Bárbara Grazielle Firmino de Araújo

Sistema de Visão de Máquina para Detecção e Localização Automática de Peças Utilizando Raspberry Pi

João Pessoa - PB

Março de 2019

Bárbara Grazielle Firmino de Araújo

Sistema de Visão de Máquina para Detecção e Localização Automática de Peças Utilizando Raspberry Pi

Dissertação de Mestrado submetida ao Programa de Pós-Graduação em Engenharia Elétrica do Instituto Federal de Educação, Ciência e Tecnologia da Paraíba, como requisito necessário à obtenção do grau de Mestre em Ciências no Domínio da Engenharia Elétrica. Área de Concentração: Processamento de Sinais.

Orientador: Prof. Dr. Carlos Danilo Miranda Regis

Coorientador: Prof. Dr. Rafael Franklin Alves Silva

João Pessoa - PB, Março de 2019

Bárbara Grazielle F. de Araújo - bgfaraujo@gmail.com

Dados Internacionais de Catalogação na Publicação – CIP
Biblioteca Nilo Peçanha – IFPB, *campus* João Pessoa

A663s Araújo, Bárbara Grazielle Firmino de.
Sistema de visão de máquina para detecção e localização automática de peças utilizando Raspberry Pi / Bárbara Grazielle Firmino de Araújo. – 2019.
102 f. : il.

Dissertação (Mestrado em Engenharia Elétrica) – Instituto Federal de Educação, Ciência e Tecnologia da Paraíba – IFPB / Coordenação de Pós-Graduação em Engenharia Elétrica, 2019.
Orientador: Prof. Dr. Carlos Danilo Miranda Regis.
Coorientador: Prof. Dr. Rafael Franklin Alves Silva Silva.

1. Robô. 2. Raspberry Pi. 3. Visão computacional. 4. Engenharia elétrica. I. Título.

CDU 621.865.8

Bibliotecária responsável Taize Araújo da Silva – CRB15/536

Sistema de Visão de Máquina para Detecção e Localização Automática de Peças Utilizando Raspberry Pi

Dissertação de Mestrado submetida ao Programa de Pós-Graduação em Engenharia Elétrica do Instituto Federal de Educação, Ciência e Tecnologia da Paraíba, como requisito necessário à obtenção do grau de Mestre em Ciências no Domínio da Engenharia Elétrica.

Dissertação de Mestrado defendida e aprovada em 26 / 03 / 19 .

BANCA EXAMINADORA

Carlos Danilo Miranda Regis, Dr. IFPB (Orientador)

Rafael Franklin Alves Silva Silva (Coorientador)

Suzete Élida Nóbrega Correia , Dra. IFPB (Examinadora Interna)

Leonardo Vidal Batista, Dr. UFPB (Examinador Externo)

João Pessoa – PB
Março de 2019

Agradecimentos

Agradeço primeiramente ao apoio prestado pela minha família, especialmente minha mãe, Cláudia Firmino, e minha avó, Maria das Graças, mulheres incríveis que me ensinaram a valorizar o conhecimento e a educação.

Agradeço ao meu noivo, Luiz Guilherme, pelo apoio durante todo o período do mestrado.

À minha irmã, Roberta Firmino, pelos conselhos e palavras de conforto durante muitos momentos difíceis.

Ao meu orientador, coorientador e ao professor João Bosco que me auxiliaram na conclusão dessa etapa tão importante.

Aos meus amigos pelas conversas, descontrações e desabafos.

Ao IFPB campus João Pessoa, por proporcionar uma infraestrutura que permitiu o desenvolvimento deste trabalho, bem como, um corpo técnico e docente capacitado e proativo.

“Nada lhe pertence mais que seus sonhos.”

(Friedrich Nietzsche)

Resumo

Na perspectiva da indústria 4.0, a tomada de decisão está cada vez mais descentralizada devido à inserção de sistemas inteligentes no contexto industrial, que tornam viável o desenvolvimento de linhas de produção mais flexíveis. Os sistemas de visão de máquina (VM) são sistemas inteligentes compostos por câmeras, *hardwares* de processamento, periféricos de comunicação e podem ser classificados como sensores de alto nível, dado que são capazes de extrair informações complexas de um determinado cenário. Quando associados à robótica, a VM é capaz de auxiliar na captura automática de diferentes tipos de objetos por manipuladores robóticos, atividade conhecida como *pick and place*. Neste trabalho é implementado um sistema de VM dedicado à detecção e localização das coordenadas dos objetos cilíndricos posicionadas aleatoriamente sobre uma plataforma, em seguida o sistema embarcado envia as coordenadas ao controlador de um braço robótico industrial. A câmera utilizada está disposta acima da plataforma e captura uma imagem da vista superior dos objetos, a partir desta imagem, os objetos são detectados e seus centroides são localizados. No âmbito desta pesquisa, são utilizados apenas objetos cilíndricos de mesmas dimensões com cores distintas. A arquitetura do sistema de VM é baseada no uso de tecnologias acessíveis, por exemplo, os *hardwares* Raspberry Pi e PiCamera, assim como o *framework* OpenCV, que disponibiliza um conjunto de funcionalidades *open source* para o desenvolvimento de sistemas de Visão Computacional e Processamento Digital de Imagens. Distorções inerentes à lente da câmera digital foram corrigidas e a homografia entre o plano do robô e o da imagem foi estabelecida aplicando o método DLT *(Direct Linear Transform)*. Foram testados seis métodos de detecção diferentes e seus resultados são analisados e comparados. Além disso, foram realizados testes de captura, repetibilidade do sistema e erro. Os resultados experimentais indicam que um manipulador robótico equipado com o sistema de detecção e localização proposto possui uma taxa de erro máximo na coordenada de 2,65 mm que permite a captura automática das peças-alvo, no entanto ao usar imagem do tipo *RAW* esse erro cai para 0,157 mm.

Palavras-Chave: Robô, Raspberry Pi, Visão Computacional.

Abstract

In the view of industry 4.0, decision-making is increasingly decentralized due to the insertion of intelligent systems in the industrial context that make the development of more flexible production lines feasible. Machine vision systems (MVs) are intelligent systems composed of cameras, processing hardware, and peripheral devices, and can be classified as high-level sensors because they are capable of extracting complex information from scenario. When associated to robotics, the VM is able to assist in the automatic capture of different types of objects by robotic manipulators, an activity known as pick and place. In this work is implemented a VM system dedicated to the detection and location of coordinates of cylindrical objects placed randomly on a platform, then the embedded system sends coordinates to the controller of an industrial robot arm. The camera used is positioned above the platform and captures an image of the top view of the objects, from this image, the objects are detected and their centroids are located. In the scope of this research, only cylindrical objects of the same dimensions with distinct colors are used. The architecture of the VM system, based on the use of accessible technologies, for example, the Raspberry Pi and PiCamera hardware, as well as the OpenCV framework, which provides a set of open source functionalities for the development of systems of Computational Vision and Digital Image Processing. Distortions inherent in the digital camera lens were corrected and homography between the theft plane and that of the image was established using the Direct Linear Transform (DLT) method. Six different detection methods were tested and their results were analyzed and compared. In addition, capture tests, system repeatability and error were performed. The experimental results indicate that a robot manipulator equipped with the detection system and localized to the proposed one has a maximum error rate 2.65 mm that allows automatic capture of the pieces, when using image of RAW type this error drops to 0.157mm.

Keywords: Robot, Raspberry Pi, Computer Vision.

Lista de Figuras

Lista de Tabelas

Lista de Abreviaturas e Siglas

AHE *Adaptive Histogram Equalization* - Equalização de Histograma Adaptativo

BSD *Berkley Software Distribution* - Distribuição de Software Berkley

CCD *Charge-coupled Device* - Dispositivo de Carga Acoplada

CLAHE *Contrast Limited Adaptive Histogram Equalization* - Equalização Adaptativa de Histograma com Limitaçao de Contraste

CMOS *Complementary Metal-oxide Semiconductor* - Semicondutor Complementar de Oxido de Metal

CNI Confederação Nacional da Indústria

CSI *Camera Serial Interface* - Interface Serial da Câmera

DLT *Direct Linear Transform* - Transformação Linear Direta

DoG *Difference of Gaussians* - Diferença de Gaussianas

DSI *Display Serial Interface* - Interface Serial do Display

DSP *Digital Signal Processor* - Processador Digital de Sinais

FPR *False Positive Rate* - Taxa de Falsos-positivos

FPS *Frames-per-Second* - Frames por Segundo

HSV *Hue Saturation and Value*

HSB *Hue Saturation and Brightness*

IoT *Internet of Things* - Internet das Coisas

IPU *Image Processing Units* - Unidade de Processamento de Imagens

L-CNN *Lightweigt Concolutional Neural Network* - Rede Neural Convolucional Leve

LIC Limite Inferior de Controle

LSC Limite Superior de Controle

ML *Machine Learnig* - Aprendizagem de Máquina

MMC Máquina de Medição por Coordenadas

MPS *Modular Process System* - Sistema de Processos Mudular

RGB *Red Green Blue* - Vermelho Verde Azul

RPi Raspberry Pi

PDI Processamento Digital de Imagens

PNG *Potable Network Graphics*

SCC Sistema de Coordenadas da Câmera

SCI Sistema de Coordenadas da Imagem

SCM Sistema de Coordenadas de Mundo

SCP Sistema de Coordenadas em Pixels

SCADA Supervisory Control and Data Acquisition - Sistemas de Supervisão e Aquisição de Dados

SBC *Single Board Computers* - Computador de Placa Única

SSD *Single Shot Detector*

VC Visão Computacional

VM Visão de Máquina

VVC Valor Verdadeiro Convencional

Sumário

1 Introdução

Nos últimos anos, a quarta revolução industrial, também conhecida como Indústria 4.0, surge com novos conceitos acerca das relações de produtividade e manufatura (Sishi and Telukdarie, 2017). A inserção de tecnologias como IoT (*Internet of Things*), algoritmos de aprendizagem e Visão Computacional (VC) na indústria permitiu o desenvolvimento de robôs inteligentes capazes de tomar decisões de forma autônoma, elevando os níveis de flexibilidade e adaptabilidade dos processos produtivos. As fábricas inteligentes (*smart factories*), projetadas sob essa nova perspectiva, são mais competitivas no mercado, pois adotam um modelo de produção de alta variedade que atende a demanda personalizada de cada consumidor (Rüßmann et al., 2015).

No modelo de produção tradicional, cada célula composta por máquinas, sensores e sistemas está conectada a uma unidade central de processamento, no entanto, as indústrias modernas vêm adotando um novo modelo baseado no controle de processo distribuído, caracterizado por um fluxo de produção descentralizado (Thoben et al., 2017). Isto é possível, pois os dispositivos são enriquecidos com unidades embarcadas de processamento que permitem a análise local de informações por sistemas inteligentes, flexibilizando a manufatura de produtos com alta qualidade e custos reduzidos (Hermann et al., 2015).

A VC é um exemplo de sistema inteligente composto por câmera, *hardware* e *software* capaz de extrair informações de alto nível de uma ou mais imagens (Shah, 1997). Por definição, a VC é a área de conhecimento que busca realizar o processo de modelagem e replicação da visão humana por meio de algoritmos destinados a reconstrução e compreensão de uma cena 3D a partir de imagens 2D, aplicando técnicas para aquisição, análise e processamento de imagens (Szeliski, 2010) ou a partir do treinamento de algoritmos de Aprendizagem de Máquina (*Machine Learning*-ML), acelerando e refinando os processos de classificação, detecção e localização de objetos.

Quando os sistemas de VC são empregados com maior ênfase em aplicações industriais, estes podem ser denominados de sistemas de Visão de Máquina (VM), embora não existam restrições definidas quanto ao uso das duas terminologias. Os sistemas de VM estão presentes no contexto industrial em tarefas como inspeção de peças pintadas ou soldadas, no controle de qualidade, no trabalho colaborativo entre robôs e seres humanos e na detecção automática de objetos em posições randômicas, auxiliando os manipuladores robóticos industriais na captura automática de

peças (Pérez et al., 2016).

A detecção e localização de objetos é um campo da VC que busca extrair informações sobre um objeto-alvo, identificando sua localização na cena. Nesta problemática, o principal desafio consiste em prover uma alta precisão e uma performance robusta, independentemente das condições de luminosidade externa, do fundo da imagem, geometria do objeto e qualidade da imagem capturada. No entanto, em aplicações reais, a iluminação externa não controlada provocam ruídos na imagem, consequentemente comprometendo o desempenho dos algoritmos de detecção (Jenamani et al., 2017).

Grande parte das aplicações que utilizam VM ainda são processadas por *desktops*, contudo, este dispositivo é limitado em portabilidade, peso e tamanho, além de apresentarem alto consumo quando comparados com os sistemas embarcados. A utilização de sistemas embarcados associados à VM cresceu significativamente nos últimos anos em virtude do avanço tecnológico que possibilitou a miniaturização dos computadores em dispositivos conhecidos como SBCs (do inglês, *Single Board Computers*), baseado na arquitetura ARM e em sistemas operacionais derivados do Linux. Um dos SBCs mais difundidos em aplicações de *Embedded Vision* é o Raspberry Pi (RPi). Equipado com uma GPU (*Graphics Processing Unit*) integrada, o RPi é capaz de processar algoritmos de VC em tempo-real (Horak and Zalud, 2016).

Embora inicialmente desenvolvido para aplicações educacionais, o RPi está sendo cada vez mais imerso no cenário industrial, por exemplo, no monitoramento remoto de dispositivos utilizando o conceito de IoT (Kadiyala et al., 2017), na computação robótica em nuvem Krishna et al. (2016) e no controle espacial de robôs industriais utilizando VC (Szabó and Gontean, 2016). Alinhada a essa nova perspectiva, diversas empresas têm se voltado ao RPi como solução para aplicações industriais, tendo como exemplo a Modbus Foundation que desenvolveu bibliotecas específicas para incluir o RPi no protocolo de comunicação serial Modbus, bastante difundido entre dispositivos presentes na indústria. Outras empresas oferecem estruturas para abarcar o RPi, tornando-o robusto às intempéries inerentes ao ambiente industrial.

O presente trabalho propõe um sistema de VM para detecção de objetos, robusto à variação das condições de luminosidade do meio e capaz de auxiliar os manipuladores robóticos industriais na captura de peças de diferentes formas e cores. A saída do sistema deve expressar a posição dos objetos alvo de captura, levando em consideração aspectos como a presença de distorção na

imagem, erros de perspectiva e a conversão do sistema de coordenadas da imagem (dada em *pixel*) para o sistema de coordenadas do robô (dado em milímetros). O dispositivo embarcado RPi é encarregado de realizar todo o processamento da imagem desde a captura até a etapa de comunicação com o manipulador robótico industrial. Para validação do trabalho são feitos experimentos com a manipulação de peças identificadas pelo sistema de VM, bem como testes de repetibilidade e verificação do erro entre o sistema proposto e uma máquina de medição por coordenadas (MMC).

1.1 Motivação

O desenvolvimento da Industria 4.0 no Brasil permeia diversos desafios, sendo o maior deles, o investimento em equipamentos que incorporem novas tecnologias digitais ao longo da cadeia produtiva, permitindo a integração desde o momento do pedido de compra à distribuição do produto final. A Confederação Nacional da Indústria (CNI) realizou uma pesquisa com empresários de indústrias de pequeno, médio e grande porte em que avaliava aspectos acerca da adoção de tecnologias digitais.

A pesquis concluiu que 42% das empresas desconhecem a importância das tecnologias digitais para a competitividade da indústria e mais da metade delas (52%) não utilizam nenhuma tecnologia digital de uma lista com 10 opções, dentre estas, a automação digital com linhas flexíveis, sistemas de *clound computing*, IoT e até mesmo monitoramento com sistemas SCADA (CNI, 2016).

O desconhecimento é significativamente maior nas empresas de pequeno porte devido a fatores como a falta de recursos financeiros para investimento em novas tecnologias e na capacitação dos funcionários. A falta de estratégias para a disseminação de informações acerca da digitalização industrial gera dificuldades na identificação das tecnologias mais adequadas para cada setor.

Os sistemas de VM fazem parte das tecnologias que possibilitam a flexibilização das linhas de produção automatizadas, sobretudo quando estão associados aos manipuladores robóticos industriais, no entanto, grande parte dos sistemas de VM comercializáveis, possuem uma arquitetura fechada e muitas vezes são dependentes de *softwares* proprietários, o que torna o preço desses produtos pouco atrativo para indústrias de pequeno e médio porte.

Diante do cenário presente na indústria brasileira, a principal motivação desta pesquisa é desenvolver um sistema de VC de baixo custo, capaz de se adequar ao contexto industrial. O algoritmo proposto é baseado em bibliotecas *open source* como por exemplo a OpenCV (*Open*

Source Computer Vision Library), liberada sob a licença BSD (*Berkeley Software Distribution*), a qual, é gratuita tanto para uso acadêmico quanto comercial (team, 2018).

Para tornar o algoritmo flexível à iluminação externa não controlada, é proposto um algoritmo robusto à iluminação não-uniforme, de modo que seja possível eliminar aparatos extras como sistemas de iluminação ou equipamentos de proteção para conter os efeitos da iluminação externa. Neste sentido, optou-se pelo uso de um dispositivo embarcado competitivo, capaz de realizar aplicações de VM em tempo real (Kumar et al., 2018), mas que também tivesse um custo acessível.

1.2 Objetivo Geral

Desenvolver um algoritmo integrado ao sistema de VM embarcado na RPi para detecção automática de objetos geométricos dispostos aleatoriamente sobre a plataforma de trabalho. O algoritmo deve ser capaz de detectar objetos cilíndricos com superfície nas cores preto, amarelo, vermelho, verde, azul claro e escuro. Além da detecção dos objetos na cena, o algoritmo deve informar a localização no sistema de coordenadas mundo - nesse caso, em milímetros- levando em consideração as variações de iluminação do meio externo. Por fim, o dispositivo embarcado deve comunicar-se com um manipulador robótico industrial informando a localização do objeto-alvo.

1.2.1 Objetivos Específicos

Com a finalidade de atingir o objetivo geral, são propostos os seguintes objetivos específicos:

- Empregar um algoritmo direcionado a realização da calibração da câmera, estimando os parâmetros intrínsecos e os coeficientes de distorção;
- Aplicar a correção da distorção radial e tangencial na imagem;
- Usar a transformação de perspectiva;
- Comparar técnicas de segmentação e segmentação por cor para detectar a superfície dos cilindros.
- Estimar a matriz de homografia responsável pela correspondência entre o sistema de coordenadas da imagem e o sistema de coordenadas do robô;

- Aplicar métricas para avaliar a eficiência do algoritmo proposto no sistema embarcado;
- Realizar testes para medir o erro e repetibilidade do sistema proposto.

1.3 Organização do Trabalho

O presente trabalho está estruturado da seguinte forma: no Capítulo 2 é apresentada a revisão bibliográfica, incluindo o estado da arte. No Capítulo 3, é apresentada a fundamentação teórica, que contém os embasamentos necessários para a compreensão dos capítulos seguintes. No Capítulo 4, é apresentado o método adotado na calibração, eliminação da distorção, correção de perspectiva e uma breve discussão acerca das possíveis técnicas para detecção dos objetos. No Capítulo 5, são apesentados os resultados e discussões acerca da eficiência dos algoritmos e da captura de peças e por fim, no Capitulo 6, são apresentadas as conclusões finais e projeções para trabalhos futuros.

2 Fundamentação Teórica

Neste Capítulo é abordada uma visão geral sobre o processo de calibração da câmera baseado no método proposto por Zhang (2000). Inicialmente são apresentados conceitos introdutórios sobre algumas propriedades da geometria projetiva, em seguida são discutidos os processos matemáticos de calibração da câmera e eliminação de distorção. Também são discutidos nesta seção o método DLT (do inglês, *Direct Linear Transform*) responsável por estimar os parâmetros da matriz de homografia utilizada na conversão dos sistemas de coordenadas da imagem para o sistema de coordenada de mundo como também na correção das distorções de perspectiva. Por fim, são apresentadas as técnicas de detecção de objeto até então utilizadas neste trabalho.

2.1 Geometria Projetiva e Coordenadas Homogêneas

A geometria projetiva é o campo da matemática que representa o processo de formação da imagem a partir da projeção em perspectiva de uma cena no espaço tridimensional em uma imagem bidimensional. Esta projeção é realizada mediante a definição de um centro de projeções, de modo que, cada raio que cruza este centro de projeções intersecta o plano projetivo em um único ponto correspondente (Richter-Gebert, 2011).

Na geometria projetiva os objetos diminuem de tamanho à medida que se afastam do centro de projeções, essa propriedade faz com que pontos situados em retas paralelas pareçam convergir para um ponto de interseção entre essas retas, portanto, o paralelismo, a distância, os ângulos e as formas não são preservados na geometria projetiva (Richter-Gebert, 2011).

A geometria euclidiana clássica descreve os elementos contidos no espaço físico 3D mediante axiomas e postulados, no entanto, essa representação não é suficiente quando os objetos são visualizados por um ponto de vista, neste caso, é preciso incluir o fator perspectiva, que vai de encontro ao quinto postulado da geometria euclidiana referente ao paralelismo entre retas. Para a geometria clássica duas retas podem se intersectar em um único ponto, entretanto, existem infinitos pares de retas paralelas que por definição só se tocariam no infinito (Hartshorne, 2002).

Um ponto no espaço euclidiano pode ser representado por um par ordenado de números reais $(x, y) \in \mathbb{R}^2$. Esta associação é realizada por um sistema de coordenadas cartesianas, em que duas retas perpendiculares são usadas como eixos. Na geometria projetiva são utilizadas

coordenadas homogêneas no lugar do sistema de coordenadas cartesianas devido a vantagem da representação dos pontos e retas no infinito por uma representação finita.

Se na geometria euclidiana um ponto é definido por um vetor bidimensional $\mathbf{x} = [x\ y]^{\mathrm{T}}$, na geometria projetiva este mesmo ponto é representado em coordenadas homogêneas $\tilde{\mathrm{x}} = [x\ y\ w]^{\mathrm{T}}$, acrescentando uma dimensão. A conversão do ponto $\mathbf{x}$ do espaço euclidiano $\mathbb{R}^2$ para o espaço projetivo $\mathbb{P}^2$ é dado por $\mathbf{x} = \left[\frac{x}{w}\ \frac{y}{w}\right]^{\mathrm{T}}$, em que w é denominado peso ou fator de escala para $w \neq 0$. Desse modo, o ponto cartesiano $\mathbf{x}$ corresponde à uma classe de vetores tridimensionais equivalentes $[wx, yw, w]$, incluindo o caso particular $[x, y, 1]$. Os pontos no infinito em $\mathbb{P}^2$ possuem a forma $[x\ y\ 0]^{\mathrm{T}}$, em que $x \neq 0$ ou $y \neq 0$ e são denominados de pontos impróprios (Casse, 2006).

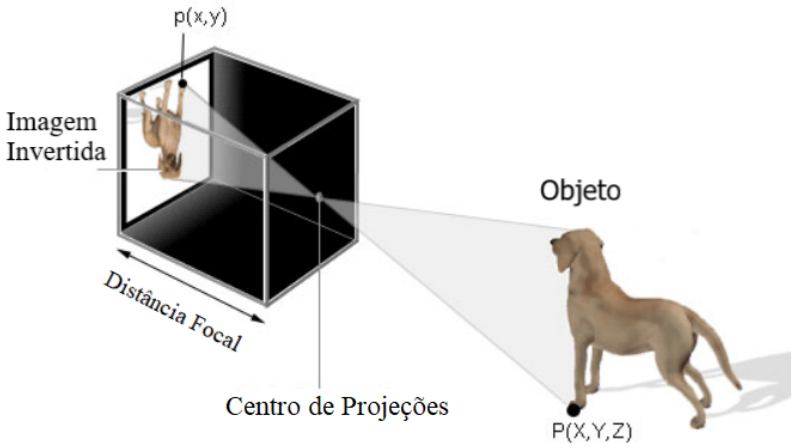
2.2 Calibração da Câmera

A calibração da câmera é uma etapa essencial para aplicações de VC que necessitam de informações do sistema métrico. Esse processo busca determinar os parâmetros intrínsecos à geometria de formação de imagens, tais como distância focal, ponto central de projeção, tamanho real de cada *pixel*, distorções incorporadas pela física das lentes e a inclinação entre os eixos de projeções. A calibração também considera os parâmetros extrínsecos ao sistema que determinam a translação e rotação da câmera em relação a um determinado sistema de coordenadas de mundo (Chen et al., 2013).

Para compreender o processo de calibração da câmera, é preciso entender como ocorre formação de uma imagem na câmera. Este fenômeno é conduzido com base na geometria projetiva que descreve os objetos a partir de um ponto de projeção, ou seja, considerando os efeitos da perspectiva. O mapeamento do espaço 3D para o plano 2D é realizado a partir da projeção perspectiva associada ao modelo de câmera *pinhole*.

O modelo matemático câmera *pinhole* é um dos mais utilizados para modelar o funcionamento de uma câmera, pois é matematicamente conveniente e apesar de sua simplicidade, é capaz de fornecer uma aproximação aceitável do processo de formação da imagem. Seu princípio é baseado na câmara escura de orifício, em que os raios de luz provenientes de um objeto, entram pelo furo contido no centro de um dos lados da câmara e formam uma projeção 2D invertida deste objeto de acordo com o que é ilustrado na Figura 1.

Figura 1: Modelo de Câmera Pinhole.



Fonte: Adaptado de Anagram Engeneering[1]

Assume-se que o orifício presente no modelo câmera *pinhole* é reduzido a um único ponto, desta forma é garantido que o ponto $P(X, Y, Z)$ tenha como projeção o ponto $p(x, y)$ no plano da imagem. também conhecido como centro óptico, sendo assim, cada raio de luz que passa por este orifício corresponde a um único ponto no plano da imagem, no entanto, esta simplificação é fisicamente impossível.

Na câmera pinhole cada ponto na imagem é uma projeção de um cone de raios com um pequeno ângulo. O diâmetro do furo influencia em aspectos da imagem como nitidez e luminosidade, uma vez que quanto menor for o diâmetro melhor será a nitidez da imagem, em contrapartida, um furo pequeno compromete a luminosidade.

Para atenuar este efeito, são inseridas lentes para aumentar a intensidade de luz na imagem pela focalização de raios de luz da cena. As lentes, por sua vez, incorporam à geometria da câmera distorções radiais e tangenciais que podem ser modeladas e corrigidas matematicamente.

2.2.1 Transformações da Câmera

O processo de calibração da câmera tem como objetivo relacionar pontos tridimensionais com suas respectivas projeções bidimensionais no plano da imagem. Para realizar esta projeção é

[1] Disponível em: <https://www.anagram.at/diplomarbeit/the-pinhole-camera/>. Acesso em fev. 2017.

necessário representar os pontos em diferentes sistemas de coordenadas:

- **Sistema de Coordenadas do Mundo (SCM)**: Sistema de Coordenadas tridimensionais (métrico) formado pelos eixos X_w, Y_w e Z_w com origem no ponto O_w e escolhido de forma conveniente a definir as coordenadas de cada ponto da cena. As coordenadas de um ponto P_w neste sistema serão denotadas como $[X_w\ Y_w\ Z_w]^{\mathrm{T}}$.

- **Sistema de Coordenadas da Câmera (SCC)**: Sistema de Coordenadas tridimensionais (métrico) formado pelos eixos X_c, Y_c, e Z_c, sendo Z_c o eixo principal ou eixo óptico com origem no centro de projeções da câmera O_c. A distância entre a origem deste sistema e o plano da imagem é conhecida como distância focal f. Para este Sistema de Coordenadas denomina-se $[X_c\ Y_c\ Z_c]^{\mathrm{T}}$ como as coordenadas do ponto P_c presente neste sistema.

