



Universidade de Brasília - UnB
Faculdade UnB Gama - FGA
Engenharia Eletrônica

Codificação de Imagens em Baixo Consumo Para Comunicações em Minas Subterrâneas

Autor: André Mateus R. Dantas
Orientador: Prof. Dr. Diogo Caetano Garcia

Brasília, DF

2015



André Mateus R. Dantas

Codificação de Imagens em Baixo Consumo Para Comunicações em Minas Subterrâneas

Monografia submetida ao curso de graduação em Engenharia Eletrônica da Universidade de Brasília, como requisito parcial para obtenção do Título de Bacharel em Engenharia Eletrônica.

Universidade de Brasília - UnB

Faculdade UnB Gama - FGA

Orientador: Prof. Dr. Diogo Caetano Garcia

Coorientador: Prof. Dr. Leonardo Aguayo

Brasília, DF

2015

André Mateus R. Dantas

Codificação de Imagens em Baixo Consumo Para Comunicações em Minas Subterrâneas/ André Mateus R. Dantas. – Brasília, DF, 2015-

60 p. : il. (algumas color.) ; 30 cm.

Orientador: Prof. Dr. Diogo Caetano Garcia

Trabalho de Conclusão de Curso – Universidade de Brasília - UnB
Faculdade UnB Gama - FGA , 2015.

1. CODIFICAÇÃO. 2. COMUNICAÇÃO. I. Prof. Dr. Diogo Caetano Garcia.
II. Universidade de Brasília. III. Faculdade UnB Gama. IV. Codificação de
Imagens em Baixo Consumo Para Comunicações em Minas Subterrâneas

CDU 02:141:005.6

André Mateus R. Dantas

Codificação de Imagens em Baixo Consumo Para Comunicações em Minas Subterrâneas

Monografia submetida ao curso de graduação em Engenharia Eletrônica da Universidade de Brasília, como requisito parcial para obtenção do Título de Bacharel em Engenharia Eletrônica.

Trabalho aprovado. Brasília, DF, 26 de julho de 2015:

Prof. Dr. Diogo Caetano Garcia
Orientador

**Prof. Dr. Cristiano Jacques Miosso
Rodrigues Mendes**
Convidado 1

Prof. Dr. Tiago Alves da Fonseca
Convidado 2

Brasília, DF
2015

Resumo

A mineração subterrânea é uma das atividades mais extremas da indústria moderna, sendo realizada em ambientes relativamente perigosos. Alguns dos principais riscos estão relacionados a acidentes dentro da mina, como desabamentos, incêndios, explosões e inundações, que podem inviabilizar comunicações com fio. Para trabalhadores sujeitos a estas circunstâncias, a comunicação sem fio pode oferecer as melhores chances de sobrevivência. O presente trabalho propõe a implementação de um sistema de comunicação de imagens em baixa largura de banda e baixo consumo, permitindo o envio de informações vitais por operários presos em minas para equipes de resgate. O sistema proposto é baseado na codificação interpolativa de imagens, aonde a resolução da imagem de entrada é reduzida antes da codificação, e recuperada após a decodificação.

Palavras-chave: Codificação de Imagens. Comunicação para Minas Subterrâneas. DCT. H.264. JPEG

Abstract

Underground mining is one of the most extreme occupations from several perspectives. First of all, mining operations are carried out in very hazardous environments. Some of the main risks relate to accidents in the mine, such as roof falls , fires, explosions, floods, among others, these factors may hinder wired communiaction. For workers under these circumstances, wireless communications may offer the best chances of survival. This paper proposes the implementation of a low-power, low-bandwidth image communication system for underground mines, allowing the transmission of vital information by workers trapped in mines to rescue teams. The proposed system is based on interpolative image coding, where the input image's resolution is reduced before encoding, and recovered after decoding.

Key-words: Image Coding. Underground Mine Communications. DCT. H.264. JPEG

Lista de ilustrações

Figura 1 – Cenas: (a) natural; (b) sintética. Fonte: (1).	18
Figura 2 – Amostragem espacial e temporal de uma sequência de vídeo.	18
Figura 3 – Imagens com mesmo tamanho real e diferentes resoluções. Fonte: (2).	19
Figura 4 – Componentes no espaço RGB: (a) imagem original; (b) vermelho; (c) verde; (d) azul. Fonte: (1).	20
Figura 5 – (a) decimação $I_D(u, v)$ de uma imagem $I^D(u, v)$ de tamanho 8x8 por um fator 4; (b) operação geral de decimação: a imagem $I(u, v)$ é convoluída com o filtro $H_D(z_1, z_2)$, gerando uma versão passabaixas I_{PB1}^D , e depois decimada por um fator M, gerando uma versão $I(u, v)$ de menor resolução. Fonte: (3).	22
Figura 6 – (a) interpolação $I^I(u, v)$ de uma imagem $I(u, v)$ de tamanho 2x2 por um fator 4; (b) operação geral de interpolação: a imagem $I(u, v)$ é interpolada por um fator M e depois convoluída com o H_I , gerando uma versão em maior resolução de $I(u, v)$. Fonte: (3).	23
Figura 7 – Bloco 10x10 <i>pixels</i> da imgem de teste Lena, equivalente a matriz 2.12. Fonte: (1).	25
Figura 8 – Escala arbitrária redimensionando com uma ou ambas as transformadas locais. Nota-se que a dimensão é dada em blocos (por exemplo, o tamanho da primeira imagem é de I blocos por I blocos, onde cada bloco é de tamanho 8x8, o tamanho da segunda imagem ainda é I blocos por I blocos, mas o tamanho de cada bloco é $N \times N$). Fonte: (4).	26
Figura 9 – Etapas de codificação baseada em DCT. Fonte: (5).	28
Figura 10 – Etapas de decodificação baseada em DCT. Fonte: (5).	29
Figura 11 – Preparação dos coeficientes da DCT para o codificador de entropia. Fonte: (5).	30
Figura 12 – Estrutura típica de um codificador H.264 (6).	31
Figura 13 – Estrutura típica de um decodificador H.264 (6).	31
Figura 14 – (a) uma cena estática de uma sequência de vídeo é capturada com baixa luminosidade; (b) após equalização de histograma a placa do automóvel continua ilegível devido ao ruído natural da imagem. Fonte: (7).	33
Figura 15 – Legibilidade da placa como resultado da média do conjunto cada vez maior de quadros. Fonte: (7).	34
Figura 16 – (a) região de interesse é capturada em uma cena dinâmica; (b) a SR estima a cena subjacente a partir de 50 imagens. A reconstrução possui o triplo da resolução em relação à imagem original. Fonte: (7).	34

Figura 17 – Exemplo de separação de quadro em componentes de baixa e alta frequência.	35
Figura 18 – Forma geral da super-resolução baseada em exemplos (8).	35
Figura 19 – Super-resolução baseada em exemplos: a imagem interpolada I_0^B recebe informações de alta frequência a partir de uma imagem em alta resolução I_j separada em versões com componentes de baixa e alta frequência, I_j^B e I_j^A . Fonte: (3).	36
Figura 20 – Arquitetura de codificação em resolução mista.	36
Figura 21 – Fluxograma do algoritmo de super-resolução proposta (3).	37
Figura 22 – Diagrama de blocos proposto para solução.	39
Figura 23 – Kit de desenvolvimento <i>Raspberry Pi 2 Model B</i> (9).	40
Figura 24 – Mapa de pinos do <i>Raspberry Pi 2 Modelo B</i> (9).	41
Figura 25 – Sequências de vídeo : (a) <i>akiyo</i> , (b) <i>foreman</i> , (c) <i>bowing</i> , (d) <i>carphone</i> , (e) <i>crew</i> , (f) <i>flower</i> , (g) <i>footbal</i> e (h) <i>mobile</i>	44
Figura 26 – Curva PSNR médio x Taxa, para a sequência de vídeo <i>Football</i> , com um fator de decimação $M = 2$	46
Figura 27 – Curva PSNR médio x Taxa, para a sequência de vídeo <i>Crew</i> , com um fator de decimação $M = 2$	46
Figura 28 – <i>Foreman</i> - Quadro 6, $M = 2$, $Qscale = 10$: (a) Imagem Original, (b)Imagem Interpolada (PSNR = 21.950 dB), (c)Imagem Super-resolvida (PSNR = 21.776 dB).	49
Figura 29 – <i>Football</i> - Quadro 4, $M = 2$, $Qscale = 10$: (a) Imagem Original, (b)Imagen Interpolada (PSNR = 16.847 dB), (c)Imagen Super-resolvida (PSNR = 17.176 dB).	50
Figura 30 – <i>Mobile</i> - Quadro 4, $M = 2$, $Qscale = 10$: (a) Imagem Original, (b)Imagen Interpolada (PSNR = 16.532 dB), (c)Imagen Super-resolvida (PSNR = 17.902 dB).	51
Figura 31 – <i>Foreman</i> - Quadro 30, $M = 2$, $Qscale = 10$: (a) Imagem Original, (b)Imagen Interpolada (PSNR = 26.399 dB), (c)Imagen Super-resolvida (PSNR = 31.503 dB).	52
Figura 32 – <i>Football</i> - Quadro 3, $M = 2$, $Qscale = 10$: (a) Imagem Original, (b)Imagen Interpolada (PSNR = 22.946 dB), (c)Imagen Super-resolvida (PSNR = 24.286 dB).	53
Figura 33 – <i>Mobile</i> - Quadro 33, $M = 2$, $Qscale = 10$: (a) Imagem Original, (b)Imagen Interpolada (PSNR = 18.756 dB), (c)Imagen Super-resolvida (PSNR = 21.211 dB).	54

Lista de abreviaturas e siglas

ARM	<i>Advanced RISC Machine</i>
AVC	<i>Advanced Video Coding</i>
CCIR	<i>International Consultative Committee on Broadcasting</i>
DCT	<i>Discrete cosine transform</i>
DTTs	<i>Discrete trigonometric transforms</i>
ELF	<i>Extremely Low Frequency</i>
FDCT	<i>Forward discrete cosine transform</i>
IDCT	<i>Inverse discrete cosine transform</i>
ISO	<i>International Standards Organisation</i>
ITU	<i>International Telecommunications Union</i>
JPEG	<i>Joint Photographic Experts Group</i>
MAC	<i>Multiply and Accumulate</i>
MPEG	<i>Moving Picture Experts Group</i>
MSE	<i>Medium Square Error</i>
PSNR	<i>Peak Signal to Noise Ratio</i>
SR	Super-resolução
TTE	<i>Through-the-earth</i>

Sumário

1	INTRODUÇÃO	15
1.1	Contextualização e Justificativa	15
1.2	Objetivos	16
1.2.1	Objetivos Gerais	16
1.2.2	Objetivos Específicos	16
2	FUNDAMENTAÇÃO TEÓRICA	17
2.1	Imagens Digitais	17
2.1.1	Imagens Monocromáticas	17
2.1.2	Representação de Cenas Naturais e Sintéticas	18
2.1.3	Conceito de Resolução	19
2.1.4	Espaço de Cores	19
2.1.4.1	O sistema de cores RGB	20
2.1.4.2	O sistema de cores YCrCb	21
2.2	Mudança de resolução de imagens	21
2.2.1	Decimação	22
2.2.2	Interpolação	22
2.2.3	Transformadas Discretas de Cosseno	23
2.3	Compressão e Codificação de Imagens e Vídeos Digitais	26
2.3.1	Compressão Sem Perdas	27
2.3.1.1	Entropia	27
2.3.2	Compressão com perdas	28
2.3.3	Padrão JPEG	28
2.3.4	Sistema H.264/AVC	30
2.4	Métricas de Qualidade	31
3	ALGORITMO PROPOSTO	33
3.1	Super-resolução por exemplos	34
3.2	Solução proposta	36
4	MATERIAIS E MÉTODOS	39
4.1	Transmissor	39
4.1.1	Hardware	39
4.1.2	Software	42
4.2	Receptor	42
4.2.1	Hardware	42

4.2.2	<i>Software</i>	42
5	RESULTADOS EXPERIMENTAIS	43
5.1	Resultados Objetivos	45
5.1.1	Teste com codificação JPEG	45
5.1.2	Teste com codificação H.264/AVC	47
5.1.3	Resultados Subjetivos	48
5.1.4	Teste com codificação JPEG	49
5.1.5	Teste com codificação H.264/AVC	51
6	CONSIDERAÇÕES FINAIS	55
	Referências	57

1 Introdução

O presente capítulo evidencia a contextualização/justificativa e objetivos deste trabalho.

1.1 Contextualização e Justificativa

A mineração consiste na extração, elaboração e beneficiamento de minerais que se encontram em estado natural sólido, líquido ou gasoso. Esta atividade é uma atividade comum no cotidiano das pessoas, desde artigos em vidro (areia) e cerâmica (argila) até a fabricação de remédios e aparelhos eletrônicos [10].

Ainda de acordo com [10], a indústria de mineração tem um papel vital na economia global. Em 2014, a capitalização de mercado estimada de empresas mineradoras globais é de cerca de 962 bilhões dólares americanos. Uma grande parte destas operações é subterrânea e envolvem equipamentos e processos especializados. Os sistemas de comunicação possuem o papel de garantir a segurança dos operários e otimizar o processo de mineração. O tamanho estimado do mercado de equipamentos de mineração subterrânea por si só foi calculado em cerca de 45 milhões dólares em 2014. Parte desse montante é atribuído aos sistemas de comunicação.

De acordo com [11], os sistemas de comunicação atuais usados em minas subterrâneas empregam sistemas com fio. No entanto, estes sistemas de comunicação a cabo podem parar de operar devido a diversas complicações (incêndios, desabamentos, explosões, inundações, entre outras desastres). Os sistemas de comunicação sem fio tem melhor probabilidade de sobreviver a desastres. Isso se dá pelo fato de terem pouca ou nenhuma dependência de um condutor sólido que deve permanecer intacto. Entretanto, os desafios técnicos tornam difícil estabelecer um sistema de comunicação prático sem fio em minas subterrâneas.

