



PROCESSAMENTO EMBARCADO DE IMAGENS COM BAIXOS NÍVEIS DE CONTRASTE E ILUMINAÇÃO PARA REALCE DE OBJETOS OCULTOS

Nilton José Rizzo

Tese de Doutorado apresentada ao Programa de Pós-graduação em Engenharia de Sistemas e Computação, COPPE, da Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários à obtenção do título de Doutor em Engenharia de Sistemas e Computação.

Orientador: Ricardo Cordeiro de Farias

Rio de Janeiro

Dezembro de 2020

PROCESSAMENTO EMBARCADO DE IMAGENS COM BAIXOS NÍVEIS DE
CONTRASTE E ILUMINAÇÃO PARA REALCE DE OBJETOS OCULTOS

Nilton José Rizzo

TESE SUBMETIDA AO CORPO DOCENTE DO INSTITUTO ALBERTO
LUIZ COIMBRA DE PÓS-GRADUAÇÃO E PESQUISA DE ENGENHARIA
DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS
REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE DOUTOR
EM CIÊNCIAS EM ENGENHARIA DE SISTEMAS E COMPUTAÇÃO.

Orientador: Ricardo Cordeiro de Farias

Aprovada por: Prof. Ricardo Cordeiro de Farias

Prof. Alexandre de Assis Bento Lima

Prof. Cláudio Esperança

Prof. Esteban Walter Gonzalez Clua

Prof. Luiz Marcos Garcia Gonçalves

RIO DE JANEIRO, RJ – BRASIL

DEZEMBRO DE 2020

Rizzo, Nilton José

Processamento Embarcado de Imagens com Baixos Níveis de Contraste e Iluminação para Realce de Objetos Ocultos/Nilton José Rizzo. – Rio de Janeiro: UFRJ/COPPE, 2020.

XIII, 69 p.: il.; 29, 7cm.

Orientador: Ricardo Cordeiro de Farias

Tese (doutorado) – UFRJ/COPPE/Programa de Engenharia de Sistemas e Computação, 2020.

Referências Bibliográficas: p. 56 – 61.

1. Ressonância Estocástica.
 2. Imagens com Baixa Iluminação.
 3. Imagens com Baixos Níveis de Contraste.
 4. Redução de Ruídos.
 5. Sistemas Embarcados.
 6. Raspberry PI.
- I. Farias, Ricardo Cordeiro de. II. Universidade Federal do Rio de Janeiro, COPPE, Programa de Engenharia de Sistemas e Computação. III. Título.

*Dedico ao meus pais, esposa e
filhos que sempre me apoiaram
nas minhas tomadas de decisões
em relação a evolução nos
estudos, cada vez mais
complexos. Deixo aqui meu
eterno agradecimento.*

Agradecimentos

Em primeiro lugar, agradecer a Deus que possibilitou os caminhos até aqui, ao professor Ricardo Cordeiro de Farias que acreditou que era possível a realização deste empreendimento, a meus pais, Maria José Rizzo e Paschoal Rizzo, que me deram todo o apoio para que eu pudesse chegar até aqui, sendo a sua dedicação como incentivadores da carreira acadêmica e de evolução moral.

A minha família, na figura de minha querida esposa Ana Cristina e de meus filhos Gabriel e Isabelle que souberam entender os desafios e tropeços necessários para se chegar até aqui.

Aos meus amigos e colegas do Laboratório de Computação Gráfica (LCG) que sempre estiveram dispostos a superar divergências de opiniões e que de uma maneira ou de outra me apoiaram nesta constante atualização de conhecimentos que é a nossa área da computação.

Aos meus amigos e colegas da Universidade Federal Rural do Rio de Janeiro, em especial dos departamentos de computação e matemática, que me apoiaram para que eu pudesse concluir este trabalho de maneira eficiente.

Agradecer ao Leonardo Gurgel Gomes pela paciência e ajuda na depuração e criação de ambiente virtual de *docker* e *python* para executar algoritmos de IA para comparação e teste.

Obrigado a todos vocês.

Resumo da Tese apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Doutor em Ciências (D.Sc.)

PROCESSAMENTO EMBARCADO DE IMAGENS COM BAIXOS NÍVEIS DE CONTRASTE E ILUMINAÇÃO PARA REALCE DE OBJETOS OCULTOS

Nilton José Rizzo

Dezembro/2020

Orientador: Ricardo Cordeiro de Farias

Programa: Engenharia de Sistemas e Computação

Apresentamos neste trabalho, um estudo sobre realce de imagens com baixíssimos níveis de contraste e de iluminação, para auxiliar o reconhecimento de objetos ocultos. Tendo como objetivo a utilização em equipamentos de poder computacional limitado, por exemplo o Raspberry PI. O método proposto utiliza a ressonância estocástica aliado a correção de histograma que fornece uma imagem final com a qualidade esperada. Para executar este estudo, foram realizadas pesquisas em vários artigos em que a técnica da ressonância estocástica foi utilizada individualmente ou agregada a outros métodos, como está descrito na seção de revisão bibliográfica. Os resultados obtidos foram comparados com técnicas envolvendo inteligência artificial, e os resultados foram expressivos. O conjunto de imagens utilizados nesta pesquisa constitui-se de imagens próprias e de imagens públicas utilizadas para teste dos métodos de inteligência artificial que utilizamos para comparar o resultado com o nosso método.

Abstract of Thesis presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Doctor of Science (D.Sc.)

EMBEDDED IMAGE PROCESSING WITH LOW LEVELS OF CONTRAST
AND ILLUMINATION FOR ENHANCING HIDDEN OBJECTS

Nilton José Rizzo

December/2020

Advisor: Ricardo Cordeiro de Farias

Department: Systems Engineering and Computer Science

In this work, we present a study on image enhancement with very low levels of contrast and lighting, to help the recognition of hidden objects. The application is aimed in equipments with limited computational power, such as Raspberry PI. The proposed method uses stochastic resonance combined with histogram correction that provides a final image with the expected quality. In order to carry out this study, researches were carried out in several papers in which the stochastic resonance technique was used either individually or summed with other methods, as described in the related works review section. The results obtained were compared with techniques using artificial intelligence, and the results were quite interesting. The set of images used in this research consists of own images and public images used to test and train the methods of artificial intelligence.

Sumário

Lista de Figuras	x
Lista de Tabelas	xiii
1 Introdução	1
1.1 Motivação	2
1.2 Objetivo	3
1.3 Metodologia	4
1.4 Organização deste trabalho	6
2 Revisão Bibliográfica	8
2.1 Aquisição de imagem	8
2.2 Demosaicing	9
2.3 Ressonância estocástica	13
2.4 Qualificação da imagem	18
2.5 Métodos de inteligência artificial	23
2.6 Escolha do sistema embarcado	23
3 Método Proposto	24
3.1 Método	24
3.2 Conjunto de imagens	26
3.3 Conversão de formato	28
3.4 Reconstrução dos planos de cor	29
3.5 Ressonância estocástica	29
3.6 Fator de parada	31
3.6.1 Índice de qualidade	31
3.6.2 Fator F	32
3.7 Redução de ruídos	33
3.8 Ambiente de teste	33
4 Resultados e Discussões	36

4.1	Obtenção das imagens	36
4.2	Nomenclatura dos arquivos	36
4.3	Resultados	37
4.3.1	Método proposto	37
4.3.2	Tempo de processamento	40
4.3.3	Condição de parada	42
4.3.4	Comparação com métodos a inteligência artificial	43
5	Conclusões	52
5.1	Contribuições	54
5.2	Trabalhos futuros	54
Referências Bibliográficas		56
A	Acesso ao conteúdo deste trabalho	62
A.1	Acesso ao código fonte	62
A.2	Acesso as figuras	62
B	Gráficos da condição de parada	63
B.1	Introdução	63
C	Analise de histograma	66
C.1	Introdução	66

Listas de Figuras

1.1	Foto típica noturna em <i>smartphones</i>	2
1.2	Ampliação da imagem para visualizar os artefatos/ruído	3
1.3	Copa do Laboratório, com luzes apagadas - Nikon D90	4
1.4	Copa do Laboratório, com luzes acesas - Nikon D90	4
1.5	Imagen de um Raspberry PI 3	5
1.6	Fluxograma do processo	7
2.1	Copa do laboratório - Imagem Raw adquirida com a Nikon D90.	9
2.2	Procissão em Paraty - Imagens JPG adquirida com a Nikon DF.	10
2.3	Filtro de Bayer - fonte de luz, filtro e sensor	11
2.4	Ampliação de 800% de uma imagem RAW	11
2.5	Diferentes filtros Bayer	12
2.6	Representação matricial do filtro de Bayer	12
2.7	Imagen do Big Ben em 256 tons de cinza, com variação do nível de comparação do sinal ([1])	14
2.8	Gráfico de uma função Bi estável - ([2])	16
2.9	Gráfico de uma partícula oscilando entre os dois vales - ([2])	16
2.10	Círculo somador com N limites - ([3])	17
2.11	Tempo - ([4])	17
2.12	Resultado comparativos do trabalho de Gupta et al - ([5])	18
2.13	Imagen corrigida - ([5])	18
2.14	Resultado do processo de ressonância estocástica com o filtro Weiner	19
2.15	Imagen de testes	21
2.16	Figura utilizada para a medição de qualidade - ([6])	22
3.1	Representação do espaço de cor HSV	27
3.2	Imagen 1 - Comportamento da média, desvio padrão e do fator F	32
3.3	Imagen 5 - Comportamento da média, desvio padrão e do fator F	32
3.4	Fotos originais utilizadas no trabalho	34
3.5	Fotos originais do conjunto de fotos obtidas em: https://paperswithcode.com/task/low-light-image-enhancement	35

4.1	Imagen imagem 9 processada em 200 ciclos da Ressonância Estocástica	40
4.2	Imagen 5 - processada com os parâmetros P1 + FG + H	41
4.3	Torre Eiffel	44
4.4	Parque	44
4.5	Quadro na parede	44
4.6	Igreja	44
4.7	Pôr do sol e montanhas	45
4.8	Telhado	45
4.9	Fundo do mar	45
4.10	Satélite	45
4.11	Pôr do sol e monumento	45
4.12	Vaso de planta	45
4.13	Bambu	46
4.14	Meninas	46
4.15	Procissão em Paraty	46
4.16	Procissão em Paraty	46
4.17	Torre Eiffel	47
4.18	Parque	47
4.19	Quadro na parede	47
4.20	Igreja	48
4.21	Pôr do sol nas montanhas	48
4.22	Telhado	48
4.23	Fundo do mar	49
4.24	Satélite	49
4.25	Pôr do sol e o templo	50
4.26	Vaso de flores	50
4.27	Bambu	50
4.28	Meninas	51
4.29	Procissão em Paraty	51
4.30	Procissão em Paraty	51
B.1	Gráficos da condição de parada da imagem 1	63
B.2	Gráficos da condição de parada da imagem 2	63
B.3	Gráficos da condição de parada da imagem 3	64
B.4	Gráficos da condição de parada da imagem 4	64
B.5	Gráficos da condição de parada da imagem 5	64
B.6	Gráficos da condição de parada da imagem 6	64
B.7	Gráficos da condição de parada da imagem 7	64
B.8	Gráficos da condição de parada da imagem 8	65

B.9	Gráficos da condição de parada da imagem 9	65
B.10	Gráficos da condição de parada da imagem 10	65
C.1	Análise de histograma para imagem 1	66
C.2	Análise de histograma para imagem 2	67
C.3	Análise de histograma para imagem 3	67
C.4	Análise de histograma para imagem 4	67
C.5	Análise de histograma para imagem 5	68
C.6	Análise de histograma para imagem 6	68
C.7	Análise de histograma para imagem 7	68
C.8	Análise de histograma para imagem 8	69
C.9	Análise de histograma para imagem 9	69
C.10	Análise de histograma para imagem 10	69

Lista de Tabelas

1.1	Especificações do Raspberry PI 3B+ ([7]).	6
2.1	Comparação entre os métodos propostos por Gupta at al. ([5]).	17
2.2	Comparação entre os métodos por Wueller, D. ([8]).	20
2.3	Comparação entre os métodos por Wang, Z. e Bovik, A. C. ([6]).	22
3.1	Informações sobre as fotos utilizadas neste trabalho	27
3.2	Codificação dos parâmetros utilizados	28
3.3	Lista de imagens utilizadas na comparação deste método com os métodos de IA	28
4.1	Codificação dos arquivos processados	37
4.2	Codificação para simplificar a leitura das imagens	37
4.3	Tempo para processar 200 iterações da imagem 9 com $\Delta_t = 0.001$. .	39
4.4	Tempo Total gasto	42

Capítulo 1

Introdução

A computação gráfica surgiu na década de 50, com a construção do primeiro computador capaz de operar em tempo real e ter uma saída gráfica rudimentar. Seu nome era Whirlwind ([9], [10]) foi construído pelo MIT. Deste então é uma área que está sempre em constante evolução e subdividida em várias sub-áreas tais como: processamento de imagens, visão computacional, simulações e jogos. Sob este aspecto é uma área que procura novos desafios para desenvolver e aprimorar seus métodos. Sendo uma área dinâmica, haverá novos campos a serem explorados

Neste sentido, este trabalho vem trazer uma pesquisa na área de processamento de imagens por um método capaz de realçar objetos ocultos em imagens com baixa iluminação e níveis de contraste, passível de ser executado de forma automata em computadores com baixo poder computacional e sem a necessidade de uso de GPU (*Graphic Processors Unit*). Uma das várias possibilidades de aplicação deste estudo é a implementação de autômatos para movimentação noturna em áreas com baixa iluminação, dando capacidade de reconhecimento de objetos em seu caminho.

Há vários métodos de realce de níveis de contraste, tais como correção por histograma, correção do canal gama, entre outros. Escolhemos utilizar a associação dos métodos de ressonância estocástica com o de correção por histograma, por estes serem fáceis de utilizar em computadores sem GPU e necessitar muito pouco poder computacional.

A escolha da ressonância estocástica como método base do nosso estudo, nos leva a um objetivo secundário que é elaborar um método de parada para a ressonância estocástica, já que este é um método recursivo e necessita de monitoração para saber se a imagem atual está melhor que a imagem anterior, o que inviabilizaria a sua utilização automata em equipamentos embarcados.

1.1 Motivação

As imagens noturnas ou em ambientes mal iluminados tem como características baixa iluminação, baixo nível de contraste e níveis elevados de ruídos, com isso a perda é sensível de informações da cena que se quer capturar. Na figura 1.1 vemos um exemplo de uma foto obtida por um celular SamSung S7 no modo automático em um local que possui baixa iluminação. Partindo desta ideia, surgiu o projeto de pesquisa para tentar melhorar essas imagens e realçar os objetos ocultos nessas condições de iluminação. Outro exemplo de imagem real com esse tipo de problema é apresentado na figura 1.3, onde mostra uma imagem obtida com a câmera Nikon D90 em modo manual, com contraste quase nulo e praticamente toda escura. Já a figura 1.4, é a mesma cena com iluminação normal, apresenta alto nível de contraste. Foram pesquisados vários métodos que pudessem se ajustar aos parâmetros necessários, que são: baixa iluminação, nível de ruído elevado e necessidade de poder computacional baixo. O que melhor se adequou foi a ressonância estocástica, e o estudo prosseguiu para este método.

A ideia básica está em conceber um método que torne simples e automática a correção dessas imagens e passível de ser embarcada em dispositivos simples, como o Raspberry PI [7]. Na tabela 1.1 podemos ver as especificações do modelo utilizado.



Figura 1.1: Foto típica noturna em *smartphones*

O método proposto permite que as fotos sejam tratadas e melhoradas com baixo comprometimento do seu conteúdo, ou seja, sem a introdução de novos artefatos na imagem.

Os artefatos são provenientes do alto nível de ruído gerado pela baixa iluminação, como podemos ver na figura 1.2. As manchas na imagens são ruídos gerados pela captura da câmera que não consegue interpretar corretamente os níveis de iluminação recebido.



Figura 1.2: Ampliação da imagem para visualizar os artefatos/ruído

O trabalho consiste que a partir de uma imagem de qualquer tipo ou um quadro de um vídeo, aplicar um conjunto de métodos simples de melhoria de níveis de contraste, como por exemplo a correção por histograma, associado a ressonância estocástica, que obtém, simultaneamente redução de ruídos e o incremento nos níveis de contraste e iluminação, e em seguida aplicarmos testes para medir a qualidade da imagem resultante. O resultado da pesquisa poderá ser utilizado além de melhoria das fotos, em imagens de câmeras de vigilância, melhorando a sua qualidade e percepção visual, como em outros cenários onde a iluminação esteja prejudicada, fornecendo níveis de visibilidade reduzida.

1.2 Objetivo

Como objetivo geral, conseguir um sistema autônomo de correção de imagens, sem a necessidade de uma pessoa para avaliar se ficou boa ou não. O trabalho visa criar um método de correção único para uma variedade de imagens em condições semelhantes, aplicar um coeficiente de qualidade que possa se aproximar do sistema de visão humana, embora de difícil caracterização pois o sistema humano tem algumas variáveis que denotam da experiência anterior do ser humano. O método é composto por um conjunto de procedimentos mais simples ([11]), como por exemplo: ajuste de histograma e ajuste do fator gama da imagem ([12]), associado e a ressonância estocástica ([2]) para a correção de níveis de contrastes e redução de ruído e a criação de um índice de avaliação de qualidade da imagem que indique a melhoria da



Figura 1.3: Copa do Laboratório, com luzes apagadas - Nikon D90



Figura 1.4: Copa do Laboratório, com luzes acesas - Nikon D90

imagem final em relação a imagem inicial, para que o método possa ser totalmente automático e passível de ser embarcado.

1.3 Metodologia

A metodologia consiste em corrigir a imagem até que apresente um índice de melhoria de qualidade significativo e redução de artefatos na imagem final. Para alcançarmos este objetivo, conectamos alguns procedimentos de processamento de imagem (figura 1.6), onde cada passo será estudado e melhorado para obtermos a melhor correção possível para um grupo de imagens diversificado. Basicamente atuaremos em quatro frentes, a saber: *Aquisição* - possibilidade de usar várias formas de captura de imagem, *Melhoria da imagem* - que encadeando-se vários métodos para um melhor aproveitamento, e o *controle de qualidade da imagem*, onde iremos trabalhar



Figura 1.5: Imagem de um RaspBerry PI 3

alguns índices de correção, tentando sempre aprimorá-los, para que não seja necessário a intervenção humana no processo, *aplicação embarcada* - que será a colocação de todo o sistema em um RaspBerry PI com a webcam Logitech 270HD.

O conjunto de imagens utilizados vieram de duas fontes distintas. O primeiro conjunto de imagens foi obtido utilizando um celular Samsung S7, uma câmera fotográfica Nikon D90 ([13]), uma câmera Nikon DF ([14]) e uma webcam Logitech 270HD. Já o segundo conjunto é de referência de três artigos sobre correção de imagens utilizando redes neurais.

Com o celular, além das fotos, foram feitos pequenos vídeos em ambientes com baixíssima iluminação, sem o uso do flash, para simular a câmera embarcada no RaspBerry PI. A câmera fotográfica Nikon foi configurada para modo manual, onde variamos o tempo de abertura, de exposição e a ISO, produzindo efeitos de variação de iluminação. A webcam foi utilizada com a sua configuração padrão. As imagens com câmeras fotográficas podem ser em formato RAW, que possibilita que nenhuma intervenção seja aplicada pela máquina, sendo assim, as imagens estão *limpas* e *crua* para aplicarmos todos os procedimentos sugerido neste trabalho. Todas as imagens estarão disponíveis no *google drive* para uso de outros pesquisadores que quiserem utilizar-as. O endereço é: <http://encurtador.com.br/sSVY7>

Em seguida a captura da imagem, convertemos do formato de arquivo proprietário, da Samsung (DNG) ou da Nikon (NEF), para um formato aberto que possamos utilizar. Para esse fim utilizou-se o programa de código aberto e livre, dcraw ([15]), que converteu os arquivos para o formato aberto PGM.

Processador SoC	Broadcom BCM2837B0 quad-core A53 (ARMv8) 64-bit @ 1.4GHz
GPU	Broadcom Videocore-IV
RAM	1GB LPDDR2 SDRAM
Rede	Gigabit Ethernet (via USB channel), 2.4GHz and 5GHz 802.11b/g/n/ac Wi-Fi
Bluetooth	Bluetooth 4.2, Bluetooth Low Energy (BLE)
Armazenamento	Micro-SD
GPIO	40-pin GPIO header, populated
Portas	HDMI, 3.5mm analogue audio-video jack, 4x USB 2.0, Ethernet, Camera Serial Interface (CSI), Display Serial Interface (DSI)
Dimensões	82mm x 56mm x 19.5mm, 50g

Tabela 1.1: Especificações do Raspberry PI 3B+ ([7]).

Para o ambiente de desenvolvimento e testes, optou-se em utilizar o sistema operacional FreeBSD ([16]), versão 13-current, em uma arquitetura de 64bits sobre um processador FX-8300 da empresa AMD ([17]) com 8Gbytes de memória RAM e rodando o servidor gráfico X Window ([18]) e interface gráfica Mate ([19]), todos sendo programas livres e de código aberto.

