

DEDICATÓRIA

À minha família pela oportunidade de chegar até aqui com todo apoio e torcida.

Aos amigos, pela presença constante durante toda a jornada que termina nesse momento.

AGRADECIMENTOS

Inicialmente, agradeço ao Centro Universitário de Brasília e ao coordenador do curso Abiézer Fernandes pela disponibilidade e atenção.

Ao professor Francisco Javier pela confiança e pelo interesse, transmitindo segurança em uma orientação sempre eficiente.

Aos professores Luís Cláudio, José Julimá e Thiago Toribio pela boa vontade e pelo auxílio sempre que solicitados, durante todo o período de curso.

Ao Dr. Pedro Renato de Paula Brandão – Neurologia/USP-Ribeirão Preto, pela solicitude e pela cortesia dos dados usados neste trabalho.

A toda minha família pelo apoio incondicional, em especial aos meus pais Jeanne e Leo com torcida e presença constante; e à minha irmã Lilian com preocupação, ajuda e companheirismo.

A meus amigos recentes e de longa data pelo carinho, compreensão nas horas de ausência e incentivo em todos os momentos.

A meus amigos de faculdade pela companhia diária e pelo excelente trabalho em equipe realizado durante os últimos anos.

A meus amigos de trabalho que tornaram meu início de carreira o mais agradável e saudoso possível e que, de alguma forma, deram sua contribuição.

À Damares Monte pela inicial sugestão do tema, que despertou o interesse e o compromisso no desenvolvimento deste trabalho.

A todos que contribuíram, direta ou indiretamente, para a conclusão desta jornada, tanto em momentos de dificuldade e concentração quanto em momentos de lazer.

Muito Obrigada.

EPÍGRAFE

Nenhum trabalho de qualidade pode ser feito sem concentração e auto sacrifício, esforço e dúvida.

Max Beerbohm

RESUMO

Processamento de imagens é uma vasta área em foco de muitos pesquisadores há algumas décadas. Uma das aplicações significativas se dá no imageamento médico e, entre todos os tipos de processamento de imagens médicas digitais, as tarefas mais comuns e muitas vezes necessárias são a filtragem de ruídos e a compressão de imagens. Ruídos são comuns em imagens médicas digitais, devido à possível corrupção durante a transmissão ou mesmo a aquisição destas. Apesar do aspecto em geral indesejável, o fator mais importante é que o ruído pode cobrir e reduzir a visibilidade de determinadas características importantes da imagem. O objetivo da restauração é trazer a imagem original de volta, tanto quanto possível, a partir de seu estado de degradação. A Tomografia Computadorizada (CT) ou a Ressonância Magnética (MR), presentes no imageamento médico, produzem imagens internas do corpo humano no formato digital. Uma vez que estas técnicas geram imagens que requerem grande espaço em disco, a compressão é necessária para fins de armazenamento e transmissão. Muitos métodos atuais oferecem uma taxa de compressão alta, mas com considerável perda de qualidade, o que não é interessante no caso de imagens de diagnósticos. Vários fatores são relevantes para a redução da quantidade de bits de informação, a fim de manter uma qualidade de imagem aceitável. Este projeto visa descrever e analisar a codificação sem perdas, com uso dos códigos de Huffman e códigos por Comprimento Corrido, além da comparação entre vários tipos de filtragem para a restauração da imagem degradada, relacionando-as com a qualidade da imagem após os processos de decodificação ou restauração.

Palavras Chave: Degradação de imagem, compressão sem perdas, filtro de Wiener, ruído gaussiano, ruído sal e pimenta, borrão de movimento, filtros espaciais, filtro inverso, codificação de Huffman, codificação por comprimento corrido, qualidade da imagem, CT, MRI

ABSTRACT

Image processing is a vast area that has been on focus of many researchers for several decades. One of the most significant implementation is in medical imaging and, among all types of digital medical image processing, the most popular and often necessary tasks are the noise filtering and the image compression. Noises normally exists in medical digital images, due to possible corruption during their acquisition or even their transmission. Although noise gives an image a generally undesirable appearance, the most significant factor is that noise can cover and reduce the visibility of certain features within the image. The purpose of image restoration is to bring back as much as possible the original image from its degraded state. Computed Tomography (CT) or Magnetic Resonance (MR) exams, featured in medical imaging, produce human body pictures in digital form. Since these imaging techniques generate huge amounts of data, compression is required for storage and transmission. Many current schemes provide a high compression rate, but with considerable loss of quality, which is not interesting in case of diagnostic images. Several factors are relevant to reducing the amount of information bits, in order to maintain an acceptable image quality. This project aims to describe and analyze the lossless image coding by Huffman and Run Length codes, in addition to the comparison of various types of filtering for restoration of degraded image, relating them to image quality after decoding or restoration processes.

Key words: Image degradation, lossless compression, Wiener filter, gaussian noise, salt and pepper noise, motion blur, spacial filters, inverse filter, Huffman encoding, run length encoding, image quality, CT, MRI

SUMÁRIO

LISTA DE FIGURAS	xi
LISTA DE SÍMBOLOS E ABREVIATURAS	xiii
LISTA DE TABELAS	xiv
CAPÍTULO 1 - INTRODUÇÃO	15
1.1 – Motivação e Apresentação do problema	15
1.2 – Objetivos do Trabalho	17
1.3 – Escopo do Trabalho	18
1.4 – Estrutura da Monografia.....	18
CAPÍTULO 2 – CONCEITOS BÁSICOS	20
2.1 – Histórico do Processamento de Imagens e Imagens Médicas Digitais	20
2.2 – Fundamentos da Imagem Digital Monocromática	21
2.2.1 – Topologia da imagem	23
2.2.2 – Histograma	24
2.3 – Armazenamento e Transmissão.....	25
CAPÍTULO 3 – RESTAURAÇÃO DE IMAGENS DIGITAIS	27
3.1 – Modelos e Tipos de Degradação	28
3.1.1 – Ruído gaussiano.....	28
3.1.2 – Ruído impulsivo	29
3.1.3 – Borrão de movimento	30
3.2 – Técnicas de Restauração de Imagens Degradadas	31
3.2.1 – Restauração apenas com ruído (filtros espaciais).....	32
3.2.2 – Restauração com função de degradação (filtro inverso e filtro de Wiener)	35
3.3 – Medidas de Qualidade e Critérios de Fidelidade de Imagens	37
3.3.1 – Erro quadrático médio	38
3.3.2 – Relação sinal-ruído	38
3.3.3 – Relação sinal-ruído de pico	39
3.3.4 – Índice universal de qualidade da imagem.....	39
CAPÍTULO 4 – COMPRESSÃO DE IMAGENS DIGITAIS	41
4.1 – Fundamentos da Compressão de Imagens.....	41
4.2 – Modelo Genérico de Compressão de Imagens	43
4.3 – Classificação para Compressão de Imagens	44
4.4 – Taxas de Compressão	46
4.5 – Compressão sem Perdas ou Compactação da Imagem.....	47

4.5.1 – Teoria da informação	48
4.6 – Codificação e Compactação de Imagens	51
4.6.1 – Código de Huffman	51
4.6.2 – Código por comprimento corrido	54
CAPÍTULO 5 – MODELO PROPOSTO	57
5.1 – Apresentação Geral.....	57
5.2 – Implementação.....	58
CAPÍTULO 6 – APLICAÇÃO DO MODELO PROPOSTO.....	67
6.1 – Resultados Obtidos	67
6.2 – Análise de Resultados.....	84
CAPÍTULO 7 - CONCLUSÃO	86
7.1 – Conclusão	86
7.2 – Sugestões para Trabalhos Futuros	87
ANEXO A – Código do Índice Universal de Qualidade da Imagem.	93
ANEXO B – Código para Decodificação de Huffman.	96
APÊNDICE A – Código do Modelo Proposto.....	99

LISTA DE FIGURAS

Figura 2.1 – Etapas de digitalização da imagem	22
Figura 2.2 – Matriz $M \times N$ da imagem.....	22
Figura 2.3 – Conceitos de 4-vizinhança, vizinhança diagonal e 8-vizinhança.....	24
Figura 2.4 – Histograma de imagem e escala de cinza.....	24
Figura 2.5 – Formato de Histograma x Contraste da imagem.....	25
Figura 3.1 – Modelo de degradação e restauração da imagem digital	27
Figura 3.2 – Ruído gaussiano e PDF correspondente.....	29
Figura 3.3 – Ruído impulsivo (<i>Salt and Pepper</i>) e PDF correspondente	30
Figura 3.4 – Imagem genérica com borrão de movimento.....	31
Figura 3.5 – Histogramas dos modelos de ruídos.....	31
Figura 3.6 – Degradação de imagem apenas com ruído.....	32
Figura 3.7 – Operação do filtro da mediana	34
Figura 3.8 – Comparativo entre Filtragem Inversa com ruído e Filtragem Wiener	36
Figura 3.9 – Imagens: Original, com borrão de movimento e pós filtragem-inversa.....	37
Figura 4.1 – Codificador (a) e decodificador (b) da fonte.....	43
Figura 4.2 – Taxionomia dos métodos de compressão.....	46
Figura 4.3 – Construção das palavras código do Algoritmo de Huffman	52
Figura 4.4 – Imagem Binária (a) RLE por linhas (b)	55
Figura 4.5 – Variantes RLE: (a) linhas, (b) colunas, (c) quadros e (d) zigzag.....	56
Figura 5.1 – Fluxograma do modelo proposto	58
Figura 5.2 – Tela inicial da GUI.....	60
Figura 5.3 – Tela do Simulador de Degradação (Simulador.fig)	61
Figura 5.4 – Tela de comparação dos filtros espaciais (GUICompare.fig)	62
Figura 5.5 – Tela do filtro inverso (GUIRestaura.fig).....	63
Figura 5.6 – Tela do filtro de Wiener (GUIRestaura.fig).....	63
Figura 5.7 – Tela de Codificação de Huffman (GUIHuffman.fig).....	64
Figura 5.8 – Tela de RLE (GUIRle.fig).....	65
Figura 5.9 – Tela de Medição da Qualidade da imagem (GUIAnalisar.fig)	66
Figura 5.10 – ImagemOriginal, (a)Ruído Gaussiano, (b) Ruído Sal e Pimenta (c) Ruído Sal e Pimenta + Borrão de Movimento e (d) Borrão de movimento.....	67
Figura 5.11 – Histogramas: ImagemOriginal, (a)Ruído Gaussiano, (b) Ruído Sal e Pimenta (c) Ruído Sal e Pimenta + Borrão de Movimento (d) Borrão de Movimento	69
Figura 5.12 – Filtros espaciais para ruído gaussiano.....	70
Figura 5.13 – Filtros espaciais para ruído impulsivo.	71
Figura 5.14 – Filtros espaciais para ruído de impulso + borrão de movimento	72
Figura 5.15 – Filtragem inversa para imagem com (a)Ruído Gaussiano, (b) Ruído Sal e Pimenta (c) Ruído Sal e Pimenta + Borrão de Movimento (d) Borrão de Movimento.....	73
Figura 5.16 – Resultados de MSE, SNR, PSNR e UIQI para filtragem inversa de imagem com borrão de movimento.....	74
Figura 5.17 – Resultados visuais e de MSE, SNR, PSNR e UIQI para filtragem de Wiener de imagem com (a)Ruído Gaussiano, (b) Ruído Sal e Pimenta (c) Ruído Sal e Pimenta + Borrão de Movimento (d) Borrão de Movimento.....	75
Figura 5.18 – (a) Imagem de teste (b) Matriz de pixels da imagem teste original (c) Palavra código gerada na codificação.	77
Figura 5.19 – Medidas de avaliação da codificação de Huffman.....	78
Figura 5.20 – Matriz de intensidade da imagem decodificada.	78

Figura 5.21 – Imagem diagnóstica decodificada por Huffman, medidas de avaliação e palavra código gerada.....	79
Figura 5.22 – Medidas de qualidade da imagem decodificada por Huffman em relação à imagem original.....	80
Figura 5.23 – Imagem codificada e decodificada por RLE.....	81
Figura 5.24 – (a) Matriz da imagem decodificada por RLE (b) parte da palavra código.....	82
Figura 5.25 – Resultado de codificação de MR por RLE.....	83
Figura 5.26 – Medidas de qualidade da imagem decodificada por RLE em relação à imagem original.....	83

LISTA DE SÍMBOLOS E ABREVIATURAS

- ACR** – Colégio Americano de Radiologia, do inglês *American College of Radiology*
- bpp** – Bits por *pixel*
- C_R** – Taxa de Compressão
- CT** – Tomografia Computadorizada, do inglês *Computerized Tomography*
- DEP** – Densidades espectrais de potência
- DICOM** – Padrão de Comunicação de Imagens Digitais em Medicina, do inglês *Digital Imaging and Communications in Medicine*
- DPI** – Pontos por polegada, do inglês *Dots Per Inch*
- GUI** – Interface Gráfica de Usuário, do inglês *Graphical User Interface*
- HIS** – Sistemas de Informação Hospitalar, do inglês *Hospital Information Systems*
- HVS** – Sistema Visual Humano, do inglês *Human Visual System*
- MAE** – Erro Absoluto Médio, do inglês *Mean Absolute Error*
- MSE** – Erro Quadrático Médio, do inglês *Mean Squared Error*
- MR** – Ressonância Magnética, do inglês *Magnetic Resonance*
- NEMA** – Associação de Fabricantes de Equipamentos Elétricos, do inglês *National Electrical Manufactures Association*
- OTF** – Função de Transferência Ótica, do inglês *Optical Transfer Function*
- PACS** - Sistemas de Arquivo e Comunicação de Imagens, do inglês *Picture Archiving and Communication Systems*
- PDF** – Função densidade de probabilidade
- pixel*** – Elemento de Imagem, do inglês *Picture Element*
- PSF** – Função de Espalhamento Pontual, do inglês *Point Spread Function*
- PSNR** – Razão Sinal-Ruído de Pico, do inglês *Peak Signal to Noise Ratio*
- R_D** – Redundância de dados relativa
- RGB** – Padrão Vermelho, Verde e Azul, do inglês *Red Green and Blue*
- RLE** – Codificação por comprimento corrido, do inglês *Run-Length Encoding*.
- RMSE** – Raiz quadrada do Erro Quadrático Médio, do inglês *Root Mean Squared Error*
- SNR** – Razão Sinal-Ruído, do inglês *Signal to Noise Ratio*
- TIFF** – Formato de Arquivo por Tag de Imagem, do inglês *Tag Image File Format*
- UIQI** – Índice de Qualidade da Imagem, do inglês *Universal Image Quality Index*
- VLC** – Código com tamanho variável, do inglês *Variable-Length Coding*

LISTA DE TABELAS

Tabela 4.1 – Imagem genérica 4x8 com 256 tons de cinza.....	49
Tabela 4.2 – Probabilidade de cada nível de cinza em imagem.	49
Tabela 4.3 – Probabilidade de par de tons em imagem.	50
Tabela 6.1 – Resultados Obtidos para Imagem Degradada.....	68
Tabela 6.2 – Métricas para imagem restaurada por filtro inverso	73
Tabela 6.3 – Métricas para imagem restaurada por filtro de Wiener	74
Tabela 6.4 – Resultados Obtidos para SNR	76
Tabela 6.5 – Resultados Obtidos para MSE.....	76
Tabela 6.6 – Resultados Obtidos para Codificação de Huffman.....	80
Tabela 6.7 – Resultados Obtidos para Codificação RLE	84

CAPÍTULO 1 - INTRODUÇÃO

1.1 – Motivação e Apresentação do problema

Em virtude dos avanços da tecnologia, as imagens médicas hoje são em grande parte adquiridas ou armazenadas digitalmente, especialmente imagens em tons de cinza, usadas na radiologia. Porém, sua manipulação computacional vem se tornando complexa em virtude da demanda advinda dos profissionais da saúde para o uso clínico dessas imagens. Tal forma de obtenção de dados médicos teve início nos anos 70, quando ocorreram descobertas revolucionárias que modernizaram a medicina diagnóstica (GROSS, 1998).

O diagnóstico por meio de imagens representa uma importante ferramenta da medicina moderna, por fornecer uma visão interna de quase todas as partes do corpo humano. A aplicação de métodos de análise e processamento digital de imagens pode proporcionar ao médico informações extremamente importantes, sem a necessidade de intervenções maiores, o que amplia a conveniência de exames convencionais como a tomografia computadorizada (do inglês, CT) ou a ressonância magnética (do inglês, MR).

Exames como os citados envolvem a geração de um volume considerável de dados, que precisam ser armazenados satisfatoriamente de forma a se obter um registro para análises e referências futuras. Esses exames envolvem quantidades de dados da ordem de 10 a 50 MB por sessão (RATIB, 1995).

Tal volume de dados representa um grande inconveniente em se tratando das possibilidades de armazenamento das imagens e de transmissão dessas para possíveis diagnósticos à distância, por exemplo. A situação ocorre em Sistemas de Informação Hospitalar (do inglês, HIS), nos quais as imagens são disponibilizadas por meio de redes de comunicação de dados (ROMUALDO, 2009).

Uma questão que entra em pauta se dá na evolução tecnológica hoje, que conduz ao desenvolvimento e à implantação de redes de comunicação digital de banda larga, além de equipamentos para armazenamento em massa de grande capacidade, o que suportaria o volume de dados demandado na aquisição e na transmissão de imagens diagnósticas. A contraposição se baseia também no constante aumento de resolução nas imagens digitalizadas

e por consequência, no volume de dados. Além disso, na tendência de ampliação do domínio das redes de comunicações de dados, integrando-se à internet e no uso de redes sem fio, o que requer economia e escalabilidade da banda passante (CLUNIE, 2000).

Os HIS evoluíram a partir dos Sistemas de Arquivo e Comunicação de Imagens (do inglês, PACS), que são tecnologias em larga escala para intercâmbio de imagens médicas de forma integrada. A tendência de implantação dos PACS em instituições hospitalares tem impulsionado o estudo de técnicas de compressão de dados aplicadas às imagens médicas (KIL et. al, 2006).

O objetivo da compressão é reduzir a quantidade de dados necessária para a representação de uma imagem digital e, então, diminuir o custo de armazenamento e de transmissão desta.

A partir disso foram diversas as pesquisas sobre formas de compressão e foram alcançadas taxas de compressão significativas, mas que acarretaram no descarte de algumas informações presentes nas imagens e que podem ser diagnosticamente importantes. Dessa forma, para imagens de cunho médico, são apropriados métodos de compressão sem perdas, mesmo que as taxas de compressão sejam menores. Isso significa que a imagem reconstruída a partir da imagem comprimida é idêntica à imagem original em todos os sentidos.

Em experimentos da radiologia, é frequente a necessidade de aquisição de imagens com longos intervalos de tempo (GAMBA et. al, 1999). Isso implica, algumas vezes, em imagens com baixa Relação Sinal-Ruído (do inglês, SNR), ou seja, os ruídos obtidos na aquisição da imagem possuem maior efeito sobre essa. Além disso, em virtude de algum tipo de interferência, a aquisição de imagens digitais pode resultar em uma imagem com pouca nitidez. Em se tratando de imagens complexas como as utilizadas para fins de diagnóstico, é necessário o uso de filtros capazes de reduzir os ruídos ou borrões aparentes, sem que a imagem seja afetada (GAMBA & DELPY, 1998).

O padrão de comunicação de imagens digitais em medicina (do inglês, DICOM), criado pela Associação de Fabricantes de Equipamentos Elétricos (do inglês, NEMA), é usado na medicina com objetivo de criar uma padronização na comunicação e no armazenamento das imagens produzidas por equipamentos que fazem exames médicos. Com o crescimento da informática e o aumento do volume das imagens geradas, foi necessário padronizar os procedimentos para que equipamentos de plataformas diferentes compartilhassem suas informações uns com os outros, possibilitando o surgimento de diagnósticos mais detalhados

e à distância. Isso faz com que os governos e entidades economizem nos custos em atendimentos médicos à população, evitando remoções desnecessárias, com salas de laudo e diagnósticos virtuais (KAUR & KAUSHAL, 2008).

Imagens médicas digitais, advindas de exames como a ressonância magnética e a tomografia computadorizada, são comumente geradas no formato DICOM. Para este trabalho, as imagens são convertidas para o Formato de Arquivo por Tag de Imagem (do inglês, TIFF), mantendo-se a resolução espacial e o contraste original.

O TIFF é um dos formatos mais usados para aplicações científicas. Isso pela permissão do armazenamento de quaisquer informações junto à imagem e é capaz de armazenar imagens de até 16 bits por pixel, inclusive com valores em ponto flutuante (BAUERMANN, 2008).

1.2 – Objetivos do Trabalho

Em virtude do apresentado, torna-se imprescindível aplicar técnicas de compressão de dados em imagens, visando menor custo de armazenamento e de transmissão dessas, mas garantindo expansão posterior de tais dados para que a reconstrução seja idêntica a original e passível de processamento. Além disso, para imagens com alta SNR, em virtude de interferências ou mesmo de corrupções nos canais de transmissão, é necessária a aplicação de filtros para recuperação da qualidade da imagem.

A escolha do método apropriado, tanto para compressão de dados quanto para o filtro, depende do conhecimento da teoria do método, bem como, do prévio conhecimento das características presentes na imagem.

O objetivo deste trabalho consiste, por meio do desenvolvimento de interface gráfica, na avaliação de dois métodos de codificação de dados que estão inclusos em compressões de imagens sem perdas, como o Código de Huffman e o Código por Comprimento Corrido (do inglês, RLE). Além disso, objetiva-se a simulação dos ruídos gaussiano e impulsivo e do borrão de movimento nas imagens médicas digitais para a posterior comparação entre os resultados de métodos de restauração por filtragem espacial, filtragem inversa e filtragem de Wiener.

1.3 – Escopo do Trabalho

As imagens obtidas para uso no projeto foram cedidas pelo Dr. Pedro Renato de Paula Brandão – Hospital das Clínicas/Neurologia/USP – Ribeirão Preto, o qual garante a validade e originalidade de tais imagens, advindas de exames de ressonância magnética e de tomografia computadorizada. As imagens diagnósticas usadas neste trabalho estão disponíveis no formato DICOM e são convertidas para os formatos TIFF (.tif ou .tiff) e Bitmap (.bmp).

O projeto contempla a criação de uma interface gráfica de usuário (do inglês, GUI), desenvolvida no MATLAB® versão 7.10.0.499 (R2010a), capaz de simular os ruídos gaussiano e impulsivo, adicionando-os às imagens e diminuindo a SNR dessas; incluindo também a adição simulada de borrões de movimento. A partir disso, a GUI possui a funcionalidade de restauração da imagem por meio de filtros espaciais e também por filtragem inversa, não incluindo sua forma adaptativa, e filtragem de Wiener. Não são incluídas todas as formas de restauração de imagem existentes, selecionando-se, dentre as desenvolvidas, aquela que possui melhor desempenho para cada caso.

Em adição, a GUI abrange as codificações de Huffman e RLE como parte para compressões de dados sem perda, conhecidas como *Lossless*. Não se contempla todo o processo de compressão, mas sim a codificação e a decodificação das imagens, visando à redução de dados necessários para representação dessas e a posterior reconstrução desses dados de forma idêntica à original.

A qualidade da imagem restaurada em relação à imagem original será avaliada em outra funcionalidade, na qual se apresentam os cálculos dos valores de Erro Quadrático Médio (do inglês, MSE), Relação Sinal-Ruído (do inglês, SNR), Relação Sinal-Ruído de Pico (do inglês, PSNR) e o Índice Universal de Qualidade da Imagem (do inglês, UIQI). Esse último, proposto por (WANG & BOVIK, 2002).

1.4 – Estrutura da Monografia

Este trabalho está estruturado em seis capítulos. No primeiro capítulo estão descritos a motivação relativa ao tema escolhido, a apresentação do problema, o escopo do projeto e a estrutura da monografia.

No segundo capítulo, inicialmente é apresentado um histórico do processamento digital de imagens. Em seguida, comenta-se sobre os fundamentos da imagem digital e, finalizando o capítulo, sobre as necessidades advindas da armazenagem e transmissão dessas imagens.

O capítulo terceiro discorre sobre os modelos e tipos de ruídos que podem interferir na qualidade de uma imagem digital. Está abrangida também a explanação de técnicas de restauração de imagens degradadas e sobre as métricas de qualidade e critérios de fidelidade entre uma imagem original e uma imagem processada.

No quarto capítulo estão descritos os fundamentos da compressão de imagens digitais e o modelo genérico de compressão de dados. Incluem-se comentários sobre a classificação para os tipos de compressão de dados existentes e sobre o cálculo da taxa de compressão. Abrange-se com maior destaque a compressão sem perdas e são descritos, por fim, algoritmos de compressão de dados usados como parte do processo de compactação de imagens.

O quinto capítulo descreve a metodologia utilizada para a realização dos experimentos deste trabalho, dentro do objetivo, sendo esse a criação de uma GUI com funcionalidades de simulação de ruídos, de restauração de imagens degradadas, de codificação de Huffman e RLE como etapas para um processo de compactação, e dos cálculos das medidas de qualidade das imagens processadas.

Por fim, no sexto capítulo, estão a conclusão e a apresentação de propostas para trabalhos futuros. Visando complementar as teorias estudadas, são apresentados nos apêndices teorias sobre padrões de compressão de imagens que fazem uso das codificações de Huffman e RLE, além de explanação sobre os formatos de imagens utilizados neste trabalho.

CAPÍTULO 2 – CONCEITOS BÁSICOS

2.1 – Histórico do Processamento de Imagens e Imagens Médicas Digitais

O processamento de imagens é uma área de interesse há algumas décadas, sendo ainda de importância crescente, incluído aqui a área de compressão e restauração de dados. Conforme descrito em (MARQUES, 1999), uma das primeiras aplicações em processamento de imagens remonta do começo do século, onde se buscavam formas de aprimorar a qualidade de impressão de imagens digitalizadas transmitidas por meio do sistema de transmissão de imagens por cabo submarino entre Londres e Nova Iorque. O equipamento codificava as imagens para posterior reconstrução no receptor, mas o exemplo não se enquadra totalmente no quesito de processamento de imagens pela falta de existência dos computadores na época, mas sem dúvida foi um marco para as futuras técnicas (SYMES, 1998),

O uso de técnicas computacionais no melhoramento de imagens produzidas por sondas espaciais teve início na Califórnia, Estados Unidos da América, quando imagens transmitidas da Lua foram processadas por um computador para corrigir diversas distorções advindas da câmera de bordo. Tais técnicas serviram de base para métodos de realce de restauração de imagens das missões para Lua e Marte (SYMES, 1998).

Desde então, a área de processamento de imagens cresce exponencialmente. Na medicina, procedimentos computacionais melhoram o contraste ou codificam níveis de intensidade em cores, de modo a facilitar a interpretação de imagens de raio-X, como por exemplo imagens de tomografia computadorizada e outras imagens biomédicas (SYMES, 1998). Além disso, fala-se do processamento de imagens como facilitador para o armazenamento e para a transmissão dessas.

Em 1946, grupos da Universidade de Stanford e do Instituto de Tecnologia de Massachusetts publicaram suas descobertas em ressonância magnética (do inglês, MR). Mas apenas em 1973, Paul Lauterbur e Peter Mansfield fizeram as primeiras imagens utilizando MR (TINOIS, 2000).

O aumento da frequência com a qual as informações são transmitidas, armazenadas, processadas e visualizadas no formato digital, a acentuada queda de preços que os

dispositivos eletrônicos sofrem e a melhoria do desempenho e da expansão nas redes de comunicação estão criando grandes oportunidades na área de processamento digital de imagens digitais. Da mesma forma ocorre com a necessidade do desenvolvimento de técnicas eficientes para todas essas tarefas, de forma a melhorar a integridade visual da informação de interesse.

2.2 – Fundamentos da Imagem Digital Monocromática

Uma imagem digital é composta por pixels, podendo esses serem definidos como pequenos pontos na imagem. Em geral uma imagem possui um tamanho de $M \times N$, com M pixels na direção vertical e N pixels na horizontal.

Uma imagem em escala de cinza ou de intensidade pode ser descrita matematicamente por uma função $f(x,y)$ da intensidade luminosa (equação 2.1), sendo seu valor, em qualquer ponto de coordenadas espaciais (x,y) , proporcional ao brilho (ou nível de cinza) da imagem naquele ponto (MARQUES & VIEIRA, 1999).