A ideia de usar a terra como um meio comum para a comunicação volta a Nicola Tesla, já em 1899, através do uso de ondas com frequências extremamente baixas [12]. Segundo [11], a maioria dos sistemas de comunicação sem fio utilizam topologias de comunicação em radiofrequência que exigem um caminho claro ou ao ar livre para a propagação do sinal, o que limita a comunicação para as entradas de minas adjacentes ou em torno dos pilares dos túneis. Durante emergências, o desabamento do telhado também pode bloquear ou limitar severamente a propagação de sinal de rádio convencional. No caso em que rochas (entre outros materiais de condutividade elétrica não desprezível) se tornam o meio de propagação, a alta atenuação de ondas eletromagnéticas em altas frequências

não permitem o uso desses sistemas supracitados [13].

Utilizar sinais similares aos utilizados para transmissões em ar livre é inviável, pois esses sinais são incapazes de penetrar nas rochas e demais materiais. No entanto, a atenuação de sinais eletromagnéticos em comunicações através da terra (*Through-the-earth*) diminui com a frequência, e em frequências muito baixas (abaixo de 30 kHz) é possível realizar uma comunicação entre o subterrâneo e a superfície diretamente [14].

1.2 Objetivos

1.2.1 Objetivos Gerais

O presente trabalho consiste na implementação de um codificação de imagens em baixo consumo, que viabiliza a comunicação mesmo com uma largura de banda de transmissão muito baixa, permitindo o envio de informações vitais por operários presos na mina para equipes de resgate.

1.2.2 Objetivos Específicos

- Estudar técnicas rápidas de downsampling para codificar a imagem em resolução mais baixa possível;
- Desenvolvimento do algoritmo para estas mesmas técnicas;
- Prototipar *hardware* e *software*;
- Testes de *hardware* e de *software*, visando medir a qualidade das sequências de vídeo no que chegam receptor;

2 Fundamentação Teórica

2.1 Imagens Digitais

Segundo [15], uma imagem pode ser representada como uma função de duas variáveis reais, por exemplo, $a(x,y)$ com a sendo a amplitude (ex.: brilho) da imagem nas coordenadas reais de posição (x,y) . As amplitudes da imagem serão sempre dadas por um conjunto de números reais ou inteiros, sendo este último o resultado de um processo de quantização, que converte uma faixa de números reais, por exemplo de 0,0 a 100,0%, em um número discreto de níveis.

De acordo com o mesmo autor, imagens digitais $a(m,n)$, descritas em um espaço discreto, são obtidas a partir de imagens analógicas $a(x,y)$, por sua vez descritas em um espaço contínuo, através do processo de amostragem. A equação 2.1 ilustra uma possível representação matricial para uma imagem digital.

$$a = \begin{bmatrix} a_{0,0} & a_{0,1} & \cdots & a_{0,n-1} \\ a_{1,0} & a_{1,1} & \cdots & a_{1,n-1} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m,0} & a_{m,1} & \cdots & a_{m-1,n-1} \end{bmatrix}. \quad (2.1)$$

2.1.1 Imagens Monocromáticas

De acordo com [16], o processo de digitalização das imagens requer uma amostragem espacial, este processo irá definir as quantidades M de linhas e N de colunas que caracterizam o espaço discreto em que a imagem está contida, além do número L , que no caso das imagens monocromáticas é o número de tons de cinza que cada *pixel* (elemento da imagem) pode assumir. M e N devem ser números inteiros e positivos. Entretanto, devido ao processamento, armazenamento e considerações do hardware de amostragem, o número de níveis de cinza é tipicamente uma potência inteira de 2:

$$L = 2^K, \quad (2.2)$$

onde K é um valor inteiro positivo. Assumindo que os níveis de cinza são representados por números inteiros igualmente espaçados no intervalo $[0,L-1]$, o número de *bits* B necessários para armazenar uma imagem digitalizada é:

$$b = M \times N \times K. \quad (2.3)$$

2.1.2 Representação de Cenas Naturais e Sintéticas

Uma cena de vídeo natural é geralmente composta de vários objetos, cada um com suas características próprias (forma, profundidade, textura e iluminação). A cor e o brilho de uma cena de vídeo natural mudam com diferentes graus de intensidade em toda a cena, ou seja, tem tons contínuos. Uma cena sintética, por sua vez, procura emular uma cena real ou apresentar uma cena virtual, e é geralmente desenvolvida com métodos de computação gráfica [17].

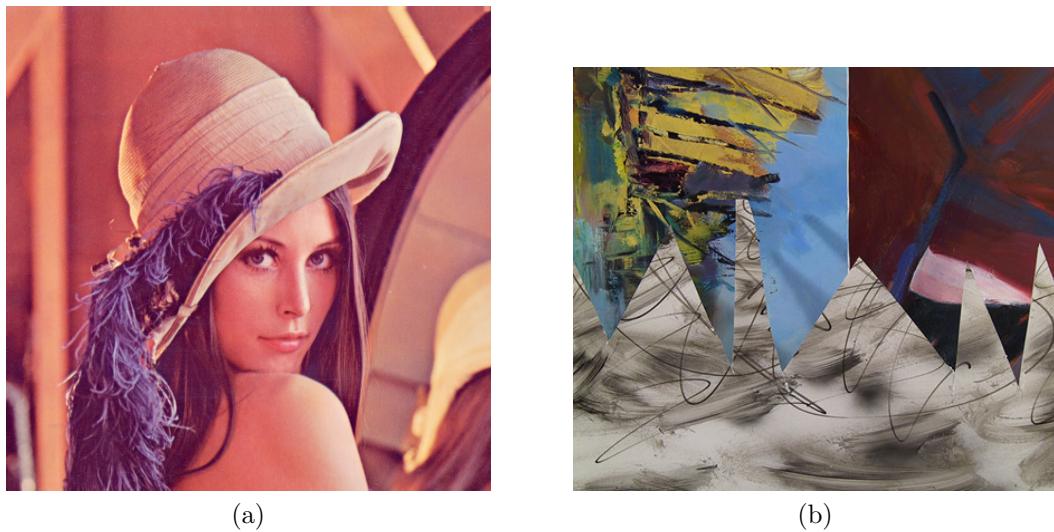


Figura 1 – Cenas: (a) natural; (b) sintética. Fonte: [1].

Segundo [3], para se representar cenas naturais e sintéticas de forma digital é necessário amostrá-las no espaço e no tempo, como mostrado na Figura 2. Para isso, são obtidas fotografias digitais das cenas (quadros) entre intervalos regulares.

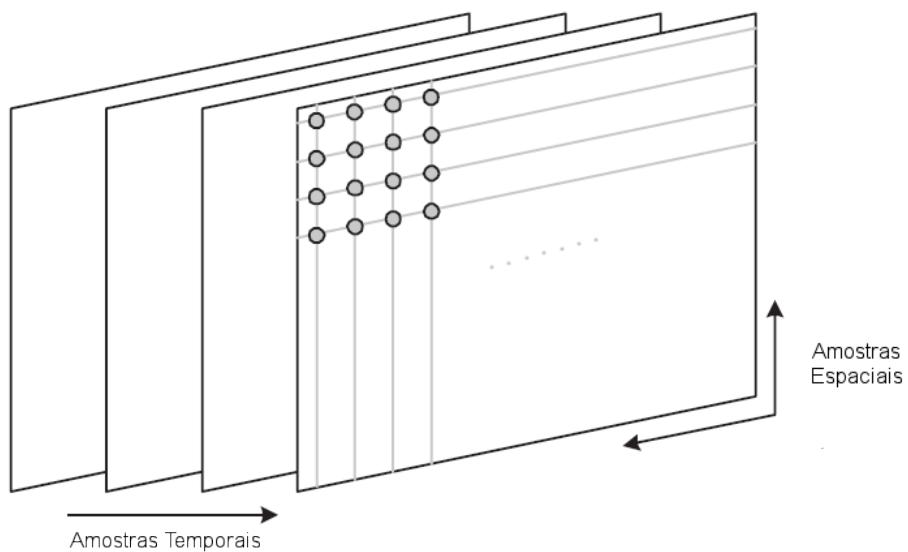


Figura 2 – Amostragem espacial e temporal de uma sequência de vídeo.

2.1.3 Conceito de Resolução

De acordo com [2], ao ser digitalizada a imagem assume um tamanho adimensional, em *pixels*. Mas, pode-se conhecer uma medida de qualidade da amostragem, conhecendo-se a razão entre o número de *pixels* obtido e o tamanho real da imagem. A esta medida dar-se o nome de resolução espacial. Em geral, é medida em pontos por polegada ou DPI (do inglês, *dots per inch*), mas pode ser também em pontos por centímetro, ou ainda em qualquer outra unidade equivalente. Quando duas imagens com tamanhos reais iguais são capturadas com resoluções espaciais diferentes, naturalmente terão o número de *pixels* diferentes e na tela aparecerão com tamanhos diferentes, como ilustrado na Figura 3.

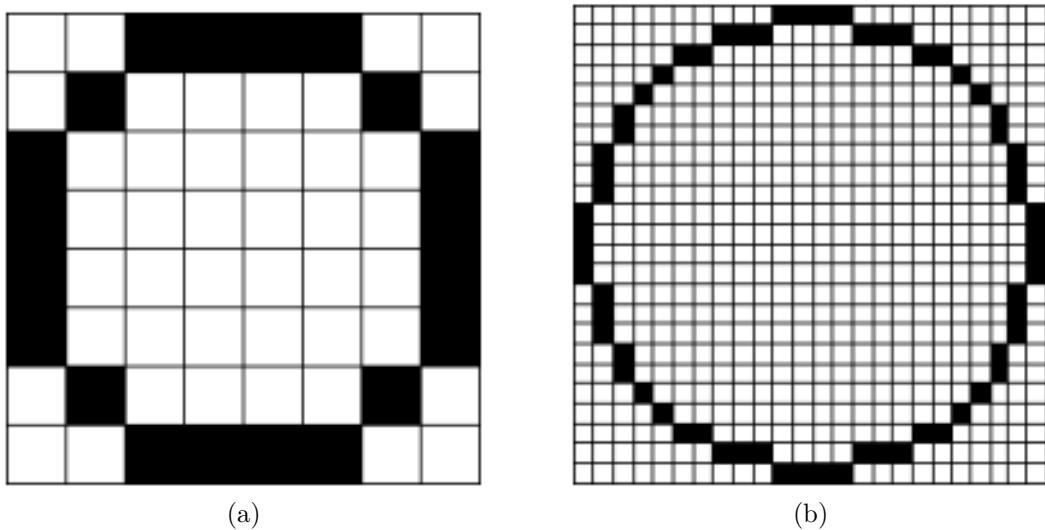


Figura 3 – Imagens com mesmo tamanho real e diferentes resoluções. Fonte: [2].

Já a resolução de tons de cinza (*Gray-level resolution*) é definida como a menor mudança discernível entre os tons de cinza. No entanto medir essas mudanças é um processo com um nível de subjetividade maior. Devido a considerações sobre o *hardware*, tem-se que o número de níveis de cinza é uma potência de 2, como já foi dito. Na maioria dos casos utilizam-se 8 bits ou 16 bits (em aplicações em que é necessária uma gama maior de níveis). Em certos casos pode-se deparar com imagens digitalizadas com 10 ou 12 bits de precisão, mas estes casos são exceções a regra geral.

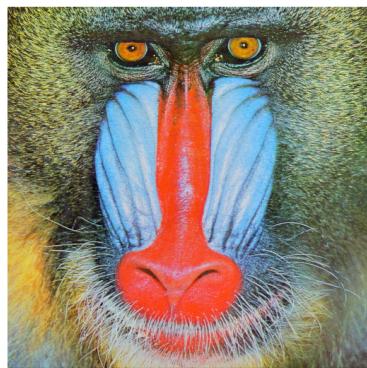
2.1.4 Espaço de Cores

De acordo com [6], na maioria das aplicações com vídeos e imagens digitais é necessário trabalhar e exibir imagens coloridas, o que faz necessário um mecanismo para capturar e representar a informação das cores. Uma imagem monocromática requer apenas um número que indique o brilho ou a luminância de cada amostra espacial. Imagens coloridas, por outro lado, requerem no mínimo três números por pixel para representar

com precisão a cor. O método escolhido para representar o brilho, luminância e cores é chamado espaço de cores.

2.1.4.1 O sistema de cores RGB

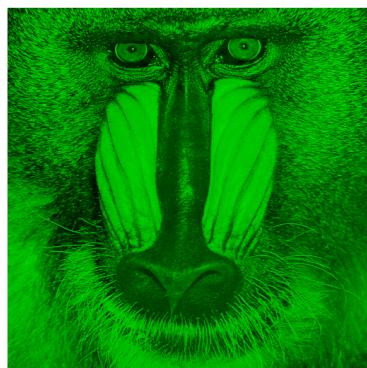
No espaço de cores RGB, a cor das amostras da imagem é representadas por três números que significam proporções relativas de vermelho (**Red**), verde (**Green**) e azul (**Blue**), essas são as três cores primárias aditivas da luz [6]. Combinando vermelho, verde e azul em variadas proporções pode-se criar qualquer cor. A Figura 4 é um exemplo de representação dos componentes vermelho, verde e azul de uma imagem colorida. A componente vermelha (Figura 4b) consistem em todas as amostras vermelhas da imagem, a componente verde (Figura 4c) consistem em todas as componentes verdes das imagens e por fim a componente azul contendo todos os amostras azuis da imagem (Figura 4d).



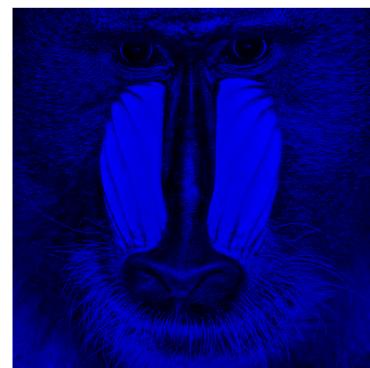
(a)



(b)



(c)



(d)

Figura 4 – Componentes no espaço RGB: (a) imagem original; (b) vermelho; (c) verde; (d) azul. Fonte: [1].

