Campus de São Carlos

RESTAURAÇÃO DE IMAGENS MÉDICAS UTILIZANDO
O FILTRO DE KALMAN

Autor: Edson Batista de Mello
Orientadora: Profª. Drª. Vilma Alves de Oliveira

UNIVERSIDADE DE SÃO PAULO

**ESCOLA DE ENGENHARIA
DE SÃO CARLOS**

RESTAURAÇÃO DE IMAGENS MÉDICAS UTILIZANDO O FILTRO DE KALMAN

Edson Batista de Mello

**Dissertação apresentada à Escola de
Engenharia de São Carlos da
Universidade de São Paulo, como
parte dos requisitos para obtenção
do título de Mestre em Engenharia
Elétrica**

Orientadora: Dr^a Vilma Alves de Oliveira

**São Carlos
1998**

Class.	TESE - EESC
Cutt.	3489
Tombo	0255/99

S/S 1031540

Ficha catalográfica preparada pela Seção de Tratamento
da Informação do Serviço de Biblioteca – EESC/USP

M527r Mello, Edson Batista de
Restauração de imagens médicas utilizando o filtro
de Kalman / Edson Batista de Mello. -- São Carlos,
1998.

Dissertação (Mestrado) -- Escola de Engenharia de
São Carlos-Universidade de São Paulo, 1998.
Área: Engenharia Elétrica.
Orientador: Prof^a. Dr^a. Vilma Alves de Oliveira.

1. Restauração de imagens. 2. Modelos AR.
3. Filtros de Kalman 2-D. I. Título.

FOLHA DE APROVAÇÃO

Candidato: Engenheiro **EDSON BATISTA DE MELLO**

Dissertação defendida e aprovada em 13.10.1998
pela Comissão Julgadora:

Vilma

Profa. Dra. **VILMA ALVES DE OLIVEIRA (Orientadora)**
(Escola de Engenharia de São Carlos - Universidade de São Paulo)

Nelson DelFINO d'Ávila Mascarenhas

Prof. Dr. **NELSON DELFINO D'ÁVILA MASCARENHAS**
(Universidade Federal de São Carlos - UFSCar)

Marcelo

Prof. Dr. **MARCELO BASÍLIO JOAQUIM**
(Escola de Engenharia de São Carlos - Universidade de São Paulo)

Denis

Prof. Doutor **DENIS VINICIUS COURY**
Coordenador da Área de Engenharia Elétrica

Jose Carlos A. Cintra

JOSÉ CARLOS A. CINTRA
Presidente da Comissão de Pós-Graduação da EESC

À minha família e amigos

À Doutora Vilma A. Oliveira, minha orientadora neste mestrado, pela ajuda prestada no decorrer do trabalho.

Ao Doutor Marcelo B. Joaquim pela prontidão com que me recebeu para solucionar as duvidas que surgiram durante a elaboração deste trabalho.

Ao Departamento de Engenharia Elétrica da EESC/USP.

Ao CNPq pela bolsa de estudos concedida.

SUMÁRIO

ÍNDICE DE FIGURAS	I
RESUMO	III
ABSTRACT	IV
1. INTRODUÇÃO	1
2. REVISÃO BIBLIOGRÁFICA.....	3
2.1 MODELOS DE IMAGEM E OBSERVAÇÃO	3
2.1.1. MODELOS DE IMAGEM.....	3
2.1.2. MODELOS DE OBSERVAÇÃO	6
2.2 RECONSTRUÇÃO	13
2.2.1. MÉTODO DA RECONSTRUÇÃO ALGÉBRICA (A.R.T)	15
2.2.2. MÉTODO DA TRANSFORMADA DE FOURIER.....	15
2.2.3. MÉTODO DA RETROPROJEÇÃO FILTRADA	16
2.2.4. PRÉ-RECONSTRUÇÃO RESTAURAÇÃO E PÓS-RECONSTRUÇÃO RESTAURAÇÃO.....	18
2.3 RESTAURAÇÃO	19
2.3.1. MÍNIMA NORMA.....	21
2.3.2. MÍNIMO ERRO QUADRÁTICO MÉDIO LINEAR.....	21
2.3.3. MÁXIMA PROBABILIDADE A POSTERIORI (MAP).....	22
2.3.4. MÍNIMOS QUADRADOS COM RESTRIÇÕES	22
2.3.5. REGULARIZAÇÃO DE MILLER	23
2.3.6. MÁXIMA ENTROPIA	23
2.3.7. ALGORITMOS ITERATIVOS COM RESTRIÇÕES	24
2.3.8. ALGORITMOS ESPECIALIZADOS.....	25

2.3.8. ALGORITMOS ESPECIALIZADOS	25
2.4 FILTROS	27
2.4.1. FILTRO DE WIENER	27
2.4.2. FILTRO DE METZ	29
2.4.3. FILTRO DE KALMAN	29
3. FORMAS DE IMPLEMENTAÇÃO DO FILTRO DE KALMAN BIDIMENSIONAL	37
3.1 FILTRO DE KALMAN 2-D ESCALAR	37
3.2 FILTRO DE KALMAN 2-D VETORIAL	39
3.3 FILTRO DE KALMAN DE ATUALIZAÇÃO REDUZIDA (RUKF)	42
4. IDENTIFICAÇÃO DOS PARÂMETROS DA IMAGEM	44
4.1 ALGORITMO DE LEVINSON	44
4.2 DETERMINAÇÃO DA ORDEM DO MODELO	50
4.3 RUÍDO DE OBSERVAÇÃO	52
5. IMPLEMENTAÇÃO DO FILTRO DE KALMAN	54
6. RESULTADOS	56
7. DISCUSSÕES E CONCLUSÕES	61
APÊNDICE	63
LISTAGEM DO PROGRAMA KALMAN E SUBROTINAS UTILIZADAS	63
REFERÊNCIAS BIBLIOGRÁFICAS	95

ÍNDICE DE FIGURAS

FIGURA 2-1 ORDEM DE VARREDURA DOS PIXELS (KAUFMAN AND TEKALP, 1991).....	5
FIGURA 2-2 REGIÕES DO SUPORTE CAUSAL NSHP. (A) SEMI-PLANO SUPERIOR. (B) SEMI-PLANO INFERIOR (MARPLE, 1987).....	5
FIGURA 2-3 REGIÃO DO SUPORTE CAUSAL QP PARA O PRIMEIRO QUADRANTE (MARPLE, 1987).....	6
FIGURA 2-4 SISTEMAS DE COORDENADAS (X,Y) E RESPECTIVA CONVERSÃO PARA COORDENADAS POLARES (R,φ) (NAIME, 1994).	14
FIGURA 2-5 (A) DOIS PERFIS DE UM OBJETO RETANGULAR CONSTITUÍDO DE MATERIAL HOMOGÊNEO SÃO MOSTRADOS GRAFICAMENTE. (B) ESTES PERFIS SÃO RETROPROJETADOS NO PLANO X-Y E SUPERPOSTOS FORMANDO UMA IMAGEM APROXIMADA DO OBJETO ORIGINAL (BROOKS AND DI CHIRO, 1976).....	16
FIGURA 2-6 RETROPROJEÇÃO. (A) ILUSTRAÇÃO DE QUATRO PERFIS DE UM OBJETO CIRCULAR. NOTE QUE HÁ A FORMAÇÃO DE UM CONJUNTO DE PIXELS FORA DO OBJETO DEVIDO ÀS CONTRIBUIÇÕES POSITIVAS DOS PERFIS A E B. ESTA OCORRÊNCIA PRODUZ UM ARTEFATO EM FORMA DE ESTRELA, COM ILUSTRADO EM (B) (BROOKS E DI CHIRO,1976).	17
FIGURA 2-7 RETROPROJEÇÃO FILTRADA. OS PERFIS FILTRADOS ($\bar{p}$) SÃO RETROPROJETADOS PARA RECONSTRUIR O OBJETO CIRCULAR ORIGINAL. O PONTO EXTERNO RECEBE CONTRIBUIÇÕES POSITIVAS DOS PERFIS A E B E DESLOCAMENTOS NEGATIVOS DOS PERFIS C E D, ELIMINANDO O EFEITO ESTRELA (BROOKS AND DI CHIRO, 1976).....	17
FIGURA 3-1 DIAGRAMA DE BLOCOS DE UM FILTRO DE KALMAN BIDIMENSIONAL PADRÃO, (BOULFELFEL, RANGAYYAN, HAHN, KLOIBER AND KUDUVALLI, 1994).	41
FIGURA 3-2 REGIÕES DE SUPORTE E SUPORTE REDUZIDA.....	42

FIGURA 3-3 TRANSIÇÃO DO FILTRO DE KALMAN DE ORDEM REDUZIDA.....	43
FIGURA 6-1 RESULTADOS DO FILTROS DE WIENER E KALMAN: (A) IMAGEM DEGRADADA; (B) IMAGEM RESTAURADA COM O FILTRO DE WIENER (c) IMAGEM RESTAURADA COM O FILTRO DE KALMAN.	57
FIGURA 6-2 RESULTADOS DA DETECÇÃO DE BORDAS UTILIZANDO O CORELDRAW NA IMAGEM 5.1: (A) IMAGEM DEGRADADA; (B) IMAGEM RESTAURADA COM O FILTRO DE WIENER (c) IMAGEM RESTAURADA COM O FILTRO DE KALMAN.....	58
FIGURA 6-3 GRÁFICO INTENSIDADE DO PIXEL PELA COLUNA DA IMAGEM.....	58
FIGURA 6-4 – IMAGENS DEGRADADAS DE UM CICLO CARDÍACO	59
FIGURA 6-5 – IMAGENS RESTAURADAS DO CICLO CARDÍACO.	60

RESUMO

Neste trabalho técnicas de restauração de imagens aplicadas à filtragem de imagens médicas foram estudadas. Considera-se uma abordagem recursiva de filtragem e suas diversas implementações em duas dimensões. A implementação utilizada neste trabalho foi a do filtro de Kalman de atualização reduzida (RUKF). Na implementação do filtro de Kalman de atualização reduzida um quarto de plano (QP) foi tomado como região de suporte e um modelo autoregressivo bidimensional (AR 2-D) foi utilizado como modelo de imagem. Os parâmetros do modelo AR 2-D e a variância do ruído foram encontrados através de uma implementação do algoritmo de Levinson para duas dimensões baseada no algoritmo de Levinson em configuração multicanal. A ordem do modelo AR 2-D foi determinada pelo critério de informação de Akaike (AIC). Para análise de resultados o filtro de Kalman de atualização reduzida foi aplicado em uma imagem planar, considerada invariante no espaço e com ruído de observação não estacionário, e os resultados comparados àqueles obtidos com o filtro de Wiener.

Palavras-chave: Restauração de imagens; modelos AR; filtro de Kalman 2-D.

ABSTRACT

In this work image restoration techniques for the filtering of medical images are studied. Emphasis is given to the recursive approach to image restoration and its different implementations are described. The implementation used in the restoration procedure is the reduced update Kalman filter (RUKF). In the implementation of the reduced update Kalman filter a quarter plane is adopted as the region of support and a 2-D autoregressive (AR) model is used as the image model. The parameters of the 2-D AR model and the variance of the driving noise are found by a 2-D implementation of the Levinson algorithm. The model order of the 2-D AR model is determined by the Akaike information criterion (AIC). For the analysis of the results, the reduced update Kalman filter is applied to a space invariant plane image with nonstationary noise. The results are compared to the results of the Wiener filter.

Keywords: Image restoration; models AR; 2-D Kalman filter.

1. INTRODUÇÃO

As imagens médicas estão sujeitas a degradações as quais são introduzidas na imagem devido a imperfeições no sistema de imagens, resultando em uma reprodução imperfeita da cena original. Cada componente no sistema de aquisição de imagem (lente, sensor, digitalizador) contribui para o borramento da imagem. A imagem é também borrada quando há movimento relativo entre o objeto e o sensor, bem como quando o sensor não está no mesmo plano focal das lentes (falta de foco).

As fontes mais comuns de ruídos são os ruídos eletrônicos ou fotoelétricos, ruídos nos grãos do filme e ruídos de quantização. Ruídos eletrônicos resultam do movimento térmico dos elétrons nos componentes eletrônicos e também da quantização durante a digitalização da imagem. Ruídos fotoelétricos devem-se à natureza estatística da luz, e ao processo fotoelétrico de conversão no sensor de imagem. Ruídos nos grãos do filme se devem à ocorrência de grãos de prata de baixa qualidade no filme que grava a imagem.

Denomina-se quantização o processo pelo qual uma imagem com tons contínuos é convertida em uma imagem com tons discretos. A redução no número de tons de cinza em imagens monocromáticas, pela diminuição do número de níveis de quantização, conduz ao fenômeno denominado falsos contornos, que é caracterizado pelo aparecimento de transições visíveis de tons de cinza em regiões com variação relativamente suave de tonalidade (Mascarenhas e Velasco, 1989).

A necessidade de restauração de imagem tem sido evidente em várias aplicações tais como: fotografia, medicina, astronomia, visão de robôs, fotografia de satélites e aplicações militares.

O objetivo da restauração de imagens é recuperar a cena original a partir da cena degradada. Técnicas de restauração de imagens são orientadas visando a modelagem de degradações, borramentos e ruídos, para a aplicação de um procedimento inverso visando reconstruir a cena original. Esta é a distinção das

técnicas de realce de imagem, que são designadas para manipular uma imagem com a finalidade de produzir melhores resultados para um observador, sem o uso de modelos particulares de degradação.

Na prática a exata restauração da cena original a partir dos dados da imagem observada pode não ser possível ainda que se conheça as características do sistema de degradação, devido ao mal condicionamento do problema de restauração de imagens e à presença de ruídos. Deste modo, métodos de restauração de imagens tentam melhorar a imagem para torná-la o mais próximo possível da cena original.

No presente trabalho é realizado um estudo sobre o filtro de Kalman, o qual é muito utilizado no processamento de sinais (1-D), e suas formas de implementação para o casos de processamento de imagens (2-D). Neste trabalho é apresentada uma implementação do filtro de Kalman de atualização reduzida feita em linguagem Fortran, a qual oferece um bom desempenho de velocidade de processamento o que a torna adequada para a implementação deste filtro o qual requer um grande tempo de processamento, pois restaura a imagem pixel a pixel. A linguagem Fortran também oferece a vantagem de não limitar o tamanho de memória para cada variável tornando possível armazenar as matrizes de covariância do erro que requerem uma grande quantidade de memória de armazenamento. A utilização do filtro de Kalman na restauração de imagens, é avaliada através da comparação do seu desempenho com o do largamente utilizado filtro de Wiener .

O trabalho é organizado como segue. No Capítulo 2 é feita uma revisão bibliográfica dos modelos de imagem e borramento e métodos de restauração. No Capítulo 3 são apresentadas formas de implementação do filtro de Kalman e no Capítulo 4 são descritos os métodos de obtenção dos parâmetros da imagem. O capítulo 5 descreve a implementação do filtro de Kalman de atualização reduzida realizada. No Capítulo 6 são mostrados os resultados, e finalmente no Capítulo 7 são apresentadas as discussões e conclusões.

2. REVISÃO BIBLIOGRÁFICA

2.1 MODELOS DE IMAGEM E OBSERVAÇÃO

Todo algoritmo de restauração de imagem é baseado em modelos que estabelecem a relação entre a entrada (imagem ideal) e a saída (imagem degradada observada) do sistema de aquisição de imagem. O sucesso da restauração de imagem em uma dada aplicação depende do quanto o modelo matemático adotado se encaixa bem nas características de entrada/saída do sistema de imagens.

Neste capítulo apresentamos um estudo sobre os modelos matemáticos de imagem e modelos matemáticos de observação. Este estudo tem como base os artigos de Sezan and Tekalp (1990) e Kaufman and Tekalp (1991).

2.1.1. MODELOS DE IMAGEM

O papel do modelo de imagem é prover uma descrição quantitativa das características da imagem requeridas por um determinado tipo de processamento.

Uma imagem digital pode ser definida como sendo uma matriz $N \times N$ de pixels. Cada pixel é caracterizado por uma intensidade de nível de cinza, a qual está ligada ao logaritmo da exposição.

Imagens podem ser representadas como uma sequência 2-D qualquer determinística ou aleatória, a qual pode ser expressa em um vetor compacto usando uma notação lexicográfica. No caso determinístico supõe-se serem membros de um espaço Hilbertiano apropriado (igual a um espaço Euclidiano com usual produto interno e norma). No caso aleatório, supõe-se que as imagens têm uma certa função distribuição de probabilidade (pdf).

Uma informação a priori sobre a imagem ideal pode ser utilizada para definir um critério de solução, ou restrições sobre a solução. Em geral cada pedaço de uma informação a priori ou restrição pode ser vista como modelo de uma certa propriedade da solução. Por exemplo, no contexto teórico de restauração de imagens, a solução pode ser restrita a um membro de um espaço Hilbertiano bem definido, tal como o conjunto de todas as imagens satisfazendo um certo critério de suavização.

