

**UNIVERSIDADE DE CAXIAS DO SUL  
ÁREA DO CONHECIMENTO DE CIÊNCIAS EXATAS E ENGENHARIAS**

**WILLIAM TRENTIN MACHADO**

**INSPEÇÃO AUTOMÁTICA DE FRASCOS UTILIZANDO VISÃO  
COMPUTACIONAL EM SISTEMA EMBARCADO**

**BENTO GONÇALVES**

**2019**

**WILLIAM TRENTIN MACHADO**

**INSPEÇÃO AUTOMÁTICA DE FRASCOS UTILIZANDO VISÃO  
COMPUTACIONAL EM SISTEMA EMBARCADO**

Trabalho de Conclusão de Curso apresentado no Curso de Engenharia Elétrica do Campus Universitário da Região dos Vinhedos, da Universidade de Caxias do Sul, como requisito parcial à obtenção do título de Engenheiro Eletricista

Orientador:  
Prof. Me. Patric Janner Marques

**BENTO GONÇALVES**

**2019**

**WILLIAM TRENTIN MACHADO**

**INSPEÇÃO AUTOMÁTICA DE FRASCOS UTILIZANDO VISÃO  
COMPUTACIONAL EM SISTEMA EMBARCADO**

Trabalho de Conclusão de Curso apresentado no Curso de Engenharia Elétrica do Campus Universitário da Região dos Vinhedos, da Universidade de Caxias do Sul, como requisito parcial à obtenção do título de Engenheiro Eletricista

Orientador:  
Prof. Me. Patric Janner Marques

**Aprovado em:** \_\_\_\_/\_\_\_\_/\_\_\_\_.

**Banca Examinadora**

---

Orientador Prof. Me. Patric Janner Marques  
Universidade de Caxias do Sul – RS

---

Profa. Dra. Marilda Machado Spindola  
Universidade de Caxias do Sul – RS

---

Prof. Dr. Guilherme Holsbach Costa  
Universidade de Caxias do Sul – RS

## **AGRADECIMENTOS**

Agradeço aos meus pais por todo o incentivo, suporte e amor dedicados a mim durante esta caminhada e toda a minha vida.

À minha irmã por ter ajudado na minha criação com muito carinho.

Aos meus amigos pelos momentos de descontração inesquecíveis e pelo apoio nas horas difíceis.

Ao Prof. Patric Janner Marques pelo conhecimento agregado, orientações e paciência para que o trabalho fosse realizado da melhor forma possível.

## RESUMO

O aumento da capacidade produtiva, bem como a ocorrência de mercados consumidores mais exigentes, têm impulsionado o uso de sistemas de inspeção visual automática destinados ao controle de qualidade e gerenciamento da produção. Entretanto, estes sistemas ainda apresentam um custo elevado, o que pode tornar sua aplicação inviável em empresas de pequeno e médio porte. Tendo estas questões como motivação, este trabalho propõe o desenvolvimento e a avaliação de um sistema de inspeção visual automática de frascos utilizando dispositivos de baixo custo. Esta inspeção visa a detecção de presença de tampa em dois modelos de frascos diferentes, através de imagens capturadas em linha de envase a velocidade nominal. Para tal desenvolvimento, uma revisão sobre os tópicos de visão computacional pertinentes ao trabalho e uma análise do estado da arte sobre inspeção visual automática de frascos são apresentadas. A partir destas revisões e considerando-se as particularidades do processo e as limitações impostas pelo *hardware*, foram selecionadas 4 diferentes abordagens de processamento de imagens para avaliação. Neste trabalho estes métodos foram denominados como “método das linhas”, “método das áreas”, “método K-means” e “método ORB”. Ao todo, a base de dados gerada contém 313 imagens do conjunto de frasco branco e tampa azul e 125 imagens do conjunto de frasco verde e tampa verde. Os resultados indicam que para o modelo de frasco branco o método que apresentou os melhores resultados foi o método das áreas. Com este método a inspeção do frasco branco atingiu uma acurácia de 100% e um tempo máximo de inspeção de 311,9 ms, não ultrapassando o limite de tempo de inspeção imposto pela velocidade de produção da linha de envase. Para o modelo de frasco verde o melhor resultado foi obtido através do método ORB. Com este método foi possível atingir um valor de acurácia de 100%. No entanto, o maior tempo de inspeção do método ORB para o frasco verde foi de 460,3 ms, ultrapassando o limite de tempo de inspeção imposto pela linha de envase. Observa-se que imagens com menor variação de cor e intensidade entre as regiões de interesse demandam algoritmos de processamento mais robustos e, como consequência, aumenta-se o tempo de inspeção do sistema.

**Palavras-chave:** Visão Computacional. Extração de Características. Inspeção Visual Automática. Tampa de Frasco.

## ABSTRACT

The increase in production capacity as well as the occurrence of more demanding consumer markets have driven the use of automatic visual inspection systems aimed at quality control and production management. However, these systems still present a high cost, which can make their application infeasible in small and medium enterprises. With these issues as motivation, this work proposes the development and evaluation of an automatic visual inspection system of bottles using low cost devices. This inspection aims at detecting the presence of cap in two different bottle models, through images captured in the packaging line at nominal speed. For such development, a review on the topics of computer vision pertinent to the work and an analysis of the state of art on automatic visual inspection of bottles are presented. From these reviews and considering the particularities of the process and the limitations imposed by the hardware, 4 different image processing approaches were implemented for evaluation. In this work these methods were denominated as "line method", "area method", "K-means method" and "ORB method". Altogether, the generated database contains 313 images of the white bottle and blue cap set and 125 images of the green bottle and green cap set. The results indicate that for the white bottle model the method that presented the best results was the areas method. With this method the inspection of the white bottle reached an accuracy of 100% and a maximum inspection time of 311.9 ms, not exceeding the inspection time limit imposed by the speed of the production line. For the green bottle model the best result was obtained using the ORB method. With this method it was possible to reach an accuracy value of 100%. However, the longest inspection time of the ORB method for the green bottle was 460.3 ms, exceeding the inspection time limit imposed by the packaging line. It is observed that images with less variation of color and intensity between the regions of interest demand more robust processing algorithms. As a consequence, the inspection time of the system is increased.

**Keywords:** Computer Vision. Feature Extractor. Automatic Visual Inspection. Bottle Cap.

## LISTA DE ILUSTRAÇÕES

Figura 1 - Etapas do processamento de imagem e as saídas obtidas. ....	15
Figura 2 - Exemplo de sistema de inspeção de nível de líquido em garrafas.....	18
Figura 3 - Processo de aquisição de imagem digital por um sensor matricial. ....	20
Figura 4 - Captura de um alvo com diferentes velocidades do obturador.....	21
Figura 5 - Diagrama temporal do funcionamento de um sensor <i>rolling shutter</i> .....	22
Figura 6 - Formas básicas de representação de imagens digitais.....	25
Figura 7 - Espaço de cores RGB como um cubo tridimensional.....	26
Figura 8 - Exemplo de aplicação do detector de bordas de Canny. ....	28
Figura 9 - Comparação de pontos para extração de características de imagens. ....	31
Figura 10 - Distribuição da região de avaliação de cantos. ....	31
Figura 11 - Distribuição dos pares de teste de acordo com os métodos propostos..	34
Figura 12 - Linhas horizontais para inspeção de tampa e nível de líquido. ....	42
Figura 13 - Histograma dos autores Abdelhedi, Taouil e Hadjkacem (2012).....	43
Figura 14 - Diagrama do método proposto pelos autores Liu <i>et al.</i> (2017).....	44
Figura 15 - Exemplo das condições de inspeção da tampa. ....	49
Figura 16 - Disposição da câmera e dispositivos para aquisição de imagens.....	50
Figura 17 - Fluxograma de aquisição de imagem.....	53
Figura 18 - Distorções encontradas nas imagens capturadas.....	53
Figura 19 - Sequência de captura de frascos com diferentes posicionamentos.....	54
Figura 20 - Fluxograma do método das linhas.....	56
Figura 21 - Imagens geradas na etapa 4 do método das linhas. ....	57
Figura 22 - Fluxograma do método das áreas. ....	58
Figura 23 - Cálculo do histograma para o frasco branco com tampa. ....	59
Figura 24 - Cálculo do histograma para o frasco verde com tampa. ....	59
Figura 25 - Imagens de saída da sexta etapa do método das áreas.....	60
Figura 26 - Fluxograma do método K-means. ....	62
Figura 27 - Imagens de saída da etapa 3 do método K-means.....	63
Figura 28 - Fluxograma do método ORB.....	64
Figura 29 - Pontos de interesse localizados na ROI pelo método ORB. ....	65
Figura 30 - Amostras do tempo de aquisição das imagens. ....	68
Figura 31 - Tempos de processamento do método das linhas para o frasco branco. .....	70

Figura 32 - Tempos de processamento do método das áreas para o frasco branco.	73
Figura 33 - Tempos de processamento do método das áreas para o frasco verde..	74
Figura 34 - Tempos de processamento do método K-means para o frasco branco.	76
Figura 35 - Tempos de processamento do método K-means para o frasco verde. ..	77
Figura 36 - Tempos de processamento do método ORB para o frasco branco.....	80
Figura 37 - Tempos de processamento do método ORB para o frasco verde.....	80
Figura 38 - Comparativo de desempenho dos métodos.....	81
Figura 39 - Comparativo entre os tempos de inspeção. ....	82

## LISTA DE TABELAS

Tabela 1 - Comparação entre o sistema visual humano e um sistema de visão computacional.....	16
Tabela 2 - Comparativo entre as fontes de iluminação.....	24
Tabela 3 - Especificações técnicas da Raspberry Pi 3 Model B.....	37
Tabela 4 - Exemplo de matriz de confusão para um dado classificador.....	39
Tabela 5 - Resultados obtidos pelos autores Kumchoo e Chiracharit (2018). ....	47
Tabela 6 - Resumo do banco de imagens de frascos.....	52
Tabela 7 – Parâmetros medidos da etapa de aquisição de imagens. ....	52
Tabela 8 - Parâmetros da etapa de aquisição de imagens.....	69
Tabela 9 - Matriz de confusão do método das linhas para o frasco branco.....	70
Tabela 10 - Resultados obtidos pelo método das linhas. ....	71
Tabela 11 - Matriz de confusão do método das áreas para o frasco branco. ....	72
Tabela 12 - Matriz de confusão do método das áreas para o frasco verde. ....	73
Tabela 13 - Resultados obtidos pelo método das áreas.....	74
Tabela 14 - Matriz de confusão do método K-means para o frasco branco. ....	75
Tabela 15 - Matriz de confusão do método K-means para o frasco verde. ....	76
Tabela 16 - Resultados obtidos pelo método K-means. ....	77
Tabela 17 - Matriz de confusão do método ORB para o frasco branco.....	78
Tabela 18 - Matriz de confusão do método ORB para o frasco verde.....	79
Tabela 19 - Resultados do método ORB.....	79

## LISTA DE ABREVIATURAS E SIGLAS

SVA	Sistema de Visão Artificial
RGB	<i>Red, Green, Blue</i>
CCD	<i>Charge-coupled Device</i>
CMOS	<i>Complementary Metal Oxide Semiconductor</i>
FPS	<i>Frames Per Second</i>
GPU	<i>Graphics Processing Unit</i>
SIFT	<i>Scale-Invariant Feature Transform</i>
SURF	<i>Speeded-Up Robust Feature</i>
ORB	<i>Oriented FAST and Rotated BRIEF</i>
FAST	<i>Features from Accelerated Segment Test</i>
BRIEF	<i>Binary Robust Independent Elementary Features</i>
HOG	<i>Histogram of Oriented Gradients</i>
ROI	<i>Region Of Interest</i>
LBP	<i>Local Binary Pattern</i>
SVM	<i>Support Vector Machine</i>
HSV	<i>Hue Saturation Value</i>
PET	Polietileno Tereftalato
NCC	<i>Normalized Cross-Correlation</i>
SUSAN	<i>Smallest Univalve Segment Assimilating Nucleus</i>
CPU	<i>Central Processing Units</i>
RAM	<i>Random Access Memory</i>
PEAD	Polietileno de Alta Densidade

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO.....</b>	<b>12</b>
1.1	OBJETIVO GERAL .....	13
1.2	OBJETIVOS ESPECÍFICOS.....	13
1.3	ESCOPO E RESTRIÇÕES .....	13
1.4	APRESENTAÇÃO DO TRABALHO .....	14
<b>2</b>	<b>REVISÃO BIBLIOGRÁFICA .....</b>	<b>15</b>
2.1	VISÃO COMPUTACIONAL E INSPEÇÃO AUTOMÁTICA .....	15
2.2	AQUISIÇÃO E REPRESENTAÇÃO DE IMAGENS DIGITAIS .....	19
<b>2.2.1</b>	<b>Aquisição de imagens digitais.....</b>	<b>19</b>
2.2.1.1	Aspectos da captura de objetos em movimento .....	21
2.2.1.2	Iluminação controlada.....	23
<b>2.2.2</b>	<b>Representação de imagens digitais .....</b>	<b>25</b>
2.2.2.1	Espaço de cores RGB .....	26
2.3	SEGMENTAÇÃO E EXTRAÇÃO DE CARACTERÍSTICAS.....	27
<b>2.3.1</b>	<b>Detector de bordas de Canny .....</b>	<b>27</b>
<b>2.3.2</b>	<b>Extração e descrição de características.....</b>	<b>30</b>
2.3.2.1	Features from Accelerated Segment Test (FAST).....	30
2.3.2.2	Binary Robust Independent Elementary Features (BRIEF) .....	32
2.3.2.3	Oriented FAST and Rotated BRIEF (ORB).....	34
2.3.2.4	Comparação de características .....	36
2.4	SISTEMAS EMBARCADOS E FERRAMENTAS DE DESENVOLVIMENTO .....	36
<b>2.4.1</b>	<b>Raspberry Pi 3 Model B e Câmera.....</b>	<b>37</b>
<b>2.4.2</b>	<b>Biblioteca OpenCV.....</b>	<b>38</b>
2.5	AVALIAÇÃO DE DESEMPENHO DE CLASSIFICADORES.....	38
<b>3</b>	<b>ESTADO DA ARTE .....</b>	<b>41</b>
<b>4</b>	<b>DESENVOLVIMENTO DO TRABALHO .....</b>	<b>48</b>
4.1	IMPLEMENTAÇÃO DO SISTEMA.....	48
4.2	AQUISIÇÃO E GERAÇÃO DO BANCO DE IMAGENS .....	51
4.3	ALGORITMOS DE INSPEÇÃO.....	54
<b>4.3.1</b>	<b>Método das linhas.....</b>	<b>55</b>

4.3.2	<b>Método das áreas.....</b>	<b>58</b>
4.3.3	<b>Método K-means .....</b>	<b>61</b>
4.3.4	<b>Método ORB .....</b>	<b>64</b>
4.4	AVALIAÇÃO DE DESEMPENHO .....	67
5	<b>AVALIAÇÃO DO SISTEMA .....</b>	<b>68</b>
5.1	RESULTADOS OBTIDOS PARA CADA MÉTODO APLICADO .....	68
5.1.1	<b>Método das Linhas .....</b>	<b>69</b>
5.1.2	<b>Método das áreas.....</b>	<b>72</b>
5.1.3	<b>Método K-means .....</b>	<b>75</b>
5.1.4	<b>Método ORB .....</b>	<b>78</b>
5.2	COMPARAÇÃO E ANÁLISE DOS RESULTADOS OBTIDOS.....	81
6	<b>CONCLUSÃO.....</b>	<b>83</b>

## 1 INTRODUÇÃO

Os processos de inspeção visual automática têm ajudado a indústria a manter ou melhorar a qualidade de seus produtos mesmo sob condições de aumento da demanda produtiva. Com mercados consumidores cada vez mais exigentes quanto a qualidade dos produtos consumidos, fez-se necessário o aumento da atenção dada ao controle de qualidade. Além do quesito de qualidade, diversas aplicações requerem que o monitoramento e o controle sob o produto seja realizado durante o processo de industrialização (BALDNER *et al.*, 2017).

Inicialmente, os processos de inspeção visual eram realizados exclusivamente por humanos, mas este cenário está mudando com o tempo. Observa-se que a visão humana é melhor na tarefa de interpretação qualitativa de um cenário complexo e não estruturado. Porém, quando se tratam de análises quantitativas e em altas velocidades, um sistema de inspeção computacional se destaca por sua precisão e repetibilidade (COGNEX, 2018). Além destas características, a inspeção realizada por humanos está sujeita a fadiga ao longo da jornada de trabalho, possíveis distrações e subjetividades que diminuem a assertividade do processo (DUAN; WANG; LIU, 2004, p. 842).

O aumento da capacidade de processamento dos dispositivos eletrônicos e o desenvolvimento de novas técnicas vêm permitindo diferentes aplicações utilizando visão computacional, tanto em nível acadêmico quanto industrial. Como exemplo, pode-se citar sistemas que aplicam a visão computacional com o intuito de inspecionar seringas médicas em linha de produção, podendo avaliar até 600 seringas por minuto (JIA, 2009). Outro exemplo, é o trabalho realizado pelos autores Azhar *et al.* (2016) que utiliza os conceitos de visão computacional e diferentes classificadores para a detecção de buracos em estradas asfaltadas.

Os fabricantes de sistemas de visão encontrados atualmente no mercado desenvolveram novas plataformas e arquiteturas visando a melhoria na qualidade de inspeção e a redução de custo do sistema (COGNEX, 2018). Entretanto, mesmo com os esforços dos fabricantes, esses dispositivos costumam apresentar um alto valor agregado, o que por vezes torna a sua aplicação inviável em empresas de pequeno e médio porte.

Diante do contexto apresentado, este trabalho propõe o desenvolvimento e a avaliação de um sistema de visão de baixo custo para a inspeção de frascos em linha de produção. O sistema proposto deverá ser capaz de identificar a presença da tampa no frasco, gerando uma sinalização no caso de não conformidade. Para tal, será utilizado um sistema embarcado em conjunto com uma câmera modular, com intuito de obter-se um sistema de baixo custo. Torna-se necessária uma avaliação quanto as limitações do *hardware* e a performance dos algoritmos de visão computacional, adequados à situação do processo de envase e à plataforma embarcada escolhida.

### 1.1 OBJETIVO GERAL

Este trabalho tem como objetivo geral o desenvolvimento e a avaliação de um sistema de visão para a inspeção de frascos em uma linha de envase utilizando dispositivos de baixo custo, como uma plataforma embarcada e uma câmera modular.

### 1.2 OBJETIVOS ESPECÍFICOS

Para atingir o objetivo geral, os seguintes objetivos específicos podem ser destacados:

- Definição da plataforma embarcada e da câmera modular adequadas às condições da aplicação;
- Criação da base de dados com imagens de frascos coletadas em linha de produção;
- Escolha e implementação dos algoritmos para detecção da presença de tampas em frascos baseado no estado da arte;
- Avaliação da performance do sistema quanto a adequação às condições da aplicação utilizando-se diferentes algoritmos em um sistema embarcado de baixo custo.

### 1.3 ESCOPO E RESTRIÇÕES

O sistema proposto será avaliado considerando-se os conjuntos de frascos e tampas encontrados na base de dados criada. Por isso, para a inspeção de outros

modelos de tampa ou frasco seria necessária a realização de ajustes de parâmetros e *setups* específicos no sistema.

O sistema a ser desenvolvido não será instalado em uma linha de produção real e avaliado nestas condições, visto que a inspeção será executada posteriormente à aquisição de todas as imagens da base de dados a ser criada.

Desta forma, não serão contempladas no desenvolvimento do sistema as questões relacionadas ao dispositivo de rejeite dos frascos reprovados, sendo que o mesmo se limitará a classificação de conformidade ou defeito dos frascos avaliados.

#### 1.4 APRESENTAÇÃO DO TRABALHO

Este trabalho está dividido em 6 capítulos. O primeiro capítulo contém a introdução e as principais definições do trabalho. O segundo capítulo apresenta a revisão bibliográfica dos temas relacionados ao processamento de imagem e visão computacional, necessários para o entendimento do trabalho proposto. O terceiro capítulo apresenta uma breve análise do estado da arte sobre inspeção visual automática de frascos. No quarto capítulo aborda-se a proposta de trabalho visando atender o objetivo de desenvolver um sistema de inspeção automática de baixo custo. O quinto capítulo apresenta os resultados obtidos pela avaliação dos sistema e dos métodos de inspeção implementados. Por fim, no sexto capítulo são apresentadas algumas considerações finais e perspectivas futuras.

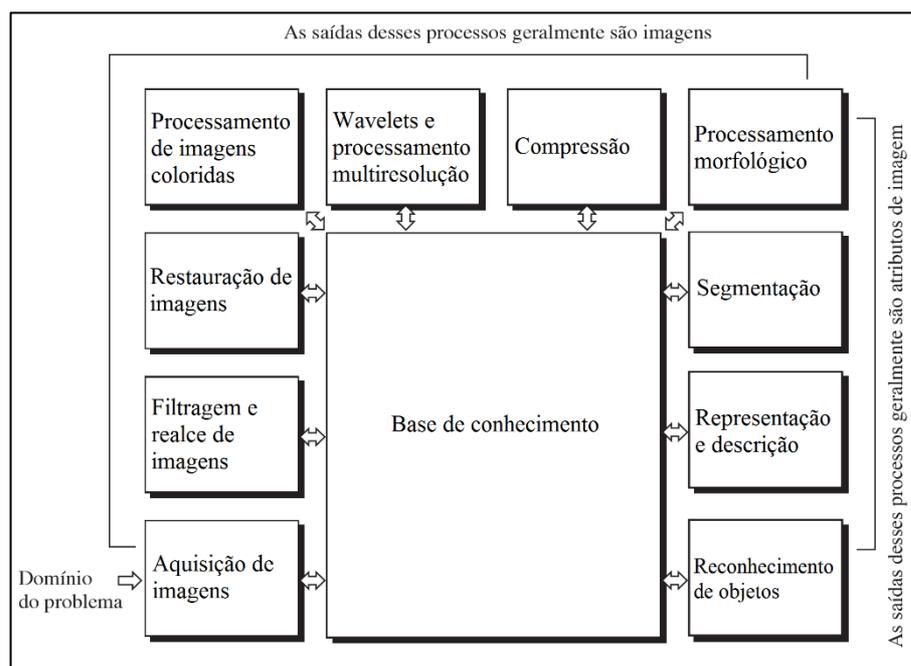
## 2 REVISÃO BIBLIOGRÁFICA

Neste capítulo será apresentada uma revisão bibliográfica sobre os temas relacionados ao trabalho proposto. Inicialmente são abordados os conceitos de visão computacional e inspeção automática. Em seguida, serão apresentados aspectos de aquisição e representação de imagens digitais, processamento e segmentação e os conceitos de detecção, descrição e comparação de características. Por último, será realizada uma breve análise dos sistemas embarcados de baixo custo e das ferramentas de visão computacional.

### 2.1 VISÃO COMPUTACIONAL E INSPEÇÃO AUTOMÁTICA

De acordo com os autores Marques Filho e Vieira Neto (1999, p. 9), define-se um Sistema de Visão Artificial (SVA) como “um sistema computadorizado capaz de adquirir, processar e interpretar imagens que correspondem a uma cena real”. Segundo Gonzalez e Woods (2010), as etapas fundamentais do processamento de imagem e reconhecimento de padrões podem ser representadas a partir de um diagrama, como pode ser visto na Figura 1.

Figura 1 - Etapas do processamento de imagem e as saídas obtidas.



Fonte: Adaptado de Gonzalez e Woods (2010)

O diagrama proposto por Gonzalez e Woods (2010) contém os passos fundamentais do processamento de imagem e reconhecimento de padrões. Além disso, são mostradas as saídas obtidas em cada etapa. Pode-se observar que as etapas que geram imagens como saída, geralmente estão relacionadas ao processamento, enquanto as etapas que geram atributos estão relacionadas com a interpretação do conteúdo da imagem.

O aumento da capacidade de processamento dos computadores e sistemas embarcados, aliado às novas arquiteturas empregadas nos sistemas de visão computacional têm permitido avanços consideráveis nas áreas de processamento de imagem e reconhecimento de padrões. Notam-se melhorias nas aplicações de inspeção industrial automática, reconhecimento de padrões em imagens aéreas, análises clínicas computacionais, detecção e reconhecimento de objetos em veículos autônomos, entre outras aplicações (DAVIES, 2012).

