



Universidade Federal do Rio Grande do Norte
Centro de Tecnologia
Graduação em Engenharia Mecatrônica

Compressão de imagens com transmissão em tempo real

Ricardo Costa Antunes de Sousa

Orientador: Prof. Dr. Adrião Duarte Dória Neto

Coorientador: Prof. Dr. Joaquim Elias de Freitas

Natal/RN

Junho de 2015.



Universidade Federal do Rio Grande do Norte
Centro de Tecnologia
Graduação em Engenharia Mecatrônica

Compressão de imagens com transmissão em tempo real

Ricardo Costa Antunes de Sousa

Orientador: Prof. Dr. Adrião Duarte Dória Neto

Coorientador: Prof. Dr. Joaquim Elias de Freitas

Trabalho de Conclusão de Curso
apresentada ao Programa de Graduação
em Engenharia de Mecatrônica da
Universidade Federal do Rio Grande do
Norte como parte dos requisitos para a
obtenção do título de Engenheiro
Mecatrônico.

Natal/RN

Junho de 2015.



Universidade Federal do Rio Grande do Norte
Centro de Tecnologia
Graduação em Engenharia Mecatrônica

Trabalho de Conclusão de Curso apresentado à banca examinadora composta pelos seguintes membros:

Prof. Dr. Adrião Duarte Dória Neto (Orientador) DCA/UFRN

Prof. Dr. Joaquim Elias de Freitas CCET/UFRN

Prof. Dr. Agostinho de Medeiros Brito Júnior DCA/UFRN

Agradecimentos

Primeiramente gostaria de agradecer ao meu sogro, amigo e professor Dr. Joaquim Elias, por todas orientações, conselhos, dedicação e companheirismo durante este trabalho. À minha noiva Gabriela, pela atenção, paciência e disposição para ajudar e revisar a escrita de cada parágrafo do trabalho. Sem vocês eu não teria conseguido.

Aos meus pais Valdir e Jacinta, que sempre me providenciaram todo o suporte e serviram de modelo para minha educação e meu caráter pessoal.

Aos meus amigos Guilherme, Harônio, João Victor, João Marcos, Márcio, Marié, Marília, Teles e Tomaz, os quais se mantiveram me motivado durante todos esses anos, principalmente nas madrugadas que passamos estudando no Departamento de Computação e Automação.

Aos professores do curso, em especial a Dr. Adrião, Dr. Darlan, Dr. Diego, Dr Luiz Felipe, Dr. Pablo, Dr. Márcio e Dr. Samaherni por toda orientação dada durante o período que estive na universidade.

A todas as pessoas do laboratório de Robótica por sempre estarem me fazendo vivenciar novos aprendizados, sendo o local que mais pude absorver conhecimento durante minha jornada na UFRN.

A todos que pude conhecer durante meu estágio da Petrobrás, em especial ao meu supervisor Jefferson; meus companheiros de estágio Amanda, Paloma, Milene, Rony, Rodrigo, Sanderson e Stephanie; além dos técnicos/engenheiros Ariel, Geraldo, Odiletil, Vicente, Pacheco, Sena, João, Lindijane, Cosme, Faria, Celso, Délio, Gilce, Márcia e Roberto. Obrigado por me proporcionarem uma oportunidade ímpar na minha vida, compartilhando experiências que irão contribuir para minha vida no mercado de trabalho.

Para finalizar, gostaria de agradecer a todas as pessoas que contribuíram de uma forma direta ou indireta para a conclusão deste trabalho e do curso.

Resumo

Este trabalho de conclusão de curso apresentará o estudo realizado para o desenvolvimento de um algoritmo capaz de comprimir e transmitir imagens coloridas em tempo real. O tipo de compressão adotado foi com perdas, sendo a Transformada de Wavelet discreta de Haar aliada à quantização os principais métodos para realizar a redução da quantidade de dados redundantes. Para a transmissão foi utilizado o protocolo UDP e uma rede que conta com um sistema de câmera conectado ao computador para realizar o envio das imagens comprimidas, além de uma estação base para receber e decodificar as informações transmitidas.

Palavras-chave: Processamento Digital de Imagens; Compressão de Imagens; Transformada de Wavelet; Haar.

Abstract

This end of course Project presents the study of the development of an algorithm to compress and transfer colour images in real time. The type of compression adopted was with losses, with the Haar Discrete Wavelet Transform allied with the quantization being the main methods to avoid redundant data. Regarding data transmission, the UDP protocol was used, as well as a network that consists of a system of cameras connected to a computer to perform the dispatch of the compressed images and a base station to receive, decode and decompress the transmitted information.

Keywords: Digital Image Process; Image Compression; Wavelet Transform; Haar.

Sumário

Sumário.....	i
Lista de Figuras.....	iii
Lista de Tabelas e Algoritmos.....	v
Lista de Símbolos e Abreviaturas.....	vii
1 Introdução.....	1
1.1 Introdução.....	1
1.2 Motivações e Objetivos.....	2
1.3 Estado da Arte.....	3
1.4 Organização da Dissertação.....	3
2 Fundamentação Teórica.....	4
2.1 Imagem Digital.....	4
2.1.1 Introdução.....	4
2.1.2 Modelo de Cores.....	5
2.1.2.1 Modelo RGB.....	6
2.2 Compressão de Imagens.....	7
2.2.1 Introdução.....	7
2.2.2 Compressão Sem Perdas.....	7
2.2.3 Compressão Com Perdas.....	7
2.2.4 Razão de Compressão.....	10
2.2.5 Critério de Fidelidade.....	10
2.2.5.1 Critério de Fidelidade Objetivo.....	11
2.2.5.2 Critério de Fidelidade Subjetivo.....	12
2.3 Transformada de Wavelet.....	13
2.3.1 Introdução.....	13
2.3.2 Transformada de Wavelet Discreta em 1D.....	14
2.3.3 As Wavelets de Haar e Daubechies.....	14

2.3.4	Matriz de Transformação.....	16
2.3.5	Transformada de Wavelet Discreta em 2D.....	18
2.3.6	Transformada Inversa de Wavelet.....	20
3	Materiais, Métodos e Resultados.....	22
3.1	Materiais.....	22
3.2	Métodos.....	23
3.3	Resultados.....	26
4	Considerações Finais.....	29
5	Referências Bibliográficas.....	30

Lista de Figuras

Figura 2.1	Componente verde da imagem de Lenna, apresentando a localização dos eixos adotados e os limites de fg correspondente à intensidade da cor no pixel.....	5
Figura 2.2	Exemplo das componentes de uma imagem RGB formando uma imagem colorida.....	6
Figura 2.3	Diagrama em blocos do processo de compressão e descompressão de imagens.....	8
Figura 2.4	Resultado de uma compressão utilizando no processo de quantização (a) 8 bits, (b) 12 bits, (c) 16 bits e (d) 20 bits, para representar os coeficientes.....	9
Figura 2.5	Exemplo de como construir R1 e R2, para aplicar a Transformada de Wavelet de Daubechies de 4 pontos, em um espaço de dimensão 12...	18
Figura 2.6	Resultado da aplicação (nas colunas e linhas respectivamente) da Transformada na Matriz que representa o sinal 2D, gerando ondas do tipo SS, SD, DS e DD.....	20
Figura 2.7	Exemplo de como aplicar as auto resoluções.....	20
Figura 3.1	Sistema utilizado para realização dos testes de compressão, transmissão, recepção e descompressão de uma imagem.....	22
Figura 3.2	Exemplo de como é determinado a distância mínima e máxima de um coeficiente.....	25
Figura 3.3	(a) Imagem Lenna Original. (b) Imagem Lenna comprimida com a razão de compressão 14,39 e PSNR 35,07 <i>dB</i> . (c) Imagem Lenna comprimida com a razão de compressão 28,91 e PSNR 33,88 <i>dB</i> .Exemplo de como é determinado a distância mínima e máxima de um coeficiente.....	27

Figura 3.4	(a) Imagem Baboon Original. (b) Imagem Baboon comprimida com a razão de compressão 13,81 e PSNR 29,98 <i>dB</i> . (c) Imagem Baboon comprimida com a razão de compressão 27,45 e PSNR 29,68 <i>dB</i> .Exemplo de como é determinado a distância mínima e máxima de um coeficiente.....	27
Figura 3.5	(a) Imagem Airplane (F16) Original. (b) Imagem Airplane (F16) comprimida com a razão de compressão 13,54 e PSNR 36,06 <i>dB</i> . (c) Imagem Airplane (F16) comprimida com a razão de compressão 27,38 e PSNR 34,43 <i>dB</i>	28
Figura 3.6	(a) Imagem Peppers Original. (b) Imagem Peppers comprimida com a razão de compressão 14,01 e PSNR 34,51 <i>dB</i> . (c) Imagem Peppers comprimida com a razão de compressão 28,39 e PSNR 33,46 <i>dB</i>	28

Lista de Tabelas e Algoritmos

Tabela 2.1	Escala de Classificação da Television Allocation Study Organization.....	12
Algoritmo 3.1	Transformação da Imagem.....	24

Lista de Abreviaturas

CMY - Cyan, Magenta, Yellow

CMYK - Cyan, Magenta, Yellow, Black

D - Dimensão

dB - Decibel

DCT - Transformada do Cosseno Discreta

DFT - Transformada de Fourier Discreta

DWT - Transformada de Wavelet Discreta

HIS - Hue, Saturation, Intensity

KLT - Transformada de Karhunen-Loève

MB - MegaBytes

ms - Milisegundos

MSE - Mean Square Error

OSI - Open System Interconnection

PSNR - Peak Signal to Noise Ratio

QMF - Quadrature Mirror Filters

RGB - Red, Green, Blue

RMSR - Root Mean Square Error

RLE - Run Length Encoder

s - Segundos

SNR - Signal to Noise Ratio

TCP - Transmission Control Protocol

UDP - User Datagram Protocol

USB - Universe Serial Bus

WHT - Transformada de Walsh-Hadamard

Capítulo 1

Introdução

1.1 Introdução

Com o crescente progresso da tecnologia digital nas últimas décadas, foi possível o avanço de inúmeras aplicações de telecomunicação que envolvem a transmissão de imagens digitais em tempo real (COSTA, 1993). Segurança, teleconferência, imagens médicas, sensoriamento remoto e reconhecimento de padrões são alguns exemplos de áreas que utilizam essa tecnologia.