No caso aleatório, modelos têm sido desenvolvidos para a pdf da imagem ideal no contexto de uma restauração de máximo a posteriori (MAP). Por exemplo, (Hunt, 1977; Trussell and Hunt, 1979), propuseram uma distribuição Gaussiana com média variante no espaço e covariância estacionária, como um modelo para a pdf da imagem. (German and German, 1984), utilizaram uma distribuição de Gibbs para modelar a pdf da imagem.

Por outro lado, se a imagem for considerada como sendo uma realização de um processo aleatório Gauss - Markov, do qual conhece-se a média e desconhece-se a autocorrelação, então ela pode ser modelada satisfatoriamente através de uma equação diferença autoregressiva (AR):

$$f(m,n) = \sum_{(k,l) \in R} a(k,l) \cdot f(m-k,n-l) + w(m,n) \quad (2.1)$$

onde f é a imagem, R é o suporte do modelo da imagem (unidimensional ou bidimensional) e $a(k,l)$ para $(k,l) \in R$ denotam os coeficientes do modelo e a região de suporte respectivamente, $w(m,n)$ denota erro de modelagem em uma distribuição Gaussiana. Os coeficientes do modelo são determinados tal que o erro de modelagem seja mínimo em termos da média quadrática.

Considerando-se a imagem sendo percorrida pixel a pixel da esquerda para direita, de cima para baixo, o valor do pixel na localização (m,n) (presente) é expresso em termos dos valores dos pixels varridos até esta localização (passado), dentro de uma região de atualização. Então como mostra a figura 2.1, uma causalidade é introduzida na representação da imagem, está ligada somente ao processo de varredura da imagem.

Figura 2-1 Ordem de varredura dos pixels (Kaufman and Tekalp, 1991).

Modelos de imagens com suporte causal são apropriados para o uso em algoritmos recursivos de restauração de imagens. O semi-plano não simétrico (NSHP) causal é o suporte causal mais usado e é amplamente utilizado no filtro de Kalman para restauração de imagens. A figura 2.2 mostra as regiões do suporte causal NSHP.

Figura 2-2 Regiões do suporte causal NSHP. (a) semi-plano superior. (b) semi-plano inferior (Marple, 1987).

Além do NSHP temos também o suporte um quarto de plano (QP), mostrado na figura 2.3.

Figura 2-3 Região do suporte causal QP para o primeiro quadrante (Marple, 1987).

Modelos AR semi-causais têm sido usados em filtros de Kalman. Além disso, modelos AR podem ser usados também no filtro de Wiener para estimação paramétrica do espectro. Note que se a imagem for não estacionária, os coeficientes do modelo são variantes no espaço. Neste caso, os coeficientes do modelo na equação (2.1) são trocados por $a(i,j,k,l)$. Alternativamente a equação (2.1) pode ser representada em notação vetorial como:

$$f(m,n) = a^T \cdot f_1(m-1,n) + w(m,n) \quad (2.2)$$

onde $f_1(m-1,n)$ é um vetor coluna consistindo dos pixels no suporte do vetor dos coeficientes do modelo.

2.1.2. MODELOS DE OBSERVAÇÃO

A degradação que ocorre durante o processo de formação e gravação da imagem pode ser dividida dentro de duas categorias gerais:

- (i) Degradações determinísticas (mas talvez desconhecidas), isto é, borramentos;
- (ii) Degradações estocásticas, isto é, ruídos.

O borramento observado e o ruído da imagem podem ser modelados como:

$$g = s(Hf) + v \quad (2.3)$$

onde g , f e v denotam os vetores representando em ordem lexicográfica a imagem observada, a imagem atual, e uma particular realização do processo de ruído aleatório aditivo, respectivamente. O operador H é o operador borramento. A componente determinística da degradação, borramento, é modelada por um mapeamento, $s(\cdot)$, da cena para o plano de imagem de coordenadas (m,n) . De forma geral este mapeamento é não linear e variante no espaço. Deve-se notar que os efeitos da não linearidade de $s(\cdot)$ têm muitas vezes sido ignorados na literatura, no desenvolvimento de algoritmos de restauração de imagens.

Considerando o operador borramento linear, o borramento pode ser invariante no ou variante no espaço. No caso de borramento invariante no espaço, H é representado por um operador convolução, onde a equação (2.3), pode ser expressa como:

$$g(m,n) = s\left(\sum_{(k,l) \in R_B} h(k,l).f(m-k, j-n)\right) + v(m,n) \quad (2.4)$$

onde h e R_B denotam o kernel e o suporte do borramento. O kernel do operador borramento é a resposta impulsional do sistema de borramento e é chamada de função espalhamento pontual (PSF, em inglês). No caso de borramento variante no espaço, H é representado por um operador superposição e a equação de observação torna-se:

$$g(m,n) = s\left(\sum_{(k,l) \in R_B(m,n)} h(m,n;k,l).f(k,l)\right) + v(m,n) \quad (2.5)$$

onde $h(m,n;k,l)$ e R_B denotam a PSF e o suporte, respectivamente, para uma localização de pixel (m,n) .

As imagens também estão sujeitas a degradações estatísticas, as quais são chamadas de ruídos.

Existem duas fontes principais de ruído em imagens digitais. São elas:

- (i) ruído do sensor;
- (ii) ruído de quantização.

A estatística do ruído do sensor depende do tipo de sensor de imagem e são usualmente sinais dependentes e possivelmente multiplicativos. No caso de restauração de imagens borradas e ruidosas, o ruído do sensor é usualmente aproximado por um processo aleatório Gaussiano aditivo e independente do sinal da imagem. Ruídos de quantização podem ser modelados por um ruído uniforme independente do sinal da imagem. Em seguida, o ruído observado $v(m,n)$, é considerado aditivo e não correlacionado com o sinal da imagem. Além disso, a maioria dos algoritmos de restauração de imagens assumem na prática que o ruído observado é um ruído Gaussiano, com média zero, para propósito de tratamento matemático. Em geral admite-se que modelos de ruídos mais sofisticados não conduzem a melhorias significativas na restauração.

Veremos agora alguns modelos simplificados de borramento.

MODELO DE BORRAMENTO LINEAR

Em um sistema de formação ideal de imagem, um ponto no plano imagem deve representar o nível de cinza de somente um ponto no plano objeto, contudo, em uma formação realista de imagem, um ponto no plano objeto é distribuído sobre uma vizinhança no plano imagem, descrito por um mapeamento $s(\cdot)$. A saída deste mapeamento $h(m,n;k,l)$ é a PSF.

Considerando que o processo de formação de imagem é linear e representa o objeto (cena), como uma combinação linear de fontes pontuais, a distribuição de densidade resultante no plano imagem é dada por:

$$g(m,n) = \sum_{(k,l) \in R_B} h(m,n;k,l) \cdot f(k,l) + v(m,n) \quad (2.6)$$

onde R_B denota o suporte do modelo de borramento. Neste modelo, a PSF varia com a localização do ponto fonte (k,l) . Então, (2.6) é conhecida como modelo de borramento linear variante no espaço.

MODELO DE BORRAMENTO LINEAR INVARIANTE NO ESPAÇO

Uma simplificação de (2.6) pode ser obtida se a formação do borramento puder ser considerada invariante no espaço. Então, (2.6) torna-se uma convolução discreta como segue:

$$g(m,n) = \sum_{(k,l) \in R_B} h(k,l) \cdot f(m-k, n-l) + v(m,n) \quad (2.7)$$

Diferente do suporte do modelo de imagem, o suporte da PSF em (2.6) e em (2.7), não necessita ser causal, e em geral todos os modelos realistas de borramento são modelados com PSF's não causais.

Considerando sistemas de imagens nos quais há conservação de energia, a conservação de energia implicará que o ponto fonte de luz deverá resultar em uma não perda de energia, isto é, no caso invariante no espaço;

$$\sum_{(k,l) \in R_B} h(k,l) = 1 \quad (2.8)$$

MODELOS DE BORRAMENTO PARAMÉTRICOS

É possível representar certos borramentos como uma função de um pequeno número de parâmetros usando a física do processo de formação do borramento. Exemplos de borramentos incluem o borramento devido ao movimento uniforme, borramento devido a falta de foco e o borramento devido a turbulências atmosféricas. Então, a PSF pode ser expressa, no caso invariante no espaço, como:

$$h(k,l) = h(k,l; \theta) \text{ para } (k,l) \in R_B(\theta) \quad (2.9)$$

onde $R_B(\theta)$ denota o suporte da PSF o qual também pode ser expresso como uma função do vetor de parâmetros θ . No caso de borramento devido ao movimento uniforme entre a câmara e a cena (Mascarenhas e Velasco, 1989), a imagem obtida $g(m,n)$ é dada por:

$$g(m,n) = \int_{-T/2}^{+T/2} f(m - m_0(t), n - n_0(t)) dt \quad (2.10)$$

onde:

T - duração da exposição;

$m_0(t)$ - componente na direção m do deslocamento;

$n_0(t)$ - componente na direção n do deslocamento.

Aplicando a transformada de Fourier, na equação (2.10), obtemos a função de transferência do sistema dada por:

$$H(u, v) = \int_{-T/2}^{+T/2} e^{-j2\pi(u m_0(t) + v n_0(t))} dt \quad (2.11)$$

O borramento devido ao movimento uniforme relativo entre a câmara e o objeto e o borramento devido à falta de foco podem ambos serem representados por um parâmetro somente, isto é, o deslocamento e o raio do disco que representa a PSF do borramento devido à falta de foco respectivamente. Esta nova simplificação da representação da PSF muitas vezes é muito útil na identificação de borramentos.

O modelo de borramento paramétrico pode ser estendido para borramentos variantes no espaço, permitindo o vetor de parâmetros variar com as coordenadas espaciais (m,n) .

MODELOS NA FORMA ESPAÇO DE ESTADO

Para usar os princípios da teoria de controle em restauração de imagem, é conveniente formular um modelo de imagem dado por (2.4), e um modelo de observação dado por (2.7), na notação estado de espaço. Para este efeito, os dados 2-D da imagem podem ser ordenados tal que a noção de passado, presente e futuro possa ser definida. Esta ordenação pode ser arbitrária, contudo, é comum ordenar os pixels em um formato de rastreamento, esquerda para direita, de cima para baixo. Os pixels passado, presente e futuro são definidos com respeito ao pixel presente (m,n) , como mostrado na figura 2.1.

Usando uma direção horizontal de recursão o estado $f(m,n)$ do pixel (m,n) consiste de pixels em ambas regiões, horizontal e vertical.

A representação espaço-estado então torna-se:

$$f(m,n) = Af(m-1,n) + Bw(m,n) \quad (2.12)$$

$$g(m,n) = Cf(m,n) + v(m,n) \quad (2.13)$$

Nestas equações, $f(m,n)$ é o vetor de estado da imagem no pixel (m,n) , $w(m,n)$ é o processo aleatório da incerteza no modelo de estado, $g(m,n)$ é a imagem borrada observada, $v(m,n)$ é o ruído de medição, e A, B e C são matrizes do sistema. A matriz de transição de estado A é esparsa consistindo primariamente de operadores de deslocamento e zeros. Supõe-se que $v(m,n)$ e $w(m,n)$ são processos de ruído Gaussiano branco com média zero, descorrelacionados e com covariâncias Q_v e Q_w , respectivamente.

Note que em (2.12) existe “uma direção de recursão”, a qual pode ser referida como “direção horizontal de recursão”. O que significa é que as recursões normalmente avançam em uma direção (m) , horizontal neste caso, até alcançarem a borda da imagem. Então ao ultrapassar a borda, o indexador horizontal (m) é zerado e o indexador vertical (n) é incrementado. As recursões então avançam normalmente

em uma direção até a próxima borda. Condições de borda convenientes são requeridas.

Por causa do número de elementos no vetor de estado ser muito grande, a representação definida por (A,B,C) é bastante difícil para identificação de sistemas e simulação. Considerando a imagem homogênea e que o borramento e filtragem são invariantes no espaço obtemos uma expressão simplificada para o filtro.

MODELOS DE EQUAÇÕES DO TIPO ENTRADA - SAÍDA

O par estado - espaço (2.12) e (2.13) não é conveniente para a identificação de parâmetros desconhecidos no modelo, desde que os sinais $f(m,n)$ não são disponíveis. Portanto, $f(m,n)$ deve ser eliminada destas equações para produzir uma equação de entrada - saída para o propósito de identificação. Para isto, combinando (2.2) e (2.7), intercambiando o somatório, e substituindo (2.7) de volta na equação resulta em:

$$g(m,n) = \sum_{(k,l) \in R} a_{k,l} g(m-k, n-l) + \sum_{(i,j) \in R_B} h_{i,j} w(m-i, n-j) + v(m,n) - \sum_{(k,l) \in R} a_{k,l} v(m-k, n-l) \quad (2.14)$$

ou equivalentemente:

$$g(m,n) = b(m,n) + v(m,n) \quad (2.15)$$

onde:

$$b(m,n) = \sum_{(k,l) \in R} a_{k,l} b(m-k, n-l) + \sum_{(i,j) \in R_B} h_{i,j} w(m-i, n-j) \quad (2.16)$$

com $b(m,n)$ denotando a imagem borrada sem ruído. As equações (2.15) e (2.16) mostram que o modelo de equação entrada - saída correspondente é um modelo 2 - D autoregressivo com média variável (ARMA).

2.2 RECONSTRUÇÃO

Denomina-se tomografia computadorizada o processo de obtenção da imagem reconstruída da seção transversal (fatia) de um corpo, utilizando-se um equipamento externo. O processo consiste na obtenção de projeções múltiplas obtidas através da passagem pelo corpo em interesse de radiações ou partículas, dentre as quais podemos citar raios γ , raios x e ondas ultra - sônicas.

Em função da densidade e número atômico do material essas radiações são atenuadas ao atravessar o corpo. As coordenadas (r,s) , descrevem o caminho de cada feixe de fóton, que é rotacionado com o mesmo ângulo da radiação transmitida como vemos na figura 2.4. Para a representação matemática das projeções o sistema de coordenadas mais conveniente é o sistema de coordenadas polares. Projeção é a integral de linha da função $f(x,y)$ que representa a função densidade do material distribuída ao longo da seção transversal, sob um certo ângulo de visão (ϕ) . Esta integral de linha é chamada de raio soma $\rho(r,\phi)$:

$$\rho(r,\phi) = \int_{r,\phi} f(x,y) ds \quad (2.17)$$

Figura 2-4 Sistemas de coordenadas (x,y) e respectiva conversão para coordenadas polares (r,ϕ) (Naime, 1994).

Se considerarmos que a radiação transmitida é constituída por fótons monoenergéticos, a intensidade do feixe transmitido será:

$$I = I_0 e^{-\int \mu(x,y) ds} \quad (2.18)$$

onde:

I - Intensidade obtida com a amostra entre a fonte e o detector;

I_0 - Intensidade obtida sem obstáculos entre a fonte e detector.

Obtém-se a função densidade que determina cada raio soma pela relação de intensidades (I/I_0), medidas no detector, substituindo f por μ , na equação (2.15).

$$\rho(r,\phi) = -\ln(I/I_0) \quad (2.19)$$

A distribuição dos coeficientes de atenuação na área da seção transversal em estudo é descrita pela função imagem. O que se obtém em cada transmissão é o raio soma dos coeficientes em todas as posições definidas pela varredura.

A matriz de contagem é montada após a varredura e a partir dela, é possível reconstruir a imagem tomográfica utilizando métodos de reconstrução.

Descrevemos a seguir alguns dos principais métodos de reconstrução .

2.2.1. MÉTODO DA RECONSTRUÇÃO ALGÉBRICA (A.R.T)

O método da reconstrução algébrica (ART, em inglês) é um método iterativo no qual a reconstrução da imagem é estimada a partir de aproximações sucessivas até que as projeções calculadas se aproximem daquelas medidas, dentro de uma certa margem de erro. Para verificar a correção da imagem calcula-se as integrais de linha e compara-as com as medidas. Depois utiliza-se cada estimativa como referência para melhorar a estimativa da reconstrução, apud Naime (1994).

2.2.2. MÉTODO DA TRANSFORMADA DE FOURIER

A transformada de Fourier é um método analítico que permite decompor uma função em uma série de componentes senoidais, onde cada uma apresenta múltiplos da frequência e fase.

A partir das componentes da função decomposta, utilizando a transformada de Fourier inversa podemos compor a função, e no caso de imagens bidimensionais esse mesmo conceito pode ser aplicado, isto é, podemos compor a imagem a partir das projeções bidimensionais em cada ângulo de projeção.

A vantagem apresentada pelo método da transformada de Fourier é a boa precisão da imagem, porém o tempo de processamento é maior que o do método da retroprojeção filtrada, uma vez que os dados são obtidos em coordenadas polares e precisam ser transformados para coordenadas retangulares.