Para o sistema embarcado, o Raspberry PI foi escolhido por ser uma plataforma barata e de fácil aquisição no mercado nacional, com bom custo/benefício. A versão do hardware escolhido foi o Raspberry PI 3B+ (1.5). Para procedimentos básicos com as imagens escolhemos a biblioteca OpenCV ([20]) que será utilizada para leitura, escrita e gerenciamento de janelas, entre outras funções básicas.

1.4 Organização deste trabalho

O segundo capítulo, *Revisão Bibliográfica* mostra de maneira sucinta conceitos, formulações, de acordo com a literatura, demonstrando os problemas atuais das soluções e seus desenvolvimentos, iniciando a escolha de métodos para serem utilizados em nosso trabalho de melhoramento da qualidade das imagens;

No terceiro capítulo, *Método Proposto* é apresentada direção tomada no trabalho de pesquisa, visando a melhoria do processo com as associações dos métodos para análise;

No quarto capítulo, *Resultados e Discussões*, apresentamos dados finais, assim como resultados dos caminhos escolhidos e onde poderemos chegar;

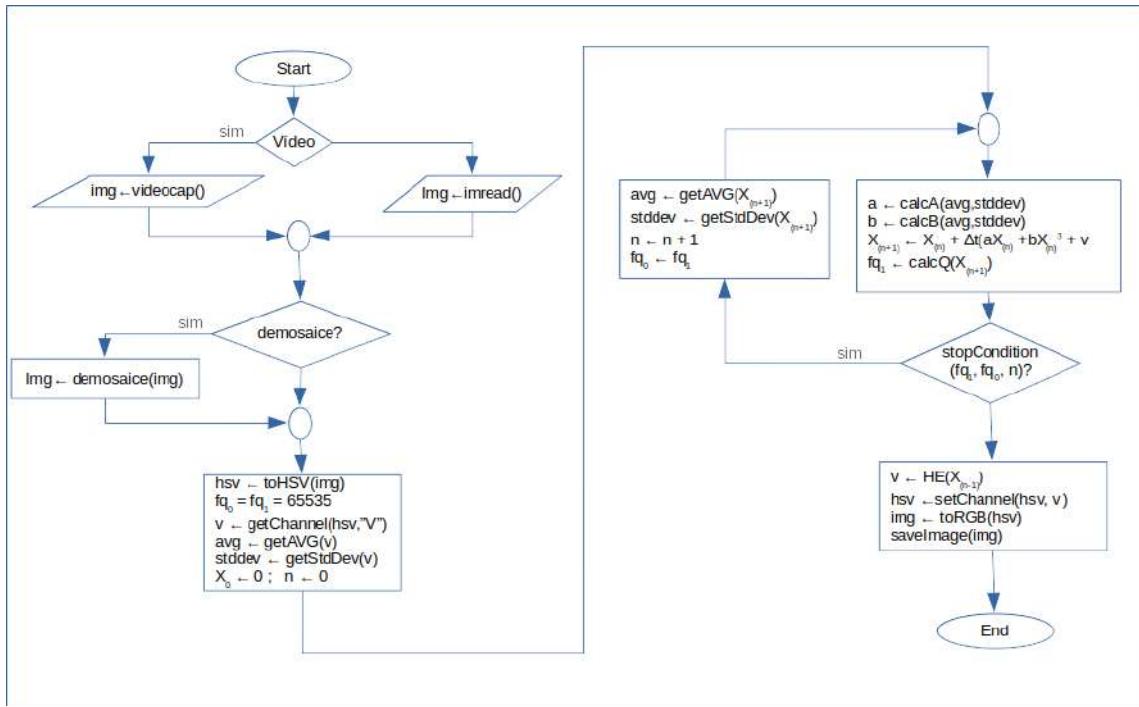


Figura 1.6: Fluxograma do processo

No quinto capítulo, Conclusões apresentamos, nossos resultados finais e a conclusão do trabalho

Capítulo 2

Revisão Bibliográfica

A revisão bibliográfica teve início pela bibliografia básica de processamento de imagens digital, como por exemplo os livros básicos na área da computação gráfica e processamento de imagens, Luiz Velho ([21]), do Gonzalez ([12]), do Scuri ([22]) e do Anil K. Jain ([23]), para fazer um estudo das técnicas de correção de contraste e redução de ruídos. Em seguida, foi feito um levantamento sobre os artigos publicados em congressos nos últimos anos sobre o tema. Muitos deles com resultados bem interessantes, outros nem tanto, mesmo assim a leitura trouxe muita clareza sobre o processo e ideias para novos trabalhos. Mesmo não sendo a ideia principal deste trabalho, foi analisado alguns artigos sobre melhoramento de imagens utilizando redes neurais, e na seção de resultados há uma comparação com esses métodos.

Para este trabalho, optou-se dividir em 4 etapas, a saber: aquisição da imagem, processamento da imagem, verificação de melhoria por índices objetivos e/ou subjetivos, e finalmente embarcar em um dispositivo de fácil acesso. Realizando busca de artigos que pudessem ajudar a desenvolver e trazer um novo olhar sobre o problema no processamento de imagens para direcionar corretamente o trabalho.

2.1 Aquisição de imagem

O primeiro passo para o trabalho foi de gerar um conjunto de imagens para testes, e o formato inicialmente pensado foi o *RAW* por apresentar informação apenas de luminância ([24]) e não ter nenhum processamento pelo equipamento de captura. Esta imagem se assemelha a uma imagens em tons de cinza, como é mostrada na figura 2.1. Para a construção do conjunto de imagens utilizadas neste trabalho, foi utilizado os seguintes equipamentos:

- Câmera Nikon D90

- Celular Samsung S7
- Webcam Logitech HD270C
- Câmera Nikon DF

Com a evolução dos estudos foi verificado que este formato não seria conveniente para o tipo padrão de entrada pelo fato de ser possível a utilização de vídeos adquiridos com o celular ou webcam, onde esse formato não estaria disponível para a captura. O sistema ainda aceita o RAW, mas faz apenas para ser o mais genérico possível.

A camera Nikon D90 foi configurada de modo manual, onde obtivemos as imagens em um formato proprietário **NEF (Nikon Eletronic Format)**, o celular foi configurado com *flash* desativado e modo automático, uma webcam e duas fotos cedidas pelo fotógrafo João Rocha, utilizando a câmera Nikon DF (figuras: 2.2a e 2.2b). As imagens obtidas em RAW foram convertida para o formato **PGM (Portable Gray Map)** pelo programa programa dcraw ([15]) e se encontrar no anexo C.



Figura 2.1: Copo do laboratório - Imagem Raw adquirida com a Nikon D90.

2.2 Demosaicing

Demosaling ou demosaicing é a técnica que reconstrói os planos de cores de uma imagem através de uma imagem RAW, devemos realizar este procedimento para obtermos a imagem final. Neste sentido as patentes da Nikon ([25] e [26]), nos fornece uma ideia básica de como restaurar os canais de cores, já que a informação



(a)



(b)

Figura 2.2: Procissão em Paraty - Imagens JPG adquirida com a Nikon DF.

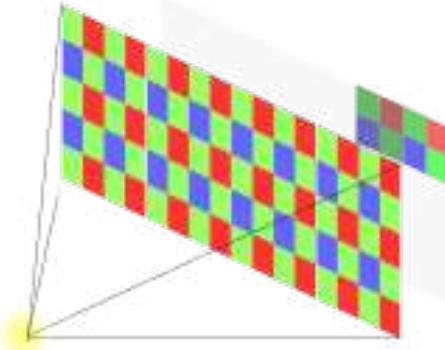


Figura 2.3: Filtro de Bayer - fonte de luz, filtro e sensor

que temos nos *pixels* da imagens referem-se a luminância, ou seja, a quantidade de luz que conseguiu atravessar o filtro Bayer (figura 2.3) colocado a frente do sensor de captura de imagem da câmera.

Esse filtro faz com que cada pixel da imagem comporte-se como sensor da quantidade luminosa que o atravessou, em uma determinada célula, com uma determinada cor. A imagem resultante pode ser vista na figura 2.4, que é uma ampliação de 800% da foto original, onde podemos verificar a formação de padrões que são resultado do filtro Bayer, deixando passar apenas as componentes das cores incidentes no sensor. Esses filtros podem ser composto de várias maneiras, mas as mais comuns são apresentadas na figura 2.5. A câmera da Nikon e o celular S7, geram os arquivo RAW em seus próprios formatos, entre outras informações, contém o tipo de filtro, a velocidade de abertura, o tempo de exposição e a ISO simulada pelo sistema.

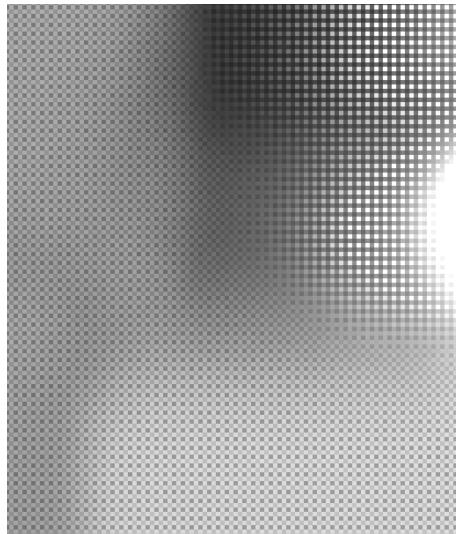


Figura 2.4: Ampliação de 800% de uma imagem RAW

Esse processo de reconstrução de canais de cores, é bem estudado e tem diversas variações, como aplicações de correção através de gradientes ([27]), correções de ganho do filtro, entre alguns trabalhos podemos destacar: Ramanath,R. et al. ([28])

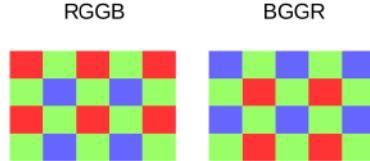


Figura 2.5: Diferentes filtros Bayer

que realiza uma comparação entre os métodos de interpolação bilinear, interpolação de matriz, interpolação baseada em gradiente, interpolação baseada em medianas e interpolação adaptativa de planos de cor, e verifica que existem ainda problemas em imagens com mudanças abruptas em níveis de luminosidade; neste outro artigo de Kimmel, R. ([29]) ele tenta solucionar o problema de vizinhanças com mudanças drásticas de níveis de iluminação (gradientes) com duas passadas, ou ainda, utilizando as patentes da própria Nikon, onde além de reconstruir os planos de cor, detecta *pixels* com defeito e os corrige. Como a lógica é implementada na própria eletrônica da câmera, o processo é extremamente rápido, traduz em cores bem fiéis e não gera artefatos na imagem.

Para a reconstrução dos planos de cores, a maioria desses trabalhos utiliza ao menos duas passadas, uma para criar o plano da cor verde e em seguida, uma passada para criar os planos das outras duas cores, azul e vermelha. Para a reconstrução do plano verde, geralmente se utiliza da média dos pixels vizinhos como podemos ver na equação 2.1. A figura 2.6 mostra um recorte do filtro bayer com a identificação de suas células.

C00	C01	C02	C03	C04	C05	C06	C07
C10	C11	C12	C13	C14	C15	C16	C17
C20	C21	C22	C23	C24	C25	C26	C27
C30	C31	C32	C33	C34	C35	C36	C37
C40	C41	C42	C43	C44	C45	C46	C47
C50	C51	C52	C53	C54	C55	C56	C57
C60	C61	C62	C63	C64	C65	C66	C67
C70	C71	C72	C73	C74	C75	C76	C77

Figura 2.6: Representação matricial do filtro de Bayer

$$[t]Vd_{(33)} = \frac{C_{23} + C_{43} + C_{33} + C_{34}}{4} \quad (2.1)$$

Já os outros planos, dependerá mais do modelo a ser seguido, porém o modelo de interpolação da matriz, considera a razão entre o pixel da cor desejada e a cor verde (equação 2.2 e 2.3) para uma melhor qualidade da cor final. Em um outro artigo, Lukin e Kubasov ([30]) fazem uma comparação entre vários métodos de interpolação, para uma avaliação qualitativa dos métodos.

$$Vm_{(33)} = \frac{Vd_{(33)} \times \left(\frac{Vm_{(32)}}{Vd_{(32)}} + \frac{Vm_{(52)}}{Vd_{(52)}} + \frac{Vm_{(34)}}{Vd_{(34)}} + \frac{Vm_{(54)}}{Vd_{(54)}} \right)}{4} \quad (2.2)$$

$$Az_{(33)} = \frac{Vd_{(33)} \times \left(\frac{Az_{(32)}}{Vd_{(32)}} + \frac{Az_{(52)}}{Vd_{(52)}} + \frac{Az_{(34)}}{Vd_{(34)}} + \frac{Az_{(54)}}{Vd_{(54)}} \right)}{4} \quad (2.3)$$

A partir deste ponto, temos a imagem com os canais no espaço de cor **RGB**, onde poderemos aplicar outras técnicas para diminuir ruídos e melhorar os níveis de contrastes. Para a aplicação neste trabalho, utilizamos a conversão de demosaicing da própria biblioteca OpenCV ([20]), que é considerada muito boa e foi suficiente para este trabalho.

2.3 Ressonância estocástica

Apesar de ser uma técnica antiga e já utilizada para outros contextos, a ressonância estocástica vem sendo estudada como ferramenta para melhoria de imagens há algum tempo, e com uma gama bem interessante de aplicações. A ressonância estocástica possibilita ao mesmo tempo, corrigir os baixos níveis de contrastes e os altos níveis de ruídos simultaneamente. Este estudo visa aplicar esta técnica em conjunto com outras para ampliar sua potencialidade. A ressonância estocástica ainda pode ser melhorada se otimizar parâmetros do sistema e adicionar ruídos simultaneamente ([31]).

Quando adquirimos um sinal, e ele está muito fraco para ser analisado, é possível agregar a este, ruído branco ou utilizar o próprio ruído do sinal para que faça o sinal entrar em *ressonância* e assim reforça-lo, este procedimento é chamado de ressonância estocástica ([32]). Este mecanismo pode elevar a relação de sinal/ruído a patamares que o sinal, que era quase indetectável, possa ser diferenciado do ruído e então analisado ([1]), para que isso ocorra, utiliza-se um potencial de limiar onde valores que conseguem suplanta-lo são reconhecidos como sinal, aqueles que não tem energia suficiente são descartados, como vemos na figura 2.7.

A equação fundamental para a utilização ([33]) advém de uma aplicação da equação de Langevin (equação 2.4).

$$dx = [x(a - x^2)]dt + \xi dW \quad (2.4)$$

onde:

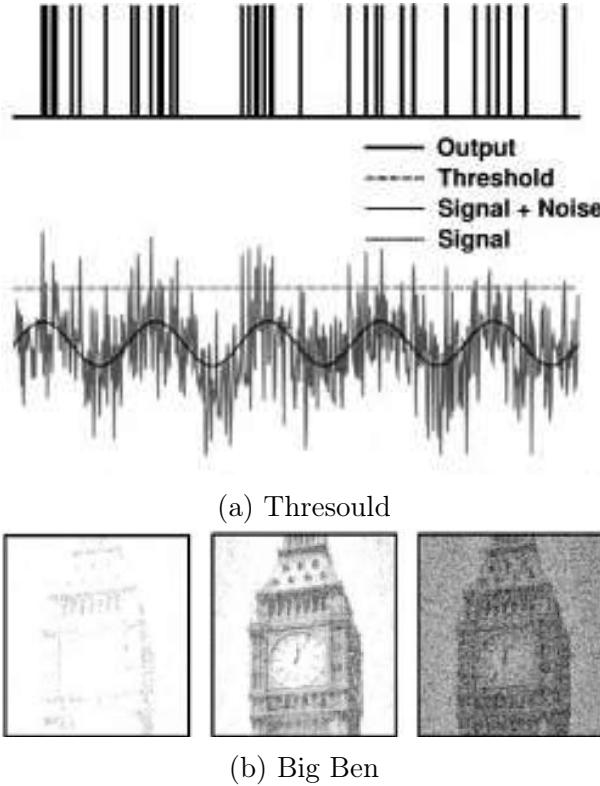


Figura 2.7: Imagem do Big Ben em 256 tons de cinza, com variação do nível de comparação do sinal ([1])

$\xi dW \rightarrow$ Ruído com média 0 (Gaussiano)

$W \rightarrow$ Processo de Wiener, onde podemos assumir um processo Gaussiano com média 0

Através da equação 2.4 podemos chegar a equação aplicada a imagens (equação), tanto no espaço da imagem quanto em espaço de frequência, como veremos nas aplicações desta técnica.

$$X_{(n+1)} = X_{(n)} + \Delta t[aX_{(n)} - bX_{(n)}^3 + B\sin(\omega t) + \sqrt{D\xi(t)}] \quad (2.5)$$

onde:

- | | |
|------------------------------------|--|
| $X_{(n)}$ | \rightarrow Imagem inicial |
| $X_{(n+1)}$ | \rightarrow Imagem corrigida |
| Δt | \rightarrow Variação temporal, entre duas iterações |
| $B\sin(\omega t) + \sqrt{D\xi(t)}$ | \rightarrow Amplitude (B) e frequência do sinal (ωt) + ruído |

Os coeficientes a e b são obtidos através da equação 2.6 que representa a relação sinal/ruído:

$$SNR = \frac{4a}{\sqrt{2}(\sigma_0\sigma_1)^2} \exp\left(-\frac{a}{2\sigma_0^2}\right) \quad (2.6)$$

$$a = 2\sigma_0^2 \quad (2.7)$$

$$b < \frac{4a^3}{27} \quad (2.8)$$

onde:

- $\sigma_0 \rightarrow$ desvio padrão da imagem
- $\sigma_1 \rightarrow$ desvio padrão do ruído

Esta é a principal equação utilizada em nosso processo, pois permite que ao mesmo tempo elevemos o sinal a níveis aceitáveis e reduzamos o ruído. Em nosso pesquisa, inicialmente os resultados não eram tão bons e tivemos que aprofundar o estudo nos parâmetros desta equação. O valor do fator b e de Δ_t foram ajustados respectivamente para 0.01 e 10^4 como são definidos nos artigos ([5] e [34]) porém as imagens ficavam ou todas preta ou totalmente branca. Variando essas variáveis entre os seguintes valores: $0.001 \leq b \leq 0.1$ e $10^3 \leq \Delta_t \leq 10^{18}$, começamos a obter resultados satisfatórios, como veremos na seção de resultados.

No artigo de Luca Gammaitoni et al. ([2]), ele descreve a utilização deste método aplicado na medicina, em sistemas neurais, em sistemas ópticos e em sistemas elétricos e magnéticos. Demonstra que uma partícula pode ficar oscilando entre um potencial qualquer Δ_V (figura 2.8), e esta mudança de estado faz com que o dado associado a esta partícula possa ser diferenciado do ruído, como mostra a figura 2.9. Quando a partícula ganha energia suficiente para ultrapassar o potencial Δ_V , conseguimos separar o sinal do ruído, mas quando a partícula perde energia e passa para o vale à esquerda, ele tem O potencial Δ_V é calculado pela equação 2.9.

$$\Delta V = -\frac{ax^2}{2} + \frac{bx^4}{4} \quad (2.9)$$

Neste outro artigo de Qinghua Ye et al ([35]), eles aplicam a ressonância estocástica em imagens de sonar, obtendo melhorias significativas na identificação da forma. O processo tem seu começo com a subdivisão da imagem em 9 setores, criando um vetor x_i , onde $1 \leq i \leq 9$, e reorganiza de forma randômica para ser processada pela ressonância estocástica. Eles não estão interessados com a melhoria visual, mas na detecção da forma, para a procura de minas subaquáticas, já que estas existem de

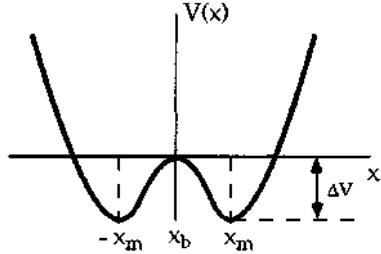


Figura 2.8: Gráfico de uma função Bi estável - ([2])

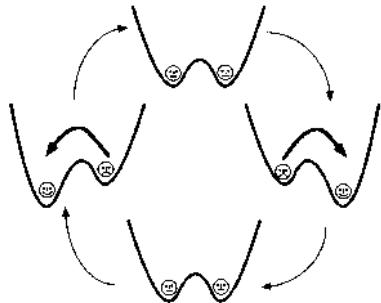


Figura 2.9: Gráfico de uma partícula oscilando entre os dois vales - ([2])

várias formas e tamanhos.

Em um estudo sobre ressonância estocástica e com multiníveis de comparação, o artigo ([3]) não é uma aplicação direta em imagens, mas em sinais elétricos (figura 2.10), porém nos mostra a eficiência do método para sistemas com níveis de comparações dinâmicos. O sinal de entrada $x_{(t)}$ é único para todas as entradas dos circuitos comparadores e o ruído $\eta_{i(t)}$ é independente para cada entrada, onde $1 \leq i \leq N$, assim pode-se aplicar níveis de comparações independentes, ao final soma-se todas as saídas dos comparadores ($y_{(t)} = \sum_{i=1}^N y_{i(t)}$), obtendo-se assim, uma melhoria do sinal como um todo.

