

**UNIVERSIDADE DO ESTADO DO AMAZONAS
ESCOLA SUPERIOR DE TECNOLOGIA
GRADUAÇÃO EM ENGENHARIA ELÉTRICA**

INGRID NUNES RODRIGUES

**DESENVOLVIMENTO DE UM DISPOSITIVO PARA REALIZAR O BALANÇO DE
BRANCO EM TELEVISORES EM TEMPO REAL UTILIZANDO A TÉCNICA DA MÉDIA
DE MÚLTIPLAS IMAGENS**

Manaus

2019

INGRID NUNES RODRIGUES

**DESENVOLVIMENTO DE UM DISPOSITIVO PARA REALIZAR O BALANÇO DE
BRANCO EM TELEVISORES EM TEMPO REAL UTILIZANDO A TÉCNICA DA MÉDIA
DE MÚLTIPLAS IMAGENS**

Projeto de pesquisa desenvolvido durante a disciplina de Trabalho de Conclusão de Curso II e apresentada à banca avaliadora do Curso de Engenharia Elétrica da Universidade do Estado do Amazonas, como pré-requisito para obtenção do título de Engenheira Eletricista.

Orientador: Jozias Parente de Oliveira, Dr.

Manaus

2019

Universidade do Estado do Amazonas – UEA
Escola Superior de Tecnologia - EST

Reitor:

Cleinaldo de Almeida Costa

Vice-Reitor:

Cleto Cavalcante de Souza Leal

Diretora da Escola Superior de Tecnologia:

Ingrid Sammyne Gadelha Figueiredo

Coordenador do Curso de Engenharia Elétrica:

Walfredo da Costa Lucena Filho

Banca Avaliadora composta por:

Prof. Jozias Parente de Oliveira (Orientador)

Prof. Karlo Homero Ferreira

Prof. Júlio Feitoza Pereira

Data da defesa: 17/12/2019.

CIP – Catalogação na Publicação

Nunes Rodrigues, Ingrid

Desenvolvimento de um dispositivo para realizar o balanço de branco em televisores em tempo real utilizando a técnica da média de múltiplas imagens / Ingrid Nunes Rodrigues; [orientada por] Jozias Parente de Oliveira. – Manaus: 2019.

51 p.: il.

Trabalho de Conclusão de Curso (Graduação em Engenharia Elétrica). Universidade do Estado do Amazonas, 2019.

1. Balanço de branco. 2. Processamento digital de imagens. 3. Média de Múltiplas Imagens. 4. *Orange Pi*. I. Parente de Oliveira, Jozias.

INGRID NUNES RODRIGUES

DESENVOLVIMENTO DE UM DISPOSITIVO PARA REALIZAR O BALANÇO
DE BRANCO EM TELEVISORES EM TEMPO REAL UTILIZANDO A TÉCNICA DA
MÉDIA DE MÚLTIPLAS IMAGENS

Pesquisa desenvolvida durante a disciplina de Trabalho de Conclusão de Curso II e apresentada à banca avaliadora do Curso de Engenharia Elétrica da Escola Superior de Tecnologia da Universidade do Estado do Amazonas, como pré-requisito para a obtenção do título de Engenheira Eletricista.

Nota obtida: _____ (_____)

Aprovada em ____/____/____.

Área de concentração: Processamento Digital de Imagens

BANCA EXAMINADORA

Orientador: Jozias Parente de Oliveira, Dr.

Avaliador: Karlo Homero Ferreira, Me.

Avaliador: Júlio Feitoza Pereira, Me.

DEDICATÓRIA

Dedico este trabalho aos amantes da tecnologia,
aos que buscam constantemente por
conhecimento e botam isso em prática.

AGRADECIMENTOS

Agradeço a Deus por ter me dado força, saúde e sabedoria para superar as dificuldades encontradas ao longo de minha trajetória acadêmica.

Aos meus pais, pois sem seu incentivo, dedicação e apoio jamais teria conseguido.

Ao meu orientador pela orientação e apoio durante a elaboração deste trabalho.

Ao meu namorado que me incentivou e não mediu esforços para me ajudar.

Aos colegas de trabalho que me ajudaram durante ensaios em laboratório.

Por fim, agradeço aos meus amigos que me acompanharam e estiveram ao meu lado durante todas as complicações da graduação, que me incentivaram e contribuíram para essa conquista.

RESUMO

O balanço de branco em TVs busca aprimorar a imagem exibida pela tela de forma que seja mais próxima possível da imagem real através da remoção da projeção de cores não realistas. Como atualmente esses ajustes são realizados apenas em fábrica, este projeto buscou desenvolver um dispositivo de baixo custo que fizesse o balanço de branco em tempo real utilizando da técnica de processamento digital de imagem denominada Média de Múltiplas Imagens. Essa técnica consiste em obter uma imagem média a partir de N imagens ruidosas de forma que a imagem resultante se aproxime cada vez mais de uma imagem livre de ruídos. Isso será necessário, uma vez que o projeto se baseia em fazer a aquisição dessas imagens através de uma câmera. O programa para fazer o processamento das imagens foi desenvolvido na linguagem *Python* utilizando a biblioteca de visão computacional *OpenCV*. Para fundamentar o projeto, foi elaborada uma pesquisa sobre o padrão de cores utilizado em televisores, o Rec. 709, balanço de branco, processamento digital de imagens, *OpenCV*, *Python* e *Orange Pi*, que será o microcomputador utilizado para fazer o processamento em tempo real. O projeto baseia-se em capturar imagens de televisores com o balanço de branco correto e incorreto, obter suas imagens médias sem ruído e fazer uma comparação entre elas por meio do método de Correlação de Pearson, onde foi estabelecido um ponto de corte para o valor do coeficiente para imagens corretamente balanceadas, em seguida é enviado um sinal para que a TV faça o autoajuste, o qual será simulado por meio de um circuito sinalizador de LED. Após a realização dos ensaios, foi possível concluir que o dispositivo pode auxiliar nesse procedimento de forma eficiente e confiável.

Palavras chave:

Balanço de branco – Processamento digital de imagens – Média de Múltiplas Imagens – Correlação de Pearson – *Orange Pi*.

ABSTRACT

White balance on TVs seeks to enhance the screen display image so that it is as close as possible to the real image by removing unrealistic color projection. As these adjustments are currently performed only at the factory, this project sought to develop a low-cost device that would perform real-time white balance using the digital image processing technique called Multiple Image Average. This technique consists of obtaining an average image from N noisy images so that the resulting image gets closer and closer to a noise free image. This will be necessary since the project is based on acquiring these images through a camera. The program for processing images was developed in Python using the OpenCV computer vision library. To support the project, a survey was conducted on the color standard used in televisions, the Rec. 709, white balance, digital image processing, OpenCV, Python and Raspberry Pi, which will be the microcomputer used for real-time processing. The project is based on capturing television images with the correct and incorrect white balance, obtaining their average images without noise, and comparing them using the Pearson Correlation method, where the coefficient cut-off value was established for correctly balanced images, then a signal is sent to the TV to self-adjust, which will be simulated via an LED signaling circuit. After the tests, it was concluded that the device can assist in this procedure efficiently and reliably.

Keywords:

White balance – Digital image processing – Multiple Image Average – Pearson Correlation – Orange Pi.

ÍNDICE DE FIGURAS

Figura 1 – Padrão Rec. 709 dentro do Diagrama de Cromaticidade CIE 1931 XYZ	14
Figura 2 - Espectro visível da Luz.....	15
Figura 3 - Diagrama de cromaticidade xy do sistema tricromático CIE 1931	16
Figura 4 - Plano cartesiano de cores RGB e cubo RGB	18
Figura 5 - (a) Balanço de branco correto. (b) Balanço de branco incorreto	20
Figura 6 - <i>Software</i> de ajuste de Balanço de Branco	21
Figura 7 - Representação de uma imagem digital	22
Figura 8 - Etapas do Processamento Digital de Imagens	23
Figura 9 - Diminuição da taxa de amostragem N de imagens respectivas	24
Figura 10 - Técnica da média de múltiplas imagens para diferentes valores de M.....	26
Figura 11 - <i>Orange Pi PC Plus H3</i>	28
Figura 12 - Fluxograma das etapas do algoritmo	30
Figura 13 - Fluxograma das etapas gerais do projeto	31
Figura 14 - Obtenção de 30 imagens de referência	32
Figura 15 - Cálculo da Imagem Média.....	32
Figura 16 - Captura das imagens de teste e cálculo da imagem média de teste	33
Figura 17 - Cálculo do coeficiente de Pearson	34
Figura 18 - Comprime as imagens de referência e de teste	34
Figura 19 - Declaração das matrizes e recebimento das componentes R, G e B.....	35
Figura 20 - Cálculo da média das matrizes R e G e coeficiente de Pearson.	35
Figura 21 - Estabelecendo uma sessão SSH com Windows PowerShell	37
Figura 22 - Distribuição das GPIOs	38
Figura 23 - Circuito sinalizador.....	38
Figura 24 - Comparação entre imagem com balanço de branco correto e incorreto.....	39
Figura 25 - Matriz de confusão.....	41
Figura 26 - (a) Imagem de referência (b) Imagem de teste	42
Figura 27 - Resultado mostrado no circuito sinalizador.....	42
Figura 28 - (a) Imagem de referência (b) Imagem de teste	43
Figura 29 - LED aceso sinalizando ajuste correto.....	43

LISTA DE EQUAÇÕES

Equação 1 – Equação da coordenada de cromaticidade x	17
Equação 2 - Equação da coordenada de cromaticidade y	17
Equação 3 - Equação da coordenada de cromaticidade z	17
Equação 4 – Transformação de matriz para obtenção do valores tristimulares	18
Equação 5 - Função bidimensional de intensidade da luz $f(x, y)$	22
Equação 6 - Quantidade de b (bits) necessários para armazenar a imagem	24
Equação 7 – Equação para cálculo de uma imagem média	25
Equação 8 - Valor esperado de $\bar{g}(x,y)$	25
Equação 9 - Desvio-padrão da imagem filtrada	25
Equação 10 - Variância da imagem filtrada	25
Equação 11 - Coeficiente de correlação de Pearson (ρ)	26
Equação 12 – Equação da acurácia	41
Equação 13 – Equação da precisão	41
Equação 14 – Equação da revocação	41
Equação 15 - Cálculo do resultado da acurácia do primeiro método	44
Equação 16 - Cálculo do resultado da precisão do primeiro método	44
Equação 17 - Cálculo do resultado da revocação do primeiro método	44
Equação 18 - Cálculo do resultado da acurácia do segundo método	44
Equação 19 - Cálculo do resultado da precisão do segundo método	45
Equação 20 - Cálculo do resultado da revocação do segundo método	45

LISTA DE TABELAS

Tabela 1 - Coordenadas de cromaticidade que definem as primárias e o branco padrão	14
Tabela 2 - Valores de estímulos e coordenadas de cromaticidade para seis iluminantes selecionados.....	18
Tabela 3 - CCT nominal e real dos Iluminantes CIE diurnos	19
Tabela 4 - Coeficientes de Pearson após comparação entre 10 imagens.....	39
Tabela 5 - Coeficientes de Pearson utilizando o segundo método	40
Tabela 6 - Resultados obtidos no primeiro método após 20 testes.....	44
Tabela 7 - Resultados obtidos no segundo método após 20 testes	44

SUMÁRIO

INTRODUÇÃO	11
1 REFERENCIAL TEÓRICO	13
1.1 O PADRÃO DE CORES REC. 709	13
1.1.1 Espaço de cores CIE 1931	14
1.1.2 Padrão Iluminante CIE D65	19
1.2 BALANÇO DE BRANCO	19
1.3 PROCESSAMENTO DIGITAL DE IMAGENS	21
1.3.1 Amostragem e Quantização	23
1.3.2 Filtragem, realce e suavização	24
1.3.3 Média de Múltiplas Imagens.....	25
1.3.4 Coeficiente de correlação de Pearson	26
1.4 <i>SOFTWARE</i>	27
1.4.1 <i>Python</i>	27
1.4.2 <i>OpenCV</i>	27
1.5 <i>ORANGE PI PC Plus H3</i>	28
2 METODOLOGIA	29
3 IMPLEMENTAÇÃO	31
3.1 AMBIENTE DE DESENVOLVIMENTO PARA VALIDAÇÃO DO ALGORITMO.....	31
3.2 VALIDAÇÃO DO ALGORITMO NO <i>SOFTWARE PYCHARM</i>	31
3.3 AMBIENTE DE DESENVOLVIMENTO EM HARDWARE.....	36
3.4 IMPLEMENTAÇÃO DE TESTES EM HARDWARE	36
3.4.1 Estabelecendo uma sessão SSH.....	36
3.4.2 Projeto e desenvolvimento do circuito sinalizador	37
4 RESULTADOS	38
4.1 RESULTADOS EM HARDWARE	40
CONCLUSÃO	46
REFERÊNCIAS BIBLIOGRÁFICAS	48

INTRODUÇÃO

A televisão revolucionou o mundo e hoje é um meio de comunicação de grande importância, mesmo depois do advento da Internet. Antes, como artigo de luxo, não passava de alguns chaviscos difíceis de decifrar, porém com o tempo sua tecnologia evoluiu, o preço diminuiu e ela conquistou seu espaço como meio de comunicação.