Uma das ferramentas mais úteis na área de processamento de imagens que permitiram o desenvolvimento dessas aplicações é a compressão, sendo essencial para qualquer propósito que envolva transmissão ou armazenamento de imagens. Todavia, é necessário um mínimo de qualidade que varia de acordo com seu emprego.

A compressão de imagens pode ser realizada com ou sem perda de informação. A realizada com perda é utilizada quando a transmissão e a redução do tamanho da imagem são mais importantes que a qualidade visual, neste caso pode ocorrer perda de informações. Já a sem perda é aplicada quando a exatidão da imagem deve ser mantida, não havendo perdas na informação (FALCÃO et al., 1998).

Entretanto, para atender as aplicações que necessitam de resposta em tempo real, a compressão pode ser problemática, pois seus métodos envolvem muitos cálculos matemáticos que demandam um custo computacional muito grande. Devido a isso, diversos estudos na área estão sendo realizados, com a finalidade de facilitar a compressão de imagens em tempo real.

1.2 Motivações e Objetivos

O objetivo principal desta dissertação é produzir um algoritmo capaz de comprimir as imagens e realizar a transmissão em tempo real. Mesmo com o avanço da capacidade de processamento isso continua sendo um desafio, pois as imagens também foram aperfeiçoadas e estão cada vez com mais qualidade, detalhes e maiores dimensões.

Uma webcam, por exemplo, costuma disponibilizar cerca de 30 imagens (ou frames) por segundo para uma aplicação de videoconferência. Para transmitir essas imagens digitais e coloridas (3 bytes/pixel) com resolução de 640x480 pixels, numa taxa de 30 frames/s, é necessário que os dados sejam transmitidos em uma velocidade de:

$$30 \frac{\text{frames}}{s} \times (640 \times 480) \frac{\text{pixels}}{\text{frames}} \times 3 \frac{\text{bytes}}{\text{pixels}} = 27.648.000 \text{ bytes/s}$$

Ou seja, em uma residência que possui uma banda larga de 10 MB não seria possível usufruir de uma aplicação como essa e de diversas outras que envolvam imagens.

Com o intuito de minimizar esse problema, será utilizada a Transformada de Wavelet de Haar a fim de realizar a compressão. Apesar desse método não proporcionar a melhor qualidade de imagem, o custo computacional exigido é o menor ao ser comparado com as outras Transformadas de Wavelets, cujo fator é essencial para aplicações em tempo real.

Neste trabalho, a compressão e transmissão de cada imagem disponibilizada por uma câmera deverá ser efetuada em menos de 33,333 milissegundos para assegurar uma taxa de 30 frames/segundo, confirmando dessa maneira, a transmissão em tempo real.

O tipo de compressão adotado foi com perdas, cuja redução da quantidade de dados redundantes foi garantida pela quantização juntamente com a remoção de coeficientes irrelevantes gerados pela Transformada de Wavelet.

Por fim, também é objetivo deste trabalho realizar um teste do algoritmo desenvolvido em uma rede que conta com um sistema de câmera conectado ao computador para realizar o envio das imagens, além de uma estação base para receber e decodificar as informações transmitidas.

1.3 Estado da Arte

Serão abordados a seguir alguns estudos sobre a compressão de imagens:

SANCHES (2001) fez estudo do uso da Transformada Wavelet para compressão sem perdas de projeções de tomografia computadorizada. Para isso, foi utilizada a Transformada Wavelet de inteiros reversível, o que permitiu realizar a compressão sem perdas.

LIMA (2002) publicou um artigo introdutório sobre a Transformada de Wavelet, se restringindo às Wavelets Discretas de Haar e Daubechies para utiliza-las na compactação de dados (imagens digitais).

GOMES (2005) em sua dissertação de mestrado estudou a compressão seletiva de imagens, desenvolvendo um algoritmo que selecionava as regiões de interesse, através de Redes Neurais, para que não houvesse perdas de informações essenciais durante a compressão.

SANTOS (2012) realizou um estudo da compressão de imagens aplicando a Função de Peano para linearizar as imagens com máxima concentração de pontos vizinhos e a Transformada Wavelet 1D para decompô-la. Estas aplicações permitiram extrair informações relevantes para o armazenamento de uma imagem com um menor custo computacional e com uma margem de erro muito pequena ao comparar as imagens original e processada.

1.4 Organização da Dissertação

Esta dissertação está organizada da seguinte forma: no Capítulo 2 é feita uma breve introdução sobre imagens digitais, compressão de imagens e Transformada de Wavelet. O Capítulo 3 aborda os materiais utilizados nesse trabalho, quais métodos foram adotados para compressão, transmissão, recepção e descompressão da imagem e os resultados obtidos. As Considerações Finais estão descritas no Capítulo 4.

Capítulo 2

Fundamentação Teórica

Este capítulo irá abordar toda a fundamentação teórica que se faz necessário para a compreensão do trabalho, informando sobre os temas de imagem digital, compressão de imagens e Transformada de Wavelet.

2.1 Imagem Digital

2.1.1 Introdução

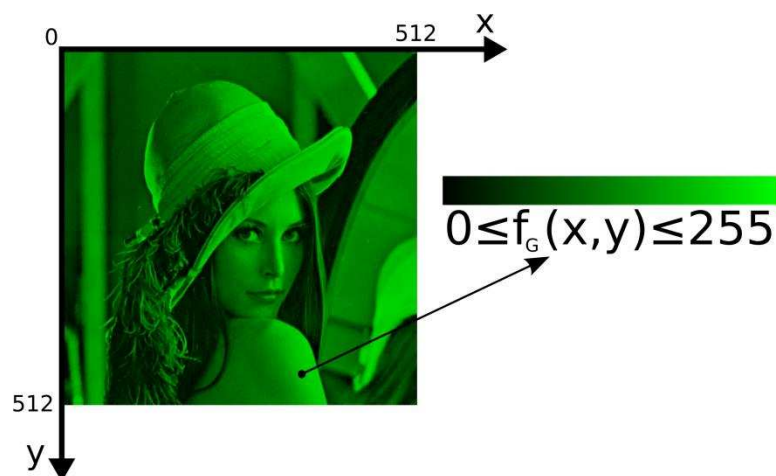
Uma imagem digital pode ser definida como sendo uma matriz de pontos, onde cada ponto (pixel) contém a informação de sua cor, podendo ser representada através de uma função bidimensional $f(x,y)$, onde x e y são coordenadas cartesianas discretas e o valor de f no ponto (x,y) corresponde à intensidade da cor no pixel. Quando um sinal analógico da câmera é obtido, deve ser submetido a um processo de discretização para que seja possível adquirir o formato da matriz mencionada. Ou seja, a amostragem converte a imagem analógica na seguinte matriz M por N pontos:

$$f(x,y) = \begin{bmatrix} f(0,0) & f(0,1) & \cdots & f(0,N-1) \\ f(1,0) & f(1,1) & \cdots & f(1,N-1) \\ \vdots & \vdots & \ddots & \vdots \\ f(M-1,0) & f(M-1,1) & \cdots & f(M-1,N-1) \end{bmatrix} \quad (2.1)$$

No caso de uma imagem colorida, os pixels possuem informações de bandas distintas de frequência, sendo necessária uma função $f(x,y)$ para cada uma delas.

A Figura 2.1 mostra que cada pixel possui uma localização espacial determinada pelos eixos x,y e que a intensidade da cor verde, representado por f_G , pode assumir valores de 0 a 255 numa imagem de 8 bits por pixel, onde 0 é a ausência de cor e 255 é a intensidade máxima da cor verde.

Figura 2.1 - Componente verde da imagem de Lenna, apresentando a localização dos eixos adotados e os limites de f_g correspondente à intensidade da cor no pixel.



Fonte: Adaptado de USC-SIPI Image Database.

2.1.2 Modelo de Cores

O objetivo de um modelo de cores é facilitar a especificação das cores em alguma forma padronizada, amplamente aceita. Essencialmente, um modelo de cores é uma especificação de um sistema de coordenadas e um subespaço dentro desse sistema no qual cada cor é representada por um único ponto (GONZALES et al., 2010).