Com bases de dados mais consolidadas e processadores de maior desempenho, algumas das atividades de inspeção visual que antes eram realizadas exclusivamente por humanos agora abrem espaço para os sistemas artificiais. Contudo, os sistemas computacionais ainda apresentam limitações relacionadas às adaptações ao ambiente, como luminosidade e contraste (MARQUES FILHO; VIEIRA NETO, 1999).

A Tabela 1 apresenta um comparativo entre o sistema de visão humano e um sistema de visão computacional. É possível destacar as vantagens de um sistema artificial para atividades repetitivas e a altas velocidades, nas quais o resultado não pode estar sujeito a subjetividades.

Tabela 1 - Comparação entre o sistema visual humano e um sistema de visão computacional

(continua)

	<b>Sistema visual humano</b>	<b>Sistema de visão artificial</b>
Espectro	Limitado à faixa de luz visível (300 nm a 700 nm) do espectro de ondas eletromagnéticas.	Pode operar em praticamente todo o espectro de radiações eletromagnéticas, dos raios X ao infravermelho.
Flexibilidade	Extremamente flexível, capaz de se adaptar a diferentes tarefas e condições de trabalho.	Normalmente inflexível, apresenta bom desempenho somente na tarefa para a qual foi projetado.

(conclusão)		
Habilidade	Pode estabelecer estimativas relativamente precisas em assuntos subjetivos.	Pode efetuar medições exatas, baseadas em contagem de pixels e, portanto, dependentes da resolução da imagem digitalizada.
Cor	Possui capacidade de interpretação subjetiva de cores.	Mede objetivamente os valores das componentes R, G e B para determinação de cor.
Sensibilidade	Capaz de se adaptar a diferentes condições de luminosidade, características físicas da superfície do objeto e distância ao objeto. Limitado na distinção de muitos níveis diferentes de cinza, simultaneamente.	Sensível ao nível e padrão de iluminação, bem como à distância em relação ao objeto e suas características físicas. Pode trabalhar com centenas de tons de cinza, conforme projeto do digitalizador.
Tempo de resposta	Elevado, da ordem de 0,1 s.	Dependente de aspectos de hardware, podendo ser tão baixo quanto 0,001 s.
2-D e 3-D	Pode executar tarefas 3-D e com múltiplos comprimentos de onda (dentro do espectro de luz visível) facilmente.	Executa tarefas 2-D com relativa facilidade, mas é lento e limitado em tarefas 3-D.
Percepção	Percebe variações de brilho em escala logarítmica. A interpretação subjetiva de brilho depende da área ao redor do objeto considerado.	Pode perceber brilho em escala linear ou logarítmica.

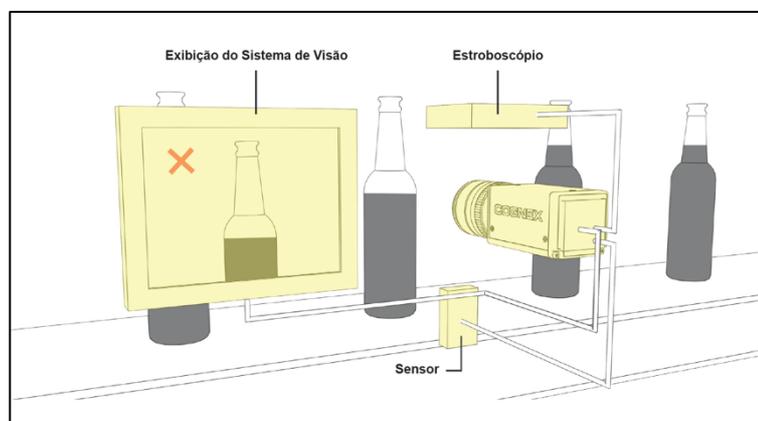
Fonte: Adaptado de Marques Filho e Vieira Neto (1999)

Atualmente, existem algumas aplicações de visão computacional voltadas a produção industrial, sendo diferenciadas principalmente pelo resultado que apresentam e a forma como este resultado é usado. Em aplicações de montagem automática, os resultados do sistema de visão são as coordenadas e a orientação do objeto a ser montado. No caso da inspeção visual automática, os produtos são avaliados e o resultado é uma condição de conformidade ou de rejeição, de acordo com os parâmetros pré-estabelecidos para o item inspecionado. Outras aplicações podem apresentar resultados úteis para a gestão da produção, como no caso de contagem de produtos produzidos e na identificação via código de barras ou similares (DAVIES, 2012).

A Figura 2 ilustra um exemplo de inspeção automática do nível de líquido em garrafas. Logo após terem sido enchidas, as garrafas passam continuamente por um sistema de inspeção baseado em visão computacional. Este sistema capta esta

passagem através de sensores de presença e registra uma imagem da garrafa. A imagem gerada é processada e comparada com um padrão de nível do líquido previamente estabelecido. Caso o resultado da comparação seja negativo, ou seja, se há menos líquido do que o valor estabelecido, o sistema gera um alerta. A partir deste alerta ações podem ser tomadas, como o descarte automático desta garrafa da linha de produção. Alguns sistemas como o da figura utilizam um estroboscópio para auxiliar na captura da imagem. Monitores também podem estar presentes no sistema para seleção de parâmetros ou exibição de informações relevantes ao processo.

Figura 2 - Exemplo de sistema de inspeção de nível de líquido em garrafas.



Fonte: Cognex (2018)

Uma característica importante da maioria das tarefas de inspeção automática é que elas ocorrem em tempo real, ou seja, o dispositivo de inspeção deve ser capaz de acompanhar a velocidade do processo de fabricação. Para atender a esta característica, as arquiteturas destes dispositivos podem contar com processadores *multicore* e *Graphics Processing Units* (GPU's), dedicadas ao processamento de imagem. Entretanto, nota-se que quanto melhor forem as especificações do *hardware* maior será o custo do dispositivo de inspeção. (DAVIES, 2012).

As etapas de aquisição de imagem, segmentação, representação e descrição e reconhecimento de objetos, presentes no diagrama da Figura 1, são consideradas as mais importantes em relação ao processo de inspeção visual destacadas na literatura (MARQUES FILHO; VIEIRA NETO, 1999). Sendo assim, estas etapas serão abordadas de forma mais detalhada nas próximas seções, ressaltando-se suas principais características e funcionalidades.

## 2.2 AQUISIÇÃO E REPRESENTAÇÃO DE IMAGENS DIGITAIS

A aquisição da imagem é um dos aspectos mais importantes dos sistemas de visão computacional. As deficiências nas capturas causadas por iluminação insuficiente, ruído ou características da câmera necessitam algoritmos complexos e alto processamento para a correção, e, em alguns casos, podem até inviabilizar a aplicação (DAVIES, 2012). Nesta seção estão revisadas as principais características relacionadas a aquisição e representação de imagens em formato digital.

### 2.2.1 Aquisição de imagens digitais

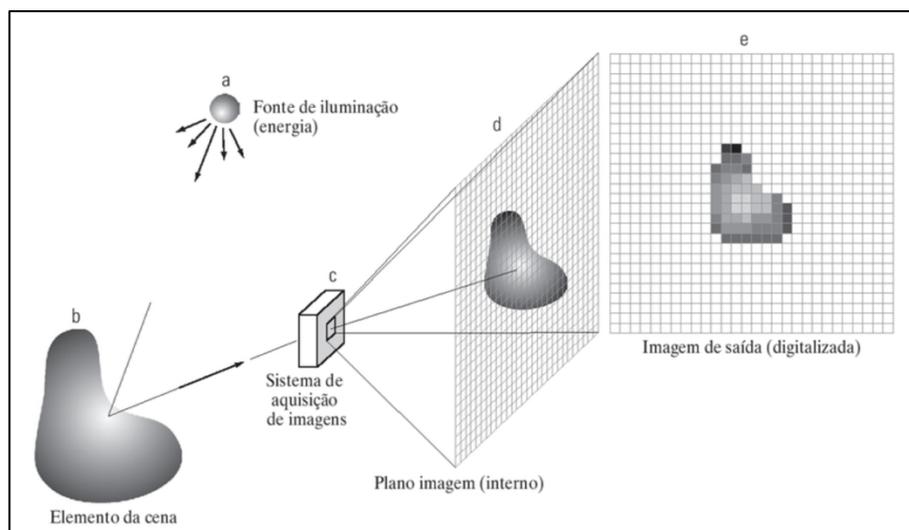
Existem algumas formas de aquisição de imagens, como dispositivos com um único sensor de captura, sensores por varredura de linha e sensores matriciais. Neste trabalho será revisada a formação de imagens através da captura da luz<sup>1</sup> refletida por um objeto em um sensor matricial. De forma resumida, pode-se dizer que a energia da radiação eletromagnética que reflete dos objetos, combinada com o material do sensor será transformada em um sinal de tensão elétrica. O sinal de saída dos sensores é a resposta analógica à recepção da luz, que será amostrada e quantizada para a formação da imagem (GONZALEZ; WOODS, 2010).

A Figura 3 exemplifica o processo típico de aquisição de uma imagem em que a energia provida por uma fonte de iluminação é refletida por um objeto da cena. Esta energia refletida é projetada em um plano focal que coincide com o arranjo dos sensores. A matriz de sensores ajusta a sua saída de acordo com a integral do sinal eletromagnético de entrada. Da leitura e conversão dos sinais de cada célula da matriz de sensores, é formada a imagem digital (GONZALEZ; WOODS, 2010). Nota-se que no caso de imagens coloridas, cada elemento do sensor terá 3 canais coletando o sinal no espaço de cores primárias. Neste caso, por meio do sistema de cores denominado *Red, Green, Blue* (RGB).

---

<sup>1</sup> Segundo Gonzalez e Woods (2010), luz é um tipo particular de radiação eletromagnética que pode ser percebida pelo olho humano, sendo que a banda visível do espectro esta compreendida em comprimentos de onda que vão de 430 nm (violeta) a 790 nm (vermelho).

Figura 3 - Processo de aquisição de imagem digital por um sensor matricial.



Fonte: Gonzalez e Woods (2010).

Entre os principais dispositivos sensores para conversão da imagem em sinais elétricos estão o sensor *Charge-Coupled Device* (CCD) e o *Complementary Metal Oxide Semiconductor* (CMOS). Ambos se caracterizam por serem sensores de carga acoplada, ou seja, armazenam e convertem em sinal elétrico uma carga proporcional a intensidade luminosa a que foram expostos. As principais diferenças entre os dois modelos consistem no menor custo, menor consumo energético, tamanho reduzido e menor interferência a ruído do sensor CMOS em relação ao CCD. Esses fatores fazem com que o sensor CMOS venha ocupando o espaço do CCD (MARQUES FILHO; VIEIRA NETO, 1999).

O processo de digitalização da imagem pode ser dividido em duas etapas: a amostragem e a quantização. Por amostragem entende-se como a etapa de digitalização das coordenadas, ou seja, transformar uma imagem contínua no espaço em diversos elementos de uma matriz finita de pixels. No processo de quantização são digitalizados os valores de luminosidade de cada pixel. Com isso, obtém-se valores discretos tanto para as coordenadas da imagem quanto para os valores de carga de cada pixel. Observa-se que a qualidade da imagem está diretamente relacionada a quantidade de pixels amostrados e ao número de níveis discretos de intensidade utilizados na quantização. Usualmente para imagens monocromáticas são utilizados 8 bits para a quantização de intensidade, gerando 256 níveis que variam do preto, ou valor 0, ao branco, ou valor 255 (GONZALEZ; WOODS, 2010).

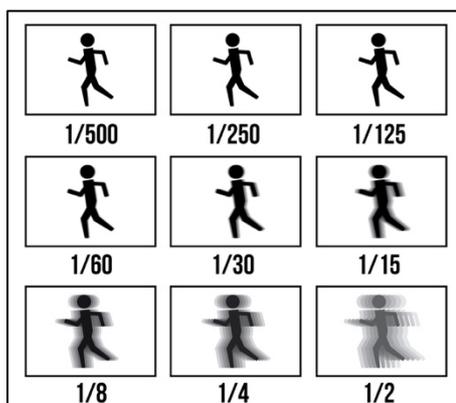
### 2.2.1.1 Aspectos da captura de objetos em movimento

A captura de imagens com objetos em movimento apresenta algumas particularidades que podem ser prejudiciais ao processo de inspeção visual automática. Na fotografia artística o efeito de *motion blur*, ou borrão de movimento, é amplamente utilizado para criar a percepção de ação. Entretanto, nas imagens utilizadas para inspeção visual esse efeito pode ocasionar diversos problemas, como, a impossibilidade de localizar, medir ou identificar um objeto em uma cena. Este efeito é causado principalmente pelas características de velocidade do obturador e forma de leitura da matriz de pixels da câmera (DAVIES, 2012).

O obturador de uma câmera fotográfica pode ser considerado com uma cortina. No momento de disparo da fotografia o obturador se abre deixando a luz entrar no sensor. Instantes depois, o mesmo se fecha escondendo o sensor e bloqueando a entrada de luz. Ao tempo que o obturador fica aberto é dado o nome de tempo de exposição, que pode ser considerado como sendo o inverso da velocidade do obturador. Se a velocidade do obturador for lenta, ou o tempo de exposição for longo, os alvos em movimento na foto aparecerão borrados no sentido do movimento. Por outro lado, capturas realizadas com baixo tempo de exposição ajudam a “congelar” a cena, aumentando a nitidez do alvo em movimento (MANSUROV, 2018).

A Figura 4 ilustra o resultado do efeito de *motion blur*. Nela é exibido o resultado da captura de um boneco fictício em movimento utilizando-se diferentes tempos de exposição. O primeiro quadro traz a captura com 2 ms de exposição e o último quadro com 0,5 s exposição.

Figura 4 - Captura de um alvo com diferentes velocidades do obturador.



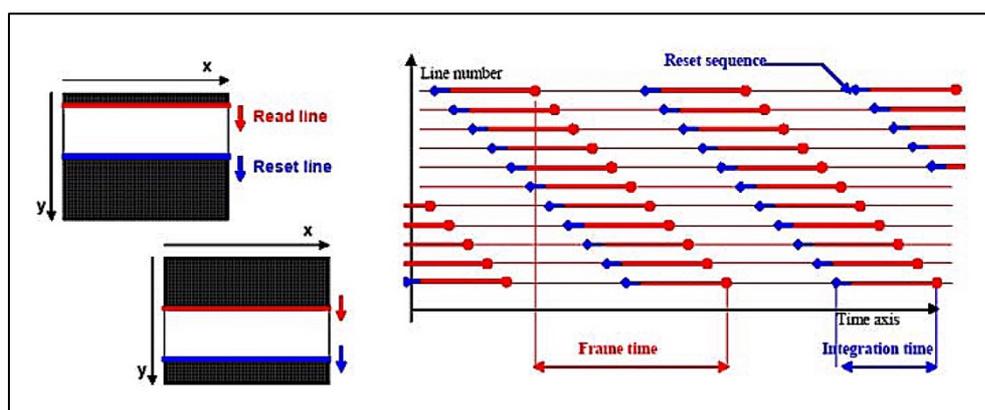
Fonte: Creative Live (2018)

O *motion blur* causado pelo tempo de exposição pode ser produzido por todos os tipos de câmera. Porém, além deste efeito, as capturas realizadas com câmeras CMOS podem apresentar outra forma de distorção devido a sua arquitetura e a forma como realizam a captura de um *frame*<sup>2</sup>.

Na arquitetura *global shutter*, normalmente associada a sensores CCD e obturadores mecânicos, a leitura de todos os pixels do sensor é feita no mesmo instante de tempo. Já na arquitetura *rolling shutter*, normalmente encontrada em câmeras CMOS com obturadores eletrônicos, esta leitura é realizada linha a linha, percorrendo-se a matriz em um sentido definido (JONES, 2018).

A Figura 5 mostra o processo de captura de um *frame* pela arquitetura *rolling shutter*. A linha azul na imagem representa a varredura de *reset*. Simultaneamente, a linha vermelha percorre a matriz representando a leitura dos valores armazenados nos elementos das linhas. Depois da leitura da última linha o *frame* estará completo, e, então, o processo será iniciado novamente para a captura do próximo *frame*. Note-se que neste tipo de câmera o tempo de exposição é dado pela diferença temporal entre o momento de *reset* e de leitura de cada linha do sensor (CHIAO; LEE; LAI, 2013).

Figura 5 - Diagrama temporal do funcionamento de um sensor *rolling shutter*.



Fonte: Ait-aider, Bartoli e Andreff (2007)

<sup>2</sup> Segundo Marques Filho e Vieira Neto (1999) *frame* (ou quadro) pode ser definido como sendo uma imagem dentro de uma sequência de imagens. Usualmente utilizado no contexto de imagem retirada de um vídeo.

Em movimentos horizontais a distorção aparece como uma inclinação dos elementos da cena, para a direita ou para a esquerda. No caso de movimentos verticais, a distorção aparece deixando os objetos da cena alongados ou comprimidos em relação a sua real dimensão. Para reduzir a distorção causada pelo efeito de *rolling shutter* e *motion blur* aconselha-se a diminuição do tempo de exposição (CHIAO; LEE; LAI, 2013).

#### 2.2.1.2 Iluminação controlada

Uma iluminação adequada pode simplificar o processo de inspeção automática através da produção de imagens mais nítidas e precisas. Segundo Cognex (2012):

Uma boa imagem é o resultado de foco e iluminação adequados. Imagens com contraste ruim e iluminação irregular exigem muito mais esforço para o computador de imagem, o que, em última análise, aumenta o tempo de processamento e pode aumentar o número de análises incorretas das imagens.

A escolha da técnica de iluminação adequada está diretamente relacionada com a aplicação. De acordo com Cognex (2012), existem 5 técnicas básicas de iluminação para inspeção automática computacional, sendo as principais técnicas e aplicações correspondentes:

- Iluminação de fundo: utilizada para acentuar as formas do objeto inspecionado através da intensificação das silhuetas. Produz um contraste adequado, o que ajuda a evidenciar as bordas das peças. Amplamente aplicada no processo de aferição de peças. Não é indicada para a análises de superfícies, uma vez que o alto contraste dificulta a visualização.
- Iluminação frontal direta: apresenta considerável facilidade de montagem e um resultado satisfatório no aumento do contraste. Este método pode ser utilizado para a inspeção de superfícies. Para minimizar o efeito das sombras criadas, podem ser utilizados mais do que um ponto de luz, como no caso dos anéis de iluminação. Luzes difusas auxiliam na uniformidade da iluminação e diminuem a criação de sombras.
- Iluminação estruturada: neste método normalmente são usados lasers ou fibras óticas para criar uma iluminação regular e colimada em uma forma conhecida. As medidas são feitas avaliando-se a distorção do reflexo da luz em relação ao

seu formato original e conhecido. Empregada principalmente na inspeção de relevos e diferenças de alturas em superfícies. Este método não é aconselhado para ser utilizado com objetos em movimento ou que apresentem características de absorção da luz.

- Iluminação difusa *on-axis*: usada em aplicações com o objetivo de inspecionar superfícies planas e brilhantes, ou o interior de cavidades. Através do uso de um espelho prateado, normalmente com 50% de reflexão, esta técnica permite que se gere uma iluminação perfeitamente alinhada em relação ao ângulo da câmera, que estará posicionada atrás do espelho.
- Iluminação difusa *off-axis*: popularmente conhecida como luz de dia nublado, esta técnica é utilizada para a eliminação de sombras e pontos de alto brilho encontrados, principalmente, em peças metálicas. Nesta técnica, antes da luz ser direcionada a peça, ela passa por uma superfícies difusa, normalmente em formato de cúpula. Desta forma, evita-se que o objeto inspecionado reflita um ponto de alto brilho concentrado.

As fontes de iluminação podem ser do tipo contínua ou estroboscópica. Fontes contínuas são adequadas para a inspeção de superfícies e aplicações gerais. As fontes estroboscópicas podem ser utilizadas para o efeito de congelamento da ação, auxiliando na captura de objetos em movimento. Porém, esta luz precisa ser precisamente controlada e sincronizada com o tempo de exposição da câmera. Do contrário, as imagens poderão aparecer com iluminações desparelhas (COGNEX, 2012).

A fonte de luz também é um quesito que pode ser adequado de acordo com a aplicação proposta. A Tabela 2 apresenta um comparativo qualitativo entre as principais fontes de luz artificiais, segundo Cognex (2012).

Tabela 2 - Comparativo entre as fontes de iluminação.

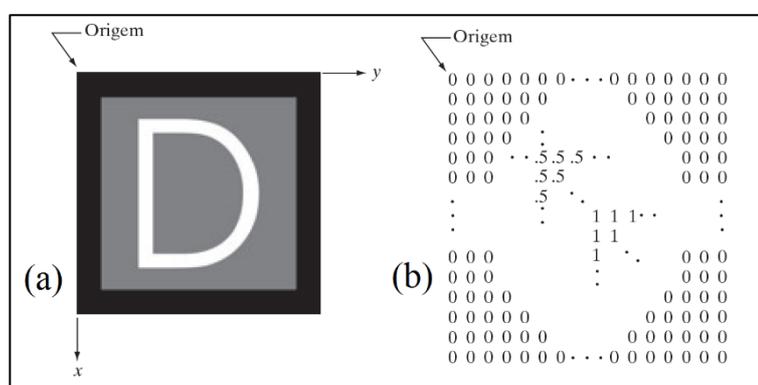
<b>Tipo</b>	<b>Estroboscópica</b>	<b>Custo</b>	<b>Intensidade</b>	<b>Vida</b>	<b>Calor</b>
Halógena	Ruim	Razoável	Bom	Ruim	Ruim
Incandescente	Ruim	Bom	Razoável	Ruim	Ruim
Fluorescente	Ruim	Bom	Ruim	Razoável	Razoável
Laser	Bom	Ruim	Bom	Bom	Bom
Xenon	Bom	Ruim	Bom	Ruim	Ruim
LED	Bom	Razoável	Bom	Bom	Bom

Fonte: Cognex (2012)

## 2.2.2 Representação de imagens digitais

A representação de imagens em forma matricial é amplamente utilizada e está presente na maioria dos dispositivos como monitores, impressoras e câmeras. A Figura 6a ilustra esta representação com imagem em escala de cinza, posta sobre os eixos  $x$  e  $y$ . A Figura 6b ilustra de forma numérica o valor de intensidade dos pixels que formam a imagem presente na Figura 6a (GONZALEZ; WOODS, 2010).

Figura 6 - Formas básicas de representação de imagens digitais.



Fonte: Adaptado de Gonzalez e Woods (2010).

A função que representa uma imagem também pode ser apresentada com uma matriz  $M \times N$ , onde a intensidade de cada pixel é o valor contido em cada elemento da matriz, conforme a Equação 1 (GONZALEZ; WOODS, 2010).

$$f(x,y) = \begin{bmatrix} f(0, 0) & f(0, 1) & \dots & f(0, N-1) \\ f(1, 0) & f(1, 1) & \dots & f(1, N-1) \\ \vdots & \vdots & \dots & \vdots \\ f(M-1, 0) & f(M-1, 1) & \dots & f(M-1, N-1) \end{bmatrix} \quad (1)$$

Os valores de  $M$  e  $N$  presentes na Matriz 1 não costumam apresentar limitações, além de serem inteiros e positivos, já que estão relacionados somente com a resolução da imagem. Porém, para os valores de intensidade ( $L$ ) algumas restrições são impostas. Usualmente, este valor é quantizado como uma potência inteira de 2, sendo, tipicamente, formado por 8 bits para a representação de intensidade. O número total de bits  $b$  necessários para a representação de uma imagem pode ser calculado

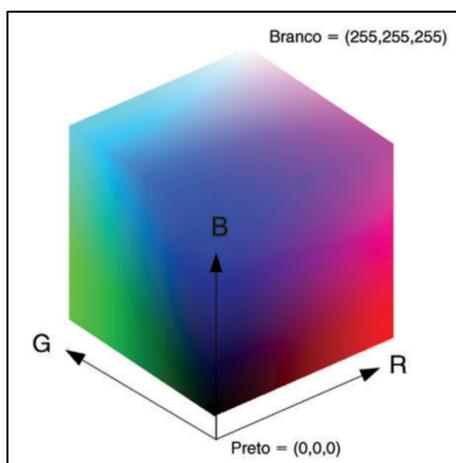
pela Equação 2. Nesta equação,  $k$  é o valor de bits por elemento da matriz,  $M$  o número de linhas e  $N$  o número de colunas (GONZALEZ; WOODS, 2010).

$$b = M \times N \times k \quad (2)$$

### 2.2.2.1 Espaço de cores RGB

Atualmente, é comum que os dispositivos de captura de imagem realizem o registro de forma colorida e utilizem uma representação no espaço de cores RGB. As imagens no espaço de cores RGB podem ser consideradas como matrizes tridimensionais, ou conceitualmente, como sendo 3 planos bidimensionais, cada um contendo as informações de um canal de cor. A Figura 7 ilustra o espaço de cores como sendo um cubo em que cada vetor contém a porção de um dos canais. Os três eixos desse vetor possuem a mesma faixa de valores, que variam de 0 a 255 em imagens com 1 byte na escala de intensidade (SOLOMON; BRECKON, 2013).