A função $f(x,y)$ representa o produto da interação entre a iluminância $i(x,y)$, ou seja, a quantidade de luz que incide sobre o objeto, e as propriedades de refletância, que podem ser representadas pela função $r(x,y)$, cujo valor exprime a fração de luz incidente que o objeto vai transmitir ou refletir ao ponto (x,y) .

$$f(x, y) = i(x, y) \cdot r(x, y)$$

$$0 < i(x, y) < \infty \quad (2.1)$$

$$0 < r(x, y) < 1$$

Iluminância (lúmen/m²) e refletância(%) de uma imagem.
 FONTE: Modificado de (MARQUES & VIEIRA, 1999)

No caso de uma imagem que possui informações em intervalos ou bandas distintas de frequência, é necessária uma função $f(x,y)$ para cada banda. É o caso de imagens coloridas no padrão vermelho, verde e azul (do inglês, RGB), formadas pela informação das cores primárias. As técnicas de processamento de imagens descritas neste trabalho tratam fundamentalmente com imagens monocromáticas, bidimensionais e estáticas.

Para conversão de um objeto ou cena real em uma imagem digital, são seguidas algumas etapas sendo a primeira delas a aquisição. Essa é realizada por meio de sensores que captam radiações refletidas ou alteradas, sendo essas as que atravessam um objeto, transformando-as em um sinal elétrico. (MARQUES & VIEIRA, 1999).

A segunda etapa se dá na digitalização desse sinal com uma conversão analógico-digital, composta da amostragem e da quantização (figura 2.1). Basicamente, a amostragem converte a imagem analógica em uma matriz de M por N pontos (figura 2.2), cada qual denominado elemento de imagem (*pixel*). Para maiores valores de M e N , implica-se uma imagem de maior resolução.

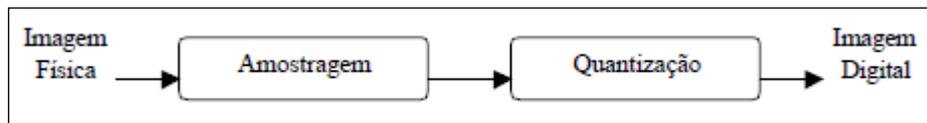


Figura 2.1 – Etapas de digitalização da imagem

FONTE: Modificado de (OLIVEIRA, 2003)

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

Figura 2.2 – Matriz $M \times N$ da imagem

FONTE: Modificado de (MARQUES & VIEIRA, 1999)

Sendo G o número de níveis de cinza (equação 2.2), cada elemento de imagem ou pixel tem associado um valor G_{min} ou $0 \leq f(x,y) \leq G_{max}$ ou $2^m - 1$, tal que o intervalo $[G_{min}, G_{max}]$ é denominado escala de cinza e quanto maior o valor de m , maior o número de níveis de cinza presentes. A intensidade f de uma imagem monocromática nas coordenadas (x,y) é chamado de nível de cinza da imagem naquele ponto. Uma convenção comum é atribuir a cor preta ao nível de cinza mais escuro (valor 0) e atribuir a cor branca ao nível de cinza mais claro (valor 255) (SCHWARTZ & PEDRINI, 2007).

Já na quantização, deve-se definir que valores de M , N e m são adequados, do ponto de vista de qualidade da imagem e da quantidade de bytes necessários para armazená-la. Após a amostragem, a imagem fica na forma discreta no tempo, mas apresenta ainda certa quantidade

de amplitudes. Define-se então a quantidade de níveis de cinza que representa a imagem. Os valores de M , N e da quantidade de níveis de quantização m definem o tamanho da imagem sem compressão. O número de bits necessários para representar um *pixel* na imagem é dado por b , pela equação (2.2) a seguir (OLIVEIRA, 2003).

$$\begin{aligned} b &= N.M.m \\ G &= 2^m \end{aligned} \quad (2.2)$$

Número de bits para representação de imagem e o nível de cinza
 FONTE: (OLIVEIRA, 2003)

Ao ser digitalizada a imagem em pixel assume um tamanho adimensional. Apesar disso, é reconhecida uma medida de qualidade da amostragem, sendo a razão entre o número de pixels obtidos e o tamanho da imagem real ou equivalente. É a conhecida resolução, medida em pontos por polegada (do inglês, DPI) (SCURI, 2002).

2.2.1 – Topologia da imagem

Uma imagem digital é uma imagem $f(x,y)$ discretizada tanto espacialmente quanto em amplitude. Portanto, uma imagem digital pode ser vista como uma matriz cujas linhas e colunas identificam um ponto na imagem. O valor numérico de tal ponto corresponde ao nível de cinza da imagem naquele ponto.

Um pixel p , de coordenadas (x,y) , tem 4 vizinhos horizontais e verticais, cujas coordenadas são $(x+1, y)$, $(x-1, y)$, $(x, y+1)$ e $(x, y-1)$. Estes pixels formam a chamada "4-vizinhança" de p , que será designada $N4(p)$. Os quatro vizinhos diagonais de p são os pixels de coordenadas $(x-1, y-1)$, $(x-1, y+1)$, $(x+1, y-1)$ e $(x+1, y+1)$, que constituem o conjunto $Nd(p)$. Dessa forma, a "8-vizinhança" de p é definida como a união entre $N4(p)$ e $Nd(p)$ (figura 2.3) (GONZALES & WOODS, 2001).

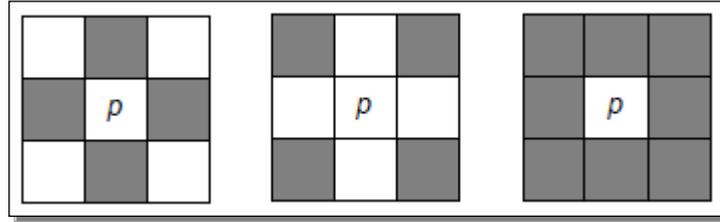


Figura 2.3 – Conceitos de 4-vizinhança, vizinhança diagonal e 8-vizinhança.
FONTE: (GONZALES & WOODS, 2001)

2.2.2 – Histograma

O histograma de uma imagem corresponde à distribuição dos níveis de cinza da imagem, o qual pode ser representado por um gráfico indicando o número de pixels na imagem para cada nível de cinza.

Várias medidas estatísticas podem ser obtidas a partir do histograma de uma imagem, tais como os valores mínimos e máximos, o valor médio, a variância e o desvio padrão dos níveis de cinza da imagem. Os histogramas também representam a Função de Densidade de Probabilidades (do inglês, PDF), ou seja, representam a probabilidade de se encontrar determinado tom de cinza na imagem (figura 2.4) (SILVA, 2001).

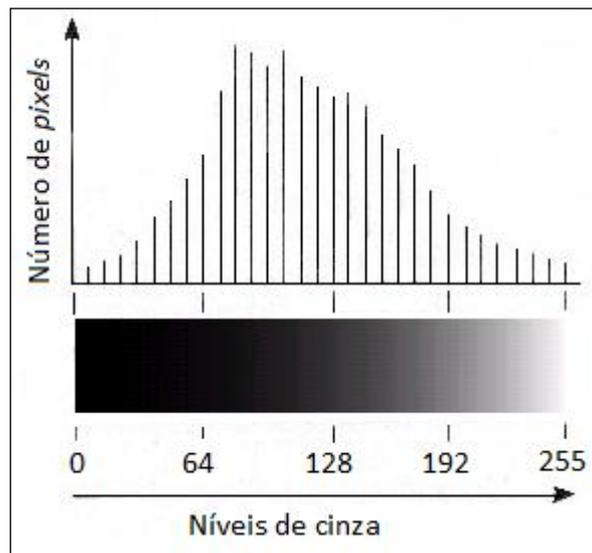


Figura 2.4 – Histograma de imagem e escala de cinza
FONTE: Modificado de (LABGIS, 2011)

Segundo (LABGIS, 2011), infere-se do formato de um histograma informações sobre características de contraste de uma imagem. Um histograma estreito indica uma imagem de baixo contraste e um histograma multimodal indica a existência de regiões com diferente brilho (figura 2.5).

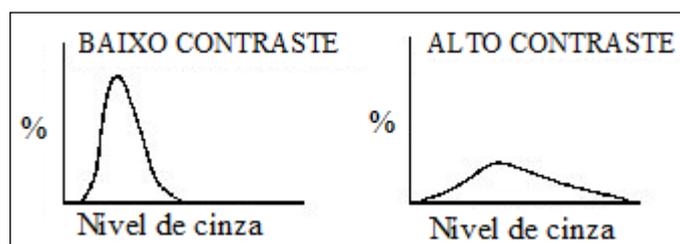


Figura 2.5 – Formato de Histograma x Contraste da imagem

FONTE: *Modificado de (LABGIS, 2011)*

Segundo (NETO, 2010), de acordo com a equação (2.3), a escala de cinza normalizada é representada por r_k , sendo $0 \leq r_k \leq 1$. Já o número de níveis de cinza da imagem é representado por G , sendo $0 \leq k \leq G-1$. N representa o número total de pixels na imagem e n_k o número de pixels correspondentes ao nível de cinza k . Por fim, a probabilidade do k -ésimo nível de cinza é representado por $p(r_k)$.

$$p(r_k) = \frac{n_k}{N} \quad (2.3)$$

Cálculo de elementos do histograma.

FONTE: *(NETO, 2010)*

2.3 – Armazenamento e Transmissão

O armazenamento de imagens médicas de maneira padronizada tem proporcionado inúmeros fatores favoráveis, como: possibilidade de criação de uma biblioteca digital de imagens médicas, permitindo a organização, catalogação, disponibilização de acesso, conservação e apoio ao ensino.

No manejo de informação dentro de um hospital, por meio de uma rede de computadores, demonstrou-se que é possível utilizar sistemas computadorizados para

melhorar o gerenciamento dos pacientes, geração e distribuição de relatórios, facilidades de utilização dos recursos disponíveis, localização dos filmes e funcionamento do setor de radiologia (STOLFI, 2000).

Atualmente, imagens advindas de CT e MR requerem um grande espaço físico em departamentos de radiologia. Surgiu daí um conceito de armazenagem conhecido como Sistema de Arquivamento e Comunicação de Imagens (do inglês, PACS) (STOLFI, 2000). A finalidade deste é possibilitar, com o uso de softwares, a visualização e outras operações com imagens armazenadas nos servidores de imagens médicas digitais.

Este sistema, em conjunto com os sistemas de informação radiológica e de informação hospitalar (do inglês, HIS) forma a base para um serviço de radiologia informatizado. A implantação desse serviço traz melhorias no quesito de acessibilidade e integração de informações, pela vinculação de imagens ao registro médico eletrônico do paciente (CANTADORI & FURUIE, 2002).

Quando as taxas de transmissão proporcionadas pelas redes locais convencionais não suportam o volume de dados necessário para transmissão em tempo real, o acesso normalmente é mais demorado. Assim sendo, os dados são antes transferidos integralmente para a estação de trabalho para serem posteriormente visualizados.

No caso de acesso remoto, onde as taxas de transmissão por Modem muitas vezes não superam 10 a 20 kb/s, a transmissão de um exame completo é impraticável, a menos que se empreguem processos de compressão de dados (STOLFI, 2000).

Dentro de um Sistema Integrado de Informações Médicas, nos moldes de um HIS, torna-se necessário prover o acesso aos resultados de exames através de uma rede de comunicação de dados, tanto dentro do âmbito de uma instituição como remotamente (RATIB, 1995). Imediatamente após a realização de um exame, antes do seu arquivamento, as imagens geradas são armazenadas temporariamente em um servidor, sendo acessíveis através de estações de trabalho dentro do hospital.

Todo o esforço para reduzir o espaço ocupado pela imagem tem de levar em consideração a sua qualidade. Dessa forma, a interoperacionalidade de sistemas relacionados com a transferência de imagem digital, tal como o DICOM, é conseguida por meio da melhor relação do binômio nível de compressão/qualidade (MARTINS, 2002).

CAPÍTULO 3 – RESTAURAÇÃO DE IMAGENS DIGITAIS

A restauração de imagens consiste no retorno ao estado original da imagem por meio da remoção ou redução da degradação, ocorrida durante sua aquisição, incluindo o borramento introduzido pelo sistema óptico, bem como ruídos eletrônicos. Técnicas de restauração são orientadas para a modelagem da degradação e para a aplicação do processo inverso.

Embora haja discussões entre realce de imagens e restauração de imagens, o realce é em grande parte um processo subjetivo, baseado essencialmente em aspectos visualmente agradáveis para o espectador. Já a restauração se dá como um processo objetivo, visando recuperar imagens degradadas com uso de conhecimentos prévios do fenômeno de degradação (GONZALES & WOODS, 2001).

Como evidenciado na figura (3.1) a seguir, assume-se que a função de degradação H junto ao termo aditivo do ruído $\eta(x,y)$ operam em uma imagem de entrada $f(x,y)$ e produzem a imagem degradada $g(x,y)$. Em vista disso, o objetivo se dá na estimativa da imagem original.

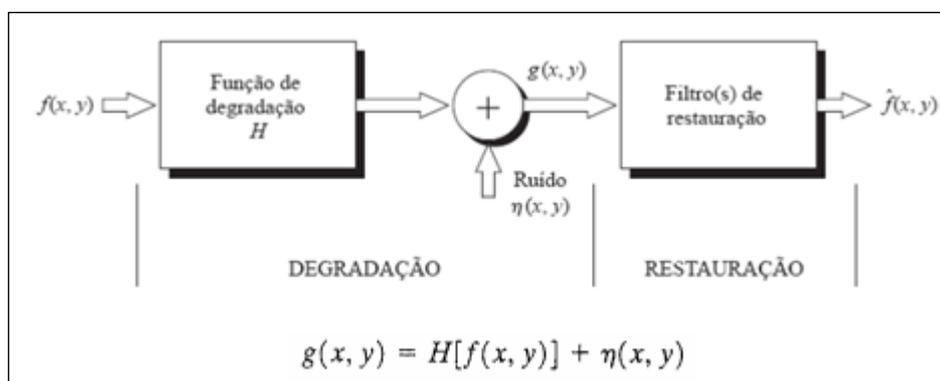


Figura 3.1 – Modelo de degradação e restauração da imagem digital
 FONTE: Modificado de (GONZALES & WOODS, 2001)

Se H for um processo linear com posição invariante, a imagem degradada em domínio espacial segue como mostra a equação (3.1), sendo h a representação em nível de espaço da função de degradação. Seguindo a representação em domínio de frequência (equação 3.2), com $F(u,v)$, $H(u,v)$ e $N(u,v)$ sendo as respectivas Transformadas de Fourier de $f(x,y)$, $h(x,y)$ e $\eta(x,y)$:

$$g(x, y) = h(x, y) * f(x, y) + n(x, y) \quad (3.1)$$

Imagem degradada $g(x, y)$, sendo “*” a convolução.

FONTE: (GONZALES & WOODS, 2001)

$$G(u, v) = H(u, v)F(u, v) + N(u, v) \quad (3.2)$$

Imagem degradada $G(u, v)$ em domínio de frequência

FONTE: (GONZALES & WOODS, 2001)

$H(u, v)$ é também chamada por Função de Transferência Ótica (do inglês, OTF) enquanto $h(x, y)$ é conhecida por Função de Espalhamento Pontual (do inglês, PSF).

3.1 – Modelos e Tipos de Degradação

A maior parte da degradação por ruído ou borrões vem durante a aquisição ou transmissão das imagens digitais. Essas podem ser afetadas por diversos fatores como níveis de luz e temperatura ou até mesmo movimentações durante o processo de aquisição. Assume-se que ruídos são independentes de coordenadas espaciais e não há correlação entre os valores dos pixels e os valores dos componentes do ruído (GONZALES & WOODS, 2001).

Segundo (SEARA, 1998), toda a aquisição de imagens está sujeita a algum tipo de ruído. Devido à natureza randômica não há como prevê-los, e por isso, não há como se obter uma medida precisa dos ruídos. Em detrimento disso, existem alguns casos em que ele pode ser caracterizado pelo efeito na imagem. Nesses casos, o ruído é geralmente expresso por meio de uma distribuição de probabilidade com uma média específica, além de um desvio padrão, por uma Função Densidade de Probabilidades (do inglês, PDF) que é sobreposta à PDF da imagem original. À seguir são discorridos alguns dos tipos mais comuns de ruídos.

3.1.1 – Ruído gaussiano

O ruído gaussiano é normalmente distribuído e aparece na imagem pelo ruído térmico no amplificador e por outros efeitos. O ruído gaussiano é caracterizado pela ocorrência de *pixels* com valores de intensidade que variam conforme a distribuição gaussiana (SCHWARTZ & PEDRINI, 2007).

A figura (3.2) mostra a PDF do ruído randômico gaussiano, na qual μ é a média e σ é o desvio padrão.

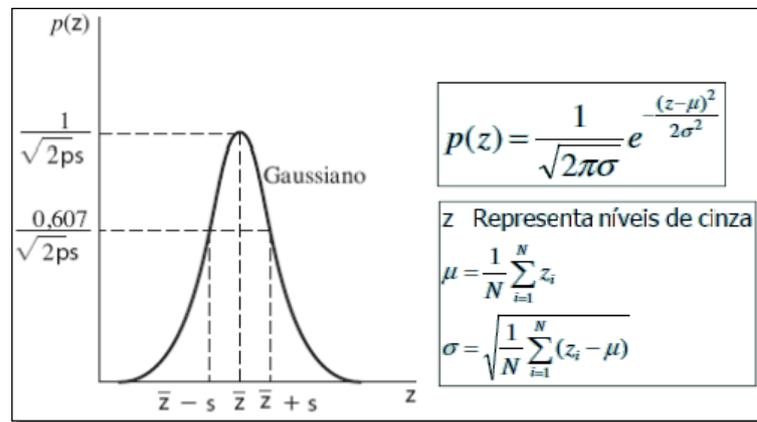


Figura 3.2 – Ruído gaussiano e PDF correspondente

FONTE: Modificado de (REN & CAVALCANTI, 2010)

Aproximadamente 70% dos valores estão contidos entre $\mu \pm \sigma$ e 90% dos valores estão contidos entre $\mu \pm 2\sigma$. Apesar de, teoricamente, a PDF ser não nula de $-\infty$ a $+\infty$, costuma-se considerar a função igual a 0 além de $\mu \pm 3\sigma$ (REN & CAVALCANTI, 2010).

O ruído gaussiano se faz útil para a modelagem de processos naturais que introduzem ruídos (MOGER, 2008).

3.1.2 – Ruído impulsivo

O modelo de ruído impulsivo, mais conhecido como *Salt and Pepper* (Sal e Pimenta), é representado pela ocorrência aleatória de *pixels* brancos (Sal) e pretos (Pimenta). Possui apenas dois valores possíveis, a e b, e a probabilidade de obtenção de cada um é menor que 0.1, de modo que o ruído não domine a imagem por completo.

Esse tipo de ruído é causado a partir de erros na transmissão de dados. Ocorre uma diferença bruta nos tons de cinza existentes entre *pixels* e sua vizinhança, devido ao fato de que os *pixels* corrompidos ou possuem alguns bits alterados, ou estes *pixels* são alterados para um valor máximo. O ruído impulsivo surge quando ocorre a modificação alternada do *pixel*

para 0 ou para o máximo, dando à imagem uma aparência de mistura de sal e pimenta, daí seu nome (RIBEIRO, 2006).

O ruído é especificado de acordo com a figura (3.3). Se $b > a$, a intensidade b aparecerá como um ponto claro na imagem e a aparecerá como um ponto escuro.

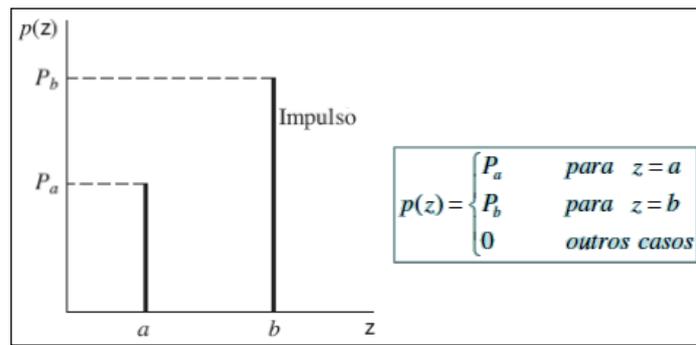


Figura 3.3 – Ruído impulsivo (*Salt and Pepper*) e PDF correspondente
 FONTE: Modificado de (REN & CAVALCANTI, 2010)

3.1.3 – Borrão de movimento

Além dos ruídos, podem ser também significativas as alterações nas imagens por meio de borrões. Conhecidos como borrões de movimento, são resultados do movimento relativo entre a fonte de aquisição e o objeto focado. Trata-se de um efeito com grande capacidade de degradação da imagem (BEN-EZRA & NAYAR, 2004).

O sensor de uma câmera acumula a luz recebida durante um determinado período de tempo. A imagem registrada se torna borrada, desfocada ou distorcida, como na figura (3.4), devido ao fato de que não foi registrado apenas um instante, mas sim todas as posições que o objeto focado ocupou durante o tempo necessário de captação do sensor.



Figura 3.4 – Imagem genérica com borrão de movimento
 FONTE: (GONZALES & WOODS, 2001)

3.2 – Técnicas de Restauração de Imagens Degradadas

Um ruído periódico vem tipicamente por interferências elétricas ou eletromagnéticas durante a aquisição da imagem, sendo que seus parâmetros para a PDF são estimados em domínio de frequência. O ruído pode ser identificado por meio da análise do espectro da imagem, uma vez que surge sobre a forma de picos, como mostra a figura (3.5).

Quando apenas existe uma imagem para analisar, os parâmetros podem ser obtidos analisando uma região homogênea da imagem para obter a média e o desvio padrão, sendo o tipo de distribuição identificado pela análise do histograma e pela aparência da imagem (REN & CAVALCANTI, 2010).

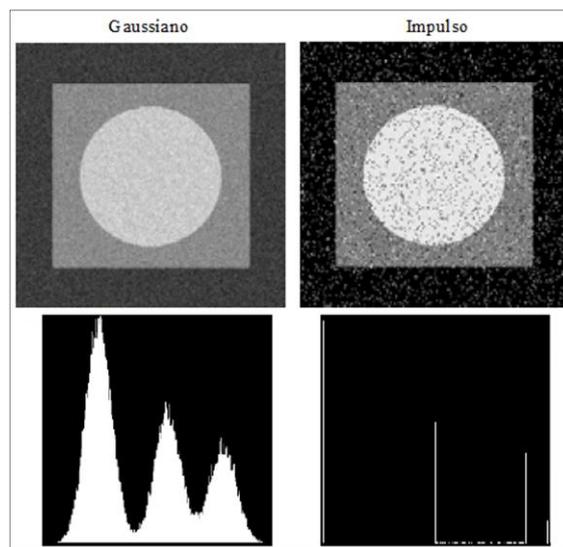


Figura 3.5 – Histogramas dos modelos de ruídos
 FONTE: Modificado de (REN & CAVALCANTI, 2010)

A restauração de imagens degradadas por borrões de movimento, requerem uma estimativa dos parâmetros da PSF. A função para borrões de movimento é caracterizada pelos parâmetros de comprimento e direção. O comprimento do borrão é o número de posições de *pixels* pelos quais um *pixel* é deslocado de sua posição original, enquanto a direção é o ângulo tomado pelo borrão. (MATHWORKS, 2011).

3.2.1 – Restauração apenas com ruído (filtros espaciais)

Quando o ruído é a única forma de degradação presente, como representado na figura (3.6), os termos desse são desconhecidos, não sendo possível uma simples subtração (Equações 3.3 e 3.4). Dessa forma, são bons candidatos os filtros espaciais (GONZALES & WOODS, 2001).

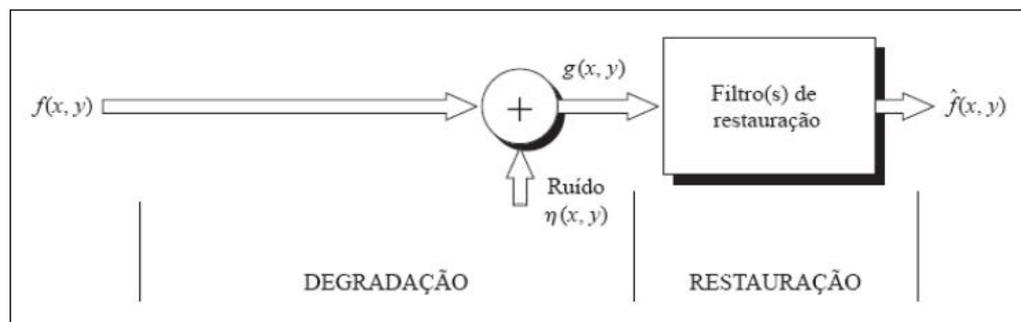


Figura 3.6 – Degradação de imagem apenas com ruído.
 FONTE: Modificado de (MOGER, 2008)

$$g(x, y) = f(x, y) + n(x, y) \quad (3.3)$$

$$G(u, v) = F(u, v) + N(u, v) \quad (3.4)$$

Funções de imagem degradada em domínio espacial e de frequência.

FONTE: (REN & CAVALCANTI, 2010)

Esses métodos de filtragem operam diretamente sobre a matriz de pixels da imagem digitalizada e normalmente utilizam operações entre a imagem original e uma máscara especialmente construída. As máscaras são chamadas de filtros espaciais (THOMÉ, 2004).

Ao ser aplicada a máscara com centro numa posição (i,j) , sendo i o número de uma dada linha e j o número de uma dada coluna da imagem, substitui-se o valor do *pixel* na posição (i,j) por um novo valor, o qual depende dos valores dos *pixels* vizinhos e dos pesos da máscara. O novo nível de cinza do ponto depende do seu nível de cinza original e do de outros pontos considerados como vizinhança (THOMÉ, 2004).

Ainda segundo (THOMÉ, 2004), existem dois tipos de filtros conhecidos como filtros lineares e não lineares. Os primeiros suavizam e minimizam efeitos do ruído, não alterando o nível médio de cinza da imagem. Já o segundo tipo aplica transformações sem o compromisso de manter o nível de cinza da imagem original. O que acontece com o tipo linear é uma suavização de contornos, problema abordado pelo tipo não linear que evita uma filtragem homogênea em regiões próximas a esses contornos. É abordado neste trabalho o filtro não linear da média aritmética.

Uma classe de filtros bastante empregada na eliminação de ruídos, com preservação de contornos, são os filtros não lineares estatísticos da ordem também inclusos neste trabalho: filtro da mediana; filtro de valor máximo e filtro de valor mínimo.