No artigo de Rajlaxmi Chouhan ([4]), é demonstrada uma aplicação para imagens com regiões escuras e regiões com brilho, onde a imagem é melhorada para diminuir a faixa dinâmica da imagem, para que os níveis disparese sejam normalizados e trazidos para uma faixa mais agradável aos olhos. Esta aplicação da ressonância estocástica é feita no espaço de imagem (figura 2.11).

O artigo de Nidhi Gupta et al ([5]), utiliza além da ressonância estocástica, as transformações de *wavelet* e de cosseno em associação, em uma tentativa de melhorar o método uma vez que ela é aplicável tanto em domínio da frequência quanto no domínio da imagem. A figura 2.12a é a imagem original, a figura 2.12b é processada pelos três métodos, e a figura 2.12c é processada apenas pela transformação *wavelet* e a ressonância estocástica. Quando é feita a transformada, a ressonância estocástica é aplicada apenas nas partes mais escuras. Este artigo fornece também, uma

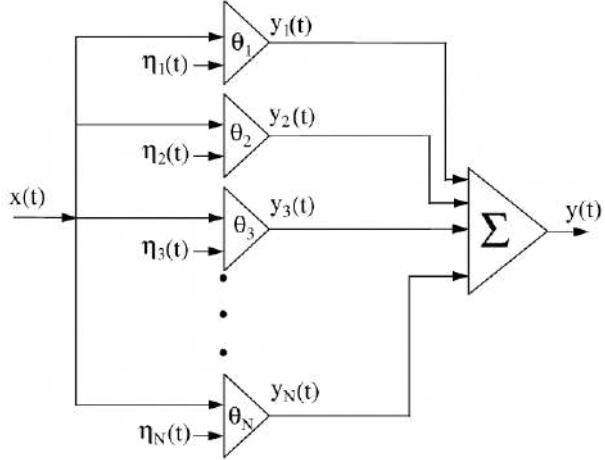


Figura 2.10: Circuito somador com N limites - ([3])



(a) Original

(b) Corrigida

Figura 2.11: Templo - ([4])

comparação entre os métodos: Equalização de Histogramas, *Single Scale Retinex* (SSR), *Multi Scale Retinex* (MSE), como podemos ver nas fotos (figura 2.13) e em seus tempos de processamento na tabela 2.1

O artigo de Jian Wang et al ([36]), é proposto um método para aplicar o filtro de Weiner para redução de ruídos em conjunto com a ressonância estocástica, onde obteve-se uma redução bem acentuada dos ruídos. O filtro de Weiner tem a sua aplicabilidade para eliminação de ruídos e do efeito de borramento da imagem, com isso obtemos imagens um pouco mais nítidas. Apesar de ter sido aplicado apenas em imagens em tons de cinza, obteve considerável resposta no melhoramento, como

Métodos	Tempo (s)
DWT + DCT + DSR	1,8
DWT + DSR	2,52
MSR	7,8
SSR	1,3
HE	2,76

Tabela 2.1: Comparação entre os métodos propostos por Gupta at al. ([5]).

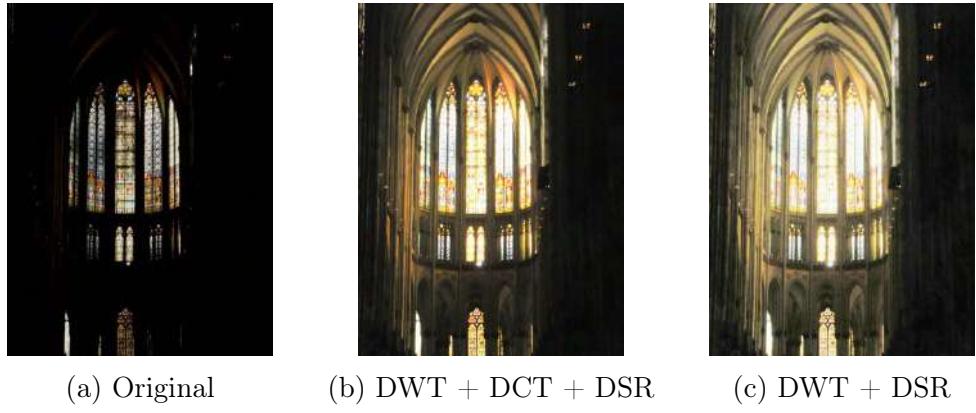


Figura 2.12: Resultado comparativos do trabalho de Gupta et al - ([5])

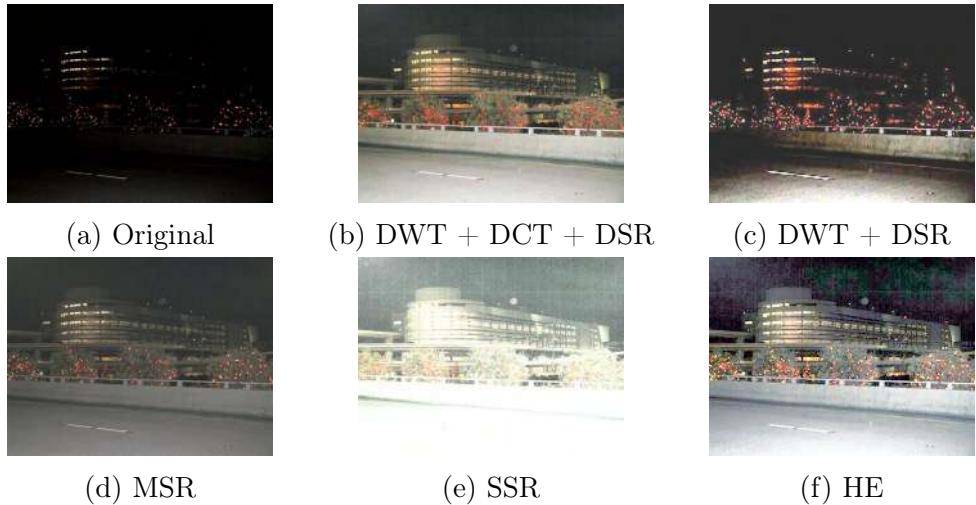


Figura 2.13: Imagem corrigida - ([5])

podemos ver na figura 2.14.

Além dos estudos de artigos sobre a ressonância estocástica como método primário de melhoria de imagens, foi realizado estudos sobre outros métodos e analisar sua possibilidade de ser agregados a este trabalho, entre eles: melhoria de imagens subaquáticas ([37]), melhoria de contraste através de análise de brilho dominante ([38]), uma revisão sobre vários métodos de melhoria de contrastes ([11]).

2.4 Qualificação da imagem

Para finalizar o estudo da revisão da bibliografia, foi feita uma busca por artigos sobre índice de qualidade de imagens, onde vimos que sempre existe uma porção de subjetividade a ser tratada, que é a diversidade humana. Essa diversidade, eventualmente, tem conceitos divergentes de imagens boas, pois cada indivíduo que vê um conjunto de imagens tem experiências diferentes e pode gerar percepção diferente entre imagens que não lhe são agradáveis aos olhos e com isso gerar um falso

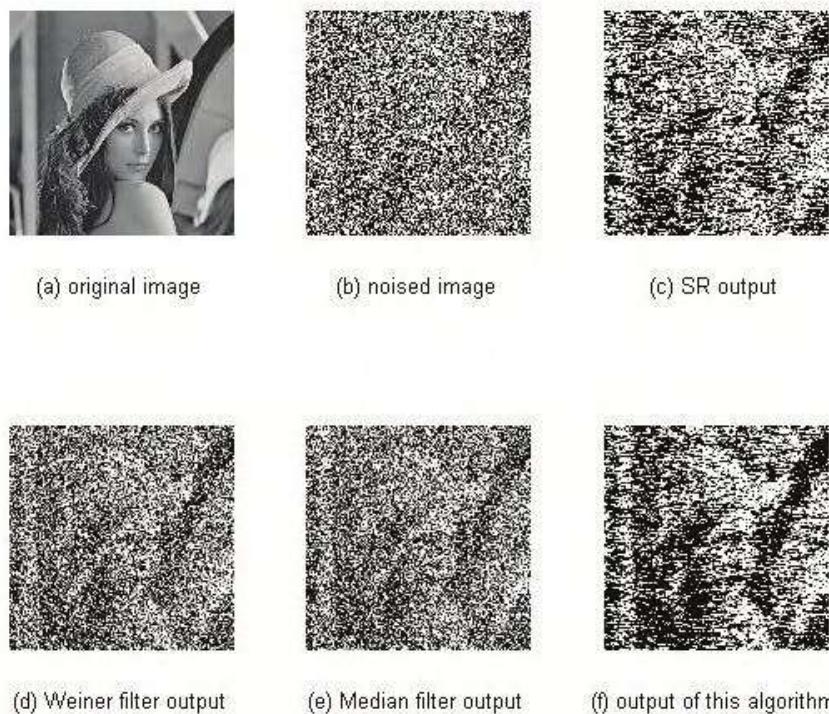


Figura 2.14: Resultado do processo de ressonância estocástica com o filtro Weiner

negativo ou falso positivo.

Para tentar evitar esse tipo de abordagem, a procura de índices objetivos, que possa ser calculado facilmente e obter uma comparação justa e eficiente para que o método proposto de correção possa ter resultados bem próximos do ideal.

Iniciando nossos estudos sobre este tópico, o artigo de Dietmar Wueller et al ([8]) onde analisa a padronização de análise de imagens e suas tolerâncias pela *International Organization for Standardization* (ISO) sob número **ISO 19264** aplicada a equipamentos de aquisição de imagens, como câmeras e *scanners*, como vemos na tabela 2.2. Esse estudo traça a qualidade do equipamento através da captura de uma imagem padronizada e então realiza a aferição do sistema por esta imagem. A imagem padrão é vista na figura 2.15.

Apesar de serem aplicados para verificação de qualidade de um sistema, pode-se fazer aproximações para trabalhar com imagens, onde obtemos uma imagem padrão e tentaremos ver o quanto nosso método consegue chegar desta imagem.

O artigo de Zhou Wang ([6]), nos mostra o quanto é difícil caracterizar um índice de qualidade de imagem objetivamente, e mesmo assim atribui um fator Q para cálculo de qualidade de imagem sendo:

Métodos	Excelente	Muito Bom	Bom
Tone reproduction	$\Delta L \leq +/- 2$	$\Delta L \leq +/- 3$	$\Delta L \leq +/- 5$
Gain Modulation (high-lights)	0.8 - 1.1	0.7 - 1.2	0.6 - 1.5
Gain Modulation (other tones)	0.6 - 1.4	0.4 - 1.7	0.1 - 2.0
Noise	< 3	< 4	< 6
Dynamic range	≥ 2.3	≥ 2.1	≥ 1.9
Banding (visual inspection)	None	None	Slight
Defect pixels (visual inspection)	None	< 0.1 per million	< 1 per million
White balance	$\Delta C \leq +/- 2$	$\Delta C \leq +/- 3$	$\Delta C \leq +/- 5$
Color reproduction (Max)	$\Delta E \leq +/- 10$	$\Delta E \leq +/- 15$	$\Delta E \leq +/- 18$
Color reproduction (Mean)	$\Delta E \leq +/- 4$	$\Delta E \leq +/- 5$	$\Delta E \leq +/- 5$
Sampling rate (Diff. claimed/obtained)	$\leq 1\%$	$\leq 2\%$	$\leq 3\%$
Resolution (ISO 16067)			
Percentage of claimed sampling rate	$\geq 85\%$	$\geq 80\%$	$\geq 70\%$
Sharpening (Max SFR contrast value)	≤ 1.05	≤ 1.1	≤ 1.2
Acutance	≥ 0.67	≥ 0.50	≥ 0.40
Illumination non-uniformity ($\leq A3$)	$\Delta L \leq 3$	$\Delta L \leq 3$	$\Delta L \leq 3$
Illumination non-uniformity ($\leq A2 > A3$)	$\Delta L \leq 4$	$\Delta L \leq 5$	$\Delta L \leq 5$
Illumination non-uniformity ($\leq A0 > A2$)	$\Delta L \leq 5$	$\Delta L \leq 6$	$\Delta L \leq 6$
Color mis-registration	≤ 0.4 pixel	≤ 0.7 pixel	≤ 1.0 pixel
Distortion	$\leq 1.5\%$	$\leq 2.0\%$	$\leq 5.0\%$

Tabela 2.2: Comparação entre os métodos por Wueller, D. ([8]).

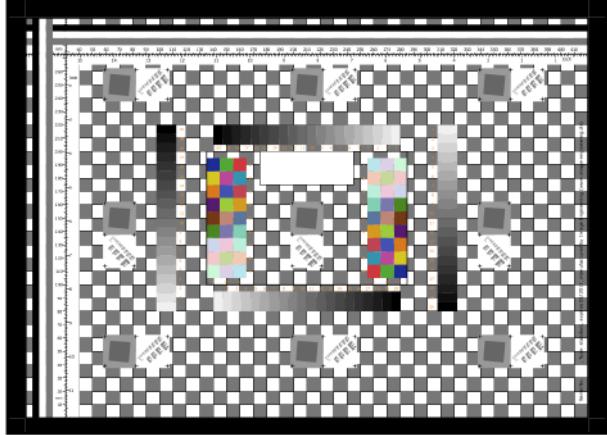


Figura 2.15: Imagem de testes

$$Q = \frac{\sigma_{xy}}{\sigma_x \sigma_y} \times \frac{2\bar{x}\bar{y}}{\bar{x}^2 + \bar{y}^2} \times \frac{2\sigma_x \sigma_y}{\sigma_x^2 + \sigma_y^2} \quad (2.10)$$

Onde:

$$\bar{x} = \frac{1}{N} \sum_{i=1}^N x_i \quad (2.11)$$

$$\bar{y} = \frac{1}{N} \sum_{i=1}^N y_i \quad (2.12)$$

$$\sigma_x^2 = \frac{1}{N-1} \sum_{i=1}^N (x_i - \bar{x})^2 \quad (2.13)$$

$$\sigma_y^2 = \frac{1}{N-1} \sum_{i=1}^N (y_i - \bar{y})^2 \quad (2.14)$$

$$\sigma_{xy} = \frac{1}{N-1} \sum_{i=1}^N (x_i - \bar{x})(y_i - \bar{y}) \quad (2.15)$$

Aplicando este índice nas imagens da figura 2.16, obtém-se a tabela 2.3, com os fatores de qualidade Q e o índice de erro médio quadrático (MSE).

Outro artigo interessante para nosso estudo é o de Ahmet M. Eskicioglu et al ([39]) nos traz um revisão de vários índices de teste de imagens já utilizados com certa eficiência. Apesar dos índices serem aplicados basicamente a imagens em tons de cinza, é um artigo com informações bem valiosas para nosso estudo, pois nos mostra

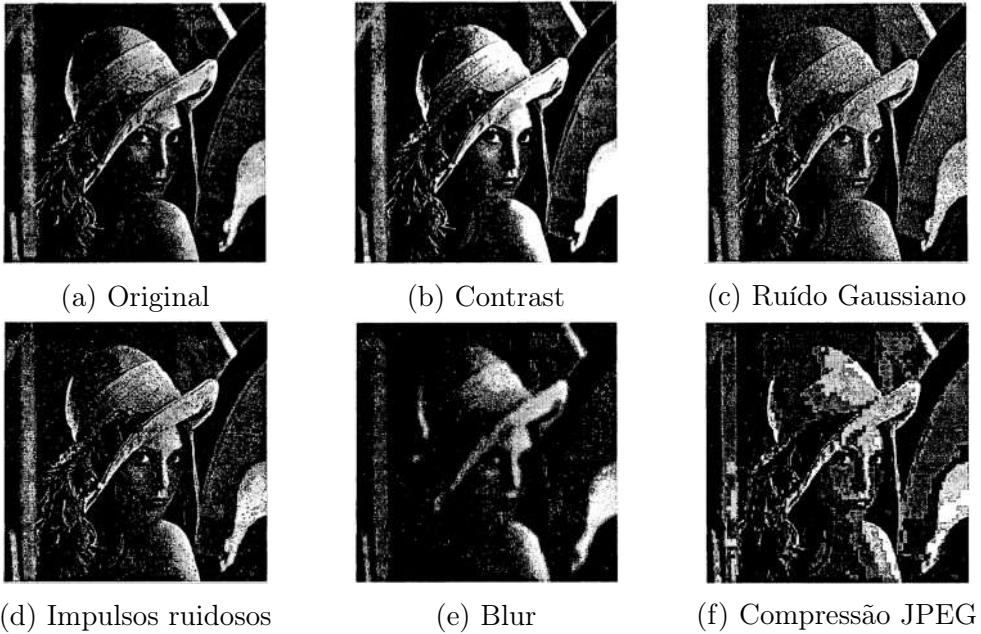


Figura 2.16: Figura utilizada para a medição de qualidade - ([6])

a maioria dos métodos já desenvolvidos para este propósito. No artigo temos ainda a subdivisão dos métodos em duas grandes famílias, o teste subjetivo e o qualitativo. O primeiro é realizado por pessoas, através de comparações entre a imagem original e a restaurada ou através de uma simples imagem restaurada, já o qualitativo são índices calculados com ajuda da matemática e da estatística, onde obtemos valores numéricos para análise. Em outro artigo do mesmo autor ([40]), ele nos traz a performance de alguns dos métodos descritos no primeiro artigo, analisando cada um deles. Para ele, é difícil descrever ou analisar a performance de um conjunto uma vez que obtém-se um número escalar, sem uma grandeza física definida e por isso a dificuldade em comparar os métodos. Nesse estudo ele analisou os seguintes métodos: *Average Difference, Structural Content, Normalize of Cross-Correlation, Correlation Quality, Maximum Difference, Image Fidelity, Weighted Distance, Laplacian Mean Square Error, Peak Mean Square Error, Normalize Absolute Error, Normalize Mean Square Error, L_p -norm, Hosaka Plot, Histogram.*

Figura	MSE	Q
Original	—	—
Contraste	225	0.9372
Ruído Gaussiano	225	0.3891
Impulsos	225	0.6494
Blur	225	0.3461
Compressão JPEG	215	0.2876

Tabela 2.3: Comparação entre os métodos por Wang, Z. e Bovik, A. C. ([6]).

Outro índice de qualidade que pode ser utilizado é a entropia de Shannon ([41]).

2.5 Métodos de inteligência artificial

No intuito de analisar outros métodos e compara-los com o método proposto foram estudados três métodos que utilizam inteligência artificial para fazer a correção das imagens.

O primeiro deles utiliza uma *Deep Convolutional Network* para realizar a correção através de curvas pré estabelecidas dentro da rede. O autor denominou de Zero Deep Curve Estimation, onde a rede faz a a correção dos níveis de contraste através de uma curva otimizada ([42]). Este método não necessita de pares de imagens com baixa iluminação/normal para aprender.

O segundo emprega O método Retinex, que faz a decomposição da imagem em reflectância e iluminação, aplicando rede neuronal. O autor denomina este método de RetinexNet ([43]). Este método necessita de um par de imagem com baixa iluminação e iluminação normal para o aprendizado.

O terceiro método chamado de *Multi-Branch Low-Light Enhancement Network* ou apenas MBLLEN ([44]), consiste em utilizar várias sub-redes neurais realizando diferentes processos, tais como: redução de ruídos, melhoria de contraste. Ao final são combinadas as imagens resultantes em uma única imagem.

2.6 Escolha do sistema embarcado

O Raspberry PI é uma arquitetura baseada em processadores ARM que consegue poder computacional em uma placa um pouco maior que um cartão de crédito, com várias portas de comunicação e uma porta para uma câmera digital. A versão do sistema 3B+ foi a escolhida por sua compatibilidade de hardware com o sistema operacional escolhido.

Chegando ao fim da revisão bibliográfica, onde fica claro que os pontos onde podemos implementar um diferencial de nosso método, são eles: aplicar a ressonância estocástica como método básico de melhoramento, associado a outros e uma intervenção na maneira de como melhorar um índice de qualidade que possa trazer ao conjunto confiabilidade suficiente para deixá-lo automático.

Capítulo 3

Método Proposto

O método proposto consiste de um conjunto de processos agregados e organizados de forma a obter a melhoria final de uma imagem e posterior visualização de objetos que estão ocultos nas regiões de baixo níveis de contraste. A informação de entrada do método pode estar em um desses formatos: RAW, JPEG ou PNG, um vídeo em MP4 ou ainda via captura direta da câmera do Raspberry PI ou de uma webcam.

Apesar de atualmente a maioria do processamento de imagens utilizar sistemas com várias *GPU*, não utilizamos neste trabalho para otimizar o processo, para que ele possa ser embarcado em hardware mais modestos. Com essa ideia em mente, seguimos com o projeto para rodar em *CPU* apenas. Com isso nosso método fica de fácil implementação em qualquer plataforma.

3.1 Método

Este método foi inicialmente idealizado para pós processamento da imagem, mas com aprofundamento das pesquisas, notou-se que era possível utilizar-lo de forma a ser embarcado em um hardware modesto, como um Raspberry PI, de maneira totalmente automata.

Para dar início ao processo, adquire uma imagem através de um dos seguintes meios: vídeo capturado através de uma webcam ou celular ou uma foto. Para o processamento desta imagem, vamos utilizar a ressonância estocástica associado com a correção por histograma, e ao final aplicar um filtro para eliminar o ruído ainda existente..