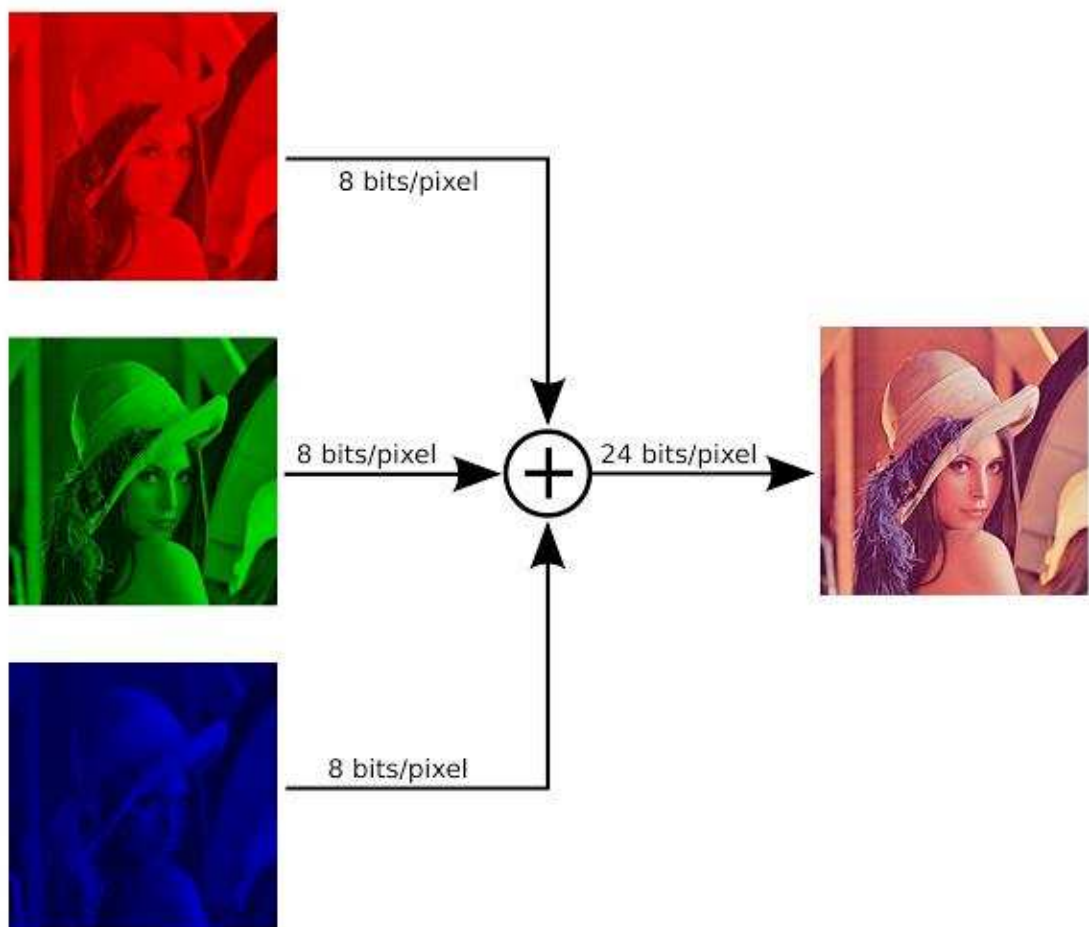
Na área de processamento digital de imagens, os modelos mais usados atualmente são o modelo RGB (*red, green, blue* – vermelho, verde, azul) para monitores coloridos e câmeras de vídeo em cores, o modelo CMY (*cyan, magenta, yellow* – ciano, magenta, amarelo) e o modelo CMYK (*cyan, magenta, yellow, black* – ciano, magenta, amarelo, preto) para a impressão colorida, o modelo HSI (*hue, saturation, intensity* – matiz, saturação, intensidade), que corresponde estreitamente à forma como os humanos descrevem e interpretam, as cores; e o modelo YUV, que foi utilizado amplamente na transição da televisão analógica para colorida. Além desses, existem inúmeros modelos de cores em uso atualmente, em virtude da ciência constituir um campo amplo que engloba muitas áreas de aplicação.

A seguir, é feita uma breve introdução sobre o Modelo RGB, ao qual é utilizado em todo o trabalho.

2.1.2.1 Modelo RGB

No modelo de cores RGB (do inglês Red, Green, Blue), exemplificado na Figura 2.2, cada pixel possui uma componente da cor vermelha, verde e azul. Geralmente, cada componente é representada por 8 bits, ou seja, cada cor pode possuir até 256 níveis diferentes. Os três valores combinados (24 bits) podem representar 16.777.216 cores diferentes.

Figura 2.2 - Exemplo das componentes de uma imagem RGB formando uma imagem colorida.



Fonte: Adaptado de USC-SIPI Image Database.

2.2 Compressão de Imagens

2.2.1 Introdução

Um dos maiores desafios da área de processamento digital de imagens é contornar o problema da grande quantidade de bytes necessários para armazenar ou transmitir à distância uma imagem digitalizada. Esse problema é agravado quando se trabalha com vídeo digital e ambos os casos se utilizam de algoritmos e técnicas que permitam a redução do volume de dados exigido para representar uma imagem ou sequência de imagens. As técnicas de compressão buscam esse feito partindo de uma premissa básica, que é a eliminação de dados redundantes. Estas técnicas costumam ser divididas em dois grupos: aquelas nas quais toda informação original é preservada (técnicas de compressão sem perdas) e aquelas em que ocorre uma perda tolerável de informação, com o objetivo de atingir maiores taxas de compressão (técnicas de compressão com perdas) (MARQUES et al., 1999).

2.2.2 Compressão Sem Perdas

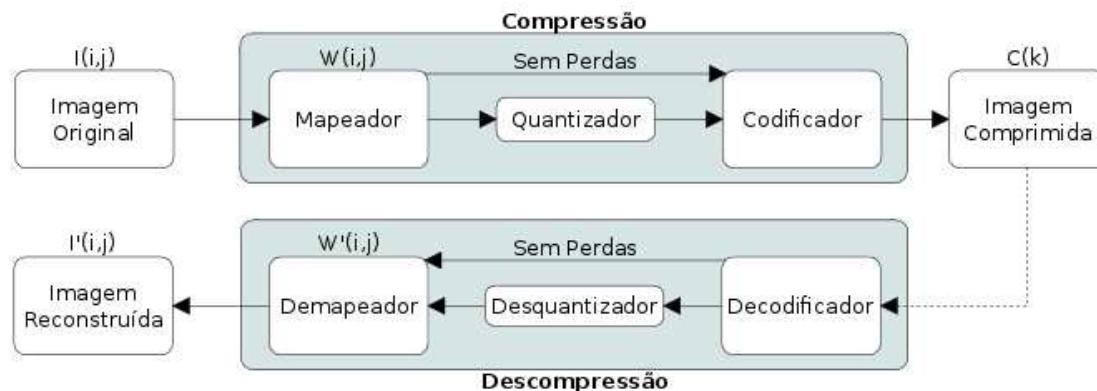
Nas aplicações que envolvam transmissão de textos, banco de dados, programas executáveis ou imagens médicas digitais, por exemplo, é extremamente importante que as informações sejam exatamente preservadas quando comprimidas. Para isso, se faz necessário o uso de compressão sem perdas. Algumas técnicas que podem ser utilizadas sem que haja perda são: Codificação de Huffman, Codificação Aritimética, Codificação por Plano de Bits, RLE (Run Length Encoder), entre outras (GONZALES et al., 2010).

2.2.3 Compressão Com Perdas

Diferentemente da compressão sem perdas, a compressão com perdas não prioriza a exatidão de suas informações e sim procura estabelecer uma maior taxa de compressão de dados. No entanto, existem casos onde mesmo com a alta taxa de compressão sua perda é imperceptível a olho nu (SANCHES, 2001).

Geralmente, as técnicas de compressão com perdas possuem as seguintes etapas: mapeamento da imagem, quantização e codificação. A Figura 2.3 demonstra o processo de compressão e descompressão:

Figura 2.3 - Diagrama em blocos do processo de compressão e descompressão de imagens.



Fonte: Adaptado de SANCHES, 2001.

Partindo de uma imagem original $I(i, j)$, o bloco mapeador descorrelaciona os dados da imagem, mapeando os dados para um novo domínio $W(i, j)$. Geralmente, a implementação desse bloco é feita por uma transformada. Algumas transformadas que podem ser utilizadas são a Transformada de Karhunen-Loève (KLT), Transformada de Fourier Discreta (DFT), Transformada do Cosseno Discreta (DCT), Transformada de Walsh-Hadamard (WHT) e a Transformada de Wavelet Discreta (DWT). Neste trabalho, foi utilizada a DWT, sendo abordada no capítulo seguinte.

Na etapa de quantização, costuma-se aplicar um limiar com o objetivo de remover os coeficientes iguais ou próximos de zero, aumentando a taxa de compressão. Em razão disso, esta etapa geralmente implica na perda de informações, impedindo a reconstrução exata dos dados.

Outra maneira de se quantizar é determinar a quantidade de bits para representar os coeficientes. Quanto mais bits forem utilizados para representar um valor, mais precisa será sua representação. Em uma imagem, caso o número de bits seja muito pequeno, pode acarretar na perda de qualidade visual, mais especificamente, na distinção das cores. Abaixo, pode-se observar exemplos desse tipo de quantização.

Figura 2.4 – Resultado de uma compressão utilizando no processo de quantização (a) 8 bits, (b) 12 bits, (c) 16 bits e (d) 20 bits, para representar os coeficientes.



Fonte: Adaptado de USC-SIPI Image Database.

Na Figura (2.4) é perceptível que as imagens (c) e (d) são muito parecidas, porém se for desejado uma maior compressão, a letra (c) será a melhor escolha, pois consegue garantir uma boa qualidade utilizando menos informações.

Quando não ocorrer nenhuma quantização, é possível recuperar completamente os dados originais. Logo, em uma compressão sem perdas o bloco quantizador é descartado.

Por fim, a etapa de codificação tem a função de transformar os dados para uma representação unidimensional $C(k)$, podendo ser utilizada uma das técnicas de codificação, tais como: codificação de Huffman, Shannon-Fano ou aritmética.

Já a reconstrução da imagem $I'(i,j)$ é realizada efetuando as operações inversas do processo de compressão.

