} + S \quad (\text{E 4.13})$$

Para testar a potencialidade do método, procedemos a dizimação seguida de interpolação, retendo 1/2 e 1/4 das amostras. Nas imagens reconstruídas, foi realizada uma análise da densidade de blocos com determinada SNR, para cada caso. Somou-se, então, o número de blocos acima de determinado limiar de SNR, obtendo potencialmente os valores de "a" e "b" indicando a quantidade de blocos que poderiam descartar refinamentos adicionais. A Figura 4.4 mostra alguns gráficos destas densidades, indicando densidade de blocos acima de determinado limiar. Estes resultados nos indicam que é possível pensar em um sistema de multiresolução que possa levar a economias maiores de transmissão, obedecendo certos critérios de fidelidade.

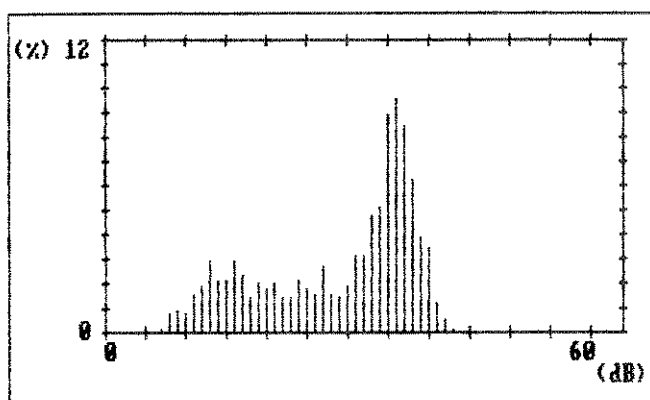
4.4) SIMULAÇÕES

Foram realizadas simulações sobre as 6 imagens de resolução 512x512 (Capítulo 1) disponíveis. Comparou-se os resultados de codificação entre os sistemas DPCM convencional e em pirâmide. Para determinação dos diversos parâmetros envolvidos no projeto dos sistemas empregados nas simulações, foram seguidas as seguintes premissas:

- Técnicas empregadas devem ser extremamente simples, apenas o necessário para uma comparação.
- O mesmo DPCM deve ser aplicado para a imagem inteira e para $x_2(n)$.
- Quantizadores devem ser escalares, simples e fixos. Os resultados devem ser analisados pela entropia de saída destes quantizadores.



Total SNR = 21.8dB
Acima 35dB : 10.8 %



Total SNR = 27.8dB
Acima 35dB : 42.3 %

Figura 4.4 - Densidade de blocos por SNR para imagem GIRL
dizimada e reconstruída. a) Sub-amostragem, fator 4. b)
Sub-amostragem, fator 2. Indica-se montante de blocos com SNR
acima de limiar de 35dB.

O quantizador Q_0 do DPCM deve ter bom desempenho, mas Q_1 e Q_2 (na Figura 4.1) devem possuir menos níveis e valores de reconstrução mais espaçados. Desta forma, explorar-se-á a menor visibilidade de erros nas maiores frequências.

Com esta última diretriz, tentar-se-á reduzir a entropia de $L_1(n)$ e $L_0(n)$ com aumento do erro na reconstrução da imagem. Como se utiliza o mesmo sistema DPCM e quantizadores diferentes, nada é imposto à entropia total (ou taxa de bits). Assim, os sistemas terão taxa e degradação diferentes entre si e não há comparação direta.

Utilizou-se o interpolador de (E 4.10) e o preditor de (E 4.7). Para o quantizador Q_0 do DPCM, escolheu-se um quantizador escalar fixo de 35 níveis [5]. Para os quantizadores de $L_1(n)$ e $L_0(n)$, em primeiro lugar optou-se pelo uso do mesmo quantizador fixo de 35 níveis, para depois lançar mão do projeto otimizado destes em função de pesquisas nas imagens de teste.

Com todos os quantizadores idênticos (de 35 níveis), simulamos o sistema híbrido, comparando-o ao convencional, através da imagem de teste "Girl", a qual julgamos ser a de melhor qualidade de digitalização e que contenha maior significância em altas frequências. Desta forma mediríamos uma imagem que poderia fornecer baixa performance do DPCM híbrido. Os resultados, se não foram dos mais surpreendentes, foram satisfatórios, e ambas medidas de degradação e taxa média de entropia foram praticamente idênticas. A relação sinal ruído ficou em torno de 38.64dB tanto para o DPCM híbrido quanto para o convencional, o mesmo ocorrendo para a entropia dos sinais quantizados, ficando em rigorosamente 3.05 bits/pel. Todavia, ainda poderíamos dispor das outras vantagens sobre o sistema convencional citadas anteriormente. Com os resultados idênticos, demonstra-se, a princípio, que o sistema híbrido é no mínimo comparável ao convencional.

Estes resultados não são suficientemente satisfatórios e partiu-se para otimização dos quantizadores dos erros de interpolação, seguindo as premissas assumidas anteriormente (principalmente a última). Para tal enfrentamos o problema clássico da alocação de bits a diversos quantizadores. Ao contrário dos casos da codificação por transformadas e sub-bandas, que usam os sinais de entrada não quantizados, aqui enfrentamos um problema não linear. Note que o erro de interpolação, antes da quantização por Q_1 , depende do sinal $\bar{x}_2(n)$, decodificado localmente, portanto, de um quantizador não linear embutido ao DPCM. Da mesma forma, o sinal de entrada de Q_2 depende da decodificação local de $x_1(n)$ e, logo, dos outros quantizadores envolvidos. Uma forma interativa pode ser explorada para alocação ótima, mas não existem garantias de sua convergência. Suponha que haja total desconexão entre os erros de quantização e das previsões inter ou extrapolativas. O algoritmo de recursão seria algo como

σ_{L2}^2 , σ_{L1}^2 e σ_{L0}^2 : as variâncias dos erros sem quantização.

σ_{Q0}^2 , σ_{Q1}^2 e σ_{Q2}^2 : as variâncias dos ruídos de quantização.

σ_0^2 , σ_1^2 e σ_2^2 : as variâncias de entrada de cada quantizador.

1)Determine os σ_{Lk}^2 .

2)Estime que os σ_{Qk}^2 sejam nulos, obtendo os σ_k^2 .

3)Faça o algoritmo de alocação desejado para os σ_k^2 .

4)Com quantizadores otimizados, calcule os σ_{Qk}^2 .

5)Some σ_{Qk}^2 aos σ_{Lk}^2 , já que supôs-se desconexão.

6)Obtenha os σ_k^2 .

7)Pare sob algum critério de convergência ou vá ao passo (3).

Este algoritmo aparenta ser por demais complexo para ser feito rotineiramente. De qualquer forma, a melhor maneira de enfrentarmos as não linearidades deste projeto seria a adoção de uma estratégia de quantização adaptativa. O algoritmo deveria ser suficientemente robusto e adequado às características dos sinais envolvidos. Poder-se-ia, até, supor densidades de probabilidade laplacianas, por exemplo, e chavear quantizadores de diferentes número de níveis. Contudo, as formas citadas foram descartadas de nossas simulações por desobedecerem as premissas assumidas para este trabalho em particular.

A outra forma de obter estes quantizadores, baseou-se nas probabilidades de erro medidas em pesquisas, sem quantizadores, e noções intuitivas de erro permissível em torno do zero nas quantizações. A partir destas pesquisas obteve-se as densidades discretas de probabilidade dos sinais a serem quantizados. Seja esta densidade tal que $P(x(n)=i)=p(i)$, $i \in [-255,255]$, $\forall n$, esta densidade. O quantizador é otimizado, no sentido de minimizar o erro médio quadrático introduzido pela quantização, se atendermos as seguintes equações recorrentes, que são extensões discretas das encontradas em [1] [4] :

$$\sum_{e=x_{k-1}}^{x_k} (e-y_k) p(e) = 0 \quad (E 4.14)$$

$$y_{k+1} = x_k + (x_k - y_k) \quad (E 4.15)$$

Os x_k são os níveis limiares de decisão e os y_k os valores de reconstrução. São duas equações recorrentes, mas, para simplificar, adotaremos leis de quantização simétricas. Logo, o intervalo central (em torno do zero) atenderá (E 4.14) e devemos fazer $y_0=0$ e escolher um valor adequado para x_0 . A partir dos valores de y_0 e x_0 e conhecendo a função $p(i)$, a partir das pesquisas, obtém-se a lei desejada do quantizador.

A alocação de quantizadores foi decidida da seguinte forma: Q_0 escolhido arbitrariamente como sendo o quantizador de 35 níveis para luminância de [5], com isto teremos um quantizador eficiente para o DPCM; Q_1 depende de Q_0 e não de Q_2 , de forma que foram feitas análises nas imagens disponíveis, levantou-se $p(i)$ para dado Q_0 e rodou-se o algoritmo do parágrafo prévio, para dado x_0 , deixando-o decidir sobre o número de níveis do quantizador; repetiu-se o procedimento para Q_2 , dados Q_0 e Q_1 encontrados. Desta forma o algoritmo de alocação é ótimo, dada a restrição de escolha de Q_0 e de x_0 , no sentido de minimizar o erro médio de quantização para os níveis $L_0(n)$ e $L_1(n)$. Note, também, que são avaliados momentos de ordem superior à variância, mas não há controle sobre a taxa total de bits, neste algoritmo de alocação.

Assim foram encontrados os quantizadores Q_1 e Q_2 (para $L_1(n), L_0(n)$ respectivamente) com $x_0=1,5$, como tendo as leis descritas na Tabela 4.4 (a) e (b), respectivamente. Com $x_0=0,5$ obteve-se a lei da Tabela 4.4(c) para Q_2 .

Para os sinais de cromaticidade, C_R e C_B , foi aplicado o mesmo DPCM utilizado para luminância, mas guardando a relação 4:2:2 [7]. Q_1 e Q_2 foram projetados de forma similar e são dados na Tabela 4.4 (d) e (e), respectivamente, para C_R e na 4.4 (f) e (g) para C_B . Para cromaticidade ajustamos $x_0=2$ no projeto de Q_2 .

Obteve-se os resultados da Tabela 4.5, na simulação da codificação dos sinais de luminância. Nesta tabela são indicadas as imagens testadas, a entropia total média dos quantizadores e a relação sinal-ruído, para três métodos distintos:

Esquema A - DPCM convencional

Esquema B - DPCM piramidal com Q_2 dado pela Tabela 4.4(b).

Esquema C - DPCM piramidal com Q_2 dado pela Tabela 4.4(c).

Tabela 4.4-Níveis de decisão e reconstrução dos quantizadores

$Q_1 - x_0 = 1,5$

$x_{k-1} - x_k$	y_k
0-1	0
2-6	3
7-14	10
15-25	19
26-43	32
44-255	55

(a)

$Q_2 - x_0 = 1,5$

$x_{k-1} - x_k$	y_k
0-1	0
2-7	3
8-24	12
25-255	27

(b)

$Q_2 - x_0 = 0,5$

$x_{k-1} - x_k$	y_k
0-0	0
1-2	1
3-7	4
8-24	12
25-255	37

(c)

$CR - Q_1$

$x_{k-1} - x_k$	y_k
0-1	0
2-7	3
8-255	12

(d)

$CR - Q_2$

$x_{k-1} - x_k$	y_k
0-1	0
2-62	4
63-255	121

(e)

$CS - Q_1$

$x_{k-1} - x_k$	y_k
0-1	0
2-12	3
13-255	22

(f)

$CS - Q_2$

$x_{k-1} - x_k$	y_k
0-1	0
2-255	4

(g)

Tabela 4.5 - Resultados de codificações.

ESQUEMA	A		B		C	
IMAGEM	Hq(bpp)	SNR(dB)	Hq(bpp)	SNR(dB)	Hq(bpp)	SNR(dB)
ZELDA	2.33	45.4	1.45	42.4	2.00	43.8
COZINHA	2.44	45.2	1.77	42.9	2.26	44.3
SALA	2.69	43.7	2.06	39.3	2.51	39.8
PRAIA	3.15	41.8	2.47	37.3	2.86	37.6
CAMERAMAN	3.03	43.7	2.37	40.3	2.74	40.7
GIRL	3.05	38.7	2.45	34.0	2.80	34.4

Dos dados obtidos, percebe-se que houve alguma economia de transmissão através do esquema híbrido, mas às custas de queda de alguns decibéis na relação sinal-ruído. Contudo, a inspeção das imagens processadas revela que subjetivamente não há diferenças, mesmo prestando atenção aos detalhes. Comparações lado a lado são feitas no apêndice, onde foram segmentadas regiões de cada imagem e colocadas adjacente às imagens originais e as processadas. Não se nota diferenças significativas entre a imagem original e as processadas, sob os esquemas (A), (B) e (C). Este é o ponto principal, o de que conseguiu-se reduções de taxa, mas com degradações que não sensibilizam fortemente o olho humano, dado que as imagens codificadas sob o esquema (B) eram indistinguíveis das originais para todos aqueles que serviram à nossa experiência de observação das imagens, diretamente no monitor.

