

**FUNDAÇÃO OSWALDO ARANHA  
CENTRO UNIVERSITÁRIO DE VOLTA REDONDA  
CURSO DE GRADUAÇÃO EM ENGENHARIA ELÉTRICA  
TRABALHO DE CONCLUSÃO DO CURSO**

**DIOGO VIANA SOARES DIAS  
JOSÉ CARLOS TORRES JUNIOR  
LUCAS RAFAEL CONS DUQUE**

**VISÃO COMPUTACIONAL APLICADA A CENTRALIZAÇÃO DE  
TIRAS**

**VOLTA REDONDA  
2018**

**FUNDAÇÃO OSWALDO ARANHA  
CENTRO UNIVERSITÁRIO DE VOLTA REDONDA  
CURSO DE GRADUAÇÃO EM ENGENHARIA ELÉTRICA  
TRABALHO DE CONCLUSÃO DO CURSO**

**VISÃO COMPUTACIONAL APLICADA A CENTRALIZAÇÃO DE  
TIRAS**

Trabalho de Conclusão de Curso apresentado ao Curso de Engenharia Elétrica do Uni-FOA como requisito à obtenção do título de Engenheiros Eletricistas.

**Alunos:**

Diogo Viana Soares Dias  
José Carlos Torres Junior  
Lucas Rafael Cons Duque

**Orientador:**

Prof. Me. Orlando Moreira Guedes Junior

**VOLTA REDONDA**

**2018**

## FOLHA DE APROVAÇÃO

**Curso:** Engenharia Elétrica

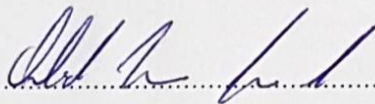
**Acadêmico: Matrícula:** Diogo Viana Soares Dias 201410505; José Carlos Torres Júnior 201410105; Lucas Rafael Cons Duque 201410541.

**Título do TCC:** Visão Computacional aplicada e centralização de tiras.

Apresentado publicamente perante a Banca Avaliadora, como parte dos requisitos para conclusão do Curso de Engenharia Elétrica.

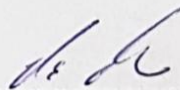
Aprovada em 17 de novembro de 2019

**Banca Avaliadora:**



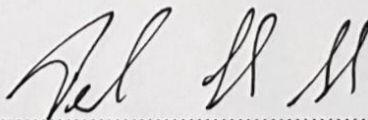
Professor Orientador

Orlando Moreira Guedes Junior, Mestre, UniFOA



Professor Avaliador

Edson de Paula Carvalho, Mestre, UniFOA



Professor Avaliador

Péricles Guedes Alves, Doutor, UniFOA

## DEDICATÓRIA

A Deus, que nos criou e pelo fôlego de vida.  
Aos familiares e professores que foram  
nosso apoio nesta jornada.

## **AGRADECIMENTOS**

Agradecemos primeiramente a Deus por ter nos guiado nessa difícil jornada.

Agradecemos aos familiares que nos proporcionaram tamanho conhecimento para cumprir esta etapa da vida.

Agradecemos também a todos os professores envolvidos na formação educacional e profissional de grande valia.

## RESUMO

Os processos industriais contínuos de tiras de aço consistem basicamente em desenrolar a bobina no início do processo, passar por todas as etapas como solda, acumuladores, imersão, laminadores, inspeção e novamente são enroladas ao final. Esse percurso realizado nas etapas do processo pode chegar à grandes distâncias. Para sustentação das tiras na linha são empregados rolos, no qual se utiliza de sistemas de controle e monitoramento para garantir o perfeito posicionamento do produto em sua superfície.

É condição indispensável para a qualidade do produto manter as tiras na posição central dos rolos. Nesse contexto este trabalho apresenta uma proposta de sistema de monitoramento e centralização de tiras por meio de visão computacional. O monitoramento consiste na utilização de câmera integrada a um computador por meio de rede de comunicação TCP/IP.

Desenvolveu-se um software capaz de atender as necessidades de centralização de objetos, a fim de solucionar o problema de descentralização de tiras dos processos siderúrgicos, mas também podendo ser aplicado a qualquer outro processo industrial com o mesmo objetivo.

A estrutura do sistema é constituída de um computador hospedeiro com software programável em Python, câmera IP, DAQ (dispositivo de aquisição de dados) com o qual se realiza o gerenciamento dos sinais.

A validação do sistema é realizada por meio de testes em software previstos na proposta. Os resultados satisfatórios observados nos testes realizados com o software mostram que essa proposta é capaz de atender a finalidade a que se destina.

**Palavras-chave:** Processo contínuo, Python, Centralização de Tiras, Visão Computacional.

## SUMÁRIO

1	INTRODUÇÃO .....	12
1.1	Justificativa.....	12
1.2	Objetivos Gerais.....	13
1.3	Objetivos Específicos .....	13
1.4	Metodologia.....	13
2	FUNDAMENTOS TEÓRICOS .....	14
2.1	Visão computacional .....	14
2.2	Processamento digital de imagens.....	15
2.3	Aquisição de Imagens .....	16
2.3.1	Sensor Único.....	17
2.3.2	Sensores por varredura de linha .....	18
2.3.3	Sensores matriciais .....	19
2.4	Amostragem e quantização de imagens .....	19
2.5	Pixel .....	22
2.6	Pré-processamento .....	24
2.7	Espaço de cor .....	25
2.8	Histograma .....	26
2.9	Calibração da câmera .....	27
2.10	Segmentação .....	30
2.11	Detecção de borda .....	30
2.12	Operadores .....	32
2.12.1	Operador Sobel .....	33
2.13	Limiarização .....	35
2.14	Controle de Processos .....	36

2.15	Controle de Bordas .....	37
2.16	Controle de posição pelas bordas .....	38
2.17	Controle de posição pelo centro .....	39
3	DESENVOLVIMENTO.....	41
3.1	Método Proposto .....	41
3.2	Solução proposta .....	41
3.3	Ambiente de Desenvolvimento.....	43
3.3.1	Python .....	43
3.3.2	OpenCV.....	43
3.4	Software .....	43
3.4.1	Captura de imagem .....	43
3.4.2	Processos para calibração de câmera .....	44
3.4.3	Processos para detecção de bordas e limiarização .....	45
3.4.4	Representação Final .....	47
4	RESULTADOS EXPERIMENTAIS .....	49
4.1	Considerações .....	49
4.2	Experimento para Calibração de Câmera .....	49
4.3	Resultado da Calibração de Câmera .....	50
4.4	Experimento para Centralização de Tira .....	52
4.5	Resultado da Centralização de Tiras .....	55
5	CONCLUSÃO.....	58
6	REFERÊNCIAS BIBLIOGRÁFICAS .....	59

## LISTA DE ABREVIATURAS E SIGLAS

AC – Alternative Current (Corrente Alternada)

CCD – Charge Coupled Device (Dispositivo Acoplado de Carga)

CMOS – Complementary Metal Oxide Semiconductor (Semicondutor Complementar de Óxido Metálico)

CPC – Center Position Control (Controle de Posição Central)

CPU – Central Processing Unit (Unidade Central de Processamento)

DAQ – Data Acquisition (Aquisição de Dados)

DC – Direct Current (Corrente Contínua)

EPC – Edge Position Control (Controle de Posição de Borda)

MPM – Metros por minuto

PX – Pixel

RGB – Espaço de Cores (Vermelho, Amarelo, Azul)

TCP/IP – Protocolo de Comunicação

## LISTA DE FIGURAS

Figura 1 - Visão Humana e Visão computacional.....	14
Figura 2 - Passos fundamentais em processamento digital de imagens.....	15
Figura 3 - Representação dos Eixos .....	16
Figura 4- Estrutura Física Fotodiodo. ....	17
Figura 5 - Combinação de único sensor com movimento. ....	18
Figura 6 -Arranjo Linear de Sensores. ....	18
Figura 7 - Matriz de Sensores. ....	19
Figura 8 – Produção imagem digital (a) Imagem contínua. (b) Amostragem e quantização.....	20
Figura 9 – Resultados da amostragem e quantização. ....	21
Figura 10 - (a) Representação gráfica da imagem. (b) Representação como matriz de intensidade. (c) Representação como uma matriz numérica. ....	22
Figura 11 - Vizinhança Horizontal e Vertical.....	23
Figura 12 - Vizinhança Diagonal. ....	23
Figura 13 - Vizinhança de 8 pixels. ....	23
Figura 14 - Coordenadas para o Computador.....	24
Figura 15 – Bandas de cor no modelo RGB.....	25
Figura 16 - Formação de cores pelo espaço de cor RGB. ....	26
Figura 17 – Valores que compreendem uma imagem em tons de Cinza. ....	26
Figura 18 - (a) Imagem digital. (b) Histograma.....	27
Figura 19 - Imagem com distorção radial. ....	28
Figura 20 - Imagem sem distorção.....	28
Figura 21- Vetor Gradiente.....	31
Figura 22- (a) Imagens com descontinuidades. (b) Sinal da descontinuidade. (c) Primeira derivada do sinal. (d) Segunda derivada do sinal. ....	32
Figura 23- Máscaras utilizadas no operador de Sobel. ....	33
Figura 24 - Magnitude dos gradientes e ângulos das bordas.....	34
Figura 25 - Processo realizado pelo operador Sobel. ....	34
Figura 26–Imagem antes da Limiarização. ....	35
Figura 27 - Imagem Depois da Limiarização.....	35
Figura 28-(a) Sistema de malha aberta. (b) Sistema de malha fechada. ....	37
Figura 29 - Representação do controle do sistema de centralização pelas bordas. .	38

Figura 30 - Representação do controle do sistema de centralização pelo centro .....	40
Figura 31 – Diagrama de bloco do controle do sistema de centralização. ....	41
Figura 32 - Componentes que forma o sistema de centralização. ....	42
Figura 33 - Sistema para aplicação do projeto. ....	42
Figura 34 - Representação dos cantos extraídos.....	44
Figura 35 - Imagem depois da Detecção de bordas.....	45
Figura 36 - Imagem antes e depois da detecção de bordas e limiarização.....	46
Figura 37 – Histograma para centralização.....	46
Figura 38 - Fluxograma do Algoritmo de centralização. ....	47
Figura 39 - Algoritmo em Python de centralização.....	47
Figura 40 - Representação final gerada pelo software.....	48
Figura 41 – Imagens usadas para calibração experimental. ....	50
Figura 42 – Imagens com marcação de cantos.....	51
Figura 43 – Imagens sem distorções da câmera.....	52
Figura 44 - Protótipo montado para testes. ....	53
Figura 45 – Tira usada para os testes.....	53
Figura 46 – Sequência e imagens usadas para os testes de centralização.....	54
Figura 47 – Sequência e imagens processadas nos testes de centralização. ....	55

## LISTA DE TABELAS

Tabela 1 - Resultados do algoritmo de centralização.....	56
Tabela 2 – Resultados do cálculo de largura .....	56
Tabela 3 – Tempos de resposta.....	57

## 1. INTRODUÇÃO

Mediante o avanço das tecnologias pertinentes à automação, desencadeou-se assim o estudo e a procura por métodos alternativos de controle que sejam mais eficazes quanto: a flexibilidade, qualidade e produção em função do tempo. Os processos automatizados propiciam maior segurança, pois reduzem a necessidade da intervenção humana, uma vez que a sua estrutura dispõe de métodos mecânicos, elétricos e computacionais.

A laminação é um dos processos que se encontra bastante automatizado. A automação neste tipo de processo está relacionada à precisão do acabamento das chapas e placas metálicas, para isso é preciso monitorar todo o processo de modo que o produto final atenda às exigências requisitadas pelos usuários de cada produto. Desencadeando assim a imprescindibilidade da criação de sistemas inteligentes capazes de adquirir informações, supervisioná-las e apresentar respostas a diferentes situações pré-programadas.

A visão computacional associada a este tipo de processo proporciona uma maior versatilidade, pois sistemas computacionais não estão sujeitos ao cansaço, ou seja, apontam falhas que poderiam passar imperceptíveis ao olhar de um operário, tornando assim o processo mais seguro e de boa produtividade.

### 1.1 Justificativa

Um problema visto frequentemente em processos de laminação é o deslocamento irregular das tiras de aço laminado, que por alguma adversidade desviam de seu percurso original, acarretando prejuízos assim como a interrupção da linha de produção. O projeto tem como finalidade a correta centralização de tiras para que o material não se desloque nos processos de laminação vindo a bater na estrutura causando um amassado e danificando a sua estrutura. Sendo assim, destina-se à criação de uma programação capaz de adquirir e analisar imagens em tempo real para a inspeção e controle mediante um processo contínuo cujo trabalho exige um monitoramento constante.

## 1.2 Objetivos Gerais

Apresentar um método alternativo de controle que traga confiabilidade, segurança e disponibilidade, realçando a importância da visão computacional em processos industriais com os quais é necessária a centralização de objetos a fim de atender diferentes necessidades.

Sendo assim, é de fundamental importância a pesquisa por tecnologias alternativas que sejam capazes de reduzir e/ou eliminar adversidades que venham a ocasionar falhas em processos industriais.

## 1.3 Objetivos Específicos

Aplicação de conhecimentos em programação para solucionar problemas relacionados à centralização de tiras através do desenvolvimento de um software que seja útil e aplicável às indústrias destinadas à laminação de aço, tendo em vista o aprimoramento da qualidade de aços laminados diminuindo as perdas metálicas no descarte de ponta e cauda das bobinas de aço ocasionadas por amassados e diferentes alongamentos de fibras.

Atualmente os sistemas de centralização utilizam sensores fotoelétricos que necessitam de inspeção periodicamente para que garantam a integridade dos processos, a proposta em si consiste na eliminação de tais sensores, atribuindo a sua função a câmera que ficará responsável por inspecionar o processo.

## 1.4 Metodologia

Com o intuito de efetivar esse trabalho dispomos dos seguintes métodos:

- Pesquisa bibliográfica sobre o tema;
- Testes e simulações utilizando software programável;
- Apresentação de um método alternativo de realimentação;
- Levantamento de equipamentos necessários para o correto funcionamento;
- Desenvolvimento de software em Python para controle.

## 2 FUNDAMENTOS TEÓRICOS

### 2.1 Visão computacional

A visão computacional visa representar a visão humana usando software e hardware. Sua função é reconstruir, interromper e compreender uma cena em 3D a partir de imagens que provêm em plano 2D. Pode-se dizer que a visão computacional é muito mais que apenas ver, pois além de ver o processo envolve a percepção junto com muita análise. Em resumo a visão computacional é o estudo da extração de dados e informações de imagens (BALLARD & BROWN, 1982).

O reconhecimento de imagens e seu processamento integram de forma geral o processo de visão computacional, com estes integrantes é possível melhorar e adequar as imagens para os algoritmos de visão computacional.

A visão biológica é análoga a visão computacional, enquanto uma estuda os processos do cérebro para formação e percepção de imagens a outra estuda os algoritmos que fazem a máquina enxergar. Para ser eficaz a visão computacional se relaciona com a física no que rege a ótica e lentes, que são os olhos do projeto. Outro campo também relacionado é o processamento de sinais, processamento de imagens, visão de máquina e outros.

A Figura 1 faz um comparativo entre a visão humana e a computacional. Pode-se dizer que a visão computacional vai além do processamento de imagens, pois esta é responsável pela obtenção e a manipulação dos dados de uma imagem, sendo elas processadas ou não, para diferentes fins.