Reinventando-se do preto e branco ao Ultra Full HD, os aparelhos televisores oferecem o que há de mais atual em termos tecnológicos na busca de agradar seus consumidores. Para isso, tornam-se necessários alguns testes para validação da qualidade da imagem reproduzida, entre eles o teste de Balanço de Branco (*White Balance*), que por sua vez remove as projeções de cores não realistas, ou seja, ajusta os níveis de cores RGB (*Red, Green, Blue*), para se obter o branco padrão denominado D65, que é o tom da luz branca ao meio dia descrito no espaço de cores CIE 1931 (SCHANDA, 2007).

Atualmente, esse procedimento é realizado por meio de equipamentos como o analisador de cor CA-310 e o espectroradiômetro CS-2000, que são muito caros e complexos, além de exigirem um conhecimento técnico prévio sobre tal ajuste, ou seja, é realizado apenas nas fábricas.

Os ajustes de fábrica seguem uma norma descrita pela VESA (*Video Electronics Standards Association*), em relação ao padrão de medição de displays de tela plana, cuja especificação consiste em:

Medir a luminância e as coordenadas de cromaticidade (opcionalmente a temperatura de cor correlacionada [CCT]) do branco em tela cheia no centro da tela. Com unidades: cd/m^2 , para luminância, nenhuma para as coordenadas de cromaticidade e K para a temperatura de cor. A luminância do branco em tela cheia (o "brilho" da tela) é uma das métricas mais importantes para uma exibição e é usada para várias outras medições e cálculos.

Tendo isso em vista, esse procedimento se torna extremamente importante quando se trata de oferecer um produto de qualidade aos consumidores, além de que ter esse tipo de teste através de uma interface de baixo custo, torna-o mais acessível aos amantes de tecnologia e até mesmo a população mais comum. Portanto, para o desenvolvimento deste projeto, além de serem necessários os conhecimentos adquiridos ao longo do curso, como Circuitos Elétricos, Eletrônica Analógica, Eletrônica Digital, Processamento Digital de Imagens, Sistemas Microprocessados e Microcontroladores, também se faz necessário o estudo de novos conceitos como Espaço de Cores e o processo de Balanço de Branco em televisores.

Visando permitir que esse procedimento seja realizado de maneira mais didática, sem a necessidade da aquisição desses equipamentos e também possa ser feito pelo próprio usuário, este trabalho propõe que tal ajuste seja realizado por meio de processamento digital de imagens, utilizando a técnica da Média de Múltiplas Imagens, onde o programa foi desenvolvido em Python utilizando a biblioteca de visão computacional *OpenCV* e posteriormente foi elaborado um protótipo para realizar esse processamento em tempo real.

Este trabalho está estruturado em quatro seções primárias listadas a seguir:

- a) Referencial teórico: serão apresentados conceitos fundamentais das tecnologias, normas, etapas de teste e equipamentos necessários para o entendimento do projeto.
- b) Metodologia: serão descritos cada passo durante a aquisição das imagens tanto de teste quanto a imagem padrão já ajustada, desenvolvimento da programação em *Python* e implementação da programação no *Orange Pi*.
- c) Implementação: será descrito o que foi feito para realizar a montagem do projeto completo com o objetivo de possibilitar o início da realização de testes.
- d) Análise e interpretação dos resultados: tem o objetivo de apresentar, interpretar e analisar os dados obtidos através da realização de testes após a conclusão do desenvolvimento do dispositivo.

1 REFERENCIAL TEÓRICO

Considerando a problemática e a hipótese deste projeto, é de fundamental importância a aquisição de conhecimentos pertinentes ao balanço de branco em televisores, como o espaço de cores da CIE (*Commission Internationale de l' Eclairage*), que cria padrões internacionais relacionados a luz e cor, além disso, também se faz necessário o estudo de processamento digital de imagens, bem como a técnica de obtenção da imagem média, que segue o Teorema do Limite Central, onde será considerado um número de amostras igual a 30 quadros, conforme descrevem Montgomery e Runger:

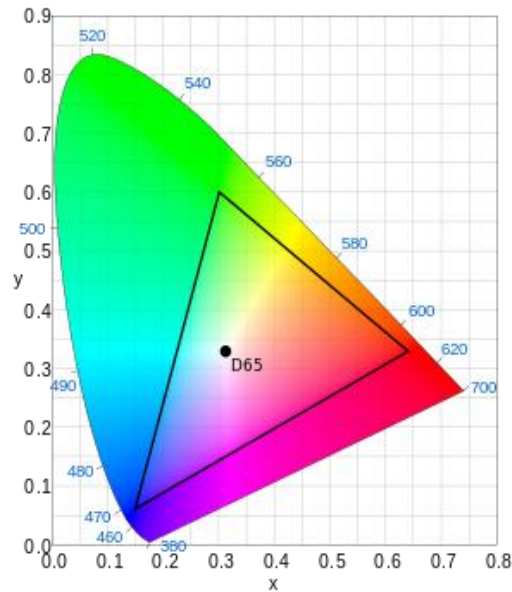
Em muitos casos de interesse prático, se $n \geq 30$, a aproximação normal será satisfatória, independentemente da forma da população. Se $n < 30$, o teorema do limite central funcionará, se a distribuição da população não for muito diferente da normal. (Estatística Aplicada e Probabilidade para Engenheiros, 2009, p. 142).

Do mesmo modo, será descrito sobre microcomputadores, especificamente o *Orange Pi*, que fará o processamento do código com os parâmetros que serão comparados mediante teste. Para o desenvolvimento do software do dispositivo, é necessário conhecer a linguagem de programação *Python* e a biblioteca que será utilizada para visão computacional *OpenCV*.

1.1 O PADRÃO DE CORES REC. 709

O Rec. 709, também conhecido como ITU-R BT. 709, refere-se a sistemas de HDTV e padroniza parâmetros técnicos como a velocidade de quadros por segundo, o espaço de cores e o tipo de gama da imagem. A primeira edição dessa norma foi aprovada em 1990 e está localizada na região mostrada na Figura 1, dentro do espaço de cores CIE 1931 XYZ, onde as cores inclusas na gama de cores Rec. 709 estão dentro do triângulo que conecta as primárias, assim como o iluminante D65 (RECOMMENDATION ITU-R BT.709-6, 2015).

Figura 1 – Padrão Rec. 709 dentro do Diagrama de Cromaticidade CIE 1931 XYZ



Fonte: (BECKER, PACINE e LEMOS, 2018)

O Setor de Radiocomunicação da União Internacional de Telecomunicações (ITU-R), recomenda os parâmetros do espaço de cores RGB conforme a Tabela 1 abaixo.

Tabela 1. Coordenadas de cromaticidade que definem as primárias e o branco padrão.

Espaço colorido	Ponto branco		Primárias					
	x W	y W	x R	y R	x G	y G	x B	y B
ITU-R BT.709	0,3127	0,329	0,64	0,33	0,3	0,6	0,15	0,06

Fonte: (RECOMMENDATION ITU-R BT.709-6, 2015)

1.1.1 Espaço de cores CIE 1931

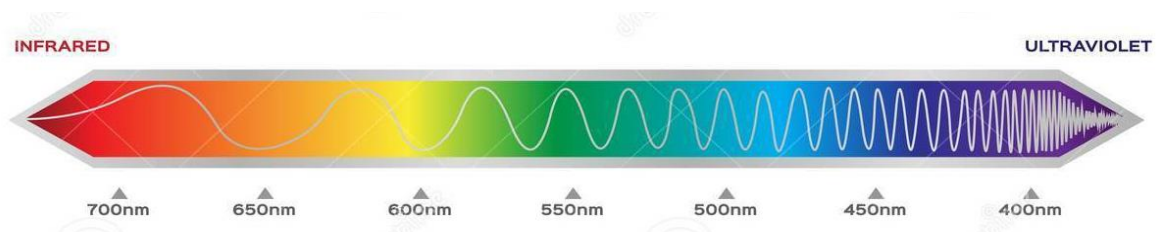
A colorimetria, no que lhe concerne, descreve objetivamente os correlatos físicos da percepção de cores. Em 1931, a Comissão Internacional de Iluminação (CIE) definiu o Observador Padrão e os Iluminantes Padrões para Colorimetria dentro do Espaço de Cores CIE 1931 XYZ, cujos fundamentos ainda são válidos, e muitos métodos de avaliação de cores industriais são baseados nesse padrão (SCHANDA, 2007).

O principal fundamento da luz é seu comprimento de onda. Os seres humanos podem perceber a luz com ondas tão pequenas quanto 380nm e tão grandes quanto 780 nm. O sol emite luz através de muitos comprimentos de onda e os objetos ao nosso redor refletem alguns e

absorvem outros. A luz, ao atravessar os meios ópticos oculares, excita moléculas fotossensíveis dos fotorreceptores da retina que, por sua vez, iniciam o processo de codificação da informação presente nos raios luminosos até que após extenso processamento neural em vários níveis da retina, do tálamo e do córtex cerebral, ocorre a percepção da cor (BACKHAUS, 1998).

Como mostra a Figura 2, a sensação de luz é produzida pela radiação visível de 380 nm a 780 nm, onde a região de onda curta desta radiação produz geralmente a sensação de luz azul, radiação com comprimentos de onda entre 520 nm e 550 nm é vista como luz verde, e acima de cerca de 650 nm percebe-se que a luz geralmente é de cor vermelha (SCHANDA, 2007).

Figura 2. Espectro visível da Luz



Fonte: (DREAMSTIME, 2015)

Existem dois métodos fundamentais para produzir estímulos coloridos: mistura de cores aditivas e subtrativas. Em misturas de cores aditivas, as luzes são misturadas, como em displays de TV em cores, onde a sensação de cor aos olhos é produzida pela mistura aditiva de minúsculas luzes vermelhas, verdes e azuis, onde os pontos são tão próximos que os olhos são incapazes de determiná-los espacialmente e vemos a mistura das luzes. De outro modo, na mistura de cores subtrativas, os corantes removem alguma parte do espectro visível e a sobreposição de vários corantes de diferentes concentrações mudará a cor da luz transmitida.

Na colorimetria básica, a descrição dos resultados das experiências de correspondência de cores, é construída com base na mistura de cores aditivas pois suas leis são mais simples do que as da mistura de cores subtrativas. Sendo as quatro leis empíricas básicas:

- 1- Toda impressão de cor pode ser analisada em três elementos matematicamente determináveis, o matiz, o brilho da cor e o brilho do branco misturado.
- 2- Em segundo lugar, assumimos que, se uma das duas luzes de mistura estiver continuamente alterada (enquanto a outra permanece inalterada), a impressão da luz mista também é continuamente alterada.

3- Duas cores, ambas com o mesmo matiz e a mesma proporção de branco misturado, também dão cores misturadas idênticas, não importando de que cores homogêneas elas possam ser compostas.

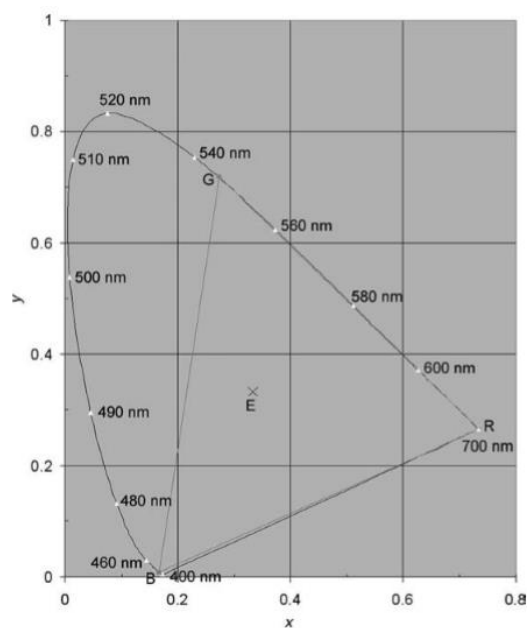
4- A intensidade total de qualquer mistura é a soma das intensidades das luzes misturadas.