2.2.3. MÉTODO DA RETROPROJEÇÃO FILTRADA

Os métodos analíticos são mais rápidos, pois se baseiam nas soluções matemáticas exatas das equações da imagem. Entre esses métodos se encontra o da Retroprojeção Filtrada, que é o mais usado na maioria dos tomógrafos comerciais. A figura 2.5 mostra que somente duas projeções são suficientes para a seção retangular de um material com distribuição homogênea das densidades.

Figura 2-5 (a) Dois perfis de um objeto retangular constituído de material homogêneo são mostrados graficamente. (b) Estes perfis são retroprojetados no plano x-y e superpostos formando uma imagem aproximada do objeto original (Brooks and Di Chiro, 1976).

Realiza-se a reconstrução retroprojetando-se cada perfil no plano x-y, isto é, aplica-se a magnitude de cada raio soma para todos os pontos constituintes do mesmo (figura 2.5 (b)).

Os problemas apresentados no método da retroprojeção simples se devem ao fato de cada raio soma ser aplicado a todos os pontos do raio e não somente aos pontos de alta densidade. A presença deste problema é mais acentuada quando a amostra tem áreas de alta densidade discretas, o que produzirá o artefato “estrela”.

Figura 2-6 Retroprojeção. (a) Ilustração de quatro perfis de um objeto circular. Note que há a formação de um conjunto de pixels fora do objeto devido às contribuições positivas dos perfis A e B. Esta ocorrência produz um artefato em forma de estrela, com ilustrado em (b) (Brooks e Di Chiro, 1976).

O método encontrado para eliminar estes artefatos provocados pela retroprojeção, é fazer uma filtragem apropriada dos perfis antes deles serem retroprojetados. A aplicação do filtro introduzirá componentes em fase opostas àquelas dos pontos externos, cancelando-se assim o efeito estrela na imagem.

Os três tipos de filtragem comumente aplicados aos perfis são:

- (i) Filtragem de Fourier;
- (ii) Filtragem de Radon;
- (iii) Filtragem por Convolução.

Figura 2-7 Retroprojeção filtrada. Os perfis filtrados ($\bar{p}$) são retroprojetados para reconstruir o objeto circular original. O ponto externo recebe contribuições positivas dos perfis A e B e deslocamentos negativos dos perfis C e D, eliminando o efeito estrela (Brooks and Di Chiro, 1976).

A filtragem de Fourier consiste em calcular a transformada de Fourier $S_0(w)$ das projeções, multiplicá-las por $|w|$ e retroprojetá-las sobre a imagem (Mascarenhas e Velasco, 1989).

O método da filtragem por convolução (Lewitt, 1983), devido à sua simplicidade e alta velocidade de processamento é o mais empregado na maioria dos tomógrafos comerciais.

2.2.4. PRÉ-RECONSTRUÇÃO RESTAURAÇÃO E PÓS-RECONSTRUÇÃO RESTAURAÇÃO

A reconstrução da imagem pode ser realizada antes ou após a restauração da imagem, caracterizando pré-reconstrução restauração ou pós-reconstrução restauração, respectivamente. A seguir são descritos os dois procedimentos (Boufelfel, et al., 1992).

Quando projeções planares são utilizadas para a reconstrução tomográfica, o borrimento bidimensional que existe em cada projeção torna-se tridimensional no bloco de imagens tomográficas computadorizadas com emissão de fóton simples (SPECT, em inglês). A PSF Gaussiana bidimensional formada em cada projeção planar contribui para formação de uma PSF tridimensional depois da reconstrução. Por causa dos dados da projeção planar serem filtrados com uma função rampa e interpolados no plano x-y de imagens SPECT, a PSF ao longo do eixo z (no plano entre camadas) é ligeiramente diferente da PSF no plano SPECT.

Pós-reconstrução restauração trata somente com uma camada SPECT por vez, e portanto, ignora a componente da função borrimento entre camadas. Por causa desta deficiência da pós reconstrução restauração, a pré-reconstrução restauração apresenta melhores resultados para pequenos objetos, quando a função borrimento pode ser considerada invariante no espaço.

Pré-reconstrução restauração de imagens de medicina nuclear tem desempenho melhor que a pós-reconstrução restauração se um modelo apropriado puder ser escolhido para as projeções planares, e se a imagem tiver pequenas dimensões. Pós-reconstrução restauração ignora o borrimento entre camadas o que

leva somente a uma restauração parcial da imagem SPECT. Pré reconstrução restauração necessita de um modelo espectral diferente para cada projeção planar, aumenta o tempo de processamento. As vantagens da pós reconstrução restauração são o pequeno tempo de processamento e a relativa invariância espectral da função borramento no plano SPECT, no caso de imagens de pequenas dimensões. A primeira vantagem é compensadora se um grande número de imagens SPECT são restauradas, neste caso é vantajoso aplicar um processo de restauração para completar a série de projeções de imagens. Além disso, a pré-reconstrução restauração não requer que o objeto esteja localizado no eixo de rotação da câmara, porque cada projeção pode ser restaurada com uma função transferência de modulação (MTF, em inglês) da correspondente distância entre a fonte e o colimador (SCD, em inglês).

2.3 RESTAURAÇÃO

Restauração de imagem pode ser definida como o processo de anulação da degradação da imagem baseado em um modelo matemático desta degradação. A finalidade da restauração de imagem é remover borramentos e ruídos adquiridos na aquisição da imagem. Para isso, técnicas de restauração baseiam-se nos modelos de degradação e aplica o processo inverso para recuperar a imagem. Isto envolve usualmente a formulação de um critério que produza alguma estimativa ótima do resultado desejado.

Imagens de medicina nuclear sofrem de estatística e resolução espacial pobres. A estatística pobre resulta do pequeno número de fótons que podem ser obtidos para cada imagem. O número de fótons é limitado pela baixa sensibilidade do colimador, a baixa dose de radiofármacos, o tempo pelo qual a distribuição do radiofármaco permanece constante e a atenuação interna por vários tecidos do corpo. O processo de restauração de imagem em medicina nuclear não é simples devido ao fenômeno do borramento depender da distância entre o objeto e a câmara (isto é, variante ao deslocamento). O processo de degradação de imagem poderá ser considerado invariante com o deslocamento se o objeto for pequeno quando comparado com a distância entre a fonte e o colimador.

A projeção de imagem SPECT usualmente possui estatística mais pobre que as projeções planares convencionais. Isto se deve ao tempo limitado de aquisição para cada projeção (Boufelfel, Rangayyan, Hahn and Kloiber, 1992).

Se a presença de ruídos é desconsiderada no modelo de degradação, uma técnica simples de restauração (deconvolução), pode ser obtida, na qual a resposta em frequência do filtro de correção, $H_1(u,v)$, é encontrado como sendo o inverso da resposta em frequência da PSF $H(u,v)$, isto é :

$$H_1(u,v) = [H(u,v)]^{-1} \quad (2.20)$$

O filtro inverso (2.20) pode ser eficientemente implementado no domínio da frequência usando a transformada rápida de Fourier (FFT). Contudo é muitas vezes impraticável devido a presença de ruído. Devido ao fato da inversão direta da função de transferência do borramento ter usualmente uma magnitude de altas frequências grande e ser infinita onde a função de transferência do borramento tem zeros, resulta numa excessiva amplificação no ruído do sensor para estas frequências.

Técnicas de deconvolução regularizada usam informações sobre o processo de geração da imagem tentando não carregar a função de transferência do filtro inverso para estas frequências na tentativa de limitar a amplificação do ruído durante o processo de restauração do sinal, entretanto, inevitavelmente o filtro regularizado desvia-se da exata inversão destas frequências. Regularização da inversão pode ser realizada via processamento iterativo determinístico de sinal ou via técnicas estocásticas de filtragem ótima. As técnicas de regularização determinísticas estão ligadas intimamente com a teoria de processamento de sinal e são usualmente implementadas por processos iterativos com o intuito de aliviar o problema computacional. Técnicas estocásticas por outro lado utilizam a teoria da filtragem ótima, a qual está ligada com ambos, processamento de sinal e teoria de controle. Na filtragem ótima o critério do mínimo erro médio quadrático (MMSE, em inglês) tem sido escolhido, por causa da sua simplicidade matemática.

A seguir descrevemos alguns dos principais métodos de restauração, cuja finalidade é encontrar uma solução para que a imagem satisfaça a um critério de otimização pré-determinado, como em (Sezan and Tekalp, 1990).

2.3.1. MÍNIMA NORMA

Este método consiste em aplicar o operador borramento inverso (quando este existir), à imagem observada para se obter uma estimativa. Esta estimativa difere da imagem atual por um termo adicional de erro $D^{-1}n$. Este termo depende da natureza do borramento e do ruído, e manifesta-se como uma amplificação do ruído. A filtragem inversa é extremamente sensível a ruídos. Quando a inversa do operador borramento não existir, uma das soluções é encontrada, com base no critério dos mínimos quadrados. Esta solução minimiza a norma residual de:

$$r(f) = Hf - g \quad (2.21)$$

2.3.2. MÍNIMO ERRO QUADRÁTICO MÉDIO LINEAR

O método do mínimo erro quadrático médio linear (LMMSE), encontra uma estimativa linear da imagem ideal, para a qual a média quadrática entre a imagem ideal e a imagem estimada é mínima. O operador linear atua na imagem observada para determinar uma estimativa, a qual é obtida com base em uma informação estatística de segunda ordem sobre a imagem e o processo de ruído, conhecidos a prior.

Como exemplo do método LMMSE, podemos citar o filtro de Wiener, para processo estacionários e modelo de borramento invariante no espaço, e o filtro de Kalman.

2.3.3. MÁXIMA PROBABILIDADE A POSTERIORI (MAP)

Neste método o critério é encontrar a estimativa que maximize a densidade de probabilidade da imagem atual, condicionada à observação.

Através da aplicação da regra de Bayes, a densidade a posteriori pode ser expressa em termos da densidade de probabilidade da imagem observada, condicionada à imagem ideal (densidade a priori), a imagem ideal e a imagem observada. As duas primeiras densidades são determinadas a partir do conhecimento a priori da distribuição de probabilidade da imagem atual e do processo de ruído. A densidade de probabilidade da imagem observada não afeta a maximização da densidade a posteriori.

O método MAP difere dos demais métodos discutidos por considerar as características de não linearidade do sensor de imagem. No caso de sensor linear e considerando o processo Gaussiano, a estimativa do MAP reduz-se à estimativa do LMMSE.

2.3.4. MÍNIMOS QUADRADOS COM RESTRIÇÕES

O método dos mínimos quadrados com restrições (CLS), é baseado na regularização de Tikhonov (Sezan e Tekalp, 1990), neste método a estimativa da imagem é definida com sendo membro do conjunto:

$$C_{\sigma} = \{f : ||Hf - g||^2 \leq \sigma^2\} \quad (2.22)$$

que minimiza uma função de estabilização, geralmente da forma:

$$J(f) = ||Af||^2 \quad (2.23)$$

onde o operador é escolhido de tal modo que a minimização de J , incorpore no problema alguma informação a priori a respeito da imagem ideal. Quando A é o operador Laplaciano, isto implica em uma suavização da imagem. C_σ é um conjunto de soluções que reproduzirá a observação, dentro de uma incerteza definida pelo parâmetro σ , que está relacionado com a variância do ruído. A condição para uma solução única é que o operador inverso A^{-1} exista.

2.3.5. REGULARIZAÇÃO DE MILLER

Na regularização de Miller a estimativa é definida como membro da interseção dos conjuntos C_σ dado em (2.22) e C_ϵ dado por:

$$C_\epsilon = \{x : \|Ax\|^2 \leq \epsilon^2\} \quad (2.24)$$

que minimiza o funcional

$$J(f) = \|Hf - g\|^2 + (\sigma/\epsilon) \|Af\|^2 \quad (2.25)$$

O operador A é escolhido para que C_ϵ represente alguma espécie de informação a priori sobre a imagem ideal. Os parâmetros σ e ϵ devem ser tais que a interseção não seja vazia. Como no CLS a condição para solução única é que A seja inversível.

2.3.6. MÁXIMA ENTROPIA

O método da máxima entropia (ME), maximiza a entropia da imagem sujeita à restrições de que a solução resultante deva reproduzir a observação exatamente, ou

dentro de uma faixa de incerteza tolerável definida pela estatística do ruído observado. A maximização da entropia reflete as propriedades de suavização da imagem atual. Um aspecto importante na máxima entropia é que a restrição de positividade da imagem é imposta implicitamente na solução porque a entropia é definida em termos do logaritmo natural dos pixels da imagem. O problema de restauração pelo método da máxima entropia pode ser formulado de tal forma a maximizar a entropia dada por:

$$(-\sum_{j=1}^{N^2} f_j \ln f_j) \quad (2.26)$$

com restrições dadas pelo modelo de observação.

2.3.7. ALGORITMOS ITERATIVOS COM RESTRIÇÕES

Os algoritmos iterativos baseados em restrições podem ser expressos pela equação:

$$Cf_k = f_{k+1} \quad (2.27)$$

onde C é um operador linear ou não linear.

O operador C é definido com base em restrições da imagem, tal que se uma imagem g satisfaz estas restrições então:

$$Cg = g \quad (2.28)$$

Na equação (2.28), g é um ponto fixado do operador C que pode ser tomado como uma estimativa da imagem ideal. Em geral o operador restrição (C), pode não ter um ponto fixado, isto é, as iterações podem não convergir para um único ponto.

fixado. Neste caso, condições adicionais são requeridas para garantir a convergência para um ponto fixado.

2.3.8. ALGORITMOS ESPECIALIZADOS

RESTAURAÇÃO DE IMAGENS DEGRADADAS POR BORRAMENTOS VARIANTES NO ESPAÇO

Devido ao fato da transformada de Fourier não poder ser utilizada em borramentos variantes no espaço, utilizam-se algoritmos implementados no domínio espacial. Métodos especializados são desenvolvidos para o problema de restauração variante no espaço, e estes métodos podem ser classificados dentro de três categorias diferentes.

- (i) Particionamento;
- (ii) Transformação de coordenadas;
- (iii) Abordagens direta.

No método do particionamento podemos assumir que o borramento é aproximadamente invariante no espaço para pequenas regiões da imagem. Então podemos restaurar a imagem aplicando técnicas de restauração para borramentos invariante no espaço em pequenas regiões da imagem. O método do particionamento apresenta bons resultados em muitos casos, contudo não representa a realidade.

O método da transformação de coordenadas se baseia no fato que para certos tipos de borramentos variantes no espaço, tais como no caso em que o movimento relativo entre o objeto e o sistema de imagem, não se realizar em trajetórias paralelas (Robbins and Huang, 1972), a transformação de coordenadas (CRT), pode ser aplicada à imagem observada para que o borramento nas novas coordenadas seja invariante no espaço. Então a imagem poderá ser restaurada um filtro invariante no espaço e depois transformada de volta às coordenadas originais. É importante notar que as propriedades estatísticas da imagem e do processo de ruído são afetadas pela transformação de coordenadas, como por exemplo, as características do sistema de

coordenadas espacial original não são mantidas na transformação de coordenadas. Além disso, o método da transformação de coordenadas se aplica somente a uma pequena classe de borramentos variantes no espaço.

O método das abordagens direta consiste em solucionar iterativamente a equação de observação.

RESTAURAÇÃO DE IMAGENS DEGRADADAS POR BORRAMENTOS ALEATÓRIOS

Quando a PSF varia aleatoriamente devido a turbulências ou movimento relativo aleatório entre o objeto e o sistema de gravação de imagens, há a ocorrência de borramentos aleatórios, onde o problema de identificação do borramento resume-se em identificar as propriedades estatísticas do borramento.

Uma solução simples para o problema do borramento aleatório é usar o valor esperado do processo de borramento em algoritmo de restauração.

RESTAURAÇÃO DE IMAGEM COM REDUÇÃO DE ARTEFATOS

Restauração linear e invariante no espaço, introduz artefatos à imagem restaurada (Langendijk, Biemond and Boekke, 1988; Sezan and Tekalp, 1990). Estes artefatos são classificados como:

- (i) Truncamento de borda;
- (ii) Ruído filtrado;
- (iii) Desvio do filtro;
- (iv) Artefatos de erro na PSF.

Uma análise das origens e características destes artefatos é dada em Sezan and Tekalp (1990).

O truncamento de borda ocorre devido a erros no modelo de convolução da imagem borrada nas bordas, e pode ser reduzido através de um pré-processamento das bordas da imagem observada. Artefatos de ruído filtrado ocorre devido à

filtragem do ruído pelo filtro de restauração. O desvio do filtro de restauração do filtro inverso provoca um aumento nos artefatos devido ao desvio do filtro. Artefatos de erro na PSF devem-se a erros na modelagem e identificação da PSF do borramento.

A redução dos artefatos possibilita a implementação de algoritmos não lineares e variantes no espaço.

2.4 FILTROS

A seguir apresentamos uma breve descrição a respeito dos filtros de Wiener, Meta e Kalman.