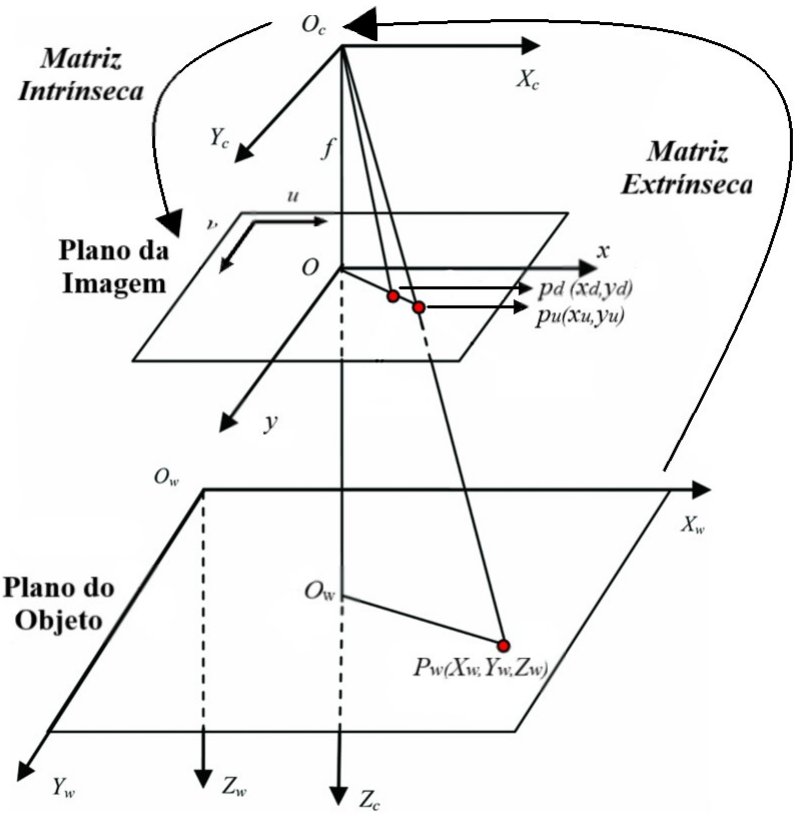
- **Sistema de Coordenadas da Imagem (SCI)**: Sistema de Coordenadas bidimensional (métrico) situado no plano da imagem, também conhecido como plano de projeção. A origem deste sistema é denominado de ponto principal dado por O com coordenadas $[c_x\ c_y]^{\mathrm{T}}$ e corresponde ao ponto em que o eixo principal intersecta o plano da imagem. Um ponto pertencente ao plano da imagem é denotado por p com coordenadas $[x\ y]^{\mathrm{T}}$ quando é utilizado o modelo de câmera *Pinhole*. Entretanto, na modelagem matemática da distorção, define-se $[x_d\ y_d]^T$ como coordenadas dos pontos com distorção e $[x_u, y_u]^{\mathrm{T}}$ coordenadas dos pontos com distorção corrigida.

- **Sistema de Coordenadas em *Pixels* (SCP)**: Sistema de Coordenadas bidimensional formado pelos eixos u e v paralelos aos eixos x e y do SCC. Os pontos desse sistema de coordenadas são expressos em *pixel*. Usualmente, adota-se como centro deste sistema o canto superior esquerdo da imagem.

O fluxo de transformações entre os sistemas de coordenadas é ilustrado na Figura 2. Inicialmente é calculada a matriz extrínseca que determina a posição da câmera em relação a um certo SCM. Essa transformação faz com que um ponto dado em um SCM passe a ser representado no SCC. Em seguida, calcula-se a matriz intrínseca, que descreve as características ópticas e geométricas da câmera, bem como as distorções inerentes ao sistema de lentes, dessa forma, é

possível descobrir a relação entre o SCC com SCI, conhecendo a dimensão de um *pixel* no sistema métrico é possível realizar a transformação do SCC para o SCI.

Figura 2: Transformações entre Sistemas de Coordenadas durante o processo de Calibração da Câmera.



Fonte: Adaptado de Li and Zhang (2011).

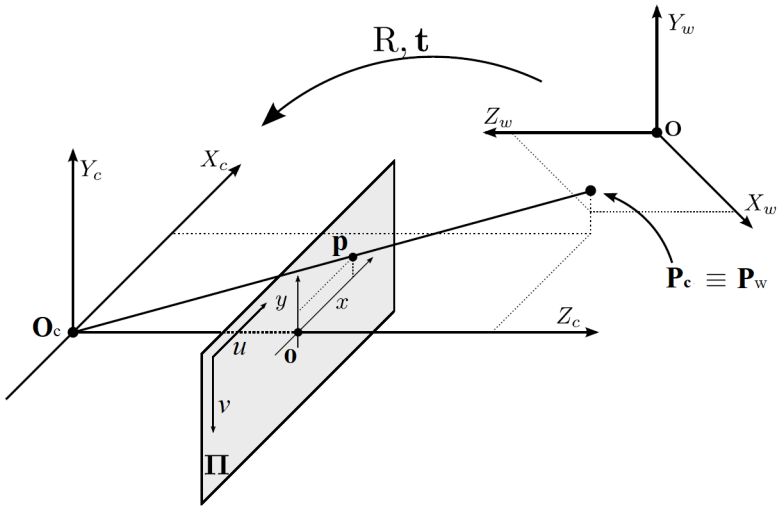
Transformação do SCM para SCC

A transformação do SCM para o SCC é necessária, pois é preciso identificar o posicionamento da câmera em relação a um SCM conhecido, portanto, é realizada uma transformação do SCM para o SCC, sendo assim, os pontos do espaço tridimensional passam a ter representações equivalentes, a primeira em função da origem O_w e a segunda em função da origem O_c.

A relação entre os sistemas de coordenadas é dada por uma matriz de transformação que

envolve translação e rotação, conforme ilustra a Figura 3.

Figura 3: Transformação Rígida ente SCM e SCC.



Fonte: Adaptado de Laureano (2013)

A translação é composta por um vetor t de dimensões 3×1, em que cada elemento corresponde a translação relativa das origens dos dois sistemas. Já a rotação R é uma matriz 3×3 formada pela combinação de outras 3 matrizes que representam as rotações nos eixos Zc, Yc e Xc com ângulos de rotação ϕ, γ e θ, respectivamente.

$$R = R_x(\phi)R_y(\gamma)R_z(\theta) = \begin{bmatrix} cos\phi & -sen\phi & 0 \\ sen\phi & cos\phi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} cos\gamma & 0 & sen\gamma \\ 0 & 1 & 0 \\ -sen\gamma & 0 & cos\gamma \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos\theta & -sen\theta \\ 0 & sen\theta & cos\theta \end{bmatrix}. \quad (1)$$

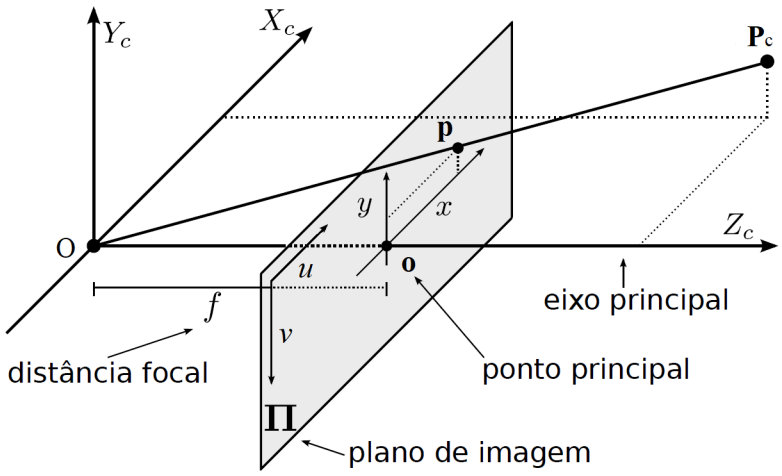
A matriz extrínseca é formada pela pela matriz R e o vetor t, dessa forma, os pontos P_c e P_w são relacionados por:

$$\begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \end{bmatrix} + \begin{bmatrix} t_1 \\ t_2 \\ t_3 \end{bmatrix}. \quad (2)$$

Mudança do SCC para SCI

Esta transformação descreve a projeção perspectiva do ponto P_c pertencente ao SCC em coordenadas do SCI, adotando o modelo câmera *pinhole*. Neste modelo, ocorre uma aproximação, de modo que, assume-se que o orifício da câmera é formado por um único ponto, desconsiderando o sistema de lentes. Na Seção 2.2.2 é abordado o processo de correção da distorção radial e tangencial provocada pelas lentes. A Figura 4 ilustra a formação de uma imagem no modelo de câmera *Pinhole*.

Figura 4: Relação entre SCC e SCI.



Fonte: Adaptado de Laureano (2013).

Neste modelo, a distância focal f é uma característica construtiva da câmera e corresponde a distância entre o centro de projeções O_c e o plano da imagem. O ponto $P_c = [X_c\ Y_c\ Z_c]^{\mathrm{T}}$ situado no SCC é mapeado no plano da imagem no ponto $p = [x\ y]^{\mathrm{T}}$ por meio da transformação projetiva, portanto, os pontos passam a ser representadas em coordenadas homogêneas.

Se todos os pontos forem representados em coordenadas homogêneas, a transformação de coordenadas no espaço 3D para o espaço 2D é dada pela matriz de projeções:

$$\begin{bmatrix} x \\ y \\ w \end{bmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix} = \tilde{p} = M\tilde{P}_c, \tag{3}$$

em que a representação do ponto $\tilde{p}$ no sistema cartesiano é $p = \left[\frac{x}{w}\right]^{\mathrm{T}}$.

Por meio das relações de semelhança de triângulos percebe-se que o ponto P_c é mapeado no ponto $[f\frac{X_c}{Z_c}\ f\frac{Y_c}{Z_c}\ f]^{\mathrm{T}}$ no plano da imagem, logo, $p = [f\frac{X_c}{Z_c}\ f\frac{Y_c}{Z_c}]^{\mathrm{T}}$.

Transformação do SCI para SCP

A presente transformação consiste em converter as coordenadas do SCI, ainda no sistema métrico, para coordenadas do SCP dadas em unidades de *pixel*. Neste sentido, é necessário identificar parâmetros característicos da câmera como o fator de escala η_x e η_y, referente ao comprimento do *pixel* em milímetros, τ, dado pela inclinação existente entre os eixos u e v e a coordenada do ponto principal $[c_x\ c_y]^{\mathrm{T}}$. Caso os *pixels* do sensor da câmera sejam exatamente quadrados $\eta_x = \eta_y$. Como regra geral, adota-se a origem do sistema de coordenadas da imagem como sendo o *pixel* mais à esquerda e mais acima, portanto, é necessário transladar em relação ao ponto principal $[c_x\ c_y]^{\mathrm{T}}$. Essa transformação entre planos é dada por:

$$\begin{bmatrix} u \\ v \\ w \end{bmatrix} = \begin{bmatrix} \eta_x & \tau & c_x \\ 0 & \eta_x & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix}. \tag{4}$$

É possível, no entanto, realizar a conversão do SCC para o SCP associando a Equação 3 com a Equação 4, formando a matriz de parâmetros intrínsecos K.

$$\begin{bmatrix} u \\ v \\ w \end{bmatrix} = \begin{bmatrix} f\eta_x & \tau & c_x \\ 0 & f\eta_x & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix} = \tilde{p} = K\ P_c. \tag{5}$$

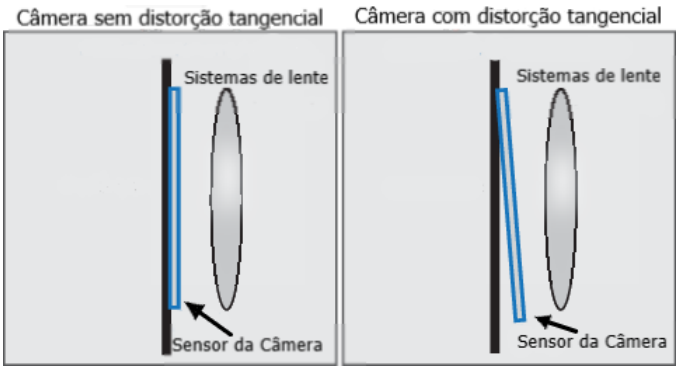
2.2.2 Distorção das Lentes

A distorção é uma característica inerente ao sistema óptico, especialmente às lentes. Existem dois principais tipos de distorção presente nas lentes de câmeras digitais: a distorção radial e a distorção tangencial.

A distorção radial pode ser encarada como sendo a parcela não desejável da refração de um raio de luz ao atravessar uma lente, formando um ângulo com o eixo óptico, provocando um deslocamento dos pontos da imagem, ou seja, a distorção incorpora ao mapeamento de um segmento de reta do sistema de coordenadas de mundo um efeito de curvatura na imagem. Portanto, a eliminação da distorção preserva a propriedade da colinearidade entre os planos, essencial para que as coordenadas do SCP, possam ser convertidas para o SCM.

A distorção tangencial é causada por um desalinhamento físico dos elementos que constituem a lente, fazendo com que a lente não fique paralela ao sensor da câmera, ilustrado na Figura 5. Este tipo de distorção é mais frequente em câmeras com foco ajustável, embora o impacto desta pertubação seja menor nas câmeras mais modernas (Park et al., 2009).

Figura 5: Causas da Distorção Tangencial.



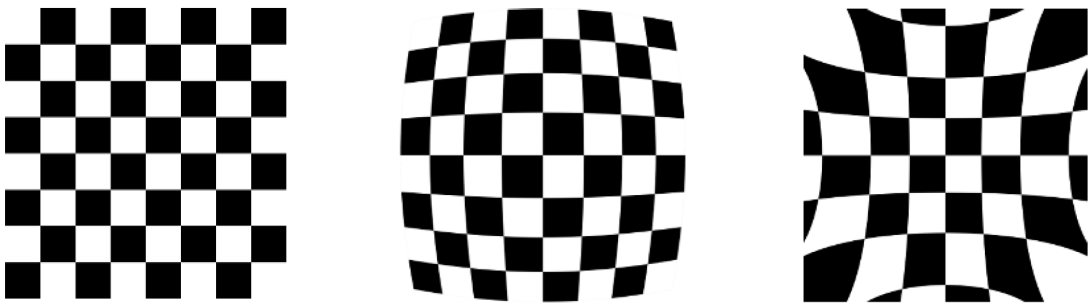
Fonte: Adaptado do site Mathworks[2]

A distorção radial exerce maior influência nos modelos de lentes comumente utilizados, tendo seus efeitos ilustrados na Figura 6. A distorção radial positiva provoca na imagem o aspecto denominado de efeito “barril”, ilustrado na Figura 6 (b) e acomete, na maioria dos casos, lentes com uma pequena distância focal e um grande campo de visão. O efeito visual desta distorção faz

[2] Disponível em: <https://www.mathworks.com/help/vision/ug/camera-calibration.html>. Acesso em fev. 2017.

com que o centro da imagem fique mais ampliado em relação às bordas. Por outro lado, a distorção radial negativa ilustrada na Figura 6 (c) causa na imagem o aspecto de "almofada"caracterizado por uma ampliação nas bordas da imagem em relação ao centro. Este tipo de distorção é mais frequente em lentes usadas para longo alcance, ou seja, que possuem uma maior distância focal e consequentemente menor campo de visão. Algumas lentes possuem os dois tipos de distorção incorporados, a este tipo de distorção dá-se o nome de distorção "bigode" e suas principais características são a ampliação do centro da imagem, assim como na distorção radial positiva, e as suas extremidades apresentam o efeito da distorção radial negativa.

Figura 6: Efeitos da Distorção. (a) Imagem: sem Distorção. (b) Imagem: Distorção Radial Positiva. (c) Imagem: Distorção Radial Negativa.



(a) (b) (c)

Fonte: Documentação da OpenCV[3]

O método para a modelagem da distorção considera a distorção radial e tangencial, adotando k_1, k_2 e k_3 como coeficientes da distorção radial e p_1 e p_2 coeficientes da distorção tangencial.

Sabendo que a distorção radial provoca o mapeamento de uma reta em uma linha com uma determinada curvatura (Park et al., 2009), a correção deste efeito é definida pela Equação 6, cujas coordenadas (x_d, y_d) são os pontos com distorção e (x_u, y_u) são os pontos com a distorção radial corrigida.

[3]Disponível em: < https://docs.opencv.org/2.4/modules/calib3d/doc/camera_calibration_and_3d_reconstruction.html>. Acesso em jun. 2017

$$
\begin{aligned}
x_u &= x_d(1 + k_1 r^2 + k_2 r^4 + k_3 r^6) \\
y_u &= y_d(1 + k_1 r^2 + k_2 r^4 + k_3 r^6).
\end{aligned}
\tag{6}
$$

A correção da distorção tangencial pode ser modelada pela Equação 7, em que $r^2 = x^2 + y^2$ representa o deslocamento radial em relação ao centro óptico:

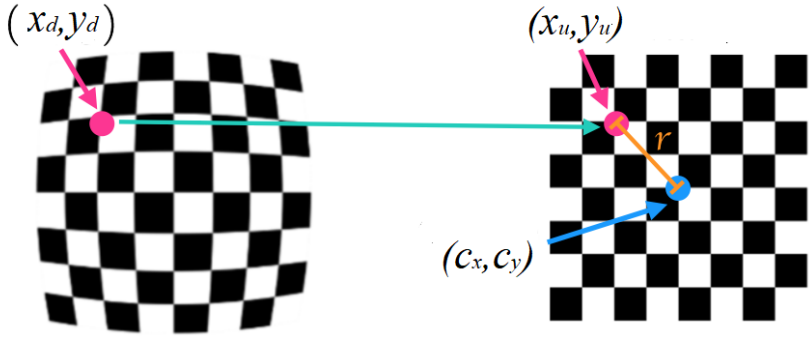
$$
\begin{aligned}
x_u &= x_d + [2p_1 x_u y_u + p_2(r^2 + 2x_u^2)] \\
y_u &= y_d + [p_1(r^2 + 2x_u^2) + 2p_2 x_u y_u],
\end{aligned}
\tag{7}
$$

portanto a modelagem matemática final incluindo a distorção tangencial e radial é expressa por:

$$
\begin{aligned}
x_u &= x_d(1 + k_1 r^2 + k_2 r^4 + k_3 r^6) + 2p_1 x_u y_u + p_2(r^2 + 2x_u^2) \\
y_u &= y_d(1 + k_1 r^2 + k_2 r^4 + k_3 r^6) + p_1(r^2 + 2y_u^2) + 2p_2 x_u y_u.
\end{aligned}
\tag{8}
$$

Após a identificação dos coeficientes polinomiais de distorção radial k_1, k_2 e k_3 e os coeficiente de distorção tangencial p_1 e p_2, é realizado o mapeamento dos pontos, gerando uma nova imagem com as coordenadas corrigidas.

Figura 7: Correção da Distorção.



Fonte: Elaborado pela Autora.

2.2.3 Distorção Perspectiva

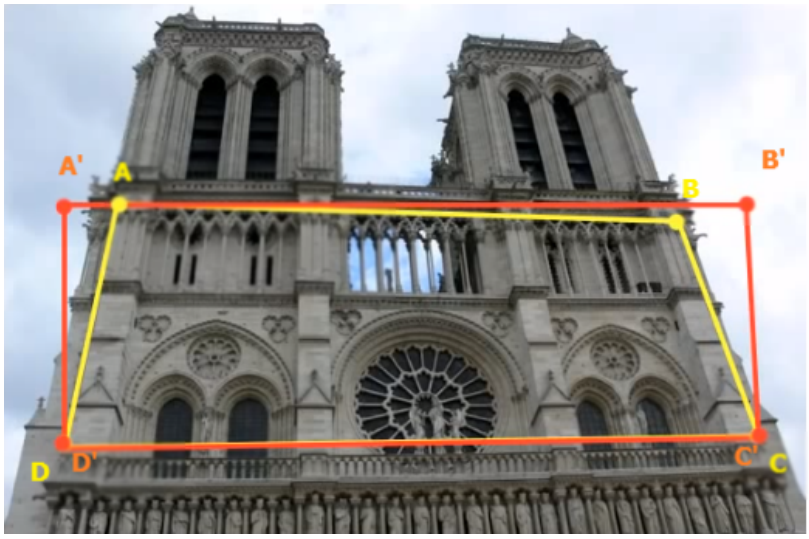
O fenômeno da distorção perspectiva está relacionada à características inerentes da geometria projetiva tais como a redução da dimensão dos objetos à medida que se afastam do centro de projeções e a convergência de linhas paralelas no plano de projeções. Fatores como o desalinhamento perpendicular da câmera em relação ao objeto e a utilização de lentes grande-angular intensificam este tipo de distorção.

A correção da distorção perspectiva é realizado pela transformação perspectiva que é fundamentalmente um mapeamento entre pontos pertencentes a dois planos distintos, preservando a relação de colinearidade entre estes. Para realizar esta transformação é necessário obter, no mínimo, uma correspondência de 4 pontos, entre a imagem de origem e a imagem de destino. A Figura 8 ilustra um caso de transformação de perspectiva a partir da correspondência entre os vértices A, B, C e D com os vértices A', B', C' e D'.

Conhecendo os valores das coordenadas dos vértices é possível relacioná-las em coordenadas homogêneas para estimar uma matriz de homografia capaz de relacionar o plano do objeto com o plano sem distorções de perspectiva. Os cálculos referentes à matriz de homografia são detalhados na Seção 2.3.

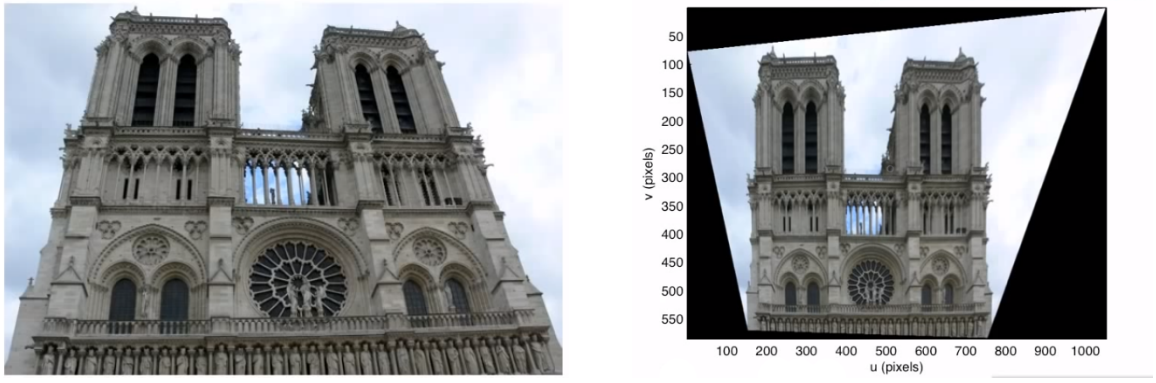
Com as coordenadas dos pontos A', B', C' e D', bem como os coeficientes da matriz H, aplica-se técnicas que atuam na correção da distorção causada pela perspectiva, sendo a técnica de deformação *perspective warping* a mais empregada nesta problemática (Heckbert, 1999). O *perspective warping* foi empregado na Figura 9 para corrigir a distorção de perspectiva e consiste no mapeamento do espaço de entrada (x, y) no espaço de saída (x', y').

Figura 8: Transformação de Perspectiva.



Fonte: Adaptado de Peter Corke[4]

Figura 9: Efeitos da Perspectiva. (a) Imagem: com Distorção Perspectiva. (b) Imagem com Distorção de Perspectiva Corrigida.



(a) (b)

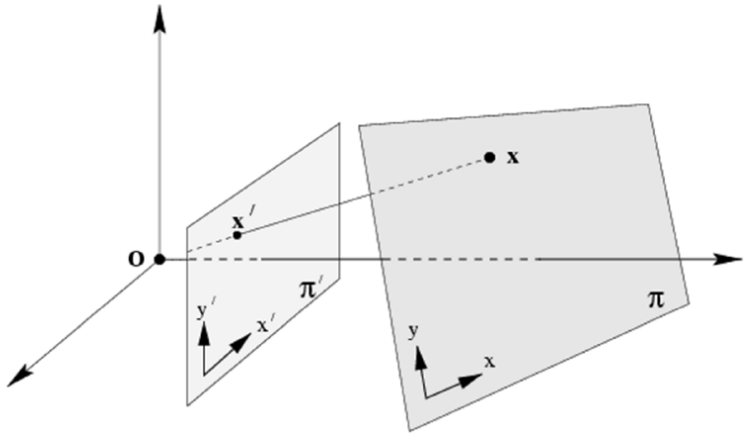
Fonte: Adaptado de Peter Corke[8]

[4] Disponível em: <https://www.youtube.comwatch?v=fVJeJMWZcq8list=PLMBUCrSudtjMBRy-w0DQDLdzYq5$_{P}gekNindex = 8 > .Acessoemjun.$2017.

2.3 Homografia Planar

A Homografia é uma transformação projetiva planar que corresponde ao mapeamento de um conjunto de pontos entre dois planos distintos. O cálculo da homografia é feito quando se deseja descobrir a relação entre coordenadas em *pixel* de uma imagem com as coordenadas de mundo, considerando que estas estão dispostas em um plano π. Desta forma, as coordenadas de mundo podem ser representadas de acordo com a Figura 10.

Figura 10: Homografia entre o plano π e plano π'



Fonte: Szeliski (2010)

A matriz de homografia determina a relação de um ponto em um mundo planar e a posição de sua representação em uma imagem. Seus coeficientes são dados pela relação entre a matriz de parâmetros intrínsecos e a matriz de parâmetros extrínsecos resultantes do processo de calibração, conforme descritas na Equação 9.

$$\begin{bmatrix} u \\ v \\ w \end{bmatrix} = \begin{bmatrix} f\eta_x & \tau & c_x \\ 0 & f\eta_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix}. \tag{9}$$

Tendo em vista que as coordenadas de mundo estão fixas em um mesmo plano, conclui-se que valores de $Z_w = 0$, portanto as coordenadas do ponto P_w são representadas por:

$$P_w = \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} = \begin{bmatrix} x_w \\ y_w \\ 0 \\ 1 \end{bmatrix}. \tag{10}$$

Reescrevendo a Equação 9, assumindo que as coordenadas de mundo estão no plano, a terceira coluna da matriz de rotação é eliminada.

$$\begin{bmatrix} u \\ v \\ w \end{bmatrix} = \begin{bmatrix} f\eta_x & \tau & c_x \\ 0 & f\eta_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & t_1 \\ r_{21} & r_{22} & t_2 \\ r_{31} & r_{32} & t_3 \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ 1 \end{bmatrix}, \tag{11}$$

portanto, a relação entre a posição de um ponto no SCM e a posição da sua projeção no SCP é dada pela matriz de Homografia H de dimensões 3x3 dada por:

$$\begin{bmatrix} u \\ v \\ w \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ 1 \end{bmatrix}. \tag{12}$$

Sabendo que a matriz H é uma matriz não-singular, ou seja, não é possível utilizar a sua inversa H^{-1} para calcular o valor das coordenadas de mundo $[X_w, Y_w, 1]^T$, nesse caso, é preciso estimar os valores dos coeficientes da matriz de homografia de modo que seja possível conhecer a sua inversa. O método da Transformação Direta Linear DLT (do inglês, *Direct Linear Transform*) proposto por Abdel and Karara (1971) é capaz de estimar os coeficientes da matriz de homografia de 8 graus de liberdade a partir de 4 coordenadas correspondentes, uma vez que é possível obter duas equações linearmente independentes por cada correspondência.

