

UNIVERSIDADE FEDERAL DE SANTA CATARINA
CENTRO TECNOLÓGICO DE JOINVILLE
CURSO DE ENGENHARIA MECATRÔNICA

GUSTAVO HENRIQUE GRÜTZMANN

IMPLEMENTAÇÃO DO MÉTODO DE RASTREAMENTO POR MARCADORES PARA
RECONHECIMENTO DO MOVIMENTO DA MÃO COM REPLICABILIDADE EM
GARRA ROBÓTICA

Joinville
2024

GUSTAVO HENRIQUE GRÜTZMANN

IMPLEMENTAÇÃO DO MÉTODO DE RASTREAMENTO POR MARCADORES PARA
RECONHECIMENTO DO MOVIMENTO DA MÃO COM REPLICABILIDADE EM
GARRA ROBÓTICA

Trabalho apresentado como requisito parcial para obtenção do título de bacharel em Engenharia Mecatrônica, no Curso de Engenharia Mecatrônica, do Centro Tecnológico de Joinville, da Universidade Federal de Santa Catarina.

Orientadora : Dra. Tatiana Renata Garcia

Joinville
2024

Dedico este trabalho à minha família, que fez o impossível para que meu sonho se tornasse possível.

AGRADECIMENTOS

Agradeço primeiramente à Universidade Federal de Santa Catarina por proporcionar um ambiente de qualidade em que tive o prazer de concluir o meu curso de graduação, com profissionais qualificados que me deram toda a educação e conhecimento necessário para minha evolução pessoal e acadêmica.

Agradeço à minha orientadora, Dra. Tatiana Renata Garcia, por ter abraçado minha proposta de tema e por ter me auxiliado incontáveis vezes, durante todo o percurso da minha graduação.

Agradeço à toda minha família pelo apoio incondicional, em especial meu pai Roque, minha mãe Marlene e minha irmã Daniella, por terem acreditado no meu sonho e por me ajudarem a torná-lo realidade, mesmo distantes se fizeram muito presentes nos meus dias.

Agradeço aos meus amigos e colegas que me acompanharam nesta jornada, tornando meus dias mais felizes e sendo como uma segunda família pra mim, levarei todas as boas amizades para a vida.

Por fim, agradeço a todos que de alguma maneira se fizeram presentes na minha vida. Meu muito obrigado a cada um de vocês.

RESUMO

Com o avanço da tecnologia e sua crescente inserção nas indústrias brasileiras, torna-se mais acessível a integração entre humanos e máquinas para melhorar a produção e facilitar o trabalho. Considerando isso, em conjunto com o aumento gradativo em casos de acidentes envolvendo mãos, este trabalho apresenta um método de rastreamento do movimento da mão, com o intuito de montar um protótipo acessível para realizar interações com um atuador mecatrônico, na forma de braço robótico. Buscando atender requisitos de que garantam fluidez, como tempo de resposta e exatidão. Por meio de pesquisas bibliográficas, foi escolhido o método de rastreamento por marcadores e implementado em etapas para validar cada parte da aplicação. Para a validação do sistema, foram realizados testes quantitativos de tempo e de assertividade, em que resultados satisfatórios foram apontados, indicando o sucesso do trabalho, da mesma forma, melhorias foram apontadas para trabalhos futuros.

Palavras-chave: Rastreamento de Movimento. Processamento de Imagem. Segurança.

ABSTRACT

With the advancement of technology and its increasing incorporation into Brazilian industries, the integration between humans and machines in order to improve production and facilitate work has become more accessible. Considering this, along with the gradual rise in accidents involving hands, this study presents a hand movement tracking method aimed at developing an accessible prototype for interactions with a mechatronic actuator. The goal is to meet requirements that ensure fluidity, such as response time and accuracy. Through bibliographic research, the marker-based tracking method was chosen and implemented in stages to validate each part of the application. For system validation, quantitative tests on time and accuracy were conducted, yielding satisfactory results that indicate the success of the study. Additionally, suggestions for improvements were proposed for future research.

Keywords: Motion Tracking. Image Processing. Safety.

LISTA DE FIGURAS

Figura 1 – Rastreamento por temperatura	16
Figura 2 – Rastreamento por subtração de fundo	17
Figura 3 – Rastreamento por correlação	17
Figura 4 – Rastreamento por marcadores	18
Figura 5 – Experimento de dispersão da luz branca	19
Figura 6 – Cubo RGB unitário	20
Figura 7 – Espaço de cores HSV	22
Figura 8 – Imagem em escala de cinza e sua matriz de intensidade	23
Figura 9 – Fluxograma	26
Figura 10 – Protótipo do sistema	30
Figura 11 – Luva	31
Figura 12 – Braço Robótico	32
Figura 13 – Circuito Eletrônico	33
Figura 14 – Obtenção dos valores HSV	34
Figura 15 – Janela de rastreamento	35
Figura 16 – Movimentos do braço robótico	36
Figura 17 – Tempo de processamento	39
Figura 18 – Rastreamento da cor verde	40
Figura 19 – Rastreamento da cor azul	40
Figura 20 – Rastreamento da cor amarelo	41
Figura 21 – Rastreamento da cor rosa	41