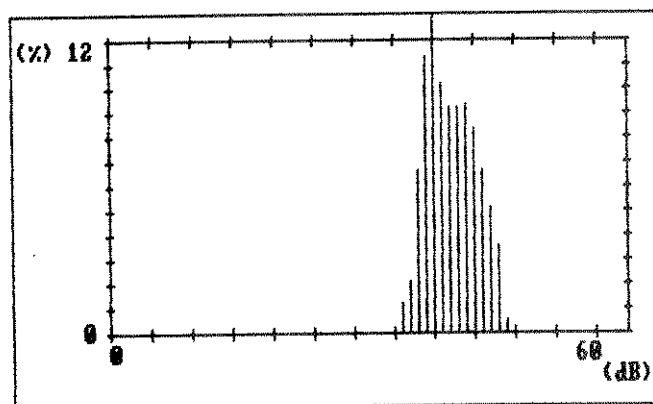
Para os sinais de crominância, utilizou-se apenas a imagem "Zelda", buscando evitar repetitividade. Os resultados são mostrados na Tabela 4.6, novamente realizando comparações. A medida de degradação foi a raiz quadrada do erro médio quadrático (MSE), que representa o desvio médio do erro de codificação em torno do zero. Note que se utilizou os quantizadores da Tabela 4.4, mas com o mesmo DPCM usado na codificação do sinal de luminância. Ainda, note que a correspondência entre os sinais atendida foi 4:2:2, de forma que C_r e C_b possuem metade das amostras em relação ao sinal Y . Desta feita, para medir a entropia final da codificação da imagem, devemos somar os valores da Tabela 4.5 com a metade dos valores da Tabela 4.6.

Tabela 4.6 - Imagem ZELDA

COMPONENTE	CONVENCIONAL		PIRÂMIDE	
	Hq (bpp)	DESVIO	Hq (bpp)	DESVIO
C_r	1.90	0.443	1.21	0.943
C_b	1.84	0.426	1.12	0.896

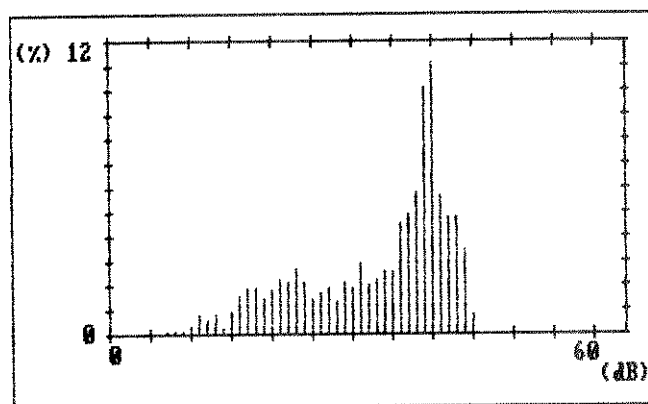
Os resultados desta tabela indicam grande economia de taxa de bits, novamente com aumento do erro, mas desta vez com maiores vantagens ainda. O desvio médio dobrou, mas ainda ficou abaixo da unidade, ou seja, não houve erro médio superior a um nível de quantização. Levando em conta que há sempre um arredondamento de resultados para 256 níveis, o erro introduzido se compara a tais erros de quantização. Para reforçar este ponto de vista, foi percorrido o caminho inverso do descrito para obtenção das componentes, obtendo os sinais RGB a partir do conjunto YCrCb. Não se notou degradação das componentes individualmente, comparando-as aos sinais originais. Outro ponto relevante diz respeito aos erro de interpolação nestes sinais. Por serem muito próximos ao zero seria possível utilizar técnicas de *run-length* e devido ao baixo número de níveis empregados na quantização escalar, poder-se-ia usar eficientemente quantização vetorial.

Por fim, histogramas similares aos descritos na Seção 4.3, foram empregados ao sinais de luminância das imagens "Girl" e "Zelda" comparando as imagens original e codificada pelo esquema (B) (Figura 4.5). Desta vez, utilizou-se também uma ponderação para o erro. Tentando explorar a menor sensibilidade visual humana a erros de alta frequência, filtrou-se o erro com um filtro bidimensional passa-baixas com atenuação bem suave, para atenuar as componentes expúrias de alta frequência. Repetiu-se, então, as avaliações de relações sinal-ruído, mas com erros ponderados, às imagens "Zelda", "Girl" e "Cameraman" (Figura 4.6). De uma forma ampla, não há muita diferença entre os histogramas para diferentes métodos aplicados a uma mesma imagem.



Total SNR = 42.4dB
Acima 40dB : 78.4 %

(a)

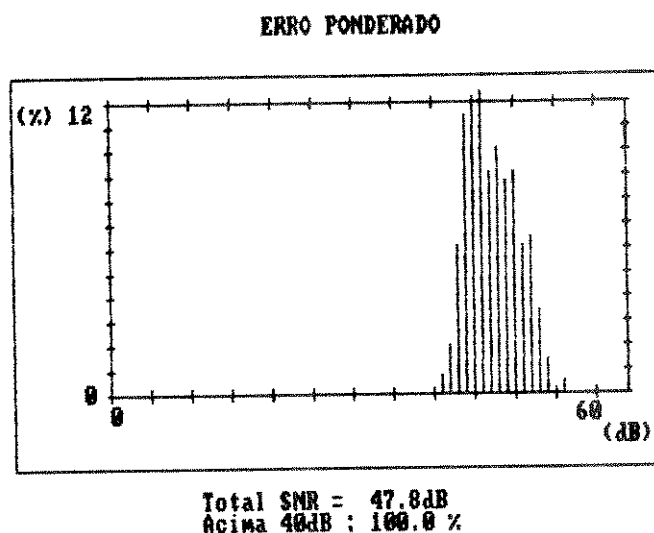


Total SNR = 34.0dB
Acima 40dB : 30.6 %

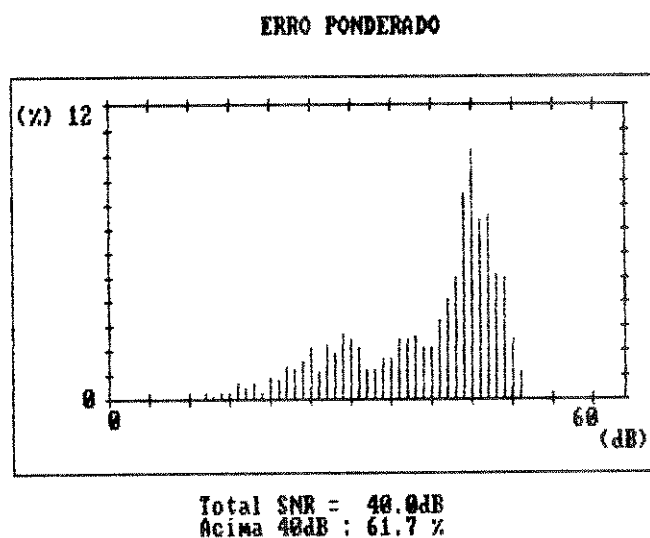
(b)

Figura 4.7 - Densidade de blocos com determinada SNR para o esquema DPCM piramidal (B), na codificação das imagens (a) "Zelda" e (b) "Girl" .

(a)



(b)



(c)

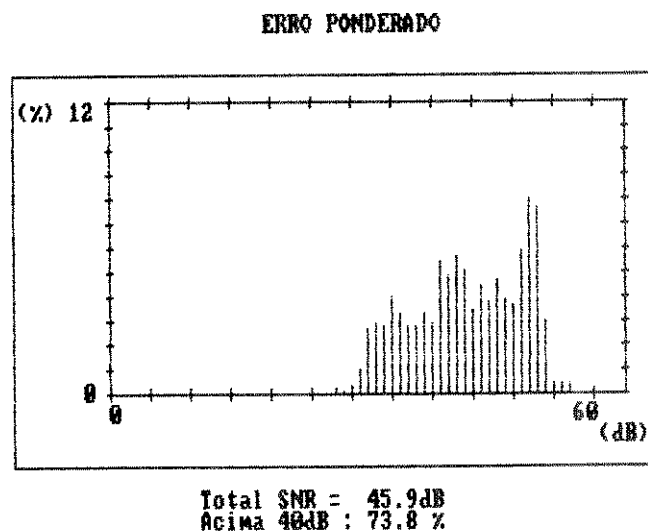


Figura 4.8 - Reaplicação do método da figura anterior, com erro ponderado, às imagens (a) "Zelda", (b) "Girl" e (c) "Cameramen".

4.5) CONSIDERAÇÕES ADICIONAIS

Com vistas a uma implementação do DPCM em Pirâmide de 3 níveis, alguns fatores devem ser levados em conta de forma a torná-lo eficiente e aprimorado, explorando melhor suas características.

Vários pontos do sistema descrito na Figura 4.1 poderiam ser alterados de forma a torná-lo um sistema adaptativo. Adaptação poderia ser incorporada a :

- Preditor do DPCM, com técnicas mais eficientes.
- Quantizador do DPCM.
- Quantizadores Q_1 e Q_2 .
- Interpoladores.
- Fator de sub-amostragem em cada passo.

Com os quantizadores adaptativos, certamente a eficiência aumentará, inclusive se levarmos em conta as diferentes características de cada sinal sobre os quais tais dispositivos atuam. Quanto ao interpolador, sua adaptação em termos de coeficientes, não deve levar a grandes benefícios, mas grande ferramenta seria um método chaveado, no qual pudesse se decidir sobre a forma de interpolação a ser empregada. Esta forma de implementação é comum em preditores extrapolativos, e para a interpolação poder-se-ia explorar interpolação intercampo, bidimensional, temporal, horizontal, vertical, tridimensional ou outras formas híbridas. Ainda sobre a interpolação, já discutimos o fato de não ser eficiente, a princípio, uma interpolação fixa bidimensional, mas para os sinais de crominância, podemos dentro do padrão 4:2:2, empregar dizimação hexagonal. Como visto no Capítulo 3, seria possível encontrar interpoladores bidimensionais simples e eficientes para o caso em questão. Mesmo um simples interpolador planar levaria a grandes resultados. Por fim, seria possível e aconselhável explorar quantização vetorial para $L_0(n)$ e $L_1(n)$, assim como estimação de movimento no processo de interpolação.

Para proteção contra erros de canal, é necessário um estudo intenso para uma transmissão confiável. Contudo, algumas diretivas seriam úteis. Em primeiro lugar, para erros longos, seria melhor adotar segmentações em quadros. Com isto, a informação seria apenas substituída por um padrão qualquer até que se obtenha um reset, recuperando a sincronia. O ponto principal, em realidade, é que no DPCM em pirâmide, possuímos uma vantagem a mais, não citada, sobre o sistema convencional, que reside no fato que a informação importante está apenas em $L_2(n)$. Erros ocasionais em $L_1(n)$ ou $L_0(n)$ não seriam muito sensibilizadores ao observador, logo devemos proteger $L_2(n)$ fortemente contra erros de canal, mas não necessariamente os outros níveis. Por exemplo, quando encontrássemos erro, nestes níveis, os omitiríamos, como na transmissão condicional por blocos, sendo penalizados apenas com uma pior definição de altas frequências naquela região.

