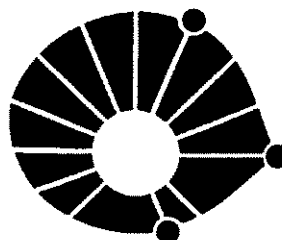


defendida por: Wilson Lobaina  
e aprovada pela Comissão  
Julgada em 13/08/00  
Luiz César Martini  
Orientador

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



**UNICAMP**

UNICAMP

BIBLIOTECA CENTRAL

SEÇÃO CIRCULANTE

# **Uma Análise dos Filtros Condicionais Seletores de Posição e sua Aplicação na Restauração de Imagens Digitais**

**Wilson Lobaina Pérez**

**Orientador: Prof. Dr. Luiz César Martini**

Banca examinadora:

Prof. Dr. Luiz César Martini  
DECOM/FEEC/UNICAMP  
Prof. Dra. Edna Lúcia Flores  
FEE/UFU – Uberlândia - MG  
Prof. Dr. João Baptista T. Yabu-uti  
DECOM/FEEC/UNICAMP  
Prof. Dr. Yuzo Iano  
DECOM/FEEC/UNICAMP

Dissertação de Mestrado apresentada à  
Faculdade de Engenharia Elétrica e de  
Computação (FEEC) da Universidade  
Estadual de Campinas (UNICAMP), como  
parte dos requisitos exigidos para obtenção  
do título de Mestre em Engenharia  
Elétrica.

**Área de Concentração:**

**Telecomunicações e Telemática**

Dissertação de Mestrado  
Campinas – SP – Brasil  
Setembro, 2000



200108038

UNIDADE <u>BC</u>	
N.º CHAMADA:	
<u>TIUNICAMP</u>	
<u>L781a</u>	
V.	Ex.
TOMBO BC/ <u>44547</u>	
PROC. <u>16-392/02</u>	
C <input type="checkbox"/>	D <input checked="" type="checkbox"/>
PREÇO <u>R\$ 11,00</u>	
DATA <u>17/05/01</u>	
N.º CPD	

CM00156297-3

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

L781a Lobaina Pérez, Wilson  
Uma análise dos filtros condicionais seletores de posição e sua aplicação na restauração de imagens digitais / Wilson Lobaina Pérez.--Campinas, SP: [s.n.], 2000.

Orientador: Luiz César Martini  
Dissertação (mestrado) - Universidade Estadual de Campinas, Faculdade de Engenharia Elétrica e de Computação.

1. Telecomunicações. 2. Processamento de sinais – Técnicas digitais. 3. Processamento de imagens – Técnicas digitais. 4. Filtros elétricos digitais. I. Martini, Luiz César. II. Universidade Estadual de Campinas. Faculdade de Engenharia Elétrica e de Computação. III. Título.

*“Ser buenos es el único modo de ser dichosos y ser cultos el único modo de ser libres”.*

*José Martí*

À minha mãe, meus irmãos, meus  
sobrinhos, meus avôs, minha namorada, e  
em especial à memória de meu pai.

## RESUMO

---

Neste trabalho é estudada e analisada uma classe de filtros chamada **Filtros Condicionais Seletores de Posição “RCRS”**. Os filtros **RCRS** são analisados dentro de uma classe mais geral denominada **Filtros Seletores de Posição “RS”** (*Rank Seletion*), os quais caraterizam-se em colocar na suas saídas uma ordem estatística de um conjunto (amostras) de observação. Muitos outros filtros propostos previamente (ex. *filtro mediano*, o *filtro mediano ponderado central “CWM”*, o *filtro de ordem estatística ponderada “WOS”*), e baseados na ordem de posição podem ser formulados através dos filtros **RS**. A única diferença de tais filtros está em qual informação é usada para decidir a ordem estatística selecionada que será colocada como saída. O tipo de informação usada nos filtros **RCRS** corresponde com o posicionamento das amostras observadas na entrada de onde surge o nome de **Filtros Condicionais Seletores de Posição** (*Rank Conditioned Rank Selection Filters*). Neste trabalho são apresentados também os resultados das simulações computacionais que ilustram o desempenho do filtro **RCRS**, assim como, a comparação com outros tipos de filtros aplicados na restauração de imagens.

### 1.1. Abstract

In this work, a class of nonlinear filters called *rank conditioned rank selection (RCRS)* filters is studied and developed. The **RCRS** filters are analyzed within the general framework of *rank selection (RS)* filters, which are filters constrained to output an order statistic from the observation set. Many filters previously proposed (e.g. median filter, *center weighted median filter “CWM”*, *weighted order statistic filter “WOS”*), rank order based can be formulated as **RS** filters. The only difference between such filters is in the information used in deciding which order statistic to output. The information used by the **RCRS** filters is the ranks of selected input samples, hence the name *rank conditioned rank selection* filters. In this work, computer simulation results that illustrate the performance of **RCRS** filters in comparison with other techniques in image restoration are presented.

## AGRADECIMENTOS

---

Agradeço muito ao Professor Dr. Luiz Cesár Martini, pela sua orientação essencial ao desenvolvimento deste trabalho, pelo ensino propiciado e pelo convívio enriquecedor nos aspectos profissional e pessoal.

À minha namorada Rosângela, pela sua ajuda incondicional, seus pensamentos sempre positivos, ainda nos momentos de puro “*stress*”, pelo seu carinho, sua compreensão e sobre tudo pela sua paciência para comigo.

À minha família: minha mãe, meus irmãos, sobrinhos e meus avôs, meus tios e primos, pela força, apoio, carinho e incentivo.

À família Lopes Góes, que aceitou-me como um filho nos momentos que mais precisava, e fez sentir-me como no meu próprio lar.

Ao meu amigo José Antonio pela sua força e as grandes contribuições neste trabalho. Sem a sua incomparável ajuda não teria sido possível a culminação deste trabalho. Também a sua esposa Diana pelo seu apoio e generosidade.

Aos meus grandes amigos Rebeca, Yudelka, Julian, Omar, Adrian, Alexei e Isabel, embora longe, para mim, muito presentes neste e em todos os momentos.

À todos os colegas e amigos do LABORATORIO DECOM, particularmente a Raquel, Rubem Toledo, Evandro, Alessandra, Fabricio, Ricardo, Raul, Carlos, e muitos outros pela sua companhia diária e companheirismo, em especial a Felipe.

Aos meus amigos Maximino, Emilia, Victor, Oselis, Vicente, Zaida, Rosabel, Franklin, Liset, Evelio, Manuel, Luis, Rigoberto, Carmencita, Lino, Jaime e ao resto dos amigos cubanos, por termos nascido em Cuba, essa terra linda banhada pelo mar Caribe, e por trazer toda nossa alegria e sabor caribenho até o Brasil, um belo país com pessoas muito solidárias e alegres.

Aos meus amigos brasileiros Rubem Milagre, David, Luciano, Silas, pela sua amizade de sempre.

À todos os amigos e funcionários de FEEC, em especial a Esther, Cida, Noemia, e todo o pessoal da CPG-FEEC, pela sua ajuda, bons conselhos e suas orientações sempre precisas.

Ao CNPq (Conselho Nacional de Desenvolvimento Científico e Tecnológico) pelo seu suporte financeiro, através da bolsa de estudos que permitiu que eu pudesse me dedicar em tempo integral no desenvolvimento deste trabalho.

# ÍNDICE

---

<b>CAPÍTULO 1. INTRODUÇÃO .....</b>	<b>1</b>
1.1. PREFÁCIO .....	1
1.2. ORGANIZAÇÃO DO TRABALHO .....	2
<b>CAPÍTULO 2. IMAGEM DIGITAL E TÉCNICAS BÁSICAS DE               PROCESSAMENTO. FUNDAMENTOS. ....</b>	<b>4</b>
2.1. REPRESENTAÇÃO DE IMAGENS DIGITAIS .....	4
2.2. UM MODELO SIMPLES DE IMAGEM .....	5
2.3. AMOSTRAGEM E QUANTIZAÇÃO .....	7
2.4. ALGUNS RELACIONAMENTOS BÁSICOS ENTRE <i>PIXELS</i> .....	9
2.4.1. <i>Definições</i> .....	9
2.4.2. <i>Os vizinhos de um pixel</i> .....	10
2.4.3. <i>Conectividade</i> .....	10
2.4.4. <i>Operações aritméticas</i> .....	11
2.5. TÉCNICAS BÁSICAS DE PROCESSAMENTO .....	13
2.5.1. <i>Fundamentos</i> .....	13
2.5.2. <i>Técnicas de Processamento baseadas em Pontos da Imagem</i> .....	13
2.5.3. <i>Técnicas de Processamento por Área ou Região da Imagem</i> .....	14
2.5.4. <i>Filtragem Espacial</i> .....	15
2.5.5. <i>Filtros espaciais lineares</i> .....	16
2.5.6. <i>Filtros Espaciais Não Lineares</i> .....	22
2.5.6.1. <i>Filtro condicional de média aritmética</i> .....	23
2.5.6.2. <i>Filtro mediano</i> .....	24
2.5.6.3. <i>Filtros Seletores de Posição (RS)</i> .....	25
2.6. CONSIDERAÇÕES FINAIS SOBRE O CAPÍTULO .....	27
<b>CAPÍTULO 3. FILTRO CONDICIONAL SELETOR DE POSIÇÃO “RCRS”. ....</b>	<b>28</b>
3.1. INTRODUÇÃO .....	28
3.2. ESTRUTURA DOS FILTROS SELETORES DE POSIÇÃO .....	29
3.3. DEFINIÇÃO DO FILTRO CONDICIONAL SELETOR DE POSIÇÃO (RCRS) .....	33
3.4. RELAÇÃO ENTRE OS FILTROS RCRS E OUTRAS CLASSES DE FILTROS .....	37
3.5. OTIMIZAÇÃO .....	40
3.6. CONSIDERAÇÕES FINAIS SOBRE O CAPÍTULO .....	48
<b>CAPÍTULO 4. RESULTADOS EXPERIMENTAIS .....</b>	<b>49</b>
4.1. ANÁLISE QUANTITATIVA. ....	49
4.2. ANÁLISE SUBJETIVA .....	54

4.3. CONSIDERAÇÕES FINAIS DO CAPÍTULO .....	59
<b>CAPÍTULO 5. CONCLUSÕES E TRABALHOS FUTUROS .....</b>	<b>60</b>
<b>REFERÊNCIAS BIBLIOGRÁFICAS .....</b>	<b>62</b>
<b>ANEXO 1 – TAMANHO DA JANELA E RELAÇÃO SINAL /RUÍDO .....</b>	<b>65</b>
<b>ANEXO 2 – FONTE DE DEGRADAÇÃO DE IMAGENS .....</b>	<b>67</b>
<b>ANEXO 3 – FUNÇÃO “IMNOISE” DO MATLAB UTILIZADA PARA CONTAMINAR A IMAGEM COM VÁRIOS TIPOS DE RUÍDOS .....</b>	<b>70</b>
<b>ANEXO 4 – EXEMPLO DIDÁTICO DO PROCESSO DE FILTRAGEM UTILIZANDO O FILTRO RCRS .....</b>	<b>71</b>
PROCESSO DE TREINAMENTO .....	73
PROCESSO DE FILTRAGEM .....	76



## LISTA DE FIGURAS

---

<b>Figura 2.1.</b> Convenção dos eixos para representação de imagens digitais. _____	4
<b>Figura 2.2.</b> <i>Pixels</i> adjacentes: (a) adjacentes por fronteira ; (b) adjacentes por esquina. ____	9
<b>Figura 2.3.</b> Processo de convolução utilizando uma máscara $3 \times 3$ . _____	17
<b>Figura 2.4.</b> Três máscaras que permitem a filtragem passa – baixas. _____	19
<b>Figura 2.5.</b> Processamento de uma imagem com um filtro <b>passa–baixas</b> : a) imagem original, b) imagem contaminada por ruído, c) imagem processada. _____	19
<b>Figura 2.6.</b> Três máscaras utilizadas como filtros passa – altas. _____	20
<b>Figura 2.7.</b> Processamento de uma imagem por um filtro passa–altas: a) imagem original, b) imagem resultante logo depois de ser processada com a máscara da Figura 2.6a. _____	20
<b>Figura 2.8.</b> (a) Seções transversais dos filtros simétricos no domínio da frequência. (b) Seções transversais dos filtros no domínio espacial. _____	21
<b>Figura 2.9.</b> Uma janela $3 \times 3$ com coeficientes arbitrários (pesos) $w_i$ . _____	21
<b>Figura 2.10.</b> Exemplo de um algoritmo de suavização utilizando um filtro condicional de média aritmética. _____	23
<b>Figura 2.11.</b> Ilustração do procedimento utilizado pelo filtro mediano. _____	24
<b>Figura 2.12.</b> (a) Imagem original, (b) Imagem contaminada com ruído impulsivo o filtro mediano, (c) Imagem depois de passar pelo filtro mediano $5 \times 5$ _____	25
<b>Figura 3.1.</b> Diagrama de blocos da estrutura do filtro seletor de posição _____	31
<b>Figura 3.2.</b> Função $S(r_\delta)$ do filtro <b>RCM</b> com parâmetro $k$ . _____	35
<b>Figura 3.3.</b> Função $S(r_\delta)$ do filtro <b>CWM</b> com parâmetro $k$ . _____	37
<b>Figura 3.4.</b> Função Ótima $S_{opt}(r_\delta)$ para um filtro <b>RCRS</b> de janela $9 \times 9$ treinado com a imagem “Elaine” contaminada com ruído impulsivo. _____	45
<b>Figura 3.5.</b> Função ótima $S_{opt}(r_\delta)$ para um filtro <b>RCRS</b> de janela $9 \times 9$ treinado com a imagem “Elaine” contaminada com ruído Gaussiano aditivo. _____	46
<b>Figura 3.6.</b> Função ótima $S_{opt}(r_\delta)$ para um filtro <b>RCRS</b> de janela $9 \times 9$ treinado com a imagem “Elaine” contaminada com ruído impulsivo. _____	47

<b>Figura 4.1.</b> (a) Imagem original da “Elaine”(teste) de tamanho 512 x 512 representada com 8 bits/pixel, (b) Imagem do “Boat (treino) de tamanho 512 x 512 com 8 bits/pixel, (c) Imagem da “Elaine” corrompida com ruído impulsivo com densidade $D = 0,20$ .	50
<b>Figura 4.2.</b> O MAE dos filtros <i>RCRS(1,2,3)</i> , <i>mediano</i> , <i>CWM</i> , <i>SWOS</i> , <i>RCM</i> e <i>Wiener</i> , filtrando a imagem “Elaine” corrompida com ruído impulsivo de densidade $D$ . (a) Resultado do erro <i>MEA</i> utilizando a janela 3x3, (b) Resultado do erro <i>MEA</i> utilizando a janela 5x5.	51
<b>Figura 4.3.</b> MAE dos filtros <i>RCRS</i> de primeira ( <i>RCRS1</i> ) e segunda ordem ( <i>RCRS2</i> ), filtrando as imagens “Elaine” e “Boat” corrompidas com ruído impulsivo.	52
<b>Figura 4.4.</b> MAE dos filtros <i>RCRS</i> , <i>mediano</i> , <i>CWM</i> , <i>SWOS</i> , <i>RCM</i> ao filtrar a imagem “Elaine” corrompida com ruído Gaussiano branco, aditivo, utilizando a janela de observação com janela 7x7.	53
<b>Figura 4.5.</b> Imagem original de “Elaine”.	54
<b>Figura 4.6.</b> Imagem de “Elaine” corrompida por ruído impulsivo do tipo “sal e pimenta” com densidade de ruído $D = 0,20$ .	55
<b>Figura 4.7.</b> Imagem de “Elaine” restaurada pelo filtro <i>Wiener2</i> com janela de tamanho 5x5, $IRSN = 8.4\text{ dB}$ .	55
<b>Figura 4.8.</b> Imagem de “Elaine” restaurada pelo filtro <i>CWM</i> com janela de tamanho 5x5, $k=6$ , $IRSN = 13.7\text{ dB}$ .	56
<b>Figura 4.9.</b> Imagem de “Elaine” restaurada pelo filtro <i>SWOS</i> utilizando uma janela de tamanho 5x5 e com $k = 6$ , $IRSN = 11.6\text{ dB}$ .	57
<b>Figura 4.10.</b> Imagem de “Elaine” restaurada pelo filtro <i>mediano</i> utilizando uma janela de tamanho 5x5, $IRSN = 16.4\text{ dB}$ .	57
<b>Figura 4.11.</b> Imagem de “Elaine” restaurada pelo filtro <i>RCRS</i> de primeira ordem utilizando uma janela de tamanho 5x5, $IRSN = 19.5\text{ dB}$ .	58
<b>Figura 4.12.</b> Imagem “Elaine” restaurada com o filtro <i>RCRS2</i> de segunda ordem, utilizando uma janela de tamanho 5x5, $IRSN = 19.9\text{ dB}$ .	59
<b>Figura A4.1.</b> (a) Imagem original da “Elaine” 512x512 pixels, (b) Imagem da “Elaine” corrompida com ruído do tipo “sal e pimenta” com densidade $D = 0,20$	71

- Figura A4.2.** Sub-região da imagem original “Elaine” e a submatriz “IO” com os valores de níveis de cinza dos *pixels* da sub-região analisada. \_\_\_\_\_ 71
- Figura A4.3.** Sub-região da imagem de “Elaine” contaminada com ruído e a submatriz “IR” com os valores de níveis de cinza dos *pixels* da sub-região contaminada. \_\_ 72
- Figura A4.4.** Região da imagem de “Elaine” (a) Região corrompida com ruído, (b) Região filtrada com um filtro *RCRS* de primeira ordem com janela de tamanho  $3 \times 1$ . \_\_\_\_ 72

## LISTA DE TABELAS

---

<b>Tabela A1.1.</b> <i>SNR</i> para janelas de vários tamanhos. _____	66
<b>Tabela A1.2.</b> Tamanhos de janelas recomendáveis para + 30 dB de ganho em regiões planas, assumindo $P_s = 128^2$ . _____	66
<b>Tabela A4.1.</b> Vetores observação “ <b>x</b> ” e vetores ordenados “ <b>x<sup>r</sup></b> ”. _____	73
<b>Tabela A4.2.</b> Vetores Diferença <b>P(m)</b> _____	74
<b>Tabela A4.3.</b> Vetores Erro Cumulativo “ <b>R</b> ” _____	75
<b>Tabela A4.4.</b> Valores do parâmetro <i>k</i> _____	76
<b>Tabela A4.5.</b> Submatrizes das sub-regiões da imagem original “ <b>IO</b> ”, imagem corrompida com ruído “ <b>IR</b> ” e da imagem restaurada “ <b>IM</b> ” respectivamente. _____	77

# CAPÍTULO 1. INTRODUÇÃO

---

## 1.1. Prefácio

O desenvolvimento de métodos de filtragem **não lineares** para o processamento de imagens tem alcançado significativos avanços nestes últimos anos, devido que os mesmos constituem uma solução efetiva onde não é adequado a utilização de técnicas lineares. [Arce, 1994, Barner, 1998]. Os filtros baseados em ordem estatística (*order statistics*), [Lin, 1996], conhecidos também como filtros seletores de posição “**RS**” (*rank selection filters*) representam a maior categoria, dentro da classe geral de filtros não lineares. A esta categoria pertencem por exemplo: o **filtro mediano**, o **filtro mediano ponderado central** “**CWM**” (*center weighted median filter*) [Yu, 1994], o **filtro de ordem estatística ponderada** “**WOS**” (*weighted order statistic filter*) [Yin, 1997], etc. Ao contrário dos filtros lineares; os filtros baseados em ordem estatística exibem excelentes propriedades de robustez no processamento de imagens. Eles se caracterizam por minimizar o ruído, ao mesmo tempo que evitam o borramento extensivo da imagem.

Este trabalho estuda o desempenho da classe não linear de filtros chamados **Filtros Condicionais Seletores de Posição “RCRS”** (*rank conditioned rank selection filters*) que se desenvolvem dentro de uma categoria mais geral que corresponde aos filtros seletores **RS**. Estes últimos tem como principal característica selecionar como saída uma ordem estatística de um conjunto de amostras da entrada. A diferença de tais filtros está em que informação é utilizada para decidir qual é a ordem estatística selecionada como saída. O tipo de informação usada nos filtros **RCRS** corresponde ao posicionamento (*rank*) das amostras de entrada de onde surgiu o nome de filtros condicionais seletores de posição [Hardie, 1994].

A extrema necessidade de recuperar a informação de imagens degradadas e a grande habilidade que os filtros **RCRS** tem para minimizar o ruído sem alterar as estruturas da imagem, motivaram o estudo sobre esta técnica de filtragem em restauração de imagens.

Para tanto, além dos fundamentos de processamento de imagens digitais, foram estudados os princípios da filtragem espacial. Também foram implementados algoritmos com o objetivo de simular a restauração de imagens contaminadas com ruído envolvendo filtros *RCRS*.

## **1.2. Organização do trabalho**

Para uma melhor compreensão e conveniência do leitor, este trabalho encontra-se estruturado da seguinte maneira:

### **Capítulo 1: Introdução.**

### **Capítulo 2: Imagem Digital e Filtragem Espacial. Fundamentos.**

Este capítulo tem o propósito de apresentar vários conceitos básicos relacionados a imagens digitais, a filtragem espacial e algumas notações utilizadas ao longo deste trabalho. A primeira seção mostra a representação de imagens digitais. A segunda seção apresenta um modelo simples de imagem. A terceira seção descreve brevemente os conceitos de amostragem e quantização. A quarta seção mostra alguns relacionamentos básicos entre pixels. Finalmente a quinta seção apresenta várias técnicas básicas de processamento de imagens digitais.

### **Capítulo 3: Filtro Condicional Seletor de Posição (*RCRS*).**

Este capítulo introduz a classe de filtros *RCRS*. Inicia-se com um preâmbulo, no qual é realizada uma introdução sobre este tipo de filtragem. Logo na segunda seção mostra-se a estrutura dos filtros *RCRS*. Na terceira seção apresenta-se a definição dos filtros *RCRS*. Após a definição do filtro, na quarta seção mostra-se a relação dos filtros *RCRS* com outras classes de filtros. Finalmente a quinta seção apresenta um processo de otimização sobre a classe de filtros *RCRS*.

## **Capítulo 4: Simulações e Resultados.**

Neste capítulo são apresentados os resultados obtidos depois da restauração de uma imagem contaminada com ruído (impulsivo e Gaussiano). Aqui é realizada uma análise quantitativa do desempenho de várias técnicas de restauração. São mostrados diferentes gráficos com as curvas dos Erros Médios Absolutos (**MAE**). Também são apresentadas várias imagens processadas para uma avaliação subjetiva

## **Capítulo 5: Conclusões e Recomendações para Trabalhos Futuros.**

No capítulo 5 são realizadas conclusões sobre os resultados obtidos no quarto capítulo deste trabalho e comenta-se sobre os trabalhos futuros que poderão ser realizados.