LISTA DE TABELAS

Tabela 1 – Componentes do sistema eletrônico.	33
---	----

SUMÁRIO

1	INTRODUÇÃO	11
1.1	OBJETIVOS	12
1.1.1	Objetivo Geral	12
1.1.2	Objetivos Específicos	12
2	FUNDAMENTAÇÃO TEÓRICA	13
2.1	TECNOLOGIA NAS FÁBRICAS BRASILEIRAS	13
2.2	MÉTODOS DE INTEGRAÇÃO DA TECNOLOGIA	13
2.2.1	Manipulação 3D	14
2.2.2	Navegação	14
2.2.3	Entrada simbólica	15
2.2.4	Controle do sistema	15
2.3	RASTREAMENTO DA MÃO	15
2.3.1	Técnica de Temperatura	16
2.3.2	Técnica de Subtração de Fundo	16
2.3.3	Técnicas de Correlação	17
2.3.4	Técnicas de Marcadores	18
2.4	COMPONENTES DE UM SISTEMA DE RASTREAMENTO	19
2.4.1	ESPAÇO DE CORES	19
2.4.1.1	RGB	20
2.4.1.2	CMY e CMYK	21
2.4.1.3	HSV	21
2.4.2	PIXEL	22
2.4.2.1	ESCALA DE CINZA	22
2.4.3	Limiarização	23
2.4.4	HARDWARE	24
3	METODOLOGIA	25
3.1	DESCRIÇÃO DO SISTEMA PROPOSTO	25
3.1.1	MÉTODO DE RASTREAMENTO	25
3.1.2	SISTEMA DE TRANSMISSÃO DOS MOVIMENTOS DOS DEDOS: LUVA COLORIDA	27
3.1.3	SISTEMA DE CAPTURA DE IMAGEM	27
3.1.4	SISTEMA DE PROCESSAMENTO DE IMAGEM	28
3.1.5	SISTEMA DE SOFTWARE	28
3.1.6	SISTEMA DE ATUAÇÃO MECÂNICA	28

4	DESENVOLVIMENTO	29
4.1	DESENVOLVIMENTO E PROTOTIPAGEM DOS SISTEMAS	29
4.1.1	DESENVOLVIMENTO DA LUVA COLORIDA	30
4.1.2	DESENVOLVIMENTO MECÂNICO DO BRAÇO ROBÓTICO	31
4.1.3	DESENVOLVIMENTO ELETRÔNICO DO BRAÇO ROBÓTICO	32
4.2	ALGORITMOS PROPOSTOS	33
4.2.1	OBTENÇÃO DE VALORES HSV	34
4.2.2	RASTREAMENTO	35
4.2.3	MOVIMENTAÇÃO DO BRAÇO ROBÓTICO	37
5	RESULTADOS	38
5.1	RESULTADOS DOS TESTES	38
5.1.1	AVALIAÇÃO DA VELOCIDADE DE PROCESSAMENTO	38
5.1.2	AVALIAÇÃO DA ROBUSTEZ À VARIAÇÃO DA LUZ	39
5.2	DISCUSSÃO DOS RESULTADOS	41
5.2.1	INTERPRETAÇÃO DOS DADOS OBTIDOS NOS TESTES	42
5.2.2	LIMITAÇÕES DO SISTEMA DESENVOLVIDO	42
6	CONCLUSÕES	43
6.1	POSSÍVEIS MELHORIAS	43
	REFERÊNCIAS	45
	APÊNDICE A	47
	ANEXO A	52

1 INTRODUÇÃO

A adoção de novas tecnologias digitais nos ambientes industriais vem apresentando uma crescente nos últimos anos, impulsionada pelos avanços tecnológicos e pela busca da eficiência operacional. Conforme apresentado pela Confederação Nacional da Indústria (CNI), por meio da Sondagem Especial 83: Indústria 4.0 Cinco Anos Depois, em 2021, 69% das indústrias brasileiras utilizavam pelo menos uma tecnologia digital. Esta pesquisa realizada com mais de mil empresas comenta sobre os benefícios na adesão à tecnologia, sendo eles o aumento na produtividade, melhora na qualidade de produtos e diminuição de custos (CONFEDERAÇÃO NACIONAL DA INDÚSTRIA, 2022).

Contudo, na busca por acelerar os processos fabris, a integração entre humanos e tecnologia pode apresentar falhas e riscos aos operadores. Segundo o Anuário Estatístico de Acidentes do Trabalho – AEAT do Ministério da Previdência Social (MPS), no Brasil, aproximadamente 23% dos acidentes registrados no trabalho em 2022 afetaram mãos, variando desde fraturas leves até amputamentos. Esse tipo de acidente representa a ocorrência mais comum nos registros de empresas para o referido ano, evidenciando a necessidade de medidas de segurança adicionais nos ambientes de trabalho (MINISTÉRIO DA PREVIDÊNCIA SOCIAL, 2022).

Além disso, o MPS relatou que o número de acidentes do trabalho aumentou nos últimos anos, registrando 465 mil acidentes em 2020, 580 mil em 2021 (aumento de 24%) e 648 mil em 2022 (incremento de 11%). Esses dados indicam um aumento na periculosidade dos ambientes fabris, resultando em impactos negativos para os colaboradores e gerando custos significativos para empresas e governo. Conforme o Tribunal Superior do Trabalho (2022) "No ano de 2021, segundo dados do Instituto Nacional do Seguro Social, foram gastos 17,7 bilhões de reais com auxílio-doença e 70,6 bilhões de reais com aposentadoria por invalidez", destacando o impacto econômico dessas ocorrências para o setor público e privado.

Visando minimizar os riscos associados a atividades de alta periculosidade, propõe-se um método de rastreamento dos movimentos da mão humana por imagem, como os desenvolvidos por Agostinho (2014), Conti e Gonzaga (2011) e Freitas (2011). O objetivo é aplicar esse método em um sistema mecatrônico, que simule tarefas com complexidade de movimentos, permitindo a realização de atividades perigosas de forma mais segura ao usuário, reduzindo os riscos de acidentes.

Para a implementação desse método, o presente trabalho inicia-se com uma pesquisa bibliográfica, como meio de contextualizar a inserção da tecnologia no setor industrial brasileiro nos últimos anos, apresentando trabalhos relacionados e explicitação dos métodos e elementos pertinentes ao objetivo. Em seguida, o

desenvolvimento do projeto é detalhado, apresentando as escolhas feitas e embasadas nas informações coletadas, juntamente com os resultados obtidos por meio de testes realizados com o sistema construído. Por fim, são discutidas as conclusões e análises dos resultados, discorrendo sobre as reais aplicações possíveis para o sistema, sugestões de melhorias para o projeto e demais considerações para o aprimoramento do projeto.

1.1 OBJETIVOS

1.1.1 Objetivo Geral

Desenvolver um método robusto e de baixo custo, que realize o rastreamento do movimento da mão e que possa ser integrado a um sistema mecatrônico, na forma de um braço robótico, de modo a contribuir para a diminuição de acidentes em ambientes fabris.

1.1.2 Objetivos Específicos

- Definir por meio de análise de viabilidade o método adequado;
- Analisar os elementos necessários para a aplicação;
- Aplicar o sistema e definir os limites do projeto.