Figura 7 - Espaço de cores RGB como um cubo tridimensional.



Fonte: Solomon e Breckon (2013)

Entretanto, segundo Solomon e Breckon (2013), a conversão de uma imagem colorida em escala de cinza é o ponto inicial do processamento e análise computacional de imagens, pois, este processo simplifica e reduz significativamente a quantidade de dados. Ainda Solomon e Breckon (2013):

Embora uma imagem em escala de cinza contenha menos informação do que uma imagem em cores, a maioria das informações importantes e relacionadas às características da imagem é mantida, como bordas, regiões, manchas, junções etc. Algoritmos de detecção de características e de processamento operam sobre a versão da imagem em escala de cinza.

Conforme apresentado por Solomon e Breckon (2013), uma imagem em RGB pode ser convertida em escala de cinza através da Equação 3. Nesta equação,  $(n, m)$  representa a posição do pixel na imagem em escala de cinza,  $(n, m, c)$  a posição do pixel em um dado canal de cor e  $I_{cores}$  a intensidade luminosa no canal.

$$I_{escala\ de\ cinza}(n, m) = \alpha I_{cores}(n, m, r) + \beta I_{cores}(n, m, g) + \gamma I_{cores}(n, m, b) \quad (3)$$

Nota-se pela equação que a conversão é feita pela soma ponderada dos canais vermelho, verde e azul, e leva-se em consideração os coeficientes  $\alpha$ ,  $\beta$  e  $\gamma$ . Esses coeficientes são utilizados porque a percepção de cor do olho humano é naturalmente mais sensível as cores vermelha e verde, o que torna a distribuição não uniforme.

## 2.3 SEGMENTAÇÃO E EXTRAÇÃO DE CARACTERÍSTICAS

Nesta seção são abordados os conceitos de processamento, segmentação, extração e descrição de características e comparação de características, pertinentes a proposta deste trabalho. Estes conceitos são componentes essenciais do sistema de visão computacional e estão presentes em aplicações com diferentes finalidades, que vão desde inspeção automática industrial a sistemas de visão para veículos autônomos (DAVIES, 2012).

### 2.3.1 Detector de bordas de Canny

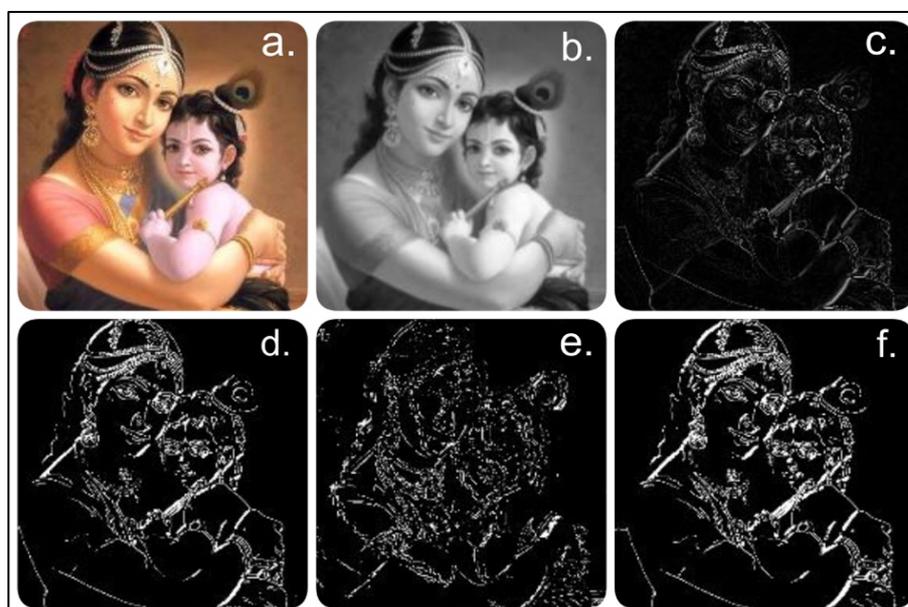
Segundo os autores Solomon e Breckon (2013), o método de Canny é reconhecido como o mais completo método de detecção de borda. Este método visa a satisfação dos seguintes critérios:

- Todas as bordas deverão ser localizadas e devem ser o mais próximas possível das bordas verdadeiras;
- As distância entre o ponto de borda marcado pelo detector e o centro da borda original deve ser mínima;

- O detector não deve gerar mais do que um ponto de borda para a mesma borda original.

O procedimento para aplicação da técnica de Canny é composto pelas etapas de suavização da imagem, determinação da intensidade da borda, cálculo de direção da borda, digitalização da direção da borda, supressão não máxima e, por último, histerese. A Figura 8 ilustra o resultado de algumas destas etapas, iniciando pela imagem original na Figura 8a e terminado com o resultado final do procedimento na Figura 8f.

Figura 8 - Exemplo de aplicação do detector de bordas de Canny.



Fonte: Adaptado de Project (2012).

A primeira etapa do processo de detecção de Canny consiste em suavizar a imagem de entrada através da aplicação de um filtro gaussiano<sup>3</sup>. Esta etapa é aplicada porque os operadores de gradiente são sensíveis ao ruído. Observa-se que quanto maior o tamanho do núcleo do filtro aplicado maior será a redução do ruído. Contudo, núcleos maiores implicam em um maior erro na localização das bordas. A Figura 8b ilustra o resultado da conversão para escala de cinza e aplicação do filtro gaussiano na imagem presente na Figura 8a.

<sup>3</sup> Conforme Gonzalez e Woods (2010), o filtro gaussiano é definido como um filtro linear de suavização, onde a máscara utilizada é composta por pesos diferentes para cada posição. A distribuição dos pesos segue uma distribuição gaussiana. O resultado da suavização é dado pela média ponderada.

Na segunda etapa ocorre a determinação da intensidade da borda. Através da aplicação dos operadores de Sobel<sup>4</sup> são obtidas as primeiras derivadas nos sentidos horizontal e vertical. A soma dos módulos destas duas derivadas será o valor de intensidade da borda.

Na terceira etapa é calculado o ângulo de direção da borda. Este ângulo em graus pode ser encontrado através a Equação 4, onde  $G_y(x,y)$  é o valor da derivada na direção vertical,  $G_x(x,y)$  é o valor da derivada na direção horizontal e  $\theta$  é o ângulo de direção da borda.

$$\theta = \tan^{-1} \frac{G_y(x,y)}{G_x(x,y)} \quad (4)$$

Na quarta etapa deve-se transformar o valor de direção da borda encontrado anteriormente em um valor discreto determinado. Considerando-se o ponto central de uma região de 3x3 pixels, pode-se definir 4 orientações possíveis: 0° para vizinhos na horizontal, 90° para vizinhos na vertical, 45° e -45° para vizinho nas diagonais.

A técnica de supressão não máxima é aplicada na quinta etapa. Nesta fase os pontos da imagem que não foram classificados como uma região de borda são suprimidos, formando assim uma região fina na imagem de saída. O resultado desta etapa do processo esta ilustrado na Figura 8c.

O último passo consiste em rastrear os pixels que não foram suprimidos para aplicação do limiar através do conceito de histerese. Para a histerese são utilizados dois valores de *threshold* distintos, sendo um baixo denominado  $T_1$  e um alto denominado  $T_2$ . Os pontos que estiverem acima do valor  $T_2$  são diretamente considerados bordas fortes. Os pontos que estiverem abaixo do valor  $T_1$  serão automaticamente rejeitados como borda. Os pontos entre os valores  $T_1$  e  $T_2$  que estiverem conectados a uma borda forte serão mantidos, do contrário, os pontos serão suprimidos. As Figuras 8d, 8e e 8f representam a aplicação do limiar para as bordas fortes, fracas e o resultado final após uso da histerese, respectivamente.

---

<sup>4</sup> Os operadores de Sobel, ou filtros de Sobel, foram propostos para aproximar a derivada de primeira ordem do gradiente da imagem. Estas derivadas indicam os pontos de transição abrupta da luminosidade na imagem, característica dos locais que onde ocorrem as bordas (SOLOMON; BRECKON, 2013).

### 2.3.2 Extração e descrição de características

A extração e a comparação de características está na base de problemas de visão computacional, como, detecção e identificação de objetos, inspeção de superfícies, reconhecimento de faces, entre outras. Entre os principais algoritmos para extração e descrição de características em tempo real estão *Scale-Invariant Feature Transform* (SIFT), *Speeded-Up Robust Feature* (SURF) e o *Oriented FAST and Rotated BRIEF* (ORB) (RUBLEE *et al.*, 2011; NOBLE, 2016; KUMAR; K, 2014).

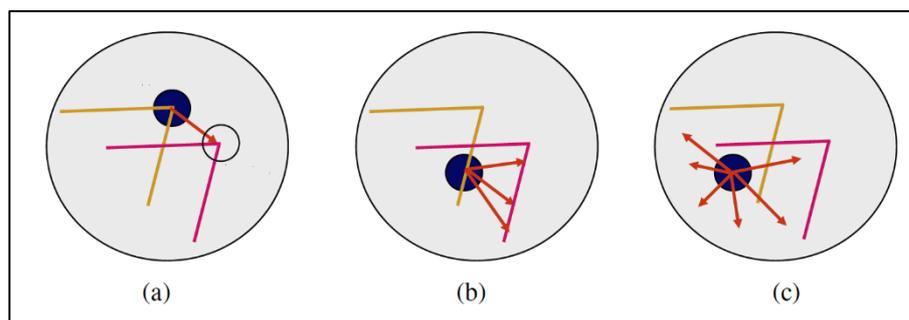
Dentre os algoritmos de extração mais reconhecidos na literatura, o ORB se destaca por ser indicado para aplicações em tempo real, podendo ser tão preciso quanto SIFT e mais preciso do que o SURF. Em termos de tempo de processamento, o descritor ORB é praticamente duas ordens de grandeza mais rápido quando comparado ao SIFT (RUBLEE *et al.*, 2011).

#### 2.3.2.1 Features from Accelerated Segment Test (FAST)

Uma das formas de extração de características de imagens é através da detecção de cantos. Os cantos de uma imagem podem ser definidos como os locais onde uma borda troca de direção de forma abrupta. Para exemplificar este caso, observa-se a Figura 9, que ilustra a comparação de duas imagens sobrepostas. Na Figura 9c, o ponto escolhido para a comparação é uma região plana e com pouca variação de intensidade. Como consequência da pequena variação presente na área avaliada o mesmo ponto pode ser encontrado em vários locais na imagem.

Observando a Figura 9b, o ponto selecionado para comparação é uma borda. Apesar da borda da imagem ser mais específica do que uma região plana, ela ainda pode apresentar mais do que um ponto em comum ao longo de um segmento. Por fim, na Figura 9a o canto da imagem é utilizado como característica para a comparação. Observa-se que esta característica se sobressai em relação as duas anteriores, visto que os cantos tendem a aparecer em locais específicos e não se replicam ao longo da imagem (SZELISKI, 2010).

Figura 9 - Comparação de pontos para extração de características de imagens.

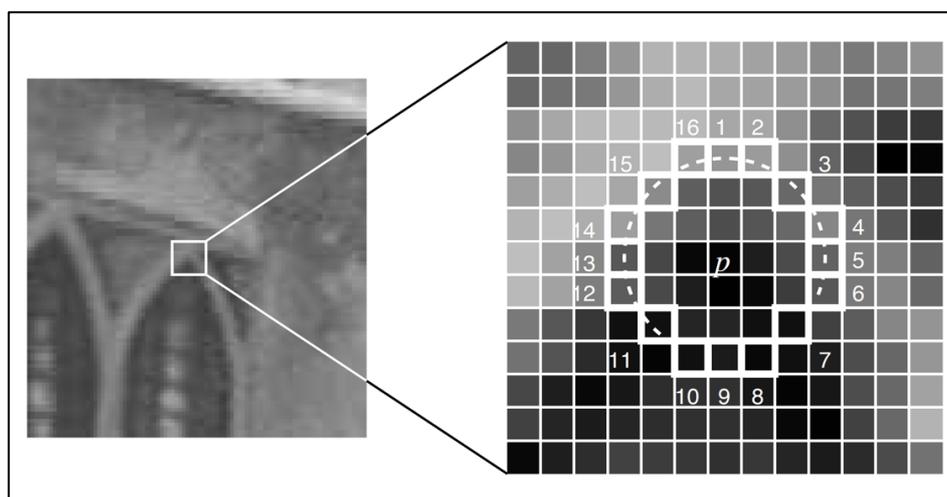


Fonte: Adaptado de Szeliski (2010)

Com o intuito de otimizar o processamento para aplicações de tempo real, os autores Rosten e Drummond (2010) desenvolveram um algoritmo denominado FAST para detecção de cantos. Este algoritmo é baseado na análise de um conjunto de pixels ao redor de um ponto de interesse e na utilização de aprendizado de máquina para a classificação desse conjunto.

A primeira etapa proposta no algoritmo consiste na seleção de um pixel  $p$  para a avaliação. Em seguida, um círculo de 16 pixels ao redor do ponto de interesse é avaliado com base em um valor de *threshold*  $t$  definido previamente. O pixel avaliado é classificado como um canto, se houver um determinado número  $n$  de pixels contínuos com valor de intensidade superior a  $p + t$ , ou inferior de  $p - t$ . A Figura 10 exibe a seleção de um ponto de avaliação, e demonstra a forma como são distribuídos os 16 pixels ao redor deste ponto.

Figura 10 - Distribuição da região de avaliação de cantos.



Fonte: Rosten e Drummond (2010).

Com intuito de elevar o desempenho da técnica, o algoritmo executa um teste rápido verificando primeiramente os pixels 1, 9, 5 e 13, representados na distribuição ilustrada pela Figura 10. Se um número mínimo de 3 destes 4 testes for superior a  $p + t$ , ou inferior a  $p - t$ , os pixels restantes são verificados. Caso contrário, o pixel  $p$  é descartado como possível canto.

Apesar de uma performance adequada para aplicações de tempo real, os autores enumeram alguns pontos fracos deste detector de cantos:

1. O teste rápido dos 4 primeiros pixels não generaliza bem para um valor de  $n$  pixels contínuos menor do que 12.
2. A escolha e a distribuição dos pixels do teste rápido possuem suposições implícitas a distribuição das características aparentes.
3. Vários cantos são detectados próximos uns aos outros.

A fim de solucionar os dois primeiros pontos fracos do teste rápido e elevar o desempenho do algoritmo, os autores utilizaram uma abordagem de aprendizado de máquina. Entretanto, nota-se que o conjunto de treinamento não abrange todos os possíveis cantos que podem ser encontrados em imagens. Este fato faz com que este método seja ligeiramente menos heurístico do que o modelo convencional.

Já para atender o terceiro ponto fraco citado pelos autores, estes aplicam a técnica de supressão não máxima. Nesta abordagem são somadas as diferenças entre os pixels na seção contínua  $n$  em relação ao pixel central. O ponto que obtiver a maior soma de diferenças dentro de uma região determinada será mantido, enquanto o restante será suprimido.

### 2.3.2.2 Binary Robust Independent Elementary Features (BRIF)

Os algoritmos descritores de características têm com função realizar a descrição de um trecho da imagem na forma de um vetor de informações. Estes trechos podem ser áreas no entorno de um ponto de interesse ou um canto da imagem anteriormente localizado. Os algoritmos como o SIFT e o SURF usam como base o Histogram of Oriented Gradients (HOG) para realizar a descrição destes trechos. Outros algoritmos, como o BRIF, realizam esta descrição através da criação de vetores binários.

Os algoritmos baseado em HOG apresentam um alto custo computacional visto que para os cálculos dos gradientes da imagem são necessárias as derivadas de cada pixel do trecho a ser descrito. Já no caso dos descritores binários, a sua construção esta baseada na comparação direta de pares de pixels dentro do trecho da imagem a ser descrita. Esta característica faz com que os descritores binários apresentem um menor custo computacional (CALONDER *et al.*, 2010).

Como intuito de diminuir o tempo de processamento, os autores Calonder *et al.* (2010) desenvolveram um descritor de características binário denominado BRIEF. A primeira etapa deste algoritmo consiste na aplicação de um filtro gaussiano para a redução de ruído. Após a imagem estar filtrada, este algoritmo seleciona um pixel denominado  $p1$ , e então, seleciona um segundo pixel  $p2$ . Se o valor de intensidade do pixel  $p1$  for superior ao valor de intensidade do pixel  $p2$ , armazena-se o valor 1 dentro de um vetor de bits. Caso contrário, se o valor de  $p1$  for igual ou inferior ao valor de  $p2$ , armazena-se o valor 0 neste mesmo vetor. A escolha do número de pares de pixels que serão testados pode ser 128, 256 ou 512, sendo que esta escolha também determina o tamanho do vetor de descrição.

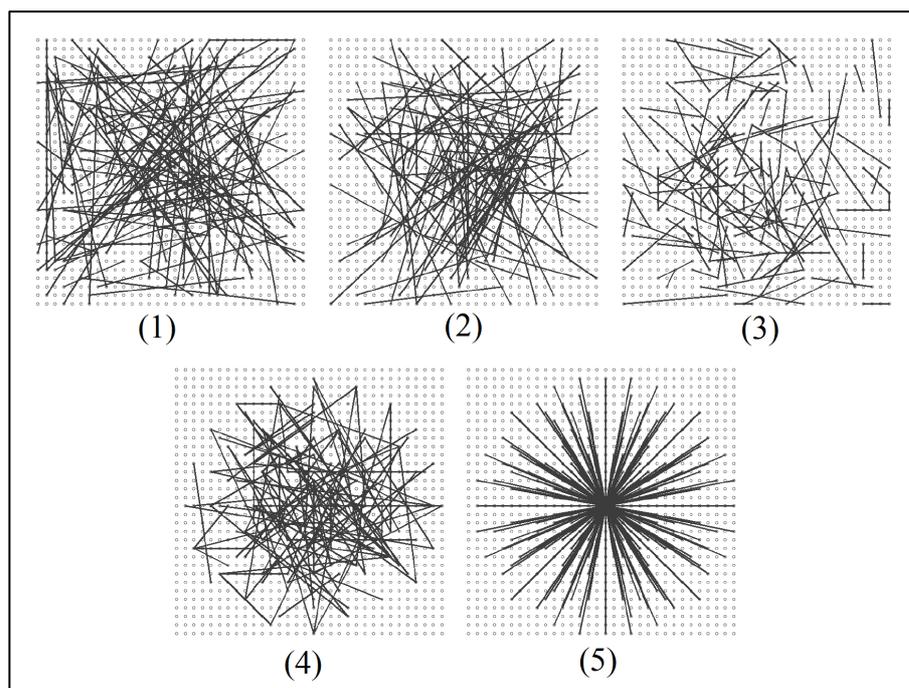
A geração do vetor que descreverá a imagem pode ser feita a partir de inúmeras combinações de pares de pixels dentro de um mesmo trecho. Com o objetivo de encontrar a distribuição que melhor descreve o trecho analisado os autores realizaram 5 simulações. A Figura 11 ilustra a distribuição utilizada em cada uma destas simulações. Nesta figura, cada ponto representa um pixel e cada traço representa um par de teste, sendo que o ponto de referência está no centro da região descrita. Foram traçados 128 pares em cada método de distribuição.

Dentre os métodos de distribuição presentes na Figura 11, todos, exceto o método 3, foram traçados aleatoriamente. Os métodos de distribuição ilustrados nesta figura foram determinados segundo os seguintes critérios:

1. Os pontos  $p1$  e  $p2$  são aleatoriamente amostrados;
2. Os pixels  $p1$  e  $p2$  são amostrados aleatoriamente usando uma distribuição gaussiana, o que significa que os locais mais próximos do centro do trecho são preferidos;
3. A primeira localização  $p1$  é amostrada a partir de uma distribuição gaussiana centrada em torno da origem, enquanto a segunda localização  $p2$  é amostrada a partir de outra gaussiana centrada em  $p1$ . Este método força os testes a serem mais locais;

4. Os pontos  $p1$  e  $p2$  são amostrados aleatoriamente a partir de uma coordenada polar;
5. O ponto  $p1$  está sempre localizado na origem. Os pontos  $p2$  são distribuídos em uma coordenada polar em relação a  $p1$ .

Figura 11 - Distribuição dos pares de teste de acordo com os métodos propostos.



Fonte: Adaptado de Calonder *et al.* (2010)

De acordo com os autores Calonder *et al.* (2010), o método de distribuição número 2 foi o que obteve a maior taxa de reconhecimento nas simulações realizadas. Por este motivo, este foi o método adotado no descritor BRIEF.

### 2.3.2.3 Oriented FAST and Rotated BRIEF (ORB)

O algoritmo ORB foi desenvolvido pelos autores Rublee *et al.* (2011) com base no detector de pontos principais FAST e no descritor de características BRIEF. Como forma de melhoria, foram implementadas funções adicionais no algoritmo ORB. A primeira melhoria foi o uso de um mecanismo de compensação da orientação, tornando-o invariante a rotação. Além disso, no algoritmo ORB utilizou-se um método de aprendizagem para identificação dos pares de amostragem ideais, enquanto no BRIEF são utilizados pares de amostragem escolhidos aleatoriamente.

A compensação de rotação utiliza a relação angular entre o centroide e o ponto de interesse ou canto da imagem. Para o cálculo do ângulo de rotação considera-se que o canto da imagem está centrado em relação ao trecho que está sendo avaliado. O cálculo do centroide inicia pelo cálculo dos momentos do trecho avaliado através da Equação 5, onde,  $m_{pq}$  representa o momento da imagem,  $x$  e  $y$  são as coordenadas de cada ponto avaliado,  $I(x,y)$  é o valor de intensidade do pixel,  $p$  e  $q$  representam os índices da ordem do momento calculado.

$$m_{pq} = \sum_{x,y} x^p y^q I(x,y) \quad (5)$$

Com o resultado dos momentos da imagem pode-se encontrar as coordenadas do centroide através da Equação 6, onde,  $m_{00}$ ,  $m_{01}$  e  $m_{10}$  são os momentos encontrados para cada ordem e  $C$  é a coordenada do centroide encontrado.

$$C = \left( \frac{m_{10}}{m_{00}}, \frac{m_{01}}{m_{00}} \right) \quad (6)$$

Um vetor pode então ser definido a partir do centro do trecho avaliado até o centroide calculado. O ângulo desse vetor em relação ao ponto central pode ser encontrado a partir da Equação 7, onde  $atan2$  é a versão sensível ao quadrante da função arcotangente.

$$\theta = atan2(m_{01}, m_{10}) \quad (7)$$

Depois de encontrada a orientação do trecho avaliado, pode-se aplicar uma rotação canônica, obtendo-se desta forma a invariância de rotação proposta.

A seleção dos pares de teste apresenta duas importantes propriedades. A primeira propriedade é a autonomia dos pares de teste. Pares de teste que não estão relacionados entre si trazem novas informações para o descritor, maximizando assim a quantidade de informação que o descritor carrega.