A colorimetria CIE baseia-se nestas leis empíricas, desde que as condições de observação, a exposição prévia do olho do observador, e pessoa que faz a correspondência é mantida a mesma segundo Grassmann (1853, apud SCHANDA, 2007).

O sistema de cores CIE 1931 XYZ é conhecido pelo seu diagrama de cromaticidade, mostrado na Figura 3. A forma tridimensional do sistema pode ser obtida calculando-se as coordenadas do terceiro eixo (Y), o eixo de luminância, somadas às coordenadas dos dois eixos de cromaticidade (x e y).

No diagrama estão representadas não apenas as cores puras ou monocromáticas como, também, as misturas delas para cada coordenada cromática. As coordenadas para as cores puras são representadas no locus espectral, o qual ocupa mais de dois terços da borda do diagrama, correspondendo aos comprimentos de onda entre 380 nm e 780 nm. O branco, por sua vez, está localizado em sua porção mais central ($x = 0,33$; $y = 0,33$) e a sua determinação resulta da combinação dos três comprimentos de onda adotados como primárias, 700 nm, 546,1 nm e 435,8 nm. Esses valores foram obtidos em 1931 segundo Guild (1931, apud SIMÕES, 2004).

Figura 3. Diagrama de cromaticidade xy do sistema tricromático CIE 1931



Fonte: (SCHANDA, 2007)

No sistema, X, Y e Z representam cada um dos valores absolutos da mistura de cores das três primárias necessárias para a equiparação com cada uma das cores espectrais. Os valores de x, y e z representam as coordenadas de cromaticidade e são obtidas pelas equações 1, 2 e 3:

$$x = \frac{X}{X + Y + Z} \quad (1)$$

$$y = \frac{Y}{X + Y + Z} \quad (2)$$

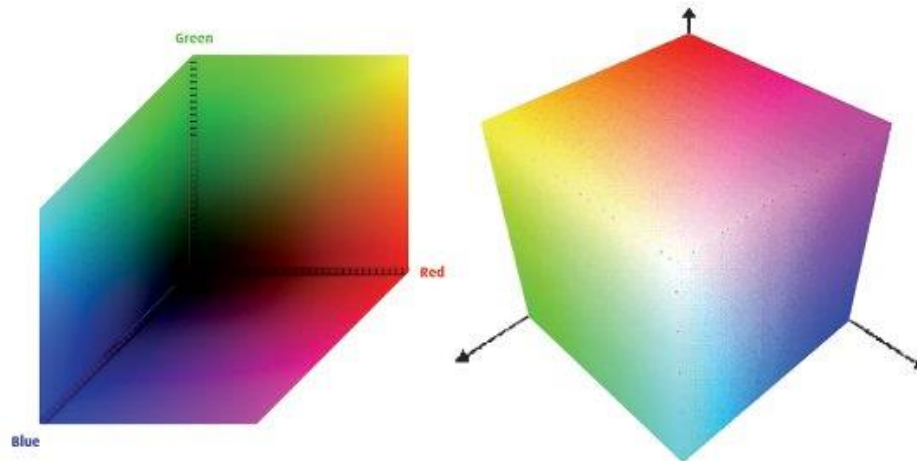
$$z = \frac{Z}{X + Y + Z} \quad (3)$$

Onde $x + y + z = 1$, portanto, é suficiente descrever a cromaticidade com dois números, geralmente x e y, porém a terceira, z, pode ser obtida por diferença, uma vez que $z = 1 - x - y$. Estas três coordenadas correspondem às proporções de cada uma das três primárias para constituir cada cor espectral. Desta forma, as cores têm informações nas três dimensões, enquanto o branco e o preto estão apenas no eixo Y que representa o nível de luminância expresso em cd/m^2 . Não se deve esquecer, no entanto, que um estímulo de cor pode ser descrito apenas com três grandezas características, assim, se x, y são usados, é preciso citar também Y.

Evidentemente as TVs utilizam o sistema RGB, já que nenhuma transformação é necessária para exibição de informações na tela. Dessa forma, o sistema pode ser visualizado como um cubo com os três eixos correspondentes a vermelho, verde e azul, como mostra a Figura 4. O canto inferior, quando vermelho = verde = azul = 0, é preto, enquanto o canto superior oposto, onde vermelho = verde = azul = 255 (para um sistema de exibição de 8 bits por canal), é branco.

Assim, a conversão de valores de pixel de imagem RGB para os valores de estímulo CIE XYZ pode ser obtida usando um processo de dois estágios, onde primeiramente é preciso calcular a relação entre os valores de pixel da imagem de entrada e a intensidade exibida. Essa relação é uma função de transferência, muitas vezes simplificada para gama. As funções de transferência geralmente diferem para cada canal, portanto, são medidas de maneira independente.

Figura 4. Plano cartesiano de cores RGB e cubo RGB



Fonte: (RIBEIRO, 2005)

O segundo estágio é transformar os valores exibidos em vermelho, verde e azul para os valores tristimulares CIE. Isso é mais facilmente realizado usando a transformação de matriz mostrada na equação 4:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} X_r & X_g & X_b \\ Y_r & Y_g & Y_b \\ Z_r & Z_g & Z_b \end{bmatrix} * \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (4)$$

Onde X, Y, Z são os valores de estímulo CIE desejados, R, G, B são os valores RGB exibidos obtidos das funções de transferência.

Para o Iluminante D65 os valores de X, Y e Z já são conhecidos e são mostrados na Tabela 2, bem como outros iluminantes normalmente usados.

Tabela 2. Valores de estímulos e coordenadas de cromaticidade para seis iluminantes selecionados

Iluminantes	X	Y	Z	x	Y
Iluminante CIE A	109,85	100	35,58	0,44758	0,40745
Iluminante CIE D65	95,04	100	108,88	0,31272	0,32903
Iluminante CIE D50	96,4	100	82,5	0,34567	0,3585
Iluminante CIE D55	95,68	100	92,14	0,33243	0,34744
Iluminante CIE D75	94,97	100	122,61	0,29903	0,3148
Iluminante C	98,07	100	118,22	0,31006	0,46089

Fonte: (SCHANDA, 2007)

1.1.2 Padrão Iluminante CIE D65

Iluminante é uma luz definida por sua energia espectral, porém, pode originar de uma fonte física ou natural. A luz de uma lâmpada em seu ambiente de trabalho ou casa origina então de uma fonte iluminante, manipulável, que pode ter sua energia espectral medida com equipamento adequado. O CIE estabelece o padrão dessa energia espectral que cada modelo de iluminante deve ter para sua comercialização. Um iluminante natural, porém, nem sempre pode ter sua energia espectral medida.

Para o órgão visual humano, a iluminação "natural" é a luz do dia, portanto, parece evidente que iluminantes fossem selecionados durante as fases da luz do dia. Entre essas fases, a CIE selecionou o D65 e segundo Schanda (2007):

O iluminante padrão D65 deve representar luz diurna média e ter uma temperatura de cor correlacionada de aproximadamente 6500 K. O iluminante padrão D65 da CIE deve ser usado em todos os cálculos colorimétricos que exigem luz diurna representativa, a menos que haja motivos específicos para usar um iluminante diferente. Sabe-se que variações na distribuição da energia espectral relativa da luz do dia ocorrem, particularmente na região espectral ultravioleta, em função da estação do ano, da hora do dia e da localização geográfica. No entanto, o iluminante padrão D65 deve ser usado, dependendo da disponibilidade de informações adicionais sobre essas variações.

Tabela 3. CCT nominal e real dos Iluminantes CIE diurnos

Iluminante	CCT Nominal, K	CCT Real (Aproximada), K
Iluminante CIE D50	5000	5003
Iluminante CIE D55	5500	5503
Iluminante CIE D65	6500	6504
Iluminante CIE D75	7500	7504

Fonte: (SCHANDA, 2007)

Os padrões D50, D55 e D75, mostrados na Tabela 3, são definidos como iluminantes diurnos preferidos caso o D65 não for aplicável. A indústria de artes gráficas, por exemplo, selecionou o D50 como seu iluminante de referência, por estar entre a luz natural média e a luz incandescente. Outras tecnologias podem usar diferentes iluminantes de referência.

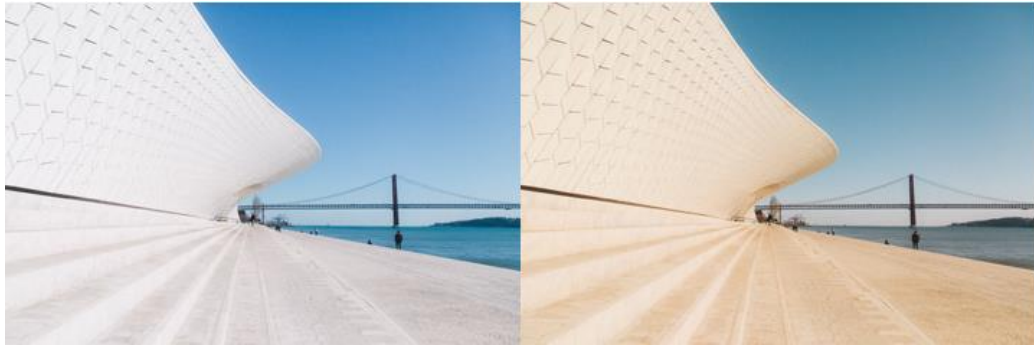
1.2 BALANÇO DE BRANCO

O balanço de branco, como dito anteriormente, é um processo que busca a remoção de cores não realistas de uma imagem. Esse conceito também é bastante utilizado em fotografia,

uma vez que esse ajuste faz com que a imagem capturada por uma câmera pareça natural, mesmo que a fonte de luz não seja.

Portanto, para que esse ajuste seja feito corretamente deve-se levar em consideração a temperatura de cor de uma fonte de luz, ou seja, quão quente ou fria ela é. A Figura 5 mostra os efeitos devido ao balanço de branco correto e incorreto.

Figura 5. (a) Balanço de branco correto. (b) Balanço de branco incorreto.



Fonte: (VIERO, 2018)

Essa definição é análoga quando se trata de televisores, uma vez que diferentes fontes de luzes transmitem diferentes temperaturas de cor dependendo do ambiente e para isso houve a necessidade de se padronizar a temperatura de cor de 6500 K para obtenção do branco padrão.

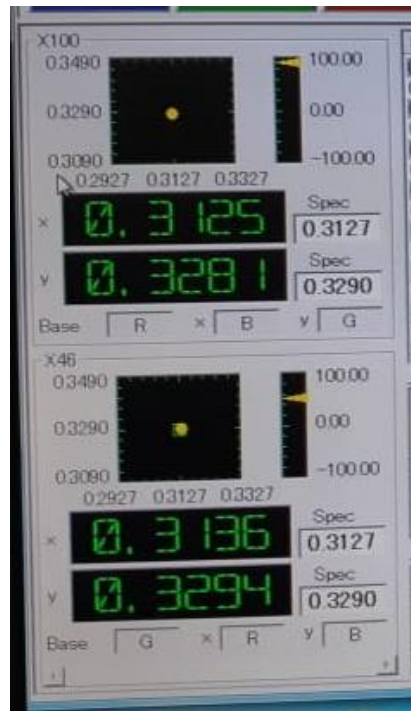
Como já mencionado, a maioria dos televisores utiliza o padrão Rec. 709 que define os parâmetros para realizar o balanço de branco, ou seja, as coordenadas do branco padrão denominado Iluminante D65, sendo elas $x = 0,3127$ e $y = 0,3290$.

No teste realizado em fábrica é feita uma leitura da tela da TV quando é exibida uma imagem em branco. Essa leitura é feita por meio do Analisador de Cor, sendo comumente usado o CA-310, que se comunica com um software que contém os parâmetros do balanço de branco pré-estabelecidos, como mostrado na Figura 6, uma vez que cada fabricante utiliza um software diferente.

A comunicação serial entre os equipamentos e a TV é feita através do protocolo RS232 que por sua vez envia e recebe dados através de um USB serial e um cabo de interface.

Quando a leitura das coordenadas x e y da tela em branco é diferente da especificação, o software obriga a TV a ajustar seus parâmetros até que o ajuste esteja dentro da tolerância também pré-estabelecida, ou seja, o analisador de cor lê continuamente os parâmetros da tela e enquanto o *software* envia um comando de auto ajuste para o televisor.

Figura 6. Software de ajuste de Balanço de Branco



Fonte: Própria

Faz-se também o ajuste das coordenadas para cada temperatura de cor, sendo elas normal, quente e frio, uma vez que o software contém o mapa de registradores da televisão e pode acessá-los para que essa mudança de temperatura ocorra durante o teste. Ao final do ajuste o software sinaliza que a TV está pronta para uso.