A matriz H possui 9 elementos, no entanto, como esta é invariante a escala, fixa-se o valor de fixa-se $h_{33} = 1$, uma vez que a matriz H multiplicada por um fator de escala qualquer também será uma solução da Equação 12.

Reescrevendo as equações que representam as duas primeiras linhas da Equação 12, já que a terceira linha é linearmente dependente das duas primeiras, chega-se a:

$$\begin{cases} uX_w.h_{31} + uY_w.h_{32} + u = X_w.h_{11} + Y_w.h_{12} + h_{13} \\ vX_w.h_{31} + vY_w.h_{32} + v = X_w.h_{21} + Y_w.h_{22} + h_{23} \end{cases} \tag{13}$$

As duas equações linearmente independentes do sistema descrito acima possuem 8 incógnitas que foram obtidas a partir de um único ponto na imagem e seu respectivo correspondente. Havendo n pontos, o conjunto de $2n$ equações podem ser representadas matricialmente pela Equação 14 que representa o sistema linear $Ax = b$.

$$\begin{bmatrix} X_w^1 & Y_w^1 & 1 & 0 & 0 & 0 & -u^1X_w^1 & -u^1Y_w^1 \\ 0 & 0 & 0 & X_w^1 & Y_w^1 & 1 & -v^1X_w^1 & -v^1Y_w^1 \\ X_w^2 & Y_w^2 & 1 & 0 & 0 & 0 & -u^2X_w^2 & -u^2Y_w^2 \\ 0 & 0 & 0 & X_w^2 & Y_w^2 & 1 & -v^2X_w^2 & -v^2Y_w^2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ X_w^n & Y_w^n & 1 & 0 & 0 & 0 & -u^nX_w^n & -u^nY_w^n \\ 0 & 0 & 0 & X_w^n & Y_w^n & 1 & -v^nX_w^n & -v^nY_w^n \end{bmatrix} \begin{bmatrix} h_{11} \\ h_{12} \\ h_{13} \\ h_{21} \\ h_{22} \\ h_{23} \\ h_{31} \\ h_{32} \end{bmatrix} = \begin{bmatrix} u^1 \\ v^1 \\ u^2 \\ v^2 \\ \vdots \\ u^n \\ v^n \end{bmatrix}, \tag{14}$$

em que A é uma matriz de tamanho $2N \times 8$, a matriz x possui dimensão 8×1 contém os coeficientes da matriz de homografia, enquanto a matriz b de tamanho $2N \times 1$ contém as coordenadas dos pontos na imagem.

Sabendo que o sistema possui 8 incógnitas e $2n$ equações, conclui-se que são necessários no mínimo 4 pontos para calcular a homografia. Com exatamente 4 pontos, é possível calcular a matriz A e consequentemente determinar os coeficientes de distorção contidos na matriz x.

$$x = A^{-1}b. \tag{15}$$

Com $n > 4$, o sistema de equações é sobredeterminado, ou seja, o número de equações é maior que o número de incógnitas, sendo assim, esse problema deve ser resolvido de forma a minimizar o erro quadrático médio, através da pseudo-inversa:

$$x = (A^TA)^{-1}A^Tb. \tag{16}$$

Rearranjando o vetor x, obtém-se a matriz de homografia inversa, capaz de identificar pontos no SCM com base nas coordenadas de uma imagem:

$$\begin{bmatrix} X \\ Y \\ 1 \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix}^{-1} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}. \tag{17}$$

Como existem ruídos no sistema, quanto maior o número de pontos utilizados, melhor será a qualidade do resultado.

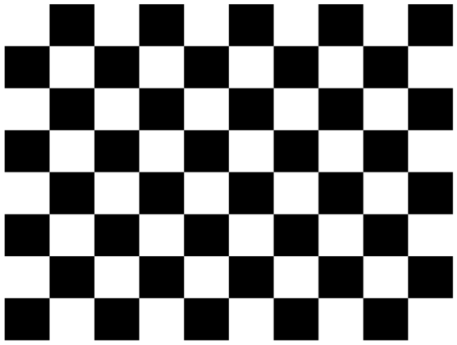
2.4 Padrão de Calibração

Os parâmetros de calibração são estimados relacionando pontos conhecidos no ambiente com suas respectivas projeções no plano da imagem e para isso utilizam-se padrões de calibração contendo pontos de referência com valores previamente conhecidos. Os padrões de calibração tem a função de tornar mais prática a identificação das coordenadas de cada ponto de referência. Estes padrões podem ser tridimensionais (Salvi et al., 2002), bidimensionais ou unidimensionais (Zhang, 2004). A calibração utilizando objetos 3D se baseia na identificação precisa da geometria de um determinado objeto, ou padrão tridimensional composto por um conjunto de pontos de referencia contidos em dois ou três planos perpendiculares entre si (Zhang et al., 1997). A calibração usando objetos unidimensionais consiste na utilização de pontos de referência alinhados sobre uma reta, embora o processo de calibração seja similar aos que empregam padrões 2D, neste caso é preciso extrair um número bem maior de imagens (Loaiza et al., 2011).

A facilidade construtiva dos padrões de calibração 2D e a popularidade dos seus algoritmos tornou este método mais difundido em aplicações de calibração de câmera. Dentre os padrões de calibração 2D mais utilizados, destaca-se o padrão planar quadriculado similar ao tabuleiro de xadrez, cujos pontos de referência estão localizados nos cantos onde os quadriláteros de cores diferentes se tocam. Os valores encontrados durante o processo de calibração da câmera são totalmente dependente da precisão do algoritmo de detecção dos pontos de referência. Fatores como iluminação inadequada e baixa qualidade de impressão da folha de calibração afetam a precisão do sistema. Para minimizar a incerteza nos cálculos dos parâmetros de calibração, aumenta-se o conjunto de coordenadas relativas aos pontos de referência, elevando também o número de ima-

gens capturadas, portanto, são capturadas várias imagens do padrão de calibração em diferentes orientações, assim como ilustrado a Figura 11.

Figura 11: Padrão de Calibração com pontos de referência marcados.



Fonte: (Zhang, 2000).

A dimensão das arestas dos quadriláteros assim como as coordenadas cartesianas dos pontos de referência são parâmetros essenciais para relacionar as coordenadas SCM com os pontos do SCP. Para obter parâmetros de calibração mais exatos é necessário um algoritmo de detecção de cantos robusto, de forma que consiga detectar os pontos de referência com precisão em ambientes com condições de iluminação não-uniformes.

2.5 Detecção do Objeto

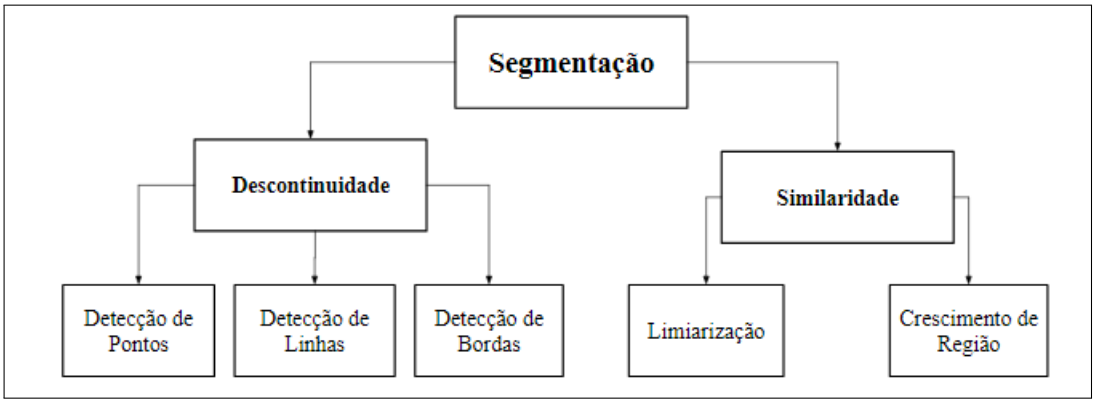
A segmentação de imagens é uma das principais técnicas de análise de imagens, cujo objetivo principal é ressaltar características relevantes para a aplicação em questão com base em aspectos relativos ao valor de intensidade de *pixel*, segregando a imagem em regiões específicas. As regiões de interesse são denominadas de objeto, nelas estão contidas informações relevantes para a interpretação das imagens, por outro lado, as regiões menos relevantes são denominadas como *background*. Após a segmentação o objeto é descrito por características geométricas ou topológicas tais como área, forma, textura, centro de massa e centroide.

A detecção de objetos em imagens consiste na identificação dos *pixels* que caracterizam um objeto presente em uma cena. O número de técnicas destinadas a detecção de objetos tem

crescido rapidamente em virtude da robustez dos métodos de aprendizagem de máquina voltadas para a otimização da segmentação de imagens, ou o uso daqueles que utilizam a segmentação nos primeiros estágios de classificação.

As principais técnicas de segmentação são agrupadas de acordo com duas propriedades relacionadas aos níveis de intensidade da imagem: Descontinuidade ou Similaridade. Os métodos baseados em descontinuidade levam em consideração as mudanças abruptas nos níveis de intensidade da imagem que podem constituir pontos isolados, linhas ou bordas presentes na imagem, enquanto os métodos de segmentação por similaridade subdivide a imagem em regiões de acordo com alguma característica, sendo normalmente usado o valor de limiar. O diagrama ilustrado na Figura 12 aponta as distinções entre alguns tipos de técnicas de segmentação utilizadas na área de Processamento Digital de Imagens (PDI) (Filho and Neto, 1999).

Figura 12: Diagrama das Técnicas de Segmentação de Imagens.



Fonte: Elaborado pela autora.

O desempenho do processo de segmentação está diretamente relacionado à extração de características da imagem, no entanto, isso pode ser afetado por variáveis como ruído, iluminação não controlada ou problemas relacionados à aquisição da imagem.

2.5.1 Limiarização

A limiarização é a técnica de segmentação que consiste na análise de todos os *pixels* que compõem a imagem com o intuito de determinar se estão especificados acima ou abaixo de um valor

de limiar, ou seja, os *pixels* com intensidade acima do valor de limiar assumirão o nível lógico 1, ao passo que, os *pixels* com valores de intensidade abaixo do limiar, assumirão o nível lógico 0. O resultado desse processo é uma imagem binarizada com apenas dois níveis de intensidade, cujo nível máximo representa o objeto e nível mínimo representa o *background* da imagem.

$$g(x,y) = \begin{cases} 0, \text{ se } f(x,y) \leq T \\ 255, \text{ caso contrário} \end{cases} \tag{18}$$

em que T é o valor de limiar global, $f(x,y)$ é o valor de intensidade da imagem monocromática de 8-*bits* e $g(x,y)$ o valor de intensidade da imagem binarizada.

Esta técnica de binarização é denominada de limiarização global, pois apresenta apenas um valor de limiar em toda imagem. A escolha do limiar é uma tarefa de extrema importância, uma vez que seu valor pode afetar diretamente a detecção do objeto. Existem diversas técnicas de análise de histogramas com o intuito de achar o melhor valor que separe as duas classes, no entanto, nem sempre é possível empregar este tipo de análise, pois a imagem pode conter ruídos que dificultam essa estimativa.

De acordo com Pedrini and Schwartz (2008) não é recomendado o uso da limiarização global em casos cuja imagem apresente grandes variações entre os níveis de cinza do objeto e do fundo da imagem provocadas pela iluminação não-uniforme, problemas durante a aquisição ou outros fatores.

A alternativa mais apropriada nessas circunstâncias é o uso da limiarização local. Nesta técnica são definidos diferentes valores de limiar para cada região da imagem, tornando o processo de segmentação mais flexível as condições não-uniformes de iluminação. A seleção do valor de limiar local pode ser associado aos valores da média, mediana, desvio padrão ou a outras funções de uma vizinhança local. A Equação 19 descreve o processo de limiarização local por três métodos distintos, pelo valor da média, mediana e a média dos valores máximos e mínimos, respectivamente.

$$T = Media_v(p)$$

$$T = Mediana_v(p) \tag{19}$$

$$T = \frac{min_v(p) + max_v(p)}{2}$$

em que v representa a vizinhança local ao ponto p da imagem.

O tamanho da vizinhança é um parâmetro que deve ser informado à função de limiarização local e deve ser escolhido um tamanho grande o suficiente para conter partes do objeto e do fundo da imagem, por outro lado, uma janela muito grande não é capaz de fornecer valores de limiar adequados para imagens com iluminação não-uniforme.

2.6 K-means

O algoritmo K-means ou K-médias é um método de *clustering* iterativo que visa agrupar elementos em *k* grupos pré-determinados. Foi desenvolvido por MacQueen (1967) e atualmente faz parte dos algoritmos de *Machine Learning* com aprendizagem não supervisionada, sendo assim, o *K-means* é aplicado em diversas áreas do conhecimento.

O algoritmo possui duas fases. Na primeira fase são calculados os centroides de cada grupo e na segunda fase, cada ponto pertencente ao conjunto de dados é alocado no grupo mais próximo de acordo com o cálculo da distância Euclidiana média entre o centroide e o respectivo ponto. Após o agrupamento, novos valores de centroides são calculados até que a distância média entre os pontos de cada grupo seja minimizada.

No campo da visão computacional, o *K-means* pode ser utilizado na segmentação de imagens em tons de cinza. Considerando o conjunto de *pixels* que formam uma imagem como o conjunto de dados inicial do algoritmo, o agrupamento é realizado a partir dos valores dos centroides que representam um nível de intensidade luminosa na imagem. A priori são definidas as quantidades de grupos, ou seja, em quantos segmentos a imagem será dividida, em seguida, para cada *pixel* da imagem é calculada a distância *d* entre o centro e cada *pixel* da imagem.

Seja *p(x,y)* o *pixel* que será agrupado em um c_k centroide, esta distância é obtida pela relação:

$$d = \|p(x, y) - c_k\| . \tag{20}$$

Os *pixels* são atribuídos a um centroide com base no cálculo da distância d. Na iteração seguinte são calculadas novas posições para os centroides:

$$c_k = \frac{1}{k} \sum_{y \epsilon c_k} \sum_{x \epsilon c_k} p(x, y). \tag{21}$$

2.7 Detecção de Cores

O processamento de imagens coloridas é uma importante tarefa da área de processamento digital de imagens, pois em alguns casos a identificação de um objeto só pode ser realizada diante da detecção da cor.

As imagens digitais coloridas são representadas em um espaço de cor, sendo o mais comum o RGB, composto pelos canais de cores primárias vermelho, azul e verde, respectivamente. Esse padrão de cor é um dos mais utilizados devido a compatibilidade com a maioria dos *hardwares* e também com o olho humano que é mais receptivo para enxergar a combinação de cores primárias.

As imagens coloridas são formadas pela junção de três componentes de imagem de 8 *bits*, sendo assim, uma imagem RGB é um trio de valores (R,G,B) com uma profundidade de 24 *bits* e com capacidade de representação de até $(2^8)^3 = 16.777.216$ cores (Gonzalez and Woods, 2010).

Na Figura 13 é ilustrado a decomposição de uma imagem RGB em seus três canais de cores. Na Figura 13(a) tem-se uma imagem no espaço de cor RGB, na sequência os canais azul na Figura 13(b), verde na Figura 13(c) e vermelho na Figura 13(d).

Figura 13: Decomposição dos canais de cores. (a) Imagem colorida no espaço de cor RGB. (b) Canal Azul da Imagem. (c) Canal Verde da Imagem. (d) Canal Vermelho.

(a) (b)

(c) (d)

Fonte: Gonzalez and Woods (2010).

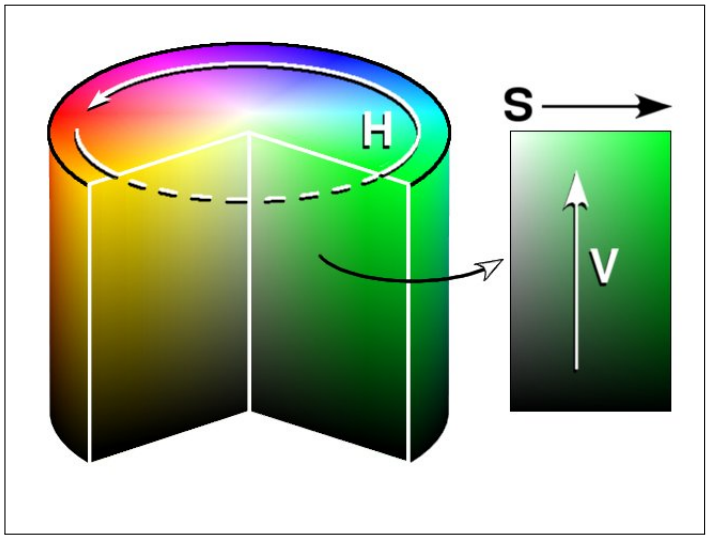
Apesar do padrão de cor RGB apresentar uma série de vantagens, na maioria das aplicações envolvendo a detecção de cor de uma imagem é preciso realizar conversões para outros padrões de cores. Isso acontece porque a interpretação humana das cores não é feita a partir da associação de porcentagem das cores primárias.

2.7.1 Espaço de Cor HSV

O espaço de cor HSV é formado por três componentes, são eles a matiz (*Hue*), a saturação (*Saturation*) e o brilho (*Value*). A matiz é referente a cor pura, a saturação é a medida de diluição de uma cor pura por luz branca e o brilho é uma variável subjetiva que indica a quantidade de brilho na imagem (Gonzalez and Woods, 2010). O fato de ter uma variável capaz de representar a cor pura, faz com que o padrão de cor HSV seja um dos mais utilizados em algoritmos de detecção de cor.

A Figura 14 ilustra o espaço de cor HSV. A matiz é representada pelo disco superior do cilindo HSV, seus valores variam de $0°$ a $360°$, podendo ser normalizados entre 0% e o 100%. Cada fatia do disco representa uma cor arbitrária. O comprimento do vetor horizontal S indica a saturação que varia em uma faixa entre 0% e 100%, da mesma forma o comprimento do vetor vertical V indica o brilho da cor que também varia entre 0% e 100%.

Figura 14: Espaço de Cor HSV.



Fonte: Su et al. (2011).

2.8 Equalização de Histograma

A equalização de histograma é realizada a partir uma função de transformação que tem como objetivo ajustar a distribuição dos *pixels* de uma imagem, distribuindo-os uniformemente ao longo da escala de níveis de intensidade (Gonzalez and Woods, 2006).

Por definição, um histograma de uma imagem indica o número de *pixels* contidos em um determinado nível de cor (para imagens coloridas) ou nível de cinza no intervalo $[0,\ L-1]$, em que L indica a quantidade de níveis de intensidade. Quando os valores do histograma são normalizados é possível afirmar estatisticamente qual a probabilidade de incidência do *pixel* n_k no k-ésimo valor de intensidade r_k.

$$p_r(r_k) = \frac{n_k}{MN}, \qquad k = 0, 1, 2, 3, ...L-1, \tag{22}$$

em que MN é o número total de *pixels* da imagem, portanto, o gráfico de $p_r(r_k)$ em relação ao valores de r_k é definido como histograma normalizado.

A equalização de histograma é visto como uma operação de transformação global com r representando os valores de intensidade da imagem de entrada, de modo que $r = 0$ representa o preto e $r = L-1$ representa o branco,

$$s = T(r), \qquad 0 \leq r \leq L-1, \tag{23}$$

dessa forma, a imagem resultante desse processo será obtida pelo mapeamento de cada *pixel* da imagem de entrada com nível de intensidade r_k em um *pixel* correspondente de nível s_k

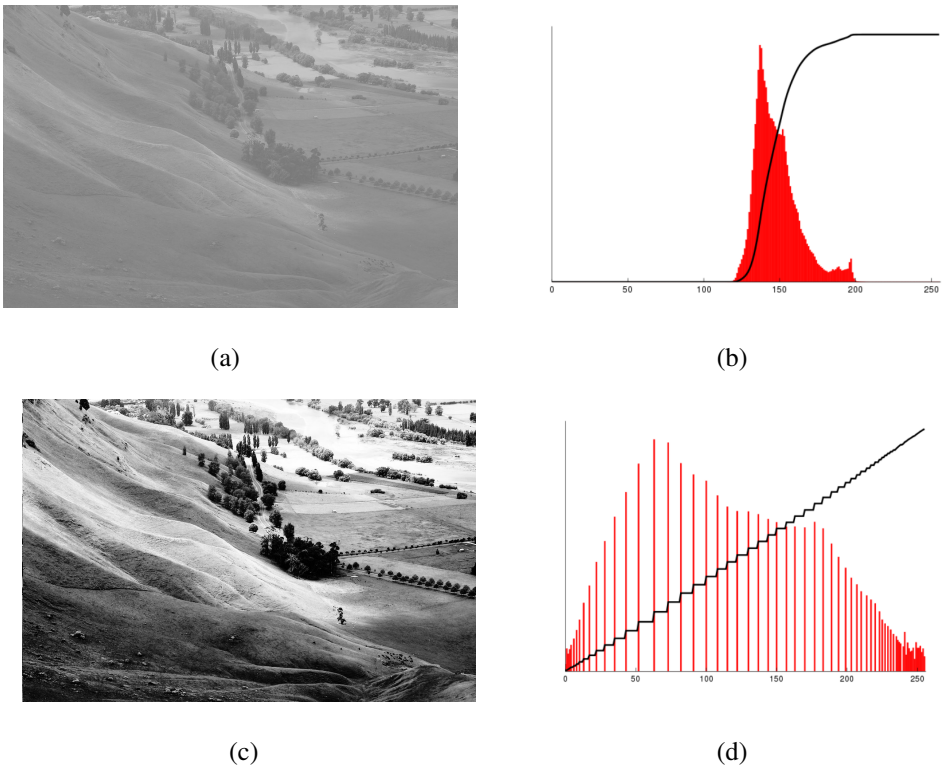
$$s_k = T(r_k) = (L-1)\sum_{j=0}^{k} p_r(r_j) = \frac{(L-1)}{MN}\sum_{j=0}^{k} n_j. \tag{24}$$

Existe uma relação entre o histograma e o contraste da imagem. Os histogramas com aparência mais concentrada em uma pequena faixa de intensidades, representam imagens com baixo contraste, ou seja, o realce também é baixo, uma vez que não existe uma transição tão marcante nas bordas das imagens.

A Figura 15 ilustra a técnica de equalização de histograma. Na Figura 15 (a) é ilustrada uma imagem de baixo contraste e seu respectivo histograma (Figura 15 (b)) onde é possível obser-

var uma frequência de distribuição em uma pequena faixa de níveis de cinza, resultando em uma imagem com uma aparência cinza, desbotada e sem brilho. Após a equalização, a Figura 15 (c) ilustra um contraste mais acentuado, onde nota-se uma riqueza maior de detalhes na imagem. Na Figura 15 (d) é ilustrado uma distribuição mais uniforme dos *pixels* ao longo do histograma.

Figura 15: Equalização de Histograma. (a) Imagem com baixo contraste. (b) Histograma da imagem de baixo contraste. (c) Imagem após a Equalização do Histograma. (d) Histograma Equalizado.



(a) (b)

(c) (d)

Fonte: Elaborado pela autora.

A equalização de histograma pode ser classificada como uma técnica de transformação global utilizada como pré-processamento em imagens capturadas com diferentes condições de iluminação. No entanto, o realce de contraste pela equalização do histograma apresenta algumas limitações, sobretudo em imagens com muitas regiões claras e escuras. Nestes casos, a equalização local de histograma é mais indicada.

A equalização local de histograma (AHE, do inglês, *Adaptive Histogram Equalization*) con-

siste em calcular vários histogramas para cada região da imagem visando redistribuir os valores de luminosidade na imagem de forma mais uniforme. No entanto, essa técnica tende a acentuar ruídos em regiões homogêneas. Para evitar esse tipo de efeito, aplica-se a limitação de contraste, resultando na técnica de Equalização Local de Histograma Limitada por Contraste (CLAHE, do inglês, *Contrast Limited Adaptive Histogram Equalization*).

O algoritmo CLAHE é uma versão aprimorada do algoritmo AHE, pois com a limitação de contraste é possível atenuar os ruídos em regiões homogêneas provocados pela AHE. Regiões homogêneas são identificadas no histograma como picos em uma curta faixa de níveis de intensidade, portanto, limitar o contraste, em termos práticos, é limitar estas regiões do histogramas com o chamado *clip limit* ou limitação de pico (Zuiderveld, 1994). Os *pixels* que estavam acima dessa limitação são redistribuídos em todo o histograma para manter a contagem total do histograma.

2.9 Momentos Invariantes

Os momentos invariantes têm sido utilizado em aplicações de detecção de objetos e sua principal vantagem é a invariância à rotação, translação e escala. Esses podem ser calculados pelo método inicialmente proposto por Hu (1962), que consiste em extrair características geométricas de um objeto presente na imagem binarizada.

Após ressaltar os objetos de interesse na imagem por meio da segmentação, é realizado o cálculo dos momentos da imagem para localizar o centroide dos objetos. Um momento padrão ou regular é invariante à translação e permite o cálculo da área e do centroide de um objeto. O momento de ordem $(p + q)$ de imagens $f(x, y)$ é definidos conforme a Equação 25.

$$M_{pq} = \sum_{1}^{nx} \sum_{1}^{ny} x^p y^q f(x, y) \tag{25}$$

Os valores de nx e ny representam a largura e altura da imagem respectivamente, enquanto $f(x, y)$ é a intensidade do *pixel* da imagem binarizada que assume apenas dois valores 0 ou 1. Desta forma, o momento M_{00} corresponde à área do objeto na imagem.

$$M_{00} = \sum_{1}^{nx} \sum_{1}^{ny} x^0 y^0 f(x, y) = \sum_{1}^{nx} \sum_{1}^{ny} f(x, y) \tag{26}$$

O momento de ordem $p = 1$ e $q = 0$ fornece o somatório das coordenadas no eixo x,

enquanto que o momento de ordem $p = 0$ e $q = 1$ fornece o somatório de todas as coordenadas do eixo y. Dividindo estes momentos pela área do objetos, obtém-se o ponto correspondente ao centroide.

$$\overline{x} = \frac{M_{10}}{M_{00}} \qquad \overline{y} = \frac{M_{01}}{M_{00}} \tag{27}$$

A coordenada $(\bar{x}, \bar{y})$ corresponde ao centroide do objeto-alvo que deve ser convertido para o sistema de coordenadas de mundo e posteriormente enviado ao manipulador robótico.

3 Revisão Bibliográfica

Muitos trabalhos vêm adotando técnicas para detecção automática dos objetos presentes em uma cena, a partir de sistemas de VM. Este Capítulo traz uma breve introdução aos conceitos de detecção de objetos e calibração de câmeras, em seguida, é realizada uma descrição das recentes pesquisas apresentadas pela comunidade científica voltadas à aplicação de algoritmos de VC em tarefas que auxiliam manipuladores robóticos industriais na detecção de peças.

3.1 Considerações Iniciais

As atividades de *pick-and-place*[5] executadas por robôs dotados de VM vêm sendo cada vez mais adotadas em função do aumento na produtividade e flexibilidade nas operações, uma vez que é possível reconhecer automaticamente diferentes objetos presentes em uma cena, informar as respectivas posições e orientações ao manipulador robótico sem que haja necessidade da intervenção humana (Liu et al., 2012).

