

INSTITUTO FEDERAL DE SANTA CATARINA

MURILO SILVEIRA

ESTUDO DA UTILIZAÇÃO DE VISÃO COMPUTACIONAL EMBARCADA EM
SISTEMAS DE MANUFATURA

Joinville

2022

MURILO SILVEIRA

ESTUDO DA UTILIZAÇÃO DE VISÃO COMPUTACIONAL EMBARCADA EM
SISTEMAS DE MANUFATURA

Monografia apresentada ao
Curso de Bacharelado em
Engenharia Elétrica do
Campus Joinville do
Instituto Federal de Santa
Catarina para a obtenção do
diploma de Engenheiro
Eletricista

Orientador: Stefano Romeu
Zeplin, Me. Eng.

Joinville

2022

Silveira, Murilo.

Estudo da utilização de visão computacional embarcada em sistemas de manufatura / Murilo Silveira – Joinville, SC, 2022.

124 p.

Trabalho de Conclusão de Curso (Graduação) - Instituto Federal de Educação Ciência e Tecnologia de Santa Catarina, Curso de Bacharelado em Engenharia Elétrica, Joinville, 2022.

Orientador: Stefano Romeu Zeplin.

1. Sistemas embarcados de visão. 2. Viabilidade. 3. Sistemas de manufatura.
I. Instituto Federal de Educação Ciência e Tecnologia de Santa Catarina. II.
Título.

MURILO SILVEIRA

ESTUDO DA UTILIZAÇÃO DE VISÃO COMPUTACIONAL EMBARCADA EM
SISTEMAS DE MANUFATURA

Este trabalho foi julgado adequado para obtenção do título em Engenheiro Eletricista, pelo Instituto Federal de Educação, Ciência e Tecnologia de Santa Catarina Campus Joinville, e aprovado na sua forma final pela comissão avaliadora abaixo indicada.

Joinville, 22, fevereiro de 2022.



Prof. Stefano Romeu Zeplin, Me. Eng.

Orientador

Instituto Federal de Santa Catarina Campus Joinville



Prof. Neury Boaretto, Dr. Eng.

Instituto Federal de Santa Catarina Campus Joinville



Prof. Rodrigo Coral, Dr. Eng.

Instituto Federal de Santa Catarina Campus Joinville

AGRADECIMENTOS

À Santíssima Trindade, à Nossa Senhora de Fátima e a São José por me abençoarem e darem a Graça de poder frequentar o curso de graduação.

À minha família, pelo amor, paciência e incentivo incondicional em toda a trajetória da graduação.

Aos meus colegas de engenharia durante toda a graduação, pelos momentos de descontração e principalmente pelas ajudas em momentos de dificuldades.

Aos docentes do Instituto Federal de Santa Catarina Campus Joinville, por todo o conhecimento transmitido, toda dedicação e todos os incentivos repassados.

E a todos que direta e indiretamente fizeram parte da minha caminhada até a formação, deixo aqui o meu muito obrigado.

RESUMO

Em um cenário onde a indústria de manufatura visa a alta competitividade no mercado, aliado a crescente exigência dos consumidores por produtos de alta qualidade, resultam na busca por soluções que aumentem a eficácia dos processos de manufatura. Uma dessas alternativas é o uso de visão computacional, porém como qualquer aplicação se leva em consideração a relação custo-benefício. Nesse sentido, o presente trabalho tem como objetivo estudar a viabilidade da utilização de sistemas embarcados no desenvolvimento de algoritmos de visão computacional para aplicação na manufatura. Para alcançar esse objetivo o trabalho fundamentou-se através de revisão bibliográfica sobre sistemas de visão e sistemas embarcados, além da execução de um algoritmo de medição em um sistema de visão embarcado. Demonstrou-se a partir dos dados coletados, a importância do domínio das características e peculiaridades de cada componente do sistema de visão embarcado e por meio do algoritmo de medição se pode comprovar que a viabilidade do uso dos sistemas embarcados de visão computacional está atrelada com a precisão requerida pelo processo produtivo, demonstrados através dos resultados de incerteza padrão amostral e repetitividade obtidos.

Palavras-chave: Sistemas Embarcados de Visão. Viabilidade. Sistema de Manufatura.

ABSTRACT

In a scenario where the manufacturing industry aims to high competitiveness in the market, combined with the growing demand of consumers for high quality products, result in the search for solutions that increase the effectiveness of manufacturing processes. One of these alternatives is the use of computer vision, but like any application, the cost-benefit relationship is taken into account. In this sense, the present work aims to study the feasibility of using embedded systems in the development of computer vision algorithms for application in manufacturing. To achieve this objective, the work was based on a literature review on vision systems and embedded systems, in addition to the execution of a measurement algorithm by an embedded vision system. It was demonstrated from the collected data, the importance of mastering the characteristics and peculiarities of each component of the embedded vision system and through the measurement algorithm it can be proved that the feasibility of using embedded computer vision systems is linked to the precision required by the production process, demonstrated through the sample standard uncertainty and repeatability results obtained.

Keywords: Embedded Vision Systems. Viability. Manufacturing System.

LISTA DE ILUSTRAÇÕES

Figura 1: Etapas do Processamento Digital de Imagens.....	19
Figura 2: O efeito da iluminação de um rosto por quatro direções diferentes	20
Figura 3: Os raios de luz refletidos por um objeto e focados por uma lente delgada convexa.....	21
Figura 4: Imagem captada com o sensor na distância correta (esquerda) e incorreta (direita)	21
Figura 5: Representação de um vetor 10x10 de amostragem de uma imagem	22
Figura 6: Amostragem e quantização de uma imagem de valor contínuo	23
Figura 7: Representação ideal de uma borda em degrau	25
Figura 8: Gradiente em uma borda	26
Figura 9: Vários círculos passam pela célula correspondente ao centro do círculo identificado pela Transformada de Hough	27
Figura 10: Aproximação poligonal	28
Figura 11: Arquitetura von Neumann	31
Figura 12: Arquitetura de Harvard	32
Figura 13: Exemplo de condicionamento de sinal de entrada	33
Figura 14: Uso de iluminação diascópica destacando o conteúdo interno da garrafa	37
Figura 15: Exemplo de um objeto antes(a) e após a digitalização da imagem (b)	38
Figura 16: Raspberry Pi 3.....	42
Figura 17: Exemplo de configuração do processador Cortex-A53	44
Figura 18: A Porta GPIO.....	45
Figura 19: Logitech C270	46
Figura 20: Raspberry Pi Camera v1 versão 1.3	47
Figura 21: Ambiente de captura das imagens	49
Figura 22: Imagem capturada com resolução de 720p (à esquerda) e 240p	50
Figura 23: Alteração dos parâmetros de matiz, saturação e luminosidade da câmera	51
Figura 24: Método para alterar o foco da câmera Logitech C270	51
Figura 25: Longa distância de foco da câmera Logitech C270.....	52
Figura 26: Curta distância de foco da câmera Logitech C270.....	52

Figura 27: Distância ajustada de foco da câmera Logitech C270	53
Figura 28: Imagens com inversão vertical (à direita) e horizontal	54
Figura 29: Imagens com acuidade baixa (à esquerda) e alta	55
Figura 30: Imagem com ajustes na saturação, contraste e brilho, destacando o objeto em análise (à direita).....	55
Figura 31: Captura do objeto pela Raspberry Pi Camera v1	56
Figura 32: Captura do objeto pela Logitech C270	57
Figura 33: Matiz, saturação e luminosidade do laranja capturado pela Logitech C270	57
Figura 34: Matiz, saturação e luminosidade do laranja capturado pela Raspberry Pi Camera v1	58
Figura 35: Histograma da imagem capturada pela Logitech C270.....	58
Figura 36: Histograma da imagem capturada pela Raspberry Pi Camera v1	59
Figura 37: Iluminação direcional frontal difusa capturada por Logitech C270 (à direita) e Raspberry Pi Camera v1	60
Figura 38: Iluminação frontal difusa capturada por Logitech C270 (à direita) e Raspberry Pi Camera v1	61
Figura 39: Níveis de saturação para imagem capturada pela Logitech C270	61
Figura 40: Níveis de saturação para imagem capturada pela Raspberry Pi Camera v1	62
Figura 41: Iluminação diascópica capturada por Logitech C270 (à direita) e Raspberry Pi Camera v1.....	63
Figura 42: Lâmpadas halógenas como fonte de luz capturada por Logitech C270 (à direita) e Raspberry Pi Camera v1	64
Figura 43: Lâmpadas LED como fonte de luz capturada por Logitech C270 (à direita) e Raspberry Pi Camera v1	65
Figura 44: Imagens com intensidade luminosa menor (à direita) e maior capturada pela Logitech C270	66
Figura 45: Histograma das imagens com intensidade luminosa menor (acima) e maior capturada pela Logitech C270.....	67
Figura 46: Imagens com intensidade luminosa menor (à direita) e maior capturada pela Raspberry Pi Camera v1	67
Figura 47: Histograma das imagens com intensidade luminosa menor (acima) e maior	

capturada pela Raspberry Pi Camera v1	68
Figura 48: Imagem adquirida em escala de nível cinza (à esquerda) e imagem com filtro gaussiano aplicado	70
Figura 49: Aplicação do operador Canny na imagem.....	70
Figura 50: Aplicação do algoritmo <i>findContours</i> na imagem	71
Figura 51: Imagem com retângulo ao entorno do objeto (à esquerda) e imagem com algoritmo <i>boxPoints</i> aplicado.....	71
Figura 52: Organização dos pontos extremos do objeto	72
Figura 53: Aquisição de comprimento e largura em posições diferentes	73
Figura 54: Quadrantes de calibração do pixel	74
Figura 55: Pontos adquiridos do algoritmo de aquisição dos pontos extremos do objeto	75
Figura 56: Varredura do eixo X com variação do eixo Y a fim de obter pontos que satisfazem a equação do segmento de reta.....	76
Figura 57: Segmentos de reta localizados nos respectivos quadrantes de calibração	77
Figura 58: Menu inicial da interface.....	78
Figura 59: Interface de aquisição de imagens para calibração do pixel	78
Figura 60: Interface de aquisição do valor verdadeiro convencional do mensurando	79
Figura 61: Interface dos resultados da calibração do pixel	79
Figura 62: Interface de configuração de parâmetros da aplicação.....	80
Figura 63: Aplicação em funcionamento	80
Figura 64: Sistema de visão embarcada com câmera Logitech C270 a 345 mm do plano	82
Figura 65: Sistema de visão embarcada com câmera Logitech C270 a 250 mm do plano	83
Figura 66: Sistema de visão embarcada com câmera Logitech C270 a 160 mm do plano	85
Figura 67: Sistema de visão embarcada com câmera Raspberry Pi Camera v1 a 345 mm do plano.....	87
Figura 68: Sistema de visão embarcada com câmera Raspberry Pi Camera v1 a 250 mm do plano.....	88

Figura 69: Sistema de visão embarcada com câmera Raspberry Pi Camera v1 a 160 mm do plano.....	90
Figura 70: Medições com Logitech C270 a 345 mm de altura com correção.....	95
Figura 71: Distribuição normal das medições de comprimento com Logitech C270 a 345 mm de altura	96
Figura 72: Distribuição normal das medições de largura com Logitech C270 a 345 mm de altura	96
Figura 73: Medições com Logitech C270 a 250 mm de altura com correção.....	97
Figura 74: Distribuição normal das medições de comprimento com Logitech C270 a 250 mm de altura	98
Figura 75: Distribuição normal das medições de largura com Logitech C270 a 250 mm de altura	98
Figura 76: Medições com Raspberry Pi Camera v1 a 250 mm de altura com correção	99
Figura 77: Distribuição normal das medições de comprimento com Raspberry Pi Camera v1 a 250 mm de altura	100
Figura 78: Distribuição normal das medições de largura com Raspberry Pi Camera v1 a 250 mm de altura	101
Figura 79: Medições com Logitech C270 a 160 mm de altura com correção.....	102
Figura 80: Distribuição normal das medições de comprimento com Logitech C270 a 160 mm de altura	103
Figura 81: Distribuição normal das medições de largura com Logitech C270 a 160 mm de altura	103
Figura 82: Medições com Raspberry Pi Camera v1 a 160 mm de altura com correção	104
Figura 83: Distribuição normal das medições de comprimento com Raspberry Pi Camera v1 a 160 mm de altura	105
Figura 84: Distribuição normal das medições de largura com Raspberry Pi Camera v1 a 160 mm de altura	105

LISTA DE TABELAS

Tabela 1: Resultados das medições em pixel.....	74
Tabela 2: Medições da largura do objeto com câmera Logitech C270 a 345 mm do plano	82
Tabela 3: Medições do comprimento do objeto com câmera Logitech C270 a 345 mm do plano	82
Tabela 4: Medições da largura do objeto com câmera Logitech C270 a 250 mm do plano	84
Tabela 5: Medições do comprimento do objeto com câmera Logitech C270 a 250 mm do plano	84
Tabela 6: Medições da largura do objeto com câmera Logitech C270 a 160 mm do plano	85
Tabela 7: Medições do comprimento do objeto com câmera Logitech C270 a 160 mm do plano	86
Tabela 8: Medições da largura do objeto com câmera Raspberry Pi Camera v1a 250 mm do plano.....	88
Tabela 9: Medições do comprimento do objeto com câmera Raspberry Pi Camera v1a 250 mm do plano.....	88
Tabela 10: Medições da largura do objeto com câmera Raspberry Pi Camera v1a 160 mm do plano.....	90
Tabela 11: Medições do comprimento do objeto com câmera Raspberry Pi Camera v1a 160 mm do plano.....	90
Tabela 12: Resultado das medições com Logitech C270 a 345 mm do plano	94
Tabela 13: Resultado das medições com Logitech C270 a 250 mm do plano	97
Tabela 14: Resultado das medições com Raspberry Pi Camera v1 a 250 mm do plano	99
Tabela 15: Resultado das medições com Logitech C270 a 160 mm do plano	101
Tabela 16: Resultado das medições com Raspberry Pi Camera v1 a 160 mm do plano	103
Tabela 17: Custo da aplicação	106

SUMÁRIO

1	INTRODUÇÃO.....	15
1.1	Objetivos	16
1.1.1	Objetivo geral.....	16
1.1.2	Objetivos específicos	16
2	FUNDAMENTAÇÃO TEÓRICA.....	18
2.1	Processamento digital de imagens	18
2.1.1	Aquisição da Imagem	19
2.1.2	Pré - Processamento	23
2.1.3	Segmentação da Imagem	24
2.1.4	Representação e Descrição.....	28
2.2	Sistemas Embarcados.....	29
2.2.1	Hardware	29
2.2.2	Arquitetura e Processadores.....	31
2.2.3	Interface.....	33
2.2.4	Comunicação	33
2.2.5	Software.....	34
2.3	Medição dimensional por visão computacional.....	35
2.3.1	Calibração do Pixel	36
2.3.2	Erros em medições por imagem	36
2.3.3	Incerteza na medição por imagem.....	38
3	DESENVOLVIMENTO	40
3.1	Escopo do Trabalho.....	40
3.2	Ferramentas Utilizadas.....	40
3.2.1	Raspberry Pi	40
3.2.2	Câmera webcam Logitech C270.....	46
3.2.3	Raspberry Pi Camera v1.....	47
3.2.4	OpenCV	47

3.2.5	Estrutura do Sistema Embarcado de Visão	48
3.3	Avaliação das Câmeras	49
3.3.1	Configuração de parâmetros da câmera Logitech C270.....	49
3.3.2	Configuração de Parâmetros da Raspberry Pi Camera v1	53
3.4	Nível de saturação das imagens.....	56
3.5	Geometria do sistema de iluminação.....	59
3.5.1	Geometria 45°: 0° - iluminação direcional frontal difusa	59
3.5.2	Geometria 45°: 0° - iluminação frontal difusa.....	60
3.5.3	Iluminação diascópica.....	63
3.5.4	Iluminação frontal difusa – lâmpadas halógenas	64
3.5.5	Iluminação frontal difusa – lâmpadas LED.....	65
3.5.6	Iluminação frontal difusa – variação da intensidade luminosa	66
3.6	Algoritmo de Medição Bidimensional	69
3.6.1	Aquisição dos pontos extremos do objeto	69
3.6.2	Calibração do pixel a partir de divisão de quadrantes	73
3.6.3	Separação dos segmentos de reta	75
3.7	Interface de operação do usuário.....	77
3.8	Aquisição de dimensões com o sistema embarcado de visão computacional	81
3.8.1	Câmera Logitech C270 com altura de medição de 345 mm	81
3.8.2	Câmera Logitech C270 com altura de medição de 250 mm	83
3.8.3	Câmera Logitech C270 com altura de medição de 160 mm	84
3.8.4	Câmera Raspberry Pi Camera v1 com altura de medição de 345 mm	86
3.8.5	Câmera Raspberry Pi Camera v1 com altura de medição de 250 mm	87
3.8.6	Câmera Raspberry Pi Camera v1 com altura de medição de 160 mm	89
4	ANÁLISE E DISCUSSÃO DOS RESULTADOS.....	92
4.1	Custo dos dispositivos	106
5	CONCLUSÃO.....	108
	REFERÊNCIAS	111
	APÊNDICE 1	118

APÊNDICE 2	119
APÊNDICE 3	120
APÊNDICE 4	123

1 INTRODUÇÃO

Crescentemente, percebe-se a exigência dos consumidores por produtos e serviços de qualidade. Como resultado as empresas têm se esforçado cada vez mais para tentar garantir e encontro desse desejo por parte dos indivíduos (BRITTO,2016). O mesmo processo ocorre na indústria de manufatura, que vem buscando soluções cada vez mais autônomas para aumentar a eficácia de seus processos (COGNEX, 2021).

Uma dessas alternativas, é o uso de visão computacional nos processos de manufatura. Essa é a ciência responsável pela visão de uma máquina, de forma que extrai informações significativas a partir de imagens capturadas por câmeras, sensores, scanners entre outros dispositivos. E com tais informações é possível manipular, reconhecer e tomar ações sobre os objetos que compõem a imagem (HARRY BALLARD; BROWN, 1982).

Uma das aplicações desses sistemas de visão computacional é a medição dimensional de objetos, tal parâmetro é utilizado não só com o objetivo de controle de qualidade em indústrias de manufatura, aprovando ou rejeitando objetos, mas também de orientar a fabricação de produtos, e ter uma diminuição de custos no processo como um todo (SENAI; CTS, 1996).

Além disso, o uso de visão computacional se torna uma alternativa aos métodos manuais de inspeção como uma ferramenta para ensaios não destrutivos, visto que não existe contato com o objeto de medição, além de eliminar aspectos da inspeção humana, como lentidão, fadiga e subjetividade (LETA et al., 2016).

Apesar dos inegáveis ganhos para o processo produtivo, o alto custo de aquisição de sistemas que utilizam visão computacional pode muitas vezes dificultar a implementação dessas soluções na indústria (ATWELL, 2021).

Nessa perspectiva, torna-se imperativa a discussão sobre a utilização de sistemas embarcados de visão computacional em sistemas de manufatura, visto a alta competitividade do mercado e conseqüentemente a constante necessidade de redução de custos na fabricação de produtos manufaturados.

Considerando-se os elementos do cenário até aqui evidenciados, estabelece-se como questão central de investigação do presente trabalho: os sistemas

embarcados podem ser uma alternativa de baixo custo para aplicações na manufatura que necessitam de visão computacional?

Para o efetivo desenvolvimento sobre a questão central de investigação do presente trabalho em um corpo consistente de análise e argumentação, este estudo tem como base uma pesquisa exploratória, e apresentação de resultados quantitativos, a partir da coleta de informações de fontes primárias e secundárias.

No primeiro capítulo, serão abordadas etapas e conceitos de processamento digital de imagens importantes para a medição dimensional de objetos, assim como o conceito de sistemas embarcados e seus principais componentes.

No segundo capítulo são caracterizadas algumas placas Raspberry Pi, assim como câmeras, com relação a parâmetros e comportamento em diferentes situações do sistema embarcado de visão computacional.

No terceiro capítulo, é abordado o algoritmo de medição bidimensional de tal forma a explicar seu funcionamento e particularidades.

Já no quarto capítulo, são apresentados resultados de aplicação do algoritmo em um sistema didático de manufatura.

Ao final, são apresentadas discussões sobre os resultados obtidos.

1.1 Objetivos

1.1.1 Objetivo geral

Estudar a viabilidade da utilização de sistemas embarcados no desenvolvimento de algoritmos de visão computacional para aplicação na manufatura.

1.1.2 Objetivos específicos

- Estudar conceitos de sistema de visão
- Estudar conceitos de sistemas embarcados
- Estudar as características de câmeras e placas Raspberry Pi
- Implementar algoritmo de medição bidimensional
- Avaliar o desempenho do algoritmo de medição bidimensional em sistemas

didáticos de manufatura dos laboratórios do IFSC.

2. FUNDAMENTAÇÃO TEÓRICA

Nessa seção do trabalho será realizada a abordagem introdutória sobre visão computacional, realizando-se um estudo sobre etapas e conceitos de processamento digital de imagens que permitem a medição dimensional de objetos. Além disso, será realizado um estudo acerca dos sistemas embarcados e seus principais componentes.

1.2 Processamento digital de imagens

Uma imagem pode ser representada por uma função de intensidade luminosa $f(x, y)$, onde a amplitude nas posições espaciais x e y representam a intensidade ou o brilho naquele ponto. Essa função $f(x, y)$ é formada por dois componentes: a iluminância e a reflectância (GONZALEZ; WOODS, 2009).

$$f(x, y) = i(x, y)r(x, y) \quad (1.0)$$

A iluminância é determinada pela fonte de iluminação, e em teoria pode ter valores entre 0 e infinito, mas para a luz visível, por exemplo em um dia claro, o sol pode produzir mais do que 90000 lm/m² sobre a superfície da terra, porém em uma noite de lua cheia há uma iluminação média de 0,1 lm/m² (GONZALEZ; WOODS, 2009).

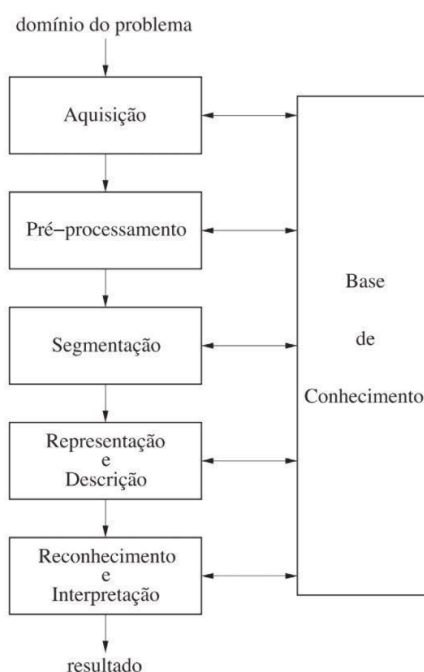
Já a reflectância é determinada pela característica do objeto, a intensidade igual à 0 condiz para uma situação de absorção total, já a intensidade igual à 1 para reflectância total. Porém esses limites também são teóricos, os valores comuns para o veludo preto é de 0,01 e 0,93 para neve (GONZALEZ; WOODS, 2009).

Entretanto essas imagens formadas pelas duas variáveis citadas, não se restringem àquelas capturadas por câmeras digitais que estão presentes na banda visível do espectro eletromagnético (GONZALEZ e WOODS, 2010), mas também por uma gama de sensores, capazes de captar diversos tipos de radiação, como a imagem por ressonância magnética (MRI) e imagem por tomografia computadorizada por emissão de pósitrons (PET), como afirma (BOVIK, 2009).

Independente da sua forma de aquisição, o processamento digital de imagens

pode ser dividido em algumas etapas essenciais para a obtenção de informações úteis, como apresentado na Figura 1.

Figura 1: Etapas do Processamento Digital de Imagens



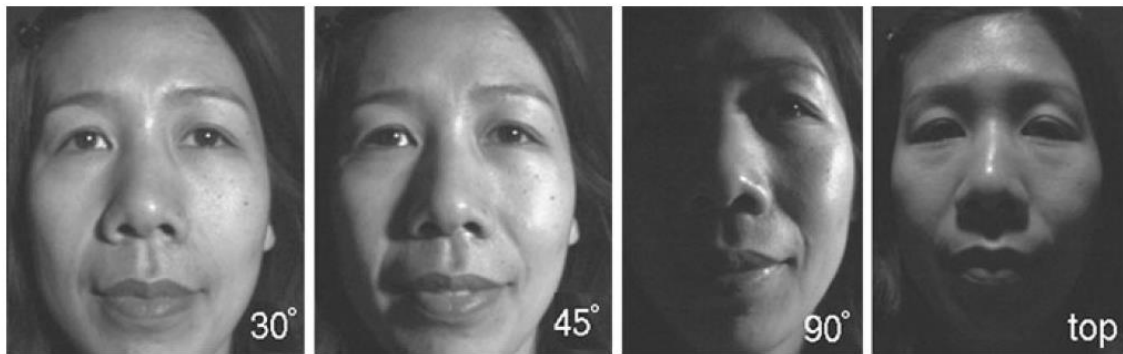
Fonte: Pedrini e Schwartz (2007)

1.2.1 Aquisição da Imagem

Segundo Moeslund (2012), o processo de conversão de uma imagem para uma matéria gerenciável é chamado de aquisição de imagem, e para esse processo ocorrer é necessária uma energia mensurável.

Para haver a captura de uma imagem, existe a necessidade de que alguma fonte de energia ilumine a área de interesse, e na maioria das vezes é aplicado uma luz com comprimento de onda pertencente ao espectro visível, mas outras faixas também podem ser aplicadas. O posicionamento geográfico dessa luz é de suma importância para a captura da imagem. Como apresentado na Figura 2, a diferença entre as quatro fotos são as posições da fonte de luz, no caso uma lâmpada (MOESLUND, 2012).

Figura 2: O efeito da iluminação de um rosto por quatro direções diferentes



Fonte: Moeslund (2012)

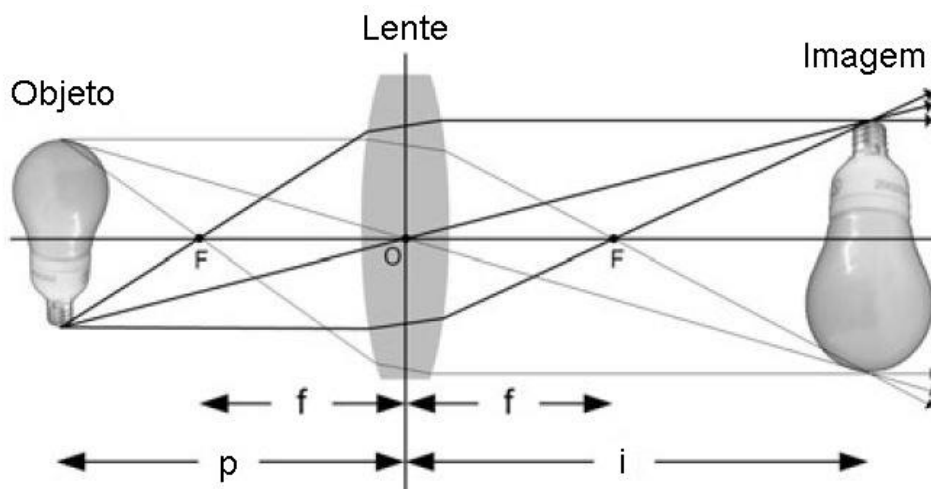
Com o objeto de interesse iluminado, a luz refletida do objeto pode ser capturada pelo sensor. Porém uma simples exposição de um sensor a um objeto faria com que raios de luz dos seus diferentes pontos atingissem o sensor, assim como pontos repetidos e de objetos periféricos ao de interesse. Essas condições resultam em uma imagem incorreta (MOESLUND, 2012).

Para solucionar esse problema é colocado uma espécie de barreira entre o objeto de interesse e o sensor, de forma a permitir que os raios de luz atravessem a barreira por um orifício, e para um maior aproveitamento dos raios, o orifício é substituído por um sistema óptico (MOESLUND, 2012).

Esse sistema é formado principalmente por lentes delgadas de superfície convexas, uma vez que estas têm a capacidade de convergir para um ponto focal dois raios paralelos que refratam a lente, como apresentado na Figura 3. A equação abaixo relaciona a distância focal f , distância do objeto p e a distância da imagem i (RESNICK; WALKER; HALLIDAY, 2016).

$$\frac{1}{f} = \frac{1}{p} + \frac{1}{i} \quad (2.0)$$

Figura 3: Os raios de luz refletidos por um objeto e focados por uma lente delgada convexa



Fonte: Adaptado de Moeslund (2012)

A fim de haver uma captação eficiente da imagem, o posicionamento do sensor deve acompanhar a distância i . E a partir da equação 2.0, pode-se notar que para o caso onde a distância f é fixa, a variação da distância do objeto à lente p também irá variar à distância da imagem i , e se a distância do sensor que capta a imagem não for alterada, o efeito é uma imagem fora de foco como apresentado na Figura 4 (MOESLUND, 2012).

Figura 4: Imagem captada com o sensor na distância correta (esquerda) e incorreta (direita)



Fonte: Moeslund (2012)

Com o intuito de se realizar a aquisição de imagens com o foco ajustado, pode-se utilizar sensores do tipo CCD (*Charged Coupled Device*), e cada um desses pode ser fabricado com diversas propriedades sensoras dispostas de forma matricial.

O arranjo desses sensores, posicionados com a distância correta do ponto focal f , produzem saídas proporcionais as integrais da energia luminosa projetadas sobre as superfícies de cada sensor (GONZALEZ; WOODS, 2009). E para o processamento digital desses sinais, dois passos são essenciais: a amostragem e a quantização (BOVIK, 2009).

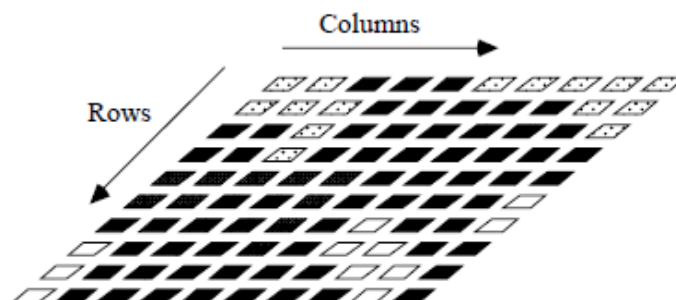
A amostragem é o processo de converter um sinal do espaço de tempo contínuo para o tempo discreto (BOVIK, 2009). Com o intuito de haver uma conversão completa do sinal, deve-se atender a inequação abaixo:

$$\Omega_s \geq 2\Omega_c \quad (3.0)$$

A inequação 3.0 apresenta a relação entre a frequência de amostragem Ω_s e a frequência do sinal amostrado Ω_c , onde $\Omega = 2\Omega_c$ é chamada de frequência de Nyquist do sinal no tempo contínuo (DINIZ; SILVA; NETTO, 2014).

A imagem digital amostrada é um vetor de com valores inteiros positivos organizados em um formato de linha e coluna, sendo que cada posição desse vetor é chamada de elemento de imagem, ou pixel, conforme Figura 5. A fim de simplificar o endereçamento computacional dessas amostras, o número de linhas e colunas desse vetor geralmente é uma potência de base 2 (BOVIK, 2009).