2.4.1. FILTRO DE WIENER

O desenvolvimento do filtro de Wiener é baseado na consideração de que o sistema de imagem é linear e estacionário e também na consideração de que o ruído é aditivo com variância constante e descorrelacionado do objeto.

Com base nestas considerações o filtro de Wiener é desenvolvido para encontrar a estimativa linear:

$$\hat{f}(m,n) = L[g(m,n)] \quad (2.29)$$

onde L é um operador linear que minimiza:

$$E\{(f(m,n) - \hat{f}(m,n))^2\} \quad (2.30)$$

Derivando o estimador obtemos uma expressão para o filtro de Wiener dada no domínio da frequência por:

$$W(u, v) = \frac{MTF^2(u, v)}{MTF(u, v) \left[MTF^2(u, v) + \frac{N_p(u, v)}{P_F(u, v)} \right]} \quad (2.31)$$

onde:

MTF - função transferência de modulação;

N_p - potência espectral do ruído;

P_F - potência espectral do objeto.

O filtro de Wiener é utilizado na supressão de ruídos. O primeiro termo da equação (2.31), é o filtro inverso, usado para recuperar a resolução (deconvolução) (King, Doherty and Schwinger, 1983). O segundo termo é um filtro passa - baixa que depende da magnitude da razão entre a potência espectral do ruído e a potência espectral do objeto. A razão entre a potência espectral do ruído e a potência espectral do objeto, governa a atenuação das frequências espaciais na imagem filtrada.

Se o ruído for muito pequeno ou tender a zero, isto é, $P_n \rightarrow 0$ o filtro de Wiener se reduz ao filtro inverso dado em (2.20).

O problema na implementação do filtro de Wiener consiste na determinação da potência espectral do ruído e potência espectral do objeto.

Em imagens degradadas por ruído de Poisson, a potência espectral da imagem é igual ao borramento do objeto mais a potência espectral do ruído. A potência espectral do ruído terá uma média constante com valor igual a média total dos pixels do objeto $\bar{N}$ (King, Doherty and Schwinger, 1983). Assim podemos escrever:

$$P_g(u, v) = \bar{N} + MTF^2(u, v).P_f(u, v) \quad (2.32)$$

onde:

$P_g(u,v)$ - potência espectral da imagem;

$\bar{N}$ - média total dos pixels do objeto.

2.4.2. FILTRO DE METZ

O filtro de Metz em uma dimensão é definido como:

$$M(f) = \text{MTF}(f)^{-1} [1 - (1 - \text{MTF}(f)^2)^x] \quad (2.33)$$

onde:

MTF - função de transferência de modulação;

f - frequência espacial;

x - fator que controla a faixa de atuação do filtro inverso.

O filtro de Metz é constituído do produto do filtro inverso com um filtro passa - baixa. A magnitude de x determina a frequência para a qual o filtro passa - baixa se torna dominante. O fator x é tomado como uma função da contagem da imagem, otimizado com a finalidade de encontrar o mínimo erro quadrático médio (MSE), para um conjunto de imagens bidimensionais.

2.4.3. FILTRO DE KALMAN

O filtro de Kalman é obtido recursivamente e determina a solução do LMMSE. Ele se baseia em uma representação espaço de estado do sistema de imagem (Sezan and Tekalp, 1990). Os vetores de estado são definidos a partir dos dados da imagem. Na implementação do filtro de Kalman assume-se que a imagem é um processo aleatório de Gauss-Markov, cujo modelo de imagem é representado pela equação (2.1). Em sua implementação, em primeiro lugar o filtro de Kalman utiliza um preditor das variáveis de estado, com base no estado anterior do sistema. Após a estimação dos coeficientes do modelo, esses coeficientes são utilizados no modelo de imagem para estimar o estado presente do sistema.

A seguir descrevemos o equacionamento do filtro de Kalman para uma dimensão.

FILTRO DE KALMAN UNIDIMENSIONAL

Dadas as equações dos modelos de imagem e observação:

$$f_m = a_{m-1}f_{m-1} + w_{m-1} \quad (2.34)$$

$$g_m = H_m f_m + v_m \quad (2.35)$$

e a equação de atualização:

$$\hat{f}_m(+) = k_m^1 \hat{f}_m(-) + \bar{k}_m g_m \quad (2.36)$$

onde:

$\hat{f}_m(-)$ - estimativa a prior de f_m

$\hat{f}_m(+)$ - estimativa a posteriori de f_m

O problema de otimização consiste em encontrar os valores para k_m^1 e $\bar{k}_m$ que satisfaçam o principio da ortogonalidade:

$$E\{[f_m - \hat{f}_m(+)]g_i^T\} = 0, i = 1, 2, 3, \dots, M-1 \quad (2.37)$$

$$E\{[f_m - \hat{f}_m(+)]g_m^T\} = 0 \quad (2.38)$$

Substituindo-se as equações (2.34) e (2.36) na equação (2.37), com M igual a dimensão de g temos:

$$E\{a_{m-1}f_{m-1} + w_{m-1} - k_m^1 \hat{f}_m(-) - \bar{k}_m g_m\}g_i^T = 0, i=1, \dots, M-1 \quad (2.39)$$

Substituindo o valor de g_m pela equação (2.35), podemos rescrever (2.39) como:

$$E\{a_{m-1}f_{m-1} + w_{m-1} - k_m^1 \hat{f}_m(-) - \bar{k}_m H_m f_m - \bar{k}_m v_m\}g_i^T = 0, i=1, \dots, M-1 \quad (2.40)$$

Usando o fato de que as seqüências de ruídos aleatórios w_m e v_m são descorrelacionadas, isto é:

$$E\{w_m g_i^T\} = 0, i=1, \dots, M-1$$

$$E\{v_m g_i^T\} = 0, i=1, \dots, M-1$$

e a equação (2.38), podemos reduzir a equação (2.40) à forma:

$$a_m E f_{k-1} g_i^T + E w_{k-1} g_i^T - k_k^1 E \hat{f}_k(-) g_i^T - \bar{k}_k H_k a_{k-1} E f_{k-1} g_i^T - \bar{k}_k E v_k g_i^T = 0$$

$$a_{m-1} E f_{m-1} g_i^T - k_m^1 E \hat{f}_m(-) g_i^T - \bar{k}_m H_m a_{m-1} E f_{m-1} g_i^T = 0$$

$$E\{[f_m - \bar{k}_m H_m f_m - k_m^1 f_m] - k_m^1 (\hat{f}_m(-) - f_m)\}g_i^T = 0$$

$$[I - k_m^1 - \bar{k}_m H_m] E f_m g_i^T = 0 \quad (2.41)$$

A equação (2.41) é satisfeita para qualquer f_m desde que:

$$K_m^1 = I - \bar{K}_m H_m \quad (2.42)$$

Definindo os vetores de estimativa de erros como:

$$\tilde{f}_m(+) = \hat{f}_m^{\Delta}(+) - f_m \quad (2.43)$$

$$\tilde{f}_m(-) = \hat{f}_m^{\Delta}(-) - f_m \quad (2.44)$$

$$\tilde{g}_m = g_m^{\Delta}(-) - g_m = H_m \hat{f}_m(-) - g_m \quad (2.45)$$

onde:

$\tilde{f}_m(+)$ - vetor estimativa do erro depois da atualização.

$\tilde{f}_m(-)$ - vetor estimativa do erro antes da atualização.

A equação (2.38) torna-se:

$$E\{f_m - \hat{f}_m(+)\} \hat{g}_m^T = 0 \quad (2.46)$$

Subtraindo a equação (2.35) da equação (2.46):

$$E\{f_m - \hat{f}_m(+)\} \tilde{g}_m^T = 0 \quad (2.47)$$

Substituindo f_m , $\hat{f}_m(+)$ e $\tilde{g}_m$ pelas equações (2.34), (2.36) e (2.45) respectivamente:

$$E\{a_{k-1}f_{k-1} + w_{k-1} - k_k^1 \hat{f}_k(-) - \bar{k}_k g_k\} [H_k \hat{f}_k(-) - g_k]^T = 0$$

Contudo:

$$E\{w_m g_m^T\} = E\{w_m \hat{f}_m(+)^T\} = 0$$

$$E\{a_{m-1}f_{m-1} - k_m^1 \hat{f}_m(-) - \bar{k}_m g_m\} [H_m \hat{f}_m(-) - g_m]^T = 0$$

Substituindo k_m^1 , g_m e $\hat{f}_m(-)$, e usando o fato de que $E\{\tilde{f}_m(-)v_m^T\} = 0$, podemos rescrever o último resultado como:

$$0 = E\{[a_{m-1}f_{m-1} - \hat{f}_m(-) + \bar{k}_m H_m \hat{f}_m(-) - \bar{k}_m v_m] [H_m \hat{f}_m(-) - H_m f_m - v_m]^T\}$$

$$= E\{[(f_m - \hat{f}_m(-)) - \bar{k}_m H_m (f_m - \hat{f}_m(-)) - \bar{k}_m v_m] [H_m \tilde{f}_m(-) - v_m]^T\}$$

$$= E\{[\tilde{f}_m(-) - \bar{k}_m H_m (-) \tilde{f}_m - \bar{k}_m v_m] [H_m \tilde{f}_m(-) - v_m]^T\}$$

Por definição a matriz de covariância do erro a prior é:

$$P_m(-) = E\{\tilde{f}_m(-) \tilde{f}_m^T(-)\}$$

que satisfaz a equação:

$$[I - \bar{k}_m H_m] P_m(-) H_m^T - \bar{k}_m R_m = 0$$

portanto o ganho do filtro de Kalman é expresso como:

$$\bar{k}_m = P_m(-) H_m^T [H_m P_m(-) H_m^T + R_m]^{-1} \quad (2.48)$$

Observamos na equação (2.48) que o ganho do filtro de Kalman é uma função da covariância a prior.

Por de definição a covariância a posteriori (matriz de covariância do erro após a atualização), é dada pela formula:

$$P_m(-) = E\{\tilde{f}_m(-)\tilde{f}_m^T(-)\} \quad (2.49)$$

Substituindo a equação (2.42), na equação (2.36), obtemos as equações:

$$\hat{f}_m(+) = (I - \bar{k}_m H_m) \hat{f}_m(-) + \bar{k}_m g_m$$

$$\hat{f}_m(+) = \hat{f}_m(-) + \bar{k}_m [g_m - H_m \hat{f}_m(-)] \quad (2.50)$$

Substituindo a equação (2.35) na equação (2.50) e subtraindo f_m de ambos os lados da equação obtemos:

$$\hat{f}_m(+) - f_m = \hat{f}_m(-) + \bar{k}_m H_m f_m + \bar{k}_m v_m - \bar{k}_m H_m \hat{f}_m(-) - f_m$$

$$\tilde{f}_m(+) = \tilde{f}_m(-) - \bar{k}_m H_m \tilde{f}_m(-) + \bar{k}_m v_m$$

$$\tilde{f}_m(+) = (I - \bar{k}_m H_m) \tilde{f}_m(-) + \bar{k}_m v_m \quad (2.51)$$

Substituindo a equação (2.51) na equação (2.49), e observando o fato de que $E\{\tilde{f}_m(-)v_m^T\} = 0$, obtemos:

$$P_m(+) = E\{(I - \bar{k}_m H_m) \tilde{f}_m(-) \tilde{f}_m^T(-) (I - \bar{k}_m H_m)^T + \bar{k}_m v_m v_m^T \bar{k}_m^T\}$$

$$P_m(+) = E\{[I - \bar{k}_m H_m] \tilde{f}_m(-) \tilde{f}_m^T(-) [I - \bar{k}_m H_m]^T + \bar{k}_m v_m v_m^T \bar{k}_m^T\}$$

$$P_m(+) = (I - \bar{k}_m H_m) P_m(-) (I - \bar{k}_m H_m)^T + \bar{k}_m R_m \bar{k}_m^T \quad (2.52)$$

A equação (2.52), é conhecida como “forma de Joseph”, da covariância atualizada e foi desenvolvida por P.D.Joseph. Substituindo $\bar{k}_m$ da equação (2.48), podemos expressar (2.52) da seguinte forma:

$$\begin{aligned} P_m(+) &= P_m(-) - \bar{k}_m H_m P_m(-) - P_m(-) H_m^T \bar{k}_m^T + \bar{k}_m H_m P_m(-) H_m^T \bar{k}_m^T + \bar{k}_m R_m \bar{k}_m^T \\ &= (I - \bar{k}_m H_m) P_m(-) - P_m(-) H_m^T \bar{k}_m^T + \underbrace{\bar{k}_m^T (H_m P_m(-) H_m^T + R_m)}_{P_m(-) H_m^T} \bar{k}_m^T \\ P_m(+) &= (I - \bar{k}_m H_m) P_m(-) \end{aligned} \quad (2.53)$$

Dada as equações da covariância a prior e da estimativa de estado,

$$\begin{aligned} P_m(-) &= E\{\tilde{f}_m(-) \tilde{f}_m^T(-)\} \\ \hat{f}_m(-) &= a_{m-1} \hat{f}_{m-1}(+) \end{aligned} \quad (2.54)$$

respectivamente. Subtraindo f_k de ambos os lados da equação (2.54), obtemos as equações:

$$\begin{aligned} \hat{f}_m(-) - f_m &= a_{m-1} \hat{f}_{m-1}(+) - f_m \\ \tilde{f}_m(-) &= a_{m-1} [\hat{f}_{m-1}(+) - f_{m-1}] - w_{m-1} \\ \tilde{f}_m(-) &= a_{m-1} \tilde{f}_{m-1}(+) - w_{m-1} \end{aligned}$$

multiplicando ambos os lados da equação por $\tilde{f}_m^T$ e extraíndo a expectância e usando o fato de que $E\{\tilde{f}_{m-1} w_{m-1}^T\} = 0$, obtemos o resultado:

$$P_m(-) = a_{m-1} P_{m-1}(+) a_{m-1}^T + Q_{m-1} \quad (2.55)$$

O equacionamento completo do filtro de Kalman para uma dimensão é dado na tabela 2.1.

Modelo dinâmico do sistema:
$\mathbf{f}_m = \mathbf{a}_{m-1} \mathbf{f}_{m-1} + \mathbf{w}_{m-1}$ $\mathbf{w}_m \sim N(0, \mathbf{Q}_m)$
Modelo de borramento:
$\mathbf{g}_m = \mathbf{H}_m \mathbf{f}_m + \mathbf{v}_m$ $\mathbf{v}_m \sim N(0, \mathbf{R}_m)$
Condições iniciais:
$E\{\mathbf{f}_0\} = \hat{\mathbf{f}}_0$ $E\{\tilde{\mathbf{f}}_0 \tilde{\mathbf{f}}_0^T\} = \mathbf{P}_0$
Consideração independente:
$E\{\mathbf{w}_k \mathbf{v}_j^T\} = 0 \text{ para todo } k \neq j$
Estimativa de estado (equação (2.54)):
$\hat{\mathbf{f}}_m(-) = \mathbf{a}_{m-1} \hat{\mathbf{f}}_{m-1}(+)$
Covariância do erro (equação (2.55)):
$\mathbf{P}_m(-) = \mathbf{a}_{m-1} \mathbf{P}_{m-1} \mathbf{a}_{m-1}^T + \mathbf{Q}_{m-1}$
Estimativa de estado atualizada (equação (2.50))
$\hat{\mathbf{f}}_m(+) = \hat{\mathbf{f}}_m(-) + \bar{\mathbf{k}}_m [\mathbf{g}_m - \mathbf{H}_m \hat{\mathbf{f}}_m(-)]$
Covariância do erro atualizada (equação (2.53)):
$\mathbf{P}_m(+) = (\mathbf{I} - \bar{\mathbf{k}}_m \mathbf{H}_m) \mathbf{P}_m(-)$
Ganho do filtro de Kalman (equação (2.48)):
$\bar{\mathbf{k}}_m = \mathbf{P}_m(-) \mathbf{H}_m^T [\mathbf{H}_m \mathbf{P}_m(-) \mathbf{H}_m^T + \mathbf{R}_m]^{-1}$

Tabela 2.1. Equações do filtro de Kalman

FILTRO DE KALMAN BIDIMENSIONAL

Devido ao grande sucesso do filtro de Kalman em uma dimensão tem havido muitos esforços na tentativa de estendê-lo para duas dimensões. Os problemas encontrados na implementação do filtro de Kalman 2-D estão relacionados à dificuldade em se estabelecer um modelo 2-D conveniente e à alta dimensão do vetor de estado resultante.

3. FORMAS DE IMPLEMENTAÇÃO DO FILTRO DE KALMAN BIDIMENSIONAL

3.1 FILTRO DE KALMAN 2-D ESCALAR

Uma vez que a varredura da imagem não afeta o resultado, consideramos uma imagem $N \times N$ varrida da esquerda para direita, de cima para baixo. Desta forma alguns pontos da imagem serão o passado, um ponto o presente e os pontos restantes o futuro (figura 2.1).