Figura 1 - Visão Humana e Visão computacional

Fonte: Autores, 2018.

## 2.2 Processamento digital de imagens

O processamento digital de imagens nasceu da necessidade de se realizar a transmissão e impressão de imagens (GONZALES & WOODS, 2010). Essa necessidade tem como objetivo a melhoria da imagem e a retirada de defeitos ou imperfeições não necessárias ou que atrapalhariam a interpretação da imagem.

Processar uma imagem significa aplicar diversos métodos computacionais visando à melhoria de sua qualidade em uma transmissão e para se extrair a informação presente nela.

As principais etapas para o processamento de imagens são:

- Aquisição de imagens: Obter imagem digital;
- Pré-processamento: Usar filtros e técnicas para melhorar a imagem;
- Segmentação: Isolar partes significativas da imagem;
- Representação e descrição: Representar os dados extraídos;
- Reconhecimento e interpretação: Interpretar os dados extraídos com base em um conhecimento prévio;
- Base de conhecimento: Dados iniciais necessários para o funcionamento de todos os outros processos, dados de referência.

A Figura 2 representa o fluxo dos processos.

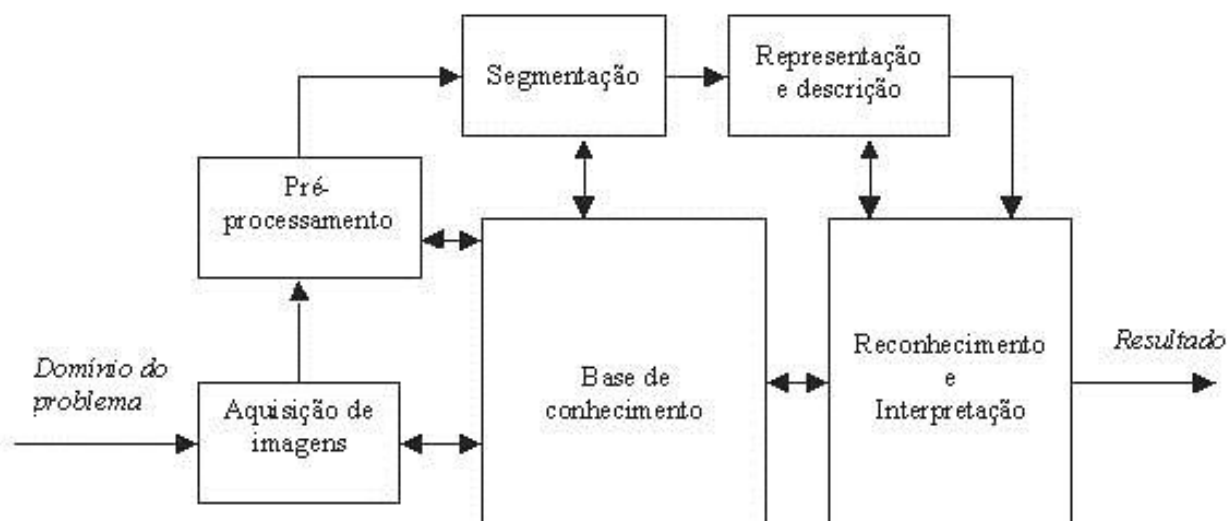


Figura 2 - Passos fundamentais em processamento digital de imagens.

Fonte: Adaptado de Gonzales e Woods, 2010.

Uma imagem pode ser representada como uma função bidimensional de intensidade da luz,  $f(x, y)$ . Onde  $x$  e  $y$  são componentes espaciais e  $f(x, y)$  o brilho da imagem naquele ponto conforme Figura 3. Uma imagem digital, conseqüentemente, pode ser representada por uma função  $f(x, y)$ , em coordenadas espaciais e em brilho. Outra forma de representar uma imagem é uma matriz de “pixels” (GONZALES & WOODS, 2010).



Figura 3 - Representação dos Eixos

Fonte: Autores, 2018

### 2.3 Aquisição de Imagens

A aquisição de imagem pode ser considerada o primeiro passo, para isso é necessário de um elemento sensor para captura da imagem e digitalização da mesma. Normalmente este sensor é uma câmera que envia um sinal digital, caso esse sinal seja analógico, é necessário um dispositivo para sua conversão.

O princípio de digitalização de imagem se dá por meio da captação de diferentes intensidades luminosas refletidas ou transmitidas por objetos iluminados que por sua vez geram sinais elétricos de diferentes amplitudes.

Existem três meios de aquisição de imagem:

- Aquisição de imagens utilizando um único sensor;
- Aquisição de imagens utilizando sensores por varredura de linha;
- Aquisição de imagens utilizando sensores matriciais.

### 2.3.1 Sensor Único

O sistema baseado em elementos semicondutores como o fotodiodo, mostrado na Figura 4, produz um sinal elétrico cuja forma de onda da tensão é proporcional a intensidade luminosa que está exposta, onde é possível aplicar filtros a entrada do elemento sensor a fim de favorecer específicos tipos de bandas como o RGB.

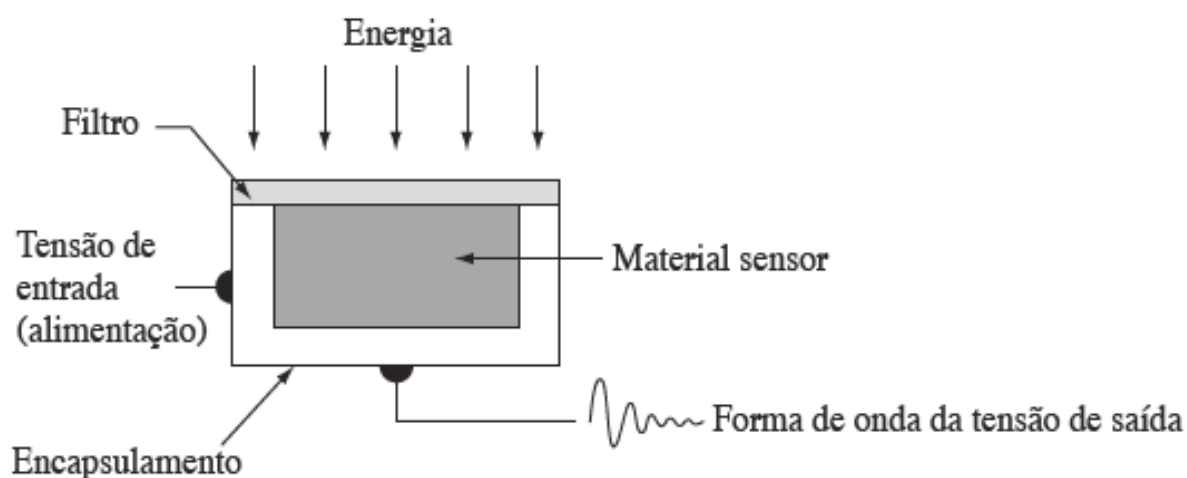


Figura 4 - Estrutura Física Fotodiodo.

Fonte: Gonzales e Woods, 2010.

Para gerar uma imagem bidimensional utilizando um único sensor é necessário que haja transições tanto no eixo X quanto no eixo Y entre o sensor e a área de imagem, representado pela Figura 5. Este modelo pode conter um alto nível de precisão, apesar de ser lento apresenta alta resolução.

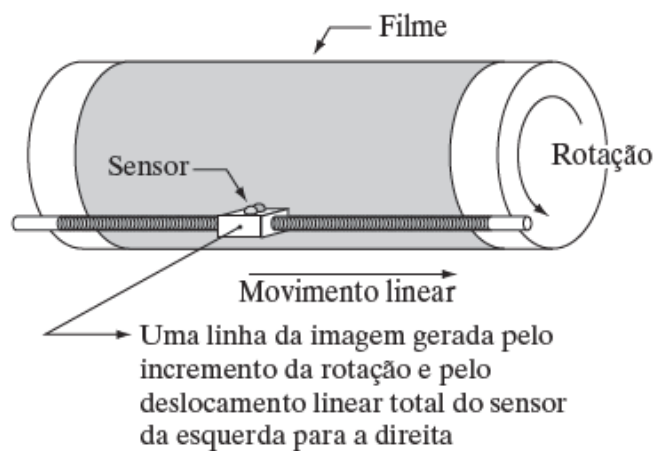


Figura 5 - Combinação de único sensor com movimento.

Fonte: Gonzales e Woods, 2010.

### 2.3.2 Sensores por varredura de linha

Segundo Gonzales e Woods (2010, p.31), utiliza de uma disposição geométrica que “consiste em um arranjo linear de sensores na forma de uma faixa de sensores. O arranjo linear dos sensores fornece elementos para a aquisição de imagens em uma direção, movimentos de direções oposta geram imagens na outra direção”, conforme a Figura 6.

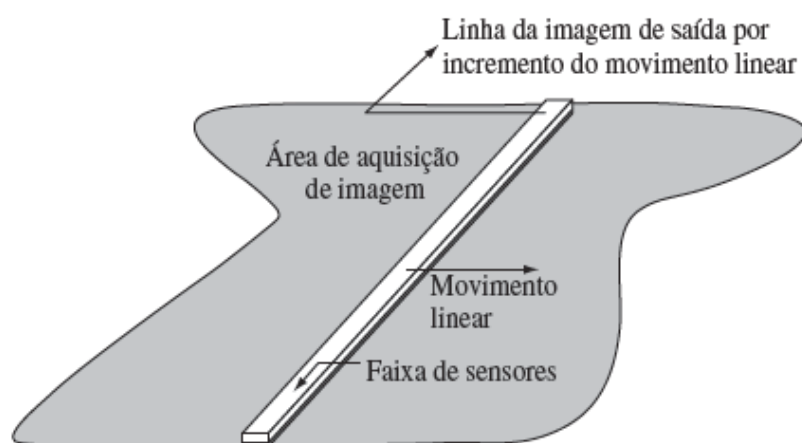


Figura 6 - Arranjo Linear de Sensores.

Fonte: Gonzales e Woods, 2010.

### 2.3.3 Sensores matriciais

São sensores dispostos de modo a formar uma matriz bidimensional, representados pela Figura 7, muito utilizado em câmeras digitais de modelo CCD que possuem diversas variedades de disposição de arranjos matriciais como, por exemplo, 4000 x 4000 elementos ou mais. Cada sensor corresponde à integral da energia luminosa projetada sobre ele, produzindo assim uma imagem com baixo nível de ruídos. Deste modo não há necessidade de movimentações dos sensores para aquisição de uma imagem completa.

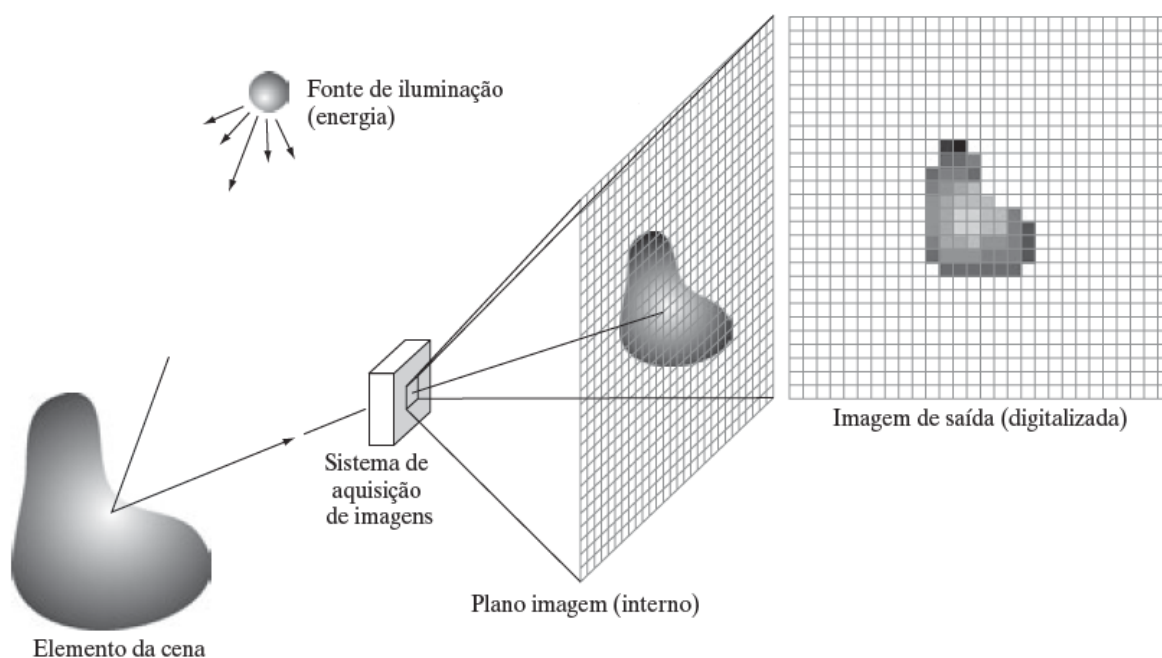


Figura 7 - Matriz de Sensores.

Fonte: Gonzales e Woods, 2010.

## 2.4 Amostragem e quantização de imagens

O objetivo é sempre gerar imagens digitais a partir de dados captados por sensores (GONZALES & WOODS, 2010). Como muitos dos sensores são analógicos, para a transformação em imagem digital existem dois processos: amostragem e quantização.

A amostragem se dá através da digitalização das coordenadas em relação à  $x$ , já a quantização está relacionada à amplitude, ou seja, a intensidade do sinal luminoso que é representada por um número inteiro entre 0 e 255 em relação ao eixo  $y$ . A

amostragem utilizando sensores matriciais não lida com movimento, portanto os limites de amostragem nas duas direções são determinados pelo número de sensores que compõem a matriz.

A Figura 8 (a) representa uma imagem para ser transformada em imagem digital. São colhidas amostras igualmente espaçadas pela reta AB que se define pelo processo de amostragem. Com isso a Figura 8 (b) representa no eixo y a intensidade (quantização) de cada ponto amostrado com escala de intensidade dividida em intervalos de preto a branco.

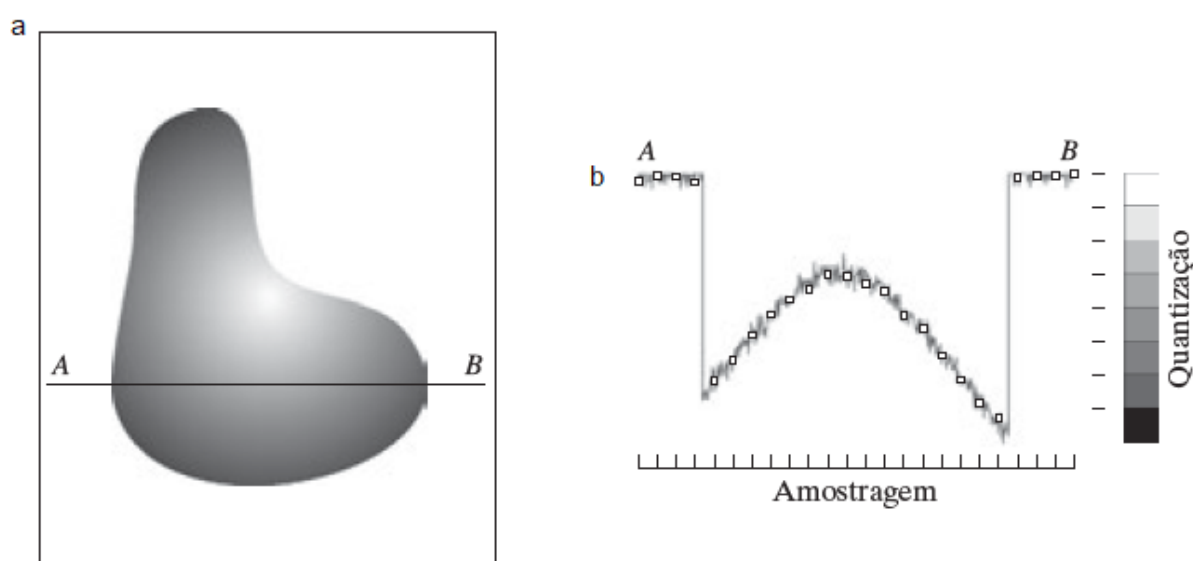


Figura 8 – Produção imagem digital (a) Imagem contínua. (b) Amostragem e quantização.