2.2.4 Razão de Compressão

Uma maneira de medir quanto a imagem foi comprimida é utilizando a razão (ou taxa) de compressão, que é a relação entre o tamanho da imagem original com o tamanho da imagem comprimida, sendo definida por:

$$C = \frac{b}{b'} \quad (2.2)$$

Onde b significa o número de bits usados para representar uma imagem antes da compressão e b' após a compressão.

A partir disso, calcula-se a porcentagem de dados redundantes da seguinte forma:

$$R = 1 - \frac{1}{C} \quad (2.3)$$

Dessa maneira, se uma imagem de 512x512 pixels que esteja no modelo RGB utilizando 24 bits/pixel for comprimida para 1.572.864 bits, a razão de compressão será de:

$$C = \frac{512 \times 512 \times 24}{1572864} = 4$$

Uma razão de compressão como 4 (ou 4:1) significa que a imagem original tem 4 bits para cada bit na imagem comprimida. A redundância correspondente é de:

$$R = 1 - \frac{1}{4} = 75\%$$

2.2.5 Critério de Fidelidade

A compressão de dados de uma imagem pode resultar numa perda de informação visual qualitativa. Ao visualizar essa perda, não existe um parâmetro de comparação, uma vez que cada pessoa enxerga e avalia ao seu modo. Em alguns casos, a redução das informações

não é perceptível a olho nu e mesmo assim é necessária uma classificação da mudança. Para isso, é utilizado um critério de avaliação dessa perda, podendo ser objetivo ou subjetivo.

2.2.5.1 Critério de Fidelidade Objetivo

Nos métodos de critério de fidelidade objetivo, é possível quantificar a perda de informações de um processo de compressão através de uma função matemática, reduzindo o tempo e o custo. Alguns dos métodos mais conhecidos são o Erro Médio Quadrático (Root Mean Square Error - RMSE), Relação Sinal Ruído (Signal to Noise Ratio - SNR) e a Relação Sinal Ruído de Pico (Peak Signal to Noise Ratio - PSNR).

Neste trabalho, o método utilizado para avaliar a qualidade da imagem foi o PSNR, que é uma relação entre o máximo possível de potência de um sinal, pela potência do ruído, comparando o sinal antes e depois de um processo de degradação, sendo *dB* (decibel) a unidade de medida. Matematicamente o PSNR de uma imagem é representado por:

$$PSNR = 10 \times \log_{10} \left(\frac{MAX_I^2}{MSE} \right) = 20 \times \log_{10} \left(\frac{MAX_I}{\sqrt{MSE}} \right) \quad (2.4)$$

Onde MAX_I é a intensidade máxima possível da cor em um pixel (no caso de uma imagem RGB de 8 bits por cor equivale a 255) e MSE (Mean Square Error) pode ser calculado por:

$$MSE = \frac{1}{m \times n} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} \|I(i, j) - I'(i, j)\|^2 \quad (2.5)$$

Onde m e n definem a dimensão da imagem avaliada, I' é a imagem descomprimida e I é a imagem original.

Analisando a Equação 2.4 é possível perceber que quanto maior é a relação entre a potência do sinal pela potência do ruído, maior será o valor do PSNR, significando uma melhor qualidade da imagem.

2.2.5.2 Critério de Fidelidade Subjetivo

Apesar do critério de fidelidade objetivo proporcionar uma maneira prática e simples de avaliar a perda de informações, as imagens descomprimidas são vistas por seres humanos. Desse modo, medir a qualidade da imagem pelas avaliações subjetivas das pessoas costuma ser fundamental, o que é feito apresentando uma imagem descomprimida a um grupo e calculando a média de suas avaliações (GONZALEZ et al., 2010). A seguir, a Tabela 2.1 apresenta os níveis do critério subjetivo em imagens:

Tabela 2.1 – Escala de Classificação da Television Allocation Study Organization.

Valor	Classificação	Descrição
1	Excelente	Uma imagem de qualidade extremamente alta, o melhor que se pode desejar.
2	Boa	Uma imagem de alta qualidade, proporcionando uma experiência visual agradável. A interferência não chega e incomodar.
3	Razoável	Uma imagem de qualidade aceitável. A interferência não chega a incomodar.
4	No limite	Uma imagem de baixa qualidade. Você gostaria que fosse melhor. A interferência incomoda um pouco.
5	Inferior	Uma imagem muito ruim, mas é possível assistir. A interferência definitivamente incomoda.
6	Inutilizável	Uma imagem tão ruim que você não assistiria.

Fonte: FRENDELL et al., 1960.

2.3 Transformada de Wavelet

2.3.1 Introdução

As Transformadas de Wavelets podem ser vistas como mecanismos para decompor ou quebrar sinais nas suas partes constituintes, permitindo analisar os dados em diferentes domínios de frequências, estudando cada parte em uma resolução diferente. Além disso, na análise de wavelets, são usadas funções que estão contidas em regiões finitas, tornando-as convenientes na aproximação de dados contínuos (SOARES, 2009).

O estudo aprofundado da Transformada de Wavelet é bastante recente e se desenvolveu principalmente nos últimos trinta anos através de contribuições de pesquisadores das mais diversas áreas como Matemática, Física, Estatística, Computação, Geofísica e Engenharia (SOARES, 2009).

A primeira literatura que se relaciona com wavelets é sobre a Wavelet de Haar. Ela foi proposta pelo matemático Alfréd Haar em 1909, entretanto o conceito de Wavelet ainda não existia naquela época, sendo proposto só em 1981 pelo geofísico Jean Morlet. Já o termo Wavelet foi inventado em 1984 pelo próprio Morlet e o físico Alex Grossman. Até essa época, a Wavelet de Haar foi a única base ortonormal de wavelets conhecida (GRAPS, 1995).

Em 1985, Stephane Mallat deu um grande salto através de seu trabalho em processamento digital de sinais, descobrindo relações entre filtros de quadratura espelhada (*quadrature mirror filters* – QMF), algoritmos piramidais e bases de wavelets ortonormais. Y. Meyer, inspirado nos trabalhos de Mallat, construiu a primeira wavelet não trivial. Ao contrário das Wavelets de Haar, as Wavelets de Meyer são continuamente diferenciáveis, no entanto eles não têm suporte compacto. Em 1986, a introdução da análise em multiresolução e a Transformada Rápida de Wavelet rápida por Mallat e Meyer forneceu a conexão entre filtros sub-bandas e wavelets. Dois anos depois, em 1988, usando o trabalho de Mallat, Ingrid Daubechies construiu uma família de wavelets com suporte compacto. Desde então, muitos trabalhos têm sido desenvolvidos em diferentes áreas, sendo uma delas, as aplicações em compressão de imagens digitais (SANCHES, 2001).

2.3.2 Transformada de Wavelet Discreta em 1D

Aplicar a transformada de um sinal consiste em mapear o seu espaço vetorial original para o espaço vetorial das wavelets. A equação que descreve essa transformação para o espaço unidimensional pode ser descrita por:

$$Y_{1D} = T_{1D} \cdot X_{1D} \quad (2.6)$$

Onde Y_{1D} é a Transformada de Wavelet do sinal amostrado em uma dimensão, T_{1D} é a matriz transformação e X_{1D} é o próprio sinal amostrado de uma dimensão. A seguir nos tópicos 2.3.3 e 2.3.4 será estudado como encontrar a matriz T_{1D} .

2.3.3 As Wavelets de Haar e Daubechies

T_H é a matriz que transforma vetores na base canônica $((1,0), (0,1))$ em vetores (wavelets) na base de Haar $H = \left\{ \left(\frac{\sqrt{2}}{2} \quad \frac{\sqrt{2}}{2} \right), \left(\frac{\sqrt{2}}{2} \quad -\frac{\sqrt{2}}{2} \right) \right\}$. Ela é empregada para mapear sinais com duas amostragens e será explicado como encontra-la adiante.

$$T_H = \begin{bmatrix} \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} \\ \frac{\sqrt{2}}{2} & -\frac{\sqrt{2}}{2} \end{bmatrix}$$

Por terem características semelhantes, as wavelets de Haar serão consideradas como um caso das Wavelets de Daubechies com 2 pontos, as quais serão abordadas nesse capítulo.

A partir do estudo de I. Daubechies, as propriedades de q pontos foram definidas e apresentadas a seguir.

Uma wavelet de q pontos quando imersa em um espaço vetorial de dimensão maior ou igual a q tem apenas q coordenadas diferentes de zero.

Basicamente existem duas wavelets, uma do tipo S e outra do tipo D . Em que $(a_1, a_2, \dots, a_{q-1}, a_q)$ correspondem ao tipo S e $(a_q - a_{q-1}, \dots, a_2 - a_1)$ ao tipo D . Sendo $(b_1, b_2, \dots, b_{q-1}, b_q)$ a wavelet do tipo D , tem-se $(b_1, b_2, \dots, b_{q-1}, b_q) = (a_q - a_{q-1}, \dots, a_2 - a_1)$, expressa na propriedade (2.7) a seguir.