O filtro de média aritmética, representado na equação (3.5) a seguir, localiza a média aritmética dos valores de *pixels* da máscara a ser sobreposta. Essa deve ser do menor tamanho possível, visto que quanto maior, maior será o borramento deixado na imagem. É um filtro baseado em um kernel que representa a forma e o tamanho da vizinhança a ser amostrada no cálculo da média (THOMÉ, 2004).

$$f(x, y) = \frac{1}{mn} \sum_{(s,t) \in S_{xy}} g(s, t) \quad (3.5)$$

Filtro de média aritmética

FONTE: Modificado de (REN & CAVALCANTI, 2010)

Seguindo pelos filtros estatísticos da ordem, o filtro da mediana (equação 3.6) reduz o ruído de forma semelhante ao filtro da média. Considera-se cada *pixel* por vez, mas ao invés de se alterar o valor pela média de valores da vizinhança, a alteração se dá pela mediana de tais valores, como mostrado na figura (3.7) (THOMÉ, 2004).

$$f(x, y) = \underset{(s, t) \in S_{xy}}{\text{mediana}} \{g(s, t)\} \quad (3.6)$$

Filtro da mediana

FONTE: *Modificado de (REN & CAVALCANTI, 2010)*

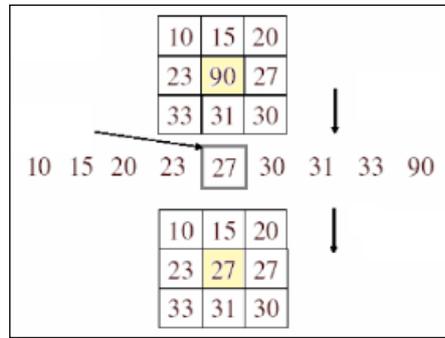


Figura 3.7 – Operação do filtro da mediana

FONTE: *(REN & CAVALCANTI, 2010)*

Já o filtro de valor máximo, segundo (THOMÉ, 2004), é representado pela equação (3.7) e seleciona o maior valor dentro de uma máscara ordenada de valores de *pixels*. Dessa forma, infere-se a redução do ruído pimenta, cujo valor da intensidade é o mais baixo de todos. A imagem se torna mais clara e na presença do ruído do tipo sal, esse se torna mais evidente.

$$f(x, y) = \underset{(s, t) \in S_{xy}}{\max} \{g(s, t)\} \quad (3.7)$$

Filtro de valor máximo

FONTE: *Modificado de (REN & CAVALCANTI, 2010)*

Por fim, o filtro de valor mínimo (equação 3.8) seleciona o menor valor dentro de uma máscara ordenada de valores de *pixels*. Sendo assim, o filtro tem bom funcionamento para ruído do tipo sal, visto que esse possui valor de intensidade máximo. A imagem filtrada tende a escurecer.

$$f(x, y) = \underset{(s, t) \in S_{xy}}{\min} \{g(s, t)\} \quad (3.8)$$

Filtro de valor mínimo

FONTE: *Modificado de (REN & CAVALCANTI, 2010)*

3.2.2 – Restauração com função de degradação (filtro inverso e filtro de Wiener)

Segundo (GONZALES & WOODS, 2001), em alguns casos é possível determinar a função de degradação e aplicá-la para restaurar toda a imagem (equação 3.9).

$$H_S(u, v) = \frac{G_S(u, v)}{F_S(u, v)} \quad (3.9)$$

Função $H(u, v)$ de degradação da imagem
 FONTE: (GONZALES & WOODS, 2001)

Em caso de disponibilidade do equipamento de aquisição da imagem, a função de degradação pode ser obtida por meio da resposta do equipamento a um impulso de amplitude A (equação 3.10).

$$H_S(u, v) = \frac{G_S(u, v)}{A} \quad (3.10)$$

Função de degradação da imagem
 FONTE: (GONZALES & WOODS, 2001)

Seguindo essa linha, com conhecimento da função de degradação da imagem, são citadas duas das formas de restauração dessa imagem: Filtragem Inversa e Filtragem Wiener.

Pela filtragem inversa, uma estimativa da imagem restaurada pode ser obtida pela Transformada inversa de Fourier de $\hat{F}(u, v)$, segundo a equação (3.11), lembrando que $G(u, v)$ é a Transformada de Fourier da imagem degradada. A partir desta equação, infere-se que, mesmo com o conhecimento da função de degradação, a restauração da imagem é limitada pelo componente de ruído, uma função randômica, cuja Transformada $N(u, v)$ não é conhecida. Apresenta-se o problema de amplificação do ruído nos pontos onde $H(u, v)$ é próxima de 0, que torna o componente de ruído dominante na estimativa (GONZALES & WOODS, 2001). Dessa forma, a filtragem inversa não obtém bons resultados para restauração de imagens ruidosas, sendo de melhor funcionalidade em casos de imagens com borrões de movimento (VIEIRA, 2010).

$$\hat{F}(u, v) = F(u, v) + \frac{N(u, v)}{H(u, v)} \quad (3.11)$$

Estimativa de imagem restaurada por filtragem inversa.

FONTE: (GONZALES & WOODS, 2001)

Já o filtro de Wiener, é um filtro linear com bons resultados para minimizar o valor do Erro Quadrático Médio (do inglês, MSE) entre as imagens original e distorcida por ruídos (MAGNUSSON, 2010). Apresenta-se como melhor opção na restauração de imagens ruidosas.

Wiener se baseia no conhecimento prévio das densidades espectrais de potência (DEP) da imagem ideal e do ruído. Assim, apesar de ser ótimo, seu desempenho depende de estimativas acuradas das DEP, quando não substituídas por constantes K . O filtro de Wiener convencional também requer que os campos do sinal e do ruído sejam homogêneos (GONZALES & WOODS, 2001). O filtro é definido pela equação (3.12).

$$F(u, v) \approx \left[\frac{1}{H(u, v)} \frac{|H(u, v)|^2}{|H(u, v)|^2 + K} \right] G(u, v) \quad (3.12)$$

Estimativa de imagem restaurada por filtragem Wiener.

FONTE: (GONZALES & WOODS, 2001)

A figura (3.8) mostra um comparativo entre resultados de restauração por filtragem inversa e por filtragem de Wiener aplicada em imagem com inserção de ruído. No caso do filtro inverso, e de acordo com a equação (3.11), nota-se que a amplificação do ruído é dominante (VIEIRA, 2010). Já na figura (3.9), compara-se a imagem original com a imagem borrada (borrão de movimento) e com a imagem filtrada por filtro inverso.

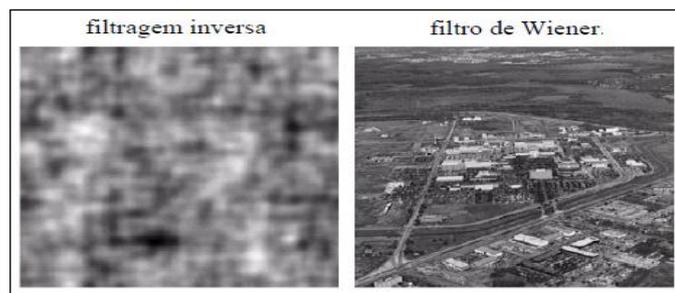


Figura 3.8 – Comparativo entre Filtragem Inversa com ruído e Filtragem Wiener

FONTE: Modificado de (GONZALES & WOODS, 2001)

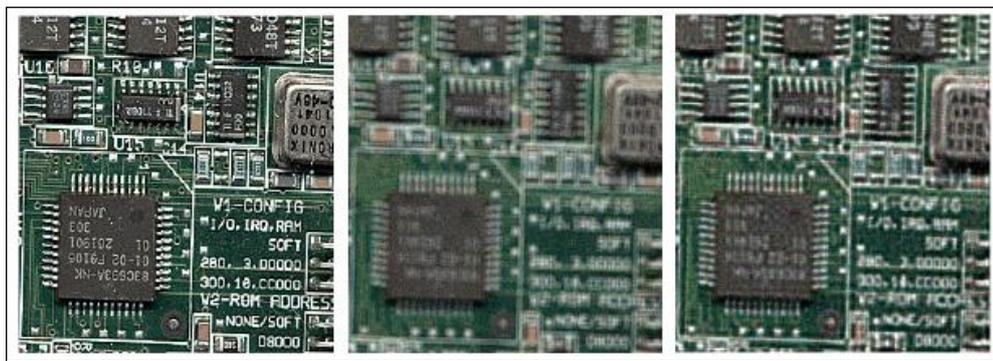


Figura 3.9 – Imagens: Original, com borrão de movimento e pós filtragem-inversa
 FONTE: *Modificado de (GONZALES & WOODS, 2001)*

3.3 – Medidas de Qualidade e Critérios de Fidelidade de Imagens

A restauração de imagens representa a reconstrução ou estimativa de uma imagem não corrompida a partir de sua versão distorcida. Há aplicações em diversas áreas, tais como medicina, exploração espacial e comércio (CHOY & CHAN, 1996).

O processo de filtragem de uma imagem tem por objetivo reconstruir uma imagem corrompida a partir de sua versão com ruído. No contexto de restauração de imagens, a expressão “qualidade de uma imagem” normalmente denota a sua fidelidade com relação à sua versão original sem ruído. Assim a aplicação de uma métrica de qualidade de imagem a um filtro digital bidimensional significa medir o aumento na qualidade da imagem devido à filtragem (CHOY & CHAN, 1996).

O objetivo da pesquisa em avaliação objetiva da qualidade de imagens é desenvolver métricas que possam prever automaticamente a qualidade da imagem. As métricas de desempenho são de grande importância numa ampla gama de aplicações, tais como aquisição de imagens, compressão, comunicação, restauração, análise, reprodução, impressão e marcas d’água (WANG & BOVIK, 2002).

Por meio das medidas de fidelidade discutidas a seguir, é estabelecido um nível de referência para sistemas e algoritmos de processamento de imagem. Esse nível de referência é utilizado neste trabalho como validação de resultados obtidos.

3.3.1 – Erro quadrático médio

O função erro é definido segundo a equação (3.13), na qual $f(x,y)$ é o nível do sinal individual da imagem original e $A(x,y)$ é o correspondente nível do sinal individual da imagem filtrada.

$$e(x,y) = (f(x,y) - A(x,y)) \quad (3.13)$$

Função erro

FONTE: (CONCI & AQUINO, 2005)

A partir da função erro, obtemos o MSE (KUMAR, 2011) segundo a equação (3.14), com N sendo o número de pixels na direção horizontal, e M o número de pixels na direção vertical. O MSE ideal se da em torno de 0.

$$MSE = \frac{1}{NM} \sum_{x=1}^N \sum_{y=1}^M (e(x,y))^2 \quad (3.14)$$

Erro Quadrático Médio

FONTE: (KUMAR, 2011)

3.3.2 – Relação sinal-ruído

A relação sinal-ruído (CONCI & AQUINO, 2005) é definida pela equação (3.15), com $A(x,y)$ sendo o nível do sinal individual da imagem filtrada e $e(x,y)$ a função erro definida anteriormente.

$$SNR_{rms} = \frac{\sum_{x=1}^N \sum_{y=1}^M (A(x,y))^2}{\sum_{x=1}^N \sum_{y=1}^M (e(x,y))^2} \quad (3.15)$$

Relação Sinal-Ruído

FONTE: (CONCI & AQUINO, 2005)

Trata-se de uma estimativa da qualidade de uma imagem reconstruída comparada com a mesma imagem original. A idéia básica é calcular um único número que reflete a qualidade

da imagem restaurada. Imagens reconstruídas com maiores métricas são consideradas melhores. Idealmente, as medidas de SNR não se equacionam com a percepção subjetiva humana. Basta lembrar que medidas de maior valor não representam imagens de maior qualidade.

3.3.3 – Relação sinal-ruído de pico

A relação sinal-ruído de pico (do inglês, PSNR) (CONCI & AQUINO, 2005) é definida pela equação (3.16) e expressa em *dB*. Trata-se de um termo da engenharia para a razão entre poder máximo de um sinal e o poder de corrupção de um ruído que interfere na fidelidade da representação daquele. A PSNR é comumente utilizada com a medida de qualidade de imagens reconstruídas após descompressão.

$$PSNR = 10 \log \frac{Max_i^2}{MSE} = 20 \log_{10} \frac{Max_i}{\sqrt{MSE}} \quad (3.16)$$

Relação Sinal-Ruído de Pico
 FONTE: (CONCI & AQUINO, 2005)

Pela equação (3.16), Max_i é o valor máximo possível de um pixel na imagem. Quando *pixels* são representados usando 8bits, o valor de Max_i é 255. Uma imagem idêntica à original resulta em uma PSNR indefinida de forma que o MSE se torna igual à zero. Quanto maior o valor da PSNR, maior a qualidade da imagem.

3.3.4 – Índice universal de qualidade da imagem

Existem basicamente duas classes para medição da qualidade de uma imagem. Matematicamente, a primeira é representada pelo MSE, o PSNR e o SNR. Já a segunda classe considera o Sistema Visual Humano (do inglês, HVS), incorporando a medida de qualidade de forma perceptiva do observador (PAPPAS & SAFRANEK, 1999)

Embora as classes citadas sejam atrativas pela baixa complexidade, (WANG & BOVIK, 2002) propuseram ainda o Índice Universal de Qualidade da Imagem (do inglês,

UIQI) com o intuito de resolver os problemas inerentes às outras métricas com relação aos resultados para alguns tipos de distorções. Para tanto, essa abordagem não depende das imagens que são testadas, das condições visuais e, tampouco, da análise de um observador. Apesar de proposto como índice de qualidade, o UIQI avalia a fidelidade entre imagens.

O índice faz uso de três componentes para o cálculo da diferença entre imagens. São eles a perda de correlação, a distorção de iluminância e a distorção de contraste. A expressão resultante do valor do UIQI é definida pela equação (3.17) a seguir.

$$Q = \frac{\sigma_{xy}}{\sigma_x \sigma_y} \frac{2xy}{(x)^2 + (y)^2} \frac{2\sigma_x \sigma_y}{\sigma_x^2 + \sigma_y^2} \quad (3.17)$$

Componentes do Índice Universal de Qualidade

FONTE: (WANG & BOVIK, 2002)

Sendo x e y a imagem original e a imagem de teste respectivamente, “o primeiro componente é o coeficiente de correlação entre x e y , o qual mede o grau de correlação linear entre x e y , variando entre -1 e 1. O melhor valor, 1, é obtido quando $y_i = ax_i + b$ para todo $i = 1, 2, \dots, N$, onde a e b são constantes e $a > 0$. Mesmo que x e y sejam linearmente relacionados, ainda podem haver distorções relativas entre eles, que são avaliadas no segundo e no terceiro componentes. O segundo componente, com valores entre 0 e 1, mede o quão próxima é a iluminância média entre x e y . O valor 1 é obtido se, e somente se, $\bar{x} = \bar{y}$. σ_x e σ_y podem ser vistos como a estimativa do contraste entre x e y e, dessa forma, o terceiro componente mede o quão similar é o contraste entre as imagens. O valor varia de 0 a 1 e o melhor resultado é obtido se, e somente se, $\sigma_x = \sigma_y$.” (WANG & BOVIK, 2002, tradução própria)

Apesar de não levar em consideração qualquer modelo HSV, o UIQI possui grande habilidade de medição de distorções estruturais ocorridas durante a degradação ou a reconstrução de imagens.

CAPÍTULO 4 – COMPRESSÃO DE IMAGENS DIGITAIS

Geralmente, hospitais e clínicas tomam por base para diagnósticos modernos as imagens geradas por raio-X, tomografia computadorizada (do inglês, CT) e ressonância magnética (do inglês, MR). Os dados brutos fornecidos por esses dispositivos de imagens normalmente ocupam grande espaço em disco. As imagens de diagnóstico para interpretação radiológica devem ser eficientemente armazenadas e transmitidas aos devidos destinos. É para acelerar a transmissão por via eletrônica e para minimizar espaço de armazenamento que as imagens médicas são frequentemente compactadas em arquivos de menor tamanho.

4.1 – Fundamentos da Compressão de Imagens

A compressão de dados se refere ao processo de reduzir, por meio de algoritmos, o tamanho físico de um bloco de informação, ou seja, trata o problema da quantidade de dados necessários para representar uma imagem digital. Em particular, na Computação Gráfica, existe o interesse em diminuir o tamanho de arquivos de imagens que precisam ser armazenados e/ou transmitidos por um canal de comunicação. Lembrando que dados são os meios pelos quais a informação é transmitida. Várias quantidades distintas de dados podem ser usadas para representar a mesma informação (WATANABE, 2005).

Geralmente, a compressão de imagens é possível em decorrência da coerência existente nos dados que a representam. Por exemplo, tomando um elemento de imagem (*pixel*) qualquer, provavelmente a cor desse pixel será igual a dos elementos vizinhos ou de outra região próxima na imagem, visto que há uma grande probabilidade de todos eles pertencerem a um mesmo objeto da imagem. Com base neste fato, com a redução ou eliminação das redundâncias, os diversos métodos de compressão visam produzir um código mais compacto que preserve as informações contidas na imagem.

A redundância é uma característica que está relacionada à distribuição estatística da informação presente na imagem. Sendo a redundância de dados um aspecto essencial no estudo de compressão de imagens digitais (MARTINS, 2002).

Segue, então, a equação (4.1), com n_1 e n_2 denotando o número de unidades portadoras de informação em dois conjuntos de dados que representam a mesma informação, sendo C_R a razão ou taxa de compressão.

$$C_R = \frac{n_1}{n_2} \quad (4.1)$$

Razão ou taxa de compressão
 FONTE: (MARQUES & VIEIRA, 1999)

“No caso em que $n_2 = n_1$, $C_R = 1$, podemos concluir que o primeiro conjunto de dados não contém nenhum dado redundante em relação ao segundo” (MARQUES & VIEIRA, 1999). Caso $n_2 \ll n_1$, a compressão de dados altamente redundantes é significativa. Já com $n_1 \ll n_2$, ocorre a expansão indesejada de dados, em virtude da pouca coerência entre *pixels*.

Em uma imagem digital existem, basicamente, três tipos de redundâncias de dados que podem ser identificadas e exploradas: redundância de codificação, redundância inter pixel e redundância psicovisual (MARQUES & VIEIRA, 1999)

Na redundância de codificação, há a atribuição de um código com tamanho variável (do inglês, VLC), em termos de número de bits. Os bits de um determinado tom de cinza que tem maior ocorrência são representados pelos códigos de comprimento menor. Ao contrário, se forem pouco frequentes, são representados por um código maior (MARTINS, 2002).

A redundância interpixel permite estimar o valor de um pixel pelo valor de seus vizinhos. Tal correlação está ligada ao relacionamento geométrico entre os objetos na imagem. Assim, a informação carregada por cada pixel individualmente é pequena. Uma variedade de nomes tem sido usada em referência a essas dependências entre os pixels, incluindo redundância espacial, redundância geométrica e redundância entre quadros (MARTINS, 2002).

Para reduzir as redundâncias interpixels em uma imagem, a matriz de pixels normalmente usada para a visualização e interpretação humana deve ser transformada para um formato mais eficiente, mas geralmente não visualizável. Por exemplo, as diferenças entre pixels adjacentes podem ser usadas para representar uma imagem. Transformações desse tipo, com remoção de redundância interpixel, são chamadas de mapeamentos. Se a imagem original puder ser reconstituída a partir da imagem transformada, então o mapeamento é chamado reversível (MARQUES & VIEIRA, 1999).

Sabendo-se que o olho humano não responde com a mesma sensibilidade a todas as informações visuais, é possível a definição de um grau de importância para tais, podendo eliminar algumas sem que haja prejuízo significativo na percepção da imagem. Trata-se da redundância psicovisual que é associada com a informação visual quantificável.

Considerando que a eliminação de dados psicovisualmente redundantes resulta em uma perda de informação quantitativa, esse procedimento é comumente referido por quantização. A operação traz perdas de informação de forma irreversível, o que resulta em uma compressão de dados com perda (MARQUES & VIEIRA, 1999).

4.2 – Modelo Genérico de Compressão de Imagens

As reduções da quantidade de dados necessária para a representação de uma informação ou imagem são tipicamente combinadas, formando sistemas práticos para compressão de imagens.

Um sistema de compressão é constituído por dois blocos estruturais distintos, o codificador e o decodificador. O primeiro é responsável pela redução ou eliminação de quaisquer redundâncias, podendo ser modelado por uma série de operações distintas. Cada uma é projetada para a redução de uma das três redundâncias (figura 4.1).

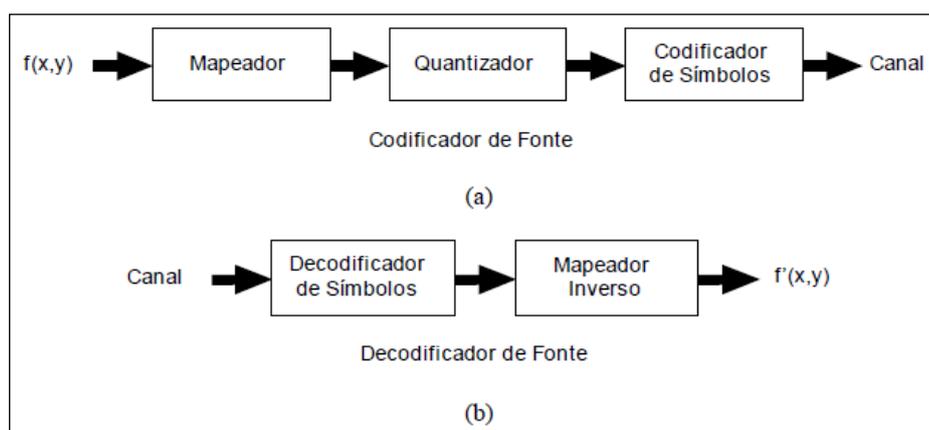


Figura 4.1 – Codificador (a) e decodificador (b) da fonte
 FONTE: Modificado de (JLS, 2003)

No primeiro estágio do codificador da fonte, o mapeador transforma os dados de entrada em um formato projetado para diminuir a redundância interpixel da imagem de entrada. Trata-se de uma operação ainda reversível. Já no segundo estágio, o quantizador reduz a precisão da saída do mapeador, de acordo com algum critério de fidelidade. A etapa diminui a redundância psicovisual da imagem original, sendo um processo irreversível e, portanto, omitido para compressões sem perdas.

No último estágio, o codificador de símbolos cria um código de comprimento fixo ou variável para representar a saída do quantizador e a mapeia de acordo com esse código. Trata-se de uma operação também reversível que reduz a redundância de codificação (JLS, 2003).

O decodificador de fonte, com apenas dois componentes, executa em ordem contrária as operações inversas dos blocos do codificador, exceto a quantização por ser irreversível.

Quando a informação própria é transferida entre uma fonte de informação e um usuário daquela informação, diz-se que a fonte de informação está conectada ao usuário de informação por um canal de informação. O canal é o meio físico que conecta a fonte ao usuário. Pode ser uma linha telefônica, um meio de propagação de ondas eletromagnéticas, ou um cabo entre dois computadores.

No caso apresentado (figura 4.1) e de acordo com a abrangência deste trabalho, não há referência ao codificador e ao decodificador de canal que exercem função caso canal de transmissão esteja contaminado por ruído ou sujeito a erro.

4.3 – Classificação para Compressão de Imagens

Uma classificação rígida dos vários métodos de compressão de imagens é muito difícil, uma vez que existe uma grande variedade de algoritmos e técnicas de compressão que utilizam princípios diferentes e que muitas vezes são utilizados simultaneamente.

De um modo geral, é possível distinguir entre as técnicas de compressão pela simetria, pela adaptabilidade e pela perda (MARTINO, 2002).

A primeira se refere à simetria da complexidade computacional entre os algoritmos de compressão e descompressão. Uma compressão simétrica possui algoritmos de codificação e decodificação associados ao mesmo tempo de processamento. Tal procedimento é

geralmente usado para aplicações que envolvem transmissão e visualização das imagens de forma simultânea (MARTINO, 2002).

Em uma compressão assimétrica, há um forte desbalanceamento do tempo de processamento entre a compressão e a descompressão. Por exemplo, em um procedimento de *backup* que tende priorizar uma compressão mais rápida em detrimento do tempo necessário para a operação inversa (MARTINO, 2002).

Em relação à classificação por adaptabilidade, fala-se de compressões não adaptativas, que se baseiam em dicionários estáticos, uma tabela pré-definida que associa símbolos a códigos de menor tamanho. Ao contrário, compressões adaptativas constroem seus dicionários em função dos dados de entrada (MARTINO, 2002).

Por fim, a classificação mais encontrada em literatura se refere à perda. No caso específico de compressão de imagens, convém notar que, em diversas circunstâncias práticas, a compressão deve obrigatoriamente ser sem perdas. Tal é o caso quando se aplicam técnicas de compressão a imagens médicas, nas quais eventuais perdas são indesejáveis e prejudiciais a uma avaliação posterior da imagem para diagnósticos.

Na situação citada, o método de compressão deve permitir a recuperação exata dos dados da imagem original. Diz-se, nesse caso, que o processo é reversível ou uma compressão sem perda (*lossless*). Caso contrário, ele é chamado irreversível, e fala-se de uma compressão com perda (*lossy*), isto é, não permitindo a reconstrução exata dos dados originais.

As técnicas ditas reversíveis são também chamadas de redução de redundância, ou sem ruído. As técnicas irreversíveis são conhecidas também por redução de entropia, ou com ruído (MARTINS, 2002).

A figura (4.2) apresenta uma taxionomia dos diversos métodos e algoritmos de compressão existentes (AGOSTINI, 2000). Destes métodos, são abordados neste trabalho os algoritmos de Huffman e o RLE, descritos adiante.

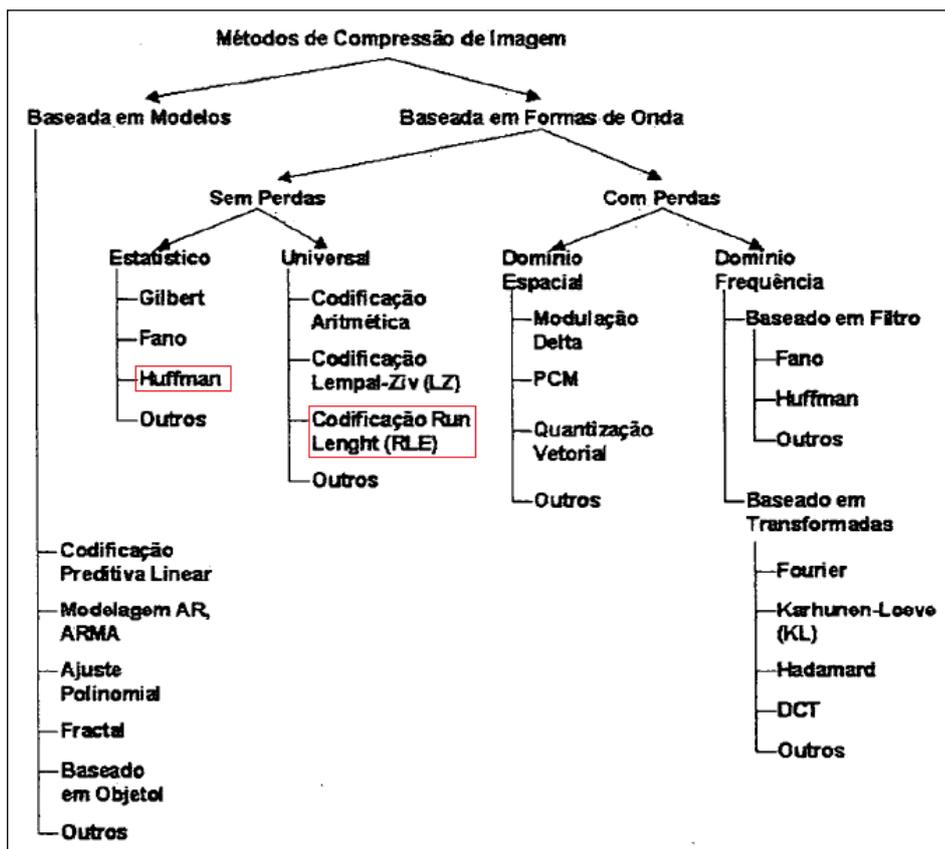


Figura 4.2 – Taxionomia dos métodos de compressão.

FONTE: Modificado de (MARTINS, 2002)

4.4 – Taxas de Compressão

A taxa de compressão é a razão entre os tamanhos da imagem comprimida e da imagem original. Esta taxa fornece uma medida de quanto menor ficou o arquivo após a aplicação da compressão. Em geral, é comum fornecer a taxa de compressão ou na forma de um vetor absoluto ou em percentual. Por exemplo, a taxa de compressão para uma imagem de 200 Kbytes que após ser comprimida ficou com 100 Kbytes de tamanho, é de 0,5 ou 50% (MELO, 1994).

São diversas as formas usadas para o cálculo das taxas de compressão, sendo que alguns cálculos são feitos em bits/bytes. Esta medida de compressão fornece o número de bits necessários por pixel depois da compressão. O valor bits por pixel (bpp) permite comparar os métodos de compressão e calcular o tamanho da imagem comprimida e a taxa de transmissão necessária para qualquer tamanho de imagem.

Outra forma se dá no uso de razões, por exemplo, 2:1. Isto significa que a quantidade de dados foi reduzida pela metade. Será utilizada neste trabalho a fórmula (em porcentagem) definida pela equação (4.2):

$$T_x = \left(1 - \frac{\text{dados comprimidos}}{\text{dados originais}}\right) \cdot 100 \quad (4.2)$$

Taxa de compressão

FONTE: (SANCHES, 2001)

Isto significa que um arquivo que não muda de tamanho após a compressão terá uma taxa de compressão de 0% (zero por cento). Um arquivo comprimido pela metade do seu tamanho original terá uma taxa de compressão de 50%. Na teoria, a taxa de compressão máxima é 100%. Isto só ocorrerá se o tamanho do arquivo comprimido for igual a zero, o qual somente ocorrerá se o tamanho dos dados originais também for zero. Se o tamanho do arquivo comprimido for maior do que o arquivo original, teremos uma taxa de compressão negativa (MARTINS, 2002).