O processo inicializa pela leitura da imagem, caso a entrada seja um vídeo ela representará um quadro (*frame*) deste vídeo. Caso a imagem esteja no formato RAW, será convertida para RGB através do processo de demosaice, e caso seja

Algorithm 1 Algoritmo completo

Input: *selectfilter* in { Gaussian, Blur, Median, Bilateral }

```
1: repeat
2:   hasimage  $\leftarrow$  true
3:   if hascamera then
4:     img  $\leftarrow$  videocap()
5:   else
6:     img  $\leftarrow$  imread()
7:   end if
8:   if empty( img ) then
9:     hasimage  $\leftarrow$  false
10:  else
11:    if img is RAW then
12:      img  $\leftarrow$  demosaice(img)
13:    end if
14:    image  $\leftarrow$  toHSV( img )
15:    channelV  $\leftarrow$  getChannel ( image, V )
16:    channelV  $\leftarrow$  SR(channelV)
17:    if channelV  $\neq \{-1\}$  then
18:      channelV  $\leftarrow$  histEnhancement( channelV )
19:      image  $\leftarrow$  restoreChannel ( image , channelV, V)
20:      image  $\leftarrow$  toRGB( image )
21:      image  $\leftarrow$  runFilter(selectfilter)
22:      showImage( image )
23:    end if
24:  end if
25: until hasimage
```

uma imagem tipo JPEG ou PNG, o processo de demosaice não será necessário. O algoritmo (1) demonstra a sequência completa do ciclo de processamento. O algoritmo 2, demonstra o cálculo da ressonância estocástica propriamente dita.

Após a leitura da imagem e sua identificação, a próxima etapa no processo é a conversão do espaço de cor RGB para o espaço de cor HSV (figura: 3.1). Essa conversão é necessária para facilitar o processamento, onde somente será utilizado para o processo o canal *V* que representa o valor de luminância. Com a imagem convertida para HSV, é extraído o canal *V* e passamos a tratar exclusivamente este canal e a ação a ser tomada é de realizar a correção pela ressonância estocástica, que será feita enquanto o fator de qualidade não for adequado.

Para a condição de parada do método da ressonância estocástica foi criado um fator de qualidade complexo que contém a razão entre os fatores de qualidade da imagem atual e da imagem anterior, com a verificação da variação da média e do desvio padrão do passo atual com o passo anterior e o limite de iterações. Esses testes são realizados a cada iteração do processo da ressonância estocástica.

Algorithm 2 Função SR

Input: $channel_V$

```
1:  $width \leftarrow channel_V.cols$ 
2:  $height \leftarrow channel_V.rows$ 
3:  $avg \leftarrow getAVG(channel_V)$ 
4:  $stddev \leftarrow getStdDev(channel_V)$ 
5:  $X_0 \leftarrow \{0\}$ 
6:  $factorQ_0 \leftarrow 65535.0$ 
7: repeat
8:    $lastAVG \leftarrow avg$ 
9:    $lastSTD \leftarrow stddev$ 
10:   $factorQ_1 \leftarrow factorQ_0$ 
11:  for  $h = 0$  to  $width$  do
12:    for  $w = 0$  to  $height$  do
13:       $X_{(h,w)} \leftarrow X_{0(h,w)} + \Delta_t[a * X_{0(h,w)} - b * X_{0(h,w)}^3 + channel_V(h,w)]$ 
14:    end for
15:  end for
16:   $avg \leftarrow getAVG(X)$ 
17:   $stddev \leftarrow getStdDev(X)$ 
18:   $factorQ_0 \leftarrow getFactor(X)$ 
19:   $X_0 \leftarrow X$ 
20:   $count \leftarrow count + 1$ 
21:   $status \leftarrow factorTest(factorQ_0, factorQ_1, avg, stddev, lastAVG, lastSTD, count)$ 
22: until  $status == OK$  OR  $status == FAIL$ 
23: if  $status == FAIL$  then
24:    $X \leftarrow \{-1\}$ 
25: end if
26: return  $X$ 
```

Ao término da ressonância estocástica há um ajuste realizado pela correção de histograma no canal V e em seguida este canal é reagrupado na imagem em HSV. Na sequência converte-se a imagem para RGB e termina aplicando-se o filtro escolhido, o padrão é o filtro gaussiano.

3.2 Conjunto de imagens

O conjunto de 10 imagens foi criado para este trabalho e a tabela 3.3 mostra as características das imagens e qual equipamento foi utilizado para aquisição. As imagens originais estão apresentadas na figura 3.5. Além deste conjunto de imagens, foi utilizado o conjunto utilizado do artigo do Zhang ([45]).

Para facilitar a leitura neste texto, utilizamos a simplificação dos parâmetros utilizados na ressonância estocástica constante na tabela 3.2. No capítulo 4, temos outra tabela para codificação dos arquivos:



Figura 3.1: Representação do espaço de cor HSV

Imagen	Largura	Altura	Máquina	Formato Original	Fonte
Imagen 1 (3.4a)	1728	3072	S7	JPEG	Foto
Imagen 2 (3.4b)	1920	1080	S7	MP4	Vídeo
Imagen 3 (3.4c)	1920	1080	D90	RAW	Foto
Imagen 4 (3.4d)	1920	1080	D90	RAW	Foto
Imagen 5 (3.4e)	4992	3292	DFr	JPEG	Foto
Imagen 6 (3.4e)	4992	3292	DF	JPEG	Foto
Imagen 7 (3.4g)	1920	1080	S7	MP4	Vídeo
Imagen 8 (3.4h)	1920	1080	S7	MP4	Vídeo
Imagen 9 (3.4i)	1920	1080	S7	MP4	Vídeo
Imagen 10 (3.4j)	640	480	LT 270HD	MP4	Vídeo

Tabela 3.1: Informações sobre as fotos utilizadas neste trabalho

As imagens foram obtidas através das máquinas fotográficas Nikon D90 com configurações manuais de abertura, tempo de exposição e a simulação de ISO do filme para obtermos uma coleção de imagens variadas de uma mesma cena, um celular Samsung S7 e a webcam USB Logitech modelo 270 HD, esses últimos no modo automático. As imagens da procissão de Paraty (figuras 3.4e e 3.4f) foram cedidas pelo fotógrafo João Rocha ([46]) para uso neste trabalho. Ele utilizou uma Nikon DF com configurações manuais para obter essas fotos.

O conjunto de imagens listadas na tabela 3.3, estão disponíveis no **Google Drive** no link que se encontra no anexo A.1, para melhor visibilidade e caso se deseje utiliza-las em outras situações.

Dentre essas imagens, destacamos a lista apresentada na tabela 3.3 para a comparação entre o método proposto e os métodos Zero-Reference Deep Curve Estimation ([42]), Deep Retinex Decomposition for Low-Light Enhacement ([43]) e MBLLEN: Low-light Image/Video Enhacement Using CNNs ([44]). Além desses métodos utilizando inteligência artificial, outros forma estudados, como: EnlightenGAN: Deep Light Enhancement without Paired Supervision ([47]), Underexposed Photo Enhancement Using Deep Illumination Estimation ([48]) e Kindling the Darkness: A Practical Low-light Image Enhancer ([45]).

Parâmetros	Δ_t	Divisor
P1	0.01	10^3
P2	0.01	10^4
P3	0.01	10^{15}
P4	0.01	10^{16}
P5	0.001	10^3
P6	0.001	10^4
P7	0.001	10^{15}
P8	0.001	10^{16}

Tabela 3.2: Codificação dos parâmetros utilizados

Arquivo	Conteúdo
imagem01	Meninas 3.4a
imagem05	Procissão em Paraty 3.4e
imagem06	Procissão em Paraty 3.4f
01	Torre Eiffel 3.5a
02	Parque 3.5b
03	Quadro na parede 3.5c
04	Igreja 3.5d
05	Por do Sol Montanhas 3.5e
07	Telhado 3.5f
16	Fundo do mar 3.5g
19	Satélite 3.5h
44	Por do sol Monumento 3.5i
47	Vaso de flores 3.5j
60	Plantação de bambu 3.5k

Tabela 3.3: Lista de imagens utilizadas na comparação deste método com os métodos de IA

3.3 Conversão de formato

As cameras Nikon e o celular Samsung geram a imagem no formato RAW, porém é um formato proprietário. Neste caso se faz necessário a utilização de um programa externo, chamado dcraw ([15]), para realizar a tarefa de conseguir ler e extrair a imagem de dentro do formato proprietário, sem nenhum tipo de interferência. Para isso deve-se utilizar a opção *-E* que está omitida da listagem das opções válidas e só foi identificada com a leitura e estudo minucioso do código do programa. Essa opção além de aumentar o nível de não processamento da imagem, força o uso das opções *-d -D -j* automaticamente como padrão, que fazem respectivamente: extrair sem cor e interpolação; sem escalas; e sem alterações nos *pixels* RAW.

Esse primeiro passo é importante, pois preservamos a imagem crua e sem nenhum retoque, que possibilitará a aplicação do método em estudo, sem a interferência indesejada de algum algoritmo de correção automática do equipamento de captura.

Como um dos objetivos é a automatização do processo esta etapa da extração e conversão só é utilizada para pós processamento e não estará disponível quando for embarcado no Raspberry PI.

3.4 Reconstrução dos planos de cor

Esta etapa do método, também chamada de demosaice, consiste em remontar os planos de cores da imagem para obter os valores corretos de cada pixel da imagem. A escolha da biblioteca OpenCV ([20]) para realizar essa reconstrução foi feita por possuir um algorítimo rápido e eficiente para este processo e também é considerado o *estado da arte*.

Esta etapa só estará disponibilizada para pós processamento pelo simples motivo de que não será necessário quando embarcado.

3.5 Ressonância estocástica

O processo da ressonância estocástica será utilizado para o melhoramento dos níveis de contrastes como também para a diminuição do nível de ruído da imagem. O que se pretende fazer nesta fase é associar algum outro método que realce esta melhoria, e aplicando filtros para que se verifique a melhor resposta possível. A equação 3.1 mostra a equação modificada para a ressonância estocástica que será utilizada. A modificação dá-se pela alteração do ruído a ser injetado, que neste caso será o da própria imagem.

Neste ponto, as tentativas iniciais do uso da ressonância estocástica foram frustrantes, porque não se conseguia chegar a um bom resultado. Foi tentado reproduzir os artigos estudados, mas as imagens que foram utilizadas não foram encontradas. Muito estudo foi necessário para a compreensão dos valores das variáveis a e b da equação da ressonância estocástica. A atenção especial foi dada a variável b já que ela é calculada através da equação 3.3, onde aplicamos um divisor D variando entre 10^3 até 10^{16} , como podemos observar na equação 3.4.

O parâmetro Δ_t teve seu valor variado de 0.001 até 0.1 em passos de múltiplos de 10, com isso geramos inúmeras imagens a serem analisadas visualmente. Essa análise foi necessária para verificar a evolução de cada passo do processo e analisar os dados da média, desvio padrão e o fator F , que é a razão entre a qualidade da imagem anterior em relação a imagem atual. Com essa análise pode-se verificar que o comportamento exponencial da média e do desvio padrão demonstrava o melhoramento da imagem quando atingiam um comportamento exponencial, sendo este um ponto de parada.

Outro fator de parada, a variável F , foi analisada e verificou-se dois comportamentos distintos, a saber:

- O decréscimo deste valor - A variável F tende a 0 (zero), quando mais próximo deste valor melhor a imagem, mas foi estimado um valor limite de 0.8. Mesmo abaixo deste valor não há uma melhoria significativa da imagem;
- A inversão da tendência de queda - Quando a variável F volta a subir, significa que a imagem teve piora, então para neste ponto.

Para cada combinação de Δ_t e o divisor D , foram necessários vários passos, e cada passo gerando uma imagens. Em média os bons resultados foram obtidos com 10 iterações. O limite imposto para considerar uma imagem sem possibilidades de correção foi de 200 iterações. Através das análises visuais, foi verificado que 50 iterações são o suficiente para que o processo termine sem que a imagem seja melhorada.

Foram geradas mais de 5000 imagens para serem analisadas visualmente, através de histogramas e o comportamento das variáveis envolvidas, demandando tempo de processamento para gerar as imagens e tempo para visualizar, analisar e comparar os resultados.

$$X_{(i,j)(n+1)} = X_{(i,j)(n)} + \Delta t[aX_{(i,j)(n)} - bX_{(i,j)(n)}^3 + V(i, j)] \quad (3.1)$$

$$i \leftarrow 0, 1, \dots MAX_COLS$$

$$j \leftarrow 0, 1, \dots MAX_ROWS$$

$$a = 2\sigma_0^2 \quad (3.2)$$

$$b < \frac{4a^3}{27} \quad (3.3)$$

$$b < \frac{4a^3}{27 * D} \quad (3.4)$$

onde:

$X_{(n)}$ \leftarrow Imagem inicial/anterior

$X_{(n+1)}$ \leftarrow Nova imagem

Δ_t \leftarrow Variação temporal

a, b \leftarrow Fatores calculados através da imagem $X_{(n)}$

V \leftarrow Ruído da própria imagem

A razão dos fatores de qualidade varia na direção do valor 0 (zero) e é considerado

que a imagem chegou em sua máxima correção quando a tendência deste fator volte a crescer, em comparação com o fator calculado para a imagem anterior, ou caso o cálculo da média e do desvio padrão do passo atual varie exponencialmente em relação ao anterior. Para o próximo passo do algoritmo consideramos a melhor imagem a referente a iteração anterior ($n - 1$).

3.6 Fator de parada

Este ponto é o mais sensível de todo processo, porque ficou difícil de medir a melhoria através de um único método confiável dentre os descritos na seção de trabalhos relacionados, seja por demanda computacional alta ou pela não tradução quantitativa da melhoria da imagem.

Analizando os resultados visualmente e os dados numéricos de cada imagem, foi possível criar um índice de qualidade próprio para que fosse utilizado como um dos fatores de parada do método.

3.6.1 Índice de qualidade

Analizando os dados numéricos das imagens resultantes, foi verificado que o comportamento exponencial da média e do desvio padrão indicavam que a imagem obteve melhoria significativa. Isso significa que na média, todos os *pixels* da imagem aumentaram de valor e a diferença entre eles teve o mesmo comportamento. O fator de qualidade Q é calculado pela razão entre a média e o desvio padrão 3.5

$$Q_n = \frac{\bar{I}_n}{\sigma_n^2} \quad (3.5)$$

Onde:

- | | | |
|--------------|--------------|---------------------------------|
| Q_n | \leftarrow | Qualidade da n-ésima imagem |
| \bar{I}_n | \leftarrow | Média da n-ésima imagem |
| σ_n^2 | \leftarrow | Desvio padrão da n-ésima imagem |

Nos gráficos das figuras 3.2 e 3.3 mostram o comportamento da média, do desvio padrão e do fator F , para as imagens 3.4a e 3.4e.

Podemos observar nos gráficos que ambas imagens tiveram comportamento semelhante. No primeiro gráfico a média (azul) tem um comportamento crescente, mas o desvio padrão (vermelho) decresce indicando que a imagem não terá bons resultados. No segundo gráfico, o comportamento de ambas variáveis tem o comportamento crescente e exponencial, indicando que a imagem obteve sucesso na sua melhoria.

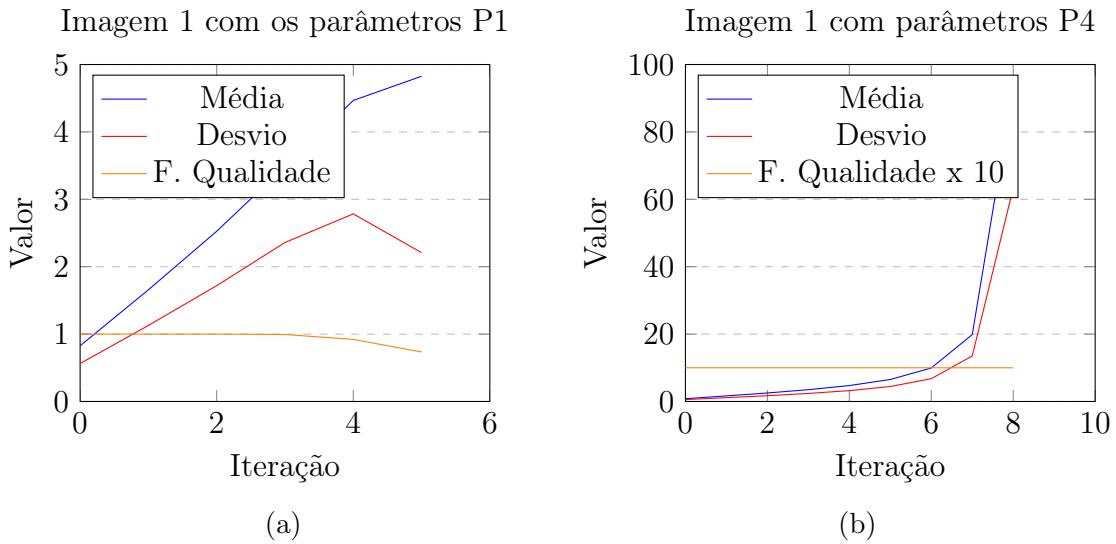


Figura 3.2: Imagem 1 - Comportamento da média, desvio padrão e do fator F

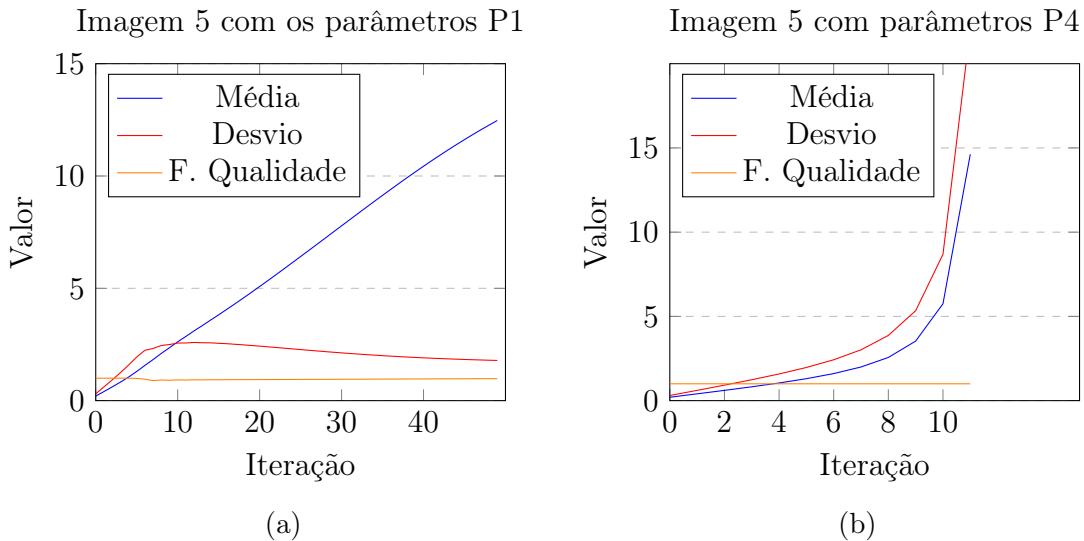


Figura 3.3: Imagem 5 - Comportamento da média, desvio padrão e do fator F

3.6.2 Fator F

Este fator é o resultado do cálculo da razão dos índices de qualidade do passo atual e passo anterior, como pode ser visto na equação 3.6. O comportamento do fator F (amarelo) é decrescente em direção ao valor 0, porém através das análises, o limite de 0.8 é o suficiente para saber se a imagem foi ou não corrigida. Sozinho diz apenas que a imagem chegou a um limite e caso não tenha ocorrido o salto exponencial nos valores da média e do desvio padrão a imagem não atingiu os indicadores necessários para ser considerada melhorada.

$$F = \frac{Q_n}{Q_{(n-1)}} \quad (3.6)$$

Onde:

$$\begin{aligned} F &\leftarrow \text{Fator de qualidade} \\ Q_n &\leftarrow \text{Qualidade da } n\text{-ésima imagem} \\ Q_{(n-1)} &\leftarrow \text{Qualidade da imagem anterior} \end{aligned}$$

3.7 Redução de ruídos

A ressonância estocástica já realiza o principal trabalho na redução de ruídos, pelo incremento do sinal paulatinamente, fazendo-o pular de um nível para o outro no sistema biestável, transformando-o em um sinal detectável. Mas só este processo não assegura uma filtragem eficiente. Para auxiliar a diminuição dos ruídos, após a ressonância estocástica, aplicamos um dos seguintes filtros: o Gaussiano, Bilateral, Blur e de Média. Os resultados serão analisados no capítulo 4.

3.8 Ambiente de teste

Para o ambiente de desenvolvimento e de testes, as máquinas utilizadas foram: um computador *desktop* com um processador AMD FX8300 @ 3,3GHz ([49]) com 8Gbytes de RAM, placa de vídeo *on board* Radeon HD 3000, com memória compartilhada de 1Gbyte e um Raspberry PI 3B+ ([50]) com processador Broadcom BCM2837B0, Cortex-A53 64-bit SoC @ 1.4 GHz com 1Gbyte de RAM. Os resultados finais foram processados em ambos ambientes de teste e no Raspberry PI, utilizamos uma câmera USB da Logitech, modelo 270 HD, com resolução máxima de 1280 x 720 pixels. A resolução adotada para os teste foi de 640x480 pixels.

