

RONALDO APARECIDO DE ABREU

**TÉCNICAS DE EQUALIZAÇÃO DE CANAIS DE
COMUNICAÇÃO APLICADAS A IMAGENS**

Dissertação apresentada à Escola
Politécnica da Universidade de
São Paulo para obtenção do
Título de Mestre em Engenharia
Elétrica.

São Paulo
2011

RONALDO APARECIDO DE ABREU

**TÉCNICAS DE EQUALIZAÇÃO DE CANAIS DE
COMUNICAÇÃO APLICADAS A IMAGENS**

Dissertação apresentada à Escola
Politécnica da Universidade de
São Paulo para obtenção do
Título de Mestre em Engenharia
Elétrica.

Área de Concentração:
Sistemas Eletrônicos

Orientador:
Prof. Dr. Magno T. M. Silva

São Paulo
2011

Este exemplar foi revisado e alterado em relação à versão original, sob responsabilidade única do autor e com a anuência de seu orientador.

São Paulo, 15 de junho de 2011.

Ronaldo Aparecido de Abreu

Prof. Dr. Magno Teófilo Madeira da Silva

Abreu, Ronaldo Aparecido de

Técnicas de equalização de canais de comunicação aplicadas a imagens. /
R.A. de Abreu. – ed. rev. – São Paulo, 2011.
85 p.

Dissertação (Mestrado) - Escola Politécnica da Universidade de São Paulo.
Departamento de Engenharia de Sistemas Eletrônicos.

1. Filtros elétricos adaptativos 2. Algoritmos para imagens 3. Processamento de Imagens I. Universidade de São Paulo. Escola Politécnica. Departamento de Engenharia de Sistemas Eletrônicos II. t.

Agradecimentos

Ao meu orientador e amigo, Prof. Magno T. M. Silva, por toda a atenção e principalmente paciência ao longo deste projeto. Depois de muito esforço e muitas conversas, conseguimos chegar a algo palpável que creio que possa nos levar a outros estudos.

Ao Prof. Miguel A. Ramirez e à Profa. Maria D. Miranda pelas sugestões dadas no exame de qualificação.

Aos demais professores que contribuíram no meu desenvolvimento acadêmico e na realização deste trabalho, em especial ao Prof. Marcio Eisencraft e ao Prof. Vanderlei Cunha Parro, os quais têm grande responsabilidade pela minha fascinação pela Engenharia.

Aos meus pais que sempre me incentivaram e que me orgulham cada dia que passa por todo apoio ao longo de minha vida.

À minha namorada Nathália pelo apoio e compreensão nos momentos mais difíceis.

Ao Eng. Benito De Domenico, meu antigo chefe e amigo, pela sua compreensão sem a qual jamais poderia iniciar este novo desafio e muito menos ter a chance de concluí-lo.

Aos colegas do Laboratório de Processamento de Sinais, em especial ao amigo João Mendes Filho pelas inúmeras conversas e ótima convivência.

Aos amigos sem exceção pelo companheirismo e incentivo a iniciar o mestrado.

Por todo apoio e conselhos ao longo destes anos, eu agradeço.

Nem tudo que se enfrenta pode ser modificado,
mas nada pode ser modificado até que seja enfrentado.

Albert Einstein

Resumo

O objetivo da desconvolução autodidata de imagens é reconstruir a imagem original a partir de uma imagem degradada sem usar informação da imagem real ou da função de degradação. O processo de reconstrução é crítico em aplicações em que a imagem original ou suas características estatísticas não são conhecidas. Fazendo um mapeamento da imagem digital antes de sua transmissão, ela pode ser interpretada como um sinal de comunicação com modulação do tipo PAM (*Pulse Amplitude Modulation*). Utilizando essa interpretação, técnicas clássicas de equalização de canais de comunicação podem ser usadas para restauração de imagens. Além disso, é usual considerar os pixels de uma imagem como um sinal não-estacionário, o que justifica o uso de algoritmos adaptativos. Neste trabalho, técnicas adaptativas usadas em equalização de canais de comunicação são aplicadas para restauração de imagens. Inicialmente, é proposta uma nova técnica de varredura a fim de minimizar alterações bruscas no sinal de entrada do filtro adaptativo. Utilizando o algoritmo *Least Mean Squares*, obtém-se uma equivalência entre funções de degradação de imagens e canais de comunicação variantes no tempo. Isso possibilitou comparar algumas funções de degradação com relação à distorção causada em imagens. Em seguida, usando um rearranjo dos elementos da matriz de entrada em um vetor, o algoritmo multimódulo regional (RMMA - *Region-based Multimodulus Algorithm*) foi estendido para restauração de imagens. Esse algoritmo é então usado para adaptação dos coeficientes do equalizador linear transversal e também do equalizador de decisão realimentada. Cabe observar que o RMMA trata um sinal de módulo não-constante como se fosse de módulo constante, o que faz com que ele se comporte como um algoritmo supervisionado, convergindo na média para a solução de Wiener. Dessa forma, o RMMA apresenta um desempenho melhor que o algoritmo do módulo constante (CMA - *Constant Modulus Algorithm*) convencional, usado em equalização autodidata de canais de comunicação. Esse comportamento também foi observado na reconstrução de imagens, através das simulações apresentadas nesta dissertação. Este estudo abre novas perspectivas de extensão de técnicas usadas em equalização de canais de comunicação para restauração de imagens. Uma delas é a possibilidade de restauração de imagens coloridas usando diversidade espacial.

Abstract

The aim of blind image deconvolution is to reconstruct the original scene from a degraded observation without using information about the true image and the point spread function. The restoration process is critical in applications, where the true image or its statistical characteristics are unknown. Mapping the pixels of the original image before its transmission, the mapped image can be interpreted as a pulse amplitude modulation (PAM) signal, used in communications systems. With this interpretation, classic equalization techniques of communication channels can be used to image restoration. Furthermore, the pixels of a true image constitute a nonstationary signal, which justifies the use of adaptive filters. In this dissertation, adaptive techniques used for equalization of communication channels are applied to image restoration. Firstly, we propose a new update path through the blurred image that consists in a combination of horizontal and vertical alternate paths. This update path minimizes the problem of abrupt changes in the adaptation of the filter and provides better conditions to the image recovery. Using the least mean squares (LMS) algorithm, we obtain an equivalence between a point spread function and a time-variant communication channel. This equivalence was used to compare some point spread functions in relation to the distortion caused in images. Secondly, reshaping the input matrix into a column vector, we extend the regional-based multimodulus algorithm (RMMA) to blind image deconvolution. This algorithm is used to update the coefficients of the linear transversal equalizer and also of the decision feedback equalizer. RMMA treats nonconstant modulus signals as constant modulus ones, which provides a behavior closer to that of a supervised algorithm. Thus, RMMA can converge in the mean to the Wiener solution and, therefore, presents a better performance when compared to the conventional constant modulus algorithm (CMA), used in blind equalization of communication channels. This behavior was also observed in image restoration, through the simulations presented in this dissertation. This study pushes back the frontiers of image processing, since different techniques used in equalization can be extended to image restoration. One of the new possibilities is the color image restoration using the spatial diversity.

Sumário

Lista de Figuras	vi
Lista de Tabelas	ix
Lista de Abreviaturas	x
Lista de Símbolos	xii
1 Introdução e formulação do problema	1
1.1 Motivação	1
1.2 Formulação do problema	3
1.3 Objetivos	6
1.4 Justificativa	7
1.5 Contribuições da Dissertação	7
1.6 Organização da Dissertação	8
2 Equalização de canais de comunicação	9
2.1 Formulação do problema - o LTE e o DFE	9
2.2 A Solução de Wiener	12
2.3 Equalização supervisionada - o algoritmo LMS	13
2.4 Equalização autodidata com algoritmos de módulo constante	15
2.4.1 O algoritmo do módulo constante - CMA	15
2.4.2 O algoritmo do módulo constante que evita divergência - DM-CMA . .	16
2.4.3 O algoritmo do módulo constante para o DFE - DFE-CMA	17
2.4.4 O algoritmo multimódulo regional - RMMA	19
2.5 Conclusão	23
3 Degradação em imagens	24
3.1 Quantização e Amostragem	24
3.2 Espalhamento (PSF-Point Spread Function)	25
3.3 Ruído aditivo	30

4	Algoritmos adaptativos para restauração de imagens	34
4.1	Definição de Janelas	34
4.2	Tratamento de borda	35
4.3	Varredura	35
4.4	Equalização supervisionada bidimensional - o algoritmo TDLMS	38
4.5	Equalização autodidata bidimensional - o TDCMA	40
4.6	Imagens tratadas como sinais unidimensionais	42
5	Soluções alternativas para restauração de imagens	43
5.1	Introdução e Método de Avaliação	43
5.2	A relação com o canal de comunicação	45
5.3	Alteração de varredura	49
5.4	O algoritmo multimódulo regional bidimensional - o TDRMMA	56
5.5	O algoritmo multimódulo com decisão realimentada- o TDRMMA - DFE	63
6	Conclusões e trabalhos futuros	72
	Referências Bibliográficas	76
	Apêndices	79
A	Fundamentos da Imagem Digital	79
A.1	A Imagem	79
A.2	O Sistema Bidimensional	80
A.3	A Transformada de Fourier e a inteligibilidade	83

Lista de Figuras

1.1	<i>(a)aplicações médicas - tomografia, [fonte: www.srccomp.com], (b)aplicações astronômicas - mapeamento estelar, [fonte: www.nasa.gov] (c) aplicações em microscopia - visualização de microorganismos, [fonte: www.hypescience.com] (d) aplicações governamentais com reconhecimento de padrões - reconhecimento de placas de automóveis, [fonte: www.reconhecimento.com], (e) aplicações meteorológicas - imagens infra-vermelho de furacões, [fonte: www.nasa.gov], (f) aplicações em visão computacional - sistemas de visualização para robôs de missões não tripuladas, [fonte: www.nasa.gov]</i>	2
1.2	<i>Representação em blocos da degradação da imagem</i>	3
1.3	<i>Diagrama em blocos da equalização no caso generalizado</i>	5
2.1	<i>Sistema de comunicação simplificado com um equalizador linear transversal.</i>	10
2.2	<i>Diagrama em blocos de um DFE</i>	11
2.3	<i>Erro do CMMA como função de $y(n)$ para sinal 8-PAM</i>	20
2.4	<i>Regiões de um sinal 8-PAM</i>	20
2.5	<i>Erro do RMMA como função de $y(n)$ para sinal 8-PAM</i>	21
2.6	<i>Erro quadrático médio ao longo das iterações considerando os algoritmos CMA ($\mu = 1 \times 10^{-4}$) e RMMA ($\mu = 7 \times 10^{-3}$); 8-PAM; $M = 11$; ausência de ruído; algoritmos normalizados.</i>	22
3.1	<i>Quantização Digital</i>	25
3.2	<i>Resolução espacial da imagem "Bellpeppers".</i>	26
3.3	<i>Blur de movimento 1-D; $L = 11$; dimensão 1×11</i>	27
3.4	<i>Blur de movimento 2-D; $L = 6$; dimensão 6×6</i>	28
3.5	<i>Blur Gaussiano; $\sigma = 3$; dimensão 1×7</i>	28
3.6	<i>Blur de turbulência atmosférica; $\sigma^2 = 0.25$; $K=1$; dimensão 7×7</i>	29
3.7	<i>Blur de desfocagem uniforme; $R = 4$</i>	30
3.8	<i>Degradação por ruído. (a) Ruído Gaussiano, (b) Ruído Rayleigh, (c) Ruído Erlang, (d) Ruído Exponencial, (e) Ruído Uniforme e (f) Ruído Sal e Pimenta</i>	33
4.1	<i>Modelos de seleção janela</i>	35
4.2	<i>Modelos de seleção de borda</i>	36

4.3	<i>Modelos de varredura de imagem</i>	37
4.4	<i>Operador de vetorização</i>	42
5.1	<i>Imagem de teste - Lenna</i>	44
5.2	<i>Representação do %MSE. (a) Representação ao longo das iterações, (b) Representação espacial</i>	44
5.3	<i>Representação do %MSE - Vista de topo</i>	45
5.4	<i>Esquema de identificação de canal</i>	46
5.5	<i>Representação unidimensional do blur por iteração - Canal ideal - Algoritmo LMS ($\mu = 0.001$, $M = 25$); 5 repetições.</i>	47
5.6	<i>Representação unidimensional do blur por iteração - Blur de Movimento 1-D 3×3 - Algoritmo LMS ($\mu = 0.001$, $M = 25$); 5 repetições.</i>	47
5.7	<i>Diagramas de pólos e zeros - Blur de Movimento 1-D 3×3</i>	48
5.8	<i>Representação unidimensional do blur por iteração - Blur Gaussiano 2-D 7×7 - Algoritmo LMS ($\mu = 0.001$, $M = 25$); 5 repetições.</i>	48
5.9	<i>Diagramas de pólos e zeros - Blur Gaussiano 2-D 7×7</i>	49
5.10	<i>Varredura Proposta</i>	50
5.11	<i>Comparação de algoritmos - TDLMS - Blur Gaussiano 2-D 7×7 - Níveis de cinza: 8 bits/pixel - Imagem 256×256 pixels</i>	51
5.12	<i>Simulações de %MSE no TDLMS - Blur Gaussiano 2-D 7×7 - Níveis de cinza: 8 bits/pixel - Imagem 256×256 pixels - sem ruído</i>	52
5.13	<i>Efeitos da equalização - Blur Gaussiano 3×3 - Níveis de cinza: 1 bit/pixel - Imagem 256×256 pixels</i>	53
5.14	<i>Comparação de algoritmos - TDCMA - Blur Gaussiano 2-D 5×5 - Níveis de cinza: 3 bits - Imagem 256×256 pixels - sem ruído</i>	54
5.15	<i>Simulações de %MSE no TDCMA - Blur Gaussiano 2-D 5×5 - Níveis de cinza: 3 bits/pixel - Imagem 256×256 pixels - sem ruído</i>	55
5.16	<i>Efeitos da recuperação no TDCMA - Blur de movimento 1-D 3×3 - Níveis de cinza: 2 bits/pixel - Imagem 256×256 pixels - sem ruído</i>	56
5.17	<i>Arranjos em blocos do TDRMMA</i>	58
5.18	<i>Efeitos da equalização - Blur Gaussiano 2-D 3×3 - Níveis de cinza: 4 bits/pixels - Imagem 256×256 pixels - sem ruído</i>	59
5.19	<i>Efeitos da equalização - Blur Gaussiano 3×3 - Níveis de cinza: 6 bits/pixels - Imagem 256×256 pixels - sem ruído</i>	60
5.20	<i>Efeitos da equalização - Blur de movimento 1-D 5×5 - Níveis de cinza: 8 bits/pixel - Imagem 256×256 pixels - sem ruído</i>	61
5.21	<i>Efeitos da equalização - Blur Gaussiano 2-D 5×5 - Níveis de cinza: 8 bits/pixel - Imagem 256×256 pixels - BSNR=18 dB</i>	62
5.22	<i>Arranjos em blocos do TDRMMA-DFE</i>	63

5.23	<i>Efeitos da equalização - Blur Gaussiano 3×3 - Níveis de cinza: 2 bits/pixel - Imagem 256×256 pixels - sem ruído</i>	64
5.24	<i>Efeitos da equalização - Blur Gaussiano 3×3 - Níveis de cinza: 3 bits/pixel - Imagem 256×256 pixels - sem ruído</i>	65
5.25	<i>Efeitos da equalização - Blur Gaussiano 5×5 - Níveis de cinza: 4 bits/pixel - Imagem 256×256 pixels - sem ruído</i>	66
5.26	<i>Efeitos da equalização - Blur Gaussiano 5×5 - Níveis de cinza: 5 bits/pixel - Imagem 256×256 pixels - sem ruído</i>	67
5.27	<i>Efeitos da equalização - Blur Gaussiano 5×5 - Níveis de cinza: 6 bits/pixel - Imagem 256×256 pixels - sem ruído</i>	68
5.28	<i>Efeitos da equalização - Blur Gaussiano 5×5 - Níveis de cinza: 7 bits/pixel - Imagem 256×256 pixels - sem ruído</i>	69
5.29	<i>Efeitos da equalização - Blur Gaussiano 5×5 - Níveis de cinza: 8 bits/pixel - Imagem 256×256 pixels</i>	70
5.30	<i>(a) Resultado da restauração com o TDRMMA (b) Resultado da restauração com o TDRMMA que evita soluções degeneradas - Blur Gaussiano 2-D 7×7 - Níveis de cinza: 3 bits/pixel - Imagem 256×256 pixels - $\mu = 10^{-3}$</i>	71
6.1	<i>Decomposição das camadas da imagem - (a) imagem colorida, (b) camada vermelha, (c) camada verde, (d) camada azul</i>	74
6.2	<i>Simulação TDRMMA com diversidade espacial - (a) imagem original, (b) imagem degradada, (c) imagem recuperada</i>	75
A.1	<i>Representação gráfica de uma sequência bidimensional</i>	80
A.2	<i>Diagrama em blocos de um sistema de imagens</i>	82
A.3	<i>Propriedades da convolução: Comutativa</i>	82
A.4	<i>Propriedades da convolução: Associativa</i>	82
A.5	<i>Propriedades da convolução: Distributiva</i>	82
A.6	<i>Propriedades da convolução: Deslocamento no espaço</i>	83
A.7	<i>Lenna</i>	84
A.8	<i>Transformada de Fourier de Lenna</i>	84
A.9	<i>Inteligibilidade - relação módulo e fase</i>	85

Lista de Tabelas

2.1	Sumário do algoritmo LMS.	14
2.2	Sumário do algoritmo NLMS.	15
2.3	Sumário do CMA.	16
2.4	Sumário do DM-CMA.	18
2.5	Sumário do Algoritmo DFE-CMA-FB.	19
2.6	Sumário do RMMA.	22
4.1	Sumário do Algoritmo <i>Two Dimensional Least-Mean Square</i> (TDLMS).	40
4.2	Constante de dispersão	41
4.3	Sumário do <i>Two Dimensional Constant Modulus Algorithm</i> (TDCMA).	41
5.1	Sumário do TDRMMA.	58

Lista de Abreviaturas

A seguir são listadas as principais abreviações usadas na dissertação. No caso de siglas consagradas na literatura internacional, optou-se por manter as mesmas em inglês.

LSI	<i>Linear Shift Invariant</i> (Linear e Invariante no Espaço)
LTE	<i>Linear Transversal Equalizer</i> (Equalizador Linear Transversal)
DFE	<i>Decision Feedback Equalizer</i> (Equalizador de Decisão Realimentada)
MSE	<i>Mean Squared Error</i> (Erro quadrático médio)
FIR	<i>Finite Impulse Response</i> (Resposta ao pulso unitário finita)
CMA	<i>Constant Modulus Algorithm</i> (Algoritmo do módulo constante)
LMS	<i>Least-Mean Square</i> (Algoritmo do Mínimos Quadrados)
NLMS	<i>Normalized Least-Mean Square</i> (LMS normalizado)
PAM	<i>Pulse Amplitude Modulation</i> (Modulação em Amplitude de Pulso)
QAM	<i>Quadrature Amplitude Modulation</i> (Modulação de Amplitude em Quadratura)
RMMA	<i>Region-based Multimodulus Algorithm</i> (Algoritmo múltimódulo regional)
SNR	<i>Signal-to-Noise Ratio</i> (Relação sinal-ruído)
BSNR	<i>Blurred Signal-to-Noise Ratio</i> (Relação Sinal Degradado e ruído)
TDCMA	<i>Two-Dimensional Constant Modulus Algorithm</i> (CMA bidimensional)
TDLMS	<i>Two-Dimensional Least-Mean Square</i> (LMS bidimensional)
TDRMMA	<i>Two-Dimensional Region-based Multimodulus Algorithm</i> (RMMA bidimensional)

Lista de Símbolos

Símbolos gerais

\mathbb{R}	conjunto dos números reais
\mathbb{N}	conjunto dos números naturais
$(\cdot)^T$	transposição de vetores ou matrizes
$\hat{\cdot}$	valor estimado
$E[\cdot]$	operador esperança matemática
z^{-1}	operador de atraso, $z^{-1}y(n) = y(n - 1)$
$ x $	módulo do escalar x
$\lceil x \rceil$	operador de arredondamento superior. Representa o próximo inteiro de x
$\ \mathbf{x}\ $	norma Euclidiana ou l_2 do vetor \mathbf{x}
σ_x^2	variância do sinal x
\mathbf{I}	matriz identidade
M	número de coeficientes do filtro
$\nabla_{\mathbf{w}} J$	Vetor Gradiente da função custo escalar J em relação ao vetor \mathbf{w}

Filtros unidimensionais

a	sinal transmitido
r	constante de dispersão que depende de estatísticas do sinal transmitido
\hat{a}	sinal recuperado após equalização
\mathbf{u}	vetor regressor de entrada do filtro
\mathbf{R}	matriz de autocorrelação do sinal de entrada do filtro
d	sinal desejado
η	ruído do canal de comunicação
e	sinal de erro
\mathbf{w}	vetor de coeficientes do filtro
\mathbf{w}_o	vetor de coeficientes ótimos
\mathbf{h}	vetor de coeficientes do canal
J	função custo

Filtros bidimensionais

$\mathbf{F}(n_1, n_2)$	imagem original
$\mathbf{H}(n_1, n_2)$	canal bidimensional
$\mathbf{G}(n_1, n_2)$	imagem degradada
$\mathbf{N}(n_1, n_2)$	ruído bidimensional
$\mathbf{D}(n_1, n_2)$	imagem desejada
$\mathbf{W}(n_1, n_2)$	matriz de coeficientes do filtro
$\hat{\mathbf{F}}(n_1, n_2)$	imagem estimada
$e(k)$	erro de estimação
n, n_1, n_2	índices de indicação de posição
$\text{vec}[\cdot]$	operador de vetorização da matriz
$\text{vec}[\cdot]^{-1}$	inversa da vetorização da matriz (rearranjo)

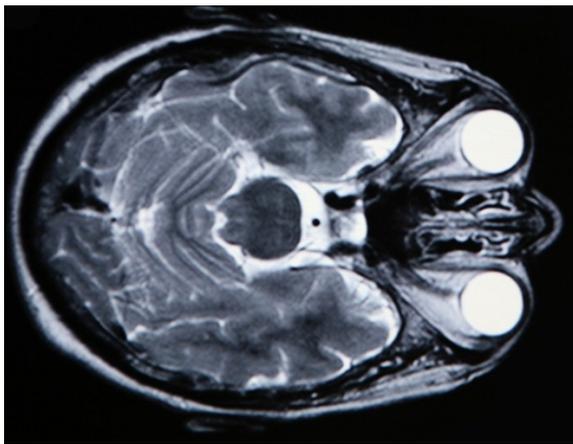
Capítulo 1

Introdução e formulação do problema

1.1 Motivação

Na atualidade, as imagens são indispensáveis para a Ciência e para o cotidiano e cada dia mais encontramos aplicações nas mais diversas áreas científicas, tecnológicas e industriais que usam imagens e realizam seu processamento para o prosseguimento de suas atividades. Seja na identificação de organismos minúsculos na área da microscopia digital [Swedlow et al., 1996] até a busca por novas galáxias e corpos celestes no campo da astronomia [Jefferies e Christou, 1993, Schulz, 1993], o processamento de imagens se torna mais presente em nossas vidas. Em todos esses casos, o objetivo sempre é de se conseguir a imagem mais fiel possível ao objeto de estudo observado tentando se aproximar ao máximo da imagem real, porém este processo nunca é perfeito [Bishop et al., 2007]. O processamento de imagens possui muitas aplicações práticas e atinge com grande impacto a área da medicina, onde inúmeros exames de imagem são realizados, como por exemplo, tomografia computadorizada, ressonância magnética e angiografia. Em aplicações mais próximas do dia-a-dia, avanços na TV digital desenvolvem as transmissões ao redor do mundo promovendo qualidade na imagem transmitida. O desenvolvimento de visão computacional desenvolve o campo da robótica gerando robôs com papéis importantíssimos na indústria e também na medicina.

A Figura 1.1 apresenta algumas ilustrações relacionadas às aplicações mencionadas anteriormente, sendo elas: (a) tomografia computadorizada usada para diagnósticos na medicina; (b) mapeamento estelar usado em astronomia; (c) microorganismos estudados em microbiologia; (d) reconhecimentos de padrões para efetuar a leitura de placas de veículos; (e) estudo de furacões em meteorologia; e (f) sonda espacial usada em visão computacional.



(a)



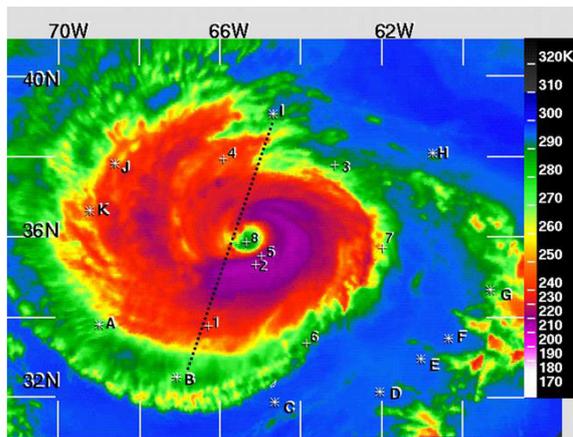
(b)



(c)



(d)



(e)



(f)

Figura 1.1: (a)aplicações médicas - tomografia, [fonte: www.srccomp.com], (b)aplicações astronômicas - mapeamento estelar, [fonte: www.nasa.gov] (c) aplicações em microscopia - visualização de microorganismos, [fonte: www.hypescience.com] (d) aplicações governamentais com reconhecimento de padrões - reconhecimento de placas de automóveis, [fonte: www.reconhecimento.com], (e) aplicações meteorológicas - imagens infra-vermelho de furacões, [fonte: www.nasa.gov], (f) aplicações em visão computacional - sistemas de visualização para robôs de missões não tripuladas, [fonte: www.nasa.gov]

1.2 Formulação do problema

Uma imagem digital é obtida pela quantização em níveis de uma imagem real. Normalmente, as aplicações usuais possuem 256 níveis de gradação para cada camada da imagem. Dados os limites finitos de níveis e de resolução (definida pelo tamanho do pixel), o próprio processo de digitalização acaba por, de certa forma, degradar a imagem original, gerando pequenas diferenças mediante comparação com a imagem observada [Lim, 1990]. Esta degradação é ínfima face à pequena perda de informação tendo em geral, a inteligibilidade da imagem mantida. Em contrapartida, em muitas aplicações, tem-se a fase de aquisição da imagem onde encontramos grandes obstáculos ao objetivo de ser fiel à imagem real. Os meios físicos, o movimento dos sistemas de aquisição, a difração da luz e as interferências externas efetuam degradação ao sinal, além de adicionar ruído, alterando as características da imagem [Kundur e Hatzinakos, 1996a,b]. A Figura 1.2¹ faz uma representação de um canal típico de imagem em que $\mathbf{F}(n_1, n_2)$ é a imagem original, $\mathbf{H}(n_1, n_2)$ é a resposta impulsiva do canal, $\mathbf{N}(n_1, n_2)$ é o ruído aditivo e $\mathbf{G}(n_1, n_2)$ é a imagem degradada, sendo n_1 e n_2 argumentos inteiros que denotam o posicionamento vertical e horizontal dos pixels respectivamente.

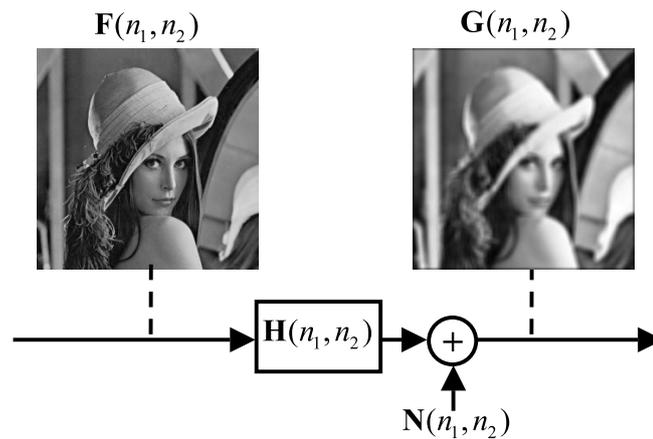


Figura 1.2: Representação em blocos da degradação da imagem

Assumindo que o canal tem comportamento linear e invariante no espaço (ou deslocamento) que a partir daqui será definido como um sistema LSI (*Linear Shift Invariant*), pode-se descrever matematicamente como sendo a convolução entre a imagem original \mathbf{F} e a resposta impulsiva do canal \mathbf{H} adicionada de ruído \mathbf{N} , ou seja,

$$\begin{aligned} \mathbf{G}(n_1, n_2) &= \sum_{k_1=-\infty}^{\infty} \sum_{k_2=-\infty}^{\infty} \mathbf{F}(k_1, k_2) \cdot \mathbf{H}(n_1 - k_1, n_2 - k_2) + \mathbf{N}(n_1, n_2) \\ &= \mathbf{F}(n_1, n_2) * \mathbf{H}(n_1, n_2) + \mathbf{N}(n_1, n_2). \end{aligned} \quad (1.1)$$

sendo que $*$ representa a operação de convolução. Detalhes sobre convolução de imagens e

¹Na Figura 1.2 se utiliza a imagem “Lenna” para esclarecer o exemplo. Essa é uma imagem largamente usada em processamento de imagens como imagem de teste e pode ser obtida, por exemplo, em www.lenna.org.

outros conceitos básicos podem ser encontrados no Apêndice A.