Figura 5: Representação de um vetor 10x10 de amostragem de uma imagem

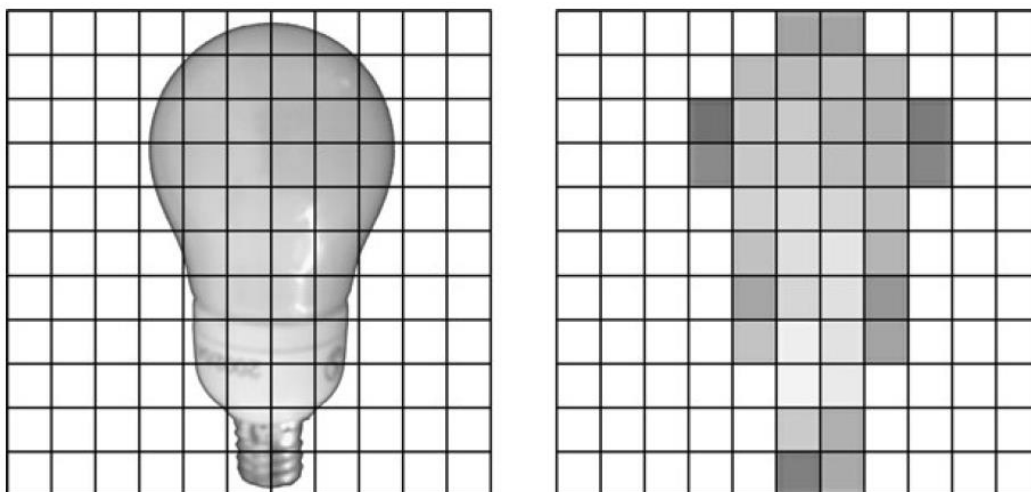


Fonte: Bovik (2009)

O valor de cada pixel desse vetor é definido pela quantização, e significa a intensidade que o fóton incide sobre o sensor, no caso deste ter frequência de leitura no espectro visível (BOVIK,2009). O resultado da amostragem e quantização de uma

imagem de valor contínuo para uma imagem de valor discreto, pode ser visto na Figura 06.

Figura 6: Amostragem e quantização de uma imagem de valor contínuo



Fonte: Moeslund (2012)

“A qualidade da imagem digital é claramente determinada, em grande parte, pelo número de amostras e de níveis discretos de intensidade utilizados na amostragem e na quantização. No entanto, o conteúdo da imagem também é um fator importante na escolha desses parâmetros.” (GONZALEZ; WOODS, 2010, p.35)

1.2.2 Pré - Processamento

A fim de remover pequenos detalhes e realizar a conexão de pequenas descontinuidades em uma imagem, utiliza-se filtros espaciais de pré-processamento para haver a suavização da imagem (GONZALEZ; WOODS, 2009). Para realizar esse processo, é necessário criar um filtro e escanear por toda a imagem, havendo a aplicação desse pixel a pixel (PEDRINI; SCHWARTZ, 2007)

Esse filtro é uma matriz bidimensional chamada de kernel, os coeficientes dessa matriz variam de acordo com a operação de filtragem utilizada (GONZALEZ; WOODS, 2009). No caso deste estudo, a operação de filtragem utilizada foi a função gaussiana, e a função para o cálculo dos coeficientes do filtro é (MOESLUND, 2012):

$$h(x, y) = e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (4.0)$$

Onde:

- x é a coordenada horizontal num plano cartesiano retangular com valor inteiro
- y é a coordenada vertical num plano cartesiano retangular com valor inteiro
- σ é o desvio padrão

Com o filtro definido, esse pode ser aplicado através da convolução da imagem com o filtro, através da equação:

$$g(x, y) = \sum_{j=-R}^R \sum_{i=-R}^R h(i, j) \cdot f(x + i, y + j) \quad (5.0)$$

Onde:

- $g(x, y)$ é a função imagem suavizada
- $h(i, j)$ é a função do filtro gaussiano
- $f(x, y)$ é função da imagem inicial

1.2.3 Segmentação da Imagem

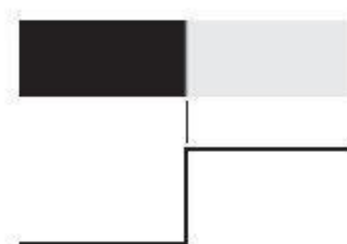
Com o objetivo de se obter as características relevantes da imagem, utilizam-se processos de segmentação para separar os dados de entrada em estruturas relevantes para a aplicação em questão, como afirma Pedrini e Schwartz (2007). Logo, a segmentação deve ser suficiente a partir do momento em que se tem as regiões de interesse da aplicação (GONZALEZ; WOODS, 2009).

Esse processo de segmentação da imagem pode-se ocorrer através da análise do nível de cinza da imagem, dessa forma busca-se detectar descontinuidades e similaridades na imagem (PEDRINI; SCHWARTZ, 2007).

1.2.3.1 Operador Canny

Segundo Gonzalez e Woods (2009) utiliza-se frequentemente a detecção de bordas com base em descontinuidades locais de intensidade, um dos modelos de borda existentes para analisar essa variação é a borda em degrau, cuja representação é apresentada na Figura 7.

Figura 7: Representação ideal de uma borda em degrau



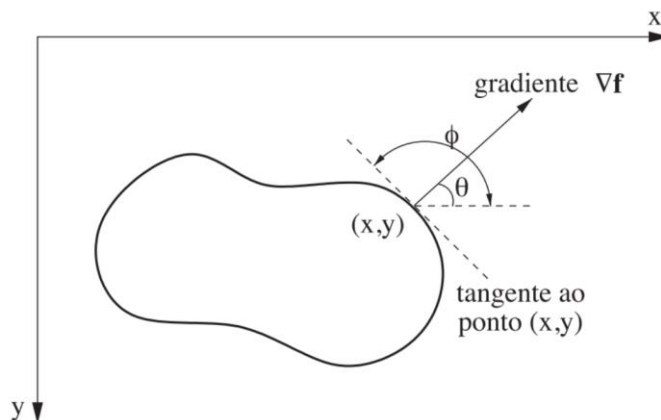
Fonte: GONZALEZ e WOODS (2010)

Um dos algoritmos mais complexos, porém, eficaz para a detecção de bordas foi desenvolvido por John Canny em 1986 e é denominado Canny (GONZALEZ; WOODS, 2009). O algoritmo tem três objetivos: ter uma baixa taxa de erros de detecção, uma boa localização dos pontos das bordas e a resposta de apenas um ponto de borda (BOVIK, 2009).

O algoritmo realiza a identificação das variações significativas dos níveis de cinza da imagem utilizando o operador gradiente, uma vez que as bordas da imagem podem ser expressas em derivadas parciais (PEDRINI; SCHWARTZ, 2007).

Para o caso de detecção de bordas, Pedrini e Schwartz (2007) caracterizam o gradiente como um vetor perpendicular à direção tangente da borda que indica a maior variação local do nível de cinza, como apresentado na Figura 8.

Figura 8: Gradiente em uma borda



Fonte: Pedrini e Schwartz (2007)

Após os pontos de magnitude localmente máxima na direção do gradiente serem conhecidos, o algoritmo elimina ruídos ainda presentes na detecção das bordas da imagem através de dois limiares, chamados T_1 e T_2 , sendo que $T_2 > T_1$ (PEDRINI; SCHWARTZ, 2007).

Esses limiares são utilizados para evitar a fragmentação das bordas em vários segmentos, uma vez que os pontos da borda com magnitude do gradiente maior que T_2 são mantidos como pontos da borda, assim como os pontos adjacentes a esses e com magnitude do gradiente superior a T_1 (PEDRINI; SCHWARTZ, 2007).

1.2.3.2 Transformada Hough

Outra técnica de segmentação de imagem utilizada é a Transformada de Hough, muito conveniente de ser aplicada em ambientes não estruturados, onde o âmbito comum de segmentação se dá pelos conjuntos de pixels estarem ou não na curva de um formato específico (GONZALEZ; WOODS, 2009).

Essa técnica consiste em mapear todos os pixels de uma imagem, e analisa o número de pixels que satisfazem uma reta com determinados parâmetros, como por exemplo a inclinação da reta e o ponto de interseção no eixo y , e assim a técnica identifica formatos a partir de conjunto de pontos com os mesmos parâmetros (BOVIK, 2009).

Além de linhas e retas, a técnica pode segmentar curvas. E a equação que

representa os pontos de uma circunferência é dado por (GONZALEZ; WOODS, 2009):

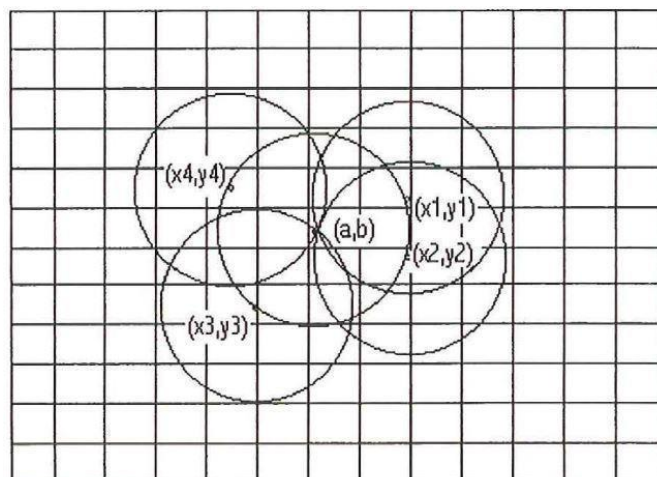
$$(x - c_1)^2 + (y - c_2)^2 = c_3^2 \quad (6.0)$$

Onde:

- x é um ponto qualquer da circunferência na coordenada horizontal de um plano cartesiano
- y é um ponto qualquer da circunferência na coordenada vertical de um plano cartesiano
- c_1 é um ponto central da circunferência na coordenada horizontal de um plano cartesiano
- c_2 é um ponto central da circunferência na coordenada vertical de um plano cartesiano
- c_3 é o raio da circunferência

Então a técnica mapeia todos os pixels da imagem transformando os em um espaço de parâmetros de três dimensões (c_1, c_2, c_3) com células em formato de cubo. A partir disso, os parâmetros c_1 e c_2 são incrementados a fim de obter o parâmetro c_3 conforme a equação 6.0. E os pixels que satisfazem a equação são acumulados na célula (GONZALEZ; WOODS, 2009). Por fim, a célula que possuir o maior número de curvas representa os parâmetros do círculo identificado, como apresentado na Figura 9.

Figura 9: Vários círculos passam pela célula correspondente ao centro do círculo identificado pela Transformada de Hough



Fonte: Marroni (2002)

Em ambas as técnicas apresentadas, ruídos podem causar erros nos métodos de segmentação de imagens, e estes causam distorções nos resultados dessa segmentação (PEDRINI; SCHWARTZ, 2007). Portanto, deve-se haver medidas de controle do ambiente a fim de diminuir a probabilidade de se obter uma segmentação imprecisa (GONZALEZ; WOODS, 2009).

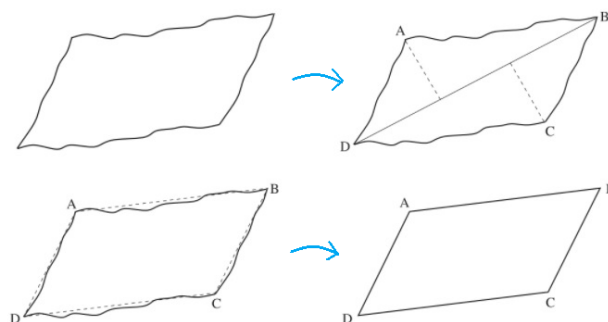
1.2.4 Representação e Descrição

Com a imagem já segmentada, segundo Pedrini e Schwartz (2007) deve-se haver a seleção do conjunto de características extraídas do objeto de interesse da imagem, sendo essa minimamente afetada por ruído, mudança de escalas, rotação e translação do objeto de interesse.

Uma técnica de representação utilizada nesse estudo é a aproximação poligonal, na qual a borda do objeto de análise pode ser aproximada a um polígono. Essa representação busca a essência da forma da borda, tendo como resultado o menor número possível de segmentos polinomiais (PEDRINI; SCHWARTZ, 2007).

A ferramenta consiste em subdividir sucessivamente um segmento de reta em duas partes até que a maior distância perpendicular entre o segmento da borda e a linha que une seus dois pontos extremos não exceda um determinado limiar, caso esse limiar seja ultrapassado, o ponto mais distante que esteja no segmento da borda se torna um vértice que subdivide o segmento inicial em dois segmentos (PEDRINI; SCHWARTZ, 2007). O caso é ilustrado na Figura 10, onde os segmentos de reta formados pelos pontos A e C até a linha AB (maior distância da borda), não excedem a distância de tal linha.

Figura 10: Aproximação poligonal



Fonte: Adaptado de Pedrini e Schwartz (2007)

1.3 Sistemas Embarcados

Sistemas embarcados são compostos por diversos componentes fixados em uma placa de circuito impresso, esses configurados corretamente permitem o funcionamento apropriado do mesmo (OLIVEIRA; ANDRADE, 2009). Esses sistemas são de suma importância para a transformação tecnológica que é vivenciado atualmente, visto que estão presentes em diversos segmentos de aplicação, como aparelhos de reprodução de imagens e som, leitor de código de barras, equipamentos de rede como roteadores, entre outros (GARCIA, 2018).

Com o intuito de obter conceitos fundamentais sobre esses sistemas, a seguir serão abordadas seções sobre o hardware e software desses sistemas.

1.3.1 Hardware

Um sistema embarcado é composto por uma placa de circuito impresso, unida a um circuito integrado de processamento com memórias, registradores, osciladores, contadores e outros componentes, orientados por um software a fim de executar uma função específica forma-se um sistema embarcado (OLIVEIRA; ANDRADE, 2009).

Segundo Oliveira e Andrade (2009) a memória é uma parte imprescindível para o funcionamento dos sistemas embarcados, visto que sem essa não é possível armazenar o software que orienta o funcionamento do sistema. Oliveira e Andrade (2009) destacam algumas tecnologias de fabricação das memórias:

Memória RAM (*Random Access Memory*): essa memória não tem a capacidade de manter os dados sem estar alimentada por energia elétrica, por isso é dita volátil. O seu aspecto aleatório advém da capacidade de acessar o endereço de destino de

leitura ou gravação de forma direta, ou seja, sem a necessidade de percorrer os outros endereços de memória.

Memória ROM (*Read Only Memory*): essa memória não permite operações de escrita apenas leitura, uma vez que é programada pelo próprio fabricante. Fatos que justificam essa tecnologia de fabricação ser não volátil, ou seja, tem a capacidade de manter os dados sem a memória estar alimentada por energia elétrica.

Memória PROM (*Programmable Read Only Memory*): memória não volátil que permite apenas operações de leitura, porém por ser fabricada ausente de dados, permite uma única gravação ao usuário.

Memória EPROM (*Electrically Programmable Read Only Memory*): essa memória faz parte da tecnologia de memória PROM, porém permite diversas gravações, condicionadas a exclusão dos dados presentes na memória através de um processo especial, como luz ultravioleta.

Memória EEPROM (*Electrically Erasable Programmable Read Only Memory*): é uma memória do tipo ROM que permite a reprogramação e exclusão dos dados via processos elétricos.

Memória FLASH: essa tecnologia de memória é não volátil e tem alta capacidade de armazenamento, além de permitir a reprogramação e exclusão dos dados via processos elétricos de maneira mais rápida se comparado com as memórias EEPROM.

Segundo Oliveira e Andrade (2009), em sistemas embarcados, as memórias são utilizadas em três seções: memória de programa, memória de dados e memória de armazenamento.

A memória de programa costuma ser uma memória FLASH, portanto não volátil e responsável por armazenar o firmware do sistema embarcado assim como o software que pode ser diversas vezes reprogramado.

Já os registros e variáveis de processamento temporário são armazenados na memória de dados, haja vista que não há a necessidade de armazenamento fixo dos mesmos, fato que assemelha a memória de dados com a memória RAM.

Para se ter um armazenamento de dados de forma segura aliado a possibilidade de realizar operações de escrita e leitura, são utilizadas pequenas memórias não voláteis do tipo EEPROM, essas armazenam dados de processadores

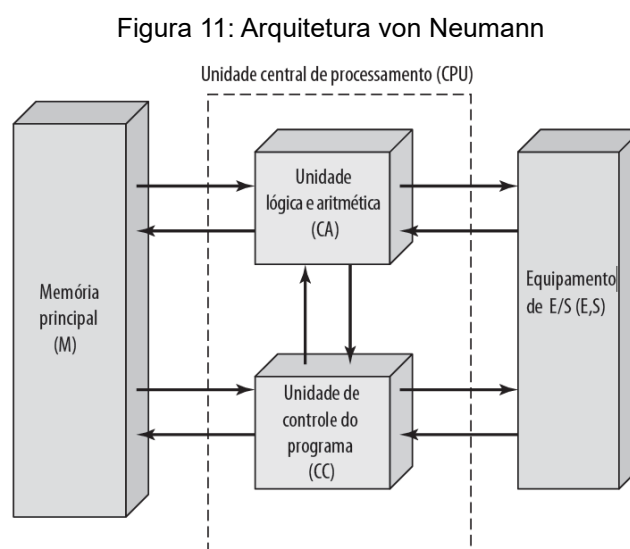
e periféricos internos.

Todas essas seções de memória citadas são ligadas aos demais componentes do sistema embarcado através de barramentos, esses são organizados conforme a arquitetura utilizada.

1.3.2 Arquitetura e Processadores

Segundo Sttalings (2010), a arquitetura de um computador rege os principais parâmetros relativos à execução lógica de um programa, como o conjunto de instruções, mecanismos de entrada e saída e técnicas de endereçamento de memória.

A primeira arquitetura de computador de impacto tecnológico foi a arquitetura de Von Neumann, essa tinha uma estrutura geral composta por uma memória principal que tem como função armazenar dados e instruções de máquina, uma unidade lógica e aritmética (ALU) que opera sobre os dados binários, uma unidade de controle na qual interpreta e executa as instruções da memória, e um sistema de entradas e saídas (STTALINGS, 2010). A representação do funcionamento dessa arquitetura é apresentada na Figura 11.



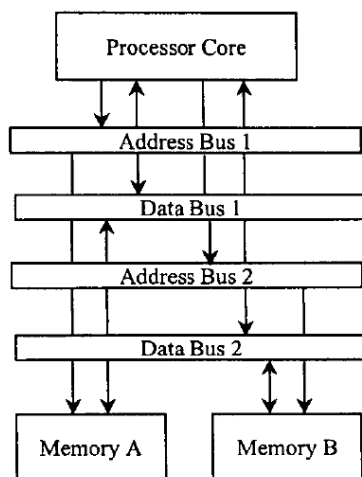
Fonte: Sttalings (2010)

Segundo Oliveira (2014), os processadores que comumente fazem uso dessa

arquitetura são aqueles da tecnologia de projeto CISC (*Complex Instruction Set Computing*) e Machado e Maia (2013) complementam afirmando que esses possuem um baixo número de registradores e as instruções são interpretadas por microprogramas. Um exemplo de processador dessa tecnologia é a x86 da Intel (OLIVEIRA; ANDRADE, 2009).

Outra arquitetura muito difundida na eletrônica digital é a arquitetura Harvard, essa arquitetura é composta por um processador conectado a duas memórias independentes, comumente sendo uma memória de programa e outra memória de dados, conforme apresentado na Figura 12 (KULARATNA, 2000).

Figura 12: Arquitetura de Harvard



Fonte: Kularatna (2000)

A grande diferença entre as duas arquiteturas citadas é o método de processamento dos dados das memórias programa e de dados. Segundo Oliveira e Andrade (2009), a arquitetura von Neumann possui apenas um barramento para acessar ambas as memórias, enquanto a arquitetura Harvard possui um barramento para a memória de dados e outro para a de programa. Dessa forma, pode-se acessar duas memórias em um mesmo ciclo de instrução (KULARATNA, 2000).

Por essa eficiência no processamento das instruções, a arquitetura Harvard é comumente utilizada em processadores com tecnologia de projeto RISC (*Reduce Instruction Set Computing*) (OLIVEIRA, 2014). Esses processadores possuem poucas instruções de máquina e são executadas diretamente no hardware (MACHADO; MAIA, 2013). Um exemplo de processador dessa tecnologia é o ARM (OLIVEIRA;

ANDRADE, 2009).

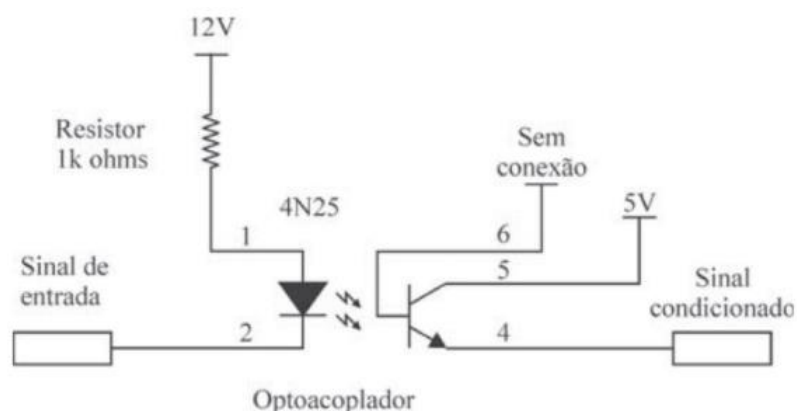
1.3.3 Interface

Segundo Oliveira e Andrade (2009), é necessário o uso de interfaces nos sistemas embarcados para haver a comunicação dos sinais de entradas e saídas da unidade de controle desses sistemas.

“As interfaces de saída correspondem aquelas que convertem os sinais de saída da unidade de controle do sistema embarcado nos sinais apropriados para o dispositivo a ser controlado.” (OLIVEIRA; ANDRADE, 2009, p.43). Oliveira e Andrade (2009) cita que é possível a utilização de transistores, relés e optoacopladores para condicionar o sinal que advém da unidade de processamento para a carga apropriada.

“As interfaces de entrada correspondem às que convertem os sinais dos dispositivos de referências (sensores), comunicação ou outro tipo nos sinais apropriados para a entrada na unidade de controle do sistema embarcado.” (OLIVEIRA; ANDRADE, 2009, p.50). Segundo o Oliveira e Andrade (2009) os optoacopladores são utilizados para realizar a interface do sinal que advém de um dispositivo externo para a unidade de processamento, condicionando o sinal. Um exemplo de condicionamento de sinal de entrada é apresentado na Figura 13.

Figura 13: Exemplo de condicionamento de sinal de entrada



Fonte: Oliveira e Andrade (2009)

1.3.4 Comunicação

A interface de comunicação de dados é a fronteira de entrada e saída de equipamentos, formada por um dispositivo que possibilita a troca de dados entre esses equipamentos. Essas interfaces podem ser divididas em duas grandes categorias, seriais e paralelas (COSTA, 2005).

Segundo Sttalings (2010), na interface paralela existem múltiplos condutores conectados a interface de entrada e saída e o equipamento no qual a comunicação é estabelecida, e múltiplos bits de dados são transferidos de maneira simultânea através deste barramento de dados. Já na comunicação serial, apenas um condutor é utilizado para transmitir a informação.

Apesar da transmissão paralela se demonstrar mais rápida, com a evolução das interfaces seriais de alta velocidade, o uso de interfaces paralelas vem se tornando incomum (STTALINGS, 2010).

Dentre as interfaces de comunicação utilizadas por sistemas embarcados, pode-se destacar a I²C (*Inter-Integrated Circuit*) e a SPI (*Serial peripheral interface*).

A interface I²C (*Inter-Integrated Circuit*), como o próprio nome sugere, tem a função de estabelecer a comunicação entre diversos circuitos integrados. É um barramento bidirecional que possui dois canais de comunicação seriais, um canal *Serial* de dados e outro de *Clock* (NXP SEMICONDUCTORS, 2014).

Outro barramento existente é o SPI (*Serial peripheral interface*) e tem a função de estabelecer uma comunicação serial síncrona de curta distância, esse protocolo é bidirecional e utiliza pelo menos 4 canais de comunicação: o canal MOSI (*Master Output, Slave Input*) que realiza a transmissão do mestre da rede, MISO (*Master Input, Slave Output*) canal de recepção do mestre da rede, SCLK (*Serial Clock*) canal responsável pelo *Clock* da rede e os canais SS (*Slave Select*) responsáveis pela seleção do escravo da rede (ALMEIDA; MORAES; SERAPHIM, 2016).

1.3.5 Software

Oliveira e Andrade (2009) afirmam que o software de um sistema embarcado é denominado de firmware, que de maneira genérica é o conjunto de regras que regem o funcionamento do sistema embarcado.

Para que o software possa ser escrito existe a necessidade de uma linguagem de programação, essa segundo Rockcontent (2019) é uma série de instruções que permitem ao programador escrever ordens, ações consecutivas e algoritmos conforme um comportamento desejado para o equipamento em desenvolvimento.

Segundo Oliveira e Andrade (2009) essas linguagens de programação podem ser divididas em dois grupos: as linguagens de baixo e alto nível. A linguagem de baixo nível é aquela que atua diretamente nos registradores do processador, sem a necessidade de conversão para a compreensão do mesmo, um exemplo de linguagem de baixo nível é a linguagem Assembly.

Já a linguagem de alto nível é aquela que abstrai diversas estruturas de controle e são implementadas de forma automática nas camadas inferiores através de compiladores, que por sua vez podem tornar as instruções maiores do que um mesmo algoritmo implementado em linguagem de baixo nível (OLIVEIRA; ANDRADE, 2009).

Haja vista que o software está armazenado na memória de programa, o mesmo pode ser executado. Os processadores costumam executar uma instrução por vez sequencialmente, quando o programa atinge a última instrução o processador pode não realizar mais nenhuma operação assim como iniciar novamente quantas vezes forem necessárias (OLIVEIRA; ANDRADE, 2009).

1.4 Medição dimensional por visão computacional

Haja vista que a inspeção visual é uma das mais antigas técnicas de avaliação de produtos e que está intimamente ligado com a produção industrial, a área de visão computacional surge como uma ferramenta versátil para o desenvolvimento de ensaios não destrutivos (END) (LETA et al., 2016).

Essa ferramenta pode ser aplicada em análise de conformidade de peças, busca de defeitos de fabricação, análise final de qualidade do produto, entre outras soluções. Além disso, o uso de visão computacional em processos de inspeção pode oferecer repetitividade em medições sem contato e exatidão, apesar de essa variar de acordo com a aplicação (LETA et al., 2016).

Leta et. al. (2016) apontam que a medição dimensional por imagem pode ocorrer seguindo alguns métodos. Um deles é o uso da imagem como auxiliar à

medição realizada por instrumentos de medida, como micrômetros e guias lineares, um exemplo são os projetores de perfil.

Outro método baseia-se na obtenção de uma relação de pixels por unidade de comprimento, através da análise de uma imagem que contenha um padrão com dimensões conhecidas, esse método é denominado de calibração de pixel (LETA et al., 2016).

Com o objetivo de obter uma medição com alta exatidão, diminuindo-se as incertezas, o terceiro método apresentado por Leta et. al. (2016) é a junção dos dois métodos já citados, utilizando instrumentos de medição de deslocamento e a calibração de pixel.

Visto que no presente estudo o método utilizado é a calibração do pixel, nos itens a seguir será detalhado sobre esse método, além de erros e incertezas de medição em visão computacional.

1.4.1 Calibração do Pixel

A calibração do pixel é um método de medição dimensional por imagem que tem como objetivo determinar a dimensão em unidade de comprimento de um pixel. Essa medição depende de alguns fatores, dentre eles pode-se destacar: a resolução da câmera, a ampliação da lente e a distância entre a câmera e o objeto a se medir (LETA et al., 2016).

A resolução da câmera se apresenta como um fator importante visto que a essa influência no comprimento do pixel, calculado através da razão entre a dimensão em unidade de comprimento do objeto medido e a quantidade de pixels que esse ocupa na imagem (LETA et al., 2016).

1.4.2 Erros em medições por imagem

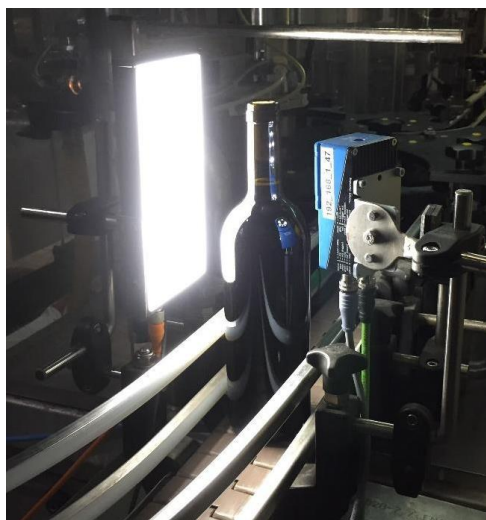
O erro de medição é definido por Junior e Sousa (2018, p.38) como “a diferença entre o valor indicado pelo sistema de medição e o valor verdadeiro do mensurando”. Em um sistema de medição por imagem esses erros podem ter origem do ambiente, operador, procedimento, instrumento e material (LETA et al., 2016).

Esses erros podem ter ocorrência sistemática, definida como a parcela previsível do erro e corresponde ao erro médio, ou aleatória, referente a parcela imprevisível do erro, na qual a repetição tende a levar a resultados diferentes (OIML, 2007).

Apesar de as medições serem automáticas em um sistema de medição por imagem, alguns parâmetros podem causar divergências consideráveis no resultado, e, portanto, deve-se realizar um procedimento de verificação e validação dos resultados (LETA et al., 2016).

Um desses fatores é a iluminação, como já citado no presente estudo, é de suma importância para a aquisição das imagens, na qual visa destacar as características do objeto sem adulterá-las, um exemplo são os objetos de vidro, que devido a seu alto índice de reflexão tornam-se de difícil leitura, porém com uma iluminação adequada pode destacar o conteúdo do vidro possibilitando o processamento da imagem (LETA et al., 2016). Um exemplo de sistema de iluminação que visa destacar o conteúdo interno da garrafa de vidro é apresentado na Figura 14.

Figura 14: Uso de iluminação diascópica destacando o conteúdo interno da garrafa



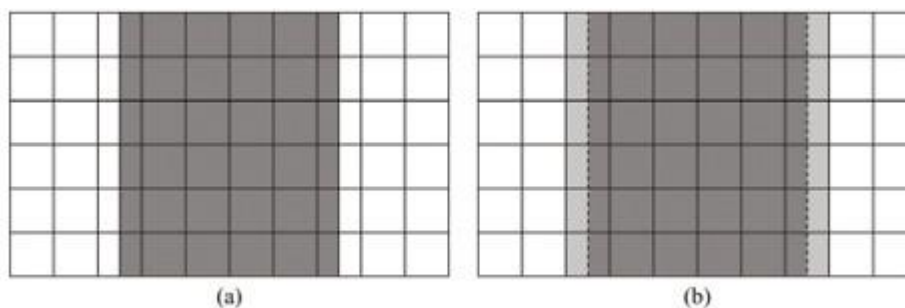
Fonte: ServiAutomation ([2020?])