4.6) REFERÊNCIAS

- [1] N.S.Jayant, P.Noll, *Digital Coding of Waveforms*, Englewood Cliffs, NJ, 1984.
- [2] M.L. Sethia, J.B.Anderson, *Interpolative DPCM*, IEEE Trans. on Communications, vol com-32, no 4, pp 729-736, June 1984.
- [3] S.Handa, H.Tanaka, *Two Dimensional Interpolative DPCM*, Elect. and Commun. in Japan, Part 1, vol 71, no 5, pp 116-123, 1988.
- [4] W.K.Pratt, *Image Transmission Techniques*, Academic Press, 1979.
- [5] CCIR Study Groups : Doc. CMTT/45E, July 1987.
- [6] H.Madhoud, D.Thomas, *Two Dimensional Adaptive LMS Algorithm*, IEEE Trans. on Circuits and Systems, CAS-35, May, 1988.
- [7] CCIR Rec. 601-1, vol. XI, 1982-1986, pp 319-328.

CAPÍTULO 5

HIERARQUIA UTILIZANDO TRANSFORMADAS SOBREPOSTAS (LT's)

Este capítulo tem como principal alvo a descrição de uma estratégia eficiente para transmissão progressiva de imagens que utilizam bancos de filtros e transformadas piramidais ou não. Não se realizou simulações, mas apenas descreveu-se métodos e critérios. Este assunto está sendo atualmente enfatizadamente estudado e novos resultados surgem continuamente e devem ser apresentados em oportunidades futuras.

5.1) BANCOS DE FILTROS E LT's

Quando se é necessário separar um sinal em bandas poderemos adotar uma estratégia como a da Figura 5.1 na qual o sinal é dividido em M bandas de frequência através de um banco de filtros [1]. Aqui trataremos o caso no qual o sinal é dividido em M bandas iguais e os filtros tenham frequências de corte em múltiplos de π/M . Desta forma os sinais denotado por $x_i(n)$ ($i=0, 1, 2, \dots, M-1$) seriam sinais passa-faixa de faixas idênticas e de largura π/M . Tendo reduzido a faixa destes sinais podemos dizimá-los de acordo com [1], sem perda de informação, se os filtros forem ideais. O conjunto de filtros que fazem a separação de tal tipo recebe a denominação de bancos de filtros de análise com dizimação máxima.

Para recuperar o sinal $x(n)$ a partir dos sinais $x_i(n)$ [1], primeiro inserimos $M-1$ amostras nulas entre amostras destes e executamos uma filtragem com um passa-faixa em cada sinal. O resultado de cada uma destas filtrações conterá uma determinada banda do sinal original de forma que a soma destes sinais recomporá $x(n)$ (assumindo, novamente, filtros ideais). O conjunto destes filtros recebe a denominação de banco de filtros de síntese para dizimação máxima.

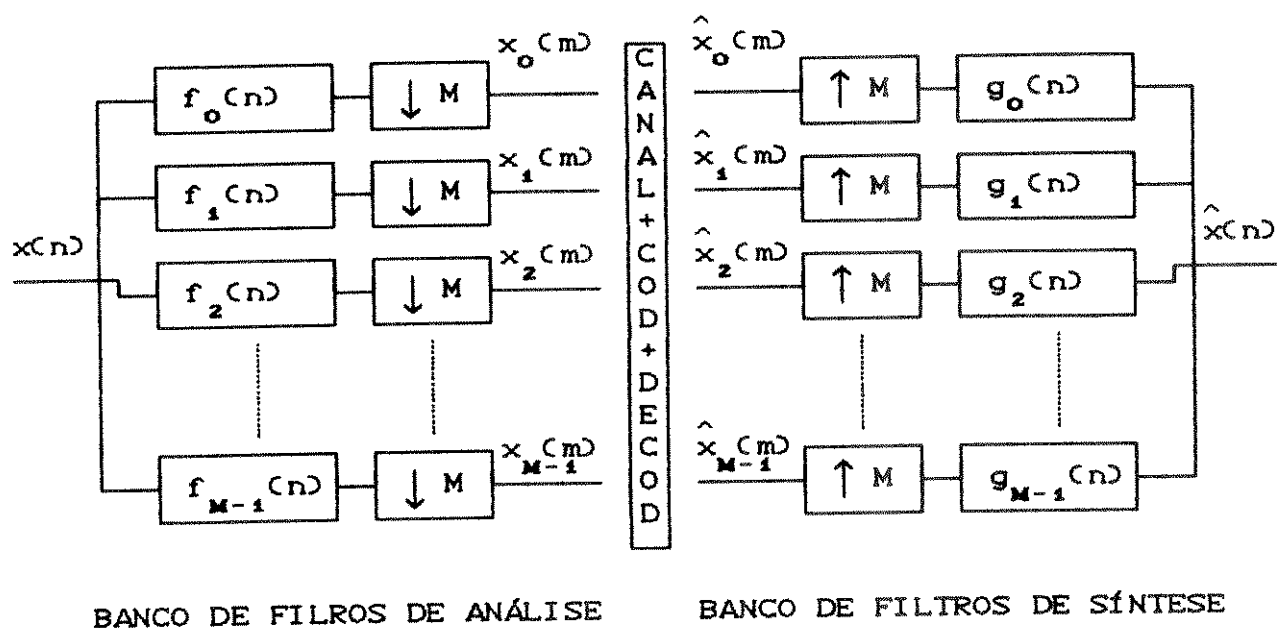


Figura 5.1 - Bancos de filtros de análise e síntese maximamente dizimados.

Na Figura 5.1 são mostrados os bancos de filtros de análise e síntese. Quando são colocados *costa-a-costa* como na figura referida, os sinais do banco de síntese podem ser processados ou codificados e sofrerem posterior agrupamento para reconstrução do sinal processado em banda completa. Um típico exemplo é o da codificação em Sub-bandas [2]. Se os filtros forem ideais e $\hat{x}_1(n) = x_1(n)$, então teremos garantidamente $\hat{x}(n) = x(n)$. De uma forma mais geral, não é necessário se exigir filtros ideais, mas se eles obedecerem determinados critérios [3], também poderemos obter reconstrução perfeita para o sistema. Utilizando filtros causais, $\hat{x}(n)$ seria uma réplica atrasada de $x(n)$. Outra consideração importante diz respeito às fases temporais relativas dos sub-amostradores e super-amostradores. Supomos aqui que estas operações são realizadas simultânea e sincronamente. Isto nos permite formular nova interpretação para o sistema da Figura 5.1.

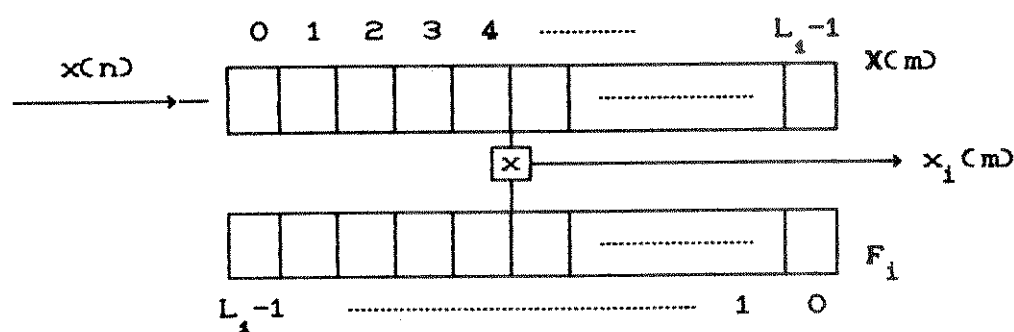


Figura 5.2 - Ilustração vetorial da filtragem subamostrada de cada ramo de análise.

Para a análise, suponha que as respostas impulsivas $f_i(n)$ tenham comprimento finito (FIR) máximo de L_1 elementos ($L_1 \geq MD$ sem perda de generalidade, já que podemos preencher as respostas menores com zeros sem alterá-la. Com isto, e supondo a dizimação sincronizada a cada M instantes de amostragem, não é difícil ver que cada ramo de análise pode ser revisto como sendo tal qual na Figura 5.2. Nesta, as amostras são agrupadas em um vetor de comprimento L_1 que se modifica a cada M amostras originais de uma vez só, como um registrador de deslocamentos de passo M . Estas amostras são ponderadas pela janela de análise dada pelo filtro $f_i(n)$, obtendo o sinal $x_i(m)$.

Em notação vetorial, agrupando as amostras de $f_i(n)$ no vetor F_i , teremos

$$x_i(m) = F_i^T X(m) \quad (E 5.1)$$

Após a obtenção das amostras para cada ramo, novo vetor $X(m)$ é formado pelo simples deslocamento do registrador, de M posições à direita, incorporando M novas amostras de $x(n)$. As M amostras mais à direita são descartadas em consequência deste deslocamento. O índice m deste vetor indica o número destes deslocamentos e equivale ao índice temporal das sequências sub-amostradas. Os filtros estão ordenados em ordem reversa para reproduzir a convolução, fazendo o produto interno dos vetores.

Para a síntese, designemos $\hat{Y}(m)$ como vetor $M \times 1$ que contenha os elementos dos $\hat{x}_i(m)$, para dado m , e designemos G_j como vetor que contenha os j -ésimos coeficientes (em ordem reversa) dos filtros do banco de síntese. Simultaneamente, as amostras dos $x_i(m)$ são multiplicadas pelos coeficientes de alguns G_j , apenas. Isto porquê, para interpolação, são inseridas amostras nulas entre amostras consecutivas dos sinais $x_i(m)$.

De forma mais geral, podemos, ainda, supor que os filtros de síntese tenham comprimento máximo L_2 e que L_2 seja múltiplo de M . Se for necessário, pode-se preencher as respostas impulsivas dos filtros com zeros até se completar tal valor (L_2). As operações de síntese podem ser representadas como

$$x(mM+i) = \sum_{k=1}^{L_2/M} G_{kM-i}^T \hat{Y}(m-1+k) \quad i=0,1,\dots,M-1 \quad (E 5.2)$$

Para estabelecer paralelos com transformadas ortogonais lembremo-nos que [2] sendo X um vetor de dados reais e A uma matriz quadrada de transformação com mesma dimensões da de X , temos

$$Y = A^T X \quad (\text{transformação do vetor } X) \quad (E 5.3)$$

Para que A seja ortogonal $A^T = A^{-1}$ e X pode ser recuperado por

$$X = B Y = [A^T]^{-1} Y = A Y \quad (E 5.4)$$

As linhas de A^T são bases ortonormais da transformada.

$$A = [b_0 \ b_1 \ b_2 \ \dots \ b_{M-1}] \quad (E 5.5)$$

Voltando ao caso dos bancos de filtros para análise e síntese, reexaminemos (E 5.1). Generalizando para qualquer i (varrendo todas as M bandas) podemos rearranjar as M equações em uma única forma matricial.

$$Y(m) = F^T X(m) \quad (E 5.6)$$

$$F = [F_0 \ F_1 \ F_2 \ \dots \ F_{M-1}] \quad (E 5.7)$$

F é uma matriz $L \times M$ e pode ser encarada como uma transformação que leva o vetor $X(m)$ de dimensão M em um sub-espço M -dimensional de $\mathbb{R}^{L1}$ gerado pelas bases de F_1 .

De forma similar, $\hat{x}(n)$, em (E 5.2), pode ser colocado em uma operação envolvendo matrizes. Generalizando, temos

$$\hat{X}(m) = \sum_{k=1}^{L2/M} G \hat{Y}(m-1+k) = \sum_{k=1}^{L2/M} \Phi_{k,m} \quad (E 5.8)$$

$$\hat{X}(m)^T = [x(mM) \ x(mM+1) \ \dots \ x(mM+M-1)]$$

$$G = [G_0 \ G_1 \ G_2 \ \dots \ G_{M-1}] \quad (E 5.9)$$

$$\Phi_{k,m} = G \hat{Y}(m-1+k) \quad (E 5.10)$$

G é uma matriz $M \times L2$ cujas linhas são os filtros de síntese. $\hat{X}(m)$ é um vetor $M \times 1$ do bloco reconstruído que é obtido pela superposição dos vetores $\Phi_{k,m}$.

Podemos considerar, então, tal processo de separação de bandas, como transformadas que se estendem por blocos vizinhos. Compare (E 5.3) com (E 5.6), (E 5.5) com (E 5.7) e (E 5.4) com (E 5.8) e (E 5.10). Teremos uma analogia mais perfeita se fizermos $L_1=L_2=L$, sendo L um múltiplo inteiro ou não de M . Novamente, não há perda de generalidade, já que, como comentado, podemos

preencher as respostas impulsivas com zeros. Por outro lado, transformadas convencionais podem ser encaradas como bancos de filtros com comprimentos M , iguais para todos os filtros. Os vetores $\hat{x}_{k,m}$ são como resultados de transformadas sobrepostas inversas aplicadas aos vetores $Y(m)$, que são produtos de transformadas sobrepostas aplicadas aos vetores $X(m)$. Reconstrução perfeita do sinal é possível, mas não pela pura transformação inversa, e sim pela superposição dos vetores resultantes. Isto é fácil e intuitivamente explicado [4] se pensarmos que de L amostras poderíamos obter L pontos de sua transformada, mas tomamos apenas M destes pontos. Com isto há *aliasing* incorporado, que se apresenta como *aliasing* temporal [1] após volta ao domínio temporal.