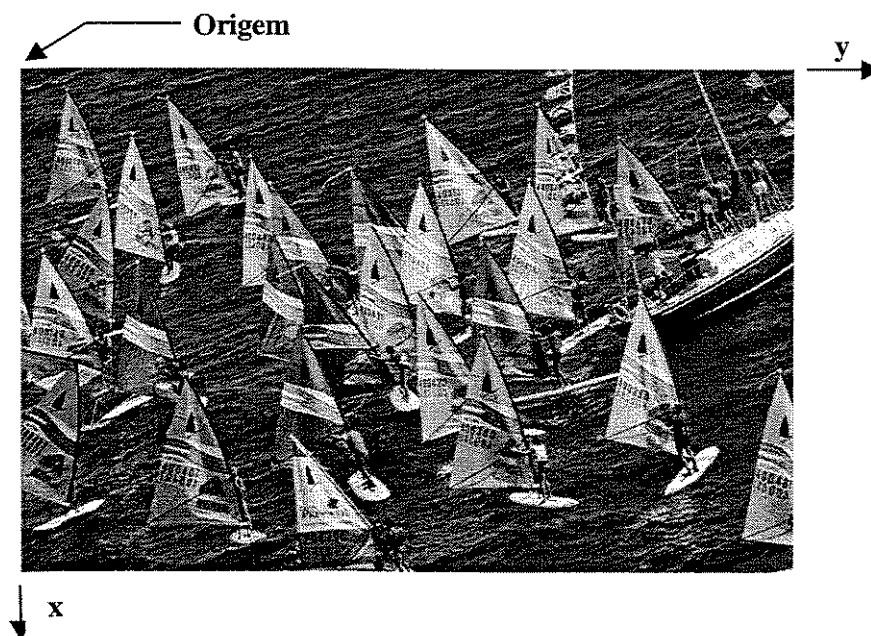
## CAPÍTULO 2. IMAGEM DIGITAL E TÉCNICAS BÁSICAS DE PROCESSAMENTO. FUNDAMENTOS.

---

Este capítulo tem o propósito de apresentar vários conceitos básicos relacionados a imagens digitais, a filtragem espacial e algumas notações utilizadas ao longo deste trabalho. A primeira seção mostra a representação de imagens digitais. A segunda seção apresenta um modelo simples de imagem. A terceira seção descreve brevemente os conceitos de amostragem e quantização. A quarta seção mostra alguns relacionamentos básicos entre *pixels*. Finalmente a quinta seção apresenta várias técnicas básicas de processamento de imagens digitais.

### 2.1. Representação de Imagens Digitais

O termo **imagem monocromática**, ou simplesmente **imagem**, refere-se à função bidimensional de intensidade da luz  $f(x,y)$ , no qual  $x$  e  $y$  denotam as coordenadas espaciais e o valor de  $f$  em qualquer ponto  $(x,y)$  é proporcional ao brilho (ou **níveis de cinza**) da imagem naquele ponto[Gonzalez, 2000]. A **Figura 2.1** ilustra a convenção dos eixos utilizada ao longo deste capítulo.



**Figura 2.1.** Convenção dos eixos para representação de imagens digitais.



Uma **imagem digital** é uma imagem  $f(x,y)$  discretizada tanto em coordenadas espaciais quanto em brilho. Uma imagem digital pode ser considerada como sendo uma matriz cujos índices de linhas e das colunas identificam um ponto na imagem, e o correspondente valor do elemento da matriz identifica o nível de cinza naquele ponto. Os elementos desta matriz são chamados de elementos da imagem, elementos da figura, “*pixels*” ou “*pels*”, estes dois últimos, abreviações de “*picture elements*” (elementos de figura).

Embora o tamanho de uma imagem digital varie de acordo com a aplicação, do ponto de vista computacional tem muitas vantagens selecionar matrizes quadradas com tamanhos e números de níveis de cinza que sejam potências de inteiras de 2. Por exemplo, um tamanho típico comparável em qualidade de imagem como a de uma TV preto e branco é uma matriz 512 x 512 com 256 níveis de cinza.

## 2.2. Um Modelo Simples De Imagem

O termo **imagem** refere-se a uma função de intensidade luminosa bidimensional, denotada por  $f(x,y)$ , em que o valor ou amplitude de  $f$ , nas coordenadas espaciais  $(x,y)$  fornece a intensidade (brilho) da imagem neste ponto [Gonzalez, 2000].

Como a luz é uma forma de energia,  $f(x,y)$  deve ser positiva e finita, isto é:

$$0 < f(x, y) < \infty \quad (2.2-1)$$

As imagens que as pessoas percebem em atividades corriqueiras consistem da luz refletida dos objetos. A natureza básica de  $f(x,y)$  pode ser caracterizada por dois componentes:

- A quantidade de luz incidindo na cena sendo observada.
- A quantidade de luz refletida pelos objetos na cena.

Apropriadamente, essas componentes são chamadas *iluminação e reflectância*, respectivamente, são representadas por  $i(x,y)$  e  $r(x,y)$ . O produto das funções  $i(x,y)$  e  $r(x,y)$  resulta  $f(x,y)$ , tal que:

$$f(x, y) = i(x, y)r(x, y) \quad (2.2-2).$$

onde

$$0 < i(x, y) < \infty \quad (2.2-3)$$

$$0 < r(x, y) < 1 \quad (2.2-4)$$

A equação (2.2-4) indica que a reflectância “r” é limitada entre 0 (absorção total) e 1 (reflectância total). A natureza de  $i(x,y)$  é determinada pela fonte e  $r(x,y)$  é obtida a partir das características dos objetos na cena.

Os valores encontrados nas equações (2.2-3) e (2.2-4) são limites teóricos. Os seguintes valores médios ilustram alguns intervalos típicos de  $i(x,y)$ . Num dia claro, o Sol pode produzir mais de 9.000 pé-candelas de iluminação na superfície da Terra. Esse valor diminui para menos de 1.000 pé-candelas em um dia nublado. Em uma noite clara, a Lua cheia gera cerca de 0,01 pé-candelas de iluminação. O nível de iluminação típico em um escritório comercial é de cerca de 100 pé-candelas. Similarmente, valores típicos de  $r(x,y)$  são 0,01 para veludo negro, 0,65 para aço inoxidável, 0,8 para uma parede branca, 0,9 para metal prateado, e 0,93 pé-candela para a neve.

Ao longo deste capítulo denomina-se a intensidade de uma imagem monocromática  $f$  nas coordenadas  $(x,y)$  de nível de cinza ( $I$ ) da imagem naquele ponto. Das equações (2.2-2) e (2.2-4) é evidente que  $I$  fica restrito ao intervalo:

$$L_{\min} \leq I \leq L_{\max} \quad (2.2-5)$$

Na teoria a única restrição sobre  $L_{\min}$  é que seja um valor positivo e sobre  $L_{\max}$  é que seja finito. Na prática  $L_{\min} = i_{\min} r_{\min}$  e  $L_{\max} = i_{\max} r_{\max}$ . Caso sejam considerados os valores

de iluminação e refletância acima pode-se esperar  $L_{min} = 0,005$  e  $L_{max} = 100$  para aplicações de processamento de imagem em ambientes fechados.

O intervalo  $[L_{min}, L_{max}]$  é denominado **escala de cinza**. Na prática é comum deslocar esse intervalo para  $[0, L]$  onde  $l = 0$  é considerado negro e  $l = L$  é considerado branco. Todos os valores intermediários são tons de cinza variando continuamente entre o branco e o negro.

### 2.3. Amostragem e Quantização

Uma função  $f(x,y)$  para ser adequada para o processamento pelo computador precisa ser digitalizada tanto espacialmente quanto em amplitude. A digitalização das coordenadas espaciais  $(x,y)$  é denominada **amostragem da imagem**, e a digitalização em amplitude é chamada **quantização em níveis de cinza** [Gonzalez, 2000].

Suponha que uma imagem contínua  $f(x,y)$  é aproximada por amostras igualmente espaçadas, arranjadas na forma de uma matriz  $N \times M$ , como mostrado na equação (2.3-1), em que cada elemento é uma quantidade discreta:

$$f(x, y) = \begin{bmatrix} f(0,0) & f(0,1) & \dots & f(0,M-1) \\ f(1,0) & f(1,1) & \dots & f(1,M-1) \\ \vdots & \vdots & \ddots & \vdots \\ f(N-1,0) & f(N-1,1) & \dots & f(N-1,M-1) \end{bmatrix} \quad (2.3-1)$$

O lado direito da equação (2.3-1) representa uma **imagem digital**. Cada elemento da matriz denomina-se elemento de imagem, **pixel**, (abreviação de “*picture element*” elemento de figura) ou **pel** como indicado na Seção 2.1. Os termos **imagem** e **pixel** serão usados no resto deste trabalho para denotar uma imagem e os seus elementos.

Em certas ocasiões, torna-se útil exprimir amostragem e quantização em termos matemáticos mais formais. Sejam  $Z$  e  $R$  os conjuntos dos números inteiros e reais respectivamente. O processo de amostragem pode ser compreendido como a partição do plano  $xy$  em uma grade, com as coordenadas de cada cruzamento da grade sendo um par de elementos obtidos do produto cartesiano  $Z \times Z$  (também representado por  $Z^2$ ), que é o conjunto de todos os pares ordenados  $(a,b)$ , onde  $a$  e  $b$  são elementos de  $Z$ . Portanto,  $f(x,y)$  é uma imagem digital se os pares  $(x,y)$  forem elementos de  $Z \times Z$  e  $f$  é uma função que atribui um valor de nível de cinza (isto é, um número inteiro) a cada par de coordenadas  $(x,y)$  distinto. Essa atribuição funcional é obviamente o processo de quantização descrito anteriormente.

Se os níveis de cinza são inteiros (como é utilizado neste capítulo e nos capítulos subseqüentes),  $R$  é substituído por  $Z$ , e uma imagem digital torna-se então uma função bidimensional (**2-D**) com valores inteiros de coordenadas e amplitude. Esse processo de digitalização envolve decisões a respeito dos valores para  $N$ ,  $M$ , e para o **número de níveis de cinza** discretos permitidos para cada pixel. A prática comum em processamento de imagens digitais é assumir que estas quantidades são potências inteiras de dois, isto é,

$$N = 2^n, \quad M = 2^k \quad (2.3-2)$$

e

$$G = 2^m \quad (2.3-3)$$

onde  $G$  é o número de níveis de cinza.

Neste trabalho é assumido que os níveis discretos são igualmente espaçados entre  $0$  e  $L$  na escala de cinza. O número,  $B$ , de *bits* necessários para armazenar uma imagem digitalizada é obtido através das equações (2.3-2) e (2.3-3):

$$B = N \times M \times m \quad (2.3-4)$$

$$\text{Se } N = M, \quad \rightarrow \quad B = N^2 m \quad (2.3-5)$$

Por exemplo, uma imagem de  $128 \times 128$  *pixels* com 64 níveis de cinza requer 98.304 *bits* para armazenamento.

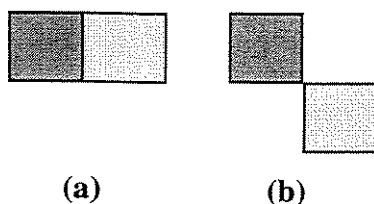
## 2.4. Alguns Relacionamentos Básicos entre *Pixels*

### 2.4.1. Definições

#### *Definição 2.1. Adjacência:*

Dois *pixels* são adjacentes entre si, se eles tem em comum uma de suas fronteiras (*adjacentes por fronteira*), ou ao menos uma de suas esquinas (*adjacentes por esquina*).

A figura 2.2 mostra *pixels* adjacentes.



**Figura 2.2.** *Pixels* adjacentes: (a) adjacentes por fronteira ; (b) adjacentes por esquina.

#### *Definição 2.2. Pixel Vizinho*

Dois *pixels* são vizinhos, se eles são adjacentes. Se os *pixels* compartilham uma de suas fronteiras, eles são chamados “**Vizinhos Diretos**”. Quando os *pixels* se tocam só em uma de suas esquinas eles são denominados “**Vizinhos Indiretos**”.

#### *Definição 2.3. Vizinhança*

Uma vizinhança de um pixel  $p$  é um submatriz  $N_{KL}$  de tamanho  $K \times L$ , onde  $K$  e  $L$  são inteiros ímpares pequenos. Esta submatriz está contida na matriz imagem  $F_{XY}$ , que contém um número finito de *pixels* vizinhos ou não de  $p$ .

$$V_p = \{p: p \in N_{KL}\}; \quad N_{KL} \subset F_{XY}; \quad K=L=3,5,9 \quad (2.4-1)$$

### 2.4.2. Os vizinhos de um pixel

Das definições 2.2 e 2.3 pode-se observar que um pixel  $p$  de coordenada  $(x,y)$  possui quatro vizinhos horizontais e verticais, cujas coordenadas são dadas por:

$$(x+1,y), (x-1,y), (x,y+1), (x,y-1)$$

O conjunto dos 4 *pixels* vizinhos é chamado **vizinhos  $p$** , e representado por  $N_4(p)$ .

Os quatro vizinhos diagonais de  $p$ , denominados de 4 vizinhos diagonais, representado por  $N_D(p)$ , possuem como coordenadas:

$$(x+1,y+1), (x+1,y-1), (x-1,y+1), (x-1,y-1)$$

Os 4 vizinhos  $N_4(p)$  e os 4 vizinhos  $N_D(p)$  formam os 8 vizinhos de  $p$ , representados por  $N_8(p)$ .

### 2.4.3. Conectividade

A conectividade entre *pixels* é um conceito importante utilizado no estabelecimento das bordas de objetos e componentes de regiões em uma imagem. Para estabelecer se dois *pixels* estão conectados é preciso determinar se eles são de alguma forma adjacentes (por exemplo são  $N_4(p)$ ), e se seus níveis de cinza satisfazem um certo critério de similaridade. Por exemplo, em uma imagem binária com valores 0 e 1 dois *pixels* podem ser 4 vizinhos mas eles podem não estar conectados a não ser que possuam o mesmo nível de cinza.

Seja  $V$  o conjunto de valores de níveis de cinza usados para definir conectividade em uma imagem binária.  $V = \{1\}$  para conectividade de *pixels* com valor 1.

Em uma imagem com níveis de cinza, para a conectividade de *pixels* com valores de níveis de cinza  $V = [32, ..., 64]$ , existem três tipos de conectividade:

- **Conectividade 4:** dois *pixels*  $p$  e  $q$ , com valores em  $V$ , são conectados de 4 se  $q$  está no conjunto  $N_4(p)$ .
- **Conectividade 8:** dois *pixels*  $p$  e  $q$ , com valores em  $V$ , são conectados de 8 se  $q$  está no conjunto  $N_8(p)$ .
- **Conectividade  $m$**  (*conectividade mista*): dois *pixels*  $p$  e  $q$ , assumindo valores em  $V$ , são conectados de  $m$  se:
  - $q$  está no conjunto  $N_4(p)$ , ou
  - $q$  está no conjunto  $N_D(p)$ , e o conjunto  $N_4(p) \cap N_4(q)$  é vazio. Este é o conjunto dos *pixels* que são 4 vizinhos de  $p$  e  $q$ , cujos valores estão em  $V$ .

A conectividade mista é uma variante da conectividade 8 introduzida para eliminar as conexões por múltiplos caminhos. A conectividade mista frequentemente aparece quando é utilizada a conectividade 8.

O caminho de um pixel  $p$  com coordenada  $(x,y)$  a um pixel com coordenada  $(x,t)$  é uma sequência de *pixels* distintos com coordenadas :

$$(x_0, y_0), (x_1, y_1), \dots, (x_n, y_n)$$

Se  $p$  e  $q$  são *pixels* de um subconjunto  $S$  de uma imagem, então  $p$  está conectado a  $q$  em  $S$  se existir um caminho de  $p$  a  $q$  consistindo inteiramente de *pixels* de  $S$ . Para qualquer pixel  $p$  em  $S$ , o conjunto dos *pixels* em  $S$  que estão conectados a  $p$  é denominado componente convexo de  $S$ . Deste modo, dois *pixels* quaisquer de um componente convexo estão conectados um ao outro. Os componentes convexas distintos são disjuntos.

#### 2.4.4. Operações aritméticas

Operações aritméticas entre *pixels* são extensivamente usadas na maioria dos ramos de processamento de imagens.

As operações aritméticas entre dois *pixels*  $p$  e  $q$  podem ser:

<b>Adição:</b>	$p + q$
<b>Subtração:</b>	$p - q$
<b>Multiplicação:</b>	$p * q$ ( <i>também <math>pq</math> e <math>pxq</math></i> )
<b>Divisão:</b>	$p \div q$

Operações aritméticas em imagens inteiras são realizadas pixel a pixel. Adição de imagens é muito usada para fazer a média de imagens para redução de ruído. A subtração de imagens é uma ferramenta básica em imagens médicas, a qual é usada para remover informação estática de fundo.

Um dos principais usos da multiplicação (ou divisão) de imagens é para corrigir sombras de níveis de cinza produzidas por não-uniformidades da iluminação ou no sensor utilizado para a aquisição da imagem.

Operações aritméticas envolvem apenas uma posição espacial de pixel por vez, de modo que elas podem ser realizadas “no local,” no sentido de que o resultado da operação aritmética realizada na posição  $(x,y)$ . Portanto, o resultado pode ser armazenado nessa posição em uma das imagens existentes, visto que essa posição não será mais utilizada nesta operação.

Além do processamento pixel a pixel em imagens inteiras, operações aritméticas são utilizadas em **operações orientadas a vizinhança**. O processamento da vizinhança é formulado em um contexto das operações por máscaras (também denominadas: janelas, *template*, molde e filtro, para denominar uma máscara). As operações com máscaras modificam o valor do pixel em função do seu próprio nível de cinza e de seus *pixels* vizinhos.



## 2.5. Técnicas Básicas de Processamento

### 2.5.1. Fundamentos

O processamento de imagens digitais é um conjunto de técnicas que operam sobre a imagem, com o objetivo de destacar alguns dos elementos que compõem a cena, de modo que possa facilitar sua posterior análise pelo ser humano, ou um sistema de visão artificial, [Lim, 1990]. Em geral, as técnicas de processamento de imagens são aplicadas para realçar, melhorar a aparência da imagem ou para destacar algum aspecto da informação contida na mesma. Estas técnicas também são utilizadas para medir, contrastar ou classificar algum elemento contido na imagem, e quando é necessário combinar imagens ou porções das mesmas ou reorganizar seu conteúdo.

As técnicas de processamento de imagens digitais podem ser classificadas em duas categorias: a primeira baseia-se em **pontos da imagens**, que agrupam as operações de processamento no qual o valor do pixel só depende de seu valor de entrada, a segunda inclui as técnicas baseadas em uma **região ou área da imagem**, no qual um pixel da imagem de saída depende tanto do valor do pixel na imagem como de sua vizinhança.

### 2.5.2. Técnicas de Processamento baseadas em Pontos da Imagem

As técnicas de processamento baseadas em pontos da imagem consistem em algoritmos que modificam o valor do pixel baseados unicamente no seu valor ou na sua localização. Nenhum outro valor de pixel é relacionado nesta transformação. O processamento é realizado varrendo pixel por pixel da imagem a ser processada. Quando a transformação aplicada depende somente do valor original do pixel, pode ser interessante utilizar a Tabela de Busca (*LUT/Look Up Table*). Caso contrário, quando é necessário considerar o valor do pixel e a sua posição, pode resultar necessário utilizar fórmulas ou uma combinação das mesmas com tabelas de busca.

De maneira geral, as técnicas de processamento baseadas em pontos da imagem não modificam as relações espaciais dentro da imagem e em consequência não modificam o grau de detalhe contido nas mesmas. Elas são simples e podem ser úteis sozinhas ou em conjunto com outras técnicas mais complexas. Estas técnicas devido a sua simplicidade, podem ser aplicadas utilizando-se um *hardware* menos complicado para uma execução em tempo real.

### **2.5.3. Técnicas de Processamento por Área ou Região da Imagem**

As técnicas de processamento digital, aplicadas sobre certa região de uma imagem, que estão relacionadas com a análise da informação que possuem os pontos localizados em torno do ponto que se deseja estudar, são usualmente denominadas: **Técnicas de Processamento por Área ou Região.**

As imagens devem ser observadas no momento de aplicar qualquer forma de processamento como uma cena digital de duas dimensões (arranjo bidimensional). Tal arranjo é denominado **matriz imagem**, que na maioria dos sistemas de processamento de imagens são matrizes quadradas. Ao obter a matriz imagem o que é feito realmente é estabelecer uma grade ou malha, no qual cada ponto da mesma é um pixel.

As técnicas de processamento baseadas em uma região têm muitas aplicações na obtenção das características primitivas da imagem, como por exemplo, a extração de contornos para seu realce, para suavizar uma imagem, para introduzir borramento dentro da mesma e para atenuar ruído aleatório. Estas técnicas usam um grupo (**vizinhança**) de *pixels* dentro da imagem a processar com o propósito de extrair informação da mesma. O pixel do centro da vizinhança normalmente é substituído por um novo valor, que é o resultado da aplicação de algum algoritmo de processamento.

O fato de considerar *pixels* de uma vizinhança, faz com que as técnicas de processamento baseadas em uma região, possuam maior custo de cálculo numérico que as técnicas baseadas em pontos da imagem. Este custo depende do tamanho da vizinhança

considerada, e do tipo de representação numérica utilizado. Porém, para maioria das aplicações e com os computadores disponíveis atualmente, pode-se obter resultados muito bons em termos de tempo de cálculo, ao processar imagens de tamanho mediano (256x256 ou 512x512).

#### **2.5.4. Filtragem Espacial**

Filtragem espacial é um conjunto de técnicas destinadas a corrigir e realçar uma imagem. A correção é a remoção de características indesejáveis, e a melhoria/realce é a acentuação de características da imagem. O uso de janelas ou máscaras espaciais para o processamento de imagens é chamado **filtragem espacial** (em contrapartida à expressão **filtragem no domínio da frequência** que usa a transformada de Fourier). As janelas ou máscaras são chamadas de **filtros espaciais** e podem ser classificados como:

1. **Filtros espaciais lineares** que geram uma matriz imagem ao aplicar um processo linear sobre a imagem de entrada.
2. **Filtros espaciais não lineares** cuja função básica é a análise estatística dos valores de níveis de cinza na vizinhança que o filtro está posicionado (exemplo: filtro mediano).

Independente do tipo de filtro espacial utilizado, os procedimentos realizados com esta filtragem são:

- a) É realizada uma única varredura na imagem de entrada pixel por pixel, segundo linhas e colunas.
- b) Cada pixel da imagem de entrada é processado, considerando uma vizinhança e utilizando um algoritmo apropriado.
- c) O novo valor do pixel obtido no item b) é colocado na imagem de saída na mesma posição que ele ocupava na imagem de entrada.