Todo o ecossistema de testes é baseado no sistema operacional FreeBSD ([16]) que está disponível em www.freebsd.org para uso. Este sistema é totalmente livre e de código aberto, possui o compilador CLANG para compilação de códigos C/C++ como padrão. Este compilador já tem suporte nativo ao padrão C++17 que será utilizado em nosso estudo.

Para poder compilar o projeto é necessário a instalação das dependências, que são:

- Opencv - Esta biblioteca fará todo o trabalho de manipulação básica da imagem, como carregar a imagem, armazenar a imagem e a sua visualização;
- SQLite3 - Esta biblioteca irá auxiliar no armazenamento dos dados de cada passo da ressonância estocástica para posterior avaliação. Os dados armazenados são: coeficientes a e b , variáveis Q_0 , F , média, desvio padrão e o tempo levado para realizar a operação da ressonância estocástica;
- bmake - Ferramenta de manipulação de arquivos *Makefile*, utilizando a sintaxe

BSD. Esta ferramenta só será necessário se for utilizado em ambientes Linux.

Todo o código ficará disponível no GitLab (www.gitlab.com) ([51]) podendo ser *clonado* através do comando: `git clone git@gitlab.com:ufrrj/tese.git`.

Como este projeto foi desenvolvido para ser compilado em qualquer sistema *Unix-Like*, como Linux ou outro BSD, ele usa como interface gráfica o servidor gráfico X Window Xorg ([18]), que é o padrão nesses sistemas. Para ser executado em sistemas baseados em *Microsoft Windows* deverá sofrer algumas alterações que deverão ser implementadas por quem for utilizar. Será necessário a instalação do Visual Studio, compilador padrão da empresa MicroSoft. Não foi testado neste último sistema operacional pois não é utilizado nos ambientes de testes.

As fotos originais encontram-se no conjunto de figuras 3.5, de forma reduzida e no anexo A.1 encontra-se o endereço para realizar a cópia, caso seja necessário.



Figura 3.4: Fotos originais utilizadas no trabalho



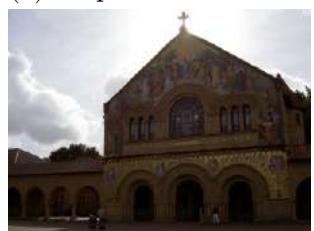
(a) Arquivo Torre Eiffel



(b) Arquivo Parque



(c) Arquivo Quadro



(d) Arquivo Igreja



(e) Arquivo Montanhas



(f) Arquivo Telhado



(g) Arquivo Fundo do Mar



(h) Arquivo Satélite



(i) Arquivo Monumento



(j) Arquivo Vaso de flores



(k) Arquivo Bambu

Figura 3.5: Fotos originais do conjunto de fotos obtidas em:
<https://paperswithcode.com/task/low-light-image-enhancement>

Capítulo 4

Resultados e Discussões

4.1 Obtenção das imagens

As imagens foram obtidas com o auxilio das câmeras Nikon D90, Nikon DF, Celular Samsung S7 e da webcam Logitech modelo 270HD. As imagens 3 e 4, foram obtidas utilizando a Nikon D90 e armazenadas em formato RAW e pré processadas pelo programa *dcraw* para a extração da imagem do formato nativo e proprietário da Nikon. Neste caso foi necessário processar essas imagens com o demosaice, para que a imagem fosse convertida para o espaço de cor RGB.

Como as imagens do tipo RAW são de difícil obtenção e por necessitar de um programa externo, optou-se por manter a estrutura do demosaice apenas para utilizar em casos onde fosse possível adquirir esse tipo de imagem de modo fácil e direto.

A escolha de utilizar imagens no formato **JPEG** ou **PNG** foi pela simples facilidade de obtenção, uma vez que são de domínio público e as câmeras já geram este tipo de imagem automaticamente (JPEG). A maioria das imagens obtidas por câmeras comuns já estão no espaço de cor RGB, inclusive as câmeras que geram vídeos de segurança e as webcams.

4.2 Nomenclatura dos arquivos

Os nomes das imagens processadas incorporam os valores de Δ_t , do divisor D e outras abreviações no nome do arquivo, como podemos ver na tabela 4.1.

Para simplificar, foram associados códigos para os nomes das imagens, para que possam ser visualizados melhor. A tabela 4.2 mostra a codificação utilizada.

A título de exemplo de codificação do nome da imagem, o arquivo *imagem01-*

codificação	Significado
DEM	Demosaice
SR	Ressonância Estocástica
DT N	Δ_t
D D	Divisor
P P	Iterações
FG	Filtros Gaussiano
FB	Filtro Blur
FBL	Filtro Bilateral
FM	Filtro Mediano
hist	Histograma

Tabela 4.1: Codificação dos arquivos processados

Composição do Nome	Código
DEM	Demosice
SR	SR
DT0.01	DT1
DT0.001	DT2
D10 ³	D1
D10 ⁴	D2
D10 ¹⁵	D3
D10 ¹⁶	D4
hist	H

Tabela 4.2: Codificação para simplificar a leitura das imagens

SR-DT0.01-D1000000000000000-P05-Gaussian-hist terá seu nome codificado para *imagem01-DT1-D4-G-H*, que representa que esta imagem foi processada pela seguinte sequência de eventos: ressonância estocástica com o $\Delta_t = 0.01$ e o divisor $D = 10^{16}$, aplicado o filtro gaussiano e em seguida a correção de histograma.

4.3 Resultados

4.3.1 Método proposto

O trabalho visa a utilização do método proposto para imagens reais, por isso, utilizamos imagens de várias fontes para que o resultado fosse o melhor possível para diferentes origens de aquisição.

Os testes para compreender o funcionamento da ressonância estocástica, demoraram um tempo além do esperado, uma vez que os artigos estudados mencionavam que o fator b na equação da ressonância estocástica (eq. 4.1) deveria ser dividido pelo valor 10^4 para maximizar a relação sinal ruído (eq. 4.4). Dessa forma, o incremento de energia da partícula seria pequeno, proporcionando um ajuste mais fino na melhoria

da imagem.

$$X_{(n)} = X_{(n-1)} + \Delta_t[aX_{(n-1)} - bX_{(n-1)}^3 + V] \quad (4.1)$$

$$a = 2\sigma_0^2 \quad (4.2)$$

$$b \ll \frac{4a^3}{27 * D} \quad (4.3)$$

onde:

- $X_{(n-1)}$ → Imagem passo anterior
- $X_{(n)}$ → Imagem atual corrigida
- Δ_t → Variação temporal, entre duas iterações
- V → é a contribuição do ruído da imagem original no sistema
- D → Divisor do fator b

Os coeficientes a e b são obtidos através das equações 4.2 e 4.3 que devem maximizar a relação sinal/ruído (equação 4.4) para melhorar a razão entre o sinal e o ruído presente nas imagens:

$$SNR = \frac{4a}{\sqrt{2}(\sigma_0\sigma_1)^2} \exp\left(-\frac{a}{2\sigma_0^2}\right) \quad (4.4)$$

onde:

- σ_0 → desvio padrão da imagem
- σ_1 → desvio padrão do ruído

Em todos os artigos estudados, a média de cinco iterações foi considerada suficiente para obter bons resultados. Para nosso conjunto de imagens, o valor da média neste patamar, não alterou a condição da imagem, mantendo-a muito escura e dando uma primeira impressão de ineficiência da ressonância estocástica para as imagens.

Contudo, a pesquisa continuou e, em novos testes, a quantidade de iterações foi alterada para 20 passando a proporcionar uma certa melhora na imagem, mas não o suficiente. Então, a quantidade de iterações foi sendo acrescida de 20, até chegar ao valor de 200, onde foi possível observar melhora em algumas das imagens, porém não como esperado e isso levou a variação do denominador de b (equação 4.3) pela multiplicação do divisor D , entre os valores 10^3 a 10^{16} .

Todos esses testes foram realizados de forma manual, visualizando a imagem resultante a cada ciclo da iteração. Com a tendência de melhora da imagem, inicialmente, a quantidade máxima de iterações foi fixada em 200 e a variação da variável Δ_t entre os valores 0.0001 e 0.1, com passos múltiplos de 10, já que este parâmetro influência diretamente o incremento da energia do pixel na nova imagem.

Nos testes realizados foi verificado que para o valor de $\Delta_t = 0.1$ a imagem tem um incremento muito rápido em cada passo, passando de um estado pré melhoria para um estado de perda de dados da imagem, inviabilizando a correção. Para o valor de $\Delta_t = 0.0001$ e $\Delta_t = 0.001$, houve imagens que demandaram as 200 iterações, e mesmo assim não chegaram a resultados satisfatórios, se compararmos a imagem original e a final. O melhor valor para Δ_t é 0.01, de acordo com que os resultados dos testes realizados indicaram. Com este valor para o Δ_t as imagens demandaram poucas iterações para obter um resultado bem interessante, porém ainda não bom o suficiente. Aplicando os filtros nas imagens corrigidas e em seguida aplicando a correção de histograma no canal V da imagem, foi possível ter um acréscimo de contraste e percepção visual bem acentuados, como podemos verificar nas imagens (figura 4.2). A imagem processada utilizou os valores $\Delta_t = 0.01$ e $D = 10^{16}$.

Para o processo da figura 4.1, a tabela 4.3 apresenta os tempos demandados para se processar as 200 iterações no *desktop*. Apesar de representar um ajuste mais fino em cada iteração, esses valores se mostram inviáveis pelo tempo necessário para a conclusão do processo, seja em um *desktop* ou em um sistema embarcado.

Analizando o histograma das duas imagens original e final, nota-se que ele ficou ainda mais comprimido e próximo a zero, isso se deve ao fato do incremento energético ser muito baixo, não sendo suficiente para ultrapassar a barreira do sistema bi-estável e alcançar o nível mais energético.

Divisor D	Tempo (s)	Divisor D	Tempo (s)
10^3	95.5768920	10^4	95.5027630
10^5	95.5585850	10^6	96.1968420
10^7	95.4601180	10^8	95.6025220
10^9	96.5098670	10^{10}	95.6894440
10^{11}	96.2910210	10^{12}	96.9593740
10^{13}	95.8980260	10^{14}	95.9215960
10^{15}	95.4984840	10^{16}	95.4513390

Tabela 4.3: Tempo para processar 200 iterações da imagem 9 com $\Delta_t = 0.001$

Além de todos esses detalhes, foi necessário utilizar uma estrutura de dados em ponto flutuante de 64bits (*double*) para as imagens, para não gerar erros de aproximações que pudessem afetar os cálculos, uma vez que o valor máximo de cor para um dado

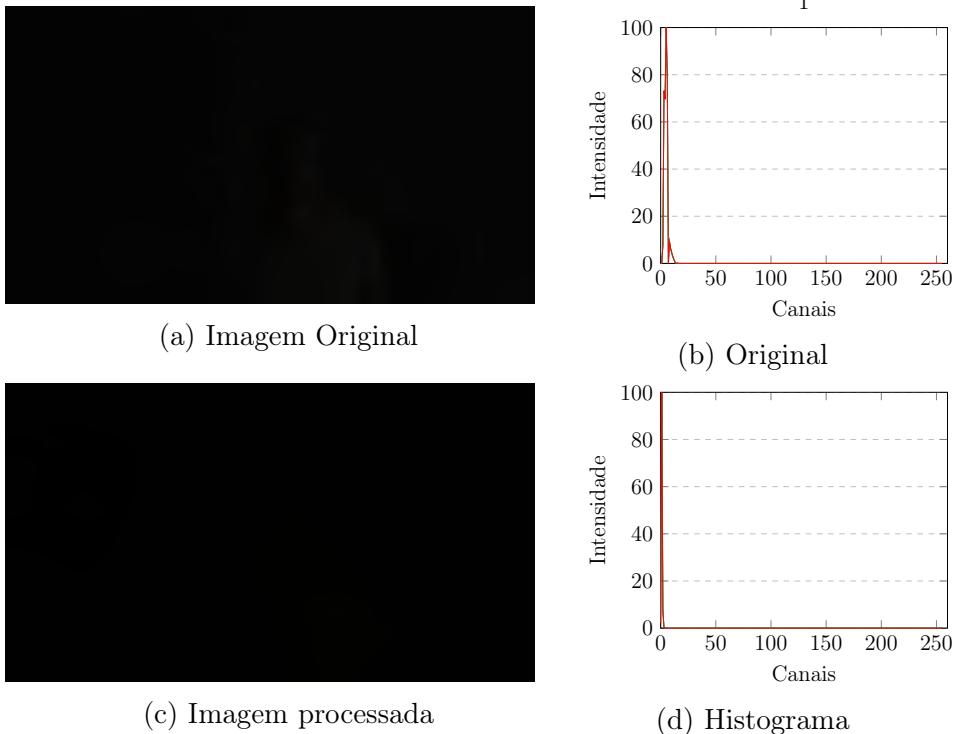


Figura 4.1: Imagem imagem 9 processada em 200 ciclos da Ressonância Estocástica

pixel é de 255 por canal. A divisão do valor do pixel por um valor da ordem de 10^{16} , necessita de uma quantidade de bits suficientemente grande para ser representado.

As dez imagens criadas para serem utilizadas neste trabalho, geraram mais de 2000 imagens para serem analisadas. Das imagens analisadas foram selecionadas as imagens processadas com valores para o divisor D iguais a: 10^3 , 10^4 , 10^{15} e 10^{16} , os valores intermediários, ficaram com resultados próximos entre si e nos extremos é possível ver a diferença entre as imagens. As imagens com o divisor D abaixo de 10^{15} não tiveram um resultado considerado satisfatório, sendo o melhor resultado para os valores de $D = 10^{15}$ ou $D = 10^{16}$.

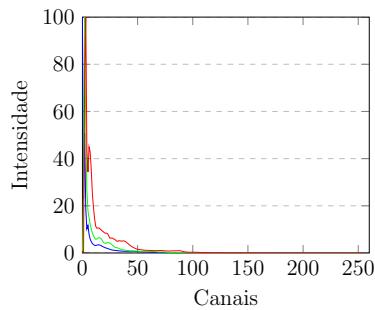
4.3.2 Tempo de processamento

A tabela 4.4 compara tempo gasto em cada uma das plataformas utilizadas no teste para processar a ressonância estocástica. A resposta apresentada pelo sistema do Raspberry PI demonstram ser bem mais expressivas quanto ao custo de tempo quando comparado ao *desktop* utilizado. Há vários fatores que podem ter levado ao tempo elevado, como por exemplo, o fato da necessidade de realizar a função de *swap* com mais frequência, uma vez que o sistema tem pouca memória física. Para imagens grandes como as imagens *imagem05* e a *imagem06* o tempo foi bem alto.

O cômputo do tempo se refere apenas ao processamento pela ressonância estocástica,



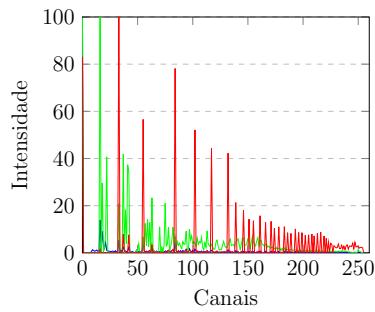
(a) Original



(b) Histograma original



(c) Imagem processada



(d) Histograma Final

Figura 4.2: Imagem 5 - processada com os parâmetros P1 + FG + H

já que os outros tempos não sofrem muita influência pela diferença entre as plataformas, uma vez que é realizado um único processamento pelo filtro e pela correção de histograma, aplicada apenas no canal V , lembrando que todo o processamento é feito no espaço de cor HSV .

Apesar do Raspberry PI ter um tempo bem mais alto do que o *desktop*, para a imagem de número 10, que é o resultado da captura pela webcam, os valores para o processamento são bem aceitáveis. Em geral, a quantidade de iluminação é o que determina o número de iterações no processo da ressonância estocástica.

Imagen	It.	Tempo(s)		Imagen	It.	Tempo(s)	
		Desktop	RPI			Desktop	RPI
01	5	2.295	5.312	06	49	57.532	142.642
	6	2.690	6.186		49	57.858	306.491
	8	3.436	8.071		10	12.763	136.753
	8	3.438	8.325		10	12.815	97.498
02	49	7.468	17.227	07	49	7.443	17.523
	9	1.496	3.466		49	7.422	17.661
	9	1.491	3.425		30	4.586	10.743
	9	1.503	3.481		30	4.590	10.917
03	49	36.750	86.305	08	49	7.407	17.712
	49	36.800	87.838		49	7.405	17.639
	17	13.189	30.670		16	2.517	5.834
	17	13.366	230.706		16	2.514	5.956
04	49	36.895	86.319	09	49	7.422	17.409
	49	36.608	85.545		49	7.429	17.829
	8	6.601	15.711		49	7.408	17.226
	8	6.615	98.976		49	7.390	17.443
05	49	57.578	138.571	10	49	1.107	2.543
	49	57.903	513.366		10	0.244	0.565
	11	13.838	127.304		10	0.243	0.559
	11	13.876	216.118		10	0.247	0.564

Tabela 4.4: Tempo Total gasto

4.3.3 Condição de parada

Para a implementação da condição de parada, foram pesquisadas algumas soluções e adotou-se 3 condições: a quantidade de iterações máxima, a razão entre a qualidade da imagem anterior e a imagem atual, gerando o fator F e a observação do comportamento da média e do desvio padrão, sendo esperado uma subida exponencial como indicativo de melhora da imagem.

O fator F tende ao valor zero, mas através de verificações subjetiva no controle da qualidade da imagem, foi verificado que valores abaixo de 0.8 não alteram a percepção visual da imagem, assim, este valor foi adotado como limiar para o fator F . Outro comportamento do fator F que deve ser observado é a subida após sucessivos declínios, que leva a parada do método. Esta mudança no comportamento de F , indica que a imagem foi comprometida e não obterá um bom resultado.

A condição de parada foi implementada com sucesso, permitindo que as imagens possam ser processadas pela quantidade de iterações ideal sem que haja perda de informações da imagem ou acréscimo de artefatos. Nos gráficos dos fatores que compõem a condição de parada, podemos analisar bem o comportamento evolutivo do processo. Nos gráficos, a curva azul corresponde a média dos *pixels* da imagem, a curva verde o desvio padrão e a curva vermelha o fator F . No gráfico referente a imagem 1 (figura B.1a), vemos que em 5 interações o fator F chegou a um valor inferior a 0.8 e com isso a ressonância estocástica encerrou o processamento. Nos gráficos das figuras B.1c e B.1d tanto a média quanto o desvio padrão subiram exponencialmente, causando a parada do processo. Apenas a imagem 9 não teve condições de uma melhoria com a ressonância estocástica com nenhum valor das variáveis Δ_t e D , visto que a imagem não tinha nenhuma iluminação direta e a cena era bem escura, obtendo uma imagem final bem ruidosa.

Os resultados dos processamentos no conjunto de imagens criado para este método estão no anexo C. Ela foram processadas com o divisor D assumindo os seguintes valores: 10^3 , 10^4 , 10^{15} e 10^{16} . As imagens estão com seus respectivos histogramas para que possam ser analisados.

Neste conjunto podemos verificar que a imagem 09 (C.9e, C.9f, C.9g C.9h) que apesar de ter uma visualização e algumas formas, a ressonância estocástica não foi eficiente em reduzir o ruído e/ou melhorar a imagem de uma forma geral, comparado as outras imagens do conjunto. Podemos visualizar em seus histogramas que apenas 3 cores a compõem. No geral os melhores resultados forma obtidos com o D assumindo valores acima de 10^{14} .

4.3.4 Comparação com métodos a inteligência artificial

Utilizando o conjunto de imagens listado na tabela 3.3, realizou-se comparações com os métodos que utilizam inteligência artificial. Para este processo de comparação utilizou-se os códigos em linguagem de programação *python* dos respectivos autores, sem nenhuma modificação, justamente para comparar a qualidade dos resultados. As figuras 4.3, 4.4, 4.5, 4.6, 4.7, 4.8, 4.9, 4.10, 4.7, 4.12, 4.13, 4.14, 4.15 e 4.16 representam os resultados dos testes na seguinte ordem: imagem original, imagens processadas pelos métodos Zero-DCE ([42]), RetinexNet ([43]), MBLLEN ([44]) e nosso método.

Analizando os resultados obtidos pelas redes neurais, podemos verificar que as imagens geradas atingiram resultado muito expressivo. E em algumas imagens, podemos verificar resultados melhores que obtidos com nosso método, como por exemplo na foto do vaso de plantas (figura 4.12). Nosso método foi pensado para imagens escu-

ras e com baixa iluminação. Esta foto do vaso está em exposição com luz ambiente bem intensa, o que acaba prejudicando o nosso método. Outra imagem para qual o nosso método não se saiu bem foi o pôr do sol nas montanhas (figura 4.7), onde se pode observar a introdução de degraus na coloração do céu, ao invés de uma transição suave entre as cores e, também na construção abaixo do céu, não foi possível uma melhoria significativa.

No geral, nosso método foi melhor que as redes RetinexNet e MBLLEN. O método RetinexNet em geral produz imagens com aspecto desfocado e pixelado, e o MBLLEN produz imagens um pouco borradadas.