Reconstrução perfeita é possível se cancelarmos este efeito apenas no domínio temporal (ou espacial) [5]. Em [6] isto é feito a partir de bancos modulados, nos quais todos os filtros são encontrados a partir de um protótipo passa-baixas através de modulação. Além disto, reconstrução perfeita é possível se obedecermos as condições impostas em [3] e algumas técnicas existem [5] [7] [8] que exibem tais propriedades. Um breve (muito breve, mesmo) resumo dos trabalhos na área pode ser encontrado em [9]. Contudo, comparando as equações 5.4 a 5.10, uma analogia mais exata pode ser encontrada se fizermos $F=G^T$ e encontraremos uma classe de bancos de filtros de comprimento L conhecidos como Transformadas Sobrepostas (*Lapped Transforms - LT's*). Para $L=2M$, existe uma transformada que permite reconstrução perfeita e boa eficiência computacional [10], conhecida como a *Lapped Orthogonal Transform* (LOT).

Os mesmos princípios podem ser aplicados para uma estrutura em árvore. Neste tipo de estrutura hierárquica, os ramos passa-faixa podem ser acoplados a outros bancos de análise. Desta forma, pode-se dividir ainda mais o espectro do sinal e conectar bancos de análise, em árvore, na forma que mais conveniente for. Similarmente, os bancos de síntese devem ser conectados em árvore de forma simétrica às conexões de bancos no sistema de análise.

5.2) HIERARQUIA EM ÁRVORE OU DECOMPOSIÇÃO DIRETA ?

Na seção precedente, partindo de uma análise sob o aspecto de convolução com núcleos, relacionamos a separação em sub-bandas do espectro com transformadas sobrepostas generalizadas. A decomposição do espectro foi realizada de forma direta, em M bandas de larguras idênticas do espectro original. No caso particular de $M=2$, teríamos dois filtros de meia-banda e cada banco de análise proporcionaria um ramo passa-altas e outro passa-baixas. Como já discutido no Capítulo 3, tal banco guardaria grande semelhança com o banco da decomposição piramidal. Àquela altura, foi citado o fato que poderíamos sub-amostrar o ramo passa-baixas, também, caso fossem atendidas certas condições para os filtros de meia-banda. Porém considerou-se tal processo como distinto, reservando capítulo posterior para tal discussão.

Retomando aquele ponto de vista, podemos considerar o caso da dizimação do ramo passa-altas como o banco de filtros com dizimação máxima (ou LD e $M=2$). Para a construção da pirâmide, seria necessária a conexão destes bancos em árvore sempre conectando-os ao ramo passa-baixas antecessor, como mostrado na Figura 5.3 para a análise. Para a decomposição bidimensional do espectro de uma imagem poderíamos realizar o processo ao longo das linhas e, em seguida, das colunas de pixels da imagem [11]. A decomposição seria, então, como na Figura 5.4a para 3 bancos conectados (apenas apresentando as frequências positivas horizontal e vertical).

A decomposição ilustrada nas Figuras 5.3 e 5.4a pode ser prolongada até um número total de passos que ainda permita dizimação do sinal. Por exemplo, se a imagem contiver 512×512 pels, não será possível conectar mais do que 9 bancos em árvore, para cada direção. A tal processo de decomposição do espectro nos referiremos como decomposição de meia-banda hierárquica, ou, simplesmente, hierárquica.

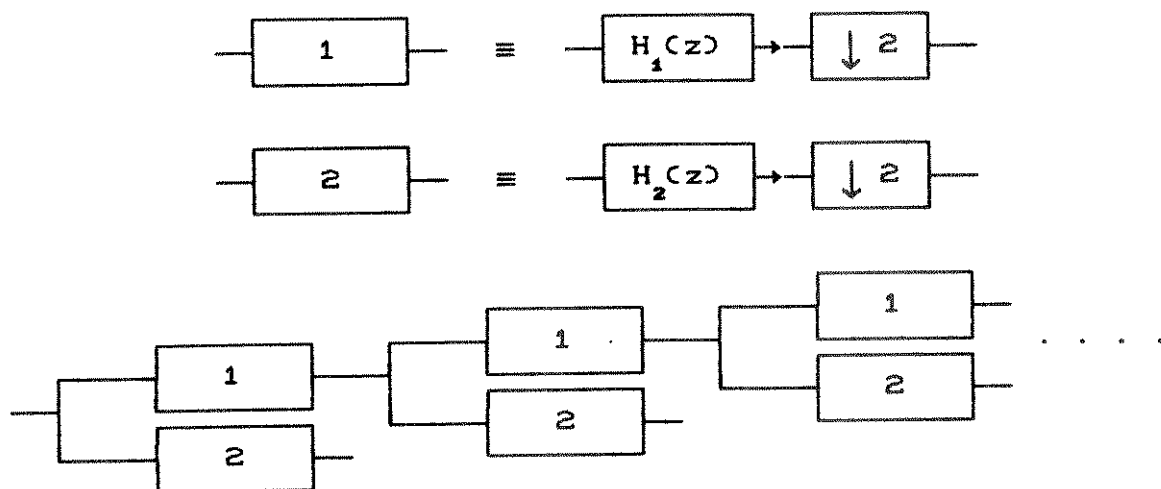


Figura 5.3 - Conexão hierárquica em árvore de bancos de filtros de meia-banda.

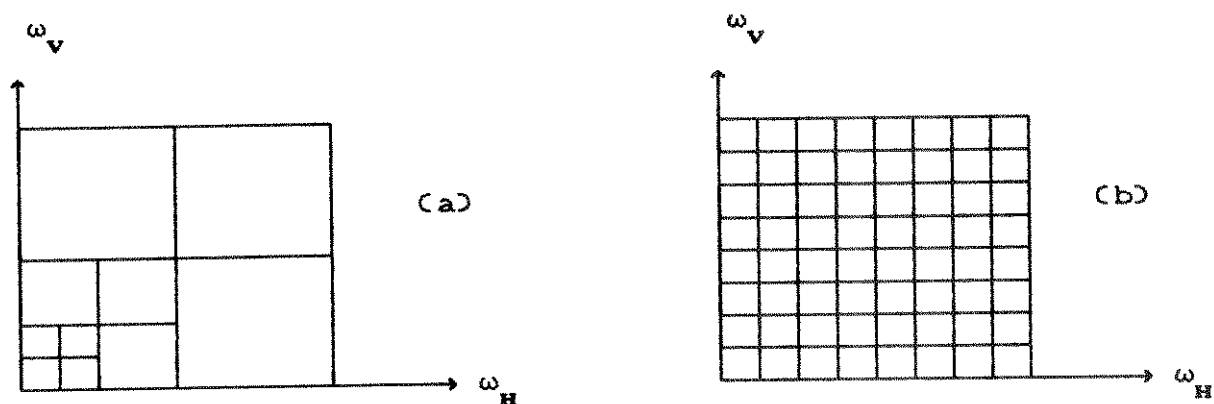


Figura 5.4 - Decomposição do espectro, supondo idealização de filtros e apresentando regiões de frequências positivas apenas.

a) Hierárquica com 3 filtros de meia banda. b) Direta com $M=8$.

No caso $M=2^n$, com apenas um banco, iremos dizer que esta é uma decomposição direta. Para o caso $M=8$, ilustra-se a decomposição do espectro na Figura 5.4b.

O caso da decomposição hierárquica apresenta grandes vantagens, como a de permitir uma menor resolução e maior seleção da banda de menor frequência (banda básica). Isto devido ao fato que, para dividir o espectro em muitas bandas diretamente, iríamos necessitar utilizar blocos muito grandes que seriam desvantajosos face a limitação do tamanho da imagem. Transformadas não sobrepostas, como a DCT ou a DFT, não sofrem muito destes problemas por assumirem repetições periódicas dos sinais a serem transformados, todavia sendo menos eficientes. A limitação citada causa o que aqui é chamado de efeito de bordas (da imagem e não dos blocos). Por outro lado, existem dois atenuantes pouco comentados na literatura. Em primeiro lugar, com decomposição hierárquica, nos bancos superiores onde a resolução é muito baixa, iremos nos defrontar com os mesmos efeitos de bordas e a sobreposição perderia sentido. Por exemplo, no nono banco de uma linha de 512 amostras, teríamos duas amostras a serem filtradas. Até onde eu possa enxergar, o filtro passa-baixas deveria calcular não mais que a média e o passa-altas a diferença entre elas. Isto reduz a eficiência e necessidade de transformadas sobrepostas nestas baixas resoluções. Outro ponto pouco discutido diz respeito à conexão em árvore de LT's de M bandas ($M \geq 2$). Com estas conexões poderíamos conseguir diversas combinações de transformadas e valores de L e M que permitissem atingir resoluções mínimas.

Um caso geral de decomposição hierárquica seria o das Transformadas Hierárquicas Sobrepostas (HLT) [12], conectando-as como na Figura 5.3 e utilizando uma transformada qualquer em cada banco. A pirâmide de diferenças reduzida [13] do Capítulo 3 é, na realidade, exemplo de transformada hierárquica não-sobreposta cuja matriz de transformação de cada banco (comparada a HLT) é dada por $\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \times \frac{1}{2}$. Já o caso de [14], como comentado, é uma fusão de decomposição direta através de DCT's com decomposição hierárquica, através desta pirâmide reduzida (ou transformada hierárquica não sobreposta). Outra relação de decomposição hierárquica é apresentada em [20], desta vez com a Transformada Wavelet Discreta.

Não entrando em detalhes, iremos apenas comparar sistemas hierárquicos (com número restrito de bancos) ao método direto, de forma que ambos atinjam a mesma resolução e faixa para a banda básica. Assim, podemos diretamente comparar as figuras 5.4(a) e 5.4(b). Neste caso é visível que a decomposição direta analisará o sinal de forma mais detalhada. Além disto, utilizando a LOT [10], MLT [15] ou LT's extendidas [16] será possível utilizar algoritmos rápidos para a decomposição direta do espectro. Por estes motivos, iremos em seguida delinear um método para transmissão progressiva baseado em LT's com decomposição direta, em detrimento da hierárquica, como utilizada em [17].

5.3) TRANSMISSÃO PROGRESSIVA COM DECOMPOSIÇÃO DIRETA.

Como primeiro passo, deve-se pensar na codificação da imagem como um todo, de forma que ao fim da transmissão obtenha-se a distorção provocada por esta codificação. Pode-se, inclusive, dimensioná-la para permitir recuperação sem erros do sinal, se for desejado. Iremos aplicar LT's ao longo das direções horizontal e vertical e de cada bloco de L^2 amostras obteremos M^2 coeficientes. Cada coeficiente corresponde ao sinal filtrado da banda correspondente com frequência de amostragem M^2 vezes menor (dizimado). Como em codificação por transformadas, será necessário alocar o número necessário de bits para cada coeficiente [2].

5.3.1) Alocação de Quantizadores.