Para o mesmo autor, o espaço de cor RGB é bem adequado para capturar e exibir imagens coloridas. Obter uma imagem RGB envolve filtrar os componentes vermelho, verde e azul da cena e capturar cada um com um conjunto de sensores separado. Para

exibir as cores de uma imagem RGB ilumina-se separadamente as componentes vermelha, verde e azul de cada *pixel* de acordo com a intensidade de cada componente. A partir de uma certa distância de visualização, os componentes separados se fundem para dar a aparência da cor “verdadeira”.

2.1.4.2 O sistema de cores YCrCb

O sistema visual humano é menos sensível a cor do que à luminância. No espaço RGB as três cores são igualmente importantes e, assim, são normalmente armazenadas todas com a mesma resolução, mas é possível representar uma imagem colorida mais eficientemente através da separação da luminância a partir da informação da cor [6].

O YCrCb é um espaço de cor definido pela CCIR (*International Consultative Committee on Broadcasting*), também referido como o espaço de cor CCIR 601 [18]. YCrCb define a informação das imagens coloridas em termos da luminância (componente Y) e de dois valores de crominância diferentes (Cr-crominância de cores vermelhas e Cb-crominância de cores azul) e não como uma combinação de cores como no espaço RGB. Quando cada *pixel* é representado assim, ao contrário do espaço de cor RGB em que cada *pixel* tem, tipicamente, 24 *bits* de informação (8 *bits* para cada plano de cor), cada *pixel* pode ser representado por apenas 12 *bits* devido a redundância da informação de crominância. Para isso, primeiro deve-se converter do espaço RGB (24 *bits/pixel*) para YCrCb (24 *bits/pixel*), da seguinte maneira:

$$\begin{bmatrix} Y \\ Cr \\ Cb \end{bmatrix} = \begin{bmatrix} 0,299 & 0,587 & 0,114 \\ -0,169 & -0.331 & 0,5 \\ 0,5 & -0.419 & -0.091 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}. \quad (2.4)$$

Após o processo de conversão, pode-se dizimar os *pixels* de Cr e Cb de modo que só um a cada quatro *pixels* permaneça representando os quatro. Ainda é possível subamostrar Cr e Cb com fator de 2, tanto na horizontal como na vertical.

Como o sistema visual humano é mais sensível à luminância do que às cores, pode-se representar as crominâncias Cr e Cb com menor resolução sem prejuízo visual aparente, reduzindo assim a quantidade de dados necessária para representar uma imagem ou *frame*, diminuindo consequentemente o tamanho do vídeo [19].

2.2 Mudança de resolução de imagens

A mudança arbitrária de resolução de imagens, surge com bastante frequência em diversas aplicações. Por exemplo, imagens e vídeos muitas vezes precisam ser redimensionados espacialmente, a fim de garantir que os dados possam trafegar nas redes de

comunicações sobre as quais eles viajam e serem exibidos nos dispositivos de exibição do usuário final sobre o qual eles vão ser apresentados [4].

O aumento ou redução da resolução de imagens são obtidas através de processos de decimação ou interpolação [20].

2.2.1 Decimação

O processo de decimação de uma imagem por um fator M consiste em manter uma linha a cada M linhas e o mesmo para as colunas. Definindo u e v como sendo as coordenadas de largura e comprimento de uma dada imagem, cada *pixel* da imagem original é indicada por $I(u, v)$, e cada *pixel* da imagem decimada por M é indicado por $I^D(u, v)$ (garcia2013tecnicas). Assim, temos:

$$I^D(u, v) = I(uM, uV) \quad (2.5)$$

Aplicando a transformada de Fourier sobre a equação 2.5 prova-se que $I^D(u, v)$ sofrerá superposição espectral (ou *aliasing*, em inglês), se a largura da faixa da transformada discreta de Fourier de $I(u, v)$ estiver fora do intervalo $\left[-\frac{\pi}{M}, \frac{\pi}{M}\right]$. Para evitar que isso aconteça, o processo de decimação é geralmente precedido por uma filtragem que preserve o intervalo, como mostrado no diagrama de blocos da Figura 5b [3] .

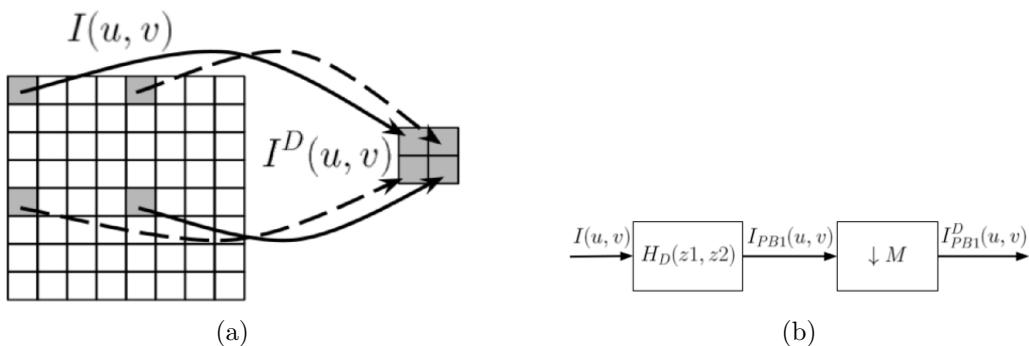


Figura 5 – (a) decimação $I^D(u, v)$ de uma imagem $I(u, v)$ de tamanho 8x8 por um fator 4; (b) operação geral de decimação: a imagem $I(u, v)$ é convoluída com o filtro $H_D(z_1, z_2)$, gerando uma versão passabaixas I_{PB1}^D , e depois decimada por um fator M , gerando uma versão $I(u, v)$ de menor resolução. Fonte: [3].

2.2.2 Interpolação

O processo de interpolação de uma imagem por um fator M consiste em acrescentar $M - 1$ zeros entre cada linha e $M - 1$ também zeros entre cada coluna da imagem, como

mostrado na Figura 6a. Definindo a imagem interpolada por M como sendo $I^I(u, v)$, então temos:

$$I^I(u, v) = \begin{cases} I(u/M, v/M), & u = jM \quad e \quad v = kM, \quad j, k \in \mathbb{Z} \\ 0, & \text{caso contrário} \end{cases} \quad (2.6)$$

Aplicando a transformada de Fourier sobre a equação 2.6, prova-se que $I^I(u, v)$ deverá ser posteriormente filtrada por um filtro passabaixas para evitar problemas de superposição espectral. O filtro, denominado $H_I(u, v)$, deverá ter ganho igual a M no intervalo de frequências $[-\frac{\pi}{M}, \frac{\pi}{M}]$, e ganho nulo em todas as outras frequências. A Figura 6b ilustra a operação geral de interpolação por um fator M , incluindo o filtro passabaixas H_I . Fonte: [3].

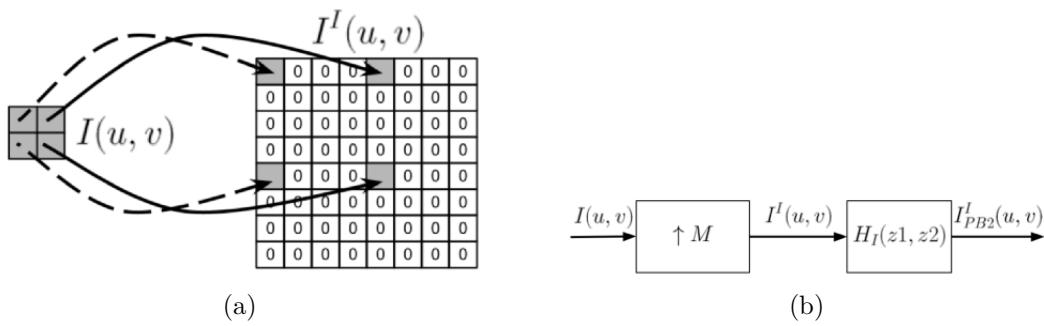


Figura 6 – (a) interpolação $I^I(u, v)$ de uma imagem $I(u, v)$ de tamanho 2x2 por um fator 4; (b) operação geral de interpolação: a imagem $I(u, v)$ é interpolada por um fator M e depois convoluída com o H_I , gerando uma versão em maior resolução de $I(u, v)$. Fonte: [3].

2.2.3 Transformadas Discretas de Cosseno

Desde que muito material multimídia passou a ser comprimido usando as populares transformadas discretas de cosseno (DCT, do inglês (Discrete cosine transform)), como nos padrões JPEG, MPEG e H.26X, uma operação de redimensionamento espacial que seja eficiente, flexível computacionalmente e que possa ser realizado no domínio da DCT é desejável. Flexibilidade computacional é especialmente importante porque uma ampla gama de opções de projeto muitas vezes se faz necessária, a fim de atender requisitos como consumo energético [4].

Segundo [21], a transformada discreta de cosseno, é uma transformada matemática que tem como funções de base os cossenos, por meio dela converte-se um bloco de amostras em um bloco correspondente de coeficientes da transformada. Essa operação é dividida em duas operações, a transformada direta (ou FDCT, do inglês *Forward discrete cosine transform*), mostrada abaixo:

$$Y = AXA^T \quad (\text{FDCT}), \quad (2.7)$$

onde a matriz X é a imagem no domínio espacial, a Y são os respectivos coeficientes das transformada e a matriz A é a matriz de transformação.

E a transformada inversa (ou IDCT, do inglês *Inverse discrete cosine transform*), descrita pela seguinte operação:

$$X = A^T Y A \quad (\text{IDCT}) \quad (2.8)$$

onde a matriz X é a imagem no domínio espacial, a Y são os respectivos coeficientes das transformada e a matriz A é a matriz de transformação. Os elementos de A podem ser calculados como mostrado abaixo: [6]

$$A_{ij} = C_i \cdot \cos\left(\frac{(2j+1)i\pi}{2N}\right)$$

$$C_i = \begin{cases} \sqrt{\frac{1}{N}}, & i = 0 \\ \sqrt{\frac{2}{N}}, & i > 0 \end{cases} \quad (2.9)$$

Assim, escrevendo as equações 2.7 e 2.8, no formato de somatórios para um espaço 2D, temos:

$$Y_{xy} = C_x \cdot C_y \cdot \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} X_{ij} \cdot \cos\left(\frac{(2j+1)y\pi}{2N}\right) \cdot \cos\left(\frac{(2i+1)x\pi}{2N}\right) \quad (2.10)$$

$$X_{ij} = \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} C_x \cdot C_y \cdot Y_{xy} \cdot \cos\left(\frac{(2j+1)y\pi}{2N}\right) \cdot \cos\left(\frac{(2i+1)x\pi}{2N}\right) \quad (2.11)$$

A DCT possui a interessante propriedade de concentrar em poucos elementos a maior parte da energia, para imagens de cenas naturais [22]. Por exemplo, considere o seguinte bloco de 10×10 pixels (Figura 7):

$$\begin{bmatrix} 181 & 192 & 188 & 185 & 173 & 170 & 177 & 182 & 180 \\ 173 & 180 & 183 & 176 & 169 & 142 & 151 & 148 & 141 \\ 156 & 138 & 141 & 157 & 152 & 130 & 124 & 125 & 123 \\ 141 & 118 & 116 & 123 & 128 & 115 & 110 & 114 & 114 \\ 126 & 115 & 108 & 111 & 110 & 110 & 108 & 109 & 108 \\ 118 & 110 & 110 & 110 & 106 & 112 & 105 & 105 & 108 \\ 103 & 109 & 109 & 108 & 104 & 111 & 103 & 104 & 103 \\ 100 & 106 & 106 & 106 & 106 & 102 & 109 & 104 & 104 \\ 101 & 102 & 107 & 114 & 102 & 99 & 106 & 106 & 105 \end{bmatrix} \quad (2.12)$$

Aplicando-se a transformada sobre essa matriz, temos:



Figura 7 – Bloco 10x10 *pixels* da imgem de teste Lena, equivalente a matriz 2.12. Fonte: [1].

$$\begin{bmatrix} 1138.22 & 40.32 & -1.46 & -8.30 & 6.03 & 12.77 & -0.55 & -2.25 & 0.29 \\ 211.52 & 32.89 & 1.58 & -6.38 & 4.62 & 7.48 & -5.93 & -0.32 & 4.32 \\ 102.24 & -2.92 & -3.39 & -17.64 & -19.38 & 0.06 & -4.05 & -6.80 & 1.86 \\ 40.69 & -10.24 & 7.43 & -0.50 & -15.50 & -14.18 & -0.10 & 0.44 & -5.08 \\ 9.69 & -14.02 & 14.88 & 3.59 & -3.60 & -0.80 & 2.05 & -0.66 & -3.09 \\ -2.47 & -14.72 & 5.38 & 5.88 & 2.28 & -2.98 & -1.90 & 0.21 & -0.72 \\ 0.86 & -8.80 & -0.22 & 0.48 & 7.29 & 1.22 & 4.61 & 3.55 & -6.01 \\ 1.70 & -6.77 & -0.60 & 2.25 & 4.92 & 1.95 & 2.62 & -1.41 & -1.25 \\ -0.79 & -2.94 & -2.37 & -0.16 & -0.10 & 0.41 & 0.40 & -0.68 & -4.85 \end{bmatrix} \quad (2.13)$$

Assim, pode-se notar que os coeficientes de maior módulo tendem a se concentrar no canto superior esquerdo.

Hoje, na literatura, há várias propostas de algoritmos de redimensionamento no domínio transformado da DCT [23, 24, 25]. Especialmente, [26] que sugerem uma técnica computacional simples e rápida para dobrar ou reduzir pela metade o tamanho da imagem usando as componentes de baixa frequência. [27], propoem algumas alterações a esta técnica. Embora o esquema modificado proporciona uma melhor qualidade de imagens redimensionadas, é computacionalmente mais intenso do que o anterior. [23] desenvolveu-se um algoritmo rápido para o fator arbitrário redução do tamanho do vídeo pré-codificado. Assim como expressa o redimensionamento espacial com uma multiplicação de matrizes entre a imagem e uma matriz construída a partir dos coeficientes do bloco 8x8 da DCT.

