

**UNIVERSIDADE ESTADUAL PAULISTA**

Faculdade de Ciências e Tecnologia de Presidente Prudente

Programa de Pós-Graduação em Matemática Aplicada e Computacional

**AVALIAÇÃO DE TÉCNICAS DE  
INTERPOLAÇÃO DE IMAGENS DIGITAIS**

Wesley Barbosa Dourado

Orientador: Prof. Dr. Aylton Pagamisse

Presidente Prudente, Agosto de 2014

**UNIVERSIDADE ESTADUAL PAULISTA**

Faculdade de Ciências e Tecnologia de Presidente Prudente

Programa de Pós-Graduação em Matemática Aplicada e Computacional

# **AVALIAÇÃO DE TÉCNICAS DE INTERPOLAÇÃO DE IMAGENS DIGITAIS**

Wesley Barbosa Dourado

Orientador: Prof. Dr. Aylton Pagamisse

Dissertação apresentada ao Programa de Pós-Graduação em Matemática Aplicada e Computacional da Faculdade de Ciências e Tecnologia da UNESP como parte dos requisitos para obtenção do título de Mestre em Matemática Aplicada e Computacional.

Presidente Prudente, Agosto de 2014

## FICHA CATALOGRÁFICA

D771a Dourado, Wesley Barbosa.  
Avaliação de Técnicas de Interpolação de Imagens Digitais / Wesley  
Barbosa Dourado. - Presidente Prudente : [s.n], 2014  
139 f. : il.

Orientador: Aylton Pagamisse  
Dissertação (mestrado) - Universidade Estadual Paulista, Faculdade de  
Ciências e Tecnologia  
Inclui bibliografia

1. Interpolação. 2. Zoom. 3. Imagem digital. I. Pagamisse, Aylton. II.  
Universidade Estadual Paulista. Faculdade de Ciências e Tecnologia. III.  
Título.

**BANCA EXAMINADORA**



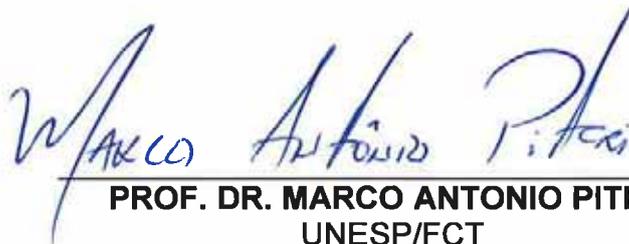
---

**PROF. DR. AYLTON PAGAMISSE**  
ORIENTADOR



---

**PROF. DR. APARECIDO NILCEU MARANA**  
UNESP/FC



---

**PROF. DR. MARCO ANTONIO PITERI**  
UNESP/FCT



---

**WESLEY BARBOSA DOURADO**

Presidente Prudente (SP), 08 de agosto de 2014.

Resultado: Aprovado

*A minha mãe, Roseneide  
e a todos que me ajudaram  
até aqui.*

---

# Agradecimentos

Em primeiro lugar agradeço a Deus pois sem Ele nada poderia fazer. Agradeço a Ele pelo entendimento e inteligência que me concede durante a caminhada. Sou grato a Ele por todos os percursos de minha vida.

Agradeço, a minha mãe Roseneide por toda paciência, incentivo e preocupação também.

Agradeço ao prof. Dr. Aylton Pagamisse pela orientação, contribuição, conselhos e paciência que sempre teve comigo. Também sou grato a todos os professores do PosMAC, em especial: Vanessa Botta, Messias Meneguette, Eniuce Menezes e Ronan Antônio, que lecionaram as disciplinas que cursei no mestrado e contribuíram muito para o meu desenvolvimento acadêmico. Também aos professores doutores Marco A. Piteri, Almir O. Artero e Aparecido Nilceu Marana pelas correções e contribuições para com este trabalho.

Agradeço aos meus amigos do PosMAC. Todos que contribuíram de maneira geral para o meu crescimento não somente acadêmico mas como cidadão. Agradecimentos especiais a Alisson Reinol, Patrícia Demetria, Irineu Palhares, Rafael Paulino e a Renata N. Imada, pelas críticas construtivas e apoio, seja através de conselhos, seja através do ato de compartilhar o conhecimento científico.

Agradeço aos alunos do 2º ano da graduação de Licenciatura em Matemática a qual realizei meu estágio de docência. A participação deles em meu crescimento como professor foi indispensável.

Agradeço aos funcionários da seção de pós-graduação pelo bom auxílio prestado no decorrer do mestrado e pela disposição de fazê-lo sempre que preciso.

À CAPES pelo apoio financeiro.

Finalmente, agradeço a todos que, diretamente ou indiretamente, contribuíram para a elaboração deste trabalho.

*“ Até aqui nos ajudou o Senhor ”*

*1 Samuel 7:12b*

---

# Resumo

Nesta dissertação é realizado um estudo comparativo sobre alguns tipos de algoritmos aplicados a imagens digitais voltado para interpolação. Este trabalho inclui os métodos clássicos, que são: replicação, bilinear, bicúbica, Lagrange e interpolação pela função *sinc*; e alguns recentes: algoritmo-localmente adaptativo, método *New Edge-Direction Interpolation* (NEDI), *improved New Edge Direction Interpolation* (iNEDI), *iterative curvature-based interpolation* (ICBI), interpolação utilizando *wavelets* redundantes e utilizando filtro bilateral. Todos os novos métodos possuem melhorias em aspectos visuais e redução de ruídos nas bordas em relação aos clássicos. Os métodos avaliados são comparados visualmente e quantitativamente utilizando as métricas estatísticas: erro médio quadrático (MSE), Raíz do Erro Médio Quadrático (RMSE), Erro Médio Quadrático Normalizado (NMSE), Relação Sinal-Ruído (SNR), Coeficiente de Correlação (CC) e Índice de Qualidade Universal (IQI). Também é realizada uma discussão dos resultados obtidos, analisando as qualidades e os defeitos dos métodos estudados. Por fim, são propostas algumas ideias para trabalhos futuros, visando aprimorar as técnicas já existentes.

**Palavras-Chave:** Interpolação. *Zoom*. Imagem digital.

# Abstract

---

*In this dissertation, a comparative study of some types of interpolation algorithms for digital images is carried out. This dissertation presents the classical methods, which are: replication, bilinear, bicubic, and Lagrange interpolation by the function sinc; and some newer methods: adaptive locally algorithm adaptive, New Interpolation Edge Direction (NEDI), improved New Edge-Direction Interpolation (iNEDI), iterative curvature-based interpolation (ICBI), interpolation method that uses redundant wavelets and interpolation method using the bilateral filter. All new methods have improvements to visual aspects and reduction of noise at the edges compared to classical. All evaluated methods are compared visually and quantitatively using statistical metrics, such as: Mean Square Error (MSE), Root Mean Square Error (RMSE), Normalized Mean Square Error (NMSE), Signal Noise Ratio (SNR), Correlation Coefficient (CC) and Universal Image Quality Index (IQI). A discussion of the results is also performed, analyzing the strengths and weaknesses of the methods studied. Finally, we propose some ideas for future work, aiming to improve existing techniques.*

**Keywords:** *Interpolation. Zoom. Digital image.*

---

# Lista de Figuras

1.1	Ilustração de <i>zoom</i> em uma imagem pelo fator de dois. . . . .	17
1.2	Sinal original e sua reamostragem com o efeito <i>aliasing</i> . . . . .	18
1.3	Exemplo do efeito <i>aliasing</i> em sinal unidimensional. . . . .	18
1.4	Exemplo do efeito <i>aliasing</i> em imagens. . . . .	19
1.5	Exemplo de <i>Jaggies</i> em imagens. . . . .	19
2.1	Esquema da interpolação do vizinho mais próximo. . . . .	23
2.2	Ampliação de imagem utilizando a interpolação pelo vizinho mais próximo. . . . .	23
2.3	Exemplificação do cálculo do método bilinear. . . . .	24
2.4	Ampliação de imagem pela interpolação bilinear. . . . .	25
2.5	Esquema da interpolação bicúbica. . . . .	26
2.6	Ampliação da imagem pela interpolação bicúbica. . . . .	26
2.7	Gráfico da função $\text{sinc}(\cdot)$ normalizada. . . . .	31
2.8	Exemplo da inserção de zeros em um sinal unidimensional. . . . .	33
2.9	Ampliação de imagem utilizando a interpolação pela função $\text{sinc}(\cdot)$ . . . . .	34
3.1	Representação da interpolação no segundo estágio. . . . .	36
3.2	Representação da interpolação no terceiro estágio. . . . .	38
3.3	Ilustração dos pontos no segundo estágio. . . . .	39
3.4	Exemplificação da terceira condição. . . . .	41
3.5	Exemplificação da quarta condição. . . . .	42
3.6	Ilustração da sexta condição. . . . .	43
3.7	Ilustração da nona condição. . . . .	44
3.8	Ilustração do terceiro estágio do método. . . . .	44
3.9	Aplicação do método adaptativo na imagem Lena. . . . .	45
3.10	Representação do primeiro passo do método NEDI. . . . .	46
3.11	Representação do cálculo dos termos $R_{kl}$ , $\hat{R}_{kl}$ , $r_k$ e $\hat{r}_k$ . . . . .	46
3.12	Representação dos pixels a serem interpolados . . . . .	48
3.13	Segundo passo do algoritmo NEDI. . . . .	48
3.14	Imagem ampliada pelo método NEDI. . . . .	49
3.15	Representação da janela circular utilizada no método iNEDI. . . . .	50

---

3.16	Ilustração da transição de regiões homogêneas separadas por uma borda. . . . .	51
3.17	Aplicação do método iNEDI na imagem Lena. . . . .	52
3.18	Ilustração dos pixels utilizados para o calculo de $\tilde{I}_{11}$ e $\tilde{I}_{22}$ . . . . .	53
3.19	Segundo passo do método ICBI. . . . .	55
3.20	Aplicação do método ICBI na imagem Lena. . . . .	56
4.1	Esquema do processo de decimação em transformada <i>wavelets</i> . . . . .	59
4.2	Esquema do processo de análise da imagem. . . . .	59
4.3	Processo de síntese para a composição da imagem. . . . .	60
4.4	Esquema do processo de transformada <i>wavelet</i> diádica. . . . .	61
4.5	Representação da interpolação do ponto central utilizando os vizinhos das diagonais. . . . .	62
4.6	Esquema da imagem após a primeira etapa do algoritmo. . . . .	63
4.7	Representação da imagem rotacionada em $45^{\circ}$ . . . . .	63
4.8	Pontos interpolados na segunda etapa. . . . .	64
4.9	Aplicação do método de interpolação que utiliza <i>wavelets</i> proposto na imagem Lena. . . . .	65
4.10	Representação da camada base e a camada de detalhes. . . . .	67
4.11	Filtragem da camada de detalhes. . . . .	68
4.12	Representação das configurações do algoritmo para o cálculo das intensidades de pixels. . . . .	69
4.13	Aplicação do método proposto na imagem Lena. . . . .	70
5.1	Degradações na Imagem Lena . . . . .	76
5.2	Exemplificação da percepção visual no contraste ( <i>Koffka Ring</i> ) . . . . .	78
5.3	Imagem Lena degradada com ruído branco. . . . .	78
5.4	Representação do foco do HVS. . . . .	79
6.1	Imagens utilizadas na análise visual. . . . .	82
6.2	Imagem Lena ampliada com método replicação. . . . .	83
6.3	Imagem copo ampliada com método replicação. . . . .	84
6.4	Imagem teste ampliada com método replicação. . . . .	84
6.5	Imagem Lena ampliada com método bilinear. . . . .	85
6.6	Imagem copo ampliada com método bilinear. . . . .	85
6.7	Imagem teste ampliada com método bilinear. . . . .	86
6.8	Imagem Lena ampliada com método bicúbico. . . . .	87
6.9	Imagem copo ampliada com método bicúbico. . . . .	87
6.10	Imagem teste ampliada com método bicúbico. . . . .	88
6.11	Imagem Lena ampliada com método <i>sinc</i> . . . . .	88
6.12	Imagem Copo ampliada com método <i>sinc</i> . . . . .	89

---

6.13 Imagem teste ampliada com método <i>sinc</i> . . . . .	89
6.14 Imagem Lena ampliada com método adaptativo. . . . .	90
6.15 Imagem copo ampliada com método adaptativo. . . . .	90
6.16 Imagem teste ampliada com método adaptativo. . . . .	91
6.17 Imagem Lena ampliada com método Nedi. . . . .	92
6.18 Imagem copo ampliada com método Nedi. . . . .	92
6.19 Imagem teste ampliada com método Nedi. . . . .	93
6.20 Imagem Lena ampliada com método iNedi. . . . .	93
6.21 Imagem copo ampliada com método iNedi. . . . .	94
6.22 Imagem teste ampliada com método iNedi. . . . .	94
6.23 Imagem Lena ampliada com método icbi. . . . .	95
6.24 Imagem copo ampliada com método icbi. . . . .	95
6.25 Imagem teste ampliada com método icbi. . . . .	96
6.26 Imagem Lena ampliada com método <i>wavelets</i> redundantes. . . . .	96
6.27 Imagem copo ampliada com método <i>wavelets</i> redundantes. . . . .	97
6.28 Imagem teste ampliada com método <i>wavelets</i> redundantes. . . . .	97
6.29 Imagem Lena ampliada com método que utiliza o filtro bilateral. . . . .	98
6.30 Imagem copo ampliada com método que utiliza o filtro bilateral. . . . .	98
6.31 Imagem teste ampliada com método que utiliza o filtro bilateral. . . . .	99
6.32 Representação das cores do modelo RGB. . . . .	100
6.33 Representação tridimensional das cores do modelo RGB. . . . .	101
6.34 Representação das cores do modelo HSI. . . . .	102
6.35 Imagem colorida escolhida. . . . .	103
6.36 Imagem colorida ampliada pelos métodos estudados. . . . .	104
6.37 Imagens utilizadas para a comparação estatística. . . . .	105
6.38 Exemplo da interferência da sub-amostragem para a comparação de métodos de interpolação. . . . .	106
6.39 Imagens testes utilizadas para a comparação visual dos métodos. . . . .	111
6.40 Aplicação dos métodos Replicação ao NEDI na imagem cameraman. . . . .	112
6.41 Aplicação dos métodos iNEDI e ICBI na imagem cameraman. . . . .	113
6.42 Aplicação dos métodos <i>wavelets</i> e bilateral na imagem cameraman. . . . .	114
6.43 Aplicação dos métodos Replicação ao NEDI na imagem Lena. . . . .	115
6.44 Aplicação dos métodos iNEDI ao bilateral na imagem Lena. . . . .	116
6.45 Aplicação dos métodos Replicação ao <i>Sinc</i> na imagem da flor. . . . .	117
6.46 Aplicação dos métodos Localmente adaptativo ao ICBI na imagem da flor. . . . .	118
6.47 Aplicação dos métodos <i>wavelets</i> ao bilateral na imagem da flor. . . . .	119
6.48 Aplicação dos métodos Replicação ao NEDI na imagem do babuíno. . . . .	120
6.49 Aplicação dos métodos iNEDI ao bilateral na imagem do babuíno. . . . .	121
6.50 Aplicação dos métodos Replicação ao <i>Sinc</i> estudados na imagem do copo. . . . .	122

6.51	Aplicação dos métodos Localmente adaptativo ao ICBI estudados na imagem do copo. . . . .	123
6.52	Aplicação dos métodos <i>wavelets</i> e bilateral na imagem do copo. . . . .	124

---

# Sumário

<b>1</b>	<b>Introdução</b>	<b>14</b>
1.1	Técnicas de Interpolação . . . . .	15
1.2	O processo de <i>Zoom</i> . . . . .	16
1.2.1	Amostragem e Reamostragem . . . . .	17
1.3	Organização do Trabalho . . . . .	20
<b>2</b>	<b>Métodos Clássicos</b>	<b>22</b>
2.1	Interpolação por vizinho mais próximo . . . . .	22
2.2	Interpolação Bilinear . . . . .	24
2.3	Interpolação Bicúbica . . . . .	25
2.4	Interpolação por Polinômios de Lagrange . . . . .	27
2.5	Interpolação com base na Convolução . . . . .	28
2.5.1	Transformada de Fourier . . . . .	28
2.5.2	Interpolação por função <i>sinc</i> . . . . .	30
<b>3</b>	<b>Métodos de Interpolação Utilizando Bordas Direcionais</b>	<b>35</b>
3.1	Algoritmo Localmente-adaptativo . . . . .	36
3.2	Adaptação do Algoritmo Baseado em Estágios . . . . .	39
3.3	O método NEDI . . . . .	45
3.4	Método iNEDI . . . . .	49
3.5	Interpolação Iterativa com Base em Curvatura . . . . .	52
<b>4</b>	<b>Outras abordagens: <i>Wavelets</i> e Filtro Bilateral</b>	<b>57</b>
4.1	Transformada <i>Wavelets</i> . . . . .	57
4.1.1	Transformada <i>Wavelets</i> Decimadas . . . . .	58
4.1.2	<i>Wavelets</i> Diádica . . . . .	60
4.2	Método Adaptativo de Super-resolução de Frame Único com <i>Wavelets</i> Redundantes . . . . .	61
4.2.1	Descrição do Algoritmo . . . . .	61
4.2.2	Super-Resolução em frame único . . . . .	62
4.2.3	Aplicação da Transformada <i>Wavelets</i> . . . . .	63

4.3	Método de Interpolação utilizando o Filtro Bilateral . . . . .	65
4.3.1	Descrição do método . . . . .	66
4.3.2	Separação em Camadas . . . . .	66
4.3.3	Filtragem da Camada de Detalhes . . . . .	67
4.3.4	Interpolação das Camadas . . . . .	68
4.3.5	Composição das camadas com realce nos Detalhes . . . . .	70
<b>5</b>	<b>Avaliação Qualitativas e Quantitativas das Imagem Digitais</b>	<b>72</b>
5.1	Métricas Clássicas de Qualidade . . . . .	72
5.1.1	Erro Máximo . . . . .	73
5.1.2	Erro Médio Absoluto . . . . .	73
5.1.3	Erro Médio Quadrático . . . . .	73
5.1.4	Raíz do Erro Médio Quadrático . . . . .	74
5.1.5	Erro Médio Quadrático Normalizado . . . . .	74
5.1.6	Relação Sinal-Ruído de Pico . . . . .	74
5.1.7	Relação Sinal-Ruído . . . . .	75
5.1.8	Covariância . . . . .	75
5.1.9	Coeficiente de Correlação . . . . .	76
5.2	Problemas na Avaliação de Qualidade de Imagens . . . . .	76
5.2.1	Índice de Similaridade Estrutural . . . . .	79
<b>6</b>	<b>Análise e Comparação dos Métodos Estudados</b>	<b>82</b>
6.1	Análise Visual dos métodos . . . . .	82
6.1.1	Interpolação por Replicação . . . . .	83
6.1.2	Interpolação Bilinear . . . . .	85
6.1.3	Interpolação Bicúbica. . . . .	86
6.1.4	Interpolação <i>Sinc</i> . . . . .	88
6.1.5	Interpolação Localmente Adaptativo . . . . .	90
6.1.6	Interpolação NEDI . . . . .	91
6.1.7	Interpolação iNEDI . . . . .	93
6.1.8	Interpolação ICBI . . . . .	95
6.1.9	Interpolação utilizando <i>wavelets</i> Redundantes . . . . .	96
6.1.10	Interpolação utilizando Filtro Bilateral . . . . .	98
6.2	Imagens Coloridas . . . . .	99
6.2.1	Modelos de Cores. . . . .	100
6.2.2	Aplicações dos métodos em imagens coloridas. . . . .	103
6.3	Comparação dos métodos . . . . .	104
6.3.1	Comparação métrica. . . . .	105
6.3.2	Análise Visual . . . . .	110

<b>7</b>	<b>Considerações Finais</b>	<b>126</b>
	<b>Referências Bibliográficas</b>	<b>129</b>
<b>A</b>	<b>Apêndice A</b>	<b>133</b>
A.1	Interpolação por convolução, função <i>sinc</i> . . . . .	133
A.2	Interpolação Localmente adaptativa . . . . .	134

# Introdução

Inúmeras são as aplicações de técnicas de interpolação em processamento de imagens digitais. Entre elas estão o uso no processo de redimensionamento de escala (*zoom*) e em rotação de imagens. O aprimoramento dos métodos existentes para ampliar a aplicabilidade tanto em áreas médicas, forenses, industriais ou de entretenimento é essencial. Desenvolver novos algoritmos que produzam imagens de alta resolução espacial sem introduzir ruídos ou distorções é uma exigência cada vez maior. O processo de *zoom* tanto quanto a rotação de imagens são muito utilizados em aparelhos celulares e em programas de edição de fotos.

Em sensoriamento remoto, tal estudo possui importantes aplicações pois melhoram a interpretação e a identificação de alvos. Para a obtenção de boas imagens em alguns satélites são utilizados sensores com alta resolução espacial. Contudo, apesar de existirem sensores comerciais deste tipo, o alto custo os torna inviável para o uso em algumas aplicações. Neste sentido o estudo de super-resolução é uma alternativa para a aquisição dessas imagens. Boa parte dos satélites utilizam sensores *Charged Coupled Device* (CCD). Alguns exemplos são: QuickBird; Ikonos II e CBERS 1, CBERS 2 e CBERS 2B (CBERS do inglês, *China Brazil Earth Resources Satellite*).

Em computação gráfica, especificamente em jogos digitais, uma interpolação adequada melhora o visual de movimentação de determinados objetos e diminui artefatos como *aliasing*. Em vídeos, a interpolação é feita entre frames sequenciais para proporcionar mais aparência de suavidade e continuidade dos movimentos. Essa mesma técnica é utilizada nas transições das imagens animadas (*Graphics Interchange Format*, GIF). Desta forma, as emendas ou tremores são atenuados, pois quanto mais imagens forem inseridas por interpolação maior será a sensação de continuidade do movimento.

O grande desafio dos métodos de interpolação está em melhorar a resolução espacial preservando as informações originais, sem acrescentar novos artefatos à imagem. Várias

---

técnicas são utilizadas para este intuito, contudo, quanto mais sofisticado o algoritmo maior seu custo computacional.

## 1.1 Técnicas de Interpolação

A interpolação é, basicamente, o processo que utiliza dados conhecidos para estimar valores em pontos desconhecidos. Em imagens, pode ser utilizada para a mudança de escalas, morfismos ou para rotações.

Existem várias técnicas que utilizam abordagens distintas de interpolação conservando os aspectos de detalhes da imagem. Os métodos avaliados neste trabalho podem ser classificados (Kumar [20]) em três tipos distintos:

- Técnicas Lineares

Utilizam filtros espaciais lineares (Gonzalez [13]) para a interpolação das imagens. Essa filtragem é efetuada utilizando uma vizinhança ou uma operação pré-definida realizada sobre os pixels da imagem. Os filtros mais comuns são: a interpolação por vizinho mais próximo, bilinear, bicúbica, quadrática, gaussiana e outros tipos de funções *splines* (Hou [15], Paik [21], Späth [29]);

- Técnicas Não-lineares

Melhoram a qualidade utilizando restrições para cada característica da imagem. Battiato [2] desenvolve um algoritmo baseado neste tipo de técnica. Já Jiang [18] otimiza uma função com base na aproximação do gradiente da imagem de alta-resolução estimada a partir da imagem de baixa-resolução. Os dois trabalhos buscam preservar as bordas definindo restrições baseadas nas direções encontradas em sua orientação;

- Técnicas de Transformadas

Usam a decomposição de uma imagem em multi-resoluções e aplicam a interpolação em cada nível obtido. Um exemplo disso é o uso da transformada *wavelets* que separa a imagem em vários sub-níveis com componentes decimadas. Desta forma, são adquiridas informações relevantes no domínio das *wavelets* que não estão aparentes em seu domínio original. Algumas aplicações de transformadas *wavelets* no processo de interpolação são dadas por Dumic [7] que emprega diferentes técnicas de decimação da imagem original e amplia cada sub-banda individualmente. Zhu [40] propõe um novo esquema de interpolação no domínio *wavelet* com base na estimativa estatística do sinal. Com o conhecimento do comportamento das bordas e as estatísticas de sinais locais, a estimativa é capaz de aumentar arestas importantes para manter a consistência da intensidade das altas frequências. Esta abordagem

visa a síntese dos componentes de altas frequências da imagem ampliada, adaptando a interpolação de acordo com o conteúdo de frequência contida em cada nível de decomposição.

## 1.2 O processo de *Zoom*

O termo *zoom* vem do inglês (*zoom lens*) e é designado a tipos específicos de lentes fotográficas que possuem a capacidade de ampliar a imagem que o usuário pretende capturar. Contudo, em processamento de imagem o termo *zoom* possui outro significado. Para diferenciar um do outro, são utilizados os termos *zoom* óptico (ampliação de imagens através de lentes) e *zoom* digital (em tratamento de imagens digitais).

O *zoom* digital é o processo pelo qual se amplia ou reduz a dimensão de uma imagem utilizando algum método digital para a reamostragem. Estes métodos são os de interpolação de imagens. A expressão *zoom* digital é popularmente utilizada quando se quer ampliar uma parcela ou a totalidade de uma imagem por meio digital. O *zoom* pode ser separado em dois tipos: o *zoom in* e o *zoom out*. O primeiro é denominado ao processo de reamostragem da imagem por um fator maior que um. Isto é, quando se deseja redimensionar a imagem para um tamanho maior em comparação ao original. Já o segundo termo é utilizado quando se deseja redimensionar o tamanho da imagem para uma escala menor. Matematicamente, se  $f : U \subset \mathbb{R}^2 \rightarrow \mathbb{R}$  é uma imagem e  $\alpha \in \mathbb{R}^+$ . Então  $g : V \subset \mathbb{R}^2 \rightarrow \mathbb{R}$  é denominado *zoom* de  $f$  se:

$$g(\alpha x, \alpha y) = f(x, y) \quad (1.1)$$

para todo  $(x, y) \in U$ .

Observe que pela Expressão 1.1, temos três situações:

- Se  $\alpha > 1$ , então a imagem  $g$  é uma ampliação da imagem  $f$  (*zoom in*);
- Se  $\alpha = 1$ , então a imagem  $g$  é exatamente a imagem  $f$ ;
- Se  $\alpha < 1$ , então a imagem  $g$  é uma redução da imagem  $f$  (*zoom out*).

Neste trabalho são estudados os métodos de interpolação que tem como consequência a ampliação da imagem original. Por este motivo, de agora em diante toda vez que for utilizado o termo isolado *zoom*, este corresponderá ao *zoom in*.

Para exemplificar, considere que seja aplicada em uma imagem o *zoom* pelo fator de dois. Isto significa que tanto a dimensão da altura, como a largura da nova imagem serão o dobro das dimensões da original (Figura 1.1).

Muitos métodos que são estudados neste trabalho são interpolações que ampliam a imagem pelo fator de  $2^n$  com  $n \in \mathbb{N}$ . Na Figura 1.1, os pontos em brancos são exatamente os pixels que serão interpolados.

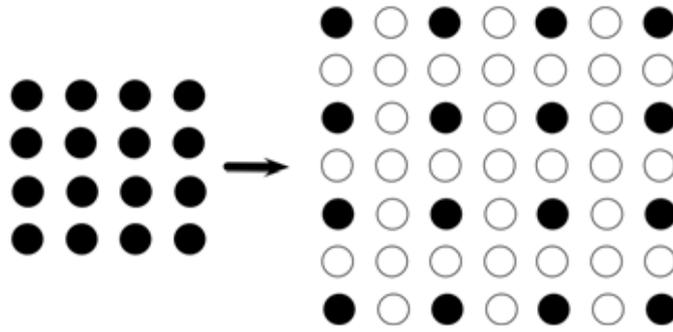


Figura 1.1: Ilustração de *zoom* em uma imagem pelo fator de dois. Os pontos pretos representam os pixels da imagem original. Os pontos brancos são os pixels a serem interpolados.

### 1.2.1 Amostragem e Reamostragem

A amostragem é o procedimento pelo qual se obtém informação sobre um todo, examinando-se apenas uma parte do mesmo. A reamostragem, por sua vez, introduz dados com o intuito de aumentar o espaço amostral com o qual se trabalha. O processo de *zoom*, por exemplo, é um tipo de reamostragem, em que ao expandir (ou contrair) uma imagem o resultado final será o conjunto de pixels originais juntamente com os novos inseridos (ou removidos). Para isso são utilizados métodos de interpolação, sendo a ferramenta essencial para a reamostragem pois insere informações através de dados previamente conhecidos.

Para que ocorra uma boa reamostragem sem a inserção de artefatos, o sinal deve obedecer ao teorema conhecido como “Teorema da Amostragem”, o qual estabelece que a partir de uma amostra discreta, sob algumas hipóteses, uma função pode ser completamente determinada, ou seja, totalmente reamostrada.

**Teorema 1** (Teorema de Nyquist (ou da Amostragem)). *Seja  $x(t)$  um sinal de banda limitada com  $x(\omega) = 0$ , se  $|\omega| > \Omega$ ,  $\forall 0 < \Omega < \infty$ . Então  $x(t)$  é determinado de modo único pelas amostras  $x(nT)$  com  $n = 0, \pm 1, \pm 2, \dots$  se  $\omega_s > 2 \cdot \Omega$ , em que  $\omega_s = \frac{2\pi}{T}$ , com  $T$  sendo o período.*

Se algum sinal reconstruído não obedecer a este teorema, alguns artefatos inseridos pela má reamostragem degradará o sinal, e este apresentará o efeito *aliasing*. Segundo Gonzalez [13], o efeito *aliasing* corresponde ao caso da subamostragem, pois os períodos de cada sinal se sobrepõe. Observe na Figura 1.2, o efeito *aliasing* em um sinal unidimensional. O sinal resultante possui reamostragem totalmente degradada em relação ao sinal original.

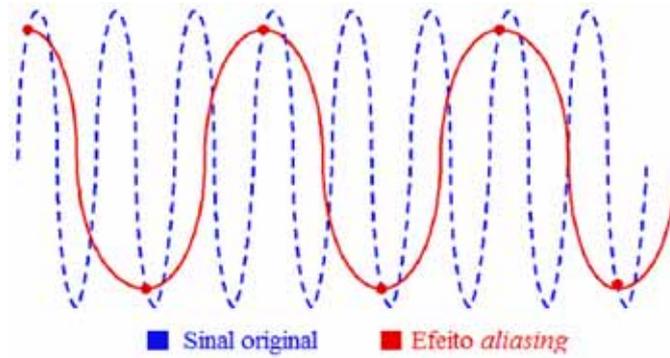
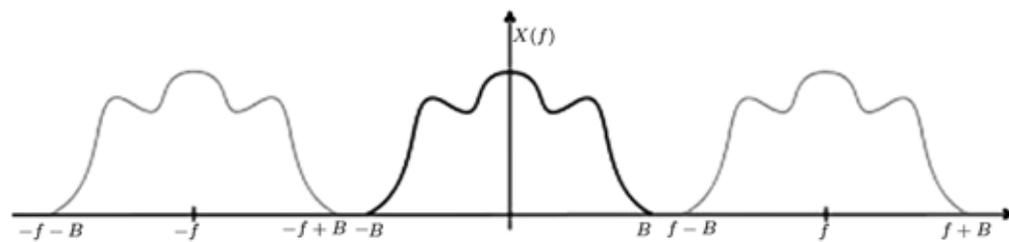
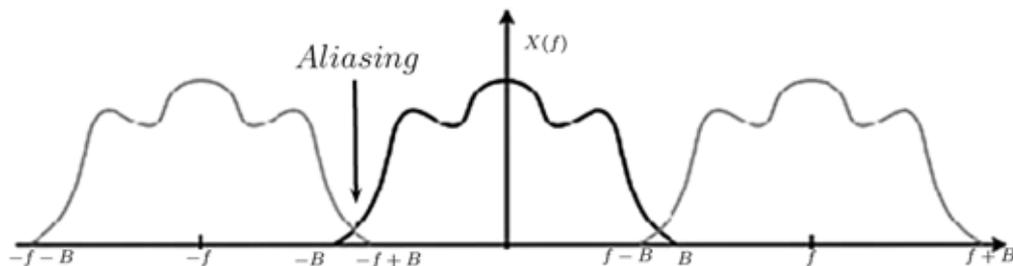


Figura 1.2: Sinal original e sua reamostragem com o efeito *aliasing*. Fonte: Retirado e adaptado a partir de [38].

Observe que na Figura 1.2, o sinal reamostrado não obedece ao teorema da amostragem. Desta maneira há perda da informação original e o sinal resultante é totalmente degradado. Na Figura 1.3 está outro exemplo do efeito *aliasing* em um sinal unidimensional.



(a) Sinal Original



(b) Sinal com efeito *aliasing*

Figura 1.3: (a) Sinal que está de acordo com o Teorema; (b) Sinal com a presença do efeito *aliasing*. Fonte: Retirado e adaptado a partir de [38].

O efeito *aliasing* degrada a imagem original inserindo ou retirando informações. Na Figura 1.4 (b), é possível ver que as listras da calça sofrem uma degradação se comparadas às listras da calça na imagem original. O efeito *aliasing*, quando introduz serrilhamento nas bordas, é denominado de efeito *jaggie*.

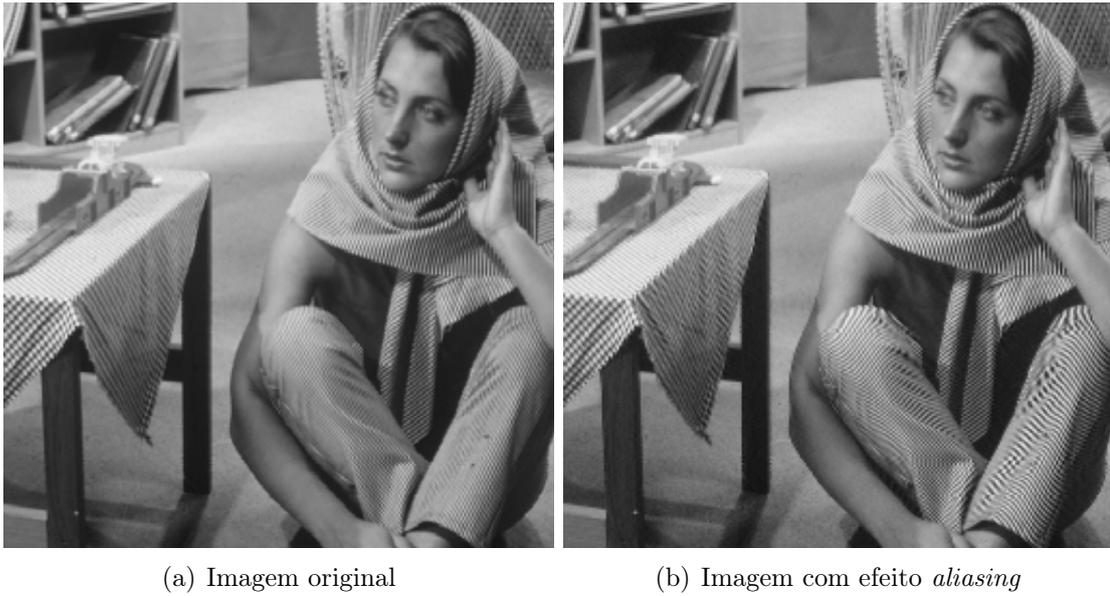


Figura 1.4: (a) Imagem original. (b) Imagem reduzida pela metade de seu tamanho por exclusão de pixels. Fonte: Adaptado a partir de [13].

Os *jaggies* são tipos específicos do efeito *aliasing* e são mais perceptíveis em regiões de alta frequências, ou seja, geralmente em regiões de bordas.

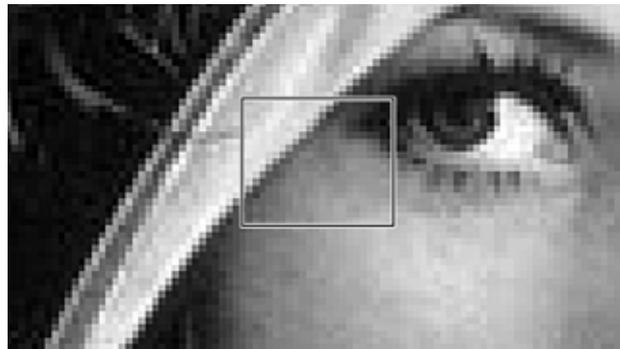


Figura 1.5: Imagem do olho da Lena apresentando o efeito *jaggie*.

É possível observar na Figura 1.5 que nas bordas do chapéu e ao redor da íris dos olhos ocorrem o efeito serrilhado.

A seguir na seção 1.3 será apresentado como este trabalho está organizado.

## 1.3 Organização do Trabalho

A dissertação está organizada em sete capítulos, como se segue:

No Capítulo 1 está a introdução deste trabalho, o qual é discutido o processo de *zoom* e classificados alguns tipos de técnicas de interpolação de imagens digitais. Também é apresentado a organização geral do trabalho.

No Capítulo 2, encontram-se os métodos mais usuais da literatura, como por exemplo a interpolação bicúbica [13, 28] utilizados em *softwares* de manipulação de imagens como *Photoshop* e em aplicativos de celulares. Os métodos de interpolação por vizinho mais próximo (*nearest*), bilinear e interpolação por polinômio de Lagrange também são discutidos. Por último também é apresentada a interpolação por convolução, em que o processo de redimensionar as escalas ocorrem no domínio das frequências.

No Capítulo 3 são apresentados métodos mais recentes que são baseados em bordas direcionais. O primeiro algoritmo é de Battiato [2], que confere várias condições impostas e através de quatro estágios verifica a presença de altas frequências em regiões locais e atribui ao novo pixel o valor calculado de acordo com a presença ou ausência de bordas. Em seguida o método de Kumar [20] que faz alterações no método do Battiato [2] e utiliza uma distribuição de pesos nos valores da região vizinha local para o cálculo. A abordagem feita por Li [22] utiliza estimativas estatísticas com correlações locais entre a nova imagem e a original. Aperfeiçoamentos deste método são propostos por Giachetti [10], que introduz dois novos algoritmos, sendo que o primeiro se baseia em [22] com mudanças no código para reduzir artefatos presentes em bordas, enquanto que o segundo utiliza as segundas derivadas direcionais para uma melhor estimativa das intensidades de cinza dos novos pixels.

O Capítulo 4 tem como foco principal apresentar outras técnicas de interpolação, que utilizam ferramentas como *wavelets* [26, 37] e filtro bilateral [14, 33].

No Capítulo 5 é realizada uma análise dos problemas envolvidos na comparação entre imagens e as métricas resultantes [36]. Métricas mais aprimoradas abrangem vários campos além da estatística. Para melhorar o cálculo da qualidade entre duas imagens, é necessário levar em consideração o processo de percepção realizado pelo sistema visual humano. Quanto mais as métricas se aproximarem desta interpretação, mais as medidas quantitativas vão refletir como a visão humana recebe e identifica semelhanças e diferenças entre duas imagens.

No Capítulo 6 é feita uma análise dos métodos e como cada um se comporta próximo à bordas. Também são mostradas algumas comparações visuais por meio de aplicações em cinco imagens que diferem nas direções dos contornos, nos detalhes e nas texturas. Algumas imagens possuem regiões mais homogêneas, enquanto outras são ricas em informações. Após ser feita uma análise qualitativa desses resultados, mostrando as deficiências

e as qualidades dos métodos, são apresentados as métricas de comparação das imagens resultantes comparados com as originais.

Finalmente, no Capítulo 7 estão as conclusões feitas por meio das comparações realizadas no Capítulo 6. Também são indicados estudos futuros como contribuição do desenvolvimento dos métodos estudados.

No Apêndice A estão as implementações de alguns algoritmos na linguagem Matlab.