2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo, serão apresentados os conceitos e definições envolvendo a evolução da tecnologia nas fábricas brasileiras e sobre sua integração nos processos produtivos. Na sequência os métodos de integração da tecnologia com seres humanos são fundamentados, com um enfoque no objetivo do trabalho que é o rastreamento da mão.

2.1 TECNOLOGIA NAS FÁBRICAS BRASILEIRAS

No estudo desenvolvido pela CNI (2022), são detalhadas as tecnologias mais adotadas pelas indústrias brasileiras, destacando-se em primeira posição os sistemas integrados de engenharia para desenvolvimento e manufatura de produtos. Essa tecnologia permite a incorporação de sensores, controladores lógicos programáveis e robôs no processo produtivo, promovendo uma maior automação e integração dos processos. A tabela completa com as formas de tecnologia utilizadas pelas empresas está disponível no anexo A do estudo, oferecendo uma visão detalhada das inovações que vêm transformando o setor industrial.

A adesão da tecnologia ocorre nos mais diferentes portes de empresas e nos mais variados âmbitos do mercado devido aos benefícios ofertados, sendo o benefício mais reconhecido pelas empresas o aumento na produtividade, mostrando que a tecnologia consegue trazer soluções específicas para cada setor. Além disso, a possibilidade de customização do produto, que a tecnologia oferece, representa uma vantagem significativa para atingir a satisfação de clientes no mercado contemporâneo (CONFEDERAÇÃO NACIONAL DA INDÚSTRIA, 2022).

Entretanto, a inserção de novas tecnologias nas fábricas exige integração com o trabalho manual dos operadores. Esse processo de integração requer uma interface adequada que permita a interação entre os trabalhadores e as ferramentas tecnológicas, facilitando o fluxo de informações e o controle dos processos.

2.2 MÉTODOS DE INTEGRAÇÃO DA TECNOLOGIA

A Interface Homem-Máquina (IHM) é responsável por conectar o mundo físico ao virtual, podendo ser adaptados para uma ampla gama de aplicações, aumentando a versatilidade e a aplicabilidade da IHM. As técnicas mais aceitas pela comunidade da área são organizadas em quatro categorias principais: manipulação, controle do sistema, navegação e entrada simbólica (TORI; KIRNER; SISCOOTTO, 2006 apud BOWMAN et al., 2004).

O desenvolvimento de uma interface responsiva e intuitiva é fundamental para criar um ambiente de uso agradável, visto que as pessoas buscam atividades que não ofereçam obstáculos para sua utilização. Uma IHM bem projetada deve apresentar desempenho que assegure a precisão e a confiabilidade das operações, além de proporcionar uma usabilidade que facilite a comunicação das intenções do usuário. A robustez também é um requisito essencial, permitindo que a interface funcione em diferentes condições e ambientes, ampliando sua aplicabilidade e garantindo a adaptabilidade do sistema a variados contextos.

2.2.1 Manipulação 3D

A seleção e manipulação de um objeto virtual em ambiente 3D, incluindo a possibilidade de mover ou modificar características desse objeto e de navegar pelo espaço virtual, é um dos métodos de interação mais abrangentes, oferecendo uma ampla gama de possibilidades para o usuário. Esse método pode ser considerado como o método mais completo, no entanto, é fundamental que a interação seja projetada para espelhar de forma precisa as ações do mundo real.

Para o aproveitamento completo desse método, é necessário uma forma de retornar as informações para o usuário, geralmente por meio de feedback visual, que consiga ilustrar claramente a interseção entre o mundo físico e o virtual. Esse feedback permite ao usuário visualizar os objetos presentes no ambiente e as ferramentas de manipulação que podem ser utilizadas. A presença desses elementos além de orientar o usuário também define as restrições e capacidades do sistema.

2.2.2 Navegação

A movimentação é um elemento crucial na construção de ambientes imersivos, pois permite ao usuário deslocar-se de um ponto a outro no mundo virtual. Esse deslocamento pode ter como finalidade a exploração (naive), onde o usuário se move livremente, ou a busca por um local específico (primed), em que a navegação é orientada por um objetivo definido. Considerado um método delicado, pois usuários diferentes possuem métodos de localização diferentes, que devem ser levados em consideração para proporcionar uma experiência intuitiva.

Para maximizar o realismo e a imersão, o ambiente físico pode ser mapeado e reproduzido virtualmente, possibilitando a simulação de um espaço similar ao real. Esse mapeamento permite ainda a inclusão de objetos virtuais que enriquecem a visualização, tornando a interação mais envolvente. A vista do usuário pode ser tratada como um personagem presente ativamente na cena e ou apenas uma câmera com visão externa para supervisionar e observar objetos, sua construção e disposição dos elementos.

2.2.3 Entrada simbólica

A técnica de entrada simbólica, que integra a inserção de caracteres e números, é fundamental para realizar tarefas de modificação de propriedades, escrita de frases e renomeação de arquivos em sistemas interativos. Devido à necessidade de precisão nos dados inseridos, o uso de reconhecimento de voz não é recomendado, pois as limitações atuais de leitores de voz podem resultar em inconsistências e erros.

Além da entrada tradicional por teclado, a técnica permite a utilização de dispositivos baseados em caneta, onde o usuário desenha diretamente os caracteres ou símbolos desejados. Essa entrada gráfica é processada pelo programa, que reconhece os caracteres representados.

2.2.4 Controle do sistema

O método de controle do sistema é amplamente utilizado para alterar o estado de um sistema, sendo frequentemente implementado por meio de menus ou comandos. Utilizado principalmente como uma função supervisória, também pode ser adaptado para manipulações ativas de objetos, dependendo do contexto e das necessidades da aplicação.

Esse método é baseado na entrada multimodal, permitindo a combinação de diferentes técnicas de interação para ampliar sua funcionalidade e acessibilidade. As técnicas mais comuns utilizadas em conjunto incluem menus gráficos, comandos de voz, ferramentas físicas e comandos de gestos (rastreamento da mão). A integração dessas entradas possibilita uma interação mais intuitiva e natural, ao mesmo tempo em que oferece flexibilidade para atender às preferências e habilidades dos usuários.