Possuindo M^2 coeficientes (C_{ij} ; $0 \leq i, j \leq M-1$), a variância de cada coeficiente depende da estatística do sinal e do tipo de transformada empregada.

$$\sigma_{ij}^2 = E [C_{ij}^2]$$

Sendo $h_i(n)$ a i -ésima base (filtro) da transformada (banco) e X_{mn} as amostras do bloco em processamento, temos :

$$Y = A X A^T \quad (\text{transformada separável})$$

$$Y = [C_{ij}] ; \quad X = [X_{ij}] ; \quad A = \begin{bmatrix} H_0^T \\ \vdots \\ H_{M-1}^T \end{bmatrix} ; \quad H_i^T = [h_i(0), \dots, h_i(L-1)]$$

$$C_{ij} = \sum_m h_i(m) \sum_n X_{mn} h_j(n) = \sum_m \sum_n h_i(m) h_j(n) X_{mn}$$

$$\begin{aligned} \sigma_{ij}^2 &= E \left[\left(\sum_m \sum_n h_i(m) h_j(n) X_{mn} \right) \left(\sum_k \sum_l h_i(k) h_j(l) X_{kl} \right) \right] = \\ &= \sum_m \sum_n \sum_k \sum_l h_i(m) h_i(k) h_j(n) h_j(l) E [X_{mn} X_{kl}] \end{aligned}$$

$$0 \leq m, n, k, l \leq L-1$$

Mas $E [X_{mn} X_{kl}] = r_x^{(m-k, n-l)}$ = função de autocorrelação bidimensional da imagem. Supondo a imagem isotrópica, amostrada em uma grade retangular equiespaçada nas direções horizontal e vertical e assumindo-a como um processo Gauss-Markov com $\rho=0.95$ em cada direção, chega-se a

$$E [X_{mn} X_{kl}] = r_x^{(m-k, n-l)} = 0.95^{\sqrt{(m-k)^2 + (n-l)^2}} \quad (E 5.11)$$

$$\sigma_{ij}^2 = \sum_m \sum_n \sum_k \sum_l h_i(m) h_i(k) h_j(n) h_j(l) r_x^{(m-k, n-l)} \quad (E 5.12)$$

De posse dos valores das variâncias, após algumas aproximações razoáveis, encontra-se algoritmo sub-ótimo para alocação de quantizadores [2] [17].

$$\tilde{b}_{ij} = R + \frac{1}{2} \log_2 \frac{\sigma_{ij}^2}{S} \quad (E 5.13)$$

R = taxa total requerida

$\tilde{b}_{ij}$ = número de bits alocado ao coeficiente C_{ij}

$$S = \left[\prod_{i,j} \sigma_{ij}^2 \right]^{1/M^2}$$

Arredondando $\tilde{b}_{ij}$ para algum valor inteiro, encontramos b_{ij} como o número inteiro de bits para cada quantizador. Obviamente, podemos encontrar não mais R , mas $\tilde{R} \approx R$ como taxa média. Em um processo iterativo, pode-se encontrar um compromisso entre os b_{ij} inteiros e R desejado [2]. A Tabela 5.1 apresenta os valores das variâncias dos coeficientes para a LOT com $M=8$ e $L=16$, bem como os b_{ij} para algumas taxas R .

Tabela 5.1 (a) - Variância dos coeficientes

	$\xrightarrow{j}$							
$\downarrow i$	53.733	3.1852	0.3886	0.2456	0.0737	0.0771	0.0356	0.0391
	3.1852	1.0767	0.2976	0.1967	0.0763	0.0716	0.0397	0.0408
	0.3886	0.2976	0.1423	0.1164	0.0539	0.0526	0.0317	0.0326
	0.2456	0.1967	0.1164	0.1124	0.0595	0.0623	0.0398	0.0415
	0.0737	0.0763	0.0539	0.0595	0.0351	0.0392	0.0264	0.0280
	0.0771	0.0716	0.0526	0.0623	0.0392	0.0460	0.0321	0.0344
	0.0356	0.0397	0.0317	0.0398	0.0264	0.0321	0.0229	0.0249
	0.0391	0.0408	0.0326	0.0415	0.0280	0.0344	0.0249	0.0271

Tabela 5.1 (b) - Algumas alocações e taxas totais.

9 7 5 5 4 4 3 4	8 6 4 4 3 3 2 3	8 5 3 3 2 2 1 2
7 6 5 5 4 4 4 4	6 5 4 4 3 3 3 3	5 4 3 3 2 2 2 2
5 5 4 4 4 4 3 3	4 4 3 3 3 3 2 2	3 3 2 2 2 2 1 1
5 5 4 4 4 4 4 4	4 4 3 3 3 3 3 3	3 3 2 2 2 2 2 2
4 4 4 4 3 4 3 3	3 3 3 3 2 3 2 2	2 2 2 2 1 2 1 1
4 4 4 4 4 4 3 3	3 3 3 3 3 3 2 2	2 2 2 2 2 2 1 1
3 4 3 4 3 3 3 3	2 3 2 3 2 2 2 2	1 2 1 2 1 1 1 1
4 4 3 4 3 3 3 3	3 3 2 3 2 2 2 2	2 2 1 2 1 1 1 1
$R \approx 4b/pel$	$R \approx 3b/pel$	$R \approx 2b/pel$

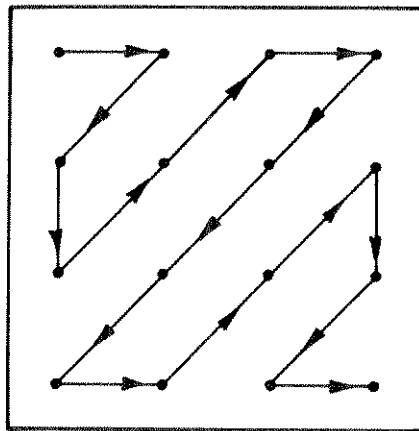


Figura 5.5 - Sequência ZIG-ZAG para transmissão sequencial dos coeficientes em um bloco transformado.

5.3.2) Critério de Transmissão Progressiva.

Dispondo os C_{ij} bidimensionalmente, poderíamos atingir um bom critério para transmissão progressiva destes coeficientes se os enviássemos hierarquicamente um a um, como, por exemplo, em ZIG-ZAG (Figura 5.5). Para a DCT este método se aproxima do ótimo, no sentido de maximizar a relação refinamento da imagem por informação transmitida, e é bastante robusto [18]. Com b_{oo}/M^2 bits/pixel poderíamos enviar a banda básica e em seguida progressivamente iríamos refinando a imagem resultante. Outra forma de se incrementar o número de passos na transmissão seria possível se transmitíssemos os bits dos coeficientes progressivamente, do mais para o menos significativo. Para melhor

explicar, suponha que ordenemos os coeficientes na direção da varredura ZIG-ZAG. Os bits menos significativos formariam um plano de bits, o mesmo ocorrendo com o segundo menos significativo, e assim por diante. As dimensões destes planos diminuiriam com aumento da significância dos bits contidos. Logo poderíamos enviar estes planos progressivamente como ilustrado na Figura 5.6. Nesta figura são indicados sentidos de percurso no envio dos coeficientes, varrendo os planos de bits.

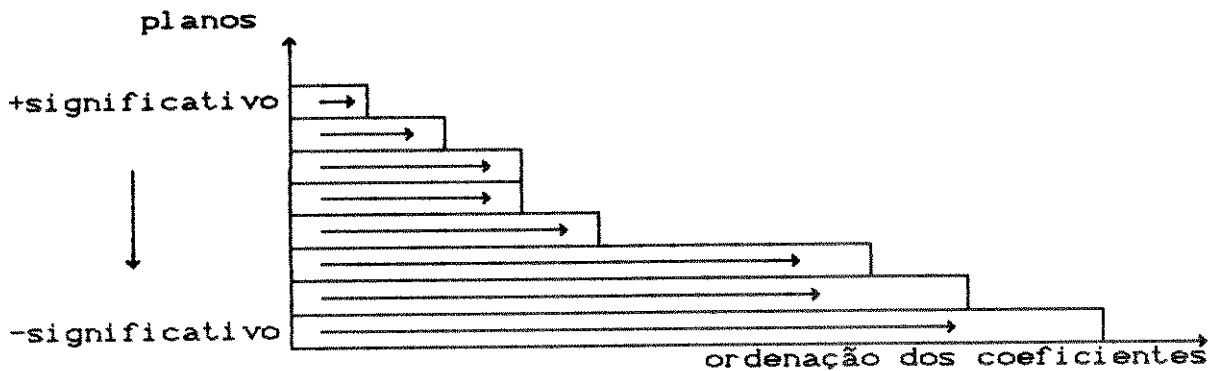


Figura 5.6 - Varredura dos coeficientes e planos de bits.

Adicionalmente, é frequentemente apontada na literatura a conveniência de se quantizar uniformemente a banda básica (C_{00}), mesmo que se aplique outros métodos de extração de redundâncias entre blocos, posteriormente. Para os outros coeficientes a distribuição laplaciana é apontada como modelo de distribuição mais adequado aos coeficientes da DCT. O mesmo aplicaremos para LOT no projeto otimizado dos quantizadores. Em [2] pode se encontrar diversas tabelas de quantizadores otimizados para vários números de bits. Como a transmissão é progressiva podemos sempre utilizar o quantizador otimizado para o número de bits acumulado até o instante para determinado coeficiente. Sendo $B_0, B_1 \dots B_n$ a sequência de bits deste coeficiente e $\lambda_k(i)$ o valor de reconstrução do nível i de um quantizador otimizado para k bits, em cada plano teríamos

$$\lambda_k(i) = \lambda_k(B_0 B_1 \dots B_k)$$

desta forma explorar-se-ia melhor a reconstrução intermediária da imagem.

Estes estágios são de vital importância na transmissão progressiva, dado que os critérios de avaliação da performance do algoritmo exige a melhor reconstrução possível a cada passo e não exclusivamente da imagem final reconstruída. Este critério aplicado a imagem final é usualmente o adotado em codificação não progressiva, mas aqui, no decorrer do processo passamos de uma taxa de bits por pixel insignificante à conclusiva, gradativamente. Deve-se exigir que o nível de reconhecimento do conteúdo da imagem seja a maior possível em cada passo e que se possa reconhecer o conteúdo geral com a menor taxa possível. Desta forma, para uma avaliação mais rigorosa, devemos medir curvas de taxa de bit por distorção incorporada à reconstrução. Pensando assim, LT's como as extendidas e as LOT's são grandes concorrentes às bases ideais para transmissão progressiva.

5.3.3) Comentários Adicionais

Foi citado um método otimizado e não-adaptativo para alocação de bits, porém esta alocação poderia ser feita adaptativamente, levando em conta as características de cada bloco [2] [19]. Temos em mente que esta alocação é necessária para uma performance de alta qualidade ou, até, reconstrução perfeita e dimensiona a quantidade de informação transmitida após o último estágio. É claro que a cada passo intermediário devemos adotar técnicas adicionais para codificação em bloco dos bits.

Além desta alocação, podemos tornar adaptativos os critérios de transmissão progressiva, encontrando a sequência ótima de coeficientes [18] e de planos, em função das particularidades de cada bloco transformados. Deve-se, contudo, agir com cautela com relação à quantidade de informação paralela que todos os sistemas de adaptação poderiam produzir.

5.4) REFERÊNCIAS

- [1] R.E. Crochiere, L.R. Rabiner, *Multirate Digital Signal Processing*, Englewood Cliffs, NJ, Prentice-Hall, 1983.
- [2] N.S. Jayant, P.Noll, *Digital Coding of Waveforms*, Englewood Cliffs, NJ, Prentice-Hall, 1984.
- [3] M. Vetterli, *A Theory of Multirate Filter Banks*, IEEE Trans. on ASSP, vol. ASSP-35, no. 3, pp 356-372, March 1987.
- [4] H.Schiller, *Overlapping Block Transform for Image Coding Preserving Equal Number of Samples and Coefficients*. Proc. SPIE, Visual Communications and Image Processing, Vol 1001, pp 834-839, 1988.
- [5] K.Nayebi, T.P.Barnwell III, M.J.T. Smith, *The Time Domain Analysis and Design of Exactly Reconstructing FIR Analysis/Synthesis Filter Banks*, Proc of ICASSP 1990, pp 1735-1738.
- [6] J.P.Princen, A.B. Bradley, *Analysis/Synthesis Filter Bank Design Based on Time Domain Aliasing Cancellation*, IEEE Trans. on ASSP, vol ASSP-34, no. 5, October 1986.
- [7] M. Vetterli, D. LeGall, *Perfect Reconstruction Filter Banks : Some Properties and Factorizations*, IEEE Trans. on ASSP, vol 37, no. 7, pp 1057-1071, July 1989.
- [8] T. Q. Nguyen, P.P.Vaidyanathan, *Structures for M-Channel Perfect Reconstruction FIR QMF Banks Which Yield Linear Phase Analysis Filters*, IEEE Trans. on ASSP, vol 38, no. 3, pp 433-446, March 1990.
- [9] T.P.Barnwell III, M.J.T. Smith, *Filter Banks for Analysis-Reconstruction Systems: A Tutorial*, Proc. of ISCAS 1990, pp 1999-2003.
- [10] H.S.Malvar, D.H.Stealin, *The LOT: Transform Coding Without Blocking Effects*, IEEE Trans. on ASSP, vol 37, pp 553-559, April 1989.