De acordo com [28], a DCT utilizada na compressão de imagens é apenas um membro da família de 16 transformadas discretas de senos e cossenos, comumente chamadas de transformadas discretas trigonométricas (DTTs). A forma convolucional dessas transformadas é chamada de convolução simétrica. [29] propõe um solução através da convolução simétrica (não mais no domínio transformado) em que pode realizar o redimensionamento com um fator arbitrário produzindo uma imagem final com qualidade melhor que o método proposto por [26], no entanto, exige um maior esforço computacional.

[4] propõe um método de redimensionamento arbitrário (descrito pela Figura 8) com base na generalização da técnica proposta por [26]. Nessa abordagem proposta (Figura 8), um mapeamento simples é construído, envolvendo uma manipulação na transformada inversa (ao voltar para o domínio espacial), seguido de um aumento combinado com uma transformada direta no domínio 8x8 DCT. Devido ao fato do método proposto redimensionar a transformada inversa e a transformada direta, é possível obter qualquer fator de escala (ao invés de somente potências de 2). Escolhendo N pontos diferentes da transformada inversa e M pontos diferentes da transformada direta, mais ou menos da imagem original pode ser usada para variar a qualidade da imagem final para um fator de escala dado por $S = O/I = M/N$. Um algoritmo semelhante a este será utilizado para o redimensionamento de imagens durante este trabalho.

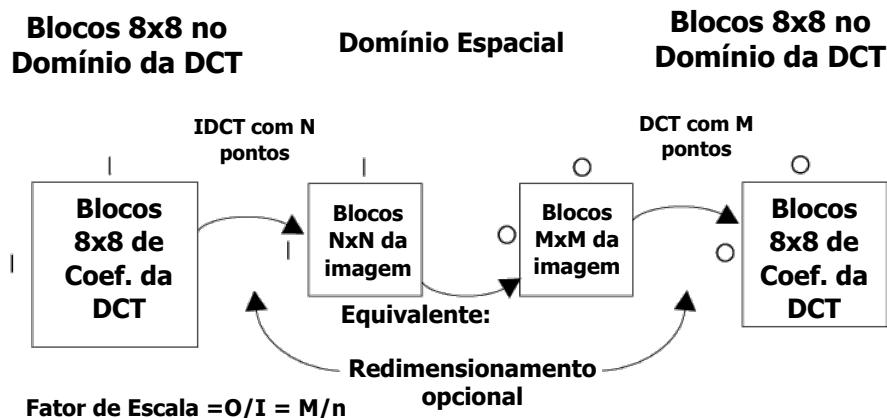


Figura 8 – Escala arbitrária redimensionando com uma ou ambas as transformadas locais.

Nota-se que a dimensão é dada em blocos (por exemplo, o tamanho da primeira imagem é de I blocos por I blocos, onde cada cada bloco é de tamanho 8x8, o tamanho da segunda imagem ainda é I blocos por I blocos, mas o tamanho de cada bloco é $N \times N$). Fonte: [4].

2.3 Compressão e Codificação de Imagens e Vídeos Digitais

O termo “compressão de dados” refere-se ao processo de redução do montante de dados exigidos para representar uma dada quantidade de informação. Deve-se esclarecer que denominamos “dados” aos meios pelos quais uma informação é transmitida [6].

2.3.1 Compressão Sem Perdas

Técnicas de compressão sem perdas, como o seu nome indica, não envolvem perda de informações. Se os dados sofreram compressão do tipo sem perdas, pode-se recuperar exatamente os dados originais a partir os dados comprimidos [30].

Compressão de texto é uma área importante para a compressão sem perdas. É muito importante que a reconstrução seja idêntica a do texto original, pois diferenças muito pequenas podem resultar em orações com significados muito diferentes. Considere as frases “Não, enviem dinheiro” e “Não enviem dinheiro”. Um argumento semelhante vale para arquivos de computador e para certos tipos de dados, tais como registros bancários [31].

Se os dados comprimidos vão ser processados (para se obter mais informação) ou “melhorados” é importante que a integridade seja preservada. Por exemplo, suponhamos que uma imagem radiológica seja comprimida de uma forma com perdas, e a diferença entre a imagem reconstruída e o original seja visualmente indetectável. Se esta imagem foi mais tarde reforçada, as diferenças anteriormente indetectáveis podem causar o aparecimento de artefatos que poderiam enganar seriamente o radiologista [31].

2.3.1.1 Entropia

Segundo [32], considerando uma fonte discreta de informação, para cada estado possível i haverá um conjunto de probabilidades (P_i) de produzir os j possíveis símbolos . Assim, há uma entropia ($H(j)$) para cada estado. A entropia média da fonte H será definida como a média ponderada de $H_i(j)$ usando como ponderações as probabilidades P_i , como mostrado a seguir:

$$\begin{aligned} H &= \sum_i P_i H_i \\ &= - \sum_{i,j} P_i(j) \log(P_i(j)) \end{aligned} \tag{2.14}$$

De acordo com mesmo autor, se a base do algoritmo na equação 2.14 for 2, a entropia representará uma medida de *bits* por símbolos, sendo essa a medida do número médio de *bits* necessários para codificar a saída da fonte.

A quantidade de informação nova transmitida por um símbolo diminui na medida em que a probabilidade de ocorrência deste símbolo aumenta. Então, os codificadores que exploram a redundância entrópica tem por objetivo transmitir o máximo de informação possível por símbolo codificado e, deste modo, representar mais informações com um número menor de *bits*. A codificação de entropia, como é chamada, utiliza diferentes técnicas e algoritmos de compressão sem perdas para atingir este objetivo [33].

2.3.2 Compressão com perdas

Técnicas de compressão com perdas provocam alguma perda de informações, assim, dados que tenham sido comprimidos utilizando técnicas com perdas geralmente não podem ser recuperados ou reconstruídos exatamente. Em troca de aceitar essa distorção na reconstrução, podemos geralmente obter taxas de compressão muito mais elevadas do que com compressão sem perdas [31].

De acordo com o mesmo autor, em muitas aplicações nota-se que a diferença entre a informação reconstruída e a original pode ser aceitável. Por exemplo, na compressão de vídeo, diferenças entre os sinais original e reconstruído podem ser toleradas, contanto que elas não resultem em artefatos inconvenientes. Assim, sequências de vídeo são geralmente comprimidas usando compressão com perdas.

2.3.3 Padrão JPEG

O padrão JPEG é um método de compressão com perdas, comumente empregado para imagens de tons contínuos, devido à sua alta eficácia e baixa complexidade computacional [34].

Este padrão é definido pelo [21], sendo largamente utilizado em máquinas fotográficas, scanners e páginas da internet.

De acordo [5], o algoritmo do padrão JPEG é descrito pelo fluxograma nas Figuras 9 e 10.

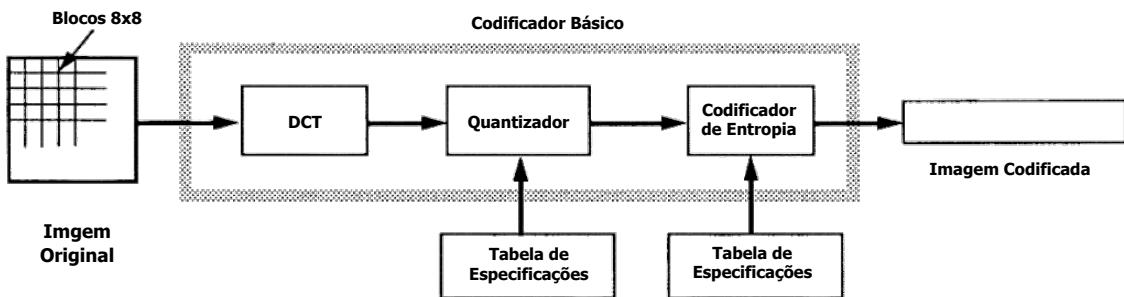


Figura 9 – Etapas de codificação baseada em DCT. Fonte: [5].

A primeira etapa pela qual a imagem passa na codificação é a FDCT. Já na última etapa, na saída do decodificador, a IDCT, que de acordo com o mesmo autor, nada mais é que uma outra transformada matemática que faz o inverso da FDCT, converte coeficientes em amostras da imagem.

De acordo [5], após a saída do FDCT, cada um dos 64 coeficientes (blocos 8x8) da DCT é quantizado uniformemente em conjunto com uma matriz de normalização de 64 elementos, que deve ser indicada pela aplicação (ou usuário) como uma entrada para o codificador. Cada elemento pode variar entre 1 e 255, o que indica o tamanho de degrau do

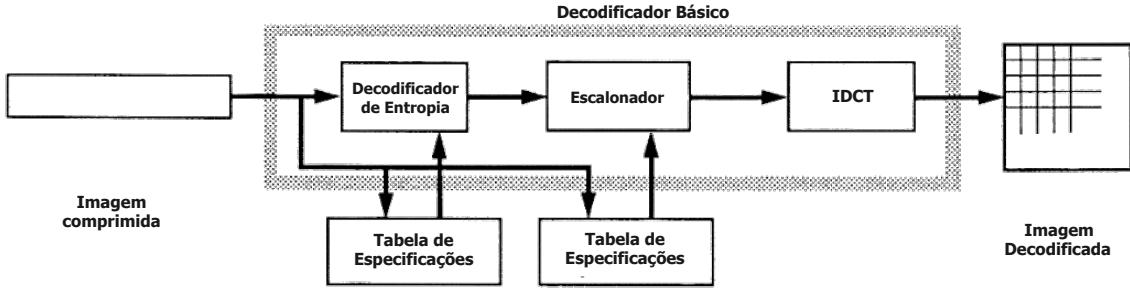


Figura 10 – Etapas de decodificação baseada em DCT. Fonte: [5].

quantizador do seu coeficiente DCT respetivo. A finalidade de quantização (*quantizer*) é conseguir uma compressão adicional, apenas por representar os coeficientes de DCT com uma precisão suficiente para atingir a desejada qualidade de imagem. Dito de outra forma, o objetivo desta etapa de processamento é descartar informações que não são visualmente significativas.

A quantização de uma matriz é definida como a divisão de cada um de seus elementos por um respetivo coeficiente de quantização (definido pela matriz de normalização), seguido de arredondamento para o inteiro mais próximo:

$$F^Q(u, v) = \left\lfloor \frac{F(u, v)}{Q(u, v)} + 0,5 \right\rfloor \quad (2.15)$$

onde: $F(u, v)$ é a matriz de coeficientes da DCT, $F^Q(u, v)$ é a matriz de coeficientes quantizados da DCT e $Q(u, v)$ é a matriz de normalização, e o operador $\lfloor . \rfloor$ executa o arredondamento para o menor valor inteiro mais próximo. Sendo assim, a operação $\lfloor +0,5 \rfloor$ executa o arredondamento para o valor inteiro mais próximo.

Ainda de acordo com [5], a inversão da quantização (*dequantization*), presente na decodificação do JPEG, representa a função inversa da quantização, que nesse caso significa uma simples multiplicação (termo a termo) pela matriz de normalização, como mostrado a seguir.

$$F^{Q'}(u, v) = F^Q(u, v) * Q(u, v) \quad (2.16)$$

onde: $F^{Q'}(u, v)$ é a matriz de coeficientes escalonados da DCT (aproximação de $F(u, v)$), $F^Q(u, v)$ é a matriz de coeficientes quantizados da DCT e $Q(u, v)$ é a matriz de normalização.

Depois da quantização, o coeficiente DC é tratado separadamente dos 63 coeficientes AC. O coeficiente DC é uma medida do valor médio das 64 amostras de imagem. Como há correlação geralmente forte entre os coeficientes DC dos blocos 8x8 adjacentes,

os coeficientes de DCT quantificados são codificados como a diferença de o termo DC do bloco anterior na ordem de codificação (zig-zag), como mostrado na Figura 11 . Este tratamento especial vale a pena, pois coeficientes DC frequentemente contêm uma fração significativa do total de energia imagem [5].

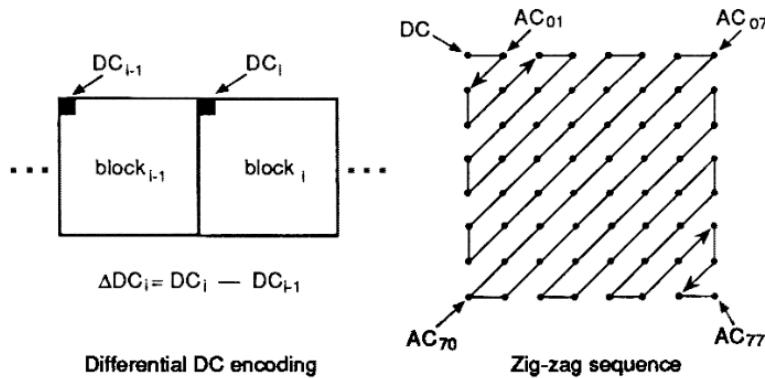


Figura 11 – Preparação dos coeficientes da DCT para o codificador de entropia. Fonte: [5].

A etapa final de codificação é a codificação de entropia. Esta etapa realiza a compressão sem perdas adicionais, codificando os coeficientes quantificados da DCT de forma mais compacta com base nas suas características estatísticas.

2.3.4 Sistema H.264/AVC

H.264/AVC é um padrão da indústria para codificação de vídeo, mas também é um formato popular para vídeo codificado, assim como um conjunto de ferramentas para a compressão de vídeo [6]. Esse padrão foi publicado pela ITU juntamente com a ISO e é conhecido por vários nomes, como: *H.264*, *MPEG-4 Part 10* e *Advanced Video Coding (AVC)*. Desenvolvido por uma equipe composta de centenas de especialistas de compressão de vídeo, o *Joint Video Team*, um esforço colaborativo entre a *Moving Picture Experts Group (MPEG)* e o *Video Coding Experts Group (VCEG)*.