A segunda propriedade da seleção é a alta variância dos pares. Alta variância torna uma característica mais discriminativa, uma vez que responde de forma diferente às entradas. A fim de atender as duas propriedades de seleção, os autores utilizam uma abordagem de aprendizagem de máquina.

Para a realização do treinamento, os autores utilizaram um conjunto contendo 300.000 trechos de imagem com cantos ou pontos de interesse. O resultado final deste treinamento são os 256 pares de teste que apresentaram a maior autonomia entre si e a maior variância. Logo, estes pares são estabelecidos como o padrão de amostragem do algoritmo ORB e resultam em um vetor de descrição com 32 bytes de tamanho.

#### 2.3.2.4 Comparação de características

Conforme os autores Calonder *et al.* (2010), uma das vantagens do uso de descritores de características binários está na forma de comparação que pode ser utilizada. Descritores binários são formados por vetores de bits que podem ser 0 ou 1, de acordo com os testes dos pares de pixels. Considerando-se esta característica, pode-se utilizar como métrica de comparação o método da distância de Hamming.

O método da distância de Hamming consiste em aplicar um operador Ou-Exclusivo (XOR) entre os bits dos descritores das duas imagens que serão comparadas. O resultado deste operador XOR será a diferença entre os dois descritores. A soma dos bits com valor 1 que resultaram da operação XOR indica a proximidade entre os dois descritores. Quanto maior for o resultado da soma, maior é a diferença entre os descritores. Caso o resultado da soma for 0, os dois descritores são idênticos, ou seja, os dois trechos de imagem representados por estes vetores são iguais.

Este método de comparação pode ser executado rapidamente em Central Processing Units (CPU) modernas, pois, geralmente estas possuem instruções específicas para a operação XOR.

## 2.4 SISTEMAS EMBARCADOS E FERRAMENTAS DE DESENVOLVIMENTO

O progresso das aplicações de visão computacional vem impulsionando o desenvolvimento de novos sistemas embarcados, ou Single Board Computers (SBC), capazes de lidar com estas aplicações. Dentre estes novos modelos de sistemas destacam-se as plataformas Nvidia Jetson TX1, Nvidia Jetson TK1 e a Raspberry Pi 3. Os dois primeiros se destacam por suas especificações contando com CPU

*multicore* e GPU dedicada, sendo indicados para aplicações de alto desempenho. Já as plataformas Raspberry Pi 3 destacam-se pelo baixo custo, especificações intermediárias, ampla conectividade a periféricos e suporte a sistema operacional Linux contado com ferramentas de desenvolvimento, como as bibliotecas OpenCV e TensorFlow (SACHAN, 2017).

### 2.4.1 Raspberry Pi 3 Model B e Câmera

A plataforma de desenvolvimento Raspberry Pi 3 Model B está entre as mais populares e utilizadas atualmente. Contando com a possibilidade de rodar sistemas operacionais de distribuição Linux como o Raspbian e Ubuntu e contendo uma porta de interface dedicada a câmera, esta placa possibilita o desenvolvimento de aplicações do visão computacional. A Tabela 3 apresenta as principais especificações técnicas desta plataforma, como, frequência de operação e modelo do processador, capacidade de memória *Random Access Memory (RAM)* e portas para interface e troca de sinais digitais (RASPBERRYPI, 2018).

Tabela 3 - Especificações técnicas da Raspberry Pi 3 Model B

Modelo da CPU	Quad Core Broadcom BCM2837 64bit
Frequência da CPU	1,2 GHz
Memória RAM	1 GB
Interfaces	40 portas digitais programáveis Interface para câmera (CSI) Conector de vídeo HDMI 4 portas USB 2.0
Conectividade	BCM43438 wireless LAN Bluetooth 4.2 BLE
Dimensões	85 x 56 x 17 mm

Fonte: Adaptado de RaspberryPi (2018).

A câmera Raspberry Pi Camera Module v2 pode ser conectada através da porta de interface CSI diretamente a placa Raspberry Pi 3 Model B. Esta câmera possui um sensor CMOS do modelo Sony IMX219 e permite a captura de imagens e vídeos em resolução de até 1920x1080 pixels a 30 FPS, ou, 640x480 pixels a até 90 FPS. O controle e as configurações desta câmera podem ser realizados através da biblioteca Picamera. Esta biblioteca é *Open Source* e disponibiliza uma interface em Python com

o módulo da câmera. Através desta interface é possível realizar algumas configurações, como, por exemplo, o controle do formato de captura, a resolução de saída da imagem, a taxa de captura, o tempo de exposição, entre outras. Além disso, é através desta interface que é realizada a aquisição de imagens por esta câmera (JONES, 2018).

#### 2.4.2 Biblioteca OpenCV

A *Open Source Computer Vision Library* (OpenCV) é uma biblioteca de software de visão computacional e aprendizado de máquina em código aberto. Esta biblioteca está escrita em linguagem C e C++ e é executada por sistemas Linux, Windows e Mac OS. Além disso, existem interfaces para programação em JAVA, Python e MATLAB (KAEHLER; BRADSKI, 2016).

A biblioteca OpenCV foi desenvolvida com foco em aplicações de tempo real e eficiência computacional. Além disso, também tem como objetivo fornecer uma infraestrutura simples, de modo que seja possível auxiliar e agilizar o desenvolvimento de sofisticadas aplicações de visão computacional. Para atender este objetivo, a biblioteca conta com mais de 500 funções incorporadas, que abrangem aplicações de inspeção industrial, segurança, calibração de câmeras, sistemas de visão *stereo*, robótica, entre outras áreas. Atualmente esta ferramenta é amplamente utilizada por cientistas e desenvolvedores, tanto para fins acadêmicos quanto comerciais (KAEHLER; BRADSKI, 2016).

### 2.5 AVALIAÇÃO DE DESEMPENHO DE CLASSIFICADORES

Uma das formas de exibição dos resultados obtidos por um classificador é através de uma matriz de confusão. A matriz de confusão pode ser utilizada para expressar os resultados obtidos por sistemas classificadores, podendo ser construída para duas ou mais classes. Considerando-se duas classes conhecidas, sendo elas a classe gato e a classe cachorro, a Tabela 4 apresenta uma matriz de confusão para expressar os resultados gerados por um dado classificador.

Tabela 4 - Exemplo de matriz de confusão para um dado classificador

		VERDADE	
		GATO	CACHORRO
ESTIMADO	GATO	TP	FP
	CACHORRO	FN	TN

Fonte: Adaptado de FUKUNAGA (2013).

Considerando a matriz de confusão apresentada na Tabela 4, a variável *true positive* (TP) corresponde ao número de gatos que foram classificados como gatos. A variável *true negative* (TN) corresponde aos cachorros que foram classificados como cachorros. A variável *false positive* (FP) indica o número de cachorros que foram classificados como gatos. Por último, a variável *false negative* (FN) representa o número de gatos que foram classificados como cachorros. Pode-se observar que as variáveis TP e TN expressam acertos do classificador, enquanto as variáveis FP e FN expressam erros de classificação.

A partir dos dados da matriz de confusão é possível que se obtenham algumas métricas de desempenho dos classificadores. Os autores Kumchoo e Chirachrit (2018) utilizam como métrica as medidas de acurácia, especificidade e sensibilidade. A acurácia correspondente à taxa de amostras positivas e negativas classificadas corretamente e pode ser calculada conforme a Equação 8.

$$Acurácia = \left( \frac{TP+TN}{TP+TN+FP+FN} \right) \times 100 \quad (8)$$

A especificidade é a parcela de amostras identificadas como negativas dentro o total de amostras negativas. A Equação 9 representa o cálculo da especificidade.

$$Especificidade = \left( \frac{TN}{TN+FP} \right) \times 100 \quad (9)$$

A sensibilidade mede a parcela de amostras classificadas como positivas dentre o total de amostras positivas. A Equação 10 representa a sensibilidade.

$$\textit{Sensibilidade} = \left( \frac{TP}{TP+FN} \right) \times 100 \quad (10)$$

### 3 ESTADO DA ARTE

Diferentes técnicas e metodologias podem ser utilizadas para a resolução de um dado problema. Neste capítulo é apresentada uma breve revisão dos últimos trabalhos relacionados a inspeção visual automática de frascos. Esta revisão se mostra importante uma vez que tende a apresentar as mais recentes técnicas adotadas no campo de visão artificial, auxiliando na tomada de decisão quanto a escolha das ferramentas para solucionar o problema proposto neste trabalho.

Em Sharma *et al.* (2015), encontra-se uma proposta para inspeção de base e topo de frascos PET, bem como a inspeção da presença de tampa no frasco e nível correto de produto. Tal proposta de inspeção de tampa resume-se em realizar a detecção de bordas em imagens geradas por uma câmera na linha de produção e, a partir da comparação das bordas de interesse determinar a presença da tampa no frasco.

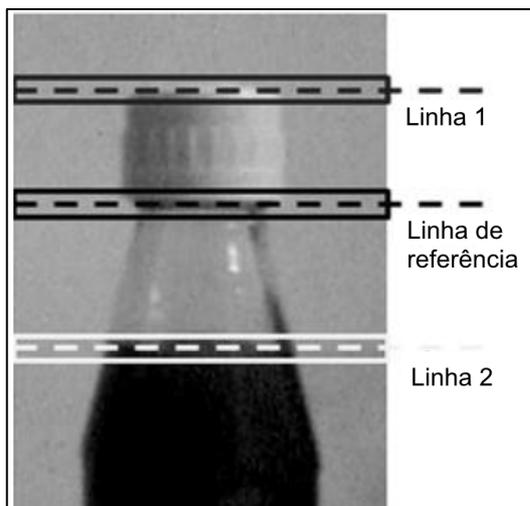
Nas etapas de inspeção da presença de tampa no frasco é utilizada uma câmera posicionada lateralmente ao frasco, sendo que esta câmera é disparada por um sensor de presença. Após a captura, a imagem originalmente colorida é transformada para escala de cinza a fim de que seja reduzida a complexidade computacional.

Como o objetivo da técnica é encontrar as linhas horizontais formadas pela base e pelo topo da tampa, o algoritmo de detecção de bordas de Canny é utilizado. Com as bordas detectadas, aplica-se a técnica de abertura morfológica para reduzir o ruído, deixando as bordas e linhas mais notáveis.

A Figura 12 ilustra o resultado obtido em uma das imagens após o processamento completo do algoritmo. Observa-se que nesta imagem estão destacadas as linhas de referência para a obtenção do resultado da inspeção.

A detecção da presença de tampa no frasco e do correto nível de líquido são feitos através da comparação de distância entre a Linha de referência e as Linhas 1 e 2, ilustradas na Figura 12. Se a distância entre a Linha de referência e a Linha 1 não for igual a altura da tampa ou se nenhuma linha horizontal for detectada na região de tolerância, o algoritmo constatará que a tampa não está posicionada corretamente. O mesmo ocorre para a inspeção de nível de produto.

Figura 12 - Linhas horizontais para inspeção de tampa e nível de líquido.



Fonte: Adaptado de Sharma *et al.* (2015)

Como forma de ampliar a avaliação experimental, os autores submeteram várias garrafas ao teste e variaram a iluminação em diversas condições. Em função disso, constatou-se que se a iluminação for imprópria, as bordas não são detectadas de forma adequada, podendo gerar uma detecção falsa.

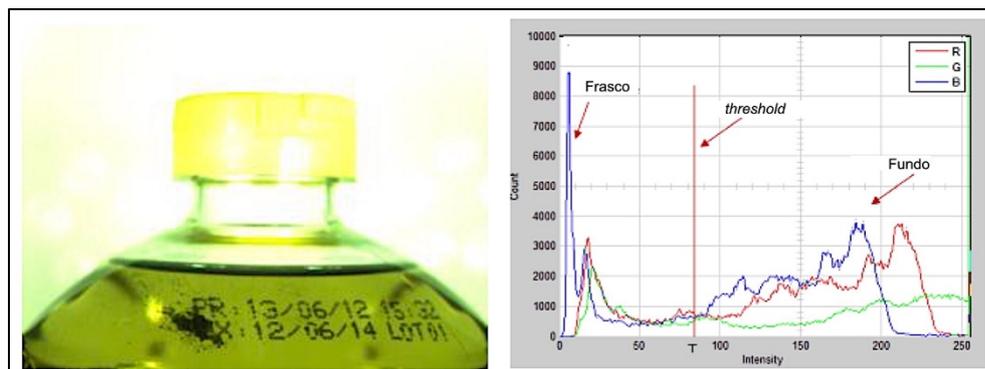
Para inspeção de nível e presença de tampa com condições de iluminação adequadas os autores obtiveram uma acurácia de 100%. No caso de iluminação inadequada, os resultados foram de 88% de acurácia.

No trabalho apresentado por Abdelhedi, Taouil e Hadjkacem (2012), os autores propõem uma solução para inspecionar a presença de tampa e o nível de produto em frascos de óleo de oliva em linha de produção. Neste trabalho, a abordagem adotada pelos autores consiste no uso de cálculos de áreas em imagens, fazendo o uso de uma câmera CCD para aquisição de imagens.

Na primeira etapa do algoritmo é utilizado um filtro gaussiano para a redução de ruído. Em seguida, calcula-se o histograma da imagem em RGB. Neste histograma é analisado qual dos canais de cor apresenta a distribuição mais adequada e que possibilite a diferenciação da tampa em relação ao restante da imagem.

A Figura 13 ilustra a imagem original e o seu histograma, onde nota-se que o canal de cores azul apresenta uma distribuição bimodal, ou seja, possui dois picos em níveis distintos, algo desejável para a aplicação da técnica adotada.

Figura 13 - Histograma dos autores Abdelhedi, Taouil e Hadjkacem (2012)



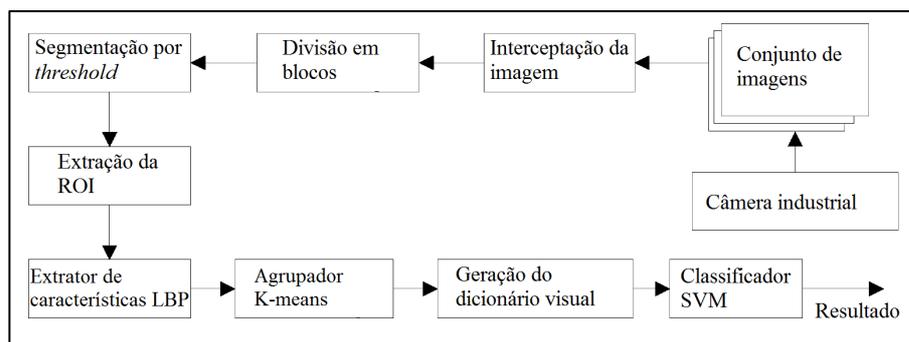
Fonte: Adaptado de Abdelhedi, Taouil e Hadjkacem (2012)

Como pode ser observado na Figura 13, o canal azul obteve os dois picos de intensidade mais distintos, e, por isso, este foi o canal de cores escolhido para a aplicação do limiarização. Para a etapa de limiarização da imagem, os autores escolheram o método de Otsu. Por último, com a imagem já binarizada, os autores realizam o cálculo da área restante na região de interesse e a comparam com um valor de área previamente definido. Caso os valores de área sejam diferentes o frasco é classificado como não conforme.

O algoritmo de processamento e inspeção foi desenvolvido na plataforma Matlab 7.12b e executado em um computador pessoal. Os autores não apresentaram em seu trabalho os resultados quanto a acurácia da técnica. Quanto ao tempo de processamento, foram coletados resultados para duas velocidades de esteira diferentes, sendo que na velocidade de 1 m/s da esteira obteve-se o tempo de processamento em torno de 0,55 ms; e com a velocidade da esteira em 0,5 m/s, o tempo de processamento foi de aproximadamente 0,6 ms.

No trabalho apresentado pelos autores Liu *et al.* (2017), uma proposta de sistema foi sugerida para a inspeção visual automática de tampas de frascos da indústria farmacêutica. A abordagem proposta é baseada em algoritmos de extração de características e aprendizado de máquina. O diagrama apresentado na Figura 14 ilustra as etapas e as técnicas do algoritmo para o sistema de inspeção.

Figura 14 - Diagrama do método proposto pelos autores Liu *et al.* (2017)



Fonte: Adaptado de Liu *et al.* (2017)

A primeira etapa consiste na aquisição da imagem através de uma câmera industrial. Na segunda etapa a imagem é dividida em blocos para, então, passar pelo processo de *threshold segmentation* (TSEG) na terceira etapa. Em seguida, os autores realizam o processo de extração da *Region Of Interest* (ROI) com base nos índices dos blocos de imagem segmentados na terceira etapa. Na quarta etapa ocorre a extração das características através do método *Local Binary Pattern* (LBP). O dicionário visual é obtido através do algoritmo de agrupamento de dados K-Means na quinta etapa. Por último, um classificador *Support Vector Machine* (SVM) é empregado para obter-se o resultado final da inspeção.

Como forma de avaliação dos resultados foram utilizadas as métricas de acurácia e tempo de processamento, sendo realizados testes para diferentes métodos de segmentação e extração de características. Utilizando o método de *threshold* para segmentação das imagens foram obtidos resultados de 90% de acurácia quando utilizadas 95 ou mais amostras no treinamento. Na comparação dos métodos de extração de características, o LBP obteve melhores resultados quando comparado aos métodos SIFT e SURF. Os resultados mostraram uma taxa de 95% de acertos quando utilizadas 200 amostras de treinamento. Com a metodologia proposta pelos autores, alcançou-se uma média de 8 avaliações de frascos por segundo. Contudo, não foi mencionado no artigo, o sistema computacional utilizado.

Um método baseado na detecção por localização e correspondência é proposto pelos autores Xie *et al.* (2017) para resolver o problema de inspeção da qualidade de encapsulamento de garrafas de Polietileno Tereftalato (PET). Para localizar a garrafa na imagem é utilizado o método *Normalized Cross-Correlation*

(NCC), que realiza a correspondência entre a imagem capturada e o modelo de referência.

Assim que a posição da garrafa é encontrada na imagem, a ROI é selecionada com base nessa posição. Os autores utilizaram o método *Smallest Univalued Segment Assimilating Nucleus* (SUSAN) para extração de borda. A detecção da tampa no frasco foi realizada diretamente pelo uso do método NCC.

A verificação do correto posicionamento da tampa é realizada por meio da medida entre o suporte do anel de lacre e a base do lacre na tampa. Já para a avaliação de tampa inclinada foram utilizadas linhas paralelas traçadas dentro da ROI e comparou-se a variação da distância entre a borda de base da tampa e as linhas previamente desenhadas.

Para o desenvolvimento, os autores Xie *et al.* (2017) utilizaram um computador industrial com o sistema operacional *Windows 7*. A biblioteca utilizada foi a *Halcon*, junto com o ambiente de desenvolvimento *Visual Studio 2010*. Os testes foram realizados através da inspeção de 2000 garrafas, divididas em 4 condições, sendo elas: tampa corretamente posicionada, tampa alta, tampa inclinada e falta de tampa. Os resultados obtidos mostram um valor percentual de precisão de 99,65%. O tempo total de processamento foi de 67 ms por *frame*.

O trabalho apresentado pelos autores Xu, Ma e Chen (2017) foca na inspeção da superfície da tampa e na verificação das condições do anel de lacre. O sistema proposto é constituído por um conjunto de 3 câmeras CCD dispostas a 120° entre si. Desta forma, toda a superfície da tampa pode ser analisada sem que haja a necessidade de rotação do frasco.

No processo de inspeção, após a captura da imagem, a segmentação é realizada através da aplicação de *threshold*. Para a avaliação da superfície da tampa foi proposto um algoritmo baseado em modelagem da imagem de fundo, comparação de imagem e *threshold* dinâmico. A detecção de quebra no anel de lacre é feita através da transformação da imagem do sistema de cores RGB para o sistema HSV, pois, o traço do *gap* que aparece entre a tampa e o anel de lacre fica mais nítido na componente de saturação da imagem.

Para o levantamento dos resultados da inspeção, os autores Xu, Ma e Chen (2017) selecionaram 1000 tampas em condições normais, 100 tampas com algum defeito na superfície, 100 tampas com o anel de lacre parcialmente quebrado e 100 tampas com o anel de lacre completamente quebrado. A aplicação foi desenvolvida

sobre a plataforma *Visual C# 2013* com a biblioteca *OpenCV*. O *hardware* de processamento é composto por um processador *Intel Core i5* com 4 GB de memória *RAM*. O sistema proposto apresentou uma acurácia média de 97,2% entre os 3 tipos de defeitos testados. A menor acurácia foi obtida na detecção de defeitos na superfície da tampa e foi de 92%.

No trabalho apresentado por Kumchoo e Chiracharit (2018), os autores propõem um método para a avaliação da rosca e do lacre de segurança em garrafas de vidro farmacêuticas com tampas “auto-rosqueantes”.

Após a captura da imagem, a segunda etapa consiste na divisão da imagem em duas ROI, sendo elas a área da rosca da tampa e a área do lacre de segurança. Na terceira etapa as duas ROI encontradas anteriormente são transformadas do espaço RGB para o espaço *Hue Saturation Value* (HSV). Na quarta etapa é realizada a extração das características da imagem por meio da aplicação do método de *threshold* e da técnica de abertura morfológica. As técnicas de abertura e *threshold* são aplicadas para a redução do ruído e supressão de objetos menores do que 40 pixels. Na última etapa é realizada a contagem dos pixel distribuídos em linhas e colunas.

A partir desta contagem de pixels, que não foram suprimidos, são gerados gráficos de barra. Estes gráficos expressam a distribuição do número de pixels divididos em segmentos ao longo das duas regiões de interesse. O sistema realiza a decisão de conformidade ou de defeito na tampa através da comparação da distribuição encontrada em cada tampa com uma distribuição predeterminada anteriormente,.

Os resultados obtidos pelos autores foram adquiridos através do teste de 86 amostras, sendo 34 amostras com tampa mal rosqueada, 22 amostras com defeito no lacre de segurança e 30 amostras em perfeito estado. Para comparação com o método proposto foram selecionados 3 especialistas em inspeção que avaliaram as mesmas amostras. As métricas utilizadas foram os valores de acurácia, especificidade e sensibilidade. A partir dos dados apresentados pela Tabela 5 é possível notar que o sistema de inspeção proposto pelos autores apresentou melhores resultados na avaliação da condição do lacre de segurança. Porém, nota-se que no teste de tampa mal rosqueada os resultados foram iguais em termos de acurácia.

Tabela 5 - Resultados obtidos pelos autores Kumchoo e Chiracharit (2018).

<b>Tipo de teste</b>	<b>Avaliador</b>	<b>Acurácia (%)</b>	<b>Especificidade (%)</b>	<b>Sensibilidade (%)</b>
Tampa solta	Sistema de processamento de imagem	85,94	83,87	87,88
	Melhor resultado entre os especialistas	85,94	100,00	80,49
Lacre de segurança	Sistema de processamento de imagem	92,31	100,00	84,62
	Melhor resultado entre os especialistas	76,92	90,91	66,67

Fonte: Adaptado de Kumchoo e Chiracharit (2018).

## 4 DESENVOLVIMENTO DO TRABALHO

Este trabalho tem como objetivo o desenvolvimento e a avaliação de um sistema para inspeção de frascos em linha de produção utilizando dispositivos de baixo custo. O critério de inspeção está relacionado a qualidade do fechamento do frasco, onde a condição de presença da tampa no frasco será verificada. Para atender este objetivo, os aspectos da implementação do sistema, o desenvolvimento dos algoritmos e a metodologia de avaliação são abordados nas seções seguintes.

### 4.1 IMPLEMENTAÇÃO DO SISTEMA

A proposta de implementação do sistema tem como fundamento a análise das principais características do cenário e da aplicação. Nesta análise foram abordadas a caracterização das condições da tampa e do frasco, a avaliação do funcionamento da linha de produção onde o sistema está inserido, a forma de controle do ambiente e a escolha do *hardware* de inspeção visando baixo custo de implementação.