### 2.5.5. Filtros espaciais lineares

Os **filtros espaciais lineares** baseiam-se exclusivamente na realização de uma combinação dos *pixels* pertencentes a uma vizinhança do pixel em estudo com uma matriz bidimensional (janela ou máscara) que descreve o processo linear a se aplicar. Esta técnica é denominada de **Convolução**.

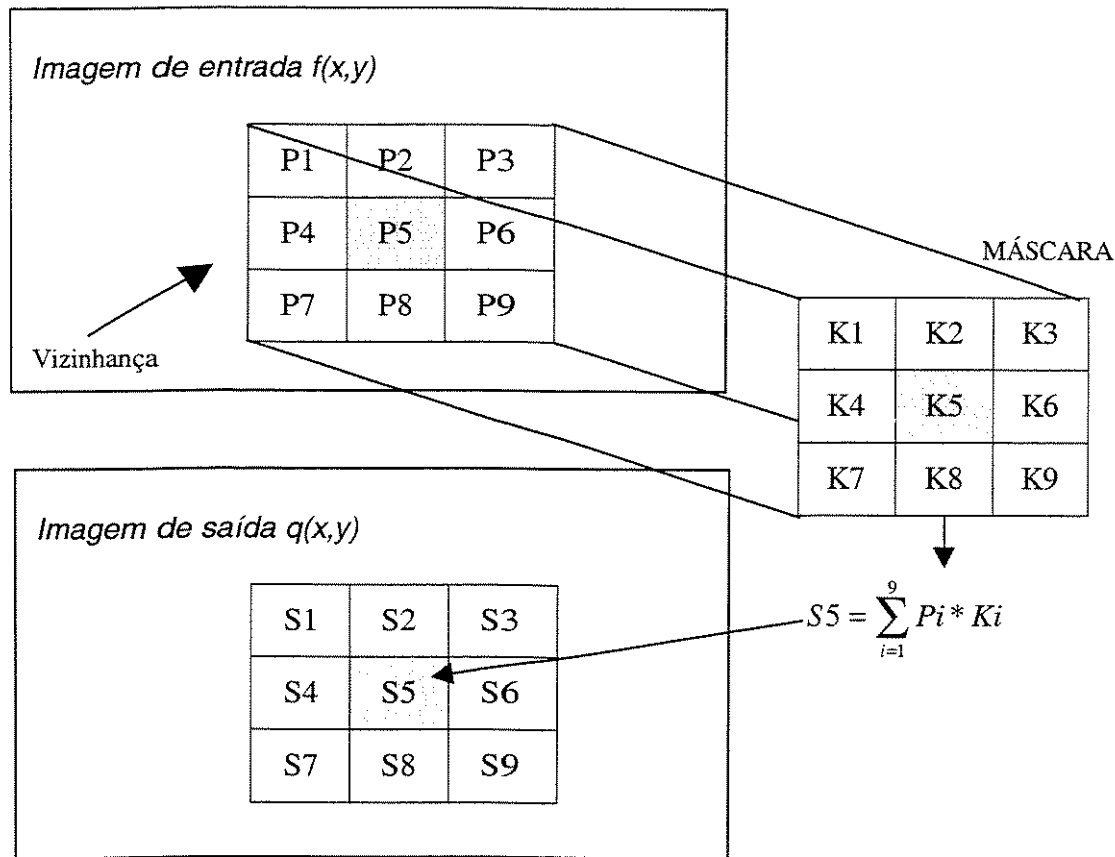
No processamento de imagens, a convolução é a extensão do caso unidimensional, mediante em que um sinal qualquer pode ser processado por um filtro arbitrário de resposta impulsiva conhecida [Oppenheim e Schaffer 1975, Lim 1990]. A convolução de uma imagem  $f(x,y)$  com o filtro (núcleo ou máscara de convolução) de resposta impulsiva  $h(x,y)$  produz uma imagem de saída  $q(x,y)$ , conforme a equação (2.5-1) onde  $m,n$  definem a vizinhança a se considerar de acordo com o tamanho da máscara de convolução  $h(x,y)$ .

$$q(x,y) = \sum_{m=-K1}^{K2} \sum_{n=-L1}^{L2} h(m,n)f(x-m,y-n) \quad (2.5-1)$$

Para a implantação direta da equação de convolução, é assumido que o tamanho da máscara de convolução é uma matriz ( $N1 \times N2$ ). Usualmente esta matriz é  $3 \times 3$  *pixels*. O tamanho da vizinhança dentro da imagem na convolução é do mesmo tamanho da máscara. Para se obter o novo valor do pixel, multiplica-se a janela pelo valor da imagem original em torno do pixel, elemento a elemento, e soma-se o produto, obtendo-se o valor do pixel na imagem  $q(x,y)$ .

A equação (2.5-1) mostra a relação entre uma imagem de entrada  $f(x,y)$  e a imagem filtrada  $q(x,y)$ , caso seja estabelecido que os coeficientes (pesos) da máscara não dependem de sua posição na imagem. Além disso, a equação (2.5-1) descreve um processo de convolução discreta realizado sobre todos os *pixels* da imagem, que é um processo que caracteriza os filtros lineares.

A **Figura 2.3** mostra o processo de convolução segundo o qual para um dado pixel dentro da imagem de entrada  $f(x,y)$ , cada pixel da vizinhança é multiplicado pelo pixel correspondente na máscara de convolução. Cada um destes produtos é somado, e o novo valor do pixel na imagem de saída  $q(x,y)$  será a soma de todos estes produtos. O processamento de toda a imagem é realizado deslocando a máscara e repetindo o mesmo procedimento para cada pixel.



**Figura 2.3.** Processo de convolução utilizando uma máscara 3x3.

A máscara de convolução usualmente possui um número ímpar de linhas e colunas. Geralmente os tamanhos destas máscaras são 3 x 3 , 5 x 5 , 7 x 7 e 9 x 9. O conteúdo das máscaras depende do tipo de processamento que se deseja realizar (ver Anexo 1). Exemplos de diferentes tipos de máscaras são apresentados em [Pratt, 1978].

O algoritmo de convolução mostrado envolve detalhes em sua implementação que devem ser resolvidos para se obter bons resultados. Um dos detalhes relaciona-se com o processamento dos *pixels* localizados nas bordas da imagem. Em tal situação a máscara não está superposta sobre os 9 *pixels* (tamanho 3 x 3) da imagem de entrada. As soluções mais diretas nesta situação são:

- Os *pixels* das bordas são ignorados durante o processo de convolução e na saída são colocados os mesmos valores que eles tinham na imagem de entrada.
- Os *pixels* localizados nas bordas são replicados para obter o número de *pixels* necessários para realizar o processo de convolução.
- Os *pixels* da primeira coluna são considerados vizinhos dos *pixels* da última coluna e os *pixels* da primeira linha são considerados vizinhos dos *pixels* da última linha.

A faixa dinâmica da imagem de saída é outro problema que ocorre no algoritmo de convolução. Os valores dos *pixels* nesta imagem precisam de outro tipo de precisão para ser representados, caso contrário, eles poderão possuir valores negativos. Em tais casos é necessário um ajuste de escala para manter a mesma precisão.

A classe de **filtros lineares** contém os denominados **filtros passa-baixas ou de suavização**. Estes filtros atenuam ou eliminam as componentes de alta-frequência no domínio de Fourier enquanto deixam as componentes de baixas frequências praticamente inalteradas. Este tipo de filtro é adequado em pré – processamento, tal como a remoção de pequenos detalhes de uma imagem antes da extração de objetos (grandes), na conexão de descontinuidades em linhas e curvas e para atenuar ou minimizar o ruído aditivo aleatório. No entanto, como as componentes de alta-frequência caracterizam bordas e outros detalhes finos de uma imagem, o efeito resultante da filtragem passa – baixas é o borramento da imagem.

A **Figura 2.4** mostra três máscaras ou janelas de convolução utilizadas freqüentemente para realizar a filtragem passa – baixas. Uma das propriedades de tais máscaras é que a soma de todos seus valores é igual a unidade.

$\frac{1}{9}$ 

1	1	1
1	1	1
1	1	1

$\frac{1}{10}$ 

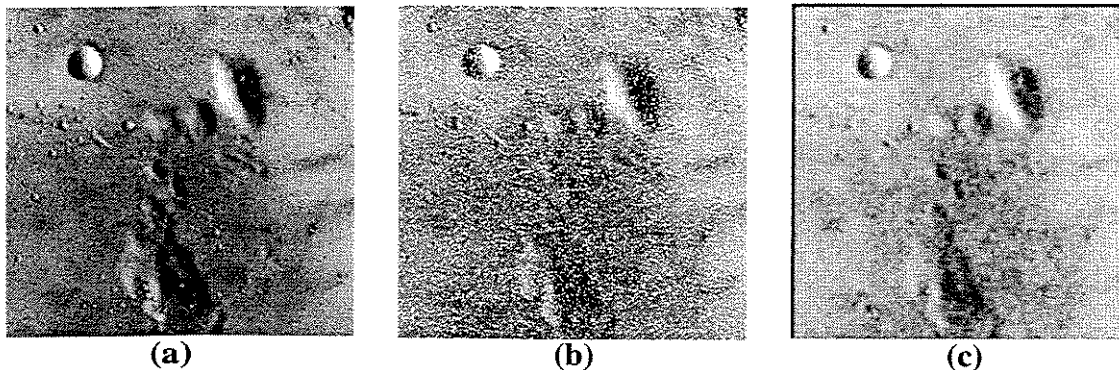
1	1	1
1	2	1
1	1	1

$\frac{1}{16}$ 

1	2	1
2	4	2
1	2	1

**Figura 2.4.** Três máscaras que permitem a filtragem passa – baixas.

A **Figura 2.5** mostra o processamento de uma imagem por um filtro passa – baixas. A Figura 2.5a mostra a imagem original. A figura 2.5b mostra a imagem original contaminada com ruído aleatório. A Figura 2.5c mostra o resultado depois de processar a imagem com um **filtro passa–baixas**. Observa-se nesta figura que o conteúdo do ruído diminui, contudo as bordas e os detalhes finos da imagem borram.



**Figura 2.5.** Processamento de uma imagem com um filtro **passa–baixas**: a) imagem original, b) imagem contaminada por ruído, c) imagem processada.

Outro tipo de filtro a se considerar na classe de filtros lineares, são os **filtros passa–altas**. Estes filtros acentuam as componentes de alta frequência de uma imagem e atenuam ou eliminam as componentes de baixas frequências. Como as componentes de baixa frequência são responsáveis pelas características que variam lentamente em uma imagem, tais como o contraste total e a intensidade média e o efeito resultante da filtragem passa–altas é uma redução destas características.

No processamento utilizando filtros passa–altas, as porções da imagem que apresentam componentes de alta frequência são ressaltadas com a utilização de níveis de cinza mais claros e as componentes de baixa frequência ficam mais escuras, correspondendo a uma

aparente acentuação das bordas e outros detalhes finos da imagem. Uns dos efeitos indesejáveis dos filtros passa–altas é que eles podem acentuar o ruído presente na imagem.

A **Figura 2.6** mostra três máscaras de convolução utilizadas como **filtro passa–altas**.

-1	-1	-1
-1	9	-1
-1	-1	-1

(a)

0	-1	0
-1	5	-1
0	-1	0

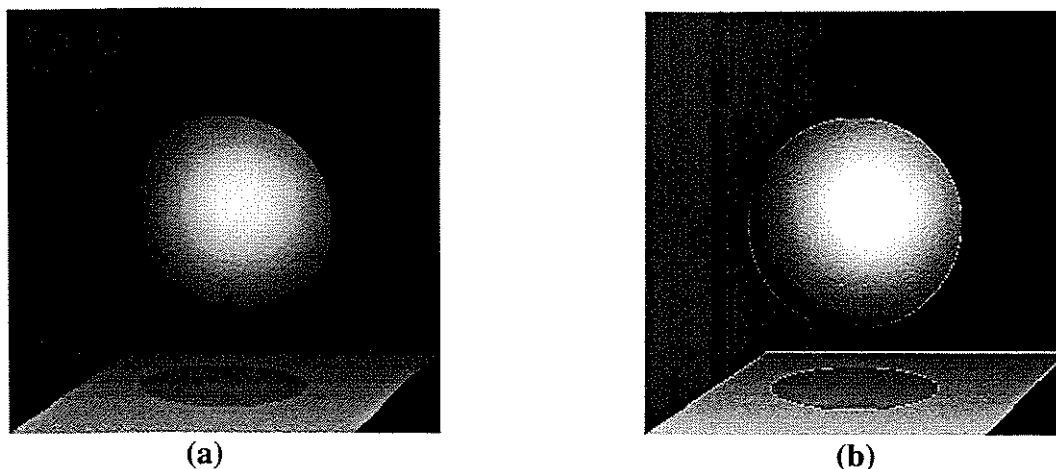
(b)

1	-2	1
-2	5	-2
1	-2	1

(c)

**Figura 2.6.** Três máscaras utilizadas como filtros passa – altas.

A **Figura 2.7** (abaixo) mostra o resultado de uma imagem após passar por um **filtros passa–altas** utilizando a máscara ou janela de convolução mostrada na Figura 2.6a. Pode-se verificar que as bordas da imagem original foram reforçadas.

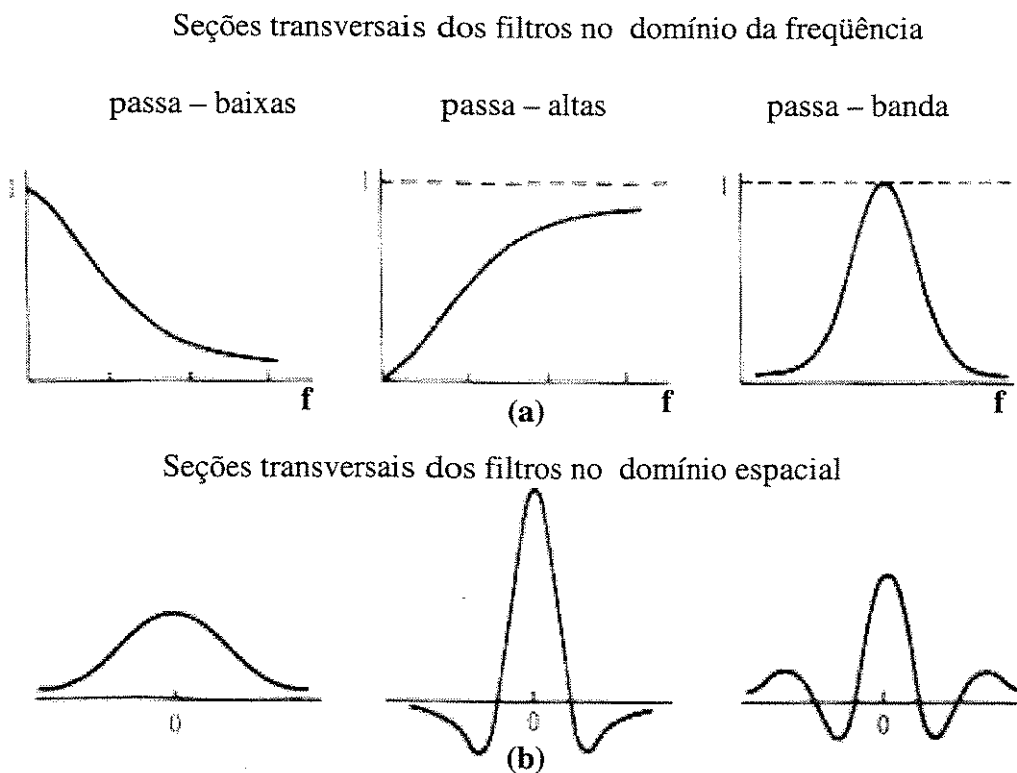


**Figura 2.7.** Processamento de uma imagem por um filtro passa–altas: a) imagem original, b) imagem resultante logo depois de ser processada com a máscara da Figura 2.6a.

Um terceiro tipo de filtragem é a **filtragem passa–banda**, que remove regiões selecionadas de frequências entre as altas e as baixas frequências. Estes filtros são usados principalmente na restauração de imagens.



A **Figura 2.8** ilustra os cortes dos **filtros passa-baixas** circularmente simétrico, **passa-altas**, e **passa-banda** no domínio da frequência e seus correspondentes **filtros espaciais**.



**Figura 2.8.** (a) Seções transversais dos filtros simétricos no domínio da frequência. (b) Seções transversais dos filtros no domínio espacial.

Independente do tipo de **filtro linear** usado, o processo de filtragem consiste em somar os produtos entre os coeficientes da janela e os valores de intensidade dos *pixels* na janela em uma posição específica da imagem. A **Figura 2.9** mostra uma janela genérica 3 x 3 com coeficientes  $w_1, w_2, w_3, w_4, w_5, w_6, w_7, w_8, w_9$ . Os níveis de cinza dos *pixels* na região da imagem onde a janela está posicionada são respectivamente:  $z_1, z_2, z_3, z_4, z_5, z_6, z_7, z_8, z_9$ .

$w_1$	$w_2$	$w_3$
$w_4$	$w_5$	$w_6$
$w_7$	$w_8$	$w_9$

**Figura 2.9.** Uma janela 3x3 com coeficientes arbitrários (pesos)  $w_i$ .

A resposta da janela linear é:

$$R = w_1 z_1 + w_2 z_2 + \dots + w_9 z_9 \quad (2.5-2)$$

O centro da janela na posição  $(x,y)$  na imagem é substituído por  $R$ . A janela é movida para a próxima posição de pixel na imagem e o processo se repete. Este processo é repetido até que todos os *pixels* da imagem tenham sido varridos.

Ao utilizar a equação (2.5-2), a resposta é a soma dos níveis de cinza de nove *pixels*, e isto faz com que  $R$  fique fora do intervalo de níveis de cinza válidos. A solução é normalizar a soma dividindo  $R$  por 9. Além disso, é usual armazenar os valores de  $R$  em uma nova imagem. Isto evita o uso de níveis de cinza na equação (2.5-2) que tenham sido alterados com o resultado da aplicação da janela anteriormente descrita.

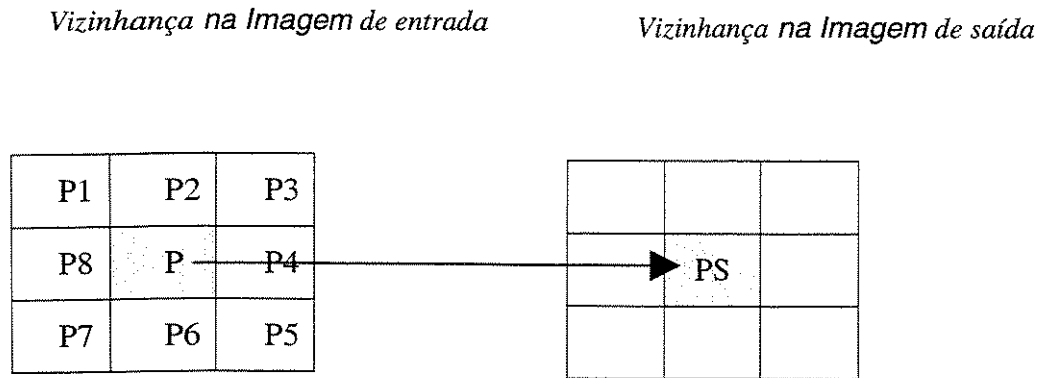
### **2.5.6. Filtros Espaciais Não Lineares**

Os **filtros espaciais não lineares** também operam em uma vizinhança. Em geral os procedimentos não-lineares destes filtros baseiam-se diretamente nos valores dos *pixels* na vizinhança considerada, não utilizando os coeficientes como descrito na equação (2.5-2). Os filtros espaciais não – lineares fazem uma análise estatística dos valores de níveis de cinza na vizinhança em que o filtro está posicionado (por exemplo: filtro mediano) [Gonzalez, 1999].

Uma das principais dificuldades da filtragem linear é que ela além de minimizar o ruído ou realçar algumas características da imagem, causa diversas alterações na estrutura do sinal (imagem) como por exemplo, o borramento das bordas e outros detalhes. Uma abordagem alternativa consiste no uso de técnicas de filtragem **não – lineares**, que caracterizam-se por alcançarem uma considerável redução de ruído, preservando as bordas e outros detalhes de interesse presentes na imagem. Nas próximas seções são mostrados alguns exemplos de filtros baseados nas técnicas não lineares.

### 2.5.6.1. Filtro condicional de média aritmética

A **Figura 2.10** mostra um algoritmo simples para realizar suavização ou filtragem passa-baixas de uma imagem contaminada com ruído aleatório. Esta técnica examina sequencialmente cada pixel e se a diferença entre o valor do nível de cinza do pixel e o valor médio do nível de cinza de seus *pixels* vizinhos é maior que um valor limiar  $U$  (este limiar depende da aplicação), o valor de nível de cinza é substituído pelo valor médio do nível de cinza de seus vizinhos. Caso contrário, o valor de nível de cinza desse pixel fica inalterado. O tamanho da janela deve ser escolhido cuidadosamente, pois em caso de ser muito grande pode introduzir um borramento apreciável da imagem. O valor limiar  $U$  pode ser determinado utilizando alguma informação do ruído que contamina a imagem, por exemplo, pode ser utilizada como informação a variância do ruído.