2.3 RASTREAMENTO DA MÃO

O rastreamento do corpo humano constitui um desafio técnico significativo, tanto pelas características do ambiente em que ocorre a captura dos movimentos quanto pela complexidade dos próprios movimentos realizados pelos seres humanos (SANTOS, 2015).

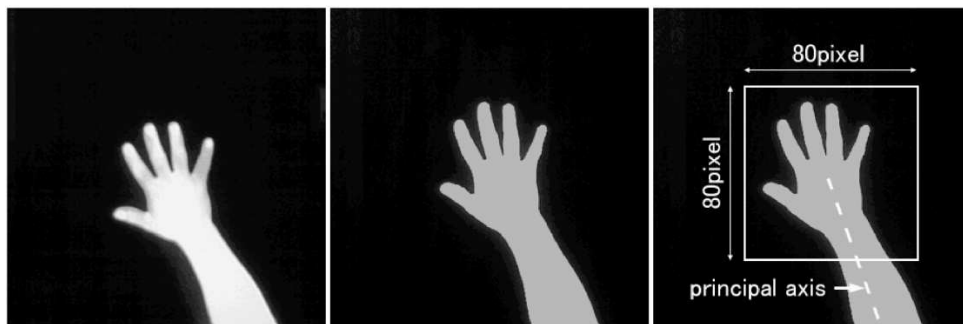
Para lidar com essas dificuldades, diversos métodos têm sido propostos com o objetivo de representar os movimentos e estados das mãos. Entre as técnicas destacam-se o rastreamento por imagem, que utiliza algoritmos de visão computacional para analisar dados capturados por câmeras; o uso de luvas com sensores integrados, capazes de medir diretamente os movimentos articulares; sensores de profundidade, que mapeiam posições tridimensionais em tempo real; e sensores de movimento, que registram aceleração e orientação (FREITAS, 2011).

2.3.1 Técnica de Temperatura

O método baseado na detecção de diferenças de temperatura utiliza o contraste térmico entre o corpo humano e o fundo da imagem para localizar e rastrear a mão. No entanto, a técnica apresenta dois limitante significativos, é essencial o uso de uma câmera infravermelha e a funcionalidade do sistema fica condicionada a um fundo que não emita calor.

Koike, Saito e Kobayashi (2001) desenvolveram um sistema interativo baseado em uma mesa que integra uma aplicação computacional, utilizou-se uma câmera infravermelha e uma câmera panorâmica. Na Figura 1 é apresentado o formato da mão, resultado do rastreamento por temperatura.

Figura 1 – Rastreamento por temperatura



Fonte: Koike et al. (2001, p. 312).

O projeto atingiu resultados exatos e satisfatórios, demonstrando alta eficiência na detecção dos dedos e na interação em tempo real. A combinação de câmeras infravermelhas e panorâmicas foi fundamental para garantir a exatidão do rastreamento. Além disso, o sistema destacou-se pela integração de projeções visuais diretamente na superfície da mesa, facilitando o uso e tornando o ambiente mais intuitivo para os usuários.

2.3.2 Técnica de Subtração de Fundo

O reconhecimento da mão do usuário por meio da cor da pele, utilizando a comparação com um fundo de cor estática, configura-se como um método de baixo custo e acessível para aplicações de rastreamento. No entanto, sua precisão pode ser impactada por variações no tom de pele, na cor do fundo e nas condições de iluminação, o que limita sua utilização em ambientes totalmente controlados.

Conti e Gonzaga (2011) desenvolveram um sistema baseado na segmentação de cor, o sistema incluiu a implementação de um programa capaz de rastrear os dedos, utilizando algoritmos para extrair informações relevantes. Adicionalmente, foi incorporado um código de anti-occlusão, com o objetivo de melhorar a robustez do método. A Figura 2 apresenta as imagens geradas pelo método de subtração de fundo.

Figura 2 – Rastreamento por subtração de fundo



Fonte: Conti e Gonzaga (2011, p. 107).

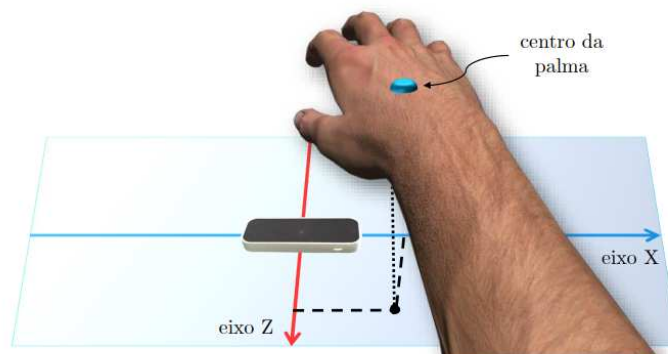
Levando em considerações as limitações do hardware utilizado e do método implementado, o sistema foi considerado eficiente por seus autores, executando com precisão as tarefas planejadas.

2.3.3 Técnicas de Correlação

A técnica de correlação é utilizada prioritariamente para estabelecer relações entre imagens ou informações estáticas e ações previamente definidas. Geralmente, sua implementação envolve o uso de inteligência artificial, que auxilia na identificação de padrões complexos e na interpretação precisa de dados visuais.

Silva (2017) aplicou sensores Leap Motion, dispositivos capazes de capturar o movimento dos dedos da mão, em conjunto com algoritmos de aprendizado de máquina. Os resultados obtidos foram satisfatórios na maior parte dos testes. No entanto, algumas situações específicas exigiram ajustes mais refinados para garantir a calibração adequada. Na Figura 3, apresenta-se o sistema construído, mostrando o funcionamento do rastreamento baseado no sensor Leap Motion.

Figura 3 – Rastreamento por correlação



Fonte: Silva (2017, p. 60).