De acordo com Vernon (1991), um sistema de VM é composto por uma ou mais câmeras, sistema de iluminação, *hardware*, *software* e uma interface de comunicação com o robô industrial ou algum outro dispositivo que atue diretamente no meio físico. No campo da automação industrial, os sistemas de VM atuam essencialmente como sensores indiretos, substituindo os limitados sensores de posição na detecção de objetos.

As técnicas de detecção podem ser classificadas em detecção 2D e 3D. A primeira se refere a detecção de objetos em um plano bidimensional. Utilizando apenas uma câmera estática é possível informar as coordenadas cartesianas (x, y) de determinado alvo, no entanto, o sistema se torna limitado na detecção de objetos de diferentes alturas. Portanto, os sistemas tridimensionais utilizam duas câmeras ou uma câmera associada a um sensor capaz de fornecer a noção de profundidade para estimar a altura dos objetos.

Contudo, a precisão com que o robô captura o objeto-alvo está diretamente ligada a calibração da câmera que é responsável pela correção da distorção provocada pelas lentes, além da conversão do sistema de coordenadas da imagem dado em *pixel* para o sistema de coordenadas de mundo[6], ge-

[5] *Pick and Place* é um termo utilizado na área robótica que faz referência ao movimento de capturar um objeto em determinada posição e posteriormente transportá-lo, alocando em outra posição.

[6] Coordenada de mundo é um termo utilizado em Visão Computacional para referir-se às coordenadas cartesianas

ralmente dado em milímetros. Portanto, Davies (2012) afirma que a calibração de câmeras permite que os sistemas de VC extrapolem a capacidade da visão humana, pois com uma câmera corretamente calibrada é possível realizar medições com uma precisão superior aos sistemas naturais de visão.

A calibração da câmera pode ser divida em duas principais etapas, a primeira busca modelar a projeção da imagem por meio de um conjunto de parâmetros, enquanto a segunda busca estimar esses parâmetros por métodos diretos ou indiretos (Zhang, 2004). Normalmente os pontos que se deseja identificar na imagem pertencem a um padrão de calibração e estes devem ser detectados com alta precisão para fornecer parâmetros confiáveis.

3.2 Sistemas de VM em atividades de *Pick-and-Place*

Em Gao et al. (2016) é desenvolvido um sistema de posicionamento baseado em visão monocular responsável pela detecção de baterias. O trabalho faz uso da arquitetura composta por uma câmera CCD fixada acima da superfície na qual as baterias são alocadas, um manipulador robótico industrial de 6 graus de liberdade e um controlador DSP TMS3206748 e FPGA conectados pelo barramento *EMIFA bus*. Os autores realizam a etapa de calibração automática da câmera utilizando a técnica *eye-to-hand*. Em seguida, Gao et al. (2016) eliminam a distorção presente na imagem para melhorar a precisão do sistema e aplicam técnicas de pré-processamento como filtro gaussiano e equalização de histograma. Na fase de detecção do objeto, é realizada a segmentação da imagem por limiarização e posteriormente a morfologia matemática para melhorar a área detectada. Os resultados indicam uma precisão com incerteza de 1 milímetro nos eixos X e Y e desvio de ângulo menor que 1 grau.

Xia and Weng (2016) abordam a aplicação de um sistema de visão monocular em tarefas de *pick-and-place*. Neste caso, o algoritmo identifica objetos de diferentes formatos geométricos, realiza o processamento das imagens em um computador convencional e envia os resultados acerca das coordenadas dos objetos-alvo ao controlador do manipulador robótico. Inicialmente é realizada a calibração da câmera pelo método de Zhang (2000), utilizando um plano de calibração similar ao tabuleiro de xadrez, 17 imagens são capturadas para estimar os parâmetros intrínsecos da câmera. O algoritmo de VC é dividido em três etapas: extração de contornos, reconhecimento de forma e

de pontos no espaço

localização e captura. Na primeira etapa é realizada a captura da imagem, em seguida, aplica-se o detector de borda Canny. Na etapa seguinte, o autor adota um novo método para reconhecimento da forma dos objetos, baseado na transformada de Hough probabilística e no método Freeman. Desta forma, é possível identificar triângulos, quadrados, pentágonos e objetos de cantos arredondados. Na etapa final, é realizada a transformação dos sistemas de coordenadas e as coordenadas baricêntricas dos objetos detectados são enviados ao controlador.

Para validar o trabalho, Xia and Weng (2016) compararam o método de Hough-Freeman com o método de reconhecimento baseado em Momentos invariantes e Momento Zernike e concluem que o método Hough-Freeman possui uma acurácia mais elevada, no entanto, para objetos com cantos arredondados é necessário uma otimização adicional.

O trabalho de Lin et al. (2016) também aborda o desenvolvimento de um sistema de detecção de peças geométricas, no entanto, fornece na saída as coordenadas 3D dos objetos, baseado no conceito de visão estéreo. A arquitetura do sistema é composta por duas câmeras fixadas no topo de uma plataforma, capturando simultaneamente duas imagens em ângulos distintos. Os autores utilizam sistematicamente três principais aspectos acerca do sistema de detecção proposto: a calibração do sistema de VM que consiste na calibração da câmera, calibração estéreo e calibração *hand-eye*, o reconhecimento dos objetos e as transformações de coordenadas. A calibração da câmera tem o objetivo de identificar parâmetros intrínsecos, extrínsecos e os coeficientes de distorção, neste caso adotou-se o método de Zhang (2000) empregado pela biblioteca OpenCV. A calibração estéreo consiste na relação entre a imagem capturada pela câmera esquerda com a imagem capturada pela câmera direita, por fim, encontra-se a matrix de transformação relacionando o plano da imagem com o plano do robô. A detecção dos objetos é realizada utilizando o método de detecção *blob* e para objetos cúbicos, também é feita a detecção dos pontos relativos as 4 arestas da face superior, logo, é possível informar a orientação destes objetos com o intuito de rotacionar a garra para a captura destas peças.

Kim et al. (2017) desenvolvem em seu trabalho um sistema de VM baseado no reconhecimento dinâmico de objetos com diferentes formas, posições, distâncias, além de não apresentar restrições quanto as condições de iluminação externa. O reconhecimento dinâmico de objetos é realizado de três formas: localização randômica de diferentes tipo de objetos a partir de uma câmera

de alta resolução, *Visual Servoing*[7] e detecção a partir de visão estéreo de objetos empilhados na estrutura *bin picking*[8].Um cenário composto por várias câmeras, sistema de iluminação, um mini-transportador para movimentar os objetos nas tarefas de detecção dinâmica (rastreamento), além de um sensor laser para a detecção da distância dos objetos foi elaborado. A seleção do tipo de câmera e do tipo de iluminação é realizada inicialmente de acordo com a atividade proposta.

Os autores ressaltam que a detecção de objetos de diferentes tipos de materiais afetam a acurácia do sistema, pois determinados materiais afetam o comportamento dos detectores de borda e incorporam distorções à imagem. Neste sentido, é proposto a associação de técnicas adaptativas como a binarização local adaptativa e o filtro de diferença de gaussianas (DoG), enquanto a binarização local adaptativa é usada para preservar as características de um objeto que está sob condições irregulares de iluminação e para extrair características das bordas da imagem, o filtro DoG apresenta melhores resultados em situações nas quais o objeto tem baixo contraste devido a efeitos da iluminação. Características locais presentes nos objetos como símbolos ou códigos de barras são utilizadas como pontos de referência para a identificação dos objetos passados para o extrator de características que usa redes neurais com *backpropagation* modificado.

Por fim, para estimar a posição dos objetos, aplica-se a transformada de Hough na detecção das linhas que compõem os objetos e conclui-se que o erro na etapa de treinamento para o reconhecimento dos objetos é de 0%; na fase de testes o erro subiu para 2,2%.

3.3 Visão Computacional utilizando Raspberry Pi

O termo Visão Embarcada (*Embedded Vision*) é utilizado para caracterizar aplicações de VM que são processadas por dispositivos embarcados. O crescente poder computacional dos SBCs aliado ao seu baixo custo, está fazendo como que estes dispositivos sejam imersos cada vez mais no contexto industrial.

Em Nair et al. (2017) é utilizado o Raspberry Pi 3 como uma IPU (*Image Processing Units*), ou seja, um dispositivo dedicado ao processamento de imagens. Os autores apontam uma série de vantagens no crescente uso dos SBCs tais como portabilidade e baixo consumo de energia quando comparados com PCs, além da facilidade na programação quando comparado com dispositivos

[7] *Visual Servoing* é um termo da área de VM que significa o controle de movimentação robótica por meio da resposta do sistema de visão

[8] *bin picking* é o nome da técnica utilizada na captura randômica de objetos que estão dentro de uma caixa

embarcados como um DSP, que exige uma programação mais próxima ao nível de *hardware*. Nair et al. (2017) concluem que o IPU baseado na Raspberry Pi possui uma alta precisão na detecção e no rastreamento de objetos.

Na literatura é possível encontrar o RPi envolvido no processamento de algoritmos mais robustos. Em Yahya Nikouei et al. (2018) é implementada uma rede neural convolucional leve (L-CNN) para a detecção de seres humanos em tempo real a partir de câmeras de vigilância. A L-CNN possui uma baixa sobrecarga na etapa de pré-processamento o que a torna mais adequada para dispositivos com processamento limitado. Os autores analisam diferentes algoritmos de *Machine Learning* para detecção de seres humanos em tempo real e compararam parâmetros como a quantidade de FPS (*frames* por segundo), a taxa de processamento da CPU, o consumo de memória e a taxa de valores falso-positivos (FPR) para medir o desempenho de cada algoritmo no RPi. Estes valores são descritos na Tabela 1.

Tabela 1: Desempenho em FPS, CPU, Memória Utilizada e Taxa Média de Valores Falso-Positivos (FPR%).

Algoritmos	**FPS**	**CPU (%)**	**Memória (MB)**	**FPR(%)**
Haar Cascaded	1,82	76,9	111,6	26,3
HOG + SVM	0,30	93,0	139,3	14,5
SSD GoogleNet	0,39	84,7	320,4	5,3
L-CNN	1.79	75,7	122,5	6,6

O algoritmo L-CNN é o segundo melhor em termos de velocidade e obteve uma taxa de falsos positivos baixa. É possível processar quase dois *frames* por segundo utilizando a arquitetura de rede neural convolucional proposta com o consumo de memória três vezes menor que o SSD GoogleNet. Além disso, devido ao menor número de filtros em cada camada, a L-CNN obteve um menor tempo de processamento que a GoogleNet.

Este trabalho elucida o comportamento da RPi no processamento de algoritmos robustos de Machine Learning e conclui que com a rede L-CNN é possível processar quase 2 *frames* por segundo, afirmando o sistema embarcado como uma possível solução para problemas de detecção de pessoas em tempo real utilizando câmeras de segurança.

3.4 Discussões do Capítulo

Os trabalhos apresentados nesse Capítulo foram divididos em dois principais eixos. O primeiro eixo agrupa trabalhos que envolvem o desenvolvimento de algoritmos de VC para auxiliar manipuladores robóticos industriais. Os trabalhos presentes no segundo eixo, tratam da aplicação do RPi em problemas que envolvem VC.

Em Gao et al. (2016) e Xia and Weng (2016) são utilizadas câmeras fixas posicionadas acima dos objetos-alvo, ambos os trabalhos não utilizam sistemas de iluminação para atenuar os efeitos da iluminação externa. A calibração da câmera foi realizada com base no método desenvolvido por Zhang (2000).

Os sistemas de visão que utilizam apenas uma câmera fixa têm uma complexidade menor quando comparados com sistemas com mais de uma câmera ou com um sensor extra, ou seja, os sistemas monoculares demandam menos poder computacional dos *hardwares*. Em contrapartida, as aplicações se tornam mais limitadas, pois não é possível realizar a detecção exata de objetos com diferentes alturas.

Em Lin et al. (2016) é utilizado o conceito de visão stereo baseado em duas câmeras fixas, nesse caso, a calibração é feita pelo método conhecido como *hand eye calibration* e consiste na captura do padrão de calibração pelo manipulador robótico. O robô é responsável pela movimentação deste padrão enquanto as câmeras realizam as capturas. A partir das imagens capturadas, é aplicado o método de calibração de Zhang (2000). Esse tipo de calibração automatizada é a mais empregada em linhas de montagem robotizadas que utilizam sistemas de VM, pois diminuem os erros causados pelos seres humanos.

O trabalho de Nair et al. (2017) utiliza a RPi como *hardware* destinado ao processamento de algoritmos de VC, já em Yahya Nikouei et al. (2018) são aplicados algoritmos de aprendizagem profunda para o rastreamento de pessoas. Essa atividade exige um maior poder computacional, devido a complexidade de características do objeto-alvo (seres humanos) como também pelo fato do rastreamento de objetos ser uma atividade que requer uma resposta mais rápida do sistema de visão.

A essência desse método aprendizagem de máquina está baseada na aplicação de redes neurais profundas, compostas por várias camadas convolucionais que extraem características que

compõem um objeto. As redes mais modernas possuem uma arquitetura mais leve, possibilitando o uso dessa nova tecnologia em sistemas embarcados que detém um poder computacional mais restrito.

Portanto, a escolha pela utilização de um sistema embarcado permite que o sistema de VM seja mais acessível, uma vez que um minicomputador é compacto, leve e é capaz de realizar o processamento de algoritmos de VC.

4 Materiais e Métodos

Este Capítulo tem como foco a descrição dos materiais e métodos utilizados nesta pesquisa, justificando a sua aplicabilidade na solução da problemática. O Capítulo encontra-se estruturado nas Seções 4.1, na qual é apresentado o ambiente onde o sistema proposto está localizado, em seguida, são apresentados na Seção 4.2 os materiais utilizados no desenvolvimento da pesquisa como Raspberry Pi e o Manipulador Robótico Industrial, descritos na Subseção 4.2.1 e Subseção 4.2.2 respectivamente. Finalmente, aborda-se o desenvolvimento do algoritmo utilizado na calibração da câmera na Subseção 4.3.2 e na detecção dos objetos, descrito na Subseção 4.3.3.

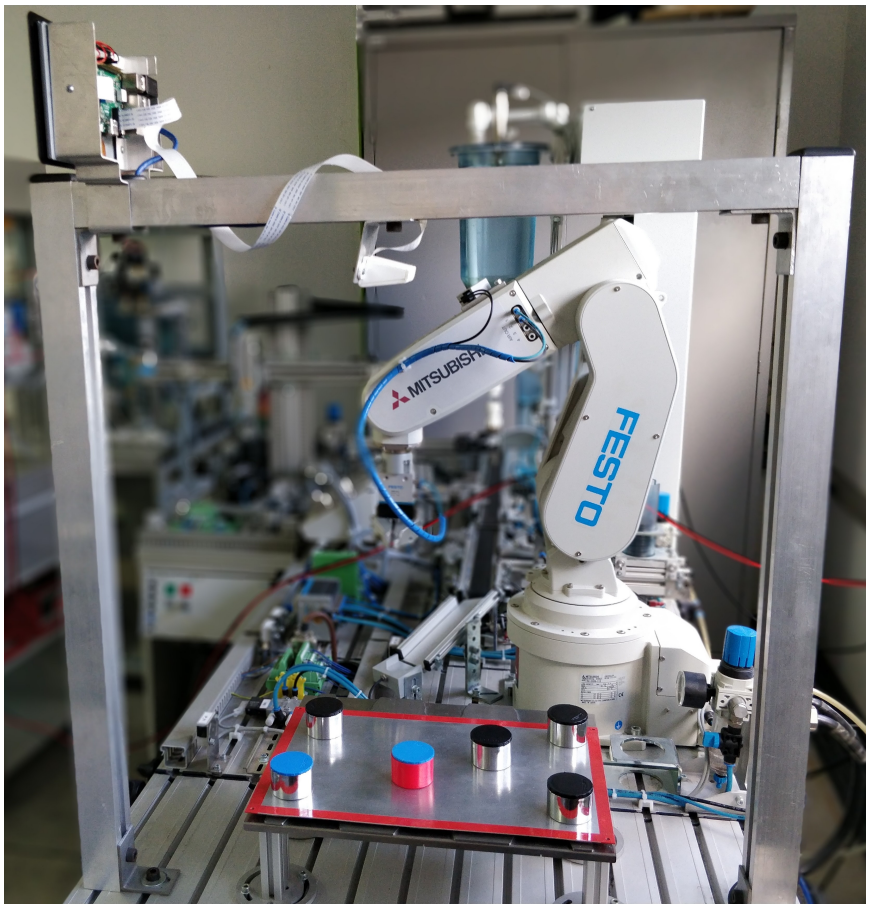
4.1 Descrição do Cenário

Esta Seção descreve o cenário de pesquisa e análise deste trabalho. A Figura 16 ilustra o manipulador robótico e o sistema de VM que encontram-se nas instalações do IFPB (Instituto Federal de Educação Ciências e Tecnologia da Paraíba) vinculados a uma bancada de manufatura (*Modular Production System* - MPS) desenvolvida pela Festo Didatics.

O sistema proposto deve inferir a posição de objetos cilíndricos de mesma altura dispostos aleatoriamente sobre uma plataforma metálica a partir de informações extraídas de uma imagem. Os cilindros possuem 40,1 mm de diâmetro e são posicionados em uma orientação fixa. Um vetor contendo a posição cartesiana (x, y) dos objetos é enviado ao robô que finalmente realiza captura e armazenamento dos cilindros em uma posição pré-definida.

A garra eletropneumática acoplada ao robô possui 42,1 mm de diâmetro de abertura. Em virtude da sua geometria, apenas objetos cilíndricos podem ser capturados. Por causa do espaço reduzido existente entre a abertura da garra e o diâmetro dos objetos a serem capturados, torna-se necessário que o sistema de VM atue corretamente a fim de evitar colisões entre o robô e a peça-alvo.

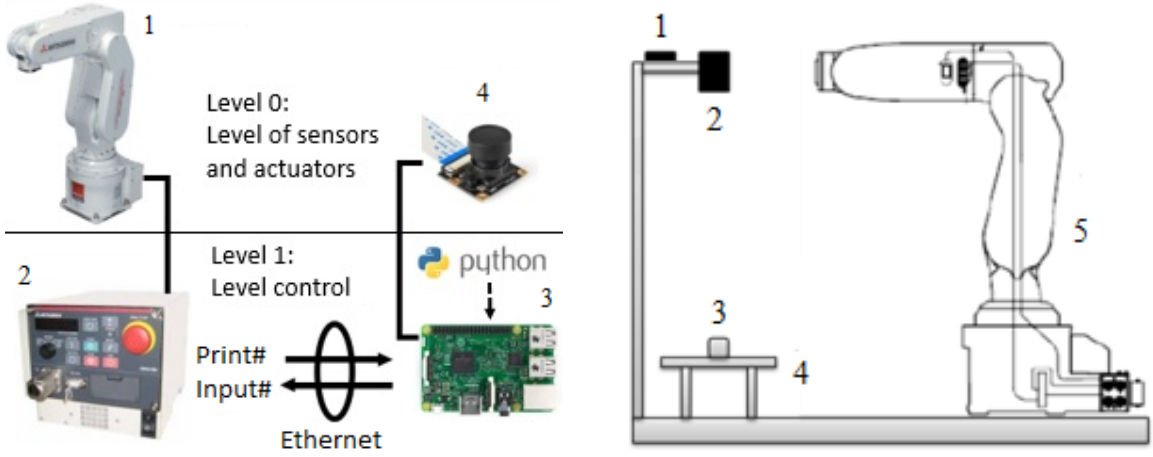
Figura 16: Instalações do sistema de VM.



Fonte: Elaborado pela autora.

Em termos de arquitetura, o sistema VM pode ser dividido em dois níveis, conforme ilustra a Figura 17(a). O nível 0 (zero) é caracterizado pela zona dos sensores/atuadores: o módulo PiCamera como sensor e o manipulador robótico como atuador. No nível 1 (um) ocorre o processamento de informações: o RPi recebe as imagens capturadas pela PiCamera, realiza o processamento digital destas, obtém as coordenadas da peça alvo e envia um vetor de coordenadas para o controlador do robô processar e direcionar o braço. A Figura 17(b) ressalta os elementos que compõem esse cenário. São eles: o sistema de VM, a plataforma onde as peças cilíndricas são dispostas e o manipulador robótico industrial.

Figura 17: Cenário de pesquisa. (a) Estrutura de comunicação: 1 - Robô RV-2SD, 2 - Controlador CR1DA-700, 3 - Raspberry Pi3, 4 - PiCamera V.1. (b) Posicionamento dos componentes: 1 - Sistema Embarcado, 2 - Câmera, 3 - Peça alvo, 4 - Plataforma, 5 - Robô.



(a) (b)

Fonte: Elaborado pela autora.

4.2 Materiais

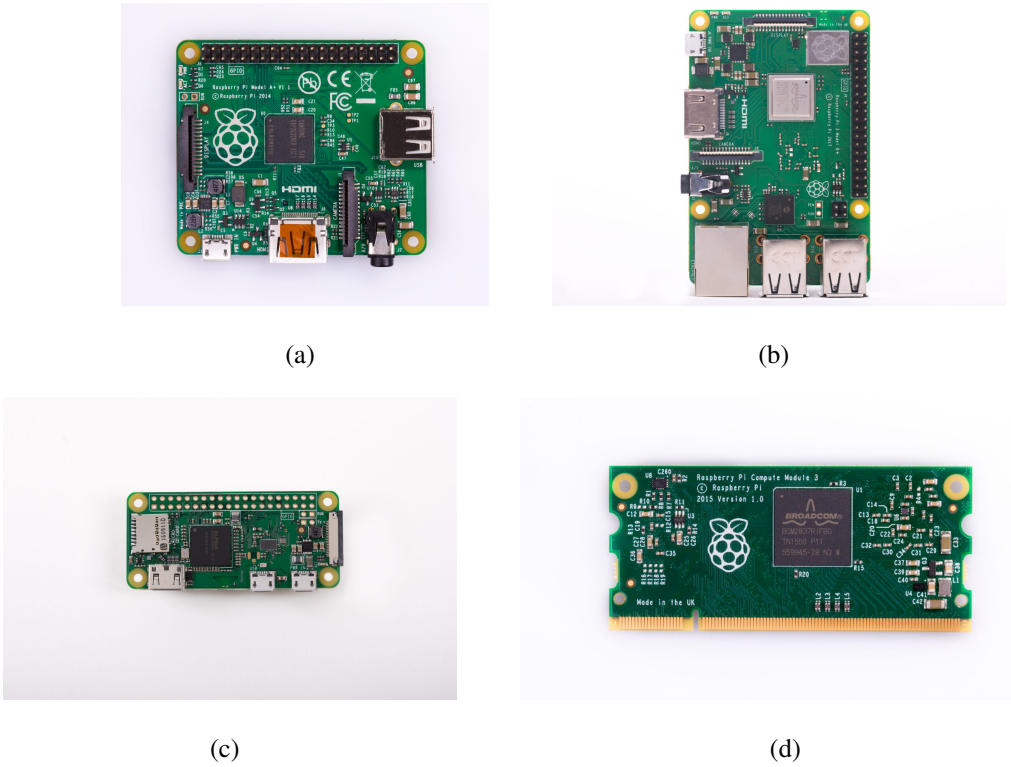
A presente Seção descreve as especificações do sistema embarcado Raspberry Pi e do Manipulador Robótico Industrial Mitsubishi modelo RV-2SDB, apontados como os principais materiais utilizados nesta pesquisa. O Raspberry Pi é o sistema embarcado que nesta aplicação encontra-se encarregado de processar a imagem e comunicar-se com o robô, informando as coordenadas do centroide dos cilindros detectados. Embora o sistema de MV proposto esteja integrado ao manipulador RV-2SDB, é possível realizar a comunicação com qualquer manipulador robótico que disponha de comunicação Ethernet.

4.2.1 Raspberry Pi

A Raspberry Pi surgiu no Reino Unido no ano de 2012 com a sua primeira versão conhecida como Raspberry Pi 1 B, seguida do seu segundo modelo mais acessível Raspberry 1 A, desde então, estes dispositivos foram ficando cada vez mais populares e atualmente existem 4 principais modelos de RPi no mercado, são eles Raspberry A, Raspberry B, *Compute Module* e Raspberry Pi Zero (Figura 18). Esse crescimento vertiginoso ocorreu devido ao preço acessível e a grande co-

munidade de estudantes, educadores, pesquisadores e desenvolvedores engajados na disseminação desta tecnologia. O objetivo inicial da Raspberry Pi Foundation, empresa que desenvolve os RPis, era criar um dispositivo com fins educacionais, o qual as crianças pudessem utilizar como ferramenta no aprendizado de programação, no entanto, por ser um sistema embarcado bastante versátil, possibilitou o desenvolvimento de aplicações que vão desde o desenvolvimento de jogos, até o uso em sistemas de Visão Computacional.

Figura 18: Principais Modelos de Raspberry Pi. (a) Raspberry Pi A+. (b) Raspberry Pi 3 B+. (c) Rasperry Pi Zero W. (d) Raspberry Pi *Compute Module*.



(a) (b)

(c) (d)

Fonte: Adaptado do site da Raspberry Pi[9]

Os testes iniciais para a implementação do sistema de VM foram realizados com a Raspberry Pi 2 B devido a disponibilidade desse dispositivo na Instituição, no entanto, a capacidade desta plataforma não supria as necessidades estabelecidas nesta pesquisa. Foram observados atrasos no envio de informações para o controlador do robô RV-2SDB bem como eventuais travamen-

[9] Disponível em: <https://www.raspberrypi.org/products/>. Acesso em jul. 2017.

tos, sobreaquecimentos e atrasos na exibição das imagens capturadas. Portanto, a Raspberry Pi 2 foi substituída pelo modelo mais recente lançado pela empresa Raspberry Pi Foundation.

A Raspberry Pi 3 B, foi lançada no mercado no ano de 2016 com funcionalidades bem mais atrativas do que a versão anterior. O sistema embarcado conta com um Sistema Operacional (SO) Raspbian Stretch baseado no Debian, desenvolvido especificamente para esta plataforma embarcada, além de memória RAM (1GB), Wi-Fi 802.11 b/g/n (2.4GHz), Bluetooth 4.1 (BCM43438) e o processador da Broadcom BCM2837 de 4 núcleos de 64 bits ARM Cortex-A53 a 1.2GHz, que traz como GPU a VideoCore IV. Este novo processador traz um incremento de 33% na velocidade de processamento e com várias melhorias de arquitetura, o que proporciona um aumento de 50-60% no desempenho face ao Raspberry Pi 2. Também possui 4 portas USB, Camera Serial Interface (CSI), *Display Serial Interface* (DSI), entrada para cartão micro SD, porta ethernet, saída de áudio e HDMI, e 40 pinos GPIO para conectar a placa a outros dispositivos (Foundation, 2016b).

O barramento CSI possibilita a conexão direta com o módulo de câmera (*board-level camera*) chamado de PiCamera com a GPU VideoCore IV. Desenvolvido pela empresa Raspberry Pi Foundation, a PiCamera atualmente conta com duas versões, a Camera Module V2 e a Pi Noir Camera V2, em que a primeira é uma câmera CMOS de 8 megapixels e a segunda uma câmera infravermelha de 8 megapixels, neste trabalho utilizou-se a PiCamera V2, conectada à RPi via cabo *flat* de 15 cm de comprimento da câmera ao barramento CSI, assim como ilustra na Figura 19.

Figura 19: Câmera PiCamera V2 conectada pelo barramento CSI ao Raspberry Pi.

Fonte: Elaborado pela Autora.