Outro aspecto determinante para a aquisição da imagem são parâmetros que influenciam a formação da imagem no sensor, como por exemplo: distância focal, abertura, profundidade de campo, tempo de exposição e valor de exposição. Esses podem gerar erros que influenciam diretamente na etapa de segmentação da imagem (LETA et al., 2016).

Segundo Leta et. al. (2016) quando existe um desalinhamento do objeto a ser medido em relação ao eixo de medição do instrumento ocorre o chamado erro de cosseno, outro erro muito comum em sistemas de medição dimensionais e para o caso de sistemas de medição por imagem pode ser causado pelo desalinhamento entre o objeto e a câmera, quando se tem erro de foco, ou pelo desalinhamento do plano perpendicular a câmera, quando a peça não está alinhada ao eixo de medição (LETA et al., 2016).

Além desses citados, pode-se destacar a importância da distinção dos objetos e suas bordas no processamento da imagem, visto que isso é um ponto fundamental para a exatidão do resultado, e a resolução da imagem está diretamente ligada a localização da borda de um objeto, visto que essa borda é discretizada qualquer parte do pixel que estiver ocupada parcialmente será preenchida por completo, como apresentado na Figura 15. Logo, deve-se considerar a incerteza de pelo menos 1 pixel na determinação da borda do objeto (LETA et al., 2016).

Figura 15: Exemplo de um objeto antes(a) e após a digitalização da imagem (b)



Fonte: Leta et. al. (2016)

1.4.3 Incerteza na medição por imagem

Para garantir a confiabilidade do resultado de um processo de medição, é expressada a incerteza de medição, e para que essa incerteza seja quantificada de maneira correta, deve-se conhecer o que a afeta (LETA et al., 2016).

Com o intuito de analisar a incerteza total do sistema de medição, é necessário tratar separadamente as fontes de incerteza, e identificar a respectiva sua contribuição para o total (LETA et al., 2016).

Para mensurar essas componentes de incerteza, segundo Leta et. al. (2016)

pode-se defini-los em dois grupos: através de valores de medições obedecendo a distribuições estatísticas ou através de certificados de calibrações, medições anteriores, julgamento científico, etc.

Leta et. al. (2016) também defende que para a determinação da incerteza de medição de um sistema de medição por imagem, devem ser analisadas fontes de erro comuns como: dilatação do padrão, erro de alinhamento, repetibilidade de medições, entre outras.

2 DESENVOLVIMENTO

Diante dos conceitos apresentados, neste capítulo serão abordados detalhes sobre a metodologia do presente trabalho, incluindo as ferramentas empregadas, características das placas Raspberry Pi 3 e Raspberry Pi 4, técnicas utilizadas para avaliação de parâmetros de câmeras, e detalhes do algoritmo de medição bidimensional de produtos manufaturados.

2.1 Escopo do Trabalho

Esse estudo abordou um escopo bem definido de aplicação para o sistema embarcado de visão que envolve a medição bidimensional, ou seja, largura e comprimento, de produtos manufaturados. Em primeiro momento, para a obtenção de parâmetros acerca de câmeras foi utilizado um ambiente de teste.

2.2 Ferramentas Utilizadas

Nessa seção do trabalho, estão listadas as ferramentas utilizadas para o desenvolvimento deste estudo, sendo os computadores Raspberry Pi 3 e Raspberry Pi 4 os principais elementos de hardware e a biblioteca OpenCV para o uso das técnicas de processamento digital de imagem já citadas neste estudo.

2.2.1 Raspberry Pi

O sistema de visão desenvolvido foi testado através de computadores Raspberry Pi, estes podem ser considerados como placas eletrônicas com poder de computação considerável e com dimensões de um cartão de crédito (UPTON; HALFACREE, 2017).

Esses dispositivos possuem saídas de vídeo, saída de áudio, interface Universal *Serial* Bus (USB) e Ethernet, assim como computadores comuns (UPTON; HALFACREE, 2017).

A criação dessas placas ocorreu no Reino Unido com a intenção curta e

despretensiosa de ser uma ferramenta de aprendizado básico em linguagem de programação a ingressantes do curso de Ciências da Computação na Universidade Cambridge, uma vez que esses estudantes apresentaram um declínio nas habilidades de programação (UPTON; HALFACREE, 2017).

Foi nesse contexto que surgiu a Raspberry Pi Foundation, fundada por Eben Upton, dr. Rob Mullins, Alan Mycroft, Jack Lang, Pete Lomas e David Braben, com o intuito de ser uma pequena instituição beneficente com grandes ideias. Em 2011, após alguns anos de desenvolvimento e melhorias no projeto do primeiro Raspberry Pi, a placa foi lançada, alcançando o sucesso no ambiente da tecnologia (UPTON; HALFACREE, 2017).

Dessa forma, o uso desse computador de propósito geral se aplica para diversas situações: um sistema de segurança doméstico, estação meteorológica, avião de controle remoto, relógios e termômetros digitais, entre outros (DONAT,2018).

Essa ampla rede de aplicações se dá por algumas características: o uso do módulo *system-on-chip* (SoC) que fornece ao Raspberry Pi o gerenciamento de processamento geral, renderização gráfica e recursos de entrada/saída aliado com interfaces USB, entradas de vídeo, entrada de áudio e conectores de entradas e saídas para uso geral, grande diferencial dos computadores comuns (UPTON; HALFACREE, 2017).

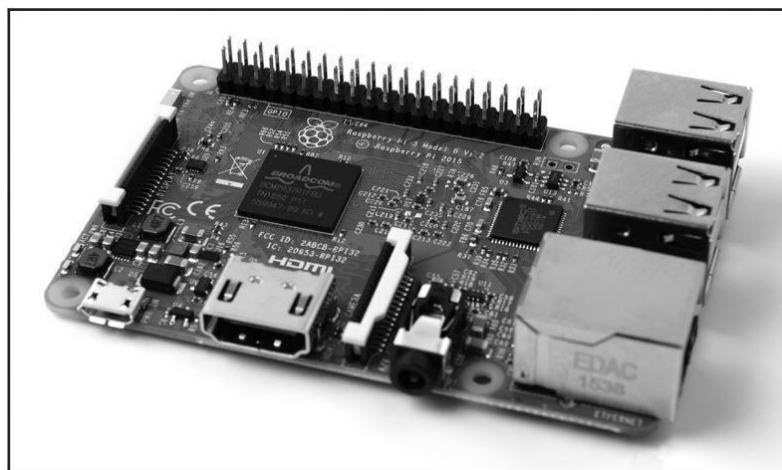
Além disso, consolidando-se com um computador, os modelos do Raspberry Pi possuem um sistema operacional chamado Raspbian, uma variante personalizada da distribuição de Linux Debian, que possui excelente desempenho, estabilidade e alta compatibilidade (UPTON; HALFACREE, 2017).

O modelo Raspberry Pi 3, apresentado na Figura 16, possui o SoC (*System on a Chip*) BCM2837 da marca Broadcom, com o processador quad-core ARM Cortex A53 que funciona à 1.2GHz, e tem velocidade de processamento 50% maior em relação ao modelo anterior Raspberry Pi 2. Além disso, esse dispositivo é capaz de realizar conexão sem fio através de Wi-fi de 2,4 GHz e Bluetooth (RASPBerry PI FOUNDATION, 2021).

Esse dispositivo possui 1 GB de memória RAM (*random-access memory*), 4 portas USB (*Universal Serial Bus*) 2.0, uma GPU (*Graphics Processing Unit*) VideoCore IV que funciona à 400MHz (RASPBerry PI FOUNDATION, 2021).

Também possui 40 pinos de entradas e saídas, esse conjunto de pinos denomina-se de porta GPIO (*general-purpose Input-Output*). Segundo Upton e Halfacree (2017) cada pino tem sua funcionalidade própria, e esses trabalham em conjunto para formar interfaces específicas.

Figura 16: Raspberry Pi 3



Fonte: Upton e Halfacree (2017)

Outro modelo presente neste estudo foi o Raspberry Pi 4, este modelo evoluiu em relação ao seu antecessor – Raspberry Pi3 - em alguns aspectos: possui um SoC BCM2711 da marca Broadcom, com o processador quad-core ARM Cortex-A72 que funciona à 1.5GHz, além de uma memória RAM de 2 GB e uma GPU VideoCore de 500 MHz com suporte à resolução 4K (RASPBerry PI FOUNDATION, 2021).

A fim de ampliar o conhecimento acerca do hardware do Raspberry Pi, nas próximas seções serão explorados detalhes do processador e interface GPIO do dispositivo.

2.2.1.1 O processador ARM

Segundo Machado e Maia (2013), o processador é a unidade responsável por realizar o gerenciamento de todo o sistema computacional, e tem como principal função controlar e executar instruções presentes na memória principal através de operações básicas de adição, subtração, comparação e movimentação de dados.

O Raspberry Pi utiliza a arquitetura de processadores ARM (*Advanced RISC*

Machine). Esses processadores utilizam a tecnologia de projeto RISC (*Reduce Instruction Set Computing*), na qual tem instruções muito simplificadas, de formato fixo e possuem apenas um ciclo de *Clock* durante a execução. Essas características refletem no desempenho limitado, uma vez que possui menor quantidade de instruções disponíveis, mas com uma alta eficiência desses processadores, visto que possuem baixo gasto de energia, baixa dissipação de calor e uma otimização das instruções (OLIVEIRA; ANDRADE, 2009).

Essas características aliadas com o pequeno die (bloco de material semiconductor – circuito integrado) possibilitaram o uso massivo dos processadores ARM em sistemas embarcados (GARCIA, 2017). No fim de 2019, a Arm Holdings anunciou a venda de 160 bilhões de chips com seu processador ARM embarcado (LIMA, 2020).

Segundo Garcia (2017) a arquitetura ARM já passou por diversas modificações a fim de atender demandas de mercado, como tecnologias de segurança, virtualização e criptografia. Esses processadores possuem diversas famílias, e essa divisão ocorre de acordo com a área de aplicação.

A família ARM CORTEX foi lançada na sétima geração da arquitetura ARM e destaca-se pela capacidade de atingir níveis de performance altamente superior se comparado com as gerações anteriores, essa família é dividida em três grupos: CORTEX-A, CORTEX-R e CORTEX-M (PEREIRA, 2009).

Segundo Pereira (2009) a família CORTEX-A possui processadores focados em aplicações de alta performance baseadas em sistemas operacionais que utilizam sistemas de memória virtual, como aplicações mobile e plataformas (GARCIA, 2017).

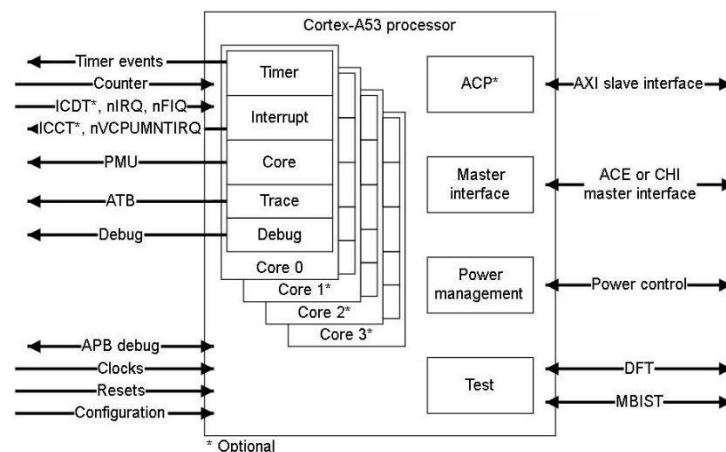
Pereira (2009) defende que a família CORTEX-R é utilizada em aplicações de tempo real e que desejam alta performance, como sistemas embarcados do ramo automotivo e de controle industrial (GARCIA, 2017).

Já a família CORTEX-M é a versão mais simplificada da família e utilizada em aplicações de baixo custo e alta eficiência energética, como microcontroladores (PEREIRA, 2009).

Apesar de ter sido lançada na sétima geração da arquitetura, a família ARM CORTEX se manteve na oitava geração e pretende-se manter na nona: ARMv9, com previsão para lançamento em 2022 (ALECRIM, 2021).

O processador ARM Cortex-A53, utilizado no Raspberry Pi 3, e o processador ARM Cortex-A72, utilizado no Raspberry Pi 4, são processadores projetados para dispositivos que visam realizar tarefas computacionais complexas, capaz de hospedar um sistema operacional com suporte a diversos softwares. Além disso, têm arquitetura 64-bit e um baixo consumo energético (ARM HOLDINGS, 2021). Um exemplo de configuração do processador Cortex-A53 é apresentado na Figura 17.

Figura 17: Exemplo de configuração do processador Cortex-A53



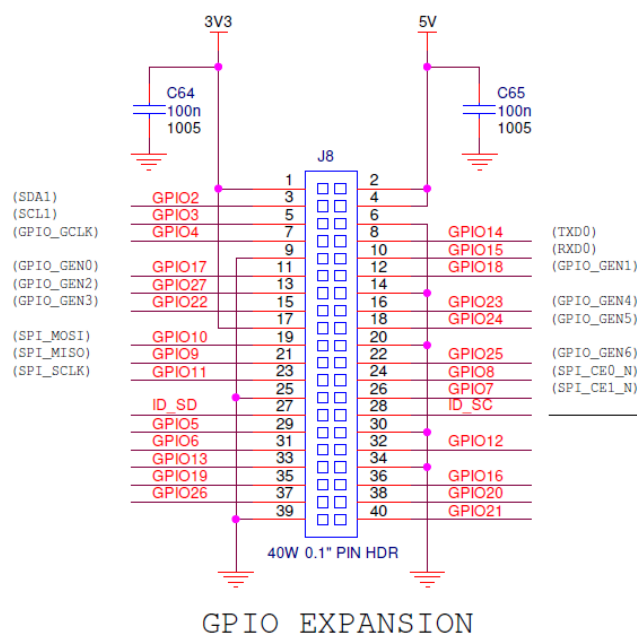
Fonte: ARM Holdings (2018)

2.2.1.2 A interface GPIO

Segundo Donat (2019), a porta GPIO (*General Purpose Input/Output*) é uma interface que permite o Raspberry Pi se conectar a diversas extensões físicas, como leds, servomotores, placas de expansão, sensores e outros dispositivos. A porta GPIO é disposta na parte superior esquerda do Raspberry Pi 3 e Raspberry Pi 4 e contém 2 fileiras de 20 pinos header do tipo macho espaçados em 2,54 mm, como apresentado na Figura 18 (UPTON; HALFACREE, 2017).

Donat (2019) defende que a interface GPIO é um grande diferencial do Raspberry Pi aos computadores comuns, uma vez que com a diminuição da disponibilidade de interfaces seriais nos computadores mais novos, tornou-se muito mais difícil essa interface com dispositivos externos, visto que a linguagem de baixo nível para o controle das portas USB não é simples e intuitiva.

Figura 18: A Porta GPIO



Fonte: Raspberry Pi Fundation e Adams (2014)

Como apresentado na Figura 17, a porta GPIO possui pinos com tensão igual a 5V (pino 2 e 4), 3.3 V (pinos 1 e 17) e 0 V (pino 6, 9, 14, 20, 25, 30, 34 e 39), porém uma vez que a interface GPIO é conectada ao SoC do Raspberry Pi, e esse opera em lógica 3.3 V, os pinos dessa interface nunca devem ser conectados a fontes de tensão superior a 3.3 V ou curto-circuitados, pois se corre o risco de danificar todo o Raspberry Pi (UPTON; HALFACREE, 2017).

A porta GPIO também disponibiliza pinos para uso geral, e esses podem ser utilizados como entrada ou saída digital, respeitando os níveis de tensão para os respectivos estados lógicos: alto para 3.3 V e baixo para 0 V (UPTON; HALFACREE, 2017).

Além dos pinos de uso geral, a porta GPIO também possui pinos dedicados a barramentos de comunicação a dispositivos externos. Um desses barramentos é o *Serial UART (Universal Asynchronous Receiver/Transmitter)*, ele realiza comunicação *Serial* através do método *full-duplex*, ou seja, dois canais de comunicação independentes, e operam com a mesma taxa de transferência e o mesmo tamanho de dados (OLIVEIRA; ANDRADE, 2009). Nas placas Raspberry Pi estudadas, os pinos 8 e 10 são dedicados para essa comunicação, sendo que o pino 8 tem a função de transmissor de dados e o pino 10 de receptor de dados (UPTON; HALFACREE,

2017).

O Raspberry Pi também possui o barramento I²C (*Inter-Integrated Circuit*) e estabelece a comunicação do SoC e dispositivos como a câmera Raspberry Pi e sensores e os pinos utilizados são: pino 3 que fornece o sinal de dados e o pino 5 que fornece o sinal de *Clock* (UPTON; HALFACREE, 2017).

Outro barramento disponível para uso na GPIO é o barramento SPI (*Serial peripheral interface*) e tem a função de estabelecer uma comunicação serial síncrona de curta distância. No Raspberry Pi, o pino 19 utiliza o canal MOSI, o pino 21 o canal MISO, o pino 23 o canal SLCK e os pinos 24 e 26 como canais SS, uma vez que possibilita a comunicação com até dois escravos (UPTON; HALFACREE, 2017).

2.2.2 Câmera webcam Logitech C270

Um dos modelos de câmeras webcam utilizados no desenvolvimento deste estudo foi o C270 da marca Logitech, apresentada na Figura 19. Essa câmera tem uma resolução máxima de 1280 x 720 pixels para a aquisição de imagens e a taxa máxima de 30 frames por segundos para a captura de vídeos (LOGITECH, 2021).

Com relação às características construtivas desse modelo, possui o foco da câmera regulável através de potenciômetro, lente de plástico e campo de visão diagonal desse modelo é de 55°. Além disso, a câmera é capaz de capturar imagens na faixa de luz visível do espectro magnético e seu sensor tem capacidade de capturar 0,9 megapixels (LOGITECH, 2021).

Figura 19: Logitech C270



Fonte: Logitech (2021)

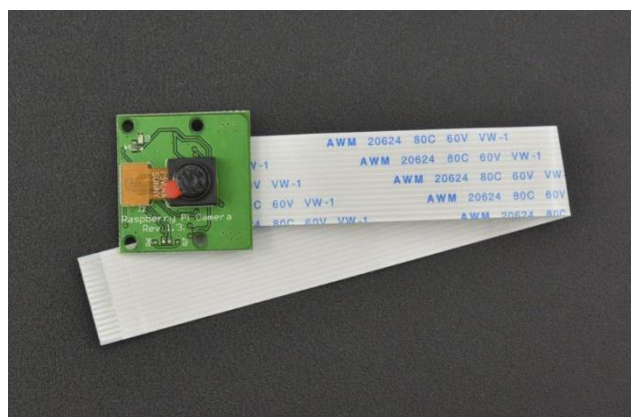
2.2.3 Raspberry Pi Camera v1

O módulo de câmera Raspberry Pi Camera v1, apresentada na Figura 20, também foi utilizado neste estudo e é dedicado exclusivamente aos dispositivos Raspberry Pi. Esse módulo tem como sensor de imagem o chip do tipo CMOS modelo OV5647 da marca OmniVision, que tem a capacidade de capturar imagens com uma resolução máxima de 2592 x 1944 pixels e vídeos com uma taxa que varia de 30 a 90 quadros por segundo, dependendo da resolução do vídeo (RASPBERRY PI FOUNDATION, 2021).

O módulo utilizado pertence a revisão 1.3 e dessa forma o sensor de imagem já citado tem a capacidade de capturar imagens de 5 megapixels. Já os módulos de câmera com revisão superior a 2.0 podem capturar imagens com resolução mais alta, de até 8 megapixels (UPTON; HALFACREE, 2017).

Com relação ao foco desse sensor de imagem, o mesmo é fixo e tem uma distância focal de 3,6 +/- 0,01 mm, campo horizontal de visão de 53,50 +/- 0,13 ° e campo vertical de visão de 41,41 +/- 0,11 ° (RASPBERRY PI FOUNDATION, 2021).

Figura 20: Raspberry Pi Camera v1 versão 1.3



Fonte: CIRCUITROCKS (2021)

2.2.4 OpenCV

OpenCV é uma biblioteca de código aberto escrita na linguagem C e C++ utilizada para o desenvolvimento de algoritmos de visão computacional. Essa biblioteca tem a capacidade de ser executada em diferentes sistemas operacionais como Linux, Windows e Mac OS X (BRADSKI; KAEHLER, 2008).

“Um dos objetivos do OpenCV é fornecer uma infraestrutura simples de visão computacional que ajude pessoas a desenvolver aplicações bastante sofisticadas de visão computacional rapidamente. A biblioteca OpenCV contém mais de 500 funções que abrangem várias áreas de visão computacional, incluindo inspeção de produtos industriais, imagens médicas, segurança, interface com usuário, calibração de câmeras, imagens estéreo e robótica.” (BRADSKI; KAEHLER, 2008, p.01).

Segundo Bradski e Kaehler (2008) a biblioteca permite que sejam desenvolvidos produtos comerciais sem a necessidade de haver código aberto ou retorno de lucros para o domínio público. Além disso, o OpenCV possui interface para as linguagens C, C++, Python e Java.

No desenvolvimento deste trabalho, foi utilizado a biblioteca OpenCV visto que existe vasta documentação, não tem custo, é compatível com a linguagem Python 3 além de ser uma biblioteca de código aberto (BRADSKI; KAEHLER, 2008).

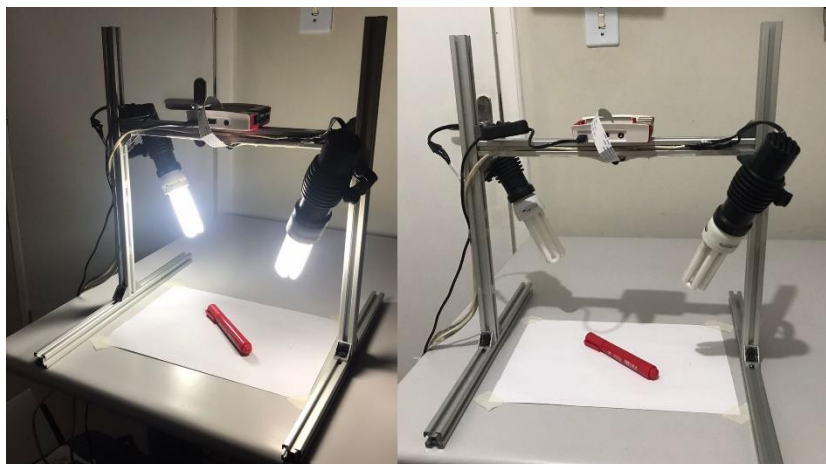
2.2.5 Estrutura do Sistema Embarcado de Visão

Com o objetivo de criar uma estrutura para ser realizado a avaliação das câmeras, dos sistemas de iluminação e do sistema embarcado de visão, foi construída uma estrutura metálica utilizando-se perfis de alumínio com largura e altura de 20 mm, apresentada na Figura 21.

Essa estrutura tem um suporte horizontal regulável para a câmera, fato que possibilita a câmera ser fixada em até 40 cm de altura em relação à base da estrutura.

Além disso, foram fixados dois receptáculos com altura regulável para comportar as lâmpadas do sistema de iluminação.

Figura 21: Ambiente de captura das imagens



Fonte: O Autor (2021)

2.3 Avaliação das Câmeras

Com o intuito de realizar a avaliação das câmeras Raspberry Pi Camera v1 e Logitech C270, foram realizados alguns testes relacionados à configuração de parâmetros das mesmas, além de testes de iluminação, nível de saturação das imagens capturadas por essas câmeras e a aplicação em um sistema de medição dimensional de objetos.

2.3.1 Configuração de parâmetros da câmera Logitech C270

Foi possível realizar o ajuste de parâmetros de aquisição da imagem através da biblioteca OpenCV, porém esses ajustes são limitados as características de projeto das câmeras, algumas dessas funções são:

- CAP_PROP_FPS: Define a taxa de aquisição de imagens da câmera, em *frames* por segundo (fps).
- CAP_PROP_FRAME_WIDTH: Define a largura do frame capturado pela câmera.
- CAP_PROP_FRAME_HEIGHT: Define a altura do frame capturado pela câmera.
- CV_CAP_PROP_BRIGHTNESS: Define o brilho do frame capturado pela

câmera.

- CV_CAP_PROP_CONTRAST: Define o contraste do frame capturado pela câmera.
- CV_CAP_PROP_SATURATION: Define a saturação do frame capturado pela câmera.
- CV_CAP_PROP_HUE: Define o matiz do frame capturado pela câmera.
- CV_CAP_PROP_GAIN: Define o ganho do frame capturado pela câmera.
- CV_CAP_PROP_EXPOSURE: Define a exposição do frame capturado pela câmera.

Através dos parâmetros “CAP_PROP_FRAME_WIDTH” e “CAP_PROP_FRAME_HEIGHT” foi possível alterar a resolução da câmera Logitech C270, que tem como resolução máxima 720p, pois a imagem capturada possui 1280 pixels de largura e 720 pixels de altura.

Na Figura 22, é apresentado a mesma imagem com resolução de 720p e 240p, essa última tem 426 pixels de largura e 240 pixels de altura.

Figura 22: Imagem capturada com resolução de 720p (à esquerda) e 240p



Fonte: O Autor (2021)

Utilizando-se como base a escala de cores HSL (*Hue-Saturation-Lightness*), foi possível alterar os níveis de matiz, saturação e luminosidade da imagem adquirida pela câmera. Conforme apresentado na Figura 23, foram definidos os valores de 99 para “CV_CAP_PROP_HUE”, 200 para “CV_CAP_PROP_SATURATION” e 115 para “CV_CAP_PROP_GAIN”, valores que geram uma cor de tom de verde. O resultado da cor formada por esses valores da escala HSV e a imagem capturada é apresentado

abaixo.

Figura 23: Alteração dos parâmetros de matiz, saturação e luminosidade da câmera



Fonte: O Autor (2021)

Através de um ajuste de um potenciômetro na câmera Logitech C270 foi possível configurar o foco da lente. Para esse ajuste, é necessário retirar a capa frontal da câmera, retirar os parafusos da segunda capa frontal, e pode-se ajustar o foco manualmente, de forma a girar para sentido horário para uma distância focal mais longa e anti-horário para mais perto, conforme apresentado na Figura 24.

Figura 24: Método para alterar o foco da câmera Logitech C270



Fonte: O Autor (2021)

Com o intuito de avaliar o funcionamento do foco da câmera, foi fixado a câmera a uma altura de 23 cm em relação ao plano de medição, inserido uma caixa de papel em seu campo de visão e variado o potenciômetro com o objetivo de analisar o comportamento da imagem, como apresentado nas Figuras 25, 26 e 27.

Figura 25: Longa distância de foco da câmera Logitech C270



Fonte: O Autor (2021)

Figura 26: Curta distância de foco da câmera Logitech C270



Fonte: O Autor (2021)

Figura 27: Distância ajustada de foco da câmera Logitech C270



Fonte: O Autor (2021)

2.3.2 Configuração de Parâmetros da Raspberry Pi Camera v1

Com a câmera Raspberry Pi Camera v1 devidamente instalada e habilitada na Raspberry Pi 3, foi possível configurar alguns parâmetros para a captura de imagens através da ferramenta de linha de comando *raspistill*, alguns desses são:

- `raspistill -o 01.jpg` : comando utilizado para capturar uma imagem com nome “01” e extensão jpg.
- `raspistill -o 01.jpg -vf` : comando utilizado para capturar uma imagem invertida verticalmente
- `raspistill -o 01.jpg -hf` : comando utilizado para capturar uma imagem invertida horizontalmente
- `raspistill -o 01.jpg -w 1920 -h 1080` : comando utilizado para modificar respectivamente a largura e a altura da imagem capturada, e consequentemente alterar a resolução da imagem.
- `Raspistill -o 01.jpg -sh -30`: comando utilizado para definir um nível de acuidade, de -100 a 100, para a captura da imagem.
- `Raspistill -o 01.jpg -co 52`: comando utilizado para definir um nível de contraste, de -100 a 100, para a captura da imagem.
- `raspistill -o 01.jpg -br 25`: comando utilizado para definir um nível de brilho, de

0 a 100, para a captura da imagem.

- `raspistill -o 01.jpg -sa 52`: comando utilizado para definir um nível de saturação, de -100 a 100, para a captura da imagem.
- `raspistill -o 01.jpg -sa 52`: comando utilizado para definir um nível de saturação, de -100 a 100, para a captura da imagem.

Utilizando os comandos “`raspistill -o 01.jpg -vf`” e “`raspistill -o 01.jpg -hf`”, foi possível realizar a inversão vertical e horizontal das imagens capturadas, conforme apresentado na Figura 28.

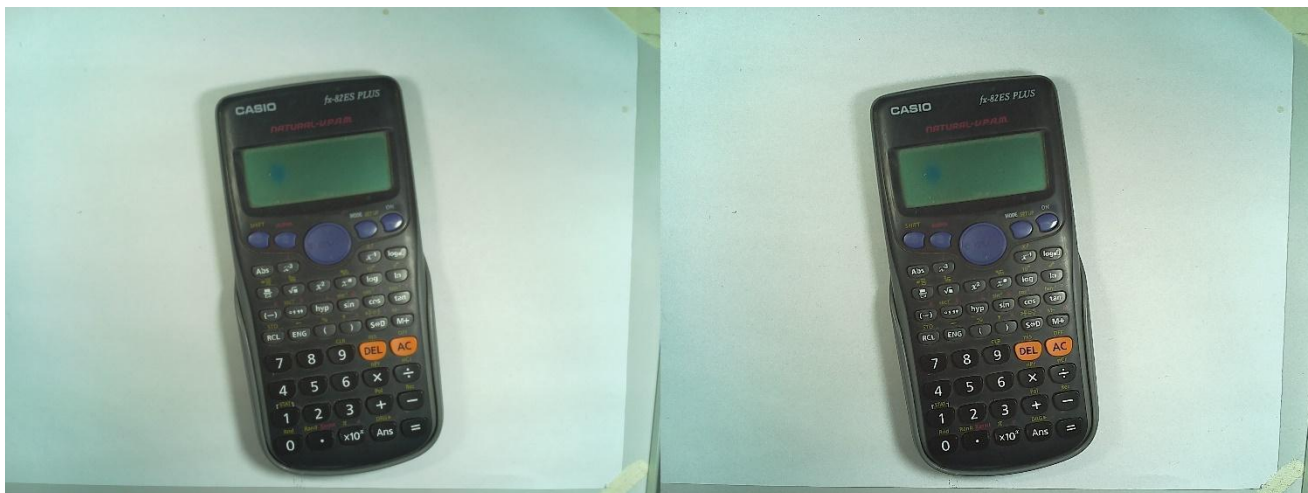
Figura 28: Imagens com inversão vertical (à direita) e horizontal



Fonte: O Autor (2021)

Além disso, foi possível alterar os parâmetros de acuidade, com intuito de modificar a definição de arestas e linhas presentes na imagem, e conseqüentemente alterar a sua nitidez de maneira diretamente proporcional que o valor absoluto do parâmetro, como apresentado na Figura 29, com as diferenças entre as amostras de uma imagem parametrizada em -100 e +97. O comando utilizado para esse teste foi “`raspistill -o 01.jpg -sh`”.