## Métodos Clássicos

Este capítulo está dividido em cinco seções que apresentam as técnicas de interpolação mais clássicas encontradas na literatura de processamento de imagens, que são de simples compreensão e de fácil implementação. Os métodos presentes neste capítulo são: replicação (ou vizinho mais próximo), bilinear, bicúbico, interpolação por polinômios de Lagrange e interpolação utilizando a função discreta *sinc*.

### 2.1 Interpolação por vizinho mais próximo

O método de interpolação por vizinho mais próximo, também é conhecida como *pixel replication*, é a interpolação mais simples de ser implementada, contudo, apresenta desvantagens por causar distorções em detalhes finos e frequentemente apresenta o efeito *jaggie* em regiões que apresentam bordas. A ideia dessa técnica é atribuir ao novo valor interpolado o valor do nível de cinza do pixel mais próximo da imagem original (Figura 2.1).

Pode-se expressar a interpolação por vizinho mais próximo por meio da Equação 2.1, sendo  $f(x', y')$  o pixel a ser interpolado, em que  $dx$  e  $dy$  são as distâncias entre os pontos originais e os interpolados nas direções  $x$  e  $y$ , respectivamente. Ou seja,  $dx = x' - x$  e  $dy = y' - y$ .

$$f(x', y') = \begin{cases} f(x, y) & \text{para } dx < 0.5 \text{ e } dy < 0.5 \\ f(x + 1, y) & \text{para } dx \geq 0.5 \text{ e } dy < 0.5 \\ f(x, y + 1) & \text{para } dx < 0.5 \text{ e } dy \geq 0.5 \\ f(x + 1, y + 1) & \text{para } dx \geq 0.5 \text{ e } dy \geq 0.5 \end{cases} \quad (2.1)$$

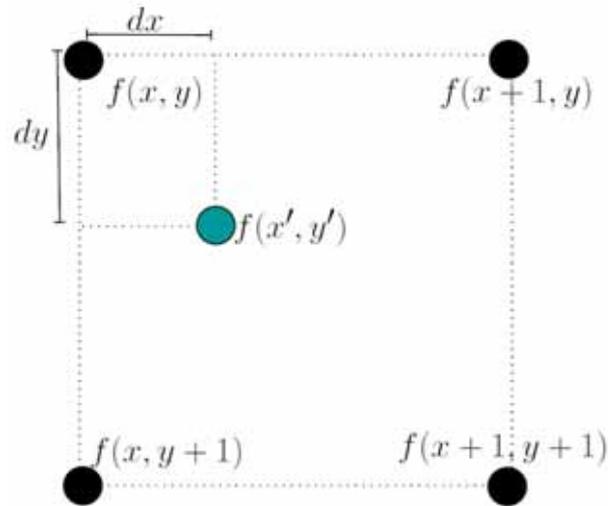


Figura 2.1: Esquema da interpolação do vizinho mais próximo. Fonte: Adaptado a partir de Pedrini [28].

Observe na Figura 2.2, uma imagem resultante da interpolação pelo vizinho mais próximo. Note o efeito serrilhado na borda do chapéu e nas curvas das bochechas do rosto ampliado.



(a) Imagem original.

(b) Região da imagem ampliada.

Figura 2.2: Ampliação da região do rosto utilizando a interpolação pelo vizinho mais próximo.

## 2.2 Interpolação Bilinear

A interpolação bilinear utiliza a média ponderada dos pixels originais que se encontram ao redor do novo pixel, como mostrado na Figura 2.1. De acordo com Pedrini [28], a interpolação bilinear é dada pela Equação 2.2.

$$f(x', y') = (1 - dx) \cdot (1 - dy) \cdot f(x, y) + dx \cdot (1 - dy) \cdot f(x + 1, y) \\ + (1 - dx) \cdot dy \cdot f(x, y + 1) + dx \cdot dy \cdot f(x + 1, y + 1) \quad (2.2)$$

Se observarmos o exemplo da Figura 2.1, o valor interpolado  $f(x', y')$  está mais próximo de  $f(x, y)$  do que  $f(x + 1, y + 1)$  e portanto o peso de  $f(x, y)$  deverá ser maior do que o peso de  $f(x + 1, y + 1)$ .

A distância máxima que estes termos podem obter é o valor de um, pois esta é a distância entre dois pixels da imagem original na mesma linha ou coluna, por exemplo,  $f(x, y)$  e  $f(x, y + 1)$ . Quanto mais distante o pixel interpolado estiver de algum ponto vizinho, menor será a influência que sofrerá deste pixel. Isto é, se  $f(x', y')$  estiver a uma distância  $dx$  de  $f(x, y)$ , então o peso atribuído a  $f(x', y')$  será de  $(1 - dx)$ , pois quanto maior for  $dx$ , menor será  $(1 - dx)$ .

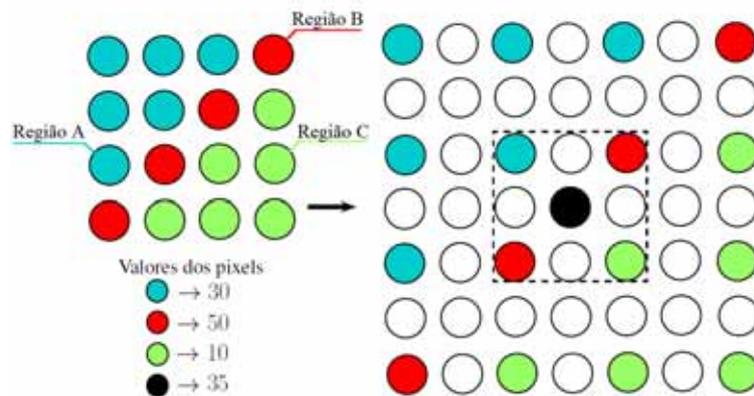


Figura 2.3: Representação de como ocorre a média ponderada na imagem utilizando a interpolação bilinear. A intensidade do pixel interpolado possui a interferência dos valores de seus vizinhos, causando a suavidade nas bordas.

O que ocorre nas bordas é a interferência de níveis de pixels de regiões distintas ao seu redor. Apesar da média ser ponderada, o fato de utilizar apenas os quatros vizinhos mais próximo ao seu redor, proporciona à imagem transições mais suaves em locais de altas frequências. Observe a Figura 2.3. Nela o ponto pintado de preto representa um dos novos pixels interpolados e este pertence à borda cuja intensidade nível de cinza é dada por  $B = 50$ . Considere que os pontos das regiões A e C tem intensidades 30 e 10 respectivamente, como indicado. O método, depois de realizar os cálculos, atribui ao ponto preto o valor de 35, apesar desse ponto pertencer à borda que possui valor 50.

Note que com essa interpolação, o efeito de serrilhado é diminuído, porém a imagem fica borrada (Figura 2.4).



Figura 2.4: Ampliação da região do rosto pela interpolação bilinear.

## 2.3 Interpolação Bicúbica

A interpolação bicúbica possui o mesmo princípio da interpolação bilinear, porém, utiliza os dezesseis pixels vizinhos ao invés de quatro pixels. É atribuído a cada um o peso correspondente à sua distância geométrica ao novo pixel e, por meio do uso de pesos baseados na *spline* cúbica, é calculado o seu valor de intensidade. A Figura 2.5 mostra o esquema da interpolação bicúbica.

Uma *B-spline* é uma *spline* que possui o menor subconjunto fechado do domínio em relação a um determinado grau  $B$ . Uma *spline* é uma curva definida matematicamente por dois ou mais pontos de controle (Chui [6]). A expressão geral da interpolação bicúbica pode ser definida pela função *B-spline* cúbica dada por:

$$f(x', y') = \sum_{m=-1}^2 \sum_{n=-1}^2 f(x+m, y+n) R(m-dx) R(dy-n) \quad (2.3)$$

em que,

$$R(s) = \frac{1}{6} [P(s+2)^3 - 4P(s+1)^3 + 6P(s)^3 - 4P(s-1)^3] \quad (2.4)$$

$$P(t) = \begin{cases} t, & t > 0 \\ 0, & t \leq 0 \end{cases}$$

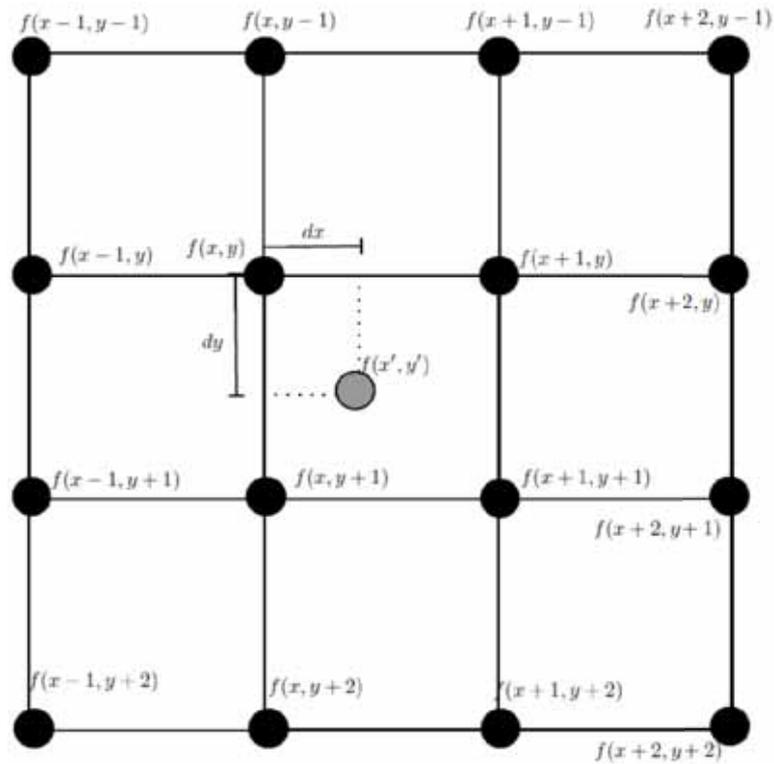


Figura 2.5: Esquema da interpolação bicúbica.  
Fonte: Adaptado a partir de Pedrini [28].

O método da interpolação bicúbica não causa o serrilhamento tão exagerado como causado pelo vizinho mais próximo e também não suaviza tanto a imagem quanto a interpolação bilinear. É frequentemente utilizado em *softwares* de edições de imagens. Observe um exemplo de aplicação da interpolação bicúbica na Figura 2.6:

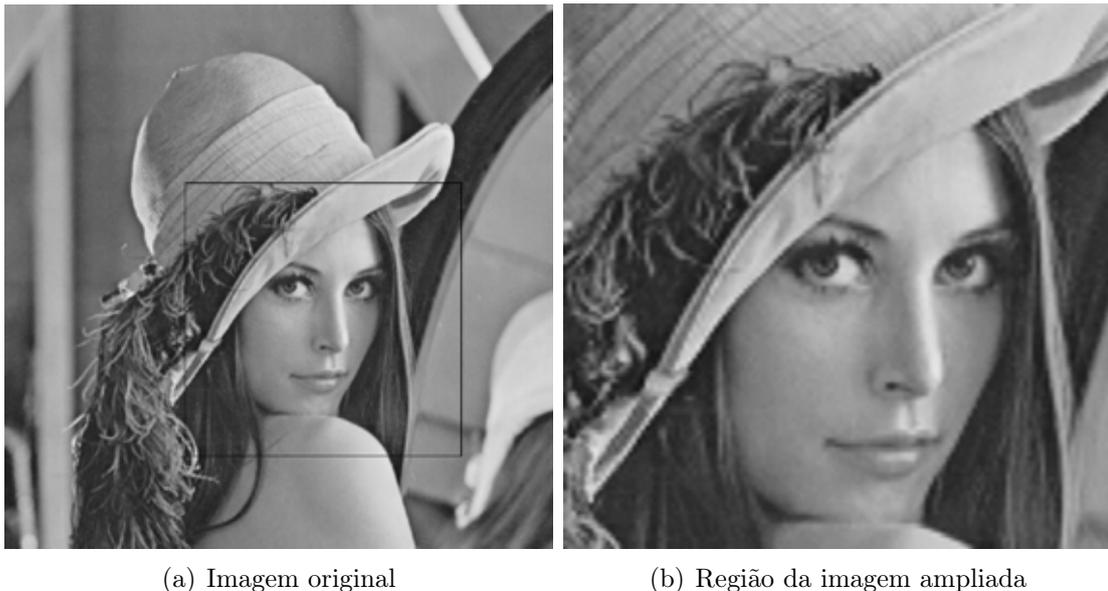


Figura 2.6: Ampliação da região do rosto pela interpolação bicúbica.

## 2.4 Interpolação por Polinômios de Lagrange

O método de interpolação por polinômios de Lagrange também utilizam os dezesseis pixels vizinhos do novo pixel para calcular o seu valor de intensidade. Antes de apresentar a expressão do método, é necessário definir o que é um polinômio de Lagrange.

Considere  $P_n(x)$  definido da seguinte maneira:

$$P_n(x) = \sum_{i=0}^n L_i(x)f(x_i) \quad (2.5)$$

em que

$$L_i(x) = \prod_{\substack{j=0 \\ j \neq i}}^n \frac{x - x_j}{x_i - x_j} \quad (2.6)$$

é chamado de polinômio interpolador de Lagrange de grau  $n$ . Segundo Späth [29] esta forma pode ser reescrita para duas dimensões:

$$P_{ij}(x, y) = \sum_{k=1}^N \sum_{\ell=1}^N a_{ijk\ell} (x - x_i)^{k-1} (y - y_j)^{\ell-1} \quad (2.7)$$

em que  $a_{ijk\ell}$  são os coeficientes do polinômio de Lagrange. Logo, a expressão do método de interpolação pode ser dada pela Equação 2.8.

$$f(x', y') = \frac{-dy(dy-1)(dy-2)L(1)}{6} + \frac{(dy+1)(dy-1)(dy-2)L(2)}{2} \quad (2.8)$$

$$+ \frac{-dy(dy+1)(dy-2)L(3)}{2} + \frac{dy(dy+1)(dy-1)L(4)}{6}$$

em que

$$L(n) = \frac{-dx(dx-1)(dx-2)f(x-1, y+n-2)}{6} +$$

$$\frac{(dx+1)(dx-1)(dx-2)f(x, y+n-2)}{2} + \quad (2.9)$$

$$\frac{-dx(dx+1)(dx-2)f(x+1, y+n-2)}{2} +$$

$$\frac{dx(dx+1)(dx-1)f(x+2, y+n-2)}{6}$$

Nas Equações 2.8 e 2.9,  $dx$  e  $dy$  são as distâncias entre o ponto interpolado e o seu vizinho (que pertence a imagem original) nas direções horizontais e verticais, respectivamente. Este método apresenta menor tempo computacional. Contudo, os resultados são visualmente semelhantes comparativamente ao método da interpolação bicúbica.

## 2.5 Interpolação com base na Convolução

Os métodos de interpolação baseada na convolução (Unser [35]) possuem importantes aplicações no processo de rotação de imagens. Dada duas funções  $f$  e  $g$ , a convolução pode ser representada por  $f * g$  e definida no caso contínuo da seguinte maneira:

$$f * g = \int_{-\infty}^{+\infty} f(\tau)g(t - \tau)d\tau \quad (2.10)$$

Seja o espaço de Hilbert  $L_2$  (Kreyszig [19]) o conjunto de todas as sequências  $s_i$ , com  $s : \mathbb{R} \mapsto \mathbb{R}$  tal que,

$$\sum_{i=1}^{\infty} |s_i|^2 < \infty \quad (2.11)$$

e sua norma dada pelo produto interno:

$$\langle s(x), r(x) \rangle_{L_2} = \int_{-\infty}^{+\infty} s(x)r(x)dx \quad (2.12)$$

com  $r(x) \in L_2$ .

Pode-se definir o espaço  $V(\varphi) \subset L_2$  como o conjunto das convoluções das funções  $\varphi$  com os elementos de  $L_2$  :

$$V(\varphi) = \left\{ s(x); s(x) = \sum_{k \in \mathbb{Z}} c(k)\varphi(x - k), c \in L_2 \right\} \quad (2.13)$$

O conjunto  $\varphi(x - k)_{k \in \mathbb{Z}}$  é uma base de *Riesz* do espaço  $V(\varphi)$  (Unser [35]), e isso implica que  $\exists A, B \in \mathbb{R}$  com  $A > 0$  e  $B > 0$ , ou seja,  $A$  e  $B$  constantes estritamente positivas, em que:

$$A \leq \sum_{k \in \mathbb{Z}} |\Phi(\omega + 2\pi k)|^2 \leq B \quad (2.14)$$

o termo  $\Phi(\omega)$  é a Transformada de Fourier da função  $\varphi(x)$ . Desta maneira, segundo o teorema da representação de Riesz (Brezis [3]), assegura-se que para cada  $s(x) \in V(\varphi)$ ,  $s(x)$  é unicamente determinado pela sequência de coeficientes  $c(k)$ .

Considerando que a convolução está intimamente ligada à transformada de Fourier, será apresentada um breve estudo sobre o assunto.

### 2.5.1 Transformada de Fourier

Seja  $f(t)$  uma função absolutamente integrável com integral finita:

$$\int_{-\infty}^{+\infty} |f(t)| dt < \infty \quad (2.15)$$

tem-se que a transformada de Fourier da função  $f(x)$  é dada da seguinte maneira:

$$\hat{f}(\omega) = \int_{-\infty}^{+\infty} f(x) e^{-i\omega x} dx \quad (2.16)$$

e sua inversa é expressa por:

$$f(x) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} \hat{f}(\omega) e^{i\omega x} d\omega \quad (2.17)$$

Seja uma função contínua de duas variáveis  $x$  e  $y$ , a transformada de Fourier no caso bidimensional é representado por:

$$\hat{f}(\mu, \nu) = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} f(x, y) \cdot e^{-i2\pi(\mu x + \nu y)} dx dy \quad (2.18)$$

e

$$f(x, y) = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} \hat{f}(\mu, \nu) \cdot e^{i2\pi(\mu x + \nu y)} d\mu d\nu \quad (2.19)$$

em que  $\mu$  e  $\nu$  são variáveis de frequência. Já no caso discreto, tem-se as séries de Fourier, apresentada na Equação 2.20.

$$\bar{x}_k = \frac{1}{n} \sum_{j=0}^{n-1} f_j \cdot e^{-\frac{2\pi i}{n} jk}, \quad (2.20)$$

com  $f_j$  dada por:

$$f_j = \sum_{k=0}^{n-1} \bar{x}_k \cdot e^{\frac{2\pi i}{n} jk} \quad (2.21)$$

A sequência de  $f_j$  é denominada de coeficientes de Fourier, com  $j \in \{1, 2, \dots, n-1\}$ .

Em imagens digitais geralmente não há a condição de periodicidade, isto é,  $f(x, 0) \neq f(x + N, 0)$  para qualquer  $x$  em uma imagem  $N \times N$ . Assim, a transformada de Fourier discreta (DFT) pode ser aplicada na função  $f$  no intervalo  $0 \leq x \leq N-1$  e  $0 \leq y \leq N-1$ . Para o caso bidimensional, tem-se:

$$\hat{f}(\mu, \nu) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) \cdot e^{-i2\pi(\frac{\mu x}{M} + \frac{\nu y}{N})} dx dy \quad (2.22)$$

e

$$f(x, y) = \sum_{\mu=0}^{M-1} \sum_{\nu=0}^{N-1} \hat{f}(\mu, \nu) \cdot e^{i2\pi(\frac{\mu x}{M} + \frac{\nu y}{N})} d\mu d\nu \quad (2.23)$$

Considere que  $\mathfrak{S}\{f(t) * g(t)\}$  seja a aplicação da transformada de Fourier na convolução de duas funções  $f(t)$  e  $g(t)$  de uma variável contínua. Ou seja:

$$\begin{aligned}\mathfrak{F}\{f(t) * g(t)\} &= \int_{-\infty}^{+\infty} \left[ \int_{-\infty}^{+\infty} f(\tau)g(t - \tau)d\tau \right] \cdot e^{-i2\pi\mu t} dt \iff \\ \mathfrak{F}\{f(t) * g(t)\} &= \int_{-\infty}^{+\infty} f(\tau) \left[ \int_{-\infty}^{+\infty} g(t - \tau)e^{-i2\pi\mu t} dt \right] d\tau \stackrel{(*)}{\iff} \end{aligned} \quad (2.24)$$

$$\mathfrak{F}\{f(t) * g(t)\} = \int_{-\infty}^{+\infty} f(\tau) [\hat{g}(\mu)e^{-i2\pi\mu t} dt] d\tau \iff$$

$$\mathfrak{F}\{f(t) * g(t)\} = \hat{g}(\mu) \cdot \hat{f}(\mu)$$

em que tem-se a passagem (\*) de tal maneira que  $\mathfrak{F}\{g(t - \tau)\} = \hat{g}(\mu)e^{-i2\pi\mu t} dt$ .

Se for aplicado a transformada inversa de Fourier no lado direito da última equação, obtém-se  $f(t) * g(t)$ . O teorema da convolução garante que a convolução entre duas funções é o mesmo que a multiplicação de suas transformadas de Fourier. O inverso também ocorre. O teorema da convolução é expressa por:

$$\begin{aligned} f(t) * g(t) &\iff \hat{g}(\mu) \cdot \hat{f}(\mu) \\ &e \\ f(t) \cdot g(t) &\iff \hat{g}(\mu) * \hat{f}(\mu) \end{aligned} \quad (2.25)$$

## 2.5.2 Interpolação por função *sinc*

Na teoria de processamento de sinais digitais e informações, a função  $sinc(\cdot)$  normalizada pode ser definida por:

$$sinc(x) = \begin{cases} \frac{sen(\pi x)}{\pi x}, & \forall x \neq 0 \\ 1 & \text{caso contrário} \end{cases} \quad (2.26)$$

a não-normalizada não possui o número  $\pi$  acompanhando a incógnita:

$$sinc(x) = \begin{cases} \frac{sen(x)}{x}, & \forall x \neq 0 \\ 1 & \text{caso contrário} \end{cases} \quad (2.27)$$

O termo “sinc” é uma contração do nome da função em latim *sinus cardinalis* (seno cardinal). A vantagem de usar a normalizada está em aplicar a integral sobre todos os pontos, pois para qualquer  $x$  o resultado será 1.

$$\int_{-\infty}^{+\infty} \frac{sen(\pi x)}{\pi x} = 1 \quad (2.28)$$

A Equação 2.28 também pode ser representada como o produtório infinito:

$$\operatorname{sinc}(x) = \frac{\operatorname{sen}(\pi x)}{\pi x} = \prod_{n=1}^{\infty} \left(1 - \frac{x^2}{n^2}\right) \quad (2.29)$$

Outra vantagem é que os zeros do  $\operatorname{sinc}(\cdot)$  não-normalizado são múltiplos não nulos de  $\pi$  enquanto que os zeros do normalizado são inteiros não nulos. É importante destacar que:

- i A  $\operatorname{sinc}$  é uma função de interpolação passa-banda, ou seja,  $\operatorname{sinc}(0) = 1$ , e  $\operatorname{sinc}(x) = 0$  para  $x$  inteiros e não-nulos, conforme pode ser observado na Figura 2.7.

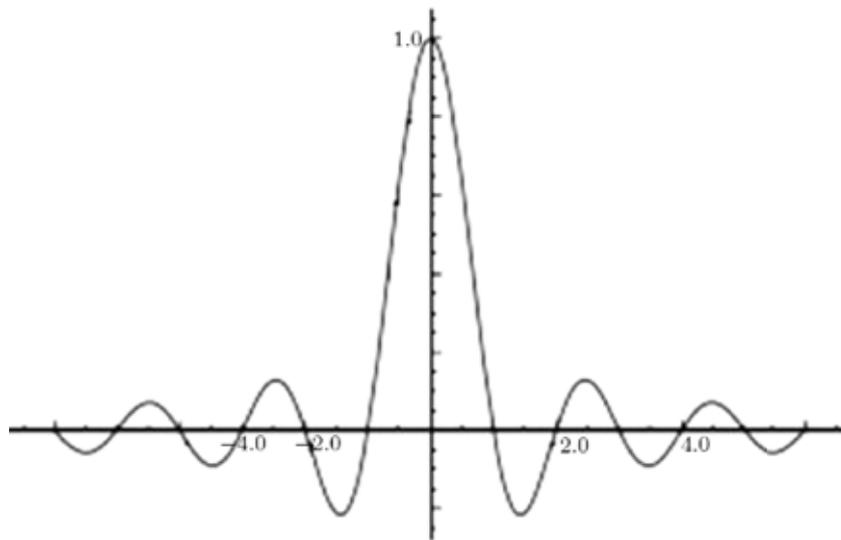


Figura 2.7: Gráfico da função  $\operatorname{sinc}(\cdot)$  normalizada.  
Fonte: Adaptado a partir de [12].

- ii As funções  $x_k(t) = \operatorname{sinc}(t - k)$  formam uma base ortonormal (Unser [34]) para as funções limitadas em banda no espaço de funções  $L(\mathbb{R})$ , cuja maior frequência angular é  $\omega(H) = \pi$ . Como a frequência angular é  $\omega = 2\pi f$ , tem-se que o ciclo de frequência mais alto é  $f_H = \frac{1}{2}$ .

Por estes motivos, de agora em diante quando for mencionada a função  $\operatorname{sinc}(\cdot)$ , esta será a normalizada. Um cálculo simples mostra que a transformada de Fourier da função  $\operatorname{sinc}(\cdot)$  é a função retangular:

$$\Pi(x) = \begin{cases} A, & |t| \leq t_0 \\ 0 & |t| > t_0 \end{cases} \quad (2.30)$$

Um dos teoremas mais importantes na teoria de sinais digitais é o teorema de Nyquist-Shannon (ou apenas Nyquist), que também é conhecido como teorema da amostragem (Teorema 1 pg. 17). O teorema de Shannon pode ser enunciado da seguinte maneira:

**Teorema 2** (Teorema de Shannon). *Seja  $f : \mathbb{R} \rightarrow \mathbb{R}$  uma função banda limitada, isto é,  $\hat{f}(\omega) = 0$ , se  $|\omega| > \Omega$  para algum  $0 < \Omega < \infty$ . Se  $\Delta t < \frac{\pi}{\Omega}$ , então  $\forall t_0 \in \mathbb{R}$ ,*

$$f(t) = \sum_{j \in \mathbb{Z}} f(t_0 + j\Delta t) \operatorname{sinc} \left( \frac{t - t_0 - j\Delta t}{\Delta t} \right) \quad (2.31)$$

Abaixo segue a Demonstração (Camargo [4]).

*Demonstração.* Considere o intervalo  $[-\Omega, \Omega]$ , expandindo a função exponencial  $e^{i\omega t}$  em sua série de Fourier, obtêm-se

$$e^{i\omega t} = \sum_{n=-\infty}^{+\infty} c_n e^{-\frac{2i\pi n\omega}{\Omega}} \quad (2.32)$$

em que

$$c_n = \frac{1}{2\Omega} \int_{-\Omega}^{\Omega} e^{i\omega t} \cdot e^{-\frac{i\pi n\omega}{\Omega}} d\omega \quad (2.33)$$

Considera-se  $\Delta t = \pi/\Omega$  e  $t_n = n\Delta t$ , temos que

$$c_n = \frac{1}{2\Omega} \int_{-\Omega}^{\Omega} e^{i\omega(t-t_n)} d\omega = \frac{\operatorname{sen}(\pi(t-t_n)/\Delta t)}{\pi(t-t_n)/\Delta t} = \operatorname{sinc} \left( \frac{t-t_n}{\Delta t} \right) \quad (2.34)$$

e então,

$$e^{i\omega t} = \sum_{n=-\infty}^{+\infty} \operatorname{sinc} \left( \frac{t-t_n}{\Delta t} \right) e^{i\omega t_n}, \omega \in [-\Omega, \Omega]. \quad (2.35)$$

Por fim, como  $f(t)$  é banda limitada, tem-se:

$$\begin{aligned} f(t) &= \frac{1}{2\pi} \int_{-\infty}^{\infty} \hat{f}(\omega) e^{i\omega t} d\omega \\ &= \frac{1}{2\pi} \int_{-\Omega}^{\Omega} \hat{f}(\omega) \sum_{n=-\infty}^{+\infty} \operatorname{sinc} \left( \frac{t-t_n}{\Delta t} \right) e^{i\omega t_n} d\omega \\ &= \sum_{n=-\infty}^{+\infty} \operatorname{sinc} \left( \frac{t-t_n}{\Delta t} \right) \frac{1}{2\pi} \int_{-\Omega}^{\Omega} \hat{f}(\omega) e^{i\omega t_n} d\omega \\ &= \sum_{n=-\infty}^{+\infty} \operatorname{sinc} \left( \frac{t-t_n}{\Delta t} \right) f(t_n), \end{aligned} \quad (2.36)$$

o que finaliza a demonstração. □

A implementação desta interpolação está no Apêndice A. A ideia utilizada está na técnica de *Zero Padding*, que é a inserção apropriada de zeros no domínio da frequência da imagem. O problema desta técnica, é que, ao inserir zeros no domínio de frequência, as regiões suaves da imagem reamostrada tendem a seguir a frequência das regiões de detalhes. Para ficar mais claro, observe a Figura 2.8. É possível observar que a inserção de zeros faz com que o sinal reamostrado mantenha as características de alguns detalhes do sinal original. A inserção de zeros no domínio do tempo pela transformada de Fourier, fornece a interpolação banda limitada no domínio de frequência. Da mesma forma, a inserção de zeros no domínio da frequência fornece interpolação no domínio do tempo. Este tipo de interpolação é a forma ideal para a reamostragem e representa justamente a imagem original com a convolução pela função  $\text{sinc}(\cdot)$ .

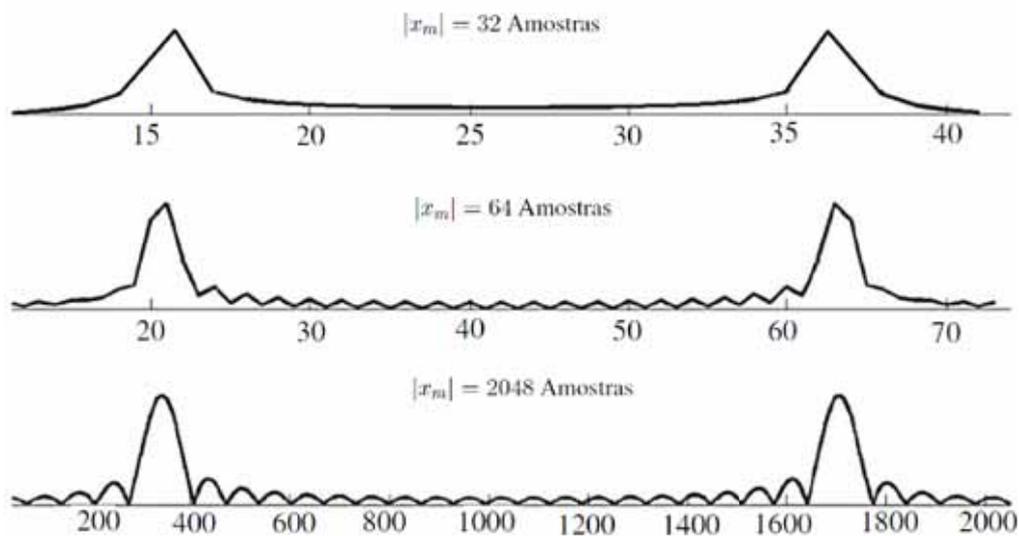


Figura 2.8: Exemplo da inserção de zeros em um sinal unidimensional.

Em imagens, a técnica *Zero Padding* introduz um efeito de “ondas”, em regiões mais homogêneas. Observe que na Figura 2.9, as regiões mais suaves da imagem possuem linhas como se fossem a propagação dos contornos das bordas.

A Figura 2.9(b) ilustra os resultados da aplicação deste método na imagem (região quadrangular) relativa a Figura 2.9(a).



(a) Imagem original

(b) Rosto Ampliado

Figura 2.9: Ampliação do rosto utilizando a interpolação pela função  $\text{sinc}(\cdot)$ .

A implementação da interpolação  $\text{sinc}$  está na seção [A.1](#) do Apêndice [A](#). Todos os outros métodos discutidos no presente capítulo não foram implementados. Para estes, foram utilizados o comando `imresize` do ambiente Matlab.

# Métodos de Interpolação Utilizando Bordas Direcionais

Neste capítulo são apresentados alguns métodos mais recentes que utilizam como base a busca da presença de bordas através da avaliação de suas direções.

Com o objetivo de preservar bordas, Battiato [2] propõe um método que tenta levar em conta informações sobre descontinuidades ou variações de luminosidade acentuadas. Para isso o algoritmo é baseado em alguns estágios, em que cada um verifica condições investigando a presença de bordas. Kumar [20] adapta o método adicionando pesos a cada pixel para a interpolação.

Em [22], Li propõe o algoritmo interpolador com bordas direcionais baseado em Allebach [1]. Este método utiliza um cálculo de estimativa da covariância local entre os pixels da imagem de baixa e alta resolução como peso da média ponderada dos pixels vizinhos. Melhorias deste algoritmo foram feitas por Giachetti [10, 11] que também apresenta o método de interpolação iterativa com base em curvatura [9]. Este interpola com base em dois passos, em que utiliza uma correlação iterativa dos pixels obtidos por meio da minimização de uma função que depende das segundas derivadas direcionais de intensidade da imagem.

Outros algoritmos de interpolação que se baseiam no método de Li foram propostos por Tam [31, 32], que aperfeiçoa a janela utilizada para o cálculo da covariância local. Esta mudança elimina o problema de acumulação de erro preservando a suavidade e a forma das bordas. Su [30] também apresenta a interpolação a partir da triangulação de dados dependentes do nível de pixel da imagens de baixa resolução. Este algoritmo possui vantagens no tempo computacional por ser mais simples do que os métodos adaptáveis. Estes dois últimos métodos não serão abordados neste trabalho, porém são exemplos de abordagens que se fundamentam na técnica de interpolação utilizando direções orientadas.

### 3.1 Algoritmo Localmente-adaptativo

Os métodos descritos nesta seção têm como objetivo aumentar a escala da imagem sem introduzir o efeito de suavização nas bordas, como ocorre nos métodos bilinear e bicúbico. O problema dos métodos clássicos é que não são capazes de realçar os detalhes de altas frequências sem introduzir artefatos, apesar de conseguir preservar as baixas frequências contidas na imagem original. O algoritmo de interpolação localmente-adaptativo utiliza os níveis de cinza dos pixels ao redor do novo para a avaliação da presença ou não de bordas.

#### – Primeiro Estágio

A imagem original é expandida por um fator de dois. Os valores de cinza da imagem original são distribuídos na nova imagem como mostrado na Figura 1.1 (pág. 17).

#### – Segundo Estágio

O algoritmo procura linha por linha todos os pontos em brancos que possuam em suas diagonais pixels da imagem original. Estes pontos são representados na Figura 3.1 pelo ponto  $X$ . Observe que os pontos referenciados genericamente por  $a$ ,  $b$ ,  $c$  e  $d$  são os pixels da imagem original. Os pontos denominados  $V_1, V_2$  são os pontos que estão entre dois pixels da imagem original na direção vertical, e  $H_1, H_2$ , na horizontal.

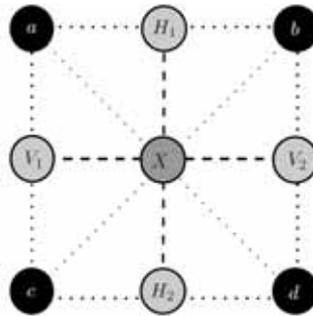


Figura 3.1: Representação do pixel  $X$  a ser interpolado no segundo estágio. Fonte: Adaptado a partir de Battiato [2].

Considere que os valores de luminância dos pontos ao redor do ponto a ser interpolado são dados por  $a$ ,  $b$ ,  $c$  e  $d$ , respectivamente, e que os valores  $T_1$  e  $T_2$  são as tolerâncias que limitam superior e inferiormente as diferenças dos níveis de cinza dos pixels da imagem atribuídos pelo usuário. Então, segundo Battiato [2], pode-se adotar os seguintes critérios:

- *uniformidade*:  $|\max(a, b, c, d) - \min(a, b, c, d)| < T_1$ , em que  $T_1$  é um valor de tolerância. Nesta etapa, o algoritmo verifica se a distância máxima entre os pontos é menor do que o valor  $T_1$  fornecido previamente. Se o valor dessa diferença for menor que o limiar então provavelmente a região é homogênea e a média é atribuída ao ponto  $X$ . Neste caso  $X = \frac{a + b + c + d}{4}$ ;
- *bordas na direção bc*: se  $|a - d| > T_2$  e  $|a - d| - |b - c| > r$ , em que  $T_2$  e  $r$  também são valores de limites e  $r > 0$ , então os níveis de cinza dos pontos  $b$  e  $c$  estão mais próximos entre si do que dos pontos  $a$  e  $d$ . Como  $a$  e  $d$  possuem disparidade em suas intensidades, então provavelmente há a presença de bordas na direção  $bc$  e o ponto  $X$  faz parte desta borda. Portanto, é atribuído ao ponto  $X$  o valor de intensidade correspondente a  $\frac{b + c}{2}$ ;
- *bordas na direção ad*: se  $|b - c| > T_2$  e  $|b - c| - |a - d| > r$ . Caso a condição anterior não seja satisfeita, é verificado se há presença de borda na direção  $ad$ . Para isto, é feita a diferença  $|b - c|$ . Se esta for maior que  $T_2$  e  $|b - c| \gg |a - d|$ , provavelmente existe uma borda que separa os pontos  $a$  e  $b$ . Portanto  $X$  recebe o valor de  $\frac{a + d}{2}$ ;
- *bordas na direção  $H_1H_2$* : se  $|a - d| > T_1$  e  $|b - c| > T_1$ , e ainda  $(a - d) * (b - c) > 0$  então  $H_1 = \frac{a + b}{2}$  e  $H_2 = \frac{c + d}{2}$ . Se não há existência de bordas nas diagonais é porque é provável que exista na horizontal ou na vertical. Para verificar qual dos dois casos, é verdadeiro, o algoritmo confere se  $|a - d|$  e  $|b - c|$  são maiores que  $T_1$ . Se isso ocorre, significa que os pixels superiores e os inferiores estão com valores muito distantes entre si.  
O próximo passo é conferir se  $(a - d) * (b - c) > 0$ . Caso esta multiplicação seja positiva, significa que o valor de  $a$  e  $b$  são ambos maiores (ou menores) do que os pontos  $d$  e  $c$ , respectivamente. Logo os pontos  $a$  e  $b$  possuem níveis de cinza mais próximos entre si e o mesmo acontece com os pontos  $c$  e  $d$ . Então é atribuído aos valores  $H_1 = \frac{a + b}{2}$  e  $H_2 = \frac{c + d}{2}$ . O ponto  $X$  é deixado indefinido.
- *bordas na direção  $V_1V_2$* : se  $|a - d| > T_1$  e  $|b - c| > T_1$ , e ainda  $(a - d) * (b - c) < 0$  então  $V_1 = \frac{a + c}{2}$  e  $V_2 = \frac{b + d}{2}$ .  
Para finalizar este estágio, caso  $|a - d| > T_1$  e  $|b - c| > T_1$  contudo  $(a - d) * (b - c) < 0$  então os valores de  $a$  está mais próximo do ponto  $c$  do que de  $b$ , e por conseguinte,  $b$  está mais próximo de  $d$  do que do ponto  $a$ . Desta maneira é atribuído a  $V_1$  o valor de  $\frac{a + c}{2}$ ; e, de  $V_2$ , o valor de  $\frac{b + d}{2}$ . Nesta etapa  $X$  também é deixado indefinido.