Quando a inteligência artificial não dispõe de dados suficientes ou não é devidamente treinada para reconhecer todos os movimentos, ela pode acabar

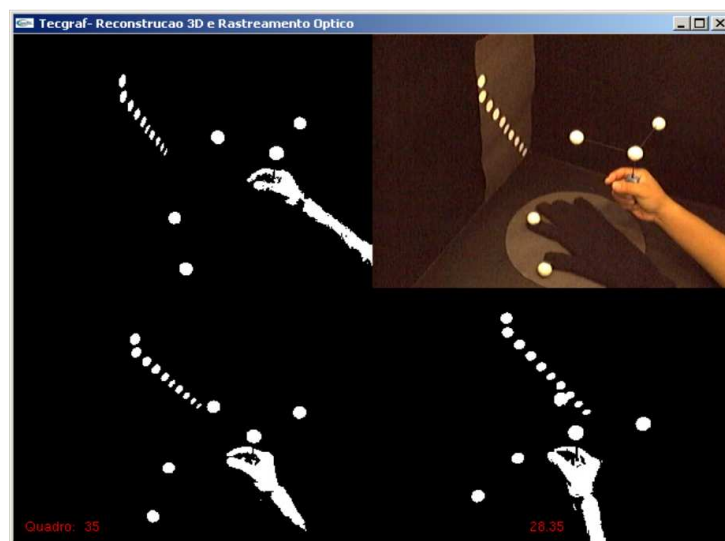
negligenciando ou interpretando de maneira imprecisa as ações realizadas pelo usuário.

2.3.4 Técnicas de Marcadores

O método de rastreamento baseado em marcadores pré-definidos utiliza cores distintas para identificar diferentes partes da mão ou pontos físicos específicos. A técnica reconhece esses marcadores por meio de imagens capturadas, permitindo manter o rastreamento contínuo e estabelecer a correspondência entre os pontos da mão e o ambiente virtual. Essa abordagem é frequentemente usada em sistemas que não exigem uma precisão milimétrica, mas que ainda necessitam de uma relação eficaz entre os elementos físicos e virtuais.

Loaiza et al. (2005) desenvolveram uma metodologia de rastreamento viável, alcançando resultados satisfatórios em cenários onde a precisão extrema não é um requisito crítico. O estudo demonstrou que é possível construir um sistema de rastreamento de baixo custo, que ainda assim cumpre com os objetivos propostos. Na Figura 4, são apresentadas as vistas das câmeras utilizadas para localizar e identificar os marcadores, ilustrando a configuração do sistema de rastreamento.

Figura 4 – Rastreamento por marcadores



Fonte: Loaiza et al. (2005, p. 1680).

Para realizar o rastreamento por imagem, as técnicas de marcadores e subtração de fundo aplicam métodos de segmentação no espaço de cores, visando separar as cores de interesse e realçá-las na imagem. Esse processo de segmentação é fundamental para destacar as regiões ou pontos específicos que precisam ser rastreados, permitindo que o sistema identifique e acompanhe esses pontos ao longo do tempo. Ambos os métodos dependem de algoritmos que segmentam a imagem com base em critérios de cor e luminosidade, o que facilita o rastreamento e a análise dos movimentos em tempo real.

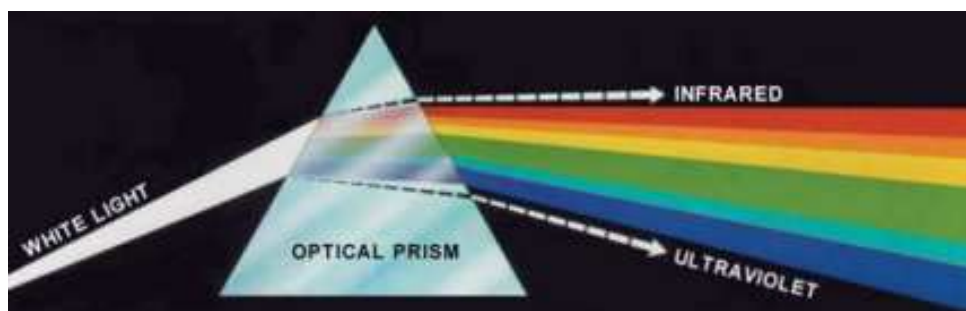
2.4 COMPONENTES DE UM SISTEMA DE RASTREAMENTO

Esta seção descreve os componentes em comum entre os sistemas de rastreamento que utilizam os métodos de subtração de fundo e de segmentação de cores, os quais foram utilizados como base para o desenvolvimento do sistema montado neste trabalho. Em ambos os métodos, a cor é um elemento fundamental para o processamento da imagem, sendo crucial para a identificação e o rastreamento dos pontos de interesse. A segmentação de cores, por sua vez, envolve a distinção de diferentes regiões da imagem com base em suas características cromáticas, permitindo a separação entre o fundo e os objetos de interesse, como a mão. Para que esse processo obtenha êxito, é necessário compreender as diferentes representações de cores, que são realizadas por meio de espaços de cores.

2.4.1 ESPAÇO DE CORES

O conceito de cor é utilizado para definir diferentes comprimentos de onda eletromagnéticas presentes na luz. O processo de percepção da cor realizada pelo cérebro é mais profundo que apenas um processo fisiológico, no entanto a natureza física da cor pode ser definida com embasamentos teóricos e experimentais. As cores percebidas em um objeto são determinadas pela natureza da luz refletida por estes objetos e como provada por Sir Isaac Newton em 1666, segundo experimento apresentado na Figura 5, a luz pode ser decomposta em uma faixa de frequência no espectro eletromagnético (GONZALES; WOODS, 2008).

Figura 5 – Experimento de dispersão da luz branca



Fonte: Gonzales; Woods (2008, p. 395)