1.3 PROCESSAMENTO DIGITAL DE IMAGENS

Primeiramente, é necessário entender como é formada uma imagem digital, para que assim, o objetivo de processar informações da mesma seja alcançado.

O termo “imagem” refere-se à função bidimensional de intensidade da luz $f(x, y)$, onde x e y correspondem às coordenadas espaciais e o valor de f em qualquer ponto (x, y) é proporcional ao brilho (ou níveis de cinza) da imagem naquele ponto. (GONZALES e WOODS, 1996).

O nível de cinza de um ponto da imagem vai depender da quantidade de luz que incide sobre a cena. A função $f(x, y)$ é caracterizada pela quantidade de iluminação que incide na cena e pela quantidade de iluminação que reflete através do objeto da cena. Essas componentes são chamadas de iluminação e refletância e são designados por $i(x, y)$ e $r(x, y)$, respectivamente. O

produto das duas funções formam $f(x, y)$, conforme na Equação 5 (FERNANDES COSTA, 2008).

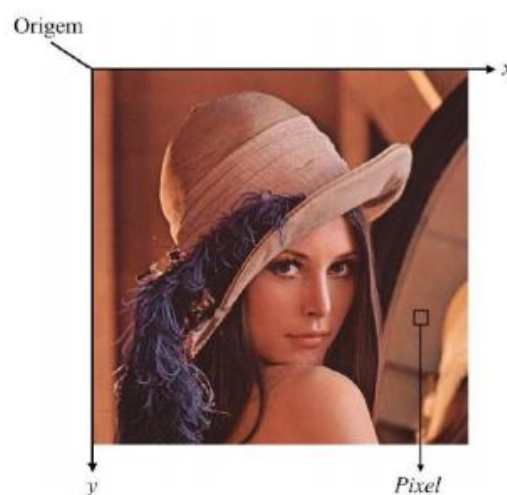
$$\begin{aligned} f(x, y) &= i(x, y) \cdot r(x, y) & (5) \\ 0 < i(x, y) &< \infty \\ 0 < r(x, y) &< 1 \end{aligned}$$

Uma “imagem digital” é uma imagem $f(x, y)$ discretizada tanto nas coordenadas espaciais como no valor do nível de cinza. Uma imagem digital pode ser considerada uma matriz na qual seus índices de linha e coluna indicam um ponto da imagem, por exemplo, na forma de um tabuleiro de xadrez, sendo que o valor correspondente ao elemento da matriz indica o nível de cinza nesse ponto.

Os elementos dessa matriz digital são chamados de elementos da imagem ou “pixel” do inglês *picture elements*, e pode ser visto na Figura 7. Um pixel é o elemento básico em uma imagem, a sua forma mais comum é retangular ou quadrada, apesar de ter dimensões finitas na representação de uma imagem digital (GONZALEZ e WOODS, 2002).

Explicado o que é uma imagem digital, pode-se dizer agora que o Processamento Digital de Imagens abrange um conjunto de hardware e software e fundamentos teóricos para obtenção da informação de interesse da imagem.

Figura 7. Representação de uma imagem digital

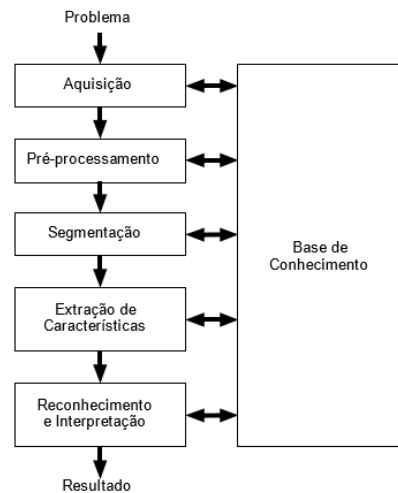


Fonte: (SÁNCHEZ- FERREIRA, 2012)

Essa interconexão pode ser observada na Figura 8, onde a primeira etapa trata-se da aquisição da imagem, seguida do pré-processamento, se necessário, que serve para melhorar a

imagem por meio de filtros de correções de ruídos e distorções, em seguida vem a segmentação que tem por objetivo dividir uma imagem em suas unidades significativas, ou seja, nos objetos de interesse que a compõem. Como o nome da etapa já diz, a extração de características busca através de descritores, extrair características das imagens resultantes da segmentação, o reconhecimento e interpretação atribui um rótulo a um objeto baseado em suas características traduzidas por seus descritores e por fim a base de conhecimento guia a operação de cada módulo de processamento e controla a interação entre eles (MARQUES FILHO e VIEIRA NETO, 1999).

Figura 8. Etapas do Processamento Digital de Imagens



Fonte: (MARQUES FILHO e VIEIRA NETO, 1999)

1.3.1 Amostragem e Quantização

Amostragem e quantização são processos de digitalizar, tanto espacialmente quanto em amplitude, uma determinada imagem.

Basicamente, podemos representar uma imagem na computação, através de uma matriz M por N , onde cada componente f da mesma representa um elemento de imagem ou *pixel*.

Portanto, para o processo de digitalização é necessário decidir alguns parâmetros, por exemplo, os valores para M , N e o número de níveis de intensidade. Não há restrições colocadas para M e N que sejam o de serem números inteiros positivos. Entretanto, devido a considerações de armazenamento e hardware de quantização, o número de níveis de intensidade tipicamente é uma potência de 2, assim $G = 2^m$. Dessa forma, quanto maiores os parâmetros de M , N e G , maior e melhor será a resolução da imagem. Porém, quanto maior for a resolução, maior será o

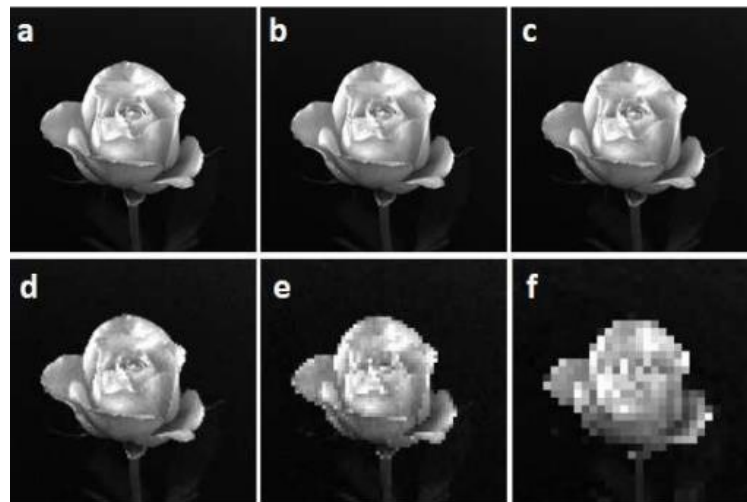
processamento e espaço para armazenamento da imagem (MARQUES FILHO e VIEIRA NETO, 1999).

O aumento do processamento ocorrerá devido às técnicas de PDI terem que percorrer uma matriz maior. O aumento de espaço para armazenamento dá-se devido à equação da quantidade de b (bits) necessários para armazenar a imagem, dada pela Equação 6 (MARQUES FILHO e VIEIRA NETO, 1999).

$$b = N \times M \times m \quad (6)$$

Assim, é fácil notar o impacto desses parâmetros na Figura 9, onde ocorre à diminuição da taxa de amostragem.

Figura 9 – Diminuição da taxa de amostragem N de imagens respectivas



Fonte: (GONZALEZ e WOODS, 1996)

1.3.2 Filtragem, realce e suavização

O principal objetivo dos filtros é processar uma imagem de modo que a imagem resultante seja mais adequada que a imagem original para uma aplicação específica, ou seja, as técnicas utilizadas são orientadas a um problema específico que se deseja resolver, sendo a interpretação dependente do observador.

Existem diversos filtros de suavização e realce, sendo normalmente classificados em duas categorias: espacial e no domínio da frequência. Os filtros no domínio espacial baseiam-se no uso de máscaras de convolução adequadas para o objetivo em questão, como a remoção de ruídos ou borramento, que operam diretamente sobre a matriz de pixels que é a imagem

digitalizada. Em contrapartida, os métodos que atuam no domínio da frequência se baseiam na modificação da transformada de Fourier da imagem.

Neste trabalho, será abordado sobre o filtro da média de múltiplas imagens que foi considerado o mais adequado para a presente aplicação.

1.3.3 Média de Múltiplas Imagens

Supondo a existência de M imagens ruidosas adquiridas em instantes de tempo diferentes, onde cada uma pode ser representada por uma função do tipo $g(x,y) = f(x,y) + j(x,y)$, onde $f(x,y)$ é a imagem original e $j(x,y)$ é um padrão de ruído aditivo de média zero e descorrelacionado, então pode-se calcular uma imagem média por meio da Equação 7 na qual o ruído terá sido minimizado. (MARQUES FILHO e VIEIRA NETO, 1999).

$$\bar{g}(x,y) = \frac{1}{M} \sum_{i=1}^M g_i(x,y) \quad (7)$$

Podendo-se afirmar também que:

$$E\{\bar{g}(x,y)\} = f(x,y) \quad (8)$$

$$\sigma_{\bar{g}(x,y)} = \frac{1}{\sqrt{M}} \sigma_{\eta(x,y)} \quad (9)$$

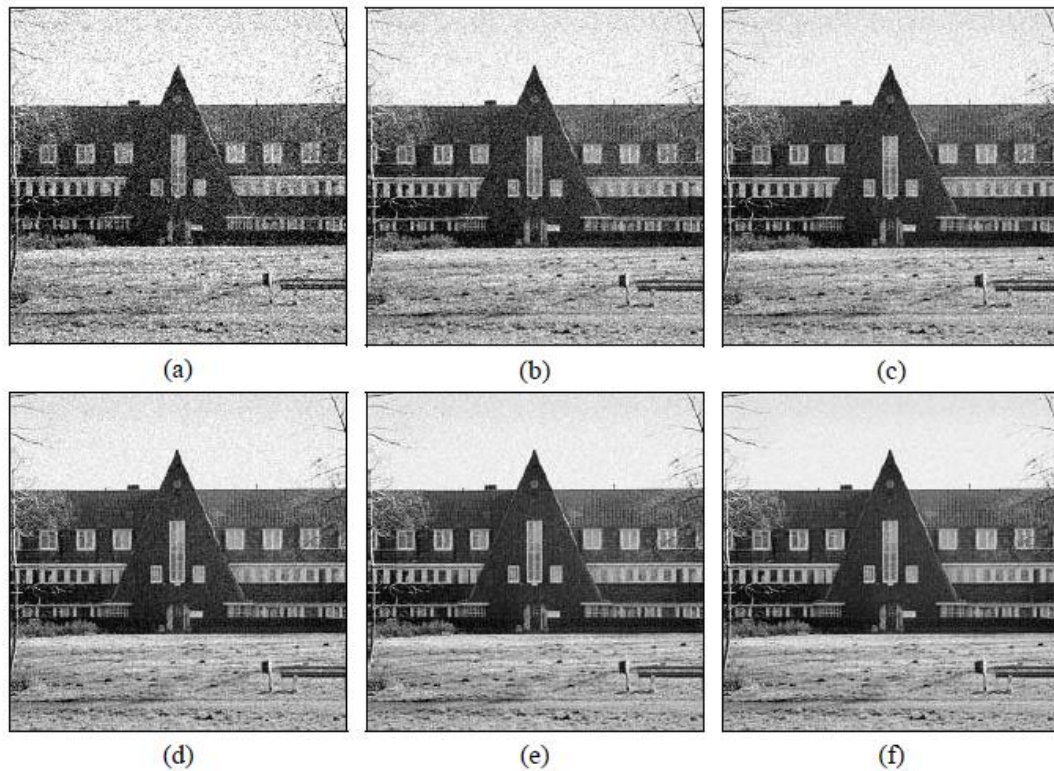
$$\sigma^2_{\bar{g}(x,y)} = \frac{1}{M} \sigma^2_{\eta(x,y)} \quad (10)$$

Onde $E\{\bar{g}(x,y)\}$ é o valor esperado de $\bar{g}(x,y)$, $\sigma^2_{\bar{g}(x,y)}$ e $\sigma^2_{\eta(x,y)}$ são as variâncias da imagem filtrada e do ruído aditivo, e por fim $\sigma_{\bar{g}(x,y)}$ e $\sigma_{\eta(x,y)}$ seus respectivos desvios-padrão. A partir dessas equações pode-se concluir que quanto maior o valor de M , menor será a variância, assim como o desvio padrão, dos pixels de $\bar{g}(x,y)$ e mais essa imagem se aproximará de $f(x,y)$.