O canal em processamento de imagens é chamado de PSF (*Point Spread Function*) e este termo será usado neste texto. Em muitos casos, a PSF é tão severa e gera uma degradação tão alta que há grande perda de inteligibilidade na imagem recebida como se pode perceber na Figura 1.2. Isto é um sério problema em algumas aplicações em que a imagem é sua única referência [Tang e Cahill, 1991, Wen e Lee, 2002, qing Qin et al., 2009]. Em aplicações astronômicas, o objeto observado está a distâncias incalculáveis e nunca foi observado antes por olhos humanos [Jefferies e Christou, 1993, Schulz, 1993]. Em algumas aplicações médicas, a fidelidade está diretamente relacionada com o tempo de exposição do indivíduo a certos tipos de radiação e nesses casos obter uma imagem com boa definição pode gerar riscos à saúde do paciente [Michailovich e Adam, 2005].

Uma solução simples para esta condição seria utilizar um filtro que invertesse a PSF, porém o ruído gera uma complicação a este processo. Pequenas perturbações efetuadas pelo ruído geram grandes desvios à resposta da equalização já que o problema é geralmente mal-condicionado devido ao fato de que a inversa da PSF possui altas magnitudes em altas frequências, o que infelizmente amplifica bastante o ruído [Kundur e Hatzinakos, 1996a]. Outro agravante é que em situações práticas, há pouca ou até nenhuma informação sobre o canal, o que torna difícil a tarefa de se fazer um filtro baseado na estimativa da PSF. Quando se tenta implementar filtros para inversão de canal, é necessário se conhecer muito bem a degradação exercida pela PSF, e nestes casos, o filtro tem seus parâmetros estabelecidos com objetivo de inverter uma resposta impulsiva específica, o que gera possibilidade de falha caso ocorra alguma mudança no meio de transmissão. Esse problema também é conhecido na literatura como desconvolução, identificação inversa, ou equalização de canal [Haykin, 2000].

Uma tentativa também para tentar resolver o problema é recorrer à solução ótima utilizando-se um filtro de Wiener [Lim, 1990, Ekstrom, 1982]. Em um primeiro momento, parece uma solução confortável ao problema, porém novamente os agravantes para este processo não são poucos. A solução ótima utiliza informação *a priori* que como discutido não está disponível no caso de algumas aplicações e ainda mais preocupante é o fato de que o funcionamento apropriado do filtro de Wiener se baseia em assumir que o sinal a ser transmitido seja estacionário. Um problema adicional é o fato do filtro de Wiener apresentar características de um passa-baixas, o que faz com que haja perda de detalhes oriundos de altas frequências [Lim, 1990, Hadhoud e Thomas, 1988].

Diante disso, vê-se uma necessidade de se poder variar os coeficientes do filtro de modo a se obter valores mais apropriados para cada região da imagem e neste ponto se nota uma oportunidade para o uso de filtros adaptativos. Na literatura atual, há vários estudos de filtros adaptativos com a finalidade de se efetuar equalização de canais de comunicação [Qureshi, 1985, Johnson et al., 1998, Haykin, 2001]. A forma de adaptação pode ser dividida em supervisionada e autoidadada. Na adaptação supervisionada, o funcionamento consiste na entrada da imagem degradada $\mathbf{G}(n_1, n_2)$, atualização do filtro através do erro $e(k)$ obtido pela

diferença entre o pixel desejado $\mathbf{D}(n_1, n_2)$ e o pixel estimado na k -ésima iteração $\hat{\mathbf{F}}_k(n_1, n_2)$. O mesmo procedimento acontece no caso autodidata com a diferença no sinal desejado $\mathbf{D}(n_1, n_2)$ que no caso supervisionado é obtido da imagem original atrasada e no caso autodidata, pode ser estimado usando-se características estatísticas da imagem original como se pode verificar na Figura 1.3, onde os dois casos estão esquematizados.

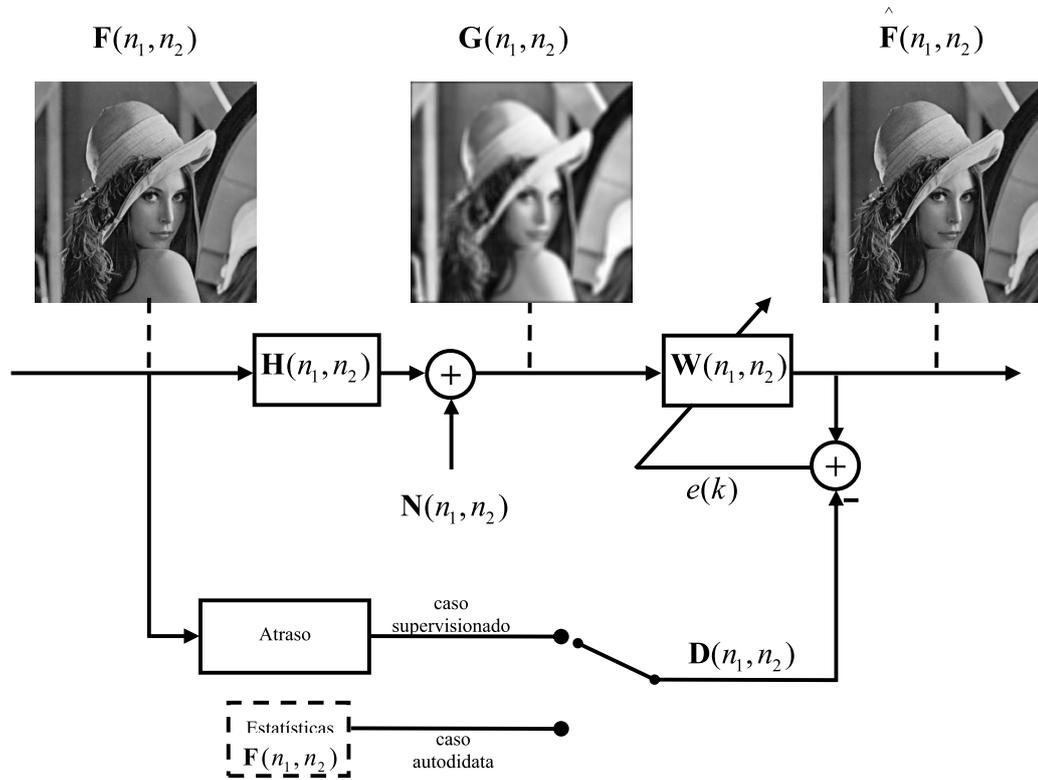


Figura 1.3: Diagrama em blocos da equalização no caso generalizado

O filtro adaptativo bidimensional pode ser considerado como uma extensão de sua versão de uma dimensão (1-D) em que há diferenças para a geração do vetor de entrada e das varreduras possíveis para sua geração. Em [Hadhoud e Thomas, 1988], por exemplo, o algoritmo LMS (*Least Mean Square*), um dos mais populares em filtragem adaptativa supervisionada, foi estendido para o caso bidimensional e aplicado em restauração de imagens. No entanto, em grande parte das aplicações práticas, não se tem a imagem original, o que remete o problema ao uso de uma solução autodidata. Dentre as soluções desse contexto, vale ressaltar a proposta em [Vural e Sethares, 2002]. Nessa referência, o algoritmo do módulo constante (CMA - *Constant Modulus Algorithm*) é estendido para restauração de imagens. Cabe observar que o CMA foi proposto em [Godard, 1980] para equalização de canais de comunicação e para o seu funcionamento correto, é necessário conhecer características estatísticas do sinal transmitido. Para se utilizar o CMA em imagens, foi proposto em [Vural e Sethares, 2002] um mapeamento da imagem original antes de sua transmissão. Assim, por exemplo, uma imagem de dois bits/pixel deve ter seus quatro níveis mapeados em ± 1 e ± 3 . Dessa forma, a imagem mapeada pode ser interpretada como um sinal de comunicação com modulação do tipo 4-PAM (*Pulse*

Amplitude Modulation). Com esse mapeamento, é possível tratar a imagem como um sinal de comunicação e conseqüentemente usar o CMA de forma adequada. Além disso, a degradação causada pela PSF pode ser interpretada como a interferência intersimbólica (ISI - *intersymbol interference*) causada por um canal de comunicação. Uma outra questão importante é a janela da imagem que deve ser definida em cada iteração para gerar o sinal de entrada do filtro adaptativo. A forma como essa janela se movimenta na imagem e as condições de borda são fundamentais para se obter um bom desempenho do algoritmo [Vural e Sethares, 2002, Hadhoud e Thomas, 1988, Samarasinghe e Kennedy, 2008].

Mesmo o CMA sendo uma boa solução, a convergência para a imagem original se torna difícil caso a imagem tenha muitos bits/pixel ou ocorra uma forte degradação. Usando o mapeamento de [Vural e Sethares, 2002], uma imagem com muitos bits/pixel é equivalente a ter um sinal de comunicação pertencente a uma constelação com um número elevado de símbolos. O desempenho do CMA é tanto pior, quanto mais símbolos existirem na constelação, sendo que o melhor desempenho ocorre para 2-PAM, cuja constelação é formada pelos símbolos ± 1 [Johnson et al., 1998]. Outra característica importante para um algoritmo adaptativo é o *tracking* que consiste na capacidade do algoritmo se adaptar a mudanças no meio de transmissão. Dessa forma, um algoritmo autodidata que apresente um bom desempenho para constelações de módulo não constante e que tenha uma boa capacidade de *tracking* é de grande interesse para restauração de imagens.

Alguns pontos ainda podem ser explorados visando uma melhoria na reconstrução. Abordagens quanto ao posicionamento da janela, considerações de borda e varredura podem melhorar a convergência do algoritmo. Além disso, soluções baseadas em equalizadores de decisão realimentada (DFE - *decision feedback equalizer*) podem apresentar melhorias no algoritmo de [Vural e Sethares, 2002].

1.3 Objetivos

Nesta dissertação de mestrado, utiliza-se o mapeamento de uma imagem em sinais de comunicação do tipo PAM antes de sua transmissão / degradação para então se utilizar técnicas de equalização autodidata na sua restauração. Para isso, são considerados filtros empregados em equalização de canais de comunicação, como o equalizador linear transversal e o equalizador de decisão realimentada [Silva, 2005]. Além disso, são utilizados algoritmos que apresentem boa capacidade de *tracking* e que consigam um bom desempenho para imagens com muitos bits/pixel, o que é equivalente a se ter muitos símbolos na constelação. Neste contexto, utiliza-se o algoritmo multimódulo regional, proposto em [Mendes Filho et al., 2009]. Por fim, é feito um estudo da influência da seleção e movimentação da janela na imagem e das condições de borda no desempenho dos algoritmos adaptativos.

1.4 Justificativa

Na literatura existem inúmeras técnicas de desconvolução autodidata de imagens. Para se ter uma ideia, o livro [Campisi e Egiazarian, 2007] contém dez capítulos de diferentes autores, que propõem técnicas de restauração. Além disso, Kundur e Hatzinakos [1996a] descrevem o estado da arte de desconvolução autodidata de imagens. Em geral, essas técnicas são complexas e apresentam um custo computacional elevado. Técnicas adaptativas para processamento de imagens apresentam a vantagem de serem mais simples computacionalmente e vêm sendo exploradas na literatura [Hadhoud e Thomas, 1988, Vural e Sethares, 2002]. Dentre essas técnicas, destacam-se os algoritmos de equalização autodidata de canais de comunicação [Silva, 2005]. Embora existam alguns trabalhos na literatura sobre a extensão do CMA para restauração de imagens, ainda há um vasto trabalho de extensão a ser feito. Por exemplo, a diversidade espacial, o equalizador de decisão realimentada e o uso de algoritmos que apresentem desempenho superior ao do CMA para sinais de módulo não constante não foram explorados em [Vural e Sethares, 2002]. Diante disso, um estudo da viabilidade de extensão dessas técnicas para a restauração de imagens é algo de interesse e justifica esta dissertação.

1.5 Contribuições da Dissertação

As principais contribuições deste trabalho são enumeradas a seguir.

1. **Relação do canal bidimensional com o canal de comunicação.** Utilizando-se um filtro adaptativo supervisionado, identificou-se o canal de comunicação (unidimensional) equivalente à PSF. Com esse estudo, pode se entender as particularidades de canais bidimensionais e o porquê da dificuldade de equalização para certos tipos de PSFs.
2. **Erro quadrático espacial.** O erro quadrático foi mostrado para cada pixel da imagem. Os diagramas de erros gerados facilitam notar os problemas na equalização e mostram que a maioria dos erros acontecem nas áreas de transições rápidas, o que é bem coerente com o uso de algoritmos adaptativos. Esta análise permite também efetuar uma boa comparação qualitativa da convergência da reconstrução da imagem.
3. **Proposta de uma varredura combinada.** A varredura proposta no Capítulo 5 tem o objetivo de minimizar as alterações bruscas da matriz de entrada do algoritmo adaptativo com o uso de uma combinação de varreduras alternadas vertical e horizontal. Isto faz com que o algoritmo tenha condições mais adequadas de convergência, melhorando a reconstrução.

4. **Proposta do TDRMMA (*Two-Dimensional Region-based Multimodulus Algorithm*)**. A proposta da extensão do RMMA para o caso bidimensional vem para ajudar a resolver o problema de como lidar com a equalização de imagens mapeadas em uma grande constelação. O uso do TDRMMA gerou melhorias perceptíveis em comparação a algoritmos adaptativos da literatura [Vural e Sethares, 2002, Kara e Vural, 2008]. Além disso a utilização do algoritmo em um DFE melhorou muito a reconstrução da imagem tendo em vista que a estrutura do DFE é mais adequada para PSFs difíceis de equalizar.

Algumas dessas contribuições foram publicadas no artigo:

- R. Abreu e M. T. M. Silva. *A multimodulus algorithm for blind image deconvolution. In Proceedings of International Workshop on Telecommunications (IWT-11)*, 2011, pp. 35-41.

1.6 Organização da Dissertação

A dissertação está dividida em 6 capítulos. No Capítulo 2, algoritmos de equalização supervisionada e autodidata são revisitados, considerando-se o equalizador linear transversal e o equalizador de decisão realimentada, a fim de serem estendidos para o caso bidimensional. No Capítulo 3, os tipos de degradação mais comuns em imagens são apresentados e simulados com modelos matemáticos. No Capítulo 4, algoritmos bidimensionais para imagens são apresentados em conjunto com particularidades da janela, da inicialização e da varredura. Além disso, é considerada uma vetorização da matrizes com objetivo de utilizar algoritmos unidimensionais. No Capítulo 5, conceitos de equalização de canais de comunicação ainda não explorados em processamento de imagens como o equalizador de decisão realimentada e o algoritmo do módulo constante regional de [Mendes Filho et al., 2009] são estendidos para o caso bidimensional. São apresentadas inúmeras simulações em que é possível verificar o bom funcionamento dessas técnicas para restauração de imagens. Finalmente no Capítulo 6, são apresentadas as conclusões e as considerações para trabalhos futuros.

Capítulo 2

Equalização de canais de comunicação

Para reconstruir imagens utilizando filtros adaptativos bidimensionais, é necessário revisar alguns conceitos e algoritmos utilizados na equalização de canais de comunicação. Neste capítulo, revisitam-se equalizadores adaptativos com o objetivo de estendê-los para a reconstrução de imagens. Na Seção 2.1, são consideradas duas estruturas de equalizadores amplamente utilizados na prática: o equalizador linear transversal (LTE - *Linear Transversal Equalizer*) e o equalizador de decisão realimentada (DFE - *Decision Feedback Equalizer*). As Seções 2.2, 2.3 e 2.4 contêm algoritmos adaptativos para a adaptação desses equalizadores.

2.1 Formulação do problema - o LTE e o DFE

Os filtros adaptativos são utilizados em várias áreas da Engenharia Elétrica. As aplicações são normalmente classificadas em quatro tipos de acordo com o objetivo do filtro e o tipo de relação que ele irá possuir com o sinal de entrada. São elas: (a) identificação de sistemas; (b) cancelamento de ruído; (c) predição; e (d) equalização [Haykin, 2001].

No caso de reconstrução de imagens, a utilização de filtros adaptativos para equalização de canais de comunicação pode ser estendida para o caso bidimensional e por isso será detalhada neste capítulo.

A Figura 2.1 contém um diagrama de blocos de um sistema de comunicação simplificado, onde o equalizador linear transversal (LTE - *Linear Transversal Equalizer*) é usado para recuperar o sinal transmitido $a(n)$. Esse sinal é degradado pela resposta impulsiva do canal $h(n)$ mais um ruído aditivo $\eta(n)$ e chega ao receptor de forma distorcida, sinal $u(n)$. A cada instante de tempo n , a saída $y(n)$ do equalizador é obtida pelo produto interno de um vetor regressor de entrada

$$\mathbf{u}(n) = [u(n) \quad u(n-1) \quad \cdots \quad u(n-M+1)]^T$$

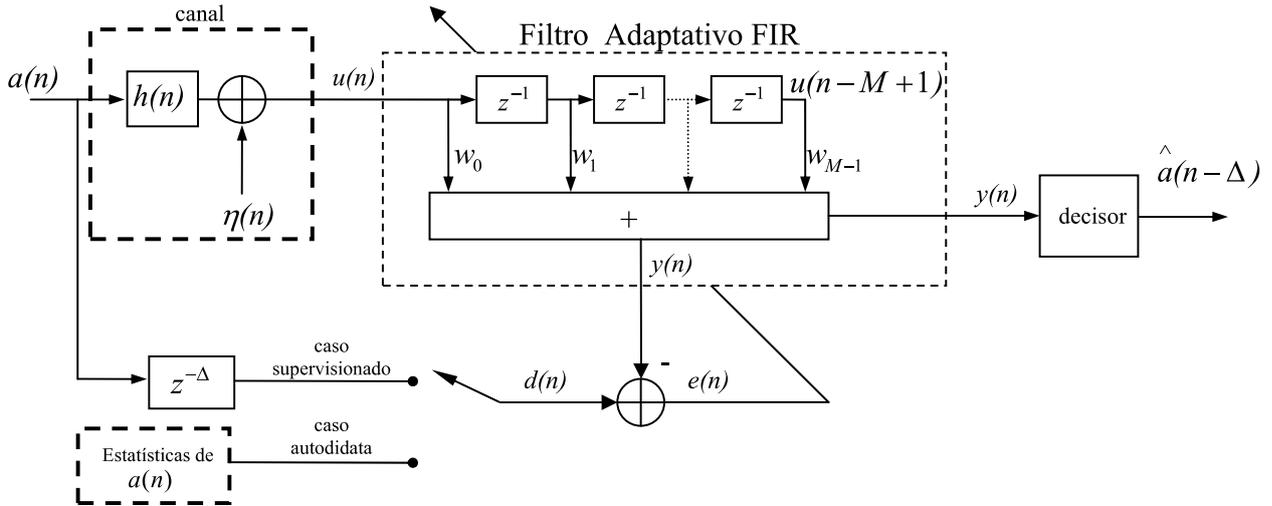


Figura 2.1: Sistema de comunicação simplificado com um equalizador linear transversal.

com o vetor de coeficientes do filtro

$$\mathbf{w} = [w_0 \ w_1 \ \cdots \ w_{M-1}]^T,$$

ou seja, $y(n) = \mathbf{u}^T(n)\mathbf{w}$. Cabe observar que n aqui representa tempo e não espaço como no processamento de imagens e que M representa o comprimento da resposta impulsiva do equalizador. O sinal $y(n)$ entra então num decisor, cuja saída deve ser uma versão atrasada do sinal transmitido, ou seja, $\hat{a}(n - \Delta)$.

A atualização dos coeficientes do filtro é realizada através do erro de estimação e pode ser feita de forma supervisionada ou autodidata. No caso supervisionado, o sinal transmitido é usado como referência e o erro de estimação é calculado efetuando-se a diferença entre sinal transmitido atrasado $d(n) = a(n - \Delta)$ e a sua estimativa $y(n)$. No caso autodidata, em que não se possui uma sequência de treinamento para referência, o sinal desejado pode ser estimado através de estatísticas de ordem superior do sinal transmitido [Silva, 2005].

As soluções que utilizam um equalizador linear transversal nem sempre apresentam bons resultados diante de canais com respostas impulsivas longas e esparsas, canais de fase não-mínima, com nulos espectrais ou com não-linearidades [Szczecinski e Gei, 2002]. Uma solução mais interessante para esses tipos de canais é o equalizador de decisão realimentada (DFE - *Decision Feedback Equalizer*) pois sua estrutura forma um sistema não-linear capaz de lidar com diferentes tipos de distorções de canais [Szczecinski e Gei, 2002]. Além disso, os coeficientes do DFE podem ser atualizados tanto de forma supervisionada ou autodidata.

A estrutura do DFE é mostrada na Figura 2.2. Neste caso, a resposta impulsiva do filtro direto (\mathbf{w}_f) e do filtro de realimentação (\mathbf{w}_b) contém respectivamente M_f e M_b coeficientes. O sinal $u(n)$ que chega ao receptor, entra no filtro direto \mathbf{w}_f e gera em sua saída a estimativa $y_f(n)$. O sinal $y_f(n)$ é então adicionado ao sinal $y_b(n)$, vindo do filtro de realimentação \mathbf{w}_b ,

e então o sinal resultante $y(n)$ passa por um decisor. Cabe observar que o fato do decisor ser definido por uma função não-linear, que neste caso está na malha de realimentação, têm-se que esse equalizador seja também não-linear. Além disso, a influência do filtro realimentação \mathbf{w}_b inclui as decisões passadas do decisor para a estimativa de novos símbolos, minimizando sobremaneira a interferência intersimbólica, quando comparado ao LTE.

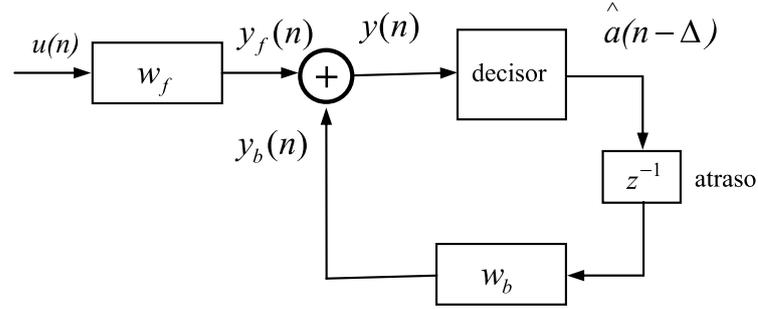


Figura 2.2: Diagrama em blocos de um DFE

Definindo o vetor regressor do filtro direto como

$$\mathbf{u}(n) = [u(n) \quad u(n-1) \quad \dots \quad u(n-M_f+1)]^T, \quad (2.1)$$

e seu vetor de coeficientes como

$$\mathbf{w}_f = [w_{f,0} \quad w_{f,1} \quad \dots \quad w_{f,M_f-1}]^T \quad (2.2)$$

obtem-se $y_f(n) = \mathbf{u}^T(n)\mathbf{w}_f$. De maneira análoga, define-se o vetor de entrada e de coeficientes do filtro de realimentação como

$$\hat{\mathbf{a}}_\Delta(n) = [\hat{a}(n-\Delta-1) \quad \hat{a}(n-\Delta-2) \quad \dots \quad \hat{a}(n-\Delta-M_b)]^T, \quad (2.3)$$

e

$$\mathbf{w}_b = [w_{b,1} \quad w_{b,2} \quad \dots \quad w_{b,M_b}]^T, \quad (2.4)$$

o que leva a $y_b(n) = \hat{\mathbf{a}}_\Delta^T(n)\mathbf{w}_b$. Assim, a saída do DFE é calculada como

$$y(n) = y_f(n) + y_b(n). \quad (2.5)$$

Para uma formulação mais direta, pode-se concatenar as entradas do filtro direto e de

realimentação como também seus coeficientes, considerando desta forma

$$\mathbf{u}_{fb}(n) = [\mathbf{u}^T(n) \quad \hat{\mathbf{a}}_{\Delta}^T(n)]^T \quad (2.6)$$

e

$$\mathbf{w}_{fb} = [\mathbf{w}_f^T \quad \mathbf{w}_b^T]^T. \quad (2.7)$$

Assim, a saída do DFE pode ser calculada como

$$y(n) = \mathbf{u}_{fb}^T(n) \mathbf{w}_{fb}. \quad (2.8)$$

2.2 A Solução de Wiener

A solução de Wiener ou filtro ótimo minimiza o erro quadrático médio e representa um limite para a equalização linear [Widrow e Stearns, 1985, Haykin, 2001]. A função custo do erro quadrático médio é dada por

$$J_{MSE} = E[e^2(n)] \quad (2.9)$$

sendo $E[\cdot]$ o operador esperança matemática, $e(n)$ o erro de estimação $e(n) = d(n) - y(n)$ e $d(n)$ o sinal desejado, que, em equalização supervisionada é o sinal transmitido atrasado, ou seja, $a(n - \Delta)$ [Widrow e Stearns, 1985].

Tomando o vetor gradiente da função custo para o caso real, obtém-se

$$\nabla_{\mathbf{w}} J_{MSE} = \frac{\partial J_{MSE}}{\partial \mathbf{w}} = 2(\mathbf{R}\mathbf{w} - \mathbf{p}), \quad (2.10)$$

sendo \mathbf{R} a matriz de autocorrelação do sinal de entrada e \mathbf{p} o vetor correlação cruzada entre o sinal de entrada e o sinal desejado, ou seja,

$$\mathbf{R} = E[\mathbf{u}(n)\mathbf{u}^T(n)] \quad (2.11)$$

e

$$\mathbf{p} = E[d(n)\mathbf{u}(n)]. \quad (2.12)$$

Igualando (2.10) a zero, se obtém a minimização da função custo e se chega à definição da equação de Wiener-Hopf dada por

$$\mathbf{w}_o = \mathbf{R}^{-1}\mathbf{p}, \quad (2.13)$$

em que \mathbf{w}_o são os coeficientes ótimos [Widrow e Stearns, 1985, Haykin, 2001].

No caso do DFE, como $\mathbf{u}_{fb}(n)$ e \mathbf{w}_{fb} são definidos respectivamente em (2.6) e (2.7) a equação (2.13) continua válida, mas as definições da matriz \mathbf{R} e do vetor \mathbf{p} , passam a ser $\mathbf{R} = E[\mathbf{u}_{fb}(n)\mathbf{u}_{fb}^T(n)]$ e $\mathbf{p} = E[\mathbf{u}_{fb}(n)d(n)]$. Cabe observar que neste caso, o vetor \mathbf{w} será

formado pelos coeficientes do filtro direto, seguidos pelos coeficientes do filtro de realimentação como em (2.7).

Embora o filtro de Wiener seja ótimo, é necessário assumir que os sinais $u(n)$ e $d(n)$ sejam estacionários. Além disso, a matriz \mathbf{R} e o vetor \mathbf{p} devem ser calculados previamente, o que faz com que seja necessário fazer a aquisição do sinal antes da filtragem sem contar o fato de que a complexidade aumenta consideravelmente com a ordem do filtro devido à inversa da matriz \mathbf{R} . Dessa forma, sua aplicação em processamento de imagens é limitada, uma vez que o sinal a ser filtrado é não-estacionário. Por esse motivo, a partir da próxima seção são revisitados algoritmos do gradiente estocástico com o intuito de serem estendidos para o processamento de imagens.

2.3 Equalização supervisionada - o algoritmo LMS

A necessidade de um sinal estacionário para o bom funcionamento do filtro de Wiener é uma condição que não se aplica às imagens reais que devem ser reconstruídas. Por esse motivo, uma proposta de algoritmo que se adapte às variações locais cabe muito bem para esta aplicação. Esta deficiência do filtro de Wiener diante da restauração faz com que ele seja usado em imagens mais comumente na diminuição de ruído [Lim, 1990]. Diante disso, é que se revisita aqui o algoritmo LMS (*Least Mean Square*).

O LMS, proposto por Widrow e Hoff em 1959, é uma versão estocástica do algoritmo do Gradiente Determinístico (*Steepest Descent*). Ele usa um procedimento iterativo que executa sucessivas correções aos coeficientes no sentido negativo do vetor gradiente a fim de minimizar o erro quadrático. Comparado com outros algoritmos, o algoritmo LMS é relativamente simples e não requer inversão de matrizes.

No método do Gradiente Determinístico, a equação de atualização do vetor de coeficientes é dada por

$$\mathbf{w}(n) = \mathbf{w}(n-1) - \frac{1}{2}\mu \nabla_{\mathbf{w}} J_{MSE}, \quad (2.14)$$

em que $\mathbf{w}(n)$ é o vetor de coeficientes do filtro na n -ésima iteração, μ é o passo de adaptação e $\nabla_{\mathbf{w}} J_{MSE}$ é definido na Equação (2.10).

O algoritmo do Gradiente Determinístico minimiza de forma exata o J_{MSE} . Escolhendo-se adequadamente o passo de adaptação μ , é possível demonstrar que $\mathbf{w}(n)$ converge para a solução de Wiener. Embora seja um procedimento iterativo que evita o cálculo da inversa da matriz \mathbf{R} , o algoritmo (2.14) ainda necessita conhecer a matriz \mathbf{R} e o vetor \mathbf{p} e, como a solução de Wiener, não é adequado para sinais não-estacionários.

O algoritmo LMS simplifica o algoritmo do Gradiente Determinístico adotando valores instantâneos para \mathbf{R} e \mathbf{p} , ou seja,

$$\hat{\mathbf{R}} = \mathbf{u}(n)\mathbf{u}^T(n) \quad (2.15)$$

e

$$\hat{\mathbf{p}} = d(n)\mathbf{u}(n). \quad (2.16)$$

Substituindo (2.15) e (2.16) em (2.10) e o resultado em (2.14) se obtém a equação de atualização dos coeficientes do algoritmo LMS, dada por

$$\mathbf{w}(n) = \mathbf{w}(n-1) + \mu e(n)\mathbf{u}(n). \quad (2.17)$$

O vetor de coeficientes pode ser inicializado com valores arbitrários, mas normalmente considera-se $\mathbf{w}(0) = \mathbf{0}$, sendo $\mathbf{0}$ um vetor coluna nulo. O algoritmo LMS está resumido no procedimento mostrado na Tabela 2.1.