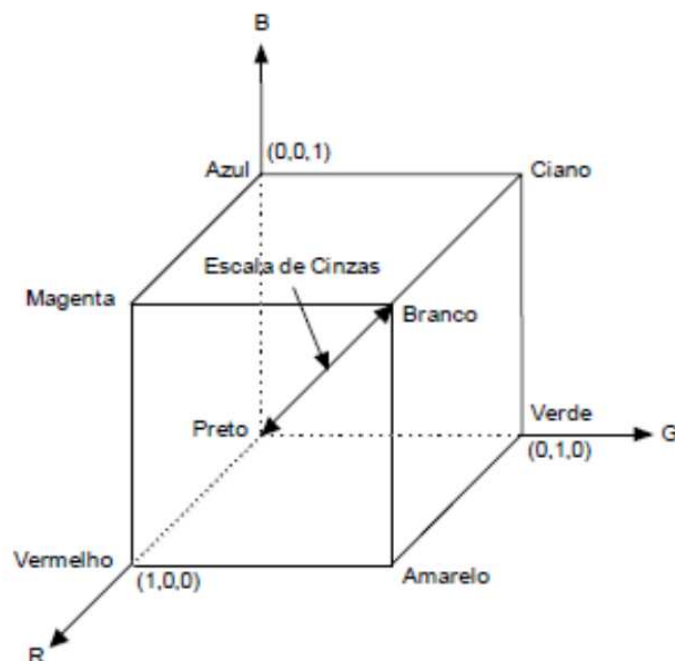
O uso de cores em métodos de rastreamento de imagens apresenta uma vantagem significativa, pois as cores funcionam como um descritor poderoso que facilita a identificação e extração de objetos ou regiões específicas em uma cena. No entanto, para realizar a detecção e o rastreamento eficientes das cores em uma imagem, é necessário compreender as definições e características dessas cores dentro de ambientes padronizados.

Ainda segundo Gonzales e Woods (2008), o espaço de cores foi criado com o propósito de representar as cores de uma forma padrão, normalmente dentro de um sistema de coordenadas onde cada cor é um ponto nesse espaço. No âmbito de processamento digital de imagens, os modelos mais utilizados na prática atualmente são o RGB (Red, Green, Blue) em monitores e câmeras. CMY (Cyan, Magenta, Yellow) e CMYK (Cyan, Magenta, Yellow, Black) para impressoras, e o modelo HSV (Hue, Saturation, Value) em sistemas de rastreamento, este último modelo apresenta a vantagem de representar as cores da maneira mais próxima que os humanos descrevem e interpretam.

2.4.1.1 RGB

No modelo RGB, as cores são representadas com base em três componentes espectrais primários: vermelho, verde e azul. Este modelo é fundamentado em um sistema cartesiano tridimensional, onde cada cor é representada por um ponto dentro de um cubo e é definida por um vetor a partir da origem, conforme ilustrado na Figura 6. Os pontos dentro deste cubo correspondem a diferentes combinações das intensidades dessas três cores primárias (GONZALES; WOODS, 2008).

Figura 6 – Cubo RGB unitário



Fonte: Fonte: Marques Filho; Vieira Neto (1999 p. 121)

Imagens representadas no modelo RGB são compostas por três canais ou componentes, correspondentes às cores primárias: vermelho, verde e azul. Cada um desses canais é uma imagem separada, e a combinação dessas imagens resulta na imagem colorida final. No processo de captura da cor de uma imagem, a técnica é

basicamente o inverso desse processo, onde a segmentação da imagem é realizada por meio da aplicação de filtros sensíveis a cada uma dessas cores primárias. Esses filtros atuam de forma a isolar as informações de cada componente de cor. Quando um filtro monocromático (sensível a uma cor específica) é aplicado em uma imagem colorida, o resultado é uma nova imagem que destaca a intensidade da cor filtrada em cada ponto da imagem original. (GONZALES; WOODS, 2008).

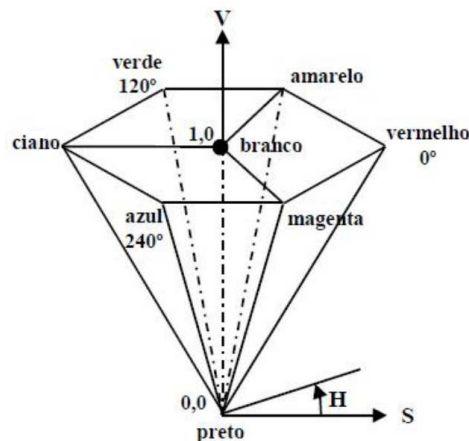
2.4.1.2 *CMY e CMYK*

Ciano, Magenta e Amarelo são as cores secundárias da luz e as cores primárias de pigmentação. Uma superfície com a cor ciano não reflete vermelho, com a cor magenta não reflete verde e com a cor amarelo não reflete azul. Em teoria, quantidades iguais de ciano, magenta e amarelo produzem preto, no entanto na prática não é alcançado um preto verdadeiro, portanto é adicionada a cor preta ao modelo (GONZALES; WOODS, 2008). Este espaço de cores é representado em conjunto com o espaço RGB na Figura 6, e é possível ver os vértices representados pelas cores Ciano, Magenta e Amarelo, que estão opostas às cores primárias.

2.4.1.3 *HSV*

Diferente dos modelos RGB e CMY, que são mais voltados para a representação técnica e a combinação de cores no espaço de luz, o modelo HSV é fundamentado na percepção visual humana, facilitando a interpretação e o ajuste de cores em sistemas digitais. O modelo de cores HSV (Matiz, Saturação e Brilho) foi desenvolvido para representar de forma mais próxima à maneira como os humanos percebem as cores. Matiz representa a cor pura, independente de ser uma cor primária ou não, a saturação dá a noção do quanto essa cor pura está diluída por luz branca, já o brilho é uma descrição, sem medição, que engloba noção de intensidade (GONZALES; WOODS, 2008). Um esquema do modelo HSV é apresentado na Figura 7.

Figura 7 – Espaço de cores HSV



Fonte: Farias apud Schowengert (2016, p. 27)

Como apresentado anteriormente, as cores RGB podem ser interpretadas como três imagens de intensidade em cores monocromáticas, portanto é possível extrair o valor de intensidade de cada uma dessas imagens componentes e aplicá-las no espaço HSV.

2.4.2 PIXEL

As imagens no ambiente virtual devem ser representadas de maneira discreta para serem exibidas em telas de computador. Para isso, utiliza-se o conceito de pixel, que é o menor elemento capaz de transmitir informações dentro de uma imagem. Compreender a definição e a funcionalidade do pixel é essencial para o entendimento dos sistemas de processamento de imagem, uma vez que ele é a unidade básica de visualização.