A câmera possui dimensões de 25×209 mm, pesando 3 gramas, utiliza o sensor Sony

IMX219 com resolução de 8 megapixels, foco manualmente ajustável, suporta os modos de vídeo 1080p 30 fps, 720p 60 fps e VGA 90 fps, bem como captura de fotos com resolução de até 3280×2464 *pixels* (Foundation, 2016a).

4.2.2 Manipulador Robótico Industrial

O robô RV-2SDB é um braço robótico industrial desenvolvido pela Mitsubishi possui 6 (seis) graus de liberdade, peso total de 19 kg, raio de alcance de 504 mm, repetibilidade de 0,02 mm, velocidade máxima de 4,4 m/s, *encoder* absoluto e *drivers* dotados de servo motores AC (*Alternating Current*). Seu controlador, modelo CR1DA-700, é programado na linguagem de Melfa Basic V, utilizando o ambiente de desenvolvimento Ciros.

O CR1DA-700 suporta três tipos de comunicação *ethernet*: através de funções de comunicação do Controlador, por meio da função *Data Link* ou usando a função de controle externo em Tempo Real. A primeira técnica normalmente é aplicada quando *softwares* ou *hardwares* proprietários específicos precisam ser configurados para estabelecerem uma comunicação com o robô, como por exemplo o *software* CIROS Studio.

A comunicação em tempo real é utilizada quando aplicações externas necessitam controlar o robô sem haver interrupções ou atrasos. A técnica de comunicação *ethernet* definida para controlar o robô foi a *Data Link*. Neste protocolo de comunicação, dados de posição do braço, por exemplo, podem ser recebidos e enviados da RPi para o controlador através de um link estabelecido utilizando os comandos de comunicação da linguagem *Melfa Basic V*.

A *Data Link Ethernet TCP/IP* possui um comportamento baseado em eventos, ou seja, enquanto não houver recebimento de dados pela entrada do canal de comunicação, o programa interno permanecerá em *standby* (Mitsubishi, 2011).

4.3 Métodos

Nesta Seção são abordados os métodos utilizados no desenvolvimento do algoritmo de calibração da câmera utilizando o tabuleiro de xadrez como padrão de calibração bem como a implementação do algoritmo de detecção dos objetos por meio da linguagem de programação Python e bibliotecas como a OpenCV para o processamento de imagens, Numpy para o cálculo matricial e Matplotlib para a exibição de gráficos e imagens.

4.3.1 Bancos de Imagens

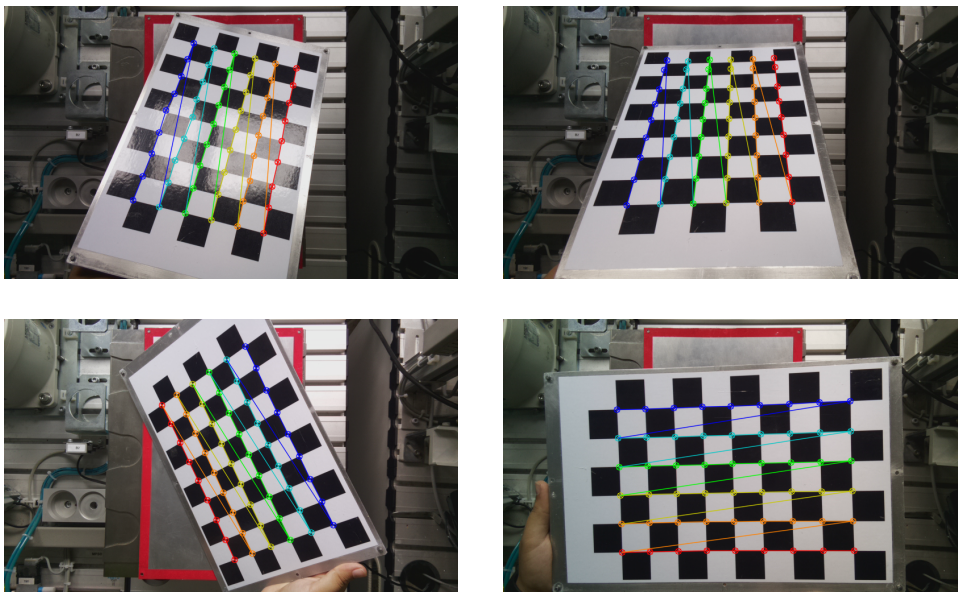
Os bancos de imagens utilizados nesta pesquisa foram feitos a partir da captura de imagens pelo sistema de VM. Com a câmera posicionada acima da plataforma, foram capturadas imagens da vista superior dos cilindros para que o algoritmo de detecção pudesse segmentar a área da superfície superior do objeto.

Foram construídos cinco bancos de imagens para finalidades distintas. O primeiro banco é referente a um conjunto de 30 imagens contendo o padrão de calibração xadrez em diferentes orientações. Essas imagens compõem o conjunto de entrada do algoritmo de calibração.

Para a construção do padrão de calibração, foi impresso em uma impressora à laser nas dimensões de uma folha A4, a imagem do tabuleiro de xadrez com nove linhas e seis colunas quadriculadas está disponibilizada no repositório da OpenCV (Documentation, 2018) cuja resolução é 1830×1330. Em seguida essa imagem foi fixada em uma superfície metálica para garantir que todos os pontos do padrão de calibração seriam coplanares.

Na Figura são ilustradas quadro das 30 imagens que constituem o banco de calibração. As linhas e pontos coloridos marcados no padrão de calibração indicam que os pontos detectados pelo algoritmo de calibração e suas conexões.

Figura 20: Banco das Imagens de Calibração



Fonte: Elaborado pela Autora

O segundo banco é um conjunto de 21 imagens com resolução 800 $\times$ 480 *pixels* no formato PNG, contendo os cilindros de tampa preta em diferentes condições de iluminação. O objetivo da criação desse banco é testar os diferentes algoritmos de detecção e avaliar o que melhor se adapta em condições de iluminação variáveis. O terceiro banco é composto pela mesma quantidade de imagens do banco anterior, no entanto, é referente as imagens dos cilindros de tampa colorida. Empregou-se a mesma metodologia na captura destas imagens.

O quarto banco de imagens armazena 21 imagens dos cilindros com tampas pretas, as imagens estão no formato *RAW* com resolução de 3280 $\times$ 2464 *pixels* e um tamanho de 14 Mb. Esse banco foi criado para a realização de testes que visam minimizar o erro do sistema de VM, pois ao aumentar a quantidade de *pixels* que representam a mesma cena na imagem, significa diminuir a resolução milímetro/*pixel*.

Por fim, criou-se o banco de imagens para testes de repetibilidade. O teste de repetibilidade avalia o grau de variabilidade de um sistema dadas as mesmas condições de entrada, em outras palavras, o objetivo é avaliar se o sistema de VM apresenta em sua saída grandes variações na detecção e localização dos objetos diante de um conjunto de imagens capturadas sob as mesmas condições.

Foram capturadas 25 imagens sequencialmente para minimizar as diferenças de iluminação entre uma imagem e outra.

Para esse teste, desenvolveu-se um protótipo impresso em 3D cuja base tinha as mesmas dimensões da plataforma detectada e sobre esta base foram impressos nove cilindros com as mesmas dimensões dos objetos alvo de captura do sistema de VM. Essa peça é posicionada acima da plataforma que fica abaixo da câmera e então as imagens sequenciais são capturadas.

4.3.2 Algoritmo de Calibração da Câmera

Neste trabalho foram implementados dois algoritmos, sendo o primeiro responsável pela calibração da câmera e o cálculo da matriz de homografia, enquanto o segundo é destinado à detecção de objetos posicionados sobre a plataforma de trabalho. Na Figura 21 é ilustrado o fluxograma adotado para a implementação do algoritmo de calibração.

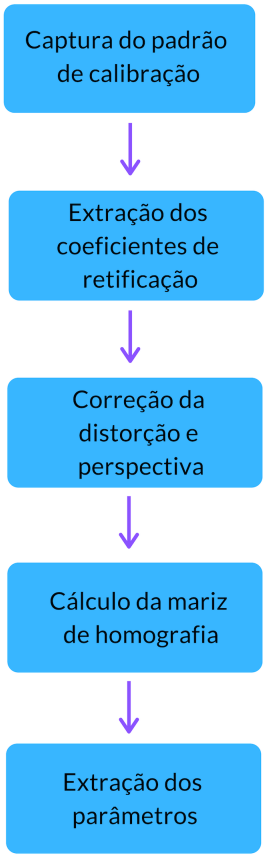
A calibração é uma atividade que deve ser realizada periodicamente ou quando houver problemas de deslocamento da câmera em relação a plataforma de trabalho. Desse modo, uma vez adquiridos os coeficientes de calibração, não será mais necessário que essa etapa seja novamente executada.

A primeira etapa consiste em carregar o banco de imagens de calibração, um conjunto com 30 imagens com uma resolução de 800×480 *pixels* com o formato PNG (*Portable Network Graphics*). Após a leitura das imagens, duas matrizes responsáveis por armazenar as coordenadas planares dos pontos de referência e suas respectivas projeções, são inicializadas.

Para detectar as projeções dos pontos de referência na imagem, aplica-se a função nativa da biblioteca OpenCV, que adota uma estratégia baseada em detecção de cantos por meio de sucessivas avaliações da imagem. Esta função recebe como parâmetros iniciais as imagens contendo o padrão de calibração e o número de pontos de referência, que nesse caso são 54 pontos, pois o padrão de calibração possui 9 colunas por 6 linhas.

Inicialmente o algoritmo a transforma a imagem que encontra-se no padrão de cor RGB para uma imagem em tons de cinza e em seguida realiza uma binarização adaptativa seguido de uma extração de contornos, ajuste de polígonos para identificar quais dos contornos são quadriláteros e por ultimo retorna em sua saída um conjunto de pontos detectados. Para melhorar a precisão é empregado um algoritmo de *subpixel* capaz de informar as coordenadas dos pontos detectados na

Figura 21: Fluxograma do Algoritmo Proposto.



Fonte: Elaborado pela autora.

imagem com uma resolução de três casas decimais.

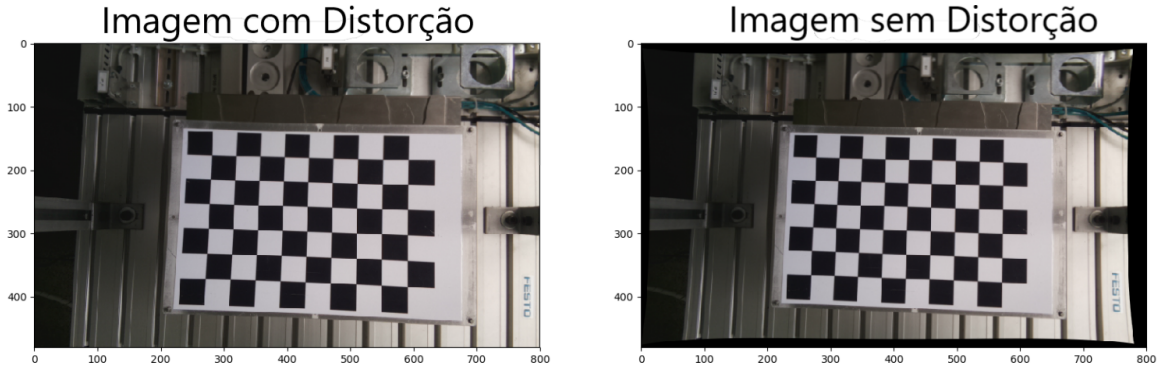
O tamanho da aresta dos quadriláteros é 25,4 mm, ou seja, a matriz que armazena as coordenadas dos pontos de referência é do tipo $P_w = [\ (0\ 0), (25,4\ 0), (50,8\ 0)...]$, logo, com as duas matrizes de coordenadas é possível identificar os parâmetros intrínsecos, extrínseco e coeficientes de distorção utilizando o modelo matemático câmera *pinhole* para estimar esses valores. Os parâmetros intrínsecos e coeficientes de distorção são fixos e deverão ser alterados caso a câmera seja trocada, enquanto os parâmetros extrínsecos são diferentes para cada orientação distinta do padrão de calibração em relação à câmera fixa. A Tabela 2 descreve os parâmetros fixos da câmera.

Tabela 2: Tabela de Parâmetros Intrínsecos e Coeficientes de Distorção.

Matriz Intrínseca		***Coeficientes de Distorção***	
f_x	632.3497	k_1	0.2015
f_y	633.4991	k_2	-0.3939
c_x	409.3057	k_3	-0.0405
c_y	238.8253	p_1	-0.0014
τ	0	p_2	-0.0024

Utilizando os parâmetros da Tabela 2 é implementada a etapa de retificação da imagem, que consiste na correção da distorção radial e tangencial das lentes e em seguida correção da distorção perspectiva. Nesta retificação, acontece o remapeamento dos *pixels* da imagem capturada em função dos coeficientes de distorção e da matriz intrínseca. O resultado desta etapa é ilustrado na Figura 22.

Figura 22: Correção da distorção.



Fonte: Elaborado pela autora.

De acordo com a imagem resultante, conclui-se que a distorção radial positiva atuava predominantemente na imagem, fazendo com que as distâncias entre as coordenadas do centro da imagem fossem maiores do que as distâncias na periferia da imagem.

A etapa de correção da distorção perspectiva é responsável por alinhar a representação da plataforma de trabalho na imagem, de modo que preserve o paralelismo nesta região da imagem.

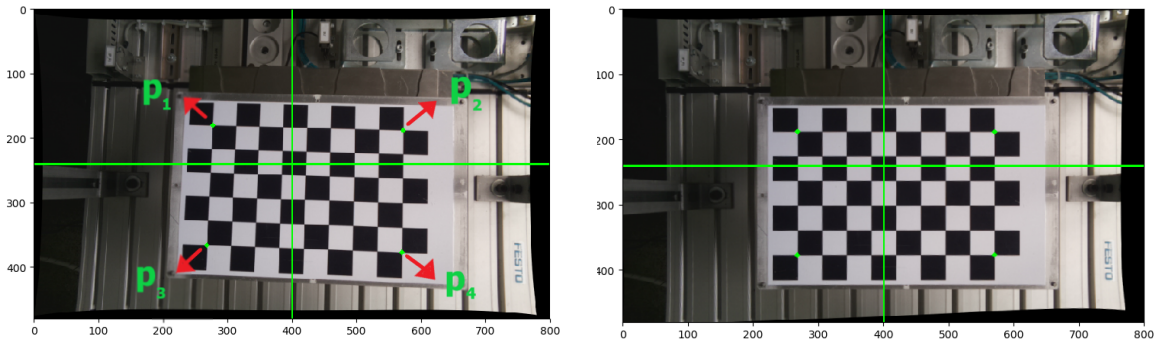
Neste processo são necessários 4 pontos na imagem de entrada e 4 pontos na imagem de saída, dos quais 3 não podem ser colineares. Para facilitar a detecção dos 4 pontos na imagem de entrada, foi utilizado o padrão de calibração que ficou posicionado sobre a plataforma de trabalho, em seguida, foi realizada a captura da imagem, a correção da distorção radial e tangencial para que o algoritmo pudesse detectar os pontos de referencia localizados nos extremos da plataforma conforme ilustra a Figura 23(a).

As coordenadas dos 4 pontos da imagem de saída são definidos diante da análise dos valores máximos e mínimos das coordenadas. Sejam $p_1 = (x_1, y_1),\ p_2 = (x_2, y_2),\ p_3 = (x_3, y_3)$ e $p_4 = (x_4, y_4)$ coordenadas da imagem e entrada, $min_x = min(x_1, x_2, x_3, x_4),\ min_y = min(y_1, y_2, y_3, y_4)$ os valores mínimos e $max_x = max(x_1, x_2, x_3, x_4),\ max_y = max(y_1, y_2, y_3, y_4)$ os valores máximos, é possível estimar os pontos da imagem da saída, de modo que os vértices formem um retângulo, logo, $p'_1 = (min_x, min_y), p'_2 = (max_x, min_y), p'_3 = (min_x, max_y)$ e $p'_4 = (max_x, max_y)$.

Com os pontos da imagem de entrada e da imagem de saída conhecidos, é possível estimar os coeficientes da matriz de homografia. Em seguida, com os coeficientes de homografia é possível aplicar a técnica de $warping$ para gerar a nova imagem de saída ilustrada na Figura 23(b).

Após o processo de retificação, as linhas verticais estão perpendiculares as linhas horizontais da imagem e foi possível recuperar o paralelismo. Com a imagem retificada inicia-se o processo para estimar os parâmetros de uma nova matriz de homografia que relaciona os pontos do SCM com as suas projeções na imagem retificada. A biblioteca OpenCV disponibiliza funções específicas para o cálculo da homografia com base na matemática descrita na Seção 2.3.

Figura 23: Correção da Distorção Perspectiva. (a) Imagem com Distorção Perspectiva. (b) Imagem sem Distorção Perspectiva.



(a) (b)

Fonte: Elaborado pela Autora.

4.3.3 Algoritmo de Detecção dos Objetos

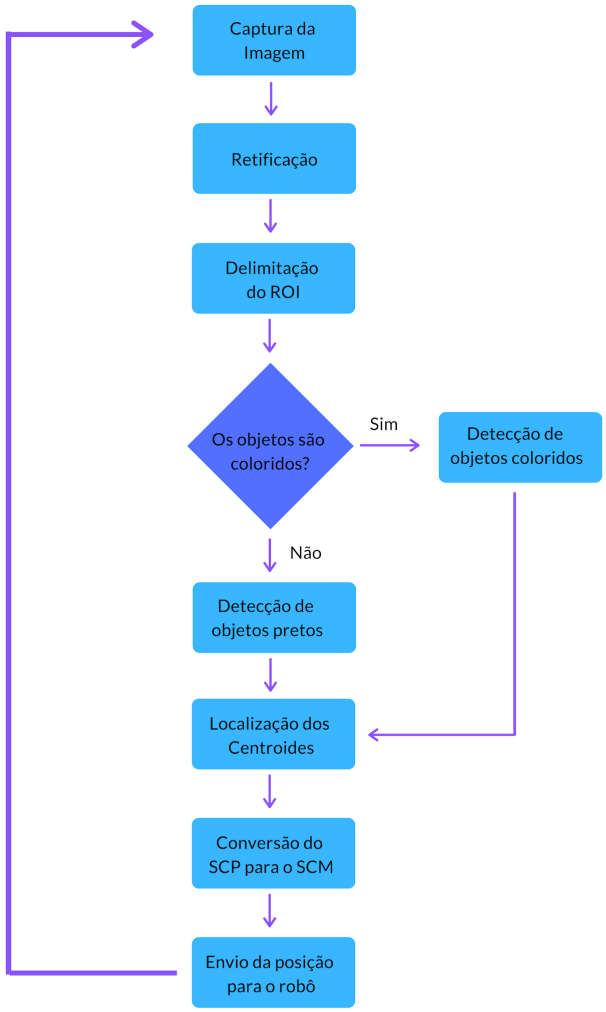
O algoritmo de detecção de objetos realiza desde a captura da imagem até a comunicação com o controlador do manipulador robótico industrial. A sequência do algoritmo de detecção está no fluxograma ilustrado na Figura 24.

Inicialmente, os cilindros são posicionados aleatoriamente dentro da região da plataforma. Estes possuem a mesma altura e podem ter tampas nas cores preto, amarelo, verde, vermelho, azul claro e azul escuro. Embora o algoritmo detecte todos os objetos-alvo presentes na plataforma, apenas uma coordenada é enviada por vez ao controlador do manipulador robótico.

Após a captura inicial da imagem, uma nova imagem só pode ser capturada após a confirmação de que a informação foi recebida pelo controlador robótico, indicando que o processo de armazenamento do cilindro foi realizado. Essa medida foi adotada para evitar possíveis oclusões na imagem causadas pelo ato de captura do manipulador robótico no momento em que o braço está sobre a plataforma.

Depois que os objetos estão posicionados sobre a plataforma, é o momento da realização da retificação da imagem. Essa etapa é uma das mais relevantes, pois é nela que acontece a remoção da distorção causada pelas lentes e pela perspectiva da câmera em relação a cena. Com os coefi-

Figura 24: Fluxograma do Algoritmo de Detecção.



Fonte: Elaborado pela Autora.

cientes extraídos da etapa de calibração, é realizada a retificação da imagem por meio de funções específicas contidas no algoritmo de detecção de objetos.

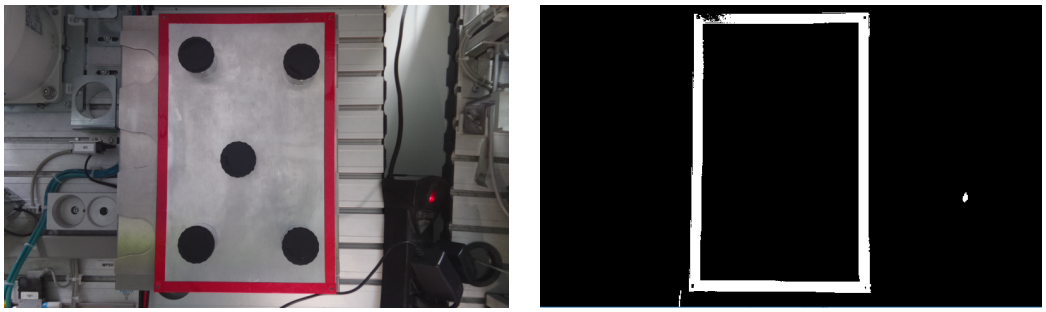
O próximo passo é a detecção da plataforma de trabalho a partir da delimitação presente na borda. Limitar uma região de interesse na imagem possibilita o direcionamento do processamento, diminuindo a identificação de falsos objetos e consequentemente otimizando a velocidade de processamento.

A mudança do padrão RGB para o padrão HSV possibilita a detecção da cor vermelha

presente na borda da plataforma, essa etapa é ilustrada na Figura 25 (b). A escolha da delimitação está relacionada a delimitação de uma região de interesse na imagem, entretanto, essa marcação também pode ser utilizada como sinalização ao usuário, indicando que aquela é a região de atuação do sistema de visão e do manipulador robótico.

A detecção da borda da plataforma permitiu a criação da máscara (Figura 25 (c)), cuja área compreende toda a região metálica da plataforma. O intuito da utilização dessa máscara é a eliminação da região além da plataforma que pode gerar a detecção de falsos-positivos. O resultado da multiplicação da máscara pela imagem é ilustrado na Figura 25 (d).

Figura 25: Etapas do Algoritmo. (a) Imagem capturada (b) Detecção do marcador. (c) Máscara para eliminação do *background*. (d) Imagem em tons de cinza.



(a) (b)

(c) (d)

Fonte: Elaborado pela Autora.

Caso os cilindros que estão sobre a plataforma tenham a tampa azul, verde, amarela ou vermelha, é utilizado o algoritmo de segmentação de cor, caso contrário, utiliza-se o algoritmo de segmentação para os cilindros de tampa preta. Em ambos os casos as imagens são binarizadas,

visando o realce da região dos objetos de interesse.

A detecção dos objetos é realizada a partir da imagem binarizada e a localização dos centroides pelo cálculo dos momentos da imagem, é realizada a conversão das coordenadas que encontram-se no SCP para o SCM. Essa conversão é feita através da matriz de homografia, cujos coeficientes foram extraídos do algoritmo de calibração.

Finalmente, inicia-se a comunicação do RPi com o controlador do manipulador robótico, utilizando o protocolo *Data Link*. Embora o algoritmo seja capaz de detectar vários cilindros em uma única imagem, apenas uma coordenada é enviada por vez ao controlador. Em seguida dá-se inicio ao processo de captura e armazenamento deste cilindro. O ciclo do algoritmo de detecção se repete até que a plataforma esteja vazia.

4.3.4 Testes de Segmentação

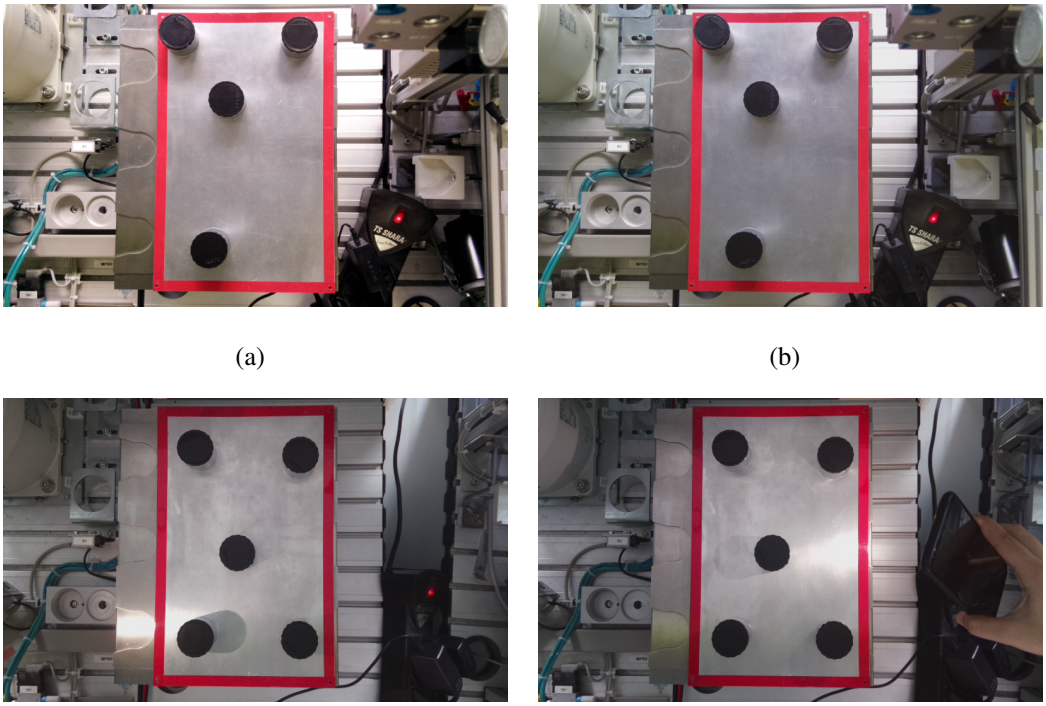
Neste trabalho é proposto um sistema de VM capaz de detectar objetos cilíndricos de mesmo tamanho e diâmetro presentes em uma cena, no entanto, esta detecção é sensível às condições de iluminação do ambiente que afetam diretamente o algoritmo de segmentação da imagem.

Devido as condições luminosas não-controladas é imprescindível a utilização de um algoritmo de segmentação flexível capaz de ressaltar os objetos-alvo de diferentes cores em relação aos demais elementos da cena. Foram capturadas imagens em diferentes horários para verificar a capacidade dos algoritmos testados na segmentação da superfície dos cilindros.

A metodologia utilizada para realização dos testes de segmentação consiste inicialmente na captura de dois conjuntos de imagens. O primeiro conjunto é composto por trinta imagens contendo os cilindros com a superfície preta, dispostos em diferentes posições e em diferentes condições de iluminação. Na sequência foram testados os algoritmos de limiarização global, limiarização adaptativa e o *K-means*.

A Figura 26 ilustra apenas quatro das imagens capturadas pelo sistema de VM ao longo do dia em diferentes condições de iluminação as quais o sistema está exposto.

Figura 26: Efeitos da Mudança de Luminosidade nas Imagens Capturadas pelo sistema de VM.



(a) (b)

(c) (d)

Fonte: Elaborado pela autora.