Inicialização do algoritmo: $\mathbf{w}(0) = \mathbf{0}, \mathbf{u}(0) = \mathbf{0}$
Para cada instante de tempo, $n = 1, 2, \dots$, calcule: $y(n) = \mathbf{w}^T(n-1)\mathbf{u}(n)$ $e(n) = d(n) - y(n)$ $\mathbf{w}(n) = \mathbf{w}(n-1) + \mu e(n)\mathbf{u}(n)$

Tabela 2.1: Sumário do algoritmo LMS.

Através de uma análise estatística de segunda ordem [Gazor e Farhang-Boroujeny, 1992, Farhang-Boroujeny, 1998], é possível mostrar que o algoritmo LMS converge na média quadrática se

$$0 < \mu < \frac{2}{3\lambda_{\max}}, \quad (2.18)$$

em que λ_{\max} é o maior autovalor da matriz de autocorrelação \mathbf{R} . Um intervalo mais restrito é dado por

$$0 < \mu < \frac{2}{3Tr[\mathbf{R}]}, \quad (2.19)$$

em que $Tr[\mathbf{R}]$ é o traço da matriz de autocorrelação [Gazor e Farhang-Boroujeny, 1992, Farhang-Boroujeny, 1998]. Como $Tr[\mathbf{R}] = M\sigma_u^2$, sendo σ_u^2 a variância do sinal de entrada, o intervalo do passo de adaptação depende explicitamente da potência do sinal de entrada, o que pode ser um problema no caso desse sinal ser não estacionário. Por este motivo, foi proposto o algoritmo LMS normalizado, o NLMS (*Normalized LMS*), que normaliza o passo de adaptação de acordo com a potência do sinal de entrada do filtro, ou seja,

$$\mu = \frac{\tilde{\mu}}{\|\mathbf{u}(n)\|^2 + \delta}, \quad (2.20)$$

sendo δ uma constante pequena positiva, usada para evitar divisão por zero $\tilde{\mu}$ um passo normalizado. Desta maneira, o algoritmo NLMS pode ser resumido no procedimento mostrado

na Tabela 2.2.

<p>Inicialização do algoritmo:</p> $\mathbf{w}(0) = \mathbf{0}, \mathbf{u}(0) = \mathbf{0}$
<p>Para cada instante de tempo, $n = 1, 2, \dots$, calcule:</p> $y(n) = \mathbf{w}^T(n-1)\mathbf{u}(n)$ $e(n) = d(n) - y(n)$ $\mathbf{w}(n) = \mathbf{w}(n-1) + \frac{\tilde{\mu}}{\ \mathbf{u}(n)\ ^2 + \delta} e(n)\mathbf{u}(n)$

Tabela 2.2: Sumário do algoritmo NLMS.

Com a normalização da potência do sinal de entrada, é possível mostrar que o algoritmo NLMS converge se [Gazor e Farhang-Boroujeny, 1992, Farhang-Boroujeny, 1998]

$$0 < \tilde{\mu} < 2. \quad (2.21)$$

Comparando (2.21) com (2.19) é possível observar que a escolha do passo de adaptação deixou de depender de potência do sinal de entrada, sendo mais adequado para sinais não-estacionários.

2.4 Equalização autodata com algoritmos de módulo constante

Na formulação do problema da reconstrução de imagens descrita no Capítulo 1 se nota que na maioria das situações práticas não há referência da imagem original. Esta seção revisita alguns algoritmos de módulo constante que serão a base utilizada para a equalização autodata bidimensional.

2.4.1 O algoritmo do módulo constante - CMA

Na equalização autodata como descrito na Figura 2.1, há pouca informação disponível do sinal desejado e da distorção exercida pelo canal. O sinal desejado pode ser estimado através de estatísticas de ordem superior do sinal transmitido. Neste contexto, um dos algoritmos mais conhecidos na literatura é o algoritmo do módulo constante (CMA - *Constant Modulus Algorithm*) que minimiza a seguinte função custo

$$J_{CMA} = E[(r - y^2(n))^2] \quad (2.22)$$

em que r é uma constante que depende de estatísticas de ordem superior do sinal transmitido definida como

$$r = \frac{E[a^4(n)]}{E[a^2(n)]}. \quad (2.23)$$

Definindo o erro de estimação como

$$e(n) = [r - y^2(n)]y(n), \quad (2.24)$$

a equação de atualização do CMA é idêntica à (2.17).

Diferente do algoritmo LMS, o vetor de coeficientes do CMA não pode ser inicializado com zeros pois neste caso não ocorre mais a atualização dos coeficientes do filtro. O algoritmo é normalmente inicializado com um vetor pino, ou seja, $\mathbf{w}(0) = [0 \ \dots \ 0 \ 1 \ 0 \ \dots \ 0]^T$. O CMA está resumido no procedimento mostrado na Tabela 2.3.

<p>Inicialização do algoritmo:</p> $r = \frac{E[a^4(n)]}{E[a^2(n)]}$ $\mathbf{w}(0) = [0 \ \dots \ 0 \ 1 \ 0 \ \dots \ 0]^T, \mathbf{u}(0) = \mathbf{0}$
<p>Para cada instante de tempo, $n = 1, 2, \dots$, calcule:</p> $y(n) = \mathbf{w}^T(n-1)\mathbf{u}(n)$ $e(n) = (r - y^2(n))y(n)$ $\mathbf{w}(n) = \mathbf{w}(n-1) + \mu e(n)\mathbf{u}(n)$

Tabela 2.3: Sumário do CMA.

Como a função custo do CMA é não convexa, ele pode convergir para mínimos locais [Johnson et al., 1998]. Além disso, se o passo de adaptação não for escolhido adequadamente ou se a inicialização estiver longe da solução ótima, o CMA pode divergir. É devido a este problema que Miranda et al. [2008] propuseram o DM-CMA (*Dual-Mode CMA*) que é revisitado a seguir.

2.4.2 O algoritmo do módulo constante que evita divergência - DM-CMA

A atualização do vetor de coeficientes do algoritmo DM-CMA é dada por

$$\mathbf{w}(n) = \mathbf{w}(n-1) + \frac{\tilde{\mu}}{\|\mathbf{u}(n)\|^2 + \delta} [d(n) - y(n)]\mathbf{u}(n) \quad (2.25)$$

em que $d(n)$ representa uma estimativa do sinal transmitido. Em [Miranda et al., 2008], o sinal $d(n)$ é calculado como

$$d(n) = \begin{cases} x(n)y(n), & \text{se } x(n) \geq 0 ; \\ 0, & \text{caso contrário} \end{cases}, \quad (2.26)$$

em que

$$x(n) = \frac{3\sigma_a^2 - y^2(n)}{3\sigma_a^2 - r} \quad (2.27)$$

sendo σ_a^2 é a variância do sinal transmitido $a(n)$.

Cabe observar que $y(n)$ também representa uma estimativa do sinal transmitido. Dessa forma, quando $d(n)$ e $y(n)$ tiverem o mesmo sinal, $x(n)$ será positivo e haverá consistência entre as duas estimativas. Neste caso, o algoritmo trabalha na chamada região de interesse como uma versão normalizada do CMA. Se $y(n)$ e $d(n)$ tiverem sinais opostos então $x(n)$ é negativo e $d(n) = 0$. Em outras palavras, a estimativa $x(n)y(n)$ é rejeitada pois o algoritmo deixa a região de interesse. Neste outro modo de operação, a adaptação do vetor de coeficientes é tal que faz com que o algoritmo volte a trabalhar na região de interesse após um intervalo de tempo. A prova matemática da estabilidade do DM-CMA se torna complexa devido a mudanças de estados do algoritmo e à natureza não-linear da função custo, porém em [Miranda et al., 2008] se pode verificar uma variedade de simulações em que o DM-CMA não diverge. O algoritmo está resumido no procedimento mostrado na Tabela 2.4.

2.4.3 O algoritmo do módulo constante para o DFE - DFE-CMA

Como explicado anteriormente, alguns tipos de canais complexos podem não ser equalizados de maneira efetiva pelo CMA em uma estrutura LTE. Levando este fato em consideração, o DFE adaptado com o CMA pode ser usado como uma solução ao problema. Neste caso, o algoritmo pode ser implementado de maneira simples pela concatenação dos vetores $\mathbf{u}(n)$ e $\hat{\mathbf{a}}_{\Delta}(n)$ gerando o vetor $\mathbf{u}_{fb}(n)$ e pela concatenação dos vetores de coeficientes do filtro direto \mathbf{w}_f e do filtro de realimentação \mathbf{w}_b gerando o vetor \mathbf{w}_{fb} como feito nas equações (2.6) e (2.7). Após essas concatenações, a equalização pode ser realizada seguindo o procedimento da Tabela 2.3. No entanto, quando se trabalha com DFEs autodatas, é possível se obter o que se chama de solução degenerada em que a saída do equalizador se torna independente do sinal de entrada devido a não-linearidade do equalizador e à função custo do módulo constante [Szczecinski e Gei, 2002]. Em uma modulação 2-PAM, por exemplo, os coeficientes do filtro direto e de realimentação podem, em algum momento, assumir respectivamente o formato

$$\mathbf{w}_f(n) = \mathbf{0} \quad \text{e} \quad \mathbf{w}_b(n) = \left[0 \quad \dots \quad 0 \quad \pm 1 \quad 0 \quad \dots \quad 0 \right]^T. \quad (2.28)$$

Em um caso como este, o erro do CMA definido na Equação (2.24) fica igual a 0 e então não ocorre qualquer atualização nos coeficientes do algoritmo. Uma forma de se evitar este

<p>Inicialização do algoritmo:</p> $r = \frac{E[a^4(n)]}{E[a^2(n)]}$ $\mathbf{w}(0) = [0 \ \dots \ 0 \ 1 \ 0 \ \dots \ 0]^T, \mathbf{u}(n) = \mathbf{0}$ $0 < \tilde{\mu} < 2$ <p>δ: constante positiva pequena</p>
<p>Para cada instante de tempo, $n = 1, 2, \dots$, calcule:</p> $y(n) = \mathbf{w}^T(n-1)\mathbf{u}(n)$ $x(n) = \frac{3\sigma_a^2 - y^2(n)}{3\sigma_a^2 - r}$ <p>se $x(n) \geq 0$</p> $d(n) = x(n)y(n)$ <p>caso contrário</p> $d(n) = 0$ <p>fim</p> $\bar{e}(n) = d(n) - y(n)$ $\mathbf{w}(n) = \mathbf{w}(n-1) + \frac{\tilde{\mu}}{\ \mathbf{u}(n)\ ^2 + \delta} \bar{e}(n)\mathbf{u}(n)$

Tabela 2.4: Sumário do DM-CMA.

problema é a inserção de algumas restrições ao algoritmo que impeçam que situações como esta aconteçam. Uma das formas de se evitar a solução degenerada é o uso do DFE-CMA-FB proposto em [Szczecinski e Gei, 2002]. Nesse algoritmo, calcula-se em cada instante de tempo a variável $C_w(n) = \|\mathbf{w}_b(n-1)\|^2 - E_{y_f}(n)$, em que $E_{y_f}(n)$ é uma estimativa da potência do sinal de saída do filtro direto. Em [Szczecinski e Gei, 2002] se demonstrou que para se impedir o aparecimento de soluções degeneradas se deve obedecer à restrição de que $C_w(n) \leq 0$. Dessa forma, quando $C_w(n) \leq 0$, $\lambda_L(n)$ é feito igual a zero e a atualização dos coeficientes do algoritmo fica idêntica à atualização do CMA convencional. Caso contrário, ele efetua ajustes nas atualizações dos vetores de coeficientes \mathbf{w}_f e \mathbf{w}_b através de $\lambda_L(n)$. Detalhes mais específicos da derivação do DFE-CMA-FB podem ser obtidos em [Szczecinski e Gei, 2002]. O DFE-CMA-FB está resumido no procedimento mostrado na Tabela 2.5, em que λ_f é um fator de esquecimento tal que $0 < \lambda_f < 1$, l_o é usualmente considerado igual a 2 e $\text{step}(n) = \begin{cases} 1, & \text{se } x \geq 0; \\ 0, & \text{caso contrário} \end{cases}$.

<p>Inicialização do algoritmo:</p> $\mathbf{w}_f(0) = [0 \ \dots \ 0 \ 1 \ 0 \ \dots \ 0]^T$ $\mathbf{w}_b(0) = \mathbf{0}, \mathbf{u}(0) = \mathbf{0}, \hat{\mathbf{a}}_\Delta(0) = \mathbf{0}, \mathbf{g}_f(0) = \mathbf{0}$ $E_{yf} = 0$
<p>Para cada instante de tempo, $n = 1, 2, \dots$, calcule:</p> $y_f(n) = \mathbf{w}_f^T(n-1)\mathbf{u}(n)$ $y_b(n) = \mathbf{w}_b^T(n-1)\hat{\mathbf{a}}_\Delta(n)$ $y(n) = y_f(n) + y_b(n)$ $e(n) = (r - y^2(n))y(n)$ $E_{yf}(n) = \lambda_f E_{yf}(n-1) + (1 - \lambda_f)y_f^2(n)$ $\mathbf{g}_f(n) = \lambda_f \mathbf{g}_f(n-1) + (1 - \lambda_f)y_f(n)\mathbf{u}(n)$ $C_w(n) = \ \mathbf{w}_b(n-1)\ ^2 - E_{yf}(n)$ $\lambda_L(n) = l_o\text{step}(C_w(n))$ $\mathbf{w}_f(n) = \mathbf{w}_f(n-1) + \mu[\lambda_L(n)\mathbf{g}_f(n) + e(n)\mathbf{u}(n)]$ $\mathbf{w}_b(n) = [1 - \mu\lambda_L(n)]\mathbf{w}_b(n-1) + \mu e(n)\hat{\mathbf{a}}_\Delta(n)$

Tabela 2.5: Sumário do Algoritmo DFE-CMA-FB.

2.4.4 O algoritmo multimódulo regional - RMMA

O CMA é largamente empregado até mesmo para casos de constelações de módulo não-constante como são os casos do N -PAM (*pulse amplitude modulation*) e o M -QAM (*quadrature amplitude modulation*) com $N > 2$ e $M > 4$ respectivamente. Apesar disso, o CMA só consegue atingir o MSE mínimo em casos de sinais de módulo constante e em um ambiente estacionário sem ruído [Johnson et al., 1998]. Na Figura 2.3, se tem um exemplo de como se comporta o erro do CMA descrito na Equação (2.24) em função do sinal de saída $y(n)$, quando se trabalha com um sinal 8-PAM. Mesmo quando $y(n)$ é igual a um dos símbolos da constelação, o erro de estimação é não nulo, o que faz com que o algoritmo continue efetuando a atualização dos coeficientes do filtro, dificultando sua convergência.

Com o objetivo de melhorar a eficiência do CMA para equalização de sinais de módulo não constante, um algoritmo multimódulo regional (RMMA - *regional multimodulus algorithm*) foi proposto em [Mendes Filho et al., 2009]. O RMMA ainda evita a divergência, rejeitando estimativas não consistentes do sinal transmitido devido ao uso da filosofia do DM-CMA de [Miranda et al., 2008], que foi revisitada na Seção 2.4.2. Comparando com o CMA, ele exhibe um desajuste consideravelmente baixo, convergência mais rápida, boa capacidade de *tracking* sem comprometer o custo computacional.

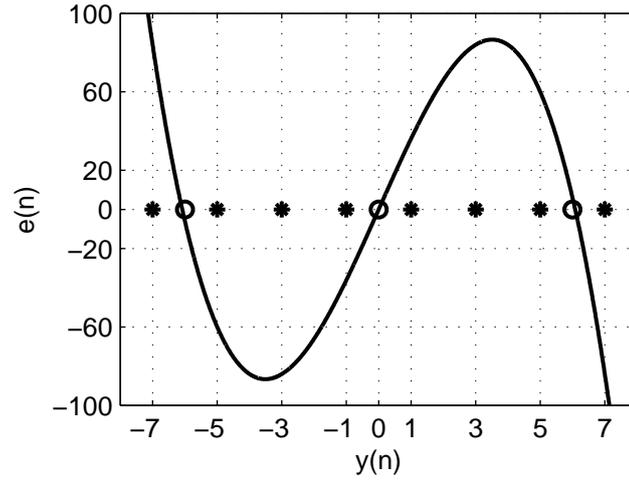


Figura 2.3: Erro do CMMA como função de $y(n)$ para sinal 8-PAM

Simplificando seus argumentos, assumindo somente entradas reais para uso posterior em imagens, a atualização dos coeficientes do RMMA é dada pela equação

$$\mathbf{w}(n) = \mathbf{w}(n-1) + \frac{\tilde{\mu}}{\delta + \|\mathbf{u}(n)\|^2} \bar{e}(n) \mathbf{u}(n), \quad (2.29)$$

em que δ é um fator de regularização (constante pequena e positiva) e $e(\bar{n})$ é o erro de estimação explicado a seguir. Considerando novamente o sinal 8-PAM ($N = 8$), se pode dividir as saídas em uma reta real dividida em regiões contendo dois símbolos da constelação, como se vê na Figura 2.4. Os centros das regiões A_k para $k = -N/4, \dots, -1, 1, \dots, N/4$ são indicados por c_k . Dada a saída $y(n)$ do equalizador e assumindo uma constelação N -PAM, com $\lceil \log_2(N) - 1 \rceil$ comparações¹, é possível identificar a região A_k à qual a saída do equalizador pertence. Usando esta informação, o erro do RMMA é definido como

$$e(\bar{n}) = |c_k| [1 - \bar{y}^2(n)] \bar{y}(n), \quad (2.30)$$

em que $\bar{y}(n) = y(n) - c_k$ é a amostra da saída transladada.

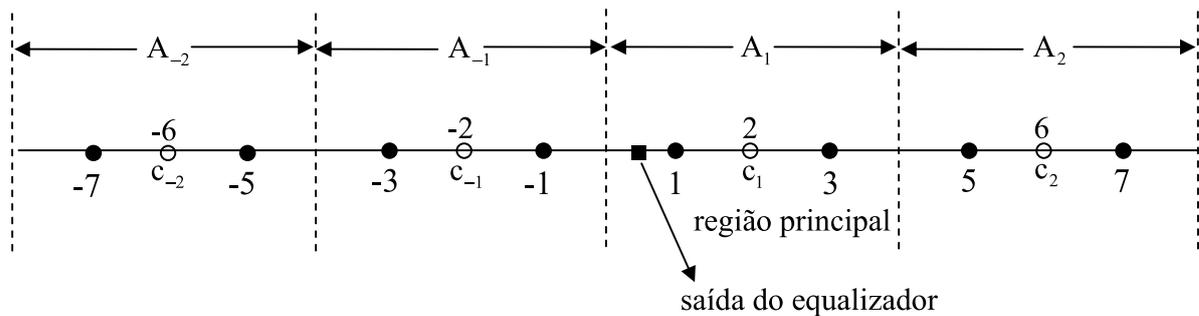


Figura 2.4: Regiões de um sinal 8-PAM

¹ $\lceil x \rceil$ representa o próximo inteiro de x .

Este erro é a repetição da forma do erro CMA (com a constante de dispersão r igual a 1) ponderado por um fator de escala como mostrado na Figura 2.5. Diferente do CMA, o erro de estimação é zero quando $y(n)$ é igual a um dos símbolos da constelação, não gerando novas modificações nos coeficientes e fazendo então com que o algoritmo se comporte melhor perante a constelações com mais símbolos. O fator de escala cria um envelope na função erro, que é essencial para a recuperação dos símbolos transmitidos, como observado em [Mendes Filho et al., 2009] e [Mendes Filho et al., 2011]. Devido ao deslocamento da estimativa $y(n)$ para a origem, tudo ocorre como se apenas os símbolos $\{\pm 1\}$ de uma constelação 2-PAM tivessem sido transmitidos. O RMMA trata constelações de módulo não constante como se fosse de módulo constante, se aproximando da solução de Wiener. Cabe observar que em [Mendes Filho et al., 2009, 2011], foram propostas soluções para melhorar a convergência do RMMA. No entanto, essas soluções não foram utilizadas nas simulações com imagens do Capítulo 5 e por isso também não foram consideradas neste capítulo. O RMMA está resumido no procedimento mostrado na Tabela 2.6.

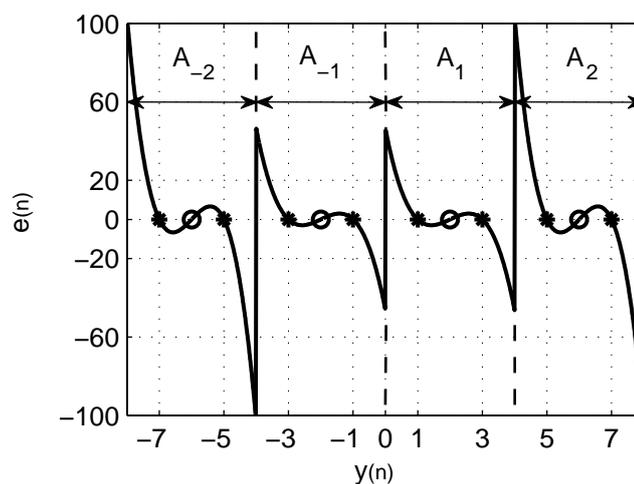


Figura 2.5: Erro do RMMA como função de $y(n)$ para sinal 8-PAM

Para verificar o bom funcionamento do RMMA em comparação ao CMA, foi feita uma simulação em que um sinal do tipo 8-PAM foi transmitido pelo canal $H(z) = 0.3 + z^{-1} + 0.3z^{-2}$ na ausência de ruído. Na Figura 2.6, são mostradas curvas do erro quadrático médio de decisão ao longo das iterações. Observa-se que o RMMA atinge a solução de Wiener que apresenta um MSE de aproximadamente -40 dB enquanto o CMA atinge apenas -22 dB. O CMA fica mais distante de Wiener quanto maior o número de símbolos da constelação. Em contrapartida, isso não acontece com o RMMA, o que pode representar uma solução interessante para a restauração de imagens.

<p>Inicialização do algoritmo:</p> $\mathbf{w}(0) = [0 \ \dots \ 0 \ 1 \ 0 \ \dots \ 0]^T, \mathbf{u}(0) = \mathbf{0}$ $0 < \tilde{\mu} < 2/ c_{\max} $
<p>Para cada instante de tempo, $n = 1, 2, \dots$, calcule:</p> $y(n) = \mathbf{w}^T(n-1)\mathbf{u}(n)$ <p>Identifique a região A_k e calcule:</p> $\bar{y}(n) = y(n) - c_k$ $\bar{x}(n) = 1.5 - 0.5\bar{y}^2(n)$ <p>se $\bar{x}(n) \geq 0$</p> $\bar{d}(n) = \bar{x}(n)\bar{y}(n)$ <p>caso contrário</p> $\bar{d}(n) = 0$ <p>fim</p> $\bar{e}(n) = c_k [\bar{d}_k(n) - \bar{y}_k(n)]$ $\mathbf{w}(n) = \mathbf{w}(n-1) + \frac{\tilde{\mu}}{\delta + \ \mathbf{u}(n)\ ^2} \bar{e}(n)\mathbf{u}(n)$

Tabela 2.6: Sumário do RMMA.

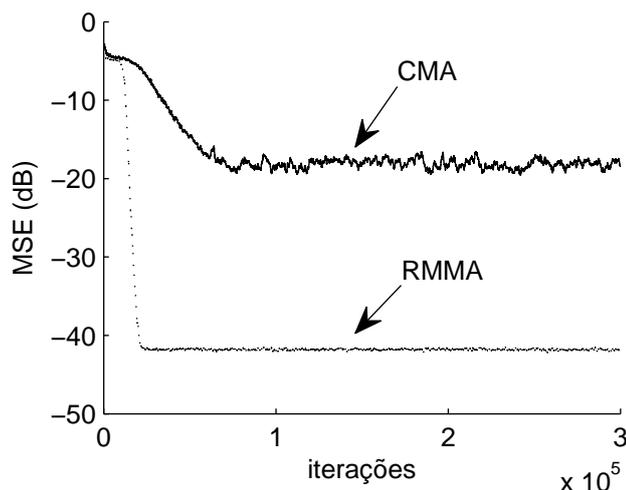


Figura 2.6: Erro quadrático médio ao longo das iterações considerando os algoritmos CMA ($\mu = 1 \times 10^{-4}$) e RMMA ($\mu = 7 \times 10^{-3}$); 8-PAM; $M = 11$; ausência de ruído; algoritmos normalizados.

2.5 Conclusão

Neste capítulo, foram revisitados algoritmos de equalização supervisionada e autodidata a fim de serem estendidos para o processamento de imagens. Diante do bom funcionamento do RMMA para constelações de módulo não constante, se considera esse algoritmo como uma boa alternativa para a restauração de imagens. Neste contexto, a utilização do DFE pode ainda gerar bons resultados principalmente para degradações complexas.

Capítulo 3

Degradação em imagens

Este Capítulo mostra os diferentes tipos de degradação que uma imagem pode sofrer. Na Seção 3.1 é considerada a degradação causada pela quantização digital. Nas Seções 3.2 e 3.3 são mostrados alguns tipos de degradação, apresentando os principais tipos de *blur* e de ruído aditivo que modelam os encontrados em aplicações reais.

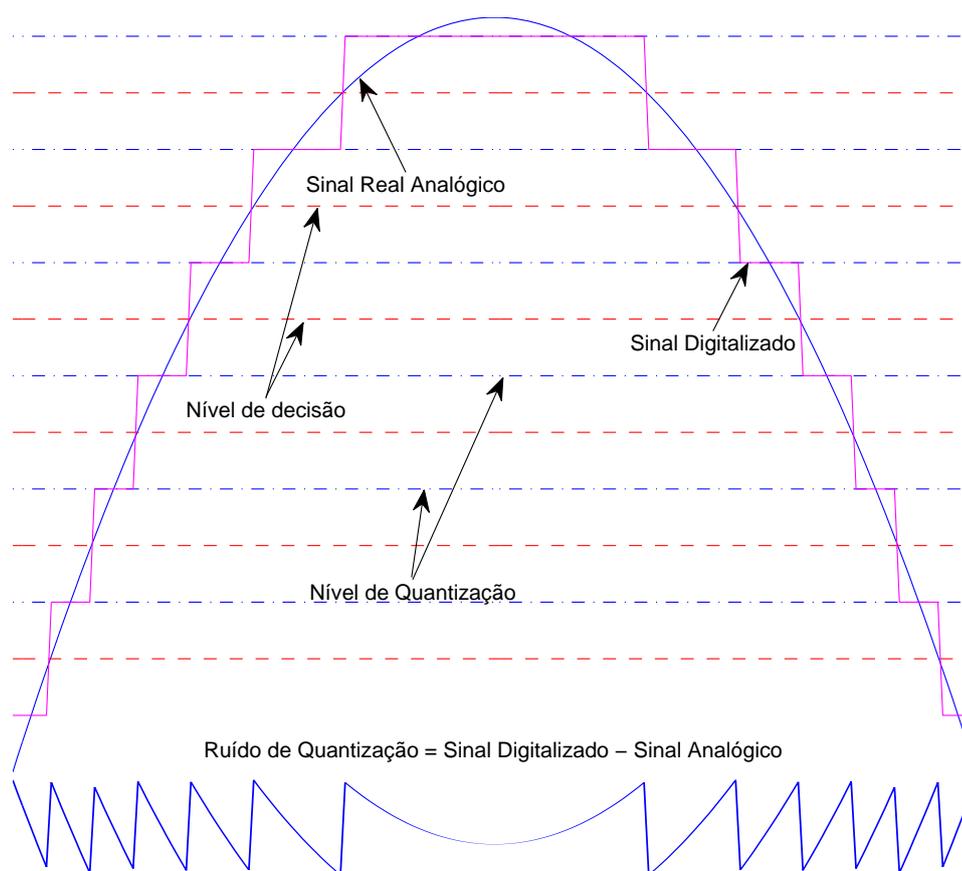
3.1 Quantização e Amostragem

O processo da aquisição da imagem é a primeira fonte de degradação pelo fato de qualquer sistema de aquisição possuir limitações físicas que geram algumas diferenças entre as imagens capturadas e as imagens reais analógicas. A digitalização de uma imagem é composta por dois processos distintos. O processo de adquirir a coordenada de um determinado ponto da imagem é a amostragem e o processo de adquirir a amplitude deste ponto é a quantização.

Uma imagem digital é normalmente quantizada em 256 níveis (os quais podem ser representados em 8 bits/pixel). Um exemplo de quantização é dado na Figura 3.1 em que o sinal analógico é digitalizado em níveis bem definidos através de níveis de decisão. Este processo cria o que se pode chamar de ruído de quantização que é o erro entre o valor real e o valor digital [Lim, 1990].

A digitalização espacial ou amostragem define um nível a uma certa coordenada da imagem. A degradação da imagem digitalizada em relação a original é tão menor quanto maior a quantidade de pixels utilizadas no processo de digitalização, gerando assim mais pontos, contornos, bordas e portanto mais definição de imagem. A Figura 3.2¹ mostra a definição de uma imagem comum a medida que se digitaliza a mesma com um menor número de pixels. Pode-se notar que a medida que se aumenta a quantidade de pixels utilizados na aquisição, a resolução da imagem se torna cada vez mais próxima a da imagem original.