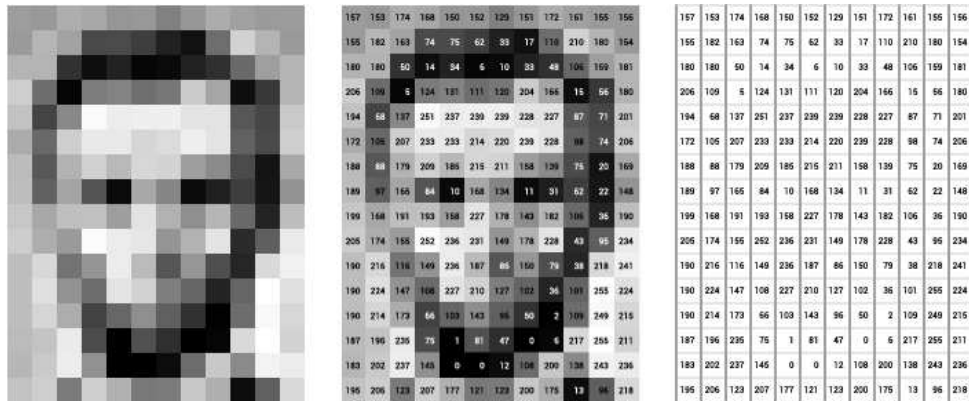
Uma imagem pode ser descrita por meio de uma função $f(x,y)$, onde x corresponde ao comprimento e y à altura da imagem. Cada pixel, portanto, tem uma coordenada única dentro dessa função, o que permite sua localização específica. Além disso, o pixel contém um valor de intensidade associado, que é responsável por definir a tonalidade ou cor da imagem. A partir desses dados, é possível representar a imagem em matrizes e calcular distâncias e comprimentos dos objetos presentes na cena (GONZALES; WOODS, 2008).

2.4.2.1 ESCALA DE CINZA

Escala de cinza é uma maneira de representar as imagens, onde cada pixel assume uma cor que varia do preto ao branco, em que todos os valores do meio são representados por diferentes tons de cinza. O valor de intensidade do pixel, usualmente segue uma faixa de valores de 0 até 255, sendo 0 preto absoluto e 255 branco absoluto

(ARAÚJO, 2019). A Figura 8 apresenta um exemplo de uma imagem em escala de cinza e a matriz que representa a intensidade de cada pixel presente na imagem.

Figura 8 – Imagem em escala de cinza e sua matriz de intensidade



Fonte: Araújo (2019, p. 13)

A matriz de intensidade desempenha um papel fundamental nos algoritmos de processamento de imagem e no reconhecimento de áreas pré-determinadas. Por meio dessa matriz, é possível estabelecer relações entre pixels vizinhos, permitindo a análise e a manipulação da estrutura da imagem. Essa abordagem facilita a identificação e o destaque de regiões que possuam características específicas, como cores ou tonalidades desejadas.

2.4.3 Limiarização

No contexto do processamento de imagem, o limiar é um valor predefinido utilizado para segmentar diferentes partes de uma imagem com base na intensidade dos pixels. O processo consiste em comparar a intensidade de cada pixel com o valor do limiar, decidindo se o pixel será classificado em uma das duas categorias possíveis, como preto ou branco. Caso a intensidade do pixel seja superior ao valor limite, ele será classificado em uma das categorias, caso contrário, será classificado na outra (MOURA, 2018).

Conforme Bertholdo (2007, p.23 e seg), a limiarização é separada em três métodos:

- Limiarização Global: compara toda a imagem com um único valor de thresholding, eficaz quando a diferença de intensidade entre o objeto e o fundo é constante em toda imagem;
- Limiarização Baseada em Entropia: encontra um valor de limiar baseado nas ocorrências de diferentes níveis de cinza presentes na imagem, portanto consegue trabalhar com mais opções além do preto e branco, com isso é possível realizar uma separação dinâmica da limiarização de diferentes objetos presentes na imagem;

- Limiarização Adaptativa: a imagem é segmentada em diversos retângulos, preferencialmente contendo apenas um pixel, para que a diferença de iluminação entre objetos não interfira na limiarização e o resultado possa ser mais assertivo.

2.4.4 HARDWARE

Os sistemas de processamento de imagem requerem uma série de componentes essenciais para a realização de suas funções. Conforme descrito por Gonzales e Woods (2008), um sistema geral de processamento de imagem é composto por vários elementos interdependentes. Entre eles, destacam-se os sensores, responsáveis pela captura da imagem, e o computador, que realiza o processamento dos dados obtidos. O computador pode variar desde um supercomputador dedicado até sistemas com configurações mais modestas, desde que atendam aos requisitos mínimos para o processamento específico.

Além disso, o sistema inclui o software, que executa as operações de processamento de imagem, utilizando os recursos do computador. O armazenamento é necessário para guardar os frames capturados pelos sensores e outras informações relevantes para o processo. Para a exibição dos resultados, é necessário um método de visualização, que permita a análise dos dados processados. Por fim, uma maneira de comunicação é crucial para garantir que todos os componentes do sistema possam trocar informações de maneira eficaz.

No trabalho proposto por Agostinho (2014), foram utilizados um par de luvas 5DT Data Glove Ultra, que consegue rastrear o movimento dos dedos, em conjunto com um controle Wii Remote, para capturar a rotação do braço e um Kinect para capturar o movimento do braço de antebraço, o processamento foi realizado em linguagem C/C++ e os resultados obtidos são mostrados na tela do computador por meio de uma interface desenvolvida em Java.

Conti e Gonzaga (2011), utilizaram uma webcam conectada a um notebook para realizar a captura do movimento da mão em um fundo preto, por meio de um algoritmo de processamento da imagem para reconhecer a ponta dos dedos, interligados por um código desenvolvido no software Matlab.

Da mesma forma Freitas (2011), utilizou um celular da Samsung e um computador da IBM para rastrear o movimento da mão, no entanto, em um fundo claro e utilizando também, bibliotecas do software Matlab. Ambos os trabalhos de Conti, Gonzaga e Freitas, apresentaram limitantes quanto a robustez do sistema, com dificuldades de rastrear a mão em ambientes que não fossem totalmente controlados ou que apresentassem objetos com a cor similar à mão do usuário.