A estrutura de um típico codec H.264 é mostrada nas Figuras 12 e 13 [6].

As imagens são processadas em unidades de macroblocos (MB), que correspondem a blocos de 16x16 pixels [6]. No codificador, é gerada uma predição para cada macrobloco a partir de blocos já codificados. Em seguida, a predição é subtraída do macrobloco gerando o macrobloco residual, que é transformado, quantizado e codificado por codificação de entropia, em processos similares aos descritos na Seção 2.3.3 .

Ainda, conforme o mesmo autor, sabe-se que a predição do macrobloco atual gerada no codificador acontece a partir de dados previamente codificados, seja usando a informação da próprio quadro (*frame*) atual (predição intra-quadro) ou utilizando informação de quadros anteriormente codificados (predição inter-quadros).

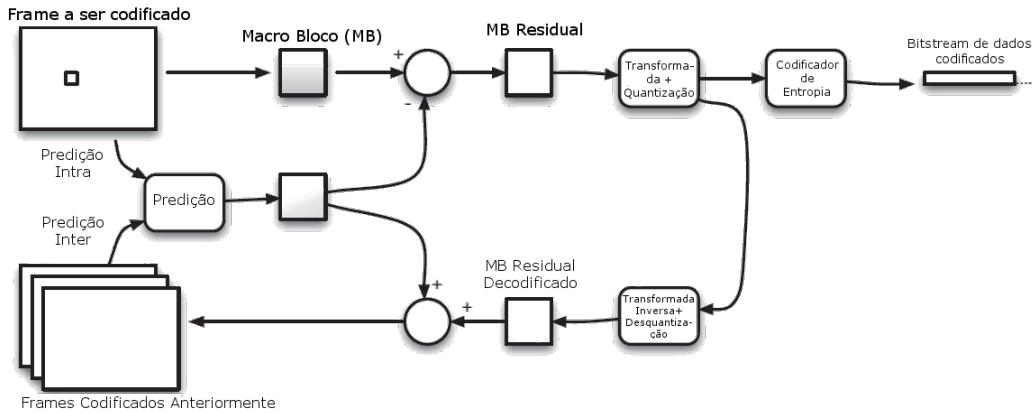


Figura 12 – Estrutura típica de um codificador H.264 [6].

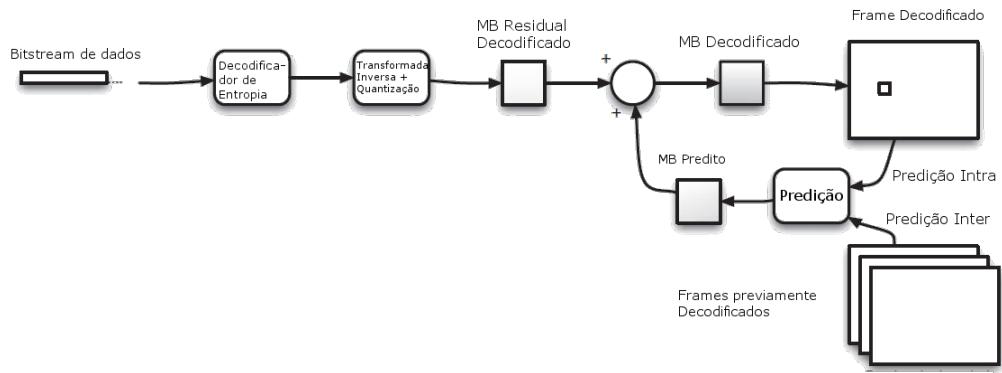


Figura 13 – Estrutura típica de um decodificador H.264 [6].

No decodificador, um macrobloco é decodificado (operação inversa à codificação de entropia), desquantizado e passa pela transformada inversa de modo a formar um macrobloco residual decodificado (similar ao macrobloco residual gerado no codificador). Ainda no decodificador é gerada a mesma predição que foi criada no codificador que é somada ao macrobloco residual decodificado para então produzir uma aproximação do macrobloco original da imagem [6].

2.4 Métricas de Qualidade

Uma vez que tenhamos desenvolvido um esquema de compressão de dados, precisamos ser capazes de medir o seu desempenho. Devido ao grande número de diferentes áreas de aplicação, condições diferentes têm sido desenvolvidas para descrever e medir o desempenho [31].

Um método comum utilizado como medida de desempenho para diversas técnicas de compressão de imagens é a relação sinal de pico/ruído (PSNR, do inglês *Peak Signal Noise Ratio*), que é a relação entre o máximo possível de potência de um sinal pela potência do ruído, quando comparamos um sinal antes e depois de um processo de degradação, ou

seja, a imagem original e a imagem comprimida. Um valor alto de PSNR significa uma alta relação entre a potência da imagem original pela potência da imagem comprimida, ou seja, uma melhor qualidade da imagem reconstruída. Em termos matemáticos, o valor do PSNR entre uma imagem original e uma imagem reconstruída é dado pela Eq. 2.17 [35] :

$$PSNR = 10 \cdot \log \left(\frac{255^2}{MSE} \right) \quad (2.17)$$

onde MSE é uma medida referente a diferença entre a sequência original de vídeo e a processada, dada por:

$$MSE = \frac{1}{M \cdot N} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} (Y_{out}(i, j) - Y_{in}(i, j))^2 \quad (2.18)$$

onde Y_{out} representa a imagem reconstruída, Y_{in} representa a imagem, N representa o número total de linhas de *pixels* da imagem, M o número de colunas e $Y(i, j, m)$ é o valor da luminância (0-255) na posição (i, j) da imagem.

Já para avaliar a degradação em uma sequência de vídeo toma-se a PSNR médio das imagens que compõe a sequência, dado por:

$$PSNR_{Médio}(k) = \frac{1}{K} \sum_{K=0}^{K-1} 10 \log_{10} \left(\frac{255^2}{MSE(K)} \right) \quad (2.19)$$

onde K é o número de quadros da sequência e o $MSE(K)$ é o MSE do K-ésimo quando da sequência de vídeo.

3 Algoritmo Proposto

O algoritmo proposto para a solução descrita neste sistema baseia-se em uma técnica clássica de super-resolução [3]. O termo super-resolução (SR) é usado para descrever processos que procuram acrescentar informações de alta frequência a uma imagem interpolada a partir de uma ou mais imagens disponíveis [36, 37, 38]. Este conjunto de imagens pode ser formado por imagens decimadas ou adquiridas por múltiplos sensores capturando uma mesma cena durante determinado período de tempo. Para cenas estáticas (Figuras 14 e 15), as observações são relacionadas por deslocamentos globais em nível de *subpixel*, geralmente ocorrendo devido a posições relativas das câmeras ou movimento do próprio sensor. Para cenas dinâmicas, as observações são relacionadas a deslocamentos de *subpixel* devidos ao movimento local dos próprios objetos juntamente com os deslocamentos globais, conforme pode ser visualizado na Figura 16. Em ambos os casos, a super-resolução é utilizada para gerar uma imagem com maior resolução espacial a partir de um conjunto de imagens em baixa resolução ou de frames de uma sequência de vídeo [39][40].

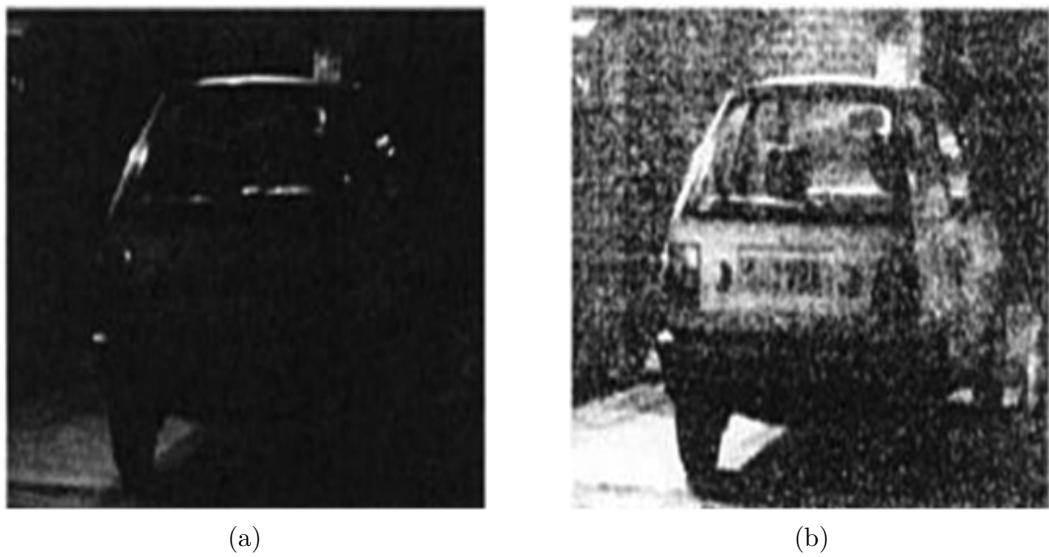


Figura 14 – (a) uma cena estática de uma sequência de vídeo é capturada com baixa luminosidade; (b) após equalização de histograma a placa do automóvel continua ilegível devido ao ruído natural da imagem. Fonte: [7].

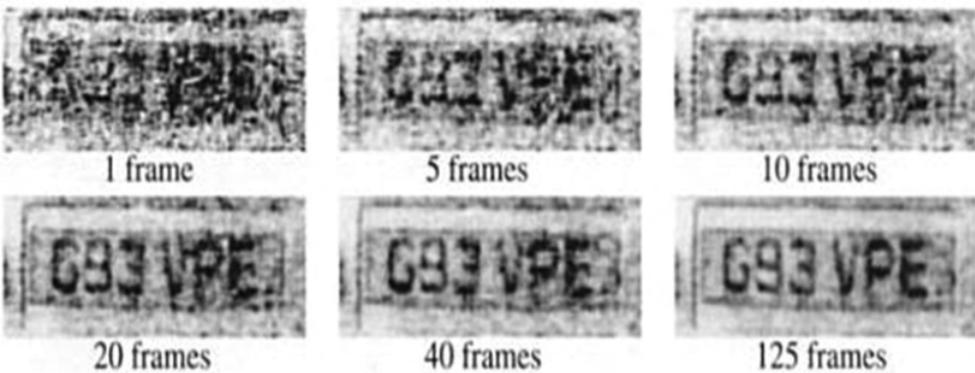


Figura 15 – Legibilidade da placa como resultado da média do conjunto cada vez maior de quadros. Fonte: [7].

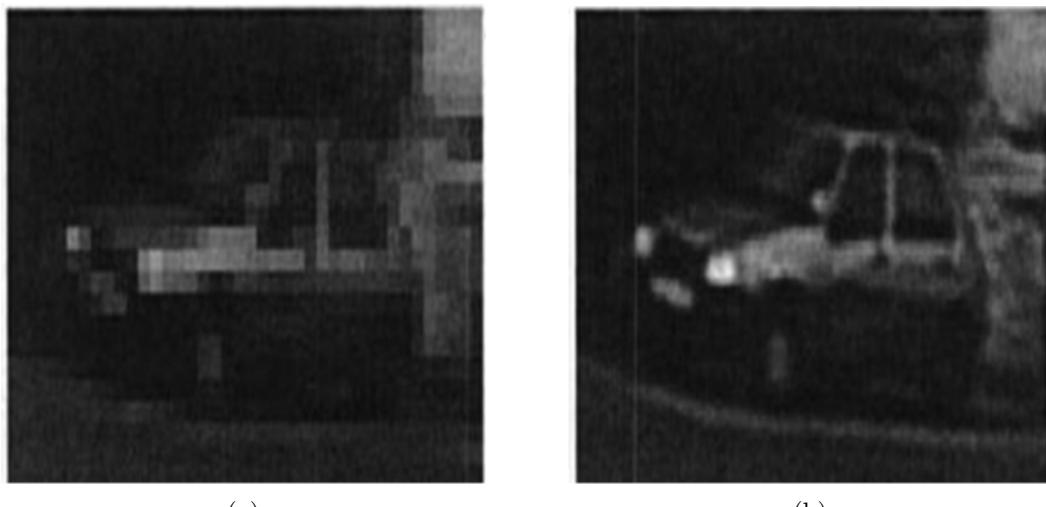


Figura 16 – (a) região de interesse é capturada em uma cena dinâmica; (b) a SR estima a cena subjacente a partir de 50 imagens. A reconstrução possui o triplo da resolução em relação à imagem original. Fonte: [7].

Pressupõe-se que a simples interpolação dessas imagens não seja suficiente para representar a cena de interesse com nitidez, fazendo-se necessário processar e combinar as diversas imagens disponíveis. É possível que todas as imagens possuam a mesma resolução, de forma que a super-resolução seja através da combinação dessas imagens (super-resolução por combinação de múltiplas imagens), ou que a imagem de interesse em baixa resolução seja processada a partir de um banco de imagens em alta resolução (super-resolução por exemplos) [3]. Neste trabalho trataremos apenas do segundo tipo de super-resolução.

3.1 Super-resolução por exemplos

No método de super-resolução baseado em exemplos, o objetivo consiste em estimar as informações, que não estão presentes na imagem original de baixa resolução, a partir

de um conjunto de amostras de imagens [41]. A princípio, não há relação direta entre a base de dados e a imagem de entrada, de forma que o algoritmo pode atender a uma ampla gama de imagens em baixa resolução, com um grande banco de dados [42].

Na super-resolução baseada em exemplos, gera-se para cada imagem em alta resolução I_j uma versão com componentes de baixa frequência, I_j^B , e uma versão de componentes de alta frequência, I_j^A , onde $I_j^A = I_j - I_j^B$, como ilustrado na Figura 17. Assim considera-se a imagem interpolada I_0 correspondente somente as componentes de baixa frequência ($I_0 = I_0^B$), o que permite obter uma relação entre a imagem de entrada (I_0) e as imagens de referência (I_j) [3].



Figura 17 – Exemplo de separação de quadro em componentes de baixa e alta frequência.