Comparando-se o modelo de imagem 2-D (equação (2.1)) com o modelo de imagem 1-D dado por:

$$f(m) = \sum_{k=1}^M a_k (m-k) + w(m) \quad (3.1)$$

onde:

M - Ordem do modelo

podemos notar que para cada janela ($M \times M$), o modelo 2-D utiliza (M^2) pontos para calcular um novo ponto da imagem estando, portanto, relacionado com a ordem do modelo como no caso 1-D em que são necessários M pontos para se

calcular um novo ponto. A medida que a janela se desloca através da imagem, valores prévios de f das M linhas anteriores serão necessários, portanto, será necessário memória suficiente para conter $(M \times N)$ elementos.

As equações para o filtro de Kalman 2-D na forma escalar são:

Estimação:

$$\hat{g}_b^{(m,n)}(m,n) = - \sum_k \sum_l a(k,l) \hat{g}_a^{(m-l,n)}(m-k,n-l) \quad (3.2)$$

$$R_b^{(m,n)}(m,n;k,l) = - \sum_o \sum_p a(o,p) R_a^{(m-l,n)}(m-o,n-p;k,l) \quad (3.3)$$

$$R_b^{(m,n)}(m,n;m,n) = - \sum_k \sum_l a(k,l) R_b^{(m,n)}(m,n;m-k,n-l) + \sigma_w^2 \quad (3.4)$$

Atualização:

$$K^{(m,n)}(i,j) = - \frac{\sum_k \sum_l h(k,l) R_b^{(m,n)}(m-k,n-l;m-i,n-j)}{\sum_k \sum_l \sum_i \sum_j h(k,l) h(i,j) R_b^{(m,n)}(m-k,n-l;m-i,n-j) + \sigma_v^2} \quad (3.5)$$

$$\hat{g}_a^{(m,n)}(i,j) = \hat{g}_b^{(m,n)}(i,j) + K^{(m,n)}(m-i,n-j)[g(m,n) - \sum_k \sum_l h(k,l) \hat{g}_b^{(m,n)}(m-k,n-l)] \quad (3.6)$$

$$R_a^{(m,n)}(i,j;k,l) = R_b^{(m,n)}(i,j;k,l) - K^{(m,n)}(m-i,n-j) \sum_o \sum_p h(o,p) R_b^{(m,n)}(m-o,n-p;k,l) \quad (3.7)$$

onde:

R - matriz de covariância do erro

K - matriz do ganho de Kalman

$\hat{g}$ - imagem restaurada

σ_w^2 - variância do ruído modelado como entrada

σ_v^2 - variância do ruído observado

subscrito b – antes da atualização

subscrito a - após atualização

3.2 FILTRO DE KALMAN 2-D VETORIAL

Esta formulação considera um vetor consistindo das linhas da imagem varridas em seqüência como se fossem contínuas. O filtro de Kalman vetorial 2-D, possui modelo de imagem dado pôr:

$$f(m,n) = Ff(m,n-1) + w(m,n) \quad (3.8)$$

e modelo de observação dado pôr:

$$g(m, n) = Hf(m, n) + v(m, n) \quad (3.9)$$

com:

$$f(m, n) = [f(m, n-1), \dots, f(m, 1); f(m-M+1, n), \dots, f(m-M+1, 1)]^T$$

$$w(m, n) = [w(m, n), 0, \dots, 0]^T$$

$$F = \begin{bmatrix} F_1 & 0 \\ I & 0 \end{bmatrix}$$

onde:

$$F_1 = [a(0, 1), \dots, a(m, n)]$$

I - matriz identidade

0 - vetor de zeros

A tabela 3.1 contém as fórmulas para o filtro de Kalman bidimensional na forma vetorial.

Estimação

$$\hat{\tilde{f}}(m,n) = F \hat{\tilde{f}}(m,n-1)$$

$$P^-(m,n) = F P^-(m,n-1) F^T + Q_w$$

Atualização

$$k(m,n) = P^-(m,n) + H^T [H P^-(m,n) H^T + Q_w]^{-1}$$

$$\hat{f}(m,n) = \hat{\tilde{f}}(m,n) + k(m,n)[g(m,n) - H \hat{\tilde{f}}(m,n)]$$

$$P(m,n) = [I - k(m,n) H] P^-(m,n)$$

Tabela 3.1. Equações do filtro de Kalman 2-D na forma vetorial.

A figura 3.1 mostra o diagrama de blocos de um filtro de Kalman bidimensional padrão.

Figura 3-1 Diagrama de blocos de um filtro de Kalman bidimensional padrão, (Boulfelfel, Rangayyan, Hahn, Kloiber and Kuduvalli, 1994).

3.3 FILTRO DE KALMAN DE ATUALIZAÇÃO REDUZIDA (RUKF)

O filtro de Kalman de atualização reduzida é uma aproximação do filtro de Kalman 2-D escalar (equação 3.2 a equação 3.7). No filtro de Kalman 2-D escalar todos os elementos do vetor de estado são atualizados a cada iteração. O principal conceito do filtro de Kalman de atualização reduzida é que as equações do filtro de Kalman são compostas de dois passos: Estimativa e atualização. Desde que a dimensão da imagem N é muito maior que a ordem do modelo M , pode-se reduzir o volume computacional através da redução do processo de atualização. Então atualiza-se somente os elementos dentro do vetor de estado que estejam dentro de uma certa distância do corrente pixel (m,n) . Este processo é uma boa aproximação devido ao fato de significantes atualizações estarem contidas dentro de uma região ao redor do pixel (m,n) .

A figura 3.2 mostra as regiões de suporte e suporte reduzida para um suporte causal de um quarto de plano.

Figura 3-2 Regiões de suporte e suporte reduzida.

O filtro de Kalman de atualização reduzida pode também ser implementado na forma vetorial, mas devido a natureza bidimensional do vetor de estado de ordem reduzida as equações do filtro de Kalman 1-D não podem ser aplicadas diretamente como no caso do filtro de Kalman vetorial. Os problemas na implementação do filtro de Kalman de ordem reduzida aparecem nas transições das bordas da imagem, isto é,

quando acaba a varredura de uma linha e iniciando-se no primeiro pixel da próxima linha, pois, o vetor $f(m,n)$ que corresponde a nova linha não pode ser expresso em função dos componentes do vetor $f(m,n-1)$, que contém elementos do final da linha anterior, desta forma a transição através das bordas da imagem não será suave.

A figura 3.3 mostra a transição do filtro de Kalman de ordem reduzida através das bordas da imagem.

Figura 3-3 Transição do filtro de Kalman de ordem reduzida.

4. IDENTIFICAÇÃO DOS PARÂMETROS DA IMAGEM

4.1 ALGORITMO DE LEVINSON

Utilizamos o algoritmo de Levinson para determinar os coeficientes do modelo $a(k,l)$ e a variância do ruído σ_w^2 Marple (1987). A equação de Yule-Walker para um processo AR causal 2-D é:

$$\sum_i \sum_j a(i,j) r_{ff}(k-i, l-j) = \begin{cases} \sigma_w^2 & \text{para } (k,l) = (0,0) \\ 0 & \text{para outros valores} \end{cases} \quad (4.1)$$

O somatório na equação (4.1) será feito numa região de suporte de um quarto de plano (QP). A estimativa da autocorrelação 2-D é dada por:

$$\hat{r}_{ff}(k,l) = \begin{cases} \frac{1}{(M-k)(N-l)} \sum_{m=0}^{M-1-k} \sum_{n=0}^{N-1-l} \hat{f}(m+k, n+l) \hat{f}^*(m,n) & \text{para } k \geq 0, l \geq 0 \\ \frac{1}{(M-k)(N-l)} \sum_{m=0}^{M-1-k} \sum_{n=-l}^{N-1-l} \hat{f}(m+k, n+l) \hat{f}^*(m,n) & \text{para } k \geq 0, l < 0 \\ \hat{r}_{ff}^*(-k, -l) & \text{para } k \leq 0, \text{ qualquer } l \end{cases} \quad (4.2)$$

onde MN - número de amostras de $f(m,n)$

Podemos arranjar as equações de Yule-Walker para um suporte QP dentro de duas matrizes blocos, ordenando os parâmetros do processo AR 2-D por linha ou coluna, onde os sobrescritos são:

r – ordenação por linha

c – ordenação por coluna

f – modelo forward

b – modelo backward

Ordenando por colunas tem-se:

$$\underline{R}^c \underline{a}^c_i = \underline{\rho}^c_i \quad (4.3)$$

na qual a matriz bloco:

$$\underline{R}^c = \begin{bmatrix} R^c[0] & R^c[1] & \dots & R^c[p_1] \\ R^c[-1] & R^c[0] & \dots & R^c[p_1-1] \\ \vdots & \vdots & & \vdots \\ R^c[-p_1] & R^c[-p_1+1] & \dots & R^c[0] \end{bmatrix} \quad (4.4)$$

onde P_1 é igual a $M \times M$, sendo M a ordem do modelo é definida em termos da matriz de elementos:

$$R^c[i] = \begin{bmatrix} r_{rr}[0, j] & r_{rr}[1, j] & \dots & r_{rr}[p_1, j] \\ r_{rr}[-1, j] & r_{rr}[0, j] & \dots & r_{rr}[p_1-1, j] \\ \vdots & \vdots & & \vdots \\ r_{rr}[p_1, j] & r_{rr}[p_1+1, j] & \dots & r_{rr}[0, j] \end{bmatrix} \quad (4.5)$$

e os vetores blocos

$$\underline{a}_1^c = \begin{bmatrix} a_1^c[0] \\ a_1^c[1] \\ \vdots \\ a_1^c[p_1] \end{bmatrix}, \quad \underline{\rho}_1^c = \begin{bmatrix} \rho_1^c \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

são definidos em termos do vetores de elementos

$$a_1^c[j] = \begin{bmatrix} a_1[0,j] \\ a_1[1,j] \\ \vdots \\ a_1[p_1,j] \end{bmatrix}, \quad \rho_1^c = \begin{bmatrix} \sigma_w^2 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \quad (4.6)$$

Cada $\mathbf{0}$ é um vetor coluna de p_1+1 zeros, e por definição $a_1[0,0]=1$. A matriz bloco Toeplitz $\underline{R}^c$ é composta de $(M+1) \times (M+1)$ matrizes de autocorrelação $R^c[j]$, as quais também são Toeplitz. A matriz $\underline{R}^c$ é, portanto, duplamente Toeplitz. O sobrescrito c é usado para indicar que a ordenação por coluna foi utilizada. O vetor bloco $\underline{a}_1^c$ de p_1+1 parâmetros autoregressivos, no qual o subscrito 1 indica que o primeiro quadrante foi utilizado, é composto de p_1+1 vetores $a_1^c[j]$, cada um de dimensão p_1+1 . O vetor bloco $\underline{\rho}_1^c$ é composto de zeros, exceto pelo primeiro elemento o qual é a variância do ruído σ_w^2 .

Da mesma forma que para o primeiro quadrante podemos expressar as equações para os demais quadrantes.

$$\underline{R}^c \underline{a}_2^c = \underline{\rho}_2^c \quad (4.7)$$

$$\underline{R}^c \underline{a}_3^c = \underline{\rho}_3^c \quad (4.8)$$

$$\underline{R}^c \underline{a}_4^c = \underline{\rho}_4^c \quad (4.9)$$

nas quais:

$$\underline{a}_2^c = \begin{bmatrix} a_2^c[0] \\ a_2^c[1] \\ \vdots \\ a_2^c[p_1] \end{bmatrix}, \quad \underline{\rho}_2^c = \begin{bmatrix} \rho_2^c \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \quad a_2^c[j] = \begin{bmatrix} a_2[p_1, j] \\ \vdots \\ a_2[1, j] \\ a_2[0, j] \end{bmatrix}, \quad \rho_2^c = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ \sigma_w^2 \end{bmatrix}$$

e

$$\underline{a}_3^c = \begin{bmatrix} a_3^c[p_1] \\ \vdots \\ a_3^c[1] \\ a_3^c[0] \end{bmatrix}, \quad \underline{\rho}_3^c = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ \rho_3^c \end{bmatrix}, \quad a_3^c[j] = \begin{bmatrix} a_3[p_1, j] \\ \vdots \\ a_3[1, j] \\ a_3[0, j] \end{bmatrix}, \quad \rho_3^c = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ \sigma_w^2 \end{bmatrix}$$

e

$$\underline{a}_4^c = \begin{bmatrix} a_4^c[p_1] \\ \vdots \\ a_4^c[1] \\ a_4^c[0] \end{bmatrix}, \quad \underline{\rho}_4^c = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ \rho_4^c \end{bmatrix}, \quad a_4^c[j] = \begin{bmatrix} a_4[0, j] \\ a_4[1, j] \\ \vdots \\ a_4[p_1, j] \end{bmatrix}, \quad \rho_4^c = \begin{bmatrix} \sigma_w^2 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

Deve-se observar que as matrizes $R^c[j]$ não são Hermitianas, mas satisfazem a relação:

$$R^c[-j] = (R^c[j])^H$$

Esta relação é suficiente para fazer a matriz bloco $\underline{R}^c$ Hermitiana, isto é:

$$\underline{R}^c = (\underline{R}^c)^H$$

Algoritmos para solucionar as equações de Yule-Walker 2-D QP podem ser desenvolvidos com base nas equações de Yule-Walker para a configuração

multicanal. As equações de Yule-Walker para um processo AR de ordem p_1 , com $p_1 + 1$ canais possuem a forma:

$$\begin{bmatrix} R_{ff}[0] & R_{ff}[1] \dots R_{ff}[p_1] \\ R_{ff}[-1] & R_{ff}[0] \dots R_{ff}[p_1-1] \\ \vdots & \vdots \\ R_{ff}[-p_1] & R_{ff}[-p_1+1] \dots R_{ff}[0] \end{bmatrix} \begin{bmatrix} I \\ A_{p_1}^H[1] \\ \vdots \\ A_{p_1}^H[p_1] \end{bmatrix} = \begin{bmatrix} P_{p_1}^f \\ 0 \\ \vdots \\ 0 \end{bmatrix} \quad (4.10)$$

$$\begin{bmatrix} R_{ff}[0] & R_{ff}[1] \dots R_{ff}[p_1] \\ R_{ff}[-1] & R_{ff}[0] \dots R_{ff}[p_1-1] \\ \vdots & \vdots \\ R_{ff}[-p_1] & R_{ff}[-p_1+1] \dots R_{ff}[0] \end{bmatrix} \begin{bmatrix} B_{p_1}^H[p_1] \\ \vdots \\ B_{p_1}^H[1] \\ I \end{bmatrix} = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ P_{p_1}^f \end{bmatrix} \quad (4.11)$$

com matriz de correlação multicanal dada por:

$$R_{ff}[k] = \begin{bmatrix} r_{00}[k] & \dots & r_{0p_1}[k] \\ r_{p_10}[k] & \dots & r_{p_1p_1}[k] \end{bmatrix} = R_{ff}^H[-k] \quad (4.12)$$

A matriz de correlação presente nas equações (4.10) e (4.11) é Hermitian e possui uma estrutura bloco Toeplitz. Estas propriedades também são características das equações Yule-Walker 2-D QP, o que sugere que a solução para as equações de Yule-Walker para 2-d QP e as equações de Yule-Walker para multicanal estão relacionadas. Comparando as equações (4.3) e (4.10), verificamos que há uma correspondência entre $\underline{R}^c$ e $\underline{R}_{ff}$, o que torna possível associar a correlação cruzada do multicanal $r_{ij}[k]$ com a autocorrelação 2-D $r_{ff}[k,l]$ da seguinte forma:

$$r_{ij}[k] = r_{ff}[k, i-j] \quad (4.13)$$

Comparando a equação (4.3) com a equação (4.10), fica claro que se

$$\rho_1^r = P_{p1}^f a_1^r[0] \quad (4.14)$$

então segue que

$$a_1^r[k] = A_{p1}^H[k] a_1^r[0] \quad (4.15)$$

O algoritmo de Levinson multicanal pode ser usado para computar a solução das equações de Yule-Walker 2-D resolvendo primeiro as matrizes P_f^{p1} e $A_{p1}[k]$ para $k=1$ até p_1 e então usar as equações (4.14) e (4.15) para encontrar os parâmetros autoregressivos e a variância para o processo 2-D.