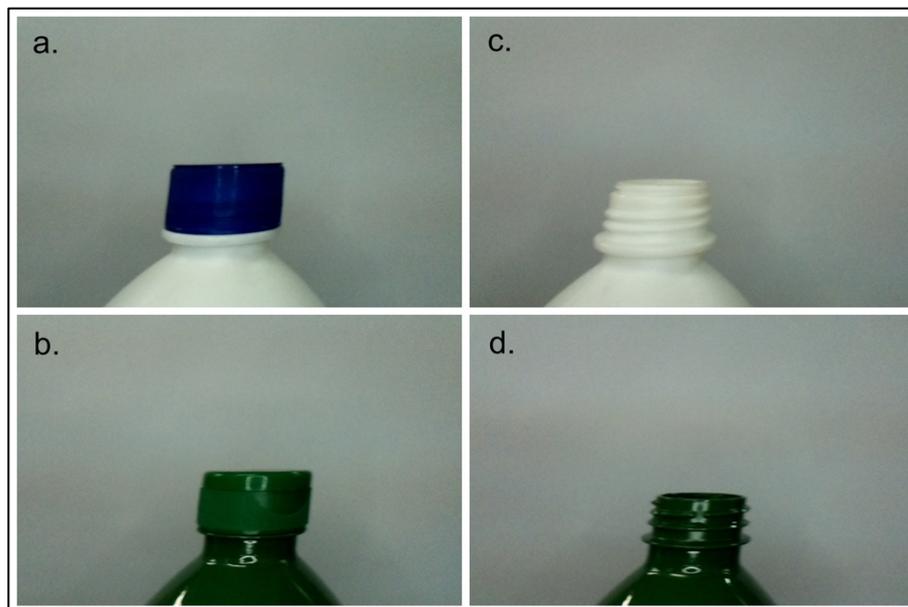
Para a avaliação experimental deste trabalho foram testados dois modelos diferentes de frascos. O primeiro modelo de frasco é fabricado em PEAD (Polietileno de Alta Densidade) de cor branca e fechado com tampa de material plástico na cor azul. Este modelo de frasco é comumente utilizado em amaciantes de roupa. O segundo modelo de frasco testado é fabricado em material PET de cor verde e fechado com tampa de material plástico também na cor verde. Este frasco PET é utilizado para envase de produtos clorados, principalmente a água sanitária.

Uma breve avaliação da Figura 15 indica as 2 condições que podem ser encontradas para cada tipo de frasco a ser inspecionado. As Figura 15a e 15b ilustram as condições para os frascos branco e verde com tampa. Nas Figuras 15c e 15d estão ilustradas as condições de frascos sem tampa no modelo branco e no modelo verde. Nota-se que as imagens presentes nesta figura foram capturadas pela câmera escolhida para o sistema e não possuem nenhum tratamento, exceto um pequeno recorte para um melhor enquadramento na figura.

Com base na avaliação das condições encontradas na Figura 15, observa-se que o sistema deverá ser capaz de classificar como “objeto em conformidade” somente as condições encontradas nas imagens 15a e 15b. Para o restante das

condições, o sistema deverá gerar uma sinalização, indicando que a inspeção encontrou uma não conformidade.

Figura 15 - Exemplo das condições de inspeção da tampa.



Fonte: O autor (2019).

O processo de envase e fechamento de frascos ocorre de forma contínua, e a velocidade deste processo pode ser medida a partir da quantidade de frascos por minuto que são entregues pela linha de produção. Esta capacidade de produção está relacionada com as velocidades das máquinas e das esteiras de transporte. Além disso, o espaçamento entre os frascos no transporte pelas esteiras também impacta na velocidade final de produção (XIE *et al.*, 2017).

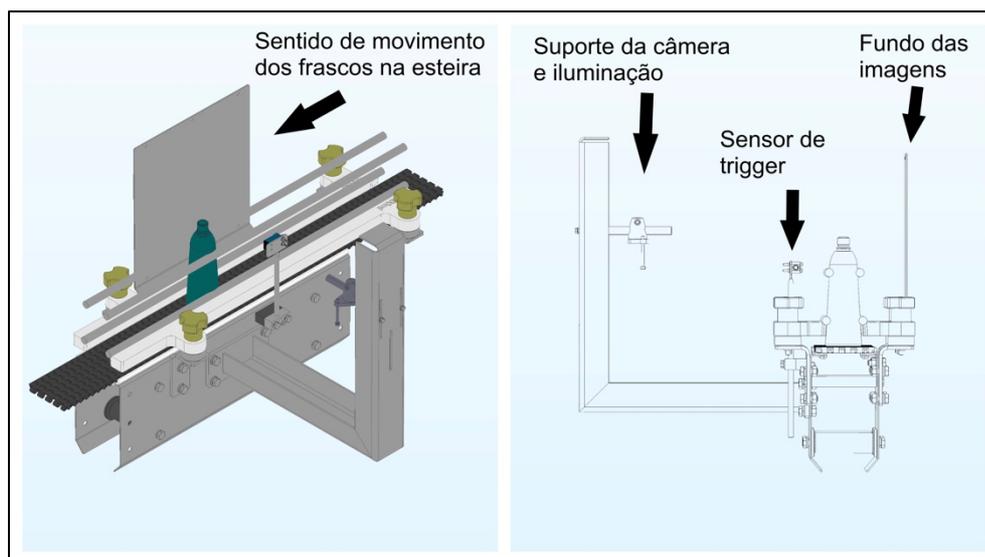
De acordo com Davies (2012), um sistema de inspeção deve ter a capacidade de operar em tempo real, ou seja, o mesmo deve ser capaz de atender a velocidade de produção da linha na qual está inserido. Tendo isso em vista, o sistema proposto neste trabalho foi avaliado com base na velocidade da linha de produção de uma envasadora de produtos saneantes da empresa Robopac Brasil (BRASIL, 2018). Sendo assim, tomou-se como base o parâmetro de velocidade de produção nominal em frascos por minuto desta máquina, sendo necessário considerar a velocidade das esteiras de transporte e o espaçamento entre frascos para um cenário típico de aplicação da máquina.

A iluminação adequada do ambiente e do objeto a ser inspecionado pode auxiliar na performance do sistema, oportunizando um processamento mais adequado e diminuindo a taxa de erros. Logo, para a iluminação do sistema foi utilizado o método de iluminação direta. Este método destaca-se por sua facilidade de montagem e capacidade de aumento do contraste.

A escolha de uma iluminação por LED também acrescenta aspectos positivos, como um custo razoável e um tempo de vida elevado se comparado com a iluminação artificial de lâmpadas incandescentes ou fluorescentes. A luminária utilizada possui formato em anel com aproximadamente 12 cm de diâmetro e produz luz de forma difusa, o que pode auxiliar na diminuição das sombras. O fundo da inspeção utilizado é formado por uma superfície uniforme na cor branca (COGNEX, 2012).

A Figura 16 esboça a disposição da câmera e dos demais dispositivos para a coleta de imagens na linha de produção. A câmera foi posicionada em um suporte de inox a aproximadamente 20 cm do centro da esteira de frascos. A iluminação foi montada no mesmo suporte que a câmera, de modo que a mesma ficou no centro do anel de LEDs. O sensor de *trigger*, utilizado para detectar a presença de um frasco e então gerar um sinal de captura para câmera, foi fixado em um suporte independente que está posicionado antes da câmera.

Figura 16 - Disposição da câmera e dispositivos para aquisição de imagens



Fonte: O autor (2019).

A escolha da plataforma para o desenvolvimento do sistema e do *hardware* para aquisição de imagem tiveram como critérios o baixo custo, a capacidade de processamento, as interfaces disponíveis e as dimensões físicas. Para o desenvolvimento do sistema foi utilizada a placa Raspberry Pi 3 Model B. A capacidade de processamento e memória da placa, aliadas as interfaces de comunicação e aquisição de imagens são características que permitem o seu uso para a aplicação proposta neste trabalho (RASPBerryPI, 2018). Além disso, a possibilidade de utilização de um sistema operacional capaz de processar ferramentas dedicadas para visão computacional, como a biblioteca OpenCV, permite agilidade no desenvolvimento da aplicação proposta (KAEHLER; BRADSKI, 2016).

A câmera Raspberry Pi V2 foi a opção escolhida para a aquisição de imagens, uma vez que possui conectividade com a placa Raspberry Pi 3 Model B via porta CSI (RASPBerryPI, 2018). A possibilidade de configuração dos parâmetros da câmera através da biblioteca Picamera também se mostra como um ponto positivo para a aplicação. Esta biblioteca permite a configuração de parâmetros como resolução, tempo de exposição, taxa de aquisição (fps), formato de captura, entre outros (JONES, 2018).

## 4.2 AQUISIÇÃO E GERAÇÃO DO BANCO DE IMAGENS

Como o problema de inspeção deste trabalho possui características da linha de produção em questão, uma base de dados composta por imagens foi criada para comparar os algoritmos e avaliar as soluções. Inicialmente, os frascos foram avaliados manualmente e as imagens foram classificadas de acordo com o modelo do frasco e a condição de presença ou falta da tampa.

A base de dados é composta por imagens de frascos verdes (PET) e brancos (PEAD) obtidas em linha de produção na condição de velocidade nominal da máquina. Foram geradas ao todo 313 imagens de frascos brancos, sendo 152 de frascos com tampa e 161 de frascos sem tampa. Para o modelo de frasco verde foram capturadas 125 imagens, sendo 61 imagens com tampa e 64 imagens sem tampa. A Tabela 6 apresenta um resumo da quantidade de imagens da base de dados.

Tabela 6 - Resumo do banco de imagens de frascos

<b>Modelo do frasco</b>	<b>Quantidade</b>
Total de frascos brancos	313
Frascos brancos com tampa	152
Frascos brancos sem tampa	161
Total de frascos verdes	125
Frascos verde com tampa	61
Frascos verdes sem tampa	64

Fonte: O autor (2019).

Durante a etapa de criação da base de dados com as imagens de frascos foram observados parâmetros referentes a linha de produção e a configurações do sistema de aquisição. Dentre estes parâmetros pode-se destacar a velocidade da esteira de frascos, a velocidade nominal de produção da linha e o tempo disponível por frasco. A Tabela 7 apresenta um resumo dos dados coletados na etapa de aquisição de imagens.

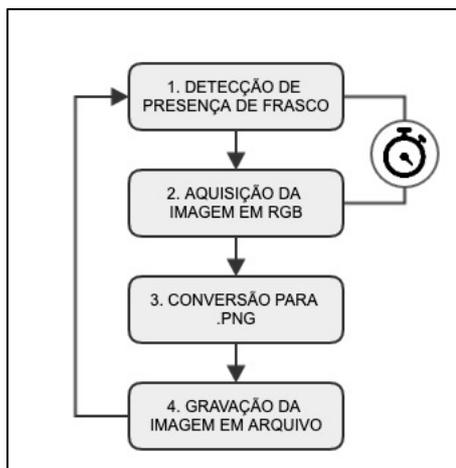
Tabela 7 – Parâmetros medidos da etapa de aquisição de imagens.

<b>Parâmetro</b>	<b>Valor / Formato</b>	<b>Unidade</b>
Velocidade da esteira de frascos	0,757	metros/s
Número de frascos/metro na esteira	3	unidades
Produção nominal aproximada	136	frascos/min
Tempo disponível por frasco	441,1	ms
Tempo de exposição	1500	us
Resolução	1280x720	pixels
<i>Framerate</i>	30	fps
Formato de captura	RGB	
Formato de gravação	PNG	

Fonte: O autor (2019).

A Figura 17 apresenta o fluxograma de aquisição das imagens em linha de produção. Observa-se pelo fluxograma que os contadores de tempo de aquisição são iniciados no momento de leitura do sensor de *trigger* e são finalizados após a formação do objeto de imagem em RGB. Com isso, não foram contabilizados os tempos das etapas de conversão e gravação das imagens, visto que em uma aplicação normal estas etapas não existiriam.

Figura 17 - Fluxograma de aquisição de imagem.



Fonte: O autor (2019).

Devido as limitações impostas pelo sistema de iluminação não foi possível utilizar valores de tempo de exposição abaixo de 1500 us. Com isso, foram encontradas algumas distorções nas imagens, como as causadas pelo efeito de *motion blur* e pelo efeito *rolling shutter* (CHIAO; LEE; LAI, 2013). A Figura 18 ilustra uma das imagens de um frasco branco em movimento capturada em linha de produção, onde duas linhas paralelas foram traçadas para ajudar a identificar a distorção causada pelo efeito de *rolling shutter*.

Figura 18 - Distorções encontradas nas imagens capturadas.

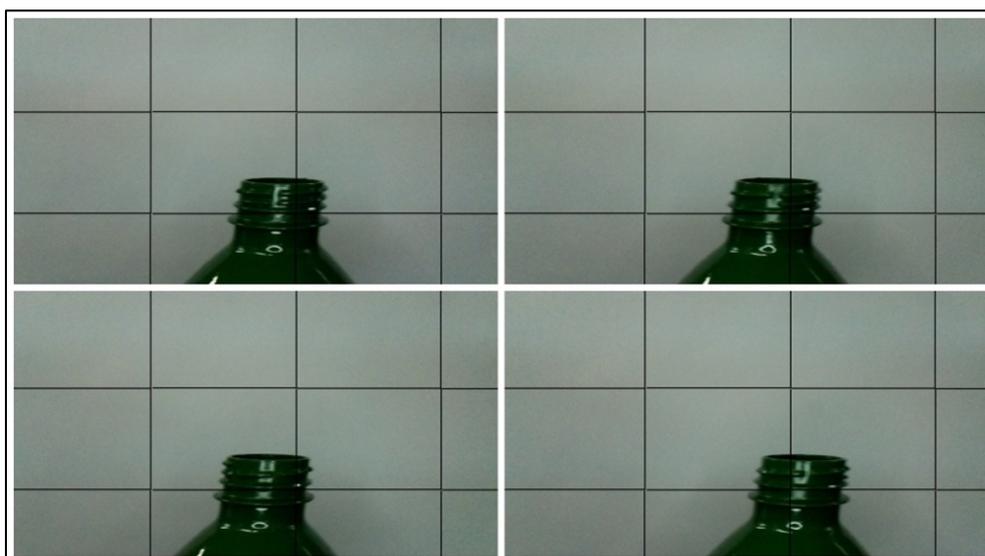


Fonte: O autor (2019).

Outros dois aspectos podem ser observados na base de dados e estão relacionados ao posicionamento dos frascos nas imagens. O primeiro aspecto está relacionado às variações de posição lateral com a qual os frascos aparecem nas

imagens. Alguns atrasos no sistema de captura, aliados a velocidade elevada da esteira de transporte e a proximidade da câmera em relação ao frasco, produzem capturas com pequenas diferenças na localização do frasco. A Figura 19 ilustra 4 capturas de frascos em sequência apresentando diferentes localizações, que são evidenciadas quando traça-se uma grade sobre cada captura.

Figura 19 - Sequência de captura de frascos com diferentes posicionamentos.



Fonte: O autor (2019).

O segundo aspecto do posicionamento dos frascos nas imagens é produzido pela folga existente entre os frascos e as guias laterais da esteira. Esta folga, que possibilita que os frascos se movam livremente pela esteira, acaba permitindo que aconteçam pequenas variações na relação de proximidade entre os frascos e a câmera. Estas variações de proximidade acabam gerando pequenas diferenças na área ocupada pelo frasco na imagem.

### 4.3 ALGORITMOS DE INSPEÇÃO

Como parte da proposta deste trabalho, buscou-se realizar um comparativo entre diferentes técnicas de processamento de imagens que viessem a resolver o problema referente à inspeção de frascos em uma linha de envase utilizando dispositivos de baixo custo. A escolha das técnicas avaliadas se baseou na análise do estado da arte. Esta seleção levou em consideração as abordagens relacionadas

ao objetivo deste trabalho, bem como a possibilidade de implementação destas técnicas em um sistema embarcado e a possibilidade de comparação de resultados.

O primeiro algoritmo implementado é o apresentado no trabalho dos autores Sharma *et al.* (2015). No artigo, os autores não definiram um nome específico à técnica adotada, logo, neste trabalho ela foi nomeada como “Método das linhas”.

A segunda abordagem avaliada é a encontrada no trabalho dos autores Abdelhedi, Taouil e Hadjkacem (2012). Esta técnica foi denominada neste trabalho como “Método das áreas”, devido às características do algoritmo.

A terceira técnica implementada foi baseada no trabalho dos autores Liu *et al.* (2017), sendo selecionada por utilizar um algoritmo robusto para segmentação e pelo fato de manipular imagens coloridas. Esta abordagem foi denominada como “Método K-means”.

O último método implementado é baseado no algoritmo de detecção e descrição de característica ORB desenvolvido pelos autores Rublee *et al.* (2011) e sua aplicação neste trabalho foi denominada como “Método ORB”.

Com intuito de trazer clareza quanto a metodologia e lógica adotada para cada técnica, bem como a sua aplicação neste trabalho, uma descrição detalhada das etapas de implementação de cada método encontra-se descrita nas seções a seguir.

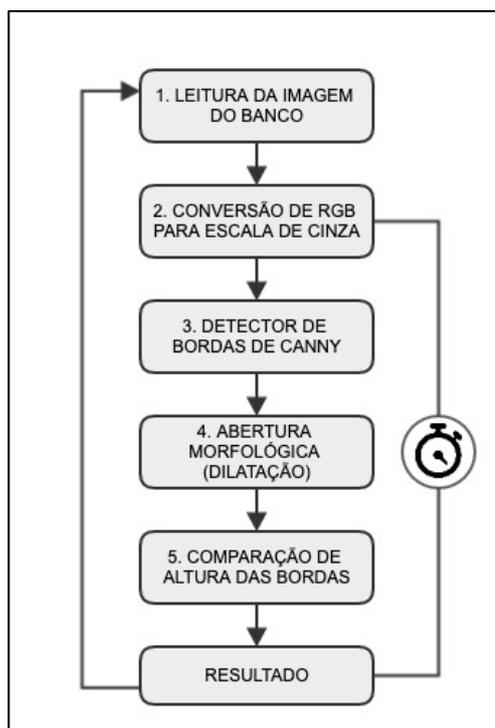
#### **4.3.1 Método das linhas**

O método das linhas tem como objetivo a localização das linhas horizontais encontradas na base e no topo das tampa nos frascos. O algoritmo implementado pode ser dividido em seis etapas. O fluxograma apresentado na Figura 20 ilustra as etapas implementadas no método das linhas.

A primeira etapa consiste no “carregamento de uma imagem”, e como isso não faz parte da técnica em si, este tempo não é considerado na análise de desempenho do algoritmo. Com isso, a contagem do tempo de processamento deste método é iniciado na segunda etapa. Esta consideração também é realizada para os demais métodos implementados neste trabalho.

Na segunda etapa, a imagem do frasco em questão é convertida para escala de cinza. Com essa conversão, a quantidade de dados é reduzida, resultando em um menor consumo computacional.

Figura 20 - Fluxograma do método das linhas.



Fonte: O autor (2019).

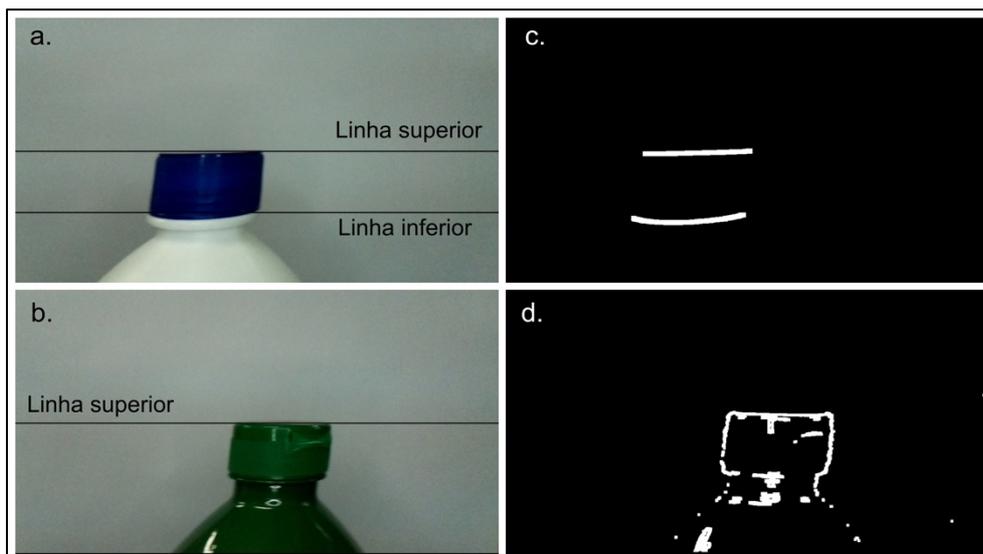
Na terceira etapa do método das linhas foi utilizado o algoritmo de detecção de bordas de Canny. A saída desta etapa são as imagens binarizadas, contendo somente as bordas que ficam dentro de valores previamente ajustados. Em seguida, foi utilizada a técnica de abertura morfológica, também chamada de dilatação, para auxiliar a tornar as bordas mais notáveis. As Figuras 21c e 21d ilustram as imagens de saída obtidas na quarta etapa para o frasco branco com tampa azul e para o frasco verde com tampa verde, respectivamente.

Pode ser observado através das Figuras 21c e 21d que o frasco branco obteve bordas mais evidentes e mensuráveis quando comparado com a saída gerada para o frasco verde. Esta diferença se deve ao fato de que a variação de intensidade presente na tampa azul é mais intensa e permite um ajuste preciso do algoritmo de detecção de bordas de Canny. Nas imagens do frasco verde não ocorre esta mesma variação de intensidade entre a tampa e o frasco, o que faz com que não seja possível evidenciar somente as bordas referentes a tampa, mesmo após a etapa de dilatação.

A quinta etapa do método das linhas consiste na verificação da distância entre as linhas “superior” e “inferior” ilustradas nas Figuras 21a e 21b. Nos casos onde não foram encontradas estas duas linhas de referência ou foram encontradas mais do que

duas linhas, atribui-se o valor zero a esta distância. Observa-se através das Figuras 21b e 21d que para as imagens do frasco verde não foi possível realizar a definição da linha inferior, pois, não foi possível detectar uma única borda que defina claramente esta linha de referência.

Figura 21 - Imagens geradas na etapa 4 do método das linhas.



Fonte: O autor (2019).

Por último, na sexta etapa é realizada a comparação entre a distância encontrada na imagem que está sendo inspecionada com uma medida de distância padrão, previamente definida. Se o valor de distância medido estiver dentro de um intervalo considerado adequado da medida padrão o frasco será classificado como conforme. Do contrário, o frasco será classificado como não conforme.

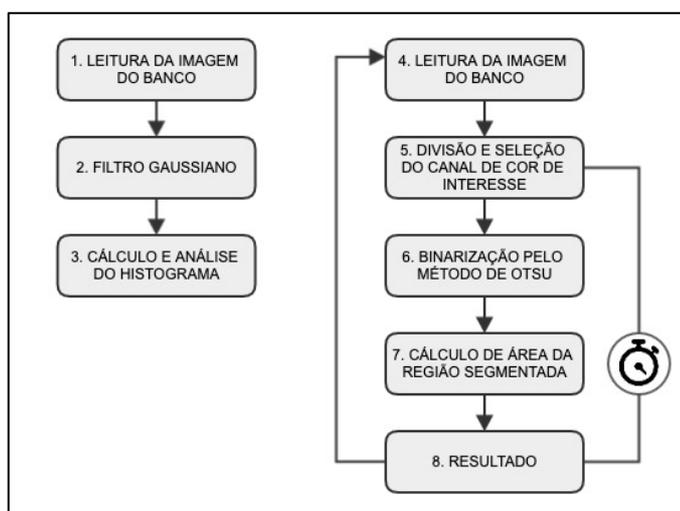
Observa-se que tanto a medida de distância padrão para comparação, quanto os parâmetros do detector de bordas de Canny e o ajuste do *kernel* utilizado para a dilatação precisam ser previamente ajustados. No caso deste trabalho estes parâmetros foram medidos e ajustados com base na primeira imagem de frasco com tampa do banco de imagens. Como estes ajustes ocorrem somente no momento de parametrização do sistema os mesmos não tiveram o seu tempo contabilizado. Em uma aplicação final estes parâmetros precisariam ser ajustados através do uso de uma interface gráfica.

### 4.3.2 Método das áreas

O método das áreas consiste na segmentação e no cálculo da área de uma determinada região de interesse, que para a proposta deste trabalho seria a região onde é encontrada a tampa do frasco. Este método é dividido em duas partes e possui 8 etapas.