Figura 29: Imagens com acuidade baixa (à esquerda) e alta



Fonte: O Autor (2021)

Também foram realizados testes variando-se os parâmetros de brilho, saturação e contraste da imagem, de forma a ficar claro a importância desses parâmetros para dar ênfase a determinadas cores e formas, conforme apresentado na Figura 30.

Figura 30: Imagem com ajustes na saturação, contraste e brilho, destacando o objeto em análise (à direita)



Fonte: O Autor (2021)

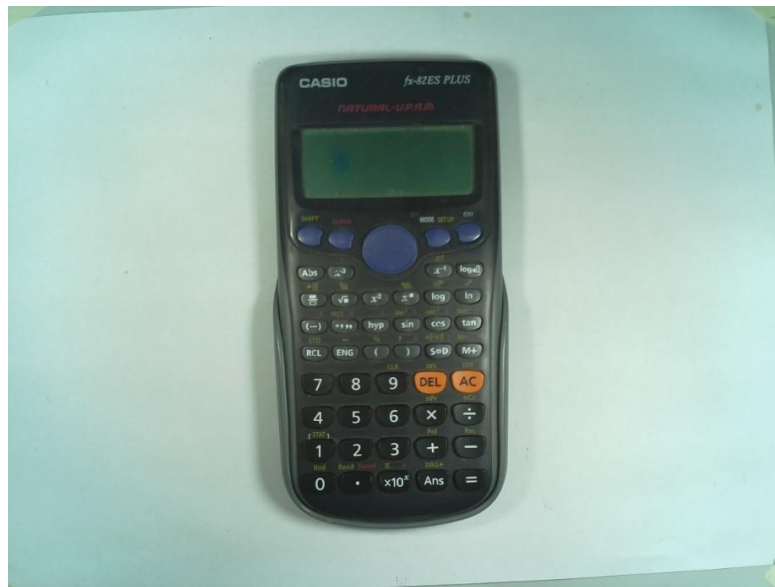
A câmera Raspberry Pi Camera v1 não tem regulagem de foco manual ou automático.

2.4 Nível de saturação das imagens

Com o intuito de analisar o nível de saturação de uma imagem capturada pela câmera Logitech C270 e Raspberry Pi Camera v1, foram realizadas duas capturas de imagem de um mesmo objeto de análise: uma calculadora. Ambas as imagens foram realizadas no mesmo ambiente, com as mesmas condições de iluminação: duas lâmpadas fluorescentes de 15 W cada, apontadas para o plano, a uma distância horizontal de aproximadamente 15 cm do objeto de análise e rotacionadas em 20° em relação a normal do plano.

A Figura 31 representa a imagem capturada pela Raspberry Pi Camera v1 foi posicionada a uma altura de 24,5 cm em relação ao plano superior do objeto em análise.

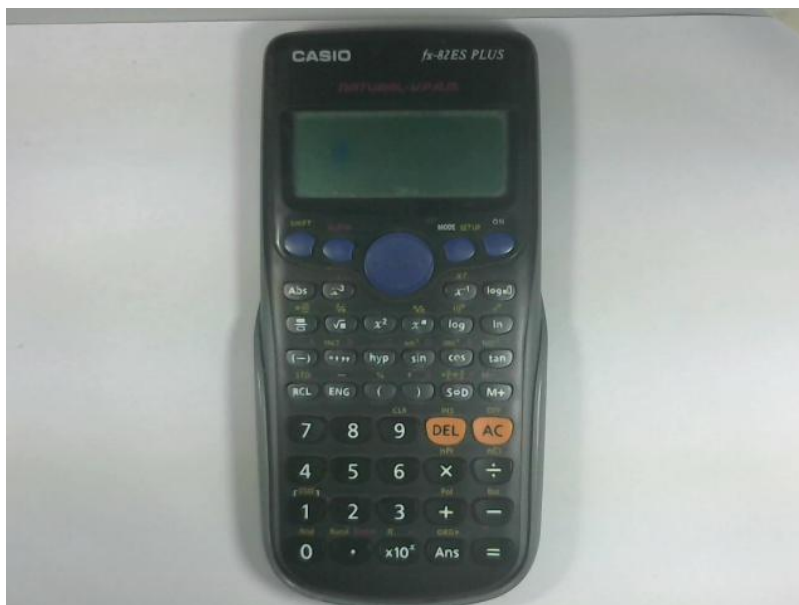
Figura 31: Captura do objeto pela Raspberry Pi Camera v1



Fonte: O Autor (2021)

Já a Figura 32 apresenta a imagem capturada pela Logitech C270 foi posicionada a uma altura de 28 cm em relação ao plano superior do objeto em análise, por conta de ser a altura mínima capaz de capturar por completo na imagem o objeto de análise. Para garantir a melhor qualidade da imagem, o foco foi ajustado manualmente para essa captura.

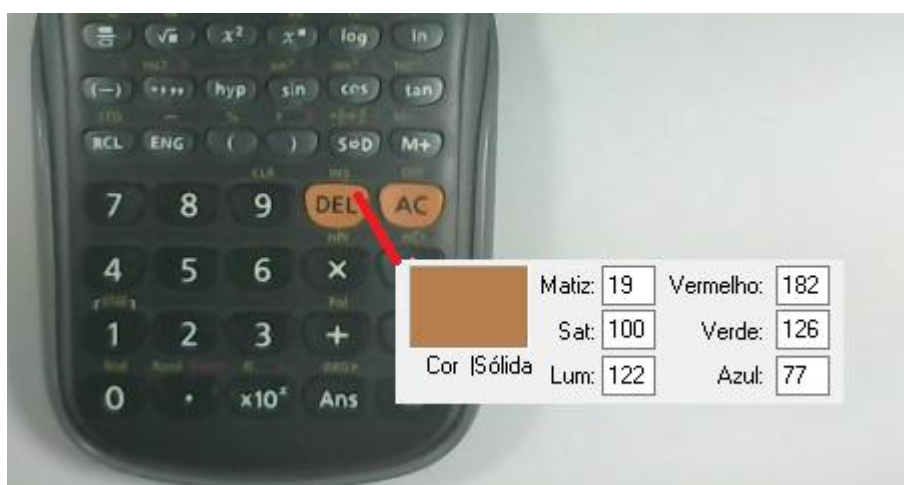
Figura 32: Captura do objeto pela Logitech C270



Fonte: O Autor (2021)

Conforme a Figura 33, pode-se analisar que a imagem capturada pela Raspberry Pi Camera v1 tem um nível de saturação mais alto, evidenciado pelo destaque das cores laranja e azul dos botões da calculadora. A cor laranja da imagem capturada pela Logitech C270 tem escala de matiz igual à 19, saturação de 100 e luminosidade de 122 no pixel localizado no par ordenado 359 e 332 do eixo cartesiano.

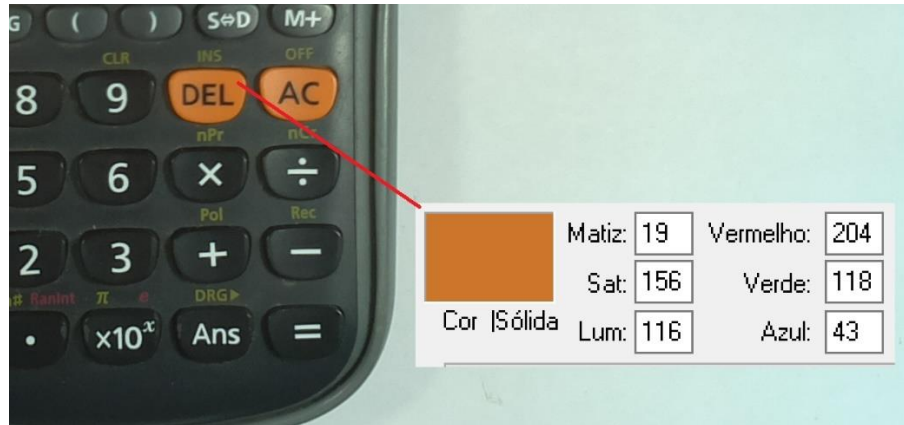
Figura 33: Matiz, saturação e luminosidade do laranja capturado pela Logitech C270



Fonte: O Autor (2021)

Já a imagem capturada pela Raspberry Pi Camera v1 tem escala de matiz igual à 19, saturação de 156 e luminosidade de 116 no pixel localizado no par ordenado 1441 e 1237 do eixo cartesiano, conforme apresentado na Figura 34.

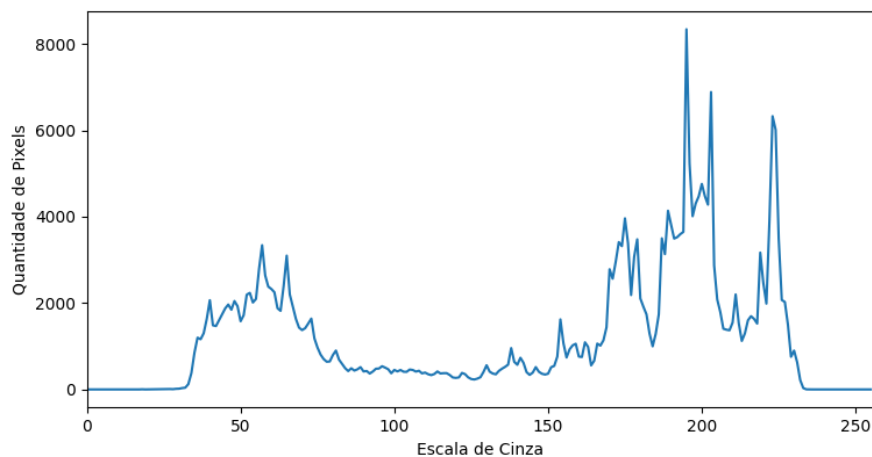
Figura 34: Matiz, saturação e luminosidade do laranja capturado pela Raspberry Pi Camera v1



Fonte: O Autor (2021)

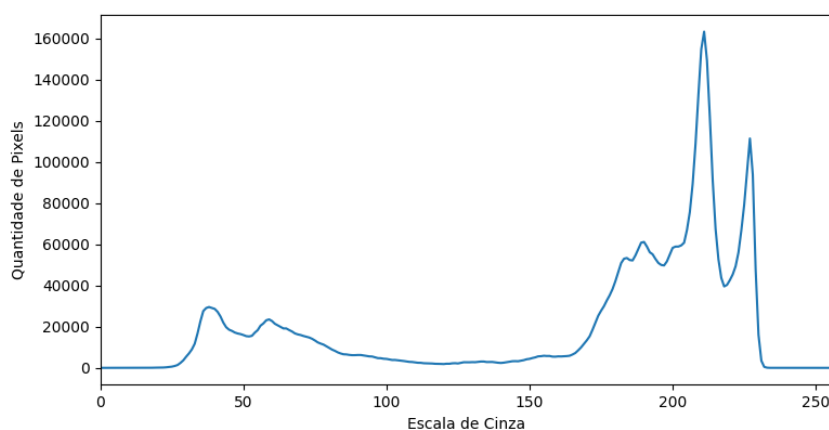
Outra característica observada através do histograma das imagens em escala monocromática é o alto contraste de ambas as imagens, visto que existe uma distribuição das intensidades em toda a escala de intensidades (GONZALEZ; WOODS, 2009). Além disso, ambas as imagens possuem predominância dos componentes no lado superior da escala de nível cinza, de forma a evidenciar que são imagens claras, conforme apresentado nas Figuras 35 e 36 (GONZALEZ; WOODS, 2009).

Figura 35: Histograma da imagem capturada pela Logitech C270



Fonte: O Autor (2021)

Figura 36: Histograma da imagem capturada pela Raspberry Pi Camera v1



Fonte: O Autor (2021)

2.5 Geometria do sistema de iluminação

Com o intuito de avaliar a importância do sistema de iluminação em um sistema de visão embarcado, foram realizados testes com algumas geometrias de sistemas de iluminação e capturado algumas imagens com as câmeras em análise.

2.5.1 Geometria 45°: 0° - iluminação direcional frontal difusa

Leta et. al. (2016) afirma que a iluminação direcional frontal difusa é realizada com apenas uma fonte de luz, posicionada a um ângulo em relação ao objeto. Para analisar a captura realizada pelas câmeras Logitech C270 e Raspberry Pi Camera v1, foi capturado uma imagem com cada câmera, utilizando-se como fonte de luz uma lâmpada fluorescente de 15W, posicionada a um ângulo de aproximadamente 45° em relação ao sensor da câmera. O resultado dessas capturas é apresentado na Figura 37.

Figura 37: Iluminação direcional frontal difusa capturada por Logitech C270 (à direita) e Raspberry Pi Camera v1



Fonte: O Autor (2021)

Através das imagens apresentadas, constata-se que essa iluminação permite um destaque aos relevos presentes nos números do relógio. Ambas as câmeras foram capazes de captar a diferença de contraste entre o lado esquerdo e direito da imagem, de maneira a destacar essa transição entre luz e sombra. Também foi possível observar que a Raspberry Pi Camera v1 capturou o reflexo da proteção de plástico superior do relógio.

2.5.2 Geometria 45°: 0° - iluminação frontal difusa

Segundo Leta et. al. (2016), a iluminação frontal difusa visa reduzir reflexos e sombras sobre o objeto, de maneira que duas fontes de luz são posicionadas em ângulos opostos ao redor da câmera.

Para analisar a captura realizada pelas câmeras Logitech C270 e Raspberry Pi Camera v1, foi capturado uma imagem com cada câmera, utilizando-se como fonte de luz duas lâmpadas fluorescentes de 15W, posicionadas de maneira oposta e a um ângulo de aproximadamente 45° em relação ao sensor da câmera, o resultado é apresentado na Figura 38.

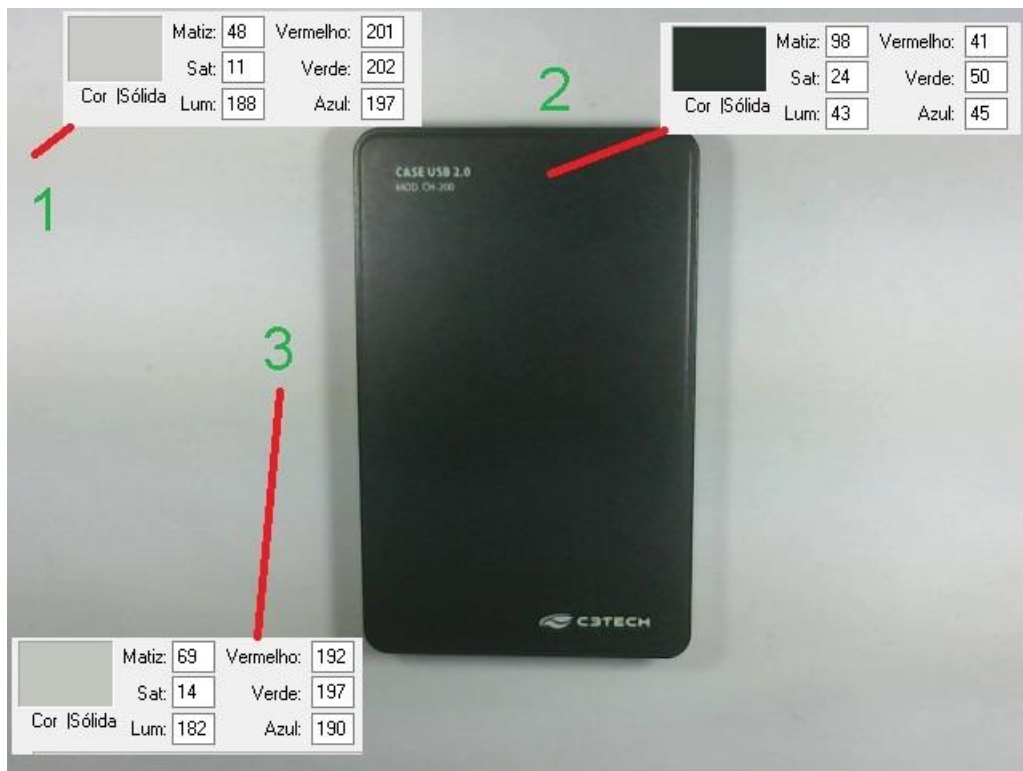
Figura 38: Iluminação frontal difusa capturada por Logitech C270 (à direita) e Raspberry Pi Camera v1



Fonte: O Autor (2021)

Foi possível constatar a diferença de saturação entre as imagens, na imagem capturada pela Logitech C270, nos pontos 1, 2, e 3, indicados na Figura 39, tiveram valores de saturação igual a 11, 24, 14 respectivamente.

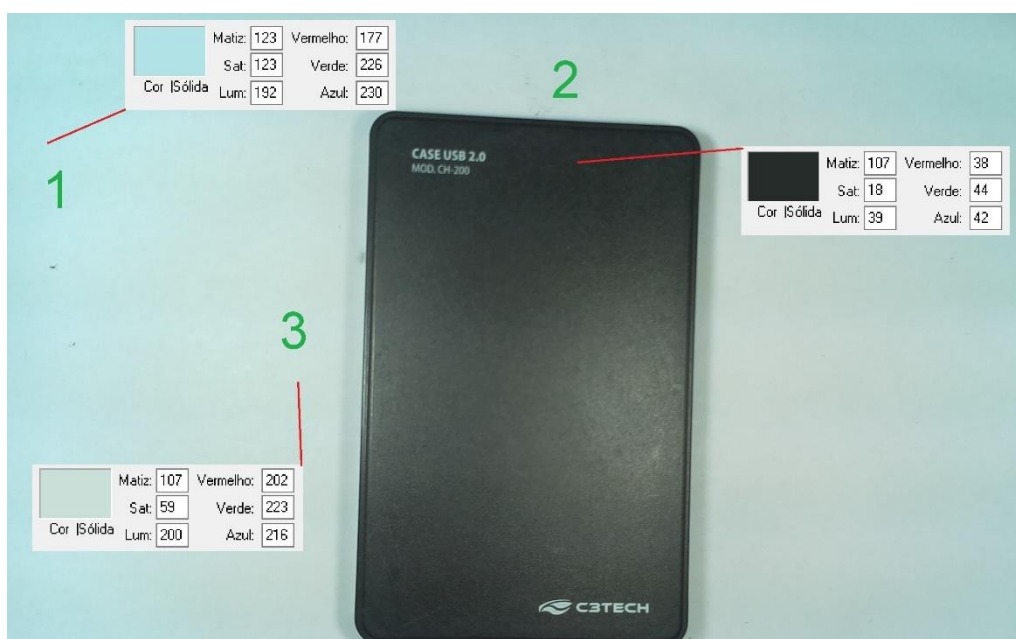
Figura 39: Níveis de saturação para imagem capturada pela Logitech C270



Fonte: O Autor (2021)

Já para a imagem capturada pela Raspberry Pi Camera v1, nos pontos 1, 2, e 3, indicados na Figura 40, tiveram valores de saturação igual a 123, 18, 59 respectivamente.

Figura 40: Níveis de saturação para imagem capturada pela Raspberry Pi Camera v1



Fonte: O Autor (2021)

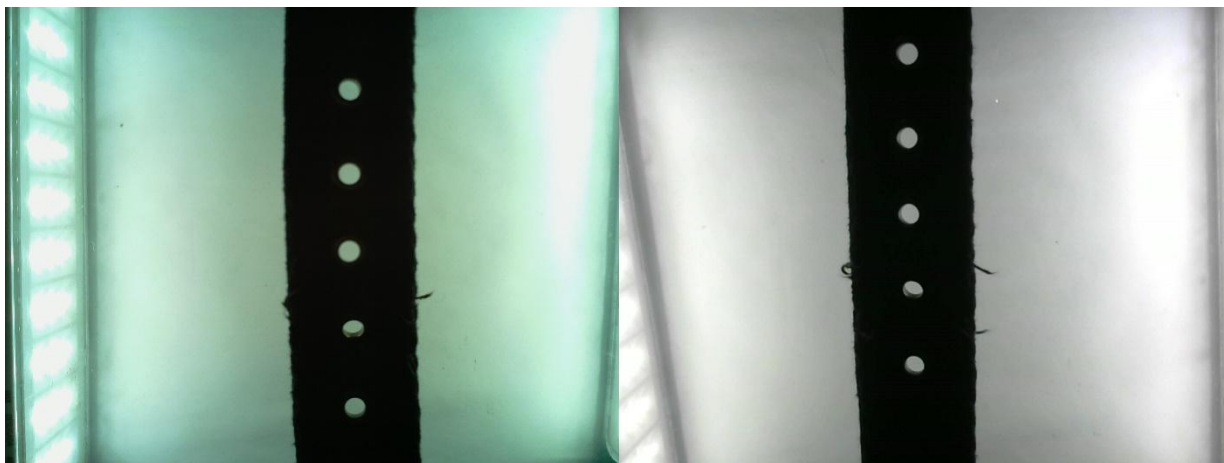
Dessa forma, pode-se notar que houve uma saturação maior na imagem capturada pela Raspberry Pi Camera v1 na base de tom claro, e uma saturação maior no tom escuro do disco rígido (HD) externo capturado pela Logitech C270.

Além disso, a Raspberry Pi Camera v1 capturou o reflexo da iluminação na parte superior do disco rígido (HD) externo. Assim como a Logitech C270 destacou as sombras nas laterais do objeto de análise.

2.5.3 Iluminação diascópica

Leta et. al. (2016) defende que a iluminação diascópica é utilizada para casos no qual se busca o destaque do contorno do objeto, de forma que a fonte de luz é posicionada atrás do objeto de análise. Para analisar a captura realizada pelas câmeras Logitech C270 e Raspberry Pi Camera v1, foi capturada uma imagem com cada câmera, utilizando-se como fonte de luz duas luminárias de 2 W, direcionadas ortogonalmente ao objeto de análise e paralelamente uma à outra.

Figura 41: Iluminação diascópica capturada por Logitech C270 (à direita) e Raspberry Pi Camera v1



Fonte: O Autor (2021)

Com base nas imagens da Figura 41, foi possível observar a capacidade que a iluminação diascópica tem de dar destaque ao contorno do objeto em análise. Ambas as câmeras puderam capturar detalhes do cinto, como os furos e pequenos fios soltos no decorrer da borda do cinto.

2.5.4 Iluminação frontal difusa – lâmpadas halógenas

Com o intuito de avaliar as características das câmeras Raspberry Pi Camera v1 e Logitech C270 com diferentes fontes de iluminação, posicionados com a geometria 45°: 0° e iluminação frontal difusa, foi realizado a captura de imagens com duas lâmpadas halógenas de 42W e temperatura de cor de 2700 K, o resultado é apresentado na Figura 42.

Figura 42: Lâmpadas halógenas como fonte de luz capturada por Logitech C270 (à direita) e Raspberry Pi Camera v1



Fonte: O Autor (2021)

Foi capaz de perceber que ambas as imagens ficaram mais escuras se comparadas às imagens com iluminação frontal difusa feitas com lâmpadas fluorescentes já apresentadas. Isso ocorreu por conta da temperatura de cor das

lâmpadas halógenas utilizadas serem mais baixas do que das lâmpadas fluorescentes, fato que fez com que as imagens ganhassem um tom amarelado.

Outro fato notado foi a dificuldade da câmera Raspberry Pi Camera v1 de capturar homogeneamente o fundo branco atrás do objeto em análise, de tal maneira que é destacado dois tons na imagem, um ao redor do objeto e outro mais afastado do mesmo.

A câmera Logitech C270 foi capaz de captar a imagem com nitidez e homogeneidade nas cores.

2.5.5 Iluminação frontal difusa – lâmpadas LED

Com a intenção de avaliar as câmeras Raspberry Pi Camera v1 e Logitech C270 em um sistema de iluminação com lâmpadas de LED, foi realizado a captura de imagens com tal sistema posicionado em geometria 45°: 0°, iluminação frontal difusa, com fontes de iluminação de duas lâmpadas LED de 6W e temperatura de cor de 6500 K, o resultado é apresentado na Figura 43.

Figura 43: Lâmpadas LED como fonte de luz capturada por Logitech C270 (à direita) e Raspberry Pi Camera v1



Fonte: O Autor (2021)

Através das imagens capturadas por ambas as câmeras, foi possível observar a presença de um ruído na imagem, caracterizado por listras de baixo contraste e de tom escuro, foi possível notar que a Logitech C270 foi mais sensível a esse ruído, já

a Raspberry Pi Camera v1 pode suprimir esse ruído com mais intensidade.

Esse ruído advém da interferência elétrica causada no momento da aquisição, é o chamado ruído periódico e pode ser reduzido através de técnicas de filtragem de imagem no domínio da frequência (GONZALEZ; WOODS, 2009).

2.5.6 Iluminação frontal difusa – variação da intensidade luminosa

Com o intuito de notar as características das imagens capturadas pelas câmeras Raspberry Pi Camera v1 e Logitech C270 com diferentes intensidades do sistema de iluminação, foram realizados dois testes: o primeiro com um sistema de iluminação na geometria 45° : 0° , iluminação frontal difusa, com fontes de iluminação de duas lâmpadas fluorescentes de 15W e temperatura de cor de 6400 K, e o segundo sistema de iluminação com mesma geométrica de iluminação, porém com lâmpadas fluorescentes de 25W e temperatura de cor de 6400 K, o resultado é apresentado na Figura 44.

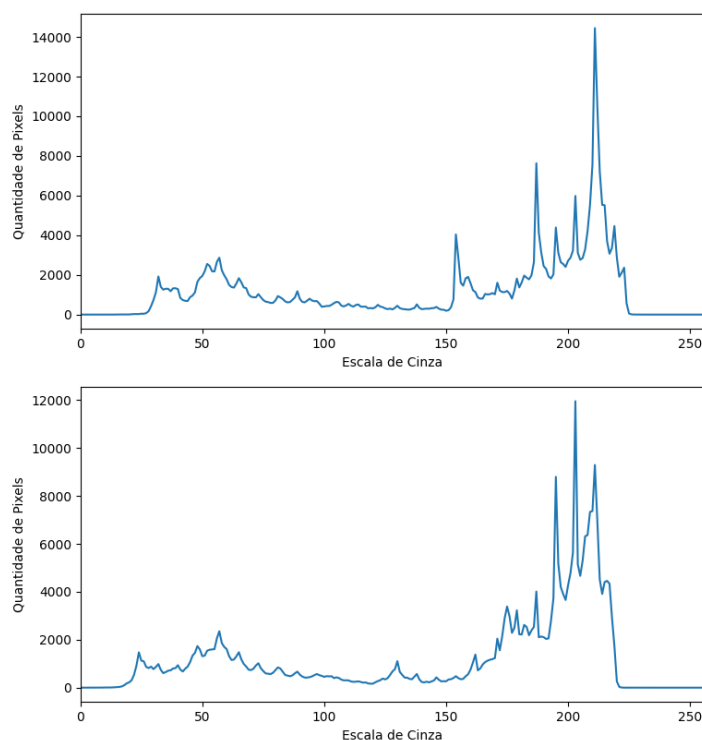
Figura 44: Imagens com intensidade luminosa menor (à direita) e maior capturada pela Logitech C270



Fonte: O Autor (2021)

Com o aumento da intensidade luminosa do sistema de iluminação, foi possível constatar que na imagem capturada pela câmera Logitech C270 manteve o seu alto contraste, porém a imagem ficou mais clara, como demonstrado nos histogramas das imagens apresentados na Figura 45.

Figura 45: Histograma das imagens com intensidade luminosa menor (acima) e maior capturada pela Logitech C270



Fonte: O Autor (2021)

Além disso, foi possível notar uma menor presença de sombras na imagem com o sistema de iluminação de maior intensidade, destacando o contorno do objeto em análise.

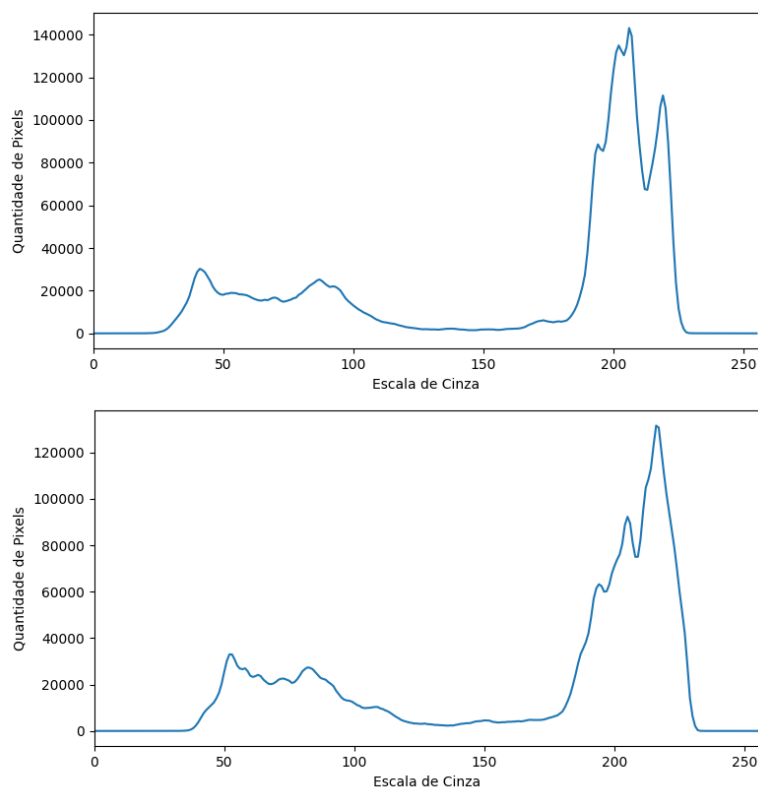
Através das imagens capturadas pela câmera Raspberry Pi Camera v1 e de seus respectivos histogramas, apresentados nas Figuras 46 e 47 respectivamente, foi possível notar que a imagem ficou mais clara, pois houve um aumento de pixels no lado superior da escala, além de uma maior homogeneidade no tom claro presente no fundo da mesma. Ademais, foi constatado a maior presença de sombra no entorno do objeto em análise, como é apresentado no histograma, com o aumento de pixels entre a região de intensidade 50 e 100 na escala de nível cinza.

Figura 46: Imagens com intensidade luminosa menor (à direita) e maior capturada pela Raspberry Pi Camera v1



Fonte: O Autor (2021)

Figura 47: Histograma das imagens com intensidade luminosa menor (acima) e maior capturada pela Raspberry Pi Camera v1



Fonte: O Autor (2021)

2.6 Algoritmo de Medição Bidimensional

Com o intuito de avaliar o funcionamento das câmeras Raspberry Pi Camera v1 e Logitech C270, além da placa Raspberry Pi 3 Model B, foi desenvolvido um algoritmo em linguagem de programação Python capaz de realizar a medição do comprimento e largura de objetos, independente do ângulo de rotação do mesmo em relação ao seu eixo.

2.6.1 Aquisição dos pontos extremos do objeto

A primeira etapa desse algoritmo foi realizar a aquisição da imagem e obter os pixels referentes ao contorno do objeto, para isso foi realizado a conversão dessa imagem adquirida pelo sistema de visão embarcado para a escala de nível cinza, e então foi aplicado a essa imagem um filtro gaussiano, a fim de reduzir os ruídos periódicos da imagem, conforme apresentado na Figura 48.

Figura 48: Imagem adquirida em escala de nível cinza (à esquerda) e imagem com filtro gaussiano aplicado



Fonte: O Autor (2022)

Com a imagem em nível cinza e com os ruídos reduzidos foi possível realizar a detecção dos contornos da imagem, para isso foi utilizado o operador Canny, com níveis mínimo e máximo de 100 e 255 respectivamente, o resultado é apresentado na Figura 49.