¹Na Figura 3.2 se utiliza a imagem “Bellpeppers” para esclarecer o exemplo. Essa é uma imagem largamente usada em processamento de imagens como imagem de teste e pode ser obtida, por exemplo, em www.hlevkin.com/TestImages/classic.htm

Figura 3.1: *Quantização Digital*

3.2 Espalhamento (PSF-Point Spread Function)

Pode-se definir a função de espalhamento (PSF) como a resposta impulsiva bidimensional produzida na ausência de ruído [Wen e Lee, 2002]. Essa função é denotada como $\mathbf{H}(n_1, n_2)$. Mesmo possuindo métodos para se estimar a PSF, em situações práticas de restauração, é extremamente difícil se obter os parâmetros da PSF uma vez que vários tipos diferentes de degradação podem afetar a imagem simultaneamente. Na literatura existem alguns modelos de degradação que estão apropriadamente dimensionados para os casos práticos mais frequentes. Estes modelos estão descritos a seguir [Wen e Lee, 2002]:

- **Blur de movimento 1-D** é o tipo de degradação originada pela aquisição de imagens de objetos em movimentos ou pelo movimento do equipamento destinado à aquisição. Nesse caso, o movimento descrito possui apenas uma direção sendo ela horizontal ou vertical, ou seja,

$$\mathbf{H}(n) = \begin{cases} \frac{1}{L}, & \text{se } -\frac{L}{2} \leq n \leq \frac{L}{2}; \\ 0, & \text{caso contrário} \end{cases} \quad (3.1)$$

em que L é a distância do deslocamento.

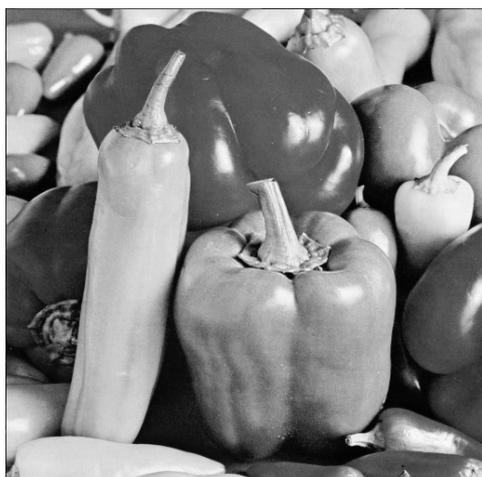
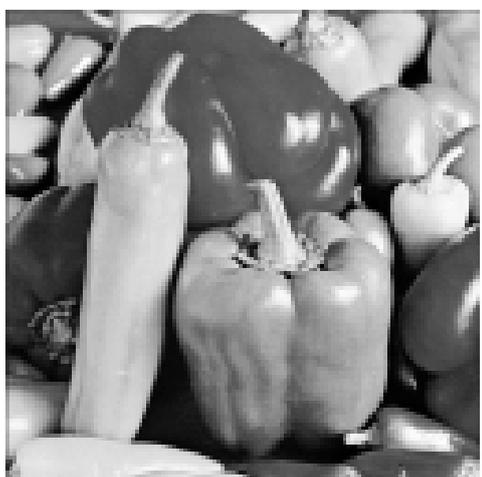
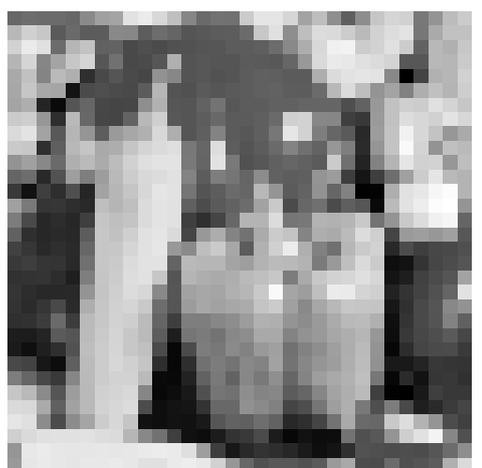
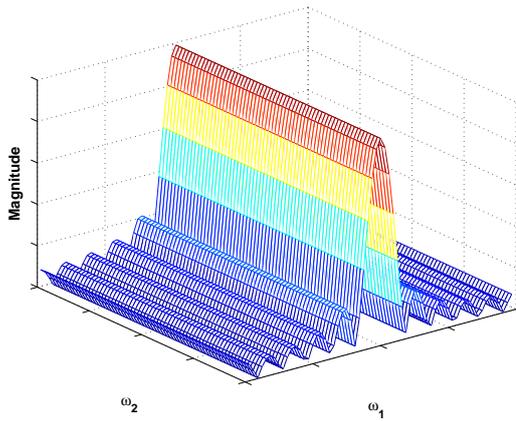
(a) 512×512 pixels(b) 256×256 pixels(c) 128×128 pixels(d) 64×64 pixels(e) 32×32 pixels(f) 16×16 pixels

Figura 3.2: Resolução espacial da imagem "Bellpeppers".

Pode-se verificar que neste caso a degradação só afeta apenas uma dimensão por vez, simulando realmente um movimento brusco durante a aquisição da imagem. Na Equação (3.1), n pode ser adotado tanto para representar o que aqui normalmente se define por n_1 ou n_2 , porém adotando apenas um destes parâmetros por vez. No exemplo da Figura 3.3(a), pode-se verificar esta situação quando se vê na resposta em frequência que a degradação só possui variação ao longo de ω_1 . A Figura 3.3(b) denota a degradação efetuada pelo *blur* 1-D.



(a) Resposta em frequência



(b) Imagem degradada

Figura 3.3: *Blur* de movimento 1-D; $L = 11$; dimensão 1×11

- **Blur de movimento 2-D** é o tipo de degradação originada pela aquisição de imagens de objetos em movimento ou pelo movimento do equipamento destinado à aquisição. Nesse caso, o movimento descrito possui dois graus de liberdade e dessa forma a direção do deslocamento do movimento pode ser variada, ou seja,

$$\mathbf{H}(n_1, n_2) = \begin{cases} \frac{1}{L^2}, & \text{se } -\frac{L}{2} \leq n_1, n_2 \leq \frac{L}{2}; \\ 0, & n_1, n_2 \leq -\frac{L}{2} \text{ ou } \frac{L}{2} \leq n_1, n_2 \end{cases} \quad (3.2)$$

em que L é a distância do deslocamento.

A resposta em frequência do filtro da Figura 3.4(a) mostra que a degradação é efetuada em ambos os eixos da imagem e a Figura 3.4(b) mostra a degradação na imagem.

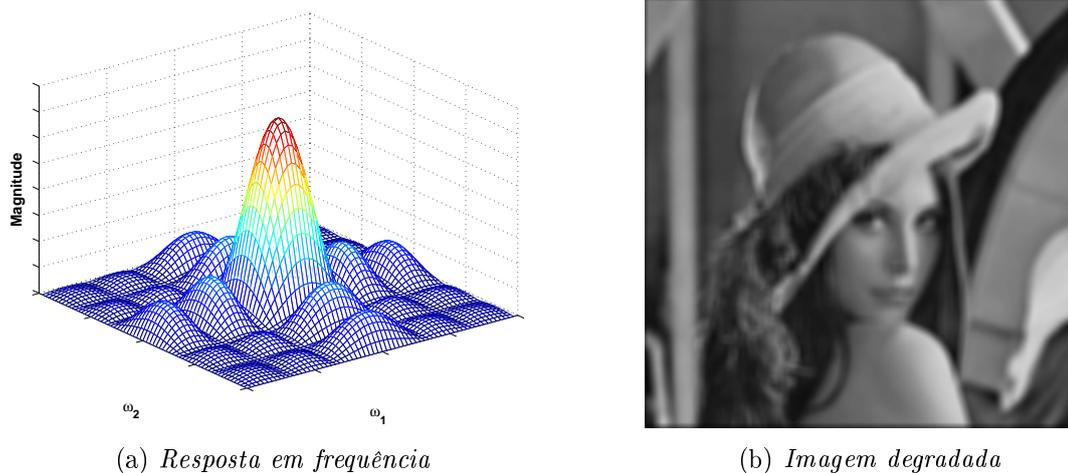


Figura 3.4: *Blur de movimento 2-D; $L = 6$; dimensão 6×6*

- **Blur Gaussiano** é o tipo de degradação baseada na distribuição Gaussiana.

$$\mathbf{H}(n) = \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{n^2}{2\sigma^2}} \quad (3.3)$$

em que σ^2 é a variância do canal e define a severidade da degradação.

Neste caso como no *blur* de movimento 1-D, a degradação afeta apenas uma dimensão da imagem por vez. Na Equação (3.3), n pode ser adotado tanto para representar o que aqui normalmente se define por n_1 ou n_2 , porém adotando apenas um destes parâmetros por vez. No exemplo da Figura 3.5(a), pode-se verificar esta situação quando se vê na resposta em frequência que a degradação só possui variação ao longo de ω_1 . A Figura 3.5(b) denota a degradação efetuada pelo *blur* Gaussiano 1-D.

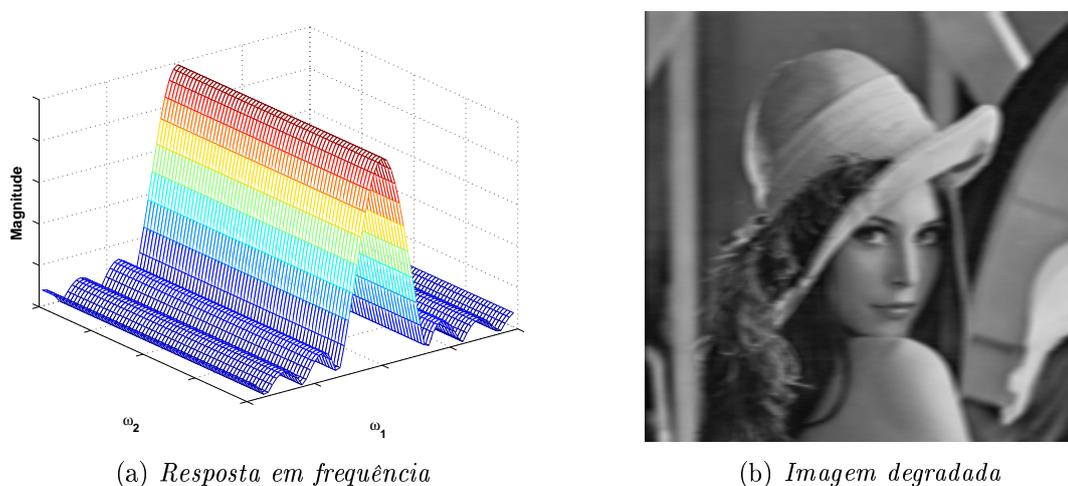


Figura 3.5: *Blur Gaussiano; $\sigma = 3$; dimensão 1×7*

- **Blur de turbulência atmosférica ou blur Gaussiano 2-D** é o tipo de degradação que atinge principalmente aplicações como sensoriamento remoto e aquisição de imagens astronômicas. Sua expressão é dada por

$$\mathbf{H}(n_1, n_2) = Ke^{-\frac{n_1^2 + n_2^2}{2\sigma^2}} \quad (3.4)$$

em K é uma constante de normalização e σ^2 é a variância do canal e define a severidade da degradação.

A resposta em frequência do filtro da Figura 3.6(a) mostra a degradação efetuada em ambos os eixos da imagem. A Figura 3.6(b) denota a degradação efetuada pelo *blur* de turbulência atmosférica.

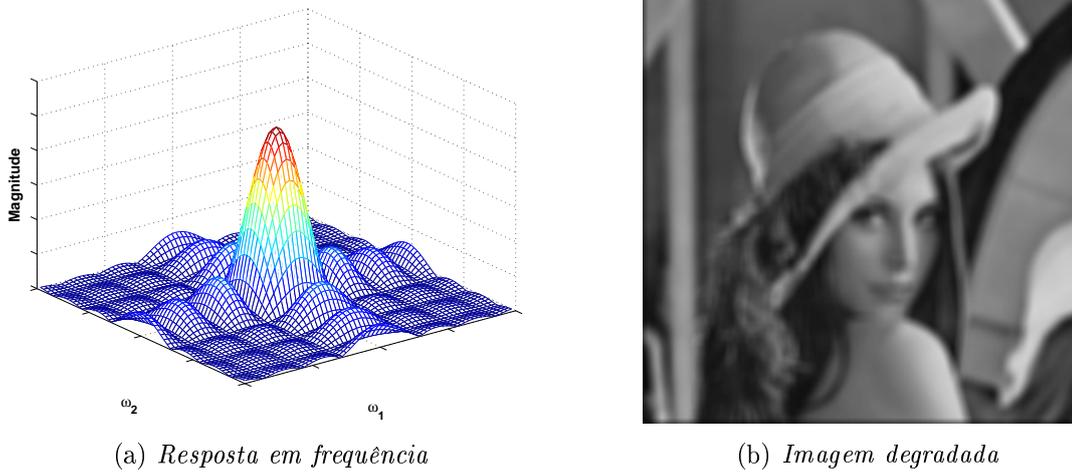


Figura 3.6: *Blur de turbulência atmosférica*; $\sigma^2 = 0.25$; $K=1$; dimensão 7×7

- **Blur de desfocagem uniforme** é uma forma de degradação que acontece principalmente em fotos e pode acontecer por diversas razões como: distância entre objeto e câmera, distância focal, efeitos de difração da luz, etc. A expressão que caracteriza essa degradação é dada por:

$$\mathbf{H}(n_1, n_2) = \begin{cases} \frac{1}{\pi R^2}, & \text{se } \sqrt{n_1^2 + n_2^2} \leq R; \\ 0, & \text{caso contrário.} \end{cases} \quad (3.5)$$

em R é o raio da desfocagem.

A resposta em frequência do filtro da Figura 3.7(a) mostra a degradação efetuada em ambos eixos da imagem de maneira circular. A Figura 3.6(b) denota a degradação efetuada pelo *blur* de desfocagem uniforme.

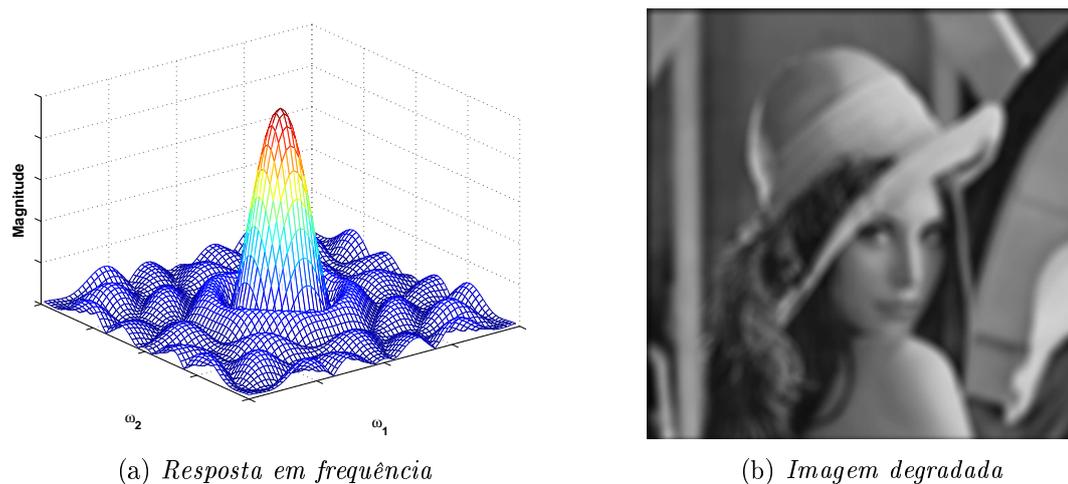


Figura 3.7: *Blur de desfocagem uniforme; $R = 4$*

Analisando o efeito destes tipos de degradação em uma imagem fica claro que dificilmente se pode notar as diferenças entre alguns tipos de *blur*. Há semelhanças entre os espectros e as imagens degradadas pelo *blur* de desfocagem uniforme, pelo *blur* Gaussiano 2-D e pelo *blur* de movimento 2-D da mesma forma que há semelhanças entre os espectros e imagens degradadas pelo *blur* de movimento 1-D e pelo *blur* Gaussiano. Em todos esses casos não se consegue diferenciar claramente um tipo de *blur* do outro em uma imagem degradada. Porém existem diferenças bem claras em seus parâmetros, o que indica que o uso de um filtro adaptativo que tenta equalizar diante de diferentes situações seria mais apropriado para tratar cada caso.

Quanto aos tipos de PSFs apresentados, cada um deles são largamente utilizados em aplicações específicas. Nesta dissertação se pretende focar nas PSFs do tipo *blur* de movimento 1-D e *blur* Gaussiano 2-D, pois englobam os dois tipos de degradação perceptíveis na imagem degradada e apresentam parâmetros bem diferentes que podem fornecer vários dados importantes por sua diversidade.

3.3 Ruído aditivo

As principais fontes de ruídos encontradas nas imagens ocorrem na sua fase de aquisição e transmissão. Na aquisição, o ruído é gerado por uma série de fatores referentes às condições ambientais e até mesmo às limitações dos sensores utilizados. Na transmissão, o ruído pode ser gerado pela interferência de outros sinais transmitidos.

Com exceção do ruído espacial periódico, se assume que o ruído é independente de coordenadas espaciais e é não-correlacionado com a imagem em si [Gonzalez e Woods, 2006]. Essas suposições não são gerais pois não se aplicam integralmente a certas aplicações, como por exemplo, em raio-x e microscopia em que a dificuldade de se lidar com o ruído é extrema. Com base no conceito de independência espacial, em que se caracteriza o ruído unicamente pelo seu comportamento estatístico, pode-se caracterizar o ruído como uma variável aleatória

descrita pela sua função densidade de probabilidade [Gonzalez e Woods, 2006]. As principais funções densidade de probabilidade encontradas no processamento de imagens são: ruído Gaussiano, ruído de Rayleigh, ruído de Erlang, ruído exponencial, ruído uniforme, ruído sal e pimenta e ruído periódico.

- **Ruído Gaussiano**

A função densidade de probabilidade é dada por

$$p(z) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(z-\mu)^2}{2\sigma^2}} \quad (3.6)$$

em que z é a variável aleatória que representa os níveis de cinza, μ é a média σ é o desvio padrão e σ^2 é a variância. Como z representa a intensidade de cada pixel, esta variável é sempre positiva [Gonzalez e Woods, 2006]. Pode-se notar seu efeito sobre uma imagem comum na Figura 3.8(a).

- **Ruído Rayleigh**

A função densidade de probabilidade é dada por

$$p(z) = \begin{cases} \frac{2}{b}(z-a)e^{-(z-a)/b} & \text{para } z \geq a; \\ 0 & \text{para } z < a. \end{cases} \quad (3.7)$$

em que a média e a variância são definidas respectivamente por

$$\mu = a + \sqrt{\frac{\pi b}{4}} \quad \text{e} \quad \sigma^2 = \frac{b(4-\pi)}{4}.$$

A Figura 3.8(b) mostra o efeito desse ruído sobre uma imagem comum.

- **Ruído Erlang**

A função densidade de probabilidade é dada por

$$p(z) = \begin{cases} \frac{a^b z^{b-1}}{(b-1)!} e^{-az} & \text{para } z \geq 0; \\ 0 & \text{para } z < 0. \end{cases} \quad (3.8)$$

em que os parâmetros a e b são definidos como $a > 0$ e b sendo um inteiro positivo. A média e variância são dadas respectivamente por

$$\mu = \frac{b}{a} \quad \text{e} \quad \sigma^2 = \frac{b}{a^2}.$$

A Figura 3.8(c) mostra o efeito desse ruído sobre uma imagem comum.

- **Ruído exponencial**

A função densidade de probabilidade é dada por

$$p(z) = \begin{cases} ae^{-az} & \text{para } z \geq 0; \\ 0 & \text{para } z < 0. \end{cases} \quad (3.9)$$

em que $a > 0$ e a média e a variância são dadas respectivamente por

$$\mu = \frac{1}{a} \quad \text{e} \quad \sigma^2 = \frac{1}{a^2}.$$

A Figura 3.8(d) mostra o efeito desse ruído sobre uma imagem comum.

- **Ruído Uniforme**

A função densidade de probabilidade é dada por

$$p(z) = \begin{cases} \frac{1}{b-a} & \text{se } a \leq z \leq b; \\ 0 & \text{caso contrário.} \end{cases} \quad (3.10)$$

em que a média e a variância são definidas respectivamente por

$$\mu = \frac{a+b}{2} \quad \text{e} \quad \sigma^2 = \frac{(b-a)^2}{12}.$$

A Figura 3.8(e) mostra o efeito desse ruído sobre uma imagem comum.

- **Ruído Sal e Pimenta**

A função densidade de probabilidade de impulso bipolar é dada por

$$p(z) = \begin{cases} P_a & \text{para } z = a; \\ P_b & \text{para } z = b; \\ 0 & \text{caso contrário.} \end{cases} \quad (3.11)$$

Resumidamente, caso $a > b$ um nível claro irá aparecer na imagem, caso contrário, um nível escuro irá aparecer. Caso P_a e P_b sejam positivos a imagem terá vários pontos claros e escuros que explica o nome desse tipo de ruído. O efeito em uma imagem real do ruído Sal e Pimenta pode ser visto na Figura 3.8(f).

Dentre os tipos de ruídos apresentados, o ruído Gaussiano é o mais considerado na literatura de processamento de imagens. Por esse motivo será utilizado nas simulações desta dissertação.



(a)



(b)



(c)



(d)



(e)



(f)

Figura 3.8: *Degradação por ruído. (a) Ruído Gaussiano, (b) Ruído Rayleigh, (c) Ruído Erlang, (d) Ruído Exponencial, (e) Ruído Uniforme e (f) Ruído Sal e Pimenta*

Capítulo 4

Algoritmos adaptativos para restauração de imagens

Neste capítulo, são detalhadas algumas técnicas específicas de processamento de imagens para o uso de algoritmos adaptativos, ou seja, definição de janelas, tratamento de borda e varredura. Além disso, dois algoritmos bidimensionais – o supervisionado (TDLMS) e o autodidata (TDCMA) – são apresentados da mesma forma que foram propostos na literatura. Em seguida, se propõe um operador para facilitar o uso de algoritmos adaptativos apresentados no Capítulo 2 para restauração de imagens.

4.1 Definição de Janelas

Para estimar um determinado pixel de uma imagem, deve-se selecionar uma janela \mathbf{U} que contenha pixels vizinhos daquele que se deseja estimar. Geralmente, a seleção da janela \mathbf{U} para a atualização dos coeficientes do filtro se dá como apresentado à esquerda da Figura 4.1. Neste caso, para se estimar o pixel marcado com o símbolo \bullet na figura, são utilizados seus vizinhos a direita e abaixo. Este tipo de seleção é largamente utilizada por vários autores devido à facilidade para programar e para geração de inicializações (veja por exemplo, [Hadhoud e Thomas, 1988] e suas referências).

Embora esse tipo de janela apresente bons resultados, uma outra possibilidade de seleção é a que está apresentada à direita da Figura 4.1. Neste caso, para se estimar o pixel \bullet , utilizam-se os pixels vizinhos a sua volta (veja por exemplo, [Vural e Sethares, 2002] e suas referências), o que pode produzir resultados melhores ainda que os obtidos com a seleção da janela descentralizada. Os dois arranjos necessitam de inicialização para alguns valores da janela quando alguns pixels na borda são selecionados. O arranjo centralizado possui mais problemas de inicialização pois esses parâmetros são necessários em todo o redor da imagem.

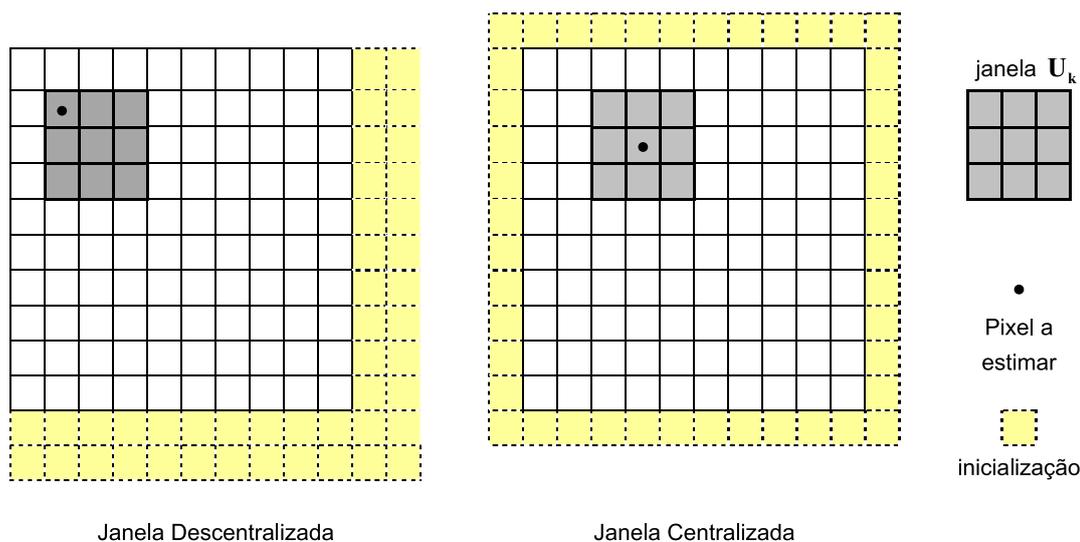


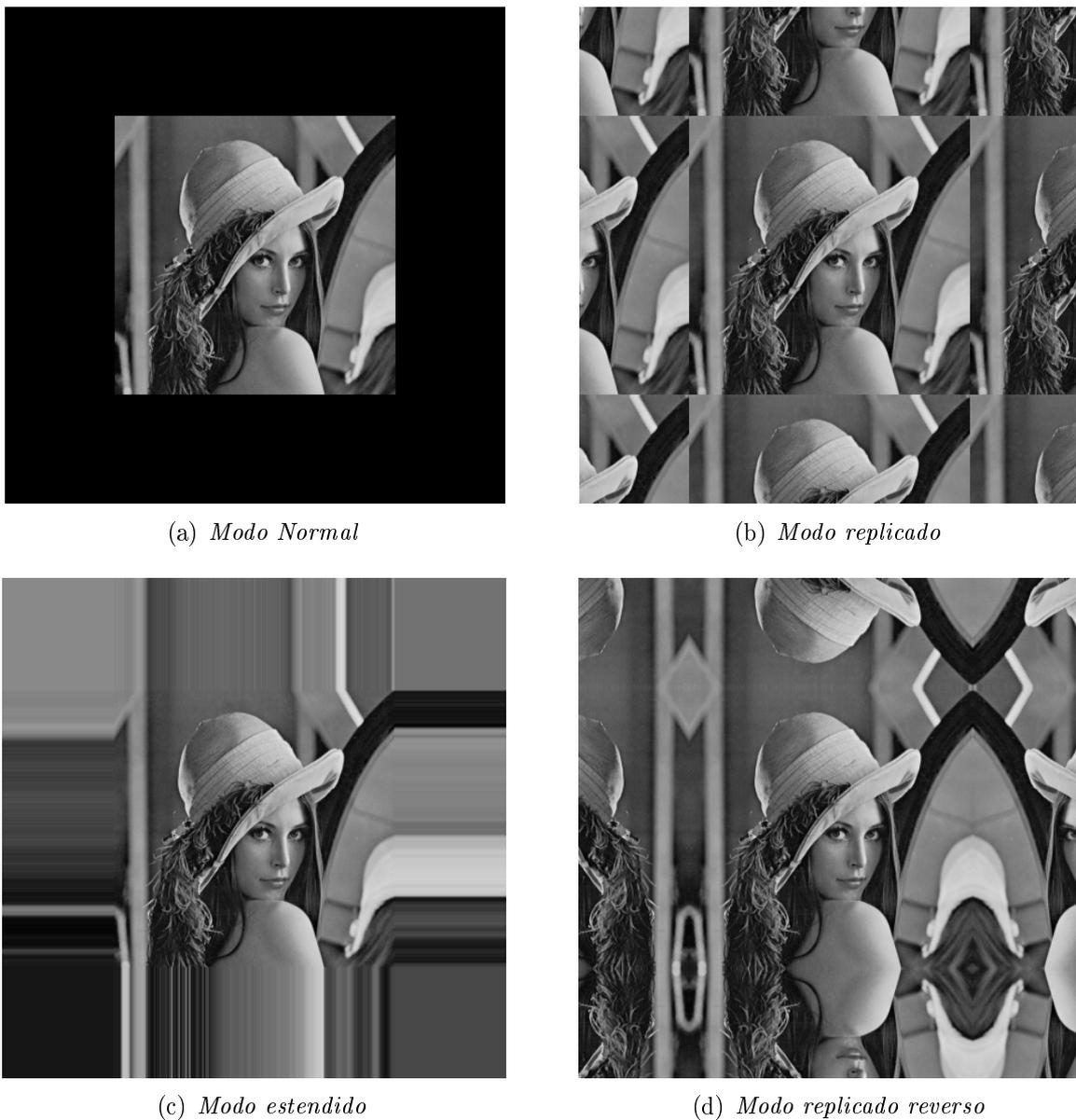
Figura 4.1: Modelos de seleção janela

4.2 Tratamento de borda

A interface de borda é encarada por muitos como condições de inicialização. Em qualquer método utilizado para seleção da janela \mathbf{U} faltarão pixels próximos do contorno da imagem para a formação adequada da janela. Há várias abordagens válidas de acordo com o tipo de processamento a ser realizado. Ao se utilizar filtros adaptativos para restauração de uma imagem, as soluções mostradas nas figuras 4.2(a) e 4.2(b) não são adequadas. Na Figura 4.2(a) considera-se o contorno da imagem como se fosse uma cor sólida, enquanto na Figura 4.2(b), a borda é constituída por repetições consecutivas da imagem. Em ambas as abordagens, ocorrem mudanças bruscas nas condições estatísticas das imagens, o que dificulta o trabalho dos filtros adaptativos. Interfaces de borda estendida como na Figura 4.2(c) ou replicada reversa como na Figura 4.2(d) fazem com que toda a área após a extremidade da imagem tenha condições semelhantes ao pixel da borda. Dessa forma, sua estimativa com filtros adaptativos pode ser mais efetiva fazendo com que o valor estimado tenha mais chances de estar próximo do valor original.

4.3 Varredura

Nos algoritmos adaptativos bidimensionais, a varredura da imagem é definida pela forma com que a janela de seleção se movimenta na imagem para se estimar cada pixel. Uma solução largamente utilizada é a mostrada na Figura 4.3(a) em que a varredura é feita da esquerda para direita e ao final da linha, a varredura continua no começo da linha subsequente [Hadhoud e Thomas, 1988]. Um problema deste tipo de varredura é que podem ocorrer mudanças muito bruscas nos coeficientes do filtro todas as vezes que uma nova linha é iniciada devido à não-estacionariedade de imagens reais.