A primeira parte da implementação tem como objetivo a definição do canal de cor que será utilizado para a segmentação, ocorrendo somente no momento de *setup* do sistema. A segunda parte do método é responsável pela avaliação de conformidade ou defeito dos frascos. Esta parte é reproduzida de maneira cíclica, pois representa a inspeção *online* da linha de produção. A Figura 22 apresenta o fluxograma da implementação do método das áreas.

Figura 22 - Fluxograma do método das áreas.



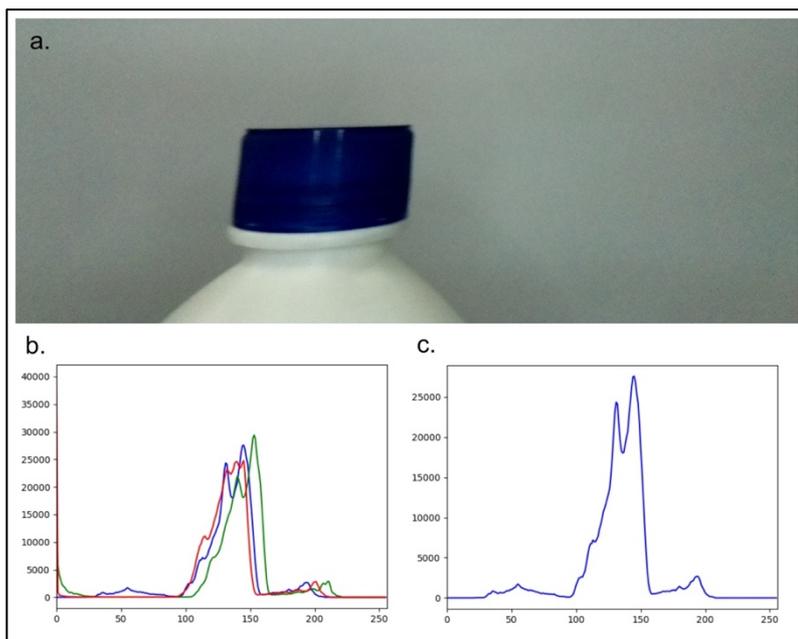
Fonte: O autor (2019).

Na segunda etapa, aplica-se na imagem carregada um filtro gaussiano, que tem como objetivo reduzir os ruídos da imagem original, melhorando a sua qualidade para as próximas etapas do processamento.

A terceira etapa da técnica consiste na realização do cálculo e na análise do histograma das imagens carregadas e filtradas. Esta etapa ocorre individualmente tanto para o modelo de frasco branco quanto para o modelo de frasco verde. A análise do histograma tem como objetivo definir em qual dos canais de cor existe a ocorrência de dois picos de intensidade distintos.

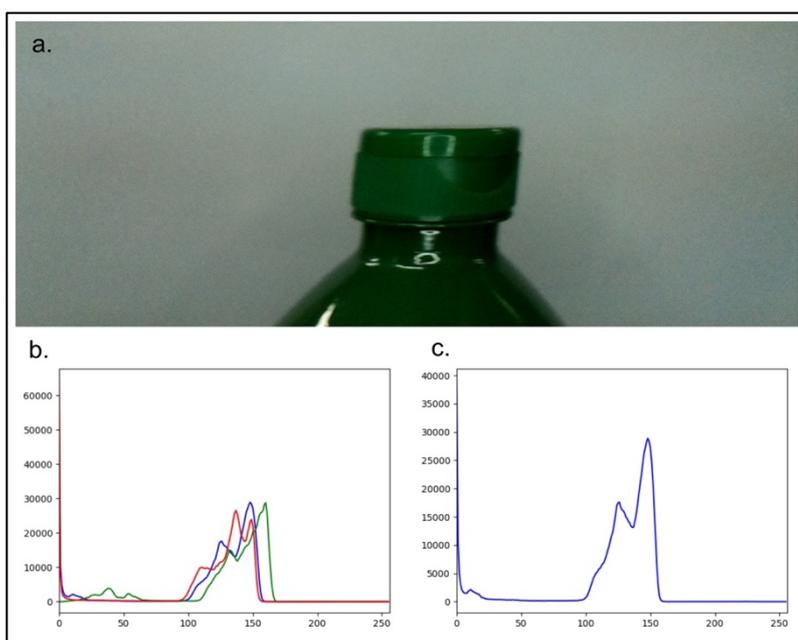
As Figuras 23a e 24a apresentam as imagens dos frascos branco e verde, respectivamente, utilizadas para os cálculos dos histogramas. As Figuras 23b e 24b apresentam os histogramas completos gerados pelas imagens originais. As Figuras 23c e 24c apresentam os histogramas do canal e cor azul de maneira isolada.

Figura 23 - Cálculo do histograma para o frasco branco com tampa.



Fonte: O autor (2019).

Figura 24 - Cálculo do histograma para o frasco verde com tampa.



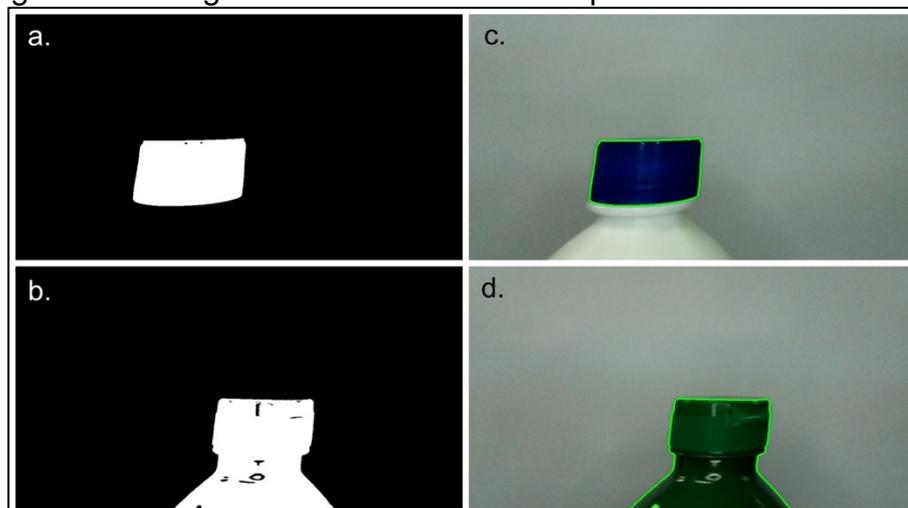
Fonte: O autor (2019).

Por meio da análise dos histogramas das Figura 23c e 24c, pode-se observar que o canal de cor azul apresentou os dois picos mais notáveis. Imagens que apresentam esta característica de histograma também são denominadas como imagens bimodais. O canal de cor azul também apresentou a melhor divisão de picos no histograma da imagem do frasco verde, e, por isso, este foi o canal de cores selecionado para a próximas etapas do algoritmo para os dois modelos de frascos.

Na quinta etapa do método das áreas ocorre a divisão da imagem carregada em 3 camadas, cada uma composta por um dos canais de cor R, G e B. Em seguida é selecionada somente a camada composta pelo componente azul da imagem original, visto que este foi o canal de cor escolhido para as próximas operações. As demais camadas foram desconsideradas para o processamento desta etapa em diante.

Para sexta etapa utilizou-se o método de Otsu para realizar a limiarização da camada selecionada anteriormente. Este método de limiarização é normalmente utilizado na segmentação de imagens bimodais, como as encontradas nesta aplicação (ABDELHEDI; TAOUIL; HADJKACEM, 2012) (KAEHLER; BRADSKI, 2016). As Figuras 25a e 25b ilustram as imagens de saída da sexta etapa para o modelo de frasco branco e verde, respectivamente. Observa-se por estas imagens que a limiarização do frasco branco conseguiu manter somente a região da tampa, enquanto a limiarização para o frasco verde não foi capaz de distinguir a região da tampa da região do frasco.

Figura 25 - Imagens de saída da sexta etapa do método das áreas.



Fonte: O autor (2019).

A sétima etapa consiste no cálculo da área segmentada na imagem de saída da etapa 6. Inicialmente utilizou-se a ferramenta de contorno de OpenCV, o que permitiu a delimitação da área desejada na imagem segmentada. As Figuras 25c e 25d ilustram com um contorno a área selecionada para o frasco branco e para o frasco verde, respectivamente. Em seguida, o cálculo da área desta região demarcada é realizado através da soma dos pixels dentro do contorno.

Para a avaliação do frasco branco considerou-se como área de interesse somente a região delimitada pela tampa. No caso de avaliação do frasco verde foi necessário assumir como área de interesse a soma da área do frasco com a área da tampa. Realizou-se esta consideração pois observa-se que a área ocupada por um frasco com tampa tende a ser maior do que a área ocupada por um frasco sem tampa, o que permite certa distinção entre as duas condições.

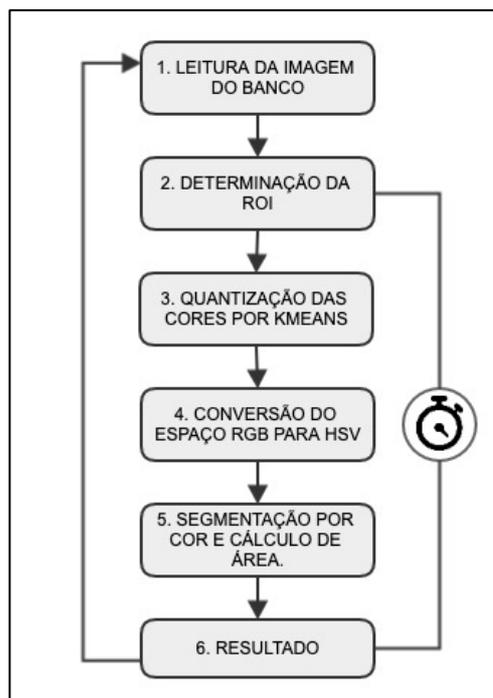
Por fim, a oitava etapa consiste na comparação entre os valores de área medidos na imagem inspecionada com um valor de área previamente definido. Caso o valor da área medida na imagem fique fora de um intervalo aceitável, o frasco será classificado como não conforme. Neste trabalho foi considerada como padrão a área calculada na primeira imagem do frasco branco com tampa presente no banco. O mesmo foi considerado para a área padrão do frasco verde.

### **4.3.3 Método K-means**

O método K-means proposto é baseado na segmentação da imagem por cores. Esta segmentação ocorre com base na imagem em RGB e visa a redução da quantidade de cores através de agrupamento. Com esta redução, a segmentação de uma única cor ou de uma faixa de cores de interesse ocorre de forma mais precisa. A implementação deste método pode ser dividida em 6 etapas, apresentadas no fluxograma da Figura 26.

Após o carregamento da imagem ocorre a determinação da ROI. Para tal foi realizada a conversão da imagem do espaço de cores RGB para escala de cinza. Em seguida, utilizou-se o algoritmo de detecção de bordas de Canny para geração de uma imagem binária na qual estarão representadas as bordas da imagem. A escolha deste detector de bordas se deve a sua baixa taxa de erros, adequada localização dos pontos e precisão de resposta (SOLOMON; BRECKON, 2013).

Figura 26 - Fluxograma do método K-means.



Fonte: O autor (2019).

Através da utilização das ferramentas de contorno foi delimitada a ROI em formato retangular no entorno do frasco e tampa. Com essa localização aproximada, é possível realizar a redução da área a ser processada nas etapas seguintes, o que possibilita um menor custo computacional e, conseqüentemente, um menor tempo de processamento.

Para realizar a quantização das cores encontradas na ROI, utilizou-se o algoritmo K-means. Esta quantização é realizada através do agrupamento das cores presentes na imagem. O agrupamento acontece de acordo com a definição de 3 parâmetros, tendo como primeiro o número de grupos de cores na qual a imagem será dividida. O segundo parâmetro determina a acurácia desejada no agrupamento, também denominado como parâmetro *epsilon*. O último parâmetro determina o número máximo de interações que o algoritmo K-means poderá realizar até obter o resultado final (KAEHLER; BRADSKI, 2016).

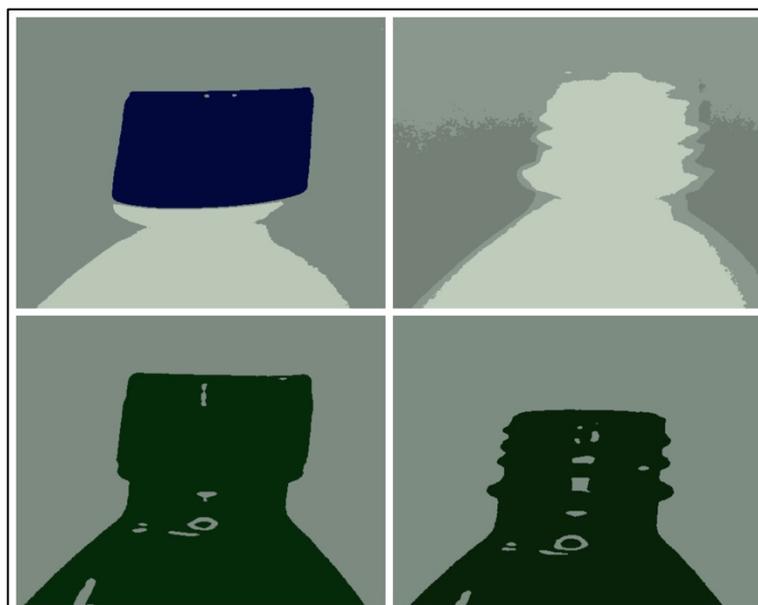
A quantização das cores do modelo de frasco branco foi realiza em 3 grupos, no qual foram agrupadas a região do fundo, do frasco e da tampa. O algoritmo K-means não apresentou resultados satisfatórios na tentativa de quantização considerando 3 grupos para o frasco verde. Esta dificuldade se deve a grande

proximidade entre as cores da tampa e do frasco. Por isso, para o modelo de frasco verde optou-se para reduzir o número de grupos para 2, no qual foram agrupadas a região da tampa e frasco como sendo um único grupo e região do fundo com sendo um segundo grupo. A Figura 27 apresenta as imagens de saída da terceira etapa do método K-means para os modelos de fracos branco e verde nas condições com e sem tampa.

Na quarta etapa do método K-means ocorre a conversão das imagens do espaço de cores RGB para o espaço de cores HSV. Esta conversão foi feita pois facilita a segmentação por cores, visto que neste espaço o parâmetro de cor e o parâmetro de intensidade são separados. Com isso o intervalo de matiz (cor), saturação e intensidade são facilmente determinados e ajustados individualmente (KAEHLER; BRADSKI, 2016).

A quinta etapa consiste na segmentação com base em um padrão de cor previamente definido na escala HSV. Para o frasco branco foi selecionada como cor de interesse a cor azul e para o frasco verde foi selecionada como cor de interesse a cor verde.

Figura 27 - Imagens de saída da etapa 3 do método K-means.



Fonte: O autor (2019).

Por último, assim como no método das áreas, o resultado da inspeção é obtido através da comparação da área segmentada com um valor de área previamente

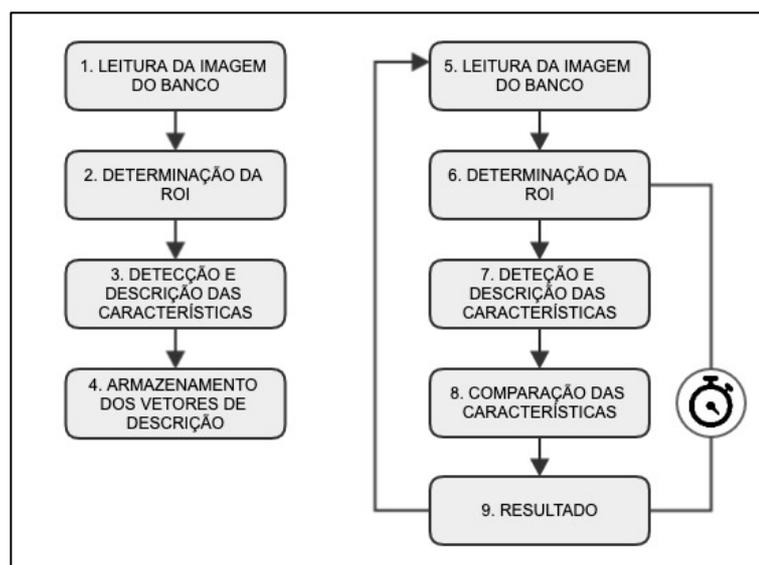
definido. Caso não seja encontrada uma região para cálculo, ou o valor de área encontrado seja diferente do intervalo de valores definido, o frasco é classificado como não conforme.

Nota-se que os parâmetros como a área padrão, o número de grupos para quantização, os valores do detector de bordas de Canny e a faixa de cor em HSV utilizado na máscara de segmentação precisam ser previamente ajustados. Considerando-se uma aplicação real, estes parâmetros poderiam ser obtidos através de uma interface gráfica, no entanto, observa-se a necessidade de uma expertise maior do usuário.

#### 4.3.4 Método ORB

O método ORB baseia-se na extração e na comparação de características dos objetos encontrados na imagem. Este método pode ser dividido em 9 etapas e estas etapas são agrupadas em duas partes. Na primeira parte, as etapas de 1 a 4 ocorrem apenas uma vez para cada modelo de frasco testado, enquanto na segunda parte, as etapas 5 a 9 ocorrem ciclicamente. O fluxograma apresentado na Figura 28 ilustra estas etapas e a ordem em que ocorrem.

Figura 28 - Fluxograma do método ORB.

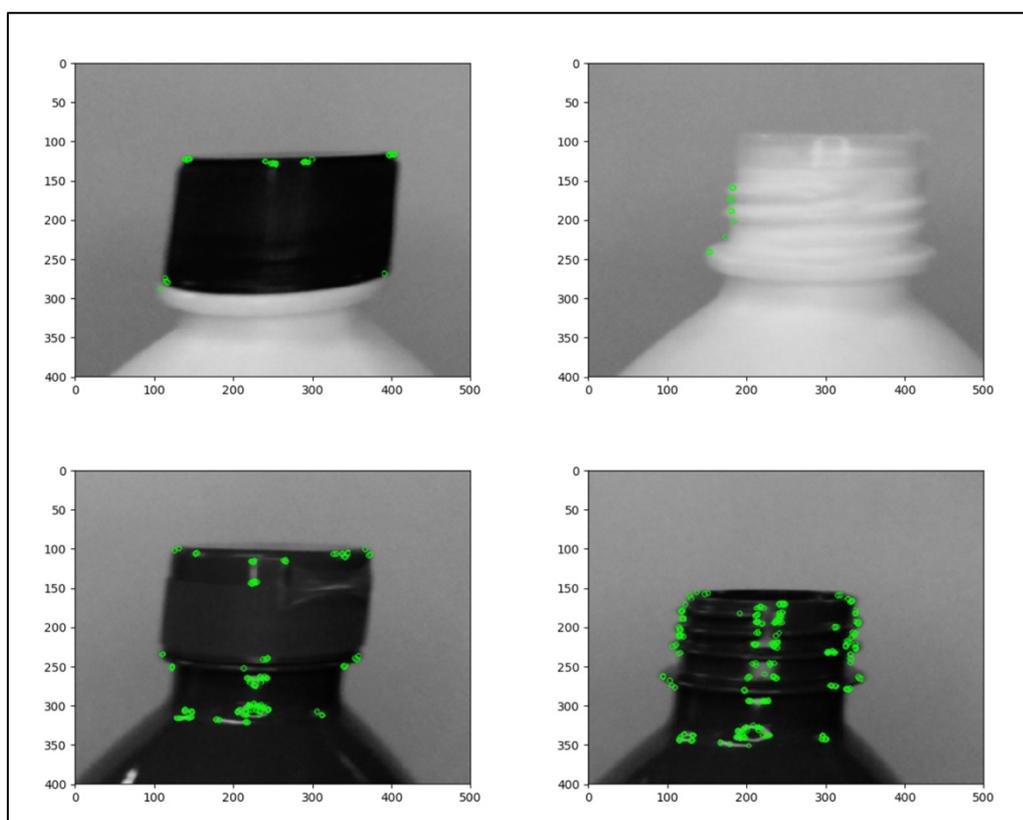


Fonte: O autor (2019).

Após o carregamento da imagem ocorre a determinação da ROI na segunda etapa. A determinação da ROI em formato retangular deste método utiliza o mesmo conceito e as mesmas ferramentas apresentadas na segunda etapa do método K-means. De forma similar, o objetivo inicial desta etapa visa a identificação aproximada das laterais e do topo do frasco e, na sequência, a redução da quantidade de informações a serem processadas.

Na terceira etapa acontece a detecção e a descrição das características da imagem. Para a execução desta etapa foi utilizado o algoritmo ORB. Este algoritmo apresenta características importantes para a aplicação proposta, como a velocidade de processamento, a invariância a escala e rotação e a geração de um descritor binário. Além destas características, o algoritmo ORB possui a vantagem de ser *Open Source* (RUBLEE *et al.*, 2011). A entrada deste algoritmo é a ROI em escala de cinza determinada na etapa anterior. As saídas desta etapa são os vetores de descrição de características encontrados na imagem de entrada. A Figura 29 ilustra os pontos de interesse detectados em imagens contendo frascos com e sem tampa dos modelos de frasco branco e verde.

Figura 29 - Pontos de interesse localizados na ROI pelo método ORB.



Fonte: O autor (2019).

A quarta etapa consiste no armazenamento dos vetores de descrição gerados pelos pontos de interesse ilustrados na Figura 29. Estes vetores armazenados serão utilizados como referência para comparação na oitava etapa deste método. Para o modelo de frasco branco utilizou-se como referência os vetores criados pela primeira imagem do frasco com tampa encontrada na base de dados. Já no caso do frasco verde, optou-se pela utilização dos vetores criados pela primeira imagem do frasco sem tampa. Esta diferença deve-se ao fato da maior concentração de pontos de interesse gerados pela área da rosca do frasco, o que possibilitou um aumento na assertividade das comparações.

Observa-se pelo fluxograma da Figura 28 que as etapas 5, 6 e 7 são equivalentes as etapas 1, 2, 3, respectivamente. Porém, na segunda parte do método ORB as etapas ocorrem de maneira cíclica, representando a inspeção *online* da linha de produção. Nota-se também que a partir da quinta etapa é iniciado o temporizador de processamento, que até então não estava sendo considerado.

Na oitava etapa ocorre a comparação dos vetores de descrição gerados nesta etapa em relação aos vetores gerados pelas imagens de referência na terceira etapa. Esta comparação foi realizada através do método da distância de Hamming. Este método é baseado na aplicação de um operador XOR entre os vetores comparados. O resultado desta operação será um número inteiro que expressa a proximidade entre os dois vetores comparados. O uso de uma operação XOR para a comparação de vetores se mostra adequada para aplicações em tempo real, uma vez que as CPU's modernas costumam possuir instruções dedicadas para este tipo de operação, o que torna o processamento otimizado (CALONDER *et al.*, 2010).

Na nona e última etapa obtém-se o resultado da inspeção. Para o modelo de frasco branco a aprovação acontece caso sejam encontradas um número mínimo de características entre os vetores da imagem do frasco que esta sendo inspecionado e os vetores da imagem de referência. No caso do frasco verde, a aprovação ocorre caso não seja encontrado um número mínimo de características compatíveis. Esta diferença ocorre devido ao fato de que os vetores de referência para o frasco verde foram gerados por uma imagem contendo um frasco sem tampa, enquanto para o modelo de frasco branco a imagem de referência continha um frasco tampado.

#### 4.4 AVALIAÇÃO DE DESEMPENHO

O resultado do processo de inspeção proposto neste trabalho pode ser tratado como um problema de classificação binária, pois o resultado da inspeção pode ser dividido em duas classes conhecidas. Com isso, uma matriz de confusão, como observada na Tabela 4 da seção 2.5, foi utilizada para expressar os resultados obtidos nos testes.