Figura 49: Aplicação do operador Canny na imagem



Fonte: O Autor (2022)

A imagem resultante do operador Canny foi utilizada no algoritmo *findContours* da biblioteca OpenCV, esse tem dois parâmetros principais: modo de recuperação de contorno e método de aproximação de contorno (OPENCV, 2021). O Modo de recuperação de contorno escolhido para esse algoritmo foi o RETR_TREE, a fim de recuperar todos os contornos e reconstruir uma hierarquia completa dos contornos agrupados (OPENCV, 2021). Tal método de reconstrução utilizado foi o

CHAIN_APPROX_SIMPLE no qual realiza a compressão horizontal, vertical e diagonal de todos os segmentos encontrados e deixam apenas os seus pontos extremos (OPENCV, 2021). O resultado da aplicação desse algoritmo é apresentado na Figura 50.

Figura 50: Aplicação do algoritmo *findContours* na imagem

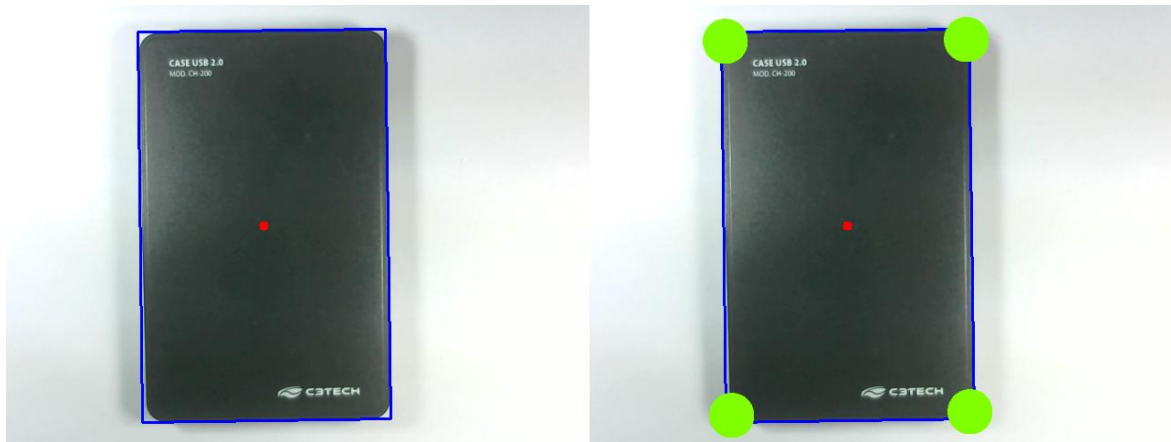


Fonte: O Autor (2022)

Por fim, foi verificada a área do polígono formado por esse contorno, a fim de evitar a detecção de pequenos objetos ao entorno que são ruídos na imagem, visto que não são objetos de interesse no algoritmo. Isso foi possível através da função *contourArea* da biblioteca OpenCV.

Então o polígono com contorno bem definido foi cercado de um retângulo, estritamente posicionado ao entorno das bordas do objeto em análise, e a este aplicado a função *boxPoints* da biblioteca OpenCV, com o objetivo de obter os quatro pontos extremos desse retângulo, como apresentado na Figura 51.

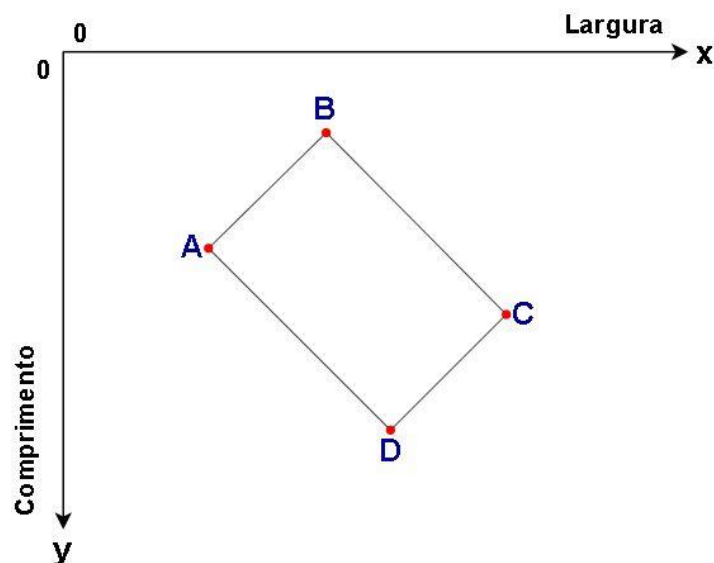
Figura 51: Imagem com retângulo ao entorno do objeto (à esquerda) e imagem com algoritmo

boxPoints aplicado

Fonte: O Autor (2022)

Uma vez que o objetivo do algoritmo foi mensurar a largura e o comprimento do objeto em análise, os dados de contorno adquirido da imagem já segmentada, foram aplicados em uma função capaz de obter 4 pontos extremos do objeto no eixo x e y da imagem. Esses pontos foram armazenados em um vetor seguindo a seguinte ordem: o ponto mais próximo do 0 no eixo X foi assignado como ponto A, já o ponto com maior valor no eixo X como ponto C, o ponto B foi aquele localizado mais próximo do 0 no eixo Y e o ponto D foi aquele localizado mais próximo ao extremo do eixo Y , conforme apresentado na Figura 52.

Figura 52: Organização dos pontos extremos do objeto



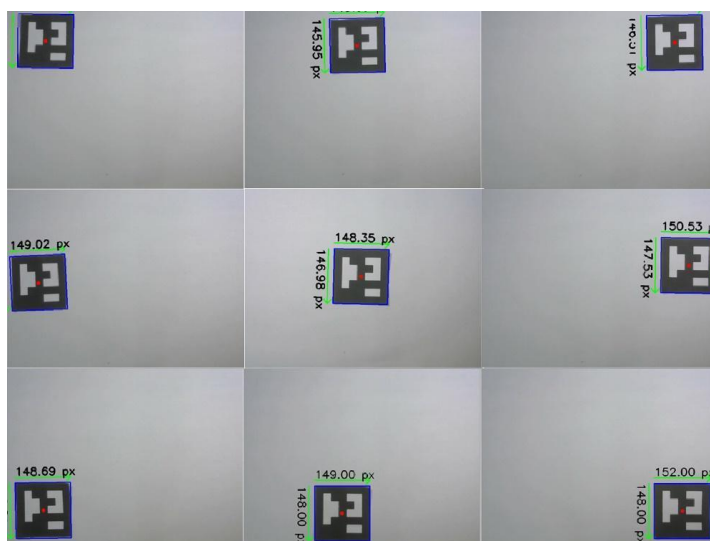
Fonte: O Autor (2021)

2.6.2 Calibração do pixel a partir de divisão de quadrantes

Como já citado, a calibração do pixel é um método de medição dimensional por imagem que tem como objetivo determinar a dimensão em unidade de comprimento de um pixel (LETA et al., 2016). Para isso foi desenvolvido um algoritmo para realizar a medição de um objeto retangular, de contorno bem definido. A partir da dimensão em pixels e conhecidas as dimensões em unidade de comprimento, foi possível encontrar a razão px/mm no eixo X e Y.

Porém, a partir de testes, foi observado que havia diferenças consideráveis nas dimensões do mesmo objeto em cantos opostos da imagem. Com o propósito de mensurar essa diferença, foram feitas aquisições de comprimento e largura em pixel do mesmo objeto em nove posições diferentes da imagem, conforme apresentado na Figura 53. Para isso foi utilizado a câmera Logitech C270, posicionada a uma altura de 30 cm do plano.

Figura 53: Aquisição de comprimento e largura em posições diferentes



Fonte: O Autor (2021)

O resultado das medições é apresentado na Tabela 1:

Tabela 1: Resultados das medições em pixel

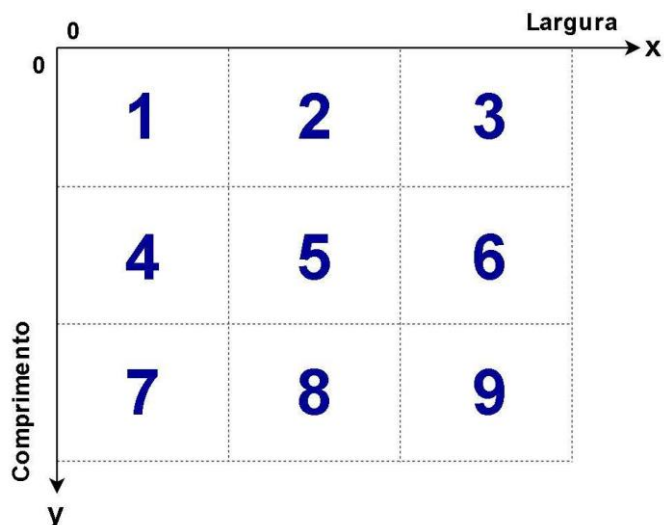
Posição	Distância no Eixo X	Distância no Eixo Y
1	148,41	144,26
2	148,09	145,95
3	149,26	146,50
4	149,02	146,16
5	148,35	146,98
6	150,53	147,53
7	148,68	148,81
8	149,00	148,00
9	152,00	148,00

Fonte: O Autor

A partir dos valores adquiridos, foi possível notar uma diferença no eixo X e Y de aproximadamente 4 pixels entre as posições 1 e 9, além de uma variação entre as posições.

Dessa forma, com a finalidade de diminuir essa variação, foi realizado um algoritmo para obter nove razões de transformação de pixel para milímetro, de forma que a imagem capturada foi dividida em nove quadrantes, conforme apresentado na Figura 54.

Figura 54: Quadrantes de calibração do pixel



Fonte: O Autor (2021)

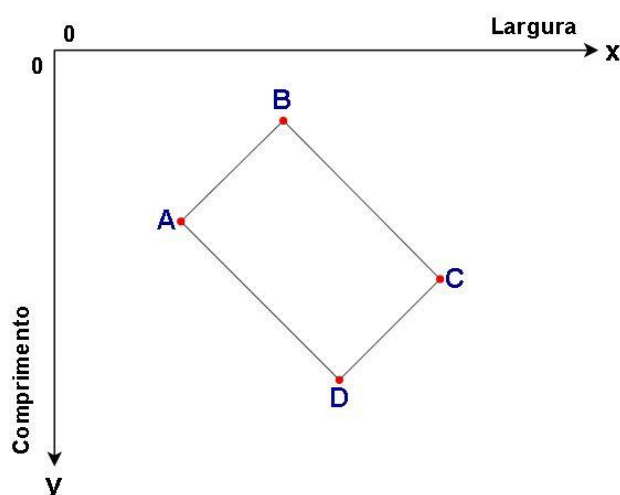
E assim, o usuário do sistema de medição posiciona um objeto retangular nas nove posições e realiza aquisições de imagens. Esse objeto deve ter contorno bem definido e dimensões menores que os respectivos quadrantes. Por fim, o usuário insere as dimensões em milímetros do objeto obtidas através de um instrumento de medição. Esses valores são utilizados de forma que se obtém nove razões de transformação de pixel para milímetro, uma para cada região delimitada na imagem acima.

2.6.3 Separação dos segmentos de reta

Haja vista que os pontos foram organizados em um vetor de forma ordenada, foi possível aplicar o método de separação dos segmentos de reta a fim de realizar a conversão da dimensão em pixel para milímetro, com as nove razões de transformação de pixel para milímetro.

Dado os pontos ordenados apresentados na Figura 55, que representam os pontos obtidos do algoritmo de aquisição dos pontos extremos do objeto explicado anteriormente, a primeira etapa do algoritmo de separação dos segmentos de reta foi o cálculo da equação linear do segmento de reta \overline{AB} . Para tal, a função verificou qual o ponto inicial e o ponto final do segmento de reta medido, a partir desses foram obtidos os coeficientes da equação linear.

Figura 55: Pontos adquiridos do algoritmo de aquisição dos pontos extremos do objeto

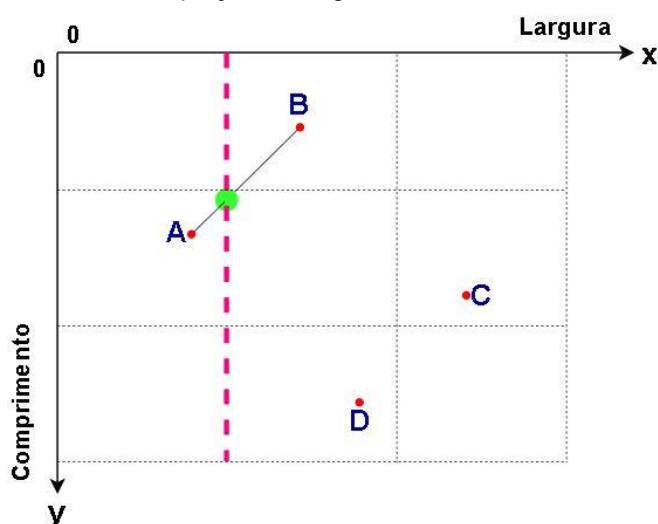


Fonte: O Autor (2021)

Uma vez que a equação da reta foi encontrada, realizou-se uma varredura nos eixos ordenados a fim de buscar pontos que satisfazem a equação, e dessa forma foi possível dividir o segmento de reta em mais segmentos nos seus respectivos quadrantes.

A varredura foi iniciada no eixo X, no qual cada par ordenado é analisado, de forma que foi atribuído um valor fixo de 33,33% da dimensão do eixo X da imagem, e o valor de Y foi variado de 0 à dimensão do eixo Y da imagem, como representado na Figura 56. Caso algum desses pontos ordenados resultantes dessa varredura satisfizesse a equação da reta já calculada, esse ponto ordenado era salvo em um *array* de pontos encontrados.

Figura 56: Varredura do eixo X com variação do eixo Y a fim de obter pontos que satisfazem a equação do segmento de reta



Fonte: O Autor (2021)

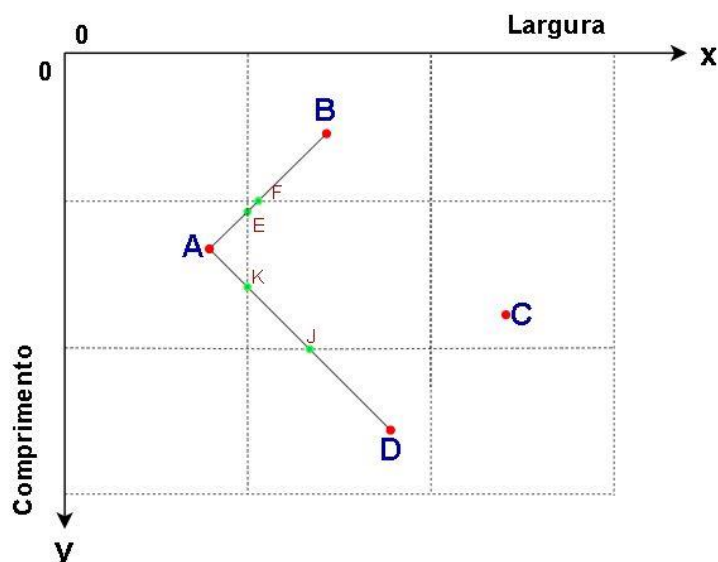
O mesmo processo de varredura ocorreu no eixo X, com seu valor atribuído em 66,66% da dimensão do eixo X da imagem. Assim como no eixo Y, no qual foram realizadas varreduras nos pontos ordenados com eixo Y atribuídos em valores iguais a 33,33% e 66,66% da dimensão do eixo Y da imagem.

O próximo segmento de reta analisado é \overline{AD} , no qual é obtido os coeficientes de equação linear e realizado a varredura nos eixos ordenados a fim de encontrar pontos que satisfaçam a equação desse segmento de reta, da mesma forma que foi realizada para \overline{AB} .

Dessa forma, foi possível encontrar todos os pares ordenados dispostos sobre

os limitantes dos quadrantes de calibração, apresentados na Figura 57. E por fim dividir os segmentos de reta que representam o contorno do objeto em vários segmentos de retas presentes dentro dos seus respectivos quadrantes.

Figura 57: Segmentos de reta localizados nos respectivos quadrantes de calibração



Fonte: O Autor (2021)

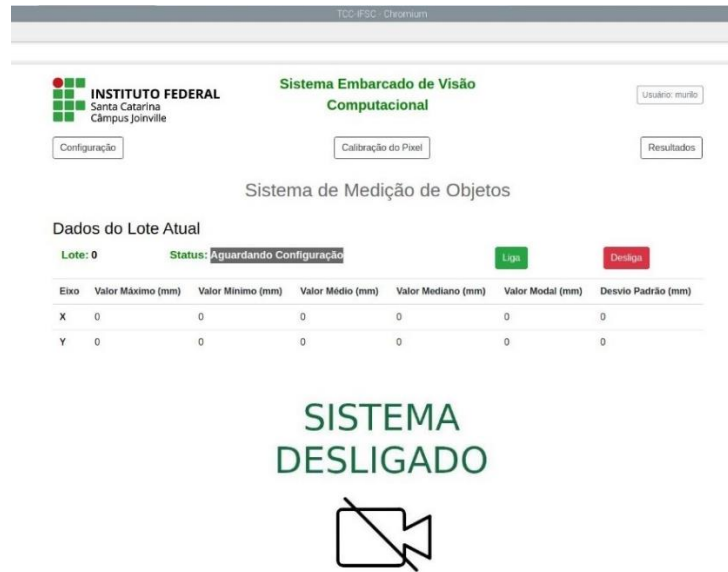
A partir desses segmentos de retas organizados, e com base em conceitos de trigonometria, foi possível encontrar a dimensão em pixels dos segmentos de reta. E baseado nas razões de proporção (px/mm) de cada quadrante, obtida através da função de calibração do pixel, foi possível encontrar a dimensão dos segmentos de reta em milímetro. Através da soma desses segmentos de reta, teve-se como resultado o comprimento e largura do objeto em análise.

2.7 Interface de operação do usuário

Para tornar fácil e intuitivo a operação desse sistema de visão computacional embarcado aplicado a um sistema de manufatura, foi desenvolvido uma interface de usuário através do framework web Django.

Nessa interface, o usuário acessa o endereço <https://127.0.0.1:8000> através de um navegador web localizado no dispositivo Raspberry Pi na qual o sistema está funcionando, a página inicial dessa interface é apresentada na Figura 58.

Figura 58: Menu inicial da interface



Fonte: O Autor (2021)

Antes de iniciar as medições, o usuário deve acessar o menu de “Calibração do Pixel”, representado na Figura 59, com o objetivo de gerar as razões de proporção (px/mm) de cada quadrante da imagem, como já comentado. Para isso, ele deverá nomear a calibração e determinar qual a câmera utilizada, e assim poderá iniciar a calibração, de forma a posicionar o objeto a ser medido dentro do quadrante indicado.

Figura 59: Interface de aquisição de imagens para calibração do pixel



Fonte: O Autor (2021)

Assim que as imagens foram adquiridas nos nove quadrantes, o usuário deve inserir o valor verdadeiro convencional do mensurando, através dos campos de altura e largura, em milímetros, como apresentado na Figura 60.

Figura 60: Interface de aquisição do valor verdadeiro convencional do mensurando

Fonte: O Autor (2021)

E por fim o resultado das razões de proporção (px/mm) é apresentado na tela, conforme a Figura 61, e o usuário deve retornar ao menu inicial.

Figura 61: Interface dos resultados da calibração do pixel

Quadrante	Escala Eixo X (px/mm)	Escala Eixo Y (px/mm)
1	0,32894448323302194	0,33400317506540184
2	0,32892097429864714	0,3340477133464758
3	0,3282681761777388	0,3321512706909437
4	0,3299523752320666	0,3333052669851501
5	0,32894736842105265	0,3333333333333333
6	0,32894736842105265	0,3333333333333333
7	0,3333333333333333	0,33557046979865773
8	0,32502371277333547	0,3346762502878272
9	0,32550702274797083	0,33346503822142226

Fonte: O Autor (2021)

Uma vez que a calibração do pixel foi realizada, o usuário deve configurar alguns parâmetros de controle da aplicação, como número de lote, valores máximos e mínimos de largura e altura permitidos, além de escolher a câmera a ser utilizada, e a calibração de pixel correspondente, essa página da interface é apresentada na

Figura 62.

Figura 62: Interface de configuração de parâmetros da aplicação

Fonte: O Autor (2021)

Por fim, o usuário deve retornar ao menu inicial e estará apto para iniciar o sistema através do botão de ligar, que irá habilitar a medição, além de apresentar o lote atual, e o status atual do sistema, como apresentado na Figura 63.

Figura 63: Aplicação em funcionamento

Eixo	Valor Máximo (mm)	Valor Mínimo (mm)	Valor Médio (mm)	Valor Mediano (mm)	Valor Modal (mm)	Desvio Padrão (mm)
X	0	0	0	0	0	0
Y	0	0	0	0	0	0

Fonte: O Autor (2021)

2.8 Aquisição de dimensões com o sistema embarcado de visão computacional

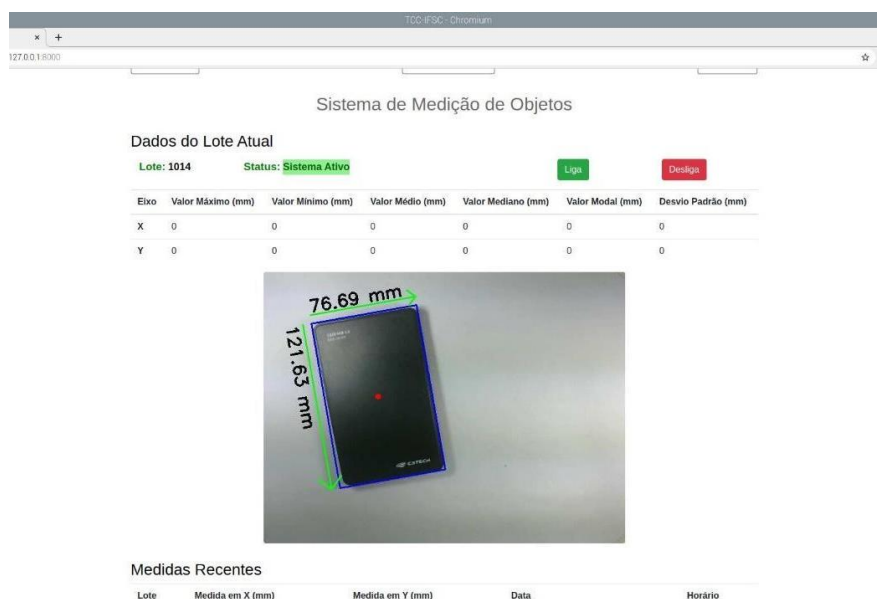
Com a intenção de avaliar as características das câmeras Raspberry Pi Camera v1 e Logitech C270, além da placa Raspberry Pi 3 Model B, aplicados a um sistema de visão computacional embarcado em sistemas de manufatura, foram realizados alguns testes que consistiam em adquirir dimensões de comprimento e largura em milímetros de objetos em variadas condições.

2.8.1 Câmera Logitech C270 com altura de medição de 345 mm

Foram realizadas aquisições de dimensões utilizando-se a câmera Logitech C270, essa estava posicionada com seu respectivo sensor a 345 mm do plano e auxiliada por um sistema de iluminação frontal difusa, com fontes de iluminação de duas lâmpadas fluorescentes de 15W e temperatura de cor de 6400 K. Para diminuir o erro da transformação de pixel para milímetro, foi realizado o processo de calibração do pixel para essa situação.

Na etapa de calibração do pixel, se pode realizar a detecção do objeto em análise e obter as relações de transformação de pixel para milímetro. E com o sistema de visão embarcada em modo de operação normal, foram realizadas medições das dimensões de um disco rígido (HD) externo, com valor verdadeiro convencional de 75 mm de largura e 120 mm de comprimento, em diversas posições dentro do campo de alcance do sistema, além de diversos ângulos de rotação do objeto em torno do próprio eixo apresentado em vermelho na Figura 64.

Figura 64: Sistema de visão embarcada com câmera Logitech C270 a 345 mm do plano



Fonte: O Autor (2021)

Os resultados das medições realizadas estão apresentados nas Tabelas 2 e 3.

Tabela 2: Medições da largura do objeto com câmera Logitech C270 a 345 mm do plano

Medição	Largura (mm)	Erro Absoluto(mm)	Erro Relativo (%)
1	76,77	1,77	2,36
2	76,84	1,84	2,45
3	78,85	3,85	5,13
4	77,89	2,89	3,85
5	77,14	2,14	2,85
6	76,69	1,69	2,25

Fonte: O Autor (2021)

Tabela 3: Medições do comprimento do objeto com câmera Logitech C270 a 345 mm do plano

Medição	Comprimento (mm)	Erro Absoluto (mm)	Erro Relativo (%)
1	122,6	2,6	2,17%
2	124,47	4,47	3,73%
3	122,29	2,29	1,91%
4	121,5	1,5	1,25%
5	123,97	3,97	3,31%
6	121,63	1,63	1,36%

Fonte: O Autor (2021)

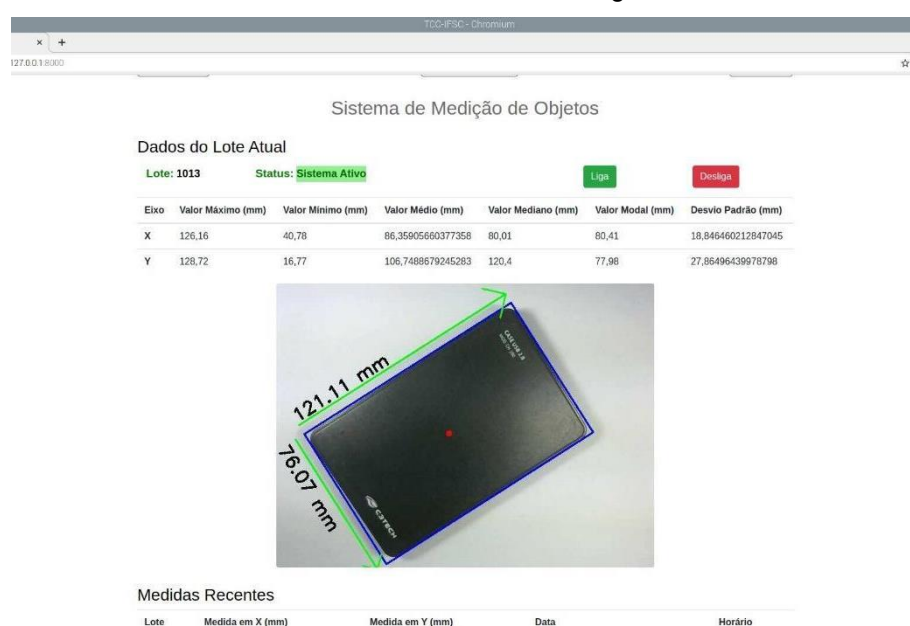
A partir dos resultados apresentados, pode-se verificar que os as taxas de erro de medição foram de no máximo 5,13% e alguns dos erros foram causados por sombra nos entornos do objeto.

2.8.2 Câmera Logitech C270 com altura de medição de 250 mm

As mesmas condições do sistema de iluminação do teste anterior foram mantidas, alterando-se a altura do sensor da câmera em relação ao plano para 250 mm, para manter o bom funcionamento do sistema, foi realizado o ajuste do foco da câmera para essa distância de foco. Já o objeto de medição foi o mesmo utilizado em teste anterior, além de ser realizado o processo de calibração do pixel para essa situação.

Na etapa de calibração do pixel, se pode realizar a detecção do objeto em análise e obter as relações de transformação de pixel para milímetro. E novamente foram adquiridas imagens em diversas posições dentro do campo de alcance do sistema, além de diversos ângulos de rotação do objeto em torno do próprio eixo. A aplicação em funcionamento para essas condições é apresentada na Figura 65.

Figura 65: Sistema de visão embarcada com câmera Logitech C270 a 250 mm do plano



Fonte: O Autor (2021)

Os resultados das medições realizadas estão apresentados nas Tabelas 4 e 5.

Tabela 4: Medições da largura do objeto com câmera Logitech C270 a 250 mm do plano

Medição	Largura (mm)	Erro Absoluto(mm)	Erro Relativo (%)
1	77,25	2,25	3,00%
2	76,07	1,07	1,43%
3	80,2	5,2	6,93%
4	79,06	4,06	5,41%
5	75,83	0,83	1,11%
6	77,03	2,03	2,71%

Fonte: O Autor (2021)

Tabela 5: Medições do comprimento do objeto com câmera Logitech C270 a 250 mm do plano

Medição	Comprimento (mm)	Erro Absoluto (mm)	Erro Relativo (%)
1	125,35	5,35	4,46%
2	121,11	1,11	0,93%
3	122,04	2,04	1,70%
4	123,02	3,02	2,52%
5	125,75	5,75	4,79%
6	124,77	4,77	3,98%

Fonte: O Autor (2021)

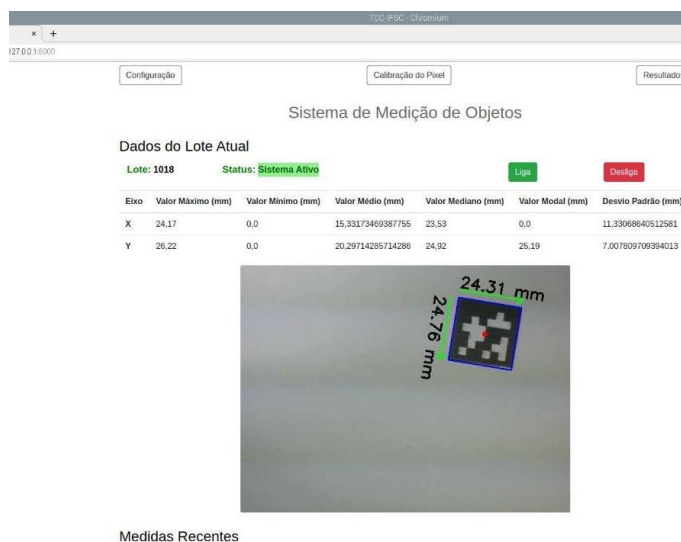
Através dos resultados apresentados, pode-se verificar que as taxas de erro de medição foram de no máximo 6,93%, tal erro causado por sombra nos entornos do objeto.

2.8.3 Câmera Logitech C270 com altura de medição de 160 mm

As mesmas condições do sistema de iluminação do teste anterior foram mantidas, alterando-se a altura do sensor da câmera em relação ao plano para 160 mm, e para manter o bom funcionamento do sistema, foi realizado o ajuste do foco da câmera para essa distância de foco. Houve a troca do objeto de medição, visto que o disco rígido (HD) externo tem dimensões superiores ao campo de alcance do sistema, já o objeto utilizado tem valor verdadeiro convencional de 25 mm de comprimento e 25 mm de largura.

Na etapa de calibração do pixel, se pode realizar a detecção do objeto em análise e obter as relações de transformação de pixel para milímetro. E novamente foram adquiridas imagens em diversas posições dentro do campo de alcance do sistema, além de diversos ângulos de rotação do objeto em torno do próprio eixo. A aplicação em funcionamento para essas condições é apresentada na Figura 66.