A Figura 10 apresenta um exemplo da técnica da média de múltiplas imagens para redução de ruídos, onde foram utilizados $M = 2, 4, 8, 16$ e 32 imagens, notando-se uma rejeição de ruídos maior na utilização de 32 imagens para o cálculo da imagem média.

Dessa forma, conclui-se que essa técnica tem um resultado satisfatório quando o número de imagens utilizadas para cálculo é significativo, uma vez que obedece ao Teorema do Limite Central, que estabelece que a soma de um grande número de termos representando ruídos aleatórios tende a produzir um ruído resultante do tipo gaussiano e independente dos tipos dos ruídos incluídos naquela soma. (MARQUES FILHO e VIEIRA NETO, 1999).

Figura 10. Técnica da média de múltiplas imagens para diferentes valores de M



Fonte: (MARQUES FILHO e VIEIRA NETO, 1999)

1.3.4 Coeficiente de correlação de Pearson

O coeficiente de correlação de Pearson (ρ) mede o grau da correlação linear entre duas variáveis quantitativas X e Y. Sendo sua fórmula a seguinte:

$$\rho = \frac{1}{n-1} \sum \left(\frac{x_i - \bar{X}}{s_x} \right) \left(\frac{y_i - \bar{Y}}{s_y} \right) \quad (11)$$

Onde x_i representa o valor da observação, \bar{X} representa a média e s_x indica o valor do desvio padrão. O mesmo deve ser feito para Y. Depois disso, o pesquisador deve somar o produto cruzado dos valores padronizados de X e Y.

O coeficiente de correlação de Pearson varia de -1 a 1, uma vez que o sinal positivo ou negativo indica direção do relacionamento e o valor sugere a força da relação entre as variáveis. Uma correlação perfeitamente positiva (1), por exemplo, indica que se o valor de uma variável aumenta, o mesmo acontece com a outra. Já numa correlação perfeitamente negativa (-1) ocorre o contrário, quando uma variável aumenta a outra diminui. O valor 0 indica que não existe correlação (FILHO e JÚNIOR, 2009).

1.4 SOFTWARE

Em uma das etapas do projeto, foi desenvolvido um programa cuja lógica determinará o funcionamento do dispositivo. Esse programa será desenvolvido na linguagem *Python* utilizando a biblioteca *OpenCV*.

1.4.1 Python

Python é uma linguagem de programação de alto nível, orientada a objetos e com semântica dinâmica. Ela possui um framework poderoso e robusto que permite um desenvolvimento rápido de websites e por isso é bastante utilizada no desenvolvimento de aplicações web. Sua principal finalidade é incentivar a produtividade e legibilidade de códigos, através de sua velocidade e estilo compacto de escrever (BLOG IMPACTA, 2018).

Ela também é uma linguagem que possui vários módulos prontos, como buscar dados em um banco de dados, ler uma página na internet, criar planilhas, etc. Em razão disso, *Python* ganhou popularidade na comunidade científica e aumentou bastante sua demanda no ano de 2018. Além disso, é compatível com a maioria dos sistemas operacionais e tem a capacidade de auxiliar outras linguagens (CAELUM, 2017)

Por fim, a linguagem *Python* possui todas as características que programadores experientes e iniciantes buscam, como fácil aprendizado, simples de programar, sintaxe intuitiva, documentação extensa, *Open source*, modularização e é multiplataforma.

1.4.2 OpenCV

OpenCV (Open Source Computer Vision Library), é uma biblioteca, originalmente desenvolvida pela Intel, livre ao uso acadêmico voltada para o desenvolvimento de aplicações na área de Visão Computacional, que nada mais é que a visão de máquinas, onde é possível

obter informações de imagens e utilizar algoritmos computacionais para descrever e analisar o conteúdo de qualquer imagem digitalizada. Essa área possui diversas aplicações, entre elas controle de qualidade de processos, orientação de robôs, navegação de veículos e análises em células (EMBARCADOS, 2017).

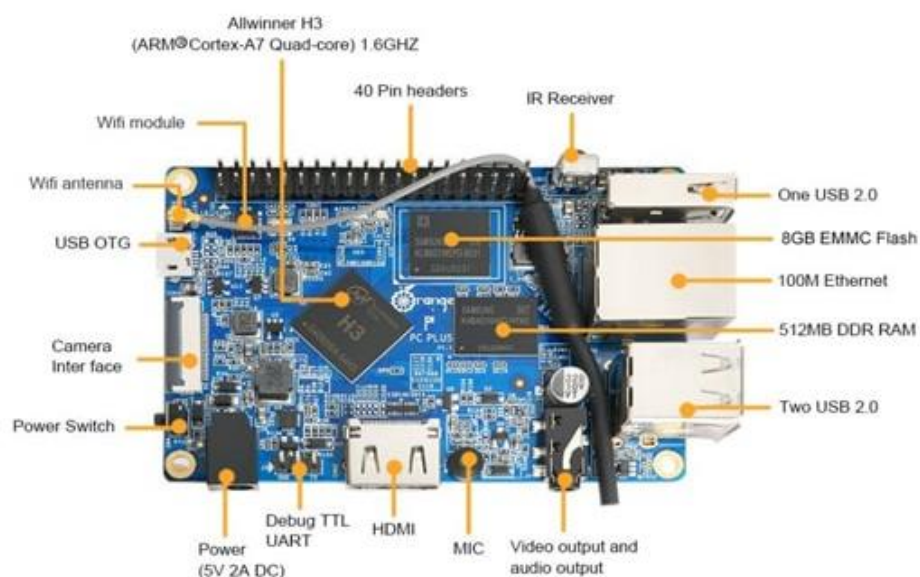
A biblioteca OpenCV foi escrita em C e C ++ e roda em Linux, Windows e Mac OS X e além disso, existe um desenvolvimento ativo nas interfaces para Python, Ruby, Matlab e outras linguagens. Possui módulos de Processamento de Imagem e Vídeo, Álgebra Linear, controle de mouse, teclado, webcam, além de mais de 350 algoritmos de visão computacional com processamento em tempo real (BRADSKI e KAEHLER, 2008).

Um dos objetivos da OpenCV é fornecer uma infraestrutur de visão computacional simples de usar que ajude as pessoas a criar rapidamente aplicativos de visão bastante sofisticados, além de avançar na área de visão computacional, visto que é uma biblioteca de código aberto e todos podem colaborar com sua otimização.

1.5 *ORANGE PI PC Plus H3*

O *Orange Pi PC Plus H3* é um microcomputador de baixo custo, que está no grupo dos SBC (Single-Board Computers) assim como o Raspberry Pi, BeagleBone, CubieBoard, etc. Seu principal diferencial entre os demais citados, são suas características de hardware e preço atraentes, custando aproximadamente \$25. Na Figura 11 pode-se ter uma visão do hardware. (EMBARCADOS, 2017).

Figura 11. *Orange Pi PC Plus H3*



Fonte: (Orange Pi, 2017)

O hardware do *Orange Pi PC Plus H3* é composto por uma CPU Quad-core, aceleração gráfica por hardware, 1 GB de memória RAM, 3 portas USB Host, 1 micro-USB OTG, porta Ethernet, WI-FI, e memória Flash de 8GB.

As finalidades mais comuns dadas ao *Orange Pi PC Plus H3* são computadores, servidores wireless, videogames, centrais de mídia, projetos de IoT (*Internet of Things*) de robótica e muitas outras, uma vez que o mesmo é *open source*. Suporta as distribuições Linux, Debian Image e Ubuntu Xenial, feitas com base no Armbian, um fabricante de distribuições Linux mais leves para ARM (Orange Pi, 2017).

2 METODOLOGIA

O presente trabalho é uma Pesquisa Aplicada, cujo objetivo é a realização de uma pesquisa exploratória e explicativa sobre o material bibliográfico adquirido. O método de abordagem utilizado será o hipotético-dedutivo e o método de procedimento de elaboração é o monográfico. Para coleta de dados será utilizada documentação indireta e análise e interpretação de dados quantitativa, assim como consulta direta com profissionais técnicos do setor de engenharia.

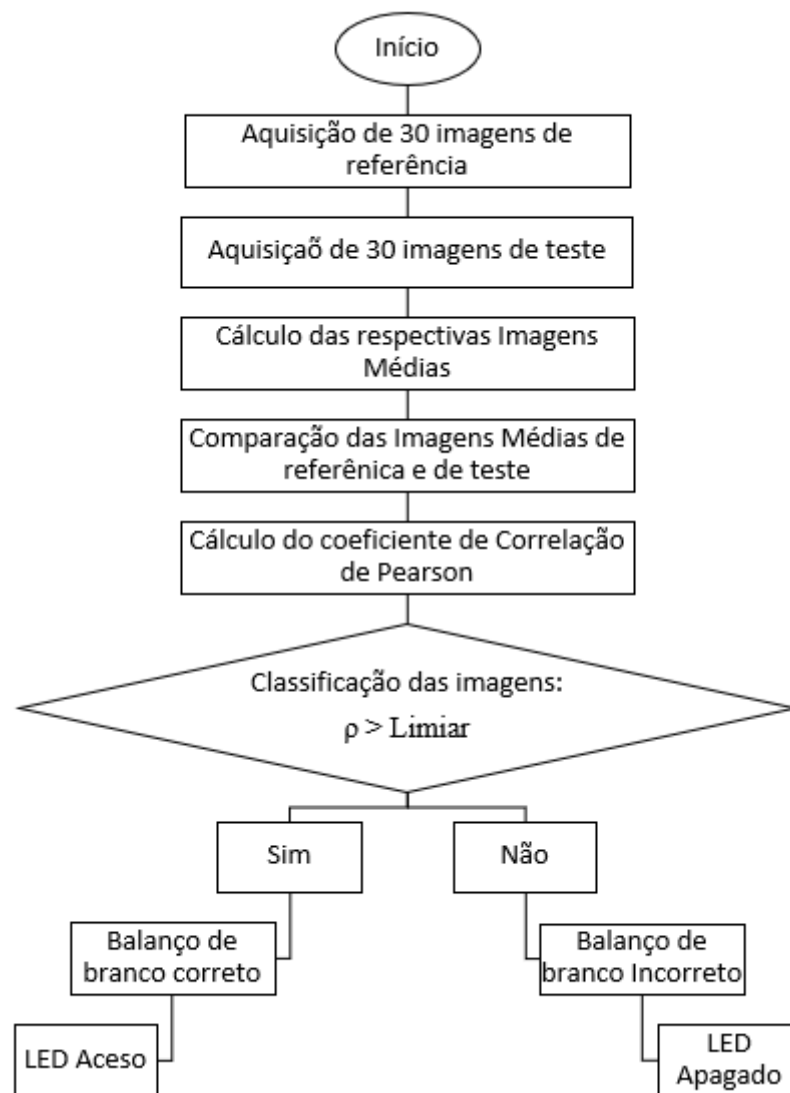
A princípio, foram realizadas pesquisas relativas a Balanço de Branco e tudo que está relacionado a esse teste, como espaço de cores, temperatura de cor e padrões a serem seguidos para realização do mesmo.

Também foi feita a validação do algoritmo em *Python* utilizando o *software PyCharm*, onde será realizada a obtenção das imagens de referência e de teste por meio da webcam do notebook e posteriormente será calculada a imagem média das 30 imagens capturadas para que seja feita comparação pelo coeficiente de Pearson.

Da mesma forma foi analisada a técnica da média de múltiplas imagens, utilizada em processamento digital de imagens para tratamento das imagens padrão e de teste, que será validada através do *software* desenvolvido em *Python*.

Logo após as pesquisas e validações foi realizada a compra dos componentes a serem utilizados na implementação do protótipo, e em seguida será iniciada a programação do código no *Orange Pi PC Plus H3*. A próxima etapa será a montagem da plataforma física, com uma saída indicada por um LED que sinaliza se a imagem está ajustada ou não. A Figura 12 mostra o fluxograma a ser seguido pelo algoritmo embarcado do dispositivo.