Figura 4.2: *Modelos de seleção de borda*

Diante disso, Ohki e Hashiguchi [1991] propuseram a forma de varredura mostrada na Figura 4.3(b). Neste método, se implementa um filtro adaptativo, cuja estimação de cada pixel depende do pixel à sua esquerda e do pixel acima. Neste caso, diferentes passos de adaptação podem ser considerados para as direções horizontal e vertical do algoritmo. Isto acontece porque duas janelas fazem a varredura simultaneamente, uma faz a varredura horizontal e a outra vertical, e pesos são adotados para cada janela a fim de atualizar os coeficientes do filtro. Geralmente se adota pesos iguais para as duas janelas, considerando assim contribuições iguais das duas direções da imagem. Este esquema é aplicado a todos os pixels da imagem com exceção da primeira linha e coluna, que não possuem informações de pixels vizinhos acima ou a esquerda. Assim, os coeficientes do filtro devem possuir condições iniciais nestes pontos, o que cria uma grande lacuna de informações a serem inicializadas, deixando o algoritmo mais

suscetível a não convergir.

Um método de varredura que aparentemente abrange melhor as condições estatísticas das regiões da imagem e ao mesmo tempo diminui o número de condições iniciais impostas é uma varredura alternada como sugerida em [Pei et al., 1993, Albiol et al., 1997] que é muito parecida com a varredura convencional com a diferença que ao final de cada linha a varredura altera sua direção como se pode verificar na Figura 4.3(c) e Figura 4.3(d). As condições estatísticas de cada região são preservadas e os transientes vistos na varredura convencional são minimizados. Algo interessante neste tipo de varredura é que ela pode ser associada através de ciclos de atualização e então se pode efetuar uma varredura alternada horizontal seguida de uma varredura alternada vertical e repetir este processo até que se obtenha convergência.

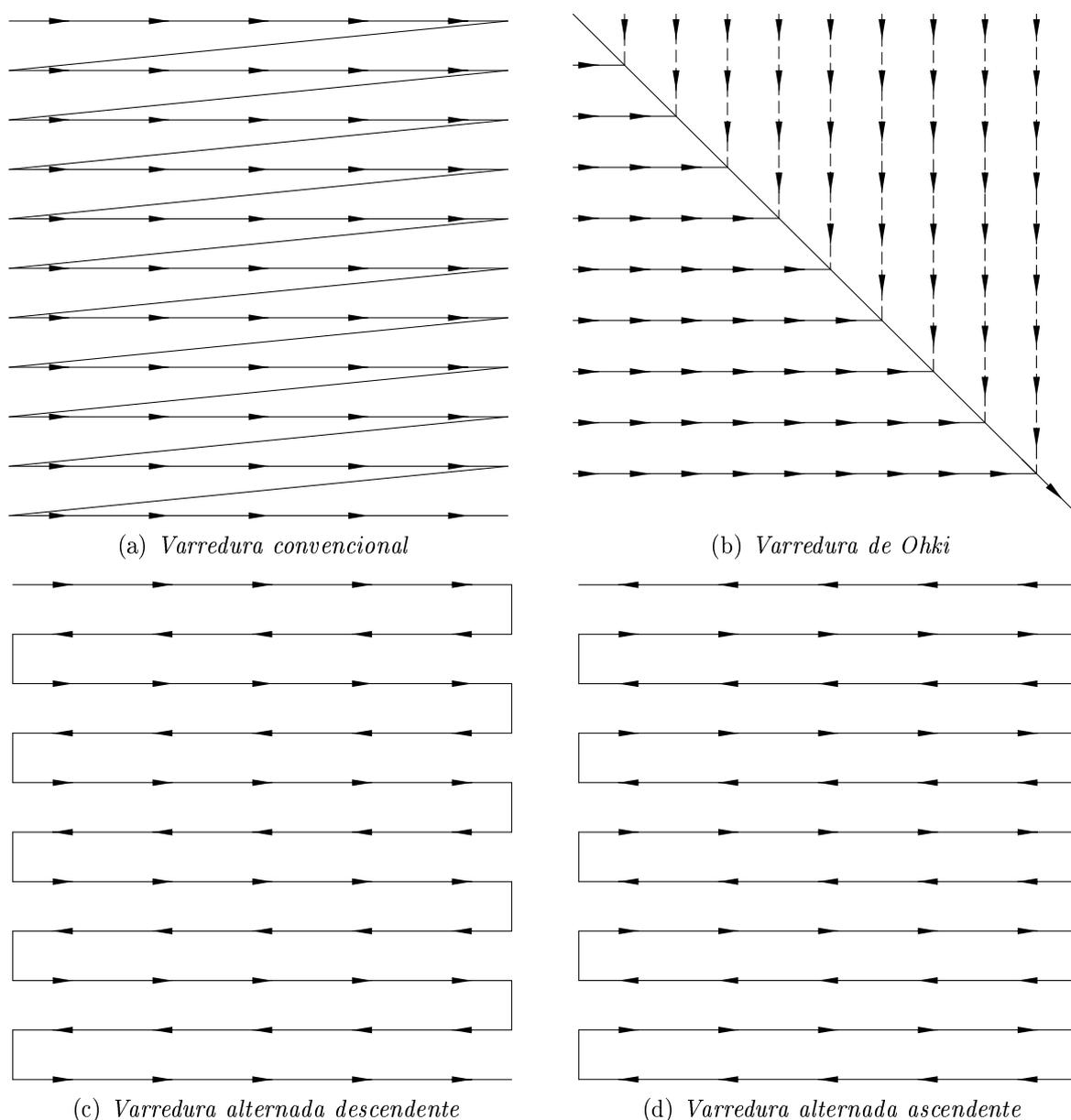


Figura 4.3: Modelos de varredura de imagem

4.4 Equalização supervisionada bidimensional - o algoritmo TDLMS

A equalização através do filtro de Wiener possui problemas porque assume condições estatísticas para a imagem que na prática não são reais, uma vez que trata a imagem como um sinal estacionário. Além disso, ela se torna complicada pela necessidade de se ter a informação da imagem original *a priori*. Utilizando o filtro de Wiener desta forma, ele se comporta como um filtro passa-baixas e não se torna uma solução efetiva para a remoção dos efeitos de *blur* do canal [Lim, 1990, Hadhoud e Thomas, 1988]. Como alternativa a estes problemas, Hadhoud e Thomas [1988] propuseram um algoritmo adaptativo que é uma extensão direta do algoritmo LMS. Este algoritmo, denominado TDLMS (*Two-Dimensional Least Mean Square*), é fácil de ser implementado e apresenta boas condições de *tracking*, se adaptando melhor às variações locais de estatísticas da imagem. O algoritmo se baseia em efetuar a reconstrução da imagem degradada \mathbf{G} , utilizando a imagem original \mathbf{F} como imagem desejada e obtendo a estimativa $\hat{\mathbf{F}}$ seguindo o modelo de equalização proposto na Figura 1.3. O erro de estimação da k -ésima iteração é definido por

$$\begin{aligned} e_k(n_1, n_2) &= \mathbf{F}(n_1, n_2) - \sum_{k_1=-\infty}^{\infty} \sum_{k_2=-\infty}^{\infty} \mathbf{W}_k(k_1, k_2) \mathbf{G}(n_1 - k_1, n_2 - k_2) \\ e_k(n_1, n_2) &= \mathbf{F}(n_1, n_2) - \hat{\mathbf{F}}(n_1, n_2). \end{aligned} \quad (4.1)$$

Fazendo as devidas extensões para o caso bidimensional, a atualização da matriz de coeficientes do filtro, segundo o método *steepest descent* é dada por

$$\mathbf{W}_k = \mathbf{W}_{k-1} - \frac{1}{2} \mu \nabla J_k, \quad (4.2)$$

em que \mathbf{W}_k é a matriz de coeficientes do filtro na k -ésima iteração, μ é o passo de adaptação do algoritmo e ∇J_k é o gradiente de $J_k \triangleq E[e_k^2(n_1, n_2)]$ em relação à matriz \mathbf{W}_{k-1} , ou seja,

$$\begin{aligned} \nabla J_k &= \left. \frac{\partial E[e_k^2(n_1, n_2)]}{\partial \mathbf{W}} \right|_{\mathbf{W}=\mathbf{W}_{k-1}} \\ &= \begin{bmatrix} \frac{\partial E[e_k^2(n_1, n_2)]}{\partial \mathbf{W}_{k-1}(0,0)} & \cdots & \frac{\partial E[e_k^2(n_1, n_2)]}{\partial \mathbf{W}_{k-1}(0, N-1)} \\ \vdots & \ddots & \vdots \\ \frac{\partial E[e_k^2(n_1, n_2)]}{\partial \mathbf{W}_{k-1}(N-1,0)} & \cdots & \frac{\partial E[e_k^2(n_1, n_2)]}{\partial \mathbf{W}_{k-1}(N-1, N-1)} \end{bmatrix}. \end{aligned} \quad (4.3)$$

Para definir um gradiente local e utilizá-lo na estimativa do pixel, é necessário definir uma janela \mathbf{U}_k da imagem degradada de mesma dimensão que a matriz de coeficientes do filtro. Existem várias maneiras de definir essa janela e sua movimentação em uma imagem,

como explicado na seções 4.1 e 4.3. Porém, no caso proposto em Hadhoud e Thomas [1988], utilizou-se uma janela descentralizada definida como

$$\mathbf{U}_k = \begin{bmatrix} \mathbf{G}(n_1, n_2) & \mathbf{G}(n_1, n_2 - 1) & \dots & \mathbf{G}(n_1, n_2 - N + 1) \\ \mathbf{G}(n_1 - 1, n_2) & \mathbf{G}(n_1 - 1, n_2 - 1) & \dots & \mathbf{G}(n_1 - 1, n_2 - N + 1) \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{G}(n_1 - N + 1, n_2) & \mathbf{G}(n_1 - N + 1, n_2 - 1) & \dots & \mathbf{G}(n_1 - N + 1, n_2 - N + 1) \end{bmatrix}. \quad (4.4)$$

em que a dimensão a matriz é $N \times N$.

Para a obtenção do TDLMS, é necessário ter uma estimativa instantânea do vetor gradiente, ou seja,

$$\widehat{\nabla} J_k = \left. \frac{\partial e_k^2(n_1, n_2)}{\partial \mathbf{W}} \right|_{\mathbf{w}=\mathbf{w}_{k-1}} = 2e_k(n_1, n_2) \frac{\partial e_k(n_1, n_2)}{\partial \mathbf{W}_{k-1}}. \quad (4.5)$$

Utilizando a definição de $e_k(n_1, n_2)$ da Equação (4.1), obtém-se

$$\begin{aligned} \left. \frac{\partial e_k(n_1, n_2)}{\partial \mathbf{W}} \right|_{\mathbf{w}=\mathbf{w}_{k-1}(n_1, n_2)} &= \\ &= \frac{\partial}{\partial \mathbf{W}_{k-1}(n_1, n_2)} \left(\mathbf{F}(n_1, n_2) - \sum_{k_1=-\infty}^{\infty} \sum_{k_2=-\infty}^{\infty} \mathbf{W}_{k-1}(k_1, k_2) \mathbf{G}(n_1 - k_1, n_2 - k_2) \right) \\ &= 0 - \mathbf{G}(n_1, n_2). \end{aligned} \quad (4.6)$$

Substituindo (4.6) em (4.5), chega-se a

$$\widehat{\nabla} J_k(n_1, n_2) = -2e_k(n_1, n_2) \mathbf{G}(n_1, n_2).$$

Utilizando a definição de \mathbf{U}_k da Equação (4.4), chega-se a conclusão que

$$\widehat{\nabla} J_k(n_1, n_2) = -2e_k(n_1, n_2) \mathbf{U}_k. \quad (4.7)$$

Utilizando essa estimativa instantânea em (4.2), a atualização dos coeficientes do algoritmo TDLMS é dada por

$$\mathbf{W}_k = \mathbf{W}_{k-1} + \mu e_k(n_1, n_2) \mathbf{U}_k. \quad (4.8)$$

Como no caso unidimensional, a estabilidade do algoritmo TDLMS proposto por Hadhoud e Thomas [1988] depende de uma escolha adequada do passo de adaptação μ . Essa escolha ainda traduz um compromisso entre velocidade de convergência e erro quadrático médio em regime. O algoritmo TDLMS está resumido no procedimento da Tabela 4.1, em que \otimes significa o somatório da multiplicação elemento a elemento de duas matrizes.

<p>Inicialização do algoritmo:</p> <p>Define-se a janela utilizada</p> <p>Define-se a condição de borda</p> <p>Define-se a varredura</p> <p>$\mathbf{W}_0 = \mathbf{0}_{M \times M}$</p>
<p>Para cada iteração, $k = 1, 2, \dots$, para cada posição, $n_1, n_2 = 1, 2, \dots$ calcule:</p> <p>$\hat{\mathbf{F}}(n_1, n_2) = \mathbf{W}_k \circledast \mathbf{U}_k$</p> <p>$e_k(n_1, n_2) = \mathbf{F}(n_1, n_2) - \hat{\mathbf{F}}(n_1, n_2)$</p> <p>$\mathbf{W}_k = \mathbf{W}_{k-1} - \mu e_k(n_1, n_2) \mathbf{U}_k$</p>

Tabela 4.1: Sumário do Algoritmo *Two Dimensional Least-Mean Square* (TDLMS).

4.5 Equalização autodata bidimensional - o TDCMA

Uma extensão do algoritmo CMA para o caso bidimensional foi apresentada em [Vural e Sethares, 2002]. Para se utilizar o CMA, é necessário fazer um mapeamento antes da degradação da imagem. Assim, por exemplo, uma imagem de dois bits/pixel deve ter seus quatro níveis mapeados em ± 1 e ± 3 . Dessa forma, uma imagem de dois bits/pixel pode ser interpretada como um sinal de comunicação com modulação do tipo 4-PAM. De uma forma geral, uma imagem com B bits/pixel deve ser mapeada no sinal $\pm 1, \pm 3, \dots, \pm (2^B - 1)$, que representa um sinal do tipo 2^B -PAM. A imagem mapeada é denotada por $\tilde{\mathbf{F}}$ e sua estimativa por $\hat{\mathbf{F}}$. Dessa forma, a extensão para duas dimensões da função custo do módulo constante instantânea é dada por

$$\hat{J}_k = \left(r - \hat{\mathbf{F}}_k^2(n_1, n_2) \right)^2, \quad (4.9)$$

em que r é a constante de dispersão da imagem real definida por

$$r = \frac{E \left[\tilde{\mathbf{F}}^4(n_1, n_2) \right]}{E \left[\tilde{\mathbf{F}}^2(n_1, n_2) \right]}. \quad (4.10)$$

A constante de dispersão r varia conforme o número de bits/pixel da imagem de acordo com a Tabela 4.2 assumindo-se que os símbolos são equiprováveis na imagem. Procedendo de forma análoga ao algoritmo TDLMS, a equação de atualização do TDCMA é dada por

$$\mathbf{W}_{k+1} = \mathbf{W}_k + \mu e_k(n_1, n_2) \mathbf{U}_k, \quad (4.11)$$

sendo $e(k)e_k(n_1, n_2) = \left(r - \hat{\mathbf{F}}_k^2(n_1, n_2) \right) \hat{\mathbf{F}}_k(n_1, n_2)$. O TDCMA está resumido no procedimento da Tabela 4.3. De forma análoga ao caso unidimensional, o TDCMA é comumente inicializado

nº de bits/pixel	Níveis de cinza	r
1	2	1.0
2	4	8.2
3	8	37.0
4	16	152.2
5	32	613.0
6	64	2456.2
7	128	9829.0
8	256	39320.0

Tabela 4.2: Constante de dispersão

com uma matriz impulso unitário.

<p>Inicialização do algoritmo:</p> <p>Define-se a janela utilizada</p> <p>Define-se a condição de borda</p> <p>Define-se a varredura</p> $r = \frac{E[\hat{\mathbf{F}}^4(n_1, n_2)]}{E[\hat{\mathbf{F}}^2(n_1, n_2)]}$ $\mathbf{W}_0 = \begin{pmatrix} 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & 1 & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & \cdots & 0 & 0 \end{pmatrix}$
<p>Para cada iteração, $k = 1, 2, \dots$, para cada posição, $n_1, n_2 = 1, 2, \dots$ calcule:</p> $\hat{\mathbf{F}}(n_1, n_2) = \mathbf{W}_k \otimes \mathbf{U}_k$ $e_k(n_1, n_2) = (\hat{\mathbf{F}}_k^2(n_1, n_2) - r)\hat{\mathbf{F}}_k$ $\mathbf{W}(n+1) = \mathbf{W}(n) - \mu e_k(n_1, n_2)\mathbf{U}_k$

Tabela 4.3: Sumário do *Two Dimensional Constant Modulus Algorithm* (TDCMA).

4.6 Imagens tratadas como sinais unidimensionais

Tanto o algoritmo TDLMS, como o TDCMA apresentados possuem características bem parecidas com seus pares unidimensionais com exceção ao tratamento devido às dimensões. Esta seção propõe o uso de um operador para rearranjar os elementos de uma matriz em um vetor coluna. Este operador de vetorização é denotado aqui como $\text{vec}[\cdot]$, sendo $\text{vec}[\mathbf{A}]$, por exemplo, um vetor coluna formado pelas colunas da matriz \mathbf{A} . O objetivo desse operador é vetorizar as matrizes de entrada \mathbf{U}_k e a matriz de coeficientes \mathbf{W}_k . A Figura 4.4 exemplifica a vetorização em um sistema bidimensional com uma matriz de coeficientes 3×3 .

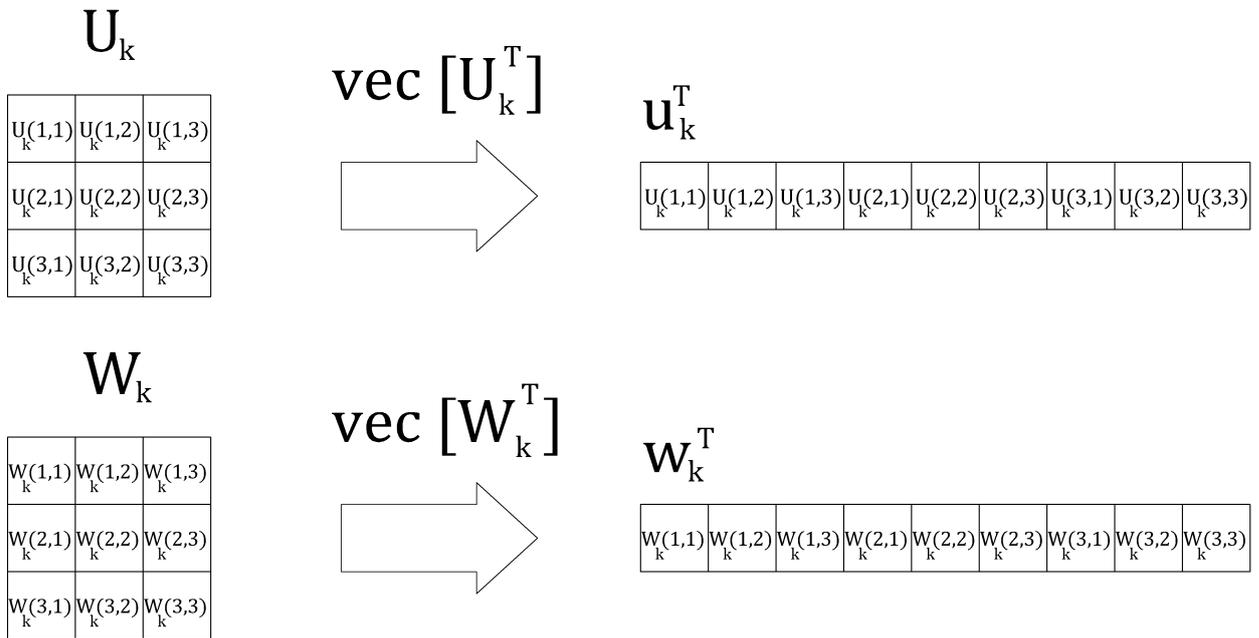


Figura 4.4: *Operador de vetorização*

Neste exemplo, não ocorre alteração no número de elementos, somente um rearranjo. Com isso, se pode obter a estimativa de um pixel através do produto interno $\hat{\mathbf{F}}(n_1, n_2) = \mathbf{w}_k^T \mathbf{u}_k$. Dessa forma, o uso deste operador pode facilitar a implementação dos algoritmos bidimensionais e estender de forma direta os algoritmos de equalização de canais do Capítulo 2 para reconstrução de imagens desde que antes da transformação se respeitem os conceitos discutidos na seções 4.1, 4.2 e 4.3.

Capítulo 5

Soluções alternativas para restauração de imagens

5.1 Introdução e Método de Avaliação

Este Capítulo tem o objetivo de apresentar extensões de algoritmos adaptativos de equalização de canais de comunicação para o caso bidimensional, com a finalidade de reconstrução de imagens. Através de simulações e comparações com algoritmos adaptativos bidimensionais encontrados na literatura, pretende-se justificar a busca de novas técnicas e aperfeiçoamentos para este campo do processamento de imagens.

As simulações foram efetuadas considerando dois dos principais tipos de PSF: o *blur* de movimento 1-D e o *blur* de turbulência atmosférica (*blur* Gaussiano 2-D). Existem autores na literatura que focam na degradação causada pelo *blur* de movimento 1-D [Kara e Vural, 2008, 2009]. Porém, o *blur* Gaussiano 2-D é raramente tratado por sua dificuldade de equalização. Na maior parte das simulações realizadas, se utilizou a imagem “Lenna”¹ com resolução de 256×256 representada na Figura 5.1. Considerou-se também a imagem “Mandril” a fim de mostrar o comportamento dos algoritmos em diferentes situações. Variou-se o número de bits/pixel das imagens de teste a fim de verificar o desempenho dos algoritmos conforme o número de símbolos da constelação aumenta.

Para a análise comparativa do erro, utilizou-se o $\%MSE$ [Kundur e Hatzinakos, 1996a,b] definido por

$$\%MSE = 100 \frac{\sum_{\forall n_1, n_2} [a\hat{\mathbf{F}}(n_1, n_2) - \mathbf{F}(n_1, n_2)]^2}{\sum_{\forall n_1, n_2} \mathbf{F}^2(n_1, n_2)}, \quad (5.1)$$

em que a é um fator de normalização dado por

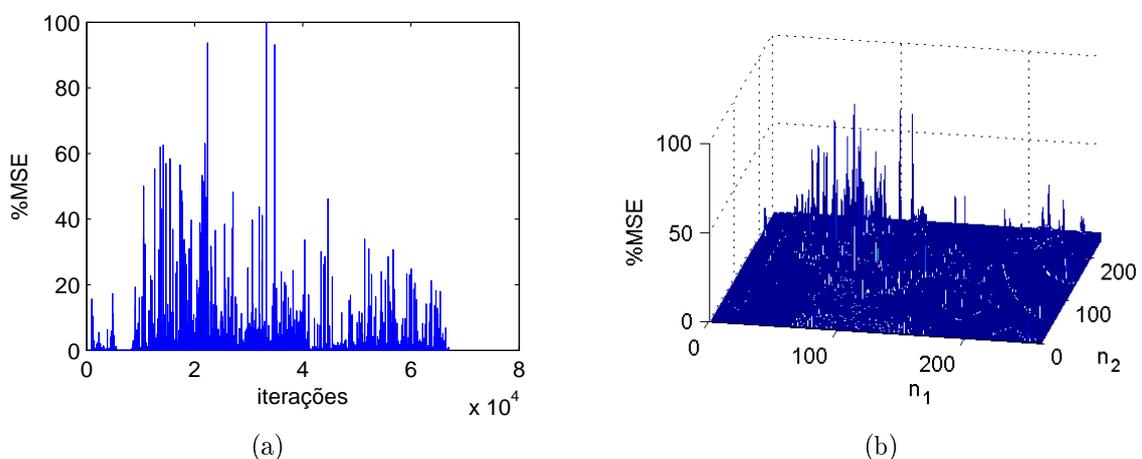
$$a = 100 \frac{\sum_{\forall n_1, n_2} \hat{\mathbf{F}}(n_1, n_2) \mathbf{F}(n_1, n_2)}{\sum_{\forall n_1, n_2} \hat{\mathbf{F}}^2(n_1, n_2)}. \quad (5.2)$$

¹a imagem “Lenna” é uma imagem largamente usada em processamento de imagens como imagem de teste e pode ser obtida, por exemplo, em www.lenna.org.

Figura 5.1: *Imagem de teste - Lenna*

Com esta medida se pode analisar o erro quadrático médio da estimativa da imagem em relação ao quadrado do pixel real. Esta medida se torna útil pois cria relações para comparações entre imagens com quantidade de níveis de cinza diferentes.

Em primeiro momento se pensou em mostrar o $\%MSE$ ao longo das iterações como visto na Figura 5.2(a) em que se tenta mostrar o progresso da equalização ao longo 65536 iterações de uma varredura completa da imagem de tamanho 256×256 . Isto foi logo descartado pois neste tipo de representação não é possível perceber como o erro está evoluindo pela imagem e quais os pontos críticos para a equalização. Devido a este problema, foi proposta uma visualização espacial como mostrado na Figura 5.2(b). Neste caso, é possível verificar os pontos de maior erro do algoritmo.

Figura 5.2: *Representação do %MSE. (a) Representação ao longo das iterações, (b) Representação espacial*

A Figura 5.3 mostra um exemplo de vista superior do erro da Figura 5.2(b). Nesta representação, a concentração de erros aumenta de acordo com o nível de cinza da imagem, ou seja, quanto mais escuro maior é o erro.



Figura 5.3: *Representação do %MSE - Vista de topo*

5.2 A relação com o canal de comunicação

Fazendo uma analogia com o caso unidimensional, os tipos de degradação geradas aqui pela PSF geram de forma equivalente uma interferência intersimbólica (ISI - *Intersymbol Interference*), ou seja, o pixel é degradado recebendo influência de seus vizinhos. No caso de imagens, essa interferência pode ocorrer nas duas dimensões ou apenas em uma delas de acordo com o tipo de *blur* que afeta a imagem.

Em equalização de canais de comunicação, sabe-se que um canal de fase mínima e/ou com nulos espectrais pronunciados é difícil de equalizar. Tendo isso em mente, foram feitas simulações para identificar a PSF ao longo das iterações. Primeiramente, uma janela centralizada de dimensão 5×5 , considerando borda replicada reversa, foi definida sobre a imagem original e esta janela foi vetorizada com o uso do operador `vec[-]` como explicado na Seção 4.6. A cada pixel da imagem original, se segue uma varredura alternada e um novo vetor é gerado. Esse vetor entra em um filtro adaptativo LMS que utiliza a imagem degradada como sinal desejado, e dessa forma, o filtro adaptativo efetua a identificação do canal, seguindo o modelo mostrado na Figura 5.4. Para a convergência do filtro adaptativo, foram considerados ciclos de repetição sobre a imagem, ou seja, se efetua a varredura sobre a imagem repetidas vezes até se obter a convergência do algoritmo.

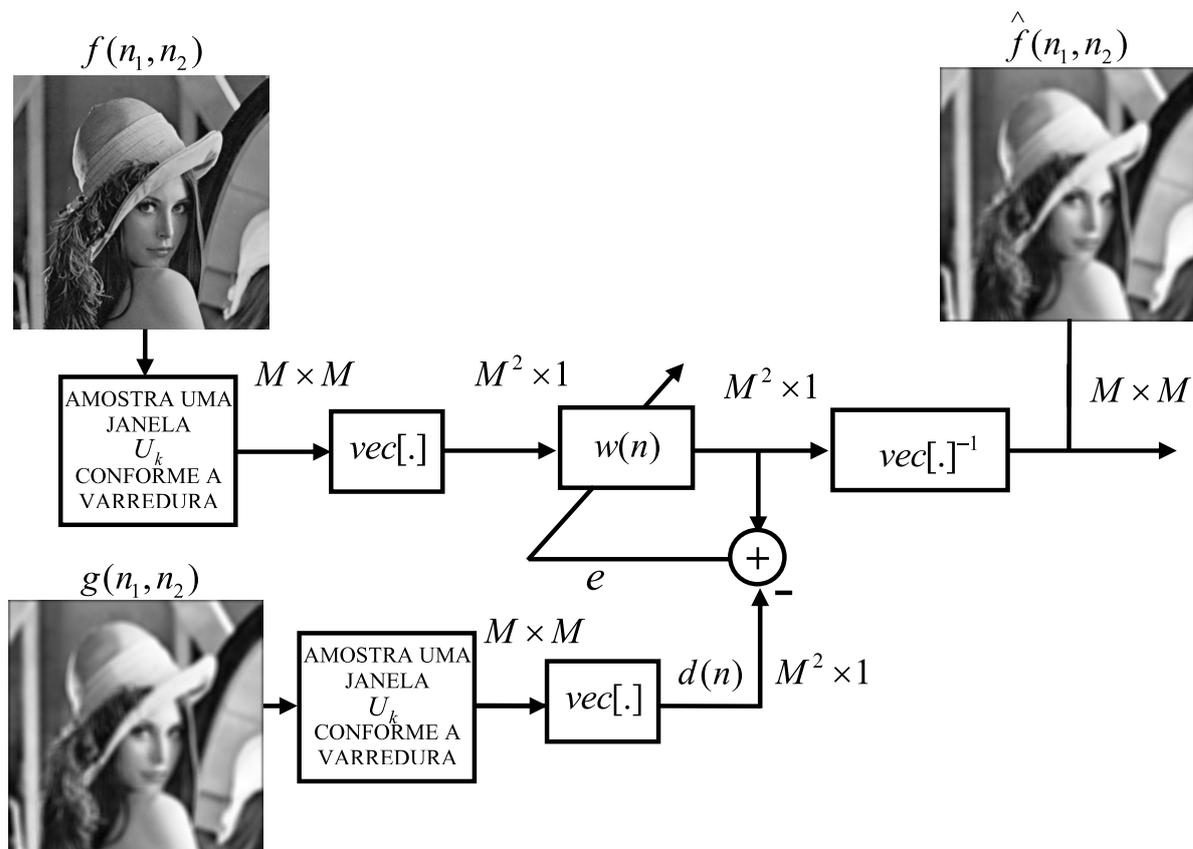


Figura 5.4: *Esquema de identificação de canal*

No caso de um canal que não gera degradação, usa-se um impulso unitário bidimensional, ou seja,

$$\mathbf{H} = \begin{pmatrix} 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & 1 & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & \cdots & 0 & 0 \end{pmatrix}. \quad (5.3)$$

A saída do sistema de reconhecimento gera uma resposta plana em 0 dB como se vê na Figura 5.5 e dessa forma se valida o funcionamento do reconhecimento perante a uma resposta conhecida.