Fonte: Adaptado de Gonzales e Woods, 2010.

Segundo Gonzales e Woods (2010, p.35), a qualidade da imagem digital é claramente determinada, em grande parte, pelo número de amostras e de níveis discretos de intensidade utilizados na amostragem e na quantização. A Figura 9 mostra uma aquisição de imagem por uma matriz de sensor e o resultado da amostragem e quantização.

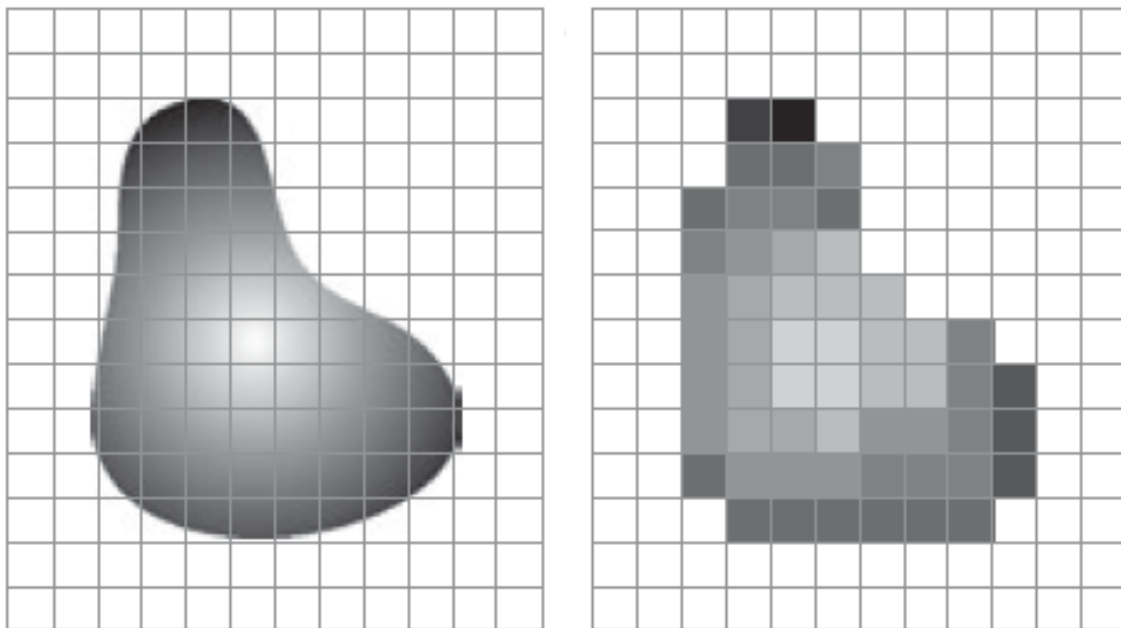


Figura 9 – Resultados da amostragem e quantização.

Fonte: Gonzales e Woods, 2010.

Uma imagem digital pode ser representada de três formas básicas. A Figura 10(a) representa a imagem digital com dois eixos espaciais e um terceiro de intensidade. A Figura 10(b) representa uma imagem semelhante a uma fotografia em que o nível de cinza é proporcional a intensidade do ponto. A terceira representação, a Figura 10(c), é a representação numérica dos valores na forma de matriz.

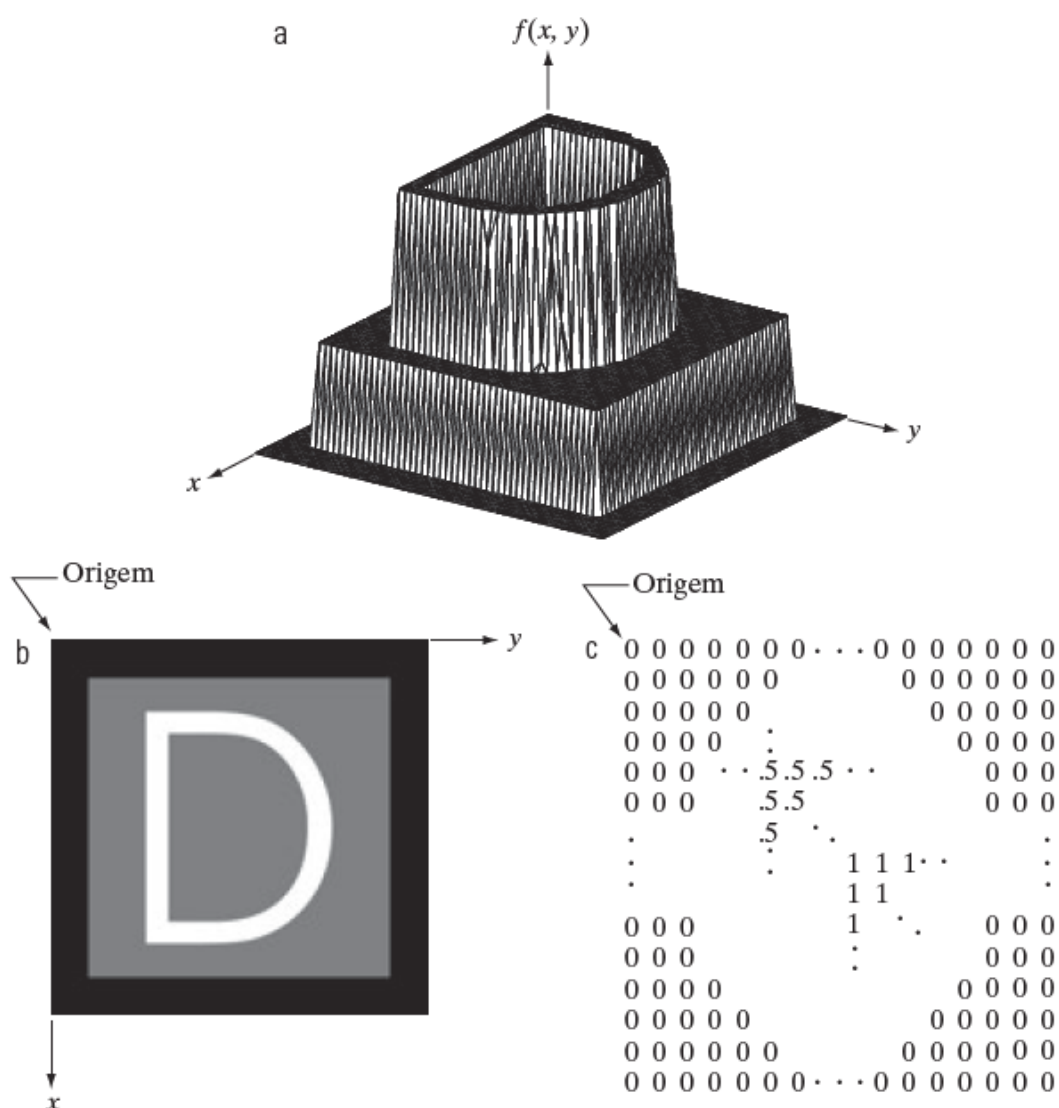


Figura 10 - (a) Representação gráfica da imagem. (b) Representação como matriz de intensidade. (c) Representação como uma matriz numérica.

Fonte: Gonzales e Woods, 2010.

## 2.5 Pixel

Pixel (P) é a menor parte de uma imagem que em conjunto constituem a imagem digital, existem diversas relações entre pixels comumente tratadas como vizinhança (V). Um pixel de coordenada  $(x, y)$  possui 4 vizinhos, 2 na vertical e 2 na horizontal, onde é possível observar que os pixels localizados na borda possuem um número menor de vizinhos observados na Figura 11.

	V	
V	P	V
	V	

Figura 11 - Vizinha Horizontal e Vertical.

Fonte: Autores, 2018.

É possível similarmente estabelecer relações de vizinhança (V) com os pixels localizados na diagonal, representados na Figura 12.

V		V
	P	
V		V

Figura 12 - Vizinha Diagonal.

Fonte: Autores, 2018.

A junção destes dois modelos gera então a vizinha de 8 pixels como pode ser visto na Figura 13.

	<b>y</b>		
	P(x-1,y-1)	P(x,y-1)	P(x+1,y-1)
<b>x</b>	P(x-1,y)	P(x,y)	P(x+1,y)
	P(x-1,y+1)	P(x,y+1)	P(x+1,y+1)

Figura 13 - Vizinha de 8 pixels.

Fonte: Adaptado de Gonzales e Woods, 2010.

O modelo se baseia no plano cartesiano, porém em computadores o plano cartesiano é ligeiramente diferente, pois o y desenvolve-se positivamente de cima para baixo como mostra a Figura 14.

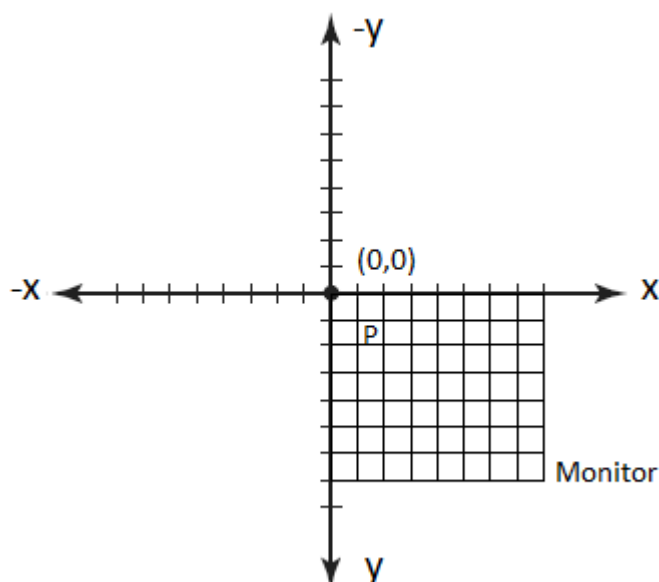


Figura 14 - Coordenadas para o Computador.

Fonte: Autores, 2018.

## 2.6 Pré-processamento

O processamento de imagens comumente é feito via software, exceto em casos de necessidade de rápido processamento que é utilizado hardware especial.

Sua principal função é melhorar a imagem para que ela possa ser mais bem aproveitada nas etapas seguintes. O pré-processamento envolve filtros, como para remoção de ruídos, ou para isolar áreas desejadas.

Segundo Gonzales e Woods (2010, p.19), “O software para o processamento de imagens consiste em módulos especializados que realizam tarefas específicas. Um bom pacote computacional também inclui a possibilidade de o usuário escrever códigos que no mínimo utilizem os módulos especializados. Os pacotes de aplicativos mais sofisticados permitem a integração desses módulos e dos comandos gerais de software a partir de pelo menos uma linguagem computacional”.

Outro fator de extrema importância está relacionado ao armazenamento das imagens, que podem ser de curto prazo, online e em arquivo para acesso.

O armazenamento de curto prazo utiliza a memória RAM do computador, ou seja, memória de execução temporária com a qual não se retém as informações, mas as disponibiliza para os processos quando requisitadas. Outra solução alternativa se aplica no uso de placas de vídeo especializadas que através de frame buffers armazenam diversas imagens que podem ser acessadas rapidamente.

Em processos que haja a necessidade de acesso frequente as informações utiliza-se a memória on-line que permite o acesso às informações que ficam retidas em disco magnético ou mídias ópticas.

O armazenamento em arquivo destina-se ao atendimento às demandas de armazenamento em massa e de baixa frequência de acesso.

## 2.7 Espaço de cor

Um espaço de cor é uma organização das cores representado por um modelo matemático que descreve a forma como as cores podem ser representadas. Com isso foram gerados vários padrões para facilitar, sendo amplamente aceito.

O modelo RGB tem suas cores denominadas como bandas, sendo assim, constituído de 3 bandas que juntas compõem a imagem colorida. Cada pixel de banda possui 3 bytes totalizando uma profundidade de 24 bits por pixel, que pode ser vista na Figura 15. Existem também outros modelos de cores, cada um baseado em um tipo de aplicação afim de atender a diferentes necessidades.

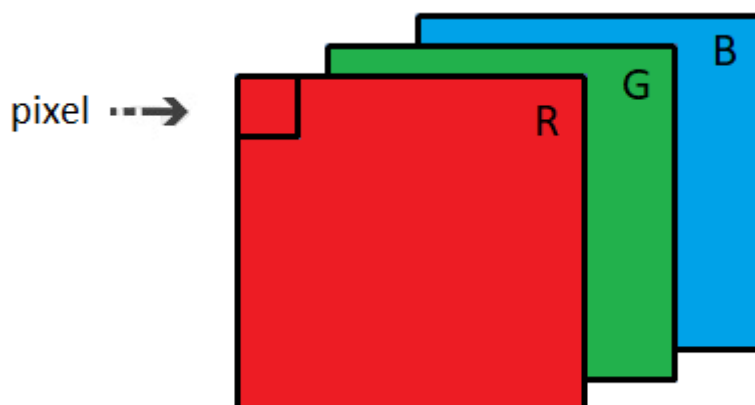


Figura 15 – Bandas de cor no modelo RGB.  
Fonte: Autores, 2018.

Os modelos mais utilizados são: RGB (vermelho, verde e azul), CMY (ciano, magenta e amarelo) e o HSI (matriz, saturação e brilho). A Figura 16 mostra o exemplo do modelo RGB.

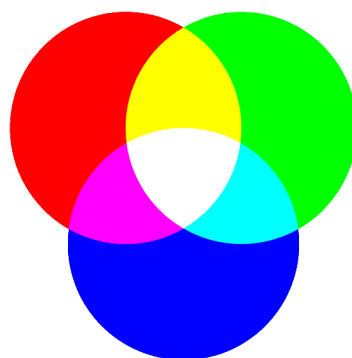


Figura 16 - Formação de cores pelo espaço de cor RGB.

Fonte: Autores, 2018.

É possível converter o modelo de cores RGB em tons de cinza, onde cada cor em tom de cinza tem valores iguais das componentes: vermelho, verde e azul. Este modelo se baseia em apenas um componente (iluminação) e compreende valores entre 0 e 255 como mostra a Figura 17.



Figura 17 – Valores que compreendem uma imagem em tons de Cinza.

Fonte: Autores, 2018.

## 2.8 Histograma

O histograma consiste em um gráfico cartesiano de frequência, que tem por objetivo ilustrar a frequência que cada tonalidade de cinza ocorre em uma imagem. Com um baixo custo de processamento computacional, o histograma é muito utilizado. A Figura 18 mostra como é um histograma.