-
- [11] J.W.Woods, S.D. O'Neil, *Sub-band Coding of Images*, IEEE Trans on ASSP, vol ASSP-34, no.5, pp 1278-1288, October 1986.
 - [12] H.S.Malvar, *Efficient Signal Coding with Hierarchical Lapped Transforms*, Proc. of ICASSP 1990.
 - [13] L. Wang, M. Goldberg, *Reduced-difference Pyramid : a Data Structure for Progressive Image Transmission*, Optical Engineering, Vol 28, #7, July 1989.
 - [14] L. Wang, M. Goldberg, *Pyramid Transform Coding Using Vector Quantization*, Proc. of ICASSP 1988, pp 812-815.
 - [15] H.S.Malvar, *Lapped Transforms for Efficient Transform/ Sub-band Coding*. IEEE Trans. on ASSP, vol 38, no. 6, June 1990.
 - [16] H.S.Malvar, *Modulated QMF Filter Banks With Perfect Reconstruction*, Elect. Letters, vol.26, no.13, pp 906-907, 21st June 1990.
 - [17] P.H. Westerink, J.Biemond, D.E. Boeckee, *Progressive Transmission of Images Using Sub-band Coding*, Proc. of ICASSP 1989, pp 1811- 1814.
 - [18] K.N. Ngan, *Image Display Techniques Using the Cosine Transform*, IEEE Trans. on ASSP, ASSP-32, No.1, pp 173-177, February 1984.
 - [19] W.H.Chen, W.K.Pratt, *Scene Adaptive Coder*, IEEE Trans. on Commun., vol com 32, no.3, pp 225-232, March 1984.
 - [20] M. Veterli, C. Herley, *Wavelets and Filter Banks : Relationships and New Results*, Proc. of ICASSP, pp 1723-1726, 1990.
-

CAPÍTULO 6

CONCLUSÃO

Sistemas que utilizam múltiplas resoluções fornecem resultados, em geral, mais satisfatórios que aqueles que utilizam processamento puramente sequencial. Embora pouco analisada sob este enfoque, a codificação por transformadas é um exemplo capaz de atingir grande performance e que cada vez mais é foco de atenção nas pesquisas correntes. A vantagem de se empregar várias resoluções em um mesmo processamento basicamente reside na distribuição de trabalho e na exploração da grande correlação entre imagens de diferentes resoluções retratando uma mesma cena. Deve ficar claro que os processos aqui discutidos pouco serviriam para sinais que não fossem tipicamente passa-baixas e não foi estudada a aplicabilidade de sistemas de multiresolução a imagens médicas de alta resolução e contraste, ou a imagens que não retratem cenas típicas de rostos ou paisagens.

A filtragem é um processamento crucial neste trabalho. O principal intento era o de aumentar o número de amostras de determinado sinal, como se este não fosse o produto de uma prévia dizimação. Desta forma as amostras existentes deviam ser mantidas e chegou-se a conclusão que os interpoladores lineares são praticamente satisfatórios para imagens que pudessem ser modeladas como processos Gauss-Markov e para baixas taxas de dizimação. Para maior precisão em estimação de bordas, aconselha-se o uso da extensão discreta do Núcleo de Convolução Cúbica que é pouco mais dispendioso, mas, mesmo assim, extremamente simples, quando se tenta duplicar a resolução do sinal. De qualquer forma, o uso da interpolação linear não é uma má prática nestes casos.

A codificação piramidal lançou novas perspectivas sobre a codificação de imagens, com técnicas eficientes para transmissão progressiva e atingindo boa capacidade de compressão. O desenvolvimento de conceituações fundamentais do método nos

permitiu elaborar um método simplificado, eliminando a prefiltração. Todavia a incorporação de *aliasing* pode ser por demais interferente e sua aplicabilidade é restrita a uma taxa de dizimação pequena. Acredita-se que não mais que três níveis da pirâmide sejam permissíveis em um esquema não adaptativo e independente da imagem. Para o caso bidimensional, entretanto, utilizando uma estrutura polifásica adequada do interpolador é possível desenvolver um novo esquema hexagonal/retangular para a pirâmide. Dobra-se o número de passos, mas atingindo maior simplicidade com eficiência compatível. Mesmo assim, não é possível atingir taxas de dizimação elevadas.

O DPCM em Pirâmide, ou híbrido inter-extrapolativo, é um sistema desenvolvido a partir das pirâmides sem prefiltração e é a maior contribuição deste trabalho. Esta nova formulação nos permite utilizar circuitos até quatro vezes mais lentos, através da distribuição do processamento, quando comparada à do DPCM convencional, fazendo, inclusive, com que o sistema híbrido seja aconselhável para codificação em altas taxas. Nestas velocidades, a maior parte do circuito é implementado em *hardware* dedicado e o intervalo entre amostras para o processamento é crítico, tornando o DPCM em Pirâmide uma atrativa ferramenta. Além disto a performance é, no mínimo, compatível e pode superar em muito o sistema puramente extrapolativo devido à utilização de interpoladores e da decomposição do processamento por componentes distintas em frequência, como em codificação em sub-bandas. Apesar de não haver filtração de análise, é um sistema eficiente, sendo mais flexível que o DPCM e mantendo a simplicidade deste. Todavia, a performance alcançada neste trabalho está longe de ser a ótima, apenas superando a performance do DPCM convencional sob as mesmas condições. Mesmo assim, devido a sua flexibilidade, foi possível reduzir a taxa, mantendo qualidade subjetiva, com sistemas simples, sem quantização ou filtrações adaptativas. Taxas em torno de 2 bits/pixel (ou menores) com filtrações unidimensionais permitiram formar imagens indistinguíveis da original e tudo indica que taxas bem mais baixas seriam possíveis se utilizarmos quantização vetorial adaptativa e explorarmos a correlação entre campos e quadros de um sinal de vídeo.

Pensando em sistemas para aplicações em transmissão progressiva, mas com algoritmos rápidos, filtros ortogonais e permitindo reconstrução perfeita, apresentamos um método utilizando a LOT como uma das formas mais eficientes para este tipo de transmissão. Por ser uma pesquisa utilizando desenvolvimentos teóricos recentes e devido à extensão deste estudo, resultados complementares estão sendo avaliados e os aqui apresentados são apenas parte de um estudo mais amplo cujos resultados serão continuamente divulgados.

É vantajoso, neste ponto, destacar as maiores contribuições desta tese que basicamente residem em aplicações da pirâmide sem prefiltração : o DPCM em Pirâmide, principalmente, mas também a formulação da pirâmide quincunx. O método de transmissão progressiva através da LOT é uma reunião inédita de técnicas já existentes, mas por não ser um trabalho completo não merece sobressair-se aqui.

Sem dúvida, o estudo de sistemas de multiresolução é uma potente frente de trabalho que promete melhor realizar os intentos dos sistemas puramente sequenciais. A existência de transformadas hierárquicas e piramidais, bem como de transformadas sobrepostas (unindo sob a mesma análise decomposição por transformadas e por sub-bandas) abre um novo caminho, sob este enfoque, permitindo rápida execução e ótimo desempenho. É necessário, cada vez mais, explorar estas técnicas, atingindo uma codificação cada vez mais eficiente.

APÊNDICE

IMAGENS ORIGINAIS E PROCESSADAS

O objetivo deste apêndice é o de incluir as imagens e fotografias referentes a tese. Optou-se pela separação para simplificar os processos de inclusão e reprodução destas.

Desta feita, iremos apresentar as imagens originais, para que possa ser avaliado o conteúdo e características destas, como imagens de teste. Elas foram digitalizadas como descrito no Capítulo 1, algumas características são apresentadas na Tabela 1.1 e são apresentadas na Figura A1, sendo seus apelidos dados no rodapé das figuras. Estas fotografias foram obtidas diretamente do monitor de vídeo com resolução de 512x480 pixels.

Na Figura A2, apresenta-se alguns resultados do Capítulo 4, destacando regiões centrais de cada uma das seis imagens, colocando lado a lado a imagem original e os resultados obtidos por cada um dos processamentos denotados por (A), (B) e (C) (Seção 4.4). Foram todos enquadrados em uma fotografia (direta do monitor, 512x480 pixels) para que os processos de aquisição, revelação e ampliação das diferentes imagens não influísse na análise comparativa. Nota-se que subjetivamente não há diferença entre as quatro imagens, sob diferentes processamentos, mesmo pensando que a compressão do esquema (B) é maior que as outras. A disposição das imagens, nas fotografias é indicada nos rodapés das figuras.

Para melhor apresentar as imagens explicitando as diferenças foram feitas ampliações, para impressão a LASER, nas Figuras A3-A8. As Figuras A3-A6 apresentam imagens originais e decodificadas (esquema B), com resolução 200x200 pixels. Pode se dizer que possuem comportamento típico, mas ainda há uma dificuldade em se perceber diferenças. Por tal motivo, ampliou-se a imagem "Girl", nas Figuras A7 e A8, destacando detalhe do olho direito com resolução de apenas 32x32 pixels. Nestas é possível visualizar claramente algumas diferenças.



Figura A1 - Imagens originais. a) Parte superior, imagem "Girl". b) Parte Inferior, imagem "Cameraman".



Figura A1 - Imagens originais. c) Parte superior, imagem "Zelda". d) Parte Inferior, imagem "Praia".

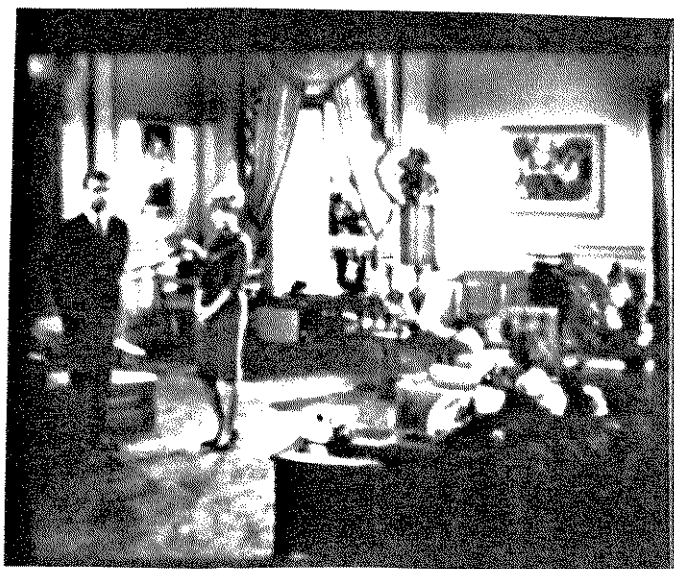


Figura A1 - Imagens originais e) Parte superior, imagem "Sala".
f) Parte Inferior, imagem "Cozinha".

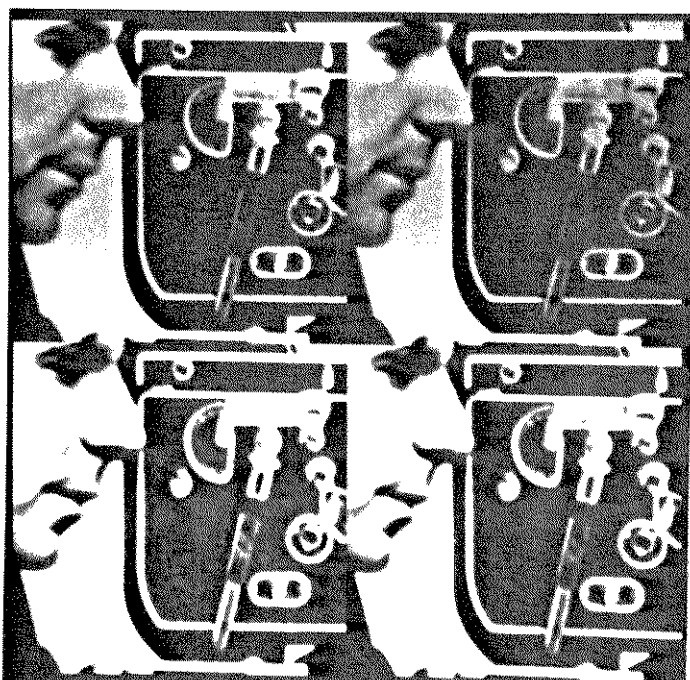


Figura A2 - Resultados comparativos entre diferentes processamentos e imagem original. Posição relativa dentro de cada fotografia: Original (Sup. Esq.), Esquema A (Inf.Dir.), Esquema B (Sup.Dir.), Esquema C (Inf.Esq.). Acima para imagem "Girl", abaixo imagem "Cameraman".