Figura 4.3: Torre Eiffel



Figura 4.4: Parque



Figura 4.5: Quadro na parede



Figura 4.6: Igreja



Figura 4.7: Pôr do sol e montanhas



Figura 4.8: Telhado

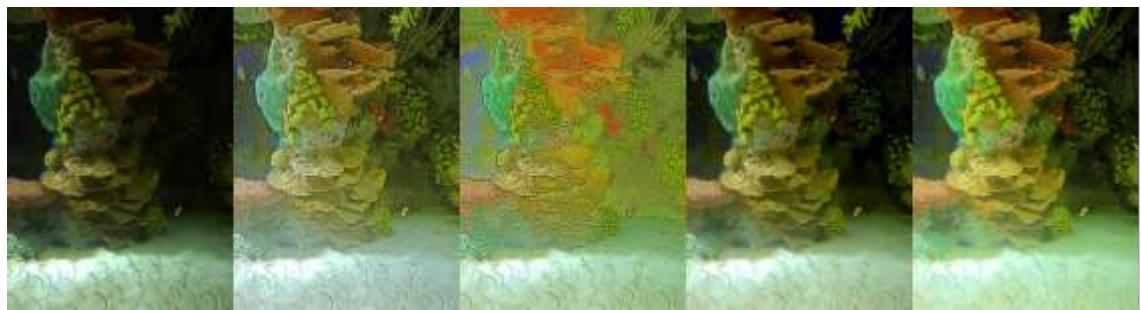


Figura 4.9: Fundo do mar



Figura 4.10: Satélite



Figura 4.11: Pôr do sol e monumento



Figura 4.12: Vaso de planta



Figura 4.13: Bambu



Figura 4.14: Meninas



Figura 4.15: Procissão em Paraty



Figura 4.16: Procissão em Paraty

Pra termos uma melhor percepção da qualidade da imagem, separamos duas pequenas seções da imagens e foi aplicado uma ampliação para uma melhor visualização. Essas imagens da seção seguem a seguinte identificação dos métodos: imagem original, Zero-DCE, RetinexNet, MBLLEN e nosso método.

Como podemos ver na imagem da Torre Eiffel (figura 4.17), o nosso método retornou a imagem sem ruído e bem nítida, comparável a obtida com o método Zero-DCE, ficando com um resultado visual acima das duas outras opções de modelo de rede neural.

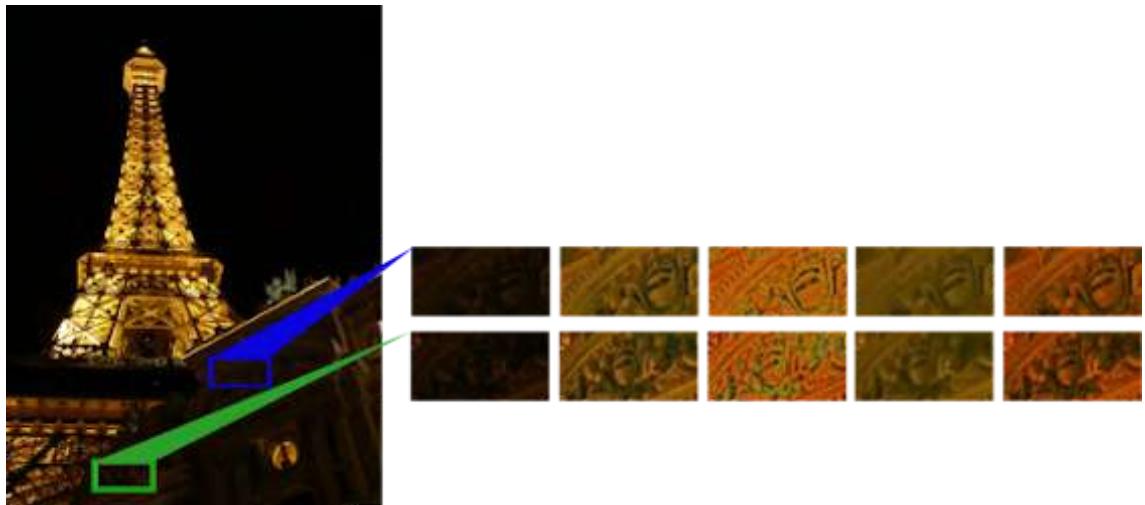


Figura 4.17: Torre Eiffel

Na imagem do parque (figura 4.18), podemos ver na primeira linha dos recortes, que o método RetinexNet também não consegue ser melhor que os demais, e isso fica bem acentuado quando avaliamos o segundo recorte, onde há um rosto que fica borrado. Novamente o nosso método fica similar aos outros dois.



Figura 4.18: Parque

Para a imagem do quadro na parede (figura 4.19), o método MBLLEN teve um resultado melhor, superando os demais. O nosso método se saiu bem trazendo um resultado melhor que o método RetinexNet.

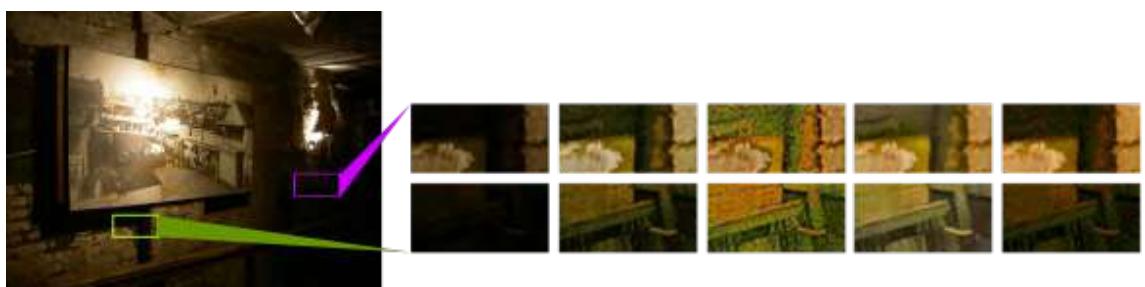


Figura 4.19: Quadro na parede

Na imagem da igreja (figura 4.20), para a fachada da igreja, nosso método teve um resultado muito bom. Já em relação à parte das portas, não teve o mesmo sucesso pelo motivo de uma grande região no topo da imagem ter um índice de luminosidade bem alto e, como foi desenvolvido para imagens com baixíssima iluminação, não teve um bom desempenho.

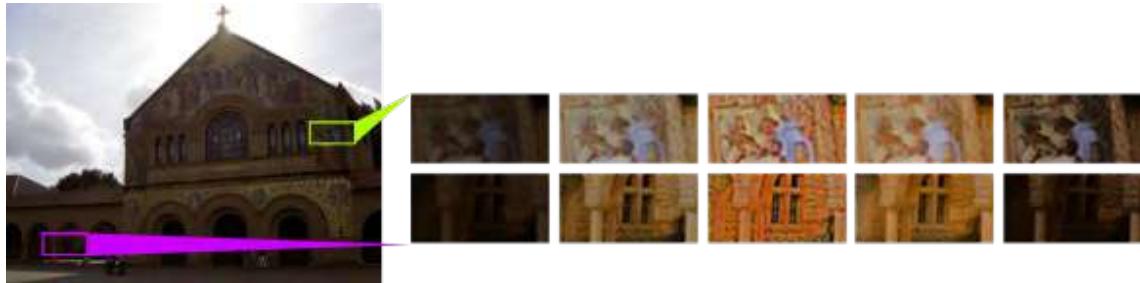


Figura 4.20: Igreja

A imagem do Pôr do Sol (figura 4.21) demonstrou que nenhum dos métodos terá bom desempenho em todos os tipos de imagem. Pela diferença de iluminação da parte mais clara com a parte mais escura, dificilmente imagens deste tipo poderão ser corrigidas automaticamente.

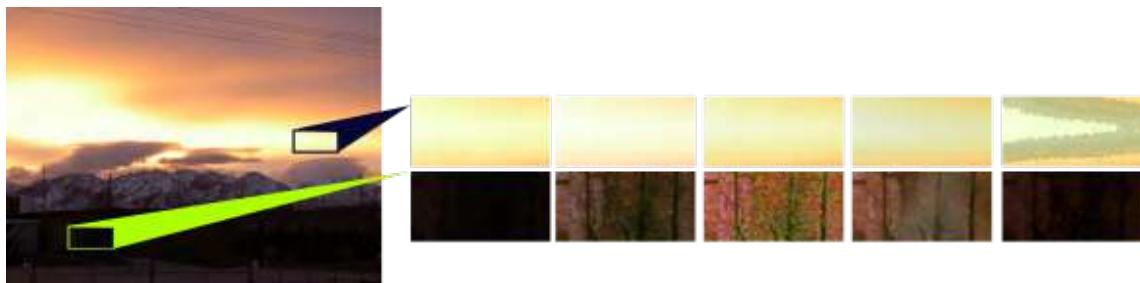


Figura 4.21: Pôr do sol nas montanhas

Quando há uma grande diferença entre as regiões distintas, nosso método não se sai muito bem, como já foi dito. Entretanto, na imagem do Telhado (figura 4.22), obteve um resultado global satisfatório.

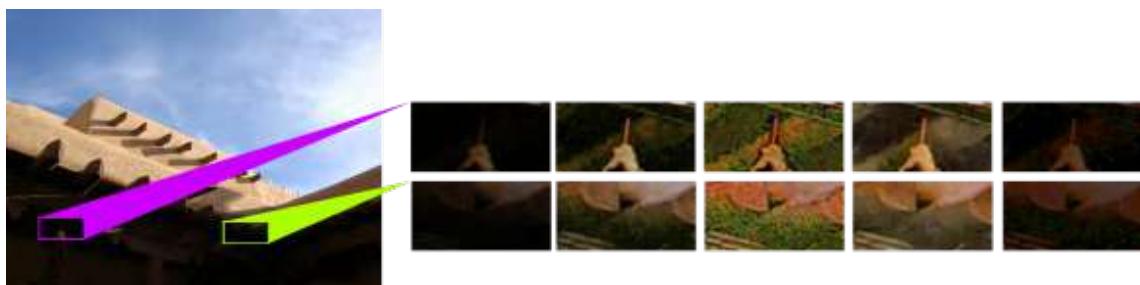


Figura 4.22: Telhado

Já na imagem do fundo do mar (figura 4.23), nosso método foi bem superior aos demais, conservando a tonalidade das cores e ampliando significativamente o contraste das regiões escuras.

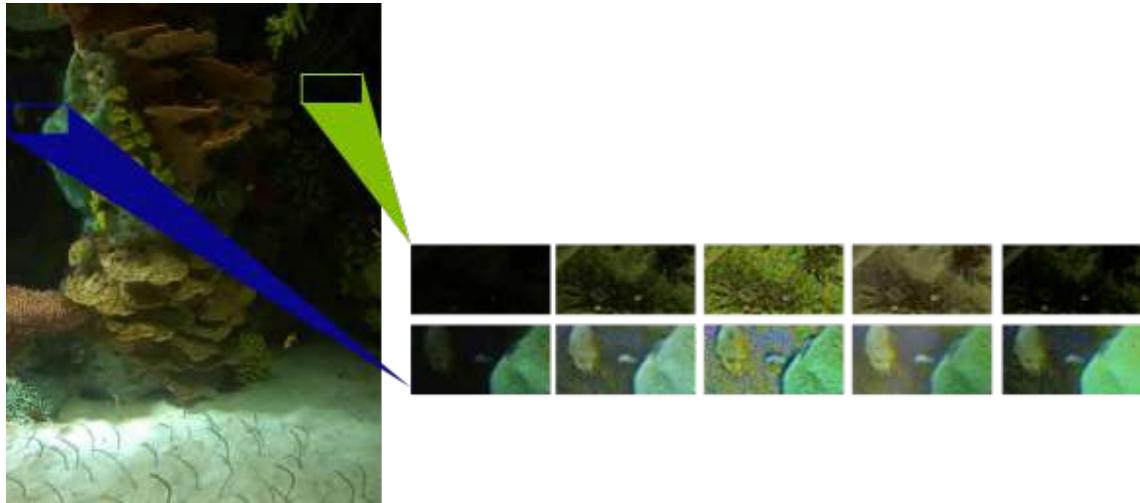


Figura 4.23: Fundo do mar

A imagem da figura 4.24 tem muita informação oculta que foi possível ser visualizada após o processamento. O nosso método foi ligeiramente superior ao método Zero-DCE e deixando os outros dois métodos para trás, em relação a qualidade da imagem final.

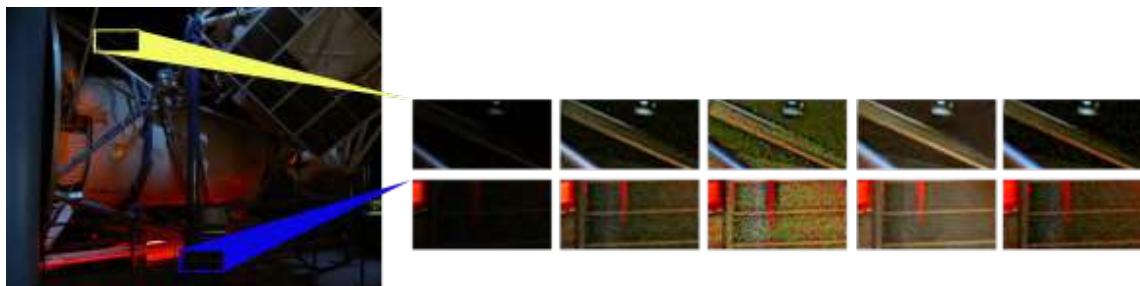


Figura 4.24: Satélite

Com relação à imagem do Templo ao Pôr do Sol (figura 4.25), o nosso método conseguiu obter um resultado muito interessante e superior os demais, uma vez que manteve a tonalidade das cores, sem dar a impressão da luz ter estourado as sombras do monumento, e mantido o desenho da grade.



Figura 4.25: Pôr do sol e o templo

Para o nosso método, a imagem do vaso pendurado na parede (figura 4.26), o resultado obtido foi ruim. Uma vez que a maior parte desta imagem apresenta uma grande área bem iluminada, e como já foi observado antes, nosso método foi desenvolvido para imagens com baixíssima iluminação, justamente o contrário desta imagem do vaso de flores. Nós realizamos este teste para confirmar a vocação do nosso método para imagens com grande área na penumbra.



Figura 4.26: Vaso de flores

A plantação de bambu (figura 4.27), foi um bom teste para verificar a ineficiência dos métodos RetinexNet e MBLLEN para este tipo de imagem. O RetinexNet teve uma saída pixelada como podemos observar nos recortes e o MBLLEN apesar de clarear bem a imagem, deixou embaçada. O nosso método, apesar de não ter dado uma correção ampla, não afetou as cores nem gerou borramento na imagem final obtendo assim, um bom resultado.



Figura 4.27: Bambu

A imagem das meninas dentro de uma casa (figura 4.28) nos mostra que o nosso método é bem interessante para preservar as nuances da imagem, como observado

na primeira linha em destaque. O método Zero-DCE embora tenha obtido melhor clareamento da imagem, demonstra uma impressão de embranquecimento na tonalidade da imagem.

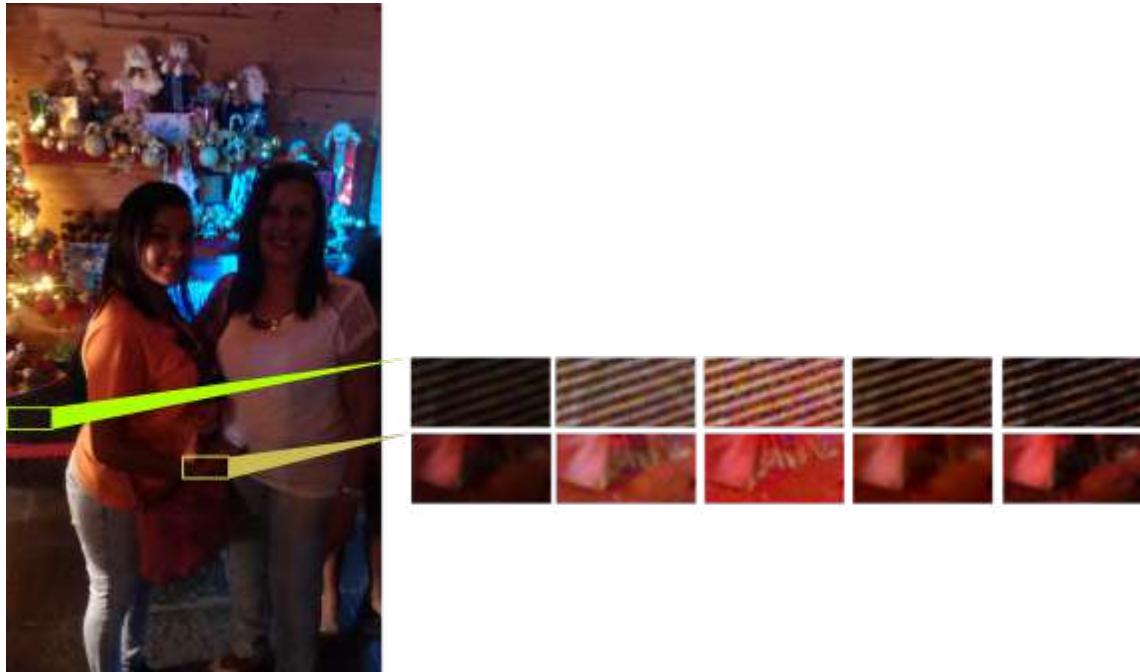


Figura 4.28: Meninas

As duas imagens da procissão em Paraty (figuras 4.29 e 4.30), dá a perfeita dimensão do nosso método, demonstrando a qualidade da melhora em uma imagem com quase a totalidade de sua informação na penumbra. Notamos que a imagem final é bem interessante.



Figura 4.29: Procissão em Paraty



Figura 4.30: Procissão em Paraty

Capítulo 5

Conclusões

A árdua tarefa de realizar pesquisas na área de processamento de imagens é a de aprimorar ou criar novos métodos que possam melhorar a vida de pessoas comuns à profissionais da área. Com este objetivo em mente, esta tese de doutorado foi realizada para trazer um novo meio de melhorar a qualidade das imagens noturnas para que possa ser visualizados objetos ocultados pela penumbra. Este objetivo foi alcançado eficientemente associando dois métodos computacionais distintos, a saber: Ressonância Estocástica e Correção de Histograma; que apresentaram resultados consistentes para as imagens do cotidiano, apesar de ainda ser um processo um pouco lento pode ser melhorado em estudos futuros. Comparado com o processamento via redes neurais, em algumas condições o resultado foi superior e a qualidade final ficou dentro do esperado (figuras: 4.17, 4.18, 4.19, 4.23, 4.24, 4.25, 4.27, 4.29 e 4.30).

Já as figuras 4.20, 4.21, 4.22 e 4.26 obtiveram desempenho insatisfatório pelo fato da cena ser mista e o método é ineficiente nesta situação, mas mesmo assim, obteve uma resposta melhor que o método RetinexNet.

Analizando os resultados dos métodos de inteligencia artificial aplicados ao problema das imagens com baixos níveis de contraste e avaliando a possibilidade de embarcar os métodos, é possível listar suas vantagens e desvantagem em seu uso.

Vantagens do uso de redes neurais:

- São rápidas - em geral são mais rápidas que os métodos convencionais por utilizarem, via de regra, cálculos simples;
- Capacidade de processamento - Utilização de paradigmas paralelos por natureza, tornam-se mais poderosas por realizarem mais cálculos em um mesmo espaço de tempo;

- São versáteis - basta ensinar corretamente, que são capazes de executar a tarefa ensinada com maestria.

Desvantagens do uso de rede neurais

- Necessitam de treinamento - sem um treinamento adequado, elas não terão acurácia necessária para solucionar problemas;
- Necessitam de GPU - em muitos ambientes embarcados, as GPU são simples e não fornecem tanto poder computacional quanto um *desktop*, o que pode inviabilizar seu uso;
- Entrada padronizada - necessitam de uma entrada formatada em um padrão que foi treinada anteriormente. Não foi possível utilizar diretamente as imagens do nosso conjunto de imagens porque algumas imagens usam cores indexadas e não era um arquivo com três canais (RGB);
- Tamanho das imagens - houve a necessidade de reduzir as dimensões das imagens do nosso conjunto de imagens para 640x480 *pixels*, uma vez que no tamanho natural geravam erros na alocação de memória.

Analisando os resultados do nosso método podemos destacar as vantagens dele:

- Hardware modesto - não necessita de um computador potente para ser executado;
- Automatizado - Totalmente automatizado, gerando imagens sem a interferência de um operador;
- Embarcável - Por utilizar um hardware mais simples, pode ser embarcado em computadores SoC de forma simplificada;
- Utiliza imagens de qualquer fonte - Utiliza qualquer tipo de imagem que possa ser processada pela biblioteca OpenCV. Isso traz muitas vantagens em relação a utilização das imagens.

Como nenhum método é ideal, nosso método possui algumas desvantagens:

- Não padronização das imagens - por aceitar imagens de qualquer tamanho e formato, o tempo gasto para processar a imagem fica atrelado ao seu tamanho;
- Imagens muito claras - por ser projetado para imagens com baixíssima iluminação, ele não funciona bem para imagens onde há iluminação normal ou uma área bem iluminada;
- Não utilização de GPU - O nosso método foi elaborado para ser utilizado em

CPU. Por outro lado, habilita o uso em qualquer hardware e em qualquer sistema operacional.