Em se tratando de taxas de compressão, apesar dos métodos de compressão com perdas não permitirem a reconstrução da versão original dos dados de entrada, é possível obter taxas de compressão mais elevadas, chegando a taxas de 10:1 até 20:1. Técnicas de compressão sem perdas, em virtude da qualidade necessária no retorno aos dados originais, possuem taxas de compressão comparativamente baixas (SANCHES, 2001).

4.5 – Compressão sem Perdas ou Compactação da Imagem

Compressão de dados sem perdas é o método de compressão que permite a recuperação exata dos dados originais após o processo de descompressão. Como dito anteriormente, os esquemas de compressão de imagens sem perdas frequentemente consistem em dois componentes distintos e independentes: modelagem e codificação.

A compressão máxima possível quando se codifica dentro de um mesmo alfabeto é representado pela entropia, termo usado na teoria da informação. Todos os algoritmos sem perdas requererão igual ou mais bits para a compressão (SANCHES, 2001).

4.5.1 – Teoria da informação

A teoria da Informação oferece a base matemática para a seguinte questão: Qual a menor quantidade de dados possível necessária para a representação de uma imagem?

Assumindo que a fonte gera uma sequência aleatória de símbolos pertencentes a um conjunto enumerável, o conjunto de símbolos da fonte $S = \{\sigma_1, \sigma_2, \dots, \sigma_L\}$ é chamado de alfabeto A da fonte. Um vetor $z = [P_1, P_2, \dots, P_L]$ representa o conjunto de todas as probabilidades dos símbolos da fonte. O par (S, z) descreve a fonte da informação. A média dessa por símbolo é dada pela equação (4.3) seguinte (SANCHES, 2001).

$$H = -\sum_{i=1}^L P_i \log_2 P_i = \sum_{i=1}^L P_i \log_2 \frac{1}{P_i} \quad (4.3)$$

Entropia

FONTE: (SANCHES, 2001)

H é o valor chamado de entropia ou incerteza da fonte. À medida que se aumenta o valor, maior a incerteza e mais informação é associada à fonte. Com símbolos equiprováveis, é maximizada a entropia e a fonte fornece a maior média de informação por símbolo. A equação (4.4) seguinte é o número médio de bits necessários para codificação não ambígua de cada símbolo de um alfabeto fonte.

$$\log_2 \frac{1}{P_i} \quad (4.4)$$

Número de bits para codificação não ambígua de símbolos

FONTE: (SANCHES, 2001)

De acordo com o Teorema da Codificação sem Ruído de (CHIU et. al, 2001), é possível codificar sem perdas, a informação gerada por uma fonte com entropia de H bits, usando em média $H+\epsilon$ bits por mensagem, onde $\epsilon \geq 0$ é uma quantidade arbitrariamente pequena. Se o tamanho dos dados é n , então $nH(S)$ é o número de bits requerido para representar S e é então chamado a complexidade de S . Caso após um processo de

descorrelação os dados obtidos não forem uniformemente distribuídos, a entropia será inferior aos n bits por *pixel* da imagem original.

O cálculo da entropia é também chamado de estimativa de primeira ordem indica o limite que se consegue obter apenas com codificação de comprimento variável. Essa é usada para reduzir redundâncias de código. A diferença entre a estimativa da entropia de mais alta ordem e a estimativa de primeira ordem indica a presença de redundâncias interpixel (JLS, 2003).

O método de estimativa de primeira ordem se dá pela construção de um modelo baseado na frequência relativa de ocorrência de cada símbolo na imagem (tabela 4.1).

Tabela 4.1 – Imagem genérica 4x8 com 256 tons de cinza

22	22	22	95	167	234	234	234
22	22	22	95	167	234	234	234
22	22	22	95	167	234	234	234
22	22	22	95	167	234	234	234

FONTE: Modificado de (JLS, 2003)

Levantando as probabilidades de cada nível de cinza na imagem proposta, segue tabela (4.2):

Tabela 4.2 – Probabilidade de cada nível de cinza em imagem.

Nível de cinza	Número de ocorrências	Probabilidade
22	12	3/8
95	4	1/8
167	4	1/8
234	12	3/8

FONTE: Modificado de (JLS, 2003)

Entrando com as probabilidades obtidas na equação (4.4), é obtida uma entropia de 1,81 bits/*pixel*, o que representa uma entropia total da fonte de aproximadamente 58 bits.

Uma estimativa de segunda ordem seria um exame da frequência relativa de blocos de dois *pixels* na imagem. Assumindo uma conexão linha a linha e do final ao início, as frequências dos pares de *pixels* são computadas como na tabela (4.3):

Tabela 4.3 – Probabilidade de par de tons em imagem.

Par de tons de cinza	Número de ocorrências	Probabilidade
(22,22)	8	1/4
(22,95)	4	1/8
(95,167)	4	1/8
(167,234)	4	1/8
(234,234)	8	1/4
(234,22)	4	1/8

FONTE: Modificado de (JLS, 2003)

A estimativa de entropia resultante do uso da equação (4.4) será $2,5/2 = 1,25$ bits/*pixel*, na qual a divisão por dois é uma consequência da consideração de dois *pixels* de cada vez.

Desse modo, o conhecimento da entropia de uma imagem, nos fornece um parâmetro importante de entendimento como o limite mínimo de representação de um *pixel*, i.e., a taxa máxima de compressão, atingida apenas com redução de redundância de codificação. Resultados de estimativas de maiores ordem sugerem taxas maiores que podem ser obtidas com a diminuição da redundância interpixel. Aplicando métodos de compressão com perdas pode-se produzir uma taxa média de bits inferior à entropia da imagem (SANCHES, 2001).

4.6 – Codificação e Compactação de Imagens

Existem diversos padrões de compressão de imagens sem perdas que fazem uso das técnicas *lossless* de codificação para realizar compressão de imagens e possuem taxas de compressão relativamente baixas em comparação com técnicas de compressão com perdas, *lossy* (RABBANI, 2002).

A correlação entre os pixels de uma imagem indica quanta informação sobre um determinado *pixel* os outros *pixels* possuem. Uma imagem com um alto grau de correlação entre os seus pixels pode ser armazenada ou transmitida com menor quantidade de bits. A redundância de informação em uma imagem está diretamente relacionada com a correlação existente entre os *pixels* dessa imagem.

As técnicas de compressão sem perdas, ou compactação, de imagens procuram representá-las com parâmetros de pouca correlação, obtidos por meio de algoritmos aplicados à imagem. Esses devem possuir um processamento completamente reversível, para que a imagem original possa ser recuperada dos dados por ele gerados.

Neste trabalho serão abordados alguns dos algoritmos/codificações mais importantes que são aplicados em imagens digitais para uma compressão *lossless*, citando em quais situações apresentam vantagens ou desvantagens e sob quais condições atingem seu desempenho ótimo.

4.6.1 – Código de Huffman

“O algoritmo foi publicado por David A. Huffman em 1952. É uma codificação sem perdas, usada em compressores populares. O código frequentemente aparece como o último estágio do regime de compressão com perdas, entre outros nos formatos JPEG e MPEG” (HUFFMAN, 2010, tradução própria). Trata-se de uma técnica simples em desenvolvimento e com baixo custo de memória.

Algoritmos que se baseiam na distribuição de probabilidades dos símbolos da fonte associam códigos de pequeno comprimento a *pixels* altamente frequentes e códigos de comprimento mais longo a *pixels* menos frequentes. Estes são genericamente denominados

VLC. Reduzem significativamente o comprimento médio efetivo, em bits, dos símbolos da fonte a serem armazenados ou transmitidos (MARTINS, 2002).

O código de Huffman segue a chamada regra de prefixo, isto é, nenhuma palavra código deverá ser o prefixo de nenhuma outra palavra código. Tal fato significa que estas palavras são inequivocamente decodificáveis, por mais que possuam comprimento variável (MARQUES & VIEIRA, 1999).

Para a codificação, dado um conjunto de símbolos com suas respectivas probabilidades de ocorrência associadas, é necessária a elaboração de uma árvore binária. Esta consiste em uma estrutura composta por nós e ramos e possui as seguintes características (MARQUES & VIEIRA, 1999):

- Todo nó depende unicamente de outro, em exclusão do nó chamado raiz ou inicial;
- Todo nó possui dependentes, exceto os chamados externos ou terminais;
- A construção da árvore é efetuada a partir dos nós terminais;
- Os símbolos são ordenados de forma decrescente de suas probabilidades, sendo indiferente a posição desses, caso ocorram probabilidades iguais.

Tomam-se, primeiramente, os dois símbolos menos prováveis – nós externos, identificando um nó com um bit 0 e outro com bit 1. Juntam-se os dois símbolos em um novo nó interno, cuja probabilidade de ocorrência será a soma das probabilidades dos símbolos anteriores. Repete-se, então, o mesmo procedimento de ordem dos símbolos, do mais provável para o menos provável, realizando novamente todos os passos até restarem somente dois nós, que formam o nó raiz com probabilidade igual a 1 (ANAHT, 1999).

Como mostra a figura (4.3), é obtido o valor 0,1 pela soma das probabilidades 0,07 e 0,03 dos símbolos s_5 e s_6 . Este pode ser posicionado antes ou depois da probabilidade de valor 0,1 do símbolo s_4 .

Símbolo	Probabilidade	Passo 1	Passo 2	Passo 3	Passo 4	Passo 5	Código
s_1	0,40	0,40	0,40	0,40	0,40	1	1
s_2	0,25	0,25	0,25	0,25	1	0,60	01
s_3	0,15	0,15	0,15	1	0,35	0	001
s_4	0,10	0,10	1	0,20	0		0001
s_5	0,07	0	0,10	0			00000
s_6	0,03	1					00001

○ Nó

Figura 4.3 – Construção das palavras código do Algoritmo de Huffman
 FONTE: Modificado de (MARQUES & VIEIRA, 1999)

Com a árvore elaborada, são geradas as palavras código formadas com início no nó raiz em direção aos terminais. Inicia-se com a associação de 0, por convenção, à probabilidade maior, e de 1 à outra probabilidade como ilustrado ainda na figura (4.3), com um 0 associado à probabilidade 0,6 e um 1 à probabilidade 0,4, no passo 4 (MARQUES & VIEIRA, 1999).

O valor 1 associado à 0,4 permanece inalterado, visto que tal probabilidade não foi decomposta. Como o símbolo de probabilidade 0,6 foi gerado a partir da combinação de dois outros símbolos na fonte reduzida à sua esquerda, o 0 usado para codificá-lo é agora atribuído a ambos os símbolos que lhe deram origem, adicionando-se mais um 0 ou 1 à direita de cada símbolo, seguindo a mesma convenção para distingui-los. O procedimento continua até o retorno aos nós das probabilidades dos símbolos. Neste ponto as palavras código ficam definidas como sendo o valor obtido pela concatenação dos valores do nó raiz até o nó terminal do símbolo em questão.

O Código de Huffman obtém máxima eficiência de compressão quando as probabilidades dos símbolos são potências de $\frac{1}{2}$. Nesse caso, a taxa de compressão atinge o valor da entropia, com redundância de código (diferença entre comprimento médio do código e entropia) igual à zero (MARQUES & VIEIRA, 1999).

A eficiência do código E é dada pela equação (4.5), sendo $H(X)$ a entropia da fonte e N o comprimento médio das palavras código. Já a redundância do código é dada pela diferença entre N e $H(X)$.

$$E = \frac{H(X)}{N} \quad (4.5)$$

Eficiência do código de Huffman
 FONTE: Modificado de (MARQUES & VIEIRA, 1999)

Com uso da equação (4.3), é possível calcular o valor da entropia para o conjunto de probabilidades da figura (4.3), resultando em 2,19183 bits/símbolo. Já o comprimento médio do código é obtido de acordo com o cálculo a seguir (equação 4.6), sendo k a quantidade total de níveis de cinza distintos na imagem, Qn_k a quantidade de bits para a representação de cada nível de cinza e $p(n_k)$ a probabilidade de ocorrência de cada nível de cinza:

$$N = \sum_{i=1}^k Q_{n_i} p(n_i) \quad (4.6)$$

$$k = 6$$

$$N = 1.(0,40) + 2.(0,25) + 3.(0,15) + 4.(0,10) + 5.(0,07) + 5.(0,03) = 2,25 \text{ bits}$$

Comprimento médio do código gerado pelo Algoritmo de Huffman

FONTE: *Modificado de (MARQUES & VIEIRA, 1999)*

É possível obter uma taxa de compressão ainda melhor quando se codificam pares de símbolos ao invés de um símbolo de cada vez. Porém, há o inconveniente de que a quantidade de palavras código cresça exponencialmente com o aumento da quantidade de símbolos codificados como n-gramas (símbolos compostos de n símbolos simples) (MARQUES & VIEIRA, 1999).

A principal desvantagem prática do código de Huffman é a necessidade de se armazenar ou transmitir a tabela de símbolos da fonte e respectivas probabilidades juntamente com os dados codificados. Em geral, algoritmos VLC necessitam da criação de uma tabela de relacionamento entre os dados compactados e os dados originais, e vice-versa. Essa deve ser incorporada ao arquivo compactado para posterior decodificação (MARQUES & VIEIRA, 1999).

4.6.2 – Código por comprimento corrido

O Código de Huffman é um dos principais métodos de remoção de redundância de codificação. Considera-se a seguir uma das várias técnicas de compressão de imagens que busca explorar redundâncias interpixel.

“A *Run-Length Encoding* (do inglês, RLE) ou Codificação por Comprimento Corrido, [...], é uma maneira simples para comprimir ocorrências seguidas do mesmo byte. Um número é usado para indicar os mesmos dados que aparecem de forma múltipla. Menos bytes são necessários para expressar a mesma informação. RLE é um esquema de compressão *lossless* simples, mas é limitada a uma compressão de 2:1 ou 3:1. [...]” (KAUR & KAUSHAL, 2008, tradução pessoal)

A RLE é uma técnica de compactação de dados sem perdas que elimina a repetição consecutiva de um determinado caractere contido na informação. Esse método tem a vantagem de eliminar a necessidade da criação de tabelas de relacionamento, incorporadas ao arquivo compactado no caso de algoritmos VLC. O desenvolvimento de arquiteturas de

compactação de dados com uso da técnica RLE é frequente em compressão de dados sem perdas (ANAHIT, 1999).

Para imagens binárias, o código *Run-Length* atua fazendo com que sequências de *pixels* consecutivos com um mesmo valor sejam codificadas em uma palavra código que representa o comprimento desta sequência. O primeiro caractere representa a quantidade de caracteres idênticos e consecutivos e o seguinte, o próprio caractere. Diz-se que algoritmo apresenta bons resultados de compactação para imagens com grande coerência de tons (TORRES, 2002).

Como exemplo, suponha a imagem binária mostrada na figura (4.4a).

0000000000000000		16 0;		
0000000000000000		16 0;		
0001111111111000		3 0; 10 1; 3 0;	35 0; 10 1;	
0001111111111000		3 0; 10 1; 3 0;	6 0; 10 1;	
0001111111111000		3 0; 10 1; 3 0;	6 0; 10 1;	
0000001111000000		6 0; 4 1; 6 0;	9 0; 4 1;	35 10 6 10 6 10 9 4 9 4 9 4 9 4 22
0000001111000000		6 0; 4 1; 6 0;	9 0; 4 1;	
0000001111000000		6 0; 4 1; 6 0;	9 0; 4 1;	
0000001111000000		6 0; 4 1; 6 0;	9 0; 4 1;	
0000001111000000		6 0; 4 1; 6 0;	9 0; 4 1;	
0000000000000000		16 0;	22 0;	

(a) (b)

Figura 4.4 – Imagem Binária (a) RLE por linhas (b)

FONTE: Modificado de (ANAHIT, 1999)

Dessa forma, entradas de dados que não possuam repetição de caracteres podem terminar por aumentar de tamanho. Para tal, é possível ainda eliminar caracteres que representem quantidades iguais a 1. Caso não haja qualquer coerência de caracteres, não haverá alteração no tamanho original.

Ainda de acordo com a figura (4.5), nota-se a variante em linhas na aplicação do algoritmo. A figura (4.5) a seguir mostra outras variantes que podem ser usadas na codificação por comprimento corrido.

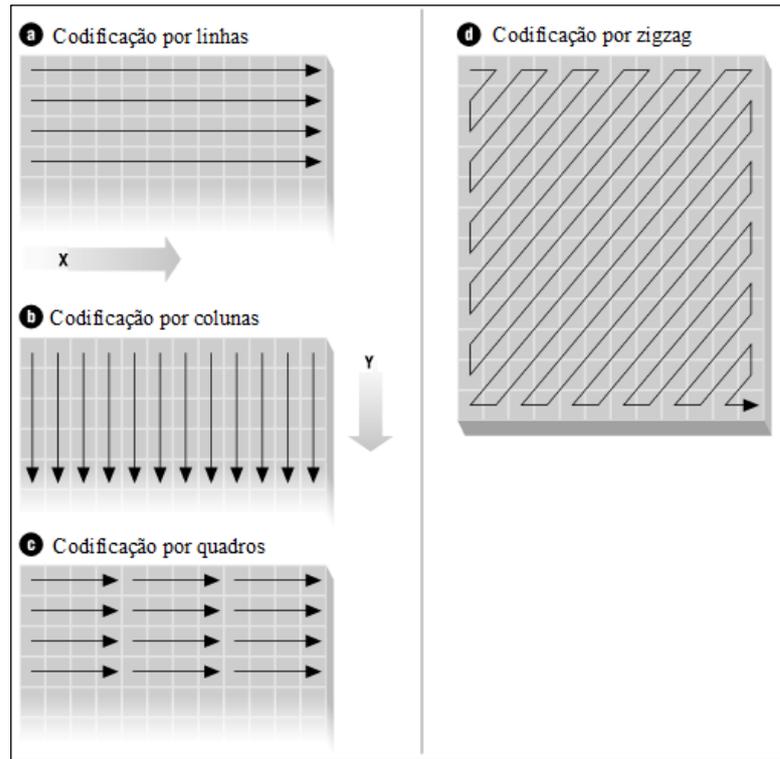


Figura 4.5 – Variantes RLE: (a) linhas, (b) colunas, (c) quadros e (d) zigzag
FONTE: *Modificada de (RLE, 2010)*

O RLE deve ser utilizado quando a informação a ser compactada contiver caracteres idênticos com repetição consecutiva. O algoritmo é para compressão simétrica, sem perdas e adaptativa e esse pode ser usado também como uma pré-compactação para outro método (MARTINO, 2002).

CAPÍTULO 5 – MODELO PROPOSTO

Este capítulo apresenta a codificação, os testes e os resultados obtidos durante o desenvolvimento do projeto, além da simulação de seu funcionamento.

5.1 – Apresentação Geral

As imagens obtidas para uso no projeto foram cedidas pelo Dr. Pedro Renato de Paula Brandão – Hospital das Clínicas/Neurologia/USP – Ribeirão Preto, o qual garante a validade e originalidade de tais imagens, advindas de exames de ressonância magnética e de tomografia computadorizada. As imagens diagnósticas usadas neste trabalho estão disponíveis no formato DICOM e são convertidas para os formatos TIFF (.tif ou .tiff) e Bitmap (.bmp). São utilizadas imagens genéricas para fins de testes e demonstração de resultados com melhor visibilidade. Estas são de resoluções baixas e com variadas intensidades de cinza.

O modelo proposto trata da interface gráfica de usuário (do inglês, GUI), desenvolvida na ferramenta MATLAB®, com funcionalidades de simulação de diversos tipos de ruídos, adicionando-os às imagens. Os ruídos simulados são o gaussiano e o impulsivo, também conhecido por sal e pimenta. Em adição, há a possibilidade de inserção de borrões de movimento na imagem a partir de uma angulação e um comprimento.

A partir da simulação do ruído, a GUI possui a funcionalidade de restauração da imagem por meio de filtros espaciais e também por filtragem inversa, não incluindo sua forma adaptativa, e filtragem de Wiener. Não são incluídas todas as formas de restauração de imagem existentes, selecionando-se nos resultados, dentre as desenvolvidas, aquela que possui melhor desempenho para cada caso. Os filtros espaciais utilizados para comparação de resultados são: média aritmética, mediana; valor máximo e valor Mínimo.

Como funcionalidades adicionais, a interface abrange as codificações de Huffman e RLE como parte para compressões de dados sem perda. Não se contempla todo o processo de compressão, mas sim a codificação e a decodificação das imagens, visando à redução de dados necessários para representação dessas e a posterior reconstrução desses dados de forma idêntica à original.

A qualidade da imagem restaurada em relação à imagem original será avaliada em outra funcionalidade, na qual se apresentam os cálculos dos valores de Erro Quadrático Médio (do inglês, MSE), Relação Sinal-Ruído (do inglês, SNR), Relação Sinal-Ruído de Pico (do inglês, PSNR) e o Índice Universal de Qualidade da Imagem (do inglês, UIQI).

A figura (5.1) mostra o fluxograma do modelo proposto, partindo do ponto no qual as imagens diagnósticas, tanto de MR quanto de CT já foram adquiridas por vias de exames.

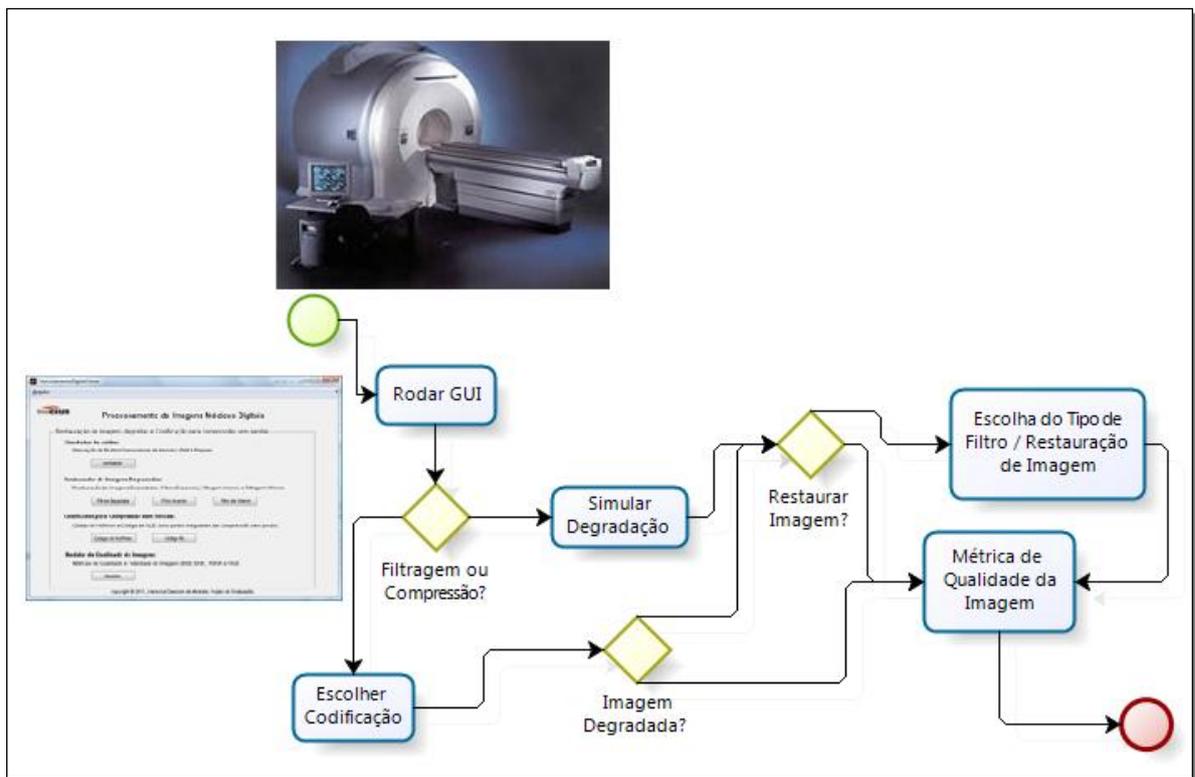


Figura 5.1 – Fluxograma do modelo proposto
FONTE: (a autora)

5.2 – Implementação

A ferramenta MATLAB® foi escolhida por fazer uso de uma linguagem com boa quantidade de funções para computação matemática. Ocorre a integração entre a computação, a visualização e a programação em um ambiente de fácil uso, no qual problemas e soluções são expressos em notações matemáticas familiares. Os usos típicos incluem a análise de dados, desenvolvimento de aplicações de engenharia, além da construção de interfaces gráficas do usuário.

Trata-se de um sistema interativo cujo elemento básico se dá por um array que não necessita de dimensionamento. Isso permite a solução de diversos problemas computacionais, especialmente os que envolvem formulações com matrizes, o que é o caso no processamento de imagens.

Para minimizar o tempo requerido na realização de pesquisas em binários, nos casos de codificação para compressão de Huffman, fez-se uso de códigos em C, disponíveis na biblioteca DIPUM, para manipulação das representações de dados com tamanho variável, visto que o MATLAB® possui melhor processamento de matrizes de tamanho uniforme (MATHWORKS, 2011). Tais códigos foram necessários porque o VLC é um dos pilares da codificação para compressão de imagens apresentadas neste trabalho.

Com uso do MATLAB® para processamento de imagens, deve-se ter em mente diversos aspectos como o carregamento das imagens, o uso do formato correto e a conversão entre os formatos disponíveis, a forma de salvar a informação em diferentes tipos de dados e a maneira de visualização em tela das imagens. São diversos os comandos específicos para cada um dessas operações e em sua maioria, necessitam do toolbox de processamento de imagens. Para este trabalho, é utilizado o *Image Processing Toolbox 7.0*, referente à *release 2010a*.

Com uso do ambiente de desenvolvimento de interface gráfica do MATLAB®, o GUIDE, foram gerados os arquivos .m que controlam o funcionamento da interface. São eles:

- *ProcessamentoDigitalHome.m*: Controle para tela inicial da interface;
- *GUISimulador.m*: Controle para simulação de ruídos Gaussiano e Impulsivo, além de borrões de movimento;
- *GUICompare.m*: Controle para comparação de resultados entre os filtros espaciais de média aritmética, mediana, valor máximo e valor mínimo;
- *GUIRestaura.m*: Controle para os filtros inverso e de Wiener;
- *GUIHuffman.m*: Controle para codificação e decodificação de Huffman;
- *GUIRle.m*: Controle para codificação e decodificação por RLE;
- *GUIAnalisar.m*: Controle para cálculos de qualidade da imagem;

A tela inicial da interface é mostrada na figura (5.2) a seguir. Encontra-se disponível o acesso a todas as demais telas/funcionalidades da aplicação, constando breve descrição de cada uma.



Figura 5.2 – Tela inicial da GUI.

FONTE: (a autora)

Acessando a funcionalidade do Simulador (figura 5.3), visualiza-se o espaço para o carregamento da imagem original, os campos para definição dos parâmetros de borrão ou de ruído, o botão para ativar a operação de degradação da imagem e o espaço para visualização da imagem degradada. Esta fica salva em memória para que seja possível seu carregamento nas demais telas da GUI.

Após a degradação da imagem, é habilitada a visualização dos histogramas de ambas as imagens. Os campos SNR, MSE, PSNR e UIQI na lateral do campo para a imagem degradada informam os parâmetros de qualidade da imagem degradada em relação à imagem original.