A Figura 18 ilustra a forma geral da super-resolução baseada em exemplos. Nela uma imagem (I_0^B) é dividida em vários macrobloco. A cada um desses blocos serão adicionadas informações de alta frequência. O dicionário com conteúdo de alta frequência é formado pelos blocos provenientes da componente de alta frequência (I_j^A) das imagens do banco de dados e a extração da informação de alta frequência ocorre pelo processo de estimativa de movimento.

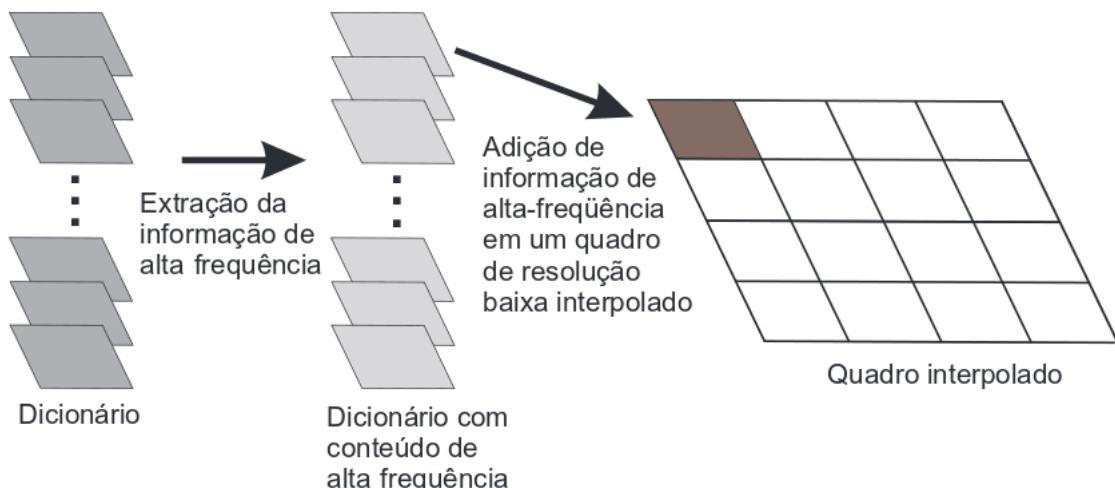


Figura 18 – Forma geral da super-resolução baseada em exemplos [8].

A estimativa de movimento é o processo realizado para encontrar os movimentos translacionais resultantes que ocorrem entre pelo menos dois quadros. Este processo consiste em buscar o melhor casamento (*matching*) de uma macrobloco de um quadro, geralmente chamado de atual, dentre um conjunto de blocos de referência [8], como ilustrado pela Figura 19.

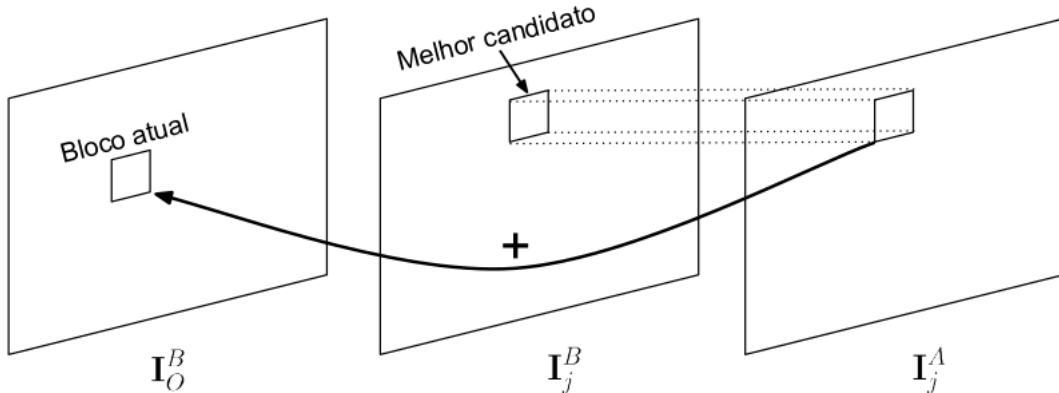


Figura 19 – Super-resolução baseada em exemplos: a imagem interpolada I_0^B recebe informações de alta frequência a partir de uma imagem em alta resolução I_j separada em versões com componentes de baixa e alta frequência, I_j^B e I_j^A .
Fonte: [3].

No caso da super-resolução por exemplos, este processo é aplicado entre os blocos da imgem interpolada e os blocos dos componentes de baixa frequência das imagens do banco de dados, o que indicará quais informações de dicionário de alta frequência devem ser adicionadas em cada macrobloco da imagem interpolada.

3.2 Solução proposta

Devido a pequena largura de banda disponível para o envio dos dados, optou-se por enviar uma sequência de vídeo com quadros em resolução mista, como ilustrado pela Figura 20.

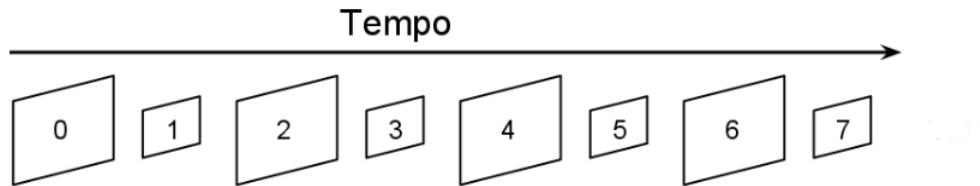


Figura 20 – Arquitetura de codificação em resolução mista.

A fim de recuperar as informações de alta frequência dos quadros em baixa resolução, apresenta-se uma técnica baseada na super-resolução por exemplos. Assim, utiliza-se

como banco de dados a imagem anterior ao quadro que será super-resolvido. O aproveitamento de informação de quadros vizinhos se deve ao fato que normalmente quadros vizinhos são altamente correlacionados [43].

O fluxograma mostrado na Figura 21, ilustra do algoritmo proposto. Dado um *frame* original decimado (I_O^D), que veio do transmissor, ele é interpolado a sua resolução original, gerando então I_O^B , que constitui uma versão de baixa frequência de I_O . Assim, de acordo com [3], pode-se assumir que :

$$\hat{I}_0 = I_O^B + \hat{I}_O^A. \quad (3.1)$$

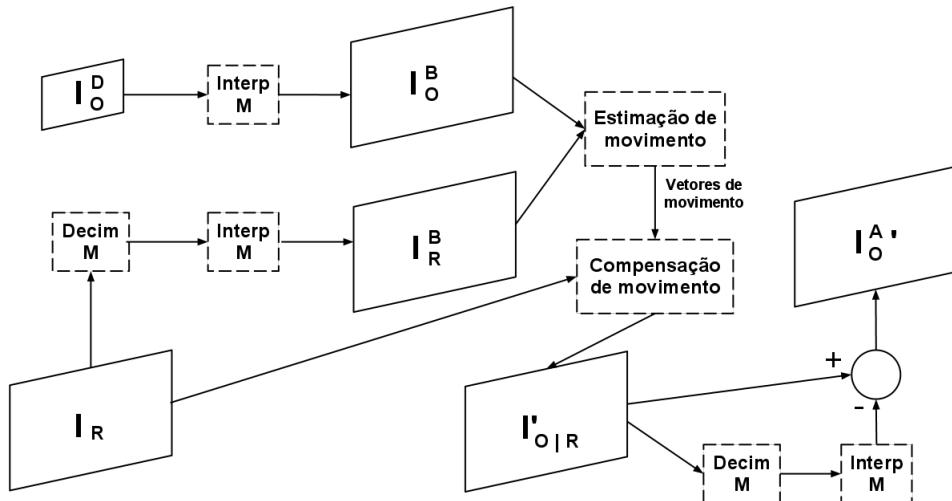


Figura 21 – Fluxograma do algoritmo de super-resolução proposta [3].

Defini-se I_R como o quadro anterior ao que será super-resolvido, que servirá como a imagem de alta resolução de referência. Para uma busca mais adequada das informações de alta frequência que serão adicionadas ao quadro decimado I_O^B é gerada sua versão de baixa frequência I_R^B , simulando o mesmo processo de degradação que gerou I_O^B através de I_O , por um processo de decimação seguido de uma interpolação utilizando o mesmo fator M que foi utilizado em I_O .

Em seguida, aplica-se o processo de busca pelos blocos semelhantes (foi utilizada a métrica MSE, discutida na Seção 2.4, para estipular a semelhança), como descrito na Seção 3.1, entre I_O^B e I_R^B . A busca de cada bloco do quadro I_O^B é realizada nos blocos do quadro I_R^B que estão dentro da janela de busca. Assim para cada bloco (u_i, v_i) , obtém-se vetores de movimento (m_u, m_v) que representam a posição do bloco mais parecido com o procurado.

Com os vetores de movimento, gera-se a estimativa de alta frequência de I_O , realizando a compensação de movimento, que consiste em reorganizar os blocos da imagem de

acordo com os vetores de movimento, na versão de altas frequências de I_R (I_R^A) obtida de acordo com a equação 3.1, ou seja, $I_R^A = I_R - I_R^B$. Assim para cada bloco (u_i, v_i) , têm-se:

$$I_R^{A'}(u_i, v_i) = I_R^A(u_i + m_u, v_i + m_v). \quad (3.2)$$

O quadro $I_R^{A'}$ já pode ser considerado uma estimativa de alta frequência de I_O , de acordo com [3]. Restando assim, somar as componentes de baixa e alta frequência para ter a estimativa de \hat{I}_O .

4 Materiais e Métodos

Na busca de uma solução para o sistema proposto, desmembrou-se o diagrama (Figura 22) a fim de modularizar o processo. A aquisição, envio e recepção dos dados, assim como as especificações de comunicação (número de portadoras, tipo de modulação, codificação de linha e canal) não foram tratados neste trabalho.

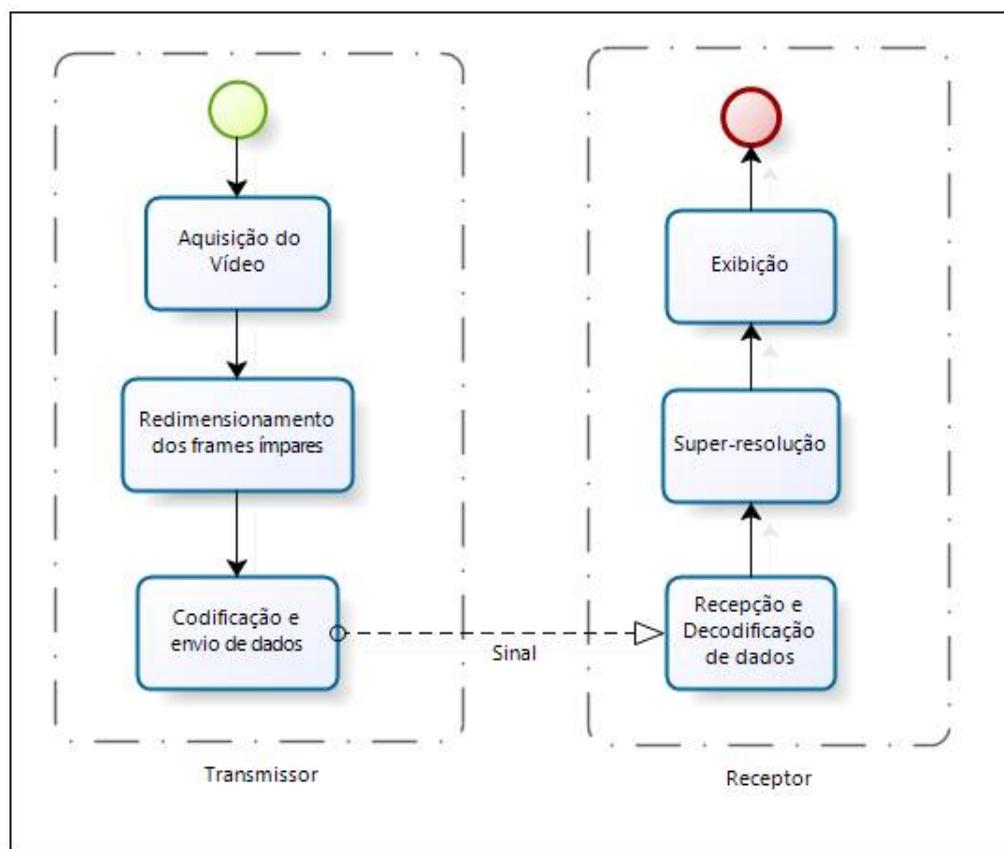


Figura 22 – Diagrama de blocos proposto para solução.

Implementou-se os blocos de redimensionamento dos *frames* ímpares, que se dá no transmissor, pelo processo de filtragem (entre quadros pares e ímpares) e decimação por um fator M destes *frames*, e o bloco de super-resolução, que se dá no receptor, que consiste na implementação do algoritmo proposto na Seção 3.2.

4.1 Transmissor

4.1.1 Hardware

Para o transmissor, escolheu-se o *kit* de desenvolvimento *Raspberry Pi 2 Model B*, mostrado na Figura 23. O *Raspberry Pi 2* é um dos dispositivos mais populares atualmente

entre estudantes, profissionais e hobbistas da área de Eletrônica. Segundo o próprio site da *Raspberry Pi Foundation* [44], o dispositivo é um computador de dimensões reduzidas e de baixo custo, capaz de se conectar a um monitor de computador ou televisão e rodar um sistema operacional baseado em Linux, sendo um dos mais populares o *Raspbian* (distribuição *Debian* voltada especificamente para o *Raspberry Pi*).



Figura 23 – *Kit* de desenvolvimento *Raspberry Pi 2 Model B* [9].