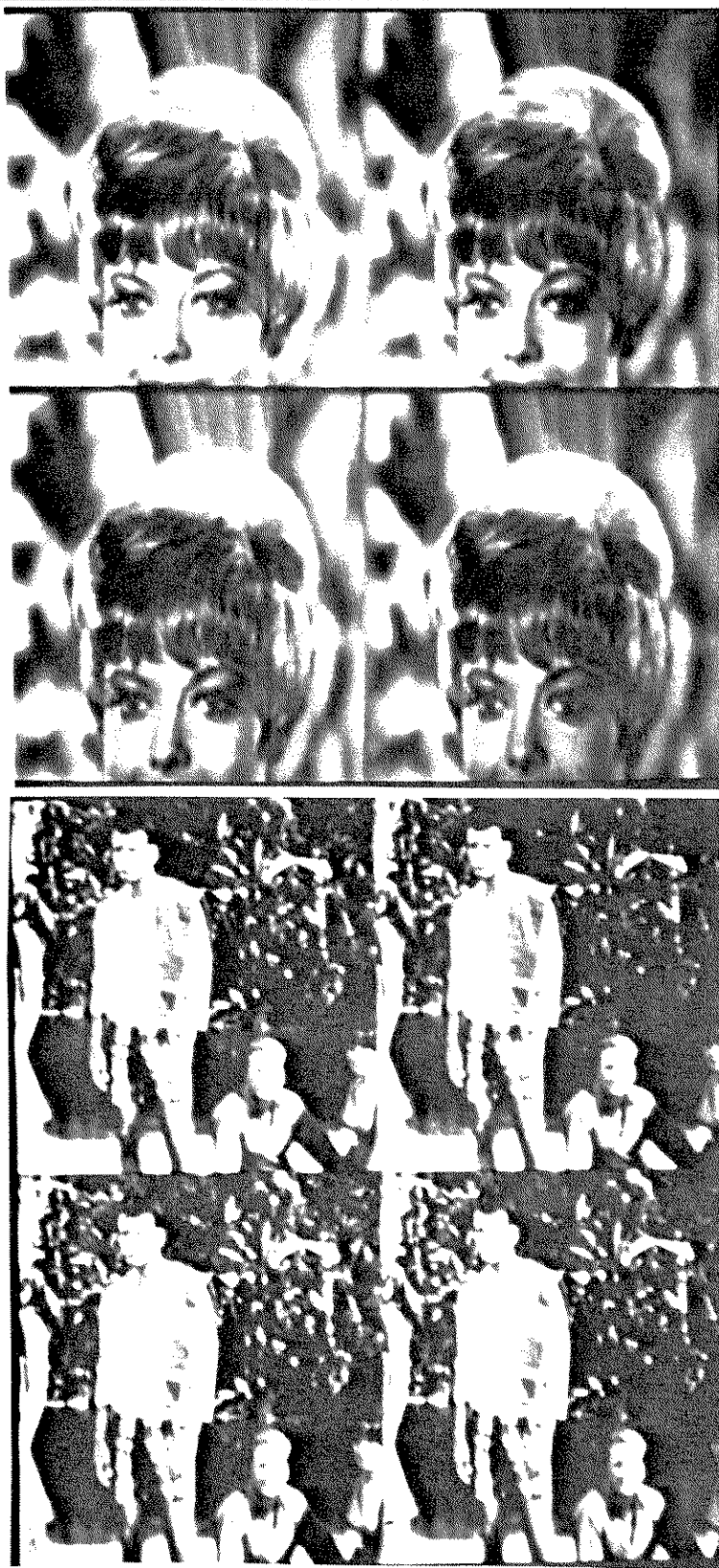


Figura A2 - Resultados comparativos entre diferentes processamentos e imagem original. Posição relativa dentro de cada fotografia: Original (Sup. Esq.), Esquema A (Inf. Dir.), Esquema B (Sup. Dir.), Esquema C (Inf. Esq.). Acima para imagem "Zelda", abaixo imagem "Praia".

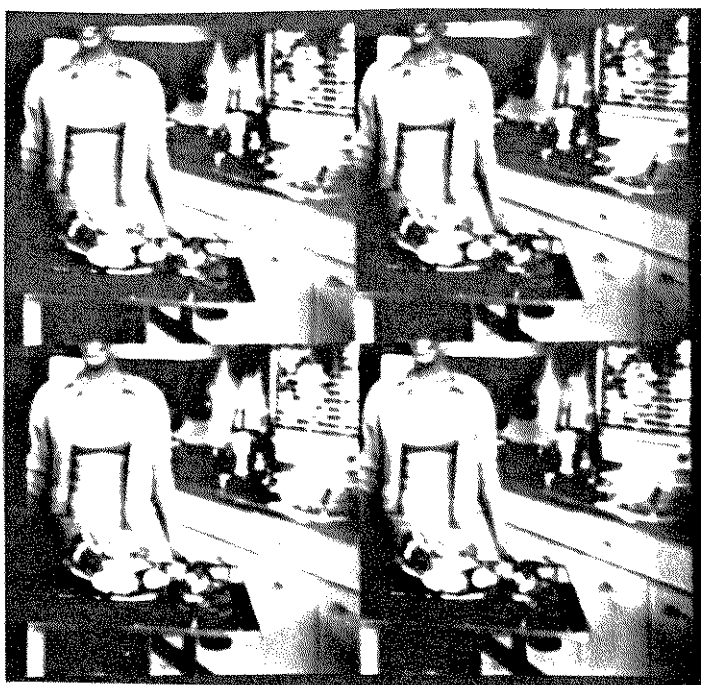


Figura A2 - Resultados comparativos entre diferentes processamentos e imagem original. Posição relativa dentro de cada fotografia: Original (Sup. Esq.), Esquema A (Inf. Dir.), Esquema B (Sup. Dir.), Esquema C (Inf. Esq.). Acima para imagem "Sala", abaixo imagem "Cozinha".

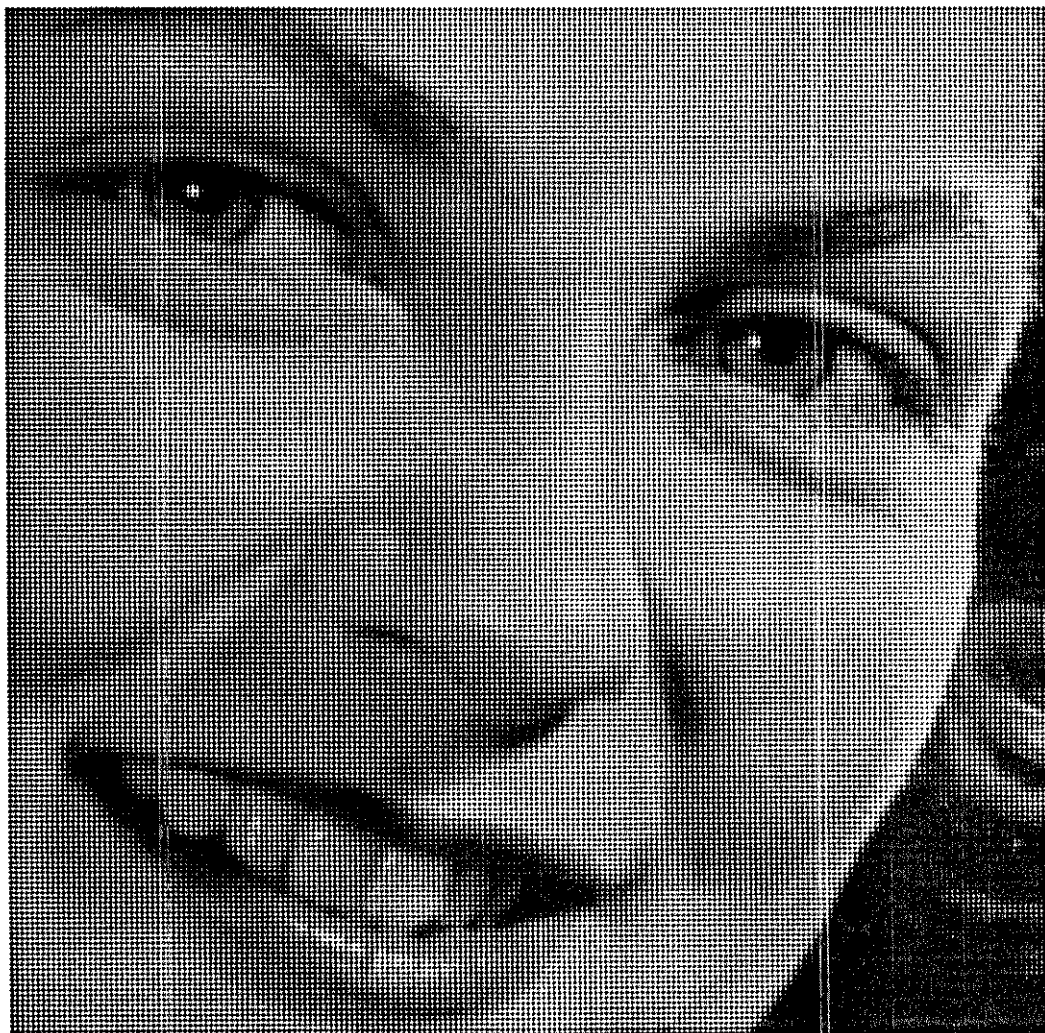


Image file is gz.dat.

Figura A3 - Imagem "Girl" original ampliada. Resolução 200x200 pels.



Image file is gpz.dat.

Figura A4 - Imagem "Girl" codificada sob DPCM em Pirâmide
(esquema B) e ampliada. Resolução 200x200 pels.

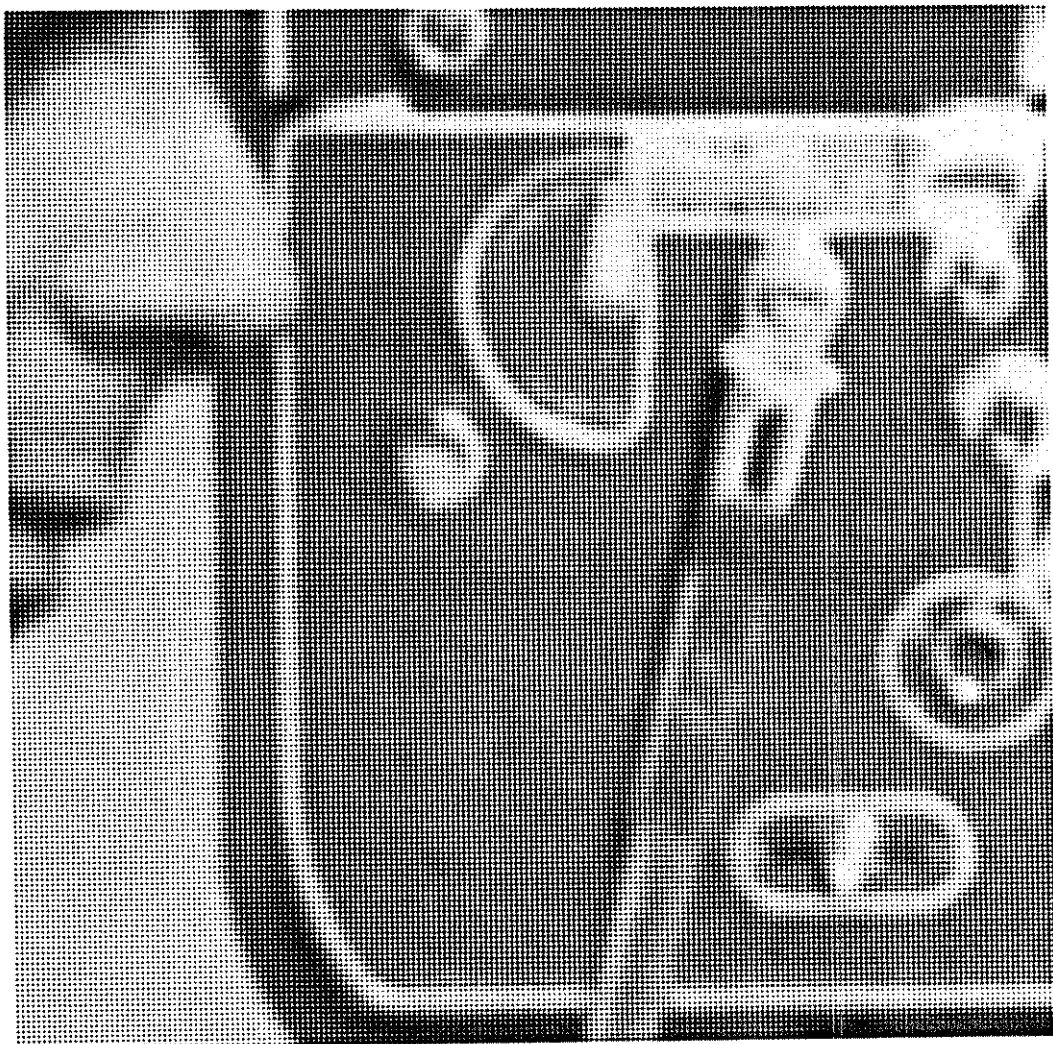


Image file is cz.dat.