Figura 66: Sistema de visão embarcada com câmera Logitech C270 a 160 mm do plano



Fonte: O Autor (2021)

Os resultados das medições realizadas estão apresentados nas Tabelas 6 e 7.

Tabela 6: Medições da largura do objeto com câmera Logitech C270 a 160 mm do plano

Medição	Largura (mm)	Erro Absoluto(mm)	Erro Relativo (%)
1	24,54	-0,46	1,84%
2	24,32	-0,68	2,72%
3	24,31	-0,69	2,76%
4	24,71	-0,29	1,16%
5	24,39	-0,61	2,44%
6	24,53	-0,47	1,88%
7	24,5	-0,5	2,00%

Fonte: O Autor (2021)

Tabela 7: Medições do comprimento do objeto com câmera Logitech C270 a 160 mm do plano

Medição	Comprimento (mm)	Erro Absoluto (mm)	Erro Relativo (%)
1	25,01	0,01	0,04%
2	24,31	-0,69	2,76%
3	24,76	-0,24	0,96%
4	24,62	-0,38	1,52%
5	24,28	-0,72	2,88%
6	24,33	-0,67	2,68%
7	24,73	-0,27	1,08%

Fonte: O Autor (2021)

Através dos resultados apresentados, pode-se verificar que as taxas de erro de medição foram de no máximo 2,88%.

2.8.4 Câmera Raspberry Pi Camera v1 com altura de medição de 345 mm

Foram realizadas aquisições de dimensões utilizando-se a câmera Raspberry Pi Camera v1, essa estava posicionada com seu respectivo sensor a 345 mm do plano e auxiliada por um sistema de iluminação frontal difusa, com fontes de iluminação de duas lâmpadas fluorescentes de 15W e temperatura de cor de 6400 K. Dessa forma, manteve-se as mesmas características do ambiente do teste realizado com a câmera Logitech C270.

Porém, no teste realizado com a Raspberry Pi Camera v1 não se obteve o mesmo desempenho, uma vez que o sistema de visão embarcado não obteve êxito na operação de calibração do pixel, visto que houveram problemas para a detecção do contorno do objeto analisado, conforme apresentado na Figura 67. Tal erro foi causado devido a necessidade de aumentar a intensidade luminosa do sistema de iluminação presente.

Figura 67: Sistema de visão embarcada com câmera Raspberry Pi Camera v1 a 345 mm do plano



Fonte: O Autor (2021)

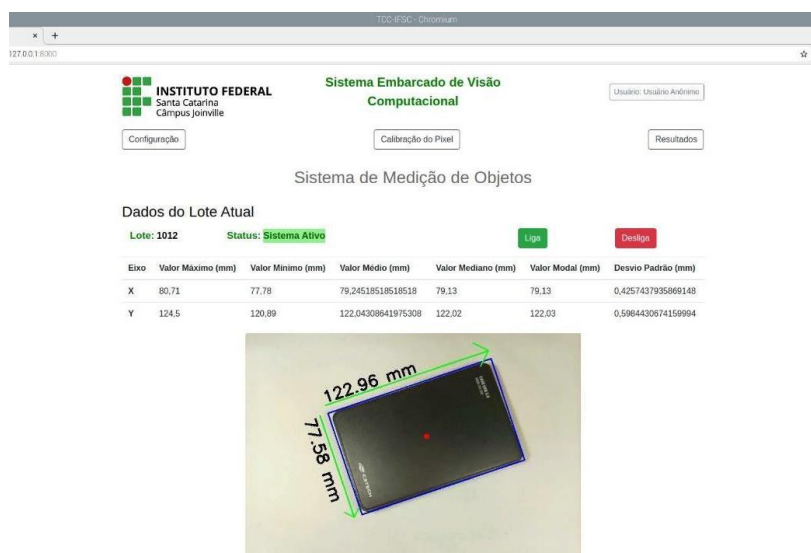
Uma vez que a relação de transformação de pixel para milímetro não foi concluída com êxito, não foi possível realizar o teste do sistema de visão embarcada em modo de operação normal com a câmera Raspberry Pi Camera v1 a altura de 354 mm.

2.8.5 Câmera Raspberry Pi Camera v1 com altura de medição de 250 mm

As mesmas condições do sistema de iluminação do teste anterior foram mantidas, alterando-se a altura do sensor da câmera em relação ao plano para 250 mm. Já o objeto de medição foi o mesmo utilizado em teste anterior, além de ser realizado o processo de calibração do pixel para essa situação.

Na etapa de calibração do pixel, se pode realizar a detecção do objeto em análise e obter as relações de transformação de pixel para milímetro. E novamente foram adquiridas imagens em diversas posições dentro do campo de alcance do sistema, além de diversos ângulos de rotação do objeto em torno do próprio eixo. A aplicação em funcionamento para essas condições é apresentada na Figura 68.

Figura 68: Sistema de visão embarcada com câmera Raspberry Pi Camera v1 a 250 mm do plano



Fonte: O Autor (2021)

Os resultados das medições realizadas estão apresentados nas Tabelas 8 e 9.

Tabela 8: Medições da largura do objeto com câmera Raspberry Pi Camera v1a 250 mm do plano

Medição	Largura (mm)	Erro Absoluto(mm)	Erro Relativo (%)
1	79,11	4,11	5,48%
2	77,58	2,58	3,44%
3	78,81	3,81	5,08%
4	79,34	4,34	5,79%
5	77,87	2,87	3,83%
6	78,6	3,6	4,80%

Fonte: O Autor (2021)

Tabela 9: Medições do comprimento do objeto com câmera Raspberry Pi Camera v1a 250 mm do plano

Medição	Comprimento (mm)	Erro Absoluto (mm)	Erro Relativo (%)
1	122,03	2,03	1,69%
2	122,96	2,96	2,47%
3	121,82	1,82	1,52%
4	122,72	2,72	2,27%
5	123,98	3,98	3,32%
6	120,38	0,38	0,32%

Fonte: O Autor (2021)

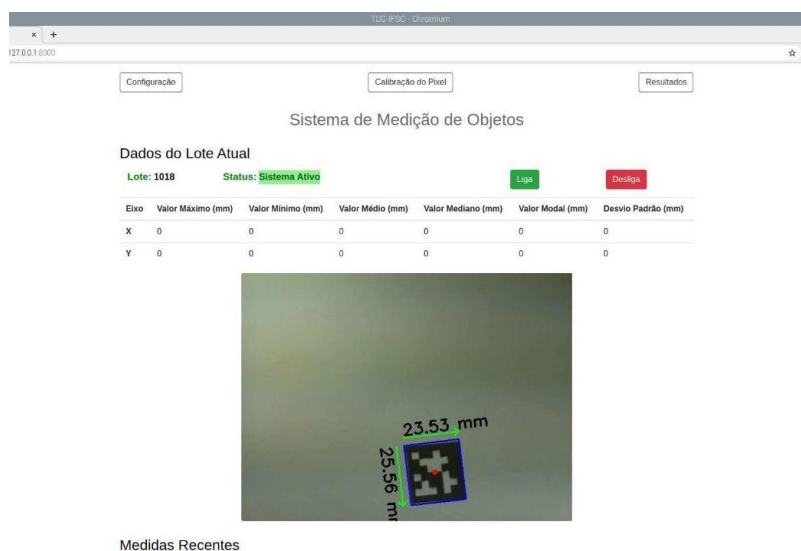
Através dos resultados apresentados, pode-se verificar que a média das taxas de erro de medição da largura do objeto foram superiores às obtidas com a câmera Logitech C270, já nas medições de comprimento a câmera Raspberry Pi Camera v1 teve melhor desempenho.

2.8.6 Câmera Raspberry Pi Camera v1 com altura de medição de 160 mm

Para o teste com a altura do sensor da câmera em relação ao plano à 160 mm, as mesmas condições do sistema de iluminação do teste anterior foram mantidas. Porém houve a troca do objeto de análise, visto que o disco rígido (HD) externo tem dimensões superiores ao campo de alcance do sistema, já o objeto utilizado tem valor verdadeiro convencional de 25 mm de comprimento e 25 mm de largura. Por fim, foi realizado o processo de calibração do pixel para essa situação.

A etapa de calibração do pixel foi realizada de maneira dificultosa, não apresentando bom resultado, visto que a detecção do objeto em análise foi dificultada por erros causados pelo sistema de iluminação do sistema embarcado ter intensidade luminosa insuficiente, apesar dos problemas, as proporções de pixel para milímetros foram obtidas. E assim, novamente foram adquiridas imagens em diversas posições dentro do campo de alcance do sistema, além de diversos ângulos de rotação do objeto em torno do próprio eixo. A aplicação em funcionamento para essas condições é apresentada na Figura 69.

Figura 69: Sistema de visão embarcada com câmera Raspberry Pi Camera v1 a 160 mm do plano



Fonte: O Autor (2021)

Os resultados das medições realizadas estão apresentados nas Tabelas 10 e 11.

Tabela 10: Medições da largura do objeto com câmera Raspberry Pi Camera v1a 160 mm do plano

Medição	Largura (mm)	Erro Absoluto(mm)	Erro Relativo (%)
1	23,74	-1,26	5,04%
2	24,92	-0,08	0,32%
3	23,53	-1,47	5,88%
4	24,71	-0,29	1,16%
5	0	-25	100,00%
6	0	-25	100,00%

Fonte: O Autor (2021)

Tabela 11: Medições do comprimento do objeto com câmera Raspberry Pi Camera v1a 160 mm do plano

Medição	Comprimento (mm)	Erro Absoluto (mm)	Erro Relativo (%)
1	25,18	0,18	0,72%
2	23,72	-1,28	5,12%
3	25,56	0,56	2,24%
4	24,62	-0,38	1,52%
5	0	-25	100,00%
6	11,68	-13,32	53,28%

Fonte: O Autor (2021)

Através dos resultados apresentados, pode-se verificar que a câmera Raspberry Pi Camera v1 não obteve bom desempenho, visto a irregularidade dos valores das medições, e a iluminação do sistema embarcado não ter intensidade luminosa suficiente. Além disso, outro fato que colaborou com esses resultados foi a falta de nitidez da imagem, pois a distância focal não estava ajustada para uma altura de 160 mm do plano de medição.

3 ANÁLISE E DISCUSSÃO DOS RESULTADOS

O presente trabalho foi iniciado através de uma pesquisa de referencial teórico acerca do processamento digital de imagem e sistemas embarcados, a fim de se obter conhecimento sobre o tema, de forma a estudar os conceitos de sistema de visão e sistemas embarcados.

Com clareza nesses conceitos, foi possível identificar as características das placas Raspberry Pi 3 e Raspberry Pi 4 e foram utilizados alguns autores com conhecimento sobre esse hardware, além de dados retirados de manuais do fabricante.

Após as características das placas, foi avaliado as câmeras Raspberry Pi Camera v1 e Logitech C270, através de coleta de dados de manuais dos respectivos fabricantes, além de testes empíricos a fim de avaliar características de imagem, comportamento em relação ao ambiente externo e versatilidade de configurações da própria câmera, como por exemplo o ajuste de foco.

Visto que as informações do hardware e as interfaces do sistema de visão embarcado foram obtidas, foi implementado um algoritmo para a medição de largura e comprimento de objetos com o propósito de poder obter dados sobre o sistema embarcado de visão, e para tornar a gestão dos dados mais eficaz, foi criada uma interface com o usuário acessado via navegador web, para obter os dados e operar o sistema.

Através do estudo dos conceitos de sistema de visão, foi possível constatar que as técnicas de processamento digital de imagem são cruciais para a obtenção de informações úteis sobre o objeto de análise, as etapas dessas são: aquisição, pré-processamento, segmentação, representação e descrição, reconhecimento e interpretação, como defende Pedrini e Schwartz (2007).

Outro ponto de extrema importância para um sistema de visão é a iluminação, Moeslund (2012) afirma que existe a necessidade de que alguma fonte de energia ilumine a área de interesse, e que na maioria das vezes é aplicado uma luz com comprimento de onda pertencente ao espectro visível, Leta et. al. (2016) apontam também que um sistema de iluminação eficiente visa destacar as características do objeto sem adulterá-las.

Moeslund (2012) complementa que para o funcionamento eficiente do sistema de visão, a captação da imagem deve ser feita com a distância de foco da câmera ajustada, caso contrário o efeito é uma imagem fora de foco.

No que se diz respeito ao sistema embarcado, foi possível compreender os principais sistemas que o compõem e que esses configurados corretamente permitem o funcionamento apropriado do mesmo, como afirma Oliveira e Andrade (2009). O mesmo autor, também evidencia as três seções de memória presentes no sistema: memória de programa, memória de dados e memória de armazenamento.

Outra informação coletada, foi a necessidade do uso de interfaces nos sistemas embarcados para haver a comunicação dos sinais de entradas e saídas da unidade de controle desses sistemas, como defende Oliveira e Andrade (2009). E tal unidade tem uma arquitetura, que rege os principais parâmetros relativos à execução lógica de um programa, como afirma Sttalings (2010). Oliveira e Andrade (2009) complementam que o software de um sistema embarcado é denominado de firmware.

Haja vista que tais conceitos teóricos de um sistema embarcado de visão foram coletados, foi possível analisar as características das câmeras, a fim de se obter o melhor resultado com as mesmas. A câmera Raspberry Pi Camera v1, apresentou limitação ao ajuste de foco, visto que não é possível, fato que causa aquisições com baixa nitidez em caso de estar a distâncias curtas do objeto de análise, como 160 mm, e também a longas como 345 mm. Além disso, não existem configurações manuais ajustáveis na câmera, apenas via software, através de bibliotecas e comandos, que permitem ajustes de brilho, saturação, contraste, resolução, entre outros.

De outra parte, a Logitech C270 possui ajuste de foco de maneira manual, que permite ao usuário do sistema de visão embarcado a adequar a câmera para cada altura que essa está posicionada, não perdendo nitidez. Além do foco, pode-se também utilizar parâmetros via software para ajustar o brilho, contraste, saturação, entre outros.

Através de imagens adquiridas em um mesmo ambiente, foi possível constatar uma diferença de saturação na aquisição realizada pelas câmeras Raspberry Pi Camera v1 e Logitech C270, visto que a primeira tem níveis de saturação mais elevados que as da segunda.

Uma vez que o sistema de iluminação é de suma importância para o sistema

de visão, foi possível constatar as diferenças de características de cada geometria, e o efeito gerado na aquisição da imagem, além de notar-se que o sistema com iluminação frontal difusa pode destacar os contornos do objeto, de forma a se tornar uma opção viável para a aplicação do algoritmo de medição de largura e comprimentos de objetos.

Através do desenvolvimento desse algoritmo, foi possível obter diversos dados de largura e comprimento de objetos, aplicados a diversos cenários de altura de captura de imagem. Também foi possível notar a importância de um sistema de iluminação eficiente para o sistema embarcado de visão.

De acordo com os testes realizados pode-se notar que o sistema embarcado de visão obteve melhores resultados com a câmera Logitech C270, para a condição de uma altura de aquisição da câmera em 345 mm em relação ao plano da imagem. Esse resultado decorreu, pois a intensidade luminosa (igual para ambas as câmeras) não foi suficiente para o sistema com a câmera Raspberry Pi Camera v1 poder operar normalmente.

A partir do arquivo com extensão “x/sx” exportado da interface de operação do usuário, com as medições realizadas pelo sistema embarcado de visão computacional, nas quais o valor verdadeiro convencional do mensurando foi de 120 mm de comprimento e 75 mm de largura, notou-se que das 41 aquisições realizadas, obteve-se os dados apresentados na Tabela 12.

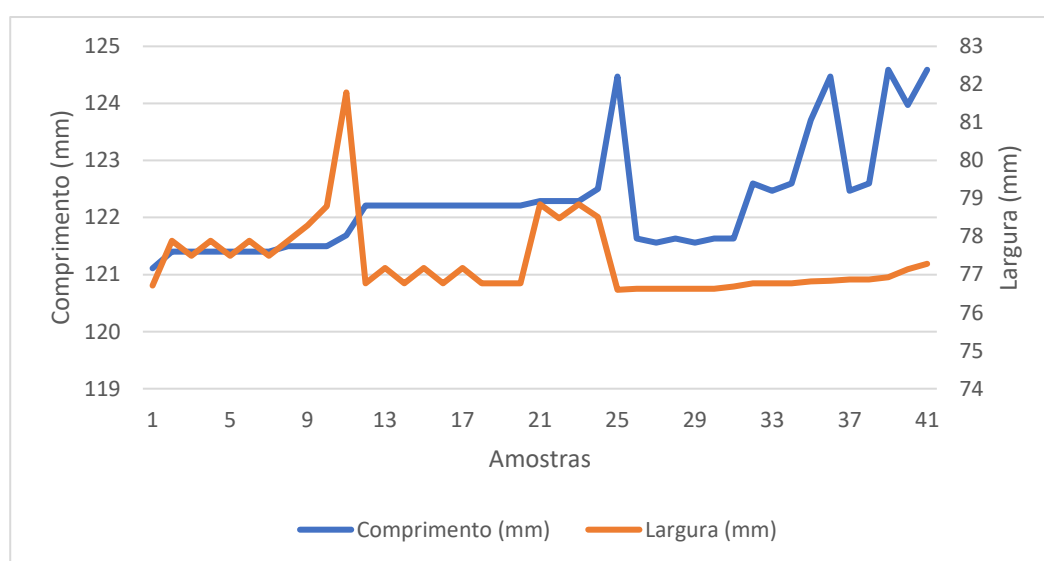
Tabela 12: Resultado das medições com Logitech C270 a 345 mm do plano

Parâmetro de Análise	Comprimento (mm)	Largura (mm)
Média	122,27	77,36
Mediana	122,21	76,87
Moda	122,21	76,77
Valor Máximo	124,59	81,79
Valor Mínimo	121,11	76,6
Desvio Padrão Amostral	0,956	0,992
Tendência	+ 2,27	+ 2,36
Repetitividade	± 1,97	± 2,05

Fonte: O Autor (2021)

Através dos dados, pode-se notar que existem tendências positivas de 2,27 mm para os dados de comprimento e 2,36 mm para os dados de largura. Esses podem ser corrigidos, através de correções de -2,27 mm e -2,36 mm respectivamente, a fim de se compensar o erro sistemático do sistema de medição. A Figura 70 apresenta a série de dados com correção aplicada:

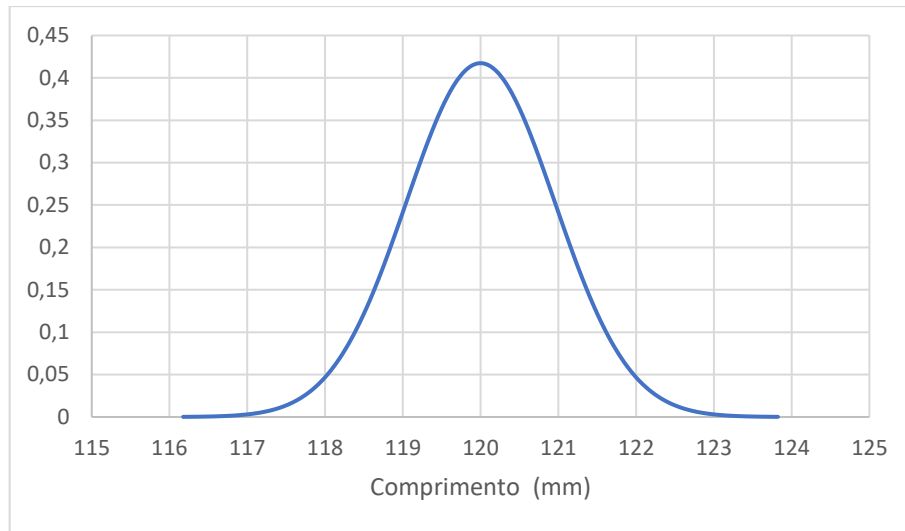
Figura 70: Medições com Logitech C270 a 345 mm de altura com correção



Fonte: O Autor (2021)

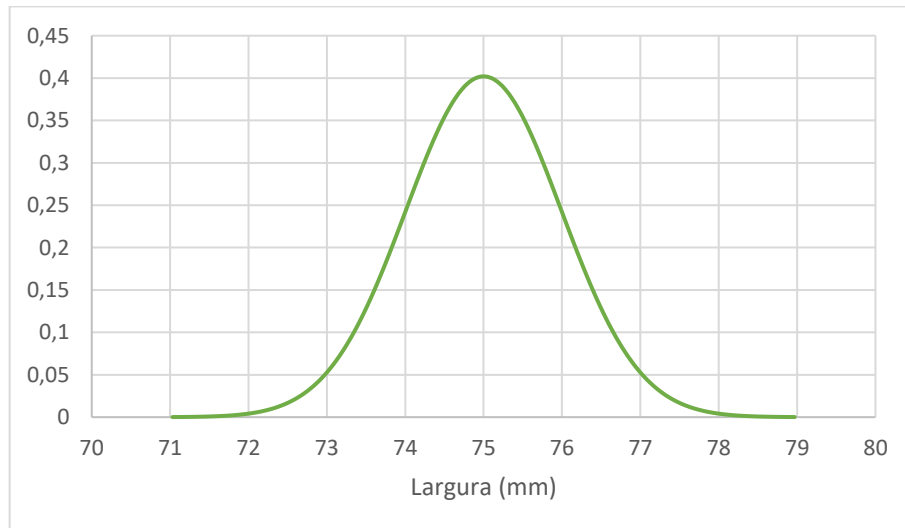
A partir dos dados apresentados nas Figuras 71 e 72, pode-se afirmar que as incertezas padrões do sistema de medição nas características determinadas, são de 0,956 mm para o comprimento e 0,992 mm para a largura, ambas associadas a 40 graus de liberdade. Além disso, pode-se afirmar que para 95,45% das medições realizadas por esse sistema de medição, o erro aleatório esperado é de $\pm 1,97$ mm para o comprimento e $\pm 2,05$ mm para a largura (JUNIOR; SOUSA, 2018).

Figura 71: Distribuição normal das medições de comprimento com Logitech C270 a 345 mm de altura



Fonte: O Autor (2021)

Figura 72: Distribuição normal das medições de largura com Logitech C270 a 345 mm de altura



Fonte: O Autor (2021)

Para a condição na qual a altura de aquisição das câmeras estava à 250 mm em relação ao plano da imagem, e o valor verdadeiro convencional do mensurando foi de 120 mm de comprimento e 75 mm de largura, foi possível obter dados com ambas as câmeras. Para a câmera Logitech C270 foram realizadas 81 aquisições,

obteve-se os seguintes dados apresentados na Tabela 13.

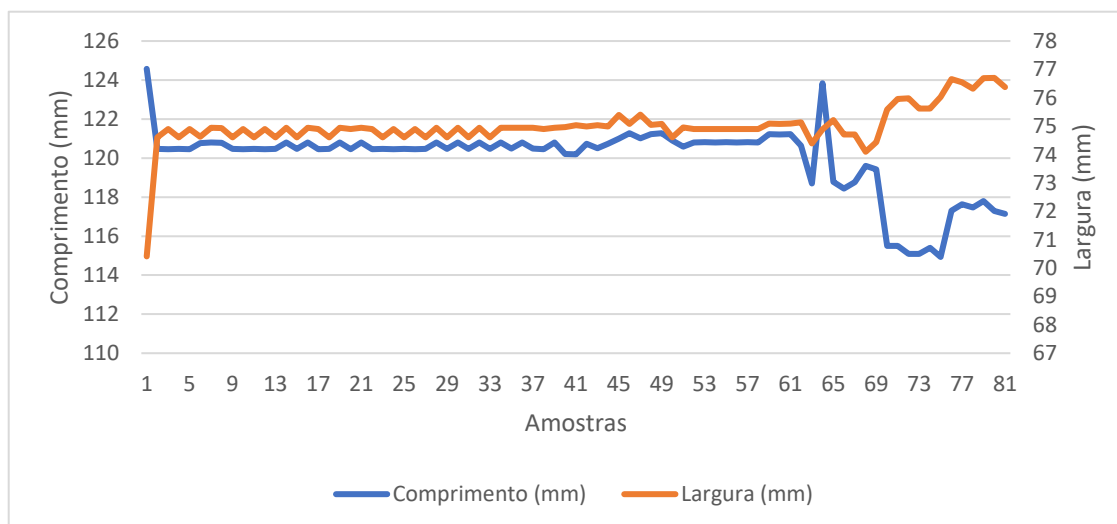
Tabela 13: Resultado das medições com Logitech C270 a 250 mm do plano

Parâmetro de Análise	Comprimento (mm)	Largura (mm)
Média	124,88	77,64
Mediana	125,35	77,58
Moda	125,35	77,54
Valor Máximo	129,45	79,34
Valor Mínimo	119,81	73,05
Desvio Padrão Amostral	1,771	0,744
Tendência	+ 4,88	+ 2,64
Repetitividade	± 3,60	± 1,51

Fonte: O Autor (2021)

A partir dos dados, pode-se notar que existem tendências positivas de 4,88 mm para os dados de comprimento e 2,64 mm para os dados de largura. Esses podem ser corrigidos, através de correções de -4,88 mm e -2,64 mm respectivamente, a fim de se compensar o erro sistemático do sistema de medição. A Figura 73 apresenta a série de dados com correção aplicada:

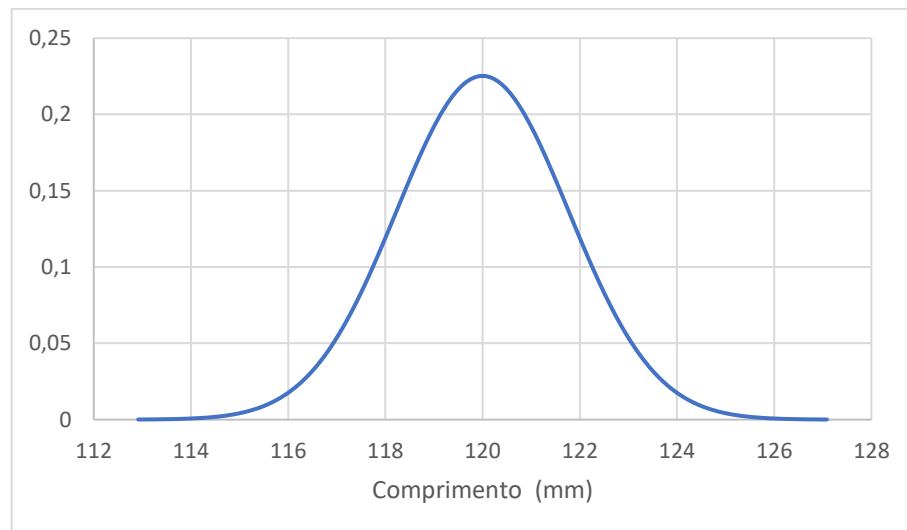
Figura 73: Medições com Logitech C270 a 250 mm de altura com correção



Fonte: O Autor (2021)

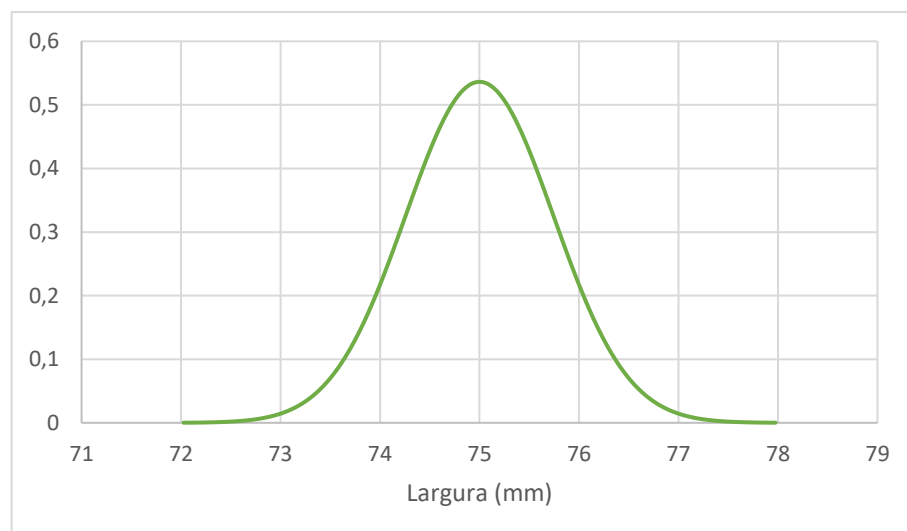
A partir dos dados apresentados nas Figuras 74 e 75, foi possível observar que as incertezas padrões do sistema de medição nas características determinadas, são de 1,771 mm para o comprimento e 0,744 mm para a largura, ambas associadas a 80 graus de liberdade. Além disso, pode-se afirmar que para 95,45% das medições realizadas por esse sistema de medição, o erro aleatório esperado é de $\pm 3,60$ mm para o comprimento e $\pm 1,51$ mm para a largura (JUNIOR; SOUSA, 2018).

Figura 74: Distribuição normal das medições de comprimento com Logitech C270 a 250 mm de altura



O Autor (2021)

Figura 75: Distribuição normal das medições de largura com Logitech C270 a 250 mm de altura



O Autor (2021)

Também foram realizadas aquisições com a câmera Raspberry Pi Camera v1, a partir das 248 aquisições realizadas, foi possível obter os seguintes resultados apresentados na Tabela 14.