Observe que após este estágio ainda ficam alguns pontos  $X_s$ ,  $H_1$ ,  $H_2$ ,  $V_1$  e  $V_2$  sem valores atribuídos.

### – Terceiro Estágio

O algoritmo retorna a fazer a varredura de todas as linha da imagem procurando pelos pontos que não foram interpolados nos estágios anteriores. Estes são apresentados pelo ponto  $P$  na Figura 3.2. Para o próximo passo, considere que as letras  $a$  e  $b$  representam os pixels da imagem original e os pontos  $X_1$  e  $X_2$  são os valores de cinza provenientes do estágio anterior.

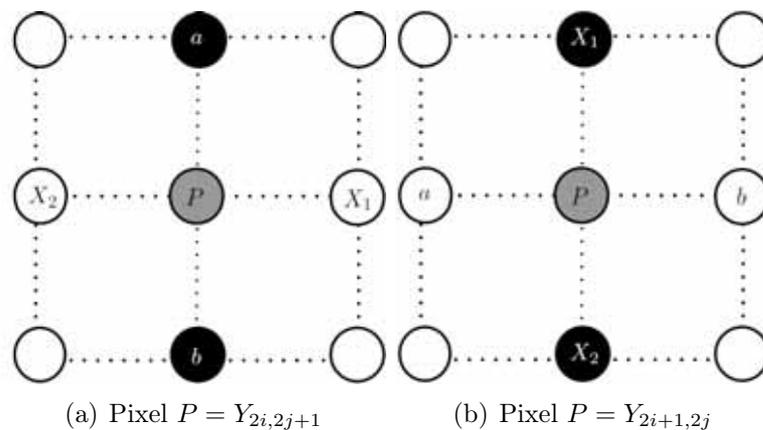


Figura 3.2: Representação da interpolação dos pontos do tipo  $Y_{2i,2j+1}$  e  $Y_{2i+1,2j}$  no terceiro estágio..  
Fonte: Adaptado a partir de Battiato [2].

- $X_1$  e  $X_2$  são pontos indefinidos; o algoritmo confere se  $|a - b| < T_1$ , se assim for, então o valor de  $P$  será  $P = \frac{a + b}{2}$ . Caso contrário,  $P$  é deixado indefinido;
- $X_1$  e  $X_2$  são pontos já definidos; verifica a existência de bordas na vertical ou na horizontal. Desta maneira, os seguintes itens são averiguados:
  - (i) (Presença de borda na direção  $X_1X_2$ ): Quando  $|a - b| > T_2$  e  $|a - b| \gg |x_1 - x_2|$ . Neste caso  $P = \frac{x_1 + x_2}{2}$ ;  
Quando isto acontece, pelo fato de  $|a - b| > T_2$ , provavelmente há uma borda que separa os valores dos pontos  $a$  e  $b$ . A distância  $ab$  é muito maior que a distância  $X_1X_2$ , ou seja, o ponto  $P$  deve pertencer a borda que passa por  $X_1X_2$ . Desta forma,  $P = \frac{x_1 + x_2}{2}$ ;
  - (ii) (Presença de borda na direção  $ab$ ): Quando  $|x_1 - x_2| > T_2$  e  $|x_1 - x_2| \gg |a - b|$ . Quando  $|x_1 - x_2| > T_2$  e  $|x_1 - x_2| \gg |a - b|$  ocorre justamente o contrário, e neste caso tem-se  $P = \frac{a + b}{2}$ ;

- Nenhuma das opções acima: Se nenhum dessas condições forem satisfeitas o ponto  $P$  é deixado indefinido e o algoritmo vai para o próximo e último estágio.

### – Quarto Estágio

Por fim, para a interpolação dos pontos deixados em branco, o método cria um vetor que possua todos os pixels vizinhos que tenham valores definidos pelos estágios anteriores e aplica a mediana neste vetor. Antes dos cálculos, contudo, todos os pixels recebem pesos adequados segundo seu valor de nível de cinza.

## 3.2 Adaptação do Algoritmo Baseado em Estágios

Este método possui três estágios, em que o primeiro consiste em ampliar a imagem em duas vezes o seu tamanho original. A implementação deste algoritmo está na seção [A.2](#) no Apêndice [A](#). O segundo e terceiro estágios serão descritos a seguir.

### – O segundo Estágio

O método verifica algumas restrições para atribuir aos novos pontos interpolados um valor que depende de seus vizinhos (pontos originais da imagem original). Para encontrar o valor destes pontos, é estipulado primeiramente um peso para cada pixel novo. Este peso será calculado de acordo com a Equação [3.1](#).

$$Peso\_do\_Pixel = |Valor\_do\_Pixel - \omega_t| \quad (3.1)$$

Como as imagens utilizadas nos testes possuem valores de níveis de cinza entre 0 e 255 tons de cinza distintos, então é escolhido a média destes valores, ou seja  $\omega_t = 128$ . Através da Equação [3.1](#) todos os pixels a serem interpolados utilizam pesos que variam com a intensidade dos originais. Para a descrição do método observe a Figura [3.3](#).

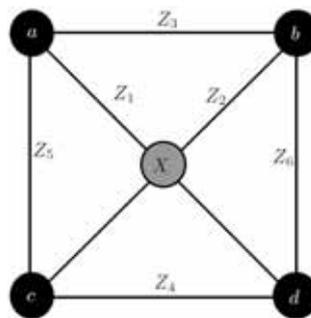


Figura 3.3: Ilustração dos pontos no segundo estágio.  
Fonte: Adaptado a partir de Kumar [\[20\]](#).

Na Figura 3.3 os pontos  $a, b, c$  e  $d$  representam os pixels da imagem original e  $X$ , o ponto a ser interpolado neste estágio. As distâncias  $Z_\kappa$ , com  $\kappa \in \{1, 2, 3, 4, 5, 6\}$ , representarão as diferenças em módulo das intensidades dos pixels como se segue:

$$\begin{aligned}
 Z_1 &= |a - d| \\
 Z_2 &= |b - c| \\
 Z_3 &= |a - b| \\
 Z_4 &= |c - d| \\
 Z_5 &= |a - c| \\
 Z_6 &= |b - d|
 \end{aligned} \tag{3.2}$$

- *Condição 1:* Verifica se todos os pontos são *uniformes*. Para tal o algoritmo depende de um fator  $K$  que será o limiar utilizado para definir se dois pontos estão muito distantes entre si ou não. No trabalho foi utilizado  $K = 10$ . Para conferir se o ponto  $X$  está em uma região homogênea, basta considerar a média dos níveis de intensidade de cada pixel até o centro dado por:

$$Z_{uniforme} = \frac{|M - a| + |M - b| + |M - c| + |M - d|}{4} \tag{3.3}$$

em que  $M = (a + b + c + d)/4$  e verificar se  $Z_{uniforme} < K$ . Se assim for, então  $X = M$ , caso contrário o algoritmo vai para a próxima condição.

- *Condição 2:* Verifica se  $Z_1 = 0$  e  $Z_2 = 0$  ou se  $Z_1$  e  $Z_2$  são menores que as distâncias horizontais e verticais. Se assim for, então:

$$X = (a \times |\omega_t - a| + b \times |\omega_t - b|) / (|\omega_t - a| + |\omega_t - b|) \tag{3.4}$$

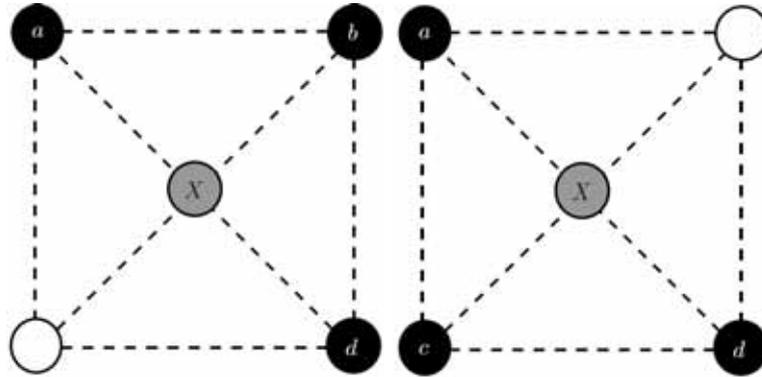
- *Condição 3:* Nesta etapa, o algoritmo verifica se tem a existência de borda na diagonal  $ad$ . Se houver nesta diagonal, a diferença entre estes pontos serão menores que as outras distâncias  $Z_k$ s. Logo o algoritmo checa:

- se  $Z_1 == 0$  ou  $Z_1 < \min \{Z_2, Z_3, Z_4, Z_5, Z_6\}$ . Se isso ocorrer, considere:

$$\begin{cases}
 B_\alpha = |b - a| + |b - d| \\
 C_\alpha = |c - a| + |c - d|
 \end{cases} \tag{3.5}$$

Então existem duas possibilidades:

- (i) Se  $B_\alpha < C_\alpha$  então existe uma borda passando por  $ad$  e o ponto  $b$  pertence a esta borda, como ilustrado na Figura 3.4 (a):



(a) Borda em  $ad$  passando por  $b$ . (b) Borda em  $ad$  passando por  $c$ .

Figura 3.4: Exemplificação da terceira condição. Os pixels em preto representam a região de uma borda. Fonte: Adaptado a partir de Kumar [20].

Então;

$$X = \frac{(a \times |\omega_t - a| + d \times |\omega_t - d| + b \times |\omega_t - b|)}{(|\omega_t - a| + |\omega_t - d| + |\omega_t - b|)} \quad (3.6)$$

- (ii) Se  $C_\alpha < B_\alpha$  então existe uma borda passando por  $ad$  e o ponto  $c$  pertence a esta borda, como ilustrado na Figura 3.4 (b). Neste caso;

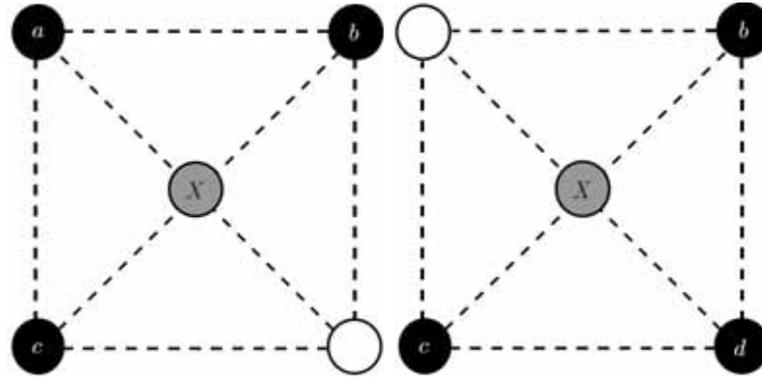
$$X = \frac{(a \times |\omega_t - a| + d \times |\omega_t - d| + c \times |\omega_t - c|)}{(|\omega_t - a| + |\omega_t - d| + |\omega_t - c|)} \quad (3.7)$$

- *Condição 4*: o algoritmo confere se há a presença de borda na direção  $bc$ . Para isto, basta averiguar se  $Z_2 = 0$  ou  $Z_2 < \min \{Z_1, Z_3, Z_4, Z_5, Z_6\}$ . Se isso acontecer, considere:

$$\begin{cases} A_\alpha = |a - b| + |a - c| \\ D_\alpha = |d - b| + |d - c| \end{cases} \quad (3.8)$$

Então;

- (i) Se  $A_\alpha < D_\alpha$  então existe uma borda passando por  $bc$  e o ponto  $a$  pertence a esta borda (Figura 3.5 (a)) :



(a) Borda em  $bc$  passando por  $a$ . (b) Borda em  $bc$  passando por  $d$ .

Figura 3.5: Exemplificação da quarta condição em que os pixels em preto representam a região de uma borda. Fonte: Adaptado a partir de Kumar [20].

Então tem-se:

$$X = \frac{(b \times |\omega_t - b| + c \times |\omega_t - c| + a \times |\omega_t - a|)}{(|\omega_t - a| + |\omega_t - c| + |\omega_t - b|)} \quad (3.9)$$

- (ii) Se  $D_\alpha < A_\alpha$  então existe uma borda passando por  $bc$  e o ponto  $d$  pertence a esta borda, como ilustrado na Figura 3.5 (b), e portanto,

$$X = \frac{(b \times |\omega_t - b| + c \times |\omega_t - c| + d \times |\omega_t - d|)}{(|\omega_t - b| + |\omega_t - c| + |\omega_t - d|)} \quad (3.10)$$

- *Condição 5:* Constata se todas as distâncias são maiores do que as horizontais. Em outras palavras, se  $Z_3 = 0$  e  $Z_4 = 0$  ou se  $Z_3$  e  $Z_4$  são menores que as distâncias diagonais e verticais. Se assim for, então:

$$X = \frac{(a \times |\omega_t - a| + c \times |\omega_t - c|)}{(|\omega_t - a| + |\omega_t - c|)} \quad (3.11)$$

- *Condição 6:* Se a borda está na direção  $ab$  a intensidade de  $Z_3$  é mínima possível. Logo  $Z_3 = 0$  ou  $Z_3 < \min \{Z_1, Z_2, Z_4, Z_5, Z_6\}$ . Então  $X$  receberá o valor da média ponderada entre os valores de  $ab$  e um valor intermediário entre  $c$  e  $d$ . Este valor intermediário será a média entre estes pontos e é denominada por  $H_2$ . Observe a Figura 3.6 (a), em que  $H_2 = \frac{c+d}{2}$ .

Neste caso

$$X = \frac{a \times |\omega_t - a| + b \times |\omega_t - b| + H_2 \times |\omega_t - H_2|}{|\omega_t - a| + |\omega_t - b| + |\omega_t - H_2|} \quad (3.12)$$

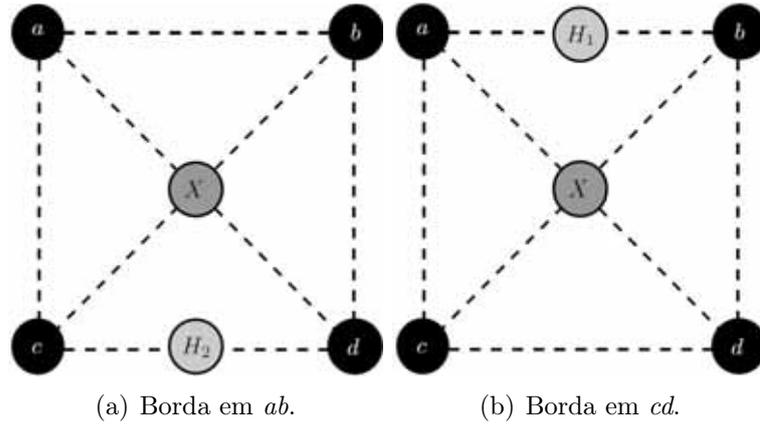


Figura 3.6: (a) Ilustração da condição 6; (b) Ilustração da condição 7.  
Fonte: Adaptado a partir de Kumar [20].

- **Condição 7:** Se a borda está na direção  $cd$ , o mesmo procedimento anterior é realizado. Desta maneira, se  $Z_4 = 0$  ou é a menor distância, então dado  $H_1 = \frac{a+b}{2}$  (Figura 3.6 (b)), o valor de  $X$  é dado pela Expressão 3.13:

$$X = \frac{c \times |\omega_t - c| + d \times |\omega_t - d| + H_1 \times |\omega_t - H_1|}{|\omega_t - c| + |\omega_t - d| + |\omega_t - H_1|} \quad (3.13)$$

- **Condição 8:** Assim como nas direções horizontais, o algoritmo verifica a existência de bordas nas verticais.
  - Se  $(Z_5 = 0$  e  $Z_6 = 0)$  ou  $(Z_5 < \min \{Z_1, Z_2, Z_3, Z_4\}$  e  $Z_6 < \min \{Z_1, Z_2, Z_3, Z_4\})$ .
  - Então

$$X = \frac{a \times |\omega_t - a| + b \times |\omega_t - b|}{|\omega_t - a| + |\omega_t - b|} \quad (3.14)$$

- **Condição 9:** Para verificar se há borda na direção  $ac$  basta conferir se  $Z_5$  possui a menor distância que os outros. Caso isso ocorra, então:

$$X = \frac{a \times |\omega_t - a| + c \times |\omega_t - c| + V_2 \times |\omega_t - V_1|}{|\omega_t - a| + |\omega_t - c| + |\omega_t - V_2|} \quad (3.15)$$

em que  $V_2 = \frac{b+d}{2}$  (Figura 3.7 (a)).

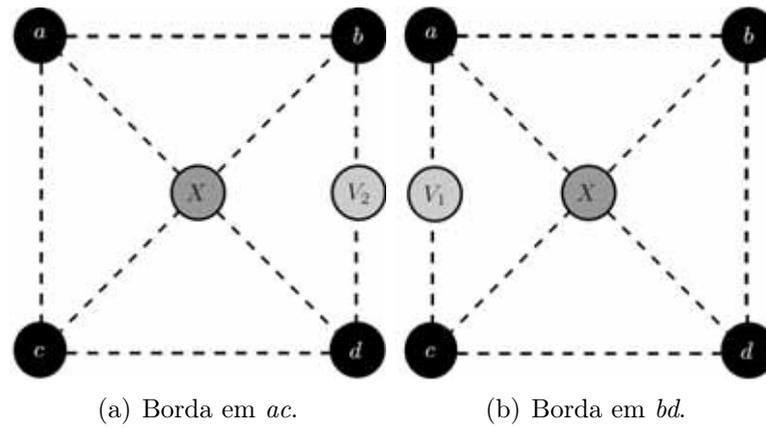


Figura 3.7: (a) Ilustração da condição 9; (b) Ilustração da condição 10.  
Fonte: Adaptado a partir de Kumar [20].

*Condição 10:* Por fim, o algoritmo verifica borda no lado  $bd$ . Se  $Z_6 = 0$  ou  $Z_6 < \min \{Z_1, Z_2, Z_3, Z_4, Z_5\}$ . Então considere  $V_1 = \frac{a+c}{2}$ , logo:

$$X = \frac{b \times |\omega_t - b| + d \times |\omega_t - d| + V_1 \times |\omega_t - V_1|}{|\omega_t - b| + |\omega_t - d| + |\omega_t - V_1|} \quad (3.16)$$

conforme a Figura 3.7 (b).

### – O Terceiro Estágio

A última etapa deste método consiste em retornar ao começo e verificar os pixels deixados indefinidos pelos estágios anteriores. O calculo destes pontos é como descrito no Estágio 2 considerando os pixels ao seu redor  $\{a, b, X_1, X_2\}$ , como mostrado na Figura 3.8. Observe que os pontos  $a$  e  $b$  são pontos da imagem original e  $X_1$  e  $X_2$  são pontos interpolados nos estágios anteriores.

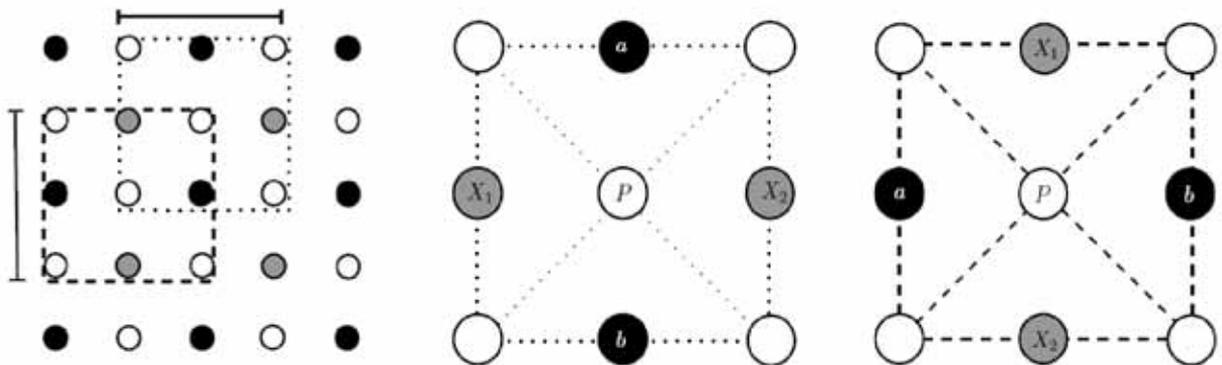


Figura 3.8: Ilustração do terceiro estágio do método.  
Fonte: Adaptado a partir de Kumar [20].

Observe o resultado obtido com a aplicação do método localmente adaptativo por Kumar [20] na Figura 3.9.



(a) Imagem original

(b) Rosto Ampliado

Figura 3.9: Aplicação do método adaptativo em imagem.

### 3.3 O método NEDI

Proposto por Li [22], o método *New Edge-Direction Interpolation* (NEDI), tem como metodologia interpolar cada ponto da imagem de alta resolução ( $Y_{2i,2j}$ ) utilizando uma estimativa da covariância da imagem de baixa resolução ( $X_{i,j}$ ). O *zoom* é realizado no fator de dois, ou seja,  $Y_{2i,2j} = X_{i,j}$ . Para isto, o algoritmo do método realiza o seguinte cálculo nos novos pixels:

$$\hat{Y}_{2i+1,2j+1} = \sum_{k=0}^1 \sum_{l=0}^1 \alpha_{2k+l} Y_{2(i+k),2(j+l)} \quad (3.17)$$

em que  $\hat{Y}_{2i+1,2j+1}$  são aqueles que possuem pixels originais nas diagonais. Na Figura 3.10, é representado um destes pixels pelo ponto cinza.

O coeficiente  $\alpha_{2k+l}$  é o peso estimado conforme a Equação 3.18.

$$\vec{\alpha} = R_{yy}^{-1} \vec{r}_y \quad (3.18)$$

Na Equação 3.18, tem-se o vetor  $\alpha = [\alpha_0, \alpha_1, \alpha_2, \alpha_3]$ ,  $R_{yy} = [R_{kl}]$  com  $l, k \in \{0, 3\}$  e  $r_y = [r_0, r_1, r_2, r_3]$ . Os termos  $[\alpha_0, \alpha_1, \alpha_2, \alpha_3]$ , são os coeficientes a serem calculados utilizando os quatro pixels originais que estão nas diagonais. O termo  $R_{kl}$  é a covariância-

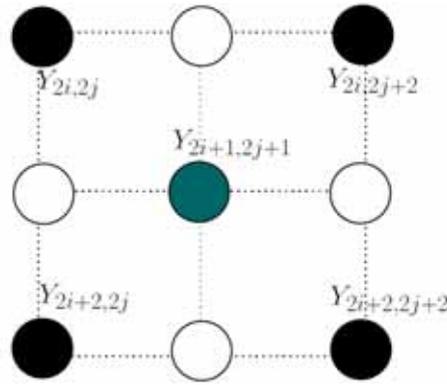


Figura 3.10: Representação da imagem expandida. Os pontos pretos são os pixels originais e o ponto cinza é o ponto interpolado  $\hat{Y}_{2i+1,2j+1}$ . Fonte: Adaptado a partir de Li [22].

local entre dois pontos originais vizinhos do ponto interpolado. Já o termo  $r_y$ , por sua vez, é o cálculo da covariância-local entre o ponto interpolado e os vizinhos mais próximos da imagem de baixa resolução. Por exemplo,  $R_{03} = E[Y_{2i,2j}Y_{2i,2j+2}]$  e  $r_0 = E[Y_{2i,2j}Y_{2i+1,2j+1}]$ , em que a função  $E[XY]$  é a esperança entre duas variáveis  $X$  e  $Y$ . Observe a Figura 3.11.

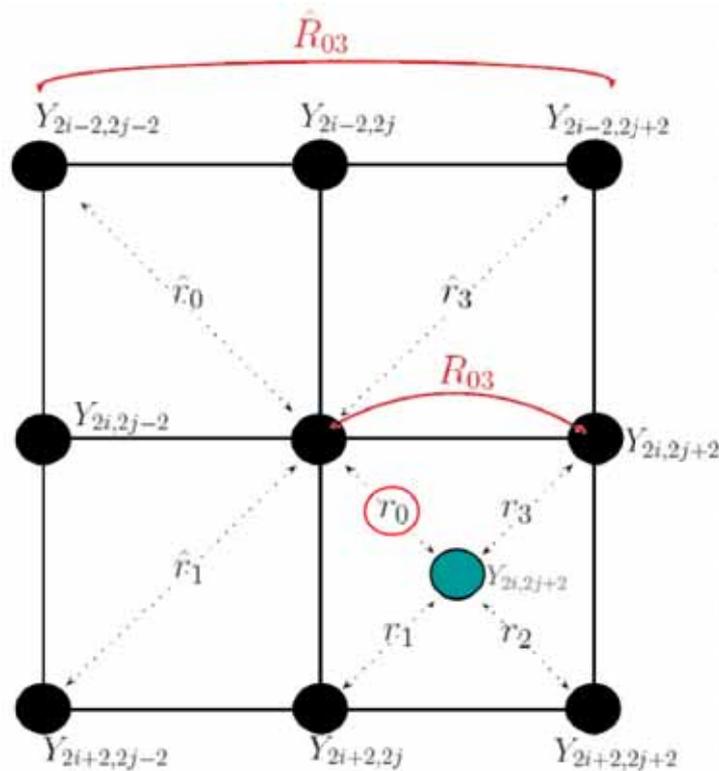


Figura 3.11: Representação do cálculo dos termos  $R_{kl}$ ,  $\hat{R}_{kl}$ ,  $r_k$  e  $\hat{r}_k$ . Os pontos pretos são os pixels originais enquanto o ponto cinza é o pixel que será interpolado pelo método. Fonte: Adaptado a partir de Li [22].

Observe que a covariância  $r_0$  circulado na Figura 3.11 precisa do valor  $Y_{2i+1,2j+1}$  para ser calculada. Contudo, não é conhecido o valor da intensidade deste pixel e portanto é necessário estimar o valor de  $r_k$ . Para isto, ao invés de utilizar  $r_k$ , é calculado o valor de  $\hat{r}_k$  com os valores dos pixels da imagem original. Desta maneira, o ponto  $Y_{2i+1,2j+1}$  é interpolado utilizando-se  $\hat{r}_k$  e  $\hat{R}_{kl}$  no calculo do coeficiente  $\alpha$ .

Segundo Li [22],  $\hat{R}_{k,l}$  pode ser calculado da seguinte maneira:

$$\hat{R}_{kl} = \frac{1}{M^2} C^T C \quad (3.19)$$

e, analogamente,  $\hat{r}_k$  é calculado da seguinte forma:

$$\hat{r}_k = \frac{1}{M^2} C^T \vec{y} \quad (3.20)$$

em que  $\vec{y} = [y_1, \dots, y_k, \dots, y_{M^2}]$  é o vetor que contém todos os pixels da janela local. A matriz  $C$  contém em cada linha o valor dos quatro vizinhos da diagonal de cada  $y_k \in \vec{y}$ . Ou seja, sendo  $\vec{y} = [y_{h_1, k_1}, y_{h_2, k_2}, \dots, y_{h_N, k_N}]^T$ , a matriz  $C$  é dada por:

$$C = \begin{pmatrix} I_{h_1-1, k_1-1} & I_{h_1-1, k_1+1} & I_{h_1+1, k_1-1} & I_{h_1+1, k_1+1} \\ I_{h_2-1, k_2-1} & I_{h_2-1, k_2+1} & I_{h_2+1, k_2-1} & I_{h_2+1, k_2+1} \\ \vdots & \vdots & \ddots & \vdots \\ I_{h_N-1, k_N-1} & I_{h_N-1, k_N+1} & I_{h_N+1, k_N-1} & I_{h_N+1, k_N+1} \end{pmatrix} \quad (3.21)$$

A Equação 3.18 pode então ser reescrita da seguinte maneira:

$$\vec{\alpha} = (C^T C)^{-1} (C^T \vec{y}) \quad (3.22)$$

Todos os coeficientes  $\alpha_i$ s podem ser resolvidos através do método de mínimos quadrados. Desta forma, o valor  $Y_{2i+1,2j+1}$  é interpolado com cada valor de  $\alpha$  da Equação 3.22 substituída na Equação 3.17. O método NEDI aplica a interpolação adaptativa baseada na covariância apenas nas áreas ao redor de bordas, e em regiões homogêneas, aplica a interpolação bilinear. Para isto, o algoritmo considera que o pixel está próximo de borda se o número que corresponde à diferença entre sua variância local estimada com os pixels da vizinhança for muito grande. Ou seja, se a variância local entre um pixel e os quatro pixels vizinhos forem menor que um limiar estipulado (*threshold*), então o algoritmo aplica a interpolação bilinear. Nos experimentos desenvolvidos ao longo deste trabalho foi utilizado *threshold*  $th = 8$ .

Após o cálculo de todos os pontos centrais  $Y_{2i+1,2j+1}$ , o algoritmo interpola os pontos do tipo  $Y_{2i+1,2j}$  que são representados pelos círculos com listras na vertical e os  $Y_{2i,2j+1}$ , representados pelos círculos com listras na horizontal (Figura 3.12).

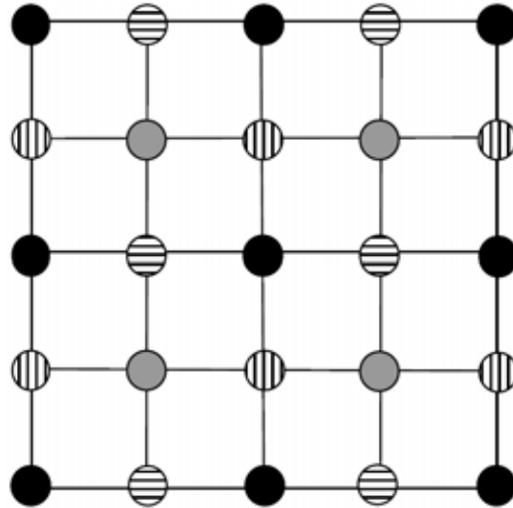


Figura 3.12: Representação dos pixels que ainda não foram interpolados (círculos listrados). Os círculos pretos representam os pixels originais enquanto os cinzas representam os pixels interpolados na etapa anterior.

Para preencher tanto os pontos  $Y_{2i,2j+1}$  quanto os  $Y_{2i+1,2j}$ , o algoritmo faz uma rotação de  $45^\circ$  e aplica a Equação 3.17 nestes pixels, conforme ilustra a Figura 3.13.

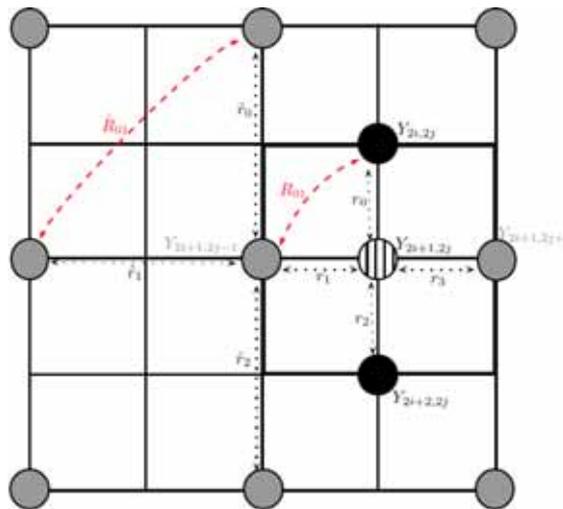


Figura 3.13: Segundo passo do algoritmo NEDI. Ilustração da interpolação do pixel  $Y_{2i+1,2j}$  através da estimativa de  $\alpha$  utilizando os termos  $\hat{R}_{kl}$  e  $\hat{r}_k$ . Fonte: Adaptado a partir de Li [22].

Na Figura 3.13 está representado a interpolação do ponto  $Y_{2i+1,2j}$ . Observe que nesta posição os pontos originais (pontos pretos) ficam na posição vertical e os estimados no primeiro passo (pontos cinzas) estão na posição horizontal. A estimativa dos termos  $R_{kl}$  e  $r_k$  são calculados da mesma maneira. Na Figura 3.14(b) pode ser observada a aplicação do método na região indicada na Figura 3.14(a).



(a) Imagem original.

(b) Rosto ampliado.

Figura 3.14: Parte da imagem ampliada pelo método NEDI.

### 3.4 Método iNEDI

O algoritmo NEDI tem como resultado a preservação da forma contínua das bordas, contudo, pode introduzir artefatos indesejáveis na imagem resultante. Giachetti [11] propõe uma modificação no método e apresenta o *improved New Edge Directed Interpolation* (iNEDI). Pode-se listar as seguintes limitações do método NEDI:

- A suposição da covariância estacionária local é violada em alguns casos;
- Introduz artefatos como *jaggies* e distorções em regiões de altas frequências;
- O sistema resultante dado pela Equação 3.21 é frequentemente mal condicionado. O uso de largas janelas melhora o condicionamento da matriz  $C^T C$ , porém, isto suaviza a imagem, dando a aparência de imagem borrada;
- Valores de pixels interpolados trocam com o brilho global;

Para resolver estes problemas, o algoritmo iNEDI apresenta as seguintes mudanças:

- A primeira modificação está no tamanho da janela utilizada por NEDI. De acordo com Giachetti [11], utilizar a janela quadrada pode introduzir artefatos direcionais e de qualquer modo faz com que o algoritmo seja não isotrópico. Em análise de imagens um operador isotrópico é aquele que responde de forma igual a variações de intensidade sem qualquer preferência por uma orientação. Os artefatos podem ser reduzidos ao calcular os parâmetros em janelas aproximadamente circulares, ou seja, janelas que variam seu formato de acordo com a posição do pixel interpolado.

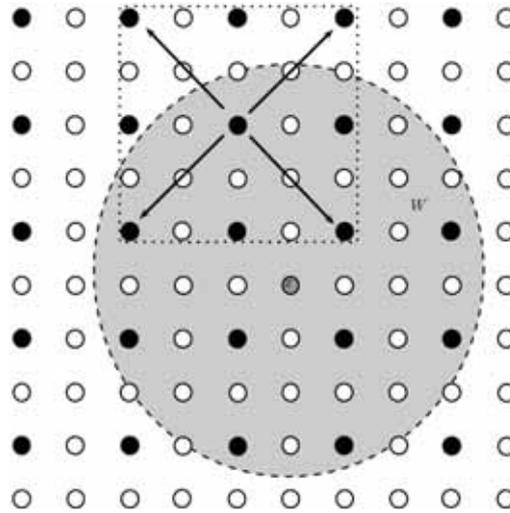


Figura 3.15: Representação da janela circular utilizada no método iNEDI.  
 Fonte: Imagem retirada e adaptada a partir de Giachetti [11].

A mudança para uma janela mais próxima possível de um círculo reduziria esses efeitos indesejáveis;

- O problema do mal condicionamento é que a matriz  $C$  pode ter uma linha ou coluna como combinação linear das outras. Neste caso o determinante da matriz  $C$  é igual a zero e sua solução dos mínimos quadrados não é única. Em outras palavras, existem vários  $\vec{\alpha}^*$ s que minimizam o termo  $\|C\vec{\alpha} - \vec{y}\|_2$ . Para resolver este problema, considere o valor de  $\vec{I}_4 = (I_{2i,2j}, I_{2i,2j+2}, I_{2i+2,2j}, I_{2i+2,2j+2})$  como sendo  $\vec{I}_4 = \vec{I}_0 + \vec{I}_{err}$ , com  $\vec{I}_{err}$  o erro. Desta maneira, o valor  $I_{2i+1,2j+1}$  interpolado é calculado pela Equação 3.23;

$$I_{2i+1,2j+1} = \vec{\alpha}^* \cdot \vec{I}_4 \quad (3.23)$$

e o seu erro quadrático é  $(\vec{\alpha}^* \cdot \vec{I}_{err})^2$ . Desta maneira, o método escolhe como única, a menor norma de solução para  $\vec{\alpha}^*$ . Esta modificação impede que a matriz  $C$  apresente o problema oriundo pela aplicação do método NEDI;

- Os valores dos pixels interpolados no método NEDI sofrem alterações no brilho. Isto ocorre porque eles não dependem somente das diferenças entre os valores dos vizinhos mas também do seus valores absolutos. Para resolver este problema, é subtraído de cada valor da matriz  $C$  a média da intensidade dos quatros vizinhos dos valores dos termos pertencente a  $C$ . Isto é:

$$C' = \begin{pmatrix} I_{h_1-1,k_1-1} - \bar{I}_{h_1,k_1} & I_{h_1-1,k_1+1} - \bar{I}_{h_1,k_1} & \cdots & \cdots \\ I_{h_2-1,k_2-1} - \bar{I}_{h_2,k_2} & I_{h_2-1,k_2+1} - \bar{I}_{h_2,k_2} & \cdots & \cdots \\ \vdots & \vdots & \ddots & \vdots \\ I_{h_N-1,k_N-1} - \bar{I}_{h_N,k_N} & I_{h_N-1,k_N+1} - \bar{I}_{h_N,k_N} & \cdots & \cdots \end{pmatrix} \quad (3.24)$$

e  $\vec{y}'$  dado por:

$$\vec{y}' = (I_{h_1,k_1} - \bar{I}_{h_1,k_1}, I_{h_2,k_2} - \bar{I}_{h_2,k_2}, \dots, I_{h_N,k_N} - \bar{I}_{h_N,k_N})^T \quad (3.25)$$

em que  $\bar{I}_{h,k} = \frac{(I_{h_1-1,k_1-1}, I_{h_1-1,k_1+1}, I_{h_1+1,k_1-1}, I_{h_1+1,k_1+1})}{4}$ . Desta forma,  $I(i, j)$  é obtido assim:

$$I(i, j) = \alpha' \cdot (I_{i-1,j-1}, I_{i-1,j+1}, I_{i+1,j-1}, I_{i+1,j+1}) + \bar{I}_{i,j} \quad (3.26)$$

- Ao invés de utilizar a interpolação bilinear, que suaviza demais as regiões homogêneas, o método iNEDI utiliza a interpolação bicúbica, o que lhe dá regiões suaves mais nítidas;
- Um outro problema do método NEDI é garantir que os pontos da janela utilizada na estimação do pixel interpolado estejam totalmente inseridos na mesma borda. O fato de pixels de regiões distintas estarem na mesma janela, pode ocasionar uma interpolação equivocada e introduzir artefatos indesejáveis nas bordas. Para solucionar este problema, o método tenta excluir todas as áreas uniformes que não estejam ligadas à borda para a interpolação (Figura 3.16).

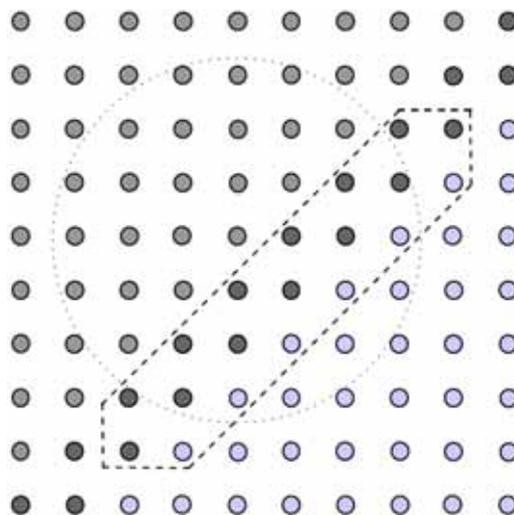


Figura 3.16: Ilustração da transição de regiões homogêneas separadas por uma borda. A máscara circular contém pixels da borda e de ambas as regiões. Imagem retirada e adaptada a partir de Giachetti [11].

Esta estratégia aumenta as chances de se ter uma boa interpolação, porém ainda há a possibilidade de ocorrer valores indesejáveis de altas frequências nesta etapa. Por essa razão é colocado mais um limiar, substituindo qualquer valor interpolado fora da faixa de intensidade dos quatro vizinhos com o mais próximo dos valores que delimitam esse intervalo.