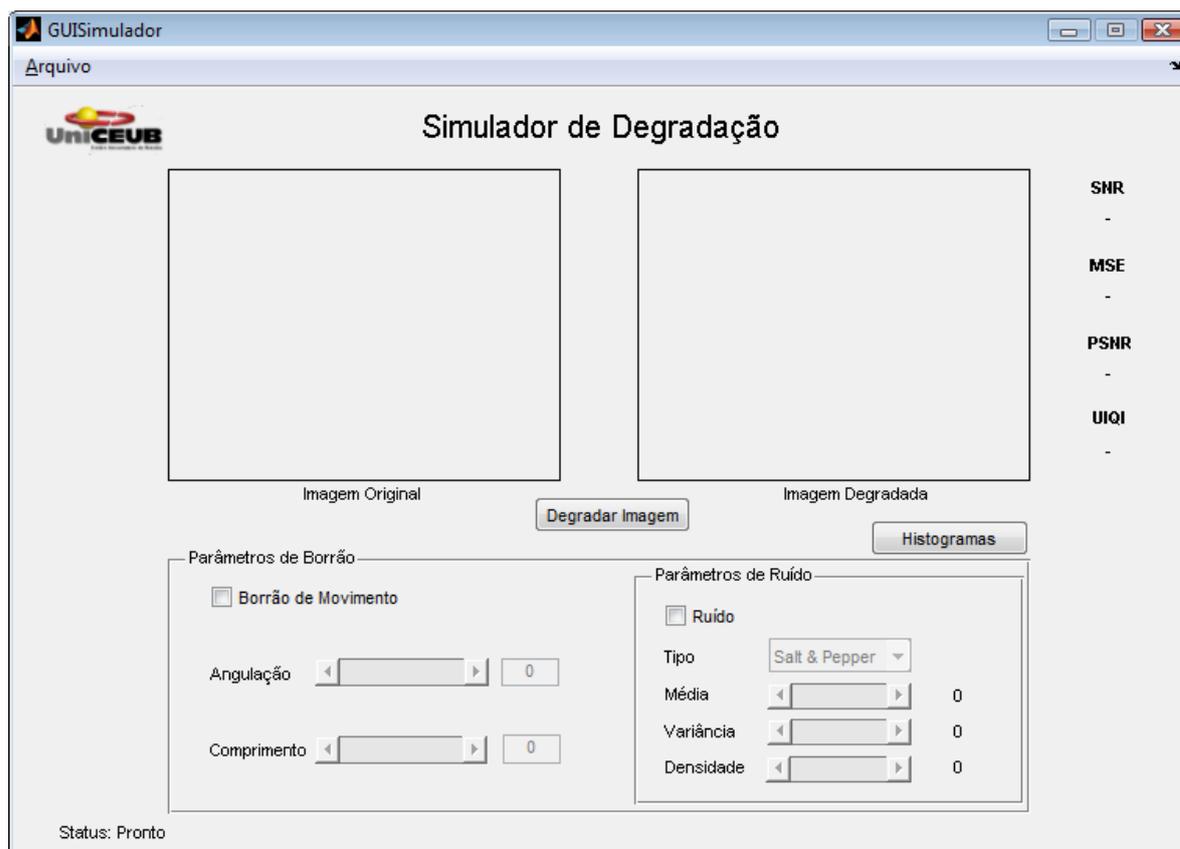


Figura 5.3 – Tela do Simulador de Degradação (Simulador.fig)
 FONTE: (a autora)

Na tela de filtros espaciais (GUICompare.fig) (figura 5.4), após a degradação da imagem de acordo com os parâmetros de simulação desejados, é possível carregá-la no espaço de “Imagem Degradada”. À medida que se pressionam os botões referentes à cada tipo de filtro, as imagens restauradas são carregadas nos respectivos espaços para visualização. São preenchidos automaticamente os campos SNR e MSE validando a qualidade da imagem restaurada perante a imagem original. Os mesmos campos correspondentes à imagem degradada são parâmetros comparativos entre essa e a imagem original anterior à degradação.

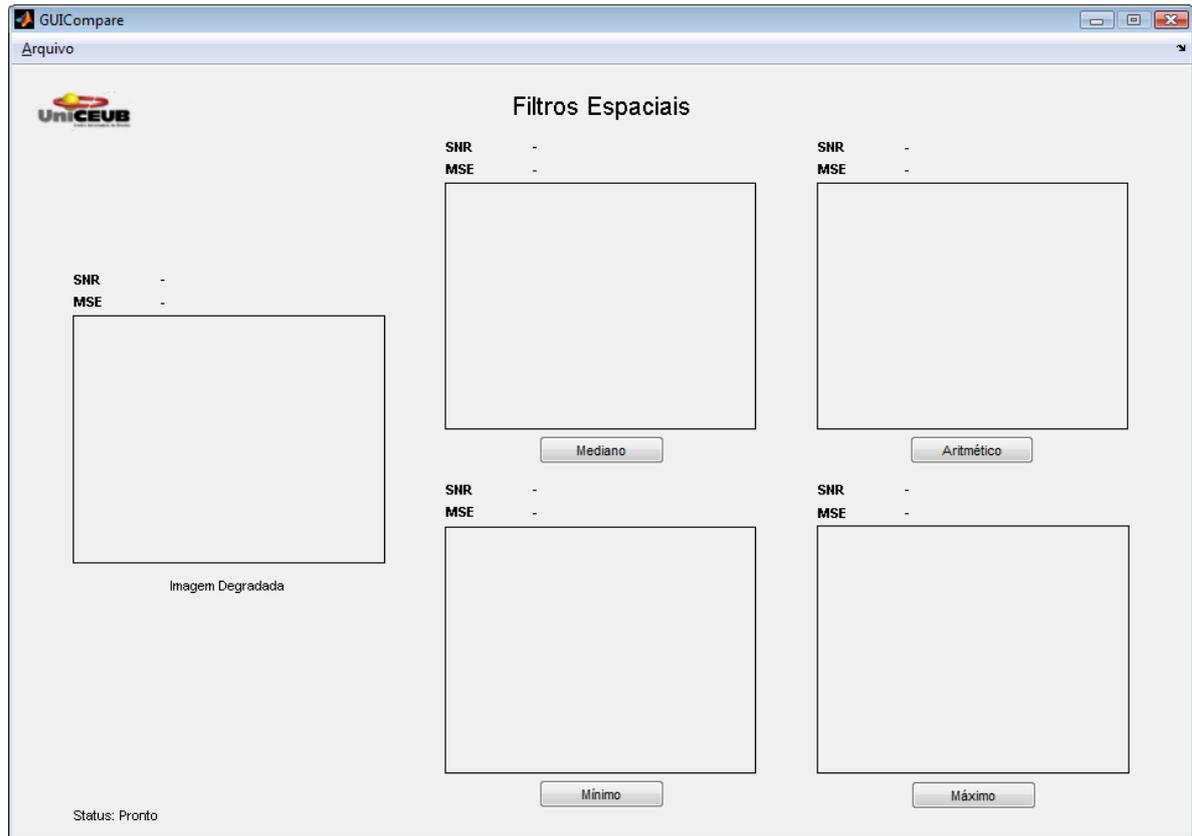


Figura 5.4 – Tela de comparação dos filtros espaciais (GUICompare.fig)
 FONTE: (a autora)

Ainda após a degradação da imagem, é possível a escolha da filtragem inversa ou da filtragem de Wiener. As figuras 5.5 e 5.6 a seguir mostram as telas referentes aos filtros respectivamente. Assim como nas funcionalidades anteriores, é no espaço “Imagem Degradada” onde será carregada a imagem previamente degradada. Em “Parâmetros do Borrão” são visíveis os parâmetros de angulação e de comprimento para o caso de existência de borrão de movimento na degradação da imagem.

O botão “Restaurar” inicia a operação de restauração da imagem, cujo resultado é carregado no espaço “Imagem Restaurada”, o qual é guardado em memória para posterior validação de qualidade. O espaço de “Zoom” permite uma visualização mais próxima dos *pixels* da imagem restaurada.

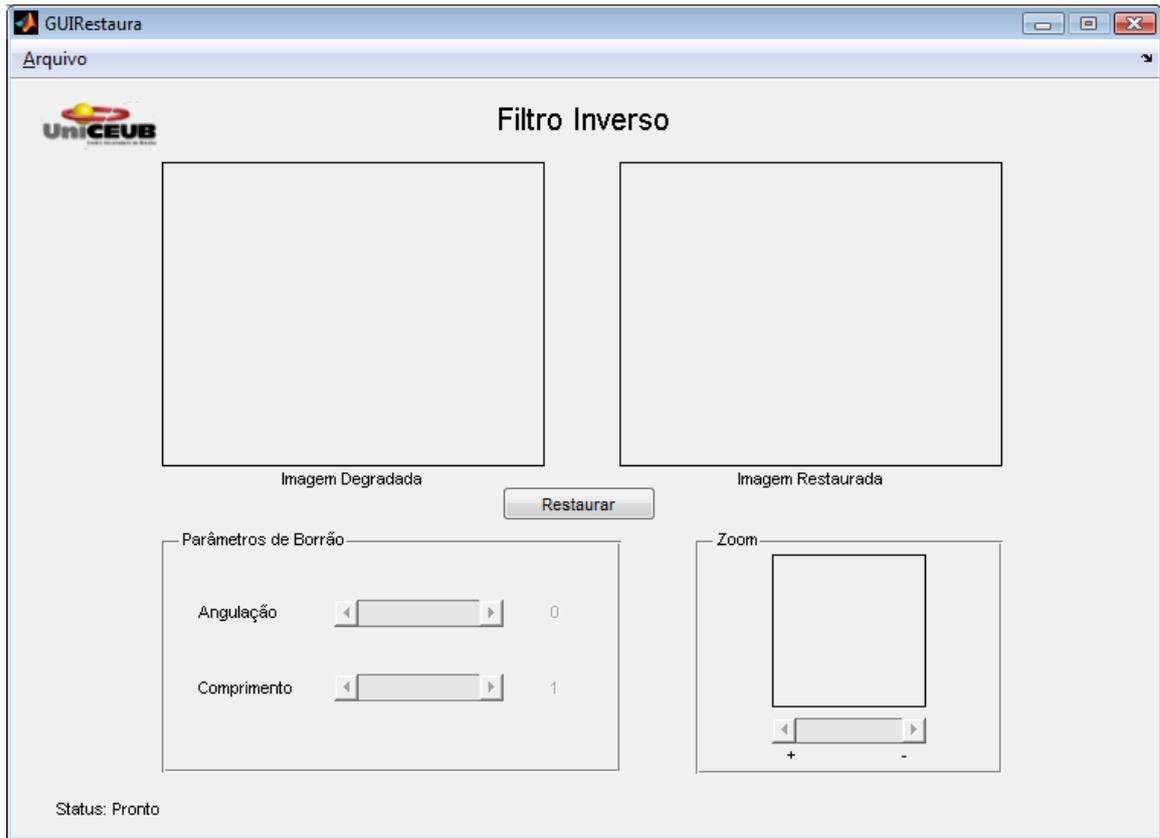


Figura 5.5 – Tela do filtro inverso (GUIRestaura.fig)
 FONTE: (a autora)

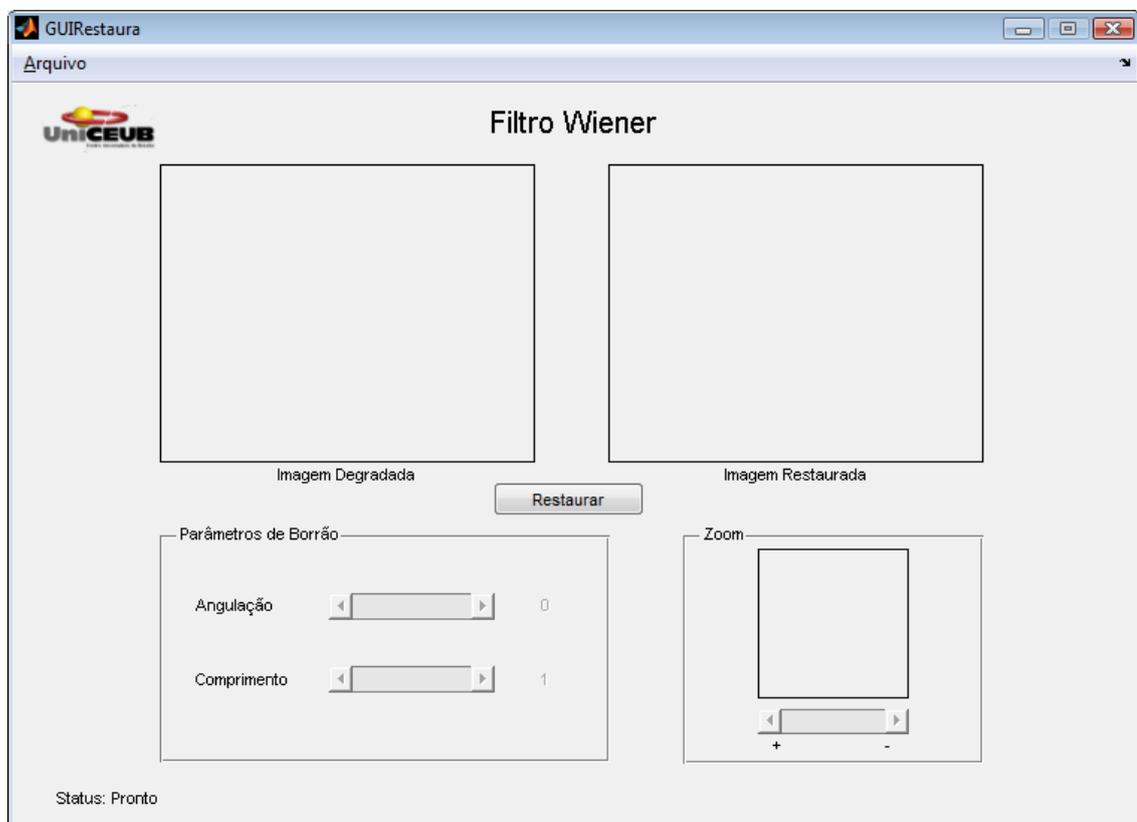


Figura 5.6 – Tela do filtro de Wiener (GUIRestaura.fig)
 FONTE: (a autora)

As funcionalidades de codificação para compressão sem perdas das imagens por Código de Huffman e por RLE são controladas a partir das telas mostradas pelas figuras 5.7 e 5.8 respectivamente. Para ambas, existe a opção de codificação em imagens de teste genéricas, de baixa resolução e que possibilitam a visualização de pixels, da matriz de intensidade e das palavras código após a compressão. A opção deve ser marcada no *checkbox* “Imagem de teste”.

Para a codificação e decodificação de Huffman, cujas operações são controladas pelos botões “Codificar Imagem” e “Decodificar Imagem”, os resultados obtidos e apresentados ao usuário são a entropia da imagem ou taxa máxima de compressão, a porcentagem removida, o nível de redundância de código da imagem, a eficiência do código e a quantidade de bits por *pixel*. Os campos “Tamanho” e “Níveis de cinza” caracterizam a imagem a ser codificada.

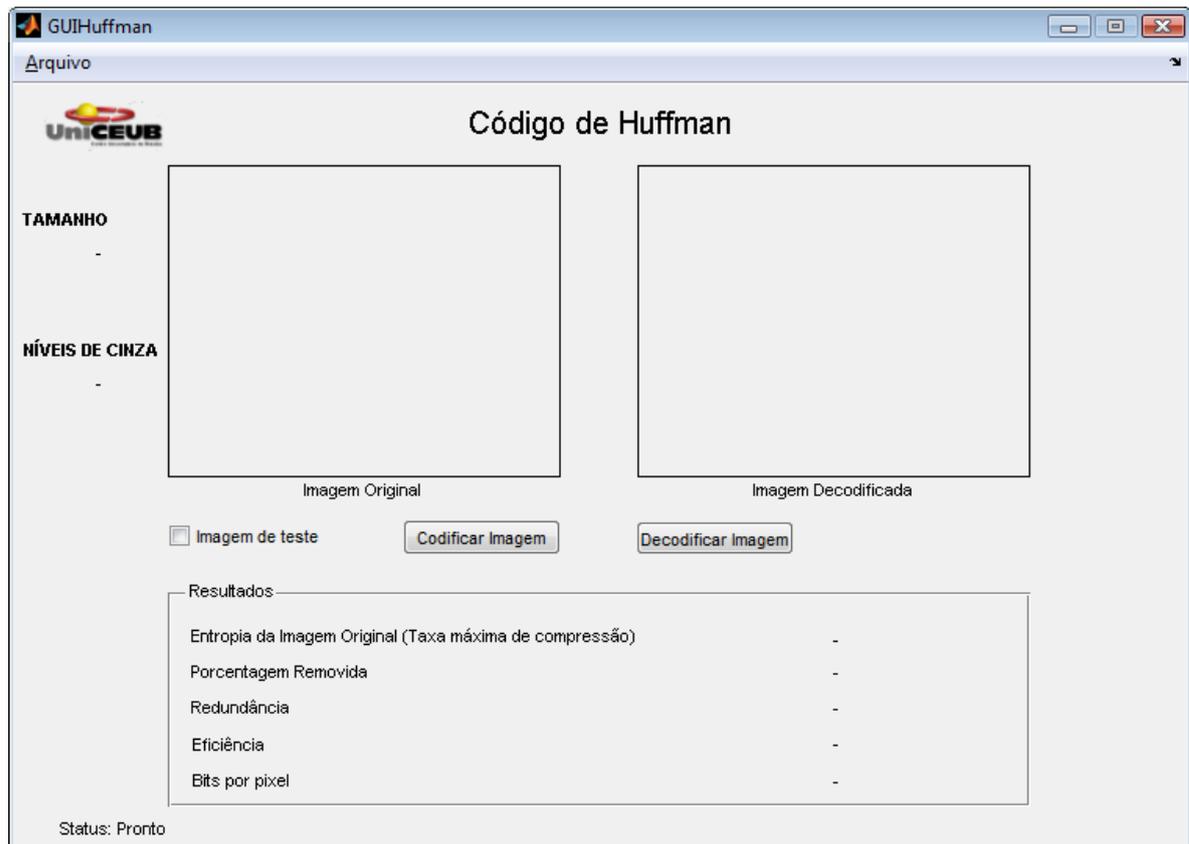


Figura 5.7 – Tela de Codificação de Huffman (GUIHuffman.fig)

FONTE: (a autora)

Para a codificação e decodificação por RLE, cujas operações são também controladas pelos botões “Codificar Imagem” e “Decodificar Imagem”, os resultados obtidos e apresentados ao usuário são o código gerado para casos de imagem de teste, a entropia da

imagem ou taxa máxima de compressão e a porcentagem removida. O campo “Tamanho” caracteriza a imagem a ser codificada.

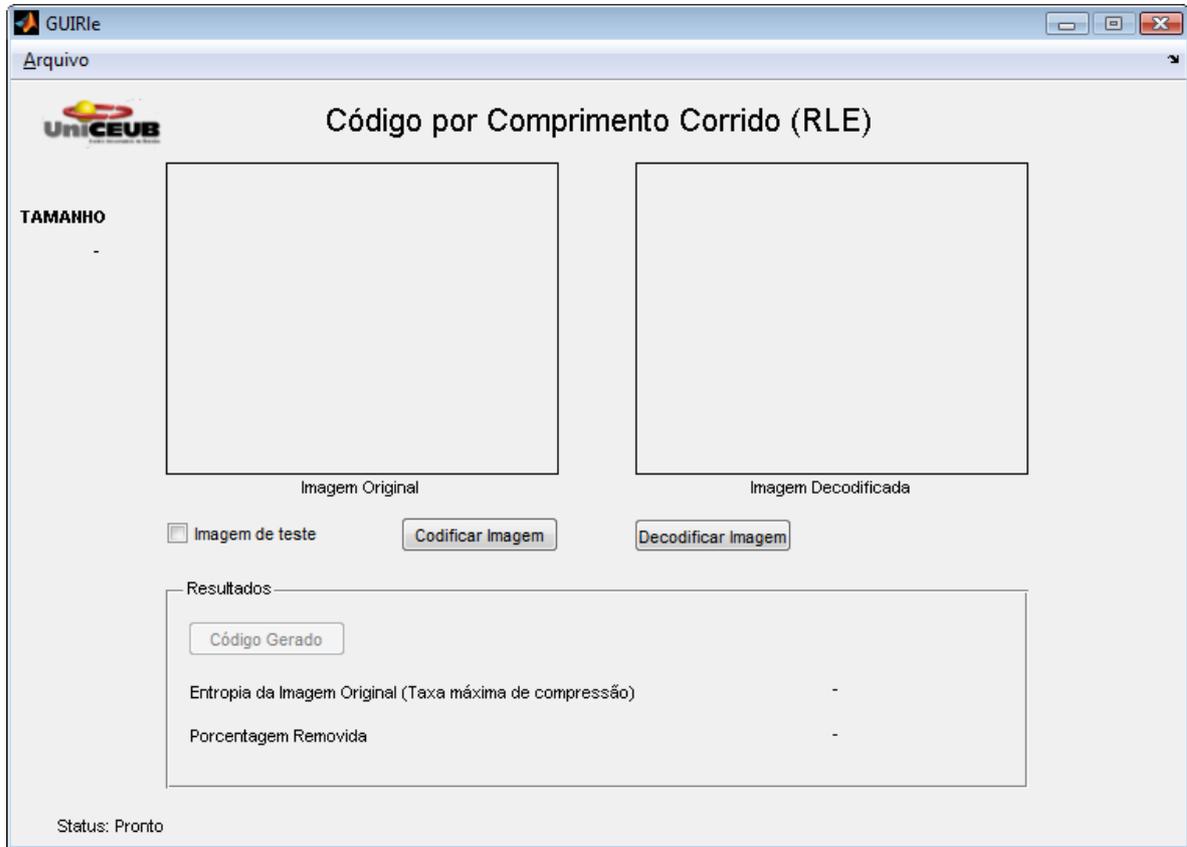


Figura 5.8 – Tela de RLE (GUIRle.fig)
 FONTE: (a autora)

Por fim, exibe-se a tela de qualificação da imagem, como na figura (5.9) cuja operação é controlada pelo botão “Analisar” após carregamento das imagens original e processada. Os resultados disponíveis para qualificação da imagem são o erro quadrático médio (do inglês, MSE), a relação sinal-ruído (do inglês, SNR), a relação sinal-ruído de pico (do inglês, PSNR) e o índice universal de qualidade da imagem (do inglês, UIQI).

O código para cálculo do índice universal de qualidade da imagem (imageQualityIndex.m) é de autoria de (WANG & BOVIK, 2002). Os autores estão em conjunto com o Laboratório de Engenharia de Imagem e Vídeo, Departamento de Engenharia Elétrica e de Computação da Universidade do Texas em Austin.

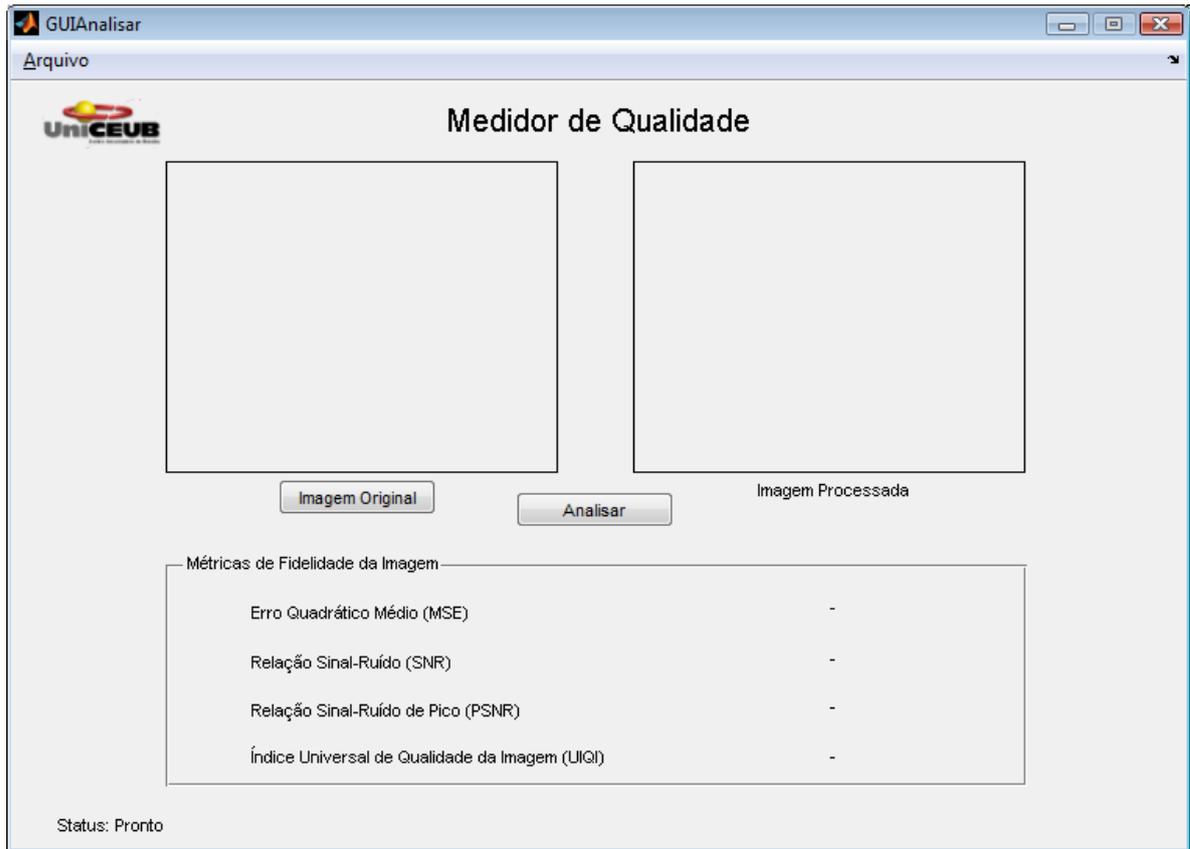


Figura 5.9 – Tela de Medição da Qualidade da imagem (GUIAnalisar.fig)
FONTE: (a autora)

CAPÍTULO 6 – APLICAÇÃO DO MODELO PROPOSTO

6.1 – Resultados Obtidos

Para demonstração dos resultados obtidos, são mostradas telas capturadas (*screen shots*) a partir das telas da aplicação em funcionamento, além de comentários sobre os resultados matemáticos e visuais obtidos.

Para a simulação de degradação, são disponibilizados dois tipos de ruídos (Gaussiano e Impulsivo) e o borrão de movimento. Para a degradação da imagem, devem-se inserir os parâmetros desejados. A figura 5.10 a seguir mostra as imagens degradadas com o ruído gaussiano (a), com o impulsivo (b), com o borrão de movimento + ruído impulsivo (c) e somente com borrão de movimento (d). Os valores respectivos para SNR, PSNR, MSE e UIQI se encontram ao lado direito de cada imagem. O cálculo de MSE é realizado de acordo com a equação (3.14), enquanto o SNR é obtido a partir da equação (3.15). Para os resultados obtidos de PSNR e UIQI foram usadas as equações (3.16) e (3.17) respectivamente.

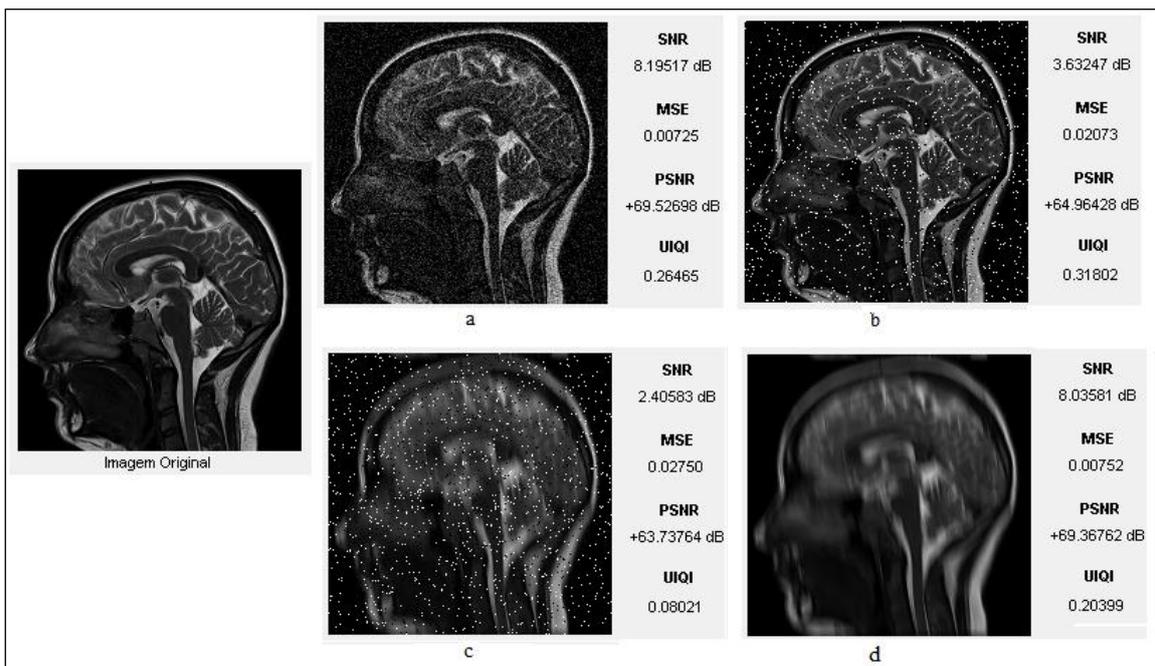


Figura 5.10 – ImagemOriginal, (a)Ruído Gaussiano, (b) Ruído Sal e Pimenta (c) Ruído Sal e Pimenta + Borrão de Movimento e (d) Borrão de movimento

FONTE: (a autora)

Para a degradação com ruído gaussiano, é utilizado o valor padrão de 0.01 para a variância. Para o ruído impulsivo (Sal e pimenta), faz-se uso do valor padrão de 0.05 para a densidade (GONZALES & WOODS, 2001). Já para o borrão de movimento, é usado valor de angulação igual a 98 e valor de comprimento igual a 34 por escolha aleatória. A comparação entre os valores de degradação pode ser realizada por meio da tabela (6.1).