A capacidade de rodar sistemas operacionais baseados em distribuições Linux foi uma característica decisiva na escolha do *kit* de desenvolvimento, pois dá ao desenvolvedor a possibilidade de utilizar *frameworks* e *codecs* já disponíveis no mercado, além da alta capacidade de processamento disponível dos quatro núcleos ARM (*Advanced RISC Machine*) e do processador gráfico dedicado. Há também módulos de aquisição e exibição de vídeo desenvolvidos e otimizados especialmente para a arquitetura deste dispositivo, o que pode facilitar o desenvolvimento futuro do bloco de aquisição e possibilitando a exibição local caso necessária. Segue um apanhado geral das características desta placa [44] :

- CPU de quatro núcleos ARM Cortex-A7 com frequência de processamento de 900 MHz;
- 1 GB de memória RAM (contra 512 MB do modelo B+);
- 4 portas USB;
- 40 pinos GPIO (do inglês, *General Purpose Input/Output*);
- Porta Full HDMI;
- Porta *Ethernet*;

- Audio jack de 3.5mm e entrada de vídeo combinados;
- Interface de câmera (CSI);
- Interface para display (DSI);
- Slot para Micro cartão SD;
- Núcleo de gráficos VideoCore IV 3D;

Dentre os 40 pinos GPIO, têm-se pinos com funções dedicadas como: tensões de entrada e referência, além de diversos tipos de comunicações (I2C, SPI, serial, etc), como mostrado na Figura 24.

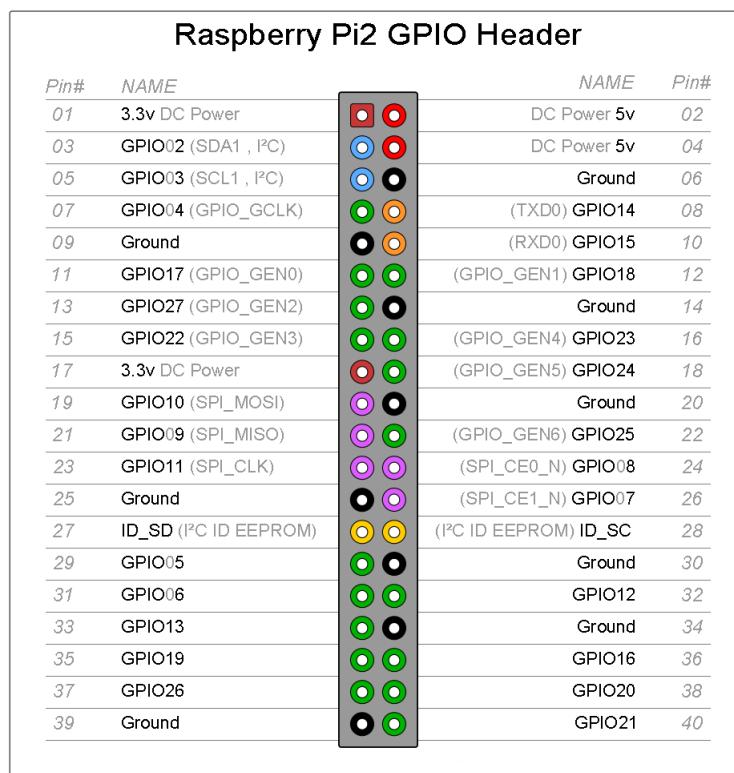


Figura 24 – Mapa de pinos do *Raspberry Pi 2* Modelo B [9].

Assim, optou-se por utilizar *Raspberry Pi2 Model B*, rodando um sistema operacional *Arch Linux Arm*, em um cartão micro SD de 8 Gb, com um *Python* versão 2.7. Utilizou-se o editor de texto *Vim*, assim como conceitos de orientação a objetos e o *multimedia framework* FFmpeg versão 2.8.2, fazendo uso principalmente do *FFmpeg Tools*, uma ferramenta que permite redimensionar, codificar e converter vídeos em diversos formatos e um *framework* de versionamento de código Git versão 2.1.4 .

4.1.2 *Software*

No transmissor trabalhou-se na separação e redimensionamento dos quadros ímpares e, também na compressão das sequências de vídeo. Para isso, desenvolveu-se um *script* na linguagem *Python*, com utilização de *FFmpeg*, um *multimedia framework* de código aberto, que por sua vez, é capaz de codificar e decodificar em diversos padrões, incluindo os dois propostos no trabalho (JPEG e H.246/AVC). Optou-se por essa linguagem devido a alta integração com o *hardware* proposto, integração com *frameworks* para cálculos científicos (*SciPy*) e experiência prévia do desenvolvedor.

A taxa de transmissão é controlada a partir de parâmetros relacionados ao coeficiente de quantização citado nas Seções 2.3.3 e 2.3.4, que definem o nível de quantização utilizado no processo.

4.2 Receptor

4.2.1 *Hardware*

Dentro do contexto proposto neste trabalho, o receptor terá amplos recursos de computação e energia, pois ele estará na superfície da mina. Já o transmissor estará exposto às condições adversas de dentro da mina. Assim, para o protótipo proposto neste trabalho, utilizou-se um computador pessoal com as seguintes configurações: um *notebook* Asus X44C-VX029R - Intel Core i3-2330M - RAM 4GB - HD 320GB, rodando um sistema operacional *Debian* 8 (Jessie), com um *GNU Octave*, versão 3.8.2. Utilizou-se também *multimedia framework* *FFmpeg* versão 2.8.2, fazendo uso principalmente do *FFmpeg Tools*, uma ferramenta que permite redimensionar, codificar e converter vídeos de diversos formatos e um *framework* de versionamento de código *Git* versão 2.1.4 .

4.2.2 *Software*

A implementação do algoritmo de super-resolução proposto para o decodificador foi feita na linguagem *Octave*, versão 3.8.2, pois ela é otimizada para o trato de matrizes e também devido à experiência prévia do desenvolvedor. O script desenvolvido recebe como entrada as sequências de vídeo vindas do transmissor, e gera as sequências de vídeo super-resolvidas com base no algoritmo apresentado no Capítulo anterior.

5 Resultados Experimentais

Este Capítulo visa apresentar os resultados dos testes realizados para avaliar o desempenho do algoritmo proposto na Seção 3.2. Foi realizada uma bateria de testes com sequências de vídeos reais, codificadas em JPEG e H.264/AVC.

Todos os testes foram realizados utilizando os *scripts* do transmissor e do receptor, escritos em *Python* e *Octave* respectivamente, com exceção do processo de estimação de movimento, que se encontra em um arquivo *mex* (linguagem C padrão ANSI), pois tratar-se de um processo complexo.

As sequências de vídeo utilizadas foram *akiyo*, *foreman*, *bowing*, *carphone*, *crew*, *flower*, *footbal* e *mobile*, visualmente apresentadas pela Figura 25 (ilustradas pelo segundo quadro de cada sequência). Com o propósito de validar o método, tais sequências foram escolhidas por possuirem diversos graus de quantidade de movimento, elementos novos/-recorrentes em cena e quantidade/intensidade de alta-frequência impondo assim diversos níveis de dificuldade à super-resolução proposta [43].



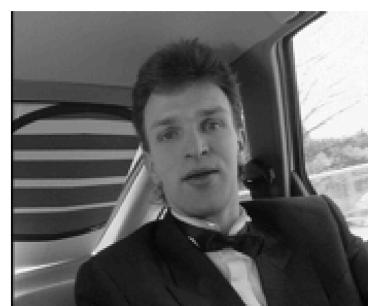
(a)



(b)



(c)



(d)



(e)



(f)



(g)



(h)

Figura 25 – Sequências de vídeo : (a) *akiyo*, (b) *foreman*, (c) *bowing*, (d) *carphone*, (e) *crew*, (f) *flower*, (g) *footbal* e (h) *mobile*.

5.1 Resultados Objetivos

Nesta Seção serão apresentados resultados objetivos, em formato de gráficos e tabelas, obtidos através de métricas definidas na Seção 2.4.

5.1.1 Teste com codificação JPEG

A fim de avaliar o desempenho do algoritmo proposto para a decodificação em uma situação prática, aplicou-se ao algoritmo sequências de vídeo codificadas dentre as seguintes condições:

- Cada sequência foi codificada separadamente utilizando o padrão JPEG, em resolução mista.
- Aplicou-se valores 2 a 31 ao parâmetro *qscale*, que é a faixa permitida pelo *software* utilizado para implementação do padrão, o que possibilitou o levantamento das curvas ilustradas nas Figuras 26 e 27.
- Utilizou-se o *framework ffmpeg*, especialmente a biblioteca *mjpeg* para implementar a codificação no padrão JPEG.
- A luminância foi escolhida para os testes, definindo os quadros originais como referência mediu-se o PSNR médio entre eles e os quadros interpolados, os quadros super-resolvidos e os quadros originais codificados e decodificados.
- A taxa considerada foi as dos quadros em baixa resolução.
- Foram utilizados macroblocos de tamanho 8x8 *pixels* e uma janela de busca de 80x80 *pixels* para o processo de estimação/compensação de movimento e para a combinação de altas frequências, e fatores de decimação $M = 2$ e $M = 4$.
- Foram medidos os ganhos médios [45] de qualidade (PSNR) para o algoritmo proposto em relação aos quadros interpolados. Estes resultados são ilustrados nas Tabelas 1 e 2.

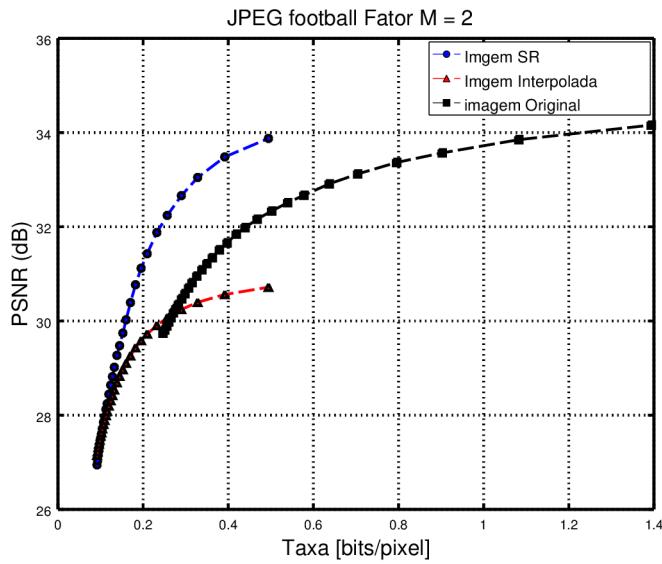


Figura 26 – Curva PSNR médio x Taxa, para a sequência de vídeo *Football*, com um fator de decimação $M = 2$.

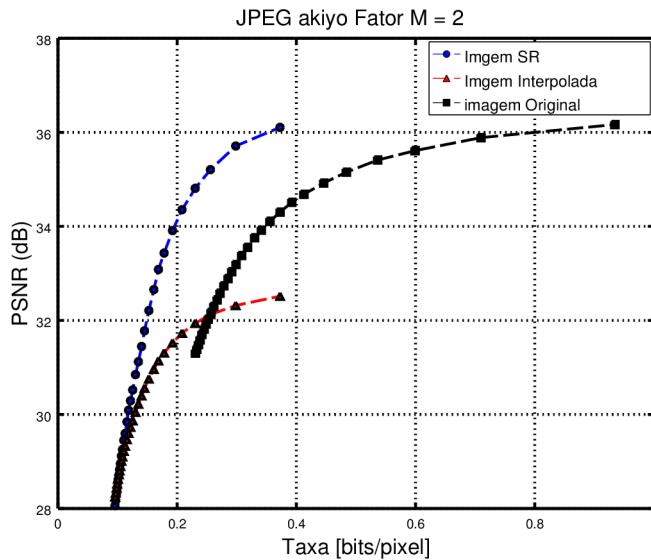


Figura 27 – Curva PSNR médio x Taxa, para a sequência de vídeo *Crew*, com um fator de decimação $M = 2$.

Os gráficos das Figuras 26 e 27, incluem duas curvas de referência: a inferior que representa a qualidade dos quadros interpolados e a superior que representa a qualidade dos quadros originais ambos codificados pelo padrão JPEG. As curvas destas duas Figuras representam o comportamento característico das curvas geradas pelas oito sequências de vídeo. As sequências de vídeo mais estáticas como *Akiyo* e *Carphone* apresentaram os melhores resultados, como esperado, consequentemente os maiores ganhos. As sequências que possuem um pouco mais de movimento como *Foreman* e *Crew* obtiveram resultados

satisfatórios. Para as sequências que apresentam muito movimento e a várias entradas de novos elementos nas cenas, como *Mobile* e *Flower*, obteu-se resultados mais sigilos, mas que ainda assim são melhores que os dos quadros interpolados.

Tabela 1 – Tabela de ganhos médios para sequências codificadas no padrão JPEG com fator de decimação $M = 2$.

Tabela de Ganhos	
Para fator de decimação $M = 2$	
Sequência	Ganho (dB)
<i>Akiyo</i>	2.0724
<i>Foreman</i>	1.5590
<i>Bowing</i>	1.6443
<i>Carphone</i>	1.8277
<i>Crew</i>	1.7288
<i>Flower</i>	1.3928
<i>Football</i>	1.5712
<i>Mobile</i>	1.2845
Média	1.6351

Tabela 2 – Tabela de ganhos médios para sequências codificadas no padrão JPEG com fator de decimação $M = 4$.

Tabela de Ganhos	
Para fator de decimação $M = 4$	
Sequência	Ganho (dB)
<i>Akiyo</i>	2.1661
<i>Foreman</i>	1.5593
<i>Bowing</i>	1.2851
<i>Carphone</i>	2.0060
<i>Crew</i>	1.9427
<i>Flower</i>	1.2137
<i>Football</i>	1.5794
<i>Mobile</i>	1.0615
Média	1.0617

5.1.2 Teste com codificação H.264/AVC

A fim de avaliar o desempenho do algoritmo proposto, sobre o padrão H.264/AVC, aplicou-se ao algoritmo sequências de vídeo codificadas dentre as seguintes condições:

- Cada sequência foi codificada separadamente utilizando o padrão H.264, em resolução mista.