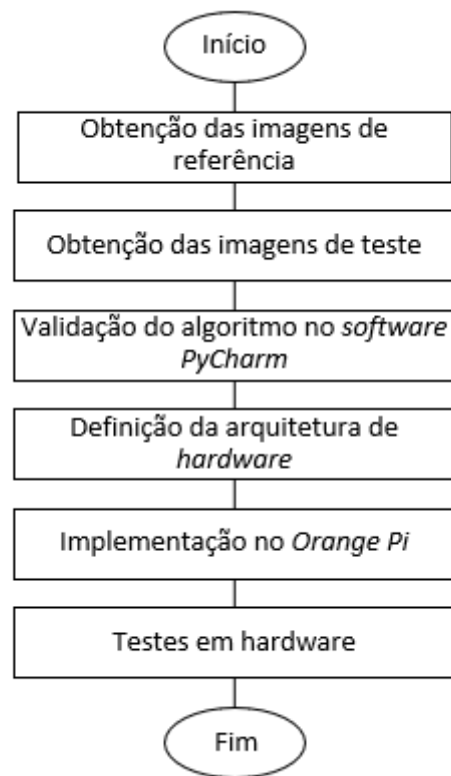
Figura 12. Fluxograma das etapas do algoritmo



O último passo foi acrescentado devido algumas dificuldades encontradas ao longo do projeto, como documentos confidenciais, impossibilitando assim o acesso à memória da TV. Porém para simular o comando de auto ajuste da TV foi implementada a etapa descrita anteriormente. Com o resultado final, foram extraídos os resultados dos testes, para então ser feita uma comparação com o procedimento anterior, a qual foi utilizada como métrica de avaliação. A Figura 13 mostra o fluxograma de funcionamento básico do projeto.

Por fim, para verificar a eficiência da aplicação do projeto, o dispositivo foi utilizado em testes de Balanço de branco em uma empresa dedicada ao ramo de eletrodomésticos do polo industrial de Manaus e os resultados obtidos foram analisados e comparados aos obtidos em testes habituais.

Figura 13. Fluxograma das etapas gerais do projeto



3 IMPLEMENTAÇÃO

Neste capítulo será abordada a forma de implementação do projeto, seguindo as etapas: Ambiente de desenvolvimento, Validação do algoritmo no *software PyCharm* e Implementação e testes em hardware (*Orange Pi*).

3.1 AMBIENTE DE DESENVOLVIMENTO PARA VALIDAÇÃO DO ALGORITMO

O ambiente de desenvolvimento desta fase do projeto foi o seguinte:

- Computador: Notebook HP, com processador Intel® Core™ i5-7200U 2.5 GHz 4GB HD 500GB;
- Sistema Operacional: Windows 10;
- Software para desenvolvimento das etapas do projeto e realização de testes: *PyCharm*;

3.2 VALIDAÇÃO DO ALGORITMO NO *SOFTWARE PYCHARM*

Para fazer a validação do algoritmo foi utilizado o *software PyCharm*, uma IDE (Ambiente Integrado de Desenvolvimento) com muitas funcionalidades que auxiliam no desenvolvimento de código além de possuir a capacidade de rodá-lo.

No bloco a seguir é feita a leitura de 30 imagens a partir da *webcam* do *notebook*. Onde a classe *VideoCapture* captura a imagem armazena no objeto *camera* e depois salva a imagem em *img* com o nome do arquivo “ImagemRef_a.jpg” isso é feito 30 vezes. Essas imagens servirão para cálculo da imagem média que será utilizada como referência, ou seja, com o balanço de branco já ajustado.

Figura 14. Obtenção de 30 imagens de referência

```

1  import cv2
2
3  a = 0
4  b = 0
5
6  camera_port = 0
7  camera = cv2.VideoCapture(camera_port)
8
9  while (a < 30):
10     retval, img = camera.read()
11     cv2.imwrite("ImagemRef_"+ str(a) + ".jpg", img)
12     a += 1
13     camera.release()

```

A seguinte etapa realiza o cálculo da imagem média de referência a partir das 30 imagens capturas acima. Primeiro será criada uma lista definida como *new_image* que receberá 3 matrizes (RGB) de tamanho 480 x 640 (resolução da imagem), sendo essas matrizes preenchidas por zeros. Em seguida serão lidos cada um dos 30 frames e cada um será somado à nova imagem criada (matrizes de zeros). Ao final a nova imagem resultante da soma é dividida por 30 (número de frames), obtendo-se então a imagem média de referência.

Figura 15. Cálculo da Imagem Média

```

15  new_image = []
16
17  for i in range(480):
18     new_image.append([])
19  for j in range(640):
20     new_image[i].append([])
21     for k in range(3):
22         new_image[i][j].append(0);
23
24
25  for i in range(30):
26     frame = cv2.imread("ImagemRef_" + str(i) + ".jpg")
27     new_image = new_image + frame
28
29  for i in range(480):
30     for j in range(640):
31         for k in range(3):
32             new_image[i][j][k] = new_image[i][j][k] / 30
33
34  cv2.imwrite('ImagemMediaRef.jpg', new_image)

```

Os mesmos códigos são utilizados para capturar as 30 imagens de teste e a imagem média de teste, conforme mostrado na Figura 16.

Figura 16. Captura das imagens de teste e cálculo da imagem média de teste

```

39 while (b < 30):
40     retval2, img2 = camera.read()
41     cv2.imwrite("ImagemTest_" + str(b) + ".jpg", img2)
42     b += 1
43     camera.release()
44
45     new_image2 = []
46
47     for i in range(480):
48         new_image2.append([])
49         for j in range(640):
50             new_image2[i].append([])
51             for k in range(3):
52                 new_image2[i][j].append(0)
53
54     for i in range(30):
55         frame2 = cv2.imread("ImagemTest_" + str(i) + ".jpg")
56         new_image2 = new_image2 + frame2
57
58     for i in range(480):
59         for j in range(640):
60             for k in range(3):
61                 new_image2[i][j][k] = new_image2[i][j][k] / 30
62
63     cv2.imwrite('ImagemMediaTest.jpg', new_image2)

```

Por fim, é realizada a comparação entre as duas imagens médias, a de referência e a de teste, por meio do coeficiente de correlação de Pearson, onde o resultado se refere ao quão relacionadas as imagens são. A Figura 17 exibe o código para esse cálculo. Como o resultado é o mais próximo de ser positivo, foi estabelecido um limiar para o valor do coeficiente, uma vez que para valores abaixo desse limiar o balanço de branco está incorreto e para valores iguais ou acima está corretamente ajustado.

Figura 17. Cálculo do coeficiente de Pearson

```

65 a = cv2.imread("ImagemMediaRef.jpg")
66 imagem1 = cv2.cvtColor(a, cv2.COLOR_BGR2GRAY)
67
68 b = cv2.imread("ImagemMediaTest.jpg")
69 imagem2 = cv2.cvtColor(b, cv2.COLOR_BGR2GRAY)
70
71 mean1, stdev1 = cv2.meanStdDev(imagem1)
72 mean2, stdev2 = cv2.meanStdDev(imagem2)
73
74 Asub = imagem1 - mean1
75 Bsub = imagem2 - mean2
76
77 Sub = Asub*Bsub
78 meanAB, stdevAB = cv2.meanStdDev(Sub)
79
80 StdAB = stdev1*stdev2
81
82 Rho = meanAB/StdAB
83
84 print(Rho)

```

Apenas essa comparação bastaria para justificar se as imagens estão ajustadas ou não, porém esse método primeiro converte a imagem em cores para preto e branco, para depois serem comparadas, ou seja, neste código são comparados níveis de cinza. Então outro código para comparar cada componente de cor R, G e B foi desenvolvido. Ele pode dizer o quão quente ou fria é a imagem, uma vez, que se a componente R for mais sobressalente, isso significa que o balanço de branco está incorreto, pois a imagem está com níveis de vermelho maiores e isso vale para as outras componentes.

Na Figura 18, pode-se observar a parte do código que comprime a imagem, pois para imagens com uma resolução muito alta o processamento fica mais lento, já que uma imagem colorida é composta por 3 matrizes.

Figura 18. Comprime as imagens de referência e de teste.

```

66 a = cv2.imread("ImagemMediaRef.jpg")
67 dim = (149, 200)
68 resized1 = cv2.resize(a, dim)
69 cv2.imwrite('ImagemMediaREFresized.jpg', resized1)
70
71 b = cv2.imread("ImagemMediaTest.jpg")
72 dim2 = (149, 200)
73 resized2 = cv2.resize(b, dim2)
74 cv2.imwrite('ImagemMediaTestresized.jpg', resized2)

```

O próximo passo é exibido na Figura 19, onde são criadas as matrizes para receber as componentes de cores R, G e B da imagem de referência e de teste já comprimidas.

Figura 19. Declaração das matrizes e recebimento das componentes R, G e B.

```

76 arrayRef = np.array(resized1)
77 arrayRefR = np.asarray([])
78 arrayRefG = np.asarray([])
79 arrayRefB = np.asarray([])
80
81 arrayTest = np.array(resized2)
82 arrayTestR = np.asarray([])
83 arrayTestG = np.asarray([])
84 arrayTestB = np.asarray([])
85
86 for i in range(arrayRef.shape[0]):
87     for j in range(arrayRef.shape[1]):
88         arrayRefR = np.append(arrayRefR, arrayRef[i, j, 0])
89         arrayRefG = np.append(arrayRefG, arrayRef[i, j, 1])
90         arrayRefB = np.append(arrayRefB, arrayRef[i, j, 2])
91
92         arrayTestR = np.append(arrayTestR, arrayTest[i, j, 0])
93         arrayTestG = np.append(arrayTestG, arrayTest[i, j, 1])
94         arrayTestB = np.append(arrayTestB, arrayTest[i, j, 2])

```

Uma função “media” foi criada para obter a média das matrizes de componentes R e G, que representam a temperatura quente. Por fim é calculado o coeficiente de Pearson entre a média das componentes R e G das imagens de referência e de teste, assim como entre as componentes B das duas imagens, isso é possível observar na Figura 20.

Figura 20. Cálculo da média das matrizes R e G e coeficiente de Pearson.

```

96 def media(array, array2):
97     m = np.array([])
98     for i in range(len(array)):
99         m = np.append(m, (array[i] + array2[i]) / 2)
100     return m
101
102
103 array1 = np.array([media(arrayRefR, arrayRefG), media(arrayTestR, arrayTestG)])
104 array2 = np.array([arrayRefB, arrayTestB])
105
106 print(np.corrcoef(array1))
107 print(np.corrcoef(array2))

```

Após calculado coeficiente são realizados alguns testes, mostrados na sessão dos resultados, para definição do limiar que representa um balanço de branco correto.

3.3 AMBIENTE DE DESENVOLVIMENTO EM HARDWARE

O ambiente de desenvolvimento para a fase de implementação em hardware foi o seguinte:

- *Orange Pi PC Plus H3*;
- Sistema Operacional: Debian GNU/Linux;
- Webcam Multilaser Nightvision 16Mp;
- Computador: Notebook HP, com processador Intel® Core™2 Duo CPU T6500 @2.1GHz e 2.10GHz, memória instalada (RAM) de 3,00GB;
- Sistema Operacional: Windows 10;
- Conexão WI-FI;
- Teclado;
- Monitor;
- Protoboard;
- Jumpers;
- LED;
- Resistor 330 Ω .

3.4 IMPLEMENTAÇÃO DE TESTES EM HARDWARE

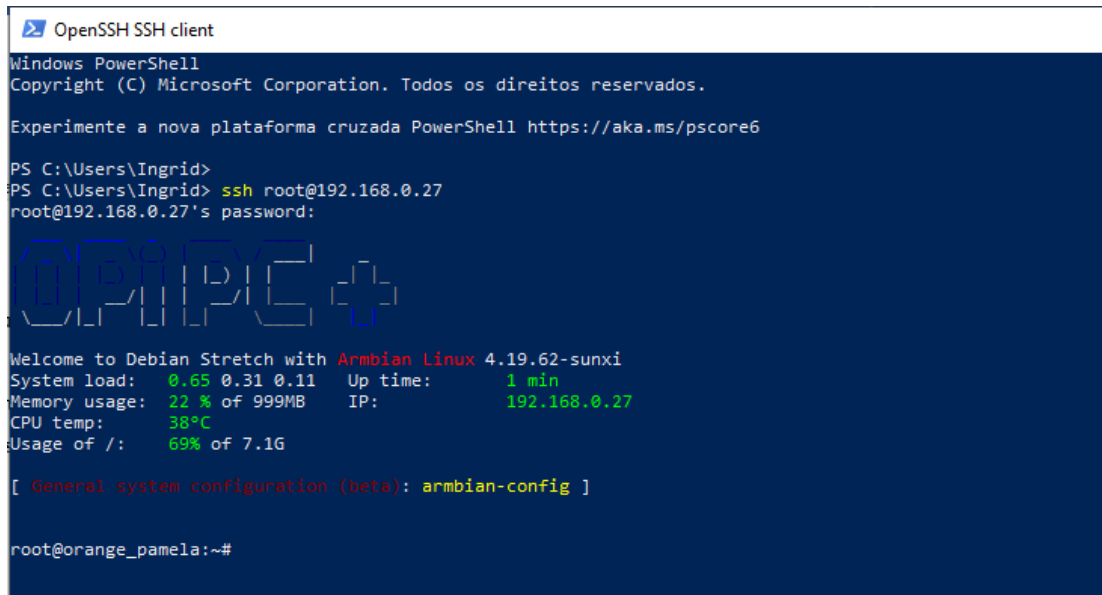
Como explicado anteriormente o *Orange Pi PC Plus H3* é um microcomputador, ou seja, tem todas as funções de um computador comum em uma simples placa, porém para torná-lo funcional são necessários alguns materiais, tais como mouse, teclado e monitor. Outra maneira de usá-lo de maneira bem prática é estabelecendo uma conexão via SSH ou VNP, neste projeto foi utilizado o SSH, um protocolo de comunicação que permite a execução de comandos em um computador remoto através da rede.