As propriedades abaixo definem as Wavelets de Daubechies de q pontos (onde $q = 4, 6, 8, \dots$), no entanto, as Wavelets de Daubechies de 2 pontos são definidas apenas pelas propriedades (2.7), (2.8) e (2.13):

$$(b_1, b_2, \dots, b_{q-1}, b_q) = (a_q, -a_{q-1}, \dots, a_2, -a_1) \quad (2.7)$$

$$\sum_{i=1}^q a_i^2 = \sum_{i=1}^q b_i^2 = 1 \quad (2.8)$$

$$\sum_{i=1}^{q-2r} a_i a_{i+2r} = 0, r = 1, 2, \dots, \left(\frac{q}{2} - 1\right) \quad (2.9)$$

$$\sum_{i=1}^{q-2r} a_i b_{i+2r} = 0, r = 1, 2, \dots, \left(\frac{q}{2} - 1\right) \quad (2.10)$$

$$\sum_{i=1}^{q-2r} b_i a_{i+2r} = 0, r = 1, 2, \dots, \left(\frac{q}{2} - 1\right) \quad (2.11)$$

$$\sum_{i=1}^{q-2r} b_i b_{i+2r} = 0, r = 1, 2, \dots, \left(\frac{q}{2} - 1\right) \quad (2.12)$$

$$\sum_{i=1}^{q-2r} b_i i^r = 0, r = 0, 1, 2, \dots, \left(\frac{q}{2} - 1\right) \quad (2.13)$$

A propriedade (2.8) indica que as duas wavelets dos tipos S e D são normais, isto é, têm módulo 1.

As propriedades (2.9), (2.10), (2.11) e (2.12) permitem que seja possível construir matrizes ortogonais (as quais serão abordadas no próximo tópico) formando as linhas desta

matriz com deslocamentos de ordem par das duas wavelets dos tipos S e D , que operam em espaços de dimensão pares maiores ou iguais a q .

A propriedade (2.13) significa que as wavelets do tipo D preservam os momentos nulos até a ordem $(\frac{q}{2} - 1)$. Ou seja, as linhas formadas por deslocamentos da wavelet do tipo D transformam sinais polinomiais de grau menor ou igual $(\frac{q}{2} - 1)$ em coordenadas nulas, e sinais que são aproximados por polinomiais em coordenadas quase nulas. Isso permite o uso destas matrizes para compressão de dados.

As soluções destes sistemas definido pelas propriedades acima definem as bases de Wavelets de Daubechies de q pontos. Por exemplo, a seguir são apresentadas as bases das wavelets dos tipos S e D para $q = 2$ e $q = 4$:

$q = 2$:

$$(a_1, a_2) = \left(\frac{\sqrt{2}}{2}, \frac{\sqrt{2}}{2} \right) \quad (2.14)$$

$$(b_1, b_2) = \left(\frac{\sqrt{2}}{2}, -\frac{\sqrt{2}}{2} \right) \quad (2.15)$$

$q = 4$:

$$(a_1, a_2, a_3, a_4) = \left(\frac{1 + \sqrt{3}}{4\sqrt{2}}, \frac{3 + \sqrt{3}}{4\sqrt{2}}, \frac{3 - \sqrt{3}}{4\sqrt{2}}, \frac{1 - \sqrt{3}}{4\sqrt{2}} \right) \quad (2.16)$$

$$(b_1, b_2, b_3, b_4) = \left(\frac{1 + \sqrt{3}}{4\sqrt{2}}, -\frac{3 + \sqrt{3}}{4\sqrt{2}}, \frac{3 - \sqrt{3}}{4\sqrt{2}}, -\frac{1 - \sqrt{3}}{4\sqrt{2}} \right) \quad (2.17)$$

Essas e as outras bases de Daubechies são fundamentais para gerar as matrizes de transformação, que serão utilizadas no cálculo da transformada de Wavelet.

2.3.4 Matriz de Transformação

Para um sinal de dimensão N , a sua matriz transformação T deve ser construída com uma dimensão $N \times N$, sendo formadas por wavelets do tipo S que formam um subespaço de

dimensão $\frac{N}{2}$ e por wavelets do tipo D , que possuem a capacidade de preservar momentos. Caso seja necessário analisar o sinal em diferentes escalas ou resoluções deve-se aplicar o processo de auto resolução, que consiste em separar novamente a transformada gerada pelas wavelets do tipo S em outros dois tipos: S e D da segunda auto-resolução. Ou seja, auto resolução é o processo de dividir os dados do tipo S em 2 até que não seja mais desejado ou possível.

A matriz transformação T será obtida após o produto das matrizes que contém cada auto resolução, ao qual é chamado de R_n , onde n é a ordem da resolução. Para uma dimensão, T pode ser expresso da seguinte maneira:

$$T_{1D} = R_n \cdot R_{n-1} \cdot \dots \cdot R_2 \cdot R_1 \quad (2.18)$$

Para formar a matriz R_l são usadas as duas wavelets básicas dos tipos S e D , uma em cada linha, repetindo e transladando de dois em dois pontos formando outras wavelets dos tipos S e D até preencher sua dimensão N . Quando $q > 2$ a translação das wavelets básicas provocam nas últimas fileiras do subespaço um estouro das colunas, sendo este problema resolvido com a aplicação das condições de contorno periódicas, isto é, os coeficientes que não couberem na linha são transladados para o início da mesma linha.

Se for desejado aplicar mais de uma auto resolução, o R_n será formado por uma sub-matriz de dimensão $\frac{N}{n} \times \frac{N}{n}$, construída nos mesmos moldes de R_l em que as wavelets dos tipos S e D são transladadas de dois em dois pontos. Além disso, a matriz R_n conta com uma sub-matriz identidade, estando localizada na diagonal da sub-matriz anterior, com a função de conservar os coeficientes do tipo D da auto resolução $n-1$. Todos os outros elementos são igualados a zero.

O exemplo abaixo na Figura 2.5 mostra como encontrar a matriz transformação com uma Wavelet de Daubechies de 4 pontos, em um espaço de dimensão 12, sendo R_2 e R_1 a matriz de segunda e primeira auto resolução, respectivamente.

A fim de facilitar a programação dos produtos matriciais as matrizes R_n foram organizadas de tal forma que as wavelets do tipo S sejam postas na primeira metade das linhas, e as do tipo D nas linhas seguintes.

$$T_{1D} = R_2 \cdot R_1 \quad (2.19)$$

Onde R_1 e R_2 estão explicitados na Figura a seguir:

Figura 2.5 – Exemplo de como construir R_1 e R_2 , para aplicar a Transformada de Wavelet de Daubechies de 4 pontos, em um espaço de dimensão 12.

Legenda:
■ Wavelets do tipo S
■ Wavelets do tipo D
■ Sub-matriz Identidade

$$R_1 = \begin{pmatrix} \begin{array}{cccccccccccc} a_1 & a_2 & a_3 & a_4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & a_1 & a_2 & a_3 & a_4 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & a_1 & a_2 & a_3 & a_4 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & a_1 & a_2 & a_3 & a_4 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & a_1 & a_2 & a_3 & a_4 \\ a_3 & a_4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & a_1 & a_2 \end{array} \\ \begin{array}{cccccccccccc} a_4 & -a_3 & a_2 & -a_1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & a_4 & -a_3 & a_2 & -a_1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & a_4 & -a_3 & a_2 & -a_1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & a_4 & -a_3 & a_2 & -a_1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & a_4 & -a_3 & a_2 & -a_1 \\ a_2 & -a_1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & a_4 & -a_3 \end{array} \end{pmatrix}$$

$$R_2 = \begin{pmatrix} \begin{array}{cccccc} a_1 & a_2 & a_3 & a_4 & 0 & 0 \\ 0 & 0 & a_1 & a_2 & a_3 & a_4 \\ a_3 & a_4 & 0 & 0 & a_1 & a_2 \end{array} & \begin{array}{cccccc} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{array} \\ \begin{array}{cccccc} a_4 & -a_3 & a_2 & -a_1 & 0 & 0 \\ 0 & 0 & a_4 & -a_3 & a_2 & -a_1 \\ a_2 & -a_1 & 0 & 0 & a_4 & -a_3 \end{array} & \begin{array}{cccccc} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{array} \\ \begin{array}{cccccc} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{array} & \begin{array}{cccccc} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{array} \end{pmatrix}$$

Fonte: Autor

2.3.5 Transformada de Wavelet Discreta em 2D

Aplicar a transformada de um sinal $2D$ consiste em mapear o espaço vetorial original de um sinal, podendo ser uma imagem, para o espaço tensorial de ordem 2 das wavelets. Para isso, são utilizadas duas matrizes de transformação com dimensões m e n , definidas da mesma maneira que uma matriz transformação $1D$ com uma auto resolução. Em outras palavras, será

aplicada a Transformada de Wavelet nas linhas e colunas a fim de gerar uma transformação 2D. Sabendo que T_L e T_C (matriz transformação das linhas e das colunas) são definidos por:

$$T_L = \begin{bmatrix} e_1 \\ e_2 \\ e_3 \\ \vdots \\ e_i \\ \vdots \\ e_m \end{bmatrix}_{m \times m} \quad T_C = \begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ \vdots \\ f_j \\ \vdots \\ f_n \end{bmatrix}_{n \times n} \quad (2.20) \text{ e } (2.21)$$

Onde e_i ($i=1, 2, \dots, m$) e f_j ($j=1, 2, \dots, n$) são os vetores da base escolhida (no nosso caso, Haar), m é número de linhas e n é o número de colunas da imagem. Além disso, considera-se que:

$$e_i = [e_{i1} \quad e_{i2} \quad \dots \quad e_{im}] \quad (2.22)$$

$$f_j = [f_{j1} \quad f_{j2} \quad \dots \quad f_{jn}] \quad (2.23)$$

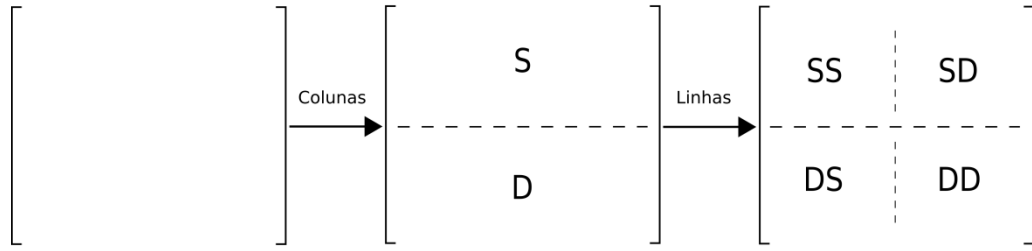
Logo, é possível relacionar a equação que descreve essa transformação entre tensores como:

$$Y_{ij} = \sum_{p=1}^n \sum_{q=1}^m e_{ip} \cdot f_{jq} \cdot X_{pq} \quad (2.24)$$

Onde Y_{ij} é cada elemento da Transformada de Wavelet da imagem amostrada, i e j são a linha e a coluna do respectivo elemento e X_{pq} é o elemento do sinal de duas dimensões, sendo p sua linha e q sua coluna.