Figura A5 - Imagem "Cameraman" original ampliada. Resolução
200x200 pels.

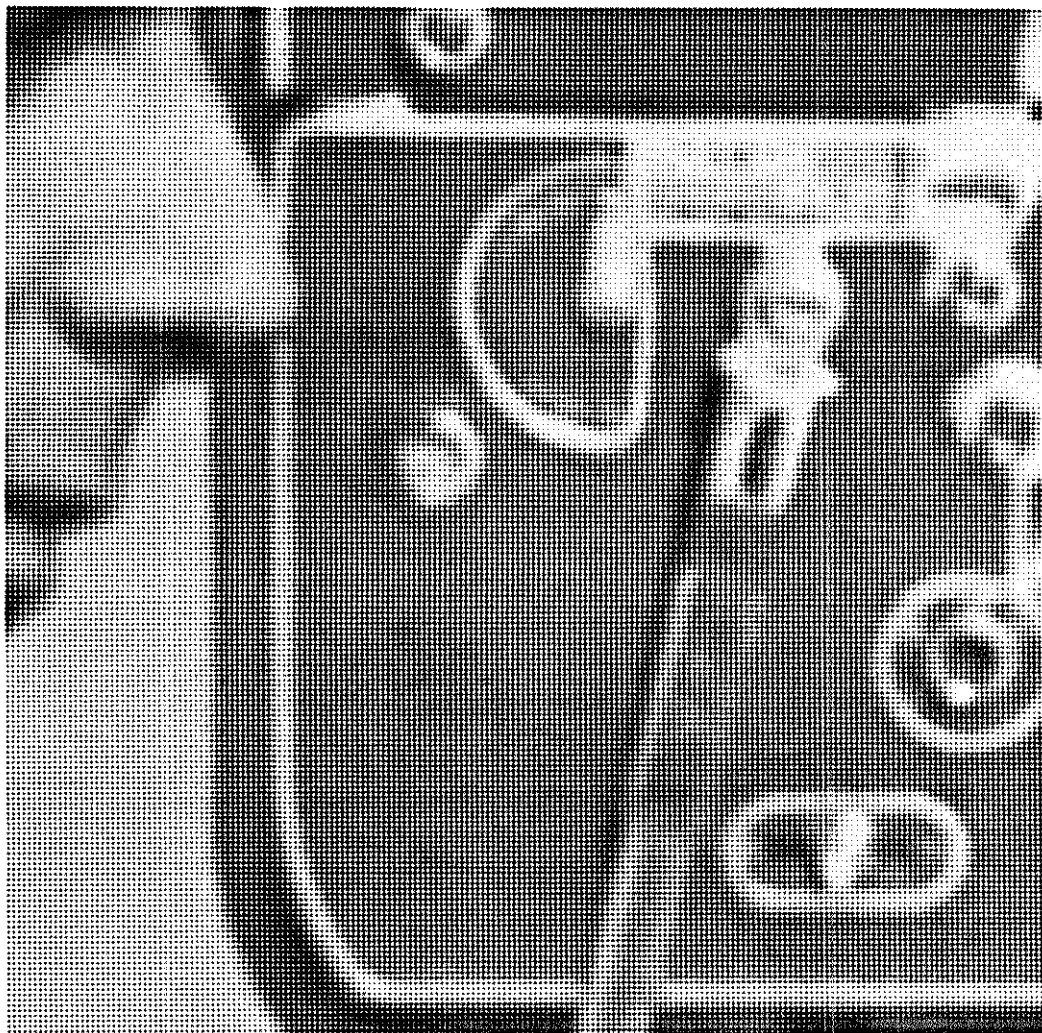


Image file is cpz.dat.

Figura A6 - Imagem "Cameraman" codificada sob DPCM em Pirâmide (esquema B) e ampliada. Resolução 200x200 pels.

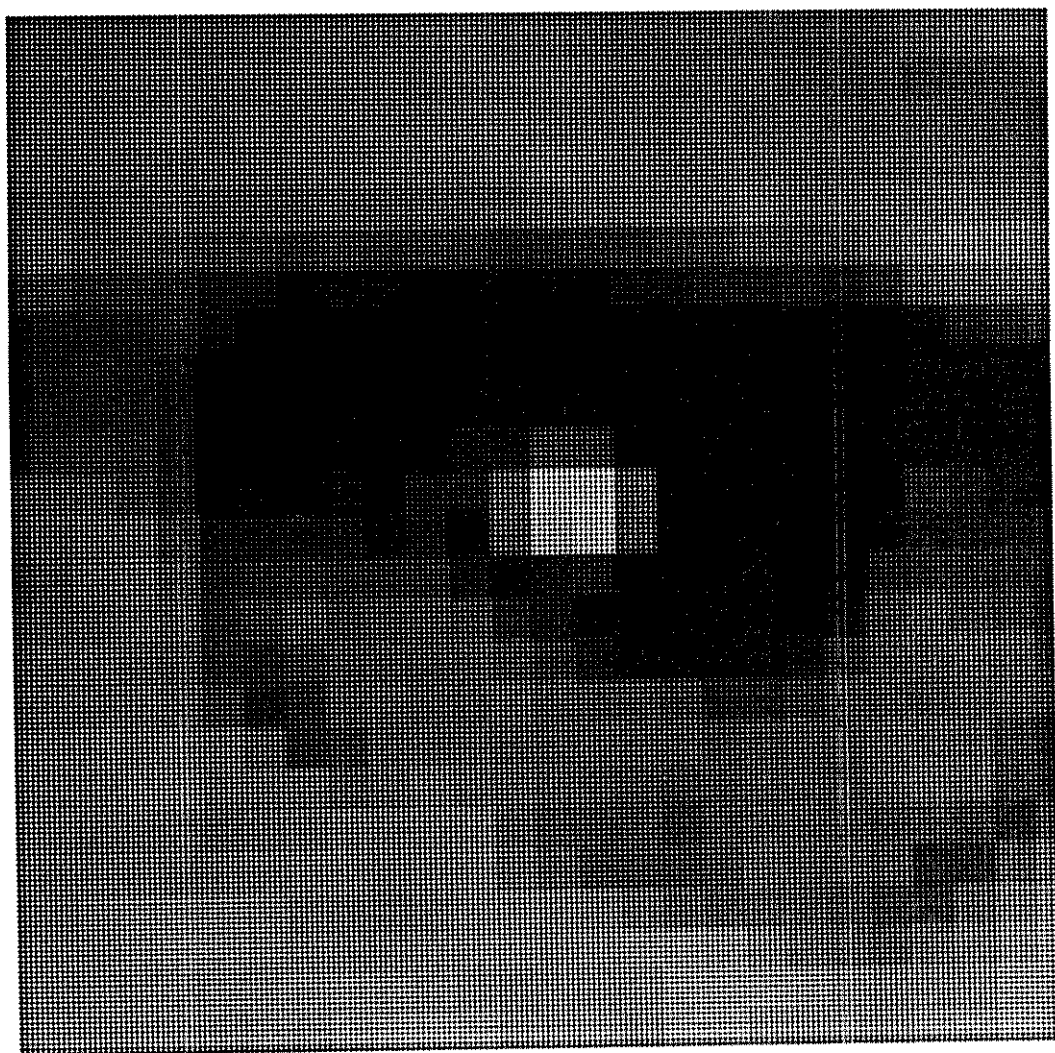


Image file is gz8.dat.

Figura A7 - Zoom do olho direito da modelo da imagem "Girl".
Original. Resolução 32x32 pels.

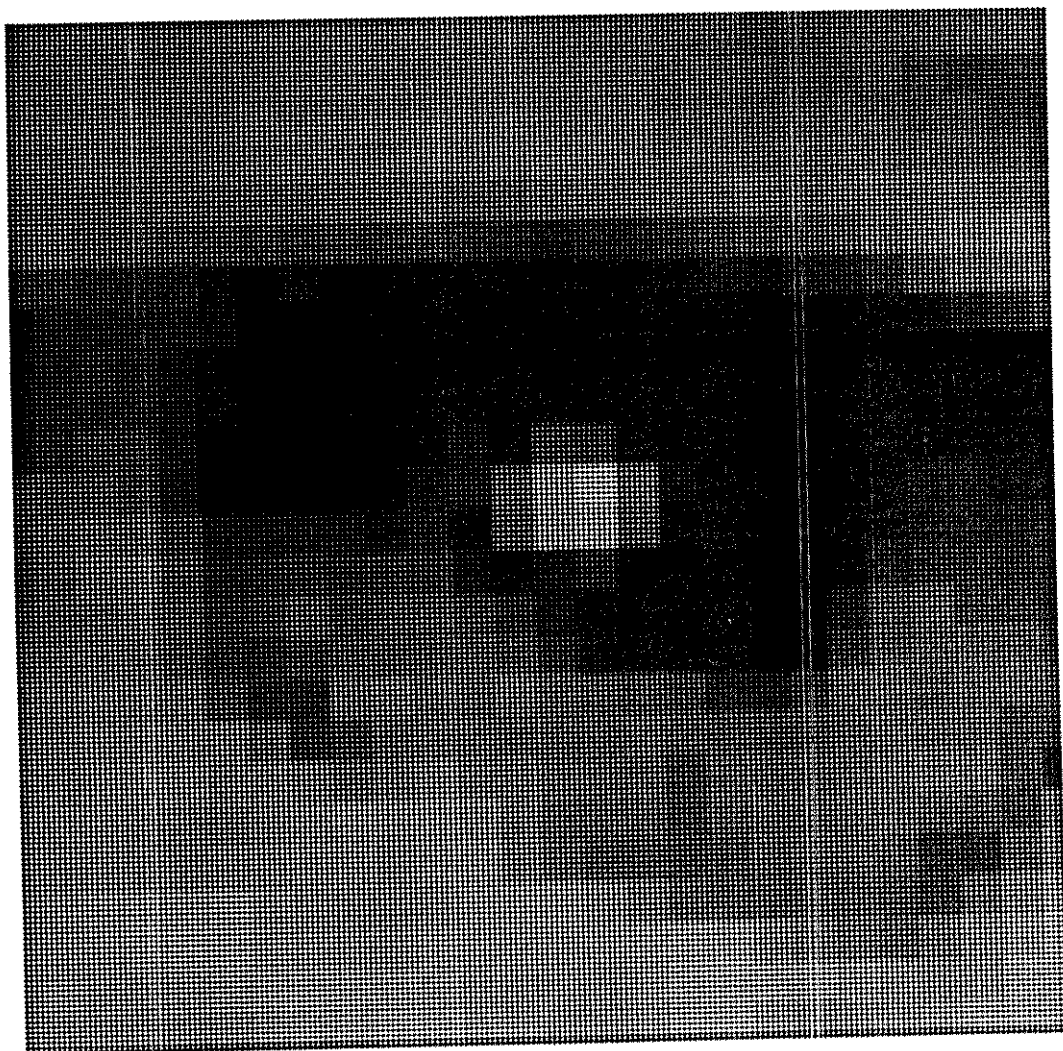


Image file is gpz8.dat.

Figura A8 - Zoom do olho direito da modelo da imagem "Girl" codificada sob DPCM em Pirâmide (esquema B). Resolução 32x32 pels.