A Figura 26 (a) foi capturada no período da manhã, momento em que o sistema de VM está recebendo iluminação solar através da janela localizada próxima a plataforma. A luz incide na parte superior da imagem com maior evidência. Na Figura 26 (b) a plataforma está recebendo uma menor incidência luminosa, no entanto, é possível observar que os cilindros estão projetando uma sombra que interfere significativamente no algoritmo de detecção.

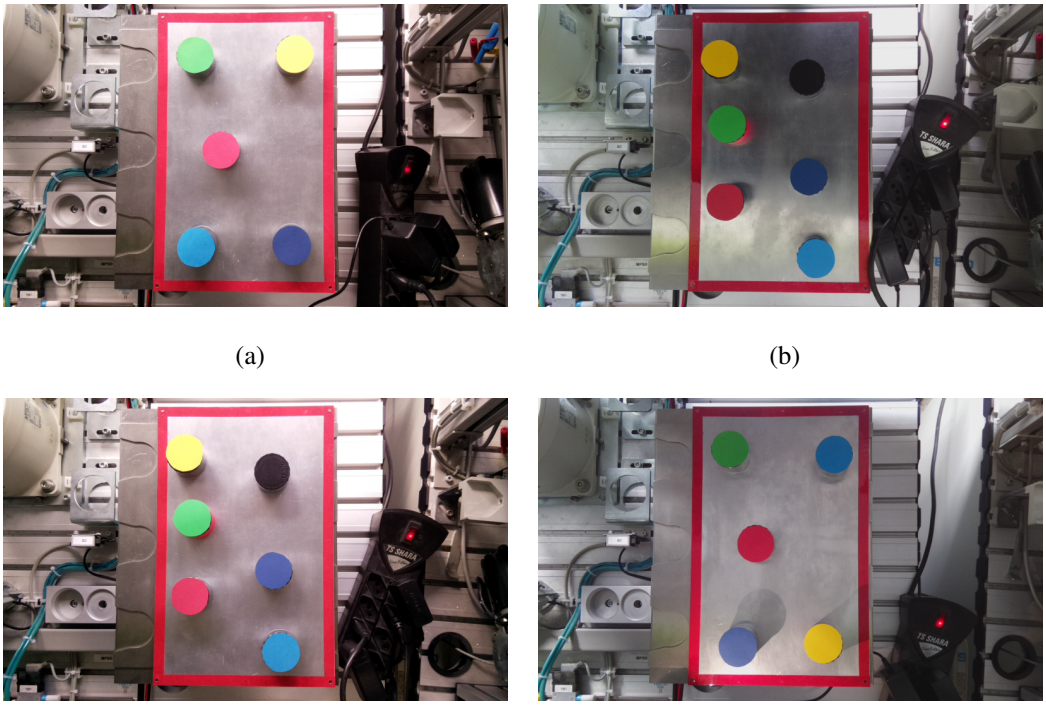
A Figura 26 (c) ilustra a projeção da sombra do cilindro do canto inferior esquerdo. Esse tipo de situação pode fazer com que o algoritmo detecte a sombra, gerando assim um falso-positivo. A Figura 26 (d) ilustra o momento em que o ambiente no qual o sistema de VM está inserido está sendo afetado por uma iluminação direta que incide na plataforma metálica.

No segundo conjunto de imagens estão contidos cilindros com as superfícies nas cores vermelho, azul ,amarelo, verde e preto. Para detectar as imagens com cilindros de superfície colorida foram capturadas mais trinta imagens que são a entrada do algoritmos de segmentação de cor pelo

padrão de cores HSV. A Figura 27 ilustra quatro exemplos. As Figuras 27 (a) e (c) estão são as mais afetadas pela iluminação externa, porém em pontos distintos da plataforma, já as Figuras 27 (b) e (d) têm uma maior presença de sombras nas imagens.

A segmentação por cor foi implementada da seguinte maneira: foi definido o range de cada cor nos canais H, S e V, posteriormente foi criada a máscara responsável pela segmentação da região correspondente a cor detectada. A imagem resultante deste processo é uma imagem binarizada cuja região da cor detectada recebe o valor de intensidade 255, correspondente a cor branca e o resto da imagem recebe o valor de intensidade 0, correspondente a cor preta. O processo descrito foi realizado para todas as respectivas cores detectadas e as imagens resultantes foram sobrepostas por meio da operação aritmética de adição.

Figura 27: Imagens capturadas pelo sistema de VM dos cilindros nas cores amarelo, vermelho, verde, azul e preto.



(a) (b)

(c) (d)

Fonte: Elaborado pela autora.

4.3.5 Análise do Erro e Repetibilidade

A repetibilidade de um sistema de medição é dada pela faixa de valores simétricos em torno do valor médio cuja taxa de erro aleatório é esperada para uma dada probabilidade. Para calcular a repetibilidade de um sistema é necessário que as medições sejam realizadas sob as mesmas condições, por exemplo, mesma iluminação e mesmo objeto medido. As repetições devem ser realizadas em um curto período de tempo para que as variações externas e probabilísticas interfiram o mínimo possível (Albertazzi and de Sousa, 2008).

Para estimar a repetibilidade do sistema de VM, aplicou-se a teoria de William Sealey Gsset conhecida na literatura como distribuição t de *Student*. A curva de distribuição t assemelha-se a curva de distribuição normal, no entanto, apresenta bordas mais alargadas que refletem maior variabilidade, característica de amostras pequenas. Além disso, a função tem um parâmetro que é a quantidade de graus de liberdade, quanto maior este parâmetro, mais semelhante à curva normal será a curva t.

A tabela ilustrada na Figura 28 ilustra a relação entre o grau de confiabilidade (colunas) e o valor de graus de liberdade (linhas).

A interseção entre o grau de confiabilidade e o número de medições é o que define o coeficiente t. Para a avaliação da repetibilidade do sistema de VM, foram capturadas 25 imagens em sequência, sob as mesmas condições, portanto, $t = 1,708$.

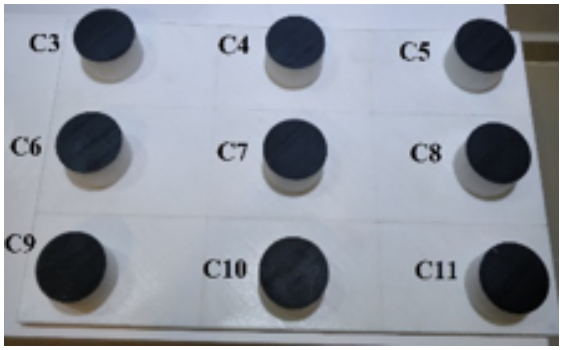
Neste ensaio adotou-se um protótipo que possui as mesmas dimensões da plataforma do sistema de VM, confeccionado em impressora 3D, contento nove cilindros fixos sobre sua superfície com as mesmas dimensões dos cilindros que são capturados pelo sistema de VM, assim como ilustra a Figura 29.

Figura 28: Tabela de Distribuição t de *Student*.

Unicaudal	75%	80%	85%	90%	95%	97,5%	99%
Bicaudal	50%	60%	70%	80%	90%	95%	98%
1	1,000	1,376	1,963	3,078	6,314	12,71	31,82
2	0,816	1,061	1,386	1,886	2,920	4,303	6,965
3	0,765	0,978	1,250	1,638	2,353	3,182	4,541
4	0,741	0,941	1,190	1,533	2,132	2,776	3,747
5	0,727	0,920	1,156	1,476	2,015	2,571	3,365
6	0,718	0,906	1,134	1,440	1,943	2,447	3,143
7	0,711	0,896	1,119	1,415	1,895	2,365	2,998
8	0,706	0,889	1,108	1,397	1,860	2,306	2,896
9	0,703	0,883	1,100	1,383	1,833	2,262	2,821
10	0,700	0,879	1,093	1,372	1,812	2,228	2,764
11	0,697	0,876	1,088	1,363	1,796	2,201	2,718
12	0,695	0,873	1,083	1,356	1,782	2,179	2,681
13	0,694	0,870	1,079	1,350	1,771	2,160	2,650
14	0,692	0,868	1,076	1,345	1,761	2,145	2,624
15	0,691	0,866	1,074	1,341	1,753	2,131	2,602
16	0,690	0,865	1,071	1,337	1,746	2,120	2,583
17	0,689	0,863	1,069	1,333	1,740	2,110	2,567
18	0,688	0,862	1,067	1,330	1,734	2,101	2,552
19	0,688	0,861	1,066	1,328	1,729	2,093	2,539
20	0,687	0,860	1,064	1,325	1,725	2,086	2,528
21	0,686	0,859	1,063	1,323	1,721	2,080	2,518
22	0,686	0,858	1,061	1,321	1,717	2,074	2,508
23	0,685	0,858	1,060	1,319	1,714	2,069	2,500
24	0,685	0,857	1,059	1,318	1,711	2,064	2,492
25	0,684	0,856	1,058	1,316	1,708	2,060	2,485
26	0,684	0,856	1,058	1,315	1,706	2,056	2,479

Fonte: Myers (2009).

Figura 29: Peça utilizada no experimento.



Fonte: Elaborado pela autora.

A função desta peça é fazer com que seja possível adotar um Valor Verdadeiro Convencional (VVC) e compara-lo com as medições realizadas pelo sistema de VM para que seja possível estimar o erro e a repetibilidade do sistema.

A primeira etapa do ensaio consiste em identificar os VVCs, caracterizado pelos centros das superfícies pretas da peça, portanto foram realizadas as medições dos centros da superfície preta da peça pela Máquina de Medição de Coordenadas (MMC) Mitutoyo modelo CRYSTA-Apex S7106 de alta confiabilidade e exatidão de $1,7\mu m$, localizada no laboratório de Engenharia de Precisão da UFPB.

As distância do cilindro de referência (C3) foram calculadas em relação aos demais centros. Esses valores foram comparados com os valores de 25 detecções sequenciais realizadas pelo sistema de VM proposto.

A repetibilidade foi obtida através da multiplicação do desvio padrão (σ) das detecções pelo coeficiente de *student* (t) para uma probabilidade de 95% .

$$\sigma = \sqrt{\frac{\sum_{i=1}^{n}(I_i - \bar{I})^2}{n-1}}, \tag{28}$$

em que I_i é a i-ésima detecção, $\bar{I}$ é a média das detecções e n é a quantidade de detecções. Uma vez obtido o valor do desvio padrão, é possível calcular a repetibilidade por $R_e = \sigma t$, em que t é o coeficiente *Student*. Cada distância composta pelas coordenadas (x,y) em relação a origem (coordenada do centro C3) terá um valor de repetibilidade associado.

Definido o valor de repetibilidade, encontram-se os valores dos Limites Inferiores de Controle e Limites Superiores de Controle, dado por $R_e - \bar{I} < \bar{I} < R_e + \bar{I}$. As medições devem estar contidas neste intervalo.

O cálculo do valor do Erro é uma medida que serve para identificar a diferença entre a posição estimada entre sistema de VM e a posição fornecida pela MMC. Foram obtidos 9 valores de erro, um para cada distância, calculados de acordo com a Equação 29, correspondente a cada centro da peça.

$$E_i = P(x_i, y_i) - VVC(x_i, y_i) \tag{29}$$

Quanto menor o valor de E, menor também é a divergência da medição do sistema de VM proposto com o VVC, indicando uma maior probabilidade na captura correta das peças pelo manipulador robótico industrial.

5 Resultados

Esta seção apresenta os resultados acerca dos algoritmos que compõem o sistema de VM bem como os testes experimentais realizados a fim de mensurar o erro e repetibilidade do sistema proposto em relação as medições oriundas de uma Máquina de Medição de Coordenadas (MMC) de alta exatidão e confiabilidade.

5.1 Resultados Experimentais

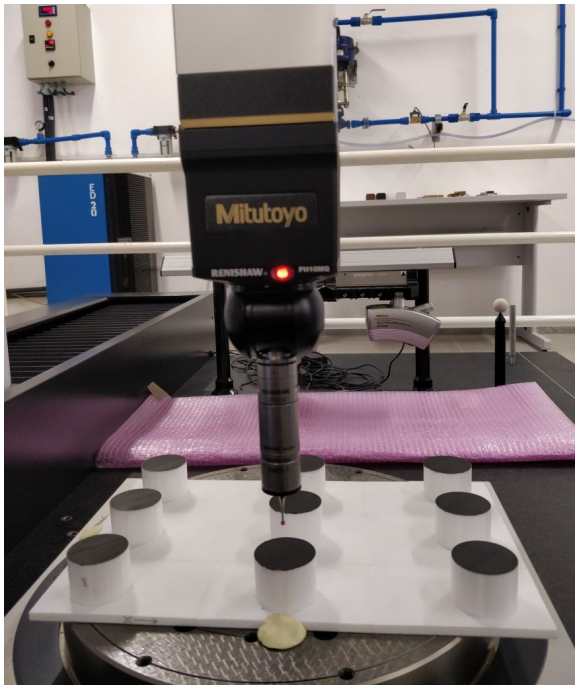
Esta seção abrange os ensaios realizados com o sistema de VM no intuito de averiguar seu comportamento diante da captura automática de peça, bem como comparar os resultados oriundos do sistema de VM proposto com os valores medidos por uma MMC modelo Mitutoyo CRYSTA-Apex S7106, dotada com exatidão de $1,7\mu m$.

Para este ensaio adotou-se um protótipo que possui as mesmas dimensões da plataforma do sistema de visão, confeccionado em impressora 3D, contento nove cilindros fixos sobre sua superfície.

Durante a medição da peça protótipo na MMC adotou-se como Valor Verdadeiro Convencional (VVC) as coordenadas das centroides dos cilindros obtidos por esta máquina. Neste processo de mensuração, convencionou-se um dos centroides como a origem do plano tangente à superfície dos cilindros. A Fig. 30 ilustra o processo de medição do protótipo.

A avaliação do erro entre o sistema de VM e a MMC aconteceu da seguinte maneira: foi calculada a distância entre os centroides de cada cilindro pela MMC, sempre em relação ao centroide do cilindro C3. Em seguida, o protótipo foi colocado sobre a plataforma de trabalho, feita a captura da imagem e identificação dos centroides pelo algoritmo de VM. O erro é estabelecido nas direções dos eixos x e y diante da diferença entre os valores obtidos pela MMC e os valores estimados pelo sistema de VM. Todas os valores de coordenadas mencionados bem como o erro estão descritos na Tabela 3. A Fig. 31 ilustra graficamente as coordenadas obtidas pela MMC, aqui consideradas como VVC, e o valores obtidos pelo sistema de VM proposto.

Figura 30: Medição por Coordenadas na Máquina Mitutoyo

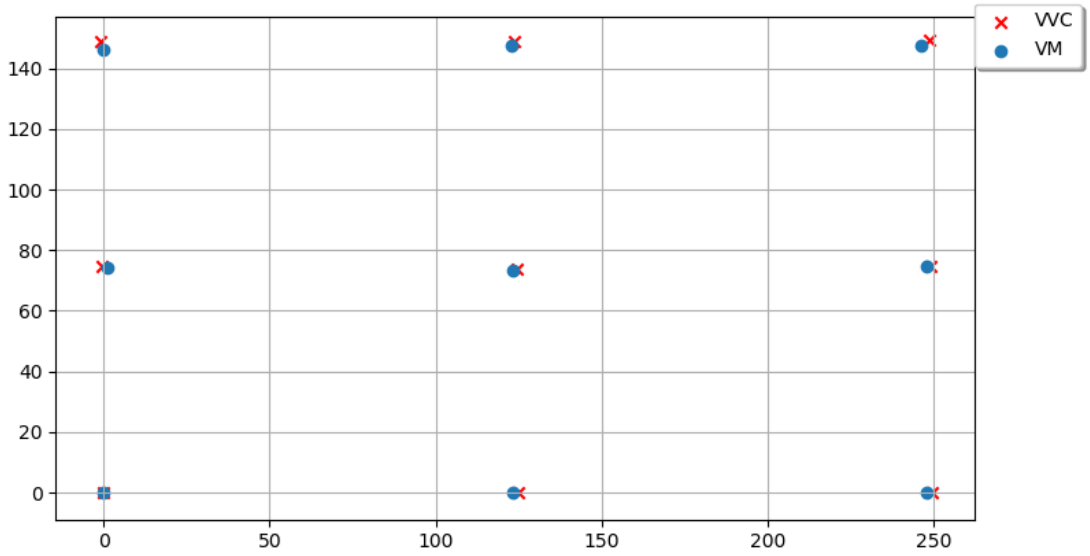


Fonte: Elaborado pela autora.

Tabela 3: Resultado Experimental e Valor do Erro.

Distâncias (mm)	**VM (mm)**		**MMC (mm)**		**Erro(mm)**	
	x	y	x	y	x	y
C3-C4	123,190	0,045	124,751	-0,063	1,561	-0,108
C3-C5	247,700	0,090	249,521	0,008	1,821	-0,082
C3-C6	1,237	74,389	-0.501	74,552	-1,738	0,163
C3-C7	123,128	73,096	124,647	73,967	1,519	0,871
C3-C8	247,694	74,455	248,968	74,698	1,274	0,243
C3-C9	-0,229	146,184	-0,746	148,834	-0,517	2,65
C3-C10	123,065	147,54	123,883	148,701	0,818	1,161
C3-C11	246,33	147,567	248,518	149,412	2,188	1,845

Figura 31: Coordenadas do Sistema Proposto e da MMC.



Fonte: Elaborado pela autora.

O teste seguinte visa investigar se um dos fatores que contribuem para o aumento do erro está relacionada a resolução da imagem padrão da PiCamera (800×600) *pixels*.

O teste de repetibilidade foi inviabilizado, pois o valor do desvio padrão no conjunto de imagens de resolução 800×480 *pixels*, é zero. Portanto, utilizar imagens de maior resolução pode viabilizar o cálculo do valor de repetibilidade, uma vez que terão mais *pixels* representando a mesma área do objeto.

Inicialmente foram realizadas capturas de 25 imagens do tipo *RAW*, com a resolução máxima da Picamera (3280×2464 *pixels*), em seguida foram aplicados os algoritmos de segmentação e calculou-se a repetibilidade do sistema de VM bem como o Limite inferior de Controle (LIC), Limite Superior de Controle e o novo valor de Erro. A Tabela 4 apresenta um exemplo desses resultados calculados apenas para uma coordenada.

Tabela 4: Resultados do sistema de VM

Algoritmos	**Repetibilidade**	**LIC**	**LSC**	**VVC**	**Erro**
Limiarização Global	0.260	124.334	124.854	124.751	0,157
Limiarização Global + CLAHE	0.713	124.109	125.536	124.751	-0,071
Limiarização Adaptativa	0.305	124.282	124.893	124.751	0,164
Limiarização Adaptativa + CLAHE	0.583	123.689	124.856	124.751	0,478
K-Means	0.897	124.694	125.689	124.751	-0,041

Nesse teste foram avaliadas a detecção dos cilindro de superfície preta utilizando os cinco algoritmos de segmentação, sendo dois deles de limiarização global, dois de limiarização adaptativa e um de agrupamento em *clusters*. Dentre estes algoritmos, o sistema de VM apresentou o menor valor de repetibilidade quando utiliza a segmentação por limiarização global, ou seja, uma menor variação na medição das coordenadas sob as mesmas condições. Ao associar essa técnica de limiarização à técnica CLAHE, neste caso utilizada como etapa de pré-processamento, notou-se um aumento significativo no valor de repetibilidade. Uma das razões pelo aumento dessa variável pode estar relacionada ao fato da CLAHE amplificar ruídos na imagem. Por outro lado, a limiarização global nesse teste apresentou um valor de erro alto em relação aos demais algoritmos testados.

O algoritmo de ML, K-means, associado a uma imagem de alta resolução, apresentou um consumo de processamento maior, tornando a compilação do programa mais lenta que o esperado. Na avaliação da repetibilidade do sistema de VM utilizando este algoritmo, é possível observar que o K-means obteve o maior valor de repetibilidade, indicando que o algoritmo apresenta maiores variabilidades quando utilizado sob as mesmas condições.

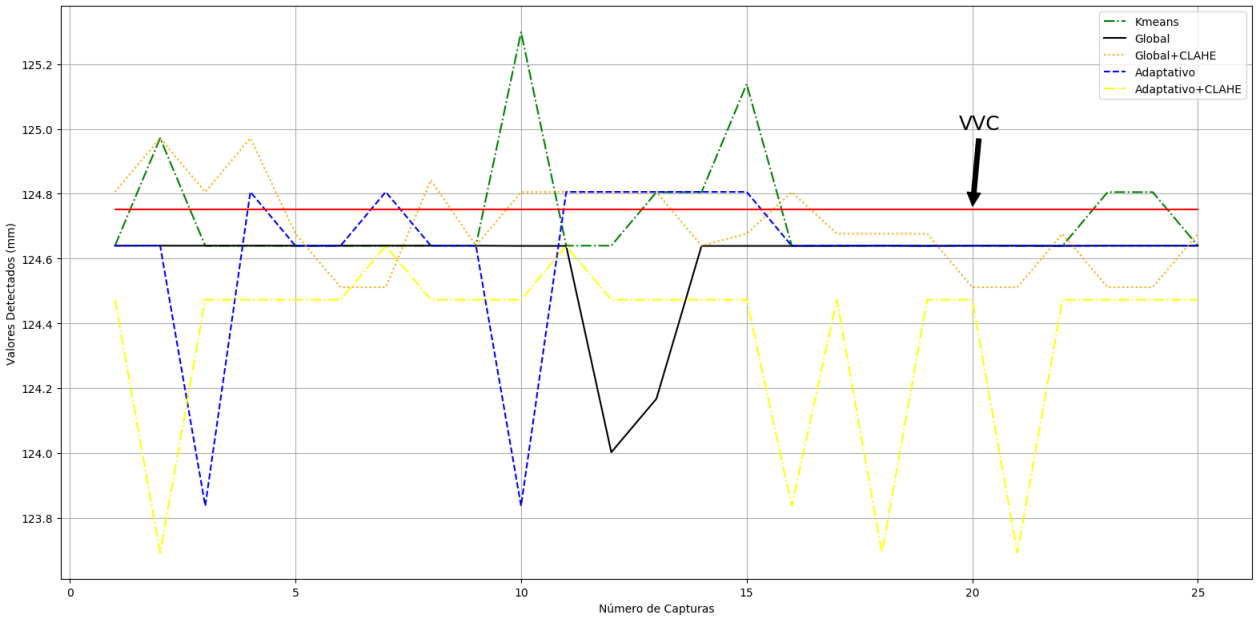
Por ser um algoritmo de inicialização randômica, o K-means pode realizar o agrupamento dos *clusters* de diferentes formas, o que pode impactar diretamente na detecção da coordenada final. No entanto, o baixo valor do erro em relação aos demais algoritmos testados indica que por mais que se obtenha uma alta variabilidade com este algoritmo, os valores resultantes estão mais próximos do VVC.

A Figura 32 ilustra as coordenadas medidas pelos cinco algoritmos testados ao longo das 25 imagens capturadas. A curva verde representa as coordenadas detectadas pelo algoritmo K-

means. No gráfico são observados picos em torno do VVC, os picos oscilam majoritariamente com valores superiores ao VVC, no entanto, é possível também identificar trechos em que a medição é constante. A curva preta é referente as coordenadas detectadas pelo algoritmo de limiarização global, nesse caso existe uma predominância de regiões constantes e apenas um pico inferior em relação ao VVC, isso significa que esse algoritmo apresenta uma repetibilidade superior ao demais algoritmos testados.

A curva laranja, representa as coordenadas detectadas pelo algoritmo de limiarização Global associado ao CLAHE. A incorporação desta etapa de pré-processamento aumentou o número de picos, consequentemente diminuindo o valor de repetibilidade do algoritmo. O mesmo fenômeno pode ser observado na curva amarela que representa as coordenadas detectadas pelo algoritmo de limiarização adaptativa associado ao CLAHE. Além de ser o algoritmo com o maior valor de erro, como descreve a Tabela 4, é também o algoritmo com o maior pico inferior ao VVC.

Figura 32: Coordenadas do Sistema Proposto e da MMC.



Fonte: Elaborado pela autora.

5.2 Desempenho do RPi

Neste teste foi analisado o desempenho da Raspberry Pi 3 ao executar o algoritmo do sistema VM responsável pela detecção, localização e definição de coordenadas de peças alvo. As variáveis monitoradas foram o percentual de processamento consumido, o percentual de memória utilizada, *frames* (imagens) processados por segundo (FPS) . Os resultados obtidos estão descritos na Tabela 5. Para obter esses valores, instalou-se na RPi a ferramenta Glances, capaz de monitorar em tempo-real informações do dispositivo.

O algoritmo de limiarização global apresentou o maior valor de FPS, sendo assim, o sistema de VM é capaz de processar até 4 frames por segundo com esse algoritmo, no entanto, como nesta aplicação os objetos não estão sendo rastreados, o valor de FPS não tem tanto peso na escolha do melhor algoritmo para esta problemática.

Tabela 5: Performance do Algoritmo na Raspberry Pi

Algoritmos	**FPS**	**Memória (%)**	**CPU(%)**
Limiarização Global	4,07	26,7	4,1
Limiarização Global + CLAHE	3,85	28,3	4,8
Limiarização Adaptativa	3,1	32,4	4,2
Limiarização Adaptativa + CLAHE	2,7	38,9	5,1
K-means	1,2	48,6	15,7
Segmentação HSV	3,2	27,5	4,2

O K-means, por ser um algoritmo de ML despende um maior custo computacional para ser executado, neste caso, o alto consumo de memória e processamento afetam o desempenho do sistema de VM, que deve ser capaz de comunicar-se com o controlador do manipulador robótico rapidamente.

5.3 Captura Aleatória

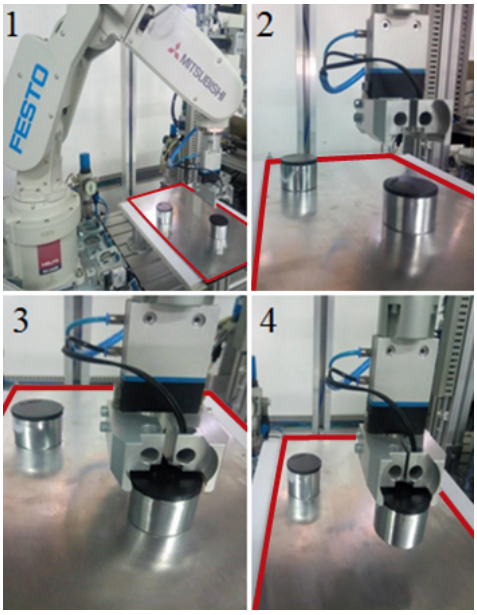
Neste teste foi analisado a captura do objeto cilíndrico quando localizado em posições aleatórias sobre a plataforma de trabalho. A imagem ilustrada na Figura 33 descreve em 4 (quatro)

cenas o movimento do braço robótico ao capturar um cilindro, em uma posição aleatória, usando o sistema de VM proposto.

As cenas 1 (um) e 2 (dois) ressaltam, respectivamente, a ida do robô ao encontro da peça alvo e o seu posicionamento sobre ela. A cena 3 (três) exibe o momento em que a garra robótica está posicionada para capturar a peça alvo. A cena 4 (quatro) exibe a captura do cilindro.

Embora o sistema de VM proposto seja capaz de proporcionar a captura de peças cilíndricas pelo manipulador robótico, é ainda necessário diminuir o valor do erro do sistema para que seja possível diminuir o número de colisões que eventualmente acontecem devido à iluminação não-uniforme que afeta sobretudo a captura nas extremidades da plataforma de trabalho.

Figura 33: Captura de cilindros.



Fonte: Elaborado pela autora.