A Transformada de Wavelet para duas dimensões assim como para uma dimensão, também produzirá coeficientes do tipo S e do tipo D , entretanto, como estão sendo aplicadas duas transformadas (uma para as linhas e outra para as colunas), serão produzidos sinais do tipo SS , SD , DS e DD , como mostra a figura abaixo:

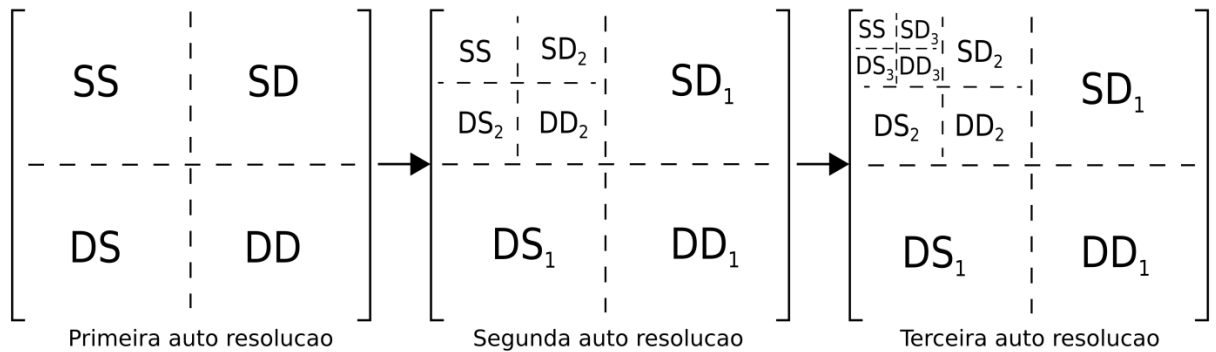
Figura 2.6 – Resultado da aplicação (nas colunas e linhas respectivamente) da Transformada na matriz que representa o sinal 2D, gerando ondas do tipo *SS*, *SD*, *DS* e *DD*.



Fonte: Autor

Além disso, quando é necessário realizar outra auto resolução deve-se transformar apenas os coeficientes do tipo *SS*, como se fosse outra imagem de dimensão $\frac{m}{2} \times \frac{n}{2}$. Ou seja, é necessário produzir duas novas matrizes de transformação T_L e T_C , com novas dimensões $\frac{m}{2}$ e $\frac{n}{2}$, descritas no tópico passado.

Figura 2.7 – Exemplo de como aplicar as auto resoluções.



Fonte: Autor

2.3.6 Transformada Inversa de Wavelet

Como foram utilizadas as wavelets para transformar os sinais amostrados, será necessária uma forma para reconstruir o sinal transformado. É por meio da transformada inversa de Wavelet que será possível essa reconstrução.

No caso de sinais com espaço unidimensional a inversa segue a seguinte equação:

$$X'_{1D} = T_{1D}^T \cdot Y_{1D} \quad (2.25)$$

Em que X'_{1D} é o sinal de uma dimensão reconstruído.

No caso de sinais com espaço bidimensional a inversa segue a seguinte equação:

$$X'_{ij} = \sum_{p=1}^n \sum_{q=1}^m e_{pi} \cdot f_{qj} \cdot Y_{pq} \quad (2.26)$$

Capítulo 3

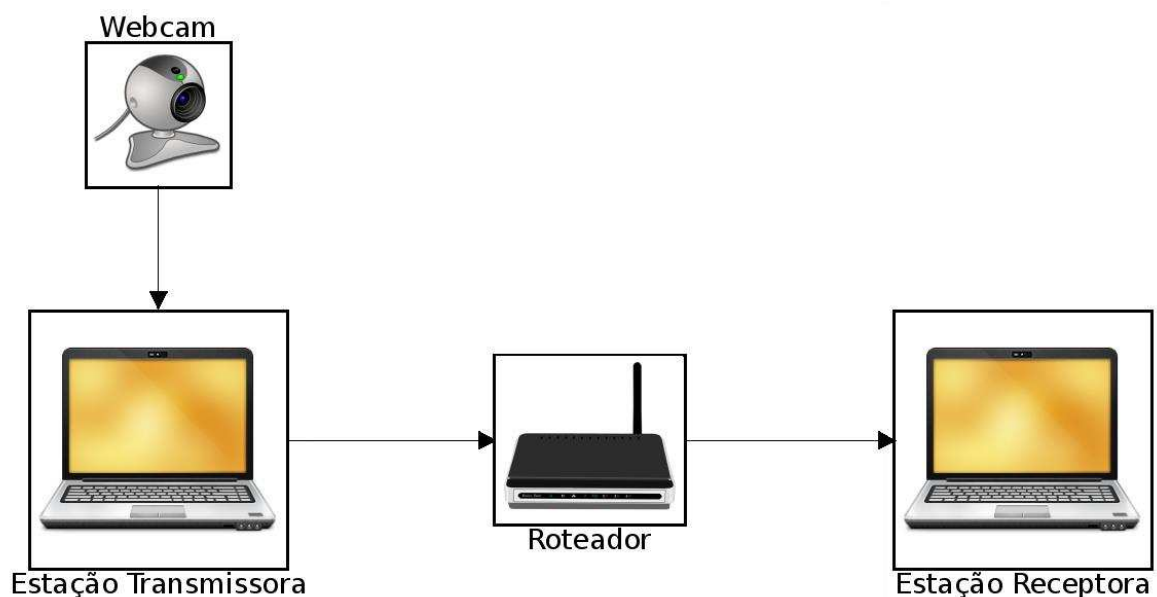
Materiais, Métodos e Resultados

Este capítulo irá abordar todo o procedimento necessário para realização dos objetivos deste trabalho. Nele será explicado a implementação do algoritmo desenvolvido, envolvendo os processos de compressão, transmissão, recepção e descompressão. Além disso, será descrito os materiais usados e os resultados obtidos.

3.1 Materiais

Para realização dos testes, foi utilizada uma rede *wireless* de computadores, que possui um sistema de câmera conectado via USB ao computador transmissor a fim de realizar o envio das imagens comprimidas. A rede conta também com outro computador para receber e descomprimir dos dados transmitidos.

Figura 3.1 – Sistema utilizado para realização dos testes de compressão, transmissão, recepção e descompressão de uma imagem.



Fonte: Adaptação de Pixabay, FancyIcons e InconSeeker.

O algoritmo foi desenvolvido com a linguagem de programação C++ e no processo de aquisição de imagens da câmera é utilizado a biblioteca OpenCV.

A transmissão e recepção dos dados podem ser implementadas utilizando os protocolos UDP ou TCP, os quais são pertencentes à 4ª camada (camada de transporte) do modelo OSI. Entretanto, o UDP se destaca para o tipo de aplicação deste trabalho, pois mesmo não existindo a garantia de entrega de pacotes (como é o caso do TCP) possui alta velocidade na troca de informações. A garantia de entrega, além de causar lentidão, não tem muita importância para uma aplicação de transmissão de imagens em tempo real, uma vez que milhares de pacotes são enviados e a perda de alguns poucos não interferem na visualização das imagens. Diante disso, foi utilizada a biblioteca QUdpSocket, que é uma biblioteca nativa do Qt Creator, a qual permite realizar uma conexão UDP entre os dois computadores.

3.2 Métodos

Os métodos realizados neste trabalho se baseiam no diagrama da Figura 2.3, em que o processo é iniciado com o algoritmo de compressão buscando efetuar as etapas de mapeamento, quantização e codificação.

Para implementação do bloco mapeador, antes deve ser calculado a base que define as Wavelets de Haar através das propriedades (2.7), (2.8) e (2.13). Com isso, foram obtidas as seguintes ondas descritas em (2.14) e (2.15).

A partir disso, no algoritmo, a Transformada de Wavelet é calculada baseada na equação (2.24) juntamente com a amostragem da imagem adquirida no formato de uma matriz. Ou seja, é calculada uma transformada bidimensional para cada camada de cor da imagem no modelo RGB. Caso seja necessário uma nova auto resolução, é preciso transformar novamente os dados do tipo *SS* e armazenar os coeficientes do tipo *SD*, *DS* e *DD*. Vale ressaltar que a necessidade de mais uma auto resolução se dá pelo objetivo de obter maiores taxas de compressão, pois os coeficientes do tipo *SS* são os que possuem maiores relevâncias e conforme é realizado mais uma auto resolução o número de coeficientes do tipo *SS* é dividido por 4.