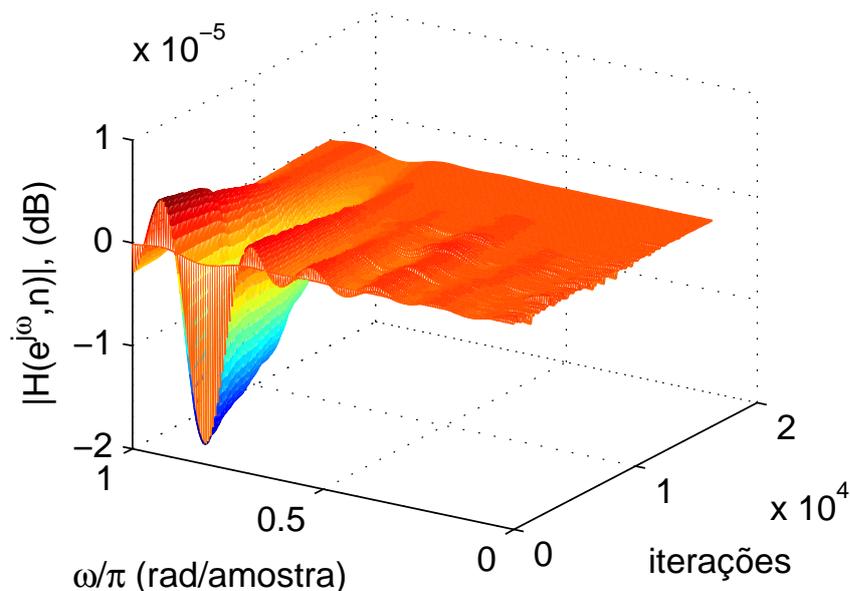


Figura 5.5: *Representação unidimensional do blur por iteração - Canal ideal - Algoritmo LMS* ($\mu = 0.001$, $M = 25$); 5 repetições.

Considerando agora os canais usados nas simulações, o *blur* de movimento 1-D é relativamente mais simples de se equalizar em relação ao *blur* Gaussiano 2-D. Essa simulação com o reconhecimento do canal pode justificar essa diferença.

Foi considerado um reconhecimento de sistema de um canal bem simples de se equalizar que é *blur* de movimento 1-D de dimensão 3×3 , utilizando o sistema da Figura 5.4. No caso, o que se obtém é mostrado na Figura 5.6 em que se tem um resposta com alguns nulos espectrais consideráveis, mas que aparenta ser constante ao longo das iterações. No diagrama de pólos e zeros da última iteração mostrada na Figura 5.7 se nota também a existência de nulos espectrais com alguns zeros sobre a circunferência unitária.

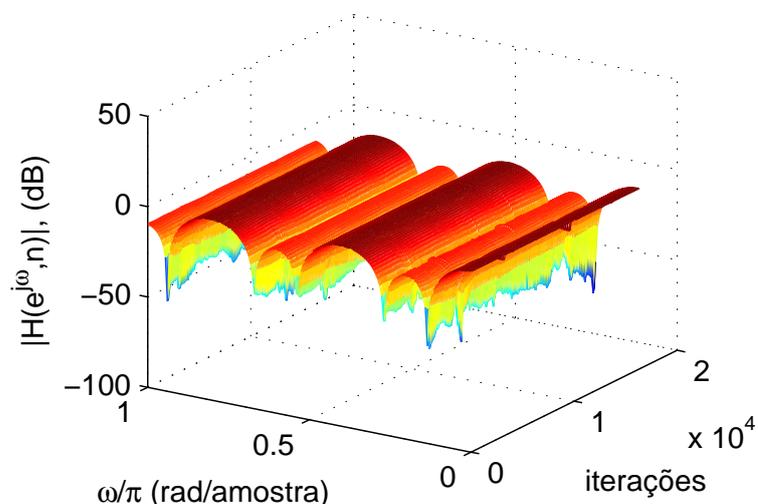


Figura 5.6: *Representação unidimensional do blur por iteração - Blur de Movimento 1-D 3×3* - Algoritmo LMS ($\mu = 0.001$, $M = 25$); 5 repetições.

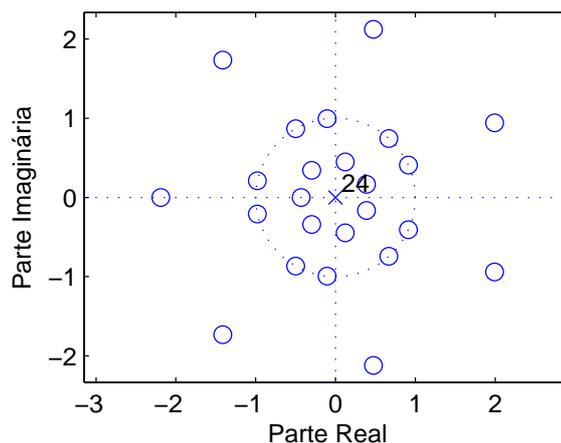


Figura 5.7: Diagramas de pólos e zeros - Blur de Movimento 1-D 3×3

Já no caso do *blur* Gaussiano 2-D, esse reconhecimento mostra um cenário completamente diferente. A Figura 5.8 mostra como é a distribuição do espectro ao longo das iterações e se nota nulos espectrais muito mais profundos em relação ao caso anterior além de que nunca ocorre uma convergência da resposta em frequência, ou seja, ela está em constante alteração a cada iteração do sistema. O diagrama de pólos e zeros mostrado na Figura 5.9 mostra quase todos os zeros situados sobre a circunferência unitária, o que torna esse canal mais difícil de equalizar.

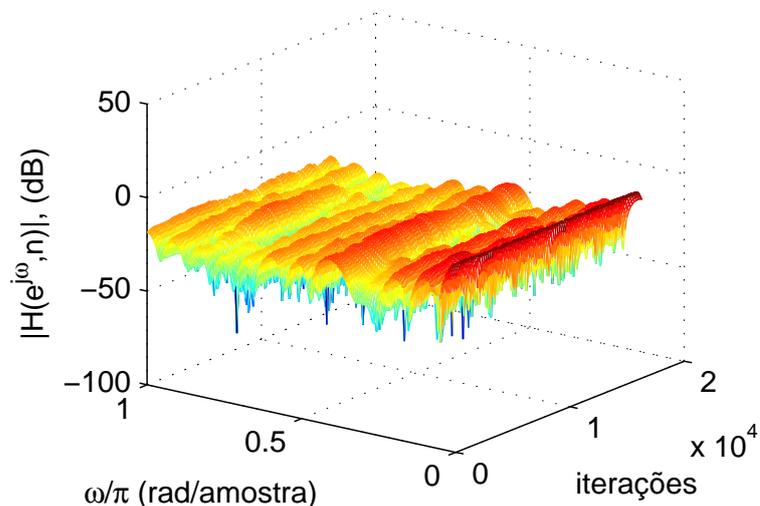


Figura 5.8: Representação unidimensional do blur por iteração - Blur Gaussiano 2-D 7×7 - Algoritmo LMS ($\mu = 0.001$, $M = 25$); 5 repetições.

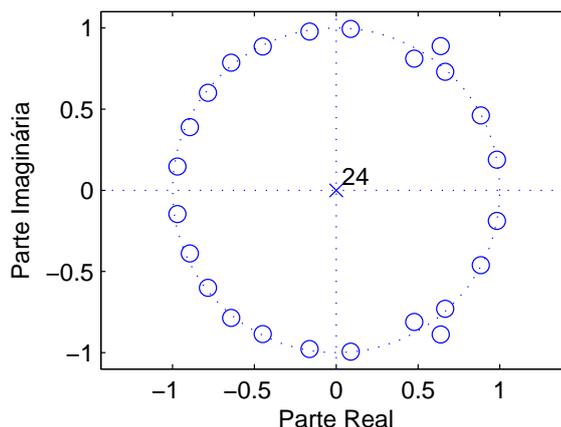


Figura 5.9: *Diagramas de pólos e zeros - Blur Gaussiano 2-D 7×7*

As características destes canais sugerem que o uso de um DFE seria de grande valia para a equalização pois sua característica não-linear pode ser útil para se sobressair ao problema dos nulos espectrais.

5.3 Alteração de varredura

Com base nos tópicos abordados, realizaram-se simulações a fim de comparar o desempenho do algoritmo TDLMS proposto por Hadhoud e Thomas [1988] e do TDCMA proposto por Vural e Sethares [2002]. Além disso, é proposta uma nova varredura a fim de facilitar a atualização dos coeficientes do filtro. Ao invés de usar a varredura convencional como mostrada na Figura 4.3(a), foi utilizada uma sequência de varreduras alternadas nos sentidos horizontal e vertical da imagem. Uma exemplificação da varredura proposta pode ser vista na Figura 5.10. A continuidade da varredura pela imagem é mantida e por essa razão a imagem é mostrada de forma invertida. O intuito é mostrar como é o comportamento da varredura proposta através da imagem na segunda vez que ela é executada. A imagem começa a ser varrida de forma vertical a partir do ponto que se terminou a primeira varredura, diminuindo assim a alteração brusca de \mathbf{U}_k e por sua vez de \mathbf{W}_k . Após o término da segunda varredura, um novo ciclo com uma varredura horizontal se inicia até que se tenha uma convergência na imagem. A atualização do filtro é feita realizando a varredura completa da imagem múltiplas vezes. Isso é feito porque a quantidade dos pixels da imagem degradada pode não ser suficiente para garantir a convergência do filtro adaptativo. Por isso, uma vez que a imagem toda foi “varrida”, começa-se novamente a varredura e esse processo é repetido até o algoritmo convergir. Em todas as simulações dessa seção, foram considerados 150 ciclos de repetição da imagem. Se o processo fosse realizado uma única vez nos algoritmos TDLMS e TDCMA, que possuem a varredura convencional, por exemplo, as primeiras linhas da imagem mostrariam uma degradação maior quando comparada ao restante, já que no início o algoritmo ainda estaria no transitório com forte influência da inicialização. Neste caso, como não se tem a

preocupação de se fazer uma aplicação em tempo real, a varredura repetida da imagem só adiciona melhorias à reconstrução.

Considerando os algoritmos TDLMS e TDCMA com a varredura convencional e com a varredura proposta, a convergência dos algoritmos para o mesmo $\%MSE$ fica clara, porém o algoritmo com a adição do conjunto de varreduras alternadas apresentou um desempenho melhor, possibilitando inclusive a utilização de passos de adaptação maiores como mostrado nas simulações seguintes.

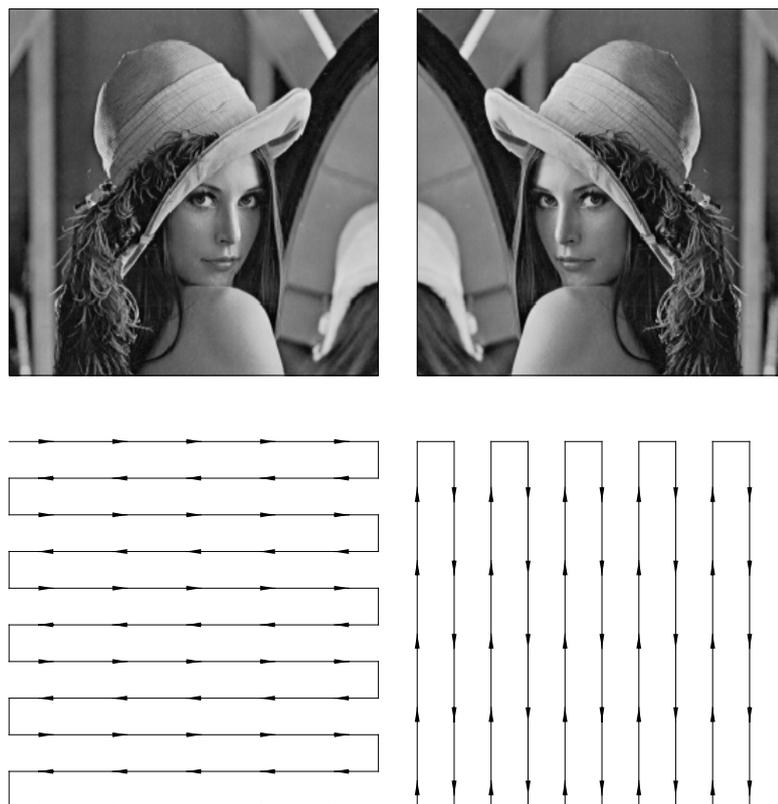


Figura 5.10: *Varredura Proposta*

No caso do TDLMS, a Figura 5.11 representa a progressão do $\%MSE$ conforme se varia o passo do algoritmo em uma imagem de 8 bits/pixel afetada por *blur* Gaussiano 2-D de dimensão 7×7 . Cabe observar que em todas as simulações dessa seção foram utilizadas uma janela centralizada e a borda replicada reversa. Considerando o $\%MSE$ geral da imagem, mostrado na Figura 5.11, observa-se algo curioso: o algoritmo com a varredura proposta apresenta um desempenho bem similar ao do algoritmo convencional para passos $\mu \in [10^{-25}, 4 \times 10^{-8}]$, mas para $\mu \in [2 \times 10^{-8}, 2 \times 10^{-6}]$ (antes da divergência), o desempenho do algoritmo com a nova varredura se torna superior. Porém, analisando o erro espacial de cada pixel para $\mu = 1.4 \times 10^{-6}$ (próximo à divergência) e para $\mu = 1.4 \times 10^{-18}$ representado pelas figuras 5.12(a) a 5.12(d), se pode fazer algumas observações. Quando se analisa os erros para cada pixel com $\mu = 1.4 \times 10^{-6}$, o algoritmo com a varredura proposta demonstra atingir valores menores de $\%MSE$ em muitas áreas da imagem como fica claro na Figura 5.12(b), enquanto

que o TDCMA com varredura convencional apresenta estimativas piores que as obtidas com a nova varredura como mostrado na Figura 5.12(a), porém nos dois casos existem regiões com grandes erros locais o que não é desejado para a reconstrução. Uma tentativa válida neste ponto seria desenvolver uma solução com passo variável em conjunto com a varredura para que se aproveite os efeitos de uma equalização com $\%MSE$ geral pequeno sem que haja regiões de extrema divergência na imagem. De maneira geral, a varredura proposta para o algoritmo apresenta resultados iguais ou melhores aos da varredura convencional.

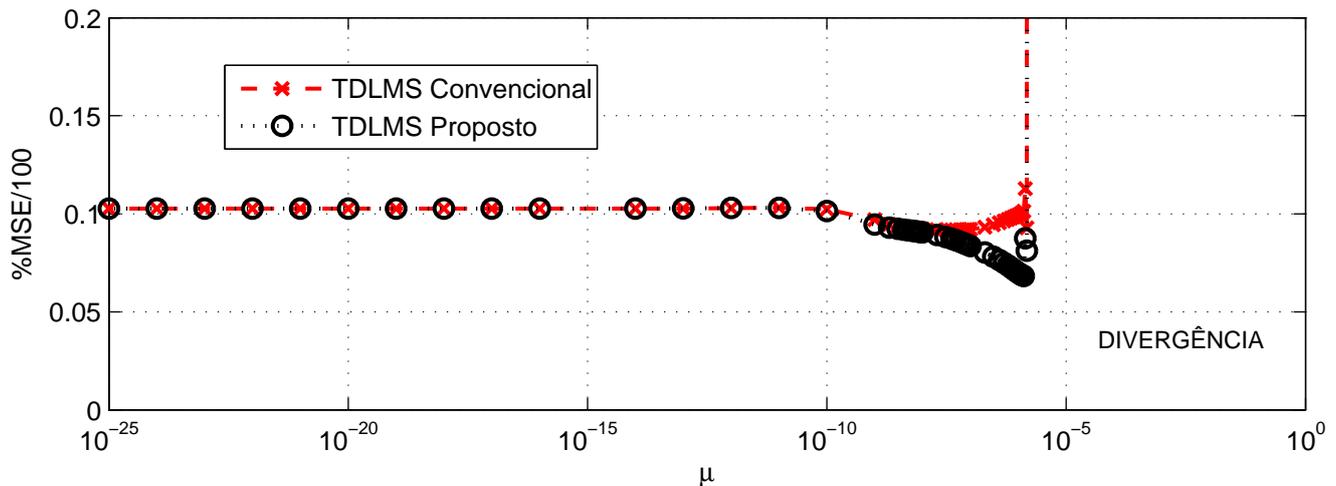


Figura 5.11: Comparação de algoritmos - TDLMS - Blur Gaussiano 2-D 7×7 - Níveis de cinza: 8 bits/pixel - Imagem 256×256 pixels

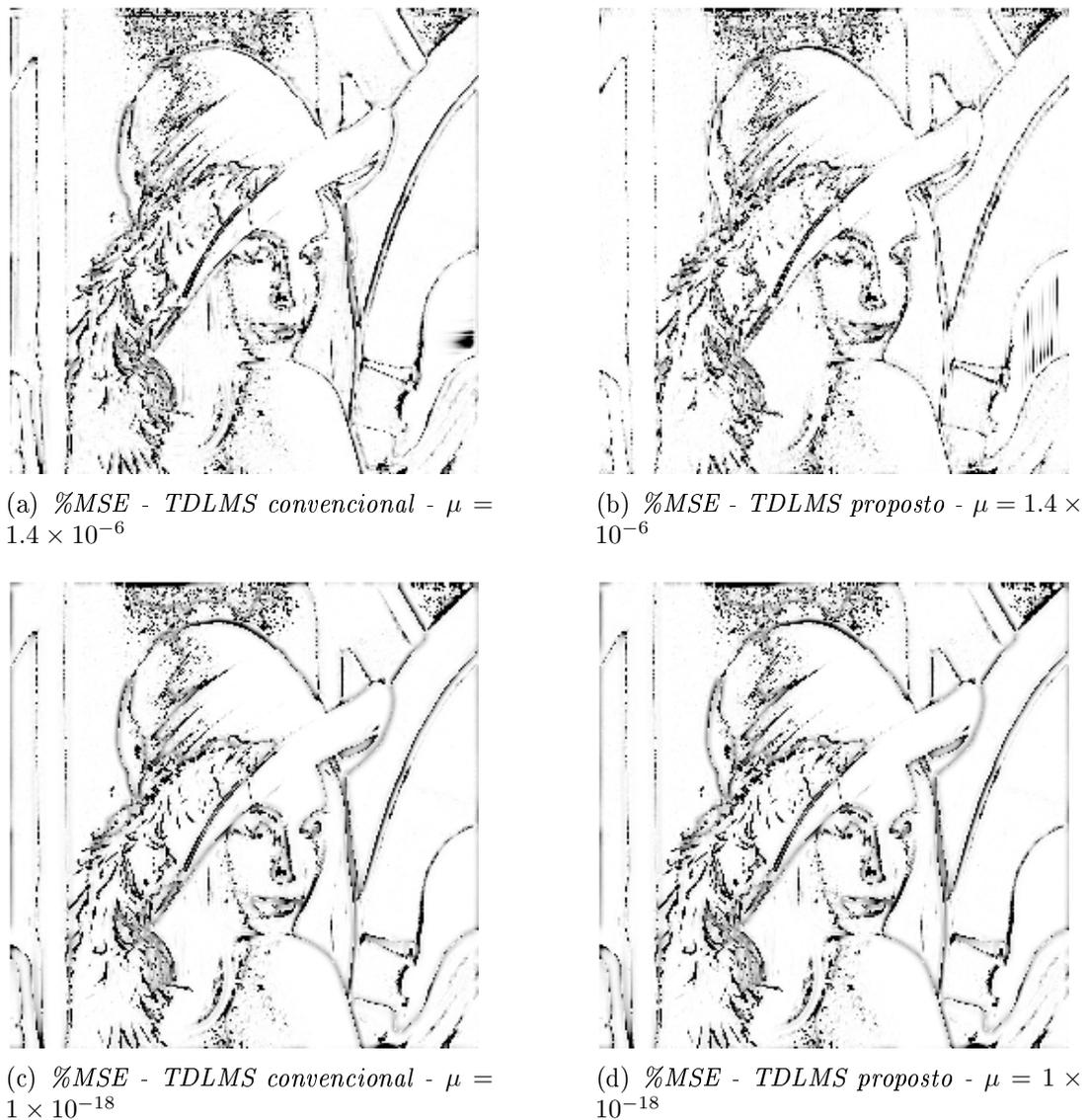


Figura 5.12: Simulações de $\%MSE$ no TDLMS - Blur Gaussiano 2-D 7×7 - Níveis de cinza: 8 bits/pixel - Imagem 256×256 pixels - sem ruído

Concentrando as simulações para o caso autodidata, a alteração de varredura também ocasionou vantagens para o TDCMA gerando também robustez à variação do passo de adaptação. Primeiramente, analisando o caso para o módulo constante, a Figura 5.13 mostra os resultados obtidos na reconstrução da imagem quantizada em 1 bit degradada por *blur* Gaussiano 2-D 3×3 . A reconstrução apresentada por ambos algoritmos é excelente recuperando quase toda a inteligibilidade perdida na degradação. No entanto, alguns pequenos detalhes da imagem original foram melhor recuperados com o TDCMA considerando a varredura proposta ($\%MSE = 1.50$) em comparação ao TDCMA convencional ($\%MSE = 5.32$).

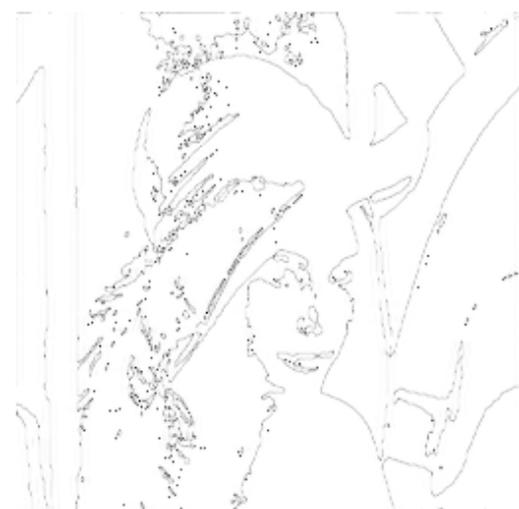
(a) *Imagem original*(b) *Imagem degradada*(c) *Imagem restaurada com o TDCMA convencional - $\mu = 10^{-4}$ - %MSE = 5.32*(d) *Imagem restaurada com o TDCMA proposto - $\mu = 10^{-4}$ - %MSE = 1.50*(e) *Erro com o TDCMA convencional - $\mu = 10^{-4}$* (f) *Erro com o TDCMA proposto - $\mu = 10^{-4}$*

Figura 5.13: *Efeitos da equalização - Blur Gaussiano 3×3 - Níveis de cinza: 1 bit/pixel - Imagem 256×256 pixels*

As figuras 5.14 e 5.15 apresentam os resultados obtidos com o algoritmo autodidata com uma imagem Lenna quantizada com 3 bits/pixel, sofrendo a degradação de *blur* Gaussiano 2-D de dimensão 5×5 . Mesmo com o passo de adaptação bem próximo a fazer o algoritmo divergir, os valores de $\%MSE$ para o TDCMA proposto são próximos ao $\%MSE$ mínimo que o filtro atinge enquanto que o erro geral do algoritmo convencional é bem alto.

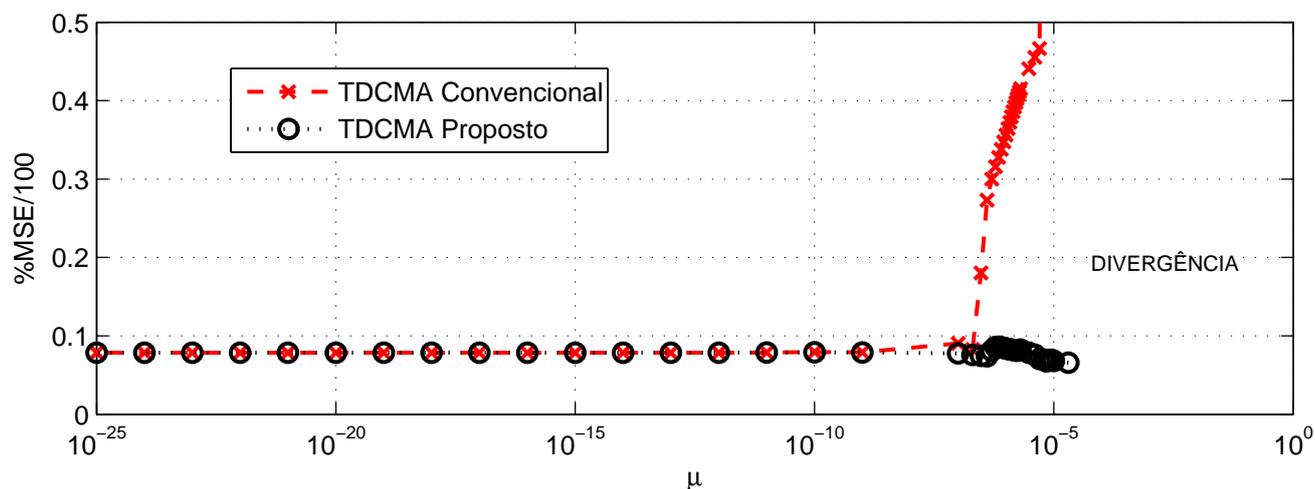


Figura 5.14: Comparação de algoritmos - TDCMA - Blur Gaussiano 2-D 5×5 - Níveis de cinza: 3 bits - Imagem 256×256 pixels - sem ruído

Na Figura 5.15 se pode ver o diagrama espacial do $\%MSE$ para alguns valores de μ , mostrando claramente a vantagem da varredura proposta em relação à variação do passo. Uma característica que cabe destacar é que tanto para o caso supervisionado quanto para o caso autodidata, a adaptação em mudanças bruscas da imagem geram grandes erros e nestes pontos a imagem dificilmente é reconstruída com sucesso, o que será motivo de estudos posteriores com finalidade de se tentar diminuir estes erros de estimativa na transição.

No caso da Figura 5.16, se tem uma demonstração visual da recuperação do TDCMA para um caso com *blur* de movimento 1-D de dimensão 3×3 e a imagem quantizada com 2 bits/pixel. Neste caso, foram apresentadas as imagens recuperadas mediante o melhor passo para cada algoritmo. Com base nisso, a recuperação com o TDCMA convencional foi feita com $\mu = 5 \times 10^{-5}$ enquanto que a recuperação com o TDCMA proposto foi feita com $\mu = 1 \times 10^{-4}$. O TDCMA proposto apresentou uma melhor equalização. Quando o algoritmo convencional utiliza o passo $\mu = 1 \times 10^{-4}$, a imagem recuperada ainda apresenta uma grande distorção. Neste ponto, cabe mencionar que o aumento de níveis de cinza na imagem aumenta o número de símbolos da constelação, o que dificulta cada vez mais a reconstrução correta da imagem. A reconstrução através deste algoritmo de uma imagem com mais de 4 bits/pixel não gera resultados satisfatórios e por essa razão se considera a seguir o algoritmo multimódulo regional proposto em [Mendes Filho et al., 2009] como uma solução alternativa para sinais de módulo não-constante.



(a) %MSE - TDCMA convencional - $\mu = 7 \times 10^{-6}$



(b) %MSE - TDCMA proposto - $\mu = 7 \times 10^{-6}$



(c) %MSE - TDCMA convencional - $\mu = 1 \times 10^{-6}$



(d) %MSE - TDCMA proposto - $\mu = 1 \times 10^{-6}$



(e) %MSE - TDCMA convencional - $\mu = 1 \times 10^{-15}$



(f) %MSE - TDCMA proposto - $\mu = 1 \times 10^{-15}$

Figura 5.15: Simulações de %MSE no TDCMA - Blur Gaussiano 2-D 5×5 - Níveis de cinza: 3 bits/pixel - Imagem 256×256 pixels - sem ruído



Figura 5.16: *Efeitos da recuperação no TDCMA - Blur de movimento 1-D 3×3 - Níveis de cinza: 2 bits/pixel - Imagem 256×256 pixels - sem ruído*

5.4 O algoritmo multimódulo regional bidimensional - o TDRMMA

Mesmo com uma melhoria na imagem através da alteração da varredura, o CMA é ineficaz em efetuar a equalização da imagem a medida que se aumenta o número de símbolos da constelação. Quanto maior o número de símbolos que filtro tem de lidar, menor deve ser o passo de adaptação e menor é o efeito do algoritmo sobre a imagem degradada. De maneira prática, o uso do CMA para imagens quantizadas com mais de 4 bits/pixel não apresenta qualquer melhoria em relação a imagem degradada [Vural e Sethares, 2002]. Diante deste problema, se propõe uma extensão do RMMA para o caso bidimensional, o TDRMMA (*Two-Dimensional*

Region-based Multidulus Algorithm). O TDRMMA pode ser implementado basicamente efetuando uma vetorização conforme mostrado na Seção 4.6 e considerando as operações da Tabela 2.6. Este algoritmo permite a reconstrução de imagens de até 8 bits/pixel, pois trata sinais de módulo não constante como de módulo constante. Efeitos do algoritmo em relação à degradação do canal sem ruído podem ser vistos da Figura 5.18 até 5.20. As imagens com 4 bits/pixel (Fig. 5.18) e 6 bits/pixel (Fig. 5.19) foram degradadas com um *blur* Gaussiano 2-D 3×3 e a imagem com 8 bits/pixel (Fig. 5.20) foi degradada com um *blur* de movimento 1-D de dimensão 5×5 . Novamente, utilizou-se uma janela centralizada, a borda replicada reversa e a varredura proposta. Em todas as simulações, foram considerados 150 ciclos de repetição da imagem. A melhoria é nítida e o realce dos detalhes demonstra que as translações do sinal de saída realizadas pelo TDRRMA lidam de maneira mais efetiva com a constelações com grande número de símbolos do que o TDCMA. Nessas figuras, tanto o erro espacial como o $\%MSE$ geral do TDRMMA sempre se apresentaram menores que os do TDCMA.