Para tornar o sistema de VM flexível à captura de objetos com diferentes geometrias é preciso desenvolver um novo modelo de garra que permita a captura de objetos em diferentes orientações.

5.4 Teste de Detecção de Cilindros Pretos

Inicialmente o algoritmo de limiarização global foi utilizado para a segmentação das imagens. O valor de limiar escolhido por métodos empíricos foi o de 50 em uma escala de 0 a 255. O resultado dessa segmentação é ilustrada nas Figuras 34 (a), (c), (e) e (g), enquanto as Figuras 34 (b), (d), (f) e (h) ilustram a detecção dos objetos sinalizada com um contorno azul. O algoritmo apresentou algumas limitações em condições específicas, por exemplo, na limiarização das regiões mais claras da imagem. Nesses casos os valores de intensidade dos *pixels* afetados por essas condições são alterados, prejudicando a detecção e gerando valores falsos-negativos. Esse efeito pode ser visualizado com mais evidencia nas Figuras 34 (a) e (e).

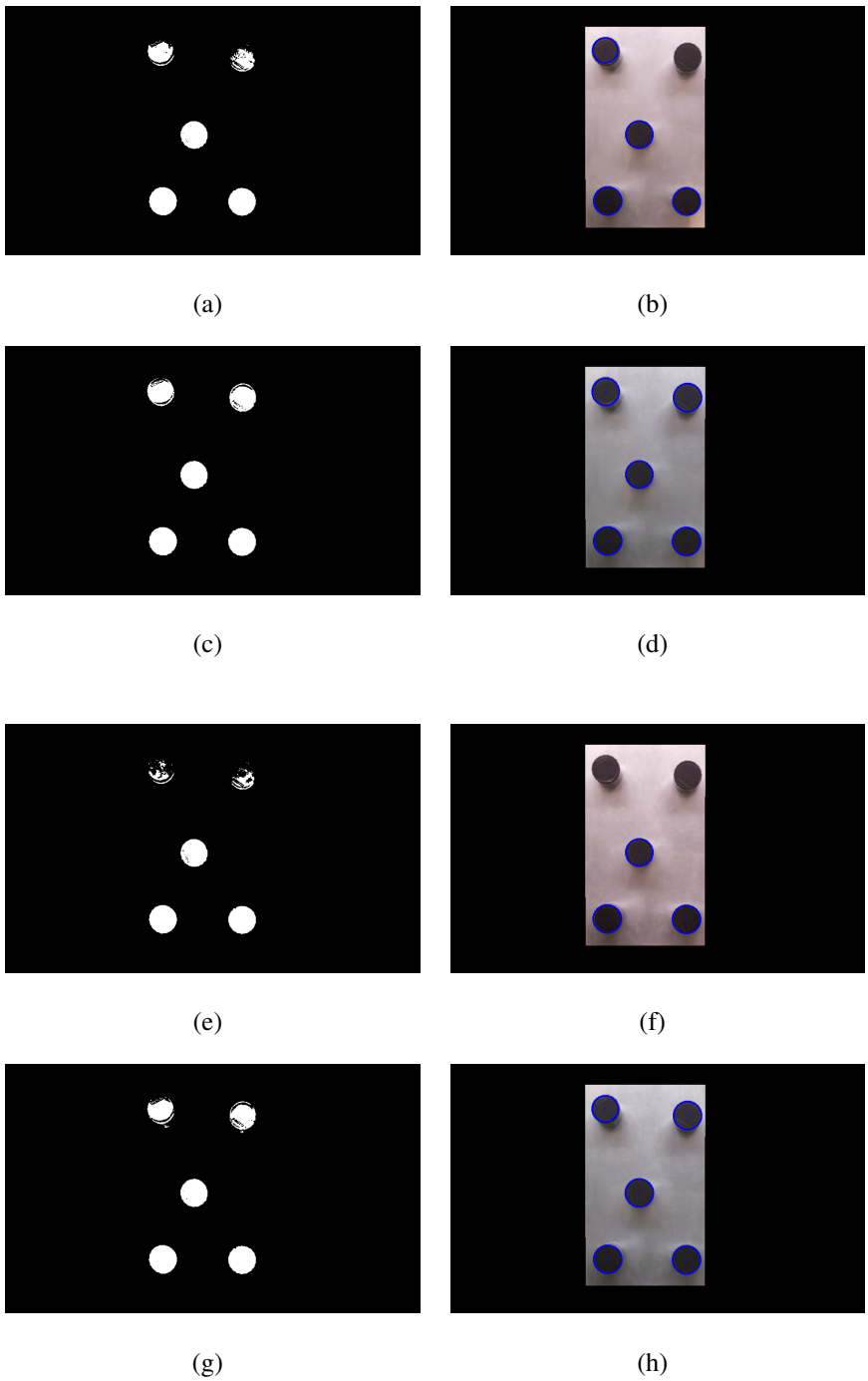
Em regiões da imagem mais escuras e até mesmo em regiões com a presença de sombras, foram onde o algoritmo obteve um melhor desempenho (Figuras 34 (d) e (h)). Com o intuito de melhorar os resultados na detecção dos cilindros pretos, sobretudo em regiões mais claras da imagem, foi realizado o teste da limiarização global associado ao algoritmo CLAHE como etapa de pré-processamento. As imagens resultantes da segmentação estão ilustrados nas Figuras 35 (a), (c), (e) e (g) e nas Figuras 35 (b), (d), (f) e (h) estão os objetos detectados pelo algoritmo. A intenção desse teste foi avaliar a influência do CLAHE nas regiões da imagem com maior incidência de luz.

Nas regiões da imagens mais afetadas pela incidência da iluminação solar externa, o algoritmo não detecta a superfície dos cilindros corretamente. Outro fator que prejudica a detecção é a presença de sombras na plataforma, ocasionando a eventual detecção de falsos-positivos.

A influência do CLAHE, acentuou ainda mais os problemas anteriormente identificados nos testes com o algoritmo de limiarização global. Nas Figuras 34 (e) e (g) a sombra do cilindro é classificada como parte do objeto, comprometendo a exatidão do valor do centroide. Portanto, o uso do CLAHE como uma etapa de pré-processamento, melhorou a performance do algoritmo em relação ao teste anterior, possibilitando a segmentação dos objetos presentes em regiões mais claras, porém, o algoritmo tornou-se mais sensível às sombras geradas pelos objetos. Posteriormente, testada a limiarização adaptativa com e sem o CLAHE como etapa de pré-processamento. Esse algoritmo divide a imagem em regiões 7×7, calcula a média e assume este valor como limiar local da imagem. Nas Figuras 36 (a), (c), (e) e (g) é ilustrada a imagem binarizada pelo algoritmo de limiarização adaptativa e nas Figuras 36 (b), (d), (f) e (h) os objetos detectados pelo algoritmo. Diferentemente

da limiarização global, a limiarização adaptativa realça apenas as bordas dos objetos.

Figura 34: Segmentação pelo Algoritmo de Limiarização Global, T = 55. (a), (c), (e) e (g) Imagens binarizadas. (b), (d), (f) e (h) Objetos detectados pelo algoritmo.



(a) (b)

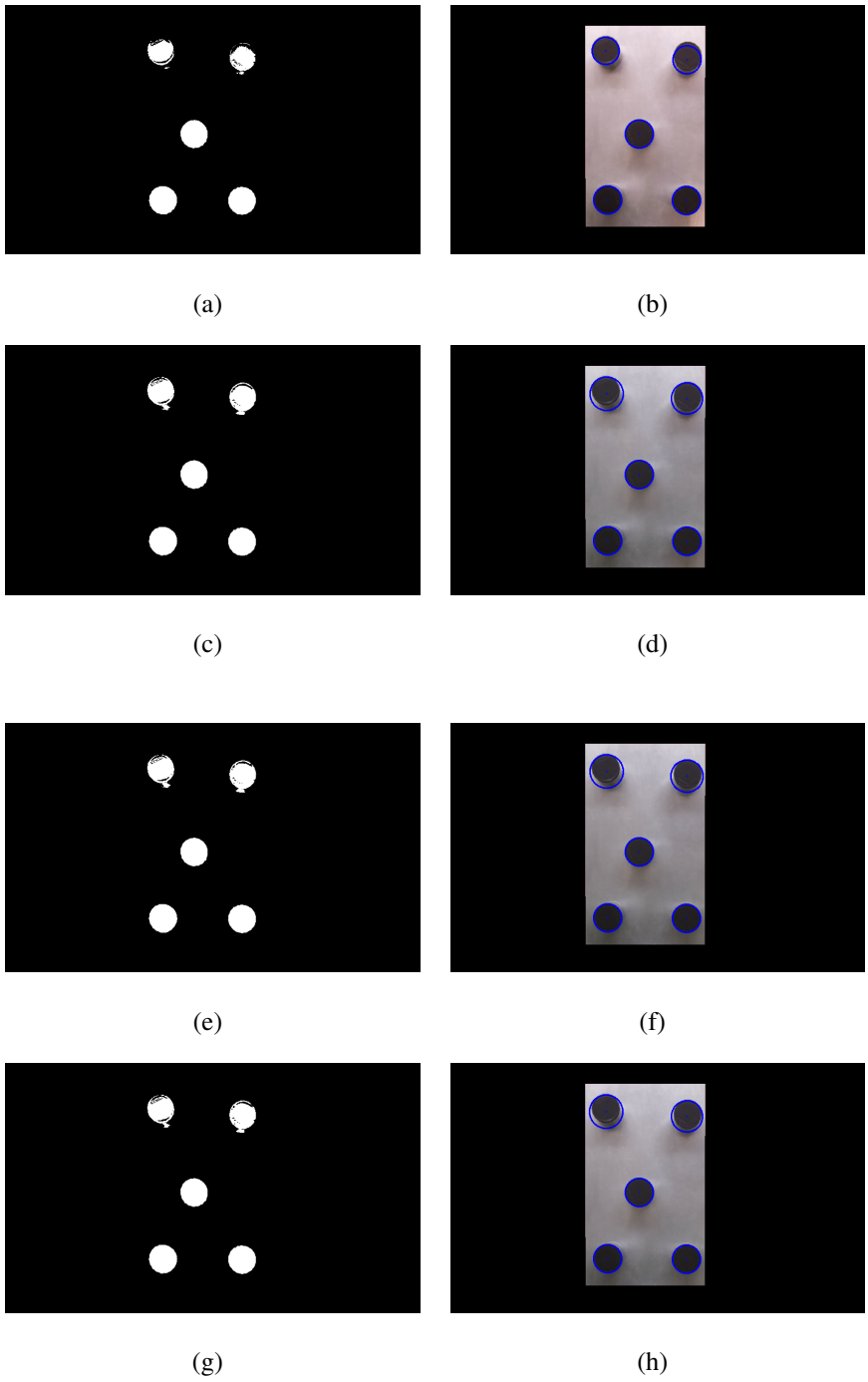
(c) (d)

(e) (f)

(g) (h)

Fonte: Elaborado pela autora.

Figura 35: Segmentação pelo Algoritmo de Limiarização Global e CLAHE. (a), (c), (e) e (g) Imagens binarizadas. (b), (d), (f) e (h) Objetos detectados pelo algoritmo.



(a) (b)

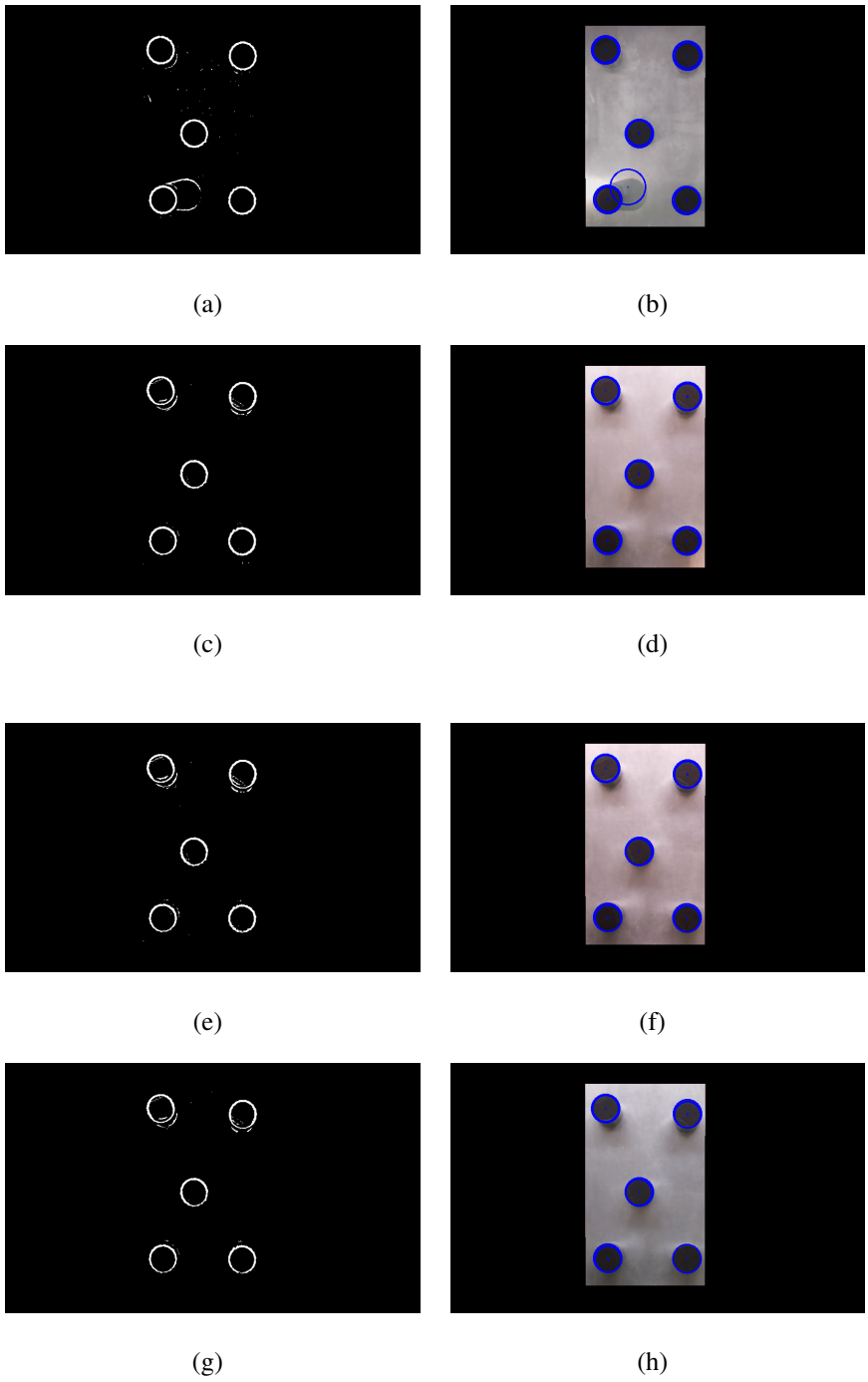
(c) (d)

(e) (f)

(g) (h)

Fonte: Elaborado pela autora.

Figura 36: Segmentação pelo Algoritmo de Limiarização Adaptativa. (a), (c), (e) e (g) Imagens binarizadas. (b), (d), (f) e (h) Objetos detectados pelo algoritmo.



(a) (b)

(c) (d)

(e) (f)

(g) (h)

Fonte: Elaborado pela autora.

Os resultados da Figura 36 ilustram casos de falha do algoritmo, por exemplo, quando a

sombra dos cilindros são realçadas, ou quando o algoritmo detecta a borda da plataforma como um possível objeto-alvo. Na tentativa de diminuir a quantidade de falsos-positivos detectados, aplicou-se o algoritmo de limiarização adaptativa associado ao CLAHE como etapa de pré-processamento.

As Figuras 37 (a), (c), (e) e (g) ilustram o resultado da associação do CLAHE com a técnica de limiarização adaptativa. A utilização de duas técnicas que atuam localmente na imagem, provocou um aumento de ruídos na região da plataforma metálica

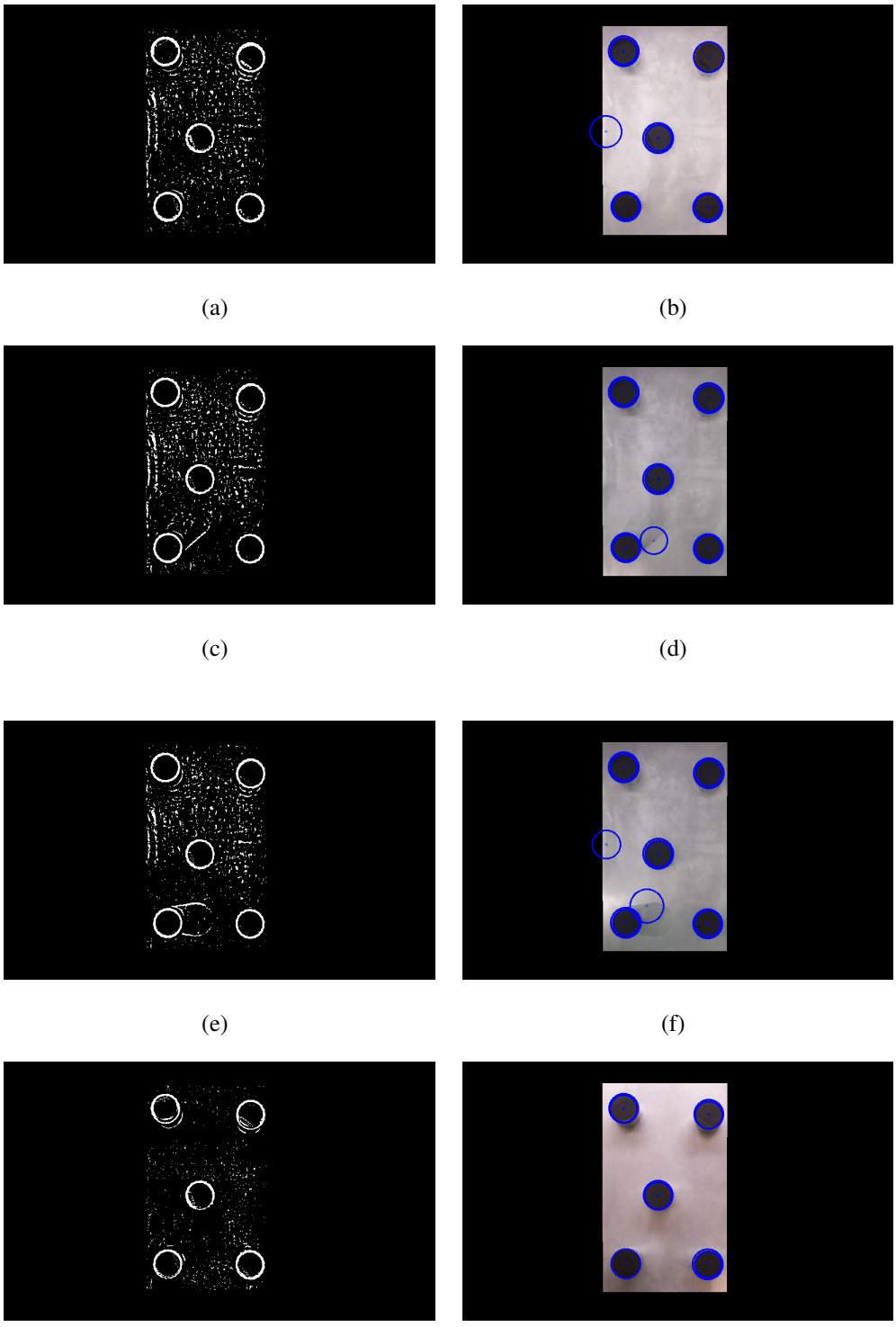
Esse algoritmo apresentou resultados positivos em casos em que a imagem sofre com a iluminação mais incidente na plataforma, um ponto positivo do algoritmo de limiarização adaptativo em relação ao algoritmo de limiarização global, como ilustra a Figura 37 (f). No entanto, os problemas relacionados a sombra dos cilindros e a borda da plataforma aumentam a quantidade de falsos-positivos detectados.

Em seguida, foi testada a segmentação pelo algoritmo de agrupamento *K-means* nas imagens ilustradas nas Figuras 38 (a), (b), (c) e (d). Ao longo dos testes, percebeu-se que ao diminuir o números de *clusters*, a segmentação das imagens com alta incidência luminosa sofria o efeito da junção da sombra do cilindro com a superfície do elemento detectado, ao passo que o aumento demasiado de *clusters* provoca uma redução da área dos objetos detectados, uma vez que a superfície dos objetos-alvo tornam-se partes de vários *clusters* distintos.

Na segmentação com o K-means utilizou-se oito *clusters*, sendo os três primeiros *clusters* considerados como pertencentes a classe dos objetos, ou seja, a todos os *pixels* pertencentes a esses *clusters* atribuiu-se o valor de intensidade 255 representado pela cor branca, aos demais atribuiu-se o valor de intensidade 0 representado pela cor preta.

O *K-means* possui características que dificultam a detecção de objetos que estão sob incidência luminosa direta. As Figuras 38 (a), (c), (e), (g) ilustram as segmentações realizadas com o algoritmo *K-means*, já as Figuras 38 (b), (d), (f), (h) ilustram as detecções dos objetos.

Figura 37: Segmentação pelo Algoritmo de Limiarização Adaptativo e CLAHE. (a), (c), (e) e (g) Imagens binarizadas. (b), (d), (f) e (h) Objetos detectados pelo algoritmo.



(a) (b)

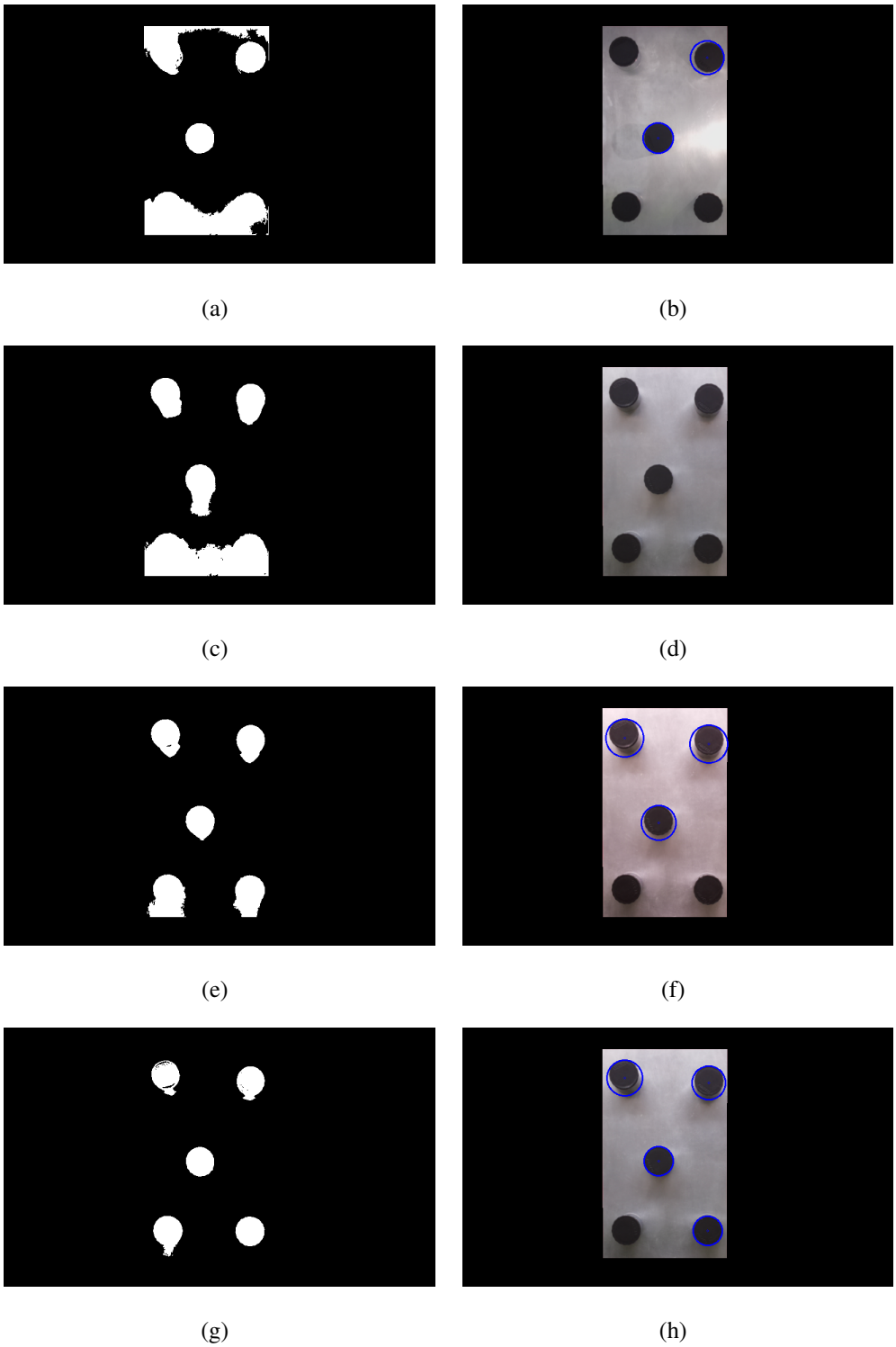
(c) (d)

(e) (f)

(g) (h)

Fonte: Elaborado pela autora.

Figura 38: Segmentação pelo algoritmo de agrupamento K-means, K = 8. (a), (c), (e) e (g) Imagens binarizadas. (b), (d), (f) e (h) Objetos detectados pelo algoritmo.



(a) (b)

(c) (d)

(e) (f)

(g) (h)

Fonte: Elaborado pela autora.

O *K-means* conseguiu detectar os objetos presentes nas imagens com uma maior iluminação (Figura 38 (h)), no entanto, não apresentou não obteve êxito na solução dos problemas que envolvem a sombra dos cilindros e a borda da plataforma.

O *K-means* é um algoritmo de agrupamento que aloca em *clusters* elementos relacionados, nesse caso, *pixels* com valores de intensidade próximos. Esse é um dos fatores que justificam a classificação da sombra do cilindro como uma região pertencente ao objeto.