Tabela 6.1 – Resultados Obtidos para Imagem Degradada

Degradação	Métricas			
	MSE	SNR	PSNR	UIQI
Gaussiano	0.00725	8.19517dB	69.526dB	0.26465
Impulsivo	0.02073	3.63247dB	64.964dB	0.31802
Impulsivo + Borrão	0.02750	2.40583dB	63.738dB	0.08021
Borrão de movimento	0.00752	8.03581dB	69.368dB	0.20399

FONTE: (a autora)

A partir das imagens degradadas, é possível gerar o histograma de cada imagem, cujo resultado é mostrado na figura (5.11). Analisando cada um dos histogramas em comparação com o da imagem original, é possível notar uma diferença mais significativa na quantidade de pixels para cada tom de cinza para a imagem degradada por ruído gaussiano e para a imagem degradada por ruído de sal e pimenta + borrão de movimento. Para a degradação somente por ruído impulsivo, o histograma mostra uma diminuição de pixels para níveis de cinza intermediários e um aumento no número de pixels para a intensidade máxima (Sal) e mínima (Pimenta). No caso da degradação somente com borrão de movimento, ocorre um aumento na probabilidade de cada intensidade na imagem, em virtude da maior quantidade de pixels por nível de cinza.

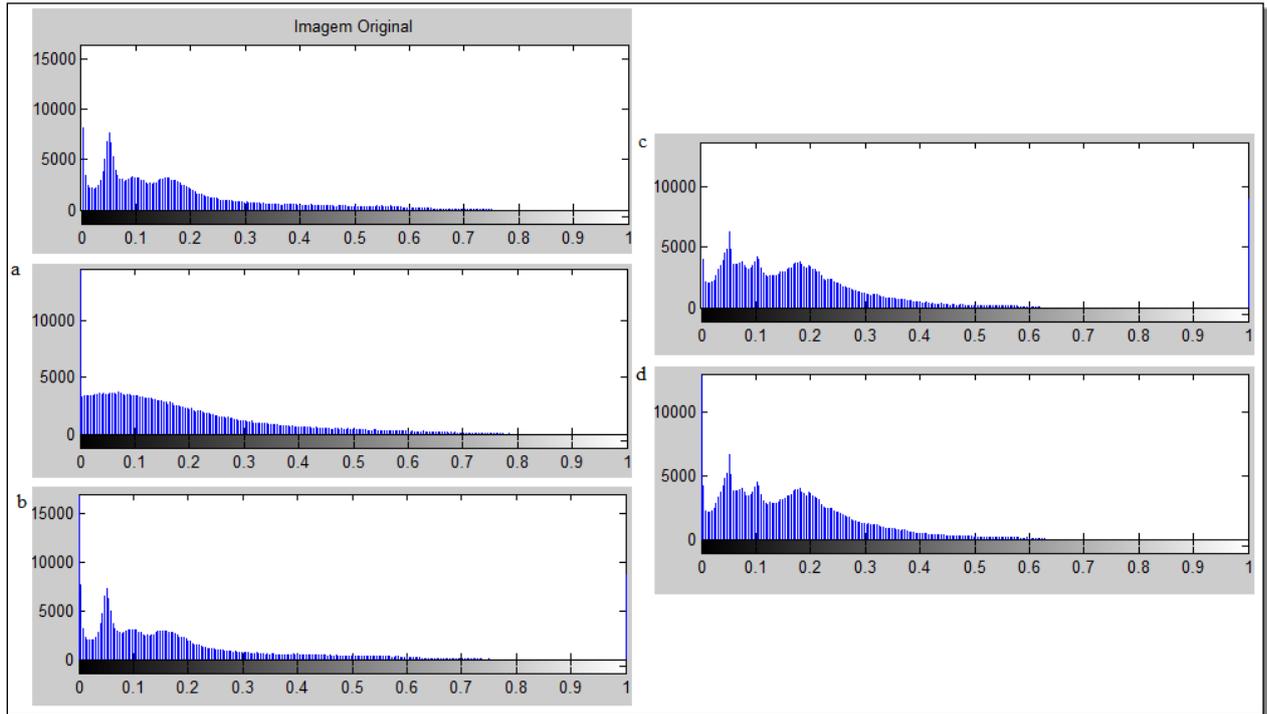


Figura 5.11 – Histogramas: ImagemOriginal, (a) Ruído Gaussiano, (b) Ruído Sal e Pimenta (c) Ruído Sal e Pimenta + Borrão de Movimento (d) Borrão de Movimento
 FONTE: (a autora)

Seguindo o fluxograma da aplicação, operando a tela de filtros espaciais, obtém-se o resultado visual mostrado pelas figuras (5.12) para ruído gaussiano, (5.13) para ruído impulsivo (Sal e Pimenta) e (5.14) para ruído impulsivo + borrão de movimento. As máscaras usadas para todos os filtros seguem o tamanho mínimo de 3x3 (MÜLLER & DARONCO, 2000). Os valores respectivos para SNR e MSE se encontram acima de cada imagem e relativos à fidelidade com a imagem original anterior à degradação.

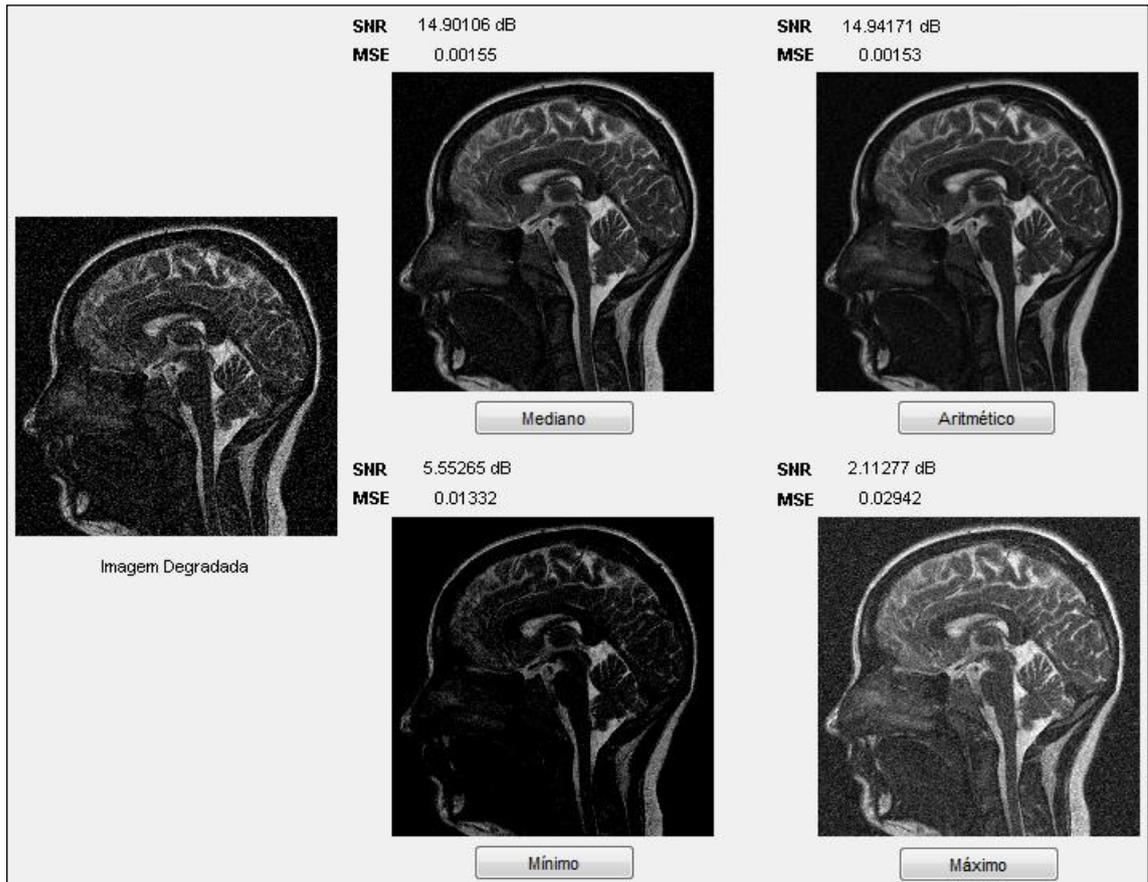


Figura 5.12 – Filtros espaciais para ruído gaussiano.

FONTE: (a autora)

Para o ruído gaussiano, em comparação com os demais filtros, o melhor resultado encontrado foi com o filtro de média aritmética, com uma relação sinal-ruído de 14,94 dB. O filtro da mediana apresenta resultado próximo com 14,90dB.

De acordo com o esperado, o filtro de valor mínimo seleciona o menor valor dentro de uma máscara ordenada de valores de *pixels*. Sendo assim, a imagem filtrada tende a escurecer. De forma contrária, o filtro de valor máximo seleciona o maior valor dentre os valores de *pixel* de uma máscara ordenada. Dessa forma, a imagem se torna mais clara.

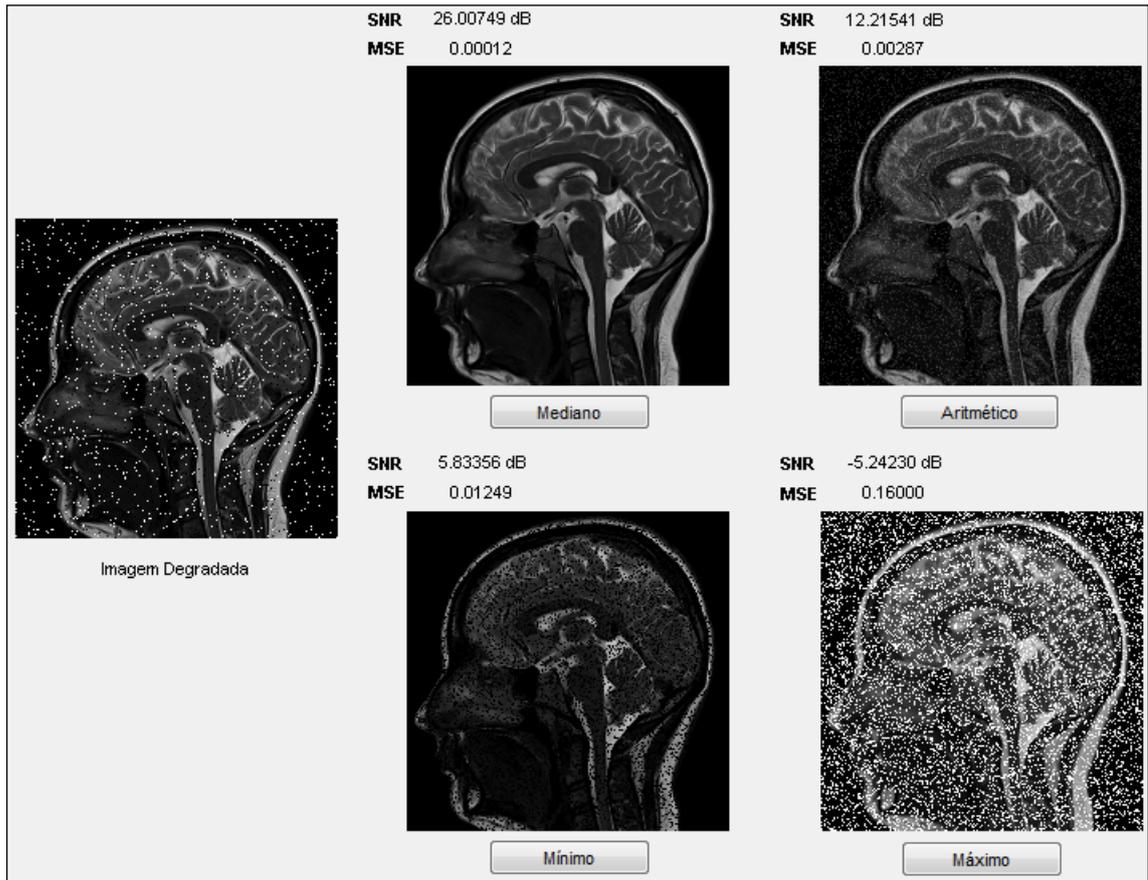


Figura 5.13 – Filtros espaciais para ruído impulsivo.

FONTE: (a autora)

No caso do ruído impulsivo, o filtro da mediana sobressai-se como a melhor opção de restauração, com relação sinal-ruído de 26,01dB. Como esperado, o filtro de valor mínimo seleciona o menor valor dentro de uma máscara ordenada de valores de *pixels*. Sendo assim, o filtro tem bom funcionamento para ruído do tipo sal, visto que esse possui valor de intensidade máximo. A imagem filtrada tende a escurecer com amplificação do ruído do tipo pimenta.

De forma contrária, o filtro de valor máximo seleciona o maior valor dentre os valores de *pixel* de uma máscara ordenada. Dessa forma, ocorre a redução do ruído pimenta, cujo valor da intensidade é o mais baixo de todos. A imagem se torna mais clara e na presença do ruído do tipo sal, o qual se torna mais evidente.

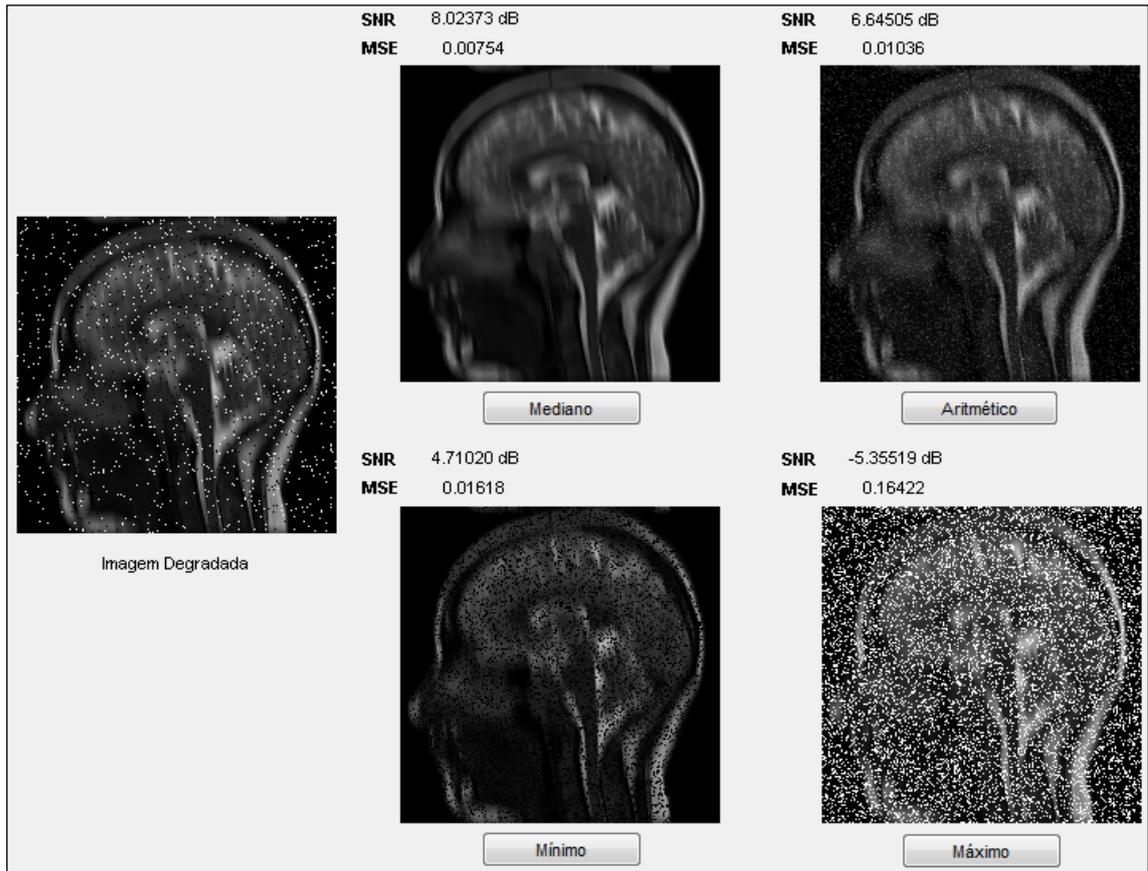


Figura 5.14 – Filtros espaciais para ruído de impulso + borrão de movimento
 FONTE: (a autora)

Pelo resultado mostrado na figura (5.14), confirma-se o funcionamento dos filtros espaciais na restauração apenas com ruídos, sendo ignorada a degradação pelo borrão de movimento. O melhor resultado permaneceu para o filtro de mediana, em virtude do ruído impulsivo, mas a imagem restaurada possui uma relação sinal-ruído de apenas 8,02dB.

Operando-se a tela de filtragem inversa e levando-se em consideração as mesmas imagens degradadas, chega-se aos resultados demonstrados pela figura (5.15). É possível confirmar que, mesmo com o conhecimento da função de degradação, a restauração da imagem é limitada pelo componente de ruído, uma função randômica, não conhecida, o que torna o componente de ruído dominante na estimativa, chegando a uma degradação total em determinado caso (figura 5.15c). Em detrimento disso, para uma degradação sem ruídos e somente com borrão de movimento, o resultado obtido com a filtragem inversa é bastante satisfatório (figura 5.15d).

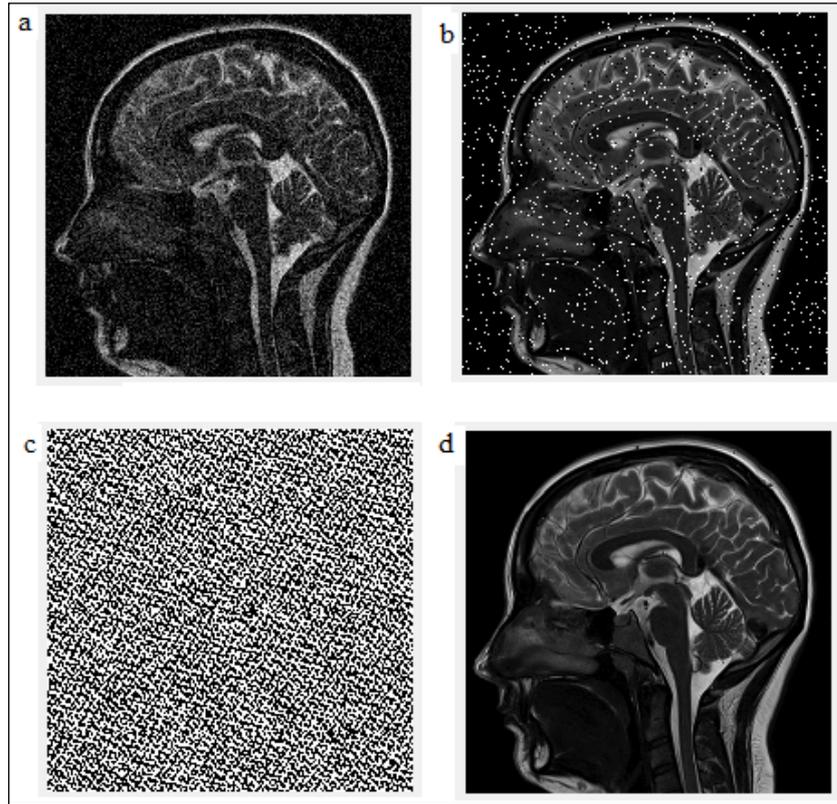


Figura 5.15 – Filtragem inversa para imagem com (a) Ruído Gaussiano, (b) Ruído Sal e Pimenta (c) Ruído Sal e Pimenta + Borrão de Movimento (d) Borrão de Movimento.

FONTE: (a autora)

Quando comparada a imagem restaurada da figura (5.15d) em relação à imagem original, geram-se os resultados matemáticos demonstrados pela tabela (6.2) e pela figura (5.16).

Tabela 6.2 – Métricas para imagem restaurada por filtro inverso

Borrão de movimento – Filtragem inversa	
MSE	0.00000
SNR	44.41dB
PSNR	105.746dB
UIQI	0.80

FONTE: (a autora)

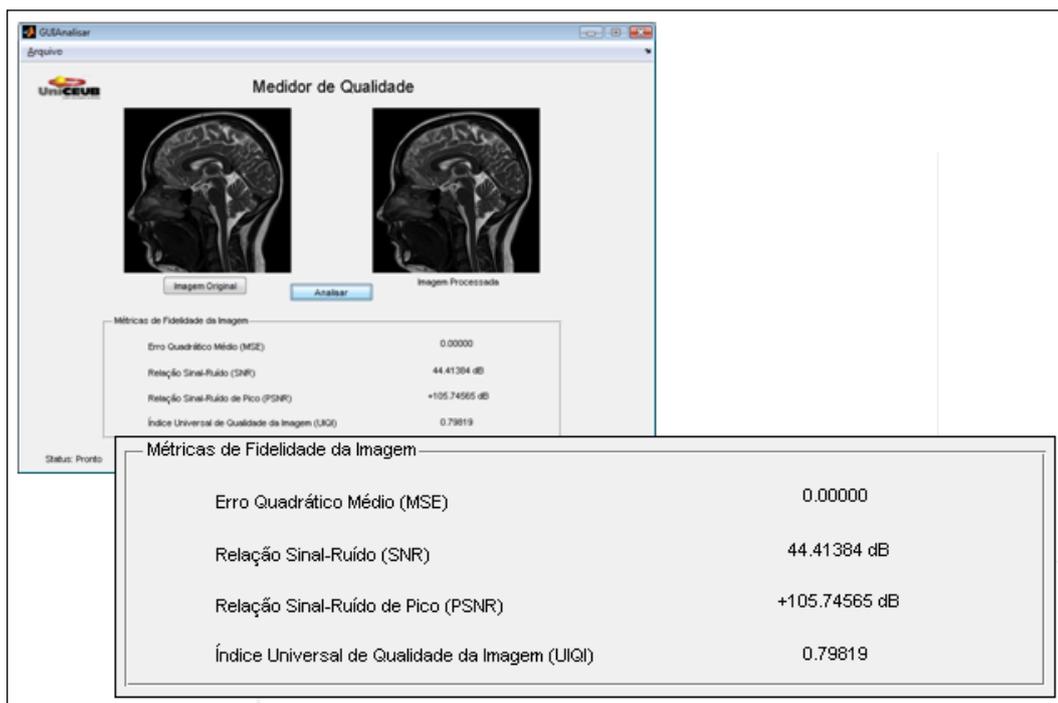


Figura 5.16 – Resultados de MSE, SNR, PSNR e UIQI para filtragem inversa de imagem com borrão de movimento.

FONTE: (a autora)

Operando-se em seguida a tela de filtragem de Wiener e ainda considerando as mesmas imagens degradadas anteriormente, as imagens restauradas são demonstradas pela tabela (6.3) figura (5.17), juntamente aos resultados de medição de qualidade da imagem restaurada.

Tabela 6.3 – Métricas para imagem restaurada por filtro de Wiener

	Gaussiano	Impulsivo	Borrão de movimento + Impulsivo	Borrão de movimento
MSE	0.00154	0.00012	0.00165	0.00106
SNR	14.967dB	25.991dB	14.618dB	16.545dB
PSNR	76.268dB	87.323dB	75.949dB	77.876dB
UIQI	0.43	0.76	0.49	0.54

FONTE: (a autora)

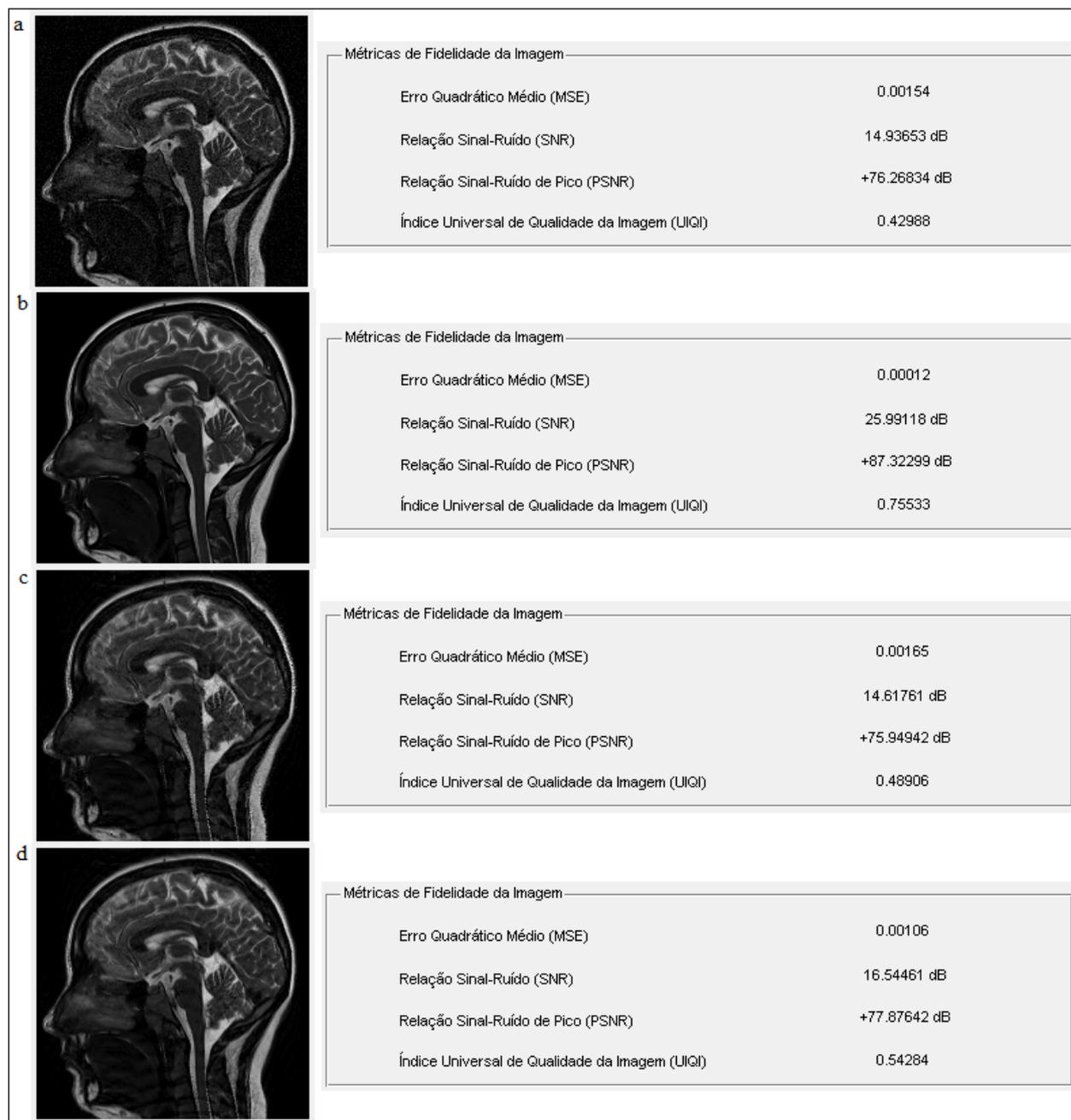


Figura 5.17 – Resultados visuais e de MSE, SNR, PSNR e UIQI para filtragem de Wiener de imagem com (a) Ruído Gaussiano, (b) Ruído Sal e Pimenta (c) Ruído Sal e Pimenta + Borrão de Movimento (d) Borrão de Movimento

FONTE: (a autora)

Os resultados gerais obtidos pela etapa de simulações de degradação e restauração de imagens podem ser resumidos segundo as tabelas 6.4 e 6.5 seguintes. A primeira se refere aos resultados em nível de relação sinal-ruído, enquanto a segunda mostra os valores obtidos para o erro quadrático médio em cada situação.