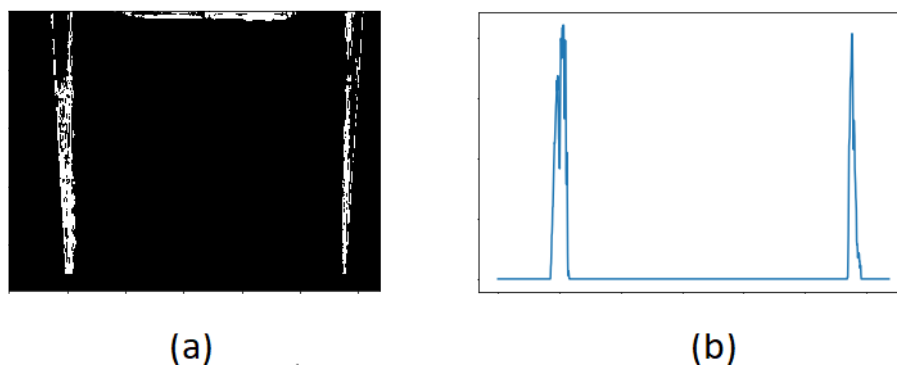


Figura 18 - (a) Imagem digital. (b) Histograma.

Fonte: Autores, 2018.

## 2.9 Calibração da câmera

A calibração de câmeras é essencial para aplicação de visão computacional em que informações métricas são necessárias. Essa calibração busca dados como distância focal, real tamanho dos pixels, ponto central de projeção e a inclinação entre os eixos de projeções. A calibração é feita com a inserção de dados de pontos conhecidos no ambiente e relacionando as projeções dentro do plano da imagem. Um dos métodos de calibração é pelo modelo de Pinhole (OPENCV, 2018).

Os principais tipos de distorção são a distorção radial e a distorção tangencial. A distorção radial faz com que as linhas retas pareçam curvas. Por exemplo, na Figura 19, na qual duas arestas de um tabuleiro de xadrez são marcadas com linhas amarela, pode-se notar que a borda do tabuleiro de xadrez não é uma linha reta e não é paralela à linha amarela. Todas as linhas retas esperadas estão salientes. A Figura 20 apresenta o resultado da imagem sem distorção.

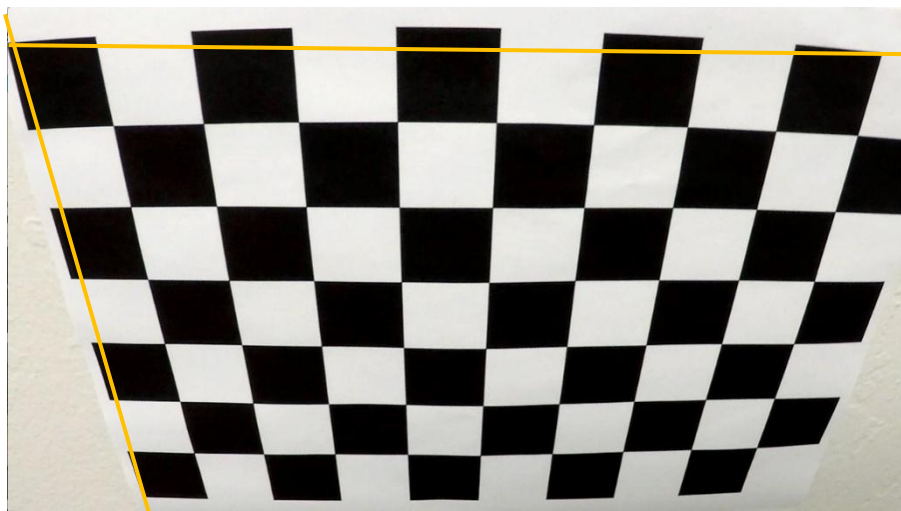


Figura 19 - Imagem com distorção radial.

Fonte: Autores, 2018.



Figura 20 - Imagem sem distorção

Fonte: Autores, 2018

A distorção radial pode ser representada conforme Equação (2.1) e (2.2):

$$x_{distorcido} = x(1 + k_1r^2 + k_2r^4 + k_3r^6) \quad (2.1)$$

$$y_{distorcido} = y(1 + k_1r^2 + k_2r^4 + k_3r^6) \quad (2.2)$$

Da mesma forma, a distorção tangencial ocorre porque a lente de captura de imagem não está alinhada perfeitamente paralela ao plano de imagem. Portanto, algumas áreas da imagem podem estar mais próximas do que o esperado (OPENCV, 2018). A quantidade de distorção tangencial pode ser representada como abaixo:

$$x_{distorcido} = x + [p_2(r^2 + 2x^2) + 2p_1xy] \quad (2.3)$$

$$y_{distorcido} = y + [p_1(r^2 + 2y^2) + 2p_2xy] \quad (2.4)$$

Em suma, há a necessidade de encontrar cinco parâmetros, conhecidos como coeficientes de distorção dados por:

$$\text{Coeficientes de Distorção} = (k_1 k_2 p_1 p_2 k_3) \quad (2.5)$$

Além disso, precisam-se de algumas outras informações, como os parâmetros intrínsecos e extrínsecos da câmera. Parâmetros intrínsecos são específicos de cada câmera. Eles incluem informações como distância focal ( $f_x, f_y$ ) e centros ópticos ( $c_x, c_y$ ). A distância focal e os centros ópticos podem ser usados para criar uma matriz de câmera, que pode ser usada para remover a distorção devido às lentes de uma câmera específica. A matriz da câmera é exclusiva para uma câmera específica, portanto, uma vez calculada, ela pode ser reutilizada em outras imagens tiradas pela mesma câmera. É expresso como uma matriz 3x3:

$$\text{Matriz da câmera} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \quad (2.6)$$

Para aplicativos estéreos, essas distorções precisam ser corrigidas primeiramente. Para encontrar esses parâmetros, devemos fornecer algumas imagens de amostra de um padrão bem definido, por exemplo, um tabuleiro de xadrez. Encontramos alguns pontos específicos dos quais já sabemos as posições relativas, como cantos quadrados no tabuleiro de xadrez. Conhecemos as coordenadas desses pontos no espaço do mundo real e sabemos as coordenadas na imagem, para que possamos resolver os coeficientes de distorção. Para melhores resultados, precisamos de pelo menos 10 padrões de teste (OPENCV, 2018).

## 2.10 Segmentação

Fracionar ou segmentar a imagem é uma parte fundamental da visão computacional, neste ponto consegue-se isolar somente a informação buscada na imagem. Como uma imagem pode ter mais de um ponto de interesse, fraciona-se esta imagem a fim de tirar as partes desnecessárias e deixar a de interesse disponível para que cada algoritmo especializado possa analisá-la. Assim que detectada a região de interesse, a segmentação é interrompida.

Segundo Gonzales e Woods (2010, p454) a “segmentação de imagens não triviais é uma das tarefas mais difíceis no processamento de imagens. A precisão da segmentação determina o sucesso ou o fracasso final dos procedimentos de análise computadorizada. Por essa razão, deve-se tomar muito cuidado para aumentar a probabilidade de se obter uma segmentação precisa. Em algumas situações, como nas aplicações de controle industrial, é possível pelo menos alguma medida de controle sobre o ambiente”.

Em sua grande maioria os processos de segmentação destinam-se a propriedades básicas como descontinuidade e similaridade. A descontinuidade se dá por meio de diferentes valores de intensidade como ocorrido em bordas de objetos, já a similaridade está associada a características semelhantes referente a um conjunto de critérios.

## 2.11 Detecção de borda

Uma borda é o limite entre duas regiões com propriedades relativamente distintas de nível de cinza (GONZALES & WOODS, 2010). Essa técnica consiste em obter pontos onde a intensidade luminosa da imagem muda repentinamente de forma abrupta. Estas mudanças podem refletir diferentes objetos sobrepostos ou descontinuação de superfícies.

A detecção é feita por meio da obtenção do gradiente da imagem, requerendo ao cálculo de derivadas parciais na posição de cada pixel. A Figura 21(a) mostra a seção de uma imagem contendo um segmento de borda reto aonde se deseja obter a borda. O gradiente é dado pelo vetor descrito na equação (2.7).

$$\nabla f = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix} \quad (2.7)$$

O ângulo do vetor gradiente é dado por:

$$\alpha(x, y) = \tan^{-1}\left(\frac{G_y}{G_x}\right). \quad (2.8)$$

A Figura 21(b) ilustra o fato de a borda ser um ponto ortogonal ao sentido do vetor gradiente no ponto.

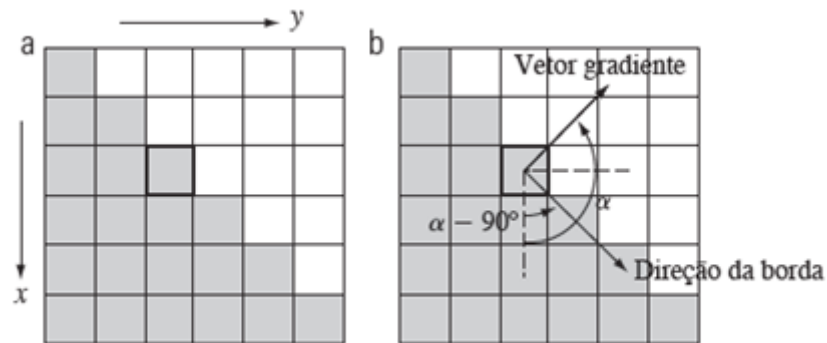


Figura 21 - Vetor Gradiente.

Fonte: Adaptado de Gonzales e Woods, 2010.

A Figura 22 ilustra a detecção de borda em uma imagem representada pelos diagramas de sinais que correspondem à borda.

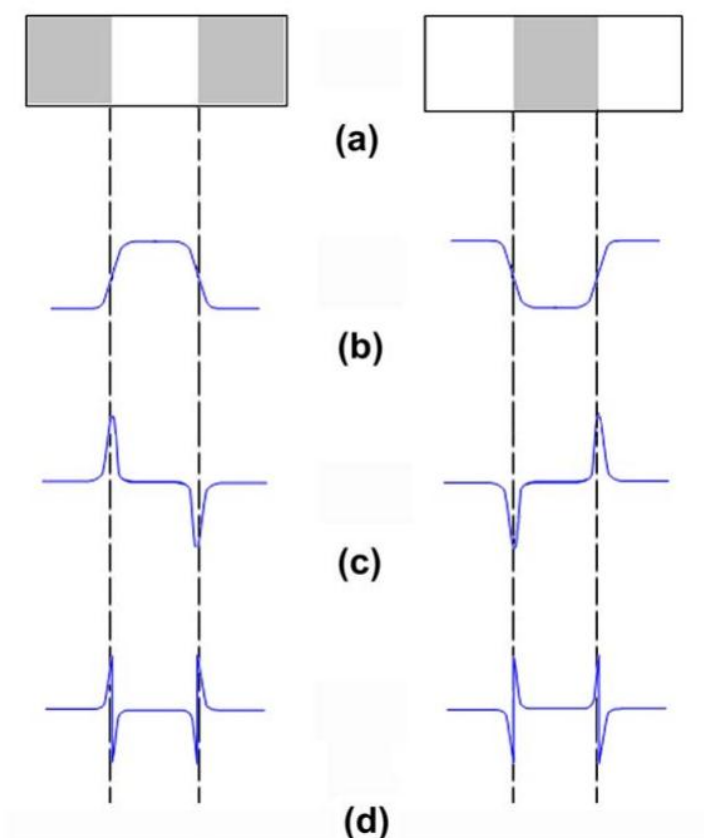


Figura 22 - (a) Imagens com discontinuidades. (b) Sinal da descontinuidade. (c) Primeira derivada do sinal. (d) Segunda derivada do sinal.

Fonte: TAKEDA, 2000

A primeira derivada apresenta o máximo e o mínimo ocorrido. A segunda derivada ilustra a passagem por zero exatamente na localização da borda. Na prática são utilizados operadores de forma a se obter uma aproximação da derivada utilizando diferenças.

## 2.12 Operadores

Existem vários modos de detectar bordas de uma imagem através do uso de filtros conhecidos como máscaras ou Kernels (Núcleo) de convolução. Estes operadores de convolução usam operadores de gradiente, que destacam as regiões com variações abruptas ou significativas em níveis de cinza (MORGAN, 2008). Existem vários operadores como os Prewitt, Roberts e Sobel. Este trabalho abordará somente o operador Sobel.

### 2.12.1 Operador Sobel

Sendo um dos mais sofisticados o operador Sobel processa independentemente cada pixel. Deste modo essa técnica determina se cada pixel presente na imagem pertence ou não a borda (MORGAN, 2008).

O operador de Sobel realça as linhas verticais e horizontais com intensidades diferentes do fundo. Para isso, duas máscaras de coeficientes,  $G_x$  e  $G_y$ , são utilizados. Como representado na Figura 23, os valores são arranjados em uma matriz de convolução de ordem 3x3.

-1	0	1
-2	0	2
-1	0	1

$G_x$

1	2	1
0	0	0
-1	-2	-1

$G_y$

Figura 23 - Máscaras utilizadas no operador de Sobel.

Fonte: TAKEDA, 2000.

O operador detecta as variações no sentido vertical e horizontal, podendo assim obter a magnitude e a direção da borda em cada pixel, como ilustra a Figura 24.

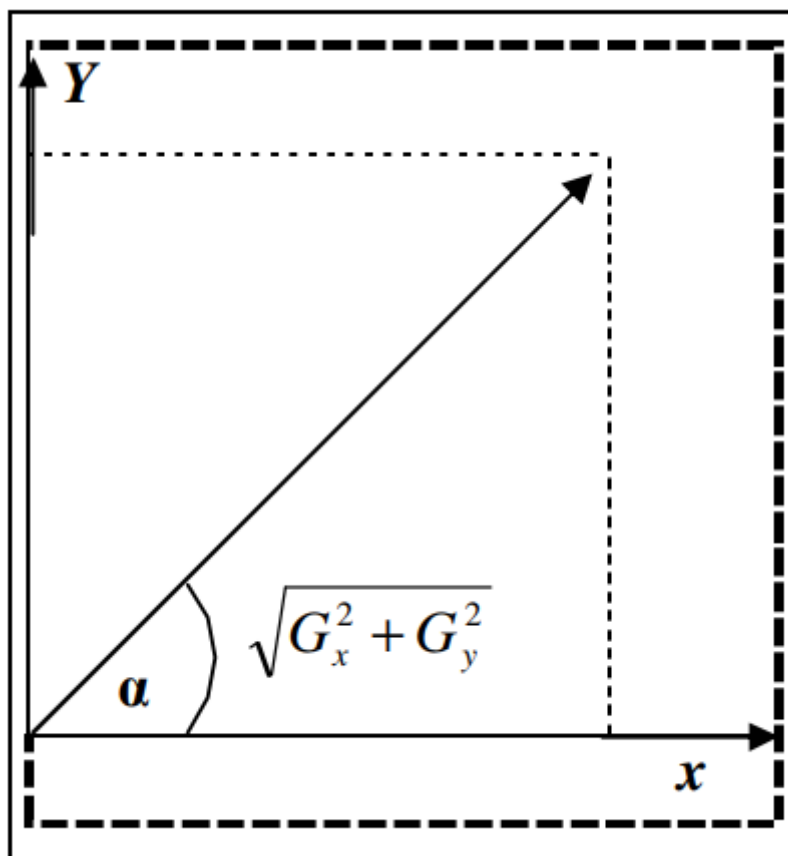


Figura 24 - Magnitude dos gradientes e ângulos das bordas.  
Fonte: MORGAN, 2008.

Segundo Fernandez (2004) a Figura 25 representa o processo que o operador de Sobel realiza sendo que  $dx$  o gradiente na vertical e  $dy$  o gradiente na horizontal.