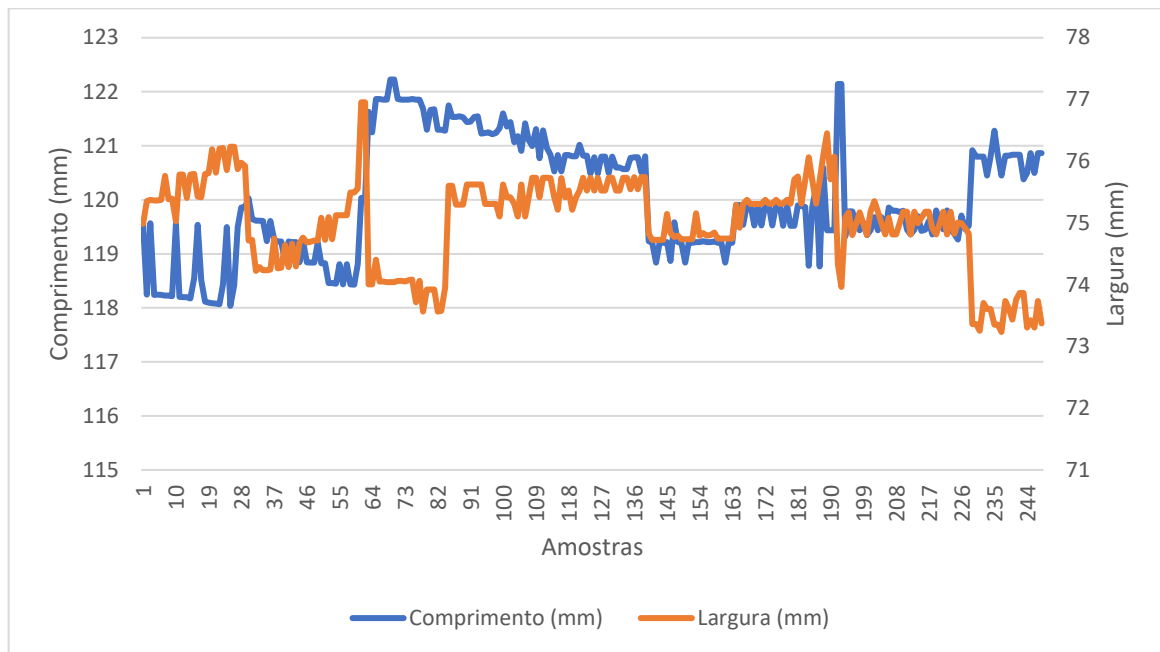
Tabela 14: Resultado das medições com Raspberry Pi Camera v1 a 250 mm do plano

Parâmetro de Análise	Comprimento (mm)	Largura (mm)
Média	122,12	78,82
Mediana	121,92	78,96
Moda	121,33	79,55
Valor Máximo	124,35	80,77
Valor Mínimo	120,16	77,05
Desvio Padrão Amostral	1,076	0,732
Tendência	+ 2,12	+ 3,82
Repetitividade	± 2,17	± 1,47

Fonte: O Autor (2021)

Como apresentado na Tabela 14, pode-se notar que existem tendências positivas de 2,12 mm para os dados de comprimento e 3,82 mm para os dados de largura. Esses podem ser corrigidos, através de correções de -2,12 mm e -3,82 mm respectivamente, a fim de se compensar o erro sistemático do sistema de medição. A Figura 76 apresenta a série de dados com correção aplicada:

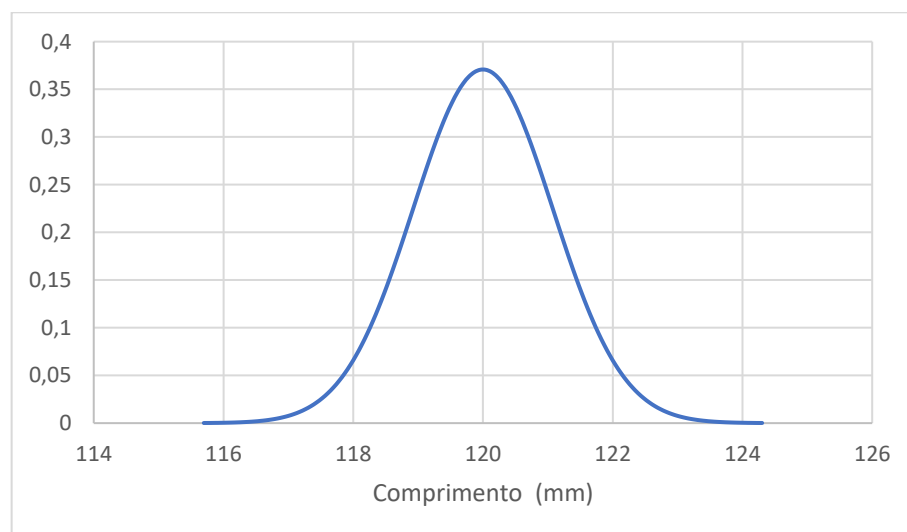
Figura 76: Medições com Raspberry Pi Camera v1 a 250 mm de altura com correção



Fonte: O Autor (2021)

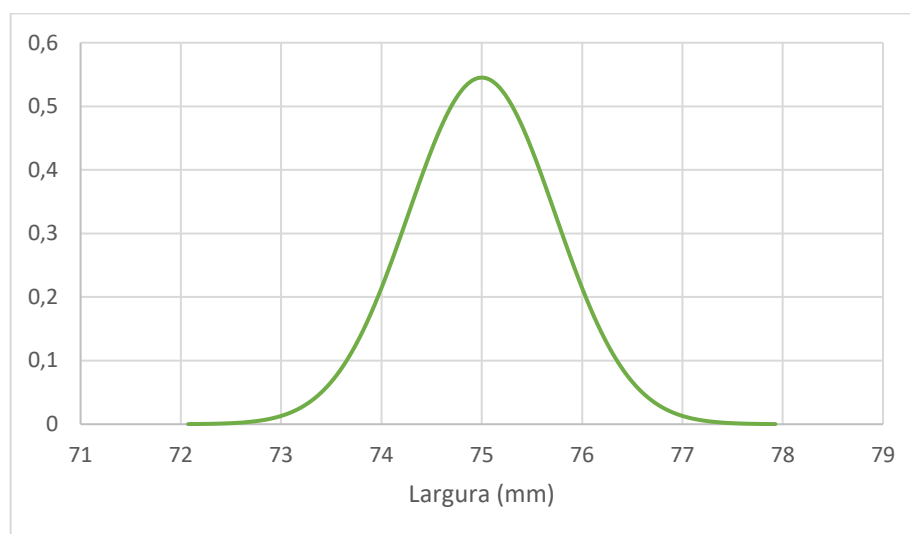
A partir dos dados apresentados nas Figuras 76 e 77, pode-se afirmar que as incertezas padrões do sistema de medição nas características determinadas, são de 1,076 mm para o comprimento e 0,732 mm para a largura, ambas associadas a 247 graus de liberdade. Além disso, pode-se afirmar que para 95,45% das medições realizadas por esse sistema de medição, o erro aleatório esperado é de 2,17 mm para o comprimento e 1,47 mm para a largura (JUNIOR; SOUSA, 2018).

Figura 77: Distribuição normal das medições de comprimento com Raspberry Pi Camera v1 a 250 mm de altura



Fonte: O Autor (2021)

Figura 78: Distribuição normal das medições de largura com Raspberry Pi Camera v1 a 250 mm de altura



Fonte: O Autor (2021)

Outro resultado encontrado foram as medições do sistema embarcado de visão com as câmeras posicionadas à 160 mm em relação ao plano da imagem, e o valor verdadeiro convencional do mensurando foi de 25 mm de comprimento e 25 mm de largura. Para ambas as câmeras foi possível obter dados, para a câmera Logitech C270 foram realizadas 46 aquisições e obteve-se os dados apresentados na Tabela 15.

Tabela 15: Resultado das medições com Logitech C270 a 160 mm do plano

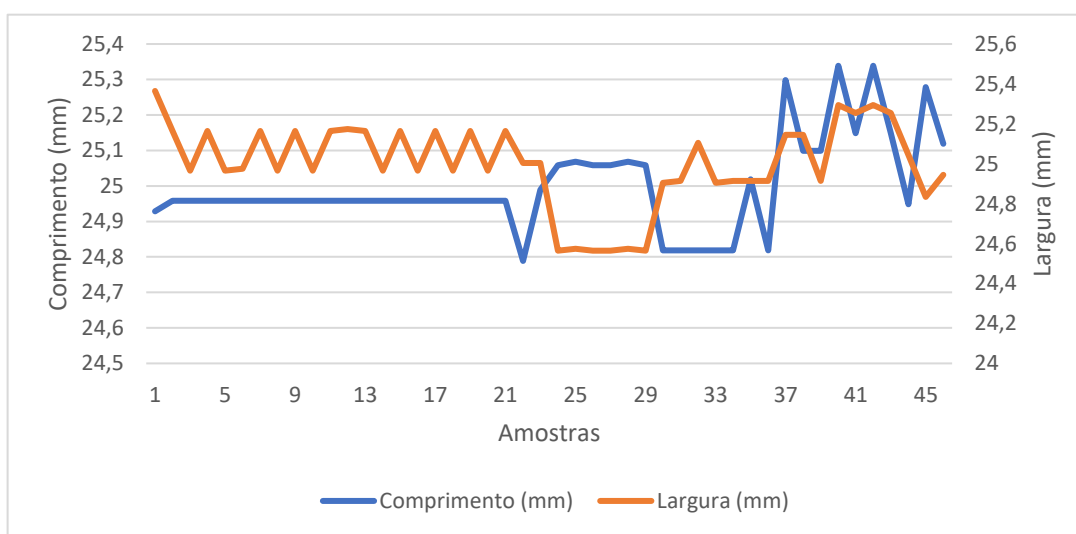
Parâmetro de Análise	Comprimento (mm)	Largura (mm)
Média	24,57	24,37
Mediana	24,53	24,34
Moda	24,53	24,53
Valor Máximo	24,91	24,73
Valor Mínimo	24,36	23,93
Desvio Padrão Amostral	0,132	0,211
Tendência	-0,43	-0,63
Repetitividade	± 0,27	± 0,44

Fonte: O Autor (2021)

Através dos dados, pode-se notar que existem tendências negativas de 0,43

mm para os dados de comprimento e 0,63 mm para os dados de largura. Esses podem ser corrigidos, através de correções de +0,43 mm e +0,63 mm respectivamente, a fim de se compensar o erro sistemático do sistema de medição. A Figura 79 apresenta a série de dados com correção aplicada:

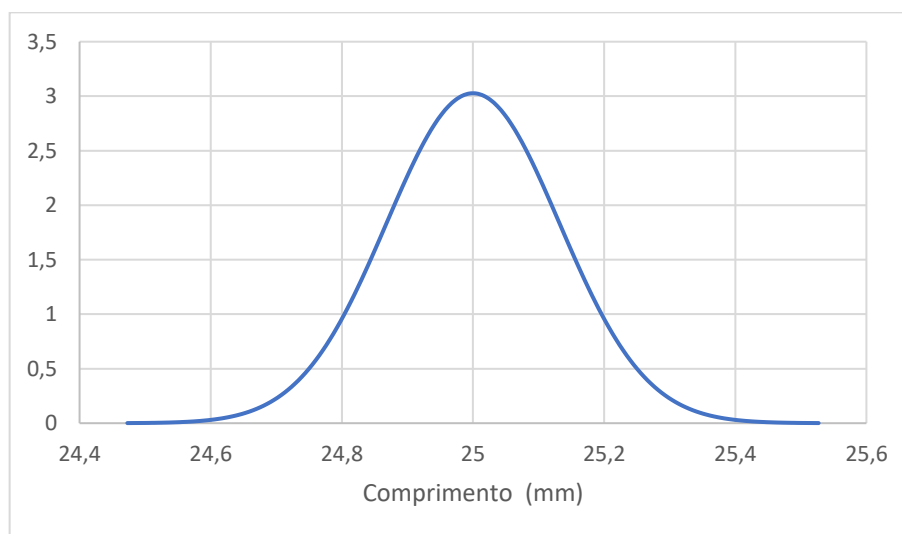
Figura 79: Medições com Logitech C270 a 160 mm de altura com correção



Fonte: O Autor (2021)

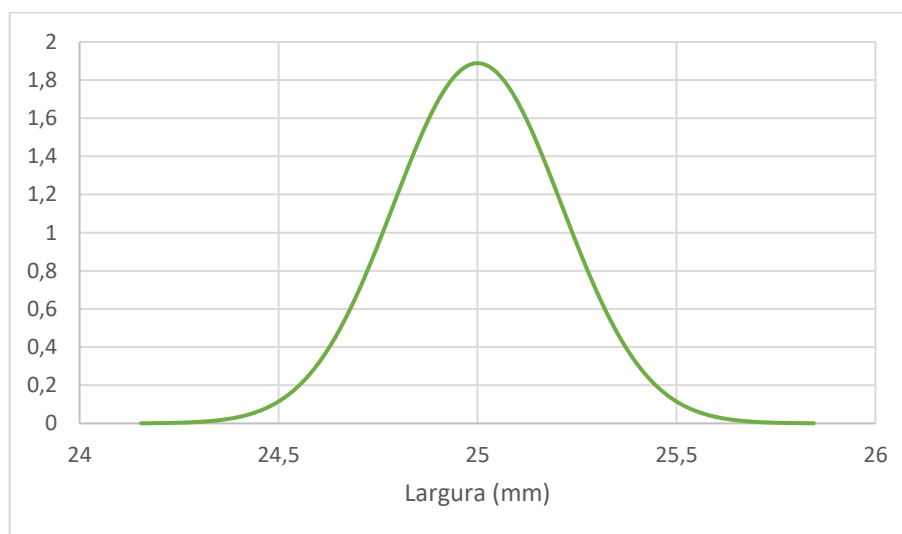
A partir dos dados apresentados nas Figuras 80 e 81, foi possível observar que as incertezas padrões do sistema de medição nas características determinadas, são de 0,132 mm para o comprimento e 0,211 mm para a largura, ambas associadas a 45 graus de liberdade. Além disso, pode-se afirmar que para 95,45% das medições realizadas por esse sistema de medição, o erro aleatório esperado é de 0,27 mm para o comprimento e 0,44 mm para a largura (JUNIOR; SOUSA, 2018).

Figura 80: Distribuição normal das medições de comprimento com Logitech C270 a 160 mm de altura



Fonte: O Autor (2021)

Figura 81: Distribuição normal das medições de largura com Logitech C270 a 160 mm de altura



Fonte: O Autor (2021)

Por fim, foram realizadas as aquisições com a câmera Raspberry Pi Camera v1, a partir das 25 aquisições realizadas, foi possível obter os seguintes resultados apresentados na Tabela 16.

Tabela 16: Resultado das medições com Raspberry Pi Camera v1 a 160 mm do plano

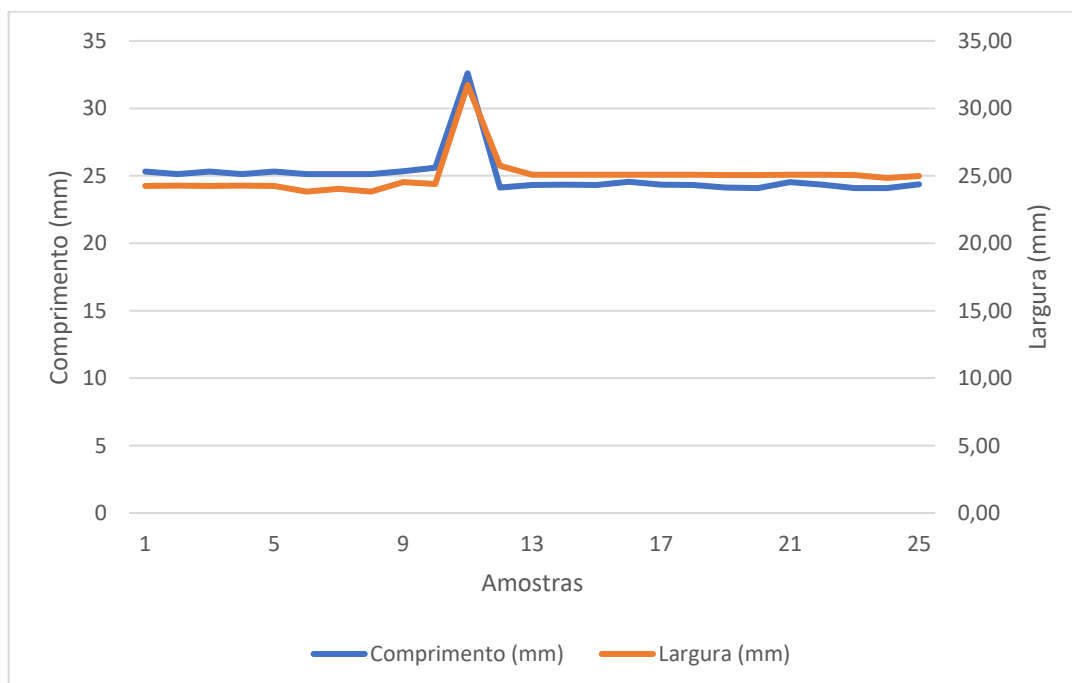
Parâmetro de Análise	Comprimento (mm)	Largura (mm)
Média	24,19	25,48

Mediana	23,72	25,54
Moda	24,51	25,56
Valor Máximo	31,79	32,22
Valor Mínimo	23,28	24,31
Desvio Padrão Amostral	1,661	1,487
Tendência	-0,81	-0,48
Repetitividade	3,54	3,17

Fonte: O Autor (2021)

A partir dos dados, pode-se notar que existem tendências negativas de -0,81 mm para os dados de comprimento e -0,48 mm para os dados de largura. Esses podem ser corrigidos, através correções de +0,81 mm e +0,48 mm respectivamente, a fim de se compensar o erro sistemático do sistema de medição. A Figura 82 apresenta a série de dados com correção aplicada:

Figura 82: Medições com Raspberry Pi Camera v1 a 160 mm de altura com correção

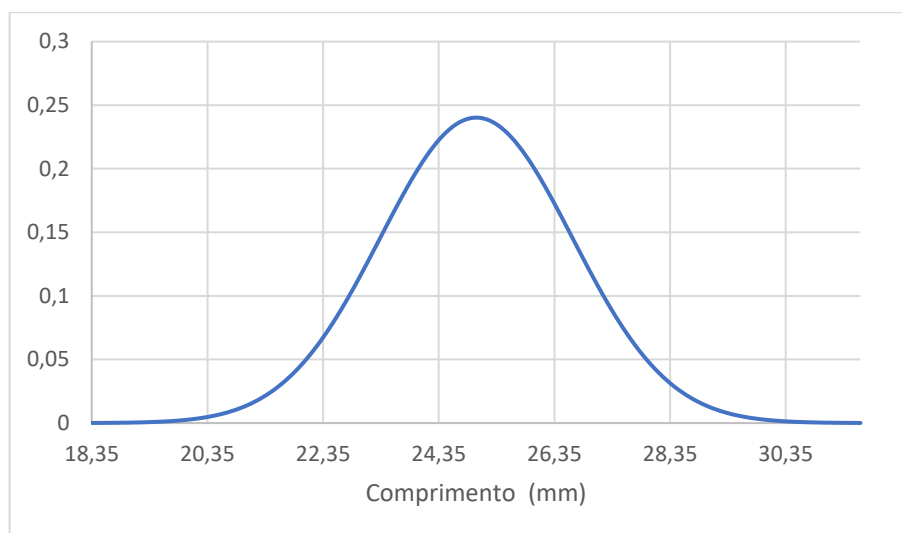


Fonte: O Autor (2021)

Como apresentado nas Figuras 83 e 84, pode-se afirmar que as incertezas padrões do sistema de medição nas características determinadas, são de 1,661 mm

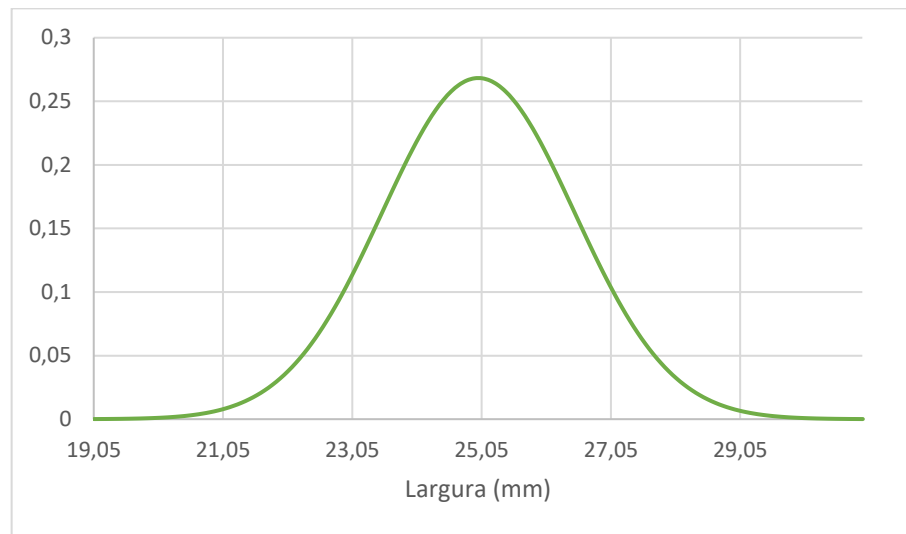
para o comprimento e 1,487 mm para a largura, ambas associadas a 24 graus de liberdade. Além disso, pode-se afirmar que para 95,45% das medições realizadas por esse sistema de medição, o erro aleatório esperado é de 3,54 mm para o comprimento e 3,17 mm para a largura (JUNIOR; SOUSA, 2018).

Figura 83: Distribuição normal das medições de comprimento com Raspberry Pi Camera v1 a 160 mm de altura



Fonte: O Autor (2021)

Figura 84: Distribuição normal das medições de largura com Raspberry Pi Camera v1 a 160 mm de altura



Fonte: O Autor (2021)

Através dos dados adquiridos, foi possível interpretar que a câmera Raspberry Pi Camera v1 apresentou pior desempenho nos testes com altura da câmera a 160 mm e 345 mm do plano da imagem, por conta da intensidade luminosa do sistema de iluminação não ser suficiente para uma aquisição de dados correta e pela falta de nitidez das imagens, tal interpretação é baseada nos parâmetros de incerteza-padrão e repetitividade do sistema de medição.

Já para o teste com altura da câmera a 250 mm do plano da imagem, ambas as câmeras puderam obter as relações de transformação de pixel para milímetro de forma a permitir a medição dimensional dos objetos, porém baseada nos parâmetros de incerteza-padrão e repetitividade do sistema de medição, pode-se notar que a câmera Raspberry Pi Camera v1 obteve melhor desempenho.

3.1 Custo dos dispositivos

Com o propósito de verificar a viabilidade econômica do uso de sistemas de visão embarcado em sistemas de manufatura, com o caso aplicado de medição dimensional de objetos, todos os custos da aplicação são apresentados na Tabela 17.

Tabela 17: Custo da aplicação

Material	Qtde.	Preço Unitário	Preço Total
Raspberry Pi 3 Model B	1	R\$ 820,00	R\$ 820,00

Raspberry Pi Camera v1	1	R\$ 45,00	R\$ 45,00
Case para Raspberry Pi 3	1	R\$ 39,39	R\$ 39,39
Cooler e dissipadores para Raspberry Pi 3 B	1	R\$ 21,80	R\$ 21,80
Fonte chaveada 5 V/3 A	1	R\$ 63,25	R\$ 63,25
Câmera Logitech C270	1	R\$ 198,35	R\$ 198,35
Lâmpada fluorescente 15W/220V	2	R\$ 27,00	R\$ 54,00
Suporte para as lâmpadas	2	R\$ 17,49	R\$ 34,98
Perfil de alumínio 20X20 V-Slot canal 6	1,85	R\$ 31,00	R\$ 57,35
Cantoneira em alumínio 20x20x17 mm	6	R\$ 5,83	R\$ 34,98
Porca martelo canal 6 M5	12	R\$ 1,66	R\$ 19,92
Parafuso allen cabeça cônica M5 10 mm	12	R\$ 1,00	R\$ 12,00
Cabos	-	-	R\$ 11,50
Valor Total	-	-	R\$ 1.412,52

Fonte: O Autor (2021)

Dessa forma, com base nos dados e custos apresentados, pode-se interpretar que os sistemas embarcados são uma alternativa de baixo custo a sistemas de visão comerciais. Uma vez que esses sistemas têm capacidade de hardware para executar aplicações em sistemas de manufatura, através da tecnologia empregada de seus respectivos processadores, além da possibilidade de comunicar-se com outros dispositivos inseridos na cadeia de um processo produtivo.

Através da aplicação desenvolvida, foi possível demonstrar a importância do domínio das características e peculiaridades de cada componente do sistema de visão embarcado a fim de que esses possam funcionar. Além disso, foi possível comprovar a versatilidade da interface do usuário desses sistemas.

E baseado nos parâmetros de incerteza-padrão e repetitividade encontrados, demonstra-se que dependendo da precisão requerida do processo produtivo, os sistemas de visão embarcada são uma alternativa de baixo custo.

4 CONCLUSÃO

Quando se iniciou o presente trabalho de pesquisa, foi possível constatar que a utilização de sistemas de visão pode impactar diretamente a indústria de manufatura, através da coleta de dados de produção e a conectividade entre dispositivos envolvidos no processo de fabricação. Porém um dos reveses dos sistemas de visão é o alto custo de aquisição. Dessa forma, foi de suma importância estudar sobre a utilização de visão computacional embarcada em sistemas de manufatura.

Diante disso, a pesquisa teve como objetivo geral estudar a viabilidade da utilização de sistemas embarcados no desenvolvimento de algoritmos de visão computacional para aplicação na manufatura, e constatou-se que o objetivo geral foi atendido pois o trabalho conseguiu verificar que tais sistemas embarcados tem capacidade técnica para tal função, e através da aplicação de algoritmos de visão se pode comprovar que a viabilidade do uso desses está atrelada com a precisão requerida pelo processo produtivo, demonstrados através dos resultados de incerteza padrão amostral e repetitividade obtidos. Também se pode destacar a versatilidade da interface com o usuário que permite o acompanhamento e análise dos dados em tempo real.

A pesquisa também teve alguns objetivos específicos, o primeiro desses foi estudar conceitos de sistema de visão, e se pode verificar que o objetivo foi atingido, uma vez que o trabalho conseguiu apresentar conceitos e processo de aquisição, representação, segmentação e suavização de imagens digitais.

Outro objetivo específico foi estudar conceitos de sistemas embarcados, de maneira que pode ser atingido uma vez que o presente trabalho apresenta conceitos sobre o hardware, processadores, interfaces de comunicação e softwares desses dispositivos.

Com base nos conceitos de sistemas embarcados e processamento digital de imagem, foi possível atingir o objetivo específico de estudar as características de câmeras, que foram testadas a partir de seus parâmetros e as influências da iluminação nas imagens por elas capturadas. Além disso, foram estudadas as características e detalhes das placas Raspberry Pi 3 e Raspberry Pi 4.

A fim de analisar a aplicação dos sistemas embarcados de visão, o trabalho também teve como objetivo específico a implementação de algoritmo de medição bidimensional através de visão computacional aplicado a produtos manufaturados. Esse objetivo específico pode ser atendido pois o trabalho conseguiu desenvolver esse algoritmo, além de uma interface para o usuário controlar e parametrizar a aplicação do sistema embarcado de visão.

O último objetivo específico do trabalho foi avaliar o desempenho do algoritmo de medição bidimensional em sistemas didáticos de manufatura dos laboratórios do IFSC campus Joinville. Foi possível demonstrar que o objetivo específico foi atingido visto que foram realizados testes, e adquiridos valores das medições de comprimento e largura dos objetos analisados. Apesar de não serem realizados nos sistemas didáticos de manufatura dos laboratórios do IFSC campus Joinville, foi feito em ambiente semelhante.

Para o efetivo desenvolvimento deste trabalho, o estudo teve como base uma pesquisa exploratória, e os resultados foram apresentados de forma quantitativa, a partir da coleta de informações de fontes primárias e secundárias. A planificação da pesquisa, inclui em primeiro lugar, o levantamento dos dados secundários, através de livros, documentos, artigos e autores significativos para o tema, como Gonzales e Woods (2009), Pedrini (2008) e Oliveira e Andrade (2009). Já os dados primários foram gerados a partir da execução de um algoritmo de medição dimensional de produtos manufaturados em uma Raspberry Pi 3 associado a duas câmeras modelo Logitech C270 e Raspberry Pi Camera v1. Essa geração de dados foi realizada em um ambiente de simulação, através de uma estrutura de alumínio para comportar o sistema embarcado de visão. Com os dados gerados, esses foram obtidos através da interface do usuário da aplicação desenvolvida.

Com os dados, discussões e conceitos apresentados no trabalho, foi possível notar que o problema de pesquisa foi completamente respondido visto que se desenvolveu um sistema embarcado de visão economicamente viável e com funcionalidade relativa à precisão requerida do processo produtivo, evidenciada pelos resultados de incerteza-padrão amostral e repetitividade. Além de serem apresentados diversos fatores importantes para a consolidação desses sistemas em aplicações de manufatura.

Entretanto, como possibilidade de melhoria do presente trabalho, poderia ter sido feito uma análise mais ampla na bibliográfica para analisar os aspectos de erro e incertezas de medição em um sistema de visão, haja vista que é de suma importância para entender as fontes de erro e quantificar a incerteza de medição de cada módulo do sistema de medição e a sua propagação no sistema como um todo.

No que se diz respeito ao algoritmo de medição por imagem, recomenda-se que seja realizada a segmentação da imagem antes da obtenção do contorno do objeto. Além de realizar dois algoritmos de calibração de pixel, o primeiro para a primeira calibração do pixel do sistema em determinadas condições, obtendo-se 100 relações de transformação de pixel para milímetro, através de uma matriz de quadrantes de 10 x 10. Já o segundo, utilizar as 9 relações de transformação de pixel para milímetro a fim de realizar ajustes rotineiros para o sistema de medição por imagem.

Já no algoritmo de separação dos segmentos de reta, realizar o cálculo da equação linear dos quatro segmentos de retas que representam as dimensões do objeto em análise, e buscar por todos os pontos que são comuns a esses segmentos e os limitantes dos quadrantes. A fim de se obter a medida em milímetro para todos os quatro segmentos, e então realizar a média dos segmentos perpendiculares, a fim de reduzir o erro.

Com relação ao sistema de iluminação, buscar uma solução que apresente menos sombras para o sistema embarcado de visão.

Outra possibilidade de melhoria é um invólucro para o sistema embarcado de visão, contendo uma interface de sinais digitais e analógicos para equipamentos específicos, como controladores lógicos programáveis.

REFERÊNCIAS

ALLIED VISION. **Embedded Vision vs. Machine Vision**. [S. l.], 28 jan. 2021. Disponível em: <https://www.azom.com/article.aspx?ArticleID=19487>. Acesso em: 7 ago. 2021.

ALMEIDA, Rodrigo Maximiano A.; MORAES, Carlos Henrique V.; SERAPHIM, Thatyana F. Piola. **Programação de Sistemas Embarcados**: Desenvolvendo Software para Microcontroladores em Linguagem C. 1. ed. Rio de Janeiro: GEN LTC, 2016. 488 p. ISBN 8535285180.

ARM HOLDINGS. **Arm Cortex-A53 MPCore Processor Technical Reference Manual**. r0p4. ed. atual. [S. l.: s. n.], 2018. 623 p. Disponível em: <https://developer.arm.com/documentation/ddi0500/j/Introduction/About-the-Cortex-A53-processor?lang=en>. Acesso em: 7 ago. 2021.

ARM HOLDINGS. **CORTEX-A53**. [S. l.], 2021. Disponível em: <https://www.arm.com/products/silicon-ip-cpu/cortex-a/cortex-a53>. Acesso em: 5 ago. 2021.

BOVIK, AI. **The Essential Guide to Image Processing**. 1. ed. Burlington: Academic Press, 2009. 853 p. ISBN 0123744571.

BRADSKI, Gary; KAEHLER, Adrian. **Learning OpenCV**. 1. ed. Sebastopol: O'Reilly Media, 2008. 580 p. ISBN 0596516134.

BRITTO, Eduardo. **Qualidade total: controle da qualidade**. São Paulo. Cengage, 2016.

COGNEX. **Industry 4.0 and Machine Vision: Industry 4.0**, or "The Industrial Internet of Things," will rely upon machine vision to revolutionize industrial automation.. [S. l.], [2021?]. Disponível em: <https://www.cognex.com/pt-br/what-is/industry-4-0-machine-vision>. Acesso em: 18 set. 2021.

COSTA, R. S. d. **Inovações tecnológicas: aquisição de dados e interface de comunicação em instrumentos de medição de massa**. 2005.

DAVIES, E. R. **Computer and Machine Vision: Theory, Algorithms, Practicalities**. 4. ed. Waltham: Academic Press, 2012. 912 p. ISBN 9780123869081.

DINIZ, Paulo S. R.; SILVA, Eduardo A. B. da; NETTO, Sergio L. **Processamento Digital de Sinais: Projeto e Análise de Sistemas**. 2. ed. Porto Alegre: Bookman, 2014. 1000 p. ISBN 8582601239.