Os resultados da aplicação do método na região indicada na Figura 3.17(a) é observado na Figura 3.17(b).



(a) Imagem original.

(b) Rosto ampliado.

Figura 3.17: Parte da imagem ampliada pelo método iNEDI.

### 3.5 Interpolação Iterativa com Base em Curvatura

Proposto por Giachetti [9, 11], o novo método tem como base um preenchimento em duas etapas e uma correção iterativa dos pixels interpolados obtidos através da minimização de uma função objetiva, dependendo das derivadas direcionais de segunda ordem da intensidade da imagem. A interpolação iterativa com base em curvatura (do inglês, *iterative curvature-based interpolation*, ICBI) utiliza restrições para derivar as funções relacionadas com o método NEDI, contudo é computacionalmente mais vantajosa por sua velocidade.

O método ICBI amplia a imagem duas vezes o tamanho original. Para o preenchimento dos novos pixels, o primeiro passo é calcular as aproximações da segunda ordem de derivadas  $\tilde{I}_{11}(2i+1, 2j+1)$  e  $\tilde{I}_{22}(2i+1, 2j+1)$ , dada respectivamente pela Equação 3.27a

e Equação 3.27b, em que  $I(i, j)$  representa o valor de intensidade da imagem na posição  $(i, j)$ .

$$\begin{aligned} \tilde{I}_{11}(2i + 1, 2j + 1) = & I(2i - 2, 2j + 2) + I(2i, 2j) + I(2i + 2, 2j - 2) - 3I(2i, 2j + 2) \\ & - 3I(2i + 2, 2j) + I(2i, 2j + 4) + I(2i + 2, 2j + 2) + I(2i + 4, 2j) \end{aligned} \quad (3.27a)$$

$$\begin{aligned} \tilde{I}_{22}(2i + 1, 2j + 1) = & I(2i, 2j - 2) + I(2i + 2, 2j) + I(2i + 4, 2j + 2) - 3I(2i + 2, 2j + 2) \\ & - 3I(2i, 2j) + I(2i - 2, 2j) + I(2i, 2j + 2) + I(2i + 2, 2j + 4) \end{aligned} \quad (3.27b)$$

Observe na Figura 3.18(a) e (b) os pixels utilizados para o cálculo das Equações 3.27a e 3.27b, respectivamente.

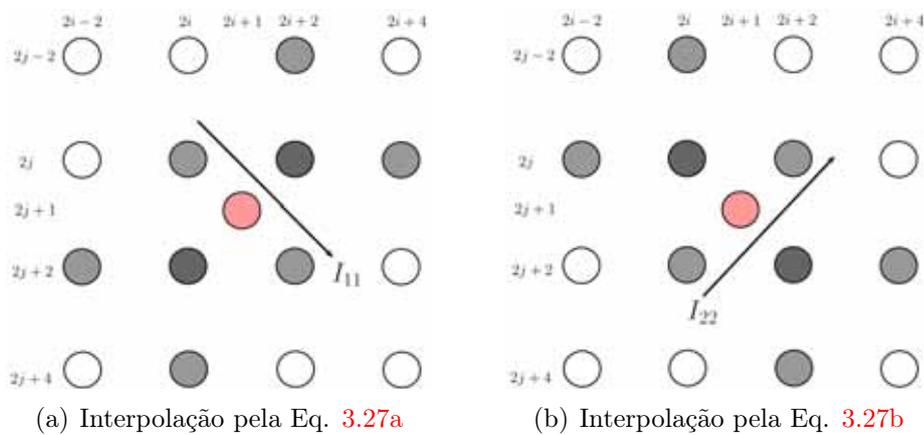


Figura 3.18: Ilustração dos pixels utilizados para o cálculo de  $\tilde{I}_{11}$  e  $\tilde{I}_{22}$ , respectivamente.  
Fonte: Imagem retirada e adaptada a partir de Giachetti [9].

Depois de calculados os termos  $\tilde{I}_{11}(2i + 1, 2j + 1)$  e  $\tilde{I}_{22}(2i + 1, 2j + 1)$ , é atribuído ao ponto  $I(2i + 1, 2j + 1)$  o valor da média dos dois vizinhos que possui a menor derivada direcional, conforme a Equação 3.28.

$$\left\{ \begin{array}{l} \frac{I(2i, 2j) + I(2i + 2, 2j + 2)}{2} \quad \text{se } \tilde{I}_{11}(2i + 1, 2j + 1) < \tilde{I}_{22}(2i + 1, 2j + 1) \\ \frac{I(2i + 2, 2j) + I(2i, 2j + 2)}{2} \quad \text{caso contrário} \end{array} \right. \quad (3.28)$$

Depois deste processo, o método ainda calcula o valor de três componentes para amenizar artefatos como *blurring* ou *jaggies* que a interpolação pela Equação 3.28 possa ocasionar. O primeiro componente é o termo que verifica a curvatura de continuidade ( $U_c$ ). O primeiro passo então é calcular  $U_c$ , dada pela Equação 3.29.

$$\begin{aligned}
U_c(2i+1, 2j+1) = & \\
& \omega_1(|(I_{11}(2i+1, 2j+1) - I_{11}(2i+2, 2j+2))| + |(I_{22}(2i+1, 2j+1) - I_{22}(2i+2, 2j+2))|) \\
& + \omega_2(|(I_{11}(2i+1, 2j+1) - I_{11}(2i+2, 2j))| + |(I_{22}(2i+1, 2j+1) - I_{22}(2i+2, 2j))|) \\
& + \omega_3(|(I_{11}(2i+1, 2j+1) - I_{11}(2i, 2j+2))| + |(I_{22}(2i+1, 2j+1) - I_{22}(2i, 2j+2))|) \\
& + \omega_4(|(I_{11}(2i+1, 2j+1) - I_{11}(2i, 2j))| + |(I_{22}(2i+1, 2j+1) - I_{22}(2i, 2j))|)
\end{aligned} \tag{3.29}$$

em que  $I_{11}$  e  $I_{22}$  são dadas pelas expressões 3.30 e 3.31, respectivamente:

$$I_{11}(2i+1, 2j+1) = I(2i-1, 2j-1) + I(2i+3, 2j+3) - 2I(2i+1, 2j+1) \tag{3.30}$$

$$I_{22}(2i+1, 2j+1) = I(2i-1, 2j+3) + I(2i+3, 2j-1) - 2I(2i+1, 2j+1) \tag{3.31}$$

e os pesos  $\omega_i$  pertencem ao conjunto  $\{0, 1\}$ , em que,  $\omega_i = 1$  se a primeira derivada é correspondente com a derivada direcional e zero caso contrário.

O termo  $U_c$  diminui a suavidade que possa ocorrer na interpolação dada pela Equação 3.28. O segundo termo é chamado de curvatura de realce, e é dada pela Equação 3.32.

$$U_e(2i+1, 2j+1) = -(|I_{11}(2i+1, 2j+1)| + |I_{22}(2i+1, 2j+1)|) \tag{3.32}$$

que resulta em uma imagem mais nítida.

O último termo (Equação 3.33) é derivado de Morse [24], que apresenta um método para melhorar a aparência da imagem em consideração a suavidade dos contornos.

$$U_i(2i+1, 2j+1) = f(I)|_{2i+1, 2j+1} I(2i+1, 2j+1) \tag{3.33}$$

em que  $f(I)$  é dada pela Equação 3.34:

$$f(I) = -\frac{I_1(i, j)^2 I_{22}(i, j) - 2I_1(i, j) I_2(i, j) I_{12}(i, j) + I_{11}(i, j)^2 I_2(i, j)}{I_1(i, j)^2 + I_2(i, j)^2} \tag{3.34}$$

em que  $I_{11}$  e  $I_{22}$  estão definidos pelas Equações 3.30 e 3.31, respectivamente, e os termos  $I_{12}$ ,  $I_1$  e  $I_2$  são dados por:

$$I_{12}(2i+1, 2j+1) = \tag{3.35}$$

$$\frac{1}{2}(I(2i+1, 2j-1) + I(2i+1, 2j+3) - I(2i-1, 2j+1) - I(2i+3, 2j+1))$$

$$I_1(2i+1, 2j+1) = \frac{1}{2}(I(2i, 2j) - I(2i+2, 2j+2)) \tag{3.36}$$

$$I_2(2i+1, 2j+1) = \frac{1}{2}(I(2i, 2j+2) - I(2i+2, 2j)) \tag{3.37}$$

Após ter calculado todos os termos, o algoritmo os reúne pela Equação 3.38:

$$U(2i + 1, 2j + 1) = \alpha \cdot U_c(2i + 1, 2j + 1) + \beta \cdot U_e(2i + 1, 2j + 1) + \gamma \cdot U_i(2i + 1, 2j + 1) \quad (3.38)$$

em que  $\alpha, \beta, \gamma$  são parâmetros escolhidos pelo usuário.

Para finalizar, é computado  $U(2i + 1, 2j + 1)$  e, a este é somado ou subtraído um pequeno valor  $\delta$  para cada pixel  $I_{2i+1,2j+1}$ . A menor intensidade destes é atribuído ao pixel interpolado. Em seguida, todo este processo é repetido para os pontos do tipo  $I_{2i+1,2j}$  e  $I_{2i,2j+1}$ . Para isto, são utilizado os pixels da imagem original que estão na vertical e horizontal, como demonstrado na Figura 3.19 (b).

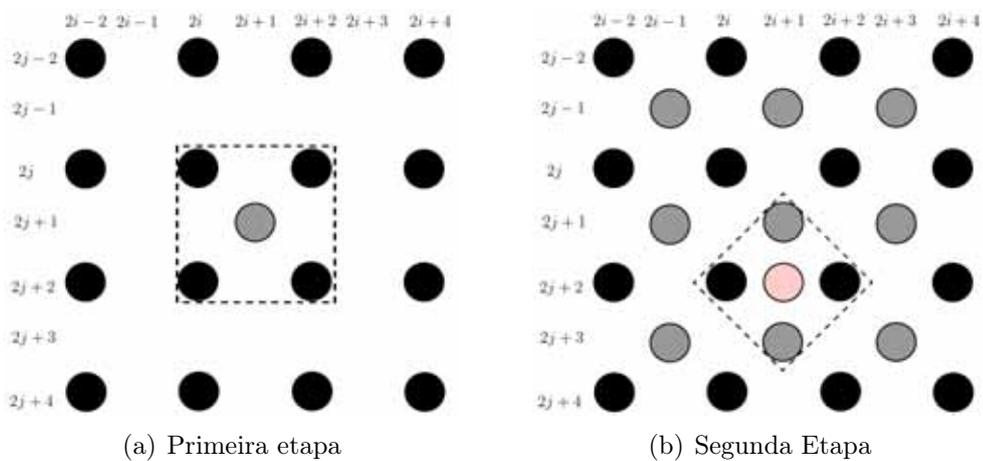


Figura 3.19: (a) Aplicação do método utilizando os quatro vizinhos da diagonal; (b) Aplicação do método utilizando os quatro vizinhos da vertical ou horizontal. Fonte: Imagem retirada e adaptada a partir de Giachetti [9].

A Figura 3.20(b) ilustra os resultados da aplicação deste método na imagem (região quadrangular) relativa a Figura 3.20(a).



Figura 3.20: Aplicação do método ICBI na região do rosto da Lena.

A implementação do método NEDI encontra-se em [22], do método iNEDI está em [11] e do icbi em [9].

## Outras abordagens: *Wavelets* e Filtro Bilateral

Neste capítulo, são incluídos duas outras abordagens, uma baseada em *wavelets* e outro no filtro bilateral. A primeira seção apresenta um resumo sobre a teoria das transformadas *wavelets* e sua aplicabilidade em super-resolução em processamento de imagens. Na segunda seção está um dos exemplos de método de interpolação proposto por Pagamisse [26], que utiliza o método adaptativo dado por Battiato [2] juntamente com *wavelets* não redundantes com o intuito de ampliar a imagem, levando-se em conta a super-resolução. Na terceira e última seção, é apresentado o algoritmo proposto por Han [14], que utiliza um novo método de interpolação baseado no filtro bilateral para a obtenção de imagens de altas resoluções.

### 4.1 Transformada *Wavelets*

Segundo Pedrini [28], uma transformada em processamento de imagens é o processo que resulta em uma alteração da representação inicial da imagem. Desta maneira é mais fácil investigar informações específicas e assim fazer uma análise mais acurada dos dados. Por exemplo, as transformadas de Fourier são alterações da função original em funções senoides complexas. Já a transformada discreta do cosseno são baseadas em cossenoides e são muito exploradas para aplicações em áreas de filtragem e compressão.

As transformadas *wavelets* por sua vez, são funções que permitem descrever as distintas frequências contidas na imagem. Diferente de Fourier, podem fornecer informações sobre a frequência e também sobre a imagem. A função *wavelet* usualmente denotada por  $\psi$  deve satisfazer a duas condições, descritas nas Equações 4.1 e 4.2.

1) Condição de admissibilidade:

$$c_\psi = 2\pi \int_{-\infty}^{\infty} \frac{|\hat{\psi}(\xi)|^2}{|\xi|} d\xi < \infty \quad (4.1)$$

que garante a invertibilidade da transformada *wavelet*. Em muitos casos, é o mesmo que exigir que a integral dessa função seja zero.

2) A função deve ser unitária, ou seja:

$$\int_{-\infty}^{\infty} |\psi(t)|^2 dt = 1 \quad (4.2)$$

No caso discreto, pode-se decompor um sinal unidimensional da seguinte maneira:

$$f(t) = \sum_k c_{1,k} \phi_{1,k}(t) + \sum_k d_{1,k} \psi_{1,k}(t) \quad (4.3)$$

A Equação 4.3 é a decomposição da função  $f(t)$  na primeira escala, em que  $\phi_{1,k}(t)$  é o filtro passa-baixa e  $\psi_{1,k}(t)$  é o filtro passa-alta. A segunda escala é dada pela Equação 4.4;

$$c_{1,k} = \sum_k c_{2,k} \phi_{2,k}(t) + \sum_k d_{2,k} \psi_{2,k}(t) \quad (4.4)$$

e assim sucessivamente.

### 4.1.1 Transformada *Wavelets* Decimadas

Duas operações importantes para o estudo da transformadas *wavelets* são a decimação (*downsampling*) e a interpolação (*upsampling*). No processo de decimação, a imagem é decomposta em quatro sub-camadas, em que, cada uma possui distintas informações da imagem original. Pelo fato da imagem ser um sinal bidimensional, é necessário uma função de escala bidimensional  $\varphi(x, y)$  e três *wavelets* bidimensionais  $\psi^H(x, y)$ ,  $\psi^V(x, y)$  e  $\psi^D(x, y)$ . Desta maneira, tem-se quatro produtos, em que a primeira é a função de escala separável:

$$\varphi(x, y) = \varphi(x)\varphi(y) \quad (4.5)$$

e as *wavelets* separáveis:

$$\psi^H(x, y) = \psi(x)\varphi(y) \quad (4.6)$$

$$\psi^V(x, y) = \varphi(x)\psi(y) \quad (4.7)$$

$$\psi^D(x, y) = \psi(x)\psi(y) \quad (4.8)$$

Essas *wavelets* medem as variações da função bidimensional. Em outras palavras, cada termo avalia as variações direcionais da imagem, ou seja,  $\psi^H(x, y)$  avalia variações nas

colunas, o que significa que possuem informações das bordas horizontais. Já  $\psi^V(x, y)$  avalia ao longo das linhas, o que fornece informações das bordas verticais e  $\psi^D(x, y)$  das diagonais.

Na Figura 4.1 está representado a decomposição das Equações 4.5 a 4.8. Ao aplicar a função *wavelet*, tem-se quatro sub-bandas, em que  $I_\phi$  é a banda suavizada da imagem, ou seja, possuem apenas as informações de baixa frequência que correspondem a regiões suaves e homogêneas. Já os detalhes de bordas, por serem altas frequências, estão contidas nas sub-bandas de detalhes em que  $I_\psi^H(x, y)$  estão as bordas horizontais,  $I_\psi^V(x, y)$  as verticais e  $I_\psi^D(x, y)$  detalhes das diagonais.

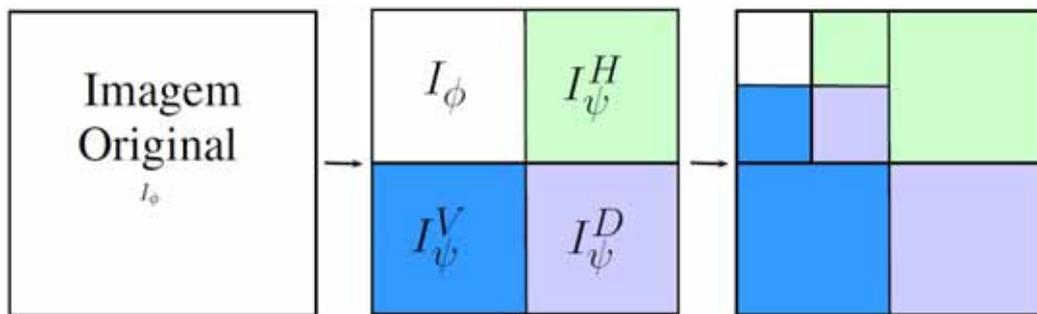


Figura 4.1: Esquema do processo de decimação em transformada *wavelets*.  
Fonte: Adaptado a partir de Castilho [5].

Observe que a cada nível houve um processo de decimação, ou seja, a informação do nível anterior foi dividida pelo fator de dois, e no segundo nível há mais divisão. O processo de decomposição citado acima está esquematizado na Figura 4.2 e Figura 4.3:

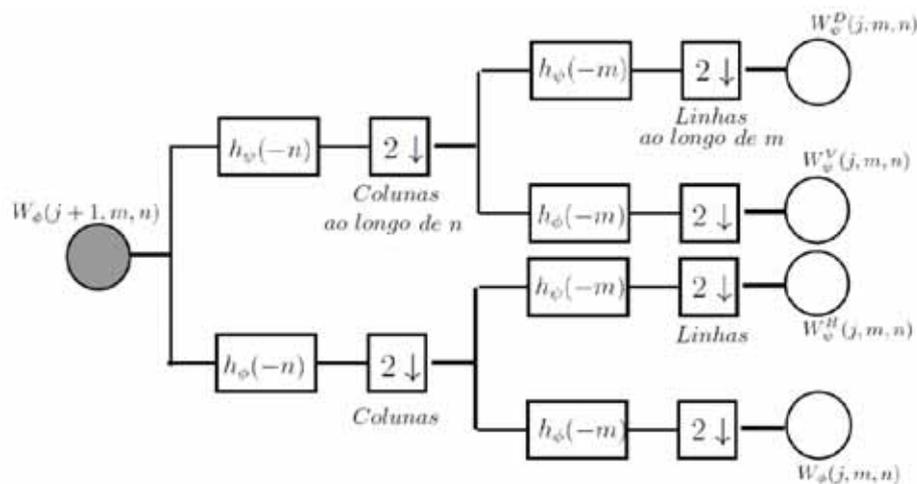


Figura 4.2: Esquema do processo de análise da imagem.  
Fonte: Adaptado a partir de Castilho [5].

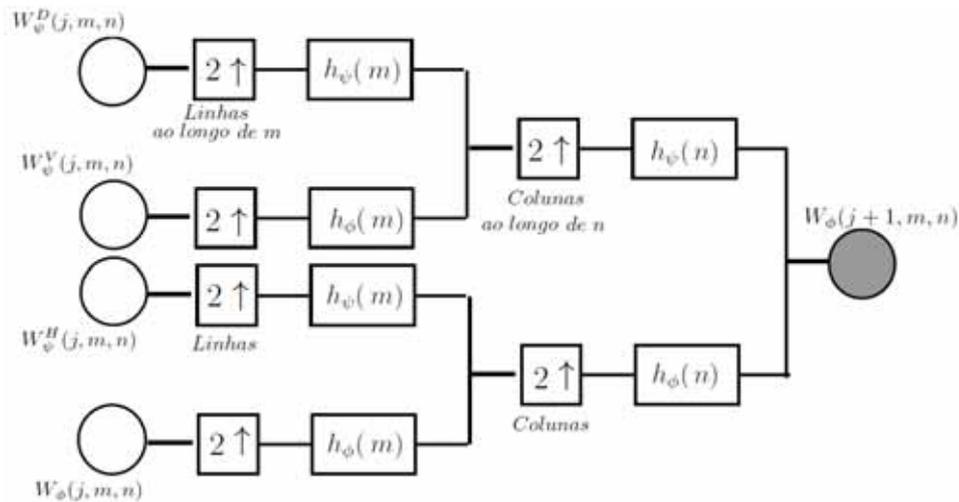


Figura 4.3: Processo de síntese para a composição da imagem.  
Fonte: Adaptado a partir de Castilho [5].

### 4.1.2 Wavelets Diádica

Assim como no processo de decimação, a transformada *wavelets* diádica também separa as informações em suave e em detalhes. Desenvolvida por Mallat [23], a imagem de entrada é separada em três sub-bandas; a camada suave, a de informações nas bordas horizontais e nas bordas verticais. A diferença desta transformada é que não há a decimação, ou seja, não há divisão pelo fator de dois. Isto significa que o tamanho de cada sub-banda em cada etapa é o mesmo do tamanho do sinal inicial. Por este motivo, esta transformada também é chamada de transformada *wavelets* redundantes.

Considere o filtro passa-baixa  $H(n)$  e o filtro passa-alta  $G(n)$  ambos correspondentes a uma função suavizante  $\varphi(t)$  e à *wavelet* mãe  $\psi(t)$ , respectivamente. Seja  $f$  o sinal (no caso a imagem) e os filtros  $H_p, G_p$ , obtidos inserindo  $2^p - 1$  zeros em cada coeficiente dos filtro  $H$  e  $G$ . A decomposição diádica discreta da imagem é dada por:

$$\begin{aligned} W_{2^{j+1}}^1 f &= S_{2^j} f * (G_j \cdot I) \\ W_{2^{j+1}}^2 f &= S_{2^j} f * (I \cdot G_j) \\ S_{2^{j+1}}^1 f &= S_{2^j} f * (H_j \cdot H_j) \end{aligned}$$

em que  $I$  é o filtro de Dirac e a operação  $*$  é a convolução. Note que nesta composição, os detalhes são separados e analisados em apenas duas camadas:  $W_k^1$ , em que possui todas as informações das bordas ao longo da direção horizontal e  $W_k^2$ , que possui todas as informações na vertical. Observe a Figura 4.4, em que cada nova camada possuirá a dimensão da anterior.

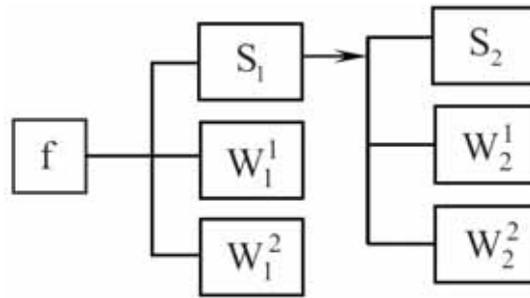


Figura 4.4: Esquema do processo de transformada *wavelet* diádica.  
Fonte: Adaptado a partir de Pagamisse [26].

## 4.2 Método Adaptativo de Super-resolução de Frame Único com *Wavelets* Redundantes

Apesar dos resultados obtidos pelas técnicas de interpolação utilizando vários frames, nem sempre é possível obter várias imagens de baixa resolução para se obter a de melhor qualidade. Por isso a super-resolução de frame único também é um tópico muito estudado. Em seu trabalho, Pagamisse [26] utiliza as *wavelets* redundantes como ferramentas para a ampliação da imagem de entrada, juntamente com cálculos da variância local para determinar os valores dos pixels interpolados (Ward [37]). Alguns pontos são obtidos baseados no cálculo de bordas locais em regiões 2x2 para obter informações adicionais da imagem. A seguir serão apresentados detalhes do método.

### 4.2.1 Descrição do Algoritmo

O método utiliza um frame único para sintetizar uma imagem em uma grade mais fina do que a original reduzindo a perda de informação de detalhes presentes. Para tal utiliza-se a expansão *wavelet* para realizar a interpolação juntamente com o cálculo de pixels, simulando um deslocamento sub-pixel na diagonal da imagem.

A transformada *wavelets* utilizada foi desenvolvida por Mallat [23] e é conhecida como transformada *wavelet* diádica discreta, em que, segue o processo de níveis sem a decimação. Ou seja, a cada nível a quantidade de informação da original é mantida em cada camada, obtendo assim uma redundância de informações.

Este algoritmo utiliza também o método adaptativo proposto por Battiato [2] para a interpolação dos pixels centrais da forma  $I_{2i+1,2j+1}$ . Pagamisse propõe pequenas alterações no método a fim de não precisar realizar o “*rebinning*”, que é o processo de escanear a imagem depois de uma etapa procurando os pontos ainda não interpolados para a próxima etapa.

Após este processo, é utilizada a ideia baseada em [37] para a interpolação dos pontos ainda sem preenchimento. Por fim é aplicado a transformada *wavelet* diádica discreta

inversa obtendo a imagem ampliada no fator de 2. A seguir cada passo será explicado com mais detalhes.

### 4.2.2 Super-Resolução em frame único

A primeira etapa para o aumento da imagem é a sua expansão, assim como nos métodos anteriores. Desta maneira, o primeiro passo do algoritmo é interpolar os pontos centrais, representados na Figura 4.5 por  $X$ , utilizando os quatros pontos vizinhos  $a, b, c$  e  $d$ .

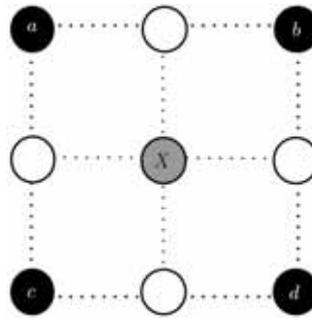


Figura 4.5: Representação da interpolação do ponto  $X$  utilizando os vizinhos das diagonais. Os pixels  $a, b, c$  e  $d$  pertence a imagem original.

Para esta interpolação Pagamisse [26] utilizou o método adaptativo segundo Battiato [2], contudo, fez pequenas alterações. Para o cálculo de  $X$  são verificadas as seguintes etapas:

- (*uniformidade*):  $|max(a, b, c, d) - min(a, b, c, d)| < T_1$ , em que  $T_1$  é um valor de tolerância (um valor limite). Neste caso  $X = \frac{a + b + c + d}{4}$ ;
- (*bordas na direção SW-NE*): se  $|a - d| > T_2$  e  $|a - d| - |b - c| > r$ , em que  $T_2$  e  $r$  também são valores de limites. Neste caso  $X = \frac{b + c}{2}$ ;
- (*bordas na direção NW-SE*): se  $|b - c| > T_2$  e  $|b - c| - |a - d| > r$ , então  $X = \frac{a + d}{2}$ ;
- (*bordas na direção NS*): se  $|a - c| > T_1$  e  $|b - d| > T_1$ , e ainda  $(a - d) * (b - c) > 0$  então se  $(a + c) > (b + d)$ , logo  $X = \frac{b + d}{2}$ , caso contrário,  $X = \frac{a + c}{2}$ ;
- (*bordas na direção EW*): se  $|a - d| > T_1$  e  $|b - c| > T_1$ ,  $(a - d) * (b - c) < 0$  e ainda se  $(a + b) > (c + d)$   $X = \frac{c + d}{2}$ , caso contrário,  $X = \frac{a + b}{2}$ .

Os parâmetros  $T_1$ ,  $T_2$  e  $r$  são fornecidos pelo usuário ao iniciar o algoritmo. Eles definem os limiares a serem tomados para verificações de regiões consideradas homogêneas ou bordas.

### 4.2.3 Aplicação da Transformada *Wavelets*

A transformada *wavelet* é obtida aplicando-se os filtros unidimensionais nas linhas e nas colunas da imagem. Entretanto, ao se aplicar a *wavelet*, os pixels em branco, indicado na Figura 4.6, não se relacionam com a informação do pixel central (pontos cinzas, obtida pela etapa anterior).

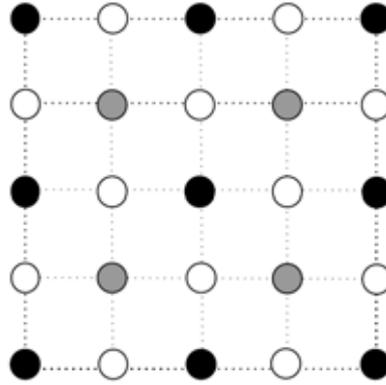


Figura 4.6: Imagem após a primeira etapa.

Para contornar este problema, é feita uma rotação na imagem de  $45^{\circ}$  graus e uma reamostragem pelo fator de dois para preencher os pontos brancos. Isto permite que estes se relacionem com os pontos pretos (pixels originais) e os pontos cinzas (Figura 4.7).

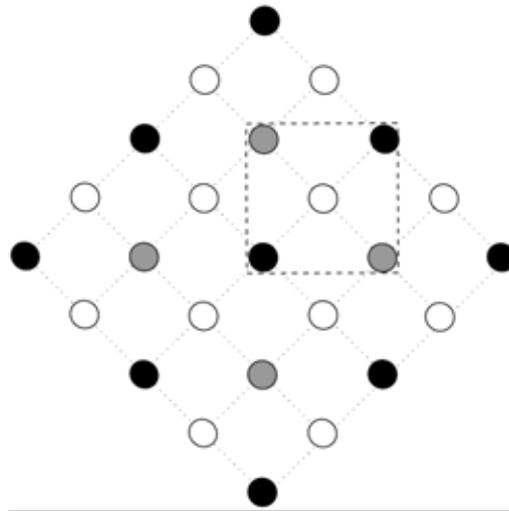


Figura 4.7: Imagem rotacionada  $45^{\circ}$ .  
Fonte: Adaptado a partir de Pagamisse [26].

Depois de rotacionados, é aplicado a transformada *wavelet* diádica na imagem obtendo três bandas: a versão suavizada, e duas de detalhes (horizontal e vertical).

Para a interpolação dos pontos em brancos mostrados na Figura 4.7 é calculado a média dos pontos que possuem a menor variância entre os quatro pontos vizinhos (Figura 4.8):

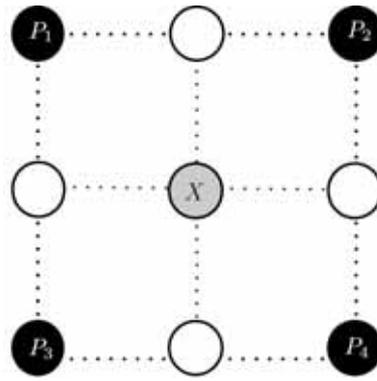


Figura 4.8: Pontos interpolados na segunda etapa.  
 Fonte: Adaptado a partir de Ward [37].

O cálculo proposto por Ward [37] para obter a menor variância consiste em construir conjuntos  $S_k$  definidos pela Expressão 4.9:

$$\begin{aligned}
 S_1 &= \{P_1, P_2, P_3\}, & S_2 &= \{P_1, P_2, P_4\} \\
 S_3 &= \{P_1, P_3, P_4\}, & S_4 &= \{P_2, P_3, P_4\}
 \end{aligned}
 \tag{4.9}$$

desta maneira, verifica-se qual dentre os conjuntos  $S_{k'}$ s com  $k \in \{1, 2, 3, 4\}$  possui os pontos  $P_k$  de tal maneira que  $k = \arg \min_j \{var(S_j)\}$  e  $j \in \{1, 2, 3, 4\}$ . Por exemplo, se  $S_1 = \min \{S_1, S_2, S_3, S_4\}$ , então os pontos  $\{P_1, P_2, P_3\}$ , que pertencem a  $S_1$ , são os que tem menor variância. Logo, o valor do ponto  $X$  será a média entre  $P_1$ ,  $P_2$  e  $P_3$ .

Para finalizar, a imagem é novamente rotacionada  $45^0$  de volta e assim, aplicado a transformada inversa obtendo a imagem com o tamanho duas vezes maior que a imagem original. A Figura 4.9(b) ilustra os resultados da aplicação deste método na imagem (região quadrangular) relativa a Figura 4.9(a).



Figura 4.9: Aplicação do método proposto por Pagamisse em parte da imagem.

### 4.3 Método de Interpolação utilizando o Filtro Bilateral

Uma nova abordagem que utiliza o filtro bilateral foi proposto por Han [14] e seu método interpola a imagem e ainda realça as bordas filtrando alguns ruídos nas regiões homogêneas. A ideia do algoritmo é utilizar o filtro bilateral para separar a imagem em duas camadas. Uma possuirá toda a informação dos detalhes da imagem original, enquanto a outra terá toda a informação de baixa frequência, ou seja, as regiões suaves. O método interpola ambas as camadas de maneira específica e no final as une com um realce nos detalhes.

O filtro Bilateral é um filtro passa-baixa que tem como objetivo a redução de ruídos. Possui um efeito de “borramento” quanto maior o desvio padrão utilizado. O valor da intensidade de cada pixel da imagem, ao ser aplicado o filtro bilateral, é substituída por uma média ponderada de valores de intensidade dos pixels vizinhos. Este peso pode ser baseada em uma distribuição Gaussiana. Segundo Tomasi [33], o filtro Bilateral pode ser definido como:

$$I_b = k^{-1}(x) \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} f(\xi) c(\xi, x) s(f(\xi), f(x)) d\xi \quad (4.10)$$

em que

$$k(x) = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} c(\xi, x) s(f(\xi), f(x)) d\xi \quad (4.11)$$

Neste caso,  $f(\xi)$  representa a função a qual é aplicado o filtro bilateral. As funções  $c(\cdot)$  e  $s(\cdot)$  no caso gaussiano, são definidos pela Expressão 4.12:

$$c(\xi, x) = e^{-\frac{1}{2} \left( \frac{d(\xi, x)}{\sigma_d} \right)^2} \quad (4.12)$$

$$s(\xi, x) = e^{-\frac{1}{2} \left( \frac{\delta(f(\xi), f(x))}{\sigma_r} \right)^2}$$

### 4.3.1 Descrição do método

Primeiramente é aplicado o filtro bilateral na imagem original, obtendo assim uma versão suavizada que recebe o nome de *camada base*. A diferença desta camada com a original resulta na *camada de detalhes*, que possui toda informação relevante das bordas. Contudo, ruídos também possuem altas frequências e estes aparecem na camada de detalhes.

Para reduzir o realce dos ruídos e assim, melhorar a imagem interpolada, é feita uma filtragem. Desta maneira, qualquer ponto isolado nos detalhes são eliminados.

Após a filtragem, ambas as camadas são expandidas individualmente. Para isto o método utilizado por Han [14] é a interpolação de preservação de bordas.

Depois de de todo este processo, junta-se as duas camadas com os detalhes realçados. Para melhor entendimento, será explicada cada passo a seguir.

### 4.3.2 Separação em Camadas

Para separar a imagem original primeiramente é aplicado o filtro bilateral na imagem. Para aplicação em imagens digitais é necessário reformular a Equação 4.10 para o discreto:

$$I_b = \frac{1}{k(x)} \sum_{y \in \Omega} c(x, y) \cdot s(I(x), I(y)) I(y) \quad (4.13)$$

em que  $I_b$  é a camada base,  $I(x)$  e  $I(y)$  são os valores dos pixels da imagem original nas posições  $x$  e  $y$ , respectivamente. Na Equação 4.13, os valores  $k(x)$ ,  $c(\cdot)$  e  $s(\cdot)$  são definidos por meio das Equações 4.14 e 4.15 respectivamente.

$$k(x) = \sum_{y \in \Omega} c(x, y) s(I(x), I(y)) \quad (4.14)$$

$$c(i, j) = e^{-\frac{\|i-j\|_2^2}{2\sigma^2}} \quad (4.15)$$

$$s(u, v) = e^{-\frac{\|u-v\|_2^2}{2\sigma^2}}$$

O parâmetro  $\sigma$  é o desvio padrão inserido inicialmente no algoritmo e  $\|\cdot\|_2^2$  é a norma euclidiana.

Aplicando este filtro em uma imagem, tem-se a camada base. Para se obter a camada de detalhe, basta fazer a diferença desta com a imagem original:

$$I_d(x) = I(x) - I_b(x) \quad (4.16)$$

Observe na Figura 4.10 a separação das camadas feita pelo algoritmo:

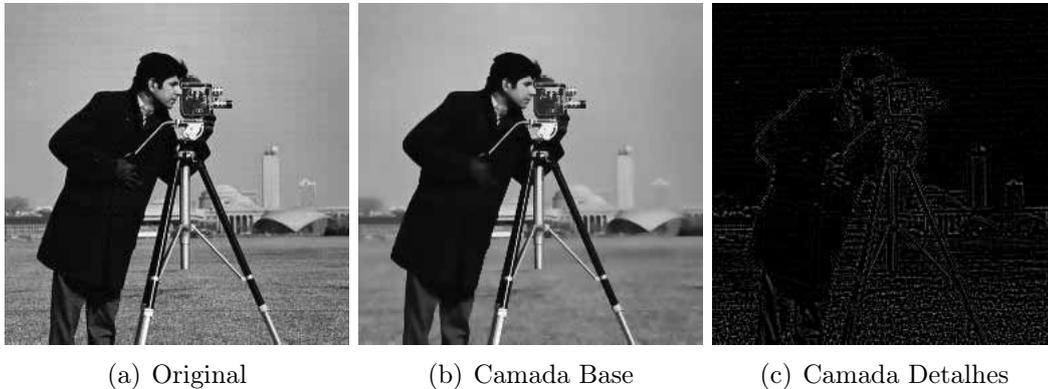


Figura 4.10: Aplicação do filtro bilateral na imagem original com  $\sigma_c = 1.8$  e  $\sigma_s = 30$ .

Quanto mais suavizada a camada base, mais grossa serão as bordas que ficam na camada de detalhes. Este controle é feito pelos desvios padrões  $\sigma_c$  e  $\sigma_s$  fornecidos inicialmente. O problema na camada base (Figura 4.10 (c)) é que todos os ruídos da imagem ficam realçadas. Portanto, é necessário filtrar esta camada de tal maneira que somente os detalhes relevantes das imagens permaneçam.

### 4.3.3 Filtragem da Camada de Detalhes

Para suprimir quaisquer ruído presente na imagem, primeiramente o algoritmo calcula os gradientes de energia de ambas as camadas:

$$E_b(x) = \sqrt{G_X^2(I_b(x)) + G_Y^2(I_b(x))} \quad (4.17)$$

$$E_d(x) = \sqrt{G_X^2(I_d(x)) + G_Y^2(I_d(x))} \quad (4.18)$$

em que  $G_X$  e  $G_Y$  são obtidos utilizando os operadores de Sobel:

$$G_X(F) = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} * F \quad (4.19)$$

$$G_Y(F) = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} * F \quad (4.20)$$

com o operador convolução (\*).

Depois de ter feitos estes cálculos é utilizado um filtro não linear com o gradiente de energia das duas camadas:

$$\hat{I}_d(x) = \frac{|E_b(x)|^2}{|E_b(x)|^2 + \gamma_1 |E_d(x)|^2} E_d(x) \quad (4.21)$$

em que, segundo Han [14],  $\gamma_1$  é um fator de regularização de controle entre a redução de ruídos e a preservação da textura da imagem definido pela Equação 4.22:

$$\gamma_1 = \frac{\sum |E_b(x)|^2}{\sum |E_d(x)|^2} \quad (4.22)$$

Observe a Figura 4.11 .