Na Figura 5.21, são mostrados resultados de simulação considerando a imagem “Mandrill” com 8 bits/pixel. Além disso, considerou-se um *blur* Gaussiano 2-D 5×5 e ruído Gaussiano de tal forma que a relação sinal ruído de imagens (BSNR-*Blurred Signal-to-Noise Ratio*) foi de 18 dB. Pode-se observar que mesmo na presença de ruído, o algoritmo também apresenta melhores resultados que os do TDCMA.

Em todos os testes realizados, os resultados foram similares, ou seja, o TDRMMA apresenta vantagens na equalização em relação ao TDCMA de [Vural e Sethares, 2002]. Porém, diante de canais mais complexos a reconstrução também não é satisfatória e a imagem recuperada pode ficar próxima à degradada. Para canais mais complexos, uma solução que usa um equalizador de decisão realimentada pode obter resultados satisfatórios.

A Figura 5.4 mostra a estrutura em blocos o TDRMMA. Considera-se que a image original é mapeada e então degradada e ruído é adicionado. Amostra-se a imagem com janelas $M \times M$ e considera-se a vetorização através do operador $vec[\cdot]$. Esses vetores são tratados diretamente com o uso do TDRMMA obtendo-se uma estimativa do pixel. O algoritmo detalhado passo a passo está descrito na Tabela 5.1.

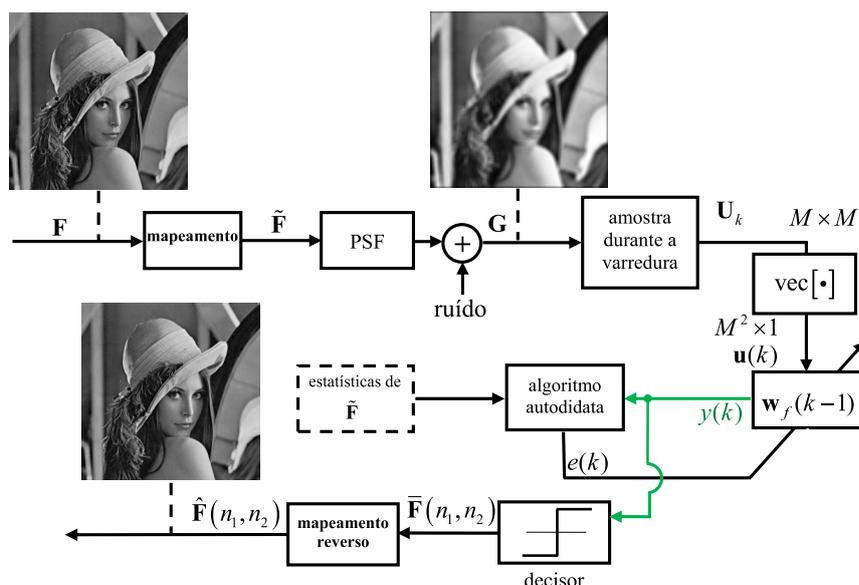


Figura 5.17: Arranjos em blocos do TDRMMA

Inicialização do algoritmo:

$$\mathbf{w}(0) = [0 \ \dots \ 0 \ 1 \ 0 \ \dots \ 0]^T, \mathbf{u}(0) = \mathbf{0}$$

Para cada iteração $k = 1, 2, \dots$ e para cada posição (n_1, n_2) durante a varredura, calcule:

Amostre \mathbf{U}_k ; $\mathbf{u}(k) = \text{vec}[\mathbf{U}_k]$

$$y(k) = \mathbf{w}^T(k-1)\mathbf{u}(k)$$

Identifique a região A_m e calcule:

$$\bar{y}(k) = y(k) - c_m$$

$$x(k) = 1.5 - 0.5\bar{y}^2(k)$$

se $x(k) \geq 0$

$$d(k) = x(k)\bar{y}(k)$$

caso contrário

$$d(k) = 0$$

fim

$$e(k) = |c_m| [d(k) - y(k)]$$

$$\mathbf{w}(k) = \mathbf{w}(k-1) + \frac{\tilde{\mu}}{\delta + \|\mathbf{u}(k)\|^2} e(k)\mathbf{u}(k)$$

$y(k)$ é remapeado para a imagem

fim

Tabela 5.1: Sumário do TDRMMA.

(a) *Imagem original*(b) *Imagem degradada*(c) *Imagem restaurada com o TDCMA - $\mu = 5 \times 10^{-9}$ - $\%MSE = 8.15$* (d) *Imagem restaurada com o TDRMMA - $\mu = 10^{-3}$ - $\%MSE = 2.73$* (e) *Erro com o TDCMA - $\mu = 5 \times 10^{-9}$* (f) *Erro com o TDRMMA - $\mu = 10^{-3}$*

Figura 5.18: Efeitos da equalização - Blur Gaussiano 2-D 3×3 - Níveis de cinza: 4 bits/pixels - Imagem 256×256 pixels - sem ruído

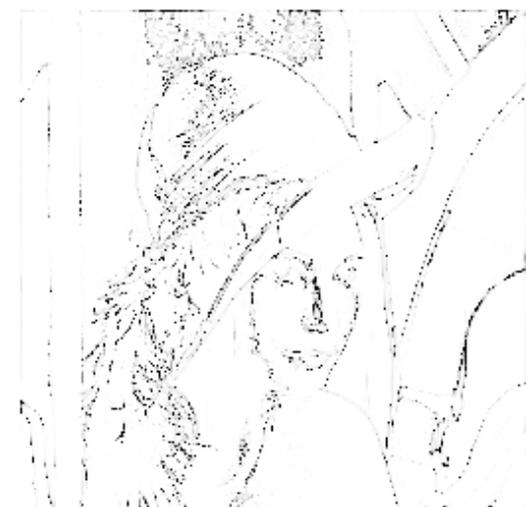
(a) *Imagem original*(b) *Imagem degradada*(c) *Imagem restaurada com o TDCMA - $\mu = 10^{-11}$ - %MSE = 3.92*(d) *Imagem restaurada com o TDRMMA - $\mu = 10^{-3}$ - %MSE = 3.26*(e) *Erro com o TDCMA - $\mu = 10^{-11}$* (f) *Erro com o TDRMMA-DFE - $\mu = 10^{-3}$*

Figura 5.19: *Efeitos da equalização - Blur Gaussiano 3×3 - Níveis de cinza: 6 bits/pixels - Imagem 256×256 pixels - sem ruído*

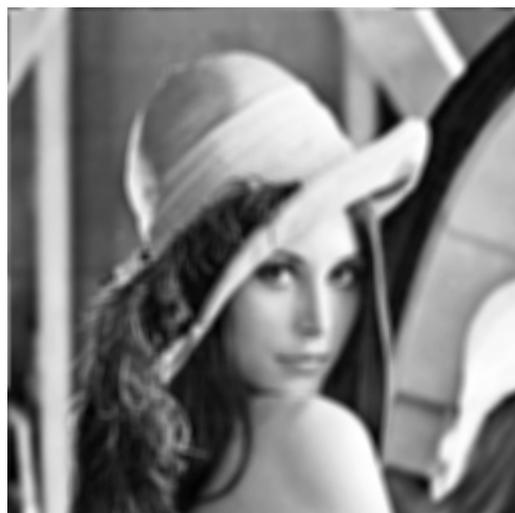
(a) *Imagem original*(b) *Imagem degradada*(c) *Imagem restaurada com o TDCMA - $\mu = 5 \times 10^{-14}$ - %MSE = 22.11*(d) *Imagem restaurada com o TDRMMA - $\mu = 10^{-3}$ - %MSE = 6.17*(e) *Erro com o TDCMA - $\mu = 5 \times 10^{-14}$* (f) *Erro com o TDRMMA - $\mu = 5 \times 10^{-3}$*

Figura 5.20: *Efeitos da equalização - Blur de movimento 1-D 5×5 - Níveis de cinza: 8 bits/pixel - Imagem 256×256 pixels - sem ruído*

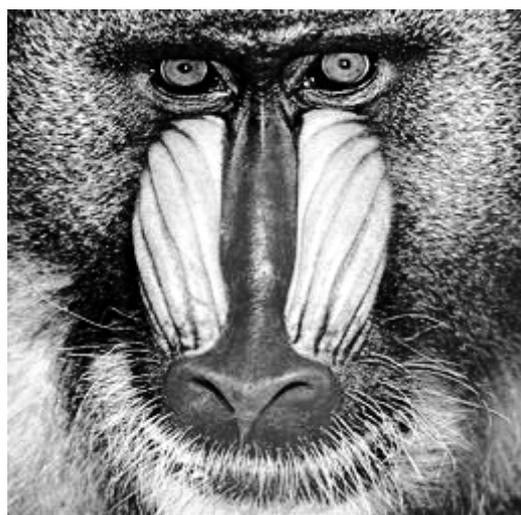
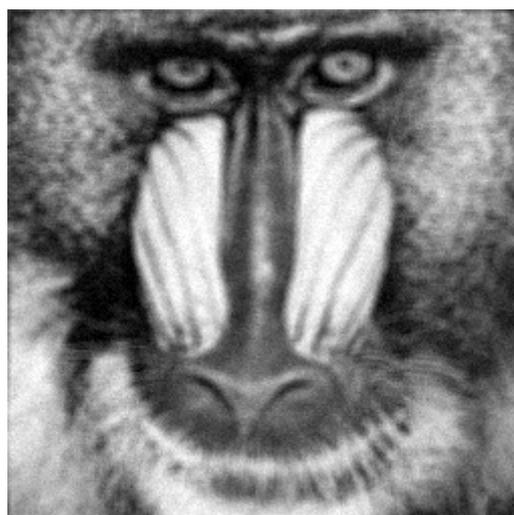
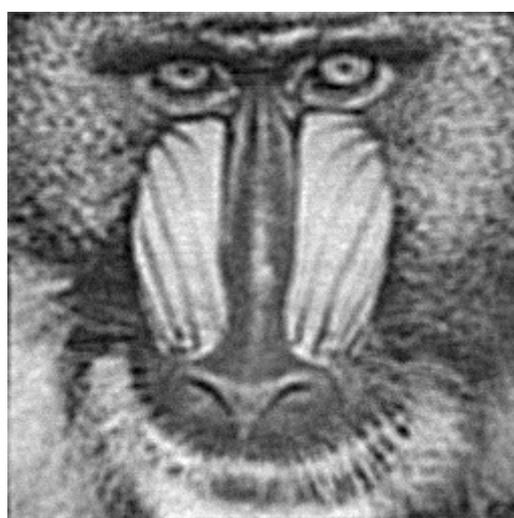
(a) *Imagem original*(b) *Imagem degradada*(c) *Imagem restaurada com o TDCMA - $\mu = 1 \times 10^{-13}$ - %MSE = 52.23*(d) *Imagem restaurada com o TDRMMA - $\mu = 10^{-3}$ - %MSE = 28.74*(e) *Erro com o TDCMA - $\mu = 10^{-13}$* (f) *Erro com o TDRMMA - $\mu = 10^{-3}$*

Figura 5.21: Efeitos da equalização - Blur Gaussiano 2-D 5×5 - Níveis de cinza: 8 bits/pixel - Imagem 256×256 pixels - BSNR=18 dB

5.5 O algoritmo multimódulo com decisão realimentada- o TDRMMA - DFE

O TDRMMA-DFE pode ser implementado basicamente efetuando a vetorização conforme mostrado na Seção 4.6 e seguindo o procedimento conforme Tabela 2.6 considerando um equalizador de decisão realimentada. Todos esses artifícios combinados, dão origem a um algoritmo interessante para as aplicações discutidas aqui. As Figuras 5.23 a 5.29 mostram a comparação da equalização considerando o TDRMMA-DFE e o TDCMA para diferentes níveis de cinza na imagem Lenna. As imagens com 2 bits/pixel (Fig. 5.23) e 3 bits/pixel (Fig. 5.24) foram degradadas com um *blur* Gaussiano 2-D 3×3 e as imagens com 4 bits/pixel (Fig. 5.25) a 8 bits/pixel (Fig. 5.29) foram degradadas também com *blur* Gaussiano 2-D, mas com dimensão 5×5 . A inserção da realimentação no equalizador faz com que o DFE seja mais robusto à complexidade do canal, gerando imagens recuperadas mais próximas da imagem original. A normalização pela potência do sinal de entrada do TDRMMA permitiu que as imagens com qualquer nível de quantização fossem sempre restauradas com um passo de adaptação $\mu = 3 \times 10^{-3}$, enquanto que com o TDCMA se tem uma grande dificuldade em se escolher o passo de adaptação: um passo muito grande causa divergência no algoritmo e um passo muito pequeno não resulta em melhorias da imagem. A Figura 5.5 mostra a estrutura em blocos o TDRMMA-DFE. Nesta estrutura, a realimentação começa a ocorrer a partir da segunda varredura da imagem. Os valores obtidos da primeira varredura são utilizados para definir as decisões passadas do algoritmo.

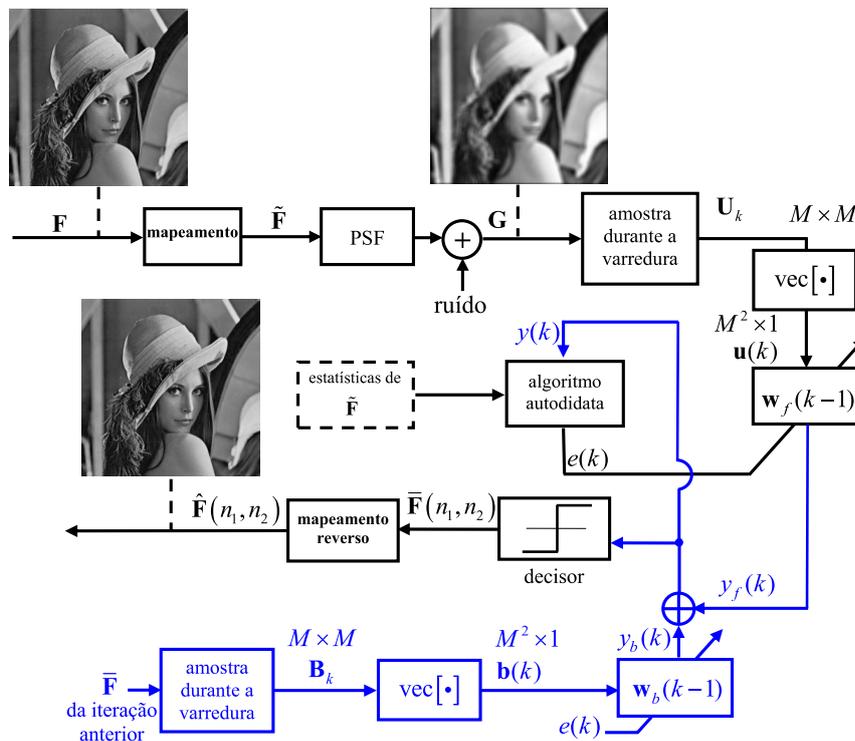


Figura 5.22: Arranjos em blocos do TDRMMA-DFE



(a) *Imagem original*



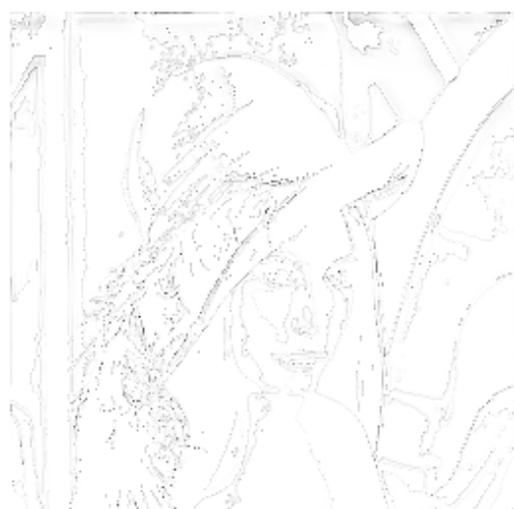
(b) *Imagem degradada*



(c) *Imagem restaurada com o TDCMA - $\mu = 10^{-6}$ - %MSE = 6.3*



(d) *Imagem restaurada com o TDRMMA-DFE - $\mu = 10^{-3}$ - %MSE = 6.3*



(e) *Erro com o TDCMA - $\mu = 10^{-6}$*



(f) *Erro com o TDRMMA-DFE - $\mu = 10^{-3}$*

Figura 5.23: *Efeitos da equalização - Blur Gaussiano 3×3 - Níveis de cinza: 2 bits/pixel - Imagem 256×256 pixels - sem ruído*



(a) *Imagem original*



(b) *Imagem degradada*



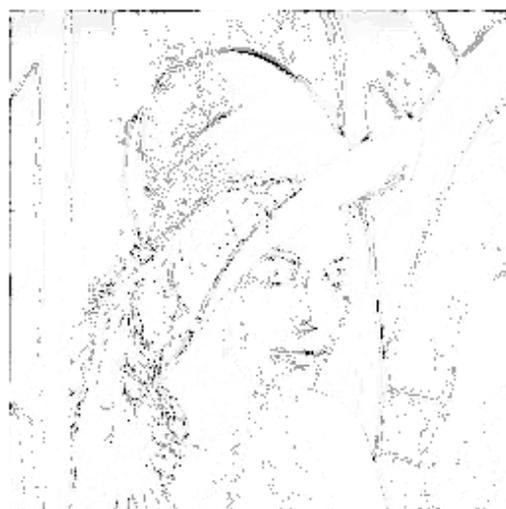
(c) *Imagem restaurada com o TDCMA - $\mu = 10^{-7}$ - %MSE = 12.0*



(d) *Imagem restaurada com o TDRMMA-DFE - $\mu = 10^{-3}$ - %MSE = 7.0*



(e) *Erro com o TDCMA - $\mu = 10^{-7}$*



(f) *Erro com o TDRMMA-DFE - $\mu = 10^{-3}$*

Figura 5.24: *Efeitos da equalização - Blur Gaussiano 3×3 - Níveis de cinza: 3 bits/pixel - Imagem 256×256 pixels - sem ruído*



(a) *Imagem original*



(b) *Imagem degradada*



(c) *Imagem restaurada com o TDCMA - $\mu = 3 \times 10^{-9}$ - $\%MSE = 20.1$*



(d) *Imagem restaurada com o TDRMMA-DFE - $\mu = 10^{-3}$ - $\%MSE = 9.6$*



(e) *Erro com o TDCMA - $\mu = 3 \times 10^{-9}$*



(f) *Erro com o TDRMMA-DFE - $\mu = 10^{-3}$*

Figura 5.25: *Efeitos da equalização - Blur Gaussiano 5×5 - Níveis de cinza: 4 bits/pixel - Imagem 256×256 pixels - sem ruído*



(a) *Imagem original*



(b) *Imagem degradada*



(c) *Imagem restaurada com o TDCMA - $\mu = 3 \times 10^{-10}$ - %MSE = 38.0*



(d) *Imagem restaurada com o TDRMMA-DFE - $\mu = 10^{-3}$ - %MSE = 10.4*



(e) *Erro com o TDCMA - $\mu = 3 \times 10^{-10}$*



(f) *Erro com o TDRMMA-DFE - $\mu = 10^{-3}$*

Figura 5.26: *Efeitos da equalização - Blur Gaussiano 5×5 - Níveis de cinza: 5 bits/pixel - Imagem 256×256 pixels - sem ruído*



(a) *Imagem original*



(b) *Imagem degradada*



(c) *Imagem restaurada com o TDCMA - $\mu = 2 \times 10^{-11}$ - %MSE = 43.8*



(d) *Imagem restaurada com o TDRMMA-DFE - $\mu = 10^{-3}$ - %MSE = 13.3*



(e) *Erro com o TDCMA - $\mu = 2 \times 10^{-11}$*



(f) *Erro com o TDRMMA-DFE - $\mu = 10^{-3}$*

Figura 5.27: *Efeitos da equalização - Blur Gaussiano 5×5 - Níveis de cinza: 6 bits/pixel - Imagem 256×256 pixels - sem ruído*



(a) *Imagem original*



(b) *Imagem degradada*



(c) *Imagem restaurada com o TDCMA - $\mu = 10^{-11}$ - %MSE = 32.0*



(d) *Imagem restaurada com o TDRMMA-DFE - $\mu = 10^{-3}$ - %MSE = 12.1*



(e) *Erro com o TDCMA - $\mu = 10^{-11}$*



(f) *Erro com o TDRMMA-DFE - $\mu = 10^{-3}$*

Figura 5.28: *Efeitos da equalização - Blur Gaussiano 5×5 - Níveis de cinza: 7 bits/pixel - Imagem 256×256 pixels - sem ruído*



(a) *Imagem original*



(b) *Imagem degradada*



(c) *Imagem restaurada com o TDCMA - $\mu = 10^{-12}$ - %MSE = 63.0*



(d) *Imagem restaurada com o TDRMMA-DFE - $\mu = 10^{-3}$ - %MSE = 13.5*



(e) *Erro com o TDCMA - $\mu = 10^{-12}$*



(f) *Erro com o TDRMMA-DFE - $\mu = 10^{-3}$*

Figura 5.29: *Efeitos da equalização - Blur Gaussiano 5×5 - Níveis de cinza: 8 bits/pixel - Imagem 256×256 pixels*

Como mencionado no Capítulo 2, quando se trabalha com DFEs autodidatas se enfrenta o problema das chamadas soluções degeneradas em que a saída do equalizador se torna independente do sinal de entrada. Esta é uma situação possível de acontecer no TDRMMA que não usa restrições como o algoritmo DFE-CMA-FB da Tabela 2.5 da página 19, como se pode ver na Figura 5.30(a). Neste caso, a solução degenerada foi induzida propositalmente, gerando uma degradação excessiva diante do números de bits/pixel que a imagem está quantizada. Com isso, por volta da metade da imagem o equalizador deixou de depender do sinal de entrada e não existe atualização dos coeficientes do filtro. Dessa maneira, o que se vê é uma faixa contínua seguindo pela metade da imagem em diante sendo uma recuperação errônea em relação à imagem original. Diante disso, se pode estender a idéia de Szczecinski e Gei [2002] que insere condições especiais para que não haja o acontecimento de soluções degeneradas. O resultado da utilização das condições estendidas do DFE-CMA-FB para o TDRMMA-DFE pode ser visto na Figura 5.30(b) que consegue equalizar a imagem (mesmo com uma intensa degradação) sob a mesma PSF e com o mesmo passo de adaptação da Figura 5.30(a).

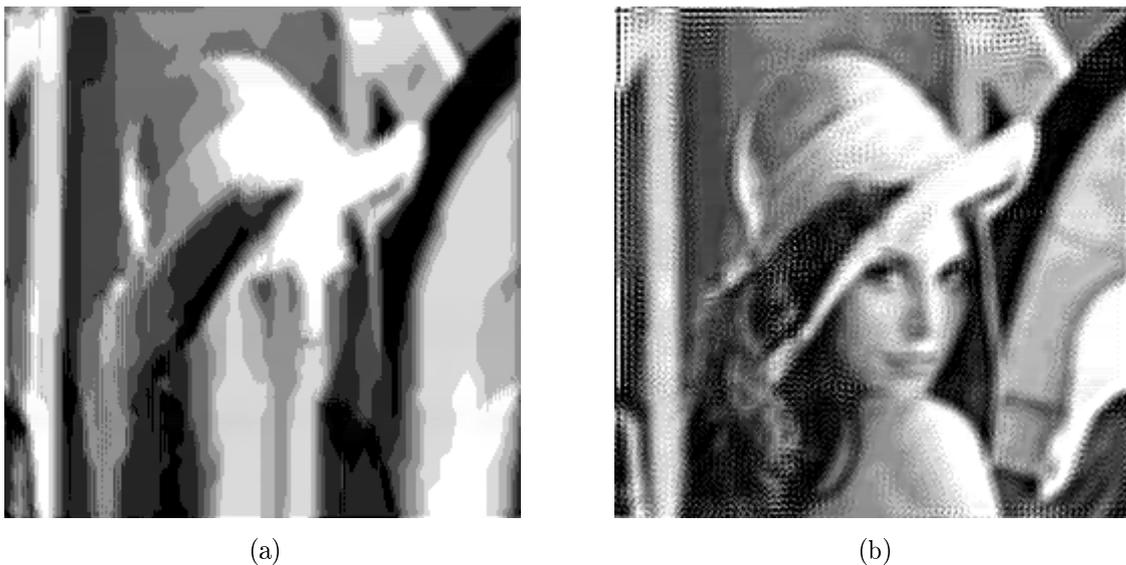


Figura 5.30: (a) Resultado da restauração com o TDRMMA (b) Resultado da restauração com o TDRMMA que evita soluções degeneradas - Blur Gaussiano 2-D 7×7 - Níveis de cinza: 3 bits/pixel - Imagem 256×256 pixels - $\mu = 10^{-3}$

Capítulo 6

Conclusões e trabalhos futuros

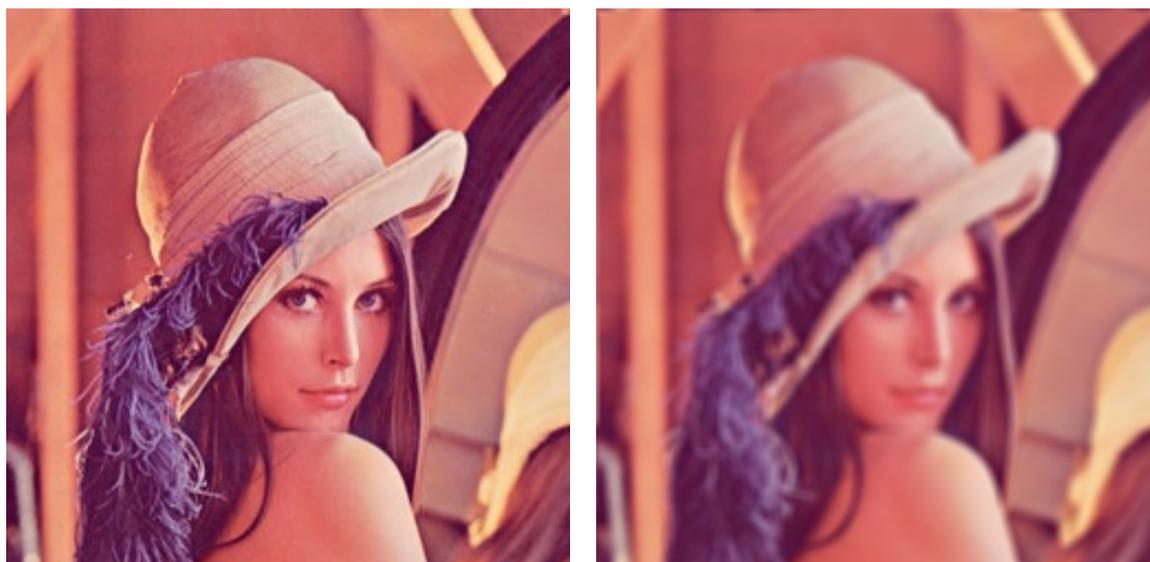
Com a revisão dos equalizadores e o entendimento das características dos filtros adaptativos bidimensionais, fica claro que há uma série de possibilidades de extensão dos conceitos para o uso na restauração de imagens. Com considerações quanto à inicialização, varredura e vetorização de janelas, é possível inclusive utilizar os algoritmos como foram propostos sem ressalvas. A extensão do RMMA apresentou resultados muito bons para se equalizar imagens com grande constelação de símbolos e forneceu uma melhoria perceptível em relação aos resultados que se obtém com o TDCMA convencional. A combinação do algoritmo TDRMMA com uma varredura que tenta evitar mudanças bruscas, uma inicialização de bordas apropriada em um equalizador de decisão realimentada se obteve bons resultados que podem ser vistos da Figura 5.18 até a Figura 5.30. Os testes com o TDRMMA-DFE mostraram que o algoritmo apresenta alguns problemas com a inserção de ruído quando implementada a realimentação. Nenhuma ressalva do gênero ocorreu usando o algoritmo em um equalizador linear transversal. Isso será investigado futuramente para que este problema possa ser contornado e então o algoritmo possa funcionar na presença de ruído sem necessidade de pré-filtragem. Os resultados visuais não foram sempre quantificados de maneira efetiva pelo índice do $%MSE$. Em [Wang et al., 2002, Wang e Bovik, 2009] se demonstrou que o uso do erro quadrático médio para medidas em imagens talvez não seja a melhor escolha, principalmente porque algumas imagens que apresentam claras diferenças visuais de degradação, apresentam índices muito próximos. De qualquer maneira, isto não invalida qualquer resultado, uma vez que aos olhos do observador, todos os resultados obtidos com o TDRMMA e o TDRMMA-DFE foram satisfatórios e superiores àqueles obtidos com o TDCMA convencional. Em trabalhos futuros, se pretende utilizar melhor a quantificação dada pela similaridade estrutural conhecida como SSIM (*Structural Similarity*) [Wang et al., 2002, Wang e Bovik, 2009] com o objetivo de aperfeiçoar as comparações entre imagens recuperadas. Em todos os casos testados, os coeficientes do filtro foram inicializados como uma matriz pino, porém como visto no Capítulo 3, há dois formatos básicos para as degradações principais e talvez uma inicialização dos coeficientes do filtro com algo próximo a uma inversa comum destas degradações pode ajudar na convergência e diminuir distorções geradas por mínimos locais da equalização.