A equação (2.24) apesar de funcional tem um custo computacional muito grande e pode ser melhorada reduzindo as inúmeras multiplicações com resultados iguais a zero. Logo,

com o intuito de melhorar a velocidade obtendo o mesmo resultado, a transformada foi implementada da seguinte forma:

Algoritmo 3.1 - Transformação da Imagem

ENTRADA: X (Imagem armazenada em um vetor unidimensional), m (número de linhas), n (número de colunas), ar (número de auto resoluções)

ENQUANTO $ar \leq 3$ **FAÇA**

PARA $i \leftarrow 0$ **ATÉ** $\left(\frac{m}{2} \times \frac{n}{2}\right) - 1$

$$Y_{SS}[i][ar] = (+X[2i] + X[2i+1] + X[2i+m] + X[2i+1+m]) \div 2$$

$$Y_{SD}[i][ar] = (+X[2i] - X[2i+1] + X[2i+m] - X[2i+1+m]) \div 2$$

$$Y_{DS}[i][ar] = (+X[2i] + X[2i+1] - X[2i+m] - X[2i+1+m]) \div 2$$

$$Y_{DD}[i][ar] = (+X[2i] - X[2i+1] - X[2i+m] + X[2i+1+m]) \div 2$$

FIM PARA

$$X \leftarrow Y_{SS}[i]$$

$$m \leftarrow \frac{m}{2}$$

$$n \leftarrow \frac{n}{2}$$

$$ar \leftarrow ar + 1$$

FIM ENQUANTO

Vale ressaltar que a informação da imagem é disponibilizada em um vetor unidimensional pelo OpenCV, em que a cada 640 elementos temos uma nova linha. Diante disso, também foi decidido armazenar os coeficientes em vetores para facilitar a transmissão.

Com a possibilidade de realizar a transformada de Wavelet com o número de auto resoluções desejadas, a implementação do processo de mapeamento é encerrada. Com isso, se inicia o processo de quantização, em que em uma compressão com perdas deve-se aplicar a técnica de limiarização. A técnica consiste em criar um histograma para analisar os coeficientes (do tipo SD , DS e DD) que serão transmitidos, determinando o limiar de corte (*threshold*) e excluindo os coeficientes de menor importância. Para isso, com a quantidade de bits utilizada por cada coeficiente juntamente com a taxa de compressão fornecida pelo usuário é calculado o limiar a fim de realizar a compressão requerida.

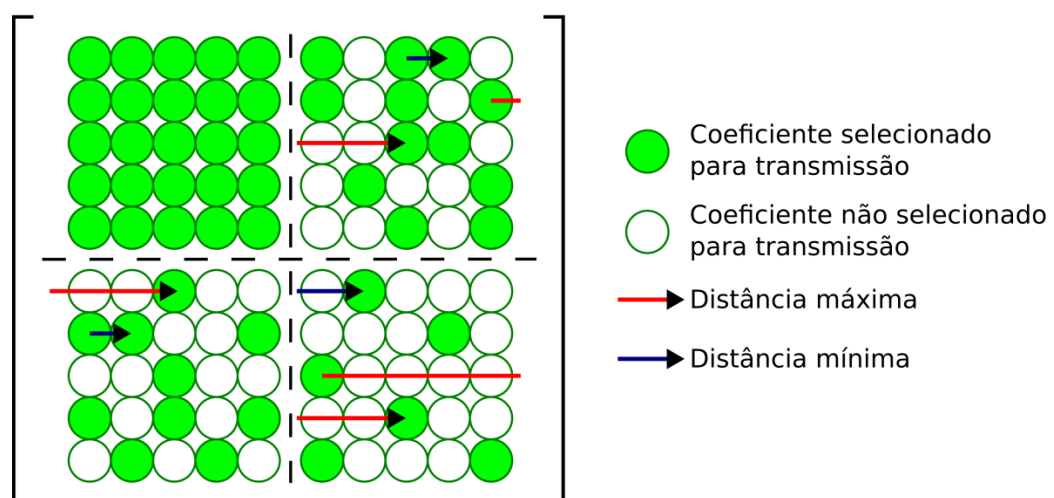
Ainda no processo de quantização, o número de bits para representar o coeficiente da transformada é determinado para que se tenha uma qualidade aceitável. Dependendo desse

número, pode-se ter como consequência a perda de qualidade visual, gerada pela falta de precisão. No entanto, se for bem escolhido, a perda pode ser insignificante a olho nu e uma maior compressão é gerada. Esse valor depende da intensidade máxima da cor, tipo de onda, auto resolução e o número de coeficientes, não sendo necessariamente distribuído igualmente entre as três cores.

Outra forma de compressão ocorre com a redução dos bits na forma de identificar os coeficientes que serão transmitidos. Com isso, ao invés de enviar as coordenadas cartesianas de cada coeficiente transmitido, é enviada a distância entre eles. Além disso, a transmissão será dividida em blocos, um para cada tipo de onda e auto resolução, a fim de determinar a quantidade de bits necessária para representar a localização dos coeficientes de cada bloco, a partir das distâncias mínima e máxima. Ou seja, para coeficientes do tipo *SS* a distância entre os transmitidos é zero, já que todos são enviados por possuírem grande quantidade de informação. Os coeficientes do tipo *DD* (que possuem valores próximos ou iguais à zero) possuem grandes distâncias entre eles, visto que poucos são transmitidos.

Por exemplo, na figura abaixo, em que foi aplicada uma auto resolução na imagem formando 4 blocos, percebe-se que todos os coeficientes do tipo *SS* são selecionados, logo não é necessário informar a distância, pois o envio dos coeficientes podem ser ordenados em sequência. Os coeficientes do tipo *DS* e *SD* possuem distância máxima igual a dois e mínima igual a zero. E os coeficientes do tipo *DD* possuem distância máxima igual a seis e mínima igual a um.

Figura 3.2 – Exemplo de como é determinado a distância mínima e máxima de um coeficiente.



Com essas informações definidas, na etapa de codificação, é possível preparar os dados em um vetor unidimensional que é usado na transmissão. Nesse vetor estão contidas as seguintes informações:

- Dimensão da imagem;
- Número de auto resoluções aplicadas;
- Mínimo e máximo dos coeficientes das wavelets do tipo *SS*, *SD*, *DS* e *DD* para cada resolução, a fim de determinar o número de bits que será utilizado para representar os coeficientes;
- Número de bits usados na distância para cada tipo de onda e auto resolução;
- Número de bits usados por cada pacote (que contém a distância e coeficientes de cada cor) de cada tipo de onda e auto resolução;
- Posições iniciais e finais dos dados das wavelets do tipo *SS*, *SD*, *DS* e *DD* para cada resolução;
- Pacotes de dados com coeficientes e distâncias.

Após a codificação, com a organização do vetor contendo as informações mencionadas, é possível realizar a transmissão dos dados.

Ao receber as informações via rede wireless, a estação base receptora irá dar início ao processo de descompressão descrito no diagrama da Figura 2.3. O primeiro passo a ser realizado na descompressão é a decodificação dos dados de acordo com o padrão enviado e a seguir é aplicada a desquantização dos dados para que seja construída a imagem transformada aproximada. Com isso, no demapeamento reconstruímos a imagem aproximada aplicando a transformada inversa descrita pela equação (2.25) e a exibindo na tela do usuário, com ajuda do OpenCV.

3.3 Resultados

Para análise dos resultados da compressão foram utilizadas algumas das imagens (no modelo de cores RGB com resolução de 512x512 pixels) mais clássicas na área de processamento digital de imagens, tais como: Lenna, Baboon, Airplane (F16) e Peppers.

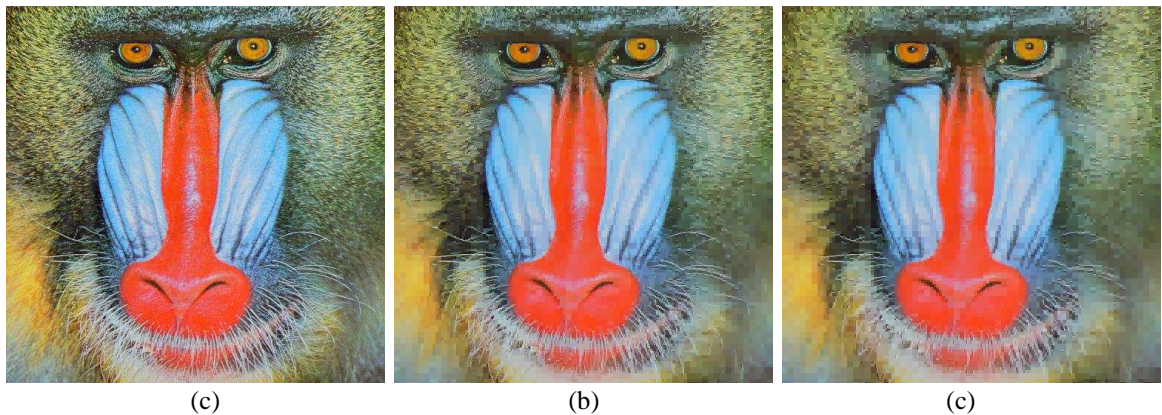
A avaliação da qualidade das imagens reconstruídas foi proporcionada pelo critério de fidelidade objetivo Relação Sinal Ruído de Pico (PSNR) definida na equação (2.4). Além disso, foi utilizada a razão de compressão definida pela equação (2.2) para quantificar o quanto foi comprimido.

Figura 3.3 – (a) Imagem Lenna Original. (b) Imagem Lenna comprimida com a razão de compressão 14,39 e PSNR 35,07 *dB*. (c) Imagem Lenna comprimida com a razão de compressão 28,91 e PSNR 33,88 *dB*.