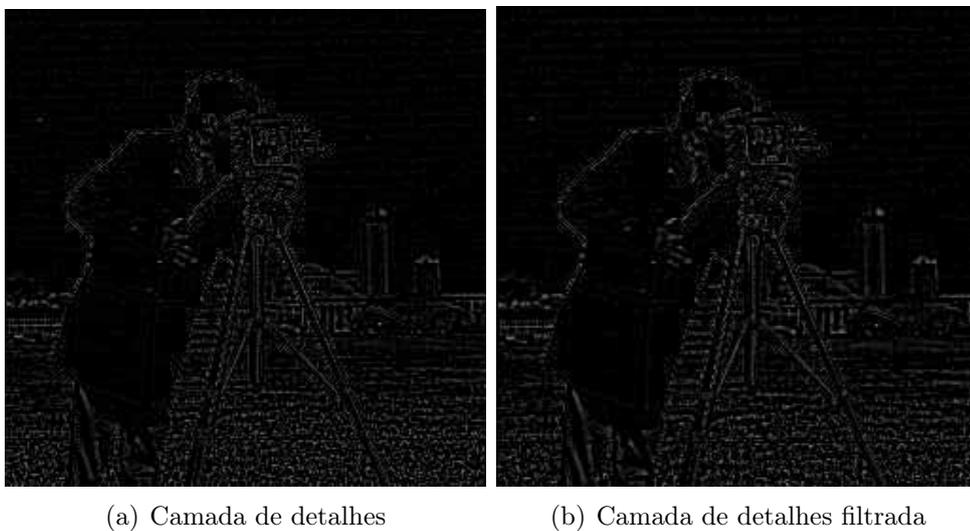


Figura 4.11: Ilustração da diferença entre a camada filtrada com a original.

#### 4.3.4 Interpolação das Camadas

É utilizado uma simples interpolação que leva em consideração a direção de borda através da correlação entre os pixels que consiste em três etapas:

- a) A primeira etapa baseia-se em expandir a imagem;
- b) Na segunda etapa, o algoritmo interpola todos os pontos  $I_c$  (Figura 4.12 (a)) aplicando o método de gradiente de base locais convencional (Hwang [17]). Para isto são utilizados os quatro pixels da diagonal que estão posicionadas ao seu redor.

Para esta interpolação é feito o seguinte cálculo:

$$d_a^* = \arg \min_{d \in \{1,2\}} (|A_d - B_d|) \quad (4.23)$$

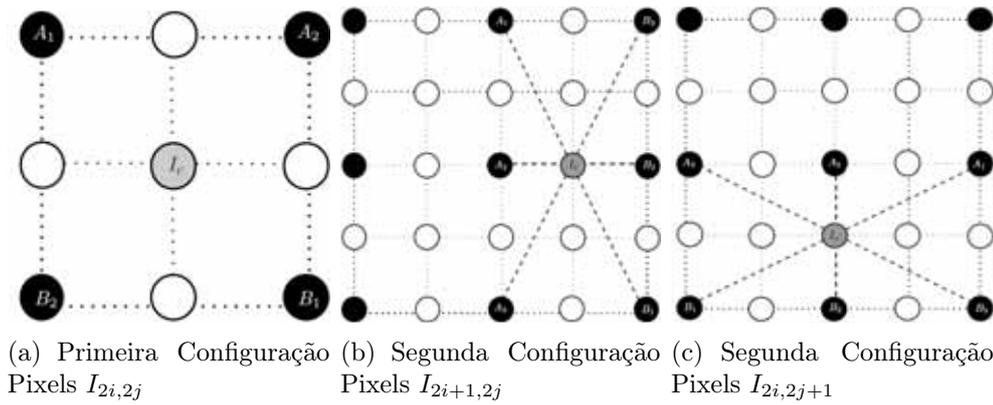


Figura 4.12: Representação das configurações do algoritmo para o cálculo das intensidades de pixels na presença de borda. Fonte: Imagem retirada e adaptada a partir de Han [14].

em que  $d_a^*$  representa a borda direcional desta primeira configuração. Desta maneira, é verificado qual diferença é menor, se  $|A_1 - B_1|$  ou  $|A_2 - B_2|$ . O valor de  $I_c$  é a média aritmética dos pontos pertencentes à menor intensidade, ou seja;

$$I(\cdot) = \frac{A_{d_a^*} + B_{d_a^*}}{2} \quad (4.24)$$

Os pontos centrais da Figura 4.12 possuirá a média dos pontos mais próximos de intensidade de tons de cinza entre si. Se houver uma borda passando por  $A_1$  e  $B_1$  por exemplo, a diferença  $|A_1 - B_1|$  será muito menor do que  $|A_2 - B_2|$ .

c) Nesta etapa o algoritmo irá preencher os pontos que sobraram, e para isso, há duas configurações possíveis:

- *Configuração 1*: Utilizado para os pixels que estão entre dois outros da imagem original na horizontal, como mostrado na Figura 4.12 (b). Estes pontos são do tipo  $I_{2n,2n+1}$ , ou seja, são pontos cuja a linha é par e a coluna é ímpar.
- *Configuração 2*: Utilizado para os pixels que estão entre dois outros da imagem original na vertical, como mostrado na Figura 4.12 (c). Em complementação da última configuração, estes últimos pontos interpolados são do tipo  $I_{2n+1,2n}$ , ou seja, são pontos cuja linha é ímpar e a coluna é par.

Assim como no item (b), para encontrar o valor de  $I_c$  é aplicada a Equação 4.24. Em que  $d_a^*$  é dada pela Equação 4.23 com  $d \in \{1, 2, 3\}$ .

Depois de aplicado a interpolação nas duas camadas, o algoritmo as compõe resultando na imagem final.

### 4.3.5 Composição das camadas com realce nos Detalhes

Para obter a imagem em alta resolução, é necessário fazer a seguinte composição:

$$\hat{I}_{final}(x) = \hat{I}_b(x) + \lambda(x)\hat{I}_d(x) \quad (4.25)$$

em que  $\hat{I}_b(x)$  e  $\hat{I}_d(x)$  são, respectivamente, a camada base e a camada de detalhes interpolados. A função  $\lambda(x)$  é um fator peso utilizado para realçar as informações na camada de detalhes (onde estão as bordas). Este fator é estimado pela Equação 4.26:

$$\lambda(x) = 1 + \frac{|\hat{E}_b(x)|}{\gamma_2|\hat{E}_d(x)|} \quad (4.26)$$

em que  $|\hat{E}_b(x)|$  é o gradiente da energia da imagem  $\hat{I}_b(x)$  e  $|\hat{E}_d(x)|$  é o gradiente da energia da camada  $\hat{I}_d(x)$ . Para prevenir uma alteração drástica e distorções de cores, Han [14] faz a seguinte restrição:

$$\lambda(x) \leq |\hat{E}_b(x)| \quad (4.27)$$

Ainda para a Equação 4.26,  $\gamma_2$  é definido como:

$$\gamma_2 = \frac{\sum |\hat{E}_b(x)|}{\sum |\hat{E}_d(x)|} \quad (4.28)$$

A Figura 4.13(b) ilustra os resultados da aplicação deste método na imagem (região quadrangular) relativa a Figura 4.13(a).



(a) Imagem original

(b) Rosto ampliado

Figura 4.13: Aplicação do método proposto na imagem Lena.

A implementação do método das wavelets foi fornecido por um dos autores. O programa da interpolação utilizando o filtro bilateral encontra-se em [16].

# Avaliação Qualitativas e Quantitativas das Imagem Digitais

Um dos problemas encontrados nos resultados obtidos usando técnicas de processamento de imagem está na análise da qualidade por meios de medidas estatísticas. Isto ocorre pois muitas vezes os métodos utilizados não levam em conta a percepção humana e desta maneira não reflete exatamente como a informação é recebida e interpretada.

A avaliação da qualidade de uma imagem digital é um assunto que ainda necessita de sofisticação em seus modelos matemáticos. Os métodos que avaliam a qualidade em sua maioria se baseam em medidas simples, como por exemplo Erro Médio Quadrático. A razão para isso é a falta do conhecimento da relação entre a estrutura do sistema visual humano (do inglês *Human Visual System*, HVS) com as estruturas estatísticas. O estudo do HVS é imprescindível quando se deseja avaliar as características de uma imagem, pois métodos puramente estatísticos não conseguem prever precisamente como o ser humano interpreta a informação visual. No entanto, tanto o conhecimento do HVS quanto a sofisticação de métricas matemáticas que modelam a percepção da visão ainda permanecem limitadas e são desafios a serem explorados.

Neste capítulo serão apresentados alguns métodos usuais da literatura indicando vantagens e desvantagens de serem utilizadas para a avaliação de qualidade de imagens.

## 5.1 Métricas Clássicas de Qualidade

Nesta seção serão apresentadas algumas técnicas clássicas para avaliação de qualidades de imagens, que são tradicionalmente utilizadas para a avaliação da eficiência numérica de alguns métodos utilizados para melhorar, filtrar, realçar ou redimensionar uma imagem digital. Contudo, cada métrica de qualidade de imagem é dependente da aplicação à qual

ela é submetida. Por exemplo, em imagens médicas são necessários métricas de qualidade mais precisas do que imagens utilizadas em entretenimento como jogos ou álbuns digitais de imagens pessoais.

### 5.1.1 Erro Máximo

A primeira métrica apresentada é a mais simples e consiste em fazer a diferença entre a imagem original  $f(x, y)$  e a imagem aproximada  $g(x, y)$ . Esta diferença é feita pixel a pixel, ou seja, as imagens a serem comparadas precisam ter a mesma dimensão. O erro máximo, cuja sigla é dada por ME (do inglês, *Maximum Error*) é expresso matematicamente pela Equação 5.1.

$$ME = \max |f(x, y) - g(x, y)| \quad (5.1)$$

Quanto menor for o valor de ME, menor a diferença e mais a imagem aproximada se assemelha à original. A análise do ME depende da aplicação, pois considere que a imagem apresente ruído e a aproximada tenha passado por um processo de filtragem. Neste caso se ME possuir um alto valor significa que a imagem filtrada é bem diferente da original. Contudo, esta diferença indica que o ruído foi reduzido ou que a imagem foi degradada mais ainda, aumentando assim a diferença entre ambas.

### 5.1.2 Erro Médio Absoluto

O erro médio absoluto (*Mean Absolute Error*, MAE) é a somatória dos módulos da diferença entre  $f(x, y)$  e  $g(x, y)$  dividido pelo tamanho da imagem. Considere que a imagem  $f(x, y)$  tenha dimensão  $M \times N$ , então a expressão é dado pela Equação 5.2.

$$MAE = \frac{1}{MN} \sum_{x=1}^{M-1} \sum_{y=0}^{N-1} |f(x, y) - g(x, y)| \quad (5.2)$$

Quanto menor o valor desta métrica, mais a nova imagem se aproxima da original.

### 5.1.3 Erro Médio Quadrático

Outra medida muito comum é o erro quadrático médio (do inglês, *Mean Square Error*, MSE) que é a soma das diferenças ao quadrado de cada ponto de duas imagens, dividido pelos tamanhos das imagens. A expressão do MSE é dada pela Equação 5.3.

$$MSE = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} [f(x, y) - g(x, y)]^2 \quad (5.3)$$

A MSE pode ser generalizada utilizando a norma  $\ell_p$ , ou também chamada de métrica de Minkowski ([3]). A medida de qualidade  $E_p$  definida em  $\ell_p$  pode ser expressa pela Equação 5.4.

$$E_p = \left( \sum_{i=0}^{M-1} |f_i - g_i|^p \right)^{\frac{1}{p}} \quad (5.4)$$

em que  $p \in [1, +\infty]$ .

A MSE é uma das métricas mais utilizadas para a comparação de imagens digitais e, quanto menor o valor desta métrica, menor a diferença entre as imagens.

#### 5.1.4 Raíz do Erro Médio Quadrático

Uma variação do MSE é a raíz do erro quadrático médio (*Root Mean Square Error*, RMSE) dada pela Equação 5.5.

$$RMSE = \sqrt{\frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} [f(x, y) - g(x, y)]^2} \quad (5.5)$$

#### 5.1.5 Erro Médio Quadrático Normalizado

Outra variação do MSE é o erro médio quadrático normalizado (*Normalized Mean Square Error*, NMSE), definido pela Equação 5.6.

$$NMSE = \frac{\sum_{x=0}^{M-1} \sum_{y=0}^{N-1} [f(x, y) - g(x, y)]^2}{\sum_{x=0}^{M-1} \sum_{y=0}^{N-1} [f(x, y)]^2} \quad (5.6)$$

Os valores do NMSE variam entre [0,1] de maneira que, quanto mais próximo de zero mais a nova imagem será semelhante da imagem original quantitativamente.

#### 5.1.6 Relação Sinal-Ruído de Pico

O *Peak Signal to Noise Ratio* (PSNR) é utilizado para avaliar a diferença global entre duas imagens. O PSNR pode ser definido a partir do MSE da seguinte maneira:

$$PSNR = 10 \cdot \log_{10} \left( \frac{L^2}{MSE} \right) \quad (5.7)$$

em que  $L$  é a maior variação entre a intensidade de nível de cinza máximo e mínimo.

### 5.1.7 Relação Sinal-Ruído

A métrica *Signal Noise Ratio* (SNR) possui uma estimativa aproximada que é calculada utilizando os valores no domínio da frequência. Esta medida relaciona a potência do sinal com a potência do ruído e pode ser determinada no domínio do espaço, e é definida pela Equação 5.8.

$$SNR = \frac{\sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y)^2}{\sum_{x=0}^{M-1} \sum_{y=0}^{N-1} [f(x, y) - g(x, y)]^2} \quad (5.8)$$

desta maneira, quanto mais próximos forem  $f(x, y)$  e  $g(x, y)$ , maior será o valor do SNR.

O SNR também pode ser expresso utilizando escala logarítmica e, desta maneira, o SNR leva em consideração a Lei de Weber para a percepção da visão humana. A lei de Weber-Fechner afirma que a resposta a qualquer estímulo é proporcional ao logaritmo da intensidade do estímulo. Esta lei aplica-se aos cinco sentidos, porém suas implicações são melhores entendidas quando se refere aos estímulos provocados pela luz e/ou pelo som. No caso do som, a unidade de medida utilizada é o Decibel (dB) definido como 10 vezes o logaritmo decimal da intensidade sonora. A mesma grandeza logarítmica descreve também a intensidade luminosa percebida, conforme a Equação 5.9.

$$SNR = 10 \cdot \log_{10} \left( \frac{\sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y)^2}{\sum_{x=0}^{M-1} \sum_{y=0}^{N-1} [f(x, y) - g(x, y)]^2} \right) \quad (5.9)$$

Quanto maior esta métrica, mais a imagem  $g(x, y)$  se aproxima da original.

### 5.1.8 Covariância

A covariância entre duas imagens  $f(x, y)$  e  $g(x, y)$  é dada por:

$$\sigma_{fg} = \frac{\sum_{x=0}^{M-1} \sum_{y=0}^{N-1} [f(x, y) - \mu_f][g(x, y) - \mu_g]}{MN} \quad (5.10)$$

em que  $\mu_f$  e  $\mu_g$  são os níveis de cinza médio das imagens  $f(x, y)$  e  $g(x, y)$ , respectivamente.

A covariância é por vezes chamada de medida de dependência linear entre  $f$  e  $g$ . Para medir o grau de dependência linear entre duas variáveis é utilizado o conceito de coeficiente de correlação.

### 5.1.9 Coeficiente de Correlação

O Coeficiente de Correlação é expresso pela Equação 5.11. A correlação é dada pela diferença entre um ponto da imagem e a média, que é representada por  $\mu_f$ ,  $\mu_g$  para  $f$  e  $g$ , respectivamente.

$$r = \frac{\sum_{x=0}^{M-1} \sum_{y=0}^{N-1} [f(x, y) - \mu_f] \cdot [g(x, y) - \mu_g]}{\sqrt{\sum_{x=0}^{M-1} \sum_{y=0}^{N-1} [f(x, y) - \mu_f]^2 \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} [g(x, y) - \mu_g]^2}} \quad (5.11)$$

em que,  $r$  varia entre  $[-1,1]$ . Quanto mais próximo o módulo de  $r$  estiver do valor 1, mais relacionadas estarão as duas imagens, e, em contrapartida, quanto mais próximo de zero menos as imagens estão relacionadas.

## 5.2 Problemas na Avaliação de Qualidade de Imagens

Para melhor entendimento do problema ocasionado com os métodos estatísticos mais comuns, considere a medida de qualidade MSE. Além de ser uma medida fácil de calcular e não muito cara computacionalmente, é muitas vezes conveniente para algoritmo de otimização, já que possui a característica de ser diferenciável e possui soluções de forma fechada que podem ser frequentemente aplicados em problemas reais quando combinadas com as ferramentas de Álgebra Linear. Por estes e outros motivos o MSE é extensivamente utilizado na literatura de processamento de imagem, comunicação e em vários campos de processamento de sinais. Contudo, o MSE tem sido criticado por sua fraca correlação com a qualidade de imagem percebida. De acordo com Wang [36], a métrica MSE não é capaz de identificar algumas degradações visuais relevantes, conforme ilustra a Figura 5.1.



Figura 5.1: Apresentação da imagem original Lena e dois tipos de degradações distintas.

Na Figura 5.1 (b), a imagem da Lena é reduzida na escala de quatro e depois redimensionada em seu tamanho original utilizando a interpolação do vizinho mais próximo. Observe que a imagem apresenta artefatos de *aliasing* o que é característico deste método de interpolação. Na Figura 5.1 (c) a imagem está degradada pelo ruído gaussiano. Apesar da evidente diferença entre as Figuras 5.1 (b) e 5.1 (c), ambas apresentam  $MSE \cong 262$ . Isto significa que o valor da métrica MSE não reflete quantitativamente o que está acontecendo na imagem. Alterações diferentes podem fornecer o mesmo número como resultado. O contrário também ocorre, pois pequenas mudanças de posicionamento ou inclinação resulta em valores bem distintos, mesmo que não haja nenhuma degradação na imagem.

Para avaliar a qualidade de uma imagem é necessário ter alguns cuidados com a métrica a ser utilizada. Para que a métrica demonstre uma eficácia na comparação entre duas imagens  $f(x, y)$  e  $g(x, y)$ , em que  $f(x, y)$  representa a imagem original (ou a imagem referencial da comparação) e a  $g(x, y)$  a imagem degradada (ou filtrada) que será comparada com a  $f(x, y)$ , é necessário uma avaliação quanto:

- (i) a iluminação da imagem;
- (ii) a visibilidade de detalhes da imagem;
- (iii) o efeito de focalização do sistema visual humano.

O primeiro é baseado na percepção luminosa do HVS. A percepção da lei de luminância obedece a lei de Weber, e pode ser expressa por:

$$K = \frac{\Delta I}{I} \quad (5.12)$$

em que  $I$  é a luminância do fundo e  $\Delta I$  é a luminância perceptível em contraste com a do fundo. A constante  $K$  é denominada de fração de Weber. De acordo com Gonzalez [13] a luminância é medida em *lumens* (lm) e é a quantidade de energia que um observador percebe de uma fonte de luz.

A iluminação está também ligada ao conceito de contraste simultâneo, que é a dependência do brilho aparente em alguma região com a intensidade do fundo de uma imagem (Figura 5.2). Observe que as Figuras 5.2 (a) e 5.2 (b) apresentam as metades separadas e que o brilho de cada semi-anel parecem levemente diferentes. Contudo é perceptível que, ao serem colocados sem nenhum obstáculo que os separem, ambas compõem um anel (anel *koffka*<sup>2</sup>) com apenas um tom de cinza (Figura 5.2 (c)). Isto demonstra o quanto a alteração do fundo pode influenciar na percepção do contraste no HVS.

---

<sup>2</sup>Do inglês *Koffka Ring*, nome dado em homenagem ao psicólogo Kurt Koffka da *Gestalt*, que estudou a importância do agrupamento em fenômenos perceptivos. Mostrou que um anel de cinza sobre um fundo claro/escuro (Figura 5.2 (c)) parece uniforme, mas quando as duas metades são divididas, como nas Figuras 5.2 (a) e (b), eles se parecem diferentes.

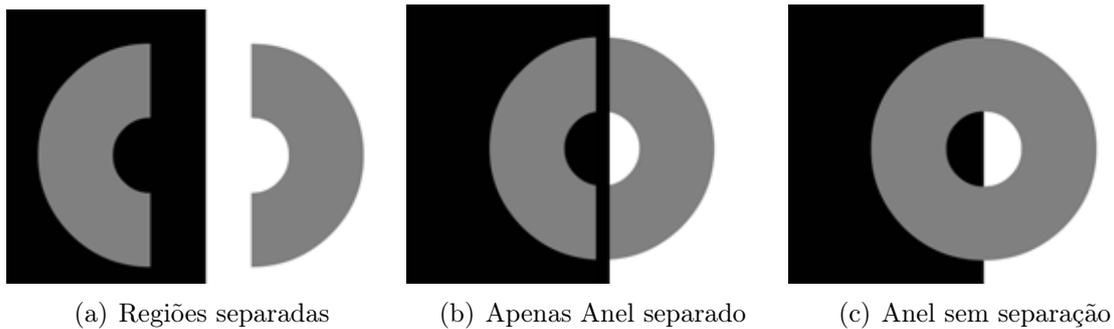


Figura 5.2: (a) Metades do anel de mesma intensidade separado; (b) Anel separado apenas por uma linha preta; (c) Anel sem separação. Fonte: Imagem retirada e adaptado a partir de [25].

O item (ii) se refere aos detalhes visuais que são perceptíveis no HVS e também tem algumas ligações à abordagem do contraste. Em geral, a redução de visibilidade de uma região da imagem é mais forte quando o sinal e o detalhe tem localização espacial, o conteúdo de frequência, e orientações semelhantes.

A Figura 5.3 demonstra uma imagem que possui os detalhes “mascarados” pelo ruído gaussiano. Pela presença do ruído, o sinal parece ser muito mais fraco em regiões com texturas do que em regiões mais lisas. Por exemplo, na região de transição da testa com o chapéu é muito mais evidente e fácil de identificar na Figura 5.3 (a) do que na Figura 5.3 (b).

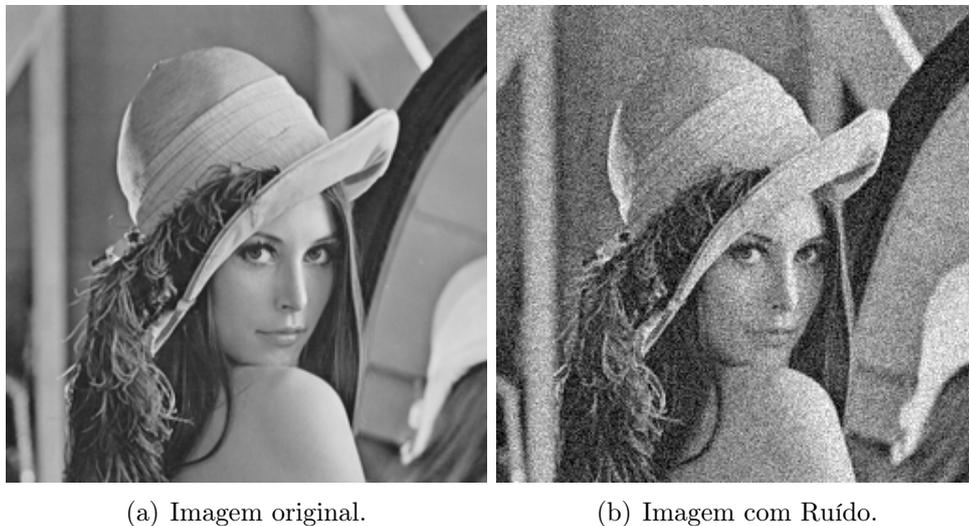


Figura 5.3: Exemplo da “mascara” ocasionada pelo ruído branco.

O item (iii) se refere ao processo do olho humano em focalizar algum objeto. Devido às distribuições não uniformes de receptores de cone e células ganglionares na região central da retina (fóvea), quando os olhos se fixam a um ponto qualquer a região em torno deste possui menor resolução. A resolução vai decrescendo rapidamente conforme o aumento da distância do ponto focado. Na Figura 5.4 é possível observar como funciona o foco, em que a região centrada possui resolução melhor do que a região ao redor. Conforme a

distância aumenta, menor a resolução do objeto. A região do foco que possui a máxima resolução espacial é chamada de “visão de foco” enquanto que a visão progressivamente de menor resolução é denominada de “visão periférica”.

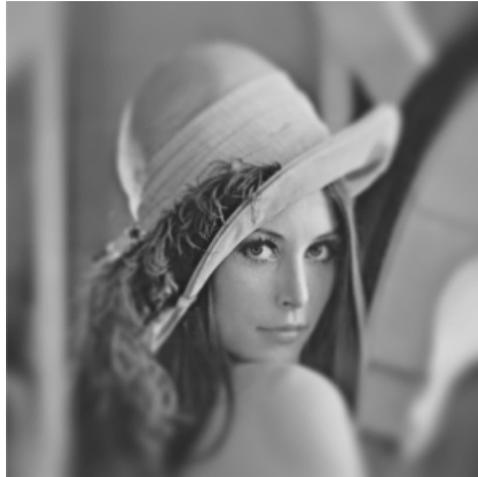


Figura 5.4: Representação do foco do HVS. A resolução da imagem fica cada vez menor quanto mais se distancia da região focada.

Tendo em vista estes três parâmetros para uma boa avaliação de qualidade de imagens, muitos algoritmos ainda vêm sendo estudados e melhorados para aproximar os resultados de correlação numérica entre duas imagens com a relação com o sistema visual humano. A seguir será apresentado uma das medidas de qualidade que leva em consideração os assuntos abordados.

## 5.2.1 Índice de Similaridade Estrutural

O índice de similaridade estrutural, ou SSIM (do inglês, *structural similarity index*) é uma função entre duas imagens  $f(x, y)$  e  $g(x, y)$ , em que uma das imagens é assumida ter uma qualidade perfeita (isto é, a imagem de referência) e a outra será a imagem degradada (ou a que será comparada). O algoritmo SSIM separa a tarefa de medida de similaridade da imagem em três comparações: luminância, contraste e estrutura.

### 5.2.1.1 Luminância

A comparação da luminância entre duas imagens  $f(x, y)$  e  $g(x, y)$  pode ser definida a partir da seguinte Equação:

$$\ell(f, g) = \frac{2\mu_f\mu_g + C_1}{\mu_f^2 + \mu_g^2 + C_1} \quad (5.13)$$

em que

$$\mu_x = \bar{x} = \frac{1}{N} \sum_{i=1}^N x_i \quad (5.14)$$

Na Equação 5.14,  $\mu_x$  representa a intensidade média estimada em uma janela com  $N$  amostras. A constante positiva  $C_1$  é adicionada para evitar que a fração tenha denominador igual a zero.

### 5.2.1.2 Contraste

É utilizado o desvio padrão como uma estimativa do contraste. Pode ser expresso pela Equação 5.15.

$$\sigma_x = \left( \frac{1}{N-1} \sum_{i=1}^N (x_i - \mu_x)^2 \right)^{\frac{1}{2}} \quad (5.15)$$

A comparação do contraste entre duas imagens é então dada pela Expressão 5.16.

$$c(f, g) = \frac{2\sigma_f\sigma_g + C_2}{\sigma_f^2 + \sigma_g^2 + C_2} \quad (5.16)$$

e, por fim, o próximo parâmetro é a comparação entre as estruturas das imagens.

### 5.2.1.3 Estrutura

A estrutura da comparação  $s(f, g)$  é definida pela Expressão 5.17:

$$s(f, g) = \frac{2\sigma_{fg} + C_3}{\sigma_f\sigma_g + C_3} \quad (5.17)$$

em que  $\sigma_{fg}$  pode ser estimado por:

$$\sigma_{xy} = \frac{1}{N-1} \sum_{i=1}^N (x_i - \mu_x)(y_i - \mu_y) \quad (5.18)$$

### 5.2.1.4 Similaridade Global

Os três componentes são relativamente independentes, pois a mudança de luminosidade e/ou contraste tem pouco impacto sobre as estruturas dos objetos na cena visual. Contudo, a combinação destes compõe uma métrica que satisfaz as seguintes condições:

- **Simetria:** ou seja  $S(f, g) = S(g, f)$ . Isto se deve pois a função  $S$  não pode sofrer alterações quando é mudada a ordem dos sinais de entrada;
- **Limite:**  $S(f, g) = 1$ . É importante um limite superior para indicar quando duas imagens são idênticas. Se houver um limite inferior, este representara a similaridade entre duas imagens completamente distintas;
- **Máximo único:** ou seja,  $S(f, g) = 1$  se e somente se  $f = g$ . A pontuação máxima é alcançada somente quando os sinais que estão sendo comparados são idênticos.

Desta maneira a medida de similaridade quantificará quaisquer variações que possam existir entre os sinais de entrada.

Esta métrica é expressa pela Equação 5.19.

$$S(f, g) = [\ell(f, g)]^\alpha \cdot [c(f, g)]^\beta \cdot [s(f, g)]^\gamma \quad (5.19)$$

em que  $\alpha > 0$ ,  $\beta > 0$  e  $\gamma > 0$  são parâmetros que ajustam a importância de cada componente. A Expressão 5.19 pode ser simplificada definindo  $C_3 = \frac{C_2}{2}$  e  $\alpha = \beta = \gamma = 1$ , desta maneira tem-se:

$$S(x, y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)} \quad (5.20)$$

Esta métrica demonstra ser superior às tradicionais como o MSE, pelo fato de levar em conta diferenciação na parte estrutural, luminância e no contraste. Wang [36] afirma que é mais aconselhável aplicá-la em blocos da imagem. Isso porque os cálculos de  $\mu_x$ ,  $\sigma_y$  e  $\sigma_{xy}$  por exemplo, são melhores computados em uma pequena região do que na imagem como um todo. Em primeiro lugar, as características estatísticas da imagem, geralmente, não exibem informações estacionárias espaciais significativas. Em segundo lugar, as distorções, que podem existir ou não, dependem das estatísticas locais da imagem. Em terceiro lugar, apenas uma área específica pode ser vista com uma resolução elevada por um observador humano a um dado instante de tempo (devido à característica do foco do HVS).

A medição localizada pode fornecer um mapa de qualidade variando espacialmente na imagem, o que proporciona mais informações sobre a degradação em regiões isoladas, e pode ser útil em algumas aplicações. Contudo a Equação 5.19 possui a desvantagem de não considerar pequenas rotações ou distorções na comparação entre duas imagens. Em outras palavras, se forem comparadas duas imagens idênticas, com uma delas um pouco rotacionada ou um pouco distorcida, o SSIM acusará uma discrepância entre elas. Portanto, esta métrica não é invariante à rotação.

Um caso especial da Equação 5.20, em que  $C_1 = C_2 = 0$  é chamado de Índice de Qualidade Universal (do inglês, *Universal Image Quality Index*, IQI), definido na Equação 5.21.

$$IQI = \frac{4\mu_x\mu_y\sigma_{xy}}{(\mu_x^2 + \mu_y^2)(\sigma_x^2 + \sigma_y^2)} \quad (5.21)$$

O IQI é mais sensível à distorção da estrutura da imagem se comparado ao MSE que é apenas sensível à energia do erro (Wang [36]). Na literatura tem-se utilizado mais a métrica IQI por sua eficácia e por não exigir nenhum parâmetro de entrada do usuário.

# Análise e Comparação dos Métodos Estudados

Este capítulo está dividido em três seções. Na primeira seção é feita uma análise das características de cada método aplicado em algumas imagens com texturas distintas. Na seção 6.2 é aplicado os métodos em imagens coloridas enquanto na seção 6.3 é realizada a comparação de todos os métodos vistos usando as métricas descritas no Capítulo 5.

## 6.1 Análise Visual dos métodos

Nesta seção será explorado como cada método se comporta em regiões próximas de bordas, que são as altas frequências da imagem. As imagens utilizadas são dadas pela Figura 6.1.



(a) Imagem da Lena.

(b) Imagem do Copo.

(c) Imagem teste.

Figura 6.1: (a) Recorte da imagem da Lena; (b) Recorte da imagem do Copo; (c) Recorte da imagem teste.

A Figura 6.1(a) possui tamanho 128x128, enquanto que a Figura 6.1(b) possui 150x150 e a Figura 6.1(c) possui 120x120. Em cada imagem foram aplicados os métodos de interpolação estudados nos Capítulos 2, 3 e 4 e calculado o gasto computacional de cada técnica. A máquina utilizada para estas estimativas de tempo, possui processador AMD Athlon(tm) X2 Dual-Core QL-65 com 2.10 GHz e 3 GB de memória Ram.

### 6.1.1 Interpolação por Replicação

O método da replicação ou (vizinho mais próximo) possui a vantagem de ser o mais simples computacionalmente e desta maneira o mais rápido também. Contudo, pela sua simplicidade ele apresenta alguns problemas. A Figura 6.2 mostra o aparecimento do efeito de serrilhamento nas bordas da imagem reamostrada. Isso se deve pela característica do próprio método já que o novo pixel possuirá o valor do nível de cinza do seu vizinho mais próximo.



Figura 6.2: (a) Imagem da Lena; (b) Parte da imagem ampliada utilizando método da replicação.

É possível ver na Figura 6.2 os *jaggies* consequente do método. Na Figura 6.3 é representada uma borda que, ao ampliá-la, os pontos a serem preenchidos recebem os valores mais próximo do original, isto causa o efeito serrilhado pois não há uma transição suave dos pixels que estão em uma região homogênea com a outra. A Figura 6.3 (b) foi ampliada pelo fator de quatro e quanto mais ampliada, maior será o efeito serrilhado.

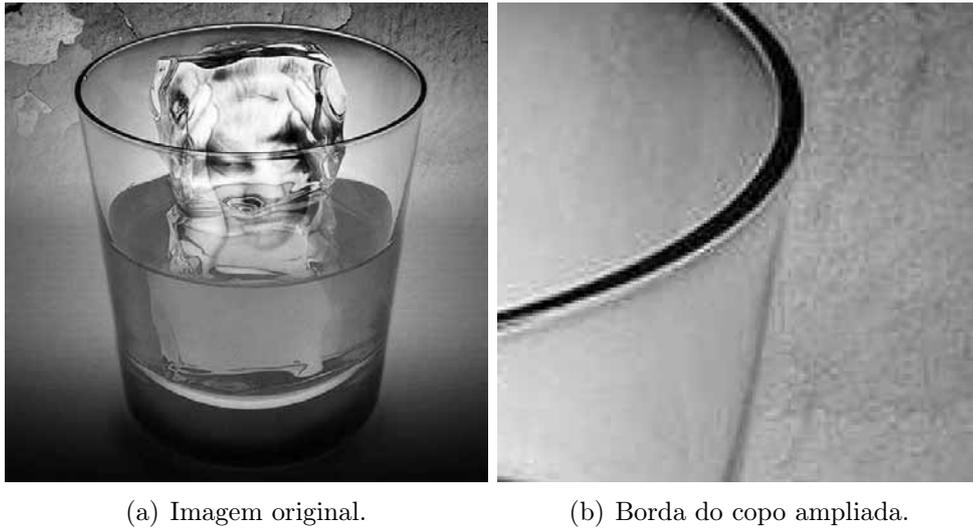


Figura 6.3: Aplicação da replicação na imagem da borda do copo.

Observe a Figura 6.4, em que aparentemente a Figura 6.4 (b) é a interpolação exata da Figura 6.4 (a). Contudo, a imagem utilizada é de baixa resolução e já apresenta bordas serrilhadas. Ao aplicar a replicação as mudanças ocasionadas na imagem não são muito aparentes.

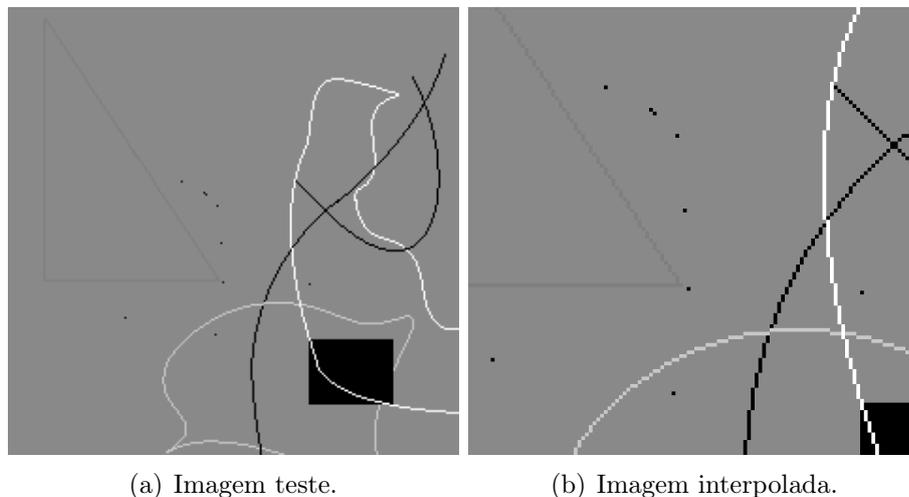


Figura 6.4: Aplicação da interpolação replicação na imagem teste. As falhas não são visíveis por se tratar de uma imagem de baixa resolução.

Sobre o tempo que o algoritmo utilizou para a interpolação, a Figura 6.2(b) demorou em média 0,003 segundos, enquanto que a Figura 6.3(b) demorou 0,0044 e a Figura 6.4(b) demorou 0,0024 segundos.

## 6.1.2 Interpolação Bilinear

A interpolação bilinear possui resultados mais satisfatórios do que o método do vizinho mais próximo. Isso se deve pela média ponderada das distâncias dos pixels ao seu redor. O método da interpolação bilinear apresenta o efeito serrilhado mais atenuado, contudo os detalhes são afetados pelas médias e resulta em um aspecto de “imagem borrada”.

A seguir a aplicação do método na Figura 6.5.



Figura 6.5: (a) Imagem da Lena; (b) Seu ombro ampliado utilizando interpolação bilinear.

Na Figura 6.5 o ombro da Lena não apresenta tantos *jaggies* como na Figura 6.2 (b), porém é demasiadamente suavizada o que aparenta ser uma imagem desfocada. Observe a Figura 6.6, em que ocorre o mesmo efeito na borda do copo ampliado.

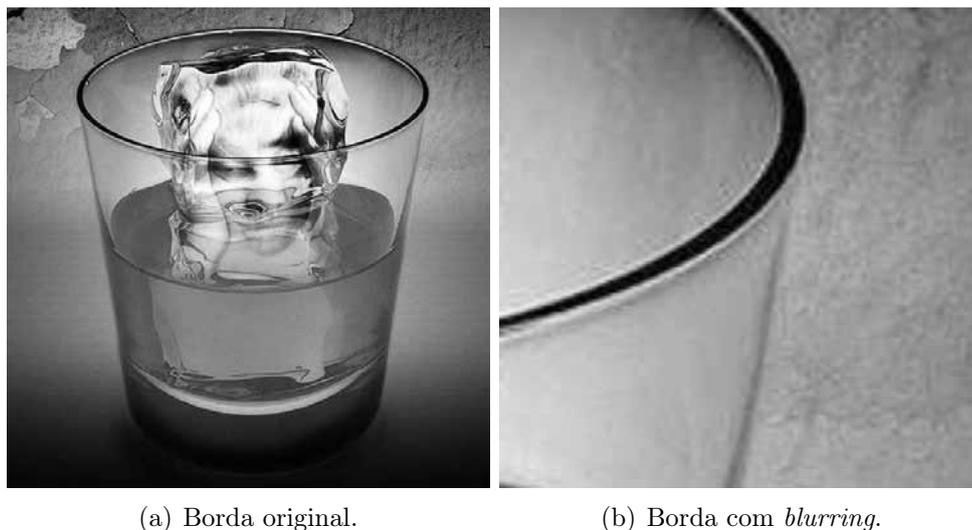


Figura 6.6: Comparação entre a borda antes e depois da aplicação da interpolação bilinear.

Na Figura 6.7 é mostrado um exemplo do que ocorre em imagens de baixa resolução quando se aplica este método.