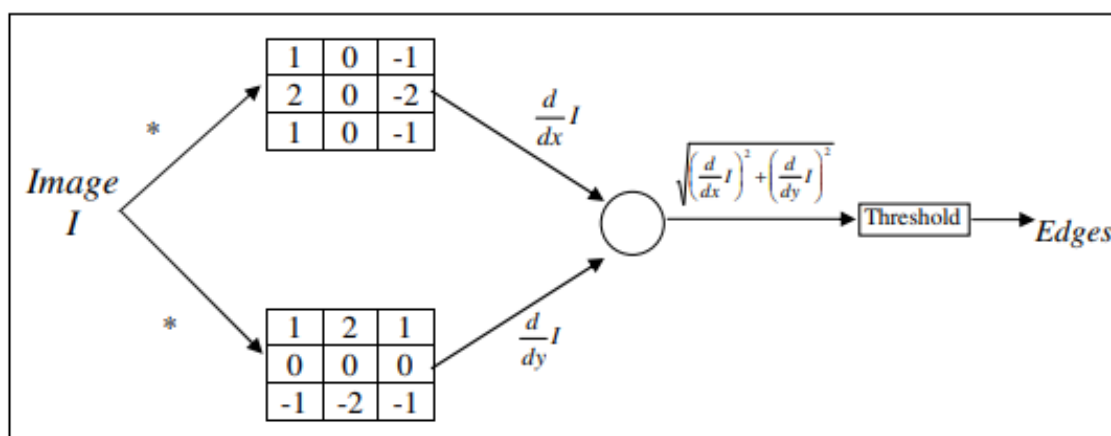


Figura 25 - Processo realizado pelo operador Sobel.  
Fonte: Fernandez, 2004.

### 2.13 Limiarização

Limiarização é um processo de segmentação que se baseia na diferença dos níveis de cinza que compõe os diferentes objetos de uma imagem, muito utilizado quando o interesse é destacar as bordas principais ou em manter a conectividade. Seus limites (L) são determinados de acordo com as características do material que se deseja isolar, podendo dividir-se em dois grupos: os dentro do limite e os fora do limite. A Figura 26 e a Figura 27 ilustram o processo de limiarização.

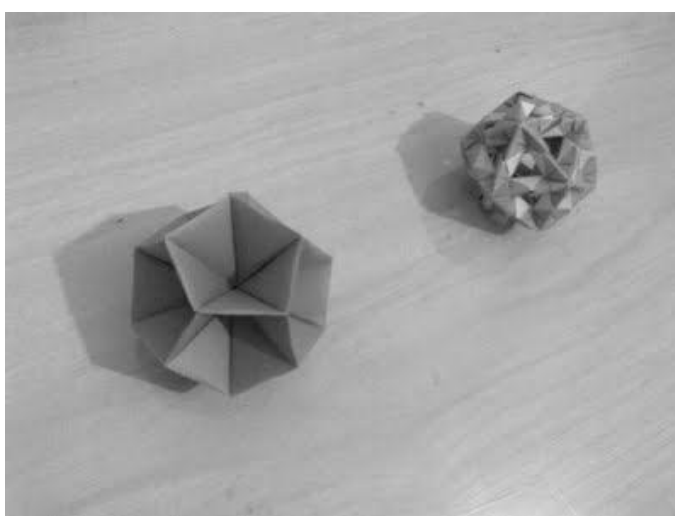


Figura 26 - Imagem antes da Limiarização.  
Fonte: Autores, 2018.

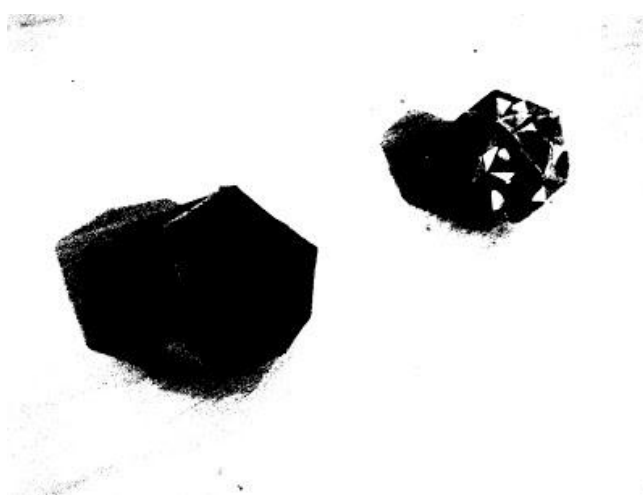


Figura 27 - Imagem Depois da Limiarização.  
Fonte: Autores, 2018.

Através de limiarização é possível obter melhorias no desempenho, pois esta técnica pode ser combinada com a detecção de bordas.

A limiarização segundo Gonzales e Woods (2010, p.77) “mapeia todas as intensidades entre 0 e 127 em 0 e todas as intensidades entre 128 e 255 em 1”. A equação que define o cálculo limiar é:

$$g(x, y) = \begin{cases} 1, & \text{se } f[x, y] > L \\ 0, & \text{caso contrário} \end{cases} \quad (2.9)$$

## 2.14 Controle de Processos

Em todos os processos industriais, as variáveis de processos estão sujeitas a variações com o tempo, como o nível de líquidos em tanques, pressão em uma tubulação e vazão de um compressor.

Essas variáveis físicas geram a necessidade da criação de sistemas de controle que atuem sobre uma variável ou condição do processo, a fim de aproximar-se de um estado estacionário. Os sistemas de controle podem ter diversas configurações, sendo as principais de malha aberta ou fechada (NISE, 2017).

O sistema de malha aberta, conforme Figura 28(a), funciona a partir de um valor de referência na entrada que se deseja obter na saída. Para tal processo, o controlador aciona um processo ou planta. No meio deste processo podem ser adicionados outros sinais como as perturbações, que podem servir de ajustes para o processo (NISE, 2017).

O sistema de malha fechada, conforme Figura 28(b), pode ser chamado de controle com realimentação, pois tem sensibilidade a perturbações e a habilidade de corrigir os seus efeitos. Isto se diferencia da malha aberta, pois contém um transdutor, ou sensor, que mede a resposta da saída de modo a indicar ao controlador a diferença entre o real e o desejado.

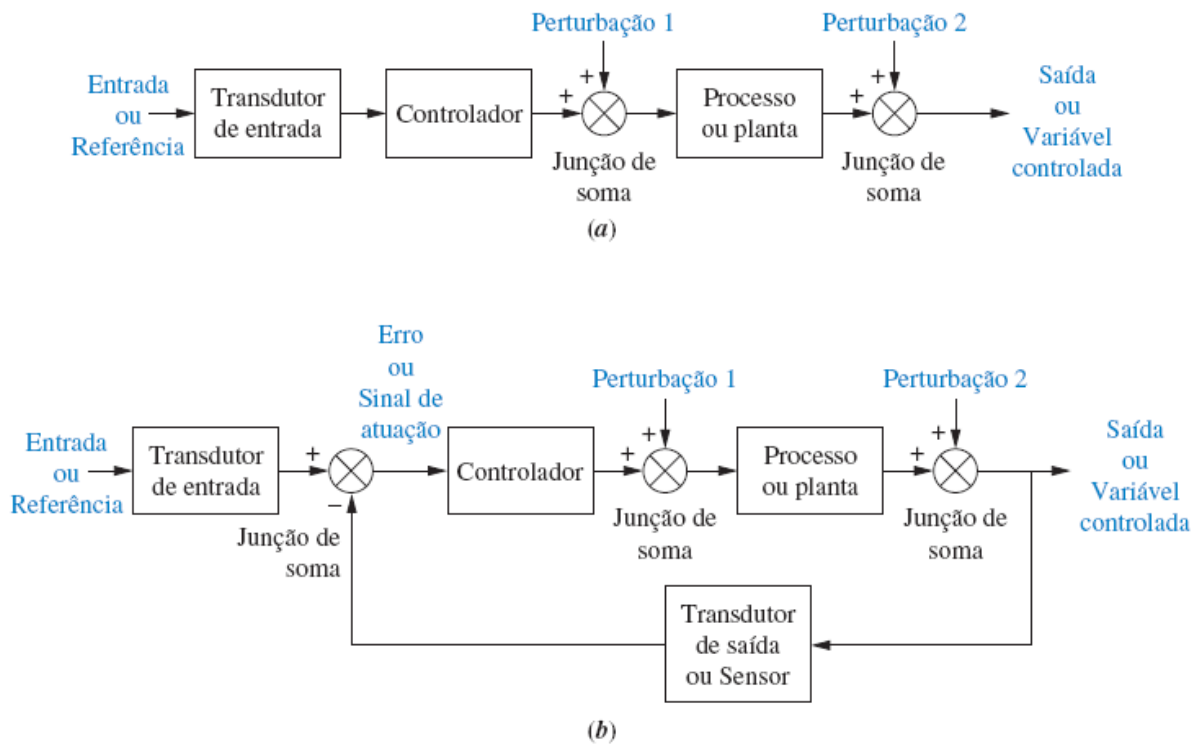


Figura 28 - (a) Sistema de malha aberta. (b) Sistema de malha fechada.

Fonte: NISE, 2017.

## 2.15 Controle de Bordas

O sistema de controle de borda visa centralizar objetos a fim de atender necessidades de diversos processos industriais. O sistema é de grande versatilidade, pois se adapta facilmente a novos ou processos existentes, tais como:

- Siderurgia;
- Indústria de metal leve;
- Indústria de rolamento de cobre;
- Indústria de impressão e encadernação;
- Indústria de plásticos e filmes;
- Indústria de borracha e química;
- Indústria de papel;
- Fibras e indústria têxtil.

A centralização é feita por meio de dois métodos que podem ser realizadas pelas bordas do objeto (EPC) ou então pelo centro do objeto (CPC).

## 2.16 Controle de posição pelas bordas

Sistema baseado no controle automático por meio de uma realimentação, que é responsável por transferir sinais de saída para a entrada do mesmo circuito (Malha fechada). Este método de controle é muito utilizado em linhas de produção contínua onde a velocidade de processamento é alta devido à necessidade de produzir em grande escala atendendo aos requisitos de produtividade

O princípio de funcionamento consiste em parametrizar uma distância lateral do objeto em relação ao sensoriamento. Uma vez que esse objeto se desloque da posição parametrizada, o controlador gera um sinal de correção para o atuador, deste modo mantendo sempre objeto dentro do valor parametrizado (NIRECO, 2007). A Figura 29 representa o sistema de controle pelas bordas.

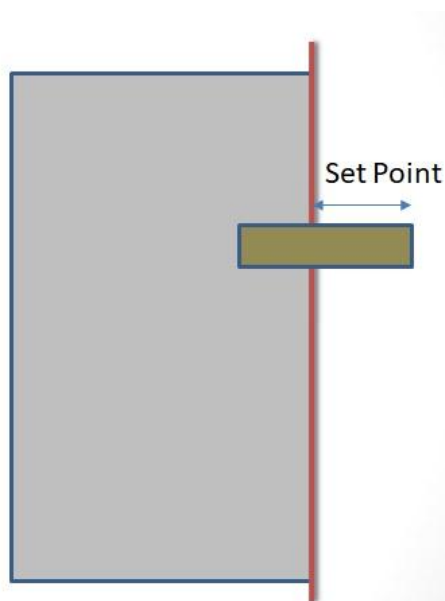


Figura 29 - Representação do controle do sistema de centralização pelas bordas.

Fonte: Autores, 2018

Este sistema se caracteriza de diversas formas como:

- A posição da borda ou a posição da linha central da tira pode ser controlado com alta precisão;
- Manutenção fácil e sem problemas podendo ser feita facilmente no local de instalação;
- Alta sensibilidade, o deslocamento da tira para dentro de + ou - 0.05 mm pode ser detectado;
- Emprego da ação integral onde a velocidade de operação é proporcional ao desvio.

Em operações como o corte lateral, que é necessário para ajustar a desigualdade da borda da tira, o sistema EPC torna essa etapa dispensável e o lado lateral da bobina é protegido reduzindo custos de produção e mão de obra, uma vez que a tira percorrerá sempre na posição desejada evitando ao máximo o desperdício.

### **2.17 Controle de posição pelo centro**

Consiste em parametrizar uma linha como referencial de centro no objeto que se deseja fazer o controle, para que se possa estabelecer uma relação entre as extremidades adjacentes e o centro do objeto. Baseando-se no plano cartesiano, a diferença entre o lado direito e o centro gera uma coordenada positiva e a diferença entre o lado esquerdo e o centro gera uma coordenada negativa. Se as duas coordenadas coincidem temos então a centralização do objeto, pois o resultado da diferença é nulo, facilitando assim a interpretação do objeto e transmitindo informações em tempo real ao controlador, que por sua vez gera sinais de correção ao atuador (NIRECO, 2007). A Figura 30 representa o sistema de controle pelo centro.

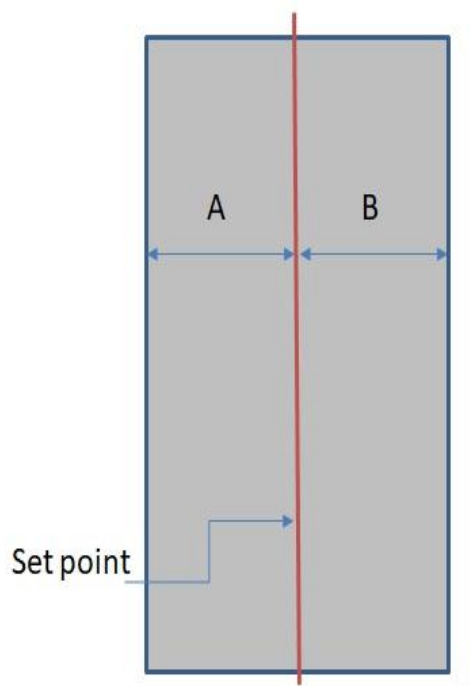


Figura 30 - Representação do controle do sistema de centralização pelo centro

Fonte: Autores, 2018

### 3 DESENVOLVIMENTO

#### 3.1 Método Proposto

O método desenvolvido é uma alternativa para sistemas de centralização de tira que usam emissores e receptores de luz como sensores para realimentar a malha de controle. A proposta deste trabalho é um sistema baseado em visão computacional aonde uma câmera juntamente com uma unidade de processamento serve como sensor para realimentar a malha de controle dos atuadores, conforme Figura 31.

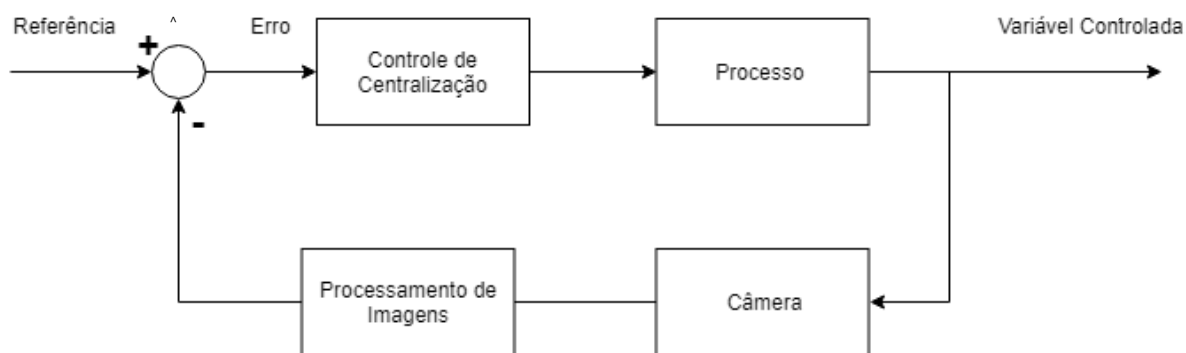


Figura 31 – Diagrama de bloco do controle do sistema de centralização.