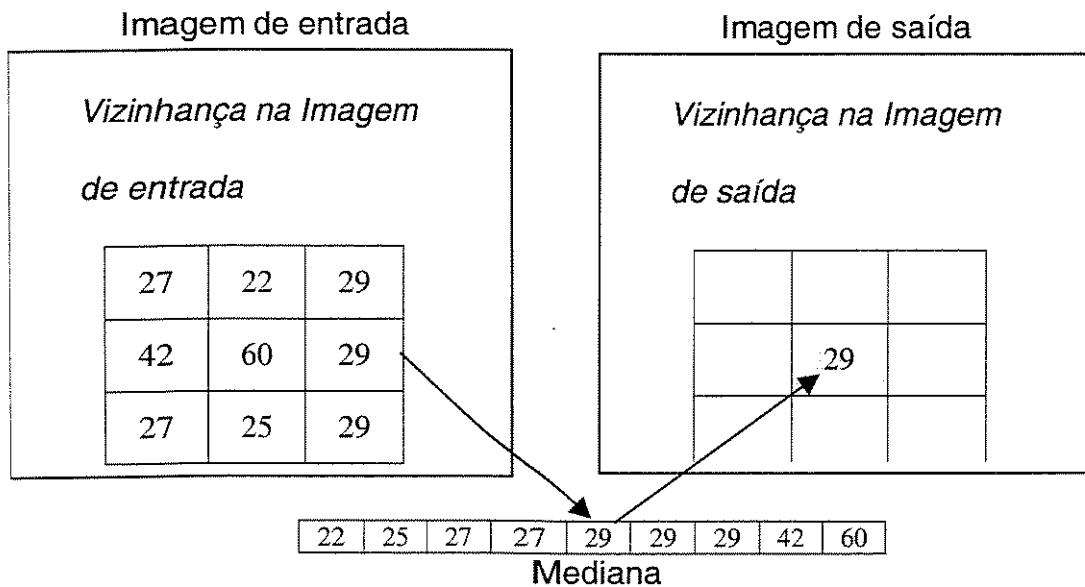


$$PS = \begin{cases} \frac{1}{8} \sum_{i=1}^8 P_i & \text{se } \left| P - \frac{1}{8} \sum_{i=1}^8 P_i \right| > U \\ P & \text{se } \left| P - \frac{1}{8} \sum_{i=1}^8 P_i \right| \leq U \end{cases}$$

**Figura 2.10.** Exemplo de um algoritmo de suavização utilizando um filtro condicional de média aritmética.

### 2.5.6.2. Filtro mediano

A filtragem mediana é uma operação não-linear útil para reduzir ruído impulsivo e do tipo "sal e pimenta" (*salt & pepper*), muitas vezes presente nas imagens. O **filtro mediano** utiliza os valores dos *pixels* contidos em uma vizinhança de tamanho ímpar para determinar o novo valor do pixel de interesse. A **Figura 2.11** apresenta o procedimento utilizado na filtragem mediana

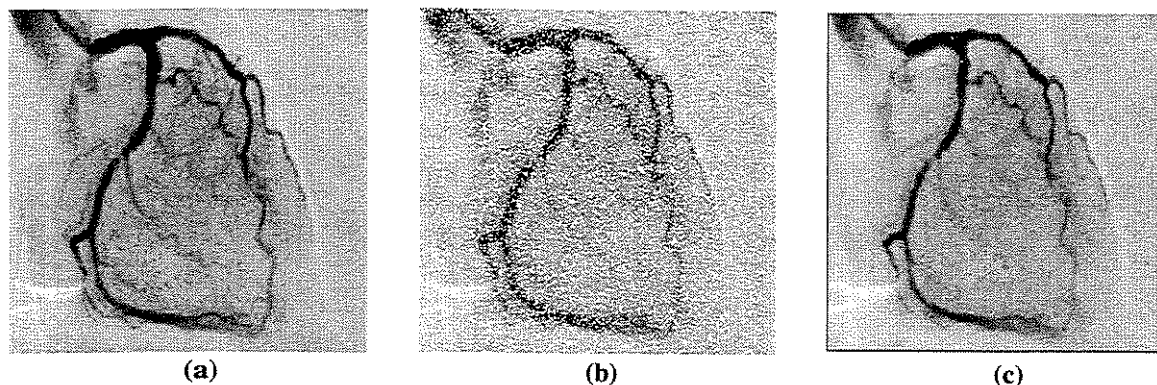


**Figura 2.11.** Ilustração do procedimento utilizado pelo filtro mediano.

O procedimento utilizado pelo filtro mediano consiste em classificar em ordem crescente todos os valores dos níveis de cinza dos *pixels* contidos na janela e substituir o pixel localizado no centro da vizinhança pelo valor do nível de cinza do pixel mediano logo depois da classificação. Isto é, se ocorre uma sequência discreta de tamanho  $N$  ímpar o valor do nível de cinza mediano desta sequência é o valor em que  $(N-1)/2$  valores de níveis de cinza são menores ou pelo menos iguais a ele e  $(N-1)/2$  valores de níveis de cinza são maiores. Por exemplo, em uma vizinhança  $3 \times 3$ , valor do nível de cinza mediano é o 5º maior valor, em uma vizinhança de  $5 \times 5$ , valor do nível de cinza mediano é o 13º maior valor, e assim por diante. Quando ocorrer de vários valores de nível de cinza em uma

vizinhança serem iguais, todos estes valores são agrupados. O custo computacional do filtro mediano está na classificação do arranjo dos valores de níveis de cinza em ordem crescente. Portanto, é muito importante utilizar algoritmos otimizados para realizar esta tarefa.

A **Figura 2.12** mostra o resultado de uma imagem após de passar por um filtro mediano  $5 \times 5$ .



**Figura 2.12.** (a) Imagem original, (b) Imagem contaminada com ruído impulsivo o filtro mediano, (c) Imagem depois de passar pelo filtro mediano  $5 \times 5$

#### **2.5.6.3. Filtros Seletores de Posição (RS)**

Tradicionalmente a restauração e a filtragem dos sinais tem sido abordadas através de métodos lineares, os quais tendem ser sensíveis ao **HTN** (*Heavy Tailed Noise*). Os filtros lineares também tendem a ser sensíveis às características não estacionárias em imagens.

Quando se aplica uma filtragem linear em imagens, geralmente junto com o ruído são minimizadas ou borradas inadmissivelmente bordas e estruturas da imagem. Tal limitação dos filtros lineares levou ao desenvolvimento de métodos não lineares de filtragem. O **filtro seletor de posição** “*Rank Selection filter (RS)*” ou de **ordem estatística** “*order statistic filter*”, coloca em sua saída uma ordem estatística de um conjunto de amostras observadas em sua entrada [Barner, 1997], [Paez, 1994].

#### **Definição 2.4.**

Dado  $N$  observações  $x_1, x_2, \dots, x_N$  de uma variável aleatória  $x$ , as ordens estatísticas são obtidas ordenando de forma ascendente o conjunto  $\{x_i\}$ . Isto produz um conjunto  $\{x_{(i)}\}$  satisfazendo:  $x_{(1)} \leq x_{(2)} \leq \dots \leq x_{(N)}$ . O conjunto  $\{x_{(i)}\}$  são as **ordens estatísticas** de  $N$  observações.

Um dos primeiros filtros não – lineares propostos foi o filtro mediano [Gonzalez, 1999]. Este filtro é bem conhecido pela sua capacidade de suprimir ruído do tipo impulsivo, preservando as bordas da imagem. Razão pela qual, ele é amplamente utilizado em aplicações de processamento de imagens. Porém, em muitas aplicações, uma das dificuldades do filtro mediano está no fato de remover estruturas do sinal, isto ocorre principalmente quando as janelas utilizadas possuem um comprimento grande. Muitas generalizações do filtro mediano têm sido propostas com a finalidade de ganhar melhoria no desempenho deste filtro. Exemplos destas propostas são :

**Filtro Mediano de Múltiplos Estágios** [Yin, 1997], **Mediano Ponderado Central (CWM)**, **Mediano Ponderado Geral (WM)** [Michael, 1996], **Filtro de Ordem Estatística Ponderada (WOS)** [Neuvo, 1994] e **Filtros de Permutações (PF)** [Hardie, 1996].

Todos estes filtros podem ser formulados como filtros **RS**, porém, eles diferem na informação que usam para selecionar a ordem estatística de saída. A tendência destes filtros **RS** mais sofisticados é ter uma melhor característica de preservação dos detalhes da imagem que o filtro mediano.

As principais vantagens do método de filtragem **RS** sobre os métodos lineares tradicionais, são:

- Os filtros **RS** tendem a preservar melhor as bordas.
- O efeito *outliers* é minimizado.

*Outliers* são elementos de um conjunto com valores pouco comuns ou distintos dos outros elementos do conjunto. Por exemplo, no conjunto dos seres humanos adultos pessoas com 2,15 metros ou com 1,20 metros de estatura podem ser considerados *outliers*. No caso de uma imagem 512x512 com 256 níveis de cinza, contaminada com ruído impulsivo, um pixel com valor igual a 255 em determinadas condições pode ser considerado um *outlier* dentro da sua vizinhança.

A preservação das bordas resulta do fato que a saída dos filtros *RS* é sempre uma das amostras que se encontram dentro da janela de observação. Assim, o processo de filtragem introduz novos pontos intermediários ou de transição. Isto possibilita que o filtro *RS* preserve as bordas agudas. Por outro lado, a habilidade destes filtros de limitar o efeito dos *outliers*, deriva da natureza da seleção dos dados ordenados. Em *HTN* (*heavy tailed noise*), os *outliers*, tendem a se localizar nas posições extremas dos dados ordenados, portanto, não selecionando como saída as amostras das posições extremas, os filtros *RS* podem fornecer uma estimativa robusta que é insensível ante qualquer nível alto de *HTN*, [Hardie, 1994].

## 2.6. Considerações finais sobre o capítulo

Este capítulo apresentou vários conceitos básicos relacionados com: as imagens digitais, a filtragem espacial, e algumas notações utilizadas ao longo deste trabalho. A primeira seção mostrou a representação de imagens digitais. A segunda seção apresentou um modelo simples de uma imagem. A terceira seção descreveu brevemente os conceitos de amostragem e quantização. A quarta seção mostrou alguns relacionamentos básicos entre *pixels*. E finalmente na quinta seção foi apresentado várias técnicas básicas de processamento de imagens digitais.

## CAPÍTULO 3. FILTRO CONDICIONAL SELETOR DE POSIÇÃO “RCRS”.

---

Este capítulo introduz-se a classe de filtros **RCRS**. Inicia-se com um preâmbulo onde é realizada uma introdução sobre este tipo de filtragem. Logo na segunda seção mostra-se a estrutura dos filtros **RCRS**. Na terceira seção apresenta-se a definição dos filtros **RCRS**. Após a definição do filtro, na quarta seção mostra-se a relação dos filtros **RCRS** com outras classes de filtros. Na quinta seção apresenta-se o processo de otimização sobre a classe de filtros **RCRS**.

### 3.1. Introdução

Os filtros condicionais seletores de posição (**RCRS**) são desenvolvidos dentro de uma área mais geral que corresponde aos **filtros seletores de posição “Rank Selection (RS)”**. Outros filtros, por exemplo, o **filtro mediano**, o **filtro mediano ponderado central “CWM”** (*center weighted median filter*), o **filtro de ordem estatística ponderada “WOS”** (*weighted order statistic filter*), e baseados na seleção de uma ordem estatística podem ser formulados através dos filtros **RS**, a única diferença destes filtros está no tipo de informação utilizada para decidir qual ordem estatística (amostra) deve ser selecionada como saída [Himayat, 1994]. O tipo de informação usada nos filtros **RCRS** para determinar qual amostra será colocada como saída, corresponde ao posicionamento das amostras de entrada. Devido a isto eles são chamados **filtros condicionais seletores de posição**.

O número de amostras selecionadas de entrada dentro da janela de observação determina a ordem do filtro **RCRS**. Esta ordem pode variar de zero (0) até o número de amostras ( $N$ ) na janela de observação. Devido a isso, o filtro **RCRS** possui uma apreciável flexibilidade na sua implementação. Os filtros de baixa ordem podem oferecer um bom desempenho, além disso, são relativamente simples de otimizar e implementar, mas quando é necessário um maior desempenho do filtro a ordem precisa ser aumentada, isto consequentemente aumenta a complexidade do filtro.



### 3.2. Estrutura dos Filtros Seletores de Posição

Antes de apresentar a estrutura geral dos filtros RS, é definida a notação que será usada neste capítulo.

Supondo que as seqüências discretas  $\{d(n)\}$  e  $\{x(n)\}$ , as quais representam respectivamente as versões desejada e corrompida de um sinal,  $n$  é um vetor de  $v$  elementos tal que  $n = [n_1, n_2, \dots, n_v]$ . Portanto, ambas as seqüências  $\{d(n)\}$  e  $\{x(n)\}$  são  $v$  dimensionais [Hardie, 1994].

Considerando uma função janela de dimensão  $v$  com  $N$  amostras passando sobre uma seqüência corrompida em uma fração pré – determinada. Em cada localização  $n$ , as  $N$  amostras observadas dentro da janela podem ser indexadas e escritas como um vector, da seguinte maneira:

$$\mathbf{x}(\mathbf{n}) = [x_1(\mathbf{n}), x_2(\mathbf{n}), \dots, x_N(\mathbf{n})] \quad (3.2-1)$$

Definindo também o vetor  $\mathbf{x}^T(\mathbf{n})$  com  $N$  amostras observadas e arranjadas em ordem crescente, da seguinte maneira:

$$\mathbf{x}^T(\mathbf{n}) = [x_{(1)}(\mathbf{n}), x_{(2)}(\mathbf{n}), \dots, x_{(N)}(\mathbf{n})] \quad (3.2-2)$$

onde

$$x_{(1)}(\mathbf{n}) \leq x_{(2)}(\mathbf{n}) \leq \dots \leq x_{(N)}(\mathbf{n})$$

Para relacionar a posição da amostra com a localização dentro da janela, define-se  $r_i(\mathbf{n})$  como a posição da amostra na localização  $i$  dentro da janela, isto é,  $x_i(\mathbf{n}) \equiv x_{(r_i(\mathbf{n}))}(\mathbf{n})$ . Adicionalmente fazemos  $\mathbf{r}(\mathbf{n}) = [r_1(\mathbf{n}), r_2(\mathbf{n}), \dots, r_N(\mathbf{n})]$ .

A partir de um conjunto de amostras observadas, pretende-se formar uma estimativa da amostra desejada na localização  $\delta$  dentro da janela, isto é, uma estimativa de  $d_\delta(\mathbf{n})$ . Esta estimativa é denominada  $\hat{d}_\delta(\mathbf{n})$  onde  $1 \leq \delta \leq N$ . Pela definição, a saída  $\hat{d}_\delta(\mathbf{n})$  do filtro RS é uma ordem estatística do vetor observação  $\mathbf{x}(\mathbf{n})$ . Para simplificar a notação, o índice  $n$  é assumido e usado explicitamente somente para fornecer uma maior compreensão.

### **Definição 3.1.**

*A saída de um filtro seletor de posição com janela de tamanho  $N$ , é:*

$$F_{RS}(x) = x_{(S(z))} \quad (3.2-3)$$

*onde*

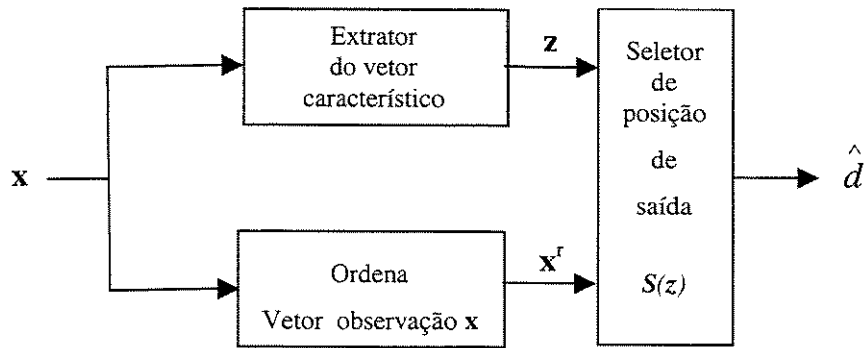
*$z$  é o vetor característico ligado ao espaço característico  $Z$ , e*

$$S: Z \mapsto \{1, 2, \dots, N\}$$

A decisão de qual das amostras do vetor  $\mathbf{x}^T$  será considerada como saída, depende do vetor característico  $z$ . Este vetor representa um subconjunto da informação contida em  $\mathbf{x}$ . Em geral, toda a informação contida no vetor de observação  $\mathbf{x}$  não é usada como base para determinar qual amostra selecionada deve ser considerada como saída.

A otimização sobre uma classe de filtro que usa toda a informação contida no vetor observação  $\mathbf{x}$  é impraticável, ou quase impossível, mesmo que as janelas usadas sejam de tamanho pequeno. Portanto é necessário extrair a informação mais relevante do vetor  $\mathbf{x}$  de acordo com a aplicação. Se for selecionado um espaço característico com uma dimensão suficientemente baixa, então a otimização é realizável [Hardie, 1994].

A **Figura 3.1** mostra o diagrama de blocos da estrutura do filtro seletor de posição.



**Figura 3.1.** Diagrama de blocos da estrutura do filtro seletor de posição

O diagrama de blocos da **Figura 3.1** mostra o vetor observação  $x$  entrando em dois blocos funcionais. Um desses blocos extrai o vetor característico  $z$ , enquanto o outro bloco produz o vetor ordenado  $x^r(n)$ . O seletor de posição na saída escolhe a amostra apropriada do vetor  $x^r(n)$  que será estimada. A forma de como qual amostra é selecionada como saída, baseia-se no vetor característico  $z$  e uma regra  $S(\cdot)$  (seletor de posição).

Através da *Definição 3.1* nota-se que  $S(\cdot)$  pode ser considerado um classificador que divide o espaço característico  $Z$  em  $N$  regiões. Cada região no espaço dividido corresponde com uma posição específica que será selecionada como saída. Sempre e quando o vetor característico  $z$  está vinculado com a  $i$ -ésima partição, a saída do filtro é a  $i$ -ésima ordem estatística.

Dado um espaço característico  $Z$ , a função  $S(\cdot)$  pode ser encontrada usando uma variedade de técnicas de classificação tradicionais. Para certos espaços característicos podem ser derivadas técnicas de otimização que minimizam o erro de estimação do filtro em correspondência com uma medida específica de distorção quantitativa. O desempenho dos filtros  $RS$  depende em grande parte da seleção do espaço característico.

Outros filtros como: o **filtro mediano**, o **filtro mediano ponderado central** “*CWM*” (*center weighted median filter*), o **filtro de ordem estatística ponderada** “*WOS*” (*weighted order statistic filter*), podem ser formulados como filtros  $RS$ . A diferença entre estes filtros está na escolha do espaço caraterístico e no classificador. Talvez o filtro mais

simples que se pode descrever nesta ampla classe de filtros *RS*, é o filtro que a saída coloca uma *ordem estatística constante*, ou seja, coloca na saída a amostra que está em uma determinada posição dentro da janela de observação. Neste caso, o classificador  $S(.)$  é simplesmente uma constante e não é função do espaço característico  $Z$ , assim,  $S(.) = k$ . Em classe de filtros inclui-se o *filtro mediano*.

O filtro mediano é bastante efetivo na supressão de **HTN** (*Heavy Tailed Noise*), e preserva ao mesmo tempo as bordas. Porém, em muitas aplicações, este filtro remove a estrutura do sinal (imagem). Além disso, oferece uma flexibilidade muito baixa na preservação dos detalhes e na suavização do ruído.

Uma segunda classe de filtros *RS* são os **filtros medianos ponderados centrais CWM**. Esta classe de filtros permite um maior controle na preservação dos detalhes e na suavização do ruído. É mostrado mais adiante neste capítulo que o vetor característico para os filtros *CWM* contém a posição parcial da amostra central da janela de observação. Embora os filtros *CWM* tenham mostrado um desempenho muito bom nas aplicações de restauração de imagens[Hardie, 1996], é bom dizer que os filtros *RCRS* que usam a informação completa da posição das amostras selecionadas possuem um melhor desempenho na restauração de imagens.

Como exemplo final, são considerados os **filtros de permutações (FP)** [Arce, 1994]. Esta classe de filtros engloba os filtros *RS*. Nos filtros *FP*, o vetor característico  $r$ , está relacionado com a ordem da posição e a posição temporal de cada amostra dentro da janela. Devido à informação contida nesse vetor, os filtros *FP* são altamente especializados. A seleção do vetor característico é particularmente efetiva nas aplicações onde é necessário a seleção de frequências.

A desvantagem do uso do  $r$  como vetor característico é o rápido crescimento do espaço característico em função do tamanho da janela. A cardinalidade do espaço característico dos filtros de permutações cresce com  $N!$ , sendo atualmente impraticável o uso de janelas que contêm mais de nove amostras. A classe dos filtros *RCRS* considerada neste trabalho é

um enlace entre os filtros de *primeira* ordem estatística ( $1^{th}$ -order *RCRS filters*) e os filtros de permutações (de *N-ésima* ordem “*Nth-order filters*”). [Hardie 1996]. Utilizando vetores característicos de dimensões reduzidas, o tamanho da janela dos filtros *RCRS* pode ser maior do que o tamanho da janela dos filtros *FP*. Os filtros *RCRS* usam a posição ou a ordem das amostras escolhidas, como base para a seleção da posição ou a ordem estatística de saída.

### 3.3. Definição do Filtro Condicional Seletor de Posição (RCRS)

O vetor característico  $z$  dos filtros condicionais seletores de posição (*RCRS*), contém as posições das amostras selecionadas no vetor observação  $x$ . As amostras selecionadas podem ser colocadas em outro vetor da seguinte maneira:  $x^* = [x_{\gamma_1}, x_{\gamma_2}, \dots, x_{\gamma_M}]$ , onde  $M$  é denominado a ordem do filtro *RCRS*, o qual toma valores no intervalo  $0 \leq M \leq N$ . As posições das amostras selecionadas (no vetor  $x^*$ ), constituem o vetor característico  $z$ , que pode ser escrito como:  $z = r^* = [r_{\gamma_1}, r_{\gamma_2}, \dots, r_{\gamma_M}]$ .

O espaço característico é definido pela expressão:  $Z = \Omega_M$ ,  
onde:

$$\Omega_M = \{[i_1, i_2, \dots, i_M] : i_j \in \{1, 2, \dots, N\}, e, i_j \neq i_k \forall j \neq k\}$$

Deste modo o espaço característico contém todas as combinações das posições, excluindo as duas combinações iguais pois essas combinações não podem ocorrer [Hardie, 1994]. Os filtros *RCRS* são definidos especificamente a seguir:

#### Definição 3.2.

A saída de um filtro *RCRS* de ordem  $M$  com janela de tamanho  $N$  é encontrada pela seguinte equação:

$$F_{RCRS}(x) = x_{(S(r^*))} \quad (3.3-1)$$

onde

$$\mathbf{r}^* = [r_{\gamma_1}, r_{\gamma_2}, \dots, r_{\gamma_M}], 0 \leq M \leq N$$

e

$$S : \Omega_M \mapsto \{1, 2, \dots, N\}$$

A ordem  $M$  do filtro **RCRS**, e a localização das amostras escolhidas no vetor  $\mathbf{x}^*$  dependem da aplicação. Para  $M = 1$ , a operação do filtro é relativamente simples de se implementar e otimizar. Esse tipo de filtro possui um bom desempenho em muitas aplicações. Se for necessário um maior desempenho do filtro **RCRS** devido a altos níveis de ruído, é necessário somar ao vetor característico uma informação adicional. O teorema a seguir especifica o número de filtros na classe de filtros **RCRS**.

### **Teorema 3.1**

*O número máximo (cardinalidade) da classe de filtros RCRS de ordem  $M$ , com janela de tamanho  $N$ , é:*

$$N^{(N!/(N-M)!)} \quad (3.3-2)$$

onde

$$0 \leq M \leq N.$$

*Prova:*

Deve ser notado que a cardinalidade do espaço característico  $\Omega_M$ , denotada como  $|\Omega_M|$ , é:

$$|\Omega_M| = N(N-1)(N-2) \dots (N-M+1) = N! / (N-M)!$$

Para cada  $\mathbf{r}^* \in \Omega_M$ , o domínio de  $S(\mathbf{r}^*)$  é  $\{1, 2, \dots, N\}$ . Deste modo para cada  $\mathbf{r}^* \in \Omega_M$ , existem  $N$  escolhas distintas para  $S(\mathbf{r}^*)$ . Consequentemente o número de filtros distintos é  $N^{|\Omega_M|}$ , que é equivalente com a equação (3.3-2). Quando  $M = 1$  existem  $N^N$  filtros **RCRS**.