O parâmetros do modelo AR 2-D para o segundo quadrante pode ser computado similarmente da solução do problema multicanal usando:

$$\rho_2^r = P_{p1}^f a_2^r[0] \quad (4.16)$$

e

$$a_2^r[k] = A_{p1}^H[k] a_2^r[0] \quad (4.17)$$

A estrutura duplamente Toeplitz da matriz de autocorrelação 2-D $R_{ff}[k]$ força a relação :

$$B_{p1}[k] = A_{p1}^H[k] \quad (4.18)$$

e a relação:

$$P_{p1}^f = P_{p1}^b \quad (4.19)$$

apesar destas relações em geral não serem válidas para o caso multicanal. Estas relações quando aplicadas ao caso 2-D simplificam o algoritmo de Levinson para multicanal como segue:

$$\Delta_{p_{l+1}} = [I \quad A_{p_l}[1] \quad \dots \quad A_{p_l}[p_l]] \begin{bmatrix} R_{ff}[p_l+1] \\ R_{ff}[p_l] \\ \vdots \\ R_{ff}[1] \end{bmatrix} \quad (4.20)$$

$$A_{p_{l+1}}[p_l+1] = -\Delta_{p_{l+1}}(P_{p_l}^f)^{-1} \quad (4.21)$$

$$A_{p_{l+1}}[k] = A_{p_l}[k] + A_{p_{l+1}}[p_l+1]A_{p_l}^H[p_l+1-k] \quad \text{para } 1 \leq k \leq p_l \quad (4.22)$$

$$P_{p_{l+1}}^f = (I - A_{p_{l+1}}[p_l+1]A_{p_{l+1}}^H[p_l+1])P_{p_l}^f \quad (4.23)$$

com condição inicial $P_0^f = R_{ff}[0]$.

No algoritmo de Levinson descrito acima o subscrito p_l representa a iteração de zero até a ordem do modelo. A cada iteração o subscrito p_l indica a ordem do modelo, portanto, o algoritmo começa calculando os parâmetros para um modelo de ordem 1 até atingir a ordem do modelo. Os passos seguidos na obtenção dos parâmetros da imagem são resumidos a seguir.

Algoritmo de Levinson

- 1 – Determinar $R_{ff}[0]$ (4.12) e escolher $P_0^f = R_{ff}[0]$. Escolher $p_l=0$.
- 2 – Determinar a matriz dos coeficientes de Levinson $A_{p_{l+1}}$ (4.21) (4.22).
- 3 – Determinar $P_{p_{l+1}}$ (4.23).
- 4- Incrementar p_l e retornar ao passo 2 até atingir a ordem do modelo da imagem.

4.2 DETERMINAÇÃO DA ORDEM DO MODELO

A determinação da ordem do modelo representa um dos problemas para se definir um modelo de imagem adequado. No presente trabalho é utilizado o critério de informação de Akaike (AIC), para a determinação da ordem do modelo de imagem.

O critério de AIC foi introduzido por Akaike (1969). Ele representa uma formulação quantitativa do princípio da parcimônia em modelos de imagens, determinando que entre modelos iguais será dada a preferência ao modelo com menor número de parâmetros.

A ordem do modelo será dada para o menor AIC definido por:

$$AIC = -2 \log(L_n) + 2\bar{k} \quad (4.24)$$

onde:

L_n - função de verossimilhança

$\bar{k}$ - número de parâmetros independentes do modelo

A função de verossimilhança depende dos parâmetros a_k do modelo, da ordem do modelo M , da variância do ruído σ_w^2 e do número de pontos N da imagem.

$$L_n(N, a_k, \sigma_w^2) = \frac{1}{(2\pi\sigma_w^2)^{N/2}} \exp\left(-\frac{1}{2\sigma_w^2} \sum_{k=1}^N E_k(a_k)\right) \quad (4.25)$$

onde:

E_k - sequência de distúrbios com média zero e variância σ_w^2 conhecida.

Denotando-se por $\hat{a}_k$ e $\hat{\sigma}_w^2$ as estimativas da máxima verossimilhança de a_k e σ_w^2 então:

$$\log(L_n(N, \hat{a}_k, \hat{\sigma}_w^2)) = -\frac{N}{2} (\log(\hat{\sigma}_w^2) + \log(2\pi - 1)) \quad (4.26)$$

Substituindo-se (4.26) em (4.24)

$$\begin{aligned} AIC &= -2\left[-\frac{N}{2}(\log(\hat{\sigma}_w^2) + \log(2\pi - 1))\right] + 2\bar{k} \\ AIC &= (\log(\hat{\sigma}_w^2) + \log(2\pi - 1)) + 2\bar{k} \end{aligned} \quad (4.27)$$

Na equação (4.26) tendo em vista que o valor de $\hat{\sigma}_w^2$ decresce com o aumento da ordem do modelo, foi introduzido o termo $2\bar{k}$ que representa um penalidade devido ao aumento do número de parâmetros do modelo. A equação (4.27) é calculada até ser encontrado o menor valor.

4.3 RUÍDO DE OBSERVAÇÃO

O processo de emissão de fótons obedece a uma distribuição de Poisson com parâmetros definidos pela fonte emissora. Além disso, é usualmente considerado que a densidade de imagem é linear com relação à intensidade do objeto. Portanto o número de emissões g em um intervalo de tempo é considerado ser uma variável aleatória cuja variância pode ser estimada com base na contagem das emissões obtidas do aparelho de aquisição de imagens. No presente trabalho a variância do ruído de observação é estimada localmente para cada janela como sendo a soma dos elementos contidos na janela $M \times M$ considerada (King, 1984):

$$\sigma_w^2(m, n) = \sum_{k=0}^M \sum_{l=0}^M g(m-k, n-l) \quad (4.28)$$

O modelo de observação considerado é da forma:

$$g = Hf + v \quad (4-29)$$

onde:

g – imagem degradada

f – imagem ideal

H – função de espalhamento

v - ruído de observação

Levando em conta que no processo de Poisson a média é igual a variância, podemos escrever:

$$E(g) = Hf = E(g^T g) \quad (4.30)$$

Portanto, temos que $E(v) = 0$ e a covariância do ruído é igual covariância de g .

5. IMPLEMENTAÇÃO DO FILTRO DE KALMAN

Nesta seção será descrita a forma de implementação do filtro de Kalman adotada neste trabalho. O filtro implementado foi o filtro de Kalman de atualização reduzida (RUKF), com um quarto de plano (QP) como região de suporte utilizando a linguagem Fortran para a sua implementação devido a sua velocidade e capacidade de alocação de memória para variáveis (não há limite de alocação), uma vez que o filtro de Kalman requer um esforço computacional elevado e uma grande quantidade de memória para armazenar as matrizes de covariância do erro. Para reduzir a necessidade de memória as variáveis relacionadas aos pixels da imagem que não estão mais sendo atualizados (fora da região de suporte), são desalocadas. Foram implementados também utilizando a linguagem Fortran o algoritmo de AIC utilizado para determinar a ordem do modelo (ordem do quarto de plano utilizado como região de suporte), e o algoritmo de Levinson que fornece os parâmetros $a(k,l)$, do modelo e σ_w^2 . A seguir é descrito os passos seguidos na restauração da imagem.

Algoritmo de restauração da imagem

1. Determinar a ordem do modelo utilizando uma implementação do critério de AIC (4.27).
2. Apresentar os arquivos da imagem degradada e da função de espalhamento na forma de vetor coluna que são depois convertidos para a forma de matriz de pontos $g(i,j)$ e $h(i,j)$, respectivamente. São adicionadas linhas na parte superior e colunas no lado esquerdo da matriz de forma que a janela com a ordem do modelo não ultrapasse os limites da matriz.
3. Considerar a imagem estacionária em uma janela com a ordem do modelo tomada no centro da imagem a qual é passada para a subrotina ACF (Anexo) que

calcula a autocorrelação 2-D da mesma (4.2). A autocorrelação da janela é passada para a subrotina LEV (Anexo), que é a implementação do algoritmo de Levinson, a qual determina os coeficientes $a(k,l)$, que são utilizados na imagem inteira, e a variância do ruído σ_w^2 .

4. Considerar as matrizes da imagem restaurada g_a e covariância do erro R_a nulas como condição inicial.
5. Estimar o pixel presente $\hat{g}^{(m,n)}(m,n)$ (3.2), utilizando seus vizinhos contidos dentro da região de suporte $\hat{g}^{(m-1,n)}(m,n)$ e os coeficientes de Levinson $a(k,l)$.
6. Estimar a matriz de covariância do erro do pixel presente $R_b^{(m,n)}(m,n,k,l)$ (3.3) e (3.4) utilizando $a(k,l)$, as matrizes de covariância do erro atualizada dos pixels vizinhos $R_a^{(m-1,n)}(m,n,k,l)$ e variância do ruído σ_w^2 .
7. Determinar o ganho do filtro de Kalman $K^{(m,n)}(i,j)$ (3.5) utilizando $R_b^{(m,n)}$, a matriz da função de espalhamento $h(k,l)$ e a variância do ruído de observação σ_v^2 .
8. Atualizar os pixels vizinhos $\hat{g}_a^{(m,n)}$ (3.6) utilizando o pixel da imagem borrada g , $K^{(m,n)}$ e h .
9. Atualizar as matrizes de covariância do erro dos pixels vizinhos $R_a^{(m,n)}$ (3.7) utilizando $R_b^{(m,n)}$, $K^{(m,n)}$ e h .
10. Restaurar o próximo pixel repetindo os passos 5 a 10. Ao restaurar o último pixel da imagem, a imagem restaurada é armazenada em um arquivo de saída e apresentada com a utilização do Matlab.

6. RESULTADOS

Neste trabalho foram estudadas várias formas de implementação do filtro de Kalman 2-D, sendo implementado o filtro de Kalman de ordem reduzida. Da mesma forma que em Boulfelfel, Rangayyan, Hahn, Kloiber and Kuduvalli (1994), um quarto de plano (QP), foi utilizado como região de suporte e um modelo autoregressivo 2-D foi utilizado como modelo de imagem. Os parâmetros $a(k,l)$ do modelo autoregressivo e a variância do ruído σ_w^2 foram obtidos através de uma implementação do algoritmo de Levinson. Na determinação da ordem do modelo foi utilizado o critério de AIC. A imagem foi considerada invariante no espaço e o ruído de observação foi modelado como uma distribuição de Poisson sendo a sua variância estimada localmente na região de suporte como sendo a soma dos elementos contidos na janela associada a esta região.

Foram adicionadas à imagem linhas na parte superior e colunas no lado esquerdo contendo zeros de modo que a janela com a ordem do modelo não ultrapassasse os limites da imagem. As janelas contendo os elementos restaurados da imagem e os elementos de covariância do erro foram tomadas inicialmente como sendo nulas, no entanto, foi assegurado um valor mínimo, de modo que o valor do ruído de observação não fosse nulo, pois, se isto ocorresse acarretaria problemas nas bordas da imagem.

Os resultados obtidos com o filtro de Kalman implementado foram comparados com os resultados de uma implementação do filtro de Wiener (Oliveira e Joaquim, 1996).

Figura 6-1 Resultados do filtros de Wiener e Kalman: (a) Imagem degradada; (b) Imagem restaurada com o filtro de Wiener (c) Imagem restaurada com o filtro de Kalman.

Pode-se observar na figura 5.1 que filtro de Kalman apresentou uma melhor resolução que o filtro de Wiener, principalmente na região central da imagem. O filtro de Wiener requer menos memória e consome um tempo computacional menor que filtro de Kalman , mas este apresenta a vantagem de não necessitar que o modelo de borramento seja invariante no espaço ou que o processo de ruído aleatório seja estacionário. A melhor resolução do filtro de Kalman quanto ao espalhamento da imagem permite que após o processamento, seja possível com auxílio de programas de detecção de bordas determinar as diferentes regiões da imagem de uma forma melhor que o filtro de Wiener. Na figura 5.2 a função contorno tipo detetor de borda do programa de desenho Coreldraw foi aplicada na figura 5.1 para determinar as regiões da imagem. Nesta figura podemos observar a melhor definição das regiões da imagem representada pela imagem tratada pelo filtro de Kalman.

Figura 6-2 Resultados da detecção de bordas utilizando o coreldraw na imagem 5.1:
 (a) Imagem degradada; (b) Imagem restaurada com o filtro de Wiener ,
 (c) Imagem restaurada com o filtro de Kalman

Podemos observar também graficamente os resultados, na figura 5.3, na qual mostra-se a intensidade dos pixels pelas colunas da imagem para uma linha escolhida.

Figura 6-3 Gráfico intensidade do pixel pela coluna da imagem.

Na figura 5.3 podemos observar que o gráfico da imagem tratada pelo filtro de Wiener apresenta uma melhor suavização mas no entanto não apresenta uma

transição entre as regiões da imagem tão satisfatória quanto a apresentada pelo filtro de Kalman.

As figuras abaixo representam as imagens degradadas e as imagens restauradas de um ciclo cardíaco.

Figura 6-4 – Imagens degradadas de um ciclo cardíaco

Figura 6-5 – Imagens restauradas do ciclo cardíaco.

7. DISCUSSÕES E CONCLUSÕES

Diversos métodos de restauração de imagens têm raízes em áreas bem desenvolvidas, tais como estimação e a solução de problemas de inversão mal condicionados e podem ser vistos como a extensão de algoritmos de processamento de sinais comumente utilizados para duas dimensões. Os métodos dos mínimos quadrados regularizados são os mais populares. A maior evolução nos métodos ocorreu com a utilização de modelos de observação e da imagem no procedimento de restauração. O filtro de Kalman encontra-se nesta categoria de métodos. Por ser um método recursivo o filtro de Kalman pode acomodar facilmente imagens de natureza não estacionárias. A sua desvantagem está no grande esforço computacional exigido devido à alta dimensão do vetor de estado.

O filtro de Kalman de atualização reduzida é uma aproximação do filtro de Kalman 2-D escalar o qual envolve dois passos: estimativa e atualização. Uma vez que a dimensão da imagem N é maior que a ordem do modelo da imagem M , o volume computacional pode ser reduzido através da redução do processo de atualização escolhendo-se uma região reduzida do suporte do modelo de imagem, sem que haja no entanto perda do desempenho do filtro, devido ao fato de que a maioria das imagens apresentam uma correlação local dos pixels, isto é, os pixels mais próximos do pixel a ser estimado contribuem de forma mais significativa para sua estimativa. Os problemas de implementação ocorrem na transição de linhas da imagem, onde há a necessidade de se adotar a condição de contorno de acrescentar zeros à imagem a fim de prover uma vizinhança aos pixels do começo da imagem. As imagens restauradas com o filtro de Kalman de atualização reduzida (RUKF), apresentam artefatos nas bordas da imagem devido principalmente ao uso de um modelo da imagem constante, o qual é inválido para a maioria das imagens. Para

reduzir os efeitos destes artefatos pode-se lançar mão de condições de contornos adequadas.

Atualmente várias novas técnicas de restauração de imagens têm sido estudadas no sentido de diminuir os artefatos nas bordas das imagens restauradas. Entre as novas técnicas de restauração de imagens podemos citar os filtros adaptativos. Os filtros adaptativos aplicam diferentes modelos para diferentes regiões da imagem. Métodos recursivos como o filtro de Kalman podem utilizar modelos adaptativos quando usam um modelo estocástico de imagem, mudando os parâmetros deste modelo para as bordas. Entre as pesquisas recentes podemos citar também os estudos na área de redes neurais e os estudos na área de identificação do borramento. A maioria das técnicas de restauração de imagens assumem que o operador convolução que representa o borramento é conhecido a priori, o que não acontece na maioria dos casos, portanto, a identificação do borramento é uma área muito importante da restauração de imagens, na qual estão sendo realizados vários trabalhos.