Fonte: Autores, 2018.

#### 3.2 Solução proposta

A abordagem para a visão computacional será de forma geral aplicada conforme fluxograma representado na Figura 32. Porém o foco deste trabalho está voltado para a captura das imagens e o processamento destas, de modo a obter um sistema realimentado, ou como comumente chamado feedback.

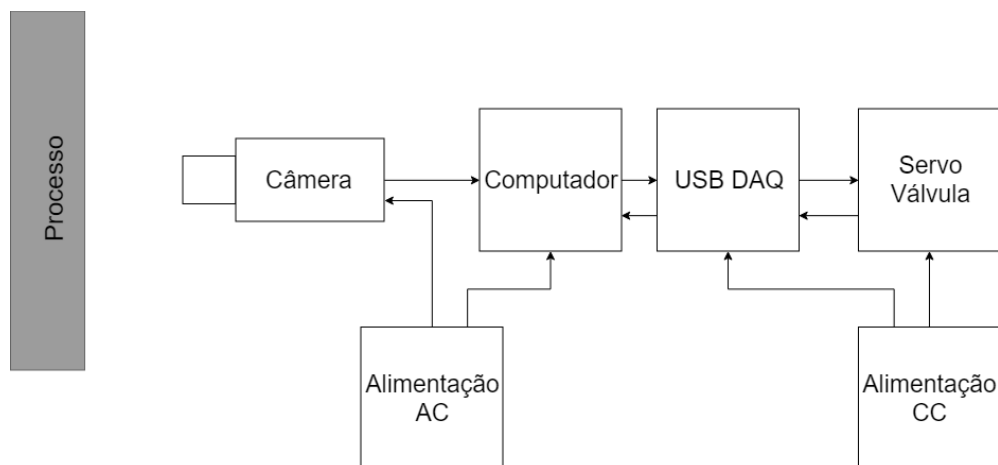


Figura 32 - Componentes que formam o sistema de centralização.

Fonte: Autores, 2018.

A Figura 33 ilustra a instalação do sistema em uma linha de laminação, em que a tira tensionada é abraçada por um rolo. Com a movimentação do rolo, por um cilindro controlado por uma servo-válvula, a posição da tira pode ser controlada.

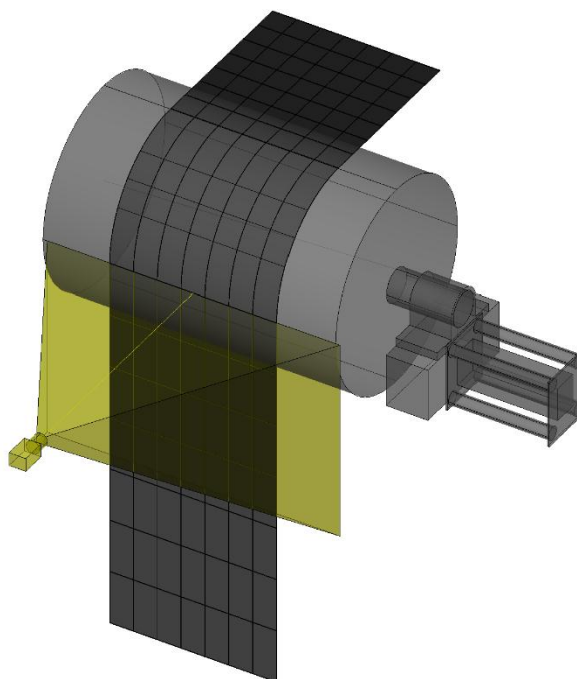


Figura 33 - Sistema para aplicação do projeto.

Fonte: Autores, 2018.

### **3.3 Ambiente de Desenvolvimento**

#### **3.3.1 Python**

Python é uma linguagem de programação de alto nível criado por Guido van Rossum em 1991. O objetivo dessa linguagem é produtividade e legibilidade. Uma linguagem moderna e relativamente muito poderosa, contendo funções capazes de realizar uma grande variedade de tarefas (Python, 2001).

Entre os muitos ambientes de programação o Spyder é um dos ambientes que traz facilidades em debugar os códigos, além de já instalar módulos científicos como: NumPy, Matplotlib e SciPy.

#### **3.3.2 OpenCV**

A OpenCV (Open Source Computer Vision Library) é uma biblioteca de função para aplicações em sistemas de visão computacional. Esta biblioteca foi desenvolvida pela Intel, podendo ser usada em sistemas operacionais como Linux e Windows (OPENCV, 2018).

A biblioteca abrange uma vasta aplicação na área de visão computacional com funções para calibração de imagem, identificação de objetos, segmentação e reconhecimento de borda.

### **3.4 Software**

#### **3.4.1 Captura de imagem**

A captura de imagem é feita por câmera IP, sendo necessário estabelecer uma conexão entre a câmera e o computador através de requisições TCP/IP. Depois de estabelecida a conexão entre ambos, a câmera envia frame a frame para o computador para que sejam executados os algoritmos computacionais. Ao finalizar cada algoritmo é solicitado um novo frame.

### 3.4.2 Processos para calibração de câmera

As câmeras são únicas e exclusivas quanto ao foco das lentes, inclinação e posição dos sensores de captura de imagem em relação ao eixo focal. A calibração é de extrema importância para processos que utilizam de visão computacional, pois a correta calibração está associada ao sucesso das etapas seguintes. Sendo assim, é necessária a remoção da distorção ocasionada pela lente da câmera.

De acordo com a documentação do OpenCV, para uma boa calibração são necessários no mínimo 10 imagens. Para este processo de calibração foram utilizadas as seguintes funções da biblioteca OpenCV: `findChessboardCorners`, `calibrateCamera` e `undistort`.

A função `findChessboardCorners` tem como objetivo encontrar as extremidades, ou cantos, de cada quadrado da mesa de xadrez. Para cada imagem é necessário encontrar os cantos e armazená-los para usar posteriormente. A Figura 34 representa os cantos extraídos.

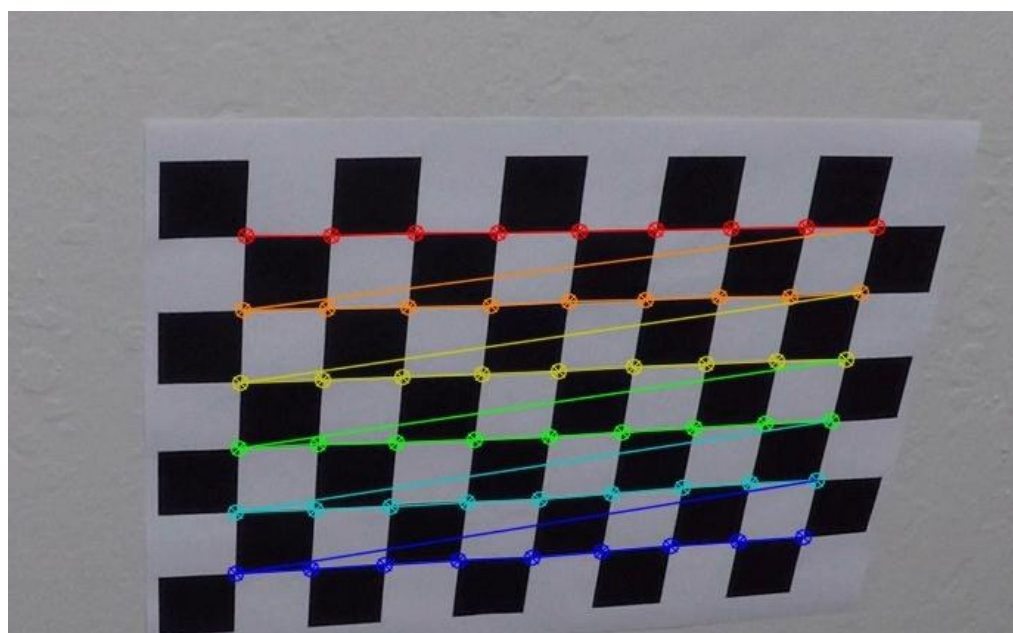


Figura 34 - Representação dos cantos extraídos.

Fonte: Autores, 2018.

Na função `calibrateCamera` estima os parâmetros intrínsecos da câmera e parâmetros extrínsecos são extraídos para cada uma das vistas. O algoritmo é baseado em Zhang (2000) e BouguetMCT. As coordenadas dos pontos de objeto 3D e suas projeções 2D correspondentes em cada vista devem ser armazenadas pela função

findChessboardCorners. Isso pode ser alcançado usando um padrão, como a mesa de xadrez.

Contudo a função calibrateCamera provê somente os valores da matriz de calibração. Para desfazer a distorção causada pelas lentes a função undistort é necessária.

### 3.4.3 Processos para Detecção de bordas e Limiarização

Para centralização a detecção de borda tem que ser precisa. Para isto é usado a função Sobel do OpenCV, o resultado é mostrado na Figura 35.

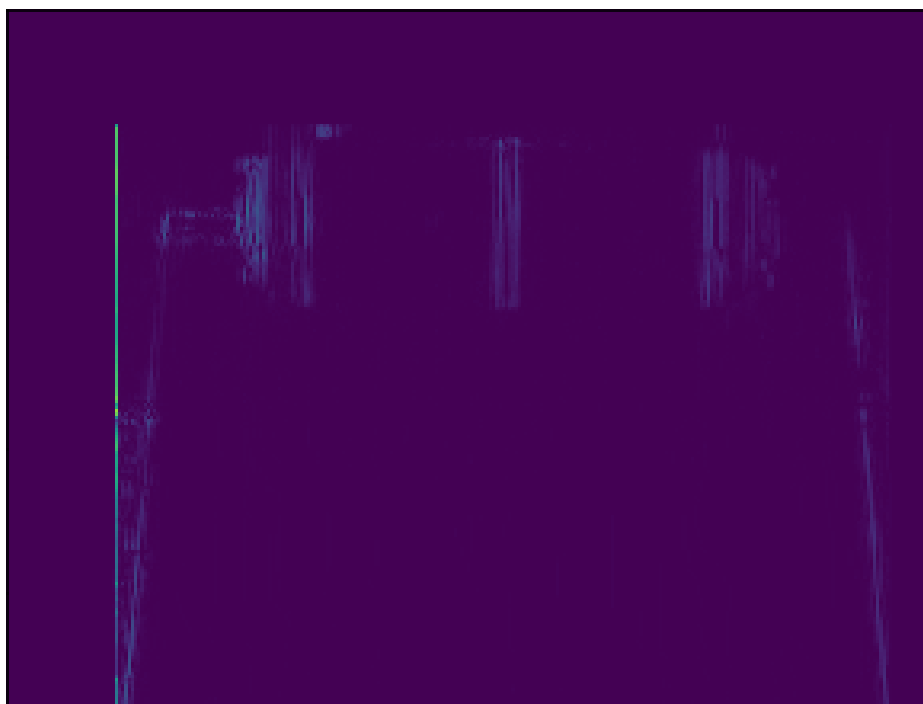


Figura 35 - Imagem depois da Detecção de bordas.

Fonte: Autores, 2018.

Porém a imagem resultado do algoritmo de detecção ainda necessita de alguns tratamentos para uma detecção mais clara. Com essa necessidade de melhoria na detecção é utilizado o processo de limiarização para destacar as bordas. O resultado da combinação dessas funções está representado na Figura 36.

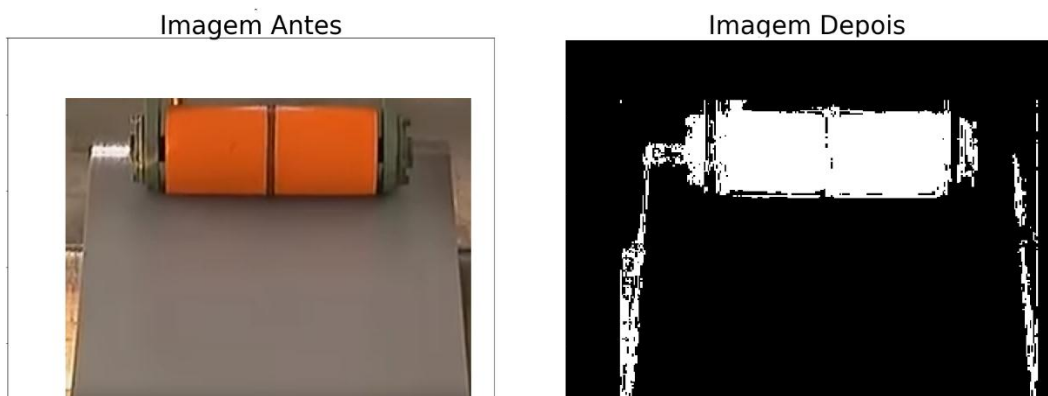


Figura 36 - Imagem antes e depois da detecção de bordas e limiarização.

Fonte: Autores, 2018

#### 3.4.4 Algoritmo de Centralização

Para funcionamento do algoritmo de centralização, o primeiro passo é montar o histograma. Com o histograma conseguimos ver as frequências das tonalidades de cinza da imagem em que se fez o processo de detecção de bordas. Observa-se que o histograma representa claramente as bordas conforme a Figura 37.

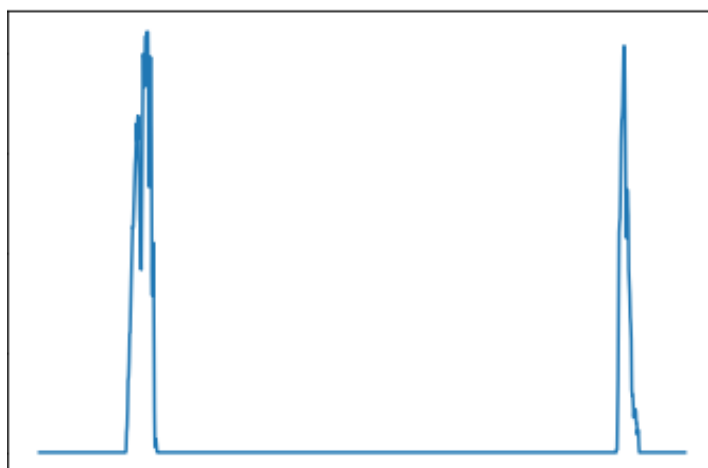


Figura 37 – Histograma para centralização.

Fonte: Autores, 2018.

O algoritmo de centralização é o responsável pela interpretação das imagens e o gerador do controle. A Figura 38 representa o fluxograma de funcionamento do algoritmo.