Este número cresce rapidamente com a ordem do filtro. Se  $M = N$  os filtros **RCRS** são equivalentes aos filtros **FP** e o número de filtros dentro desta classe cresce até  $N^N$  [Hardie 1996]. A implementação de filtros de permutações (**FP**) de janelas com tamanhos grandes (por exemplo  $N > 21$ ) é atualmente impraticável.

Apesar de não ser necessário, geralmente é útil incluir  $x_\delta$  como um elemento no vetor  $\mathbf{x}^*$  desde que  $x_\delta$  seja a amostra da estimativa. A amostra de entrada  $x_\delta$  geralmente provê a informação mais relevante da estimativa  $\hat{d}_\delta$ .

Em um filtro **RCRS** onde  $M = 1$  e  $\mathbf{x}^* = x_\delta$ . O vetor característico é  $\mathbf{r}^* = r_\delta$  e o filtro é representado pela função  $S(r_\delta)$ . Este filtro é um caso particular dos filtros **RCRS**. Se  $k \leq r_\delta \leq N - k + 1$  a saída do filtro **RCRS** é a amostra  $x_\delta$ . Caso contrário, a saída do filtro é a mediana. Devido a isto, este filtro é chamado de **filtro condicional mediano (RCM)**. A Figura 3.2 mostra o exemplo da função  $S(r_\delta)$  deste filtro.

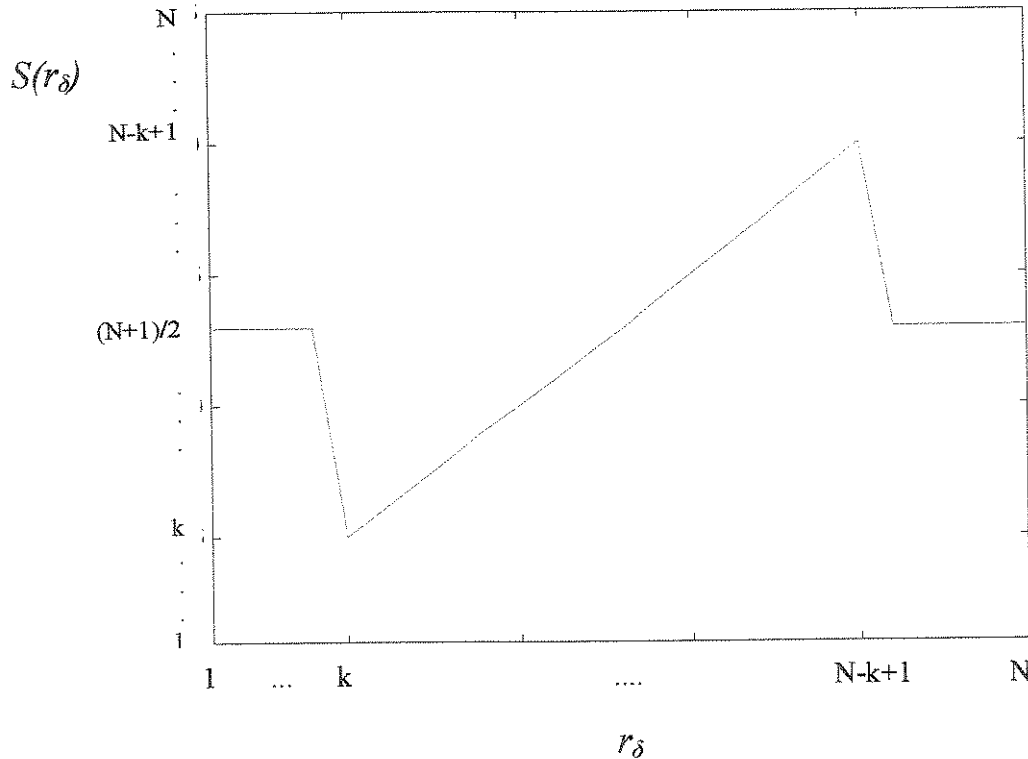


Figura 3.2. Função  $S(r_\delta)$  do filtro **RCM** com parâmetro  $k$ .

Representar graficamente a função  $S(\mathbf{r}^*)$  para  $M = 1$  e  $M = 2$  pode ser uma poderosa ajuda nas análises do funcionamento dos filtros **RCRS**.

Da **Figura 3.2** mostra que o filtro **RCM** suprime o **HTN**. Após serem ordenadas as amostras, se  $x_\delta$  ocupa uma das posições do centro da janela, a saída do filtro é a própria amostra  $x_\delta$ . Entretanto se  $x_\delta$  ocupa uma das posições extremas, a saída do filtro será a mediana.

### Exemplo 3.1

Supondo um filtro **RCM** com  $k = 2$  e com uma janela com tamanho  $5 \times 1$ .

A saída do filtro **RCM** é a amostra mediana do conjunto de observação se a amostra central é a maior ou a menor amostra na janela de observação. Em qualquer outro caso a saída é a amostra central.

Considerando  $\mathbf{x} = [2,3,1,4,5]$ . A saída do filtro **RCM** em  $\mathbf{x}$  é “3”, onde a amostra central é a amostra mínima no conjunto observado. A função  $S(r_\delta)$  do filtro **RCM** mostrada na **Figura 3.2** para os valores :  $k = 2$  e  $N = 5$ , é,

$$S(r_\delta) = \begin{cases} r_\delta & \text{se } 2 \leq r_\delta \leq 4 \\ mediana & \text{se } r_\delta < 2 \text{ ou } r_\delta > 4 \end{cases} \quad (3.3-3)$$

Como o vetor característico ( $\mathbf{r}^* = r_\delta$ ), é a posição da amostra central ( $x_\delta = x(3) = 1$ ) no vetor ordenado  $\mathbf{x}^r = [1,2,3,4,5]$ , isto é :  $\mathbf{r}^* = r_\delta = 1$ , portanto, pode-se calcular  $S(\mathbf{r}_\delta)$ , utilizando a equação (3.3-3) obtendo-se  $S(1) = 3$ . Substituindo este valor na equação (3.3-1), O filtro **RCM** considera como saída a amostra que está na posição 3. O mesmo ocorre quando  $x_\delta = x(3) = 5$ .

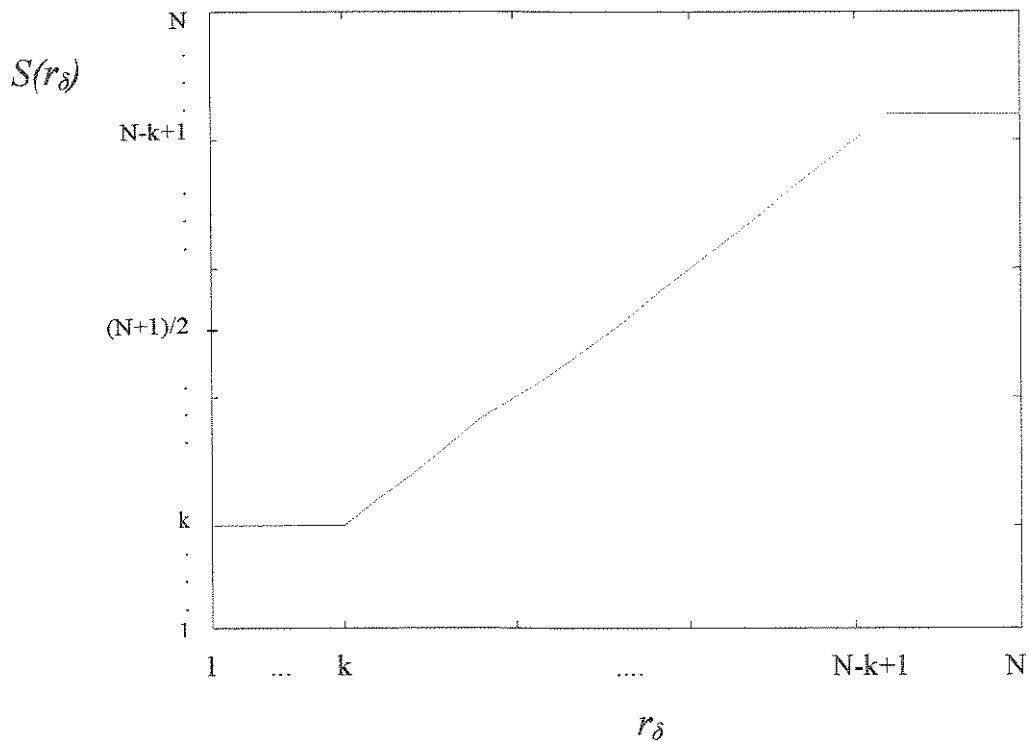


### 3.4. Relação entre os Filtros RCRS e outras classes de filtros

Nesta seção é examinada a relação entre os filtros *RCRS* e os vários tipos de filtros baseados na seleção de posição. Primeiro é considerado a relação dos filtros *RCRS*, com o **filtros medianos ponderados centrais** (*CWM*), [Yu, 1994]. A saída do filtro *CWM* é definida pela mediana sobre um conjunto estendido que inclui múltiplas amostras centrais. Esta operação pode ser escrita como:

$$F_{CWN}(x) = \text{median}\{x_1, \dots, w_\delta \diamond x_\delta, \dots, x_N\} \quad (3.4-1)$$

Onde  $\diamond$  é um operador de replicação e  $x_\delta$  representa a amostra central na janela de tamanho  $N$  (ímpar). A função  $S(r_\delta)$  do filtro *CWM* com parâmetro  $k$ , é mostrada na **Figura 3.3**.



**Figura 3.3.** Função  $S(r_\delta)$  do filtro *CWM* com parâmetro  $k$ .

A amostra central é repetida  $w_\delta$  vezes, onde  $w_\delta$  é um número inteiro, positivo, ímpar, e diferente de zero. Quando  $w_\delta = 1$ , o operador é um filtro mediano, e para  $w_\delta \geq N$ , o filtro

*CWM* é reduzido a um operador identidade. É demonstrado por S. J. Ko em [Ko, 1991], que a operação do filtro *CWM* equivale a:

$$F_{CWM}(x) = \text{median}\{x_{(k)}, x_{\delta}, , x_{(N-k+1)}\} \quad (3.4-2)$$

onde:

$$k = (N + 2 - w_{\delta})/2 \text{ para } 1 \leq w_{\delta} \leq N, \text{ e } k = 1 \text{ para } -w_{\delta} > N$$

Similarmente aos filtros *RCM*, os filtros *CWM* não alteram  $x_{\delta}$  se  $k \leq r_{\delta} \leq N - k + 1$ . Se  $r_{\delta} \leq k$ , a saída do *CWM* é  $x_{(k)}$ . Similarmente se  $N - k + 1 \leq r_{\delta}$ , a saída é  $x_{(N-k+1)}$ .

Um filtro mais geral é o **filtro de ordem estatística ponderada simples (SWOS)**, [Michael, 1996] o qual está definido pela seguinte equação:

$$F_{SWOS}(x) = \text{rank}(v)\{x_1, \dots, w_{\delta} \diamond x_{\delta}, \dots, x_N\} \quad (3.4-3)$$

O filtro *SWOS* seleciona a  $v$ -ésima posição ou ordem estatística, a partir do conjunto estendido de amostras. Como mostrado por G. R Arce em [Arce, 1994], o filtro *SWOS* equivale a:

$$F_{SWOS}(x) = \text{median}\{x_{(k)}, x_{\delta}, , x_{(l)}\} \quad (3.4-4)$$

onde

$$1 \leq l \leq N.$$

A relação entre os parâmetros na equação (3.4-3) e os parâmetros na equação (3.4-4) é determinada pelas seguintes expressões:  $v = l$  e  $w_{\delta} = l - k + 1$ . O **Teorema 3.2** relaciona os filtros *SWOS* e *RCRS*.

### Teorema 3.2

Os filtros *SWOS* são uma sub-classe dos filtros *RCRS* e estão caracterizados pela função:

$$S(r^*) = S(r_\delta) = \begin{cases} l & \text{se } l < r_\delta \\ r_\delta & \text{se } k \leq r_\delta \leq l \\ k & \text{se } r_\delta < k \end{cases} \quad (3.4-5)$$

onde

$$1 \leq k \leq l \leq N$$

*Prova:*

Considerando a equação (3.5-4), para o valor de  $x$  tal que  $r_\delta < k$ ,  $F_{SWOS}(x) = x_{(k)}$ . Se o valor de  $x$  é tal que  $l < r_\delta$ ,  $F_{SWOS}(x) = x_{(l)}$ . Se o valor de  $x$  é tal que  $k \leq r_\delta \leq l$ , então,  $F_{SWOS}(x) = x_\delta = x_{(r_\delta)}$ . Deste modo a operação de seleção da posição do filtro *SWOS* é especificada utilizando a equação (3.4-5).

O Teorema 3.2 mostra que os filtros *SWOS* usam uma informação parcial de  $x_\delta$ . A posição de saída é determinada pela faixa das posições em que  $x_\delta$  está localizada. O desempenho do filtro pode ser melhorado através da utilização de toda a informação contida em  $r_\delta$ .

A classe dos **filtros de ordem estatística ponderada (WOS)**, [Neuvo, 1994], é definida pela suavização da limitação dos filtros *SWOS* que unicamente ponderam uma simples amostra. Os filtros *WOS* ponderam todas as amostras. A saída do filtro *WOS* é determinada pela equação:

$$F_{SWOS}(x) = \text{rank}(v)\{w \diamond x\} \quad (3.4-6)$$

Onde  $w$  é o vetor dos pesos de  $N$  elementos. O  $i$ -ésimo elemento neste vetor  $w_i$  é o peso aplicado à amostra  $x_i$ . Como nos filtros *SWOS*, a saída é o elemento que está na  $v$ -ésima posição ou ordem do conjunto expandido. Através da ponderação de cada uma das amostras

na janela, os filtros **WOS** podem enfatizar ou acentuar certas amostras de observação, enquanto desacentuam outras amostras.

Apesar do esquema de ponderação do filtro **WOS** oferecer flexibilidade em seu projeto e da existência de alguma sobreposição entre as classes de filtros **WOS** e **RCRS**, alguns dos filtros **RCRS** mais interessantes não podem ser realizados como filtros **WOS**. Um exemplo de filtro **RCRS** que não pode ser projetado como filtro **WOS** é o filtro **RCM**. Os filtros **RCRS** e **WOS** requerem que os dados de observação sejam ordenados pela posição que ocupam as amostras na janela de observação. Além das operações requeridas para ordenar os dados, o filtro **WOS** executa em média,  $N/2$  adições e comparações para formar uma estimativa.

As operações adicionais exigidas pelo filtro **RCRS** para formar uma estimativa somente dependem da ordem  $M$  do filtro e não do tamanho da janela  $N$ . Um filtro **RCRS** de ordem  $M$ , requer somente  $M-1$  multiplicações e  $M(M-1)/2$  comparações adicionais. Estas comparações são necessárias quando o filtro **RCRS** gera um índice apropriado dentro de  $N/(N-M)!$  entradas da tabela de dados (*look-up table*) que armazena  $S(.)$ . Deste modo, os filtros **RCRS** de baixa ordem ( $M=1,2$ ), são mais simples de implementar que os filtros **WOS**. Porém, quando incrementa-se a ordem do filtro, o número de operações requeridas por ele para formar uma estimativa cresce além das exigidas pelo filtro **WOS**, e o número de entradas na tabela de dados (*look-up table*) pode-se tornar proibitivamente grande.

### 3.5. Otimização

O procedimento de otimização sobre a classe de filtros **RCRS** descrito aqui segue o utilizado por G. R. Arce [Arce, 1994]. Nesses trabalhos foi detalhado a otimização do filtro de permutação (**FP**) através do Erro Mínimo Normalizado  $L\eta$  (**LNE** - *Least Normed Error*). Para implementar o método de otimização **LNE**, os vetores característicos que compreendem o espaço característico devem ser indexados. Fazendo a indexação, o espaço característico pode ser expresso como:

$$\Omega_M = \{r_1, r_2, \dots, r_{|\Omega_M|}\} \quad (3.5-1)$$

No desenvolvimento precedente [Arce, 1994], a equação (3.5-1) é utilizada para escrever os vetores de observação como uma seqüência que é indexada na ordem que eles são utilizados. Os vetores observação são escritos como  $x(n_1)$ ,  $x(n_2)$ , ..... $x(n_K)$  e as estimativas desejadas correspondentes como  $d(n_1)$ ,  $d(n_2)$ , ..... $d(n_K)$ . Para um filtro **RCRS** definido por  $S(\cdot)$ , o **LNE** sobre uma seqüência de treinamento de K elementos é:

$$\sum_{i=1}^K |d(n_i) - F_{RCRS}(x(n_i))|^\eta = \sum_{i=1}^K |d(n_i) - x_{S(r^*(n_i))}|^\eta \quad (3.5-2)$$

O classificador que minimiza a equação (3.5-2) é chamado “classificador ótimo” e denotado como  $S_{opt}(\cdot)$ . O **LNE** na equação (3.5-2) pode ser particionado ou dividido de acordo com os vetores observação característicos. Considerando  $\alpha_i$  o índice do vetor característico em  $\Omega_M$  correspondente ao vetor de observação  $x(n_i)$  tal que  $r^* = r^*(n_i)$  e definindo  $\Gamma_{j,k} = \{i \in \{1,2,...,K\} : \alpha_i = j\}$ . O **LNE** total da seqüência de treinamento estimando o sinal desejado na  $k$ -ésima ordem estatística, considerando que o vetor  $r_j$  é observado, pode ser escrito como:

$$E_j(k) = \sum_{i \in \Gamma_{j,k}} |d(n_i) - x_{(k)}(n_i)|^\eta \quad (3.5-3)$$

Se para  $j \in \{1,2,...,|\Omega_M|\}$ ,  $\Gamma_{j,k} = \emptyset$ , então define-se  $E_j(k) = 0$  para algum  $k = 1,2, ...N$ . O **LNE** do filtro **RCRS** definido por  $S(\cdot)$  pode ser escrito como uma soma de erros que são divididos de acordo com o vetor característico e obtém-se:

$$\sum_{i=1}^K |d(n_i) - F_{RCRS}(x(n_i))|^\eta = \sum_{j=1}^{|\Omega_M|} E_j(S(r_j)). \quad (3.5-4)$$

É fácil mostrar que o **LNE** na equação (3.5-4) é minimizado se e somente se, cada uma das  $E_j(S(r_j))$  somas de erros é minimizada. Deste modo, o classificador ótimo do filtro **RCRS** é obtido por:

$$S_{opt}(r_j) = k : E_j(k) \leq E_j(l) \quad \forall l \neq k \quad (3.5-5)$$

para  $j = 1, 2, \dots, |\Omega_M|$

Se não existe um único erro mínimo para algum  $j$ , então deve ser utilizada uma regra de desempate (*tie-breaking*). Por exemplo, um empate entre dois valores que satisfazem a equação (3.5-6) pode ser resolvido escolhendo-se a ordem estatística correspondente a um dos erros mínimos que está mais próximo à posição da mediana. Na prática os empates são improváveis dado um número suficiente de amostras de treinamento.

O procedimento do *LNE* sempre retorna o filtro ótimo para um conjunto global de treinamento. Uma pequena norma de erro,  $\eta \leq 1$ , pode ser usada na presença de ruído *HTN*. Em tais casos os *outliers* não são consideráveis, portanto, não dominam a soma dos erros normalizados e os filtros que suavizam muito não são escolhidos. Além disso, para ruído exponencial duplicado (*doubly noise*), a norma  $\eta = 1$  conduz a uma melhor estimativa. Na presença de outros tipos de ruídos onde os *outliers* tem menos probabilidade de aparecer, por exemplo, para ruídos Gaussianos podem ser usadas normas de ordens superiores com bons resultados.

A otimização do filtro também pode ser desenvolvida recursivamente. A função  $S(\cdot)$  é atualizada quando existirem novos vetores de treinamento disponíveis. O Erro Particionado Cumulativo ( $R$ ) para atualizar a função  $S(\cdot)$  é definido como:

$$R_{j,k}(m) = \sum_{i \in \Gamma_{j,m}} |d(n_i) - x_{(k)}(n_i)|^\eta \quad (3.5-6)$$

O Erro Particionado Cumulativo  $R_{j,k}(m)$  contém o erro total ocorrido quando o filtro coloca na saída a  $k$ -ésima ordem estatística, considerando que o vetor característico  $j$  é observado até o índice  $m$  na sequência de treinamento. O erro cumulativo pode ser escrito como um vetor:

$$R_j(m) = \begin{bmatrix} R_{j,1}(m) \\ R_{j,2}(m) \\ \vdots \\ R_{j,N}(m) \end{bmatrix} \quad (3.5-7)$$

Na seqüência de treinamento, a função ótima no índice  $m$ , é determinada pelo elemento mínimo em  $R_j(m)$ , e é obtida por:

$$S_{opt}^m = k : R_{j,k}(m) = \min(R_{j,1}(m), R_{j,2}(m), \dots, R_{j,N}(m)) \quad (3.5-8)$$

para  $j = 1, 2, \dots, \lfloor \Omega_M \rfloor$

Se não existe um único elemento mínimo no vetor  $R_j(m)$ , deve ser considerada uma regra de desempate. O procedimento iterativo da otimização continua como explicado a seguir.

A função ótima é inicializada num valor predeterminado como por exemplo a mediana. Assim  $S_{opt}^0 = (N+1)/2$  ;  $R_{j,k}(0) = 0$  para  $j = 1, 2, \dots, \lfloor \Omega_M \rfloor$  e  $k = 1, 2, \dots, N$ . O índice  $m$  é inicializado com  $m = 1$  e então determina-se o índice do vetor característico  $\alpha_m$ . Em seguida, o vetor de erro cumulativo  $R_{\alpha_m}(m)$  é atualizado pela equação:

$$R_{\alpha_m}(m) = R_{\alpha_m}(m-1) + P(m) \quad (3.5-9)$$

onde:

$P(m)$  é um vetor que contém as  $L_\eta$  diferenças normalizadas entre o sinal desejado  $d(n_m)$  e cada uma das ordens estatísticas no vetor de observação.