APÊNDICE

LISTAGEM DO PROGRAMA KALMAN E SUBROTINAS UTILIZADAS

C PROGRAMA KALMAN

```

integer*2 hour2,minute2,second2,hund2,hour3,minute3,second3
      integer*2 hour1,second1,minute1,hund1
      integer*2 hour4,minute4,hour5,minute5
      integer*4 ordem,ord,nlin,ncol
character imagem*50,espariamento*50,saida*50
      integer*4 i,j,k,istat
      real*4 re[allocatable](:,:),a5[allocatable](:,:)
      real*4 tem4[allocatable](:,:),gg[allocatable](:,:)
      real*4 ga[allocatable](:,:),
      *rpp[allocatable](:,:)
      real*4 rnn[allocatable](:,:), ab[allocatable](:,:)
      real*4 A1[allocatable](:,:),A3[allocatable](:,:)
      real*4 A4[allocatable](:,:),D[allocatable](:,:)
      real*4 TEM3[allocatable](:,:),P[allocatable](:,:)
      real*4 TEMP[allocatable](:,:),TEM[allocatable](:,:)
      real*4 DELT[allocatable](:,:),rt[allocatable](:,:)
      real*4 pfft[allocatable](:,:),aa[allocatable](:,:)
      real*4 inv[allocatable](:,:),g[allocatable](:,:)
      real*4 haux[allocatable](:,:),gb[allocatable](:,:)
      real*4 ka[allocatable](:,:),hm[allocatable](:,:)

```



```

real*4 pa[allocatable](:,:,:),pb[allocatable](:,:,:,:)
integer*4 g2[allocatable](:,:),g3[allocatable](:,:)
integer*4 h2[allocatable](:,:),h3[allocatable](:,:)
integer*4 hh[allocatable](:,:),ga2[allocatable](:,:)
real*4 erro,div,num,rho,rhov,soma,conte,rhov2,gmax
integer*4 li,col,hmax,ii,kk,ll,jj,error

```

```

call gettim(hour1,minute1,second1,hund1)

```

```

istat=1

```

```

c Entrada de dados

```

```

write(*,*) 'entre com o nome do arquivo de imagem'
read(*,10) imagem

```

```

10      format(a50)

```

```

write(*,*) 'entre com o numero de linhas da imagem'
read(*,11) nlin

```

```

11      format(i5)

```

```

write(*,*) 'entre com o numero de colunas da imagem'
read(*,12) ncol

```

```

12      format(i5)

```

```

allocate(g2(nlin*ncol,1),g3(nlin,ncol),stat=error)

```

```

if (error.ne.0) then

```

```

  stop 'falha na alocação de memória'

```

```

else

```

```

end if

```

```

write(*,*) 'entre com o nome da da funcao espriamento'
read(*,13) espariamento

```

```

13      format(a50)

```

```

write(*,*) 'entre com a ordem do modelo'

```

```

read (*,16) ordem

```

```

16      format(i5)

      write(*,*) 'entre com o nome do arquivo de saida'
      read (*,15) saida
15      format(a50)
      open (5,FILE = imagem)
      read(5,20)((g2(i,j),j=1,1),i=1,nlin*ncol)
20  format(i5)
      close(5)
      k=0
      do 21 i=1,nlin
      do 22 j=1,ncol
      k=k+1
      g3(i,j)=g2(k,1)
22      continue
21      continue
      deallocate(g2,stat=error)
      if (error.ne.0) then
      stop 'falha na alocação de memória'
      else
      end if

      allocate(h2(nlin*ncol,1),h3(nlin,ncol),stat=error)
      if (error.ne.0) then
      stop 'falha na alocação de memória'
      else
      end if
      open (5,FILE = espalhamento)
      read (5,25) ((h2(i,j),j=1,1),i=1,nlin*ncol)
25      format(i5)
      k=0

```

```

do 23 i=1,nlin
do 24 j=1,ncol
k=k+1
h3(i,j)=h2(k,1)
24      continue
23      continue

close(5)

deallocate(h2,stat=error)
if (error.ne.0) then
stop 'falha na alocação de memória'
else
end if

allocate(g(nlin+ordem-1,ncol+ordem-1),stat=error)
if (error.ne.0) then
stop 'falha na alocação de memória'
else
end if

do 60 i=ordem,nlin+ordem-1
do 61 j=ordem,ncol+ordem-1
g(i,j)= real(g3(i-ordem+1,j-ordem+1))
61      continue
60      continue

deallocate(g3,stat=error)

```

```

if (error.ne.0) then
stop 'falha na alocação de memória'
else
end if

allocate(hh(ordem,ordem),stat=error)
if (error.ne.0) then
stop 'falha na alocação de memória'
else
end if
hmax=0
do 62 i=1,nlin
do 63 j=1,ncol
if (h3(i,j).gt.hmax) then
ii=i
jj=j
hmax=h3(i,j)
else
end if
63  continue
62  continue
k=0
do 70 i=ii-(ordem-1),ii
k=k+1
l=0
do 71 j=jj-(ordem-1),jj
l=l+1
hh(k,l)=h3(i,j)
71  continue
70  continue

```



```

do 505 i=1,ordem
do 506 j=1,ordem
h3(i,j)=hh(ordem+1-i,ordem+1-j)
506 continue
505 continue

allocate(haux(ordem,ordem),stat=error)
if (error.ne.0) then
stop 'falha na alocação de memória'
else
end if
do 68 i=1,ordem
do 69 j=1,ordem
haux(i,j)=real(h3(i,j))/hmax
69 continue
68 continue
deallocate(hh,stat=error)
if (error.ne.0) then
stop 'falha na alocação de memória'
else
end if
soma=0
do 72 i=1,ordem
do 73 j=1,ordem
soma=soma+haux(i,j)
73 continue
72 continue

do 74 i=1,ordem
do 75 j=1,ordem
haux(i,j)=haux(i,j)/soma

```

75 continue

74 continue

allocate(hm(ordem,ordem),stat=error)

if (error.ne.0) then

stop 'falha na alocação de memória'

else

end if

do 510 i=1,ordem

do 511 j=1,ordem

hm(i,j)=haux(i,j)

511 continue

510 continue

deallocate(haux,h3,stat=error)

if (error.ne.0) then

stop 'falha na alocação de memória'

else

end if

allocate(gg(ordem,ordem),stat=error)

if (error.ne.0) then

stop 'falha na alocação de memória'

else

end if

gmax=0

do 3362 i=1,nlin

do 3363 j=1,ncol

if (g(i,j).gt.gmax) then

ii=i

jj=j

```

        gmax=g(i,j)
    else
    end if
3363  continue
3362  continue

    rhov2=0

    i=0
    do 500 k=ii+1-ordem,ii
    i=i+1
    j=0
    do 501 l=jj+1-ordem,jj
    j=j+1
    gg(i,j)=g(k,l)

501  continue
500  continue

    allocate(re(ordem,ordem,ordem),a5(ordem,ordem,ordem),
    *TEM4(ordem,ordem),rpp(ordem,ordem),rnn(ordem,ordem),
    *ab(ordem,ordem,ordem),A1(ordem,ordem),A3(ordem,ordem),
    *A4(ordem,ordem),D(ordem,1),TEM3(ordem,ordem),P(ordem,ordem),
    *TEMP(ordem,ordem),TEM(ordem,ordem),DELT(ordem,ordem),
    *rt(ordem,ordem,ordem),pfft(ordem-1,ordem-1),aa(ordem,1),
    *inv(ordem,ordem),stat=error)
    if (error.ne.0) then
    stop 'falha na alocação de memória'
    else
    end if

```

```

call acf(ordem,gg,re,rpp,rnn)
do 55 k=1,ordem
call xpose(ordem,re(1,1,k),rt(1,1,k))
55 continue
call lev(ordem,ordem,rt,RHO,A5,TEM3,P,A3,D,DELT,TEMP,TEM,A4,
*TEM4,ab,pfft,aa,istat,inv)

```

```

deallocate(re,a5,TEM4,rpp,rnn,ab,A1,gg,A4,D,TEM3,P,TEMP,TEM,DELT,
*rt,pfft,aa,inv,stat=error)
if (error.ne.0) then
stop 'falha na alocação de memória'
else
end if

```

```

allocate(ga(nlin+ordem-1,ncol+ordem-1),
*gb(nlin+ordem-1,ncol+ordem-1),
*pa(ordem,nlin+ordem-1,ordem,ncol+ordem-1),
*pb(ordem,nlin+ordem-1,ordem,ncol+ordem-1),
*ka(ordem,ordem),stat=error)
if (error.ne.0) then
stop 'falha na alocação de memória'
else
end if

```

```

do 1100,i=1,nlin+ordem-1
do 1101,j=1,ncol+ordem-1
ga(i,j)=0
gb(i,j)=0
1101 continue

```


1100 continue

do 1102,i=1,ordem

do 1103,j=1+conte,ncol+ordem-1

conte=j-1

do 1104,k=i,ordem

do 1105,l=1+conte,ncol+ordem-1

pa(i,j,k,l)=0

pa(k,l,i,j)=0

pb(i,j,k,l)=0

pb(k,l,i,j)=0

1105 continue

conte=0

1104 continue

1103 continue

1102 continue

a3(1,1)=0

do 100,li=ordem,ncol+ordem-1

do 1110,i=1,ordem-1

do 1111,j=1,ncol+ordem-1

do 1112,k=i,ordem-1

do 1113,l=1,ncol+ordem-1

pa(i,j,k,l)=pa(i+1,j,k+1,l)

pb(i,j,k,l)=pa(i,j,k,l)

pa(k,l,i,j)=pa(i,j,k,l)

pb(k,l,i,j)=pa(k,l,i,j)

1113 continue

1112 continue

1111 continue

1110 continue

do 1114,j=1,ncol+ordem-1

conte=j-1

do 1115,k=1,ordem

do 1116,l=conte+1,ncol+ordem-1

pa(ordem,j,k,l)=0

pa(k,l,ordem,j)=0

1116 continue

conte=0

1115 continue

1114 continue

col=0

write(*,45) li+1-10

45 format('/Linha - ',i4)

do 110,col=ordem,ncol+ordem-1

if (g(li,col).eq.0) then

goto 110

else

end if

```

soma=0
do 1010 i=1,ordem
do 1011 j=1,ordem

soma=soma+a3(i,j)*ga(li+1-i,col+1-j)
1011 continue
1010 continue
ga(li,col)=-1.0*soma

soma=0.0
do 129,k=1,ordem
do 130,l=1,ordem
soma=soma+g(li+1-k,col+1-l)
130 continue
129 continue
rhov=soma

if (rhov.le.20*ordem*ordem) then
rhov=20*ordem*ordem
else
end if

do 1012 k=1,ordem
do 1013 l=col+1-ordem,col
soma=0.0
do 1014 i=1,ordem
do 1015 j=1,ordem
soma=soma+a3(i,j)*pa(ordem+1-i,col+1-j,k,l)
1015 continue
1014 continue
pb(ordem,col,k,l)=-1.0*soma

```

```

        pb(k,l,ordem,col)=pb(ordem,col,k,l)
1013 continue
1012 continue

        soma=0.0
        do 1021 k=1,ordem
        do 1022 l=1,ordem
            soma=soma+a3(k,l)*pb(ordem,col,ordem+1-k,col+1-l)
1022 continue
1021 continue
        pb(ordem,col,ordem,col)=-1.0*soma+rho

        conte=j-(col+1-ordem)
        do 1122 k=1,ordem
        do 1123 l=1,col
            pa(ordem,col,k,l)=pb(ordem,col,k,l)
            pa(k,l,ordem,col)=pa(ordem,col,k,l)
1123 continue
1122 continue

c calculo do divisor de ka

        div=0.0
        do 1030 kk=1,ordem
        do 1031 ll=1,ordem
        do 1032 ii=1,ordem
        do 1033 jj=1,ordem
            div=div+hm(kk,ll)*hm(ii,jj)*pa(ordem+1-kk,col+1-ll,ordem+1-ii,
            *col+1-jj)

```



```

1033 continue
1032 continue
1031 continue
1030 continue
    div=div+rhov

c    calculo de ka
    do 1026 i=1,ordem
    do 1027 j=1,ordem

c    calculo do numerador de ka
    num=0
    do 1028 k=1,ordem
    do 1029 l=1,ordem
        num=num+hm(k,l)*pa(ordem+1-k,col+1-l,ordem+1-i,col+1-j)
1029 continue
1028 continue
        ka(i,j)=num/div

1027 continue
1026 continue
        erro=0
        do 1034 k=1,ordem
        do 1035 l=1,ordem
            erro=erro+hm(k,l)*ga(li+1-k,col+1-l)
1035 continue
1034 continue
            erro=g(li,col)-erro
            do 1036 i=li+1-ordem,li
            do 1037 j=col+1-ordem,col
                ga(i,j)=ga(i,j)+ka(li+1-i,col+1-j)*erro

```

```

1037 continue
1036 continue

c  calculo de pa
  do 1038 i=1,ordem
    do 1039 j=col+1-ordem,col
      conte=j-(col+1-ordem)
      do 1040 k=i,ordem
        do 1041 l=(col+1-ordem)+conte,col
          soma=0.0
          do 1042 ii=1,ordem
            do 1043 jj=1,ordem
              soma=soma+hm(ii,jj)*pb(ordem+1-ii,col+1-jj,k,l)
            1043 continue
          1042 continue
          pa(i,j,k,l)=pb(i,j,k,l)-ka(ordem+1-i,col+1-j)*soma
          pa(k,l,i,j)=pa(i,j,k,l)
        1041 continue
        conte=0
      1040 continue
    1039 continue
  1038 continue
110  continue
100  continue
  allocate(ga2(nlin*ncol,1),stat=error)
  if (error.ne.0) then
    stop 'falha na alocação de memória'
  else
    end if

k=0

```

```

do 36 i=ordem,nlin+ordem-1
do 37 j=ordem,ncol+ordem-1
k=k+1
ga2(k,1)=ga(i,j)
37 continue
36 continue
open (6,FILE = saida)
c do 36 i=1,nlin
c do 37 j=1,ncol
write (6,35) ((ga2(i,j),j=1,1),i=1,nlin*ncol)
35 format(i5)

c37 continue
c36 continue
close(6)

call gettim(hour2,minute2,second2,hund2)
hour4=hour1*3600
hour5=hour2*3600
minute4=minute1*60
minute5=minute2*60
hour3=hour4+minute4+second1
minute3=hour5+minute5+second2
second3=minute3-hour3
write (6,3335) second3
3335 format(i5)

pause'gravado'

```

```

stop
end

```

```

SUBROUTINE SHOWTIME(hour, minute, second, hund)
integer*2 hour, minute, second, hund, thour

```

```

CHARACTER*1 ap

```

```

IF (hour .GT. 12) THEN

```

```

    ap = 'p'

```

```

    thour = hour - 12

```

```

ELSE

```

```

    thour = hour

```

```

    ap = 'a'

```

```

ENDIF

```

```

WRITE (*, 9002) hour, minute, second, hund

```

```

9002 FORMAT (I2, ':', I2.2, ':', I2.2, ':', I2.2, ' ', A, 'm')

```

```

END

```

c Subrotina para calcular a inversa da matriz


```

SUBROUTINE gaussj(aa,n,np,b,m,mp,ist)
integer*4 m,mp,n,np,NMAX,ist
real*4 aa(np,np),b(np,MP)
PARAMETER (NMAX=50)
integer*4 i,icol,irow,j,k,l,ll,indx(NMAX),indxr(NMAX),ipiv(NMAX)

real*4 big,dum,pivinv

ist=1
do 11 j=1,n
  ipiv(j)=0
11 continue
do 22 i=1,n
  big=0.
  do 13 j=1,n
    if(ipiv(j).ne.1)then
      do 12 k=1,n
        if (ipiv(k).eq.0) then
          if (abs(aa(j,k)).ge.big)then
            big=abs(aa(j,k))
            irow=j
            icol=k
          endif
        else if (ipiv(k).gt.1) then
          ist=0
          goto 25
        endif
      enddo
    else if (ipiv(k).gt.1) then
      ist=0
      goto 25
    endif
  enddo
  if (big.eq.0) then
    ist=0
    goto 25
  endif
  dum=1/big
  pivinv=1/dum
  do 14 k=1,n
    aa(j,k)=aa(j,k)*dum
  enddo
  b(j)=b(j)*dum
  do 15 k=1,n
    if (k.ne.j) then
      aa(k,j)=aa(k,j)-aa(k,irow)*aa(j,irow)
    endif
  enddo
  do 16 k=1,n
    if (k.ne.j) then
      b(k)=b(k)-b(j)*aa(k,irow)
    endif
  enddo
  ipiv(j)=1
22 continue
12 continue
13 continue
14 continue
15 continue
16 continue
25 continue
enddo
endif

```

```

13  continue
    ipiv(icol)=ipiv(icol)+1
    if (irow.ne.icol) then
        do 14 l=1,n
            dum=aa(irow,l)
            aa(irow,l)=aa(icol,l)
            aa(icol,l)=dum
14  continue
        do 15 l=1,m
            dum=b(irow,l)
            b(irow,l)=b(icol,l)
            b(icol,l)=dum
15  continue
    endif
    indxr(i)=irow
    indxc(i)=icol
    if (aa(icol,icol).eq.0.) then
        ist=0
        goto 25

    endif

    pivinv=1./aa(icol,icol)
    aa(icol,icol)=1.
    do 16 l=1,n
        aa(icol,l)=aa(icol,l)*pivinv
16  continue
    do 17 l=1,m
        b(icol,l)=b(icol,l)*pivinv
17  continue

```

```

do 21 ll=1,n
  if(ll.ne.icol)then
    dum=aa(ll,icol)
    aa(ll,icol)=0.
    do 18 l=1,n
      aa(ll,l)=aa(ll,l)-aa(icol,l)*dum
18    continue
    do 19 l=1,m
      b(ll,l)=b(ll,l)-b(icol,l)*dum
19    continue
  endif
21 continue
22 continue
do 24 l=n,1,-1
  if(indxr(l).ne.indxc(l))then
    do 23 k=1,n
      dum=aa(k,indxr(l))
      aa(k,indxr(l))=aa(k,indxc(l))
      aa(k,indxc(l))=dum
23    continue
  endif
24 continue
25 return
END

```

C (C) Copr. 1986-92 Numerical Recipes Software .