Tabela 6.4 – Resultados Obtidos para SNR

Degradação	SNR inicial	Filtro					
		Mediana	Média Aritmética	Mínimo	Máximo	Inverso	Wiener
Gaussiano	8.20dB	14.90dB	14.94dB	5.55dB	2.11dB	8.20dB	14.94dB
Impulsivo	3.63dB	26.01dB	12.22dB	5.83dB	-5.24dB	3.67dB	25.99dB
Impulsivo + Borrão	2.40dB	8.02dB	6.65dB	4.71dB	-5.36dB	-76.27dB	14.62dB
Borrão de movimento	8.04dB	8.03dB	8.01dB	7.78dB	7.62dB	44.41dB	16.54dB

FONTE: (a autora)

Tabela 6.5 – Resultados Obtidos para MSE

Degradação	MSE inicial	Filtro					
		Mediana	Média Aritmética	Mínimo	Máximo	Inverso	Wiener
Gaussiano	0.00725	0.00155	0.00153	0.01332	0.02942	0.00726	0.00154
Impulsivo	0.02073	0.00012	0.00287	0.01249	0.16000	0.02036	0.00012
Impulsivo + Borrão	0.02750	0.00754	0.01036	0.01618	0.16422	1000132.1	0.00165
Borrão de movimento	0.00752	0.00753	0.00757	0.00797	0.00827	0.00000	0.00106

FONTE: (a autora)

A segunda etapa de simulações para este trabalho trata da codificação para compressão sem perdas pelo código de Huffman e por RLE. Para a apresentação dos resultados, serão realizadas duas simulações para cada codificação, sendo a primeira à partir de imagem de teste com resolução extremamente pequena, o que permite a visualização das matrizes de intensidades de cinza de cada imagem e das palavras código resultantes. A segunda simulação será realizada com imagem diagnóstica de MRI de alta resolução em escala de cinza.

A adoção de uma imagem com baixa resolução se deve à complexidade e ao tempo de processamento para visualização de matrizes e de alguns resultados para uma imagem com alta resolução e tamanho. O uso da imagem de teste tem o objetivo de validar o método, devido à possível visualização dos valores de sua matriz. A imagem de teste, mostrada na figura (5.18a), possui tamanho de 30x12 *pixels* com resolução de 200 dpi ou 78,74 *pixels/cm*.

Operando-se a funcionalidade do Código de Huffman e com uso da imagem de teste, cuja matriz de intensidade é representada pela figura (5.18b), é obtido como resultado a palavra código apresentada pela figura (5.18c).

O Código de Huffman associa códigos de pequeno comprimento a *pixels* altamente frequentes e códigos de comprimento mais longo a *pixels* menos frequentes (MARTINS,

2002). Cada elemento da matriz mostrada pela figura (5.18b) é um nível de cinza na imagem de teste e é possível verificar a repetição de cada um desses, sendo alguns de maior probabilidade que outros.

Para que seja realizada a codificação, dado um conjunto de símbolos com suas respectivas probabilidades de ocorrência associadas, elabora-se uma estrutura binária cujo resultado se dá pela figura (5.18c). Códigos com maior número de bits se relacionam a elementos com maior repetição, enquanto códigos com menor número de bits são relacionados a elementos menos prováveis. Para cada nível de cinza distinto presente na imagem, um código da estrutura binária gerada é relacionado.

A figura (5.19) já mostra a imagem decodificada, cuja matriz se dá pela figura (5.20). Além disso, são mostradas diversas medidas de avaliação da codificação (tabela 6.4). Comparando-se as matrizes da imagem original (figura 5.18a) e da imagem decodificada, conclui-se que a compressão foi realizada sem perdas. Observa-se que os valores obtidos são idênticos.

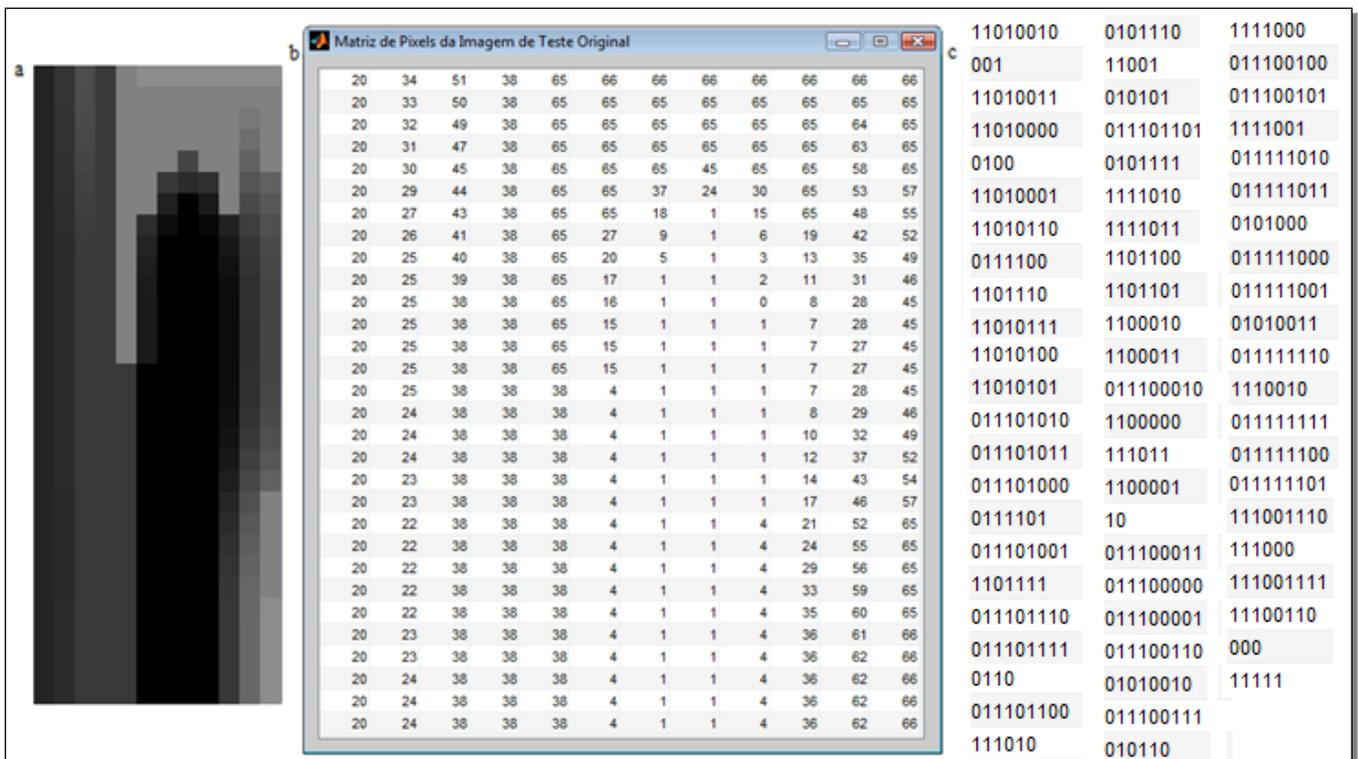


Figura 5.18 – (a) Imagem de teste (b) Matriz de pixels da imagem teste original (c) Palavra código gerada na codificação.

FONTE: (a autora)

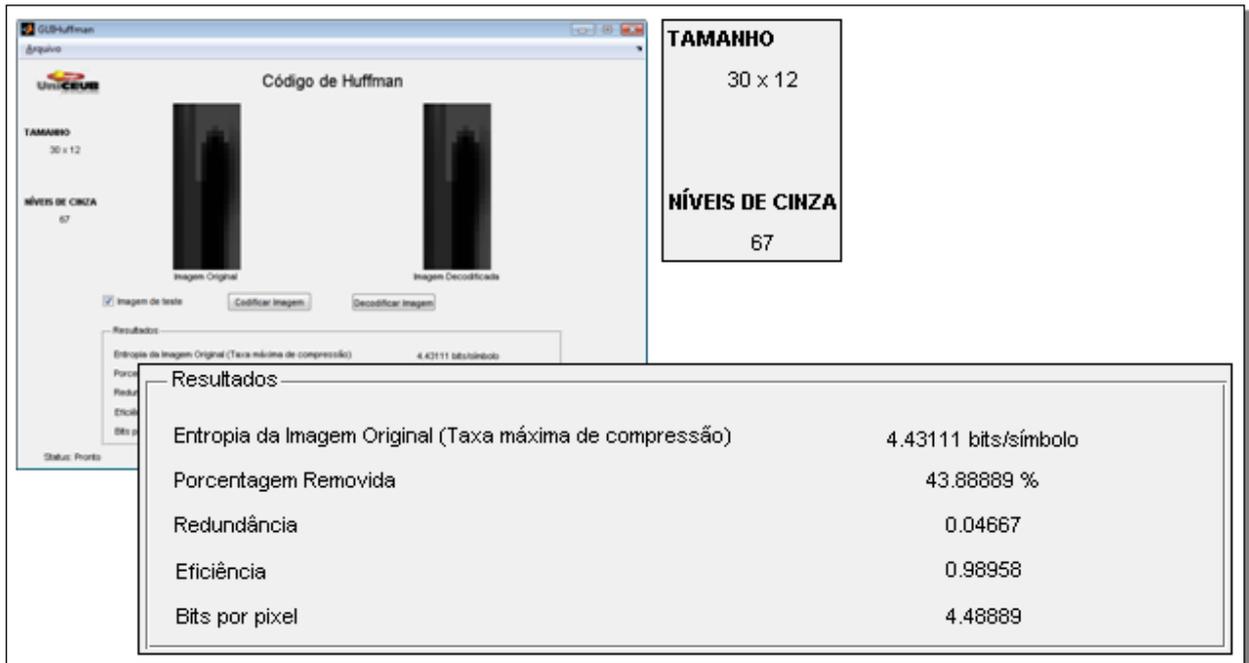


Figura 5.19 – Medidas de avaliação da codificação de Huffman.
FONTE: (a autora)

Matriz de Pixels da Imagem de Teste Decodificada

20	34	51	38	65	66	66	66	66	66	66	66
20	33	50	38	65	65	65	65	65	65	65	65
20	32	49	38	65	65	65	65	65	65	64	65
20	31	47	38	65	65	65	65	65	65	63	65
20	30	45	38	65	65	65	45	65	65	58	65
20	29	44	38	65	65	37	24	30	65	53	57
20	27	43	38	65	65	18	1	15	65	48	55
20	26	41	38	65	27	9	1	6	19	42	52
20	25	40	38	65	20	5	1	3	13	35	49
20	25	39	38	65	17	1	1	2	11	31	46
20	25	38	38	65	16	1	1	0	8	28	45
20	25	38	38	65	15	1	1	1	7	28	45
20	25	38	38	65	15	1	1	1	7	27	45
20	25	38	38	65	15	1	1	1	7	27	45
20	25	38	38	38	4	1	1	1	7	28	45
20	24	38	38	38	4	1	1	1	8	29	46
20	24	38	38	38	4	1	1	1	10	32	49
20	24	38	38	38	4	1	1	1	12	37	52
20	23	38	38	38	4	1	1	1	14	43	54
20	23	38	38	38	4	1	1	1	17	46	57
20	22	38	38	38	4	1	1	4	21	52	65
20	22	38	38	38	4	1	1	4	24	55	65
20	22	38	38	38	4	1	1	4	29	56	65
20	22	38	38	38	4	1	1	4	33	59	65
20	22	38	38	38	4	1	1	4	35	60	65
20	23	38	38	38	4	1	1	4	36	61	66
20	23	38	38	38	4	1	1	4	36	62	66
20	24	38	38	38	4	1	1	4	36	62	66
20	24	38	38	38	4	1	1	4	36	62	66
20	24	38	38	38	4	1	1	4	36	62	66

Figura 5.20 – Matriz de intensidade da imagem decodificada.
FONTE: (a autora)

Realizando o mesmo procedimento com uma imagem diagnóstica real, torna-se inviável a visualização da matriz de intensidade da imagem, em virtude de seu tamanho. A figura (5.21) mostra a codificação e a decodificação de uma imagem diagnóstica de MRI. São apresentadas a imagem original e a imagem decodificada, as medidas de avaliação da codificação e a palavra código gerada.

A codificação da imagem de MR, com tamanho de 600x600 *pixels* e resolução de 300dpi, obteve uma taxa de compressão de 42,66%, permanecendo uma redundância de código de 0.04, com uma eficiência de 0.99 e uma entropia de 4,59 *bits/pixel*.

O cálculo da taxa de compressão é realizado de acordo com a equação (4.2), fazendo uso do número de bytes da imagem original e da imagem comprimida. Para o cálculo da redundância e da eficiência são usados os valores da entropia (equação 4.3) e do comprimento médio do código (equação 4.6). A redundância se dá pela diferença entre os dois valores enquanto a eficiência é calculada pela razão entre os valores, como mostra equação (4.5).

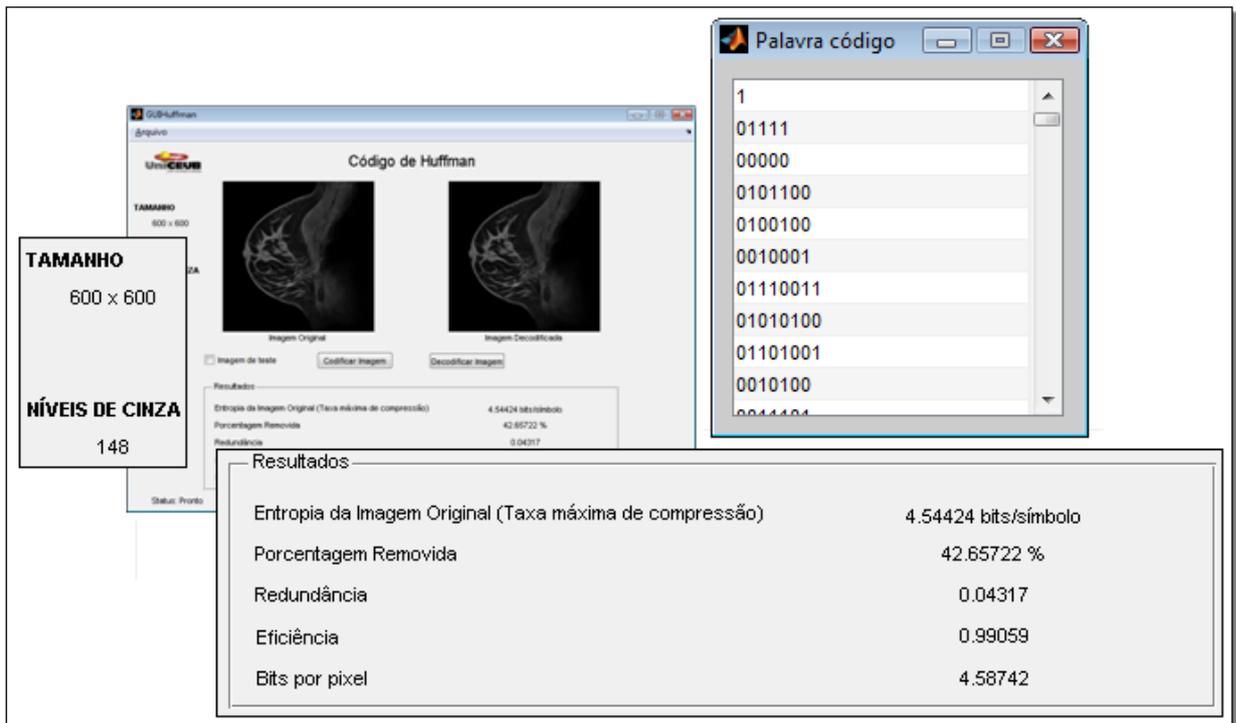


Figura 5.21 – Imagem diagnóstica decodificada por Huffman, medidas de avaliação e palavra código gerada.

FONTE: (a autora)

Para validação da compressão sem perdas, faz-se necessária a qualificação da imagem decodificada em relação à imagem original. Isso se faz pela funcionalidade de medição de qualidade e fidelidade da imagem. Pelos resultados mostrados na figura (5.22) a seguir, comprova-se que a codificação realizada obteve sucesso e a imagem final não contém perdas relativas à imagem de origem ($MSE = 0$ e $UIQI = 1$). O cálculo de MSE é realizado de acordo com a equação (3.14), quanto o UIQI faz uso da equação (3.17).

Os resultados matemáticos para a codificação de Huffman para a imagem de teste e para a imagem real podem ser avaliados pela tabela (6.4) a seguir.

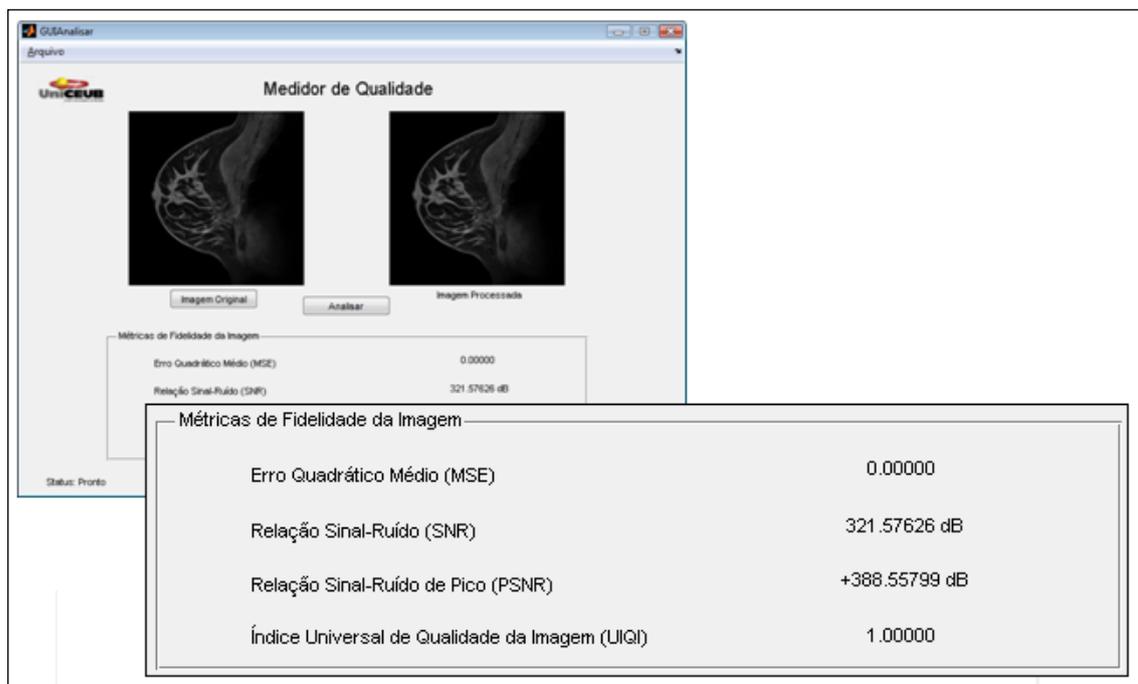


Figura 5.22 – Medidas de qualidade da imagem decodificada por Huffman em relação à imagem original.

FONTE: (a autora)

Tabela 6.6 – Resultados Obtidos para Codificação de Huffman

	Imagem de Teste	Imagem Real	Imagem Real	
Entropia	4.43111 bits/pixel	4.54424 bits/pixel	MSE	0.00000
Tx. Compressão	43.89%	42.66%	SNR	321.576dB
Redundância	0.04667	0.04317	PSNR	388.558dB
Eficiência	0.9896	0.9906	UIQI	1.00000
Bits por pixel	4.489	4.587		

FONTE: (a autora)

Considerando-se, por fim, a codificação por Comprimento Corrido, é realizado novamente o processo com a mesma imagem de teste, representada pela matriz de intensidade da figura (5.18a). A figura (5.23) seguinte mostra que o RLE reduziu a imagem em cerca de 35,56%. A matriz de intensidade da imagem decodificada é apresentada pela figura (5.24a) juntamente com a parte do código gerado (figura 5.24b).

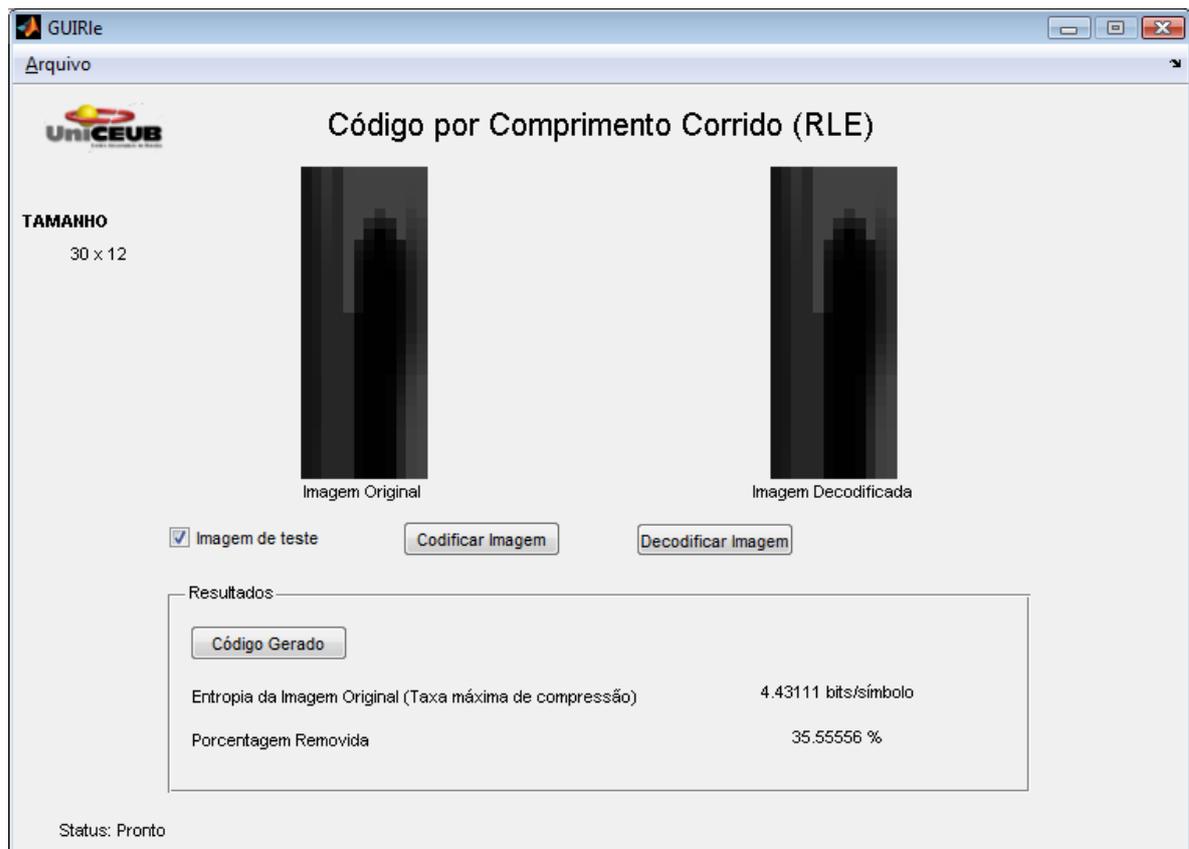


Figura 5.23 – Imagem codificada e decodificada por RLE.
FONTE: (a autora)

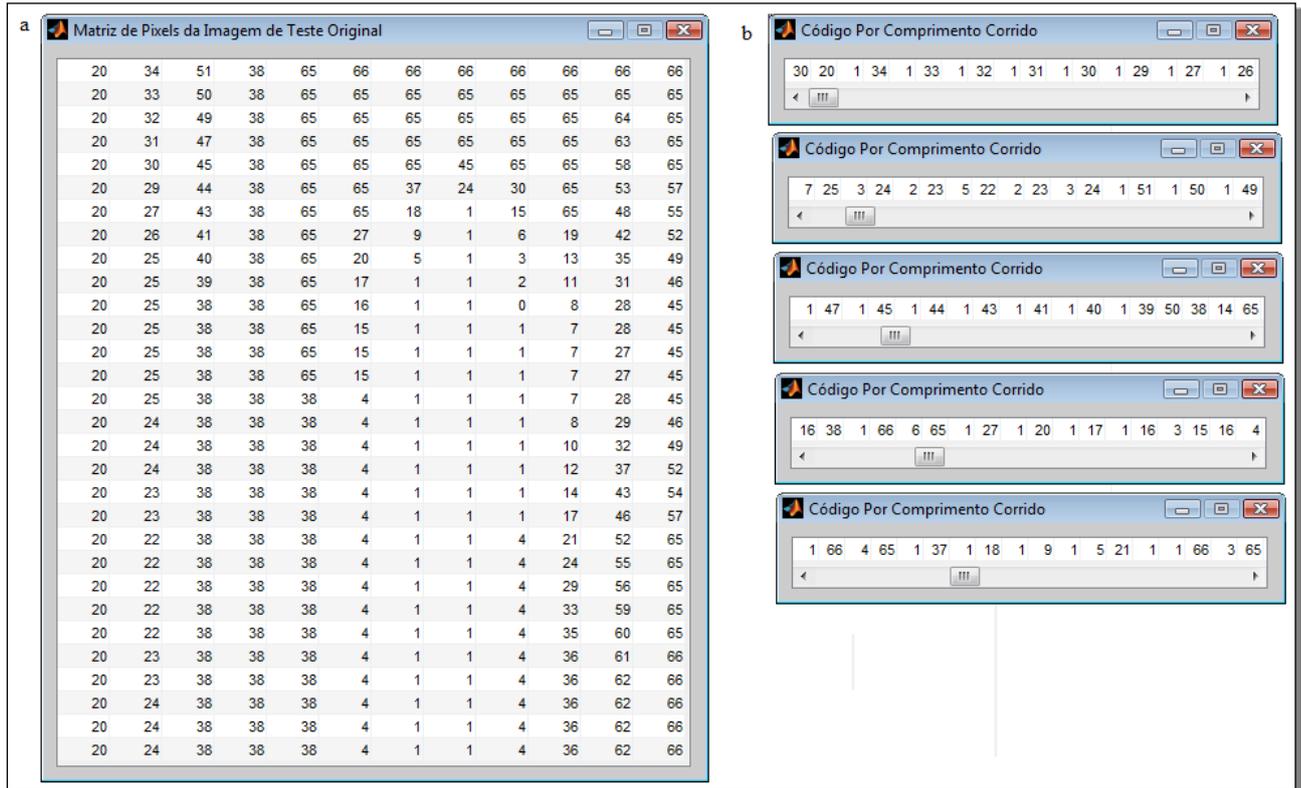


Figura 5.24 – (a) Matriz da imagem decodificada por RLE (b) parte da palavra código.

FONTE: (a autora)

Para finalizar a simulação do processamento de imagem médica digital, realiza-se a codificação da imagem de MR por RLE, com um taxa de compressão de 19,34% (figura 5.25). A fidelidade da imagem decodificada em relação à imagem original é comprovada pelos resultados mostrado na figura (5.26), com $MSE = 0$ e $UIQI = 1$, tratando-se de uma compressão sem perdas.