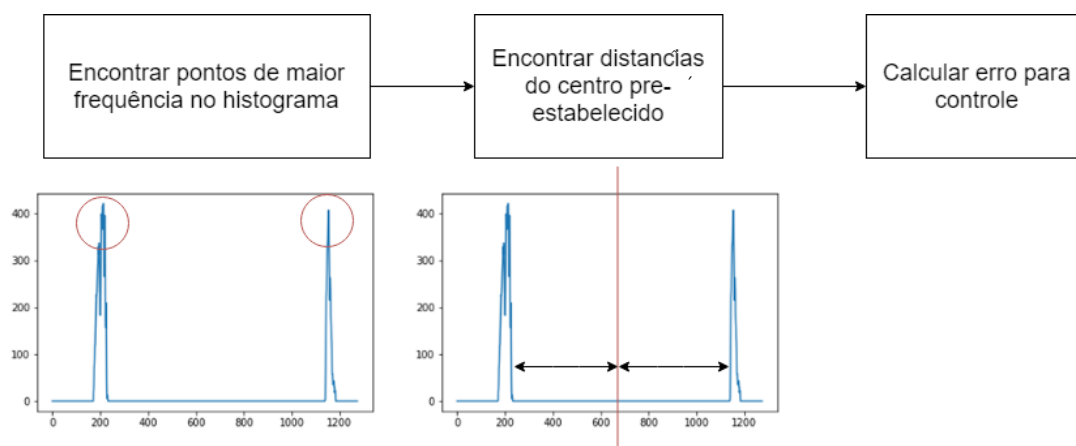


Figura 38 - Fluxograma do Algoritmo de centralização.

Fonte: Autores, 2018.

O algoritmo de centralização calcula a diferença entre as bordas e um centro pré-estabelecido, se a diferença da distância do centro para a borda direita for igual para a borda esquerda assumimos que a tira está centrada e não necessita de correção. A Figura 39 mostra o cálculo para centralização.

```

1 referencia_centro = 423 # Valor de preset
2 tira_centro = (borda_d + borda_e) / 2 #centro provido pelas bordas
3 error = referencia_centro - tira_centro

```

Figura 39 - Algoritmo em Python de centralização.

Fonte: Autores, 2018.

### 3.4.4 Representação Final

A Figura 40 mostra a representação final do software onde é indicado: número de frames analisados, tempo de execução do algoritmo em segundos, valores detectados das bordas, Set Point e Erro. Além desses dados pode-se observar uma representação da detecção de bordas no canto superior direito, onde as bordas estão destacadas.

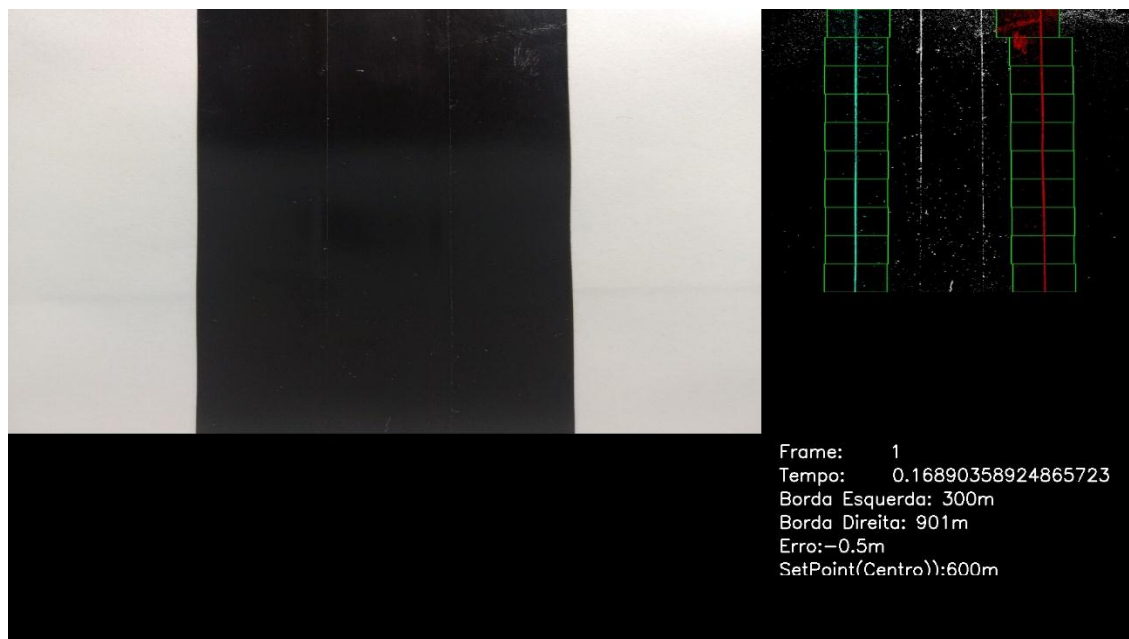


Figura 40 - Representação final gerada pelo software.

Fonte: Autores, 2018.

## **4 RESULTADOS EXPERIMENTAIS**

### **4.1 Considerações**

Neste capítulo será apresentado os resultados obtidos através do software desenvolvido. Os dados obtidos são referentes a uma simulação de centralização de uma tira de papel. Os testes são feitos por um conjunto de imagens e por aquisição de imagens pela rede (TCP/IP) por uma câmera de celular.

Para os testes experimentais serão levados em conta a precisão do algoritmo, repetibilidade dos valores e o tempo de resposta.

### **4.2 Experimento para Calibração de Câmera**

Para encontrar a matriz de calibração da câmera, foram retiradas 11 fotos do padrão de calibração, mesa de xadrez, em diferentes posições. A Figura 41 contém as imagens usadas.

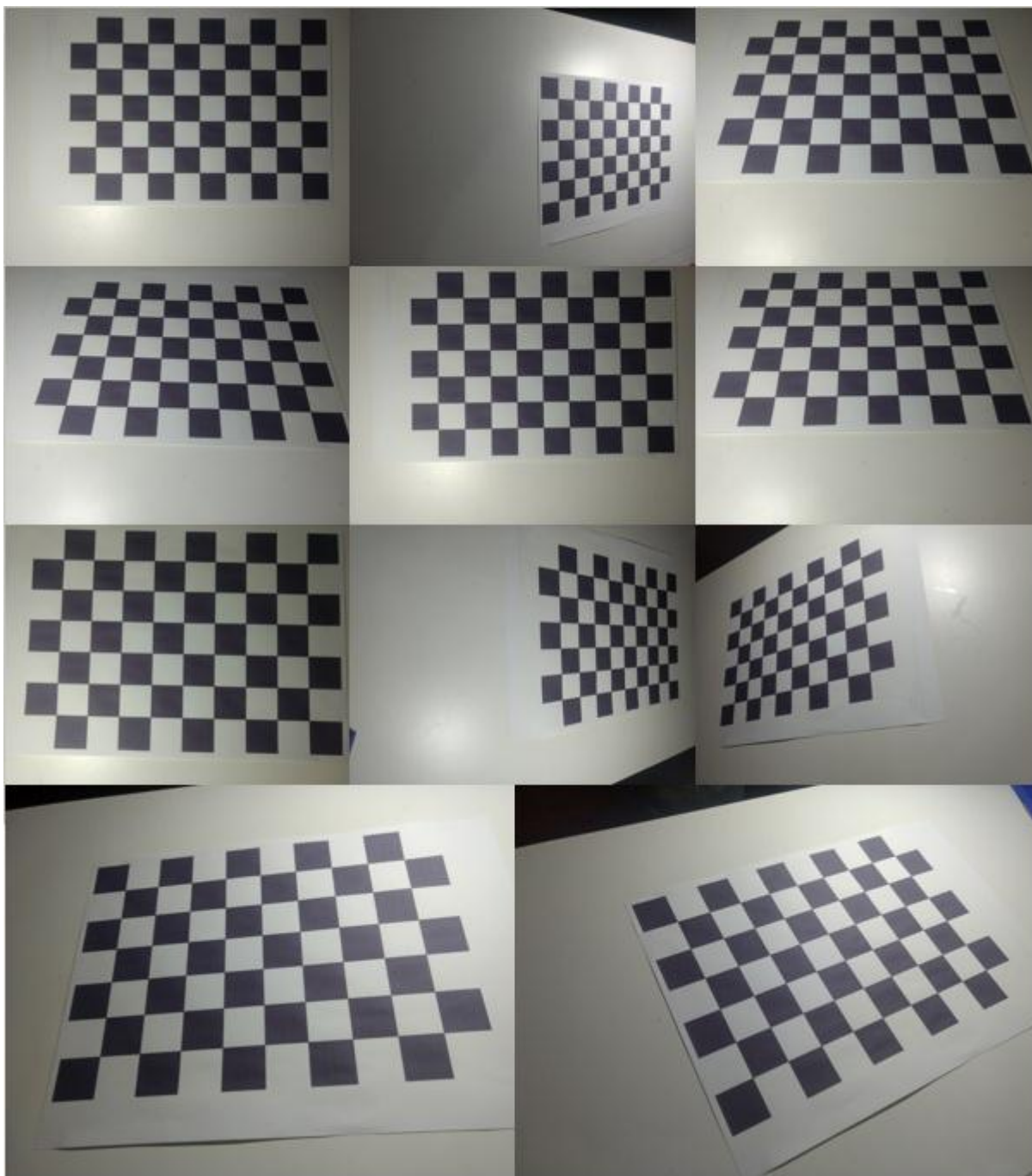


Figura 41 - Imagens usadas para calibração experimental.

Fonte: Autores, 2018.

### 4.3 Resultado da Calibração de Câmera

Após o processamento das imagens, foram geradas, conforme Figura 42, as imagens com a detecção de cantos.

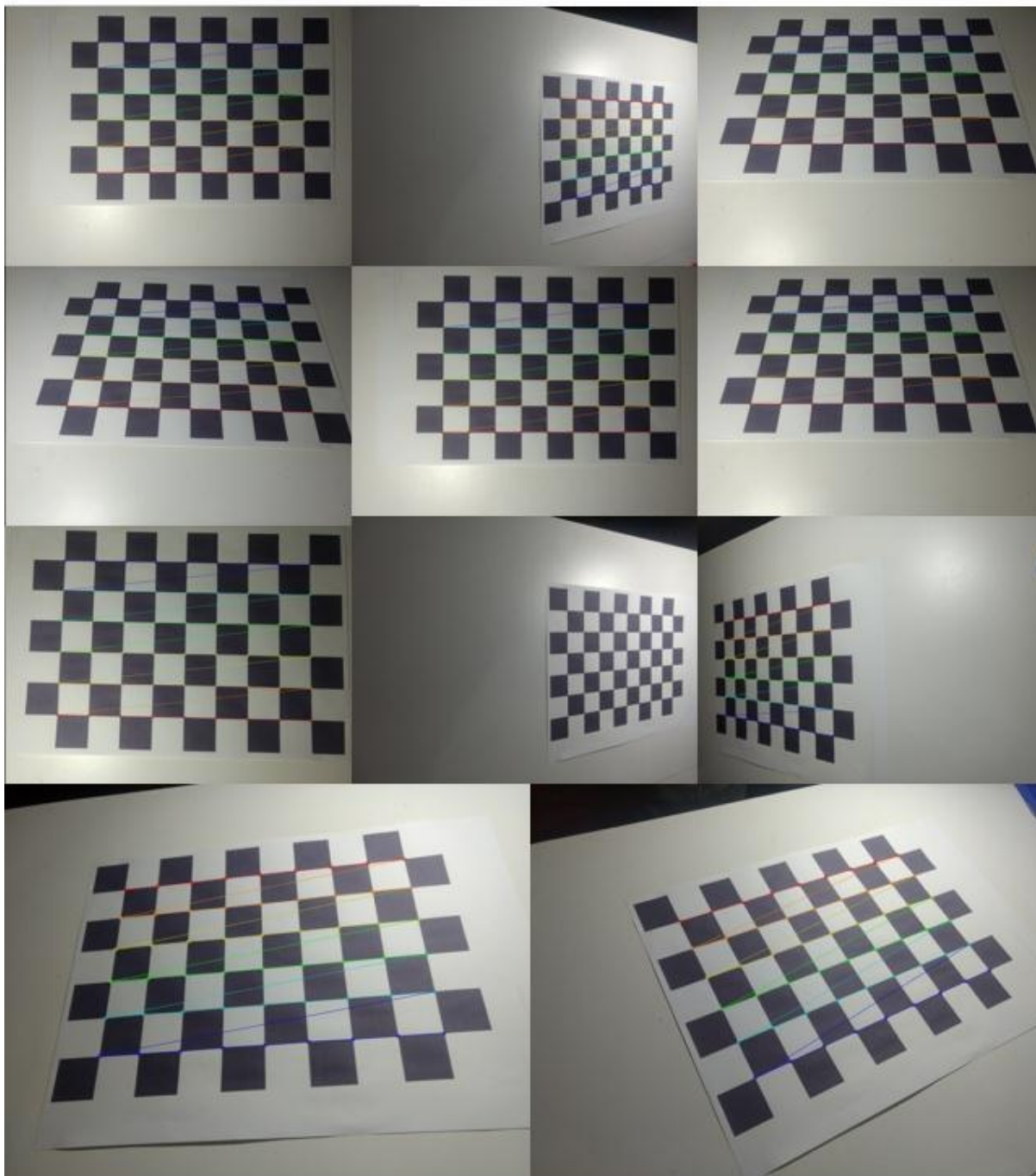


Figura 42 - Imagens com marcação de cantos.

Fonte: Autores, 2018.

Com todos esses pontos marcados e armazenados, eles são processados e é possível encontrar a matriz de calibração, conforme abaixo:

$$\text{Matriz da câmera} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1587,24 & 0 & 726,27 \\ 0 & 1585,23 & 455,75 \\ 0 & 0 & 1 \end{bmatrix} \quad (4.1)$$

Com a matriz definida foi possível desfazer as distorções da câmera. A Figura 43 representa os resultados obtidos, sendo que esta figura é a sobreposição entre a imagem original e a sem distorções, com fins de ressaltar a diferença das mesmas.



Figura 43 - Imagens sem distorções da câmera.

Fonte: Autores, 2018.

#### 4.4 Experimento para Centralização de Tira

Para nível de experimento, uma tira de papel foi colocada a uma distância de 10cm da câmera, sendo a câmera e o plano paralelos entre si. Uma serie de imagens foram tiradas para simulação e foram realizados os testes. A Figura 44 ilustra a protótipo montado para o experimento e a Figura 45 representa a tira de 5cm utilizada nos testes que equivalem a 600px.

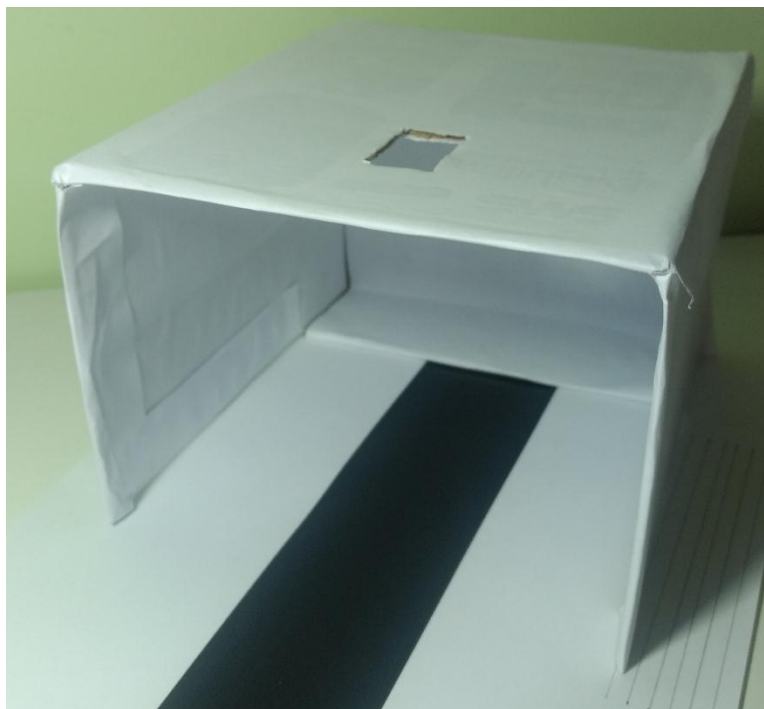


Figura 44 - Protótipo montado para testes.