SUBROUTINE MULT (M,N,Aa,Bb,Cc)

C Multiplica uma $M \times M$ matrix A por uma $M \times N$ matrix B

C colocando o resultado em uma $M \times N$ matrix C

```
integer*4 M,N
real*4 Aa(M,M),Bb(M,N),Cc(M,N),S
```

```
DO 5 J=1,N
DO 10 I=1,M
  S=0.0
  DO 20 K=1,M
    S=S+Aa(I,K)*Bb(K,J)
20  continue
  Cc(I,J)=S
10  continue
5   continue
RETURN
END
```

```
SUBROUTINE MULT2 (M,N,MM,NN,Aa,Bb,Cc)
```

C Multiplies M x M matrix A with M x N matrix B
C with result in M x N matrix C

```
integer*4 M,N,MM,NN
real*4 Aa(M,N),Bb(MM,NN),Cc(M,NN),S
```

```
DO 5 J=1,NN
DO 10 I=1,M
  S=0.0
```



```

      DO 20 K=1,MM
      S=S+Aa(I,K)*Bb(K,J)
20  continue
      Cc(I,J)=S
10  continue
5   continue

```

```

      RETURN
      END

```

```

      SUBROUTINE MULT3 (M,N,Aa,Bb,Cc)
C   Multiplies M x M matrix A with M x N matrix B
C   with result in M x N matrix C
      integer*4 M,N
      real*4 Aa(M,N),Bb,Cc(m,n)

```

```

      DO 10 I=1,M
      Cc(i,1)=Aa(I,1)*Bb
10  continue

```

```

      RETURN
      END

```

```

      SUBROUTINE ASSIGN (M,N,A,b)

```

C Coloca o conteúdo da $M \times N$ matriz A na matriz B

integer*4 m,n

real*4 A(M,N),B(M,N)

DO 10 J=1,N

DO 15 I=1,M

B(I,J)=A(I,J)

15 CONTINUE

10 CONTINUE

RETURN

END

SUBROUTINE SCALE (M,N,S,A,B)

C Multiplica a $M \times N$ matriz A pelo escalar S colocando o resultado na matriz B

real*4 S,A(M,N),B(M,N)

DO 10 J=1,N

DO 10 I=1,M

10 B(I,J)=S*A(I,J)

RETURN

END

SUBROUTINE SUB (M,N,A,B,C)

C Subtrai M x N matriz B da matriz A colocando o resultado na matriz C

integer*4 m,n

real*4 A(M,N),B(M,N),C(M,N)

DO 10 J=1,N

DO 10 I=1,M

10 C(I,J)=A(I,J)-B(I,J)

RETURN

END

SUBROUTINE ADD (M,N,A,B,C)

C Adiciona a M x N matrizes A com a B colocando o resultado na matriz C

integer*4 m,n

real*4 A(M,N),B(M,N),C(M,N)

DO 10 J=1,N

DO 10 I=1,M

10 C(I,J)=A(I,J)+B(I,J)

RETURN

END

SUBROUTINE DIV(M,N,A,B,C)

integer*4 m,n

real*4 A(M,N),C(M,N),B

DO 5 I=1,M

DO 10 J=1,N

C(I,J)=A(I,J)/B

10 CONTINUE

5 CONTINUE

RETURN

END

SUBROUTINE IDENT (M,A)

C Retorna a matriz identidade em A

real*4 A(M,M)

DO 10 J=1,M

DO 10 I=1,M

A(I,J)=(0.,0.)

IF (I.EQ. J) A(I,J)=(1.,0.)

10 CONTINUE

RETURN

END

SUBROUTINE XPOSE (M,Aa,Bb)

C Transposta conjugada da matriz A colocada na matriz B

integer*4 m

real*4 Aa(M,M),Bb(M,M)

DO 10 J=1,M

DO 10 I=1,M

10 Bb(I,J)=Aa(J,I)

RETURN

END

SUBROUTINE ZERO (M,N,A)

C Coloca zeros em uma matriz A MxN

integer*4 m,n

real*4 A(M,N)

```

DO 10 J=1,N
DO 10 I=1,M
10  A(I,J)=0.0
RETURN
END

```

```

SUBROUTINE ONE (M,N,A)
C  Constrói uma matriz A MxN constituída de 1
  real*4 A(M,N)
  DO 10 J=1,N
  DO 10 I=1,M
10  A(I,J)=1.0
  RETURN
END

```

```

SUBROUTINE acf(ior,ga,re,rp,rn)
c  Calcula a covariância da matriz ga de ordem or e armazena o resultado em
re

```

```

real*4 ga(ior,ior),rp(ior,ior),rn(ior,ior)
real*4 re(ior,ior,ior),soma,soman
integer*4 i,k,l,m,n

```

```

do 5 k=1,ior,1
do 10 l=1,ior,1

```

```

soma=0

```



```
do 15 m=1,ior+1-k,1
do 20 n=1,ior+1-l,1

soma=ga(m+k-1,n+l-1)*ga(m,n)+soma

20 continue
15 continue

rp(k,l)=soma/(ior*ior)

soman=0

do 25 m=1,ior+1-k,1
do 30 n=1,ior,1

soman=ga(m+k-1,n-l+1)*ga(m,n)+soman

30 continue
25 continue

rn(k,l)=soman/(ior*ior)

10 continue

rn(k,1)=rp(k,1)

5 continue

do 35 i=1,ior,1
```

```
do 40 k=1,ior,1
```

```
do 45 l=k,ior,1
```

```
re(k,l,i)=rp(i,l-k+1)
```

```
45 continue
```

```
40 continue
```

```
do 50 k=2,ior,1
```

```
do 55 l=1,k-1,1
```

```
re(k,l,i)=rn(i,k-l+1)
```

```
55 continue
```

```
50 continue
```

```
35 continue
```

```
return
```

```
end
```

```
SUBROUTINE lev(IP1,IP2,R,RHO,A,te3,PF,A2,C,DELTA,te1,te2,A1,  
*te4,ab,pft,aa,istat,inv)
```

```
C
```

```
C
```

```
C Calcula os parâmetros do algoritmo de Levinson para duas
```

```
real*4 R(IP1,IP1,IP1),A1(IP1,IP1),A2(IP1,IP1),te4(IP1,IP1)
```

```

real*4 A(IP1,IP1,IP1),c(IP1,1),PF(IP1,IP1),te3(IP1,IP1)
real*4 DELTA(IP1,IP1),te1(IP1,IP1),te2(IP1,IP1),S,RHO
real*4 AB(IP1,IP1,IP1),pft(ip1-1,ip1-1),aa(ip1,1),inv(ip1,ip1)
integer*4 istat
RHO=1.0

```

```

CALL ONE(IP1,IP1,A1)
CALL ONE(IP1,IP1,A2)
CALL ONE(IP1,1,C)
C
ISTAT=1
J=IP2
CALL ASSIGN (J,J,R(1,1,1),PF)
CALL IDENT(J,A(1,1,1))
CALL ASSIGN(J,J,A(1,1,1),AB(1,1,1))
M=1

```

C

C

100 CONTINUE

M=M+1

```

CALL ASSIGN (J,J,R(1,1,M),DELTA)
DO 10 K=2,M-1
    CALL MULT (J,J,A(1,1,K),R(1,1,M+1-K),te1)
10  CALL ADD (J,J,te1,DELTA,DELTA)
CALL ASSIGN (J,J,PF,te1)
CALL GAUSSJ (te1,J,J,C,1,1,ist)
if (ist.eq.0) then
    istat=0
    goto 101
end if

```

CALL ASSIGN (J,J,te1,inv)

S=-1.0

CALL SCALE (J,J,S,delta,te2)

CALL MULT (J,J,te2,te1,A(1,1,M))

CALL XPOSE (J,A(1,1,M),te1)

CALL MULT (J,J,A(1,1,M),te1,te2)

CALL IDENT (J,te1)

CALL SUB (J,J,te1,te2,te1)

CALL MULT (J,J,te1,PF,te2)

CALL ASSIGN (J,J,te2,PF)

IF (M.EQ. 2) GO TO 40

DO 30 K=2,M-1

MK=M+1-K

CALL ASSIGN (J,J,AB(1,1,K),te1)

CALL ASSIGN (J,J,AB(1,1,MK),te2)

CALL XPOSE (J,te2,te3)

CALL MULT (J,J,A(1,1,M),te3,te4)

CALL ADD (J,J,te1,te4,A(1,1,K))

30 CONTINUE

40 DO 35 K=2,M

CALL ASSIGN(J,J,A(1,1,k),AB(1,1,k))

35 CONTINUE

C

C

IF (M.LT. ip1) GO TO 100

C

```

      k=0
      do 60 i=1,ip1-1
        l=0
        k=k+1
        c(i,1)=-1.0*pf(i,ip1)
        do 65 jj=1,ip1-1
          l=l+1
          pft(k,l)=pf(i,jj)
65      continue
60      continue

      k=j-1
      CALL GAUSSJ (pft,k,k,C,1,1,ist)

      if (ist.eq.0) then
        istat=0
        goto 101
      end if

      do 70 i=1,ip1-1
        a2(i,1)=c(i,1)
70      continue
        a2(ip1,1)=1.0

        do 75 jj=1,ip1
          rho=pf(ip1,jj)*a2(jj,1)
75      continue
        do 85 i=1,ip1

```



```
      c(i,1)=a2(i,1)
85  continue
      DO 80 K=1,IP1
      CALL XPOSE (J,A(1,1,K),te1)
      CALL MULT (J,1,te1,c,aa)
      do 90 i=1,ip1
      A2(K,i)=aa(ip1+1-i,1)
90  continue
80  continue

101 RETURN
      END
```

REFERÊNCIAS BIBLIOGRÁFICAS

Akaike, H. A New Look at the Statistical Model Identification. **IEEE Transaction on Automatic Control**, v. AC-19, n.6, p.716-723, dezembro 1974.

Biemond, J.; Van Der Putten, F. G.; Woods, J. W. Identification and Restoration of Images with Symmetric Noncausal Blurs. **IEEE Transaction on Circuits and Systems**, v.35, n.4, p.385-393, abril 1988.

Boulfelfel, D.; Rangayyan, R. M.; Hahn, L. J.; Kloiber, R. Prereconstruction Restoration of Myocardial Single Photon Emission Computed Tomography Images. **IEEE Transaction on Medical Imaging**, v.11, n.3, p.336-341, setembro 1992.

Boulfelfel, D.; Rangayyan, R. M.; Hahn, L. J.; Kloiber, R.; Kuduvalli, G. R. Two-Dimensional Restoration of Single Photon Emission Computed Tomography Images Using the Kalman Filter. **IEEE Transactions on Medical Imaging**, v.13, n.1, p.102-109, março 1994.

Brooks, R. A.; Di Chiro, G. Principles of computer assisted tomography (CAT) in radiographic and radioisotopic imaging. **Physics in Medicine and Biology**, v.21, n.5, p.689-732, 1976.

Charalambous, C.; Ghaddar, F. K.; Kouris, K. Two Iterative Restoration Algorithms with Applications to Nuclear Medicine. **IEEE Transactions on Medical Imaging**, v.11, n.1, p.2-8, março 1992.

- Chellappa, R.; Kashyap, R. L. Digital Image Restoration Using Spatial Interaction Models. **IEEE Transactions on Acoustics, Speech, and Signal Processing**, v. ASSP-30, n.3, p.461-473, junho 1982.
- Cheong, P. L. C.; Morgera, S. D. Iterative Methods for Restoration Noisy Images. **IEEE Transaction on Acoustics, Speech, and Signal Processing**, v.37, n.4, p.580-585, abril 1989.
- Galatsanos, N. P.; Chin, R. T. Digital Restoration of Multichannel Images. **IEEE Transactions on Acoustics, Speech, and Signal Processing**, v.37, n.3, p.415-421, março 1989.
- Gomes, R. M.; Resende, P.; De Melo, J. C. D. Filtro de Kalman Paralelo: Aplicação em Processamento de Imagem, **10º CBA / 6º CLCA**, Rio de Janeiro, v.2, p.1132-1137, setembro de 1994.
- Gonzales, R. C.; Wintz, P. Image Restoration in **Digital Image Processing**, 2º edição, Addison-Wesley Publishing Company, p. 206-252, 1987.
- Guan, L.; Ward, R. K. Restoration of Randomly Blurred Images by the Wiener Filter. **IEEE Transactions on Acoustics, Speech, and Signal Processing**, v. 37, n.4, p.589-597, abril 1989.
- Hon, T. C.; Rangayyan, R. M.; Hahn, L. J.; Kloiber, R. Restoration of Gamma Camera-Based Nuclear Medicine Images. **IEEE Transactions on Medical Imaging**, v.8, n.4, p.354-363, dezembro 1989.
- Hunt, B. R. Digital Image Processing in **Applications of Digital Signal Processing**, Oppenheim, A. V. Ed., Prentice - Hall, p. 169-237, 1978.

- Hunt, B. R. Bayesian methods in nonlinear digital image restoration. **IEEE Trans. Comput.**, C-26(3), p.219-229, 1977.
- Kashyap, R. L.; Chellappa, R. Estimation and Choice of Neighbors in Spatial-Interaction Models of Images. **IEEE Transactions on Information Theory**, v. IT 29, n.1, janeiro 1983.
- Kaufman, H.; Tekalp, A. M. Survey of Techniques in Image Restoration. **IEEE Control Systems**, v.II, p.16-24, janeiro, 1991.
- King, A. M.; Doherty, P. W.; Schwinger, R. B. A Wiener Filter for Nuclear Medicine Images. **Med. Phys.**, 10, p.876-880, 1983.
- King, A. M.; Schwinger, R. B.; Doherty, P. W.; Penney, B. C. Two-Dimensional Filtering of SPECT Images using the Metz and Wiener Filters. **The Journal of Nuclear Medicine**, v.25, n.11, p.1235-1241, 1984.
- Koch, S.; Kaufman, H. Restoration of Spatially Varying Images Using Multiple Model Extended Kalman Filters. **IEEE Proceedings of the 32nd Conference on Decision and Control San Antonio, Texas**, p.1216-1221, dezembro 1993.
- Legendijk, R. L.; Biemond, J.; Boekee, D. E. Regularized iterative image restoration with ringing reduction. **IEEE Trans. Acoust. Speech Sig. Proc.**, p.1874-1888, 1988.
- Lewitt, R. M. Reconstruction Algorithms: Transform Methods. **IEEE Proceedings of the IEEE**, v.71, n.3, p.390-409, março 1983.
- Mallikarjuna, H. S.; Chaparro, L. F. Iterative Composite Filtering for Image Restoration. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, v.14, n.6, p.674-678, junho 1992.

Marple, S. L. Jr. Two-Dimensional Spectral Estimation in **Digital Spectral Analysis**, Prentice-Hall, p.432-461, 1987.

Mascarenhas, N. D. A.; Velasco, F. R. D. **Processamento Digital de Imagens**. 2ª edição, Kapelusz, p.4.1-5.27, 1988.

Banham R. M.; Katsaggelos K. A. Digital Image Restoration. **IEEE Signal Processing Magazine**, p. 24-41, março 1997.

Naime, J. M. Projeto e Construção de um tomógrafo Portátil para Estudos de Ciência do Solo e Plantas, em Campo. **Dissertação de mestrado em Engenharia Elétrica da Escola de Engenharia de São Carlos, USP**, p.10-16, 1994.

Oliveira, V. A; Joaquim, M. B. Implementação de filtros de Wiener para Imagens de Emissão. **Forum Nacional de Ciência e Tecnologia na Saúde**, Campos do Jordão, 13-17 outubro 1996, pp. 365-366.

Penney, B. C.; King, M. A.; Glick, S. J. Restoration of Combined Conjugate Images in SPECT: Comparison of a New Wiener Filter and the Image-Dependent Metz Filter. **IEEE Transactions on Nuclear Science**, v.37, n.2, p.707-712, abril 1990.

Rathee, S.; Koles, Z. J.; Overton, T. R. Image Restoration in Computed Tomography: The Spatially Invariant Point Spread Function. **IEEE Transactions on Medical Imaging**, v.11, n.4, p.530-538, dezembro 1992.

Robbins, G. M.; Huang, T. S. Inverse Filtering for Linear Shift-variant Imaging Systems. **Proc. IEEE**, 60(7), p.862-871, 1972.

Sezan, M. I.; Tekalp, A. M. Image Restoration and Reconstruction. **Optical Engineering**, v.29, n. 5, p.391-404, maio, 1990.

Tekalp, A. M.; Kaufman, H.; Woods, J. W. Identification of Image and Blur Parameter for the Restoration of Noncausal Blurs. **IEEE Transactions on Acoustics, Speech, and Signal Processing**, v. ASSP-34, n.4, p.963-972, agosto 1986.

Therrien, C. W.; El-Shaer, H. T. A Direct Algorithm for Computing 2-D AR Power Spectrum Estimates. **IEEE Transactions on Acoustics, Speech and Signal Processing**, v.37, n.11, p.1795-1798, novembro 1989.

Therrien, C. W.; El-Shaer, H. T. Multichannel 2-D AR Spectrum Estimation. **IEEE Transactions on Acoustics, Speech and Signal Processing**, v.37, n.11, p.1798-1801, novembro 1989.

Trussell, H. J.; Hunt, B. R. Improved methods of maximum a posteriori restoration. **IEEE Trans. Comp.**, C-27(1), p.57-62, 1979.