Dentro do contexto do processo de inspeção de frascos, pode-se redefinir o significado das variáveis TP, TN, FP e FN apresentadas na Tabela 4 através das definições de conformidade ou defeito, ilustradas na Figura 15. A variável *true positive* (TP) passa a corresponder ao número de frascos avaliados como conformes e que realmente não possuem defeitos. A variável *true negative* (TN) passa corresponder aos frascos que foram classificados como não conformes e que realmente apresentam algum defeito. A variável *false positive* (FP) indicará o número de frascos que foram classificados como conformes, mas que na verdade apresentavam algum defeito. Por último, a variável *false negative* (FN) passa a representar o número de frascos que foram avaliados como defeituosos, mas que na verdade eram conformes.

Com base nos dados obtidos pelos testes construiu-se as matrizes de confusão. A partir das informações destas matrizes algumas métricas puderam ser calculadas para a avaliação de desempenho do sistema de inspeção. Entre estas métricas pode-se destacar a acurácia, a especificidade e a sensibilidade, que são calculadas através das Equações 7, 8 e 9 respectivamente, descritas na revisão bibliográfica.

A acurácia representa a relação percentual do número de frascos corretamente avaliados e o total de frascos avaliados. No caso da métrica de sensibilidade, o resultado é a relação do total de frascos avaliados como conformes pelo total de frascos realmente conformes. A especificidade, por sua vez, representa a relação entre o número de fracos avaliados como defeituosos e o total de frascos realmente defeituosos (KUMCHOO; CHIRACHARIT, 2018).

Além das métricas apresentadas, foram avaliados os tempos de aquisição das imagens e os tempos de processamento de cada método. Estes parâmetros são importantes pois ajudam a identificar as limitações impostas pelo *hardware* e os limites de velocidade de produção ao qual o sistema pode ser submetido.

## 5 AVALIAÇÃO DO SISTEMA

Baseado nos métodos descritos no capítulo 4, este capítulo apresentará os resultados em termos de acurácia, especificidade, sensibilidade e tempo de processamento de cada método avaliado. Por fim, é apresentada uma comparação entre os resultados obtidos com intuito de avaliar o desempenho de cada abordagem aplicada no sistema proposto.

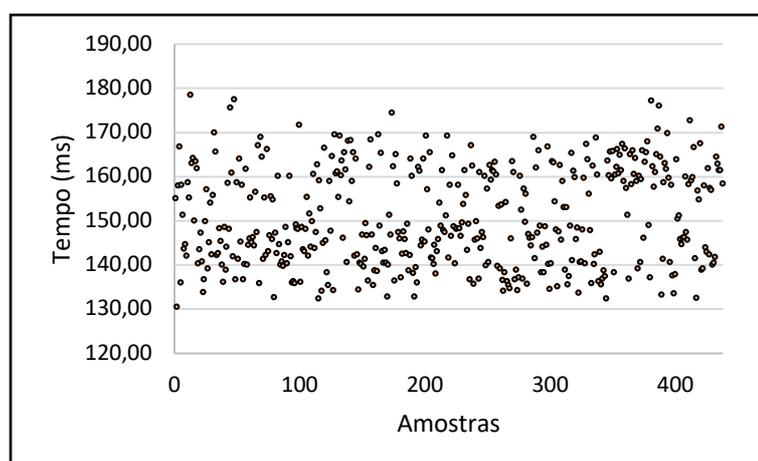
### 5.1 RESULTADOS OBTIDOS PARA CADA MÉTODO APLICADO

Nesta seção são apresentados os resultados obtidos para cada método de inspeção aplicado ao sistema proposto. Os resultados estão expressos em forma de tabelas e gráficos. Além disso, estão descritos comentários pertinentes a cada técnica implementada.

Vale observar que independentemente do método aplicado existe um tempo necessário para realizar a aquisição de imagens em linha de produção. Sendo assim, durante a etapa de aquisição das imagens na linha de produção realizou-se a medida do tempo de aquisição, conforme ilustrado no fluxograma de aquisição de imagens apresentado na Figura 17.

Neste contexto, as medidas de tempo de aquisição das imagens auxiliam na identificação dos limites de velocidade de produção impostos pelo sistema de inspeção. A Figura 30 apresenta o gráfico de dispersão com os valores de cada amostra de tempo de aquisição das 438 imagens da base de dados.

Figura 30 - Amostras do tempo de aquisição das imagens.



Fonte: O autor (2019).

Com base nos dados observados na Figura 30, realizaram-se alguns cálculos para obter informações temporais da etapa de aquisição de imagens. A Tabela 8 apresenta os parâmetros de maior tempo de aquisição, menor tempo de aquisição, a média destes valores e o seu desvio padrão expressos em milissegundos.

Tabela 8 - Parâmetros da etapa de aquisição de imagens.

<b>Parâmetro</b>	<b>Valor</b>	<b>Unidade</b>
Maior tempo de aquisição	178,50	ms
Menor tempo de aquisição	130,49	ms
Tempo médio de aquisição	150,73	ms
Desvio padrão do tempo de aquisição	9,70	ms

Fonte: O autor (2019).

Observa-se pela Figura 30 e pela Tabela 8 uma variação considerável entre os tempos de cada amostra registrada. Estas variações podem ocorrer como consequência de oscilações encontradas no sistema operacional, que são causadas principalmente por entradas de novas tarefas ou processos.

Além das oscilações naturais do sistema operacional, estas diferenças de tempo de aquisição também podem ocorrer devido a forma de funcionamento do *driver* da câmera. Utilizando-se a porta de saída de vídeo o *driver* Picamera permite que seja acessado somente um frame da sequência de frames armazenada no *buffer*. Com isso, elimina-se a necessidade de reinicialização do *driver* a cada nova aquisição, diminuindo consideravelmente o tempo de aquisição. No entanto, apesar de permitir capturas mais rápidas, a utilização da porta de vídeo para captura de imagens pode apresentar oscilações, visto que a prioridade do driver está na captura do próximo frame da sequência e não na entrega do frame de interesse (JONES, 2018).

### 5.1.1 Método das Linhas

Devido às limitações de segmentação do método das linhas, não foi possível realizar a avaliação do sistema proposto para a inspeção dos frascos verdes. Como discutido na seção 4.3.1, o método das linhas não resulta em linhas horizontais definidas para o modelo de frasco verde. Por isso, os resultados obtidos para este método são referentes somente ao frasco branco.

A Tabela 9 apresenta a matriz de confusão gerada a partir dos resultados de inspeção com o método das linhas para o frasco branco. Observa-se que foram geradas 6 avaliações de falso negativo, representando aproximadamente 1,9% do total de amostras do frasco branco.

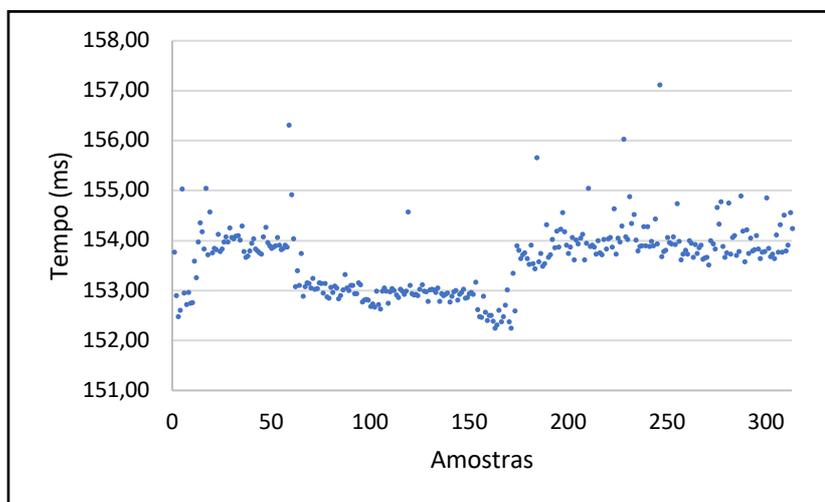
Tabela 9 - Matriz de confusão do método das linhas para o frasco branco.

		VERDADE	
		COM TAMPA	SEM TAMPA
ESTIMADO	COM TAMPA	146	0
	SEM TAMPA	6	161

Fonte: O autor (2019).

A Figura 31 ilustra a distribuição das amostras de tempo de processamento obtidos pela utilização do método da linhas para os frascos brancos nas condições de presença e falta de tampa.

Figura 31 - Tempos de processamento do método das linhas para o frasco branco.



Fonte: O autor (2019).

A partir dos dados apresentados na Figura 31, calculou-se os principais parâmetros utilizados para a avaliação do método, sendo que estes são apresentados na Tabela 10.

Tabela 10 - Resultados obtidos pelo método das linhas.

<b>Parâmetro</b>	<b>Frasco branco</b>	<b>Frasco Verde</b>	<b>Unidade</b>
Tempo de processamento médio	153,59	-	ms
Maior tempo de processamento	157,11	-	ms
Menor tempo de processamento	152,24	-	ms
Desvio padrão	0,54	-	ms
Acurácia	98%	-	
Sensibilidade	100%	-	
Especificidade	96%	-	
Tempo médio de inspeção	304,32	-	ms
Maior tempo de inspeção (pior caso)	335,61	-	ms

Fonte: O autor (2019).

Neste trabalho, definiu-se o parâmetro de “tempo médio de inspeção” como sendo a soma do tempo médio de aquisição, apresentado na Tabela 8, com o tempo médio de processamento. De forma análoga, o parâmetro de “maior tempo de inspeção” é o resultado da soma do maior tempo de aquisição com o maior tempo de processamento e representa a condição de pior caso. Estes dois parâmetros auxiliam na identificação dos limites de operação do sistema e serão utilizados nas comparações.

Observa-se pelos dados apresentados na Tabela 10 que o método das linhas obteve um elevado valor de sensibilidade para o frasco branco. Este valor demonstra que não houveram classificações erradas para a condição de frasco sem tampa. Entretanto, para a condição de frasco com tampa houveram 6 classificações incorretas, o que reduziu o parâmetro de especificidade para 96% e o valor de acurácia para 98%.

Os erros de avaliação encontrados neste método são originados principalmente pela incorreta detecção das bordas e, como consequência, a incorreta localização das linhas de referência. Isto ocorre porque os parâmetros do detector de bordas de Canny são fixos e podem não atender de maneira satisfatória todas as imagens testadas. Estes fatores são agravados quando ocorrem variações na luminosidade do ambiente.

Para este método, tanto o tempo médio de inspeção quanto o maior tempo de inspeção ficaram abaixo do limite de 441,1 ms imposto pela linha de produção. Desta forma, este método atende ao critério de velocidade de inspeção da linha de produção na qual o sistema foi inicialmente aplicado, mesmo quando se considera o pior caso.

### 5.1.2 Método das áreas

A utilização do método das áreas permitiu a avaliação dos dois modelos de frascos, diferente do método das linhas. As Tabelas 11 e 12 apresentam as matrizes de confusão geradas pela aplicação do método das áreas com as imagens de frascos brancos e com as imagens de frascos verdes, respectivamente. Destaca-se que a base de dados de imagens utilizada para avaliação é a mesma para todos os métodos.

Observa-se pela matriz de confusão apresentada na Tabela 11 que não houveram erros nas avaliações executadas pelo método das áreas para o frasco branco. Já para o modelo de frasco verde na Tabela 12, o método apresentou 24 avaliações falsas negativas, ou seja, na prática o sistema descartaria da linha de produção 24 unidades de frascos que na verdade estariam em conformidade.

Tabela 11 - Matriz de confusão do método das áreas para o frasco branco.

		VERDADE	
		COM TAMPA	SEM TAMPA
ESTIMADO	COM TAMPA	152	0
	SEM TAMPA	0	161

Fonte: O autor (2019).

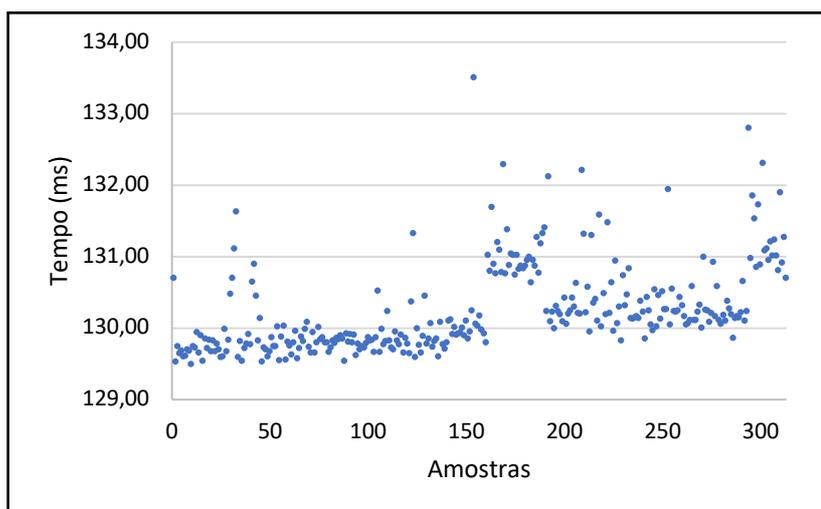
Tabela 12 - Matriz de confusão do método das áreas para o frasco verde.

		VERDADE	
		COM TAMPA	SEM TAMPA
ESTIMADO	COM TAMPA	37	0
	SEM TAMPA	24	64

Fonte: O autor (2019).

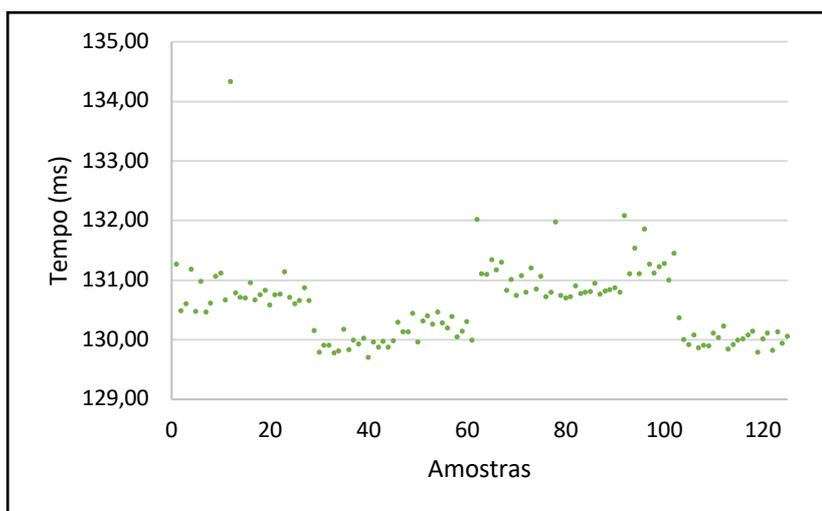
Os gráficos das Figuras 32 e 33 ilustram a distribuição das medidas de tempo de processamento do método das áreas para os modelos de frasco branco e verde, respectivamente.

Figura 32 - Tempos de processamento do método das áreas para o frasco branco.



Fonte: O autor (2019).

Figura 33 - Tempos de processamento do método das áreas para o frasco verde.



Fonte: O autor (2019).

Com base nos dados apresentados e na metodologia de avaliação adotada, calculou-se os principais parâmetros do método das áreas, permitindo assim a posterior comparação com os demais métodos avaliados. A Tabela 13 apresenta estes parâmetros calculados para os dois modelos de frascos avaliados.

Tabela 13 - Resultados obtidos pelo método das áreas.

<b>Parâmetro</b>	<b>Frasco branco</b>	<b>Frasco Verde</b>	<b>Unidade</b>
Tempo de processamento médio	130,25	130,58	ms
Maior tempo de processamento	133,50	134,33	ms
Menor tempo de processamento	129,49	129,70	ms
Desvio padrão	0,47	0,48	ms
Acurácia	100%	81%	
Sensibilidade	100%	100%	
Especificidade	100%	61%	
Tempo médio de inspeção	280,98	281,31	ms
Maior tempo de inspeção (pior caso)	311,99	312,82	ms

Fonte: O autor (2019).

O método das áreas apresentou um elevado valor de acurácia, sensibilidade e especificidade para o modelo de frasco branco, pois não houveram erros nas avaliações executadas para este frasco. Este alto índice de acertos se deve

principalmente pela segmentação precisa gerada pelo algoritmo de Otsu (ABDELHEDI; TAOUIL; HADJKACEM, 2012) (KAEHLER; BRADSKI, 2016).

No caso do modelo de frasco verde, percebe-se uma diminuição nos acertos e, como consequência, a diminuição nos parâmetros de acurácia e especificidade. Os erros de inspeção para este frasco se devem principalmente a baixa variação de intensidade entre a região da tampa e do frasco. Com isso, o algoritmo de Otsu não obteve sucesso na segmentação somente da região da tampa e foi necessário considerar uma área maior para a avaliação. Como a variação entre a área de um frasco com tampa e de um frasco sem tampa é pequena, a mesma fica mais suscetível a variações geradas pelo posicionamento do frasco, o que aumenta o número de avaliações incorretas.

Quanto a análise dos tempos de inspeção, pode-se observar que o método das áreas atende ao limite imposto pela linha de produção para os dois modelos de frascos, mesmo quando considerado o maior tempo de inspeção (pior caso).

### 5.1.3 Método K-means

As Tabelas 14 e 15 apresentam as matrizes de confusão geradas pela utilização do método K-means para o modelo de frasco branco e verde, respectivamente.

Tabela 14 - Matriz de confusão do método K-means para o frasco branco.

		VERDADE	
		COM TAMPA	SEM TAMPA
ESTIMADO	COM TAMPA	152	0
	SEM TAMPA	0	161

Fonte: O autor (2019).

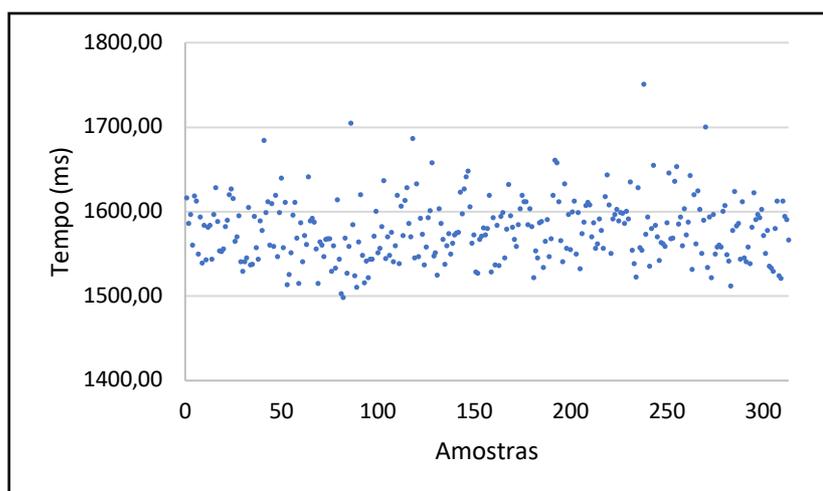
Tabela 15 - Matriz de confusão do método K-means para o frasco verde.

		VERDADE	
		COM TAMPA	SEM TAMPA
ESTIMADO	COM TAMPA	61	0
	SEM TAMPA	0	64

Fonte: O autor (2019).

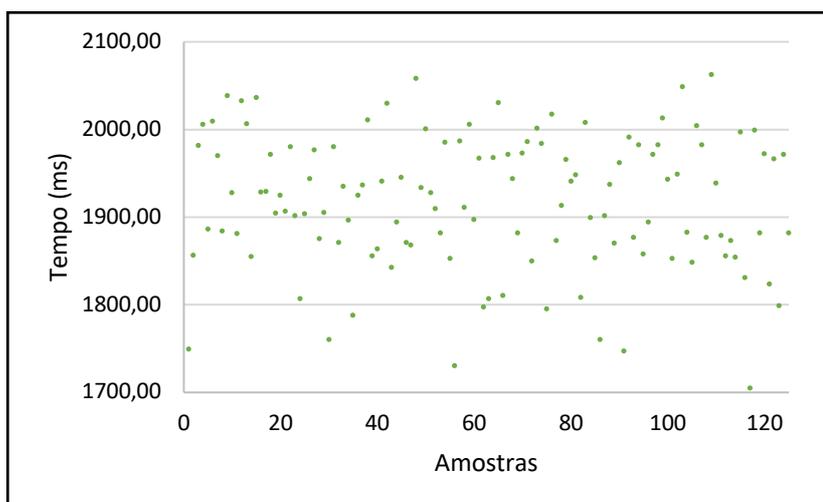
Nota-se através das matrizes de confusão apresentadas nas Tabelas 14 e 15 que este método não apresentou avaliações falsas para os dois modelos de frascos. Os gráficos apresentados nas Figuras 34 e 35 apresentam o tempos de processamento obtidos pelo método K-means para os frascos brancos e verdes, respectivamente.

Figura 34 - Tempos de processamento do método K-means para o frasco branco.



Fonte: O autor (2019).

Figura 35 - Tempos de processamento do método K-means para o frasco verde.



Fonte: O autor (2019).

Utilizando os dados apresentados nas Figuras 34 e 35 pode-se calcular os parâmetros para avaliação do método, conforme apresenta a Tabela 16.

Tabela 16 - Resultados obtidos pelo método K-means.

<b>Parâmetro</b>	<b>Frasco branco</b>	<b>Frasco Verde</b>	<b>Unidade</b>
Tempo de processamento médio	1578,54	1917,42	ms
Maior tempo de processamento	1751,29	2062,32	ms
Menor tempo de processamento	1498,64	1705,03	ms
Desvio padrão	28,98	62,69	ms
Acurácia	100%	100%	
Sensibilidade	100%	100%	
Especificidade	100%	100%	
Tempo médio de inspeção	1729,27	2068,15	ms
Maior tempo de inspeção (pior caso)	1929,78	2240,82	ms

Fonte: O autor (2019).

O método K-means não apresentou avaliações incorretas tanto para o frasco branco quanto para o frasco verde, fazendo com que todas as métricas de avaliação obtivessem o valor máximo. O aumento dos acertos na avaliação do frasco verde deve-se a segmentação mais precisa obtida pelo método K-means, quando comparado ao método das áreas. Esta melhora na segmentação permitiu um distanciamento entre os valores de área encontrados para a condição de frasco com

tampa em relação aos valores de área dos frascos sem tampa. Com isso, foi possível determinar um valor de limiar que separasse as duas classes de maneira precisa.

Com o intuito de diminuir o tempo de processamento do sistema, limitou-se em 3 a quantidade de iterações para a quantização do frasco branco e em 5 a quantidade de iterações para quantização do frasco verde. Porém, mesmo com estas limitações, o método K-means não apresentou resultados de tempo de processamento satisfatórios. Todos os resultados foram superiores ao tempo máximo permitido por frasco de 441,1 ms determinado pela velocidade nominal da linha de produção. Assim sendo, este método não permitiria a inspeção em tempo real da linha de produção na qual este trabalho foi baseado.

#### 5.1.4 Método ORB

O último método de inspeção avaliado foi o método ORB. As Tabelas 17 e 18 apresentam as matrizes de confusão deste método para os modelos de frasco branco e verde, respectivamente. Pode-se observar que não houveram erros de classificação pela utilização do método ORB para ambos os frascos.

Tabela 17 - Matriz de confusão do método ORB para o frasco branco.

		VERDADE	
		COM TAMPA	SEM TAMPA
ESTIMADO	COM TAMPA	152	0
	SEM TAMPA	0	161

Fonte: O autor (2019).

Tabela 18 - Matriz de confusão do método ORB para o frasco verde.

		VERDADE	
		COM TAMPA	SEM TAMPA
ESTIMADO	COM TAMPA	61	0
	SEM TAMPA	0	64

Fonte: O autor (2019).

As Figuras 36 e 37 apresentam os gráficos com os tempos de processamento do método ORB para o frasco branco e verde, respectivamente.

Pode-se observar através dos gráficos das Figuras 36 e 37 uma nítida diferença no tempo de processamento entre as amostras de cada modelo de frasco. Esta diferença ocorre devido ao número de pontos de interesse que são localizados em cada condição dos frascos, conforme pode ser observado na Figura 29. No modelo de frasco branco, a quantidade de pontos de interesse localizados e descritos é maior na condição de frasco com tampa, e por isso estas imagens levam mais tempo para serem processadas. Já para o modelo de frasco verde, a quantidade de pontos de interesse é maior na condição de frasco sem tampa, o que aumenta o tempo de processamento para estas imagens.