DONAT, Wolfram. **Programação do Raspberry Pi com Python: Aprenda a Programar no Pequeno Computador Mais Popular do Mundo**. 1. ed. [S. l.]: Novatec, 2019. 296 p. ISBN 8575227203.

EMBARCADOS. **ARM anuncia que já vendeu 160 bilhões de chips com seu core**. [S. l.], 27 fev. 2020. Disponível em: <https://www.embarcados.com.br/arm-160-bilhoes-de-chips/>. Acesso em: 5 ago. 2021.

EMBARCADOS. **Breve Histórico da ARM**. [S. l.], 22 ago. 2017. Disponível em: <https://www.embarcados.com.br/breve-historico-da-arm/>. Acesso em: 5 ago. 2021.

FIERCE ELETRONICS. **Fundamentals: What is embedded vision?**. [S. l.], 26 maio 2021. Disponível em: <https://www.fierceelectronics.com/electronics/fundamentals-what-embedded-vision>. Acesso em: 7 ago. 2021.

FORSYTH, David A.; PONCE, Jean. **Computational vision: A Modern Approach**. 2. ed. Nova Jersey: Pearson, 2011. 800 p. ISBN 013608592X.

GARCIA, Fernando Deluno. **Introdução aos sistemas embarcados e microcontroladores**. [S. l.], 19 jul. 2018. Disponível em: <https://www.embarcados.com.br/sistemas-embarcados-e-microcontroladores/>. Acesso em: 16 jun. 2021.

GONZALEZ, Rafael C.; WOODS, Richard C. **Processamento Digital de Imagens**. 3. ed. São Paulo: Pearson Prentice Hall, 2009. 624 p. ISBN 8576054019.

HARRY BALLARD, Dana; BROWN, Christopher M. **Computational vision**. [S. l.]: Prentice-Hall, 1982. 523 p.

JUNIOR, Armando Albertazzi G.; SOUSA, André R. de. **Fundamentos de metrologia científica e industrial: Revisada, Atualizada e Ampliada**. 2. ed. Barueri: Manole, 2018. 480 p. ISBN 8520433758.

KULARATNA, Nihal. **Modern Component Families and Circuit Block Design**. 1. ed. [S. l.]: Butterworth-Heinemann, 2000. 384 p. ISBN 0750699922.

LETA, Fabiana Rodrigues; BALDNER, Felipe de Oliveira; GOMES, Julia Freitas Santos; COSTA, Pedro Bastos. **Metrologia por Imagem**. [S. l.]: Elsevier, 2016. 312 p. ISBN 9788595154711.

LOGITECH. **C270 HD Webcam**. [S. l.], 2021. Disponível em: <https://www.logitech.com/pt-br/products/webcams/c270-hd-webcam.960-000694.html>. Acesso em: 3 ago. 2021.

LOWE, David. **The Computational vision Industry**. [S. l.], 2015. Disponível em: <https://www.cs.ubc.ca/~lowe/vision.html>. Acesso em: 18 set. 2021.

MACHADO, Francis Berenger; MAIA, Luiz Paulo. **Arquitetura de Sistemas Operacionais**. 5. ed. [S. l.]: LTC, 2013. 266 p. ISBN 8521622104.

MACHADO, Francis Berenger; MAIA, Luiz Paulo. **Arquitetura de Sistemas Operacionais**: Incluindo Exercícios com o Simulador SOSIM e Questões do ENADE. 5. ed. [S. l.]: LTC, 2013. 266 p. ISBN 8521622104.

MARRONI, L. S. **Aplicação da Transformada de Hough para Localização dos Olhos em Faces Humanas**. Dissertação (Mestrado em Engenharia Elétrica) — Universidade de São Paulo, 2002.

MOESLUND, Thomas B. **Introduction to Video and Image Processing: Building Real Systems and Applications**. 1. ed. Nova Iorque: Springer, 2012. 227 p. ISBN 1447125029.

NXP SEMICONDUCTORS. **I²C-bus specification and user manual**. 6. ed. [S. l.: s. n.], 2014. 64 p.

O QUE é uma linguagem de programação e quais os tipos existem? [S. l.], 14 maio 2019. Disponível em: <https://rockcontent.com/br/blog/linguagem-de-programacao/>. Acesso em: 18 set. 2021.

OIML. International Vocabulary of Metrology – Basic and General Concepts and Associated Terms (VIM). **OIML V 2-200**, Paris, ed. 3, p. 54, 2007. Disponível em: https://www.oiml.org/en/files/pdf_v/v002-200-e07.pdf. Acesso em: 15 set. 2021.

OLIVEIRA, André Schneider de; ANDRADE, Fernando Souza de. **Sistemas embarcados**: Hardware e Firmware na prática. 2. ed. [S. l.]: Érica, 2009. 320 p. ISBN 8536501057.

OLIVEIRA, N. M. de. **Paralelo entre as arquiteturas Von Neumann e Harvard**. 2014

OPENCV. **Structural Analysis and Shape Descriptors: ContourApproximationModes**. 3.4.15. [S. l.], 5 jul. 2021. Disponível em: https://docs.opencv.org/3.4.15/d3/dc0/group__imgproc__shape.html#ga4303f45752694956374734a03c54d5ff. Acesso em: 26 set. 2021.

OPENCV. **Structural Analysis and Shape Descriptors: contourArea()**. 3.4.15. [S. l.], 5 jul. 2021. Disponível em: https://docs.opencv.org/3.4.15/d3/dc0/group__imgproc__shape.html#ga2c759ed9f497d4a618048a2f56dc97f1. Acesso em: 26 set. 2021.

OPENCV. **Structural Analysis and Shape Descriptors: findContours()**. 3.4.15. [S. l.], 5 jul. 2021. Disponível em: https://docs.opencv.org/3.4.15/d3/dc0/group__imgproc__shape.html#ga17ed9f5d79ae97bd4c7cf18403e1689a. Acesso em: 26 set. 2021.

OPENCV. **Structural Analysis and Shape Descriptors: RetrievalModes**. 3.4.15. [S. l.], 5 jul. 2021. Disponível em: https://docs.opencv.org/3.4.15/d3/dc0/group__imgproc__shape.html#ga819779b9857cc2f8601e6526a3a5bc71. Acesso em: 26 set. 2021.

PACKT PUBLISHING LTD. **Top 10 Tools for Computational vision**. [S. l.], 5 abr. 2018. Disponível em: <https://hub.packtpub.com/top-10-computer-vision-tools/>. Acesso em: 4 ago. 2021.

PEDRINI, Hélio; SCHWARTZ, Willian Robson. **Análise De Imagens Digitais: Princípios, Algoritmos e Aplicações**. 1. ed. São Paulo: Cengage Learning, 2007. 528 p. v. 1. ISBN 8522105952.

PEREIRA, Fábio. **Tecnologia ARM: Microcontroladores de 32 bits**. 1. ed. [S. l.]: Érica,

2009. 448 p. ISBN 8536501707.

Raspberry Pi Camera v1era Rev 1.3. [S. l.], 2021. Disponível em: <https://circuit.rocks/raspberry-pi-camera.html>. Acesso em: 18 ago. 2021.

RASPBERRY PI FOUNDATION. **BCM2711.** [S. l.], 2021. Disponível em: <https://www.raspberrypi.org/documentation/hardware/raspberrypi/bcm2711/>. Acesso em: 3 ago. 2021.

RASPBERRY PI FOUNDATION. **BCM2837.** [S. l.], 2021. Disponível em: <https://www.raspberrypi.org/documentation/hardware/raspberrypi/bcm2837/>. Acesso em: 3 ago. 2021.

RASPBERRY PI FOUNDATION. **Camera Module.** [S. l.], 2021. Disponível em: <https://www.raspberrypi.org/documentation/hardware/camera/>. Acesso em: 3 ago. 2021.

RASPBERRY PI FOUNDATION. **Raspberry Pi 3 Model B.** [S. l.], 2021. Disponível em: <https://www.raspberrypi.org/products/raspberry-pi-3-model-b/>. Acesso em: 3 ago. 2021.

RASPBERRY PI FUNDATION. **Camera Module 2.** [S. l.], 2021. Disponível em: <https://www.raspberrypi.org/products/camera-module-v2/>. Acesso em: 7 ago. 2021.

RASPBERRY PI FUNDATION; ADAMS, James. **Raspberry Pi 3 Model B (Reduced Schematics).** 1.2. ed. [S. l.: s. n.], 2014. 1 p. Disponível em: https://www.raspberrypi.org/documentation/hardware/raspberrypi/schematics/rpi_SC_H_3b_1p2_reduced.pdf. Acesso em: 7 ago. 2021.

RESNICK, Robert; WALKER, Jearl; HALLIDAY, David. **Fundamentos de Física: Óptica e Física Moderna.** 10. ed. Rio de Janeiro: LTC, 2016. 448 p. v. 4. ISBN 9788521630388.

SISTEMA de Visão Artificial para verificação de nível e presença de rolha em garrafas de vinho. [S. l.], [2020?]. Disponível em: <http://www.serviautomation.pt/portfolio-items/sistema-visao-nivel-rolha-vinho/>. Acesso em: 20 set. 2021.

STTALINGS, William. **Arquitetura e Organização de Computadores**. 8. ed. São Paulo: Pearson Pratices Hall, 2010. 643 p. ISBN 978-85-7605-564-8.

SZELISKI, Richard. **Computational vision: Algorithms and Applications**. 1. ed. Nova Iorque: Springer, 2010. 832 p. ISBN 1848829345.

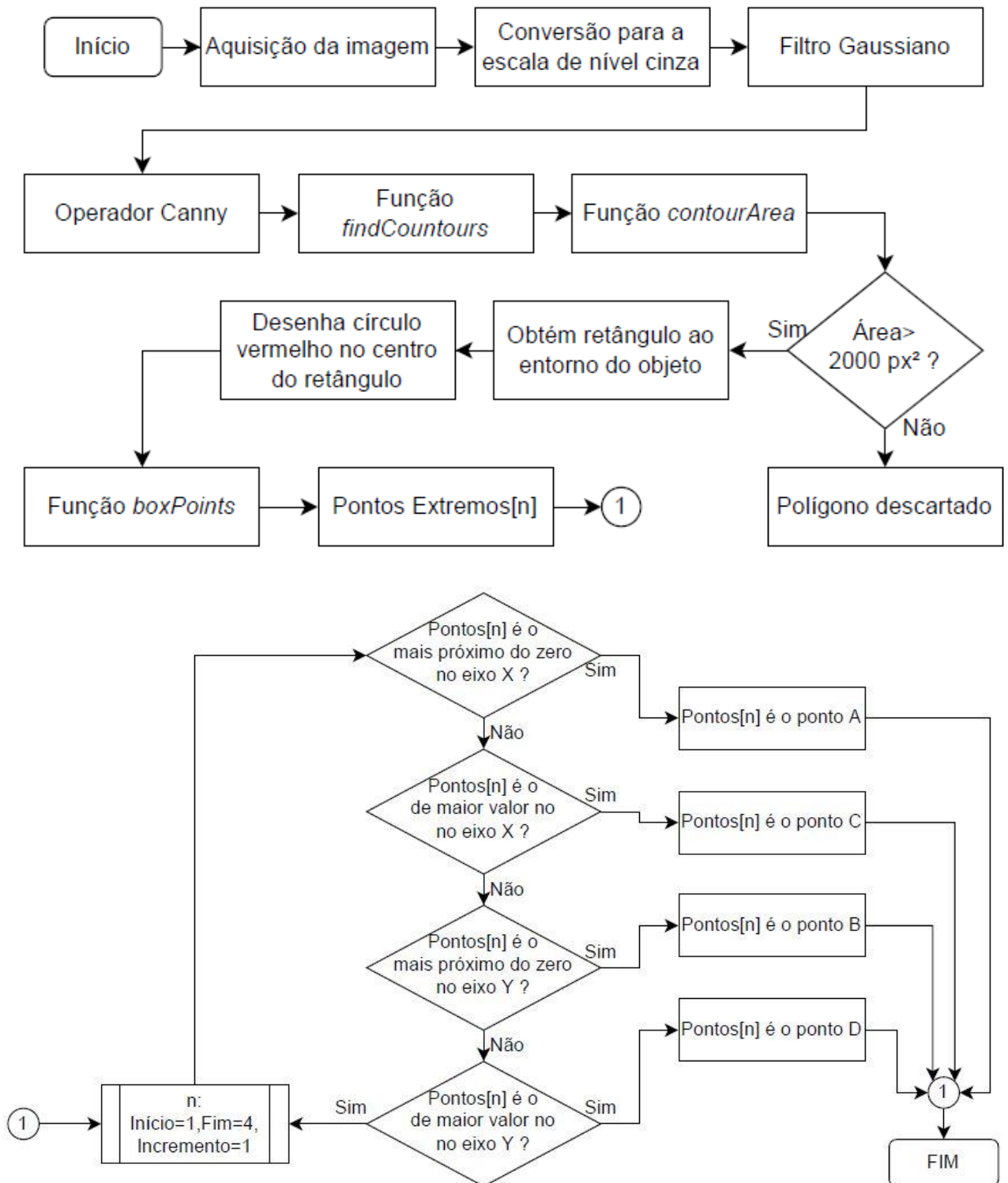
TECNOBLOG. **ARM revela ARMv9, nova arquitetura de chips mais rápida e segura**. [S. l.], 31 mar. 2021. Disponível em: <https://tecnoblog.net/427211/arm-nova-arquitetura-chips-armv9-mais-rapida-segura/>. Acesso em: 5 ago. 2021.

Universidade de São Paulo, 2002.

UPTON, Eben; HALFACREE, Gareth. **Raspberry Pi: Guia do Usuário**. 4. ed. Rio de Janeiro: Alta Books, 2017. 288 p. ISBN 8550802166.

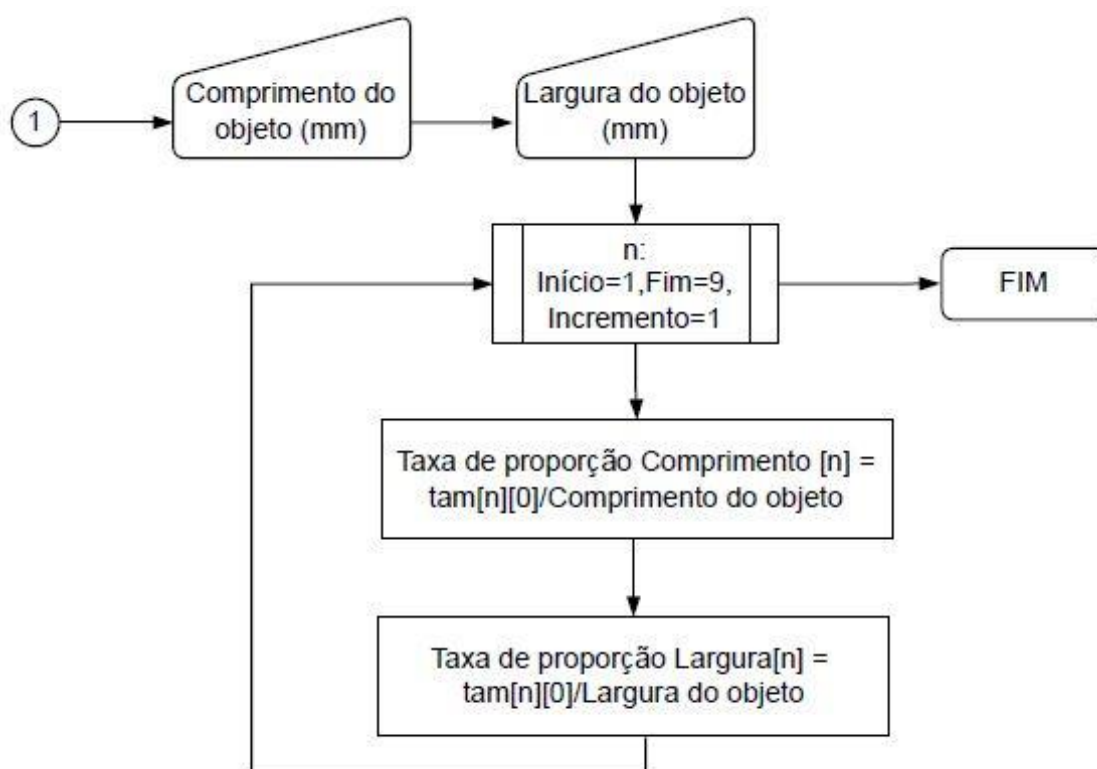
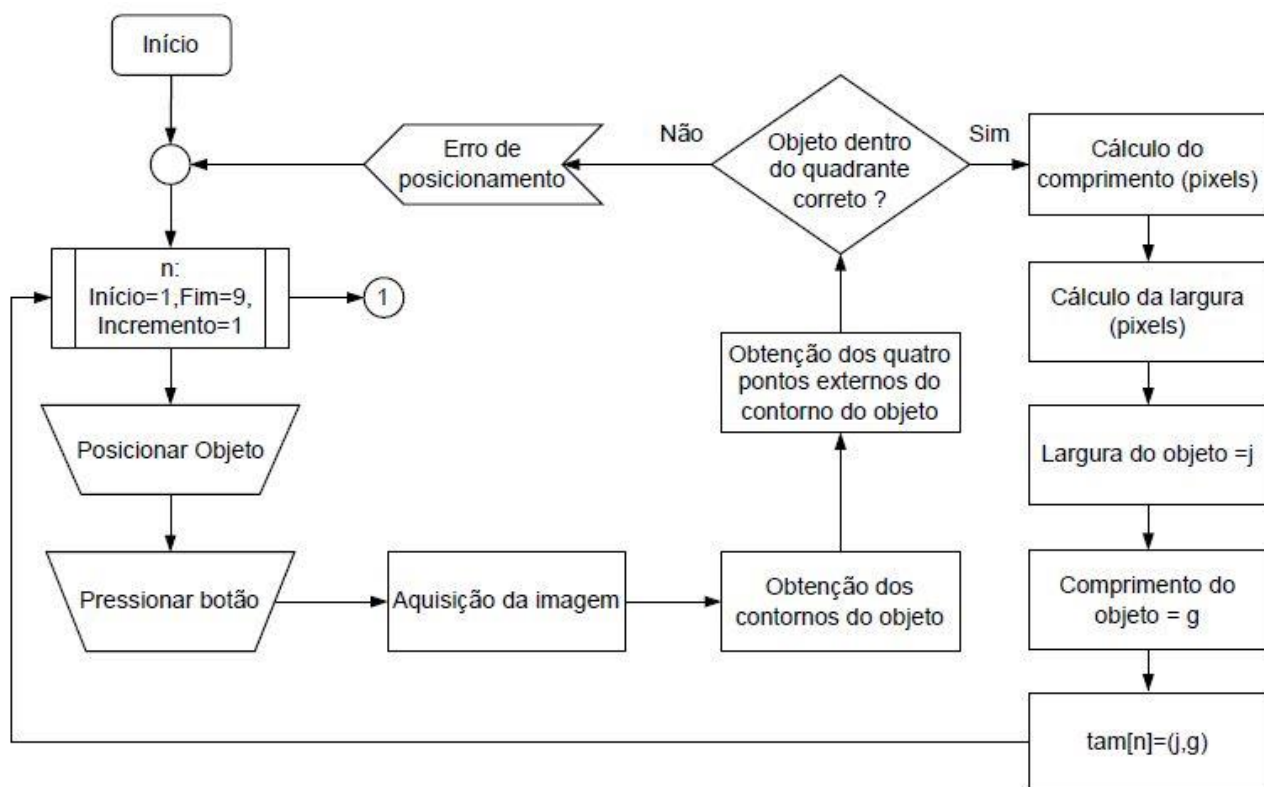
APÊNDICE 1

Fluxograma da Função de Aquisição dos Pontos Extremos do Objeto



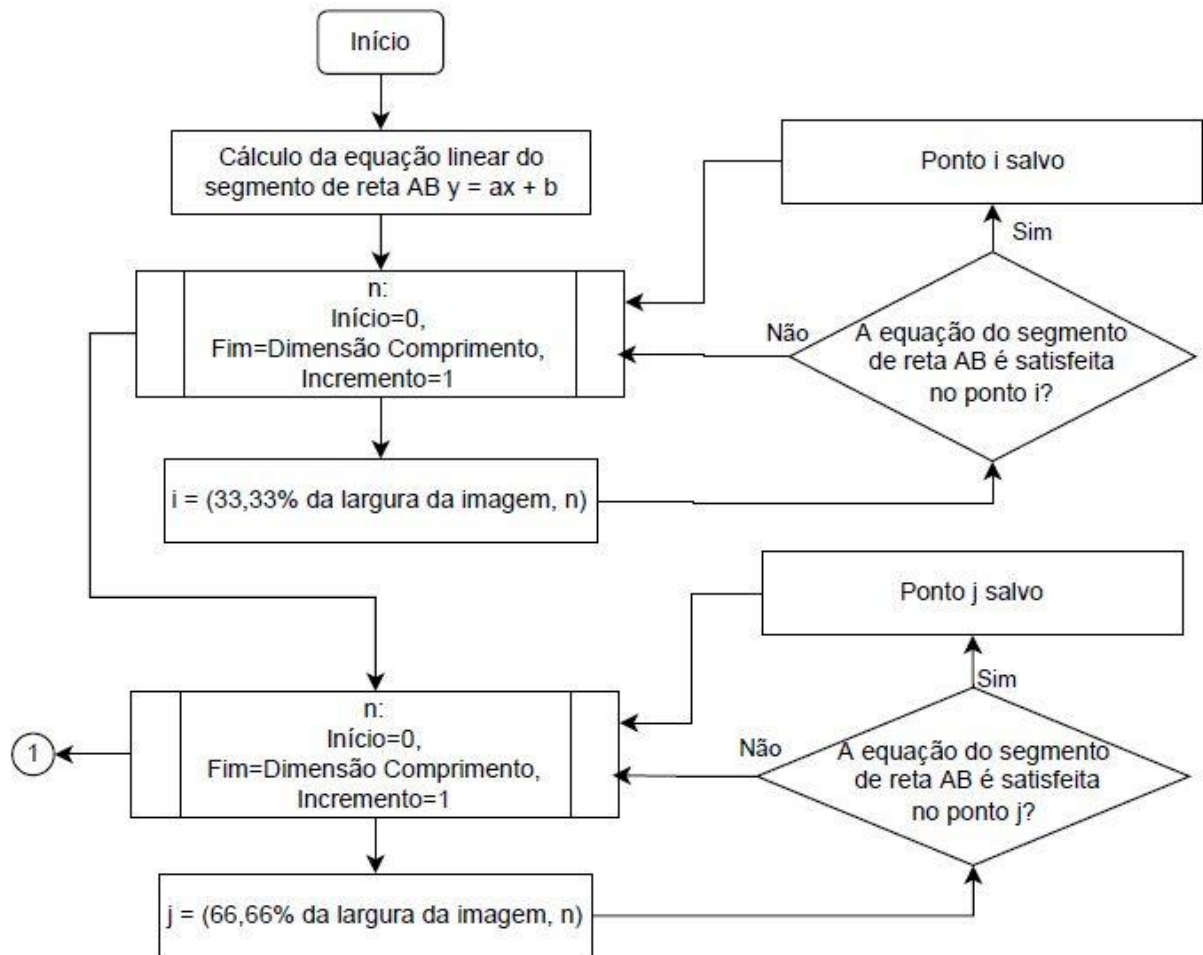
APÊNDICE 2

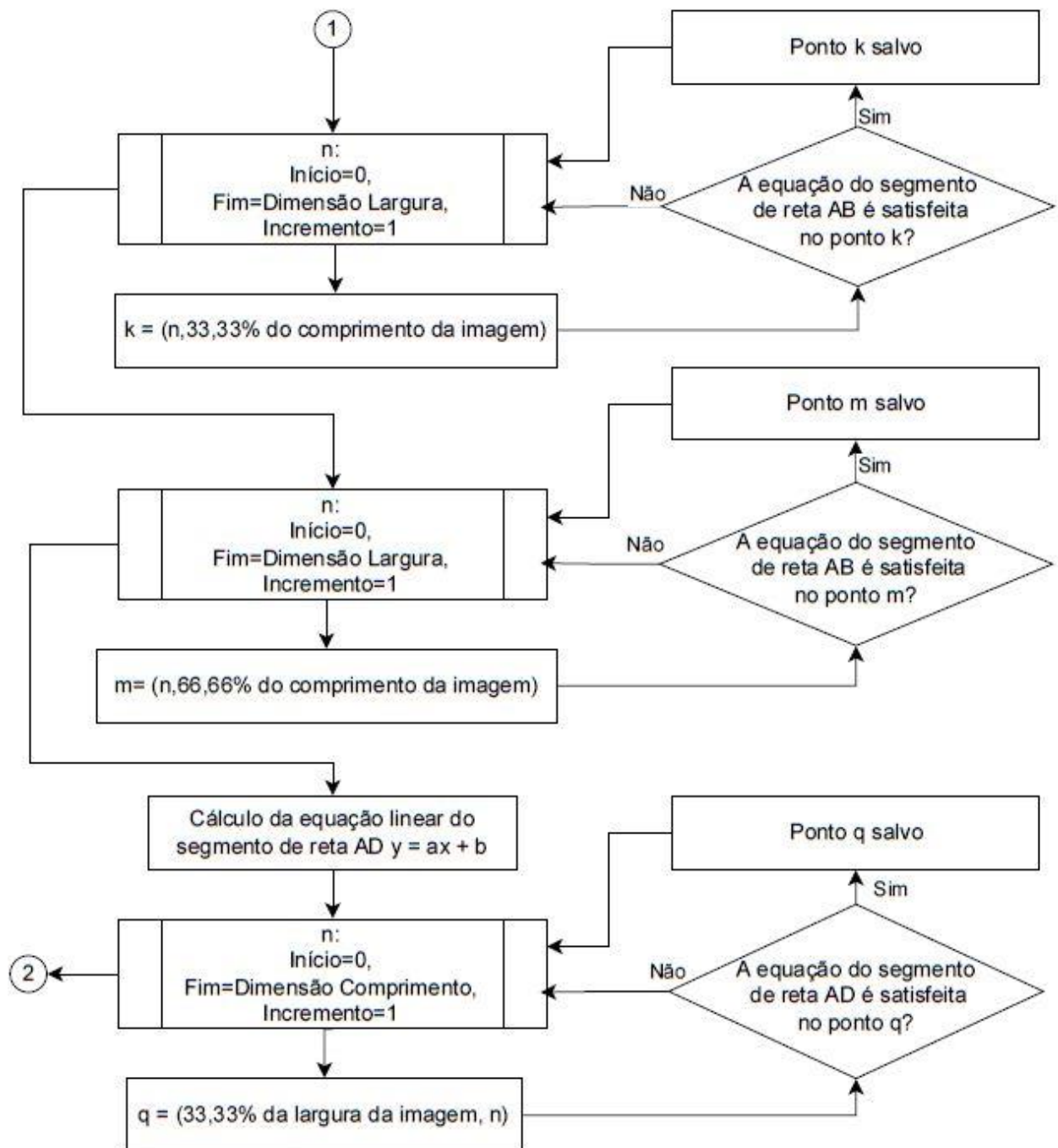
Fluxograma da Função de Calibração do Pixel

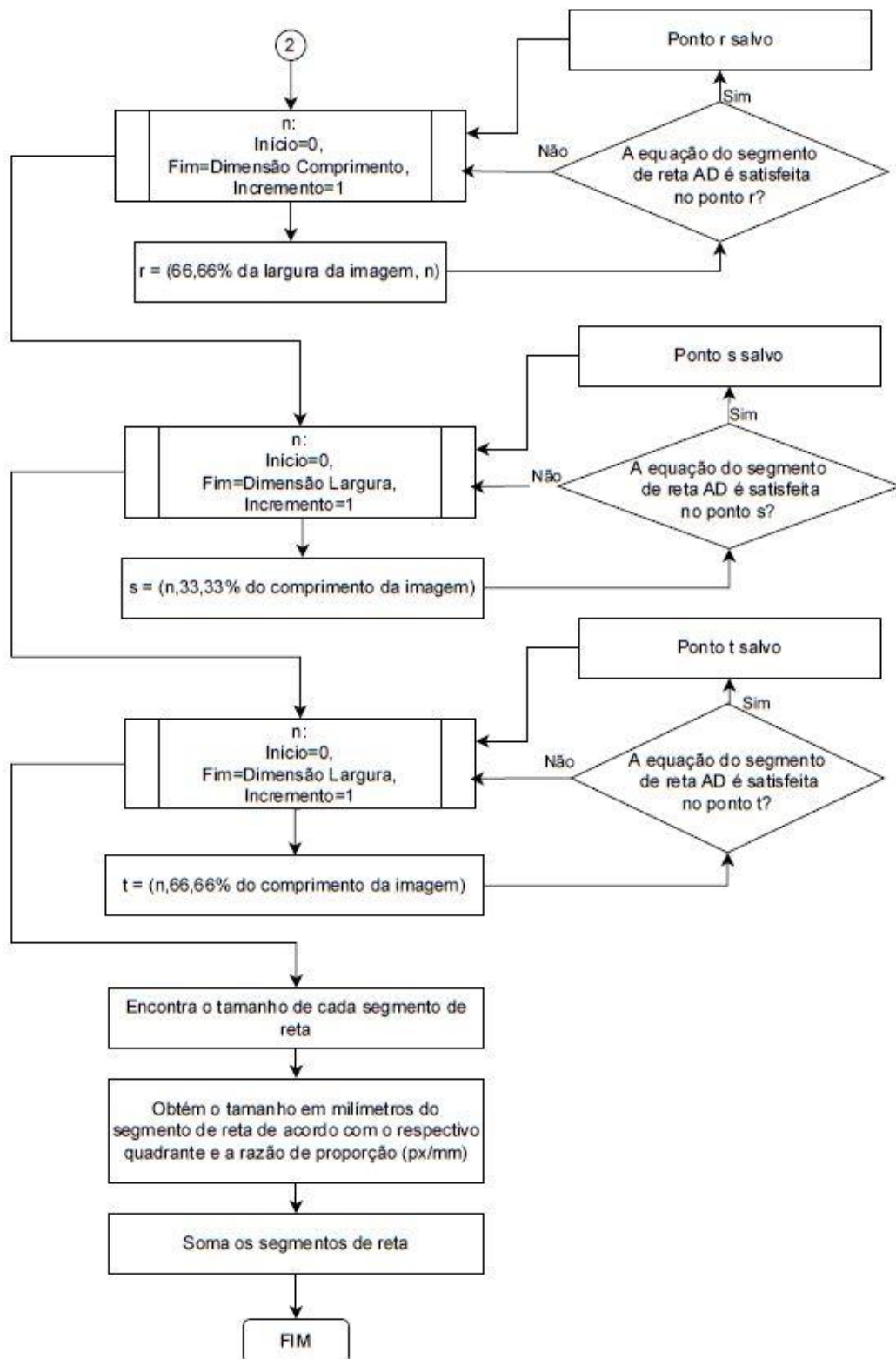


APÊNDICE 3

Fluxograma da Função de Separação dos Segmentos de Reta







APÊNDICE 4

Repositório da aplicação de medição bidimensional de objetos

https://github.com/MuriloSilveira/embedded_vision