5.5 Teste de Detecção de Cilindros Coloridos

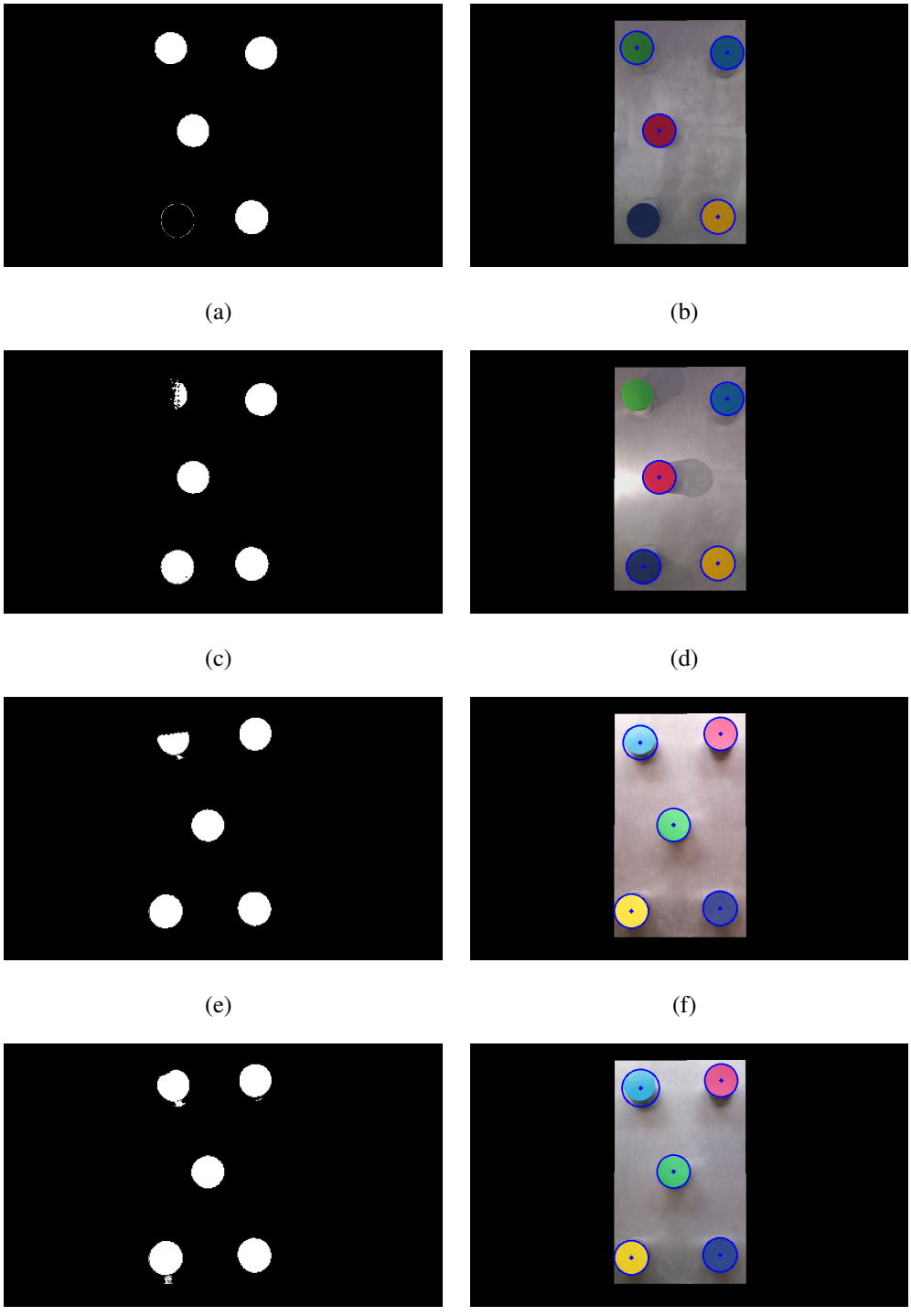
A segunda etapa dos testes envolvendo os algoritmos de segmentação consiste em detectar a superfície dos cilindros de cores distintas. Nessa problemática foi utilizado o algoritmo de detecção de cor baseado no espaço de cor HSV para as cores preto, azul, vermelho, amarelo e verde.

Foram capturadas imagens com os cilindros em diferentes posições e em diferentes condições de iluminação para avaliar se existe uma maior dificuldade de detecção de uma cor em relação a outra. As Figuras 39 (a), (c), (e) e (g) ilustram a segmentação HSV, as Figuras 39 (b), (d), (f) e (h) as respectivas detecções com esse algoritmo.

Na Figura 39 (b) a detecção dos objetos que estão situados na parte superior da plataforma ficam comprometidos devido a maior incidência luminosa nessa região. Observa-se na Figura 39 (b) este mesmo problema, no entanto o algoritmo não é sensível as sombras geradas pelos cilindros.

Um aspecto positivo deste algoritmo está relacionado a não detecção das sombras projetadas pelos cilindros na plataforma, no entanto, o intervalo de valores HSV que representam uma cor deve ser ajustado pelo projetista para que a detecção da cor possa acontecer em diferentes tipos de iluminação, portanto o desempenho desse algoritmo está diretamente relacionado com a forma com que esse intervalo de valores é parametrizado.

Figura 39: Imagens capturadas pelo sistema de VM dos cilindros nas cores amarelo, vermelho, verde, azul e preto. (a), (c), (e) e (g) Imagens binarizadas. (b), (d), (f) e (h) Objetos detectados pelo algoritmo.



(a) (b)

(c) (d)

(e) (f)

(g) (h)

Fonte: Elaborado pela autora.

5.5.1 Métricas de avaliação dos algoritmos

As métricas utilizadas neste trabalho visam avaliar quantitativamente o desempenho de cada algoritmo testado na detecção dos objetos. Utilizou-se a precisão, sensibilidade, o F1 *Score* e o IoU (*Intersection over Union*) para verificar o desempenho de cada algoritmo.

A precisão é a relação entre os valores verdadeiros-positivos (*vp*) com os valores falsos-positivos (*fp*). Em termos práticos, essa taxa indica o percentual de acertos do algoritmo em relação a todos os valores corretos. Por outro lado, a sensibilidade (ou *recall*) é a relação entre os valores verdadeiros-positivos com os valores falsos-negativos (*fn*), ou seja, indica o total de acertos em relação ao total de predições. O F1 *Score* é uma métrica que calcula a média harmônica entre a precisão e a sensibilidade. Essas relações são expressas abaixo:

$$precisão = \frac{vp}{vp + fp} \tag{30}$$

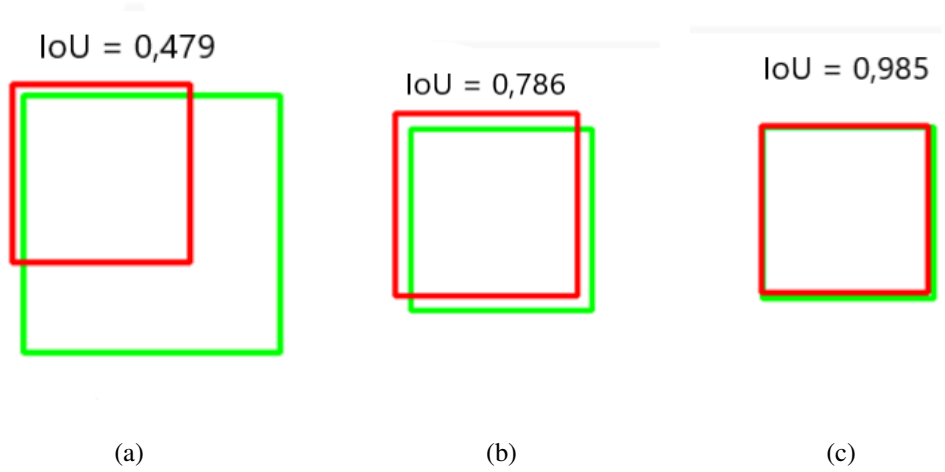
$$sensibilidade = \frac{vp}{vp + fn} \tag{31}$$

$$F1\ Score = \frac{2 \times precisão \times sensibilidade}{precisão + sensibilidade} \tag{32}$$

O IoU é uma métrica que vem sendo adotada em grandes competições de detecção de objetos como a PASCAL VOC e COCO, sua função é definir a exatidão de um detector de objetos em um conjunto de imagens (Rahman and Wang, 2016). A Figura 40 ilustra três casos de detecção e localização de objetos. O quadrado vermelho delimita as fronteiras do objeto, enquanto o quadrado verde indica a região delimitada pelo algoritmo de detecção.

O valor do IoU varia de 0 a 1, sendo 0 quando não há nenhuma sobreposição do objeto com a região detectada pelo algoritmo (falso-negativo) e 1 quando há uma completa sobreposição da região do objeto com a região detectada. Essa métrica também pode ser utilizada como limiar para classificar quando a detecção de um objeto pode ser considerado como um valor verdadeiro-positivo ou falso-positivo.

Figura 40: Exemplo de variações de IoU.



Fonte: Elaborado pela autora.

A IoU é calculada pela divisão da área de interseção do objeto com a região detectada ($A \cap B$) pela área de união do objeto com a região detectada ($A \cup B$).

Embora a precisão e sensibilidade sejam métricas comumente utilizadas em problemas relacionados a classificação de imagens, neste trabalho adotou-se essas métricas associadas ao valor do limiar obtido através da IoU, ou seja, valores acima de IoU > 0.95 são considerados verdadeiros-positivos. O valor de limiar foi definido para atender as necessidades do sistema de VM, garantindo que não aconteça eventuais colisões entre o manipulador robótico e os cilindros.

Os valores falsos-positivos são definidos mediante a relação entre o limiar, caso o IoU esteja abaixo de 0.95 e acima de 0, significa a detecção é de um falso-positivo. Quando o IoU é igual a 0, não há interseção entre a área detectada e a região marcada pelo gabarito.

Caso exista um objeto presente na cena que não foi detectado pelo algoritmo, configura um caso de falso-negativo, neste exemplo não há interseção da área detectada com a área em que o objeto ocupa na imagem. De forma análoga, caso o algoritmo detecte um objeto em uma região da imagem que não contém de fato um objeto, diante de um caso falso-positivo em que também não ocorre a interseção entre a área do objeto com a área detectada pelo algoritmo.

Nesse caso em particular em que o IoU é igual a zero é necessário avaliar visualmente se há presença de um falso-positivo ou de um falso-negativo.

Foram calculados os valores de precisão, sensibilidade, F1 *Score* para 2 conjuntos, cada um contendo 21 imagens diferentes. O primeiro conjunto é composto por imagens com os cilindros de superfície preta, utilizado para testar a detecção de objetos por meio de cinco algoritmos distintos, descritos na Tabela 6. O segundo conjunto é formado por 21 imagens contendo cilindros de superfície coloridas e foram utilizados para testar o algoritmo de segmentação HSV.

Para o sistema de VM, uma alta precisão indica que o robô conseguirá capturar mais peças sem muitos problemas. A medida que o valor da precisão diminui, isso implica que os valores falsos-positivos aumentaram, ou seja, o algoritmo está detectando mais objetos com a $IoU < 95\%$ ou em posições completamente erradas. A detecção de valores falsos-positivos neste contexto está ligado a possíveis colisões entre o robô e o cilindro ou na captura em vazio pelo robô.

Tabela 6: Precisão, Recall e F1 *Score* dos algoritmos de segmentação testados.

Algoritmos	**VP**	**FP**	**FN**	**Precisão**	**Recall**	**F1 Score**
Limiarização Global	45	59	4	43%	91,83%	58,57%
Limiarização Global + CLAHE	47	58	0	44,76%	100%	61,03%
Limiarização Adaptativa	60	45	10	57,14%	85,71%	68,82%
Limiarização Adaptativa + CLAHE	46	50	15	47,91%	75,4%	58,59%
K-Means	52	50	12	50,98%	81,25%	62,15%
Segmentação HSV	10	90	5	10%	66,66%	17,39%

Por outro lado, a sensibilidade (*recall*) fornece um indicio de que o robô realizará a captura de todos os objetos presentes na plataforma. Deixar de capturar um objeto é menos danoso ao sistema do que realizar uma captura errada que venha provocar colisões entre a garra do manipulador e a plataforma. Portanto, um algoritmo que apresenta uma maior precisão é mais aplicável neste contexto do que um algoritmo com alta sensibilidade.

O algoritmo de limiarização global com o CLAHE na etapa de pré-processamento apresentou recall 100%, ou seja, todas os objetos foram detectados pelo algorimo, no entanto, menos da metade (44,76%) foram classificados como verdadeiros-positivos. Com a introdução do CLAHE, a quantidade de acertos proporcionais aumentaram, no entanto, houve uma ligeira diminuição no *recall*.

A limiarização adaptativa foi o algoritmo que apresentou um maior valor de precisão entre todos os algoritmos testados. No entanto, o valor de *recall* foi abaixo dos valores apresentados pelo algoritmo de limiarização global devido a detecção de sombras e de eventuais ruídos presentes na plataforma. A introdução do CLAHE provocou uma diminuição da precisão e do *recall*, pois essas duas técnicas associadas potencializam ainda mais os ruídos presentes na imagem.

O K-means obteve uma precisão de $50,98\%$, a segunda maior da Tabela 6, no entanto, esse algoritmo demanda um maior poder computacional do sistema embarcado, principalmente quando o trabalha com imagens de maior resolução.

O algoritmo de segmentação por cor obteve a pior precisão entre os algoritmos testados, devida a fatores como a variação de luminosidade a sensibilidade do algoritmos às sombras projetadas pelos cilindros na plataforma. Portanto, a probabilidade de acontecerem colisões durante a captura de cilindros coloridos é maior do que durante a captura de cilindros pretos.

Para essa aplicação, o algoritmo de limiarização adaptativa é mais adequado para contexto em que a iluminação é variável, pois apresenta uma quantidade de detecções falso-positivos menor além disso, apresenta o melhor valor de F1 Score da Tabela 6. A detecção de cilindros coloridos com o método de segmentação HSV de precisão de apenas 10%, o que indica que grandes chances de colisões durante a captura de objetos com tampas coloridas.

6 Conclusões

Neste trabalho foi desenvolvido um sistema de VM capaz de detectar objetos cilíndricos de mesma altura sob condições aleatórias de posicionamento e de iluminação. Esse sistema pode ser considerado de baixo custo pois utiliza um sistema embarcado e uma câmera que são amplamente comercializados no mercado. Além disso, toda a programação implementada utiliza bibliotecas *open souce* que funcionam em no sistema operacional Raspibian, baseada em uma versão do Linux.

O sistema de VM desenvolvido neste trabalho pode auxiliar pequenas e médias industrias em atividades de manipulação robótica, controle de processos e de qualidade. No âmbito desta pesquisa, mostrou-se que é possível realizar a captura de objetos cilíndricos com um manipulador robóticos assistido pelo sistema de VM, em um cenário em que a garra possui uma abertura limitada.

Com o desenvolvimento de um algoritmo de calibração da câmera foi possível corrigir distorções de perspectiva e distorções da lente que influenciam na conversão do plano da imagem para o plano de trabalho. Todas as vezes em que o sistema de VM perder o sincronismo com o manipulador robótico, o algoritmo de calibração é empregado para parametrizar novamente os coeficientes de distorção e informar a localização da plataforma de trabalho ao robô.

Os algoritmos de limiarização global e limiarização adaptativa obtiveram os melhores valores de *recall* e precisão respectivamente, no entanto, quando associados ao CLAHE como etapa de pré-processamento, diminuíam seu desempenho. K-means, por sua vez, obteve o melhor F1 *Score*, embora apresente outras limitações relacionadas a demanda de poder computacional do sistema embarcado.

Ao utilizar a imagem do tipo *RAW* foi possível obter uma diminuição do erro, calculado pela diferença entre o valor detectado pelo sistema de VM pelo VVC obtido pela medição na MMC.

6.1 Trabalhos Futuros

A partir da implementação deste sistema de VM é possível desenvolver tanto aprimoramentos para esta aplicação, quanto outros trabalhos que visam identificar outros tipos de objetos em outros contextos diferentes. O sistema de VM além de auxiliar manipuladores robóticos na manipulação de peças, podem ser usados para aplicações de controle de qualidade e supervisão.

Recomenda-se como sugestão para trabalhos futuros, a utilização de algoritmos de *Deep Learning*, mais especificamente as CNNs (*Convolutional Neural Network*), para realizar a detecção de objetos em um contexto com iluminação variável e peças com cores distintas, visando o aumento da taxa de precisão e *recall*. Embora esses algoritmos tenham um custo computacional mais elevado, é possível embarcar esta solução devido às novas arquiteturas que estão sendo desenvolvidas.

A captura de peças de diferentes alturas é uma problemática que envolve o uso de visão 3D, neste caso, deve-se realizar uma investigação para decidir qual a melhor técnica para que o sistema de VM consiga saber a profundidade dos objetos e assim ser capaz de capturar objetos de diferentes alturas. As técnicas mais comumente empregadas utilizam a associação de duas câmeras, o que é chamado de visão *stéreo*, além disso, é possível também utilizar uma câmera juntamente a um sensor laser.

A mudança da geometria da garra do manipulador robótico, flexibiliza possíveis futuras aplicações da VM, neste sentido, é possível realizar a captura de objetos com diferentes formas. O algoritmo a ser desenvolvido deve ser capaz de identificar o melhor ponto para realizar a captura dos objetos, pois nem sempre o centroide vai representar a melhor posição para captura.

Referências

A. Y. I. Abdel and H.M. Karara. *Direct Linear Transformation from Comparator Coordinates Into Object Space Coordinates in Close-range Photogrammetry*. 1971.

A. Albertazzi and A.R. de Sousa. *Fundamentos de metrologia científica e industrial*. Manole, 2008. ISBN 9788520421161. URL https://books.google.com.br/books?id=eYXyQwAACAAJ.

Rey Casse. *Projective Geometry: An Introduction*. Oxford University Press, 1st edition, 2006. ISBN 0199298866.

Jianhui Chen, K. Benzeroual, and R. S. Allison. Robust homography for real-time image undistortion. In *2013 International Conference on 3D Imaging*, pages 1–8, Dec 2013. doi: 10.1109/IC3D.2013.6732075.

CNI. Indústria 4.0. sondagem especial. Number 66. CONFEDERAÇÃO NACIONAL DA INDÚSTRIA, Maio 2016.

E. R. Davies. *Computer and Machine Vision, Fourth Edition: Theory, Algorithms, Practicalities*. Academic Press, Inc., Orlando, FL, USA, 4th edition, 2012. ISBN 0123869080, 9780123869081.

OpenCV Documentation. Camera calibration with opencv. https://docs.opencv.org/2.4.13.7/doc/tutorials/calib3d/camera_calibration/camera_calibration.html, 2018. Accessed: 2018-08-18.

Ogê M. Filho and Hugo V. Neto. *Processamento digital de imagens.* Brasport, , Rio de Janeiro, 1999.

Raspberry Pi Foundation. Camera module v2 - specifications. https://www.raspberrypi.org/products/camera-module-v2/, 2016a. Accessed: 2018-05-25.

Raspberry Pi Foundation. Raspberry pi 3 model b - specifications. https://www.raspberrypi.org/products/raspberry-pi-3-model-b/, 2016b. Accessed: 2018-05-25.

M. Gao, Y. Yan, Y. Yang, J. Huang, and Z. He. A positioning system based on monocular vision for industrial robots. In *2016 3rd International Conference on Information Science and Control Engineering (ICISCE)*, pages 784–788, July 2016. doi: 10.1109/ICISCE.2016.172.

Rafael C. Gonzalez and R. E. Woods. *Processamento digital de imagens.* Pearson Prentice Hall, , São Paulo, 2010.

Rafael C. Gonzalez and Richard E. Woods. *Digital Image Processing (3rd Edition)*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 2006. ISBN 013168728X.

Robin Hartshorne. *Geometry: Euclid and Beyond*. Springer Verlag, 1st edition, 2002. ISBN 978-0-387-98650-0.

Paul Heckbert. Projective mappings for image warping. Technical report, in Fundamentals of Texture Mapping and Image Warping (Paul Heckbert, Master's Thesis), U.C.Berkeley, 1999.

M. Hermann, T. Pentek, and B. Otto. Design principles for industrie 4.0 scenarios: A literature review. Jan 2015.

Karel Horak and Ludek Zalud. Image processing on raspberry pi for mobile robotics. In *International Journal of Signal Processing Systems*, volume 4, pages 494–498, 12 2016. doi: 10.18178/ijsps.4.6.494-498.

Ming-Kuei Hu. Visual pattern recognition by moment invariants. *IRE Transactions on Information Theory*, 8(2):179–187, February 1962. ISSN 0096-1000. doi: 10.1109/TIT.1962.1057692.

P. Jenamani, B. P. Das, S. Rup, A. Mohapatra, and S. Mohanty. An evaluation of moving object detection under varying illumination. In *2017 2nd IEEE International Conference on Recent Trends in Electronics, Information Communication Technology (RTEICT)*, pages 1399–1403, May 2017. doi: 10.1109/RTEICT.2017.8256830.

E. Kadiyala, S. Meda, R. Basani, and S. Muthulakshmi. Global industrial process monitoring through iot using raspberry pi. In *2017 International Conference on Nextgen Electronic Technologies: Silicon to Software (ICNETS2)*, pages 260–262, March 2017. doi: 10.1109/ICNETS2.2017.8067944.

K. Kim, J. Cho, J. Pyo, S. Kang, and J. Kim. Dynamic object recognition using precise location detection and ann for robot manipulator. In *2017 International Conference on Control, Artificial Intelligence, Robotics Optimization (ICCAIRO)*, pages 237–241, May 2017. doi: 10.1109/ICCAIRO.2017.52.

B. V. S. Krishna, J. Oviya, S. Gowri, and M. Varshini. Cloud robotics in industry using raspberry pi. In *2016 Second International Conference on Science Technology Engineering and Management (ICONSTEM)*, pages 543–547, March 2016. doi: 10.1109/ICONSTEM.2016.7560952.

Vishal Kumar, Qiang Wang, Minghua Wang, Syed Rizwan, Shakir Ali, and Xuan Liu. Computer vision based object grasping 6dof robotic arm using picamera. In *2018 Conference: Conference: International Conference on Control, Automation and Robotics (ICCAR 2018)*, 03 2018.

G. T. Laureano. Deteção Topológica de Padrões de Xadrez para Calibração de Câmeras. Master's thesis, Universidade de São Paulo - USP, 2013.

Junlan Li and Dawei Zhang. Camera calibration with a near-parallel imaging system based on geometric moments. *Optical Engineering - OPT ENG*, 50, 02 2011. doi: 10.1117/1.3533032.

C. C. Lin, P. Gonzalez, M. Y. Cheng, G. Y. Luo, and T. Y. Kao. Vision based object grasping of industrial manipulator. In *2016 International Conference on Advanced Robotics and Intelligent Systems (ARIS)*, pages 1–5, Aug 2016. doi: 10.1109/ARIS.2016.7886613.

Z. Liu, B. Zhao, and H. Zhu. Research of sorting technology based on industrial robot of machine vision. In *2012 Fifth International Symposium on Computational Intelligence and Design*, volume 1, pages 57–61, Oct 2012. doi: 10.1109/ISCID.2012.23.

Manuel E. Loaiza, Alberto B. Raposo, and Marcelo Gattass. Multi-camera calibration based on an invariant pattern. *Computers Graphics*, 35(2):198 – 207, 2011. ISSN 0097-8493. doi: https://doi.org/10.1016/j.cag.2010.12.007. URL http://www.sciencedirect.com/science/article/pii/S0097849310002050. Virtual Reality in Brazil Visual Computing in Biology and Medicine Semantic 3D media and content Cultural Heritage.

J. MacQueen. Some methods for classification and analysis of multivariate observations. In *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Vo-*

lume 1: Statistics, pages 281–297, Berkeley, Calif., 1967. University of California Press. URL https://projecteuclid.org/euclid.bsmsp/1200512992.

Mitsubishi. Industrial robot: Crnq/crnd controller. instruction manual - detail explanations of functions and operations. volume 2.2.x, page 534. Mitsubishi, 2011.

Raymond H Myers. *Probabilidade e estatística para engenharia e ciências (8rd Edition)*. Pearson Education do Brasil, São Paulo, PB, Brasil, 2009. ISBN 978-85-430-1440-1.

Suraj Nair, Nikhil Somani, Artur Grunau, Emmanuel Dean-Leon, and Alois Knoll. Image processing units on ultra-low-cost embedded hardware: Algorithmic optimizations for real-time performance. *Journal of Signal Processing Systems*, pages 1–17, 8 2017. ISSN 1939-8018. doi: 10.1007/s11265-017-1267-1.

J. Park, S. C. Byun, and B. U. Lee. Lens distortion correction using ideal image coordinates. *IEEE Transactions on Consumer Electronics*, 55(3):987–991, August 2009. ISSN 0098-3063. doi: 10.1109/TCE.2009.5278053.

Hélio Pedrini and William Robson Schwartz. *Análise de imagens digitais*. Thomson Learning, São Paulo, 2008.

L. Pérez, Í. Rodríguez, N. Rodríguez, R. Usamentiaga, and D. F. García. Robot guidance using machine vision techniques in industrial environments: A comparative review. *Radiology*, 16(3): 335, 2016.

Md. Atiqur Rahman and Yang Wang. Optimizing intersection-over-union in deep neural networks for image segmentation. In *ISVC*, 2016.

Jrgen Richter-Gebert. *Perspectives on Projective Geometry: A Guided Tour Through Real and Complex Geometry*. Springer Publishing Company, Incorporated, 1st edition, 2011. ISBN 3642172857, 9783642172854.

M. Rüßmann, M. Lorenz, P. Gerbert, M. Waldner, J Justus, P. Engel, and M. Harnisch. Industry 4.0: The future of productivity and growth in manufacturing industries. Abril 2015.

Joaquim Salvi, Xavier Armangué, and Joan Batlle. A comparative review of camera calibrating methods with accuracy evaluation. *Pattern Recognition*, 35(7):1617 – 1635, 2002. ISSN 0031-3203. doi: https://doi.org/10.1016/S0031-3203(01)00126-1. URL http://www.sciencedirect.com/science/article/pii/S0031320301001261.

M. Shah. *Fundaments of Computer Vision*. n/a, Florida, Orlando, USA, 1st edition, 1997. ISBN n/a.

M. N. Sishi and A. Telukdarie. Implementation of industry 4.0 technologies in the mining industry: A case study. In *2017 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)*, pages 201–205, Dec 2017. doi: 10.1109/IEEM.2017.8289880.

C. Su, H. Chiu, and T. Hsieh. An efficient image retrieval based on hsv color space. In *2011 International Conference on Electrical and Control Engineering*, pages 5746–5749, Sept 2011. doi: 10.1109/ICECENG.2011.6058026.

R. Szabó and A. Gontean. Industrial robotic automation with raspberry pi using image processing. In *2016 International Conference on Applied Electronics (AE)*, pages 265–268, Sept 2016. doi: 10.1109/AE.2016.7577287.

Richard Szeliski. *Computer Vision: Algorithms and Applications*. Springer-Verlag, Berlin, Heidelberg, 1st edition, 2010. ISBN 1848829345, 9781848829343.

OpenCV team. Opencv - about. https://opencv.org/about.html, 2018. Accessed: 2018-05-25.

K. D. Thoben, T. West, and S. A. Wiesner. ”industry 4.0”and samart manufacturing - a review of research issue and application examples. Janeiro 2017. doi: 10.20965/ijat.2017.p0004.

David Vernon. *Machine vision - automated visual inspection and robot vision.* Prentice Hall, 1991. ISBN 978-0-13-543398-0.

K. Xia and Z. Weng. Workpieces sorting system based on industrial robot of machine vision. In *2016 3rd International Conference on Systems and Informatics (ICSAI)*, pages 422–426, Nov 2016. doi: 10.1109/ICSAI.2016.7810992.

S. Yahya Nikouei, Y. Chen, S. Song, R. Xu, B.-Y. Choi, and T. R. Faughnan. Real-Time Human Detection as an Edge Service Enabled by a Lightweight CNN. *ArXiv e-prints*, April 2018.

X. F. Zhang, A. Luo, W. Tao, and H. Burkhardt. Camera calibration based on 3d-point-grid. In Alberto Del Bimbo, editor, *Image Analysis and Processing*, pages 636–643, Berlin, Heidelberg, 1997. Springer Berlin Heidelberg. ISBN 978-3-540-69585-1.

Z. Zhang. A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(11):1330–1334, Nov 2000. ISSN 0162-8828. doi: 10.1109/34.888718.

Zhengyou Zhang. Camera calibration with one-dimensional objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(7):892–899, July 2004. ISSN 0162-8828. doi: 10.1109/TPAMI.2004.21.

Karel Zuiderveld. Graphics gems iv. chapter Contrast Limited Adaptive Histogram Equalization, pages 474–485. Academic Press Professional, Inc., San Diego, CA, USA, 1994. ISBN 0-12-336155-9. URL http://dl.acm.org/citation.cfm?id=180895.180940.