Fonte: Autores, 2018.

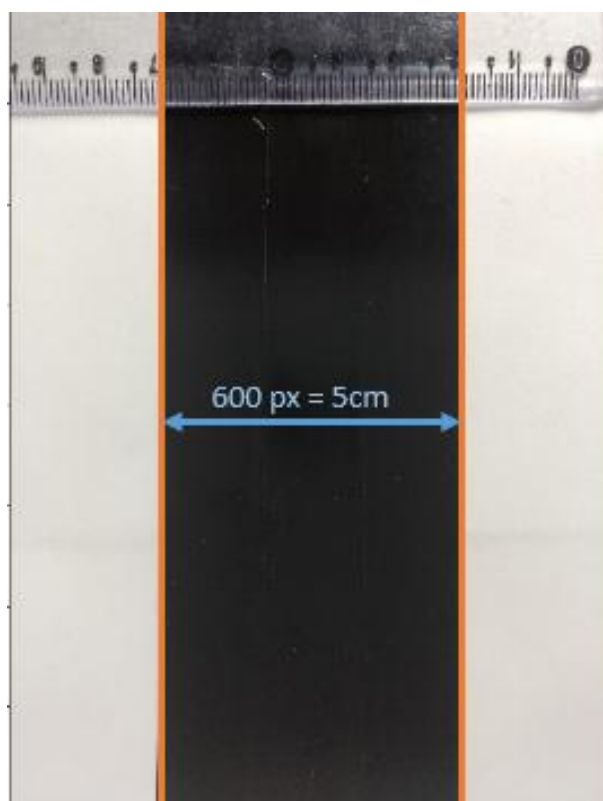


Figura 45 - Tira usada para os testes.

Fonte: Autores, 2018.

Com o sistema montado e a tira de teste devidamente posicionada, foram obtidos uma seqüência de imagens, sendo seis destas ilustradas na Figura 46.

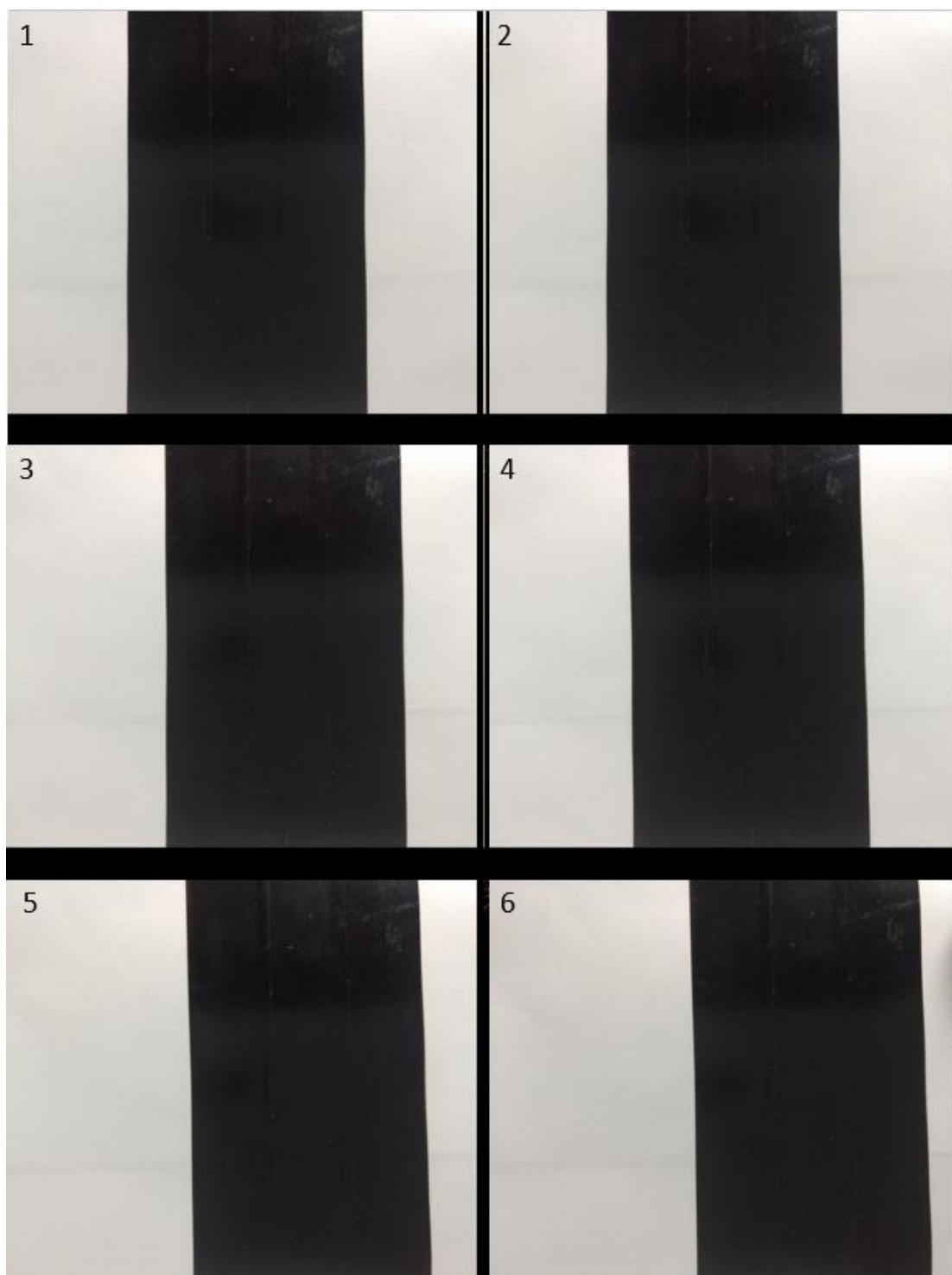


Figura 46 - Sequência e imagens usadas para os testes de centralização.

Fonte: Autores, 2018.

#### 4.5 Resultado da Centralização de Tiras

Com o processamento do algoritmo de centralização foram obtidos os valores para as bordas direitas, esquerdas e o erro entre o valor de referência e o centro encontrado. A Figura 47 contém as imagens processadas e os valores encontrados se encontram na Tabela 1.

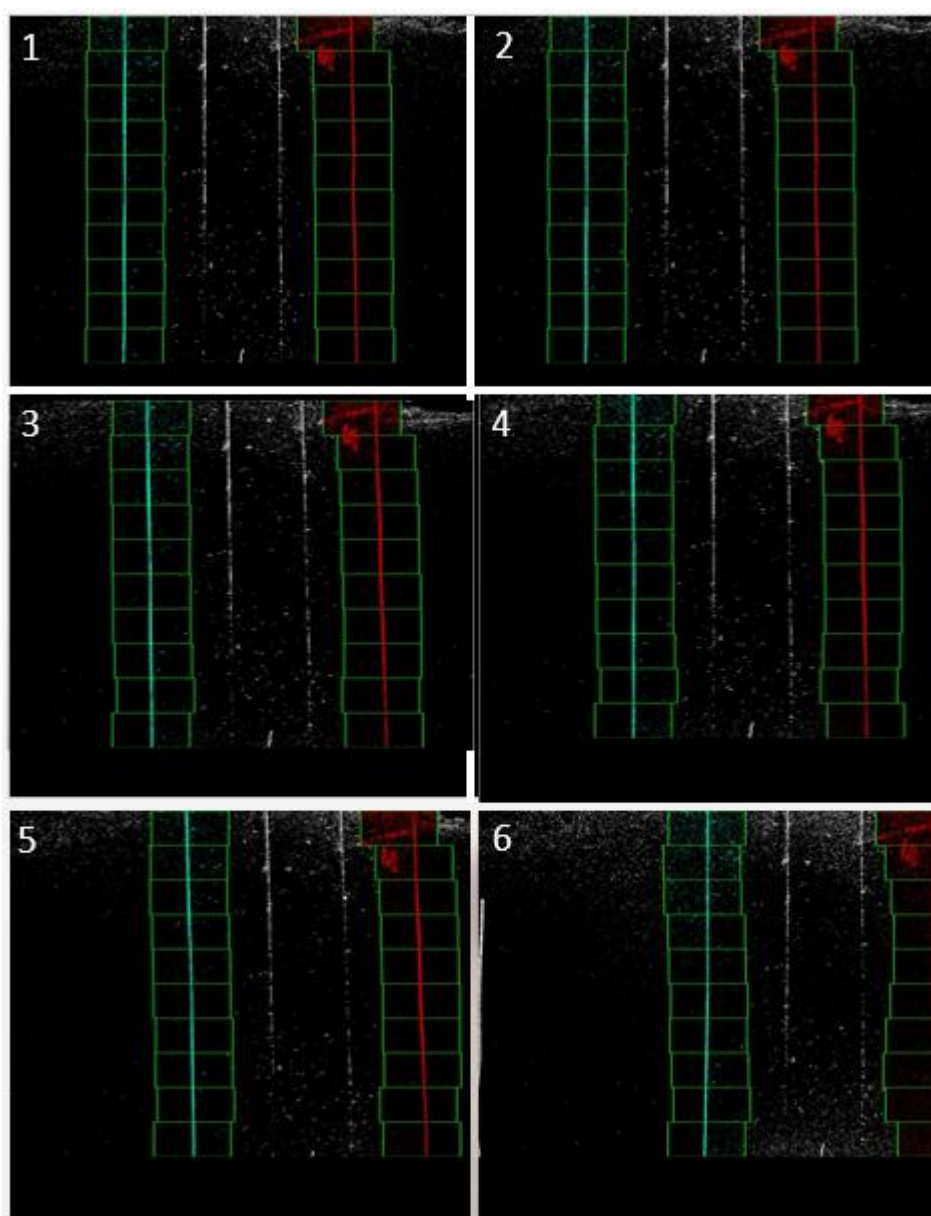


Figura 47 - Sequência e imagens processadas nos testes de centralização.

Fonte: Autores, 2018.

Tabela 1 - Resultados do algoritmo de centralização

Frame	Tempo (s)	Borda Esquerda (pixels)	Borda Direita (pixels)	Centro de Referência (pixels)	Erro (pixels)
1	0.068904	300	901	600	-0.5
2	0.048915	300	901	600	-0.5
3	0.054912	365	968	600	-66.5
4	0.050914	400	1004	600	-102
5	0.048915	478	1073	600	-175.5
6	0.051913	529	1125	600	-227

Fonte: Autores, 2018.

Porém para mensurar a precisão foi calculado, através dos valores encontrados para as bordas, a largura da tira. Sabendo que o valor da tira é de 5cm e tem a equivalência de 600px, conforme a Figura 45, foram obtidos os resultados contigo na Tabela 2.

Tabela 2 – Resultados do cálculo de largura

Frames	Borda Esquerda (pixels)	Borda Direita (pixels)	Largura calculada (pixels)	Largura real da tira (pixels)	Diferença (pixels)	Porcentagem de Variação
1	300	901	601	600	1	0,17%
2	300	901	601	600	1	0,17%
3	365	968	603	600	3	0,50%
4	400	1004	604	600	4	0,67%
5	478	1073	595	600	5	0,83%
6	529	1125	596	600	4	0,67%

Fonte: Autores, 2018.

Pelos resultados apresentados, foi obtida uma variação média de 0,50%, sendo assim obtendo uma precisão de 99,50%. Além disso, foi realizado 100 vezes o processamento de cada imagem aleatoriamente a fim de obter qual a repetibilidade, obtendo 99,9% de repetibilidade.

Outro fator levado em consideração foi o tempo para todos os processos ocorrerem, o tempo de resposta, que foram divididos em algumas etapas conforme Tabela 3. Estes tempos tem sua importância quando levado em conta que os processos in-

dustriais têm velocidades entre 100 e 500 metros por minuto (mpm). Tempo de resposta muito alto podem significar grandes quantidade de material sendo produzido sem centralização.

Tabela 3 – Tempos de resposta

<b>Descrição</b>	<b>Tempo (ms)</b>
Algoritmo + Geração de Layout final + Aquisição de Imagem pela rede	300
Algoritmo + Aquisição de Imagem pela rede	200
Aquisição de Imagem pela rede	140
Algoritmo processando imagem arquivada	55

Fonte: Autores, 2018.

Com esses tempos, nota-se que o gargalo dos testes foi à aquisição de imagem da câmera do celular, pois o trafego do celular até o computador foi de 140ms. Porém ao analisar o tempo para todo algoritmo ser processado, 55ms, e uma velocidade de processo de 500mpm, calcula-se que 45cm de material terá sido produzido. Esta quantidade de material produzida é aceitável, pois a imagem capturada pela câmera é maior 45cm, sendo assim todo material produzido terá sido centralizado.

## 5 CONCLUSÃO

Os objetivos estabelecidos para este trabalho foram alcançados, principalmente no que concerne à apresentação de uma proposta de sistema de centralização de tira. A utilização de sistemas que permitem a realização de controle por visão computacional é funcional, e esta pesquisa ressalta que apesar de simples, mostrou grande eficiência.

Os resultados satisfatórios observados nos testes realizados com o protótipo representativo de centralização de tira mostram que o sistema é factível e pode ser implementado para a aplicação a qual se destina. O sistema apresentou uma alternativa para os sistemas atuais, mostrando-se funcional.

As imagens e dados gerados foram suficientes para atender a aplicação em questão, com precisão, repetibilidade e tempo de resposta aceitáveis para vários processos.

## 6 REFERÊNCIAS BIBLIOGRÁFICAS

BALLARD, D. H., & BROWN, C. M. (1982). **Computer Vision**. Englewood Cliffs, New Jersey: Prentice-Hall.

FERNÁNDEZ, M. L. (2004). **Calibração de câmera baseado em algoritmo de detecção de cantos e inserção de objetos virtuais dentro da imagem de vídeo capturada**. Acesso em 20 de Agosto de 2018, disponível em <http://webserver2.tecgraf.puc-rio.br/~mgattass/ra/trb03/ManuelLoaiza/>

GONZALES, R. C., & WOODS, R. E. (2010). **Processamento de Imagens digitais**. São Paulo: Pearson.

JANERT, P. K. (2014). **Controle com Feedback para Sistemas de Computação**. Rio de Janeiro: LTC.

MORGAN, J. (2008). **Técnicas de Segmentação de Imagens na Geração de Programas para Máquinas de comando numérico**. Santa Maria.

NIRECO. (2007). **EPC / CPC SYSTEM**. Japão: NIRECO CORPORATION. Acesso em 10 de agosto de 2018, disponível em [http://www.nireco.jp/catalog/pdf/0502epc\\_cpc\\_system\\_e.pdf](http://www.nireco.jp/catalog/pdf/0502epc_cpc_system_e.pdf)

NISE, N. S. (2017). **Engenharia de Sistemas de Controle**. Rio de Janeiro: LTC.

OPENCV. (10 de Outubro de 2018). **OpenCV Documentation**. Acesso em 10 de Outubro de 2018, disponível em OpenCV: <https://docs.opencv.org/2.4/index.html>

Python. (2001). **Python Reference Manual**. Acesso em 5 de Setembro de 2018, disponível em <https://docs.python.org/2.0/ref/ref.html>

TAKEDA, F. B. (2000). **Sistema para Tomada de decisão baseado na geração de imagens estereoscópicas e reconhecimento de padrões circulares**.