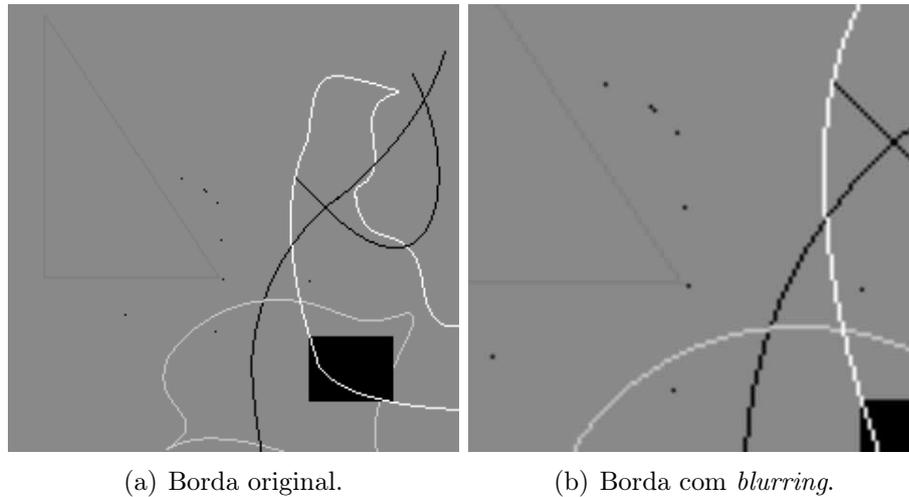


Figura 6.7: Comparação entre a borda antes e depois da aplicação da interpolação bilinear.

Os pontos isolados da Figura 6.7 (a) aparecem “desfocados” na Figura 6.7 (b). Também é evidente que as cores das bordas circulares claras são levemente alteradas por causa das regiões que as rodeiam.

Sobre o tempo que o algoritmo utilizou para a interpolação, a Figura 6.5(b) demorou em média 0,002 segundos, enquanto que a Figura 6.6(b) demorou 0,0028 e a Figura 6.7(b) demorou 0,0012 segundos.

### 6.1.3 Interpolação Bicúbica.

Em geral, o método da interpolação bicúbica é mais vantajoso do que o da bilinear, pois preserva detalhes finos da imagem (Parker [27]). Segundo Gonzalez [13], este método é utilizado em edição de imagens como *Adobe Photoshop* e *Corel Photopaint*. Observe a imagem a Figura 6.8. Os detalhes como o ombro da Lena e em regiões mais homogêneas possuem aspectos mais naturais do que a interpolação bilinear produz. O efeito de serrilhamento também é atenuado.



Figura 6.8: Imagem da Lena e seu ombro ampliado em quatro vezes o tamanho original utilizando interpolação bicúbica.

O fato de utilizar não apenas os quatro vizinhos mais próximos mas dezesseis, permite ao método bicúbico fazer melhores estimativas dos pixels. Por utilizar mais pontos juntamente com suas médias ponderadas, há mais informação sobre a região em volta daquele pixel. A Figura 6.9 mostra outro exemplo da aplicação da interpolação bicúbica.

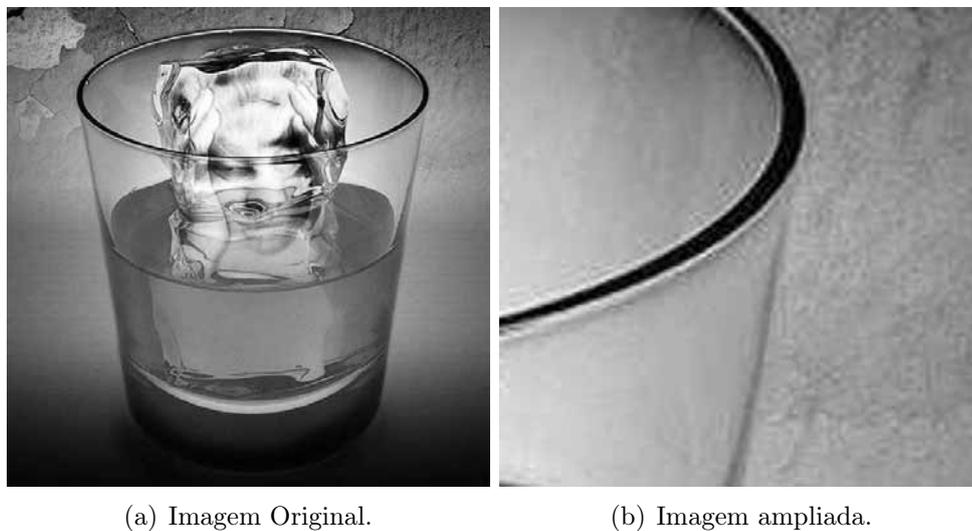


Figura 6.9: (a) Imagem do copo; (b) Parte ampliada pelo método bicúbico.

A suavidade que era causada pelo método bilinear é atenuado no bicúbico. Por ser um método simples computacionalmente e trazer, dentre os anteriores, melhor qualidade visual com aspectos mais naturais, é o mais utilizado na literatura em processamento de imagem para comparação com novos métodos.

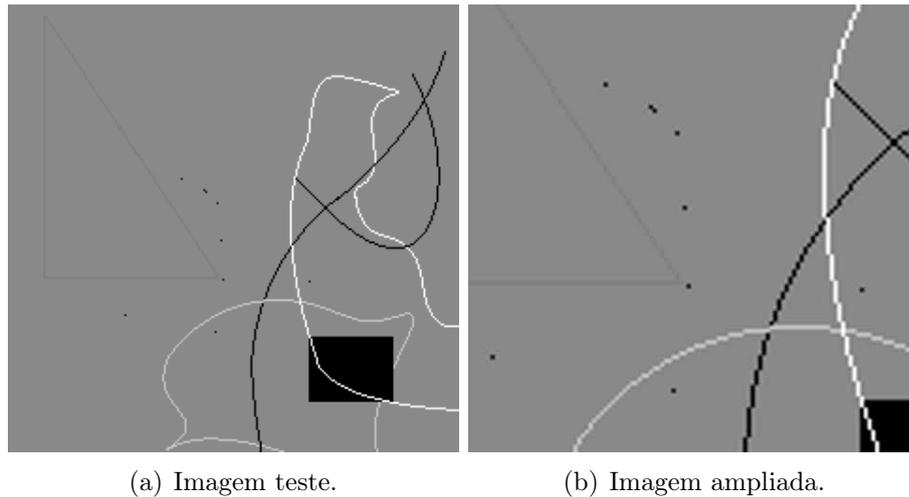


Figura 6.10: (a) Imagem original; (b) Aplicação da interpolação bicúbica.

A interpolação bicúbica não introduz tanto *jaggies*, se comparado com a replicação. Já em comparação à bilinear, o método bicúbico reduz a suavização da imagem.

Sobre o tempo que o algoritmo utilizou para a interpolação, a Figura 6.8(b) demorou em média 0,002 segundos, enquanto que a Figura 6.9(b) demorou 0,0032 e a Figura 6.10(b) demorou 0,0012 segundos.

#### 6.1.4 Interpolação *Sinc*

Na Figura 6.11 é possível perceber o aprimoramento das bordas em relação aos métodos já citados. Nas bordas não aparece o efeito de serrilhamento e os detalhes são preservados.



Figura 6.11: (a) Imagem da Lena; (b) Ombro ampliado quatro vezes utilizando a interpolação *sinc*.

Contudo, próximo às bordas aparecem artefatos pelo efeito de periodicidade, que ocorre devido à propriedades do método. Correções para este tipo de efeito de limite

de bordas foram estudadas por Yaroslavsky [39]. Na Figura 6.12 está a imagem do copo e sua borda ampliada. Em volta dessa borda está o efeito adicionado pelo método da interpolação *sinc*.

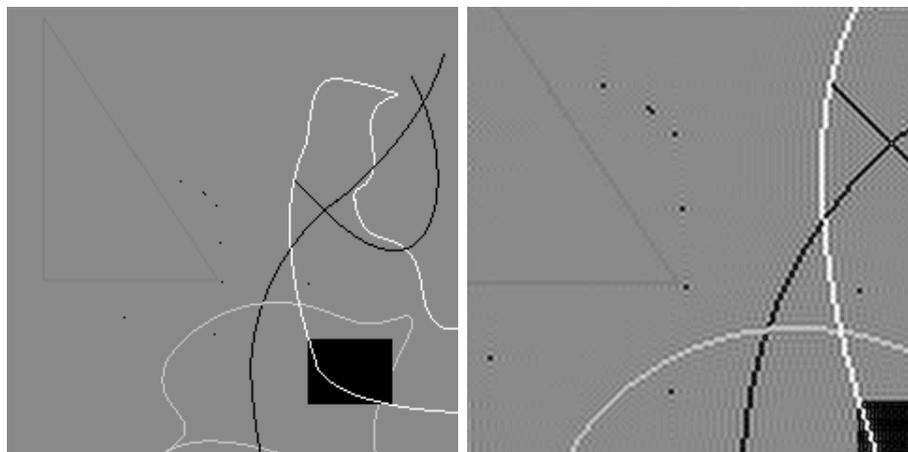


(a) Imagem Original.

(b) Aplicação do método.

Figura 6.12: Representação do que ocorre nas borda pelo método *Sinc*.

Observe a Figura 6.13. O método introduz artefatos que parecem propagação dos contornos das linhas e curvas. Apesar desta falha, o método diminui os *jaggies* e não suaviza a imagem.



(a) Imagem Original.

(b) Aplicação do método.

Figura 6.13: Representação do que ocorre nas bordas pelo método *Sinc*.

Sobre o tempo que o algoritmo utilizou para a interpolação, a Figura 6.11(b) demorou em média 0,095 segundos, enquanto que a Figura 6.12(b) demorou 0,052 e a Figura 6.13(b) demorou 0,0352 segundos.

### 6.1.5 Interpolação Localmente Adaptativo

Observe na Figura 6.14 a aplicação do método localmente adaptativo na imagem da Lena.



(a) Imagem Original.

(b) Imagem Ampliada.

Figura 6.14: Aplicação do método localmente adaptativo por Kumar [20].

Este método melhora as bordas da imagem reduzindo efeitos de serrilhamento. Contudo, produz na imagem um aspecto de “plastificado”, ou seja, diminui a naturalidade da imagem. Observe outro exemplo na Figura 6.15.



(a) Imagem original.

(b) Região da borda ampliada.

Figura 6.15: Ilustração do que ocorre na borda do copo pelo método de interpolação localmente adaptativo.

Na Figura 6.16 é mais evidente o efeito de “plastificado”. É possível também analisar que os pontos isolados da imagem original desapareceram quase todos. As bordas apresentam-se mais grossas e também mais “contínuas”. O efeito *jaggie* não aparece e nem a imagem é suavizada, porém a naturalidade é perdida dependendo das características da imagem.

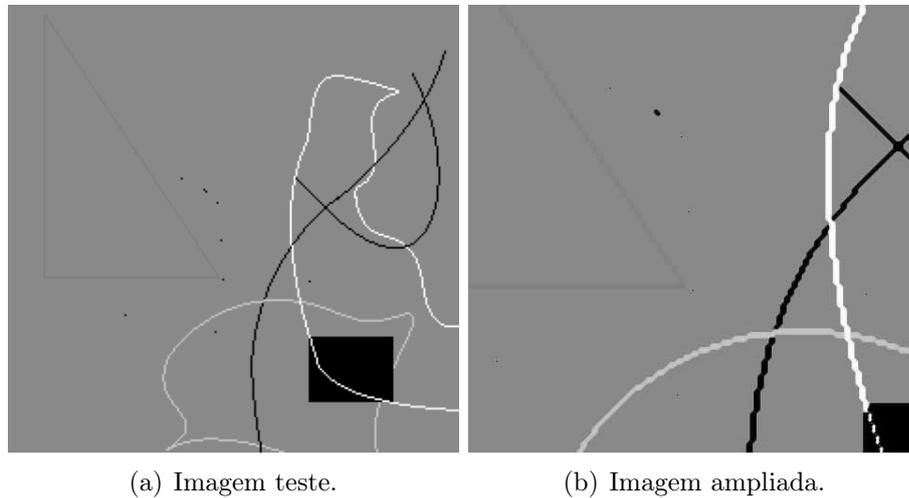


Figura 6.16: Ilustração da interpolação pelo método localmente adaptativo em imagem de baixa resolução.

Sobre o tempo que o algoritmo utilizou para a interpolação, a Figura 6.14(b) demorou em média 0,8528 segundos, enquanto que a Figura 6.15(b) demorou 1,15 e a Figura 6.16(b) demorou 0,7324 segundos.

### 6.1.6 Interpolação NEDI

O método NEDI possui como resultados, imagens com bordas mais contínuas e suaves. Se comparados com os métodos anteriores, reduz a presença de *jaggies* em regiões de curvas, contudo degrada a região que delimita a imagem. Observe na Figura 6.17 que o contorno do ombro da Lena possui curvas suaves, sem muito serrilhado.



Figura 6.17: Aplicação do método NEDI na região do ombro da Lena.

Na Figura 6.18 é possível observar o mesmo efeito de continuidade na borda do copo. Apesar de apresentar bordas mais contínuas, sua complexidade computacional decorre em um custo de tempo muito grande. Outra desvantagem do método é o fato de somente realizar *zoom* na escala de  $2^k$ ,  $k \in \mathbb{N}$ .

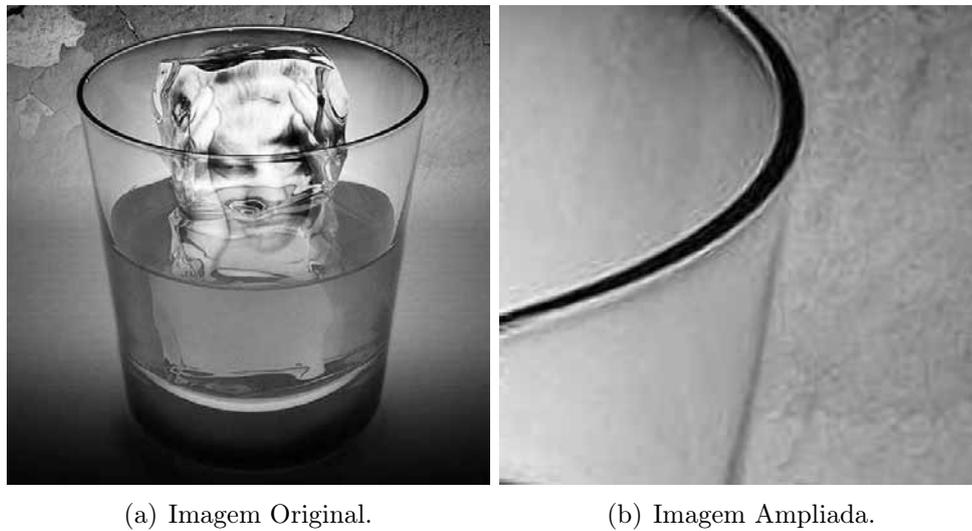


Figura 6.18: Aplicação do método NEDI na borda do copo.

A Figura 6.19 apresenta degradações em algumas regiões, como por exemplo na intersecções de curvas e retas.

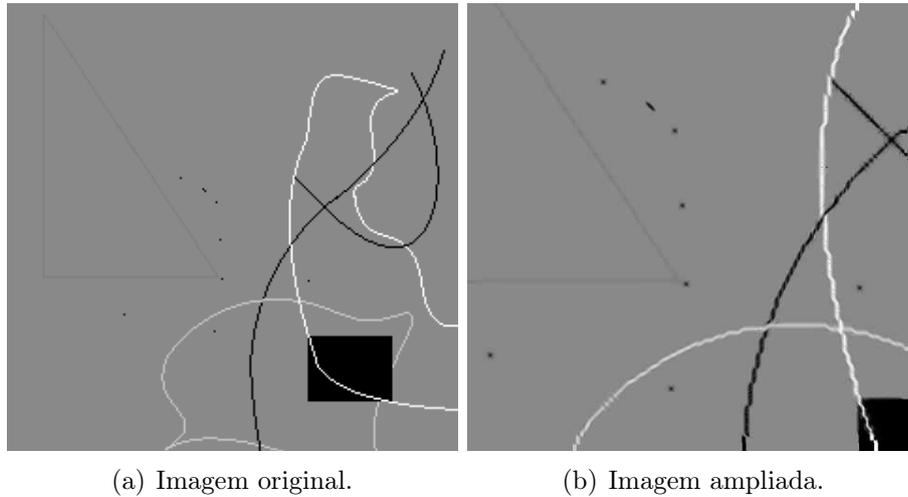


Figura 6.19: Representação do que ocorre nas bordas da imagem de baixa resolução pelo método NEDI.

Sobre o tempo que o algoritmo utilizou para a interpolação, a Figura 6.17(b) demorou em média 7,69 segundos, enquanto que a Figura 6.18(b) demorou 10,545 e a Figura 6.19(b) demorou 4,15 segundos.

### 6.1.7 Interpolação iNEDI

Na Figura 6.20 está o resultado da aplicação do método iNEDI na região do ombro da imagem da Lena. Além das qualidades provenientes do método anterior, a imagem resultante possui mais nitidez em regiões homogêneas.



Figura 6.20: Aplicação do método iNEDI na região do ombro da Lena.

Observe na Figura 6.21, que a borda do copo possui mais suavidade de contorno em relação aos métodos anteriores e não apresenta artefatos nas regiões de fronteira como na Figura 6.18.

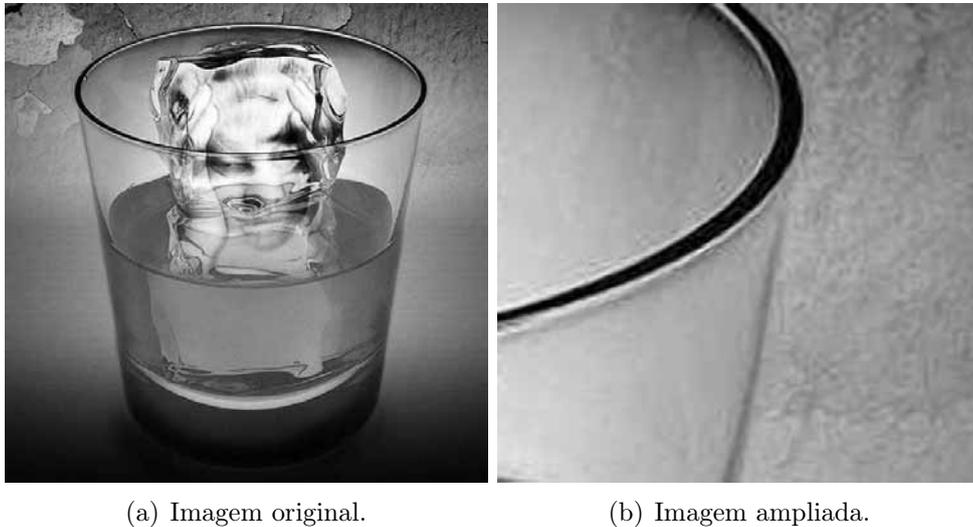


Figura 6.21: Aplicação do método iNEDI na região próximo à borda do copo.

Em relação aos anteriores, o método iNEDI traz bordas mais contínuas e não apresenta *jaggies* nem *blurring*. Contudo, em algumas imagens como no caso da Figura 6.22, o método afina em demasiado as bordas.

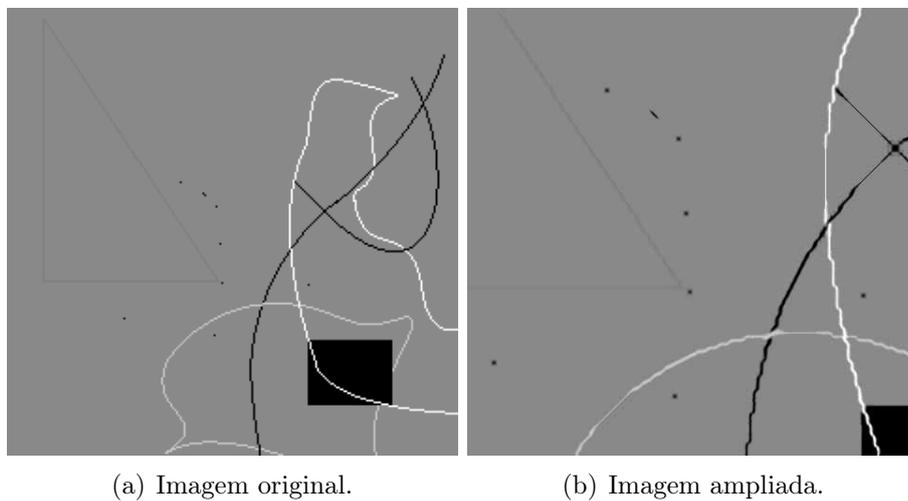


Figura 6.22: Ilustração da aplicação do método iNEDI em imagem de baixa resolução.

Sobre o tempo que o algoritmo utilizou para a interpolação, a Figura 6.20(b) demorou em média 194,36 segundos (ou aproximadamente 3 minutos e 14 segundos), enquanto que a Figura 6.21(b) demorou 255,77 (ou aproximadamente 4 minutos e 15 segundos) e a Figura 6.22(b) demorou 31,37 segundos.

### 6.1.8 Interpolação ICBI

Também desenvolvido por Giachetti [11], o algoritmo ICBI é computacionalmente mais vantajoso do que o iNEDI por causa da sua velocidade de processamento.

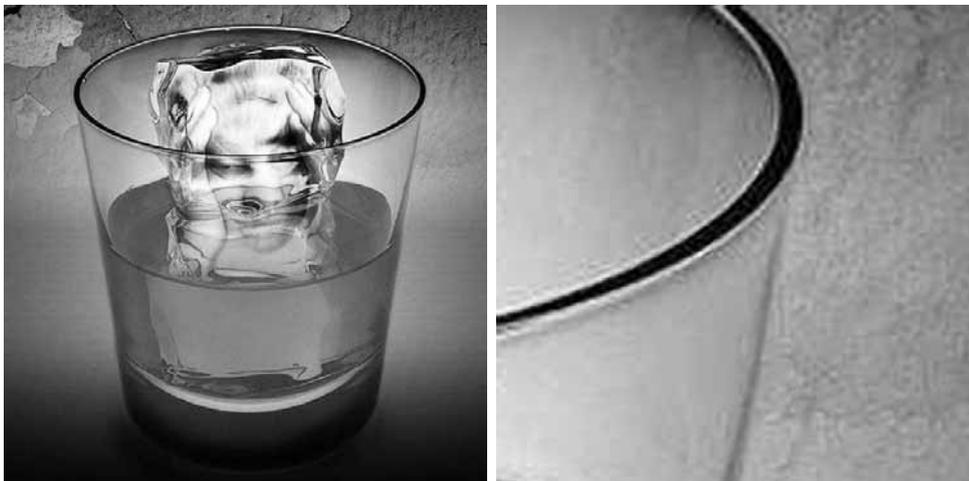


(a) Imagem original.

(b) Imagem ampliada.

Figura 6.23: Região do ombro ampliada utilizando o método ICBI.

Observe as Figuras 6.23 e 6.24. Tanto na imagem da Lena quanto na imagem do copo é visível a semelhança visual com o método iNEDI. Ambas apresentam resultados com *jaggies* e/ou *blurrings* atenuados.



(a) Imagem original.

(b) Imagem ampliada.

Figura 6.24: Aplicação do método ICBI na borda do copo.

Na Figura 6.25 é observado o que as curvas são suaves e que os pontos isolados são interpolados sem modificações. A maior desvantagem do método está no fato de só interpolar imagens na potência de 2.

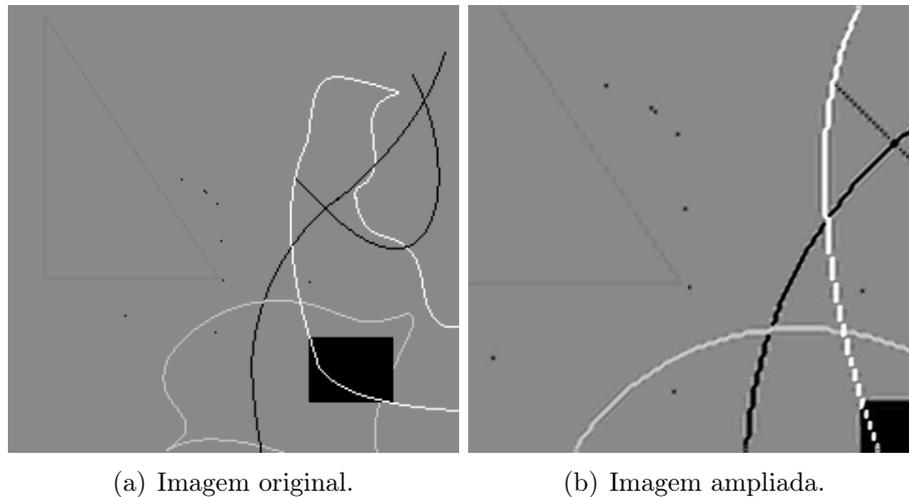


Figura 6.25: O resultado obtido na aplicação do método ICBI em imagem utilizada para teste.

Sobre o tempo que o algoritmo utilizou para a interpolação, a Figura 6.23(b) demorou em média 1,43 segundos, enquanto que a Figura 6.24(b) demorou 3,30 e a Figura 6.25(b) demorou 2,59 segundos.

### 6.1.9 Interpolação utilizando *wavelets* Redundantes

O método proposto por Pagamisse [26] reduz efeitos de serrilhado. Observe sua aplicação na Figura 6.26.



Figura 6.26: Aplicação do método que utiliza *wavelets* redundantes para a ampliação da região do ombro da Lena.

Os detalhes finos encontrado nas bordas são preservados. Contudo, este método depende de parâmetros que são sensíveis às características da imagem utilizada. Na Figura 6.27 é mostrado o resultado da aplicação do método na imagem do copo. A borda ampliada apresenta uma continuidade mais suave, por causa da redução do *jaggies*.

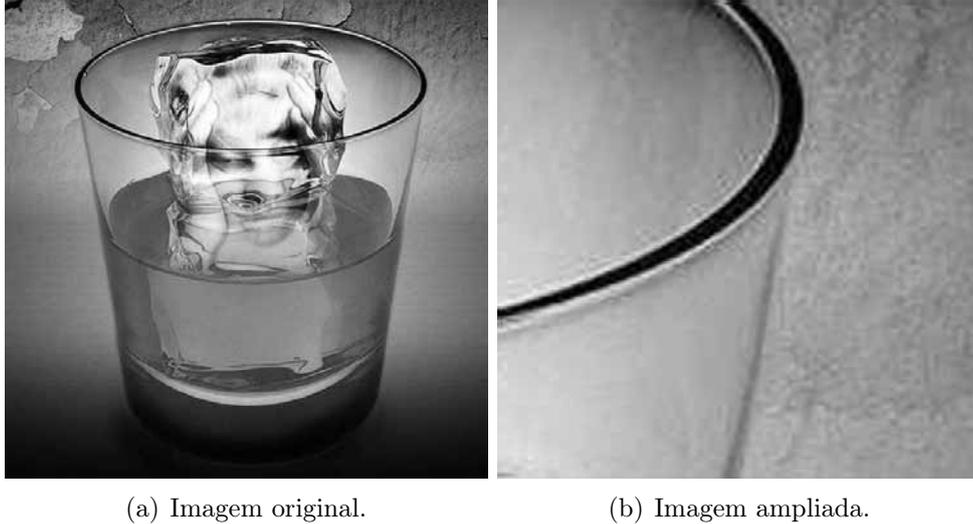


Figura 6.27: Representação do que ocorre na borda do copo pelo método de interpolação utilizando *wavelets* redundantes.

Na Figura 6.28 é possível perceber que o método preserva todos os detalhes da imagem original sem acrescentar artefatos adicionais.

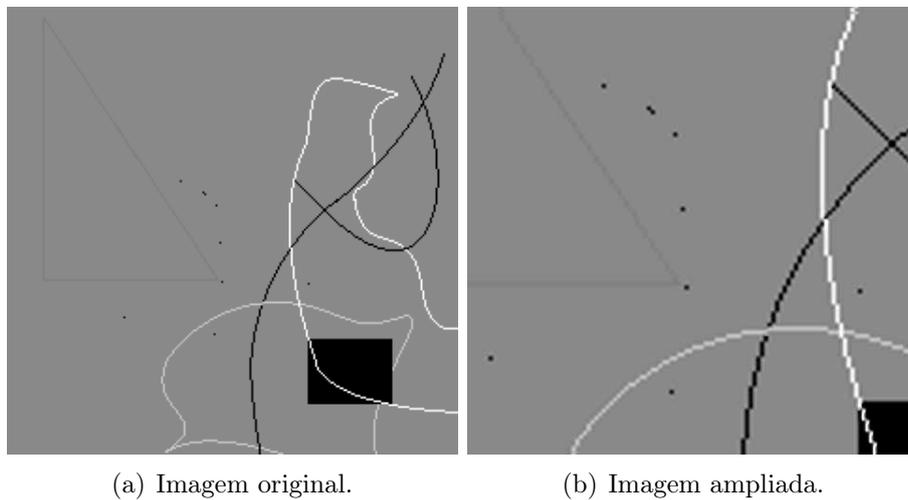


Figura 6.28: Representação do que ocorre em imagens de baixa resolução pelo método de interpolação utilizando *wavelets* redundantes.

Sobre o tempo que o algoritmo utilizou para a interpolação, a Figura 6.26(b) demorou em média 31,24 segundos, enquanto que a Figura 6.27(b) demorou 42,39 e a Figura 6.28(b) demorou 27,35 segundos.

### 6.1.10 Interpolação utilizando Filtro Bilateral

A interpolação utilizando o filtro bilateral é um dos mais recentes métodos que apresenta como resultados imagens com bordas contínuas. Por seu aspecto de suavização, a imagem de saída possui nas regiões homogêneas a redução de ruídos pela filtragem bilateral e os detalhes mais finos são preservados. Observe as Figuras 6.29 e 6.30. Ambas têm como resultados imagens mais naturais e com bordas mais suaves. Na Lena, seu ombro não apresenta *jaggies* e os detalhes de seu cabelo são preservados. Também há uma leve filtragem de ruídos em regiões com superfícies mais homogêneas. Na Figura 6.31 é visível a redução do serrilhado nas bordas. Em regiões de intersecções não há os problemas que ocorrem nos métodos NEDI e iNEDI.



Figura 6.29: Região do ombro da Lena ampliada pelo método de interpolação que utiliza o filtro bilateral.

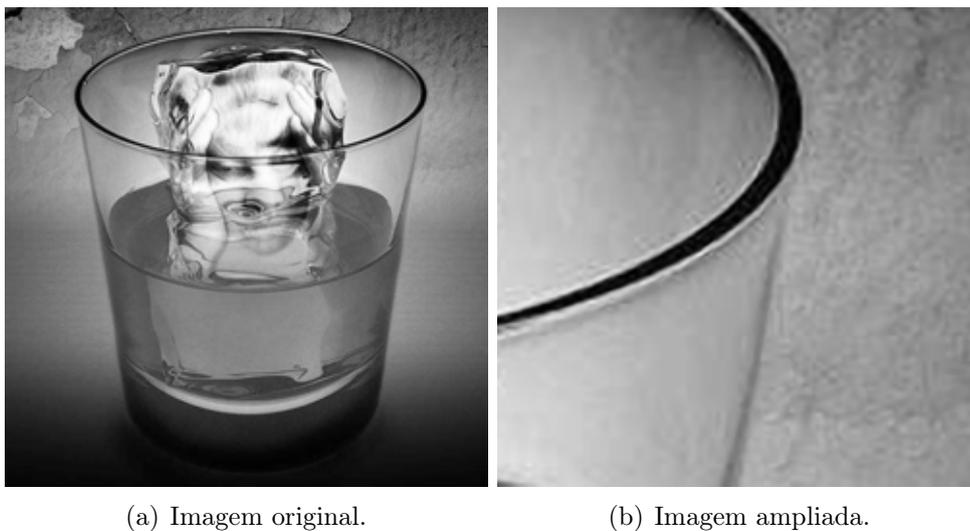


Figura 6.30: Borda do copo ampliada pelo método de interpolação que utiliza o filtro bilateral.

O algoritmo disponibilizado em [16] é executável. Por este motivo a comparação do tempo computacional não se aplica a este método.

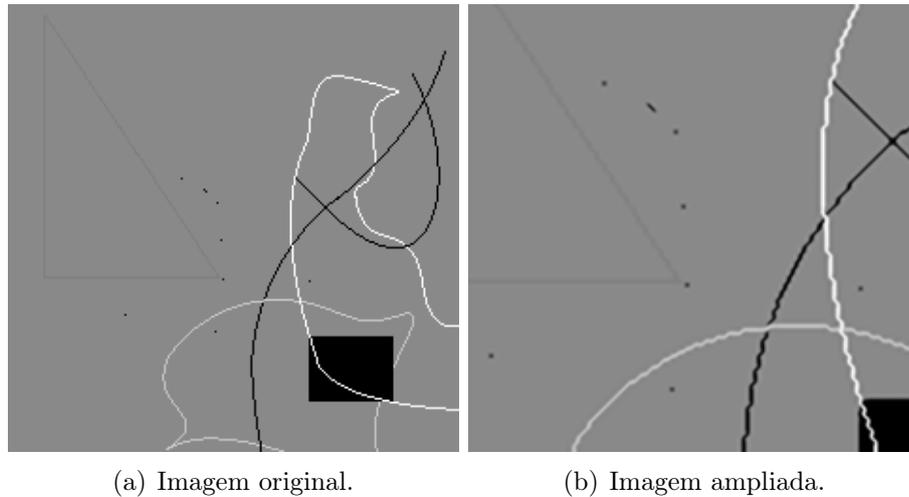


Figura 6.31: Representação do que ocorre em imagem de baixa resolução quando ampliada pelo método de interpolação utilizando filtro bilateral.

## 6.2 Imagens Coloridas

O trabalho desenvolvido se deteve em imagens em escala de cinza. Contudo, todos os métodos propostos podem ser aplicados em imagens coloridas. Para entender o processo, é necessário entender alguns fundamentos sobre cor.

O físico inglês sir Isaac Newton<sup>3</sup> descobriu que um feixe de luz solar quando atravessa um prisma de vidro, emerge espectros contínuos de cores que variam do violeta ao vermelho. Experimento que pode ser utilizado para a explicação física do surgimento do arco-íris após uma chuva. Desde então a luz foi objeto de análise para muitos físicos que estudaram suas propriedades. Em relação a cor, três valores básicos são utilizados para descrever a qualidade de uma fonte de luz cromática (Gonzalez [13]), elas são:

**radiância** : é a quantidade de energia que flui da fonte de luz. De acordo com Pedrini [28], é a quantidade infinitesimal de fluxo radiante contida em um feixe de luz;

**luminância** : este é um fator quantitativo de energia que um observador consegue “perceber” de uma fonte luminosa. Esta é medida em *lumens* (lm);

---

<sup>3</sup>Conhecido pela obra *Philosophiae Naturalis Principia Mathematica*, foi um cientista inglês, mais reconhecido como físico e matemático embora tenha sido também astrônomo, filósofo e teólogo. Sua obra é considerada uma das mais influentes na história da ciência. Publicada em 1687, descreve a lei da gravitação universal e as três leis de Newton que fundamentaram a mecânica clássica.

**brilho** : é subjetivo, e praticamente não se tem medida deste componente. Contudo, incorpora a noção acromática de intensidade e é um dos principais fatores na descrição da sensação de cores.

A luz acromática (sem cores) terá como único atributo, a intensidade. Imagens acromáticas são as imagens em níveis de cinza.

## 6.2.1 Modelos de Cores.

O modelo de cor de uma imagem é basicamente a representação tridimensional na qual cada cor é especificada por um ponto em um sistema de coordenadas. A seguir serão apresentados os modelos de cores RGB, YUV e HSI.

### 6.2.1.1 O modelo RGB

Os cones são os componentes do olho responsáveis pela percepção das cores. Estes podem ser divididos em três principais categorias de sensoriamento, correspondentes às cores azul, verde e vermelho. A visão humana é mais sensível a cor vermelha, seguido da cor verde e azul. Por este motivo, as cores vermelho (R, *red*), verde (G, *green*) e azul (B, *Blue*) são consideradas primárias. Os comprimentos de ondas destas cores, medidas em nanômetros, são aproximadamente; vermelho =  $700 \text{ nm}$ , verde =  $546,1 \text{ nm}$  e azul =  $435,8 \text{ nm}$ . Pode-se, através de composição de cores primárias, obter as cores secundárias. Observe a Figura 6.32:

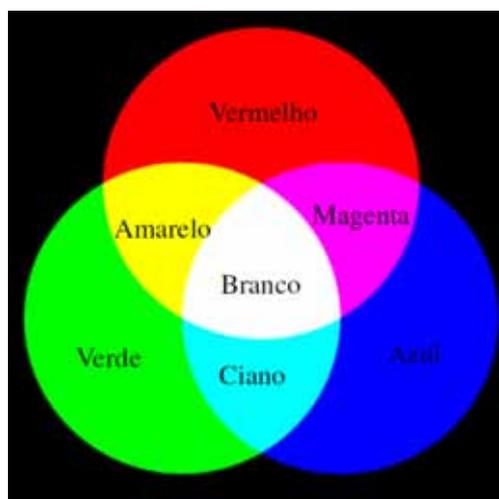


Figura 6.32: Representação das cores do modelo RGB. Fonte: Adaptado a partir de Filho [8].

O modelo RGB se baseia no cubo da Figura 6.33 em que as cores primárias, secundárias, o preto e o branco estão nos vértices. As imagens em escala de cinza estão na diagonal que atravessa o cubo (do preto ao branco).

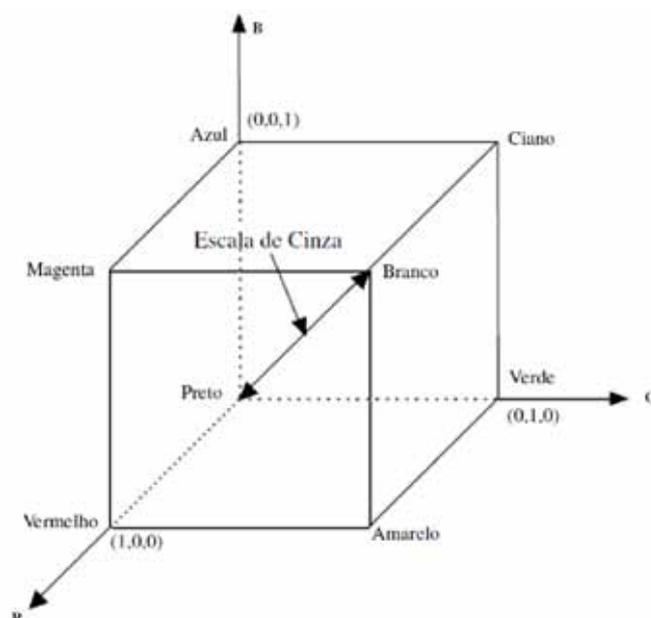


Figura 6.33: Representação tridimensional das cores do modelo RGB. Fonte: Adaptado a partir de Filho [8].

Os vértices do cubo da Figura 6.33 obedecem às seguintes relações:

**Branco** - RGB (255,255,255);

**Azul** - RGB (0,0,255);

**Vermelho** - RGB (255,0,0);

**Verde** - RGB (0,255,0);

**Amarelo** - RGB (255,255,0);

**Magenta** - RGB (255,0,255);

**Ciano** - RGB (0,255,255);

**Preto** - RGB (0,0,0).

Para a interpolação no modelo RGB, basta separar a imagem colorida em três bandas e interpolar cada uma individualmente.