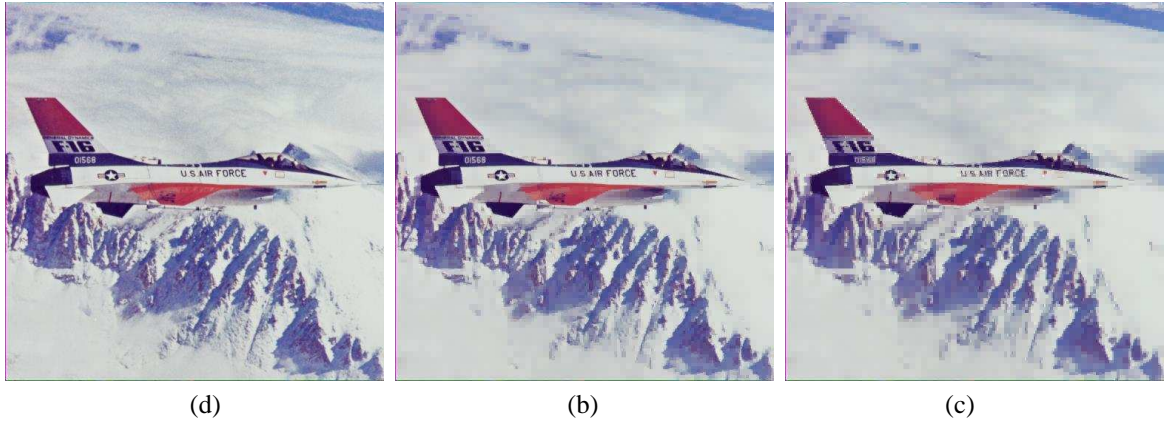
Fonte: Adaptado de USC-SIPI Image Database.

Figura 3.4 – (a) Imagem Baboon Original. (b) Imagem Baboon comprimida com a razão de compressão 13,81 e PSNR 29,98 *dB*. (c) Imagem Baboon comprimida com a razão de compressão 27,45 e PSNR 29,68 *dB*.



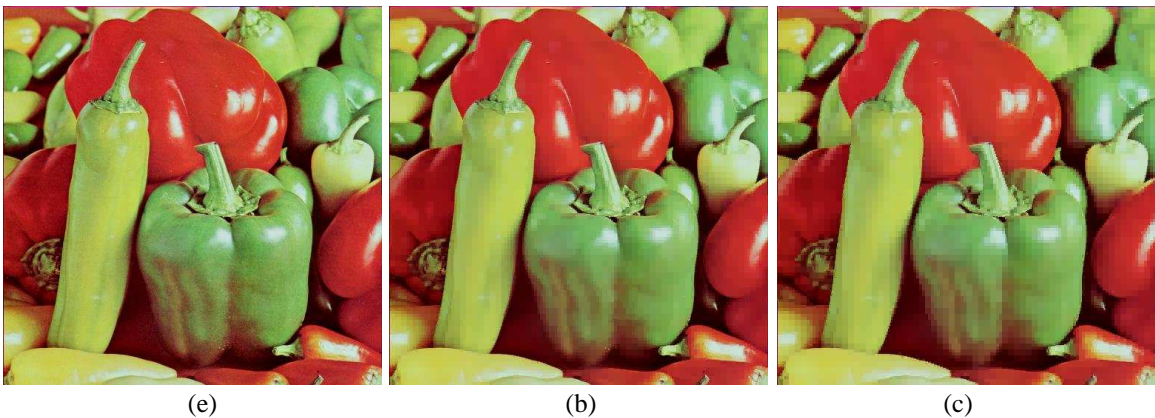
Fonte: Adaptado de USC-SIPI Image Database.

Figura 3.5 – (a) Imagem Airplane (F16) Original. (b) Imagem Airplane (F16) comprimida com a razão de compressão 13,54 e PSNR 36,06 *dB*. (c) Imagem Airplane (F16) comprimida com a razão de compressão 27,38 e PSNR 34,43 *dB*.



Fonte: Adaptado de USC-SIPI Image Database.

Figura 3.6 – (a) Imagem Peppers Original. (b) Imagem Peppers comprimida com a razão de compressão 14,01 e PSNR 34,51 *dB*. (c) Imagem Peppers comprimida com a razão de compressão 28,39 e PSNR 33,46 *dB*.



Fonte: Adaptado de USC-SIPI Image Database.

Após realizar testes de compressão em uma imagem, o próximo passo é verificar se o algoritmo é capaz de transmitir imagens de uma webcam (640x480 pixels) em tempo real. Nas etapas de compressão e transmissão o tempo médio gasto foi de 20ms com a captura de imagens. Desconsiderando a captura, com a finalidade de saber o tempo consumido apenas pelo algoritmo, o tempo cai para uma média de 12ms. Nas etapas de recepção e descompressão o tempo médio foi de 3ms.

Capítulo 4

Considerações Finais

Com a necessidade da transmissão à longa distância e em tempo real, se faz necessário uma maior banda de transmissão ou a utilização de algoritmos de compressão de imagens. Para este trabalho foi adotada a compressão de imagens a fim de resolver esse tipo de situação. Logo, foi desenvolvido um algoritmo utilizando a Transformada de Wavelet de Haar para verificar a possibilidade de aplicá-la para transmissão em tempo real.

Avaliando os resultados obtidos, é possível perceber que o algoritmo cumpriu o papel de transmissão com qualidade aceitável em tempo real, visto que a soma dos tempos de compressão, transmissão, recepção e descompressão não ultrapassaram o tempo máximo de $33,333ms$, garantindo uma sequência de imagens com $30\ frames/s$.

O algoritmo implementado, em situações futuras, pode ser aperfeiçoado e obter uma taxa de compressão ainda melhor, principalmente no processo de codificação, onde pode ser aplicada a codificação de Huffman, por exemplo. Além disso, é possível adaptar o código para testar a Transformada de Wavelet de Daubechies de 4 pontos para alcançar uma melhor qualidade visual, entretanto é necessário verificar a possibilidade de manter a transmissão em tempo real, visto que exige um custo computacional maior.

Para trabalhos posteriores, seria interessante aplicar este estudo na área de compressão de imagens 3D, a qual possui grandes dificuldades para compressão em tempo real, pois além das características de cor, a imagem possui informações de distância por cada pixel, fazendo com que a mesma fique com um volume de dados muito grande.

Referências Bibliográficas

ACHARYA, T.; RAY, A. K. **Image Processing: Principles and Applications**. Hoboken: John Wiley and Sons, 2005.

CESAR, R. M. F.; COSTA, L. F. **Neural Cell Classification by Wavelet and Multiscale Curvature**. *Biological Cybernetics*, 79(4): 322-336, 1998.

COSTA, C. A. R. **Técnicas de Compressão de Sequências de Imagens Visando Transmissão em Tempo Real**. Instituto de Matemática, Estatística, e Ciência da Computação. Curso de Pós Graduação em Ciência da Computação. Universidade Estadual de Campinas, Campinas, 1993.

DAUBECHIES, I. **The Wavelet Transform, Time-Frequency Localization and Signal Analysis**. *IEEE Trans. Information Theory*, 36, 1990.

FALCÃO, A. X.; LEITE, N. J. **Fundamentos de Processamento de Imagem Digital**. Campinas: Unicamp, 1998.

FancyIcons. Disponível em: <<http://www.fancyicons.com/free-icon/149/unleashed-icon-set/free-notebook-icon-png/>>. Acessado em: 25 de maio de 2015.

GOMES, D. M. **Compressão Seletiva de Imagens Coloridas com Detecção Automática de Regiões de Interesse**. Centro de Tecnologia. Tese de Mestrado no Programa de Pós – Graduação em Engenharia Elétrica. Universidade Federal do Rio Grande do Norte. Natal, 2005.

GONZALES, R. C.; WOODS, R. E. **Processamento de Imagens Digitais**. Pearson Prentice Hall, 3ª ed., São Paulo, 2010.

GRAPS, A. **An Introduction to Wavelets**. IEEE Computational Science & Engineering, 2(2): 50-61, USA, 1995.

IconSeeker. Disponível em: < <http://www.iconseeker.com/search-icon/refresh-cl/hardware-webcam.html>>. Acessado em: 25 de maio de 2015.

MALLAT, S. **A Theory for Multiresolution Signal Decomposition: The Wavelet Representation**. IEEE Transactions on Pattern Analysis and Machine Intelligence, 11: 674-693, 1989.

MARQUES, O. F.; VIEIRA, H. N. **Processamento Digital de Imagens**. Rio de Janeiro: Brasport, 1999.

Pixabay. Disponível em: < <http://pixabay.com/en/wireless-router-switch-network-98425/>>. Acessado em: 25 de maio de 2015.

SANCHES, I. J. **Compressão Sem Perdas de Projeções de Tomografia Computadorizada Usando a Transformada de Wavelet**. Setor de Ciências Exatas. Tese de Mestrado no Programa de Pós Graduação em Informática. Universidade Federal do Paraná. Curitiba, 2001.

SANTOS, D. T. **Compressão de Imagens Usando a Função de Peano e a Transformada Wavelet 1D**. Centro de Tecnologia. Tese de Mestrado no Programa de Pós – Graduação em Pesquisa e Desenvolvimento em Ciência e Engenharia de Petróleo. Universidade federal do Rio Grande do Norte. Natal, 2012.

SOARES, W. C. **Um Método Não-Limiar para Redução de Ruído em Sinais de Voz no Domínio Wavelet**. Automação. Tese de Doutorado no Programa de Pós – Graduação em Engenharia Elétrica. Universidade Estadual Paulista “Júlio de Mesquita Filho”. São Paulo, 2009.

USC-SIPI Image Database. Disponível em: <<http://sipi.usc.edu/database/>>. Acessado em: 01 de maio de 2015.