O vetor  $P(m)$  é obtido por:

$$P(m) = \begin{bmatrix} |d(n_m) - x_{(1)}(n_m)|^\eta \\ |d(n_m) - x_{(2)}(n_m)|^\eta \\ \vdots \\ |d(n_m) - x_{(N)}(n_m)|^\eta \end{bmatrix} \quad (3.5-10)$$

A função ótima  $S_{opt}^0(r_{\alpha_m})$  é atualizada pela equação:

$$S_{opt}^m = k : R_{\alpha_m, k}(m) = \min(R_{\alpha_m, 1}(m), R_{\alpha_m, 2}(m), \dots, R_{\alpha_m, N}(m)) \quad (3.5-11)$$

O índice  $m$  é incrementado, e o procedimento é repetido até que o final da sequência de treinamento seja alcançado ou quando se determine que o filtro já foi suficientemente treinado (foram utilizadas mais de 2 milhões de amostras). A sequência do algoritmo de treinamento recursivo [Arce, 1994] é:

1. Faça  $m = 1$ ,  $S_{opt}^0(\mathbf{r}_j) = \frac{N+1}{2}$ , e  $R_{j,k}(0) = 0$ , para  $j = 1, 2, \dots, |\Omega_M|$  e  $k = 1, 2, \dots, N$ .
2. Determine o índice do vetor característico  $\alpha_m$ .
3. Atualize  $R_{\alpha_m}(m)$ , pela equação  $R_{\alpha_m}(m) = R_{\alpha_m}(m-1) + P(m)$ .
4. Faça  $S_{opt}^m = k : R_{\alpha_m, k}(m) = \min(R_{\alpha_m, 1}(m), R_{\alpha_m, 2}(m), \dots, R_{\alpha_m, N}(m))$ .
5. Se  $m = K$  ou filtro foi suficientemente treinado, terminar o treinamento, caso contrário, incrementar  $m$  e retornar ao passo 2 do algoritmo.

As vantagens do procedimento de treinamento determinístico são:

- No processo de treinamento sempre retorna o filtro ótimo global para o conjunto de treinamento.
- Existe liberdade para escolher um erro padrão.



Além das vantagens do procedimento de treinamento determinístico, deve-se adicionar um fator exponencial de “esquecimento”  $\lambda$ , na soma dos erros normalizados estimados  $L_\eta$ , para adaptar os dados de treinamento com as mudanças estatísticas do sinal, o que faz o filtro adequado para ambientes não – estacionários. Nesta situação, a soma dos  $LNE$  a ser minimizada é :

$$E_j(k) = \sum_{i=1}^K \lambda^{(M-i)} |d(\mathbf{n}_i) - x_{(k)}(\mathbf{n}_i)|^\eta \quad (3.5-12)$$

As Figuras 3.4 a 3.6 mostram alguns exemplos de função de otimização  $S(.)$ . No treinamento foi utilizada a imagem “Elaine” de 512x512 pixels. Para obter todas as funções ótimas de classificação  $S(.)$  foi utilizada uma janela de tamanho 9x9, onde “ $\delta$ ” é o índice da amostra central da janela de observação. A Figura 3.4 mostra uma função ótima de primeira ordem  $S_{opt}(r_\delta)$  para uma imagem corrompida por ruído impulsivo de diferentes densidades.

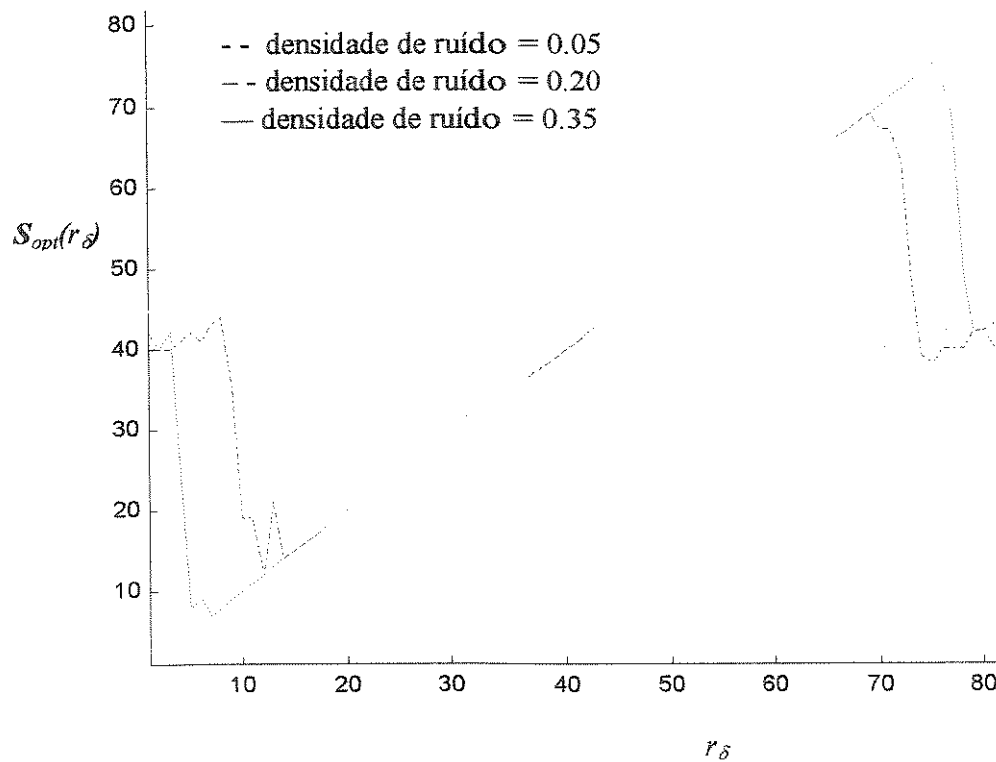
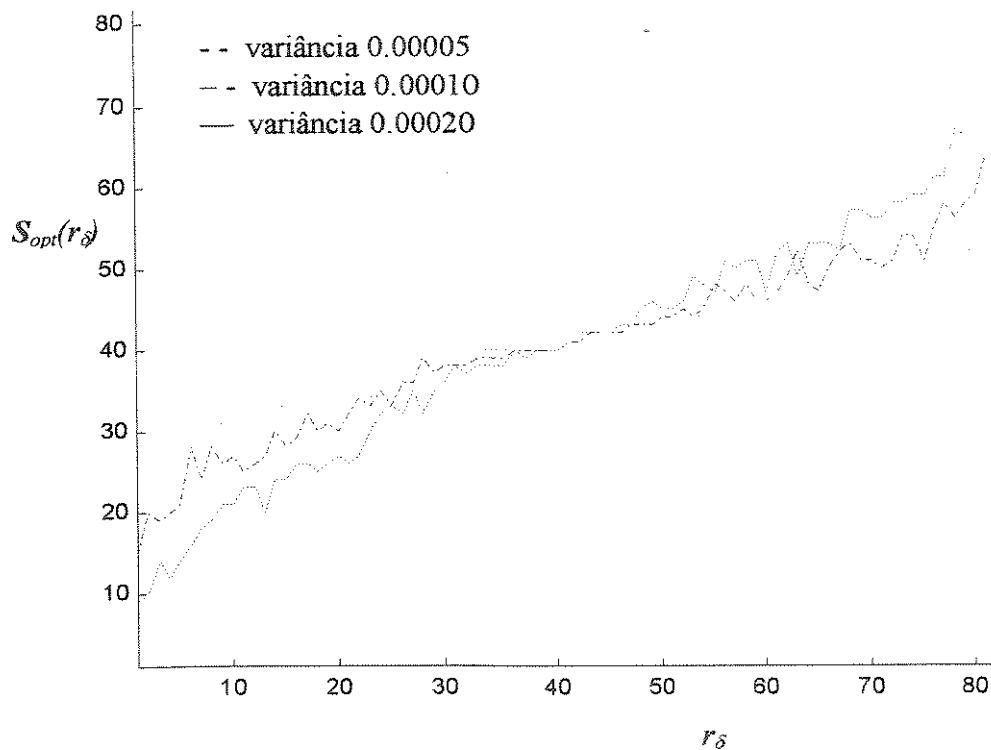


Figura 3.4. Função Ótima  $S_{opt}(r_\delta)$  para um filtro RCRS de janela 9x9 treinado com a imagem “Elaine” contaminada com ruído impulsivo.

Cada uma das funções ótimas da **Figura 3.4** tem uma região linear, em que a posição de entrada é igual a posição de saída. Porém, quando  $r_\delta$  encontra-se nas posições extremas, a saída é uma posição próxima da mediana. Isto possibilita a rejeição do impulso. Além disso, pode-se observar que o ponto de quebra (*breakpoint*) move-se a medida que aumenta a densidade de ruído.

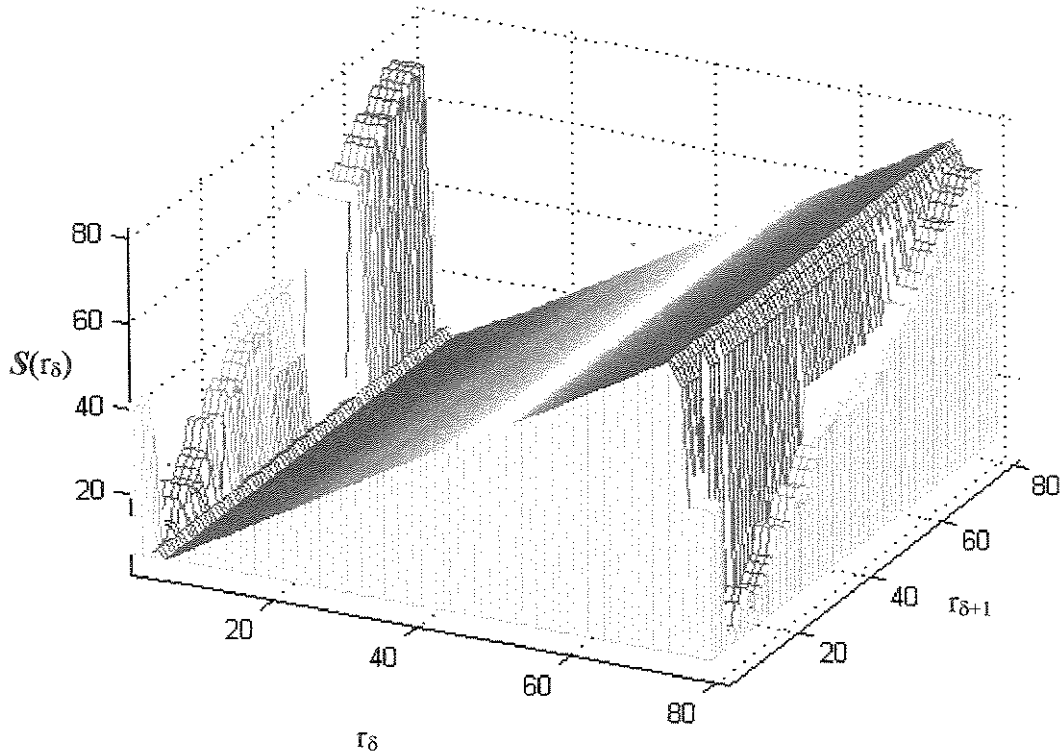
A **Figura 3.5** mostra várias funções otimizadas  $S_{opt}(r_\delta)$  para a imagem “Elaine” corrompida com vários níveis de ruído Gaussiano aditivo.



**Figura 3.5.** Função ótima  $S_{opt}(r_\delta)$  para um filtro RCRS de janela 9x9 treinado com a imagem “Elaine” contaminada com ruído Gaussiano aditivo.

Na **Figura 3.5** as funções ótimas são aproximadamente lineares, com uma inclinação inversamente proporcional ao nível de ruído. Na ausência de ruído, a função ótima possui uma inclinação de 1 que representa o filtro identidade. Quando os níveis de ruído são altos, a inclinação da função ótima aproxima-se de zero, reduzindo a um filtro mediano.

A **Figura 3.6** mostra uma função ótima de segunda ordem com uma janela de  $9 \times 9$ , onde o vetor característico é  $r^* = [r_\delta, r_{\delta+1}]$ , “ $\delta$ ” representa o índice da amostra na posição central da janela de observação e “ $\delta+1$ ” é o índice da próxima amostra à direita da amostra na posição central.



**Figura 3.6.** Função ótima  $S_{opt}(r_\delta)$  para um filtro *RCRS* de janela  $9 \times 9$  treinado com a imagem “*Elaine*” contaminada com ruído impulsivo.

No gráfico da **Figura 3.6** a altura da malha representa a posição de saída do vetor característico considerado. Nessa figura, pode-se observar que para um valor fixo de  $r_{\delta+1}$ , a forma da curva ao longo do eixo  $r_\delta$  é similar à função  $S_{opt}$  de primeira ordem. Porém, a medida que  $r_{\delta+1}$  é incrementado, a função ótima  $S_{opt}(r_\delta, r_{\delta+1})$  considera como saída a ordem estatística mais alta. Para valores pequenos de  $r_{\delta+1}$  a função ótima  $S_{opt}(r_\delta, r_{\delta+1})$  considera como saída a ordem estatística mais baixa. A informação adicional fornecida por  $r_{\delta+1}$

permite aos filtros de segunda ordem possuir regras de decisão mais sofisticadas, que possibilitam um melhor desempenho em comparação com os filtros de primeira ordem [Hardie, 1994].

### 3.6. Considerações finais sobre o capítulo

Este capítulo apresentou a classe de filtros **RCRS**. A primeira seção mostrou um preâmbulo sobre este tipo de filtragem não-linear. A segunda seção descreveu analítica e graficamente a estrutura dos filtros **RCRS**. Na terceira seção foi apresentado a definição dos filtros **RCRS**. A quarta seção mostrou a relação dos filtros **RCRS** com outras classes de filtros. Finalmente, na quinta seção ilustrou-se o processo de otimização e apresentaram-se vários exemplos de funções ótimas para classe de filtros **RCRS**.

## CAPÍTULO 4. RESULTADOS EXPERIMENTAIS

---

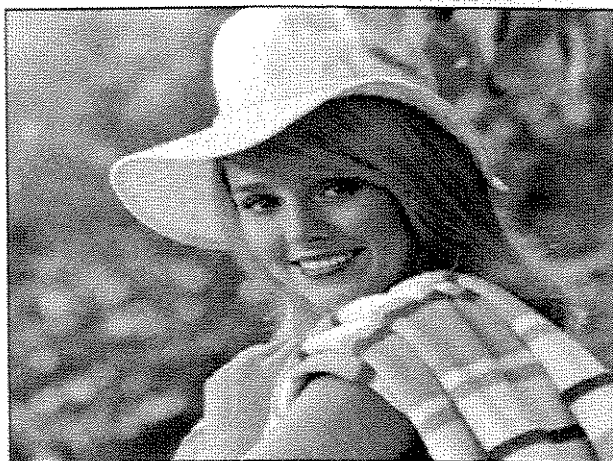
### 4.1. Análise Quantitativa.

Os filtros analisados neste trabalho podem ser utilizados em uma grande variedade de aplicações de restauração de imagens. Nesta dissertação, estes filtros são aplicados na restauração de imagens corrompidas por ruído **impulsivo** do tipo **sal e pimenta** (*salt & pepper*) e ruído **Gaussiano** branco, e aditivo, (ver Anexo 3).

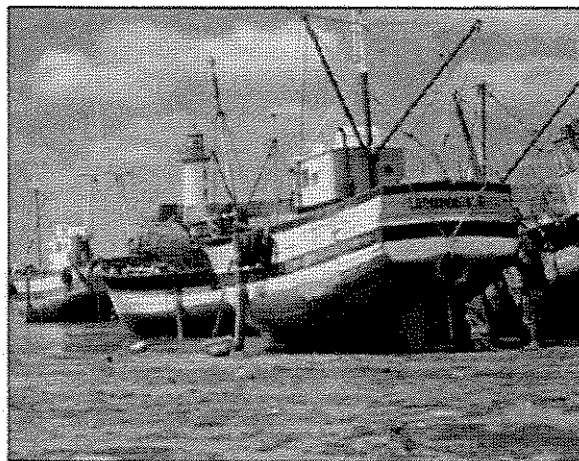
Neste capítulo são apresentados os resultados quantitativos e são mostradas várias imagens filtradas que proporcionam uma avaliação subjetiva. Nos testes foram realizadas as comparações dos resultados obtidos utilizando gráficos com as curvas dos Erros Médios Absolutos (**MAE**), dos filtros **RCRS** e os filtros **CWM**, **RCM**, **SWOS**, **Wiener** e **mediano**.

O filtro **RCRS** utilizado nos testes possui as seguintes características: no filtro denominado **RCRS1**  $M = 1$   $\gamma_1 = \delta$ , onde “ $\delta$ ” indica o índice da amostra na posição central da janela; o filtro denominado **RCRS2**  $M = 2$ ,  $\gamma_1 = \delta$ , e  $\gamma_2 = \delta + 1$ , onde “ $\delta + 1$ ”, indica o índice da amostra do lado direito da amostra central; o filtro denominado **RCRS3**  $M = 3$ ,  $\gamma_1 = \delta$ ,  $\gamma_2 = \delta + 1$ ,  $\gamma_3 = \delta - 1$ , onde  $\gamma_3 = \delta - 1$ , indica o índice da amostra do lado esquerdo da amostra central. O procedimento de treinamento usado para obter a simulação dos resultados é o Erro Mínimo Normalizado  $L_1$  (**LNE - Least Normed Error**) descrito no capítulo 3 deste trabalho. Os filtros **RCM**, **CWM** e **SWOS** ótimos foram encontrados utilizando-se as equações (3.3-3), (3.4-2) e (3.4-5) respectivamente.

A **Figura 4.1** mostra as imagens (a)“*Elaine*”, (c)“*Elaine*” corrompida com ruído impulsivo”, e (b)“*Boat*”. Esta última foi utilizada no processo de treinamento dos filtros **RCRS**, **CWM**, **RCM** e **SWOS**. Pode-se verificar que as imagens “*Elaine*” e “*Boat*” são bastante diferentes em estrutura. A finalidade de utilizar filtros empregando uma imagem diferente da imagem a ser filtrada é para mostrar o desempenho real dos filtros.



(a)



(b)

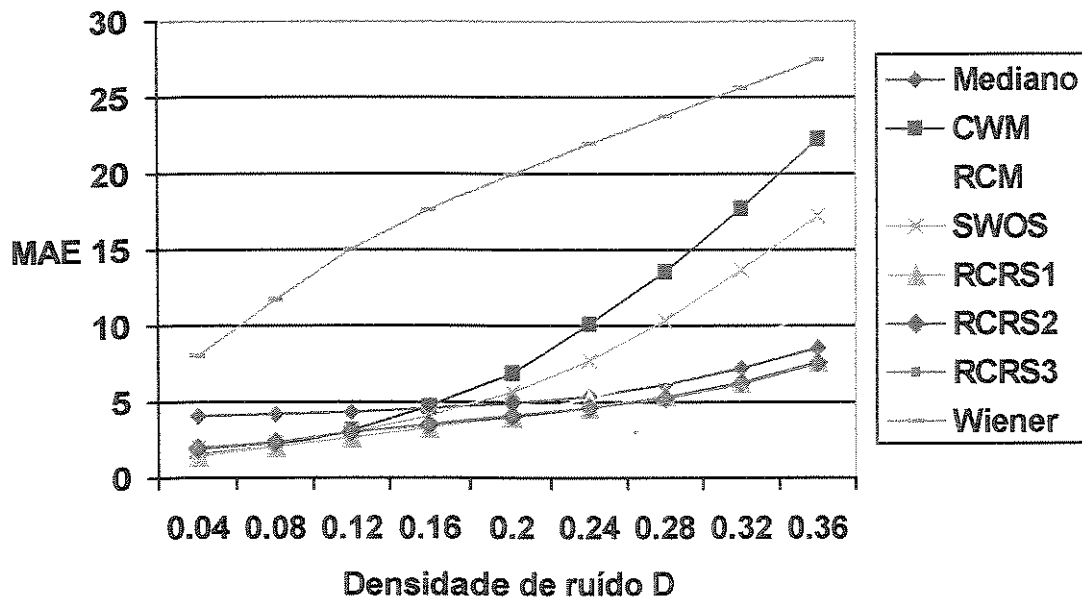


(c)

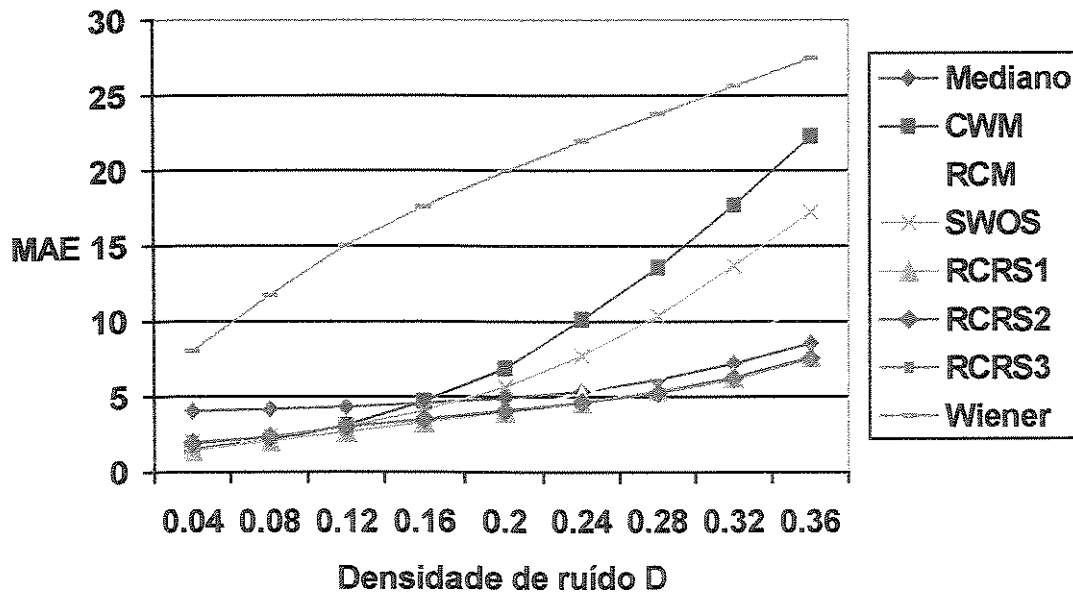
**Figura 4.1.** (a) Imagem original da “Elaine”(teste) de tamanho 512 x 512 representada com 8 bits/pixel, (b) Imagem do “Boat (treino) de tamanho 512 x 512 com 8 bits/pixel, (c) Imagem da “Elaine” corrompida com ruído impulsivo com densidade  $D = 0,20$ .

A **Figura 4.2** mostra o Erro Médio Absoluto (**MAE**) para os filtros **RCRS**, **CWM**, **SWOS**, **RCM**, **Wiener** e **Mediano**, filtrando a imagem “Elaine” corrompida por ruído impulsivo do tipo sal e pimenta com densidade  $D = 0,20$ . A **Figura 4.2(a)** mostra os erros **MAE** dos filtros com janela 3x3. Nesta figura pode-se observar que os valores de erros **MAE** dos filtros **mediano**, **CWM**, **SWOS**, **RCM** e **Wiener** (Matlab), são bastante elevados, e os valores de erro dos filtros **RCRS** são bem menores. Neste último tipo de filtro, o erro **MAE** é menor no filtro **RCRS3** de ordem três ( $M = 3$ ) do que nos filtros **RCRS2** ( $M = 2$ ) e **RCRS1** ( $M = 1$ ). Nesta figura pode-se observar que ao aumentar a densidade do ruído os erros dos filtros **CWM**, **SWOS**, **RCM**, alcançam valores comparáveis aos do filtro mediano,

inclusive estes valores chegam a ser maiores. Porém, os filtros *RCRS* continuam oferecendo valores de erro significativamente baixos.