3.4.1 Estabelecendo uma sessão SSH

Primeiramente, foi necessário utilizar o teclado e monitor para conectar a placa à rede para depois utilizar outro computador remotamente (via SSH). Para estabelecer uma sessão

remota foi utilizado o *Windows PowerShell*, nele foi especificado o computador de destino (*Orange*) através de seu endereço IP, nome de usuário e senha, depois disso o computador está pronto para executar os comandos necessários, como mostra a Figura 21.

Figura 21. Estabelecendo uma sessão SSH com Windows PowerShell



```

OpenSSH SSH client
Windows PowerShell
Copyright (C) Microsoft Corporation. Todos os direitos reservados.

Experimente a nova plataforma cruzada PowerShell https://aka.ms/pscore6

PS C:\Users\Ingrid>
PS C:\Users\Ingrid> ssh root@192.168.0.27
root@192.168.0.27's password:
[ ASCII art logo ]
Welcome to Debian Stretch with Armbian Linux 4.19.62-sunxi
System load:  0.65 0.31 0.11   Up time:      1 min
Memory usage: 22 % of 999MB   IP:          192.168.0.27
CPU temp:     38°C
Usage of /:   69% of 7.1G

[ General system configuration (beta): armbian-config ]

root@orange_pamela:~#

```

Assim que foi estabelecida a sessão, foi necessário copiar o arquivo do algoritmo desenvolvido em *Python* do *notebook* para o *Orange* e realizar um teste para saber se a placa estava identificando a webcam que seria utilizada.

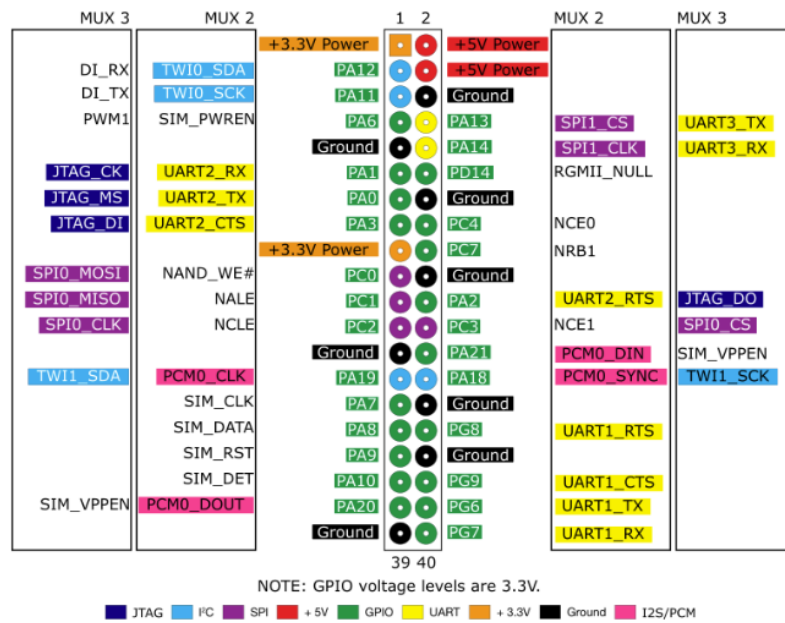
3.4.2 Projeto e desenvolvimento do circuito sinalizador

Após o código estar funcional no *Orange* foi criado um programa para acessar as GPIOs (*General Purpose Input/Output*) do microcomputador, uma vez que ele sinalizará quando o balanço de branco está correto ou não, de acordo com o limiar estabelecido para o coeficiente de correlação de Pearson. Na Figura 22 é possível visualizar a distribuição das GPIOs na barra de pinos do dispositivo.

A última etapa, foi montar um circuito sinalizador. O datasheet nos informa que a corrente máxima de saída é de 40 mA, então foi utilizado um resistor de 330 Ohms em série com o LED, a fim de limitar sua corrente em 10 mA (ALLWINNER H3, 2014). O circuito sinalizador é mostrado na Figura 23.

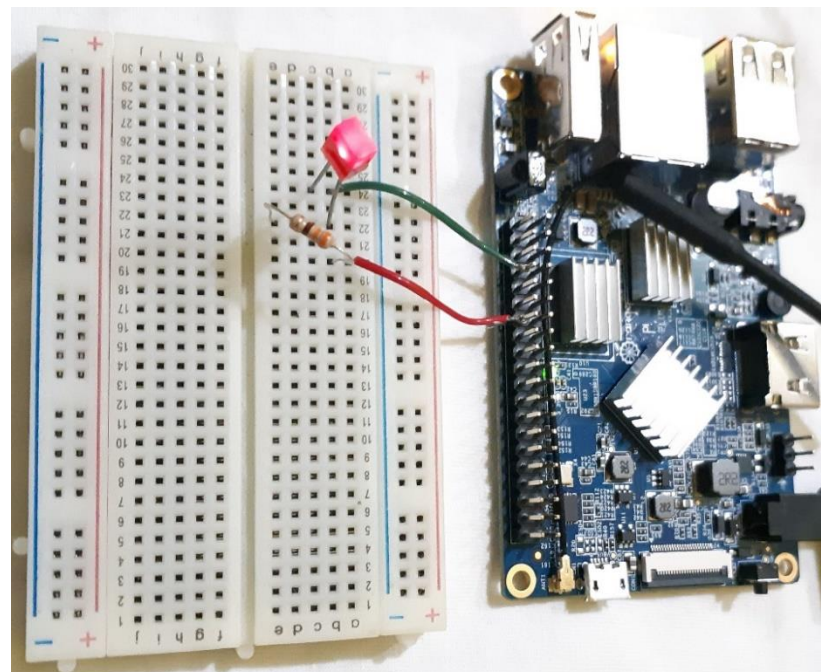
Figura 22. Distribuição das GPIOs

Orange Pi (H3 SoC) GPIO - pinout



Fonte: STACKOVERFLOW, 2017.

Figura 23. Circuito sinalizador



4 RESULTADOS

Neste capítulo são apresentados os resultados obtidos após as atividades realizadas no capítulo anterior.

Para avaliar se o balanço de branco foi realizado corretamente foi estabelecido um valor de limiar após comparação entre 10 imagens com balanço de branco correto e 10 com balanço de branco incorreto, a Figura 24 mostra uma das imagens exibidas pela TV, obtida pela câmera para fazer a comparação.

Figura 24. Comparação entre imagem com balanço de branco correto e incorreto



Primeiramente serão mostrados os resultados dos coeficientes utilizando o primeiro método, onde comparam-se os níveis de cinza, que são mostrados na Tabela 4. Esse resultado foi obtido após serem feitas 10 comparações entre imagens de referência e imagens desbalanceadas.

Tabela 4. Coeficientes de Pearson após comparação entre 10 imagens

	Coeficiente de Pearson
Comparação 1	0,9891795
Comparação 2	0,99395488
Comparação 3	0,98341411
Comparação 4	0,98567434
Comparação 5	0,96616674
Comparação 6	0,98685637
Comparação 7	0,98912491
Comparação 8	0,98941829

Comparação 9	0,98473821
Comparação 10	0,99651063

O limiar escolhido para este método foi de 0,99651063, qualquer valor abaixo disso irá indicar uma comparação entre imagens com diferentes balanços de branco, ou seja, a saída do LED não irá acender, pois o ajuste estará incorreto.

Para o segundo método, onde é feita a comparação das componentes R, G e B, foram encontrados os coeficientes da Tabela 5. Vale lembrar que foi realizada uma média das componentes R e G para que juntas representem a temperatura mais quente.

Tabela 5. Coeficientes de Pearson utilizando o segundo método.

	Coeficiente R e G	Coeficiente B
Comparação 1	0,84732665	0,84130145
Comparação 2	0,84562313	0,82658493
Comparação 3	0,81235964	0,84226973
Comparação 4	0,84992745	0,84244014
Comparação 5	0,84745265	0,43569872
Comparação 6	0,82596246	0,81271656
Comparação 7	0,58692213	0,83129165
Comparação 8	0,82478916	0,82138336
Comparação 9	0,64712746	0,81354712
Comparação 10	0,82471831	0,52489635

Neste caso, os coeficientes são menores visto que a comparação agora é feita entre cada matriz que representa uma componente. O limiar escolhido para a média das componentes R e G foi de 0,84992745 e para a componente B foi de 0,84244014, que foram os valores máximos de coeficiente obtidos.

Para representar o resultado deste método, também foi utilizado o circuito sinalizador, entretanto no algoritmo ao invés de ser comparado apenas um parâmetro, são comparados dois e é utilizada uma operação lógica AND, onde se pelo menos uma das componentes estiver abaixo do limiar, então significará que a imagem está desbalanceada.

4.1 RESULTADOS EM HARDWARE

Nesta sessão são apresentados os resultados experimentais obtidos com o método proposto, além da métrica de avaliação utilizada, sendo ela uma matriz de confusão que indica os erros e acertos do modelo, como mostra a Figura 25.

Figura 25. Matriz de confusão

		Valor Previsto	
		Positivo	Negativo
Valor Verdadeiro	Negativo	Verdadeiros Positivos	Falsos Negativos
	Positivo	Falsos Positivos	Verdadeiros Negativos

Fonte: MEDIUM,2012

Assim, o balanço de branco correto que for detectado como correto será identificado como (VP), o ajuste incorreto constatado como incorreto será (VN), quando estiver correto e for detectado como incorreto será (FP) e quando estiver incorreto e constatado como correto será identificado como (FN).

Para cálculo da acurácia, precisão e revocação, serão utilizadas as Equações 12, 13 e 14, respectivamente.

$$A = \frac{VP + VN}{VP + VN + FP + FN} \quad (12)$$

$$P = \frac{VP}{VP + FP} \quad (13)$$

$$R = \frac{VP}{VP + FN} \quad (14)$$

Onde, a acurácia indica o desempenho geral do modelo, ou seja, quantas classificações foram feitas corretamente, a precisão indica quantas classificações corretas o modelo executou

dentro da classe positiva e a revocação é a proporção entre as classificações corretas e o total identificado como correto mesmo sendo falso.

Na Figura 26 pode-se observar um teste realizado onde a imagem de referência e a imagem de teste são comparadas.

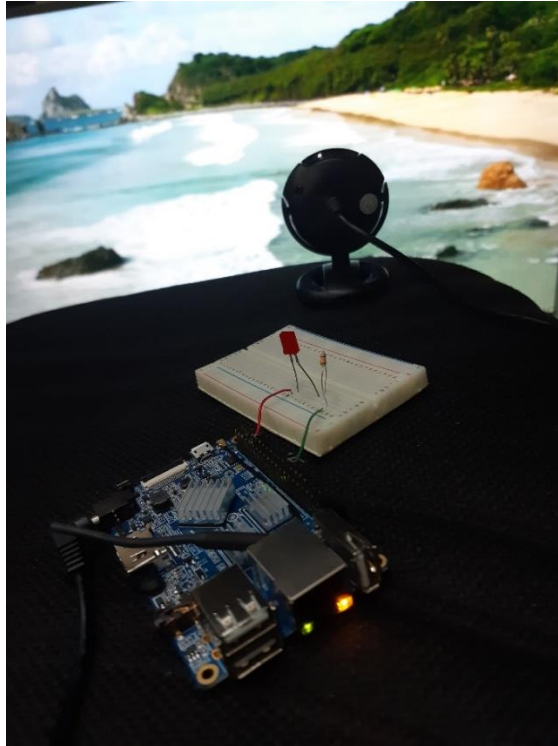
Figura 26. (a) Imagem de referência (b) Imagem de teste



A primeira análise tem resultados em que se observa um coeficiente de 0,98685637, ou seja, abaixo do estabelecido, pois a imagem de teste tem balanço de branco incorreto e foi detectado como incorreto como esperado. No segundo método a imagem também foi considerada incorreto, visto que é possível observar que a imagem de teste possui uma temperatura mais fria, ou seja, níveis maiores da componente B, resultando em coeficientes iguais a 0,84562313 e 0,81271656.

Na Figura 27, mostra a saída do circuito sinalizador apagada, indicando que a TV precisa ser ajustada.