5.1 Contribuições

1. Utilização de hardware de baixo custo - É sem dúvida uma contribuição no sentido de obter-se um sistema barato e de fácil acesso para qualquer pessoa;
2. Utilização de ferramentas livres e abertas - De maneira geral, a utilização de ferramentas livres e abertas, possibilitam a utilização por um maior número de pessoas.
3. Implementação uma condição de parada - A implementação de uma condição de parada para a ressonância estocástica, proporciona ao método um meio de decisão para a sua finalização, de maneira mais precisa e confiável;
4. União de métodos - a aplicação de 3 métodos para gerar a automação na correção de imagens noturnas, apresentaram resultados satisfatórios.

5.2 Trabalhos futuros

Como trabalhos futuros, há pontos que precisam ou podem ser melhorados no processo, tais como:

1. Paralelizar o cálculo da ressonância estocástica - O fato de muitos desses equipamentos terem processadores *multi threads*, pode-se tirar proveito disso com a paralelização dos cálculos matriciais da ressonância estocástica de um modo mais intenso.
2. Utilizar processamento na GPU - Implementar o processamento utilizando OpenCL para universalizar o acesso a diferentes equipamentos.
3. Utilizar outras plataformas de hardware - Verificar se há viabilidade de uso de outras plataformas de hardware de baixo custo que possam prover este sistema, como o BananaPI ou BeagleBone.
4. Reduzir o consumo de memória - usar técnicas de redução de uso de memória, para que se possa reduzir o uso do espaço de **swap** do sistema, fazendo o desempenho melhorar significativamente.
5. Automatizar a escolha do divisor - Uma necessidade para melhoria geral do método é automatizar o processo de escolha do divisor D da variável b na

equação da ressonância estocástica, para que o sistema pudesse escolher qual a ordem de grandeza a ser utilizada.

Referências Bibliográficas

- [1] SIMONOTTO, E., RIANI, M., SEIFE, C., et al. “Visual Perception of Stochastic Resonance”, *Phys. Rev. Lett.*, v. 78, pp. 1186–1189, Feb 1997. doi: 10.1103/PhysRevLett.78.1186. Disponível em: <<https://link.aps.org/doi/10.1103/PhysRevLett.78.1186>>.
- [2] GAMMAITONI, L., HÄNGGI, P., JUNG, P., et al. “Stochastic resonance”, *Rev. Mod. Phys.*, v. 70, pp. 223–287, Jan 1998. doi: 10.1103/RevModPhys.70.223. Disponível em: <<https://link.aps.org/doi/10.1103/RevModPhys.70.223>>.
- [3] STOCKS, N. G. “Suprathreshold Stochastic Resonance in Multilevel Threshold Systems”, *Phys. Rev. Lett.*, v. 84, pp. 2310–2313, Mar 2000. doi: 10.1103/PhysRevLett.84.2310. Disponível em: <<https://link.aps.org/doi/10.1103/PhysRevLett.84.2310>>.
- [4] CHOUSHAN, R., BISWAS, P. K. “Image enhancement and dynamic range compression using novel intensity-specific stochastic resonance-based parametric image enhancement model”. In: *IEEE International Conference on Image Processing, ICIP 2014*, 2014. ISBN: 9781479957514. doi: 10.1109/ICIP.2014.7025919.
- [5] GUPTA, N., JHA, R. K., MOHANTY, S. K. “Enhancement of dark images using dynamic stochastic resonance in combined DWT and DCT domain”. In: *2014 9th International Conference on Industrial and Information Systems (ICIIS)*, pp. 1–6, 2014. doi: 10.1109/ICIINFS.2014.7036546.
- [6] WANG, Z., BOVIK, A. C., LU, L. “Why is image quality assessment so difficult?” In: *2002 IEEE International Conference on Acoustics, Speech, and Signal Processing*, v. 4, pp. IV–3313–IV–3316, 2002. doi: 10.1109/ICASSP.2002.5745362.
- [7] FOUNDATION, T. R. P. “The Raspberry Pi Foundation”. 2017. Disponível em: <<https://www.raspberrypi.org/about/>>.

- [8] WUELLER, D., KEJSER, U. B. “Standardization of Image Quality Analysis ISO 19264”, *Archiving Conference*, v. 2016, n. 1, pp. 111–116, 2016. ISSN: 2161-8798. doi: doi:10.2352/issn.2168-3204.2016.1.0.111. Disponível em: <<https://www.ingentaconnect.com/content/ist/ac/2016/00002016/00000001/art00025>>.
- [9] SWAINE, M. R., FREIBERGER, P. A. “Whirlwind”. 2017. Disponível em: <<https://www.britannica.com/technology/Whirlwind-computer>>.
- [10] FEDORKOW, G. C. “The Whirlwind Computer at CHM”. 2018. Disponível em: <<https://computerhistory.org/blog/the-whirlwind-computer-at-chm>>.
- [11] GAYATHRI, S., MOHANAPRIYA, N., KALAAVATHI, B. “Survey on Contrast Enhancement Techniques”, *International Journal of Advanced Research in Computer and Communication Engineering*, v. 2, n. 11, 2013. ISSN: 2278-1021. Disponível em: <www.ijarcce.com>.
- [12] GONZALEZ, R., WOODS, R. *Processamento de imagens digitais*. 3 ed. , Pearson, 2009. ISBN: 9788576054016.
- [13] CORP, N. “Nikon Corporate”. 2017. Disponível em: <<http://http://www.nikon.com.br/nikon-products/product-archive/dslr-cameras/d90.html#tab-ProductDetail-ProductTabs-TechSpecs>>.
- [14] CORP, N. “Nikon Corporate”. 2017. Disponível em: <https://www.nikon.pt/pt_PT/product/digital-cameras/slr/professional/df>.
- [15] CORP, G. “GitLab Official Site”. 2020. Disponível em: <<https://www.gitlab.com>>.
- [16] TEAM, F. “FreeBSD Operationa System”. 2017. Disponível em: <<https://www.freebsd.org/about.html>>.
- [17] CORP, A. “AMD Corporation”. 2017. Disponível em: <<http://products.amd.com/pt-br/search/CPU/AMD-FX-Series/AMD-FX-8-Core-Black-Edition/FX-8300/95>>.
- [18] ORG TEAM, X. “Xorg Team”. 2020. Disponível em: <<https://www.x.org/wiki/>>.
- [19] TEAM, M. “Mate Team”. 2020. Disponível em: <<https://mate-desktop.org/pt/>>.

- [20] TEAM, O. “OpenCV Open Source Computer Vision Library”. 2017. Disponível em: <<http://opencv.org/about.html>>.
- [21] GOMES, J., VELHO, L. *Fundamentos da computação Gráfica*. 1 ed. , Instituto de Matemática Pura e Aplicada, 2008. ISBN: 978-8524402005.
- [22] SCURI, A. E. “Fundamentos da Imagem Digital”. 2002. Disponível em: <<https://webserver2.tecgraf.puc-rio.br/~scuri/download/fid.pdf>>.
- [23] JAIN, A. K. *Fundamentals of Digital Image Processing*. 1 ed. , Prentice Hall, 1998. ISBN: 978-0133361650.
- [24] CHRISTENSSON, P. “Camera RAW Definition”. 2016. Disponível em: <https://techterms.com/definition/camera_raw>.
- [25] TAO, B., TASTL, I., COOPER, T., et al. “Digital Image System and Method for combining Demosaicing and bad pixel correction”, *Patent Application Publication*, v. 41, n. 4, pp. 1118–1126, 1995.
- [26] KAKARALA, R., BAHA, I. “Digital Image System and Method for Implementing an Adaptive Demosaicing Method”, *Patent Application Publication*, 2003.
- [27] CHUNG, K. L., YANG, W. J., YAN, W. M., et al. “Demosaicing of color filter array captured images using gradient edge detection masks and adaptive heterogeneity-projection”, *IEEE Transactions on Image Processing*, 2008. ISSN: 10577149. doi: 10.1109/TIP.2008.2005561.
- [28] RAMANATH, R., SNYDER, W. E., BILBRO, G. L., et al. “Demosaicking methods for Bayer color arrays”, *Journal of Electronic Imaging*, v. 11, n. 3, pp. 306 – 315, 2002. doi: 10.1117/1.1484495. Disponível em: <<https://doi.org/10.1117/1.1484495>>.
- [29] KIMMEL, R. “Demosaicing: Image reconstruction from color CCD samples”. In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 1998. ISBN: 3540645691. doi: 10.1007/BFb0055693.
- [30] LUKIN, A., KUBASOV, D. “High-Quality Algorithm for Bayer Pattern Interpolation”, *Programming and Computer Software*, v. 30, 2004. doi: 10.1023/B:PACS.0000049512.71861.eb. Disponível em: <<https://doi.org/10.1023/B:PACS.0000049512.71861.eb>>.

- [31] WU, X., JIANG, Z.-P., REPPERGER, D. W., et al. “Enhancement of Stochastic Resonance Using Optimization Theory”, *Commun. Inf. Syst.*, v. 6, n. 1, pp. 1–18, 2006. Disponível em: <<https://projecteuclid.org:443/euclid.cis/1175791039>>.
- [32] BENZIT\$, R., SUTERA, A., VULPIANI, A. “The mechanism of stochastic resonance”, *J. Phys. A: Math. Gen. J. Phys. A: Math. Gen.*, v. 14, n. 14, pp. 453–457, 1981. Disponível em: <<http://iopscience.iop.org/0305-4470/14/11/006>>.
- [33] DEOLE, M. T., HINGWAY, S. P., SURESH, S. S. “Dynamic Stochastic Resonance for Low Contrast Image Enhancement”, *IOSR Journal of VLSI and Signal Processing Ver. II*, v. 4, n. 4, pp. 1–5, 2014. Disponível em: <www.iosrjournals.org>.
- [34] KUMAR GUPTA, A., SINGH CHAUHAN ASST, S., SHRIVASTAVA, M. “Low Contrast Image Enhancement Technique By Using Fuzzy Method”, *International Journal of Engineering Research and General Science*, v. 4, n. 2, 2016. ISSN: 2091-2730.
- [35] QINGHUA YE, HAINING HUANG, CHUNHUA ZHANG. “Image enhancement using stochastic resonance [sonar image processing applications]”. In: *2004 International Conference on Image Processing, 2004. ICIP '04.*, v. 1, pp. 263–266 Vol. 1, 2004. doi: 10.1109/ICIP.2004.1418740. Disponível em: <<https://ieeexplore.ieee.org/document/1418740>>.
- [36] WANG, J., PENG, Y., WANG, X., et al. “Study on algorithm of image restoration based on Stochastic Resonance and Weiner filtering”. In: *2015 IEEE 5th International Conference on Electronics Information and Emergency Communication*, pp. 244–247, 2015. doi: 10.1109/ICEIEC.2015.7284531.
- [37] IQBAL, K., SALAM, R., AZAM, O., et al. “Underwater Image Enhancement Using an Integrated Colour Model”, *IAENG International Journal of Computer Science*, v. 34, 01 2007.
- [38] LEE, E., KIM, S., KANG, W., et al. “Contrast Enhancement Using Dominant Brightness Level Analysis and Adaptive Intensity Transformation for Remote Sensing Images”, *IEEE Geoscience and Remote Sensing Letters*, v. 10, n. 1, 2013. doi: 10.1109/LGRS.2012.2192412.
- [39] ESKICIÖGLU, A., FISHER, P. “A survey of quality measures for gray scale image compression”. In: *9th Computing in Aerospace Conference, Aeros-*

- pace Research Central, 2012. doi: 10.2514/6.1993-4514. Disponível em: <<https://arc.aiaa.org/doi/abs/10.2514/6.1993-4514>>.
- [40] ESKICIOGLU, A. M., FISHER, P. S. “Image Quality Measures and Their Performance”, *IEEE TRANSACTIONS ON COMMUNICATIONS*, v. 43, n. 12, 1995.
- [41] RODRIGUES, E. L., FRANÇA, C. A. D., RODA, V. O., et al. “Classificação de Imagens Baseada na Entropia de Shannon”. In: *Workshop da Sub-Rede Modelagem, Imagem e Visualização*. PUC-RIO, 2000. Disponível em: <<http://iris.sel.eesc.usp.br/lavi/pdf/08.pdf>>.
- [42] GUO, C. G., LI, C., GUO, J., et al. “Zero-reference deep curve estimation for low-light image enhancement”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, pp. 1780–1789, June 2020.
- [43] WEI, C., WANG, W., YANG, W., et al. “Deep Retinex Decomposition for Low-Light Enhancement”. In: *British Machine Vision Conference*. British Machine Vision Association, 2018.
- [44] LV, F., LU, F., WU, J., et al. “MBLLEN: Low-light Image/Video Enhancement Using CNNs”. In: *British Machine Vision Conference (BMVC)*, 2018.
- [45] ZHANG, Y., ZHANG, J., GUO, X. “Kindling the Darkness: A Practical Low-light Image Enhancer”. In: *Proceedings of the 27th ACM International Conference on Multimedia*, MM ’19, pp. 1632–1640, New York, NY, USA, 2019. ACM. ISBN: 978-1-4503-6889-6. doi: 10.1145/3343031.3350926. Disponível em: <<http://doi.acm.org/10.1145/3343031.3350926>>.
- [46] FILHO, J. R. B. “Blog de Fotografia”. 2015. Disponível em: <<https://jgoffredo.blogspot.com/>>.
- [47] JIANG, Y., GONG, X., LIU, D., et al. “EnlightenGAN: Deep Light Enhancement without Paired Supervision”, *arXiv preprint arXiv:1906.06972*, 2019.
- [48] WANG, R., ZHANG, Q., FU, C.-W., et al. “Underexposed Photo Enhancement Using Deep Illumination Estimation”. In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.
- [49] DEVICES, A. M. “FX-8300”. 2020. Disponível em: <<https://www.amd.com/pt/products/cpu/fx-8300>>.

- [50] FOUNDATION, T. R. P. “The Raspberry Pi Foundation”. 2017. Disponível em: <<https://static.raspberrypi.org/files/product-briefs/Raspberry-Pi-Model-Bplus-Product-Brief.pdf>>.
- [51] COFFIN, D. “Dcraw official site”. 2017. Disponível em: <https://www.cybercom.net/~dcoffin/dcraw/index_pt.html>.

Apêndice A

Acesso ao conteúdo deste trabalho

A.1 Acesso ao código fonte

O código fonte poderá ser obtido através do link <https://gitlab.com/njrizzo/thesis-dsc>, Todo o código está escrito em C++ utilizando a especificação C++17.

A.2 Acesso as figuras

Essas imagens estão armazenadas no google drive, o link pode ser obtido através da minha página pessoal (<http://www.rizzo.eng.br/Thesis/>) ou na página do laboratório de Computação Gráfica - LCG (<http://www.lcg.ufrj.br/~rizzo/Thesis/>).

Leia as atentamente o arquivo README.md para ver informações atualizadas da documentação sobre a compilação e utilização dos programas.

Apêndice B

Gráficos da condição de parada

B.1 Introdução

Neste anexo, é possível analisar os gráficos contendo o comportamento da média, do desvio padrão e do fator F durante o processo da ressonância estocástica, vendo como a condição de parada atua.

As linhas dos gráficos correspondem as imagens do conjunto criado para este trabalho e cada linha corresponde aos valores de 10^3 , 10^4 , 10^{15} e 10^{16} .