(a)



(b)

Figura 4.2. O MAE dos filtros *RCRS(1,2,3)*, *mediano*, *CWM*, *SWOS*, *RCM* e *Wiener*, filtrando a imagem “Elaine” corrompida com ruído impulsivo de densidade  $D$ . (a) Resultado do erro *MEA* utilizando a janela  $3 \times 3$ , (b) Resultado do erro *MEA* utilizando a janela  $5 \times 5$ .

A Figura 4.2(b) mostra as curvas dos *MAE* para os filtros *RCRS(1,2,3)*, *mediano*, *CWM*, *SWOS*, *RCM* e *Wiener* com janela de tamanho  $5 \times 5$ . Pode-se observar nesta figura que os erros dos filtros *RCRS* são significativamente menores que os erros dos filtros *CWM*, *SWOS* e *mediano*. Além disso, se observa que os erros dos filtros *RCRS* para tamanho de janela  $5 \times 5$  são menores do que para janela de tamanho  $3 \times 3$ . Isto mostra que ao aumentar o tamanho da janela de observação aumenta-se a relação sinal – ruído, (ver Anexo 1).

A Figuras 4.3 mostra as curvas de erros *MAE* para os filtros *RCRS* de primeira e segunda ordem. Em uma das curvas o filtro foi treinado com a imagem “Boat” e filtrou a própria imagem com ruído. Na outra curva corresponde ao filtro treinado com a imagem “Boat” e filtrou a imagem da “Elaine”.

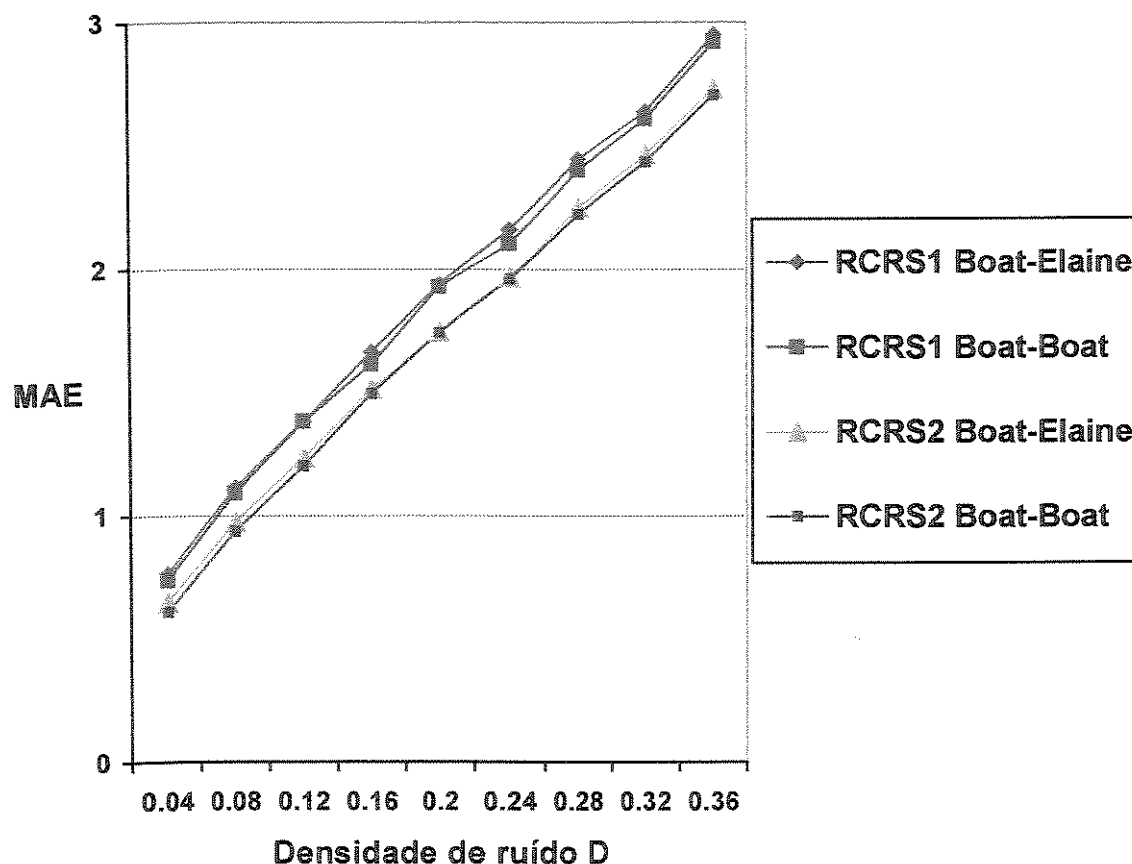


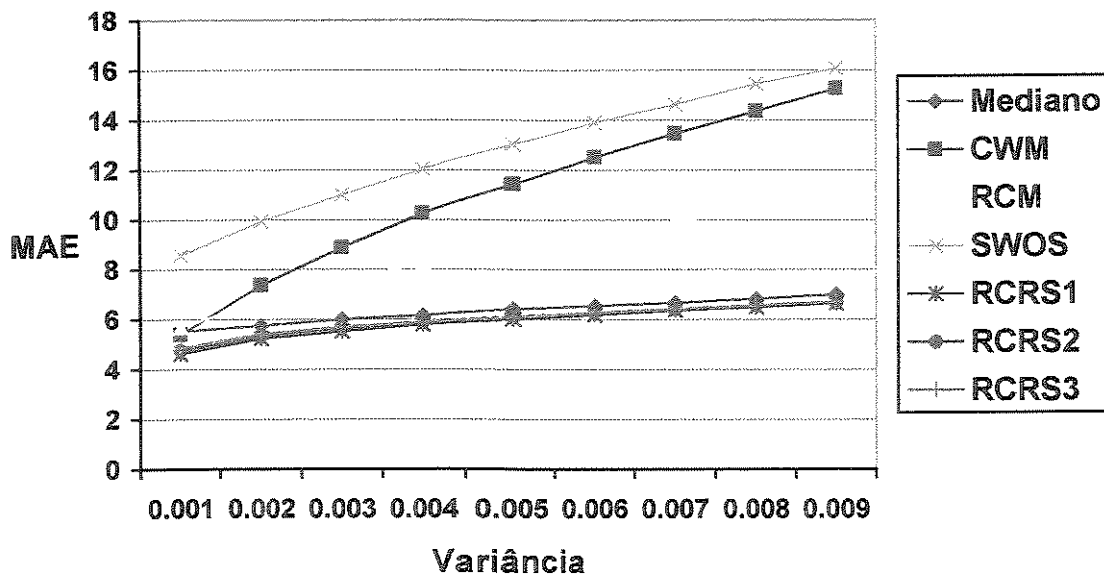
Figura 4.3. MAE dos filtros *RCRS* de primeira (*RCRS1*) e segunda ordem (*RCRS2*), filtrando as imagens “*Elaine*” e “*Boat*” corrompidas com ruído impulsivo.



Ao analisar a **Figura 4.3** pode-se observar que os filtros, que foram treinados com a imagem “*Boat*”, e que filtraram a própria imagem corrompida com ruído apresentaram um resultado um pouco melhor (menor *MAE*) do que o resultado quando eles filtraram a imagem da “*Elaine*”. Este resultado era esperado, já que quando são utilizados dados que têm estatísticas muito similares com os dados que serão filtrados, geralmente são obtidos melhores resultados.

Também pode-se verificar na **Figura 4.3** que os filtros de segunda ordem apresentam os melhores resultados. Em todos os casos foi utilizada uma janela de tamanho  $7 \times 7$ .

A **Figura 4.4** mostra os erros *MAE* quando as imagens a serem filtradas foram corrompidas com ruído Gaussiano branco, aditivo (*AWGN*) com média zero e variância  $V$ . O tamanho da janela de observação utilizada nos testes foi  $7 \times 7$ .



**Figura 4.4.** *MAE* dos filtros *RCRS*, *mediano*, *CWM*, *SWOS*, *RCM* ao filtrar a imagem “*Elaine*” corrompida com ruído Gaussiano branco, aditivo, utilizando a janela de observação com janela  $7 \times 7$ .

Ao analisar a **Figura 4.4** pode-se observar que os filtros *RCRS* apresentam novamente melhores resultados que os filtros *mediano*, *CWM*, *SWOS*, e *RCM*.

## 4.2. Análise subjetiva

Para realizar uma avaliação mais completa do desempenho dos **filtros condicionais seletores de posição (*RCRS*)** no processamento de imagens, além da análise quantitativa realizada na seção anterior, são mostradas algumas imagens nesta seção com o objetivo de comparar de maneira subjetiva o desempenho dos filtros *RCRS*, *mediano*, *CWM*, *SWOS*, *RCM* e *Wiener*.

A **Figura 4.5** mostra a imagem original de “Elaine”.



**Figura 4.5.** Imagem original de “Elaine”.

A **Figura 4.6** mostra a imagem de “Elaine” corrompida por ruído impulsivo com densidade  $D = 0,20$ .



**Figura 4.6.** Imagem de “Elaine” corrompida por ruído impulsivo do tipo “sal e pimenta” com densidade de ruído  $D = 0,20$ .

A **Figura 4.7** mostra a imagem de “Elaine” filtrada utilizando o filtro *Wiener* (ver função *Wiener2* no Matlab versão 5.3).



**Figura 4.7.** Imagem de “Elaine” restaurada pelo filtro *Wiener* com janela de tamanho  $5 \times 5$ ,  $IRSN = 8,4 \text{ dB}$ .

Na **Figura 4.7** também pode-se observar o pobre desempenho do filtro de **Wiener** com uma relação sinal/ruído muito baixa  $IRS_N = 8.4$ . (ver Anexo 2)

A **Figura 4.8** mostra a imagem de “Elaine” restaurada pelo filtro **CWM** com uma janela de tamanho  $5 \times 5$ , com parâmetro  $k=6$  (ver equação (3.4-2)).



**Figura 4.8.** Imagem de “Elaine” restaurada pelo filtro **CWM** com janela de tamanho  $5 \times 5$ ,  $k=6$ ,  $IRS_N = 13.7dB$ .

Ao analisar a **Figura 4.8** pode-se verificar que a imagem restaurada apresenta muitas alterações (impulsos), e a relação sinal/ruído  $IRS_N$  ainda é muito baixa, isto é,  $IRS_N = 13.7 dB$ .

A **Figura 4.9** mostra a imagem de “Elaine” restaurada pelo filtro **SWOS** utilizando uma janela de tamanho  $5 \times 5$ , com parâmetro  $k = 6$ .



**Figura 4.9.** Imagem de “Elaine” restaurada pelo filtro *SWOS* utilizando uma janela de tamanho  $5 \times 5$  e com  $k = 6$ ,  $IRSN = 11.6 \text{ dB}$ .

Nesta figura a relação sinal/ruído *IRNS* também é muito baixa, isto é,  $IRSN = 11.6 \text{ dB}$ . O filtro *SWOS* também deixa passar um número grande de impulsos.

A **Figura 4.10** apresenta a imagem de “Elaine” restaurada pelo filtro **mediano**  $5 \times 5$ . Aqui foi utilizada a função MEDFILT2 (ver Anexo 3 ).



**Figura 4.10.** Imagem de “Elaine” restaurada pelo filtro *mediano* utilizando uma janela de tamanho  $5 \times 5$ ,  $IRSN = 16.4 \text{ dB}$ .

O filtro **mediano** suprime muito bem os impulsos mas causa uma significativa distorção (borra) nas bordas da imagem  $IRSN = 16.4 \text{ dB}$ .

A **Figura 4.11** mostra a imagem de “Elaine” restaurada pelo filtro **RCRS** de primeira ordem (**RCRS1**).



**Figura 4.11.** Imagem de “Elaine” restaurada pelo filtro **RCRS** de primeira ordem utilizando uma janela de tamanho  $5 \times 5$ ,  $IRSN = 19.5 \text{ dB}$ .

Ao analisar a **Figura 4.11** pode-se observar que a imagem restaurada melhorou bastante em comparação a mesma imagem das figuras 4.7 a 4.10.

Embora que o filtro **RCRS1** faz um bom trabalho ( $IRSN = 19.5 \text{ dB}$ ), mesmo assim a imagem ainda possui algumas alterações.

A **Figura 4.12** apresenta a imagem restaurada usando um filtro **RCRS** de segunda ordem (**RCRS2**).



**Figura 4.12.** Imagem “Elaine” restaurada com o filtro **RCRS2** de segunda ordem, utilizando uma janela de tamanho  $5 \times 5$ ,  $IRS\text{N} = 19.9 \text{ dB}$ .

Ao analisar a **Figura 4.12** pode-se verificar que a imagem restaurada melhorou em relação a mesma imagem da figura 4.11 (imagem filtrada pelo filtro **RCRS1**), mostrando que o filtro **RCRS** de segunda ordem (**RCRS2**) possui um ótimo desempenho para restaurar imagens contaminadas com ruído. Ao calcular a relação sinal/ruído do filtro **RCRS2** foi comprovado que efetivamente é a maior de todas, isto é  $IRS\text{N} = 19.9 \text{ dB}$ .

### 4.3. Considerações finais do capítulo

Neste capítulo apresentou na primeira seção uma comparação quantitativa do desempenho de diferentes técnicas de filtragem (filtros **RCRS**, *mediano*, *CWM*, *SWOS*, *RCM* e *Wiener*). Na segunda seção foi realizada uma comparação subjetiva destas mesmas técnicas. Em ambos tipos de análises, ao comparar os resultados das simulações realizadas, pode-se verificar que o desempenho dos filtros **RCRS** é superior em relação com as outras técnicas de filtragem, isto é, devido à capacidade que possuem os filtros **RCRS** de minimizar o ruído sem alterar a estrutura da imagem.



## CAPÍTULO 5. CONCLUSÕES E TRABALHOS FUTUROS

---

Neste trabalho foi estudada e analisada a classe de filtros não – lineares **condicionais seletores de posição (RCRS)**. Estes filtros constituem uma subclasse dentro de uma classe mais geral dos filtros Seletores de Posição (**RS**). Os filtros **RS** são todos os filtros que colocam na sua saída uma amostra de um conjunto ordenado de amostras selecionadas de entrada. Muitos filtros baseados na ordem de posição, propostos anteriormente, como por exemplo, **o filtro mediano, o filtro mediano ponderado central “CWM”**(*center weighted median filter*), **o filtro de ordem estatística ponderada “WOS”** (*weighted order statistic filter*), podem ser formulados através dos filtros **RS**. Cada um desses filtros, porém, usam informações distintas para determinar qual ordem estatística se deve colocar na saída.

Os filtros **RCRS** utilizam a posição das amostras selecionadas de entrada para escolher a posição da amostra selecionada como saída. O processo de otimização determinístico utilizado neste trabalho garante o filtro ótimo para um conjunto de dados de treinamento baseado no critério do erro **LNE**. Nesta dissertação são mostrados os resultados comparativos de diferentes filtros, utilizando uma análise quantitativa através do critério do **MAE**, e é realizada uma avaliação subjetiva que permite observar o desempenho dos filtros **RCRS** em relação aos filtros **mediano, CWM, SWOS, RCM e Wiener** no processo de restauração de imagens. Os programas de treinamento e filtragem realizados neste trabalho foram feitos utilizando o aplicativo Matlab versão 5.3.

Ao analisar o comportamento dos filtros **mediano, CWM, SWOS, RCM, Wiener e RCRS** na restauração de imagens corrompidas por ruído, pode-se verificar que os filtros **RCRS** minimiza de maneira considerável o ruído sem alterar a estrutura da imagem. Outro fato importante é que ao comparar o filtros **RCRS** de primeira ordem (**RCRS1**) com o filtro **RCRS** de segunda ordem (**RCRS2**) pode-se observar que ao filtrar a imagem com o filtro **RCRS2** aparecem menos alterações na imagem que quando se realiza a filtragem com o filtro **RCRS1**.



Os trabalhos futuros que poderão ser desenvolvidos utilizando a classe de filtros não – lineares **condicionais seletores de posição (RCRS)** são :

- Implementação de programas em linguagens de alto nível (*Visual C++*, *Delphi*, *Java*, etc) [Adams, 1994] que permitam aplicar este tipo de filtragem não – linear em restauração de imagens médicas provenientes da tomografia axial computadorizada ou da Ressonância Magnética Nuclear (*RMN*), a Biologia, etc., [Belohlavek, 1997].
- Aplicar os filtros **RCRS** em codificação e compressão de imagens verificando a utilidade destes filtros no processamento de imagens digitais.

## REFERÊNCIAS BIBLIOGRÁFICAS

---

- [Adams, 1994] Adams, George B. III; Coyle, Edward J.; Lin, Liangchien; Lucke, Lori E.; Parhi, Keshab K, “ *Input compression and efficient VLSI architectures for rank order and stack filters*”, Signal Processing, vol. 38, Issue 3, August 1994, 441– 453.
- [Arce, 1994] Arce, Gonzalo R., Barner Kenneth E., “*Permutation Filters: A class of nonlinear Filters based on set Permutations*”, IEEE Transactions on Signal Processing vol. 42 no. 4, April 1994, 782 – 797.
- [Arce, 1996] Arce, Gonzalo R.; Tian Mu, “*Order statistic filter banks,*” IEEE Transactions on Image Processing, vol. 5, Issue 6, June 1996, 827 – 837.
- [Barner, 1997] Barner, Kenneth E.; Arce, Gonzalo R.,” *Design of permutation order statistic filters through group colorings*”, Circuits and Systems II: Analog and Digital Signal Processing, vol. 44, Issue 7, July 1997, 531 – 547.
- [Barner, 1998] Barner Kenneth E, Flaig, Alexander; Arce, Gonzalo R.; “*Affine order-statistic filters: medianization of linear FIR filters*”, IEEE Transactions on Signal Processing, vol. 46, Issue 8, August 1998, 2101 – 2112.
- [Belohlavek, 1997] Belohlavek M.; Greenleaf J.F, “*Detection of cardiac boundaries in echocardiographic images using a customized order statistics filter*”, Ultrasonic Imaging, vol. 19, Issue 2, 1997, 127 – 137.
- [Gonzalez, 2000] Gonzalez Rafael C; Richard Woods, “*Processamento de imagens Digitais*”, 2000.

- [Hakami, 1994] Hakami, Reza; Warter, Peter J.; Boncelet, Charles G., “*New VLSI architecture suitable for multidimensional order statistic filtering*”, IEEE Transactions on Signal Processing, vol. 42, Issue 4, April 1994, 991 – 993.
- [Hardie, 1994] Hardie, Russell C., Barner, Kenneth E.; “*Rank Conditioned Rank Selection Filters for Signal Restoration*”, IEEE Transactions on Image Processing, vol. 3, no. 2, March 1994. 192 – 206.
- [Hardie, 1996] Hardie, Russell C.; Barner Kenneth E., “*Extended Permutation Filters and their Application to Edge Enhancement*”, IEEE Transactions on Signal Processing vol. 5, no.6, June 1996, 855 – 867.
- [Himayat, 1994] Himayat, Nageen; Kassam, Saleem “*A Structure for adaptive order statistics filtering*”, IEEE Transactions of Image Processing, vol. 3, Issue 3, May 1994, 265 – 280.
- [Kataggelos, 1997] Kataggelos Aggelos K.; Banham Mark R., “*Digital Image Restoration*”, IEEE Signal Processing Magazine march 1997, 24 – 41.
- [Lim, 1990] Lim J. “*Two –Dimensional Signal Processing and Image Processing*, Prentice Hall, International Editions, New Jersey, USA, 1990.
- [Lin, 1996] Lin, Ching C.; Kuo, Chung J. “*Note on the VLSI implementation of real-time order statistic filters*”, IEEE Transactions on Signal Processing, vol. 44, Issue 5, May 1996, 1314 – 1315.
- [Michael, 1996] Ropert Michaël, Moreau de Saint-Martin François and Pelé Danielle, “*A new representation of weighted order statistic filters*”, Signal Processing, vol. 54, Issue 2, October 1996, 201 – 206.

- [Neuvo, 1994] Yrjo Neuvo, Lin Yin, “*Fast adaptation and performance characteristics of FIR – WOS Hybrid Filters*”, IEEE Transactions on signal processing vol. 42 no. 7, April 1994, 1600 – 1627.
- [Oppenheim, 1975] Oppenheim A.; R. Schaffer, “*Digital Signal Processing*”, Prentice Hall, International Editions, New Jersey, USA 1975.
- [Paez, 1994] Paez-Borralló, Jose M.; Zazo-Bello, Santiago, “*On the joint statistical characterization of the input and output of an order statistic filter*”, IEEE Transactions on Signal Processing, vol. 42, Issue 2, February 1994, 456 – 459.
- [Pratt, 1978] Pratt W. K., “*Digital Image Processing*,” John Wiley & Sons, USA 1978.
- [Yin, 1997] Yin, Lin; Astola, Jaakko; Neuvo, Yrjo, “*Adaptive multistage weighted order statistic filters based on the back-propagation algorithm*”, IEEE Transactions on Signal Processing, vol. 42, Issue 2, February 1994, 419 – 422.
- [Yu, 1994] Yu, Pao-Ta; Liao, Wei-Hsiang, “*Weighted order statistics filters - Their classification, some properties, and conversion algorithm*”, IEEE Transactions on Signal Processing, vol. 42, Issue 10, October 1994, 2678 – 2691.

## ANEXO 1 – TAMANHO DA JANELA E RELAÇÃO SINAL/RUÍDO

---

Considerando que não existe ruído Gaussiano branco aditivo (*AWGN*) e que o filtro de suavização  $F_2$  possui como amostra de saída, em um instante qualquer, a média simples de  $N$  pixels. Em regiões planas ou sem muitas variações, isto resulta na variância residual com uma média de:

$$\sigma_{residual}^2 = \frac{\sigma_n^2}{N} \quad (A1-1)$$

Depois da suavização, a relação sinal - ruído para regiões planas (em dB) é encontrada da seguinte maneira:

$$\begin{aligned} SNR_{suavização} &= 10 \log_{10} \left( \frac{P_s}{\sigma_n^2} N \right) \\ SNR_{suavização} &= 10 \log_{10} \left( \frac{P_s}{\sigma_n^2} \right) + 10 \log_{10} N \\ SNR_{suavização} &= SNR_{ruído} + 10 \log_{10} N \end{aligned} \quad (A1-2)$$

onde  $P_s$  é a potência do sinal,  $SNR_{ruído}$  é a relação sinal/ruído para regiões planas na imagem corrompida com ruído.