Figura 27. Resultado mostrado no circuito sinalizador

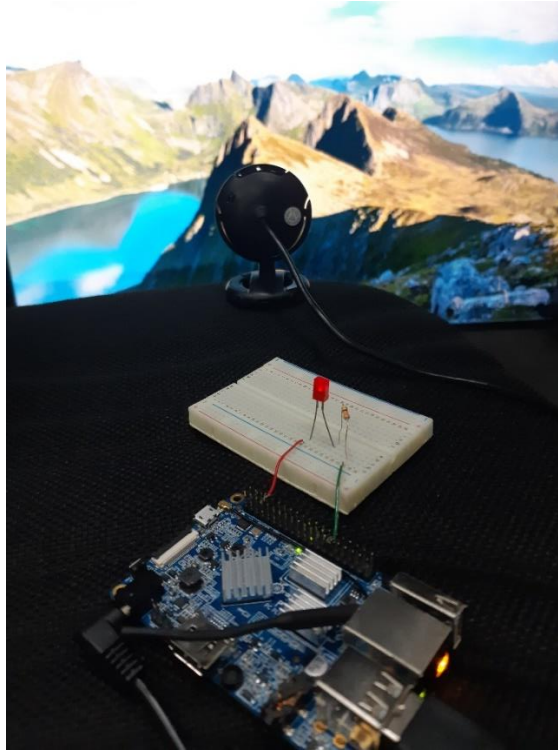


Agora é realizado um teste com imagens de TVs corretamente balanceadas, como mostra a Figura 28 e o resultado da comparação pode ser observado na Figura 29. Ambos os métodos classificaram a imagem como corretamente balanceada.

Figura 28. (a) Imagem de referência (b) Imagem de teste



Figura 29. LED aceso sinalizando ajuste correto



Após testes com 20 imagens, foi possível calcular os valores de acurácia, precisão e revocação para análise de desempenho do dispositivo desenvolvido. Dos 20 testes realizados, haviam 10 imagens com balanço de branco correto e 10 imagens com balanço de branco incorreto, onde apenas 1 imagem foi classificada inadequadamente no primeiro método, como mostra a Tabela 6.

Tabela 6. Resultados obtidos no primeiro método após 20 testes.

	Quantidade de imagens
VP	10
VN	9
FP	0
FN	1
Total	20

Nas Equações 15, 16 e 17 encontram-se os cálculos para análise de desempenho do primeiro método.

$$A = \frac{VP + VN}{VP + VN + FP + FN} = \frac{10+9}{20} = 0,95 \quad (15)$$

$$P = \frac{VP}{VP+FP} = \frac{10}{10+0} = 1 \quad (16)$$

$$R = \frac{VP}{VP+FN} = \frac{10}{11} = 0,91 \quad (17)$$

Para o segundo método, foram utilizadas as mesmas imagens, porém o resultado obtido foi considerado superior, uma vez que nenhuma imagem foi classificada incorretamente, como mostra a Tabela 7.

Tabela 7. Resultados obtidos no segundo método após 20 testes.

	Quantidade de imagens
VP	10
VN	10
FP	0
FN	0
Total	20

A análise de desempenho do segundo método é calculada nas Equações 18, 19 e 20.

$$A = \frac{VP+VN}{VP+VN+FP+FN} = \frac{10+10}{20} = 1 \quad (18)$$

$$P = \frac{VP}{VP+FP} = \frac{10}{10+0} = 1 \quad (19)$$

$$R = \frac{VP}{VP+FN} = \frac{10}{10} = 1 \quad (20)$$

CONCLUSÃO

Ao longo do desenvolvimento do trabalho, foram feitas algumas revisões sobre assuntos pertinentes a processamento digital de imagens e microcontroladores. Além disso novos conhecimentos acerca de balanço de branco e colorimetria foram adquiridos para que o trabalho pudesse ter uma boa fundamentação. Entretanto, o desenvolvimento de um sistema que faça o balanço de branco não é nada simples, uma vez que a colorimetria não depende apenas das cores, mas também dos diferentes iluminantes sob os quais uma imagem pode ser visualizada.

O trabalho teve o intuito de desenvolver um dispositivo para realizar o balanço de branco e fazer a calibração em televisores, porém ao longo do projeto foram encontrados alguns impedimentos, visto que, esse teste é realizado nos produtos finais e os projetos dos mesmos são confidenciais. Ainda assim, foi possível desenvolver o dispositivo para realizar o balanço, sendo o resultado simulado por um LED, ou seja, ao invés de o sinal ser mandado para a TV isso é simulado por um circuito sinalizador.

Foi possível observar que os resultados dos algoritmos desenvolvidos foram determinantes para avaliar os parâmetros do projeto proposto e dessa forma efetuar comparações de desempenho com diferentes resultados para cada método utilizado, o de comparação de imagens em níveis de cinza e de imagens em cores, comparando cada uma das componentes RGB. Sendo que o primeiro método mostrou-se com bom desempenho como mostrado pela acurácia, precisão e revocação obtidos, porém para imagens com pouca definição de fronteiras ou bordas ele se mostrou menos eficiente. Já os resultados do segundo método foram superiores, visto que as comparações foram mais precisas e nenhuma imagem foi incorretamente classificada.

Os resultados experimentais em hardware foram obtidos com o dispositivo desenvolvido utilizando o *Orange Pi*, que apesar de ser um microcomputador, apresentou-se de forma eficiente, porém para imagens de alta resolução o tempo de processamento ficou mais demorado, evidenciando o que foi abordado no capítulo do referencial teórico que afirma que o aumento do tempo de processamento ocorre devido às técnicas de PDI terem que percorrer uma matriz maior. Assim, o aumento de espaço para armazenamento também é um fator que deve ser considerado, pois apesar de um o *Orange Pi* ter uma memória interna, esta é limitada e acima do especificado é necessário se utilizar um cartão SD externo.

Também é necessário ressaltar que os resultados foram obtidos comparando-se imagens iguais, ou seja, foi feita a captura de uma imagem de uma TV corretamente balanceada e uma desbalanceada, porém exibindo a mesma imagem. Isso foi necessário, pois o coeficiente de

Pearson dá o resultado de quão correlacionadas as imagens são, ou seja, se fosse exibidas imagens diferentes, mesmo que ambas as TVs estivessem com balanço de branco correto, o resultado da correlação iria ser muito pequeno e sendo necessário utilizar outro método de comparação.

Diante do exposto, foi observado que o objetivo do trabalho foi alcançado, ou seja, é possível implementar uma arquitetura de hardware com o *Orange Pi*, que realize o balanço de branco em televisores em tempo real. Além disso, pode-se afirmar que o projeto obteve um resultado satisfatório ao aplicar conhecimentos adquiridos ao decorrer do curso de Engenharia Elétrica como processamento digital de imagens, lógica de programação, eletrônica analógica e digital e microcontroladores para desenvolver um dispositivo simples e eficiente visando aplicar uma solução de redução de custo de equipamentos para realização desse teste. Comprova-se então, a versatilidade que um microcomputador pode proporcionar, realizando desde pequenos projetos até o processamento digital de imagens.

REFERÊNCIAS BIBLIOGRÁFICAS

BACKHAUS, W. G. K. **Physiological and psychological simulations of color vision in humans and animals**. Berlin: de Gruyter, 1998.

BECKER, Valdecir, PACINE, Bruno e LEMOS, Guido. **Evolução da definição e resolução do vídeo**. Universidade Federal da Paraíba, Paraíba, 2018.

BLOG IMPACTA, **Conheça a Linguagem Python e saiba como é possível usá-la!** Disponível em: <<https://www.impacta.com.br/blog/2018/09/17/conheca-a-linguagem-python-e-saiba-como-e-possivel-usa-la/>>. Acesso em 20. Set. 2019.

BRADSKI, Gary e KAEHLER, Adrian. **Learning OpenCV**. 1 Ed. USA: O'Reilly Media, 2008.

CAELUM, **Apostila Python e Orientação a Objetos**. Disponível em: <<https://www.caelum.com.br/apostila-python-orientacao-objetos/o-que-e-python/>>. Acesso em 20. Set. 2019.

CANALTECH, **O que é Raspberry Pi?** Disponível em: <<https://canaltech.com.br/hardware/o-que-e-raspberry-pi/>>. Acesso em 07. Mai. 2019.

DARULA, Stanislav. KITTLER, Richard. **Cie general sky standard defining luminance distributions**. Slovak Academy of Sciences, Bratislava, 2002.

DREAMSTIME, **Spectrum**. Disponível em: <<https://pt.dreamstime.com/infographic-da-cor-do-espectro-visivel-luz-solar-image123033953>>. Acesso em 07 Mai. 2019.

EMBARCADOS, **Primeiros passos - Orange Pi PC Plus H3**. Disponível em: <<https://www.embarcados.com.br/primeiros-passos-orange-pi-pc-plus-h3/>>. Acesso em 30. Set. 2019.

FERNANDES COSTA, M. G. **Programa de Mestrado em Engenharia Elétrica: Processamento Digital de Imagens**. Universidade Federal do Amazonas, Manaus, 2008.

FILHO, D. B. F. e JÚNIOR, J. A. S. **Desvendando os Mistérios do Coeficiente de Correlação de Pearson (r)**. Revista Política Hoje, Vol. 18, n. 1, 2009.

GERENCIAMENTO DE COR E IMPRESSÃO, **Dimensões e espaços de Cor**. Disponível em: <<https://gerenciamentodecor.wordpress.com/2016/03/04/dimensoes-e-espacos-de-cor/>>. Acesso em 24. Abr. 2019.

GONZALEZ, R. e WOODS, R. **Processamento de imagens digitais**. Wilmington, USA: s.n., 1996.

GONZALEZ, R. e WOODS, R. **Digital Image Processing**. 2.ed. New Jersey: Prentice-Hall, 2002.

LIMA, M, et al. **Métodos utilizados na avaliação psicofísica da visão de cores humana**. Universidade de São Paulo, São Paulo, 2011.

MARQUES FILHO, Ô e VIEIRA NETO, H. **Processamento Digital de Imagens**. 1. Ed. Rio de Janeiro: Brasport, 1999.

MONTGOMERY, D e RUNGER, G. **Estática aplicada e probabilidade para engenheiros**. 4 Ed. Arizona: LTC, 2009.

MEDIUM, **A Beginner's Guide to (CIE) Colorimetry**. Disponível em: <<https://medium.com/hipster-color-science/a-beginners-guide-to-colorimetry-401f1830b65a>>. Acesso em 02. Mar. 2019.

MEDIUM, **Métricas de Avaliação: Acurácia, precisão, recall... quais as diferenças?** Disponível em: <https://medium.com/@vitorborbarodrigues/m%C3%A9tricas-de-avalia%C3%A7%C3%A3o-acur%C3%A1cia-precis%C3%A3o-recall-quais-as-diferen%C3%A7as-c8f05e0a513c>. Acesso em 30. Set. 2019.

ORANGE PI, **What's Orange Pi Pc Plus?** Disponível em: <<http://www.orangepi.org/orangepipcplus/>>. Acesso em 20. Set. 2019.

RECOMMENDATION ITU-R BT.709-6: **Parameter values for the HDTV standards for production and international programme exchange**. Genebra, 2015.

RIBEIRO, A. F, **Sensors for autonomous mobile robots through the use of image processing tools**. Universidade do Minho, Portugal, 2005.

SÁNCHEZ- FERREIRA, C. **Implementação do algoritmo de subtração de fundo para detecção de objetos em movimento, usando sistemas reconfiguráveis**. 2012 Dissertação (Mestrado em Sistemas Mecatrônicos). Faculdade de Tecnologia, Universidade de Brasília, Brasília, 2012.

SANTANA, C, et al. **Espaço de cores**. Universidade São Paulo, São Paulo, 2006.

SCHANDA, J. **Colorimetry: Understanding the CIE System**. 1 Ed. Canada: Wiley, 2007.

SIMÕES, A. L. A. C. **Estudo das alterações visuais cromáticas em trabalhadores expostos ao vapor de mercúrio**. Dissertação de Mestrado, Instituto de Psicologia, Universidade de São Paulo, São Paulo, 2004.

VESA, Video Electronics Standards Association. **Flat Panel Display Measurements Standard**. Califórnia, 2001.

VIERO, E. **O que é balanço dos brancos – White Balance**. Disponível em: <https://www.eduardo-monica.com/new-blog/balanco-dos-brancos-white-balance>. Acesso em 08. Jun. 2019.

WEBINSIDER, **Mitos da calibração das telas de TV**. Disponível em: <<https://webinsider.com.br/mitos-da-calibracao-das-telas-de-tv/>>. Acesso em 12. Abr. 2019.