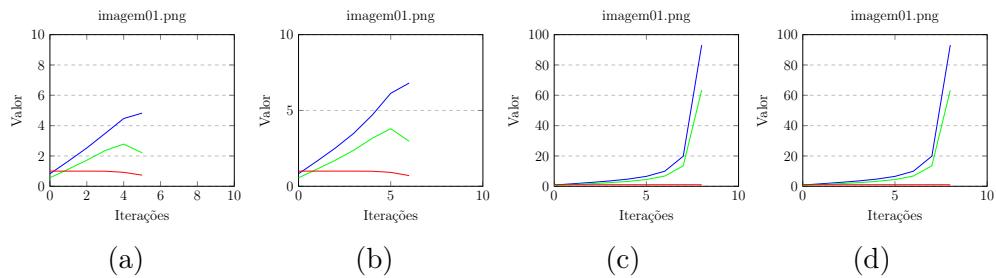


Figura B.1: Gráficos da condição de parada da imagem 1

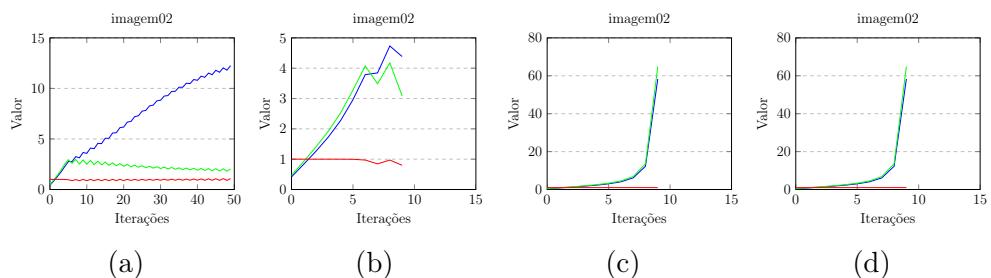


Figura B.2: Gráficos da condição de parada da imagem 2

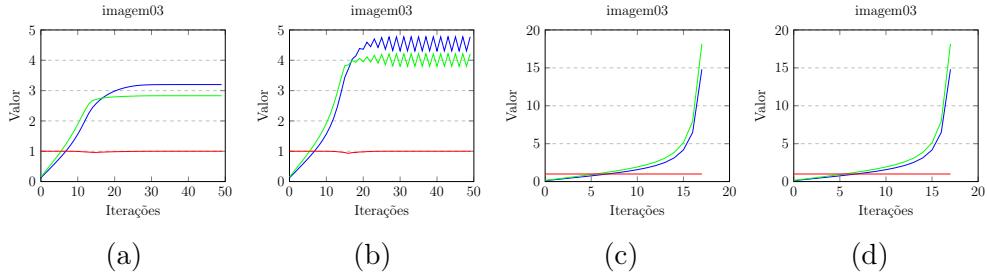


Figura B.3: Gráficos da condição de parada da imagem 3

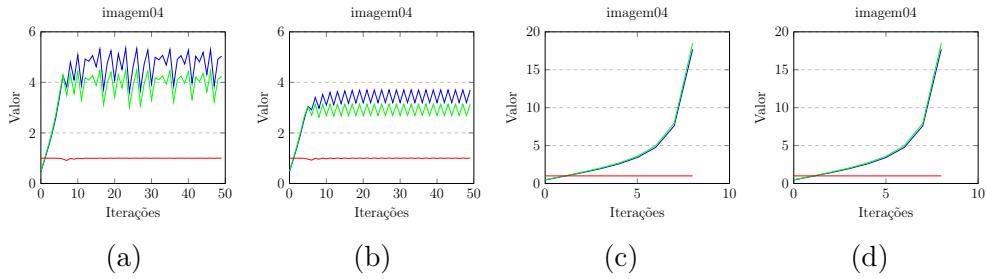


Figura B.4: Gráficos da condição de parada da imagem 4

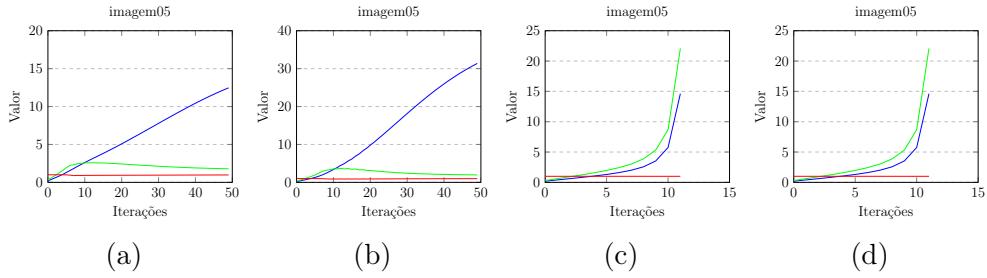


Figura B.5: Gráficos da condição de parada da imagem 5

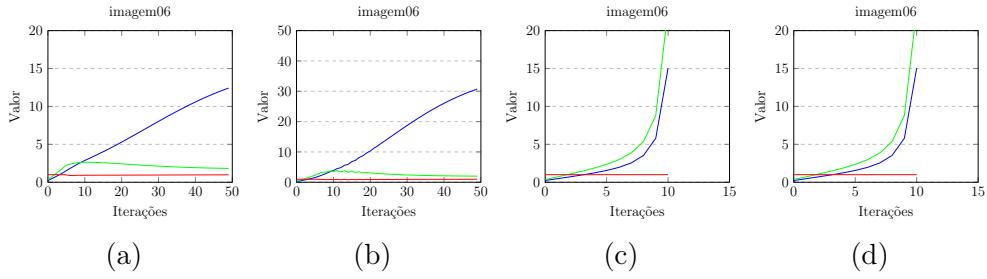


Figura B.6: Gráficos da condição de parada da imagem 6

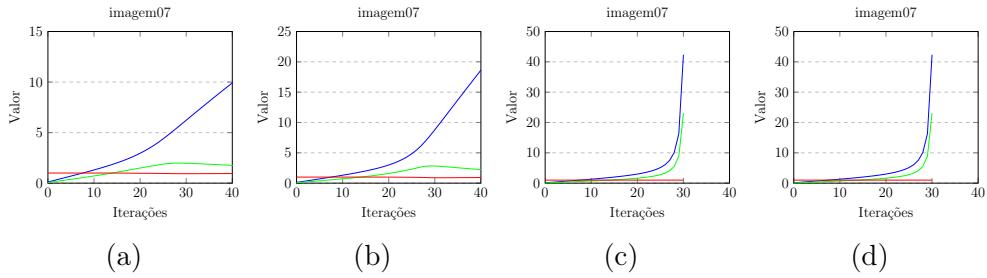


Figura B.7: Gráficos da condição de parada da imagem 7

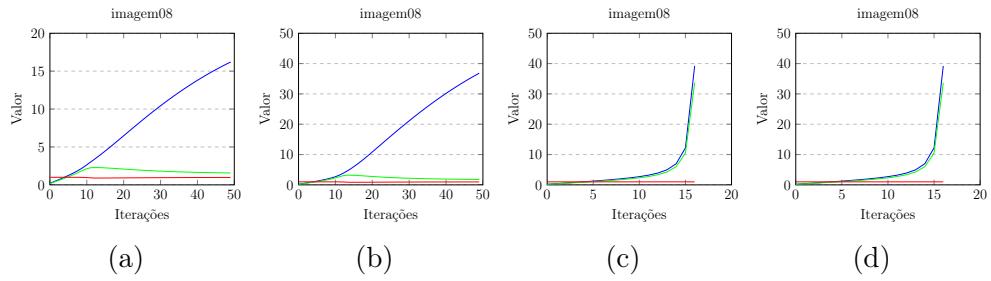


Figura B.8: Gráficos da condição de parada da imagem 8

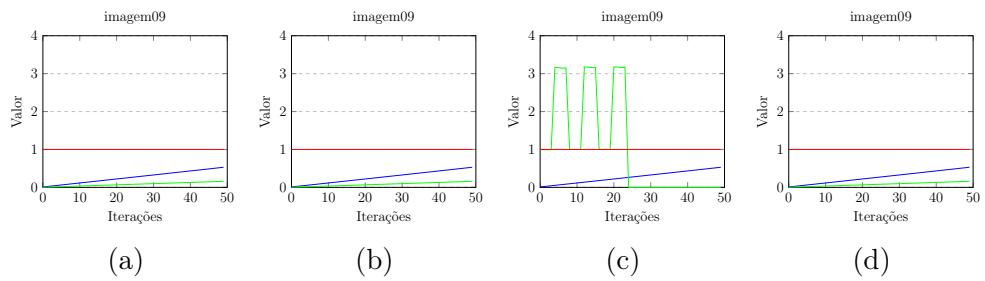


Figura B.9: Gráficos da condição de parada da imagem 9

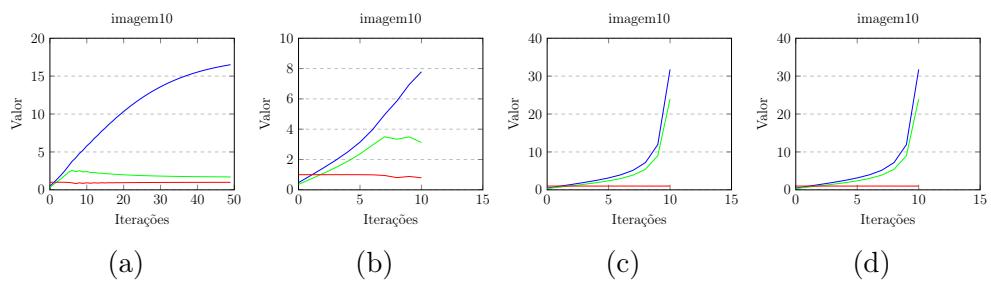


Figura B.10: Gráficos da condição de parada da imagem 10

Apêndice C

Analise de histograma

C.1 Introdução

Foram análises as imagens do conjunto de imagens criado para este método utilizando o histograma, onde podemos ver que os melhores resultados ocorrem com o valor para o divisor $D = 10^{16}$.

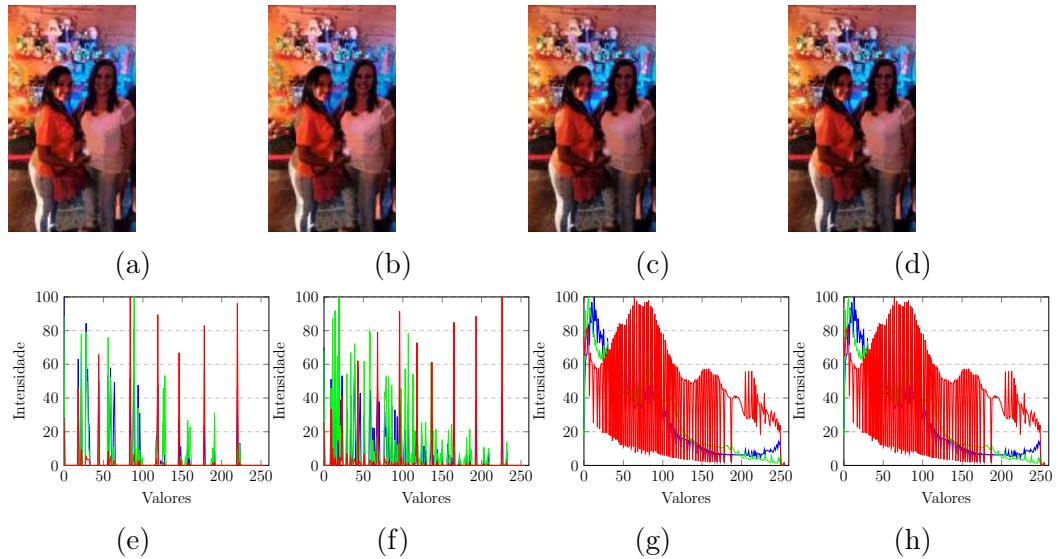


Figura C.1: Análise de histograma para imagem 1

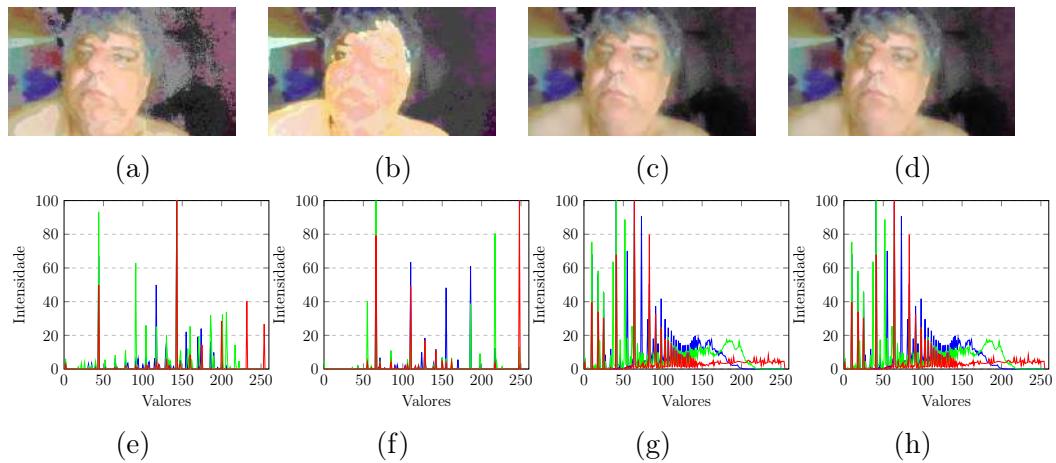


Figura C.2: Análise de histograma para imagem 2

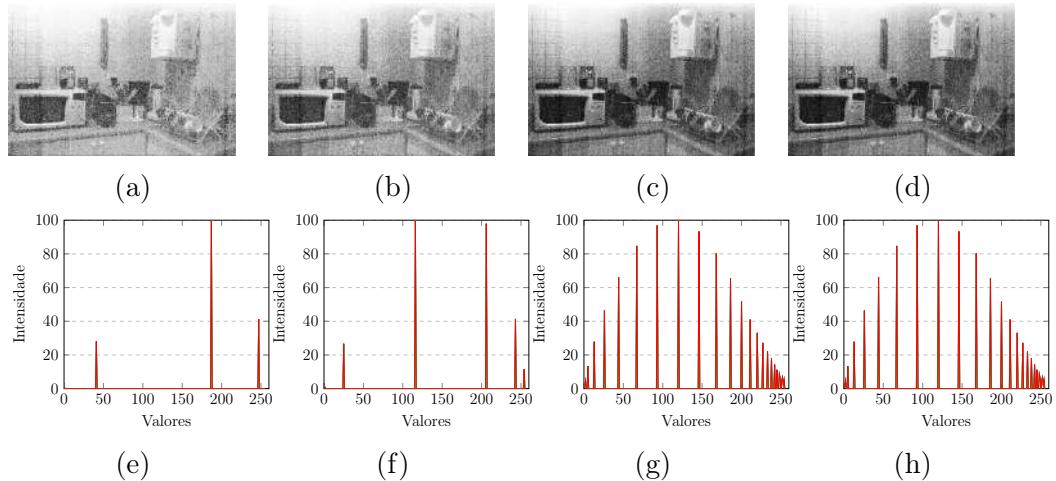


Figura C.3: Análise de histograma para imagem 3

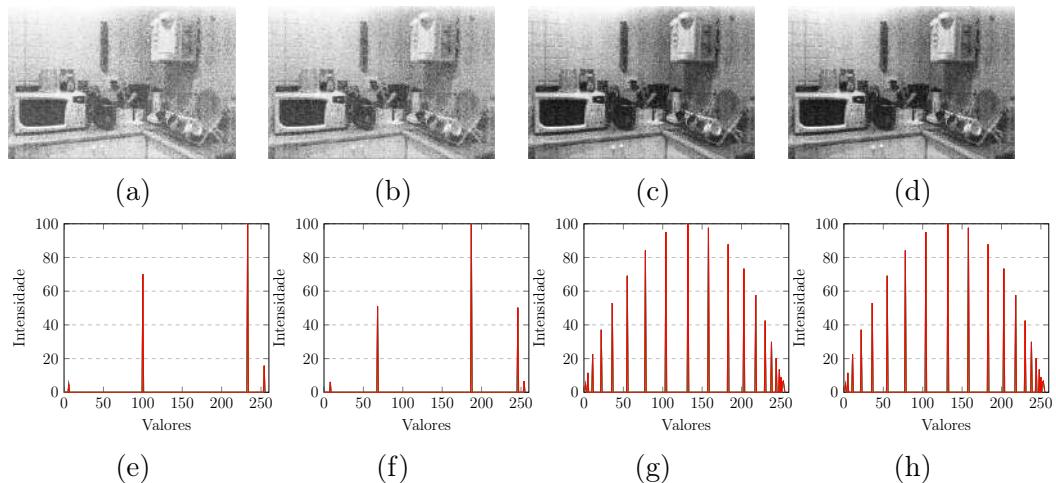


Figura C.4: Análise de histograma para imagem 4

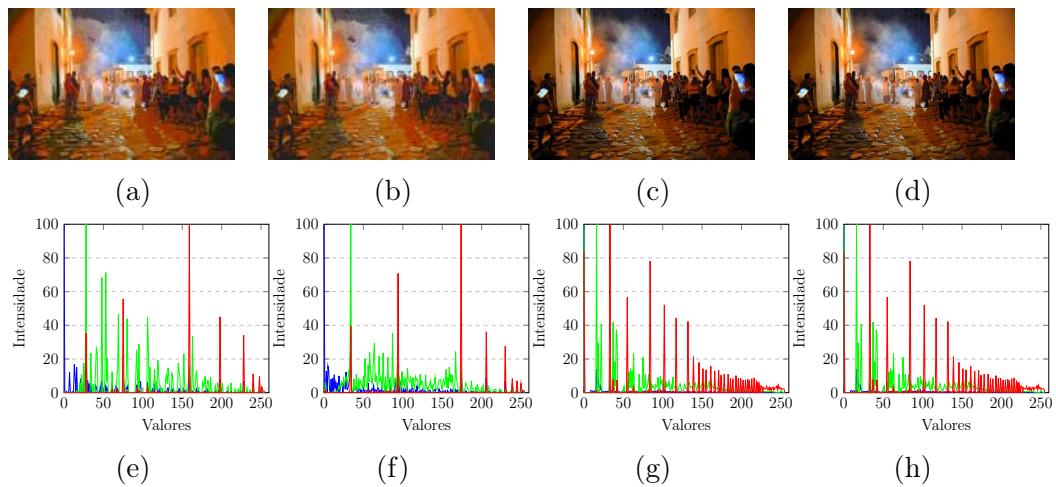


Figura C.5: Análise de histograma para imagem 5

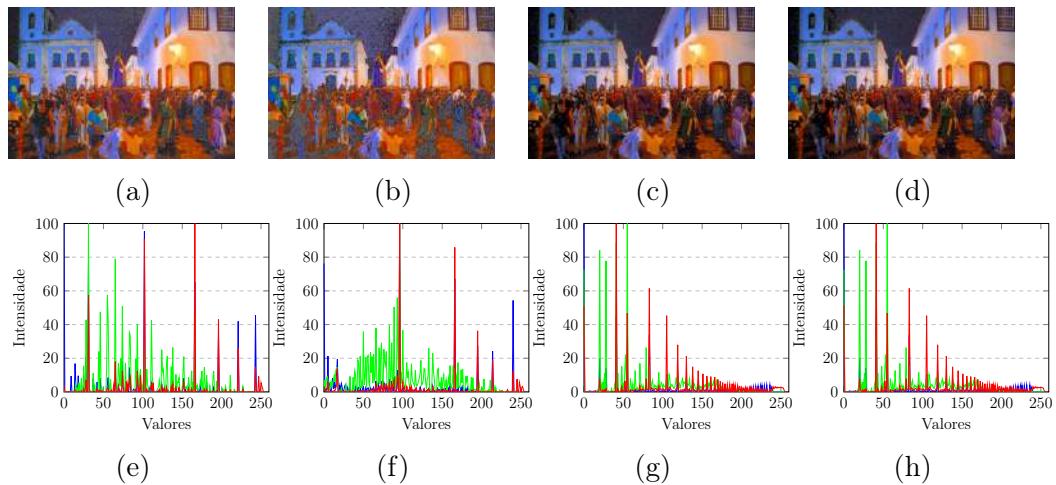


Figura C.6: Análise de histograma para imagem 6

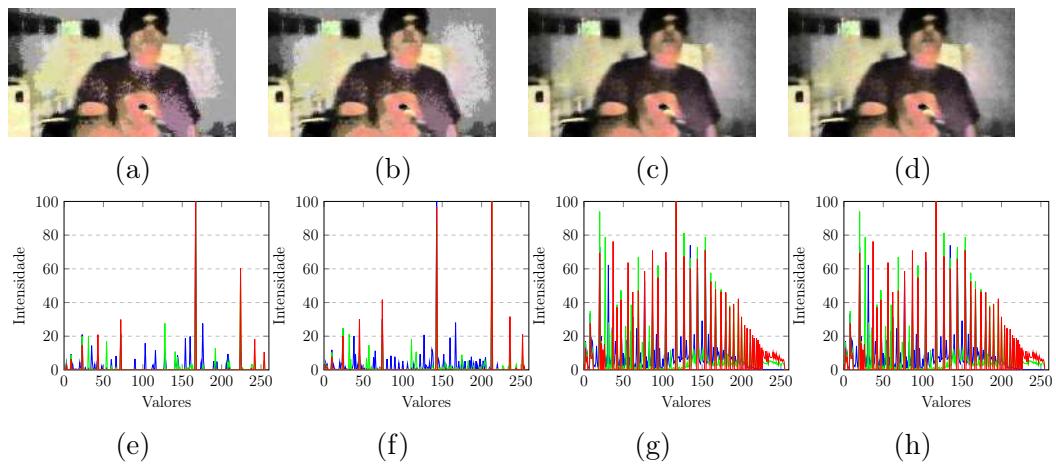


Figura C.7: Análise de histograma para imagem 7

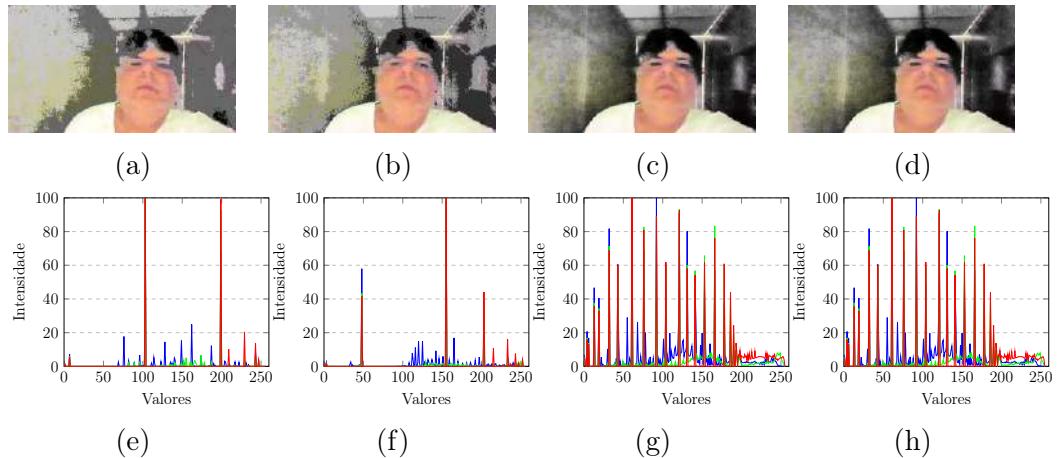


Figura C.8: Análise de histograma para imagem 8

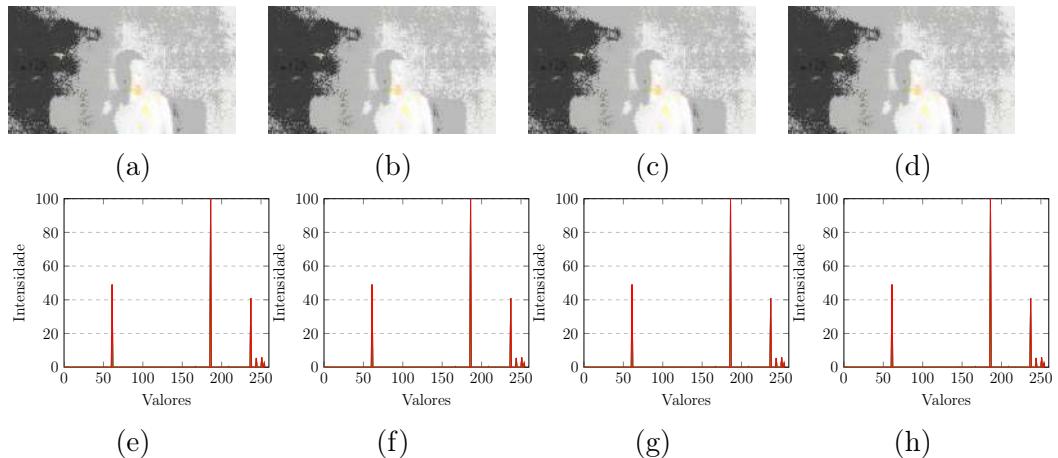


Figura C.9: Análise de histograma para imagem 9

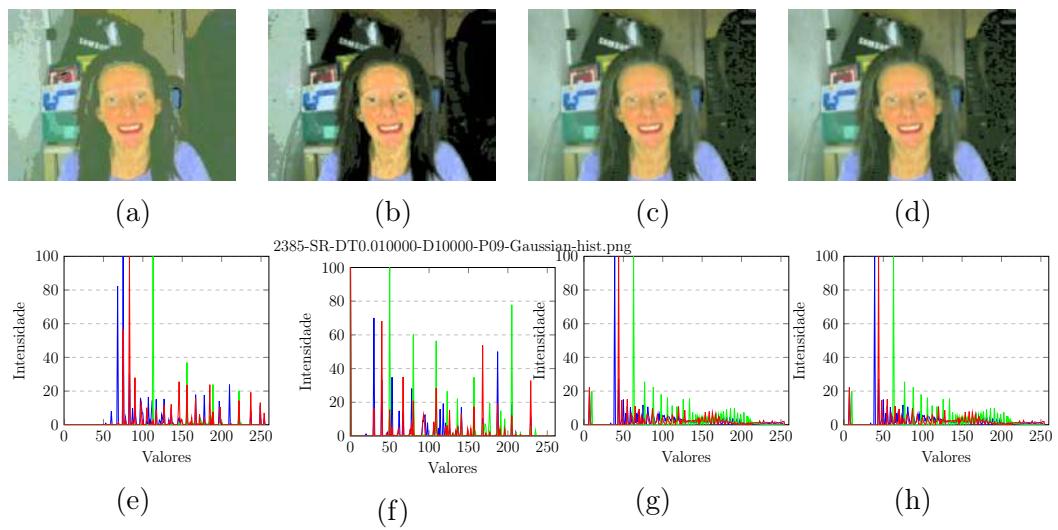


Figura C.10: Análise de histograma para imagem 10