Com base nos dados das etapas anteriores, a Tabela 19 apresenta os parâmetros calculados para comparação do método ORB.

Tabela 19 - Resultados do método ORB.

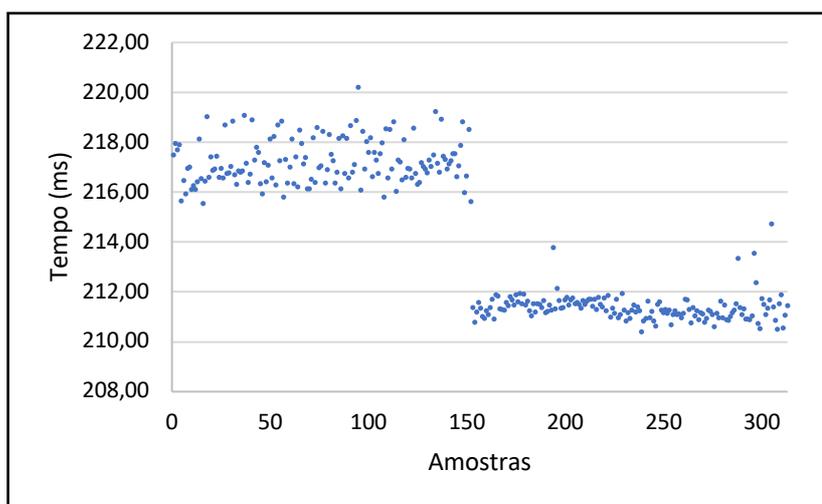
(continua)

Parâmetro	Frasco branco	Frasco Verde	Unidade
Tempo de processamento médio	214,21	253,63	ms
Maior tempo de processamento	220,19	281,84	ms
Menor tempo de processamento	210,41	241,17	ms
Desvio padrão	2,93	7,14	ms

			(conclusão)
Acurácia	100%	100%	
Sensibilidade	100%	100%	
Especificidade	100%	100%	
Tempo médio de inspeção	364,95	404,36	ms
Maior tempo de inspeção (pior caso)	398,69	460,33	ms

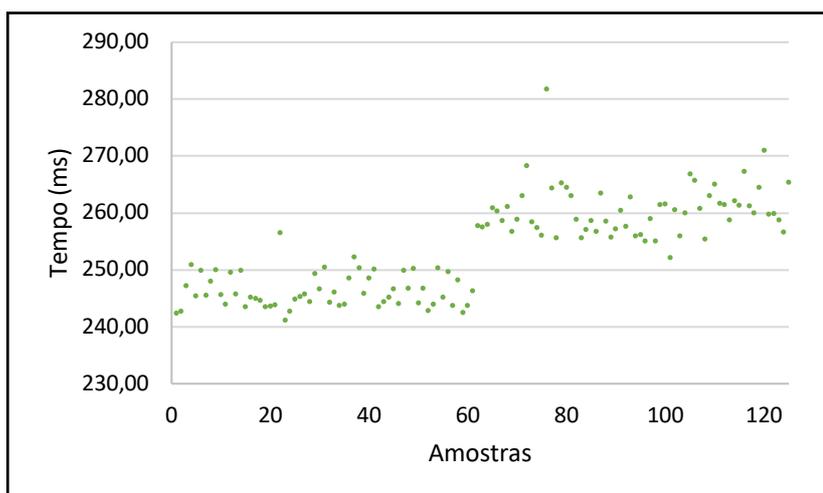
Fonte: O autor (2019).

Figura 36 - Tempos de processamento do método ORB para o frasco branco.



Fonte: O autor (2019).

Figura 37 - Tempos de processamento do método ORB para o frasco verde.



Fonte: O autor (2019).

Observa-se que o método ORB também apresentou elevados valores de acurácia, sensibilidade e especificidade, sendo que não foram executadas avaliações

incorretas nas imagens da base de dados. Estes elevados índices de acerto se devem principalmente pela boa capacidade de localização e descrição dos pontos de interesse em cada modelo e condição de frasco.

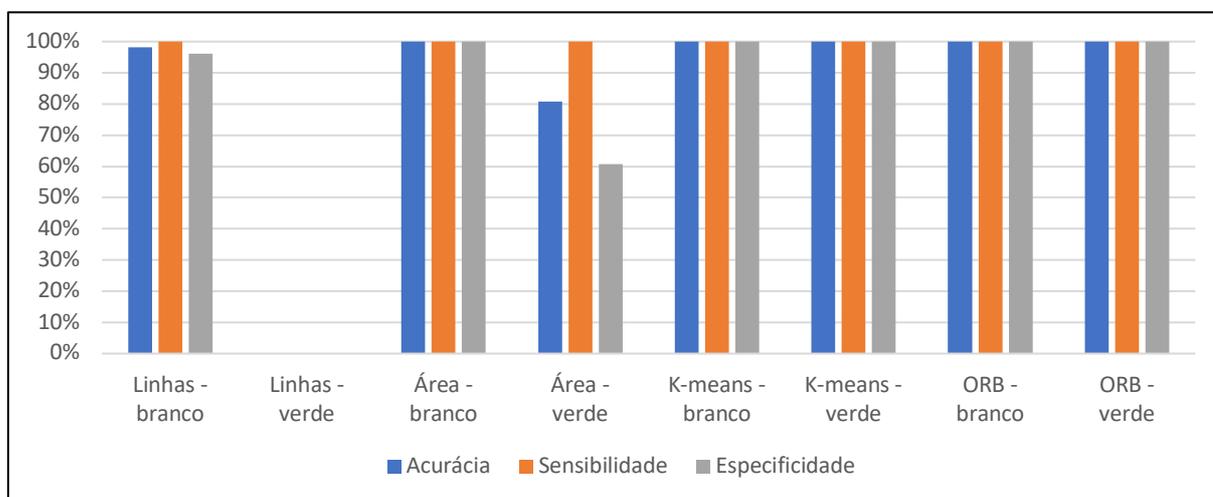
Quanto ao tempo de inspeção, pode-se observar que este método atendeu o limite de tempo de inspeção para o frasco branco, mesmo quando considerado o pior caso. Porém, nota-se que para o modelo de frasco verde este limite foi ultrapassado quando se considera o maior tempo de inspeção medido no processo.

Estas diferenças de tempo entre os modelos de frasco devem-se principalmente pela quantidade de vetores necessários para realizar a correta separação entre as condições de presença e ausência de tampa. Enquanto no modelo de frasco branco foram necessários no máximo 41 vetores para realizar a descrição e a comparação dos frascos, no modelo de frasco verde foram necessários 179 vetores.

## 5.2 COMPARAÇÃO E ANÁLISE DOS RESULTADOS OBTIDOS

Nesta seção são comparados os resultados obtidos pelos métodos implementados. Inicialmente será realizada a comparação entre o valores de acurácia, sensibilidade e especificidade de cada método. A Figura 38 apresenta um gráfico com os desempenhos de cada método agrupados por modelo de frasco. Neste gráfico as legendas são formadas pelo nome do método seguido pelo modelo de frasco avaliado conforme “Nome do método – modelo do frasco”.

Figura 38 - Comparativo de desempenho dos métodos.

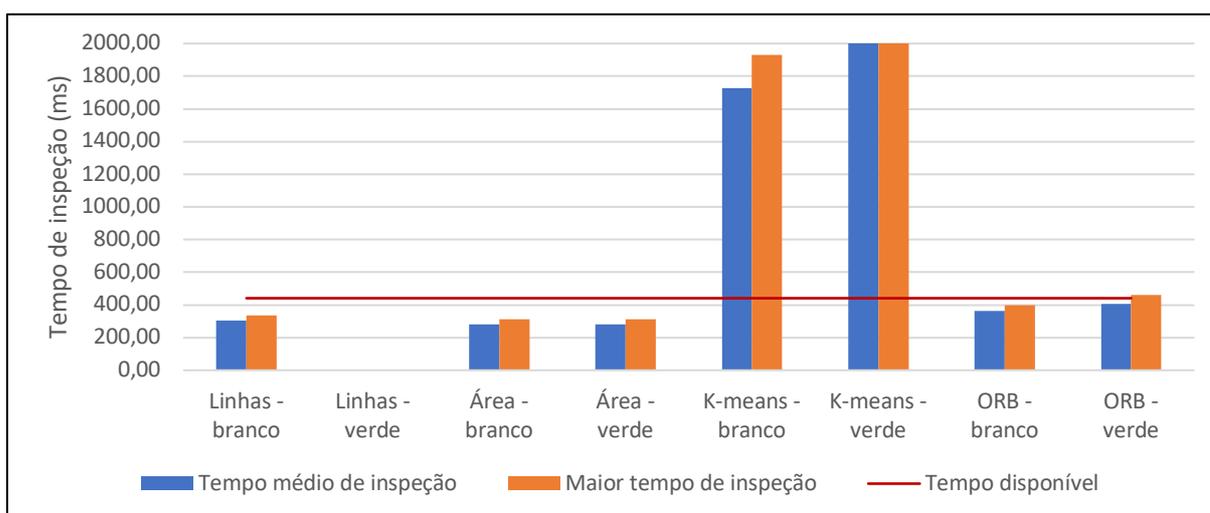


Fone: O autor (2019).

A partir da análise da Figura 38, pode-se observar que somente dois métodos obtiveram 100% de acerto para os dois modelos de frasco, sendo eles o método ORB e o método K-means. O método das áreas apresentou 100% de acurácia para o modelo de frasco branco, mas o mesmo não ocorreu para o frasco verde. No caso do método das linhas, nenhum dos modelos de frasco obteve 100% de acurácia.

Além dos parâmetros de desempenho foram comparados os parâmetros de tempo de inspeção de cada método. A Figura 39 apresenta um gráfico com os tempos médio e máximo para a inspeção de cada método e modelo de frasco. Neste figura também está traçada uma linha que representa o tempo máximo disponível por frasco, considerando a condição de velocidade de produção nominal da máquina de envase.

Figura 39 - Comparativo entre os tempos de inspeção.



Fonte: O autor (2019).

Nota-se que os métodos das linhas e das áreas obtiveram tempos de inspeção dentro do limite imposto pela velocidade de produção. No caso do método ORB, os tempos de inspeção do modelo de frasco branco também ficaram dentro do limite de tempo disponível. No entanto, para o modelo de frasco verde o método ORB ultrapassou o limite de tempo disponível quando considerado o pior caso. O método K-means apresentou resultados de tempo de processamento insatisfatórios tanto para o frasco branco quanto para o frasco verde, ficando aproximadamente 4,5 vezes acima do limite de tempo disponível.

## 6 CONCLUSÃO

A inspeção automática de frascos está diretamente ligada ao controle das linhas de envase, uma vez que defeitos, como a ocorrência de frascos destampados, podem significar diminuição na eficiência da produção e queda na qualidade do produto final. No entanto, há aplicações onde o alto custo dos sistemas de inspeção automáticos inviabilizam o seu uso. Desta forma, nota-se a importância do desenvolvimento de sistemas de inspeção de baixo custo que viabilizem a sua implementação, principalmente para indústrias de pequeno e médio porte.

Baseado neste contexto, o objetivo deste trabalho foi desenvolver um sistema de baixo custo que possibilite a detecção de presença de tampas em frascos. Para tal, utilizou-se como plataforma de desenvolvimento a placa Raspberry Pi 3 Model B em conjunto com a câmera Pi Camera Module v2, sendo que estes equipamentos são de baixo custo e de fácil aquisição no mercado nacional.

A partir do conjunto plataforma de desenvolvimento e câmera, foi criada uma base de dados contendo imagens de frascos com e sem tampa, capturadas em uma linha de produção real. Para diversificar os testes, foram capturadas imagens de dois modelos de frascos com características de cor, material e formato distintas.

Com o intuito de selecionar um método adequado para realizar esta inspeção foram avaliadas diferentes abordagens. A escolha dos métodos avaliados foi baseada no estado da arte, levando em consideração a sua adequação às características da linha de produção e às limitações impostas pelo *hardware* utilizado. Com isso, foram implementados 4 diferentes métodos, sendo denominados neste trabalho como método das linhas, método das áreas, método K-means e método ORB.

Visando à determinação da eficiência e do desempenho do algoritmo como um todo, foram tomadas medidas de acurácia, especificidade e sensibilidade dos algoritmos avaliados. Afim de atender o critério de inspeção em tempo real, foram realizadas as medidas de tempos de aquisição e de tempos de processamento de cada método avaliado.

Por meio da análise dos resultados, observou-se que os melhores resultados para a inspeção do modelo de frasco branco de material PEAD e tampa azul foi o método das áreas. Com esta abordagem foram possíveis resultados de acurácia, especificidade e sensibilidade em 100%, o que indica que não houveram erros de classificação. Este método também apresentou o menor tempo de inspeção, sendo

que, mesmo considerando-se o pior caso, obteve-se um tempo total de inspeção de 311,9 ms. Este tempo representa aproximadamente 70% do tempo total disponível por frasco da linha de produção na qual este trabalho foi baseado, o que viabilizaria a sua aplicação em uma situação real.

Já para o conjunto de frasco e tampa verde de material PET, os melhores resultados foram obtidos pelo método ORB. Com este método foram encontrados resultados de acurácia, especificidade e sensibilidade em 100%, o que também demonstra que não houveram erros de classificação deste método. Quanto ao tempo de inspeção foram necessários 460,3 ms, quando considerado o pior caso. Nota-se que este tempo é aproximadamente 4,3% acima do limite de tempo disponível por frasco na aplicação deste trabalho. Este problema poderia resultar em classificações incorretas ou defasadas, que poderiam impossibilitar o correto rejeite de um frasco defeituoso. Entretanto, um estudo mais aprofundado sobre o caso poderia encontrar alternativas para reverter esta situação problema.

Vale observar que durante a etapa de aquisição de imagens foram detectadas distorções causadas pelo efeito de *rolling shutter* do sensor e borrões causados pelo movimento dos frascos. Estes defeitos agravaram-se principalmente pela necessidade de um elevado de tempo de exposição, imposto pelas limitações da iluminação artificial utilizada. Além disso, observa-se que o tempo médio necessário para aquisição de imagens ficou elevado, representando aproximadamente 34% do tempo total disponível por frasco, o que diminui consideravelmente o tempo destinado para o processamento das imagens. O elevado tempo de aquisição foi causado pelas limitações do *hardware* utilizado e pelo modo de operação encontrado no *driver* Picamera. Logo, para a obtenção de melhores resultados quanto ao tempo de aquisição de imagens e qualidade das mesmas, poderia-se realizar a troca da Picamera por uma câmera de maior qualidade e desempenho.

A classificação resultante dos métodos avaliados, por sua vez, apresentou problemas devido à baixa variabilidade de cores e intensidades, bem como a semelhança entre algumas regiões, como é o caso encontrado para o frasco verde com tampa verde. Observa-se que quanto menor é a variação encontrada entre as regiões de interesse das imagens, mais robusto precisará ser o algoritmo de detecção.

Nota-se, também, que as variações na iluminação e nas condições do ambiente de inspeção podem afetar de maneira significativa a assertividade do sistema. Além

disso, os métodos avaliados são dependentes de parametrizações pré-estabelecidas e que, por sua vez, dependem em certo grau da expertise do usuário.

Para trabalhos futuros, sugere-se a implementação deste sistema com um conjunto maior de frascos e tampas, preferencialmente instalado em linha de produção e suscetível a diferentes condições de iluminação.

Também são sugeridas melhorias que possibilitem a aquisição de imagens em menor tempo. Estas melhorias podem se dar através de alterações no *driver* de controle Picamera ou através da substituições do *hardware* de captura das imagens. Uma interface gráfica que simplifique a etapa de parametrização poderia ser implementada, possibilitando uma diminuição no tempo de *setup* do sistema.

Outros métodos de detecção que façam uso de algoritmos mais robustos e generalistas também poderiam ser avaliados, entretanto, deve-se atentar quanto as limitações de processamento do *hardware* e a característica de operação em tempo real.

## REFERÊNCIAS

- ABDELHEDI, Slim; TAOUIL, Khaled; HADJKACEM, Bassem. Design of Automatic Vision-based Inspection System for Monitoring in an Olive Oil Bottling Line. **International Journal Of Computer Applications**. Sfax, p. 39-46. ago. 2012.
- ADLER, David. **Rolling Shutter versus Global Shutter**. 2016. Disponível em: <https://www.bhphotovideo.com/explora/video/tips-and-solutions/rolling-shutter-versus-global-shutter>. Acesso em: 23 ago. 2018.
- AIT-AIDER, Omar; BARTOLI, Adrien; ANDREFF, Nicolas. Kinematics from Lines in a Single Rolling Shutter Image. In: IEEE CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION, 1., 2007, Minneapolis. **Proceedings...** . Minneapolis: Ieee, 2007. p. 1 - 6.
- AZHAR, Kanza *et al.* Computer Vision Based Detection and Localization of Potholes in Asphalt Pavement Images. In: IEEE CANADIAN CONFERENCE ON ELECTRICAL AND COMPUTER ENGINEERING (CCECE), 1., 2016, Vancouver. **Proceedings...** . Vancouver: IEEE, 2016. p. 1 - 5.
- BALDNER, Felipe *et al.* **Metrologia por imagem**. Brasil: Elsevier Brasil, 2017. 312 p.
- BRASIL, Robopac. **Nossos cases: Saneantes**. 2018. Disponível em: <http://www.robopac.com.br/cases/saneante>. Acesso em: 13 maio 2019.
- CALONDER, Michael *et al.* BRIEF: Binary Robust Independent Elementary Features. In: 11TH EUROPEAN CONFERENCE ON COMPUTER VISION,, 11., 2010, Heraklion. **Proceedings...** . Heraklion: Springer, 2011. p. 778 - 792.
- CHIAO, Yen-hao; LEE, Tung-ying; LAI, Shang-hong. Rolling shutter correction for video with large depth of field. In: IEEE INTERNATIONAL CONFERENCE ON IMAGE PROCESSING, 1., 2013, Melbourne. **Proceedings...** . Melbourne: IEEE, 2013. p. 900 - 904.
- COGNEX CORPORATION. **Guia do especialista: como escolher a iluminação certa para aplicações de visão de máquina**. Natick: 2012.
- COGNEX CORPORATION. **Introdução ao sistema de visão: um guia para automação de processos e melhorias de qualidade**. Natick: 2018.
- DAVIES, Roy. **Computer and Machine Vision: Theory, Algorithms, Practicalities**. 4. ed. Waltham: Elsevier, 2012. 871 p.
- DUAN, Feng; WANG, Yaonan; LIU, Huanjun. A real-time machine vision system for bottle finish inspection. In: INTERNATIONAL CONFERENCE ON CONTROL, AUTOMATION, ROBOTICS AND VISION, 8, Kunming. **Proceedings...**Kunming: IEEE , 2004. p. 842-846.

FUKUNAGA, K. **Introduction to Statical Pattern Recognition**. 2<sup>a</sup>. ed. Academic Press, 2013.

GONZALEZ, Rafael C.; WOODS, Richard C.. **Processamento digital de imagens**. 3. ed. São Paulo: Pearson Prentice Hall, 2010.

JIA, Jiancheng. A Machine Vision Application for Industrial Assembly Inspection. In: INTERNATIONAL CONFERENCE ON MACHINE VISION, 2., 2009, Dubai. **Proceedings...** Dubai: IEEE, 2009. p. 172 - 176.

JONES, Dave. **Picamera 1.13 Documentation**. 2018. Disponível em: <https://media.readthedocs.org/pdf/picamera/latest/picamera.pdf>. Acesso em: 20 out. 2018.

KAEHLER, Adrian; BRADSKI, Gary. **Learning OpenCV 3: Computer Vision in C++ with the OpenCV Library**. Sebastopol: O'reilly Media, 2016. 967 p.

KUMAR, Rekhil M; K, Sreekumar. A Survey on Image Feature Descriptors. In: INTERNATIONAL JOURNAL OF COMPUTER SCIENCE AND INFORMATION TECHNOLOGIES, 6., 2014, Kottayam. **Proceedings...** . Kottayam: IJCSIT, 2014. v. 5, p. 7668 - 7673.

KUMCHOO, Wirat; CHIRACHARIT, Werapon. Detection of loose cap and safety ring for pharmaceutical glass bottles. In: 15TH INTERNATIONAL CONFERENCE ON ELECTRICAL ENGINEERING/ELECTRONICS, COMPUTER, TELECOMMUNICATIONS AND INFORMATION TECHNOLOGY, 15., 2018, Bangkok. **Proceedings...** . Chiang Rai: IEEE, 2018. p. 125 - 130.

LIU, Yuhuan *et al.* Defect Inspection of Medicine Vials Using LBP Features and SVM Classifier. In: 2017 2ND INTERNATIONAL CONFERENCE ON IMAGE, VISION AND COMPUTING, 2., 2017, Tianjin. **Proceedings...** . Chengdu: IEEE, 2017. p. 41 - 45.

LIVE, Creative. **What is Shutter Speed? How Simple Shutter Speed Settings Create Motion in Still Photos**. Disponível em: <https://www.creativelive.com/photography-guides/what-is-shutter-speed>. Acesso em: 21 ago. 2018.

MANSUROV, Nasim. **Introduction to Shutter Speed in Photography**. 2018. Disponível em: <https://photographylife.com/what-is-shutter-speed-in-photography>. Acesso em: 21 ago. 2018.

MARQUES FILHO, Ogê; VIEIRA NETO, Hugo. **Processamento Digital de Imagens**. Rio de Janeiro: Brasport, 1999.

NOBLE, Frazer K.. Comparison of OpenCV's feature detectors and feature matchers. In: INTERNATIONAL CONFERENCE ON MECHATRONICS AND MACHINE VISION IN PRACTICE (M2VIP), 23., 2016, Auckland. **Proceedings...** . Nanjing: IEEE, 2016.

PROJECT, Code. **Canny Edge Detection in C#**. 2012. Disponível em: <https://www.codeproject.com/Articles/93642/Canny-Edge-Detection-in-C>. Acesso em: 23 out. 2018.

RASPBERRYPI. Site da RaspberryPI - **Produtos**. Disponível em: <https://www.raspberrypi.org/products/>. Acesso em: 15 Out. 2018.

Rosten E., Drummond T. (2006) Machine Learning for High-Speed Corner Detection. In: Leonardis A., Bischof H., Pinz A. (eds) Computer Vision – ECCV 2006. ECCV 2006. Lecture Notes in Computer Science, vol 3951. Springer, Berlin, Heidelberg

RUBLEE, Ethan *et al.* ORB: An efficient alternative to SIFT or SURF. In: INTERNATIONAL CONFERENCE ON COMPUTER VISION, 1., 2011, Barcelona. **Proceedings...** . Menlo Park: IEEE, 2011. p. 2564 - 2571.

SACHAN, Ankit. **Embedded Computer Vision: Which device should you choose?** 2017. Disponível em: <https://www.learnopencv.com/embedded-computer-vision-which-device-should-you-choose/>. Acesso em: 29 out. 2018.

SHARMA, Shubhi *et al.* EMPTY AND FILLED BOTTLE INSPECTION SYSTEM. **ICTACT Journal On Image And Video Processing**. Chennai, p. 1122-1126. nov. 2015.

SOLOMON, Chris; BRECKON, Toby. **Fundamentos de Processamento Digital de Imagens: Uma Abordagem Prática com Exemplos em Matlab**. Rio de Janeiro: LTC, 2013. Tradução: José Rodolfo Souza.

SZELISKI, Richard. **Computer Vision: Algorithms and Applications**. London: Springer, 2010. 812 p.

XIE, Hongwei *et al.* A Rapid Inspection Method for Encapsulating Quality of PET Bottles Based on Machine Vision. In: IEEE INTERNATIONAL CONFERENCE ON COMPUTER AND COMMUNICATIONS, 3., 2017, Guangzhou. **Proceedings...** . Chengdu: IEEE, 2017. p. 2025 - 2028.

XU, Min; MA, Yue; CHEN, Shuai. Research on Real-Time Quality Inspection of PET Bottle Caps. In: IEEE INTERNATIONAL CONFERENCE ON INFORMATION AND AUTOMATION (ICIA), 1., 2017, Shenyang. **Proceedings...** . Macau: IEEE, 2017. p. 1023 - 1026.