Dentro do contexto de extensão do RMMA e do DFE, se vê a oportunidade da extensão de outros conceitos oriundos da filtragem adaptativa de sinais de comunicação para o caso bidimensional. Como se notou em todos os diagramas de erro espacial mostrados, existe sempre um pico de erro cada vez que se tenta equalizar uma alteração abrupta na imagem. Isso acontece porque o algoritmo adaptativo tenta fazer mudanças suaves baseadas no erro estimado. Com os passos utilizados aqui para se obter bons resultados de equalização, fica claro que não se pode alterar os coeficientes do filtro de forma abrupta para se conseguir uma estimativa melhor nessas regiões. É dentro deste contexto que se pretende propor soluções com o TDRMMA com um passo de adaptação variável para minimizar o erro quadrático nas regiões mencionadas. Outro conceito de equalização de canais de comunicação que pode ser adaptado para a restauração de imagens é o uso da diversidade espacial. Este conceito considera que um sinal é transmitido por vários meios e que a diversidade gerada pela degradação nos sinais recebidos adiciona informação sobre o sinal original. Com esta informação adicional obtida pela diversidade, se consegue obter algo melhor ou no mínimo tão bom quanto o caso da transmissão por um único meio. No caso da restauração de imagens, não estaria se considerando enviar o mesmo sinal por vários meios, mas sim, utilizar o fato de que em uma imagem colorida existem três camadas da imagem com diferenças entre si devido aos níveis de cada camada de cor por exemplo. Uma imagem colorida pode, por exemplo, ser decomposta por três camadas com gradação normal de 0 a 256 que constituem o sistema RGB (*red, green, blue*). Cada camada possui uma versão da imagem com diferenças entre si, porém com uma forte correlação entre si. A Figura 6.1 mostra um exemplo em que a imagem Lenna é decomposta em suas camadas, vermelha, verde e azul. No caso, o intuito é utilizar as diferenças entre essas camadas para como diversidade espacial. Na transmissão de uma imagem colorida, as três camadas são submetidas a mesma degradação, porém continuam diferentes desde sua origem. Assim se pode utilizar a estimativa do pixel de cada camada para se efetuar a estimativa da próxima devido a forte correlação.



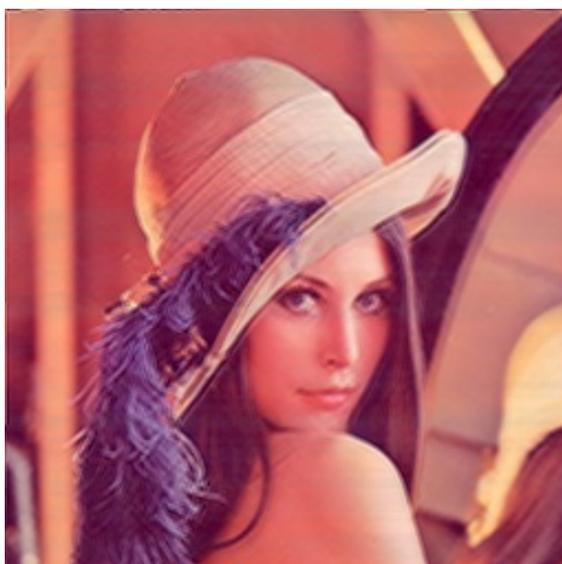
Figura 6.1: *Decomposição das camadas da imagem - (a) imagem colorida, (b) camada vermelha, (c) camada verde, (d) camada azul*

Em testes preliminares, o uso da diversidade se mostrou extremamente favorável para a restauração, recuperando a definição de várias áreas como se nota na Figura 6.2 em que se equalizou uma imagem colorida utilizando o TDRMMA-DFE e levando em conta a diversidade espacial. Esses resultados mostram uma boa oportunidade de trabalhos futuros estendendo o estudo para imagens coloridas e abrangendo as aplicações do algoritmo para questões mais práticas.



(a)

(b)



(c)

Figura 6.2: *Simulação TDRMMA com diversidade espacial - (a) imagem original, (b) imagem degradada, (c) imagem recuperada*

Referências Bibliográficas

- R. Abreu e M. T. M. Silva. A multimodulus algorithm for blind image deconvolution *In Proceedings of International Workshop on Telecommunications (IWT-11)*, pp. 35 –41, 2011.
- A. Albiol, J. M. Mossi, V. Naranjo, e L. Vergara. Two-dimensional LMS adaptation strategies for nonstationary signals. *Digital Signal Processing*, vol. 7, pp. 3 – 12, 1997.
- T. E. Bishop, S. D. Babacan, B. Amizic, A. K. Katsaggelos, T. Chan, e R. Molina. *Blind image deconvolution: problem formulation and existing approaches*. CRC press, 2007.
- P. Campisi e K. Egiazarian, editors. *Blind image deconvolution: theory and applications*. CRC press, 2007.
- M. Ekstrom. Realizable Wiener filtering in two dimensions. *Acoustics, Speech and Signal Processing, IEEE Transactions on*, vol. 30, pp. 31 –40, 1982.
- B. Farhang-Boroujeny. *Adaptive Filters - Theory and Applications*. John Wiley & Sons, West Sussex, 1998.
- S. Gazor e B. Farhang-Boroujeny. Quantization effects in transform-domain normalized LMS algorithm. *Circuits and Systems II: Analog and Digital Signal Processing, IEEE Transactions on*, vol. 39, pp. 1 –7, 1992.
- D. N. Godard. *Self-recovering equalization and carrier tracking in two dimensional data communication system*, vol. 28, pp. 1867 –1875, 1980.
- R. C. Gonzalez e R. E. Woods. *Digital Image Processing (3rd Edition)*. Prentice-Hall, Inc., 2006.
- M.M. Hadhoud e D.W. Thomas. The two-dimensional adaptive LMS (TDLMS) algorithm. *Circuits and Systems, IEEE Transactions on*, vol.35, pp.485 –494, 1988.
- S. Haykin. *Unsupervised Adaptive Filtering:Blind Deconvolution*. John Wiley & Sons, Inc., 2000.
- S. Haykin. *Adaptive Filter Theory (4th Edition)*. Prentice Hall, September 2001.

- S. M. Jefferies e J. C. Christou. Restoration of astronomical images by iterative blind deconvolution. *Astrophysical Journal*, vol. 415, 862–+, 1993.
- R. Johnson Jr, P. Schniter, T.J. Endres, J.D. Behm, D.R. Brown, e R.A. Casas. Blind equalization using the constant modulus criterion: a review. *Proceedings of the IEEE*, vol. 86, 1927–1950, 1998.
- F. Kara e C. Vural. Exact image super-resolution for pure translational motion and shift-invariant blur. In *Signal Processing and Communications Applications Conference, 2009. SIU 2009. IEEE 17th*, pp. 968–972, 2009.
- F. Kara e C. Vural. *Blind image resolution enhancement based on a 2D constant modulus algorithm Inverse Problems*, vol. 24, pp. 1–15, 2008.
- D. Kundur e D. Hatzinakos. Blind image deconvolution. *Signal Processing Magazine, IEEE*, vol. 13, pp. 43–64, 1996a.
- D. Kundur e D. Hatzinakos. Blind image deconvolution revisited. *Signal Processing Magazine, IEEE*, vol. 13 pp. 61–63, 1996b.
- J. S. Lim. *Two-dimensional signal and image processing*. Prentice-Hall, Inc., 1990.
- J. Mendes Filho, M. T. M. Silva, M. D. Miranda, e V. H. Nascimento. A region-based algorithm for blind equalization of QAM signals. In *IEEE/SP 15th Workshop on Statistical Signal Processing (SSP'09)*, pp. 685–688, 2009.
- J. Mendes Filho, M. T. M. Silva, e M. D. Miranda. A family of algorithms for blind equalization of QAM signals. In *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Process.*, pp. 3388–3391, 2011.
- O.V. Michailovich e D. Adam. A novel approach to the 2-D blind deconvolution problem in medical ultrasound. *Medical Imaging, IEEE Transactions on*, vol. 24, pp. 86–104, 2005.
- M.D. Miranda, M.T.M. Silva, e V.H. Nascimento. Avoiding divergence in the constant modulus algorithm. In *Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on*, pp. 3565–3568, 2008.
- M. Ohki e S. Hashiguchi. A new 2-D LMS adaptive algorithm. In *Acoustics, Speech, and Signal Processing, 1991. ICASSP-91., 1991 International Conference on*, vol. 3, pp. 2113–2116, 1991.
- S.C. Pei, C.Y. Lin, e C.C. Tseng. Two-dimensional LMS adaptive linear phase filters. In *Circuits and Systems, 1993., ISCAS '93, 1993 IEEE International Symposium on*, vol. 1, pp. 311–314, 1993.

- F. Q. Qin, J. Min, e H. R. Guo. A blind image restoration method based on PSF estimation. In *Software Engineering, 2009. WCSE '09. WRI World Congress on*, vol. 2, pp. 173–176, 2009.
- S.U.H. Qureshi. Adaptive equalization. *Proceedings of the IEEE*, vol. 73, pp.1349 – 1387, 1985.
- P.D. Samarasinghe e R.A. Kennedy. Minimum kurtosis CMA deconvolution for blind image restoration. In *Information and Automation for Sustainability, 2008. ICIAFS 2008. 4th International Conference on*, pp. 271–276, 2008.
- T. J. Schulz. Multiframe blind deconvolution of astronomical images. *Journal of the Optical Society of America A*, vol. 10, pp. 1064–1073, 1993.
- M. T. M. Silva. *Um estudo sobre técnicas de equalização autodidata*. Tese de Doutorado em Engenharia Elétrica, Escola Politécnica da Universidade de São Paulo, São Paulo, 2005.
- J. R. Swedlow, J. W. Sedat, e D. A. Agard. Deconvolution in optical microscopy. pp 284–309, 1996.
- L.L. Szczecinski e A. Gei. Blind decision feedback equalisers, how to avoid degenerative solutions. *Signal Processing and Communications*, vol. 82, pp. 1675–1693, 2002.
- H. Tang e L.W. Cahill. A new approach for the restoration of noisy blurred images. In *Circuits and Systems, 1991., IEEE International Symposium on*, vol.1, pp 520–523, 1991.
- C. Vural e W.A. Sethares. Blind deconvolution of noisy blurred images via dispersion minimization. In *Digital Signal Processing, 2002. DSP 2002. 2002 14th International Conference on*, vol.2, pp. 787–790, 2002.
- Z. Wang e A.C. Bovik. Mean squared error: Love it or leave it? a new look at signal fidelity measures. *Signal Processing Magazine, IEEE*, vol. 26, pp. 98–117, 2009.
- Z. Wang, A.C. Bovik, e L. Lu. Why is image quality assessment so difficult? In *Acoustics, Speech, and Signal Processing, 1993. ICASSP-93., 1993 IEEE International Conference on*, vol. 4, pp. I–V, 2002.
- C. Yen Wen e C. Hsiung Lee. Point spread functions and their applications to forensic image restoration. *Forensic Science Journal*, vol.1, pp. 15–26, 2002.
- B. Widrow e S. D. Stearns. *Adaptive signal processing*. Prentice Hall, 1985.

Apêndice A

Fundamentos da Imagem Digital

Neste apêndice são apresentados conceitos básicos de imagens e de sistemas de imagens. Descrevem-se a formação da imagem, a Transformada de Fourier bidimensional, a representação de imagens no domínio das frequências e a integridade. Cabe observar que existem inúmeros livros publicados sobre esse assunto, destacando-se dois textos clássicos: [Lim, 1990] e [Gonzalez e Woods, 2006] nos quais este apêndice foi inspirado.

A.1 A Imagem

Pode-se descrever uma imagem digital como um sinal discreto no espaço identificando sua posição por dois argumentos inteiros $\mathbf{X}(n_1, n_2)$. As imagens digitais são imagens analógicas quantizadas normalmente em 256 níveis (8 bits/pixel). Por exemplo, $\mathbf{X}(n_1, n_2)$ representa uma sequência que é definida para todos os valores inteiros de n_1 e n_2 e para os valores não inteiros a sequência é indefinida. Para o processamento digital de sinais de comunicação unidimensionais, a função impulso unitário é a principal função para a geração de qualquer sinal. No caso bidimensional a função impulso unitário é definida por

$$\delta(n_1, n_2) = \begin{cases} 1 & n_1, n_2 = 0; \\ 0 & \text{caso contrário.} \end{cases} \quad (\text{A.1})$$

Uma vez que se tem a definição do impulso unitário para o processamento digital bidimensional, pode-se definir qualquer sinal bidimensional em função do somatório de impulsos unitários deslocados no espaço, ou seja,

$$\begin{aligned} \mathbf{X}(n_1, n_2) &= \dots + \mathbf{X}(-1, -1)\delta(n_1 + 1, n_2 + 1) + \mathbf{X}(0, -1)\delta(n_1, n_2 + 1) + \mathbf{X}(0, 0)\delta(n_1, n_2) + \\ &\quad + \mathbf{X}(0, 1)\delta(n_1, n_2 - 1) + \mathbf{X}(1, 1)\delta(n_1 - 1, n_2 - 1) + \dots = \\ &= \sum_{k_1=-\infty}^{\infty} \sum_{k_2=-\infty}^{\infty} \mathbf{X}(k_1, k_2)\delta(n_1 - k_1, n_2 - k_2). \end{aligned} \quad (\text{A.2})$$

Uma maneira usualmente utilizada para representar graficamente uma sequência 2-D ou uma imagem é indicando os valores como na Figura A.1 em que os círculos abertos indicam pontos com módulo zero e os círculos preenchidos juntamente com uma indicação de valor entre parênteses demonstram pontos de valor não nulo.

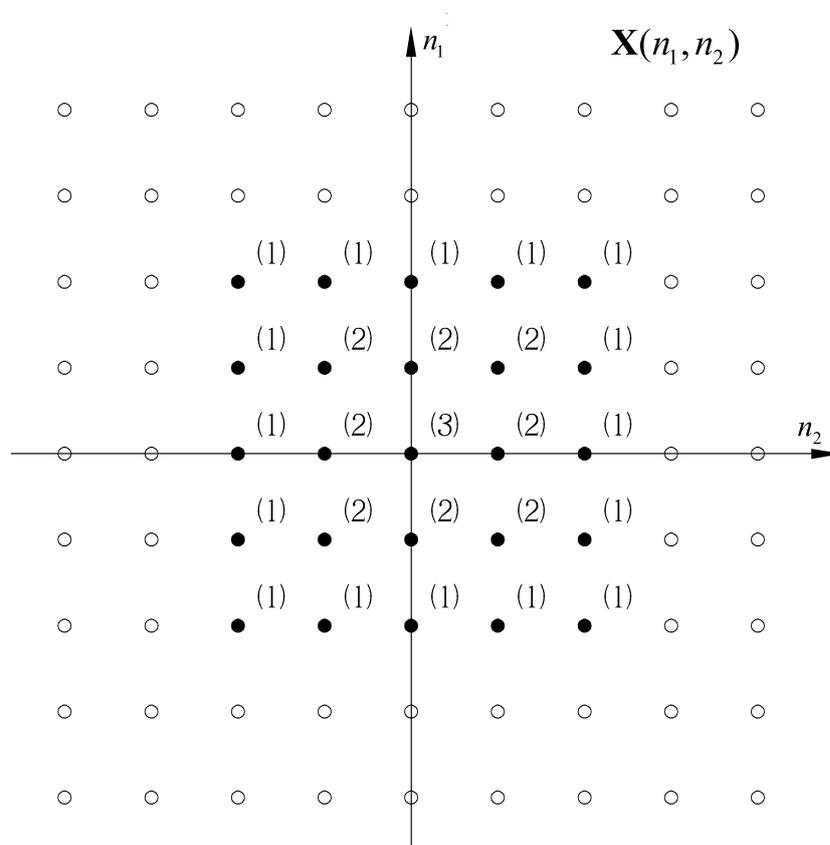


Figura A.1: *Representação gráfica de uma sequência bidimensional*

A.2 O Sistema Bidimensional

Os sistemas são as relações entre a entrada e a saída para uma dada entrada. Pode-se definir o sistema utilizando o operador $T[\cdot]$ e dessa forma um sistema bidimensional pode ser definido como

$$\mathbf{Y}(n_1, n_2) = T[\mathbf{X}(n_1, n_2)]. \quad (\text{A.3})$$

A maioria de sistemas de imagens ou sinais bidimensionais seguem as condições de linearidade e invariância no espaço. A condição da linearidade se baseia na superposição das condições da propriedade aditiva com a propriedade da homogeneidade.

Define-se a propriedade aditiva em sinais bidimensionais como

$$T[\mathbf{X}_1(n_1, n_2) + \mathbf{X}_2(n_1, n_2)] = \mathbf{Y}_1(n_1, n_2) + \mathbf{Y}_2(n_1, n_2)$$

E por sua vez, define-se a propriedade da homogeneidade como

$$T[a\mathbf{X}(n_1, n_2)] = aT[\mathbf{X}(n_1, n_2)] = a\mathbf{Y}(n_1, n_2),$$

sendo a uma constante qualquer. Com a superposição dessas propriedades tem-se a definição matemática da condição de linearidade dada por

$$T[a\mathbf{X}_1(n_1, n_2) + b\mathbf{X}_2(n_1, n_2)] = a\mathbf{Y}_1(n_1, n_2) + b\mathbf{Y}_2(n_1, n_2). \quad (\text{A.4})$$

A condição de invariância no espaço implica que a saída não é dependente da posição no sinal e sim apenas do módulo dos pontos envolvidos, ou seja,

$$T[\mathbf{X}(n_1 - k_1, n_2 - k_2)] = \mathbf{Y}(n_1 - k_1, n_2 - k_2). \quad (\text{A.5})$$

Agora combinando as equações (A.3) e (A.2) verifica-se que

$$\mathbf{Y}(n_1, n_2) = T[\mathbf{X}(n_1, n_2)] = T \left[\sum_{k_1=-\infty}^{\infty} \sum_{k_2=-\infty}^{\infty} \mathbf{X}(k_1, k_2) \delta(n_1 - k_1, n_2 - k_2) \right]. \quad (\text{A.6})$$

Aplicando a propriedades dos sistemas lineares invariantes no espaço, obtém-se

$$\mathbf{Y}(n_1, n_2) = \sum_{k_1=-\infty}^{\infty} \sum_{k_2=-\infty}^{\infty} \mathbf{X}(k_1, k_2) T[\delta(n_1 - k_1, n_2 - k_2)]. \quad (\text{A.7})$$

Desta maneira, define-se o sistema linear invariante no espaço que por simplificação é usual denotá-lo pela sigla LSI (*Linear Shift Invariant*) [Gonzalez e Woods, 2006]. Da mesma forma que acontece em sinais unidimensionais, para simplificação define-se $T[\delta(n_1, n_2)]$ como a resposta impulsiva do sistema $\mathbf{H}(n_1, n_2)$ como está esquematizado na Figura A.2. Dessa forma e com a propriedade da invariância no espaço, pode-se chegar até a definição do operador convolução bidimensional descrita como

$$\begin{aligned} \mathbf{Y}(n_1, n_2) &= \sum_{k_1=-\infty}^{\infty} \sum_{k_2=-\infty}^{\infty} \mathbf{X}(k_1, k_2) \underbrace{T[\delta(n_1 - k_1, n_2 - k_2)]}_{\mathbf{H}(n_1 - k_1, n_2 - k_2)} \\ \mathbf{Y}(n_1, n_2) &= \sum_{k_1=-\infty}^{\infty} \sum_{k_2=-\infty}^{\infty} \mathbf{X}(k_1, k_2) \mathbf{H}(n_1 - k_1, n_2 - k_2) \\ \mathbf{Y}(n_1, n_2) &= \mathbf{X}(n_1, n_2) * \mathbf{H}(n_1, n_2). \end{aligned} \quad (\text{A.8})$$

Com a definição da convolução fica simples demonstrar que o diagrama de blocos de um sistema bidimensional é uma extensão do conceito já conhecido para sinais de uma dimensão usuais. Dessa forma, as propriedades usuais da convolução também valem para sinais bidimensionais. As propriedades comutativa, associativa, distributiva e de deslocamento no espaço são revisitadas através de esquemas mostrados nas figuras A.3 a A.6.

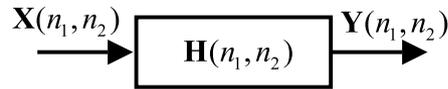


Figura A.2: Diagrama em blocos de um sistema de imagens

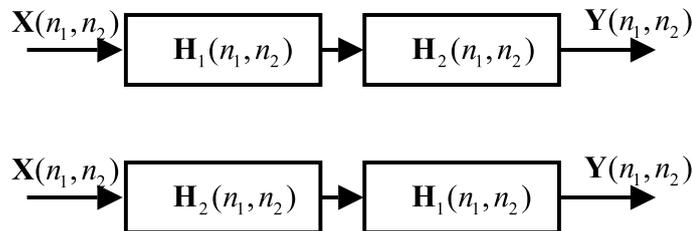


Figura A.3: Propriedades da convolução: Comutativa

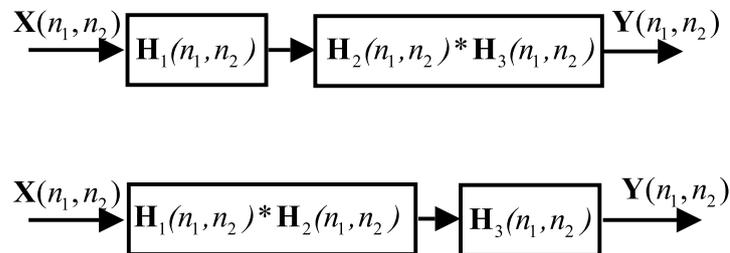


Figura A.4: Propriedades da convolução: Associativa

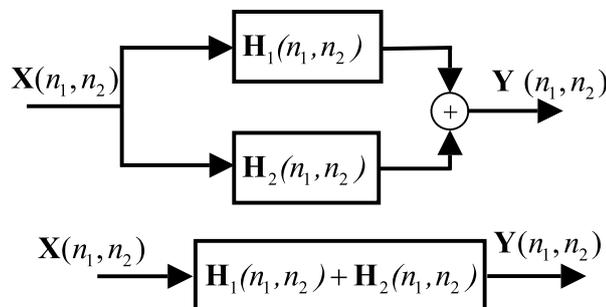


Figura A.5: Propriedades da convolução: Distributiva

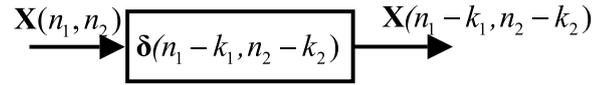


Figura A.6: *Propriedades da convolução: Deslocamento no espaço*

Em sistemas bidimensionais não há preocupação com a causalidade uma vez que a imagem completa é fornecida para ser processada em um instante de tempo [Gonzalez e Woods, 2006], já a estabilidade do sistema LSI como em sistemas unidimensionais dependem diretamente de que a resposta impulsiva do sistema seja absolutamente somável, ou seja,

$$S = \sum_{n_1=-\infty}^{\infty} \sum_{n_2=-\infty}^{\infty} |\mathbf{H}(n_1, n_2)| < \infty \quad (\text{A.9})$$

A.3 A Transformada de Fourier e a inteligibilidade

Uma vez que se tem uma sequência estável, se pode representar a mesma através de sua Transformada de Fourier usando a representação $X(\omega_1, \omega_2)e^{j\omega_1 k_1} e^{j\omega_2 k_2}$ em que $X(\omega_1, \omega_2)$ denota a magnitude da transformada e está associada a um par de exponenciais complexas.

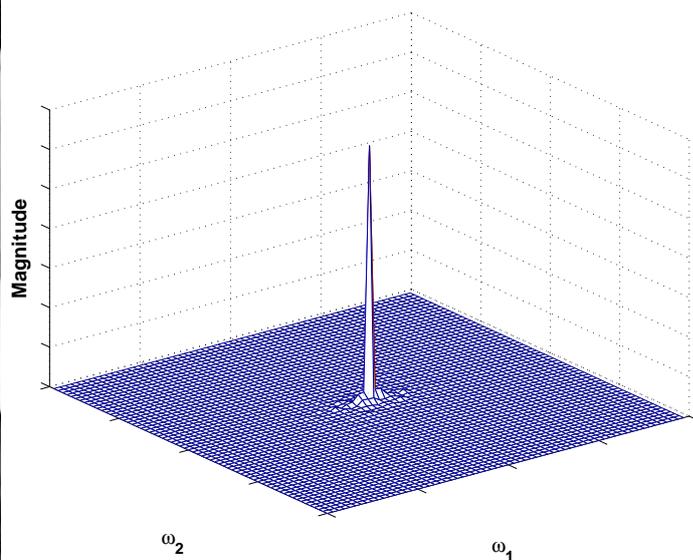
As relações entre $\mathbf{X}(k_1, k_2)$ e $X(\omega_1, \omega_2)$ são dadas pelo par transformado

$$X(\omega_1, \omega_2) = \sum_{k_1=-\infty}^{\infty} \sum_{k_2=-\infty}^{\infty} \mathbf{X}(k_1, k_2) e^{-j\omega_1 k_1} e^{-j\omega_2 k_2} \quad (\text{A.10})$$

$$\mathbf{X}(k_1, k_2) = \frac{1}{(2\pi)^2} \int_{\omega_1=-\pi}^{\pi} \int_{\omega_2=-\pi}^{\pi} X(\omega_1, \omega_2) e^{j\omega_1 k_1} e^{j\omega_2 k_2} d\omega_1 d\omega_2. \quad (\text{A.11})$$

Cabe observar que $X(\omega_1, \omega_2)$ é uma função de variáveis contínuas ω_1 e ω_2 e apresenta periodicidade em um período de $2\pi \times 2\pi$ de forma que $X(\omega_1, \omega_2) = X(\omega_1 + 2\pi, \omega_2) = X(\omega_1, \omega_2 + 2\pi)$.

A representação da Transformada de Fourier de imagens típicas é definida normalmente com grande parte da energia concentrada em uma pequena região no domínio das frequências próximo da origem e pelos eixos de ω_1 , e ω_2 . A razão para a concentração em torno da origem é que as imagens típicas possuem grandes área sem alteração de gradação e este fato gera espectros de baixas frequências e as transições bruscas contribuem com frequências baixas da mesma forma que contribuem com frequências altas. A energia concentrada pelos eixos se dá pela descontinuidade artificial da imagem em suas bordas. Uma representação da imagem Lenna no domínio das frequências é mostrada na Figura A.8 em que se nota estes conceitos. Já que grande parte da energia se concentra em uma pequena área do domínio das frequências, a imagem pode ser reconstruída usando uma pequena fração dos coeficientes da transformada de Fourier [Lim, 1990, Gonzalez e Woods, 2006].

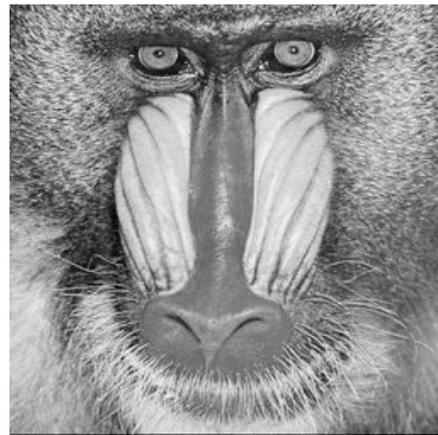
Figura A.7: *Lenna*Figura A.8: *Transformada de Fourier de Lenna*

Outra propriedade interessante de imagens é que grande parte da inteligibilidade da mesma se encontra na fase da transformada de Fourier. Esse fato pode ser constatado facilmente pela representação na Figura A.9¹. Na Figura A.9(a) e Figura A.9(b) se vê as imagens originais conservando sua amplitude e fase. Considerando a Transformada de Fourier, a Figura A.9(a) pode ser representada com a amplitude $|X_1(\omega_1, \omega_2)|$ e a fase $\theta_1(\omega_1, \omega_2)$ e a Figura A.9(b) pode ser representada com amplitude $|X_2(\omega_1, \omega_2)|$ e a fase $\theta_2(\omega_1, \omega_2)$. A partir destas representações, a Transformada de Fourier da Figura A.9(c) foi gerada com módulo $|X_1(\omega_1, \omega_2)|$ e fase $\theta_2(\omega_1, \omega_2)$ enquanto que a Figura A.9(d) foi gerada com módulo $|X_2(\omega_1, \omega_2)|$ e fase $\theta_1(\omega_1, \omega_2)$. De volta representação espacial se nota que as imagens geradas herdaram os traços e inteligibilidade da fase da imagem imposta.

¹Na Figura A.7 se utiliza as imagens "Mandrill" e "Lenna" para esclarecer o exemplo. Essas são imagens largamente usadas em Processamento de imagens como imagens de teste e pode ser obtida por exemplo em www.hlevkin.com/TestImages/classic.htm



(a) *Lenna* *original:*
 $|X_1(\omega_1, \omega_2)| \angle \theta_1(\omega_1, \omega_2)$



(b) *Mandrill* *original:*
 $|X_2(\omega_1, \omega_2)| \angle \theta_2(\omega_1, \omega_2)$



(c) *Módulo:Lenna, Fase:Mandrill*
 $|X_1(\omega_1, \omega_2)| \angle \theta_2(\omega_1, \omega_2)$



(d) *Módulo:Mandrill, Fase:Lenna*
 $|X_2(\omega_1, \omega_2)| \angle \theta_1(\omega_1, \omega_2)$

Figura A.9: *Inteligibilidade - relação módulo e fase*