- Aplicou-se valores de 16 a 51 ao parâmetro qp , a faixa permitida pelo *software* utilizado para implementação do padrão é de 0 a 51, sendo que 0 gera a codificação de melhor qualidade. Notou-se que utilizando os parâmetros de 0 a 15, a sequência decodificada tende a sequência original de tal forma que o cálculo do PSNR tende a infinito devido a limitação numérica da máquina utilizada, o que não é interessante para a abordagem do teste e por isso essa sequência de parâmetros foi desprezada.
- Utilizou-se o *framework* *ffmpeg*, especialmente a biblioteca *libx264* para implementar a codificação no padrão H.264.
- Assim como para o padrão JPEG, foi considerada somente as taxas dos quadros em baixa resolução. Utilizou-se os mesmos tamanhos de janela e macroBlocos que para o padrão JPEG.
- Calculou-se os ganhos médios [45] de qualidade (PSNR) para o algoritmo proposto em relação aos quadros interpolados. Estes resultados são ilustrados na Tabela X2.

5.1.3 Resultados Subjetivos

Observando as imagens desta Seção, pode-se perceber que mesmo para as sequências de vídeo com grande quantidade de movimento e novos elementos a cada cena, as imagens super-resolvidas são apresentadas sem detalhes visualmente incômodos.

Em geral as bordas, letras, algumas texturas e detalhes são difíceis de se ver nos quadros interpolados, no entanto ficam claramente visíveis nos quadros super-resolvidos, o que mostra que as informações de alta frequência adicionadas aos quadros interpolados possuem um bom *mathing*.

5.1.4 Teste com codificação JPEG



(a)



(b)



(c)

Figura 28 – *Foreman* - Quadro 6, $M = 2$, $Qscale = 10$: (a) Imagem Original, (b)Imagem Interpolada (PSNR = 21.950 dB), (c)Imagen Super-resolvida (PSNR = 21.776 dB).



(a)



(b)



(c)

Figura 29 – *Football* - Quadro 4, $M = 2$, $Qscale = 10$: (a) Imagem Original, (b) Imagem Interpolada (PSNR = 16.847 dB), (c) Imagem Super-resolvida (PSNR = 17.176 dB).



(a)



(b)



(c)

Figura 30 – *Mobile* - Quadro 4, $M = 2$, $Qscale = 10$: (a) Imagem Original, (b)Imagem Interpolada (PSNR = 16.532 dB), (c)Imagen Super-resolvida (PSNR = 17.902 dB).

5.1.5 Teste com codificação H.264/AVC



(a)



(b)



(c)

Figura 31 – *Foreman* - Quadro 30, $M = 2$, $Qscale = 10$: (a) Imagem Original, (b)Imagem Interpolada (PSNR = 26.399 dB), (c)Imagen Super-resolvida (PSNR = 31.503 dB).



(a)



(b)



(c)

Figura 32 – *Football* - Quadro 3, $M = 2$, $Qscale = 10$: (a) Imagem Original, (b)Imagem Interpolada (PSNR = 22.946 dB), (c)Imagen Super-resolvida (PSNR = 24.286 dB).



(a)



(b)



(c)

Figura 33 – *Mobile* - Quadro 33, $M = 2$, $Qscale = 10$: (a) Imagem Original, (b)Imagem Interpolada (PSNR = 18.756 dB), (c)Imagen Super-resolvida (PSNR = 21.211 dB).

6 Considerações Finais

O presente trabalho teve como objetivo a implementação de um sistema de comunicação de imagens em baixo consumo, mesmo com uma largura de banda de transmissão muito baixa, permitindo o envio de informações vitais por operários presos na mina para equipes de resgate. Para isso, foram elaborados os objetivos específicos (produtos de trabalho) para que o objetivo geral fosse alcançado. Os objetivos específicos foram estudar técnicas rápidas de downsampling para codificar a imagem em resolução mais baixa possível, desenvolver um algoritmo para técnicas rápidas de downsampling, prototipar, desenvolver e validar *hardware* e *software*.

Referências

- 1 MAIMONE, M. *The Computer Vision Homepage*. 2014. Citado 4 vezes nas páginas [9](#), [18](#), [20](#) e [25](#).
- 2 SCURI, A. E. Fundamentos da imagem digital. *Pontifícia Universidade Católica do Rio de Janeiro*, 1999. Citado 2 vezes nas páginas [9](#) e [19](#).
- 3 GARCIA, D. C. Técnicas de super-resolução para sistemas de vídeo de múltiplas vistas em resolução mista. 2013. Citado 11 vezes nas páginas [9](#), [10](#), [18](#), [22](#), [23](#), [33](#), [34](#), [35](#), [36](#), [37](#) e [38](#).
- 4 SALAZAR, C.; TRAN, T. D. A complexity scalable universal dct domain image resizing algorithm. *Circuits and Systems for Video Technology, IEEE Transactions on*, IEEE, v. 17, n. 4, p. 495–499, 2007. Citado 4 vezes nas páginas [9](#), [22](#), [23](#) e [26](#).
- 5 WALLACE, G. K. The jpeg still picture compression standard. *Communications of the ACM*, AcM, v. 34, n. 4, p. 30–44, 1991. Citado 4 vezes nas páginas [9](#), [28](#), [29](#) e [30](#).
- 6 RICHARDSON, I. E. *The H. 264 advanced video compression standard*. [S.l.]: John Wiley & Sons, 2011. Citado 8 vezes nas páginas [9](#), [19](#), [20](#), [21](#), [24](#), [26](#), [30](#) e [31](#).
- 7 KANG, M.; UM, D.-Y.; OH, W.-J. The digital image acquisition of high-resolution by enhancement the multiple images. *ASPRS-Proceedings-Launching the Geospatial Information Age, Washington*, 2000. Citado 3 vezes nas páginas [9](#), [33](#) e [34](#).
- 8 HUNG, E. M. Realce de vídeo para seqüências de qualidade e resolução variáveis. 2013. Citado 3 vezes nas páginas [10](#), [35](#) e [36](#).
- 9 COMMUNITY, E. *Raspberry pi 2 Model B GPIO Header*. 2015. Citado 3 vezes nas páginas [10](#), [40](#) e [41](#).
- 10 FOROOSHANI, A. E. et al. A survey of wireless communications and propagation modeling in underground mines. *Communications Surveys & Tutorials, IEEE*, IEEE, v. 15, n. 4, p. 1524–1545, 2013. Citado na página [15](#).
- 11 BARKAND, T. D.; DAMIANO, N. W.; SHUMAKER, W. Through-the-earth, two-way, mine emergency, voice communication systems. In: IEEE. *Industry Applications Conference, 2006. 41st IAS Annual Meeting. Conference Record of the 2006 IEEE*. [S.l.], 2006. v. 2, p. 955–958. Citado na página [15](#).
- 12 WHEELER, H. A. Radio wave propagation in the earth's crust. *Journal of Research of the National Bureau of Standards*, v. 65, n. 2, p. 189–191, 1961. Citado na página [15](#).
- 13 RAAB, F. H.; JOUGHIN, I. R. Signal processing for through-the-earth radio communication. *Communications, IEEE Transactions on*, IEEE, v. 43, n. 12, p. 2995–3003, 1995. Citado na página [16](#).
- 14 BANDYOPADHYAY, L.; CHAULYA, S.; MISHRA, P. Wireless communication in underground mines. *RFID-Based Sensor Networking*, Springer, 2010. Citado na página [16](#).

- 15 YOUNG, I. T.; GERBRANDS, J. J.; VLIET, L. J. V. *Fundamentals of image processing*. [S.l.]: Delft University of Technology Delft, The Netherlands, 1998. Citado na página 17.
- 16 GONZALEZ, R. C.; WOODS, R. E.; EDDINS, S. L. *Digital image processing using MATLAB*. [S.l.]: Pearson Education India, 2004. Citado na página 17.
- 17 SCHARSTEIN, D.; SZELISKI, R. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International journal of computer vision*, Springer, v. 47, n. 1-3, p. 7–42, 2002. Citado na página 18.
- 18 ACHARYA, T. *Integrated color interpolation and color space conversion algorithm from 8-bit bayer pattern RGB color space to 12-bit YCrCb color space*. Google Patents, 2002. US Patent 6,392,699. Disponível em: <<https://www.google.com/patents/US6392699>>. Citado na página 21.
- 19 ITU-R, R. B. 601-5: Studio encoding parameters of digital television for standard 4:3 and wide. *Screen*, v. 16, n. 9. Citado na página 21.
- 20 DINIZ, P. S.; SILVA, E. A. D.; NETTO, S. L. *Digital signal processing: system analysis and design*. [S.l.]: Cambridge University Press, 2010. Citado na página 22.
- 21 ITU. *ISO/IEC 10918-1 : 1993(E) CCIT Recommendation T.81*. 1993. Disponível em: <<http://www.w3.org/Graphics/JPEG/itu-t81.pdf>>. Citado 2 vezes nas páginas 23 e 28.
- 22 KHAYAM, S. A. The discrete cosine transform (dct): theory and application. *Michigan State University*, 2003. Citado na página 24.
- 23 PATIL, V. et al. A fast arbitrary down-sizing algorithm for video transcoding. In: IEEE. *Image Processing, 2006 IEEE International Conference on*. [S.l.], 2006. p. 857–860. Citado na página 25.
- 24 CHANG, S.-F.; MESSERSCHMITT, D. G. Manipulation and compositing of mc-dct compressed video. *Selected Areas in Communications, IEEE Journal on*, IEEE, v. 13, n. 1, p. 1–11, 1995. Citado na página 25.
- 25 WANG, R.-J.; CHIEN, M.-C.; CHANG, P.-C. Adaptive down-sampling video coding. In: INTERNATIONAL SOCIETY FOR OPTICS AND PHOTONICS. *IS&T/SPIE Electronic Imaging*. [S.l.], 2010. p. 75420P–75420P. Citado na página 25.
- 26 DUGAD, R.; AHUJA, N. A fast scheme for image size change in the compressed domain. *Circuits and Systems for Video Technology, IEEE Transactions on*, IEEE, v. 11, n. 4, p. 461–474, 2001. Citado 2 vezes nas páginas 25 e 26.
- 27 MUKHERJEE, J.; MITRA, S. K. Image resizing in the compressed domain using subband dct. *Circuits and Systems for Video Technology, IEEE Transactions on*, IEEE, v. 12, n. 7, p. 620–627, 2002. Citado na página 25.
- 28 MARTUCCI, S. A. Image resizing in the discrete cosine transform domain. In: IEEE. *Image Processing, 1995. Proceedings., International Conference on*. [S.l.], 1995. v. 2, p. 244–247. Citado na página 26.

- 29 PARK, H.; PARK, Y.; OH, S.-K. L/m-fold image resizing in block-dct domain using symmetric convolution. *Image Processing, IEEE Transactions on*, IEEE, v. 12, n. 9, p. 1016–1034, 2003. Citado na página 26.
- 30 UKRIT, M. F.; UMAMAGESWARI, A.; SURESH, D. G. A survey on lossless compression for medical images. *International Journal of Computer Applications*, Citeseer, v. 31, n. 8, p. 47–50, 2011. Citado na página 27.
- 31 SAYOOD, K. *Introduction to data compression*. [S.l.]: Newnes, 2012. Citado 3 vezes nas páginas 27, 28 e 31.
- 32 SHANNON, C. E. A mathematical theory of communication. *ACM SIGMOBILE Mobile Computing and Communications Review*, ACM, v. 5, n. 1, p. 3–55, 1948. Citado na página 27.
- 33 SILVA, A. M. C. da. *Um Estudo Sobre o Padrão H. 264/AVC de Compressão de Vídeo*. Tese (Doutorado) — UNIVERSIDADE CATÓLICA DE PELOTAS, 2007. Citado na página 27.
- 34 WANG, C.-Y. et al. Jpeg-based image coding algorithm at low bit rates with down-sampling and interpolation. In: IEEE. *Wireless Communications, Networking and Mobile Computing, 2008. WiCOM'08. 4th International Conference on*. [S.l.], 2008. p. 1–5. Citado na página 28.
- 35 VERGÜTZ, S. Uma combinação entre os critérios objetivo e subjetivo na classificação de imagens mamográficas comprimidas pelo método fractal. 2013. Citado na página 32.
- 36 BAKER, S.; KANADE, T. Limits on super-resolution and how to break them. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, IEEE, v. 24, n. 9, p. 1167–1183, 2002. Citado na página 33.
- 37 PARK, S. C.; PARK, M. K.; KANG, M. G. Super-resolution image reconstruction: a technical overview. *Signal Processing Magazine, IEEE*, IEEE, v. 20, n. 3, p. 21–36, 2003. Citado na página 33.
- 38 FARSIU, S. et al. Advances and challenges in super-resolution. *International Journal of Imaging Systems and Technology*, Wiley Online Library, v. 14, n. 2, p. 47–57, 2004. Citado na página 33.
- 39 FIGUEIRA, N. M.; OLIVEIRA, L. C. d. Super-resolução: técnicas existentes e possibilidade de emprego às imagens do vant vt-15. *Revista Militar de Ciência e Tecnologia*, v. 30, p. 3–19, 2013. Citado na página 33.
- 40 MILANFAR, P. *Super-resolution imaging*. [S.l.]: CRC Press, 2010. Citado na página 33.
- 41 RÉZIO, A. C. C.; SCHWARTZ, W. R.; PEDRINI, H. *Superresolução de imagens baseada em aprendizado utilizando descritores de características*. 2013. Citado na página 35.
- 42 FREEMAN, W. T.; JONES, T. R.; PASZTOR, E. C. Example-based super-resolution. *Computer Graphics and Applications, IEEE*, IEEE, v. 22, n. 2, p. 56–65, 2002. Citado na página 35.

43 SILVA, F. B. d. Super-resolução utilizando quadros chave em sequências de vídeo de resolução mista. 2009. Citado 2 vezes nas páginas [37](#) e [43](#).

44 FOUNDATION, R. P. *Raspberry pi*. 2015. Citado na página [40](#).

45 BJONTEGAARD, G. Calculation of average psnr differences between rd-curves. *Doc. VCEG-M33 ITU-T Q6/16, Austin, TX, USA, 2-4 April 2001*, 2001. Citado 2 vezes nas páginas [45](#) e [48](#).