A equação (A1-2) indica um aumento na  $SNR_{suavização}$  independente da quantia de ruído na imagem.

A **tabela A1.1** mostra o aumento de  $SNR$  utilizando-se janelas de vários tamanhos.

**Tabela A1.1.** *SNR* para janelas de vários tamanhos.

Vizinhança	N	Incremento na SNR para regiões planas
3x3	9	+9.5 dB
5x5	25	+14.0 dB
7x7	49	+16.9 dB
9x9	81	+19.1 dB
11x11	121	+20.8 dB

Embora janelas de tamanhos maiores proporcionam redução de ruído em regiões planas, quando são utilizadas janelas de tamanhos muito grandes, o filtro suaviza mais a imagem. Devido a isso, podem ser alterados detalhes importantes na imagem. O melhor é utilizar uma janela suficientemente grande para suavizar adequadamente as regiões planas da imagem. Resolvendo a equação (A1-2) para  $N$ , e assumindo que deseja-se uma **SNR = 30 dB** nas regiões planas, o tamanho adequado da janela é determinado por:

$$N \approx 1000 \frac{\sigma_n^2}{P_s} \quad (\text{A1-3})$$

A **tabela A1.2** apresenta os diferentes tamanhos de janelas, calculados a partir da equação (A1-3). Assumindo  $P_s = 128^2$ .

**Tabela A1.2.** Tamanhos de janelas recomendáveis para + 30 dB de ganho em regiões planas, assumindo  $P_s = 128^2$ .

$\sigma_n^2$	N para SNR = +30 dB Em regiões planas	Tamanhos de janelas recomendáveis
100	6	3x3
200	12	5x5
400	24	7x7
800	48	9x9
1600	98	11x11

## ANEXO 2 – FONTE DE DEGRADAÇÃO DE IMAGENS

---

Em processamento de imagens digitais existe um modelo geral para a degradação linear causada pelo borramento e o ruído aditivo, que é descrito pela superposição de somatórios na seguinte equação:

$$y(i, j) = \sum_{k=1}^M \sum_{l=1}^N h(i, j; k, l) f(k, l) + n(i, j) \quad (\text{A2-1})$$

onde:

$f(i, j)$  representa uma imagem original  $M \times N$ .

$y(i, j)$  é a imagem degradada adquirida pelo sistema de aquisição.

$h(i, j; m, n)$  é a *function point spread* (PSF) bidimensional do sistema de aquisição de imagens que em geral pode ser espacialmente variável nesta equação.

$n(i, j)$  é o ruído aditivo introduzido pelo sistema.

Usualmente, o ruído considerado é o ruído branco com distribuição Gaussiana com média zero, pois este tipo de ruído possui uma boa modelagem em imagens. A dificuldade na solução do problema de restauração com borrão espacialmente variável motivou a utilização de modelos estacionários neste tipo de problema, [Katagelos, 1997]. Nesta situação, a equação utilizada para um sistema de degradação é:

$$y(i, j) = \sum_{k=1}^M \sum_{l=1}^N h(i-k, j-l) f(k, l) + n(i, j) = h(i, j) ** f(i, j) + n(i, j) \quad (\text{A2-2})$$

onde:

**\*\*** é a convolução bidimensional

Um aspecto a ser relacionado no processamento de imagens é o tratamento de bordas. O processo de borramento descrito na equação (A2-2) é linear. Porém, usualmente esta convolução é aproximada pela convolução circular. Isto envolve o tratamento da imagem como um sinal periódico bidimensional. As bordas de uma imagem são também tratadas como extensões simétricas da imagem, ou como valores de níveis de cinza repetidos nos

pixels da borda. Estes métodos procuram minimizar a distorção nas bordas causadas pelo algoritmo de filtragem de uma imagem. Ao implementar o algoritmo de restauração é muito importante considerar como são tratadas as bordas de uma imagem.

### Blurred Signal-to-Noise Ratio (BSNR)

A maioria dos estudos de restauração de imagens, refere-se à degradação modelada por borramento e ruído aditivo em termos de uma medida chamada *de Blurred Signal-to-Noise Ratio (BSNR)* [Kataggelos, 1997]. Esta grandeza é definida em termos da variância do ruído aditivo  $\sigma_n^2$  da seguinte maneira:

$$BSNR = 10 \log_{10} \left\{ \frac{\frac{1}{MN} \sum_{i,j}^{M,N} [g(i,j) - \bar{g}(i,j)]^2}{\sigma_n^2} \right\} \quad (A2-3)$$

Para uma imagem de dimensão  $M \times N$ , onde:

$g(i,j) = y(i,j) - n(i,j)$  da equação (A2-1)

$\bar{g}(i,j) = E\{g\}$ , representa o valor esperado, ou a média de  $g$ .

### O melhoramento da relação Sinal/Ruído (ISNR)

O melhoramento da relação Sinal/Ruído (ISNR) habitualmente é feito para verificar o desempenho do algoritmo de restauração. Esta medida é calculada pela seguinte equação:

$$ISNR = 10 \log_{10} \left\{ \frac{\sum_{i,j}^{M,N} [f(i,j) - y(i,j)]^2}{\sum_{i,j}^{M,N} [f(i,j) - \hat{f}(i,j)]^2} \right\} \quad (A2-4)$$



onde:

- $f(i,j)$  e  $y(i,j)$  são as intensidades de níveis de cinza das imagens original e degradada, respectivamente.
- $\hat{f}(i,j)$  é a intensidade de nível de cinza da imagem restaurada.

Obviamente, esta medida somente pode ser calculada em simulações onde está disponível a imagem original. As medidas dos erros quadráticos médios como **ISNR** nem sempre refletem as propriedades perceptuais do sistema visual humano, porém, ela fornece uma medida objetiva padrão que possibilita comparar as diferentes técnicas de restauração, [Kataggelos, 1997].

### ANEXO 3 – FUNÇÃO “IMNOISE” DO MATLAB UTILIZADA PARA CONTAMINAR A IMAGEM COM VÁRIOS TIPOS DE RUÍDOS

---

*IMNOISE* soma ruído à imagem.

$J = \text{IMNOISE}(I, \text{TYPE}, \dots)$  soma ruído à imagem  $I$  (intensidade) do tipo  $\text{TYPE}$ .

$\text{TYPE}$  é uma das seguintes palavras:

'gaussian' ruído Gaussiano branco

'salt & pepper' ruído tipo “sal e pimenta” (“on and off” pixels)

'speckle' ruído multiplicativo

Dependendo de  $\text{TYPE}$ , podem ser considerados outros parâmetros em *IMNOISE*, como por exemplo:

- $J = \text{IMNOISE}(I, \text{'gaussian'}, M, V)$  soma à imagem  $I$  ruído Gaussiano branco com média  $M$  e variância  $V$ .
- $J = \text{IMNOISE}(I, \text{'salt \& pepper'}, D)$  soma à imagem  $I$  ruído do tipo “sal e pimenta”, onde  $D$  é a densidade do ruído.
- $J = \text{IMNOISE}(I, \text{'speckle'}, V)$  soma à imagem  $I$  ruído multiplicativo, utilizando a equação  $J = I + n * I$ , onde  $n$  é ruído aleatório uniformemente distribuído com média 0 e variância  $V$ .

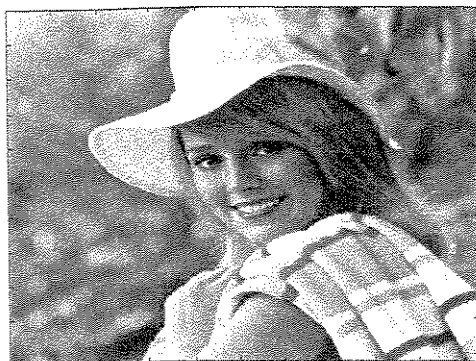
Exemplo:

-----

```
I = imread('eight.tif');  
J = imnoise(I, 'salt & pepper', 0.02);  
imshow(I), figure, imshow(J)
```

## ANEXO 4 – EXEMPLO DIDÁTICO DO PROCESSO DE FILTRAGEM UTILIZANDO O FILTRO RCRS

A Figura A4.1 mostra a imagem de “Elaine” que foi utilizada neste exemplo didático para descrever o processo de filtragem através do filtro *RCRS*.



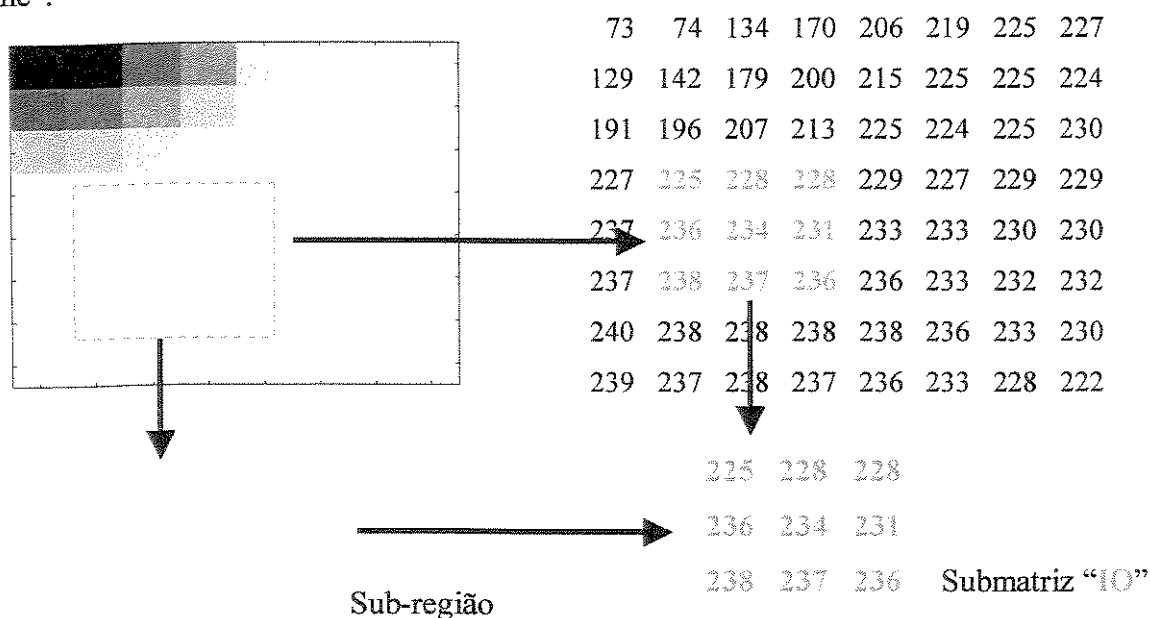
(a)



(b)

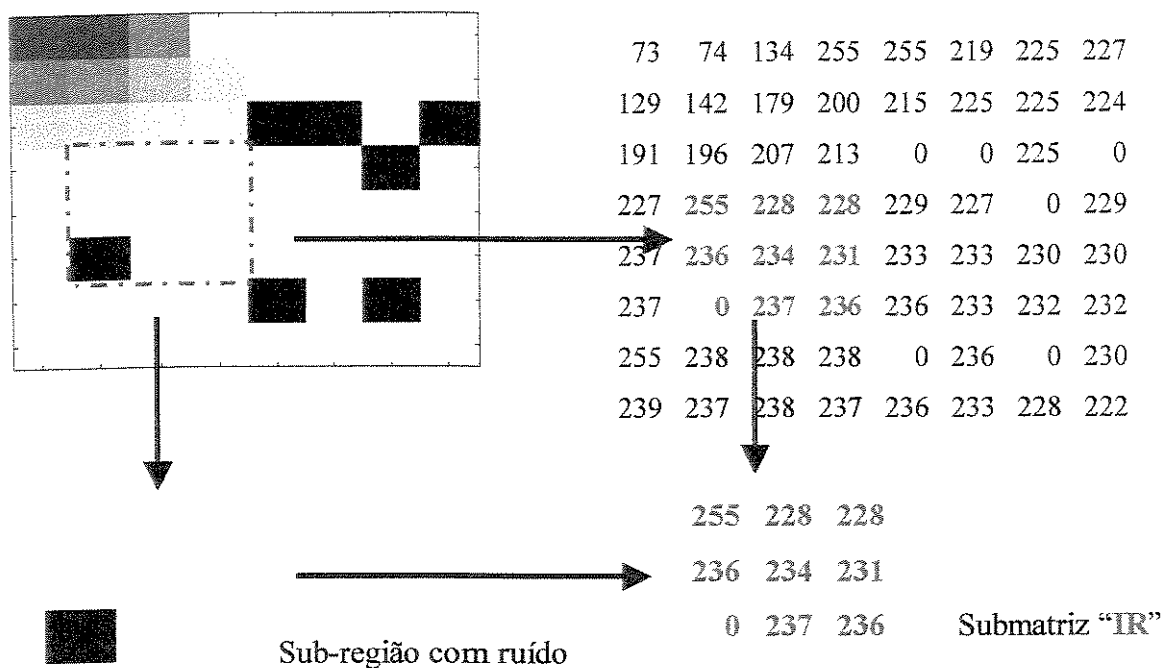
**Figura A4.1.** (a) Imagem original da “Elaine” 512x512 *pixels*, (b) Imagem da “Elaine” corrompida com ruído do tipo “sal e pimenta” com densidade  $D = 0,20$

O processo de filtragem descrito neste exemplo foi realizado em uma pequena sub-região da imagem de “Elaine”. A Figura A4.2 mostra a sub-região da imagem original “Elaine”.



**Figura A4.2.** Sub-região da imagem original “Elaine” e a submatriz “IO” com os valores de níveis de cinza dos *pixels* da sub-região analisada.

A **Figura A4.3** mostra uma sub-região da imagem “Elaine” corrompida com ruído do tipo “sal e pimenta” localizada na mesma posição que a sub-região da **Figura A4.2**.



**Figura A4.3.** Sub-região da imagem de “Elaine” contaminada com ruído e a submatriz “IR” com os valores de níveis de cinza dos *pixels* da sub-região contaminada.

A partir das sub-regiões das **Figuras A4.2** e **A4.3** foram obtidas as submatrizes imagem:

- Submatriz “IO”: com os valores de níveis de cinza dos *pixels* da sub-região da imagem original de “Elaine”.
- Submatriz “IR”: com os valores de níveis de cinza dos *pixels* da sub-região da imagem de “Elaine” contaminada com ruído tipo “sal e pimenta”.

## Processo de Treinamento

Este processo de treinamento segue procedimento de otimização sobre a classe de filtros *RCRS* descrito no capítulo 3 deste trabalho.

### Vetores observação. Vetores ordenados.

Os vetores observação “ $x$ ” foram obtidos a partir da submatriz “ $IR$ ” usando uma janela de tamanho  $3 \times 1$ . Isto é, o centro da janela é posicionado sobre a primeira amostra da submatriz “ $IR$ ” ficando dentro da janela 3 amostras (a amostra do centro da janela, e as vizinhas da esquerda e da direita da amostra do centro da janela). Cada vetor observação é constituído por três amostras. Como a submatriz “ $IR$ ” possui nove valores de níveis de cinza, colocando o centro da janela sobre cada um das amostras da submatriz “ $IR$ ”, foram obtidos nove vetores observação “ $x$ ”. Ordenando estes vetores de observação foram obtidos os vetores ordenados “ $x^r$ ”.

A Tabela A4.1 mostra esses dois conjuntos de vetores

**Tabela A4.1.** Vetores observação “ $x$ ” e vetores ordenados “ $x^r$ ”.

$IR$	Vetores observação “ $x$ ”	Vetores ordenados “ $x^r$ ”
	227 255 228	227 228 255
	255 228 228	228 228 255
237 236 234 231 233	228 228 229	228 228 229
227 255 228 228 229	237 236 234	234 236 237
237 0 237 236 236	236 234 231	231 234 236
	234 231 233	231 233 234
	237 0 237	0 237 237
	0 237 236	0 236 237
	237 236 236	236 236 237

### Vetor das amostras selecionadas $x^*$ . Vetor característico $r^*$

O vetor das amostras selecionadas “ $x^*$ ” está constituído pelas amostras escolhidas dentro do vetor de observação “ $x$ ”. O vetor característico “ $r^*$ ” relaciona a posição da amostra selecionada com a localização dentro da janela. Neste exemplo selecionou-se a amostra da posição central da janela de observação. Fazendo uma varredura por todas as amostras da submatriz  $IR$ , os vetores característicos “ $r^*$ ” são :

$$\begin{array}{lll} r_1 = 3 & r_4 = 2 & r_7 = 1 \\ r_2 = 1 & r_5 = 2 & r_8 = 3 \\ r_3 = 2 & r_6 = 1 & r_9 = 1 \end{array}$$

### Vetor Diferença “ $P(m)$ ”

$P(m)$  é um vetor que contém as  $L_\eta$  diferenças normalizadas:  $P(m) = |d(n_m) - x^r(n_m)|^\eta$ .

A Tabela A4.2 mostra os valores do vetor “ $P(m)$ ” obtidos para cada um dos vetores ordenados.

**Tabela A4.2.** Vetores Diferença  $P(m)$

$d(n)$	$x^r$	$P(m)$	$r$
225	227 228 255	$\Rightarrow$ 2 3 30	3
228	228 228 255	$\Rightarrow$ 0 0 27	1
228	228 228 229	$\Rightarrow$ 0 0 1	2
236	234 236 237	$\Rightarrow$ 2 0 1	2
234	231 234 236	$\Rightarrow$ 3 0 2	2
231	231 233 234	$\Rightarrow$ 0 2 3	1
238	0 237 237	$\Rightarrow$ 238 1 1	1
237	0 236 237	$\Rightarrow$ 237 1 0	3
236	236 236 237	$\Rightarrow$ 0 0 1	1

### Vetor Erro Cumulativo “ $R$ ”

O vetor Erro cumulativo  $R$  é dado pela seguinte expressão:

$$R_r(m) = R_r(m-1) + P(m)$$

A Tabela A4.3 mostra os valores obtidos dos vetores Erro cumulativo  $R$  para cada um dos vetores característicos.

**Tabela A4.3.** Vetores Erro Cumulativo “ $R$ ”

Vetor	r = 1	r = 2	r = 3
	0 0 27	0 0 1	2 3 30
$P(m)$	0 2 3	2 0 1	237 1 0
	238 1 1	3 0 2	
	0 0 1		
$R(m)$	238 3 32	5 0 4	239 4 30

### Função de classificação ótima ou Seletor ótimo de saída “ $S_{opt}(r)$ ”

Na sequência de treinamento a função de classificação ótima é determinada pelo elemento mínimo em  $R_j(m)$  e está dado por:

$$S_{opt}^m = k : R_{r,k}(m) = \min(R_{r,1}(m), R_{r,2}(m), \dots, R_{r,N}(m))$$

Ou seja, para uma janela de tamanho  $3 \times I$ ,  $N = 3$ , o parâmetro “ $k$ ” pode tomar os valores 1, 2 ou 3 (onde “ $k$ ” é a posição ou classificação do elemento mínimo de  $R$ ).

A **Tabela A4.4** mostra os valores de “ $k$ ” que formam a função de classificação ótima “ $Sopt(r)$ ”

**Tabela A4.4.** Valores do parâmetro  $k$

	$r = 1$			$r = 2$			$r = 3$		
$R$	238	3	32	5	0	4	239	4	30
$k$	<b>2</b>			<b>2</b>			<b>2</b>		

Portanto, a função de classificação ótima é:

$$Sopt(r) = [2 \ 2 \ 2]$$

### Processo de Filtragem

Este processo de filtragem é aplicado à mesma submatriz “**IR**” da imagem corrompida com ruído do tipo “sal e pimenta” mostrada na Figura A4.1(b).

$$IR = \begin{bmatrix} 255 & 228 & 228 \\ 236 & 234 & 231 \\ 0 & 237 & 236 \end{bmatrix}$$

No caso da primeira amostra (valor de nível de cinza “255”) da submatriz **IR**, e aplicando uma janela de tamanho  $3 \times 1$  do mesmo modo que na fase de treinamento, foram obtidos os seguintes vetores:

- **vetor de observação**  $x = [227 \ 255 \ 228]$
- **vetor ordenado**  $x^r = [227 \ 228 \ 255]$
- **vetor característico**  $r = [3]$



A saída do filtro *RCRS* é caracterizada pela seguinte equação:

$$F_{RCRS}(\mathbf{x}) = x_{(Sopt(r^*))}$$

Logo, a partir da função de classificação ótima obtida na fase de treinamento:

$$Sopt(r) = [2 \ 2 \ 2]/(3) = 2$$

A saída do filtro *RCRS* no caso da primeira amostra da submatriz *IR* é:

$$F_{RCRS} = [227 \ 228 \ 255]/(2) = 228$$

Aplicado este procedimento para todas as amostras da submatriz *IR*, foi obtida a submatriz *IM* da imagem filtrada.

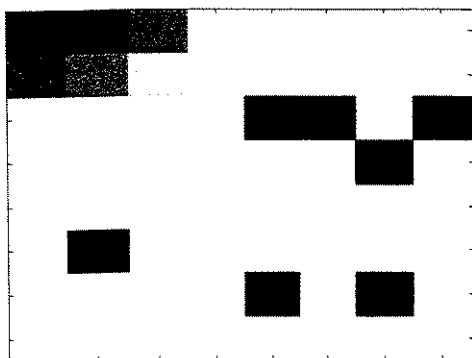
A Tabela A4.5 mostra a sub-matriz *IM* da imagem filtrada.

**Tabela A4.5.** Submatrizes das sub-regiões da imagem original “IO”, imagem corrompida com ruído “IR” e da imagem restaurada “IM” respectivamente.

IO			IR				IM		
225	228	228	255	228	228		228	228	228
236	234	231	236	234	231	⇒	236	234	233
238	237	236	0	237	236		237	236	236

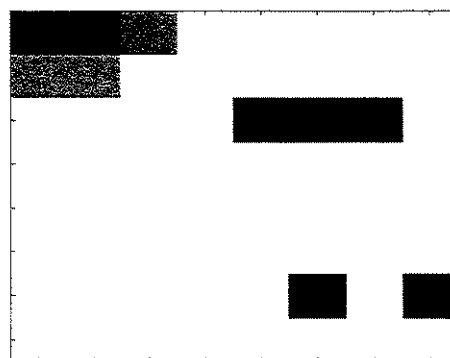
O resultado final da filtragem nesta região da imagem da “Elaine” corrompida com ruído do tipo “sal e pimenta”, onde se utiliza um filtro *RCRS* de primeira ordem com uma janela de tamanho  $3 \times 1$  é mostrado na Figura A4.4.

Região com ruído



(a)

Região filtrada



(b)

**Figura A4.4.** Região da imagem de “Elaine” (a) Região corrompida com ruído, (b) Região filtrada com um filtro *RCRS* de primeira ordem com janela de tamanho  $3 \times 1$ .