### 6.2.1.2 O modelo YUV

Utilizado nos padrões de televisão PAL (do inglês, *Phase Alternation by Line*) o modelo YUV possui três componentes. O primeiro é o Y, que corresponde à luminância e os componentes U e V codificam as informações de crominância.

Para a transformação do sistema RGB para YUV, é realizado o cálculo segundo a Equação 6.1:

$$\begin{bmatrix} Y \\ U \\ V \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.147 & -0.289 & 0.436 \\ 0.615 & -0.515 & -0.100 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (6.1)$$

em que  $0 \leq R, G, B \leq 1$ .

### 6.2.1.3 O modelo HSI

O modelo HSI separa a imagem também em três partes: matiz (H, Hue), a saturação (S, Saturation) e a intensidade (I, Intensity). Observe a combinação destes três componentes em um espaço de cores tridimensional na Figura 6.34. Qualquer ponto da superfície dessa estrutura representa uma cor puramente saturada. Os vértices correspondem as cores primárias (vermelho, verde e azul).

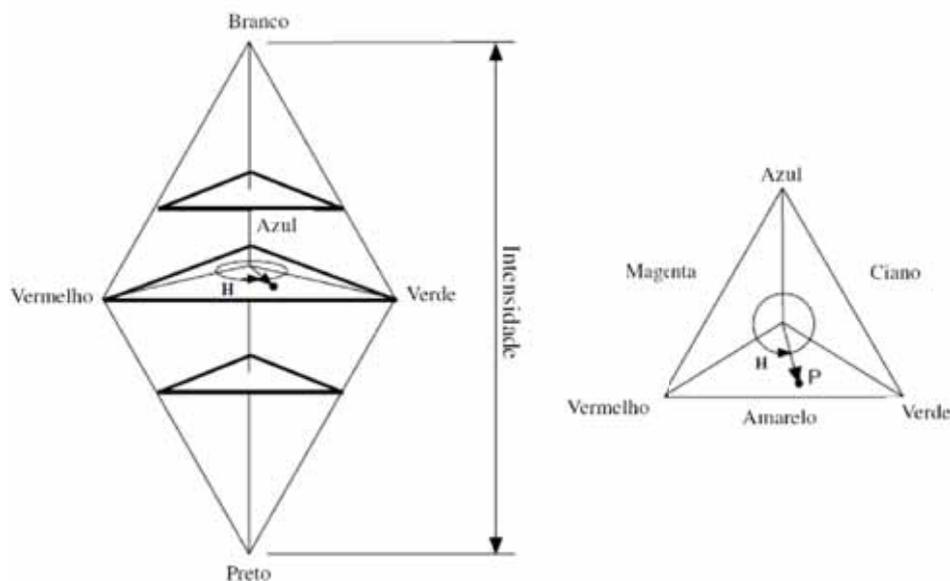


Figura 6.34: Representação das cores do modelo HSI. Fonte: Adaptado a partir de Filho [8].

A matiz  $H$  é determinada pelo ângulo formado em relação ao vértice que corresponde ao vermelho (Figura 6.34(b)). Ou seja, se  $H = 0^\circ$ , a cor é vermelha, se  $H = 60^\circ$ , a cor é amarela, e assim por diante. Os cálculos da conversão a partir das três componentes do RGB estão definidas nas Equações 6.2, 6.3 e 6.4.

$$H = \arccos \left( \frac{(R - G) + (R - B)}{2\sqrt{(R - G)^2 + (R - B)(G - B)}} \right) \quad (6.2)$$

$$S = 1 - \frac{3\min(R, G, B)}{R + G + B} \quad (6.3)$$

$$I = \frac{R + G + B}{3} \quad (6.4)$$

Desta maneira  $0^\circ \leq H \leq 360^\circ$  e  $0 \leq S, I \leq 1$ . Cores mais claras podem ser obtidas através do aumento do brilho ou da luminosidade e o contrário ocorre para cores mais escuras.

## 6.2.2 Aplicações dos métodos em imagens coloridas.

Na Figura 6.35 mostra um corte na imagem colorida de uma cesta de frutas. Esta imagem foi interpolada com alguns métodos para a visualização dos resultados em imagens coloridas. No trabalho desenvolvido, os métodos clássicos, ICBI e o bilateral interpolam a imagem colorida automaticamente. Isto significa que o algoritmo identifica se a imagem é colorida ou monocromática e aplica algum modelo de cor estudado. O método ICBI, NEDI, iNEDI e bilateral se basearam em um método específico (*demosaicking method*). Este modelo é comumente utilizado por câmeras digitais que reconstroem imagens coloridas utilizando amostras sobrepostas das imagens originais.

O método proposto por Pagamisse [26] utilizou o modelo YUV e o método adaptativo segundo Kumar [20] utiliza o modelo usual RGB. No exemplo da Figura 6.36 contudo, foi utilizado o modelo RGB para todas os algoritmos que não interpolam a imagem colorida automaticamente.

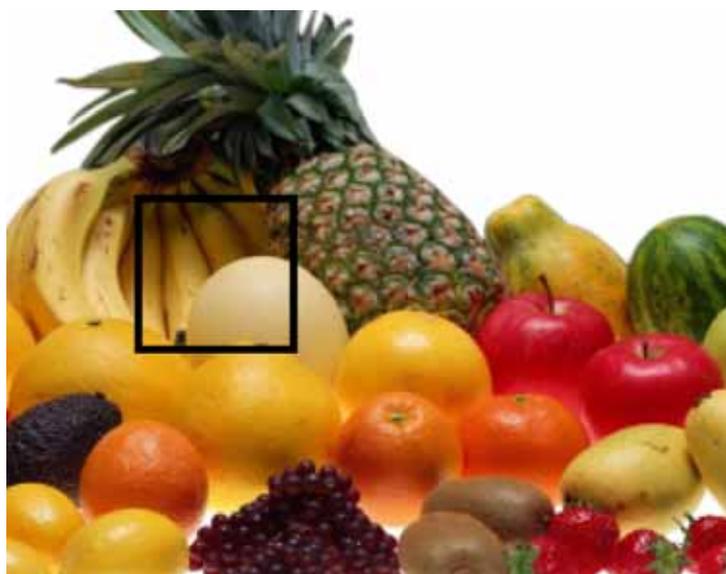


Figura 6.35: Imagem colorida utilizada para a interpolação com os distintos métodos.

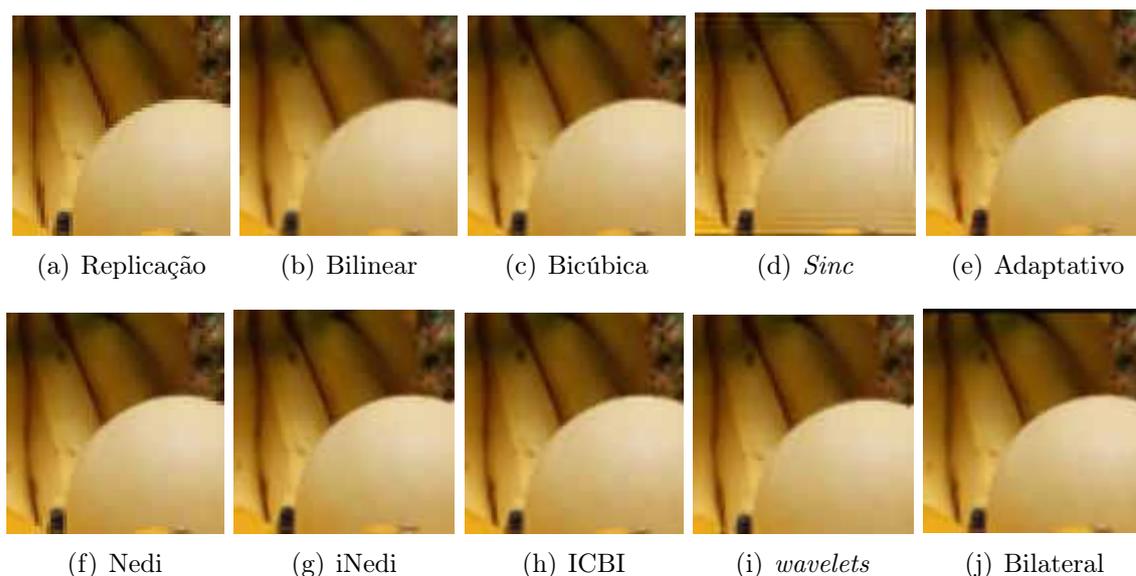


Figura 6.36: Recorte da cesta de frutas ampliada em duas vezes o tamanho original por todos os métodos vistos nos Capítulos 2, 3 e 4.

### 6.3 Comparação dos métodos

A comparação de métodos de interpolação em processamento de imagem pode ser realizada de duas maneiras: de modo visual ou por métricas estatísticas. De maneira visual a verificação da qualidade das imagens é subjetiva e, dependendo da aplicação o algoritmo, pode ser viável ou não. A análise das métricas estatísticas foram discutidas no Capítulo 5, e ainda está em aberto o desenvolvimento de um algoritmo eficaz que traduza de maneira quantitativa a presença ou a ausência de ruídos, artefatos ou mudanças significativa entre duas imagens. A qualidade de uma imagem, quando equiparada com outra, pode ser avaliada como boa ou ruim, dependendo do contexto e aplicação associada. Contudo é imprescindível analisar de diversas maneiras, levando-se em conta a qualidade visual, quantitativa e o custo computacional empregado no processo.

Esta seção está dividida em duas subseções: Na primeira (subseção 6.3.1), é realizada uma comparação de todos os métodos levando-se em conta as métricas estatísticas. Para isso, foram aplicadas as fórmulas apresentadas no Capítulo 5 em imagens distintas (Figura 6.37). Na subseção 6.3.2 são apresentados as imagens resultantes de cada método e a sua comparação visual. As imagens utilizadas como testes para a interpolação e a comparação visual estão na Figura 6.39.

### 6.3.1 Comparação métrica.

Para o estudo estatístico da qualidade dos resultados de cada método, foram utilizadas regiões específicas de cada imagem teste. Na Figura 6.37 estão as regiões utilizadas. A metodologia utilizada para a comparação métrica também é utilizada em [26, 37] e consiste em reduzir as imagens (sub-amostragem) e depois ampliá-las ao seu tamanho original com os métodos descritos nos Capítulos 2, 3 e 4.

Uma consideração importante a se fazer está no tipo de sub amostragem usado para esta técnica. Dependendo do método utilizado para diminuir a imagem, o resultado final pode apresentar mais *jaggies* ou mais suavidade. Para ilustrar, observe a Figura 6.38. Nela está a imagem original (Figura 6.38 (a)) que foi sub amostrada pela interpolação do vizinho mais próximo e pela interpolação bicúbica, (Figuras 6.38 (b) e (c), respectivamente). Ambas foram ampliadas ao seu tamanho original utilizando o método da interpolação bicúbica. Este exemplo mostra que diferentes métodos para reduzir o tamanho da imagem interferem no resultado final quando interpolado ao seu tamanho original.

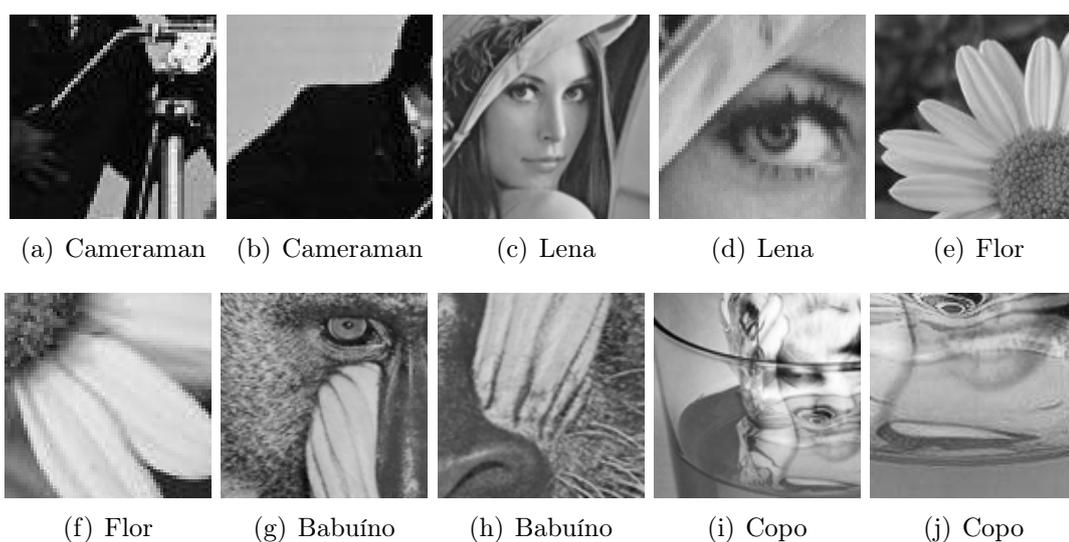


Figura 6.37: (a) Parte da imagem do cameraman tamanho 64x64; (b) Parte da imagem do cameraman tamanho 64x64; (c) Parte da imagem da Lena tamanho 256x256; (d) Parte da imagem da Lena tamanho 64x64; (e) Parte da Flor tamanho 128x128; (f) Parte da Flor tamanho 64x64; (g) Parte da imagem de babuíno (*Baboon*) tamanho 256x256; (h) Parte da imagem de babuíno tamanho 150x150; (i) parte de copo tamanho 256x256; (j) parte de copo tamanho 128x128.

Experimentos foram realizados, em que algumas imagens foram reduzidas utilizando os métodos replicação, bilinear e bicúbico e depois foram ampliadas com todos os métodos estudados. A maneira de sub-amostrar a imagem não favorece nenhum método de interpolação na reamostragem.

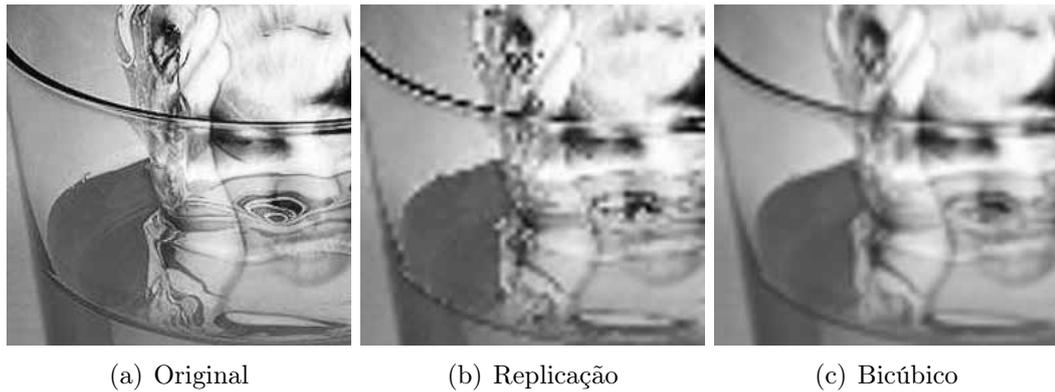


Figura 6.38: (a) imagem original; (b) ampliação utilizando a interpolação por replicação da imagem sub amostrada pela interpolação do vizinho mais próximo; (c) ampliação utilizando a interpolação bicúbica da imagem sub amostrada pela interpolação bicúbica.

Para a avaliação, neste trabalho, as imagens foram reduzidas utilizando o comando *imresize*, no ambiente Matlab. Este comando promove uma sub amostragem com *anti-aliasing*, melhorando o aspecto da imagem.

Os dados obtidos estão na Tabela 6.1. Os dois melhores resultados de cada método estão destacados com um retângulo em volta, de tal forma que a melhor performance está em negrito na cor vermelha. Como as métricas RMSE, NMSE, SNR e CC concordam com a métrica MSE, somente esta última será diferenciada para melhor destaque.

Tabela 6.1: Comparação de todos os métodos de interpolação estudadas utilizando as métricas definidas no Capítulo 5.

Imagem	Método	MSE	RMSE	NMSE	SNR	CC	IQI
Quanto:		mais próximas de 0 melhor			maior, melhor	mais próximo de 1 melhor	
Cameraman Figura 6.37 (a)	Replicação	1227.346	35.033	0.165	7.83	0.855	0.640
	Bilinear	1091.526	33.038	0.146	8.34	0.876	0.714
	Bicúbica	<b>996.242</b>	31.563	0.134	8.74	0.885	0.723
	<i>sinc</i>	1009.669	31.775	0.135	8.68	0.883	0.718
	Adaptativo	1346.886	36.700	0.181	7.43	0.843	0.701
	NEDI	1121.276	33.485	0.150	8.23	0.869	0.722
	iNEDI	1070.688	32.721	0.144	8.43	0.875	<b>0.730</b>
	ICBI	1021.754	31.965	0.137	8.63	0.881	0.728
	<i>Wavelet</i>	<b>1008.475</b>	31.756	0.135	8.69	0.884	0.719
	Bilateral	1054.355	32.471	0.141	8.49	0.878	<b>0.732</b>
Cameraman Figura 6.37 (b)	Replicação	551.833	23.491	0.044	13.60	0.956	0.797
	Bilinear	390.752	19.767	0.031	15.10	0.969	0.835
	Bicúbica	<b>339.330</b>	18.421	0.027	15.71	0.973	0.835
	<i>sinc</i>	363.463	19.065	0.029	15.41	0.971	0.844
	Adaptativo	381.011	19.519	0.030	15.21	0.970	<b>0.897</b>
	NEDI	399.108	19.978	0.032	15.01	0.969	0.844
	iNEDI	371.086	19.264	0.029	15.32	0.971	0.864

Continua na próxima página.

Tabela 6.1 – continuação da página anterior

Imagem	Método	MSE	RMSE	NMSE	SNR	CC	IQI
Quanto:		mais próximas de 0 melhor			maior, melhor	mais próximo de 1 melhor	
	ICBI	372.993	19.313	0.030	15.30	0.971	0.838
	<i>Wavelet</i>	341.430	18.478	0.027	15.69	0.973	0.834
	Bilateral	<b>331.008</b>	18.194	0.026	15.82	0.974	0.872
Lena Figura 6.37 (c)	Replicação	199.849	14.137	0.013	18.99	0.962	0.746
	Bilinear	154.111	12.414	0.010	20.11	0.972	0.831
	Bicúbica	<b>127.246</b>	11.280	0.008	20.95	0.976	0.849
	<i>sinc</i>	143.774	11.991	0.009	20.42	0.973	<b>0.851</b>
	Adaptativo	218.366	14.777	0.014	18.60	0.959	0.806
	NEDI	157.015	12.531	0.010	20.03	0.971	0.839
	iNEDI	151.295	12.300	0.010	20.19	0.972	0.842
	ICBI	145.057	12.044	0.009	20.38	0.973	<b>0.851</b>
	<i>Wavelet</i>	129.431	11.377	0.008	20.87	0.976	0.847
Bilateral	143.417	11.976	0.009	20.43	0.973	0.843	
Lena Figura 6.37 (d)	Replicação	240.549	15.510	0.011	19.49	0.929	0.756
	Bilinear	186.013	13.639	0.009	20.61	0.949	0.841
	Bicúbica	<b>146.668</b>	12.111	0.007	21.64	0.958	0.861
	<i>sinc</i>	172.200	13.123	0.008	20.94	0.950	0.862
	Adaptativo	297.639	17.252	0.014	18.57	0.913	0.810
	NEDI	200.790	14.170	0.009	20.28	0.942	0.845
	iNEDI	189.815	13.777	0.009	20.52	0.945	0.853
	ICBI	174.863	13.224	0.008	20.88	0.949	<b>0.865</b>
	<i>Wavelet</i>	628.596	25.072	0.029	15.32	0.814	0.780
Bilateral	419.581	20.484	0.020	17.07	0.874	0.783	
Flor Figura 6.37 (e)	Replicação	284.150	16.857	0.015	18.21	0.958	0.729
	Bilinear	231.329	15.210	0.012	19.10	0.966	0.781
	Bicúbica	<b>197.403</b>	14.050	0.010	19.79	0.971	0.806
	<i>sinc</i>	204.804	14.311	0.011	19.63	0.970	<b>0.822</b>
	Adaptativo	273.195	16.529	0.015	18.38	0.959	0.771
	NEDI	233.882	15.293	0.012	19.05	0.965	0.787
	iNEDI	219.410	14.812	0.012	19.33	0.967	0.789
	ICBI	201.122	14.182	0.011	19.71	0.970	0.819
	<i>Wavelet</i>	200.232	14.150	0.011	19.73	0.970	0.802
Bilateral	207.323	14.399	0.011	19.58	0.969	0.794	
Flor Figura 6.37 (f)	Replicação	331.790	18.215	0.010	20.15	0.947	0.728
	Bilinear	226.972	15.066	0.007	21.80	0.965	0.820
	Bicúbica	187.365	13.688	0.005	22.63	0.970	0.839
	<i>sinc</i>	189.191	13.755	0.006	22.59	0.970	0.845
	Adaptativo	247.734	15.740	0.007	21.42	0.961	0.800
	NEDI	202.453	14.229	0.006	22.29	0.968	0.833
	iNEDI	<b>172.008</b>	13.115	0.005	23.00	0.973	<b>0.847</b>
	ICBI	185.281	13.612	0.005	22.68	0.971	0.843
	<i>Wavelet</i>	189.533	13.767	0.006	22.58	0.970	0.836

Continua na próxima página.

Tabela 6.1 – continuação da página anterior

Imagem	Método	MSE	RMSE	NMSE	SNR	CC	IQI
Quanto:		mais próximas de 0 melhor			maior, melhor	mais próximo de 1 melhor	
Babuíno Figura 6.37 (g)	Bilateral	177.225	13.313	0.005	22.87	0.972	0.835
	Replicação	765.677	27.671	0.043	13.62	0.848	0.595
	Bilinear	738.097	27.168	0.042	13.78	0.855	0.605
	Bicúbica	705.979	26.570	0.040	13.97	0.861	0.644
	<i>sinc</i>	720.974	26.851	0.041	13.88	0.857	0.677
	Adaptativo	811.744	28.491	0.046	13.37	0.838	0.574
	NEDI	757.078	27.515	0.043	13.67	0.850	0.603
	iNEDI	743.484	27.267	0.042	13.75	0.853	0.618
	ICBI	724.259	26.912	0.041	13.86	0.857	0.646
<i>Wavelet</i>	709.961	26.645	0.040	13.95	0.860	0.637	
Bilateral	734.647	27.104	0.042	13.80	0.855	0.613	
Babuíno Figura 6.37 (h)	Replicação	436.092	20.883	0.024	16.21	0.900	0.659
	Bilinear	409.285	20.231	0.022	16.49	0.907	0.697
	Bicúbica	376.562	19.405	0.021	16.85	0.914	0.733
	<i>sinc</i>	392.015	19.799	0.022	16.67	0.910	0.757
	Adaptativo	463.417	21.527	0.025	15.95	0.893	0.669
	NEDI	399.899	19.997	0.022	16.59	0.909	0.712
	iNEDI	393.051	19.826	0.022	16.66	0.910	0.723
	ICBI	393.358	19.833	0.022	16.66	0.910	0.736
	<i>Wavelet</i>	379.745	19.487	0.021	16.81	0.914	0.726
Bilateral	392.507	19.812	0.022	16.67	0.910	0.710	
Copo Figura 6.37 (i)	Replicação	373.939	19.337	0.014	18.65	0.938	0.703
	Bilinear	331.051	18.195	0.012	19.18	0.946	0.788
	Bicúbica	299.204	17.298	0.011	19.62	0.951	0.806
	<i>sinc</i>	306.484	17.507	0.011	19.51	0.949	0.811
	Adaptativo	400.286	20.007	0.015	18.35	0.933	0.763
	NEDI	322.272	17.952	0.012	19.29	0.947	0.793
	iNEDI	310.584	17.623	0.011	19.45	0.949	0.800
	ICBI	305.535	17.480	0.011	19.53	0.949	0.807
	<i>Wavelet</i>	302.076	17.380	0.011	19.58	0.950	0.804
Bilateral	305.296	17.473	0.011	19.53	0.950	0.796	
Copo Figura 6.37 (j)	Replicação	214.812	14.656	0.009	20.55	0.916	0.716
	Bilinear	188.157	13.717	0.008	21.12	0.928	0.789
	Bicúbica	171.258	13.087	0.007	21.53	0.934	0.809
	<i>sinc</i>	173.737	13.181	0.007	21.47	0.933	0.815
	Adaptativo	224.793	14.993	0.009	20.35	0.912	0.769
	NEDI	193.710	13.918	0.008	21.00	0.925	0.790
	iNEDI	189.844	13.778	0.008	21.08	0.927	0.799
	ICBI	177.855	13.336	0.007	21.37	0.931	0.809
	<i>Wavelet</i>	172.316	13.127	0.007	21.50	0.934	0.807
Bilateral	185.856	13.633	0.008	21.18	0.928	0.793	

Para a avaliação, foram escolhidas cinco imagens com diferentes texturas. Desta maneira, é possível determinar quais métodos são mais recomendáveis de acordo com as características da imagem.

A Figura 6.37(a) apresenta como melhores métodos, referente à métrica MSE, as interpolações bicúbica, *wavelet*, *sinc* e ICBI, respectivamente. Pela Tabela 6.1 é possível observar que o método iNEDI e o bilinear obtiveram valores próximo. Os piores resultados foram do método adaptativo e replicação. As métricas RMSE, NMSE, SNR e CC possui resultados semelhantes ao do MSE. Apenas o IQI apresenta resultados que diferem dos outros. Segundo a Tabela 6.1, os métodos que apresentam melhores performances são: o bilateral, iNEDI, ICBI e bicúbica. Os que apresentam as piores taxas são os métodos da replicação e o adaptativo.

Sobre a Figura 6.37(b), o método bilateral apresenta o menor valor, seguido do bicúbico e *wavelet*. O pior resultado é do método de replicação. Pela métrica IQI o melhor método é dado pelo adaptativo seguido do bilateral e iNEDI. O pior dentre todos é a replicação.

Pela Tabela 6.1, a Figura 6.37(c) possui a bicúbica como melhor interpolação. Também o método da *wavelet* seguido do bilateral e *sinc*. Segundo as métricas MSE, RMSE, NMSE, SNR e CC os piores resultados são dados pelo adaptativo e a replicação. Pela métrica IQI, as melhores interpolações ocorrem com os métodos *sinc* e ICBI. O que apresenta o pior resultado é a replicação.

Sobre a Figura 6.37(d), a interpolação bicúbica é a melhor seguido da *sinc*, ICBI e iNEDI. Contudo as piores estão com a *wavelet* e a bilateral. Pelo IQI, as melhores são ICBI, *sinc* e bicúbica. Os métodos que não conseguiram aproximar a imagem resultante com a original foram, nesta ordem, a replicação, *wavelet* e bilateral.

A Tabela 6.1 demonstra que os melhores resultados em relação à métrica MSE, para a Figura 6.37(e), são os algoritmos de interpolação bicúbica, seguido da interpolação *wavelet* e ICBI. Os que possuem as piores performance são, os métodos antigos, a replicação e dos novos apresentados, o algoritmo adaptativo. Pelo IQI tem-se que o método que mais se aproxima da imagem original é a *sinc* seguido do ICBI. O pior se mantém com a replicação.

Os métodos que trazem os melhores desempenhos em relação à métrica MSE da Figura 6.37(f) são o iNEDI, bilateral, ICBI e bicúbico. O método de replicação é o pior dentre os outros. Em relação à métrica IQI a replicação também apresenta a pior eficiência, enquanto os melhores, são: iNEDI, *sinc* e o bicúbico.

Os métodos bicúbico, *wavelet*, *sinc* e ICBI apresentam os melhores resultados estatísticos pela métrica MSE da Figura 6.37(g). Os piores são a replicação e o método adaptativo. Pela métrica IQI os melhores são a *sinc* seguida da ICBI e bicúbica. As piores performances continuaram nos métodos replicação e adaptativo.

Na Figura 6.37(h), em relação à métrica MSE os algoritmos mais eficientes são o bicúbico e a *wavelet*. Todos os outros apresentaram valores muito próximos com exceção do

método de replicação e adaptativo, que ainda exhibe resultados de imagens muito distantes da original, quantitativamente. Em comparação com o IQI, os métodos mais eficientes são a *sinc* e o ICBI.

Na Figura 6.37(i), os métodos que apresentam os melhores desempenhos em relação à métrica MSE são: bicúbico, *wavelet*, bilateral e o ICBI. Pelo IQI, os melhores são o método da função *sinc*(·), o ICBI, a bicúbica e a *wavelet*.

Por fim, para a Figura 6.37(j), os métodos que produzem imagens que mais se aproximam de suas características originais são os métodos: bicúbico, *wavelet* e ICBI. Os que menos se aproximam são: replicação e o adaptativo. Contudo, pelo IQI os que mais apresentam bons resultados são a interpolação *sinc*, o bicúbico e a ICBI. O pior desempenho está na interpolação por replicação.

De maneira geral as técnicas que mais apresentaram melhores performances dos métodos mais usuais foi o bicúbico, enquanto dos apresentados nos Capítulos 3 e 4, foram o ICBI, *wavelets* e bilateral. Os métodos NEDI e iNEDI apresentaram, em sua maioria, valores próximos. O método que utiliza a função *sinc*(·) apresenta bons resultados numéricos em algumas imagens. Com exceção da Figura 6.37 (b), o método adaptativo apresenta como resultado imagens que ainda estão distantes de uma interpolação mais precisa. Contudo, como já esperado, o pior dentre os métodos pelas métricas utilizadas é o método do vizinho mais próximo.

É possível concluir que dentre os métodos clássicos o método bicúbico e a interpolação *sinc* são os melhores métodos pelas métricas estatísticas. Sobre os novos métodos, a interpolação ICBI e *wavelets* são os que mais possuem resultados quantitativos melhores em relação aos outros.

### 6.3.2 Análise Visual

Para esta análise, foram utilizadas todas as imagens da Figura 6.39. Observe que foi utilizada uma região específica, como foi feito na subseção anterior, e então foi realizada a comparação dos métodos entre si. Ao contrário das métricas estatísticas, a comparação visual é subjetiva e o mesmo algoritmo pode ser considerado muito ou pouco eficaz dependendo das características da imagem e do custo computacional que o usuário pretende utilizar.

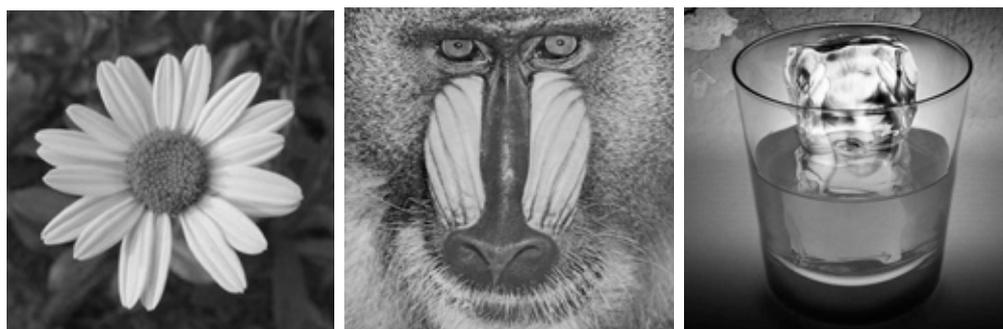
A primeira imagem usada foi a Figura 6.39 (a). Foi retirado da imagem do Cameraman uma região 128x128 e ampliada pelo fator de dois por todos os métodos estudados. Esta metodologia foi escolhida para uma investigação mais precisa no que ocorre com a imagem em determinadas regiões, quando aplicados os métodos.

Na Figura 6.40 é possível notar que a imagem resultante do método replicação apresenta a pior performance. A Figura 6.40 (a) apresenta muito *jaggies* e quaisquer ruídos



(a) Cameraman

(b) Lena



(c) Flor

(d) Babuíno

(e) Copo

Figura 6.39: Imagens testes utilizadas para a comparação visual dos métodos descrito nos Capítulos 2,3 e 4 .

isolados podem ser suprimidos ou ampliados, já que o novo pixel receberá o valor do ponto mais próximo.

Na Figura 6.40 (b) mostra o resultado da interpolação bilinear. Como já foi discutido no Capítulo 2, a imagem fica demasiadamente suavizada. A Figura 6.40 (c) apresenta o resultado do método bicúbico. Este é o mais utilizado, e apesar de fornecer uma redução de *jaggies*, não interpola as bordas de maneira eficaz para obter imagens de alta resolução. Na Figura 6.40 (d) tem-se a interpolação pela função  $\text{sinc}(\cdot)$ . As bordas da imagem, principalmente no tripé da câmera, é bem superior ao método bicúbico, porém a imagem inteira recebe a interferência de ruídos periódicos, como se fossem propagação de ondas. Na Figura 6.40 (e) está o efeito produzido pela interpolação adaptativo segundo Kumar [20]. Os *jaggies* são reduzidos, entretanto a imagem perde sua naturalidade e passa a possuir o mesmo aspecto de pintura a óleo ou imagem plastificada.



(a) Replicação



(b) Bilinear



(c) Bicúbica

(d) *Sinc*

(e) Localmente adaptativo



(f) NEDI

Figura 6.40: (a) Interpolação Replicação; (b) Interpolação Replicação (c) Interpolação Bicúbica; (d) Interpolação *Sinc*; (e) Interpolação pelo algoritmo localmente adaptativo; (f) Interpolação NEDI.

Na Figura 6.40 (f) está a aplicação do método NEDI. Nesta figura é possível observar que os *jaggies* no tripé da câmera são inexistentes e quaisquer ruídos em regiões homogêneas desaparecem. Porém alguns artefatos próximos às bordas são frequentes na imagem inteira, principalmente em regiões que delimitam a imagem. O método mantém os detalhes do fundo, contudo a imagem do cameraman e sua câmera passam a ter um aspecto não natural.

Na Figura 6.41 (a) é exibido o resultado do iNEDI, que possui uma aparência mais natural e não degrada as bordas que delimitam a imagem inteira. Este método diminui também os artefatos produzido pelo método NEDI e mantém os detalhes com bordas mais suaves sem a presença de serrilhado.

A Figura 6.41 (b) mostra o resultado do ICBI. A imagem resultante apresenta aspectos parecidos com a interpolação iNEDI, contudo não produz bordas tão suaves e apresenta alguns ruídos em volta, como no método NEDI. Apesar disso, a aparência da Figura 6.41 (b) expõe uma característica mais natural.



(a) iNEDI

(b) ICBI

Figura 6.41: (a) Interpolação iNEDI; (b) Interpolação ICBI.

O cameraman pelo método das *wavelets* (Figura 6.41 (a)) resulta em uma imagem parecida com o produzido pelo método bicúbico.

Por fim, na Figura 6.42 (b) tem-se a aplicação do método de interpolação utilizando o filtro bilateral. Este traz como resultado uma imagem do cameraman parecido com o produzido pelo iNEDI com os ruídos mais suprimidos. Apesar da boa aparência, a perna do tripé esquerdo possui uma linha branca menos contínua do que a do iNEDI. Outra desvantagem está na suavização que, apesar de ser pouco, apresenta aparência com perda de naturalidade.

(a) *Wavelets* redundantes

(b) Filtro bilateral

Figura 6.42: (a) Interpolação utilizando *wavelets* redundantes; (b) Interpolação utilizando filtro bilateral.

Dentre todos os métodos, as imagens que revelam-se melhores são os resultantes dos métodos NEDI, iNEDI e bilateral.

Na Figura 6.39 (b) está a imagem da Lena de tamanho 512x512, a qual apresenta detalhes diferentes da Figura 6.39 (a). A região retirada para a comparação é de tamanho 128x128 e inclui partes homogêneas como o ombro e a bochecha e partes ricas em detalhes, como as penas e bordas do chapéu. Na Figura 6.43 (a) está o resultado do método replicação. É possível observar que a imagem fica totalmente degradada com os *jaggies* e todos os detalhes suaves são comprometidos. A Figura 6.43 (b) mostra o bilinear que ainda apresenta serrilhado nas bordas e os detalhes são afetados pelo efeito *blurring*.

Na Figura 6.43 (c) está o método bicúbico que possui resultado parecido com a Figura 6.44 (c), da interpolação *wavelet*. Ambos mantém os detalhes das bordas mas não conseguem interpolar sem introdução de *jaggies*. Na Figura 6.43 (d) está a Lena com a aplicação da *sinc*. Perceba que o chapéu exibe uma borda contínua; isso também ocorre no contorno da bochecha e do ombro. Contudo, esta técnica introduz o efeito periódico nas regiões mais homogêneas. O método adaptativo consegue preservar as formas dos contornos do chapéu e da bochecha (Figura 6.43 (e)), entretanto, perde alguns detalhes das penas e os aspecto geral da imagem perde a naturalidade. A Figura 6.43 (f) mostra bordas bem preservadas e contínuas. O problema do NEDI nesta aplicação se encontra nas regiões que delimitam a imagem. Observe que nas penas e na região do olho são introduzidos artefatos como se fossem contornos adicionais.



(a) Replicação

(b) Bilinear



(c) Bicúbica

(d) *Sinc*

(e) Localmente adaptativo

(f) NEDI

Figura 6.43: (a) Interpolação Replicação; (b) Interpolação Bilinear; (c) Interpolação Bicúbica; (d) Interpolação *Sinc*; (e) Interpolação pelo algoritmo localmente adaptativo; (f) Interpolação NEDI.

Na Figura 6.44 (a) estes problemas não aparecem mais, pois o iNEDI consegue conservar as informações sem adicionar estes tipos de ruídos. O algoritmo ICBI é capaz de manter as informações contidas na imagem original e ainda não permite que o resultado perca o aspecto de naturalidade, como visto na Figura 6.44 (b). Em comparação com o iNEDI, o ICBI consegue conservar melhor os detalhes da pena no chapéu. Na Figura 6.44 (d) tem-se a imagem da Lena depois da aplicação da interpolação com o filtro bilateral. Os detalhes de bordas são bem preservadas assim como os ruídos nas regiões homogêneas, que são filtradas. Contudo, pela característica de suavizar, alguns detalhes são suprimidos como os riscos transversais do chapéu. Isto também ocorre nas penas e no cabelo.



Figura 6.44: (a) Interpolação iNEDI; (b) Interpolação ICBI; (c) Interpolação *wavelets* (d) Interpolação bilateral.

De maneira geral, os métodos que exibiram melhores resultados visuais nesta imagem foram: o iNEDI, o ICBI e o bilateral.

Na Figura 6.45 (a), tem-se a aplicação do método replicação em uma região que possui dimensão 128x128 da Figura 6.39 (c). As pétalas adquirem bordas completamente serrilhadas e os detalhes do centro da flor adquirem forma quadriculada. Na Figura 6.45 (b), o centro da flor perde detalhes por causa da suavização causada pela bilinear. Na Figura 6.45 (c) está a aplicação do bicúbico, em que as bordas das pétalas estão mais contínuas; contudo os ruídos também são realçados. A aplicação do método *sinc* na Figura 6.45 (d) não suaviza tanto a imagem quanto o bilinear e o bicúbico, porém a região no fundo aparecem artefatos introduzidos pelo método.



Figura 6.45: (a) Interpolação Replicação; (b) Interpolação Bilinear; (c) Interpolação Bicúbica; (d) Interpolação *Sinc*.

Na Figura 6.46 (a), o algoritmo localmente adaptativo gera uma imagem com menos *jaggies* e com ruídos reduzidos. Contudo, além do aspecto não natural, os detalhes no centro da flor são degradados. Na Figura 6.46 (b), a aplicação NEDI conserva detalhes importantes das bordas tanto nas pétalas, como no centro. Porém mantém os problemas com as regiões que delimitam a imagem. A Figura 6.46 (c) apresenta uma flor com pétalas mais contínuas e com o fundo mais suave. O método ICBI produz uma imagem (Figura 6.46 (d)) com aspectos parecidos com o iNEDI; porém possui aspecto mais natural, principalmente na região central da flor.