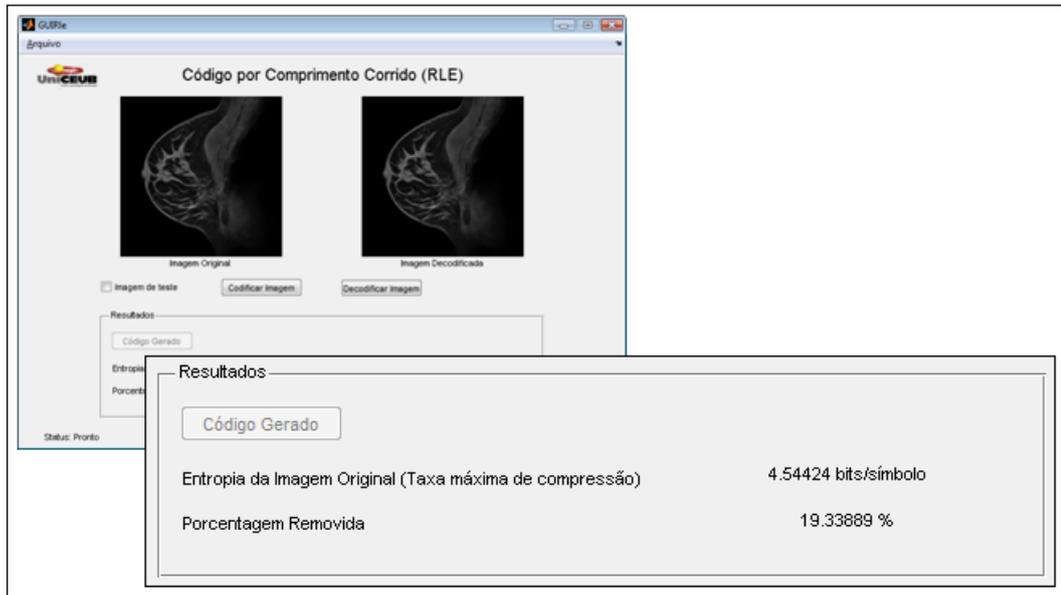


Figura 5.25 – Resultado de codificação de MR por RLE.
 FONTE: (a autora)

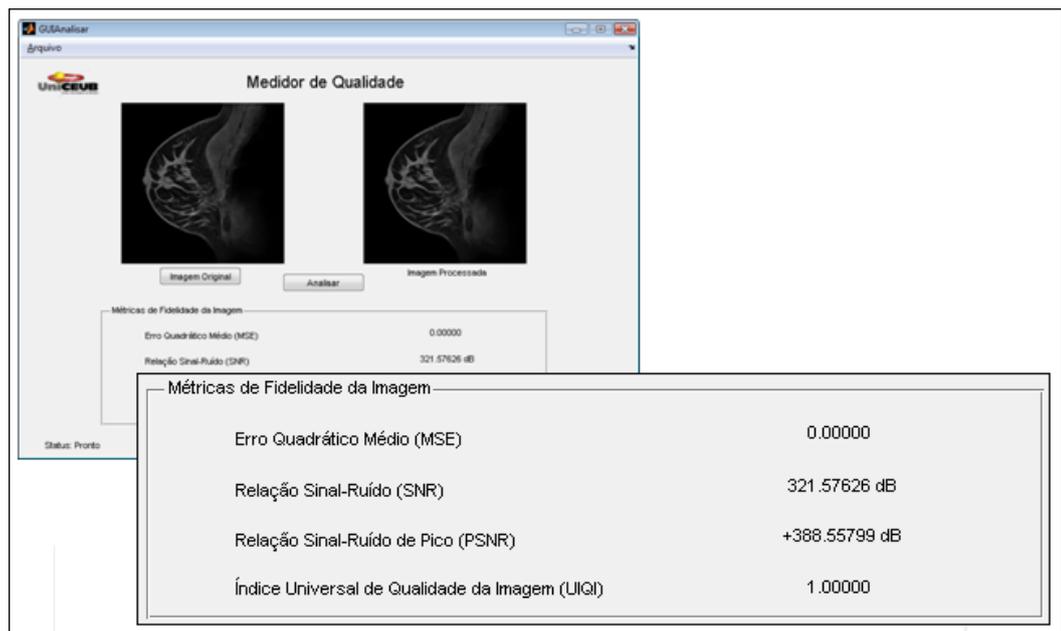


Figura 5.26 – Medidas de qualidade da imagem decodificada por RLE em relação à imagem original.

FONTE: (a autora)

Os resultados matemáticos para a codificação RLE para a imagem de teste e para a imagem real podem ser avaliados pela tabela (6.7) a seguir.

Tabela 6.7 – Resultados Obtidos para Codificação RLE

	Imagem de Teste	Imagem Real	Imagem Real	
Entropia	4.43111 bits/pixel	4.54424 bits/pixel	MSE	0.00000
Tx. Compressão	35.56%	19.34%	SNR	321.576dB
			PSNR	388.558dB
			UIQI	1.00000

FONTE: (a autora)

6.2 – Análise de Resultados

Após desenvolvimento e testes do modelo proposto, é possível realizar uma análise entre os diferentes resultados obtidos. A funcionalidade de simulação de ruídos e borrões é necessária para simular situações reais de degradação de imagens de diagnóstico. A partir disso, torna-se necessária a aplicação de filtros para restauração.

Em se tratando dos tipos de filtros abordados neste trabalho, nota-se que o filtro espacial da média aritmética e o filtro de Wiener são de eficiência semelhante na presença do ruído gaussiano, com uma relação sinal-ruído de 14,95dB para a imagem restaurada. Quando se obtém uma imagem degradada por ruído impulsivo, a imagem restaurada chega a uma SNR de 26,14dB para a filtragem espacial da Mediana e de 25,99dB para a filtragem de Wiener. Adicionando-se à imagem corrompida por ruído impulsivo uma degradação por função de transferência, nesse caso, por borrão de movimento, a filtragem de Wiener alcança um resultado de SNR de 14,62dB para a imagem restaurada. Para os três casos citados, na presença de ruídos aditivos, comprova-se a amplificação do ruído após uso do filtro inverso, como propõe a equação (3.11).

No último caso de degradação estudado, cuja imagem se apresenta corrompida pela função de degradação de movimento, o filtro inverso se mostra como a melhor opção de restauração com uma SNR de 44dB para a imagem final. Quando a imagem possui uma função de degradação além do ruído, os filtros espaciais tornam-se ineficientes.

Em se tratando de compressão de imagens médicas, foram desenvolvidas as codificações de Huffman e RLE. Para o primeiro caso, as maiores taxas de compressão se dão para imagens com níveis de cinza equiprováveis ou com probabilidades iguais a potência negativa de dois. Já para o RLE, o bom funcionamento ocorre para imagens que possuem grande quantidade de dados repetitivos, podendo ocorrer taxas de compressão negativas para imagens sem redundância interpixel. Os resultados foram satisfatórios tanto para as imagens

de teste como para as imagens reais, de maior tamanho, maior resolução e com maior quantidade de níveis de cinza. Obteve-se, para ambas as codificações, um retorno sem perdas à imagem original.

CAPÍTULO 7 - CONCLUSÃO

7.1 – Conclusão

Neste trabalho se propôs a avaliação dos métodos de codificação sem perdas de Huffman e RLE, investigando o comportamento de ambos sobre imagens médicas digitais por meio de uma interface gráfica. Isso visando à redução do volume de dados gerados no imageamento médico para posterior armazenamento ou transmissão. Além disso, a GUI proposta é capaz de degradar uma imagem de MR ou CT com ruído gaussiano ou ruído impulsivo (sal e pimenta), além da possível inserção de borrão de movimento, todos de comum ocorrência em imagens diagnósticas. Tal funcionalidade é necessária para simular situações reais de degradação, o que levanta a necessidade da aplicação de filtros para restauração, como filtros espaciais, filtro inverso e filtro de Wiener.

A simulação se torna útil à medida que nos deparamos com imagens diagnósticas geradas ou recebidas de forma degradada. Para o ruído gaussiano, os filtros da média aritmética e de Wiener se mostraram igualmente eficientes. Para casos de imagem degradada por sal e pimenta, a filtragem de Wiener e o filtro da mediana recebem destaque para uma restauração de com bom resultado. Na existência somente do ruído do tipo sal (pontos brancos), a melhor escolha se dá no filtro de mínimo, enquanto que para o ruído do tipo pimenta (pontos pretos), faz-se melhor uso do filtro de máximo.

Quando a imagem possui uma função de degradação além do ruído, os filtros espaciais tornam-se ineficientes e a melhor filtragem para uma imagem com borrão de movimento e ruído define-se como a de Wiener. O filtro inverso passa a ser a melhor opção em casos de degradação sem ruídos, comprovando-se sua limitação na presença de ruídos randômicos cuja transformada se torna desconhecida.

Em virtude do exposto, assume-se como atingidos os objetivos de comparação entre formas restauração de imagens médicas digitais degradadas, visto que estas são ferramentas importantes para medicina diagnóstica hoje e necessitam de processamentos que garantam sua qualidade.

A compressão de imagens médicas digitais faz-se necessária em virtude do volume alto de dados gerados e que se tornam custosos em se tratando das possibilidades de armazenamento e de transmissão. Por se tratarem de imagens que necessitam de alto índice de

qualidade, a compressão realizada para os devidos fins deve garantir uma descompressão fiel à origem.

Foram realizadas em imagens médicas as codificações de Huffman e RLE, amplamente utilizadas em processos de compressão de dados sem perdas. Os resultados da decodificação de ambas nas imagens de teste foram satisfatórios, assim como no uso de imagens reais de diagnóstico. Para o caso de Huffman, os melhores resultados se dão para imagens com níveis de cinza equiprováveis ou com probabilidades iguais a potência negativa de dois. Já para o RLE, o bom funcionamento ocorre para imagens que possuem grande quantidade de dados repetitivos, podendo ocorrer taxas de compressão negativas para imagens sem redundância interpixel. Dessa forma, atinge-se o segundo objetivo deste trabalho, garantindo a codificação sem perdas para as imagens de MR ou CT, dependendo da disposição de tonalidades de cinza em cada uma delas.

É possível, por fim, concluir que a possibilidade de uso real do aplicativo desenvolvido se mostra viável. Isso em virtude dos resultados obtidos próximos de valores ótimos após comparação entre as diversas formas de restauração de imagens degradadas e codificação para compressão sem perdas.

7.2 – Sugestões para Trabalhos Futuros

Durante a realização do trabalho, surgiram diversas possibilidades de aprimoramento que são sugeridas como propostas para trabalhos futuros:

- Desenvolvimento de métodos completos de compressão sem perdas, que incluam as codificações de Huffman e/ou RLE;
- Incremento da quantidade de codificações para uma comparação mais abrangente de resultados, como codificação aritmética ou de Lempel-Ziv;
- Aprimoramento da comparação entre as diversas formas de filtragem, incrementando a quantidade de filtros analisados e o uso de filtragem adaptativa;
- Aprimoramento do processamento de imagens médicas digitais, incluindo, por exemplo, a segmentação de imagens.

REFERÊNCIAS BIBLIOGRÁFICAS

(ANAHIT, 1999) ANAHIT HOVHANNISYAN, B.S.. *Comparison of lossless compression models*. Texas Tech University, 1999.

(AGOSTINI, 2000) AGOSTINI, L.V.. *Estudo de Padrões de Compressão de Imagens para Aplicações VLSI*, Trabalho Individual de Pos-Graduação, UFRGS, Porto Alegre, 2000.

(BAUERMANN, 2008) BAUERMANN, G.. *TIFF – formato de imagens sob medida para aplicações científicas*. Image Survey. Disponível em: <http://www.imagesurvey.com.br/2008/10/tiff-formato-de-imagens-sob-medida-para-aplicacoes-cientificas/>. Acesso em: 03/06/2011.

(BEN-EZRA & NAYAR, 2004) BEN-EZRA, M.; NAYAR, S.. *Motion-Based Motion Deblurring*. IEEE Transaction on pattern analysis and machine intelligence, vol. 26, no. 6, 2004.

(CANTADORI & FURUIE, 2002) CANTADORI, M.L.; FURUIE, S.S.. *Análise e Desenvolvimento de Filtros Digitais Adaptativos para Imagens Médicas*. Escola Politécnica de São Paulo, 2002.

(CHIU et. al, 2001) CHIU, E., Lin, J.; McFerron B.; Petigara, N.; Seshasai, S. *Mathematical Theory of Claude Shannon*. 2001.

(CHOY & CHAN, 1996) CHOY, S. S. O.; CHAN, Y., SIU, W.. *An improved quantitative measure of image restoration quality*. *Acoustics, Speech, and Signal Processing*. ICASSP-96. Conference Proceedings., 1996 IEEE International Conference on Volume 3, 7-10 May 1996.

(CLUNIE, 2000) CLUNIE, D.A.. *Lossless compression of grayscale medical images – Effectiveness of traditional and state of the art approaches*. 521 Plymouth Road, Suite 115, Plymouth meeting PA 19462, 2000.

(CONCI & AQUINO, 2005) CONCI, A.; AQUINO, F. R.. *Fractal coding based on image local fractal dimension. Computational & Applied Mathematics*, Petrópolis, RJ, v.24, 2005.

(GAMBA et. al, 1999) GAMBA, H.R.; Bayford, R.; Holder, D.. *Measurement of Electrical Current Density Distribution in a simple head phantom with Magnetic Resonance Imaging*, Physics in Medicine & Biology, v. 44, 1999.

(GAMBA & DELPY, 1998) GAMBA, H.R.; DELPY, D.T.. *Measurement of electrical current density distribution within the tissues of the head by Magnetic Resonance Imaging*, Medical Biological Engineering & Computing, v. 36, 1998.

(GONZALES & WOODS, 2001) GONZALES, R.C.; WOODS, R.E.. *Digital Image Processing*, Prentice Hall, 2ed.ISBN: 0-201-18075-8, 2001.

(GROSS, 1998) GROSS, M.H.. *Computer graphics in medicine: From visualization to Surgery Simulation*. Acm Siggraph Computer Graphics, New York, 1998.

(HUFFMAN, 2010) HUFFMAN. *Compression of Huffman Coding*. Disponível em: <https://blog.itu.dk/BADS-F2009/files/2009/04/46-huffman.pdf>. Acesso em: 22/05/2011.

(JLS, 2003) JLS. *Aula06Compressão*. Disponível em: <http://gec.di.uminho.pt/lesi/vpc0405/Aula06Compress%C3%A3o.pdf>. Acesso em: 22/05/2011.

(KAUR & KAUSHAL, 2008) KAUR, R.; KAUSHAL, N.. *Comparative Analysis of Various Compression Methods for Medical Images*. National Institute of Technical Teachers' Training & Research, Panjab university, Chandigarh, 2008.

(KIL et. al, 2006) KIL, S.K.; Lee, J.S.; Shen, D.F.; Ryu, J.G.; Lee, E.H.; Min, H.K.;Hong, S.H.. *Lossless medical image compression using redundancy analysis*. IJCSNS International Journal of Computer Science and 50 Network Security, vol.6 No.1A, 2006.

(KUMAR, 2011) KUMAR, S.. *An Introduction to Image Compression*. Disponível em: <http://www.debugmode.com/imagecmp>. Acesso em: 26/04/2011.

(LABGIS, 2011) LABGIS.. *Realce de Imagem* . Disponível em:

http://www.labgis.uerj.br/gis_atualizada/pdi/04_realce.htm. Acesso em: 10/06/2011.

(MAGNUSSON, 2010) MAGNUSSON, M.. *Digital Image Processing*. 2010.

(MARQUES, 1999) MARQUES, F. O.; NETO, H. V.. *Processamento Digital de Imagens*. Rio de Janeiro, Brasport, ISBN 85-7452-009-8, 1999.

(MARQUES & VIEIRA, 1999) MARQUES FILHO, O.; VIEIRA NETO, H.. *Processamento Digital de Imagens*, Rio de Janeiro: Brasport, 1999.

(MARTINO, 2002) MARTINO, M.. *Sistemas de Informações Gráficas*, 2002.

(MARTINS, 2002) MARTINS, V.. *Avaliação da taxa de compressão sem perdas em projeções de tomografia computadorizada*. Curitiba, 2002.

(MATHWORKS, 2011) MATHWORKS.. *Matlab Central*. Disponível em: <http://www.mathworks.com/matlabcentral/> . Acesso em: 22/04/2011.

(MELO, 1994) MELO, A. C. W. P.. *Compressão de Imagens em Escala de Cinza Utilizando a Transformada Wavelet*, Dissertação de Mestrado, CEFET-PR, Curitiba, 1994.

(MOGER, 2008) MOGER, J.. *Digital Image Processing for Radiographers: Noise & Image Restoration*, 2008.

(MÜLLER & DARONCO, 2000) MÜLLER, D.N.; DARONCO, E.L.. *Filtros Espaciais Passa-Baixa*. Universidade Federal do Rio Grande do Sul, Porto Alegre, 2000.

(NETO, 2010) NETO, H.V.. *Histograma e Transformações de Intensidade*. Universidade Tecnológica Federal do Paraná, 2010.

(OLIVEIRA, 2003) OLIVEIRA, F.M.. *Ferramentas de pré-processamento de Imagens*. Universidade de Brasília, 2003.

(PAPPAS & SAFRANEK, 1999) PAPPAS, T.N.; SAFRANEK, R.J.. *Perceptual Criteria for image quality evaluation*. São Paulo: Manole, 2002.

(RABBANI, 2002) RABBANI, M.. *An overview of the JPEG2000 still image compression standard*, Signal Process., Image Commun., 2002.

(RATIB, 1995) RATIB, O.. *From PACS to the World Wide Web*, Geneva University Hospital, 1995.

(REN & CAVALCANTI, 2010) REN, T.I.; CAVALCANTI, G.D.C.. *Restauração de Imagens*. Universidade Federal de Pernambuco, 2010.

(RIBEIRO, 2006) RIBEIRO, B.. *Suavização de Imagens*. Disponível em: www.ic.uff.br/~aconci/suavizacao.pdf. Acesso em: 25/04/2011.

(RLE, 2010) RLE. *Run Length Encoding*. Disponível em: http://www.fileformat.info/mirror/egff/ch09_03.htm. Acesso em : 22/05/2011.

(ROMUALDO, 2009) ROMUALDO, L.C.S.. *Restauração de imagens mamográficas digitais utilizando o filtro de Wiener no domínio de Anscombe e o filtro inverso da MTF no domínio da frequência*. São Carlos: 2009.

(SANCHES, 2001) SANCHES, I.J.. *Compressão sem perdas de projeções de tomografia computadorizada usando a transformada wavelet*. Curitiba, 2001.

(SEARA, 1998) SEARA, D.M. *Visão Geral de Detecção de Bordas*. Disponível em: www.inf.ufsc.br/~visao/bordas.html. Acesso em: 16/04/2011.

(SCURI, 2002) SCURI, A.E.. *Fundamentos da Imagem Digital*. TecGraf/PUC – Rio de Janeiro, 2002.

(SILVA, 2001) SILVA, A. M.. *Curso Processamento digital de imagens de satélite*. Centro de Eventos da PUCRS. Porto Alegre, 2001. Disponível em www.cartografia.org.br. Acesso em: 05/06/2011.

(STOLFI, 2000) STOLFI, G.. *Processos de compressão de dados aplicados a imagens médicas*. Escola Politécnica da Universidade de São Paulo, 2000.

(SCHWARTZ & PEDRINI, 2007) SCHWARTZ, W. R., PEDRINI, H. *Análise de Imagens Digitais, Princípios Algoritmos e Aplicações*. São Paulo, 2007.

(SYMES, 1998) SYMES, P.D.. *Video Compressing*. McGraw-Hill, ISBN:0-07-063344-4, 1998.

(THOMÉ, 2004) THOMÉ, A.. *Processamento de Imagens: Tratamento da Imagem -Filtros*. Universidade Federal do Rio de Janeiro, 2004.

(TINOIS, 2000) TINOIS, E.. *Imagem Funcional – PET e fMRI*. Universidade de Campinas, 2000.

(TORRES, 2002) TORRES, M.S.G.. *Compressão sem perda de imagens através de predição não causal com informação lateral de posição*. Rio de Janeiro, 2002.

(VIEIRA, 2010) VIEIRA, M.A.C.. *Introdução ao Processamento Digital de Imagens: Restauração de Imagens*. São Paulo - USP, 2010

(WANG & BOVIK, 2002) WANG, Z.; BOVIK, A.C.. A Universal Image of Quality Index. IEEE Signal Processing Letters, vol. XX, 2002.

(WATANABE, 2005) WATANABE, C.L.. *Compressão de Imagens Médicas*. São José dos Campos, 2005.

ANEXO A – Código do Índice Universal de Qualidade da Imagem.

O Código a seguir (imageQualityIndex.m) foi proposto por (WANG & BOVIK, 2002) e é utilizado para o cálculo do UIQI entre imagem original e imagem restaurada.

```
function [quality, quality_map] = imageQualityIndex (img1, img2, block_size)
%=====
% * Note: img_qi is renamed as imageQualityIndex
%
%Copyright (c) 2001 The University of Texas at Austin
%All Rights Reserved.
%
%Author   : Zhou Wang
%Version  : 1.0
%
%The authors are with the Laboratory for Image and Video Engineering
%(LIVE), Department of Electrical and Computer Engineering, The
%University of Texas at Austin, Austin, TX.
%
%Kindly report any suggestions or corrections to zwang@ece.utexas.edu
%
%Acknowledgement:
%The author would like to thank Mr. Umesh Rajashekar, the Matlab master
%in our lab, for spending his precious time and giving his kind help
%on writing this program. Without his help, this program would not
%achieve its current efficiency.
%
%=====
%
%This is an efficient implementation of the algorithm for calculating
%the universal image quality index proposed by Zhou Wang and Alan C.
%Bovik. Please refer to the paper "A Universal Image Quality Index"
%by Zhou Wang and Alan C. Bovik, published in IEEE Signal Processing
%Letters, 2001. In order to run this function, you must have Matlab's
%Image Processing Toolbox.
%
%Input  : an original image and a test image of the same size
%Output: (1) an overall quality index of the test image, with a value
```

```

%         range of [-1, 1].
%         (2) a quality map of the test image. The map has a smaller
%             size than the input images. The actual size is
%             img_size - BLOCK_SIZE + 1.
%
%Usage:
%
%1. Load the original and the test images into two matrices
%   (say img1 and img2)
%
%2. Run this function in one of the two ways:
%
%   % Choice 1 (suggested):
%   [qi qi_map] = img_qi(img1, img2);
%
%   % Choice 2:
%   [qi qi_map] = img_qi(img1, img2, BLOCK_SIZE);
%
%   The default BLOCK_SIZE is 8 (Choice 1). Otherwise, you can specify
%   it by yourself (Choice 2).
%
%3. See the results:
%
%   qi           %Gives the over quality index.
%   imshow((qi_map+1)/2) %Shows the quality map as an image.
%
%=====

if (nargin == 1 | nargin > 3)
    quality = -Inf;
    quality_map = -1*ones(size(img1));
    return;
end

if (size(img1) ~= size(img2))
    quality = -Inf;
    quality_map = -1*ones(size(img1));
    return;
end

if (nargin == 2) % sliding window, B by B
    block_size = 8;
end

N = block_size.^2;
sum2_filter = ones(block_size);

img1_sq = img1.*img1;

```

```
img2_sq = img2.*img2;
img12 = img1.*img2;

img1_sum = filter2(sum2_filter, img1, 'valid');
img2_sum = filter2(sum2_filter, img2, 'valid');
img1_sq_sum = filter2(sum2_filter, img1_sq, 'valid');
img2_sq_sum = filter2(sum2_filter, img2_sq, 'valid');
img12_sum = filter2(sum2_filter, img12, 'valid');

img12_sum_mul = img1_sum.*img2_sum;
img12_sq_sum_mul = img1_sum.*img1_sum + img2_sum.*img2_sum;
numerator = 4*(N*img12_sum - img12_sum_mul).*img12_sum_mul;
denominator1 = N*(img1_sq_sum + img2_sq_sum) - img12_sq_sum_mul;
denominator = denominator1.*img12_sq_sum_mul;

quality_map = ones(size(denominator));
index = (denominator1 == 0) & (img12_sq_sum_mul ~= 0);
quality_map(index) = 2*img12_sum_mul(index)./img12_sq_sum_mul(index);
index = (denominator ~= 0);
quality_map(index) = numerator(index)./denominator(index);

quality = mean2(quality_map);
```

ANEXO B – Código para Decodificação de Huffman.

O Código a seguir (unravel.c) foi retirado da biblioteca DIPUM e é utilizado para a etapa de decodificação de Huffman. A função minimiza o tempo requerido para realizar buscas em binários, visto que o MATLAB® possui melhor desempenho para matrizes de tamanho uniforme, em detrimento de tamanhos variáveis (VLC) (MATHWORKS, 2011).

```

#include "mex.h"
void unravel(unsigned short *hx,double *link1,double *link2,double *x,double xsz,int hxsz)
{
    int i=15,j=0,n=0;
    int k=0,k1;
    int a,b,b1,b2,c,m,L,i1;
    double z1=0,z2;

dc: while (1)
    {
        if (*(link1+n)>0)
        {
            if ((*hx+j)>>i)&0x0001
                n=*(link1+n);
            else
                n=*(link1+n)-1;
            if (i) i--;else{j++;i=15;}
            if(j>hxsz)
                mexErrMsgTxt("Out of code bits ???");
        }
        else
        {
            a=-(link1+n);
            a=a-1;
            b=a;
            if (i+1>=b)
            {
                c=(*(hx+j)>>(i-b+1)) & (unsigned short)((pow(2,b)-1));
                i=i-b+1;
                if (i) i--; else {j++;i=15;}
            }
            else
            {
                c=*(hx+j) & (unsigned short)(pow(2,i+1)-1);
                j++;
                b2=b-(i+1);
                i=15-b2+1;
                c=(c<<b2) | (*(hx+j)>>i & (unsigned short)(pow(2,b2)-1));
                i--;
            }
            if (c<pow(2,b-1))
                c=-(pow(2,b)-1-c);
            z2=z1+(double)c;
            z1=z2;
            *(x+k++)=z2;
            n=0;
            goto ac;
        }
    }
}

```

```

ac: while (1)
{
    if (*(link2+n)>0)
    {
        if (*(hx+j)>>i)&0x0001
            n=*(link2+n);
        else
            n=*(link2+n)-1;
        if (i) i--;else{j++;i=15;}
        if(j>hxsz)
            mexErrMsgTxt("Out of code bits ???");
    }
    else
    {
        a=*(link2+n);
        a=a-1;
        if (a>=1 && a=160)
        {
            if (a%10!=0)
            {
                {
                    b1=a/10;
                    b=a%10;
                }
                else
                {
                    b1=a/10-1;
                    b=10;
                }
            }
            else if (a==161)
            {
                b1=16;
                b=0;
            }
            else
            {
                n=0;
                k1=64-k%64;
                for (i1=1;i1=k1;i1++)
                    *(x+k++)=(double)0;
                if (k>=xs*64)
                    break;
                else
                    goto dc;
            }
            if (i+1>=b)
            {
                c=*(hx+j)>>(i-b+1) & (unsigned short)((pow(2,b)-1));
                i=i-b+1;
                if (i) i--; else {j++;i=15;}
            }
            else
            {
                c=*(hx+j) & (unsigned short)(pow(2,i+1)-1);
                j++;
                b2=b-(i+1);
                i=15-b2+1;
                c=(c<<b2) | (*(hx+j)>>i & (unsigned short)(pow(2,b2)-1));
                i--;
            }
            if (c<pow(2,b-1))
                c=-(pow(2,b)-1-c);
            for (m=0;m<b1;m++)
                *(x+k++)=(double)0;
            if (b1!=16)
            {
                *(x+k++)=(double)c;
            }
            n=0;
        }
    }
}
}
}

```

```
void mexFunction(int nlhs, mxArray *plhs[],
                 int nrhs, const mxArray *prhs[])
{
    double *link1,*link2,*x,xsz;
    unsigned short *hx;
    int hxsz;

    if (nrhs!=4)
        mexErrMsgTxt("Three input required.");
    else if (nlhs>1)
        mexErrMsgTxt("Too many output arguments.");

    if (!mxIsDouble(prhs[3])||mxIsComplex(prhs[3])||mxGetN(prhs[3])*mxGetM(prhs[3])!=1)
        mexErrMsgTxt("Input XSIZE must be a scalar.");

    hx=mxGetPr(prhs[0]);
    link1=mxGetPr(prhs[1]);
    link2=mxGetPr(prhs[2]);
    xsz=mxGetScalar(prhs[3]);

    hxsz=mxGetM(prhs[0]);

    plhs[0]=mxCreateDoubleMatrix(xsz*64,1,mxREAL);
    x=mxGetPr(plhs[0]);

    unravel(hx,link1,link2,x,xsz,hxsz);
}
```

APÊNDICE A – Código do Modelo Proposto.