(a) Localmente adaptativo

(b) NEDI



(c) iNEDI

(d) ICBI

Figura 6.46: (a) Interpolação pelo algoritmo localmente adaptativo; (b) Interpolação NEDI; (c) Interpolação iNEDI; (d) Interpolação ICBI.

Na Figura 6.47 (a), o método *wavelet* exibe resultados parecidos com a bicúbica. Na Figura 6.47 (b) está a aplicação do método que utiliza o filtro bilateral. A aparência da flor é parecida com a Figura 6.46 (c) (método iNEDI), porém expõe os detalhes do centro da flor com mais precisão.



Figura 6.47: (a) Interpolação utilizando *wavelets* redundantes; (b) Interpolação utilizando filtro bilateral.

De modo geral, os melhores métodos para a interpolação desta imagem são o NEDI, o iNEDI, ICBI e bilateral.

A Figura 6.48 (a) mostra a imagem com muito serrilhado nas bordas e nos detalhes. A Figura 6.48 (b), por sua vez, perde algumas informações dos pelos do babuíno por causa da suavização.

Tanto as aplicações dos métodos bicúbico (Figura 6.48 (c)), quanto das *wavelets* (Figura 6.49 (c)) apresentam resultados satisfatórios, pois a imagem possui muita textura e muitas curvas. O método *sinc* (Figura 6.48 (d)) também apresenta bom resultado visual, contudo, nas regiões homogêneas aparecem os problemas de periodicidade, o que pode comprometer a imagem. Observe o olho da Figura 6.48 (d), dentro dele é perceptível os efeitos indesejáveis causados pela interpolação *sinc*.

A Figura 6.48 (e) mostra a aplicação do algoritmo adaptativo. Este interfere nos detalhes do pelo. O mesmo ocorre pelo método NEDI, que também introduz ruídos dentro da região do olho do babuíno (Figura 6.48 (f)).



(a) Replicação

(b) Bilinear



(c) Bicúbica

(d) *Sinc*

(e) Localmente adaptativo

(f) NEDI

Figura 6.48: (a) Interpolação Replicação; (b) Interpolação Bilinear; (c) Interpolação Bicúbica; (d) Interpolação *Sinc*; (e) Interpolação pelo algoritmo localmente adaptativo; (f) Interpolação NEDI.

O método iNEDI também não consegue reamostrar os detalhes dos pelos com perfeição (Figura 6.49 (a)); contudo o contorno dos olhos são mais contínuos. Na Figura 6.49 (b), o ICBI produz uma imagem menos artificial. Apesar de contornos e bordas melhores e mais suaves, também introduz ruídos dentro do olho do babuíno. A interpolação que utiliza o filtro bilateral reduz os defeitos nas regiões mais homogêneas, porém também interfere nas regiões dos pelos, aonde estão concentrado os detalhes.



(a) iNEDI

(b) ICBI

(c) *Wavelets* redundantes

(d) Filtro bilateral

Figura 6.49: (a) Interpolação iNEDI; (b) Interpolação ICBI; (c) Interpolação utilizando *wavelets* redundantes; (d) Interpolação utilizando filtro bilateral.

De maneira geral os métodos que conseguiram interpolar a imagem e manter os aspectos e detalhes do babuíno foram os métodos ICBI, *wavelets* e bicúbica.

A ultima imagem a ser analisada é da borda de um copo de dimensão 128x128. O método replicação dada pela Figura 6.50 (a) degrada a borda pelo efeito *jaggie*. Na Figura 6.50 (b) a aplicação do bilinear diminui o serrilhado nas bordas entretanto “desfoca” a imagem em regiões de altas frequências. Na Figura 6.50 (c) o método bicúbico reduz o efeito *blurring*, contudo ainda apresenta o serrilhado nas bordas. A interpolação *sinc* (Figura 6.50 (d)) produz um resultado parecido com o bicúbico, mas adiciona ruídos nas regiões mais suaves.

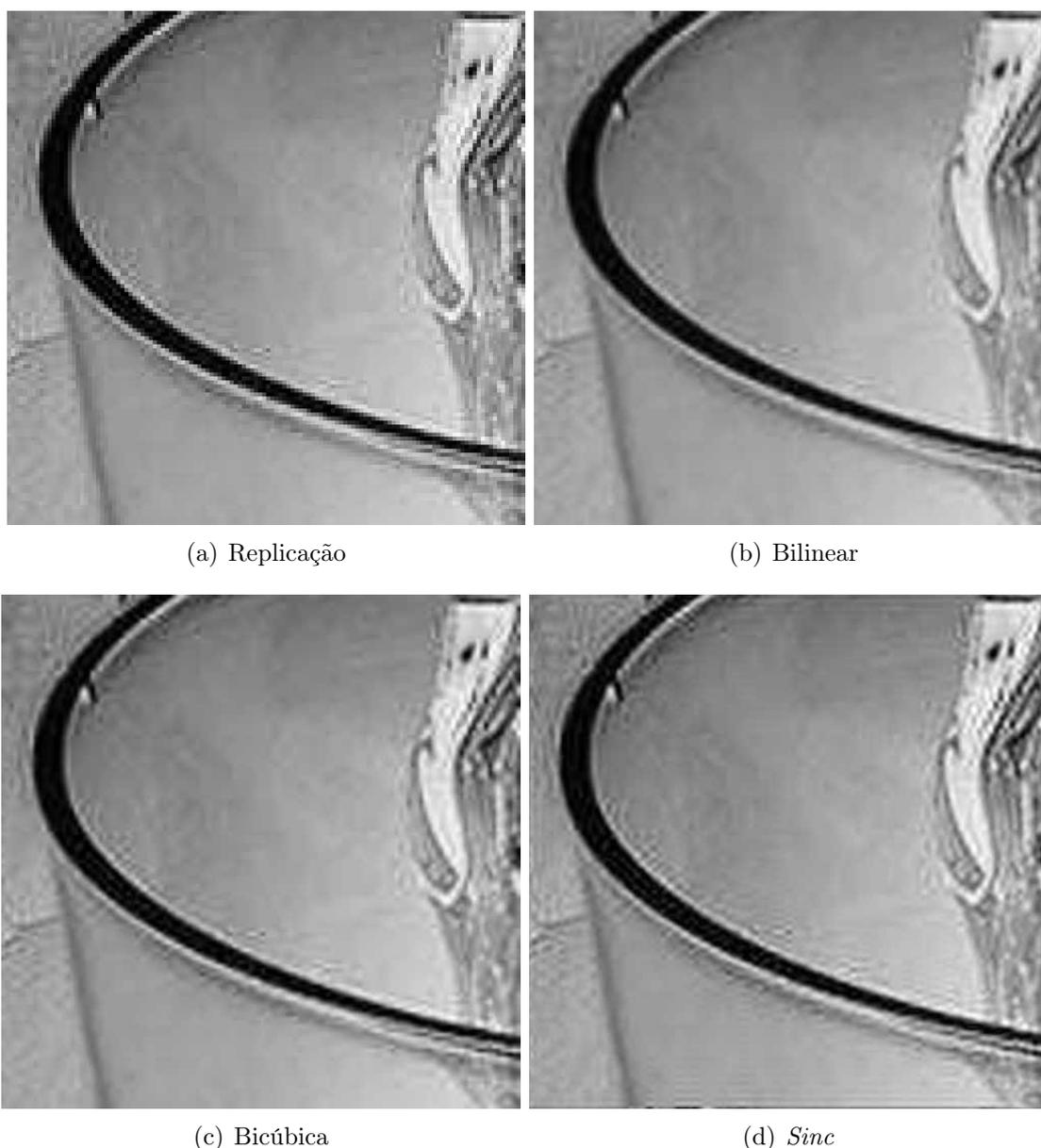


Figura 6.50: (a) Interpolação Replicação; (b) Interpolação Bilinear; (c) Interpolação Bicúbica; (d) Interpolação *Sinc*.

A interpolação localmente adaptativa está presente na Figura 6.51 (a) e imagem resultante deste método possui bordas com *jaggies* reduzidos e regiões homogêneas mais suavizadas. A desvantagem é a perda da naturalidade. A Figura 6.51 (b) mostra os re-

sultados da aplicação do NEDI. Esta exhibe bordas mais contínuas e regiões suaves mais uniforme. O problema está nos artefatos inseridos próximo das bordas do copo. O método iNEDI, Figura 6.51 (c), é parecido com o NEDI, distinguindo-se pelo fato de não degradar as bordas próximo às delimitações da imagem.

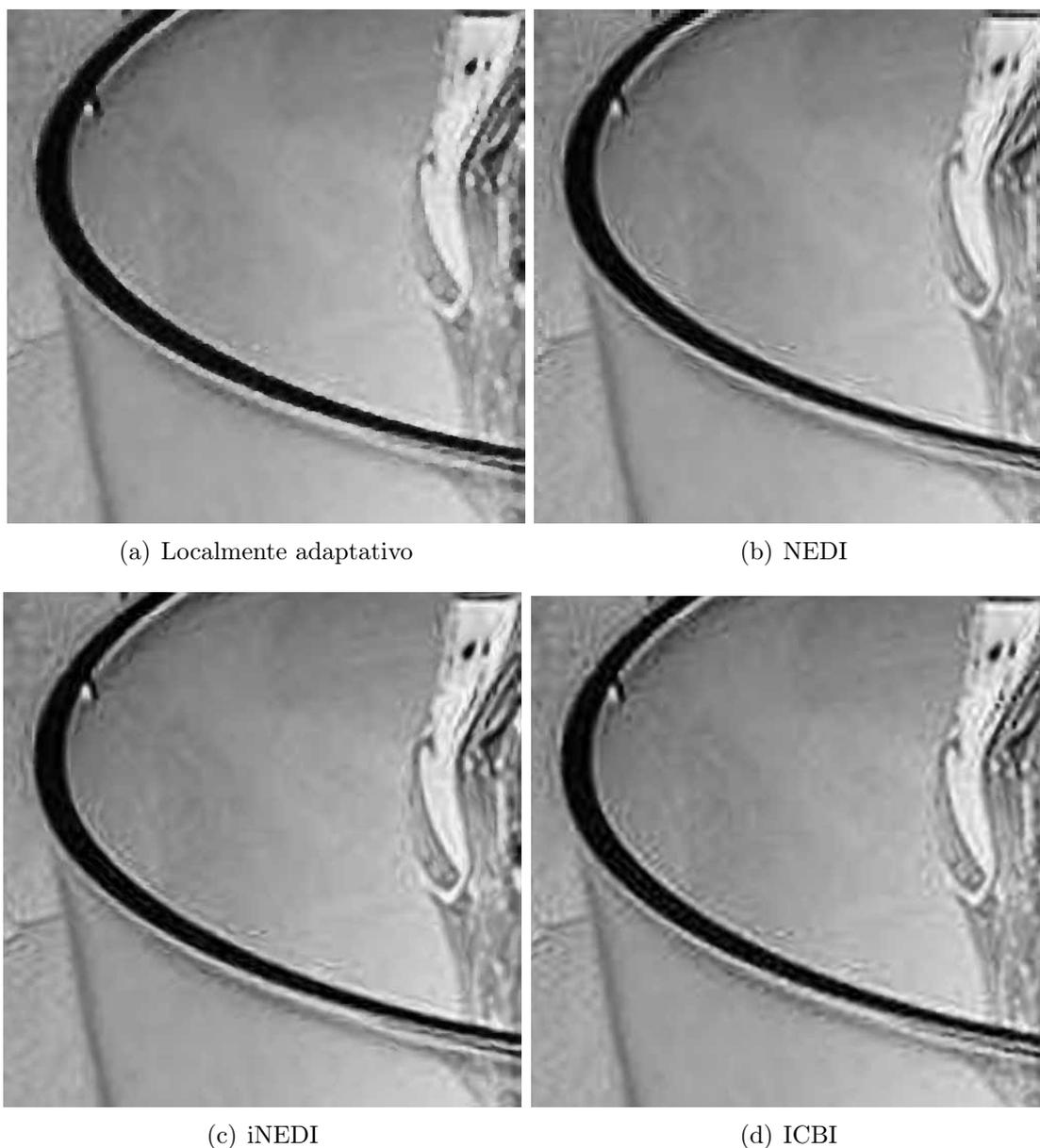


Figura 6.51: (a) Interpolação pelo algoritmo localmente adaptativo; (b) Interpolação NEDI; (c) Interpolação iNEDI; (d) Interpolação ICBI.

Nas Figuras 6.51(d) e Figura 6.52(a) os métodos ICBI e *wavelets* também produzem imagens semelhantes ao método NEDI e iNEDI, porém a borda do copo não é tão contínua. A Figura 6.52 (b) exhibe regiões homogêneas mais suavizadas. A área em volta da borda do copo apresenta menos ruído neste método.

(a) *Wavelets* redundantes

(b) Filtro bilateral

Figura 6.52: (a) Interpolação utilizando *wavelets* redundantes; (b) Interpolação utilizando filtro bilateral.

De maneira geral, esta imagem apresentou resultados semelhantes em relação a muitos métodos, podendo destacar: NEDI, iNEDI, ICBI, *Wavelet* e bilateral.

Os métodos que mais apresentaram bons resultados visuais diversificaram de imagem para imagem, pois cada uma possui características próprias. Apesar disto, os que exibiram como resultados visuais imagens com bordas mais contínuas e mais nítidas foram os métodos iNEDI, ICBI e bilateral.

Na Tabela 6.2 está um resumo das principais características de cada método.

Tabela 6.2: Vantagens e desvantagens gerais dos métodos..

Método	Vantagens	Desvantagens
Replicação	Rápido e simples computacionalmente	Introduz muito <i>jaggies</i>
Bilinear	Simple e diminui <i>jaggies</i>	Suaviza demais a imagem.
Bicúbica	Simple e diminui <i>jaggies</i> e <i>blurring</i>	Não interpola as bordas com boa continuidade.
<i>Sinc</i>	Interpola bordas com boa continuidade.	Introduz ruído periódico em torno das bordas.
Adaptativo	Rápido em relação aos métodos recentes	Perda de naturalidade na imagem.
NEDI	Bordas mais contínuas sem introduzir serrilhado	Degrada as regiões que delimitam a imagem.
iNEDI	Bordas mais contínuas; mais nitidez	Caro computacionalmente.
ICBI	Interpola sem tirar a naturalidade da imagem.	Não é tão rápida quanto os métodos clássicos.

Continua na próxima página.

Tabela 6.2 – continuação da página anterior

<b>Método</b>	<b>Vantagens</b>	<b>Desvantagens</b>
Wavelet	Não tem interferência de aspectos direcionais e tem boa interpolação das bordas de algumas imagens.	Em algumas imagens tem resultado semelhante ao bicúbico.
Bilateral	Não tem interferência de aspectos direcionais e tem boa interpolação	Suaviza a imagem; perda do realce.

## Considerações Finais

O trabalho desenvolvido explorou as vantagens e desvantagens dos métodos clássicos: replicação, bilinear, bicúbico e interpolação por função *sinc*, e dos métodos recentes: algoritmo localmente-adaptativo, NEDI, INEDI, ICBI, método com *wavelets* redundantes e método que utiliza o filtro bilateral.

No Capítulo 6, é encontrada a análise de cada método descrito no decorrer do trabalho. Os métodos mais usuais promovem interpolações com introdução de artefatos que degradam a imagem. Dos métodos mais recentes, o algoritmo adaptativo produz imagens que melhoram a continuidade das bordas. Contudo, os resultados apresentados, demonstram que este método interfere no aspecto geral, pois apesar de melhorar os contornos das bordas, prejudica a naturalidade das imagens, e assim apresenta uma aparência artificial. Sobre os métodos NEDI e iNEDI, ambos apresentam resultados parecidos. O método iNEDI apresentou, pelas avaliações numéricas, ser um dos melhores métodos estudados desde as análises estatísticas até as visuais. Todas as imagens utilizadas na comparação visual apresentaram bordas mais contínuas, detalhes finos mais preservados e regiões homogêneas com menos ruídos. Entretanto, em comparação aos outros métodos, é muito caro computacionalmente em relação ao seu tempo de processamento. Pela Tabela 6.1, o método ICBI apresentou bons resultados quantitativos em todas as métricas. Já pela comparação visual, produziu bordas contínuas, não superando apenas o método iNEDI, e também foi capaz de não afetar as características gerais da imagem, não exibindo assim, aparência artificial.

O método *wavelet* apresentou bons resultados quantitativos pela Tabela 6.1 em quase todos os casos, com exceção em relação à figura do olho da Lena, que apresentou a pior performance. No aspecto visual apresentou bons resultados nas imagens do babuíno e na borda do copo. Nas outras imagens apresentou desempenho parecido com a interpolação bicúbica. Por fim, a interpolação que utiliza o filtro bilateral trouxe resultados com

preservação de detalhes em geral. Na comparação visual, apenas na imagem do babuíno apresenta deformações, pois o pelo possui textura que é suavizada pelo filtro.

Em geral os melhores são iNEDI, ICBI e bilateral. Contudo, cada método possui suas próprias características e vale evidenciar que, a escolha do melhor dependerá dos detalhes da imagem utilizada. Por exemplo, imagens com muitas texturas é recomendável utilizar métodos que não a suavizam, como o bilateral ou bilinear. Já as que possuem muitas bordas com direções contínuas, são melhores os métodos NEDI, iNEDI e ICBI. Estas técnicas conseguem manter a continuidade de uma borda sem introdução de *jaggies*. Contudo, se a imagem tiver muitas curvas, como é o caso do babuíno, é preferível utilizar métodos que não dependam das direções das bordas, como por exemplo, a interpolação pela transformada *wavelets*, ou pelo bilateral. Este último é necessário um pequeno cuidado, pois se a imagem tiver muitos detalhes, pode ocorrer uma degradação pela característica suavizante do filtro. Imagens que possuem bordas e tem poucas regiões suaves, uma opção é utilizar a interpolação por função *sinc*. Aquelas que possuem predominância de região homogênea pode ser utilizado quase todos os métodos, pois os maiores problemas ocorrem nas bordas. Contudo, uma boa opção seria utilizar a interpolação que utiliza o filtro bilateral, pois este filtra os ruídos brancos. Por fim, se a intenção é preservar a naturalidade, a melhor opção é o ICBI. O pequeno cuidado que deve-se tomar é que este método amplia quaisquer informações de altas frequências, como os ruídos.

A proposta do trabalho é comparar alguns métodos de interpolação de imagens recentes com os clássicos usualmente utilizados. Uma sequência natural é aprimorar ou mesmo combinar os métodos avaliados. Com este intuito, para trabalhos futuros, uma proposta está em aperfeiçoar o método de interpolação que utiliza a transformada *wavelet*. Uma modificação possível seria no tipo de estimativa de pixels centrais antes da aplicação da transformada *wavelet*. O método utiliza a interpolação localmente adaptativa para este cálculo. Outra modificação estaria no cálculo das variâncias utilizadas após a rotação da imagem em  $45^\circ$ . O método usado para interpolar estes pontos é baseado no trabalho de Ward que propõe tanto a rotação quanto a estimativa. Uma sugestão que pode melhorar o algoritmo é utilizar uma nova maneira de estimar estes pixels baseado nas covariâncias entre seus vizinhos, conforme é realizado nos métodos NEDI e iNEDI.

O grande problema que envolve a interpolação de imagens está no cálculo de pixels que estão próximos de bordas ou nelas inseridas. A má interpolação resulta em imagens que apresentam bordas serrilhadas ou muito suavizadas. O método que utiliza o filtro bilateral apresenta resultados com bordas sem muito *jaggies*, contudo, apesar de não modificar a textura da imagem, causa suavização em lugares homogêneos e bordas diagonais não contínuas. Uma proposta para sofisticar o algoritmo da interpolação pelo filtro bilateral, é aplicar o método apenas em regiões de bordas. Em regiões suaves, utilizar outros métodos mais simples computacionalmente, como o bicúbico. Esta alteração pode causar melhorias na suavização das regiões próximas de bordas. Outra proposta é associar o

---

bilateral com as técnicas que utilizam bordas direcionais com a finalidade de amenizar a descontinuidade causada pelo método original em algumas imagens.

Todas as sugestões são ideias que podem futuramente originar novos algoritmos que aperfeiçoem os softwares de *zoom* nos seus diferentes tipos de usos. Em edições de imagens para entretenimento, por exemplo, é necessário expandir o avanço em algoritmos que forneçam resultados melhores que o bicúbico e em tempo real. Para imagens médicas e forenses, quanto mais sofisticado o método, melhor será identificar características observadas na imagem. Elaborar novas técnicas que amplie a imagem com bordas mais contínuas e sem a introdução de artefatos, com um custo computacional baixo é uma finalidade que facilitará as novas tecnologias que dependem da imagem digital.

O estudo de novas métricas estatísticas, que calculem com mais exatidão as diferenças entre duas imagens e que levem em consideração a percepção da visão humana, também é primordial. Quanto mais completa for a avaliação da qualidade entre imagens, melhor será a precisão das melhorias, ou distorções, causadas por diferentes métodos de interpolação. O desenvolvimento de novas métricas, ou o aperfeiçoamento das existentes, requer pesquisas mais aprofundadas, sem desconsiderar os fatores que envolvem a forma de interpretar a imagem visualizada pelo sistema visual humano. Métricas que reflitam com mais exatidão a forma do qual o homem enxerga as características das imagens são essenciais para a comparação e escolha de métodos mais acurados.

---

## Referências Bibliográficas

- [1] ALLEBACH, J.; WONG, P. Edge-directed interpolation. *International Conference on Image Processing vol. 3*, n° 6 (September 1996), pages 707–710.
- [2] BATTIATO, S.; GALLO, G., AND STANCO, F. A locally-adaptive zooming algorithm for digital images. *Image and Vision Computing vol. 20* (September 2002), pages 805–812. Dipartimento di Matematica ed Informatica - Università di Catania.
- [3] BREZIS, H. *Functional analysis, sobolev spaces and partial differential equations.*, 1 ed. Springer Science, Piscataway, 2010.
- [4] CAMARGO, A. C. Reamostragem uniforme utilizando a função *sinc*. pg. 12 - 13. dissertação de mestrado, Instituto de Matemática, Estatística e Computação Científica, UNICAMP, 2006.
- [5] CASTILHO, J. EDUARDO; DOMINGUES, M. O., PAGAMISSE, A., AND JUNIOR, O. M. *Introdução ao mundo das wavelets: Notas em Matemática Aplicada.*, vol. vol. 62. Sociedade Brasileira de Matemática Aplicada e Computacional., São Carlos, 2012.
- [6] CHUI, C. K. *An introduction to wavelets: Wavelet analysis and its applications.*, 1 ed. Academic Press, Texas, 1992.
- [7] DUMIC, E.; GRGIC, S., AND GRGIC, M. The use of wavelets in image interpolation: possibilities and limitations. *Radioengineering vol. 16*, n° 4 (December 2007), pages 101–109. University of Zagreb, Department of Wireless Communications, Zagreb, Croatia.
- [8] FILHO, O. MARQUES; NETO, H. V. *Processamento digital de imagens.*, 1 ed. Brasport, Rio de Janeiro, 1999.
- [9] GIACHETTI, A., AND ASUNI, N. Real time artifact-free image upscaling. *IEEE Transactions on Image Processing vol. 20* (October 2011), pages 2760–2768. Disponível em: <<http://www.andreagiachetti.it/icbi/>>. Acesso em 31 Outubro de 2013.

- 
- [10] GIACHETTI, A.; ASUNI, N. Accuracy improvements and artifacts removal in edge based image interpolation. *Proc. 3rd Intl. Conf. on Computer Vision Theory and Applications (VISAPP) vol. 1* (2008), pages 58–65.
- [11] GIACHETTI, A.; ASUNI, N. Fast artifact free image interpolation from a single image. *In Proc. BMVC* (2008), pages 123 – 132. Disponível em: <<http://inedi.tecnick.com/>>. Acesso em 31 Outubro de 2013.
- [12] GLASSNER, A. S. *Principles of digital image synthesis.*, 1 ed. Morgan Kaufmann Publishers Inc., San Francisco, 1994.
- [13] GONZALEZ, RAFAEL C.; WOOD, R. E. *Processamento digital de imagens. Tradução: Cristina Yamagami e Leonardo Piamonte.*, 3 ed. Pearson Prentice Hall, São Paulo, 2010.
- [14] HAN, J. WOO; HAN, J. W. K. J. H. A novel image interpolation method using the bilateral filter. *IEEE Consumer Electronics Society vol. 56* (February 2010), pages 175 – 181. IEEE Members.
- [15] HOU, H.S.; ANDREWS, H. Cubic splines for image interpolation and digital filtering. *IEEE Transactions on Acoust., Speech, Signal Processing vol. 26*, nº 6 (December 1978), pages 508–517.
- [16] HUNG, K.; SIU, W. Fast image interpolation using bilateral filter. *IET Image Processing vol. 6*, nº 7 (October 2012), pages 877–890. Disponível em: <[http://www.eie.polyu.edu.hk/~wcsiu/softmodule/3/robust\\_SAI\\_hung\\_n\\_siu\\_2.html](http://www.eie.polyu.edu.hk/~wcsiu/softmodule/3/robust_SAI_hung_n_siu_2.html)>. Acesso em 13 de Março de 2014.
- [17] HWANG, J.W.; LEE, H. Adaptive image interpolation based on local gradient features. *IEEE Signal Process. Lett. vol. 11*, nº3 (March 2004), pages 359–362.
- [18] JIANG, H.; MOLONEY, C. A new direction adaptative scheme for image interpolation. *International Conference on Image Processing vol. 3* (2002), pages 369 – 372. University of Newfoundland St. John’s, Newfoundland, Canada.
- [19] KREYSZIG, E. *Introductory functional analysis with applications.*, 1 ed. John Wiley and Sons. Inc., London, 1989.
- [20] KUMAR, M. An adaptive zooming algorithm for images. Master thesis, Thapar University, Patiala, July 2009. Thesis of Engineering in Computer Science and Engineering.

- 
- [21] LEE, S. W.; PAIK, J. K. Image interpolation using adaptive fast b-spline filtering. *in Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing vol. 5* (April 1993), vol. 177–180.
- [22] LI, X.; ORCHARD, M. New edge-directed interpolation. *IEEE Trans. on Image Processing vol. 10*, n° 10 (October 2001), pages 1521–1527. Disponível em: <<http://www.csee.wvu.edu/~xinl/source.html>>. Acesso em 31 Outubro de 2013.
- [23] MALLAT, S.; ZHONG, S. Characterization of signals from multiscale edges. *IEEE Transactions on Pattern Analyzis and machine intelligence vol. 14*, n° 7 (July 1992), pages 710 – 732.
- [24] MORSE, B.S.; SCHWARTZWALD, D. Image magnification using level-set reconstruction. *In Proc. IEEE Computer Vision and Pattern Recognition vol. 1* (February 2001), pages 333–340.
- [25] NEWS, E. *Koffka Rings*. Disponível em: <<http://sc-optom.com/2012/10/koffka-rings/>>. Acesso em 12/03/2014.
- [26] PAGAMISSE, A.; FONSECA, L. M. G. Método adaptativo de super-resolução de frame único com wavelets redundantes. *In. Anais do VI Workshop de Visão Computacional* (Julho 2010), pg. 230 – 235. Presidente Prudente. Faculdade de Ciências e Tecnologia, Unesp.
- [27] PARKER, J. A.; KENYON, R. V., AND TROXEL, D. E. Comparison of interpolating methods for image resampling. *IEEE Transactions on Medical Imaging vol. 2*, n° 1 (March 1983), pages 31–39.
- [28] PEDRINI, H.; SCHWARTZ, W. R. *Análise de imagens digitais: Princípios, Algoritmo e Aplicações*. Thomson Learning, São Paulo, 2008.
- [29] SPÄTH, H. *Two dimensional spline interpolation algorithms. Translate to English*. A. K. Peters, Massachusetts, 1993.
- [30] SU, DAN; WILLIS, P. Image interpolation by pixel-level data-dependent triangulation. *Computer Graphics Forum vol. 23* (July 2004), pages 189–201.
- [31] TAM, WINGSHAN; KOK, C., AND SIU, W. Modified edge-directed interpolation for images. *J. Electron. Imaging. vol. 19*, n° .13011 (March 2010), pages 1 – 20.
- [32] TAM, W. S.; KOK, C. W., AND SIU, W. A modified edge directed interpolation for images. *17th European Signal Processing Conference (EUSIPCO 2009). Glasgow, Scotland* (August 2009), pages 283–287.

- 
- [33] TOMASI, C. Bilateral filtering for gray and color images. *IEEE International Conference on Computer Vision, Bombay, India* (January 1998), pages 839 – 846.
- [34] UNSER, M.; SWISS, F., AND LAUSANNE, S. Sampling-50 years after shannon. *Proceedings of the IEEE vol. 88, n° 4* (April 2000), pages 569 – 587. Dipartimento di Matematica ed Informatica - Università di Catania.
- [35] UNSER, M. Convolution-based interpolation for fast, high-quality rotation of images. *IEEE Transactions on image processing. vol. 4* (October 1995), pages 1371–1381. Dipartimento di Matematica ed Informatica - Università di Catania.
- [36] WANG, Z. *Modern image quality assessment.*, 1 ed. Morgan and Claypool, Texas, 2006.
- [37] WARD, D. L. *Redundant discrete Wavelet transform based super-resolution using sub-pixel image registration.* Phd thesis, Air Force Institute of Technology, Ohio, March 2003.
- [38] WOLFCROW. Professor sampler's notes: the shannon-nyquist sampling theorem part iii. Disponível em: <<http://wolfcrow.com/blog/professor-samplers-notes-the-shannon-nyquist-sampling-theorem-part-iii/>>. Acesso em 13/03/2014.
- [39] YAROSLAVSKY, L. P. Efficient algorithm for discrete *sinc* interpolation. *Applied Optics vol. 36* (January 1997), pages 460 – 463. Institute d'Optique, Centre National de la Recherche Scientifique.
- [40] ZHU, Y.; SCHWARTZ, S. C., AND ORCHARD, M. T. Wavelet domain image interpolation via statistical estimation. *IEEE International Conference Image Proceedings vol. 3* (Oct 2001), pages 840 – 843.

# Algoritmos

Na sequência são apresentados diferentes implementações de alguns dos métodos. Vale ressaltar que foram implementados apenas àqueles, cujos originais não foram disponibilizados.

## A.1 Interpolação por convolução, função *sinc*.

Segue abaixo o algoritmo da Técnica de “Zero Padding”:

```
function zeropadding(A,l,c)

[m n]=size(A); % Tamanho da Imagem Original
B=zeros(l,c); % Tamanho da Nova Imagem
A=fft2(A); % Aplicação da Trans. de Fourier
A=fftshift(A);
%-----%
B(round(l/2-m/2):round(l/2-m/2)+m-1,round(c/2-n/2):round(c/2-n/2)+n-1)=A(1:m,1:n);
% redimensiona o espectro da imagem Original
% ao centro da Matriz de zeros B (Processo de
% zero Padding no Espectro)
%-----%
B=fftshift(B);
B=ifft2(B); % Transformação Inversa de Fourier
%-----%
k=ceil(l*c/(m*n)); % Constante que aumenta o brilho da Imagem
figure,imshow(uint8(k*abs((B)))); title('Zero Padding')
% Visualização da Imagem
```

## A.2 Interpolação Localmente adaptativa

Segue abaixo o algoritmo de interpolação Localmente Adaptativo para imagens em escala de cinza.

```
% INTERPOLAÇÃO UTILIZANDO A TESE DO MAYANK KUMAR
% OS TERMOS "K" COMO SENDO O PARÂMETRO FORNECIDO
% NA ENTRADA DO ALGORITMO.
% "A" É A IMAGEM QUE SERÁ INTERPOLADA
%-----%
% A MATRIZ AUX É UMA MATRIZ AUXILIAR QUE SERÁ UTILIZADA
% PELO ALGORITMO 'contab(AUX,K)' NA CONTAGEM AO FINAL.
%-----%
% NO ESTÁGIO 2 E 3 SÃO APLICADOS O ALGORITMO "estagios2".
% ESTE ALGORITMO TEM COMO OBJETIVO ENCONTRAR ESTUDAR A
% REGIÃO EM VOLTA DO PIXELS INTERPOLADO À PROCURA DE BORDAS
%-----%
% DEPOIS DE TODOS OS ESTÁGIO É APLICADO A FUNCTION "CONT"
% ESTA FUNCTION CONTA QUANTOS PIXEL FORAM UTILIZADOS EM CADA ESTÁGIO
% DESTA MANEIRA DÁ PARA TER UMA ESTIMATIVA NUMÉRICA SOBRE OS ESTÁGIOS
%-----%
function y=kumar(A,K,Wt)
% y=kumar(a,50,90); % para Lena
% y=kumar(a,50,90); % para Lena
%-----%
tic
%-----% ESTÁGIO 1
A=double(A);
A=bordo(A,1);
tam=2*size(A);
y=zeros(tam-1);
z=zeros(tam-1);
i=1:2:tam(1);
j=1:2:tam(2);
m=1:tam(1)/2;
n=1:tam(2)/2;
y(i,j)=A(m,n); % EXPANDE A IMAGEM
z(i,j)=1;
%-----% Cantos da imagem
```

```

for j=2:2:tam(2)-1
    y(1,j)=(A(1,j/2)+A(1,j/2+1))/2;
    y(tam(1)-1,j)=(A(tam(1)/2,j/2)+A(tam(1)/2,j/2+1))/2;
    z(1,j)=0.5;
    z(tam(1)-1,j)=0.5;
end
for i=2:2:tam(1)-1
    y(i,1)=(A(i/2,1)+A(i/2+1,1))/2;
    z(i,1)=0.5;
    y(i,tam(2)-1)=(A(i/2,tam(2)/2)+A(i/2+1,tam(2)/2))/2;
    z(i,tam(2)-1)=0.5;
end
%-----% ESTÁGIO 2
for i=2:2:tam(1)-2
    for j=2:2:tam(2)-2
        y(i,j)=estagio2(y(i-1:i+1,j-1:j+1),K,0,Wt);
        z(i,j)=0.5;
    end
end
%-----% ESTÁGIO 3
for i=3:2:tam(1)-2
    for j=2:2:tam(2)-2
        y(i,j)=estagio2(y(i-1:i+1,j-1:j+1),K,1,Wt);
        z(i,j)=0.75;
    end
end
% % -----%
for i=2:2:tam(1)-2
    for j=3:2:tam(2)-2
        y(i,j)=estagio2(y(i-1:i+1,j-1:j+1),K,2,Wt);
        z(i,j)=0.75;
    end
end
%-----%
clc
toc
%-----%
disp('-----')
contab(z,K)

```

```

disp('-----')
% -----%
figure, imshow(uint8(y)),title(sprintf('Kumar, K= %d e Wt= %d',K,Wt))
end

```

Para funcionar, este algoritmo utiliza a *function* estagio2.

```

% PARTE DO ALGORITMO DE KUMAR: function "estagio2"
function M=estagio2(A,K,z,Wt)

M=A(2,2);

if z==0
    P=A(1,1); Q=A(1,3); R=A(3,1); S=A(3,3);
elseif z==1
    P=A(1,2); R=A(2,1); Q=A(2,3); S=A(3,2);
elseif z==2
    P=A(2,1); Q=A(1,2); R=A(3,2); S=A(2,3);
end

%-----% PESOS DOS PIXELS ADJACENTES
Z1= abs(P-S); Z2=abs(Q-R); Z3= abs(P-Q);
Z4= abs(R-S); Z5= abs(P-R); Z6= abs(Q-S);
X=(P+Q+R+S)/4;
ZZ=(abs(X-P)+abs(X-Q)+abs(X-R)+abs(X-S))/4;
% ----- %

if ZZ <= K % CONDITION 1
    M=(P+Q+R+S)/4;
    %-----%
elseif (Z1<=min([Z3,Z4,Z5,Z6]) || Z2 <=min([Z3,Z4,Z5,Z6])) % CONDITION 2
    if abs(Z1-Z2)<K/2;
        M=(P*abs(Wt-P)+Q*abs(Wt-Q))/(abs(Wt-P)+abs(Wt-Q));
    elseif Z1<Z2 % CONDITION 3
        Q1=abs(Q-P)+abs(Q-S);
        R1=abs(R-P)+abs(R-S);
        if Q1<R1
            M=(P*abs(Wt-P)+S*abs(Wt-S)+Q*abs(Wt-Q))/(abs(Wt-P)+abs(Wt-S)+abs(Wt-Q));

```

---

```

    else
        M=(P*abs(Wt-P)+S*abs(Wt-S)+R*abs(Wt-R))/(abs(Wt-P)+abs(Wt-S)+abs(Wt-R));
    end
elseif Z2<Z1 % CONDITION 4
    P1=abs(P-Q)+abs(P-R);
    S1=abs(S-Q)+abs(S-R);
    if P1<S1
        M=(P*abs(Wt-P)+R*abs(Wt-R)+Q*abs(Wt-Q))/(abs(Wt-P)+abs(Wt-R)+abs(Wt-Q));
    else
        M=(Q*abs(Wt-Q)+S*abs(Wt-S)+R*abs(Wt-R))/(abs(Wt-Q)+abs(Wt-S)+abs(Wt-R));
    end
end
%-----%
elseif (Z3<=min([Z1,Z2,Z5,Z6]) || Z4 <=min([Z1,Z2,Z5,Z6])) % CONDITION 5
    if abs(Z3-Z4)<K/2
        M=(P*abs(Wt-P)+R*abs(Wt-R))/(abs(Wt-R)+abs(Wt-P));
    elseif Z3<Z4 % CONDITION 6
        D1=(R+S)/2;
        M=(P*abs(Wt-P)+Q*abs(Wt-Q)+D1*abs(Wt-D1))/(abs(Wt-P)+abs(Wt-Q)+abs(Wt-D1));
    elseif Z4<Z3 % CONDITION 7
        D1=(P+Q)/2;
        M=(R*abs(Wt-R)+S*abs(Wt-S)+D1*abs(Wt-D1))/(abs(Wt-R)+abs(Wt-S)+abs(Wt-D1));
    end
    %-----%
elseif (Z5<=min([Z1,Z2,Z3,Z4]) || Z6<=min([Z1,Z2,Z3,Z4]))% CONDITION 8
    if abs(Z5-Z6)<K/2
        M=(P*abs(Wt-P)+Q*abs(Wt-Q))/(abs(Wt-P)+abs(Wt-Q));
    elseif Z5<Z6 % CONDITION 9
        D1=(Q+S)/2;
        M=(P*abs(Wt-P)+R*abs(Wt-R)+D1*abs(Wt-D1))/(abs(Wt-P)+abs(Wt-R)+abs(Wt-D1));
    elseif Z6<Z5 % CONDITION 10
        D1=(P+R)/2;
        M=(Q*abs(Wt-Q)+S*abs(Wt-S)+D1*abs(Wt-D1))/(abs(Wt-Q)+abs(Wt-S)+abs(Wt-D1));
    end
end
end

```

---

Abaixo a *function* contab:

```
function contab(a,K)
format short
t=0;
s=0;
v=0;
ZEROS=0;
[m n]=size(a);

for i=1:m
    for j=1:n
        if a(i,j)==1
            t=t+1;
        elseif a(i,j)==0.5
            s=s+1;
        elseif a(i,j)==0.75
            v=v+1;
        else
            ZEROS=ZEROS+1;
        end
    end
end
end
disp(sprintf('A quantidade de pixels para K=%.2f,',K))
disp(sprintf('Estágio 1: %d',t));
disp(sprintf('Estágio 2: %d',s));
disp(sprintf('Estágio 3: %d',v));
disp(sprintf('Ainda não utilizados: %d',ZEROS));
```