3 METODOLOGIA

Para o desenvolvimento deste estudo, foram empregados métodos de pesquisa e de testes a fim de atender o objetivo proposto: desenvolver um método para rastrear o movimento da mão. Primeiramente, realizou-se uma pesquisa bibliográfica, apresentando e explicando todos os tópicos importantes para o entendimento do desenvolvimento do projeto, em seguida definindo o método adequado para solucionar o problema. Após a escolha do método, testes foram realizados para validar as decisões e para finalizar, o projeto foi aplicado em um braço robótico a fim de validar a robustez e precisão do código desenvolvido.

Inicialmente, foram revisados artigos e estudos previamente realizados, a fim de contextualizar o problema e destacar as abordagens e aplicações diversas já propostas. Essa pesquisa permitiu identificar aspectos favoráveis e limitações nas metodologias anteriores, facilitando a definição de um método adequado para o desenvolvimento do projeto. Com base nessas análises, foram estabelecidos os requisitos e as limitações do projeto, detalhando-se a captura de imagens, o algoritmo implementado e os materiais utilizados.

Na fase de testes, foram realizados experimentos para avaliar a assertividade e a robustez do sistema, incluindo análises quantitativas com gráficos de desempenho para identificar o funcionamento ideal. Esses testes possibilitaram ajustes que aprimoraram a funcionalidade e a aplicabilidade do sistema.

Após os testes e análises, os resultados foram documentados, destacando os problemas remanescentes e possíveis melhorias. A apresentação dos resultados inclui uma análise detalhada da eficácia do método aplicado, examinando se o projeto desenvolvido conseguiu atingir os objetivos propostos. Além disso, as funcionalidades e limitações foram discutidas, visando propor adaptações e sugerir alternativas que contribuam para o aprimoramento do método.

3.1 DESCRIÇÃO DO SISTEMA PROPOSTO

Nesta seção, serão descritos os componentes que integram o sistema proposto, assim como as justificativas técnicas para cada seleção realizada. Para isso, cada elemento foi escolhido com base em critérios específicos, como compatibilidade e custo-benefício.

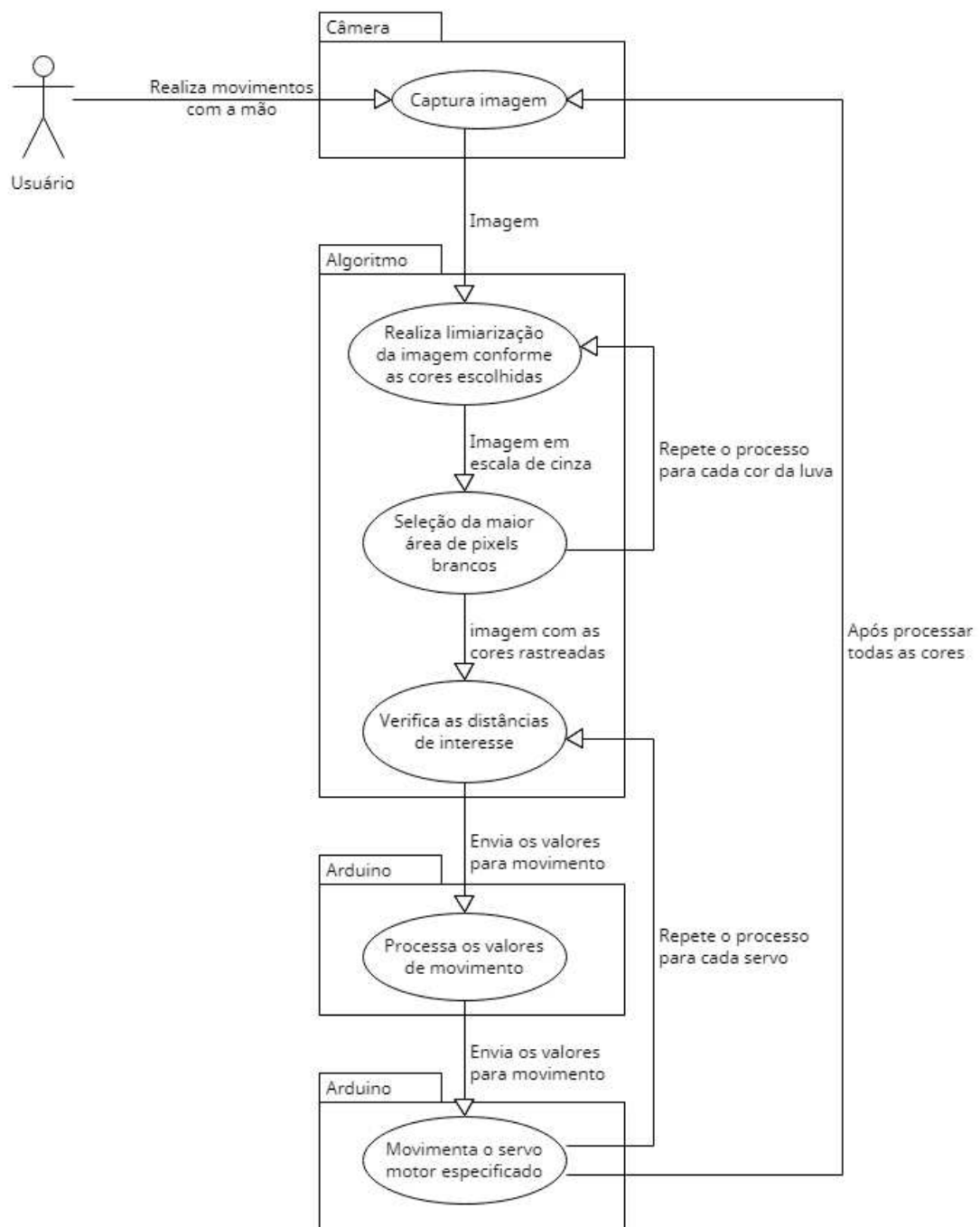
3.1.1 MÉTODO DE RASTREAMENTO

O método selecionado para realizar o rastreamento foi a combinação do reconhecimento de cores com a subtração de fundo. De acordo com a pesquisa

bibliográfica realizada, esses métodos demonstraram ser eficientes, proporcionando resultados satisfatórios sem a necessidade de equipamentos de alto custo. A utilização dessas técnicas permite o processamento de imagens de maneira eficiente, mesmo com recursos limitados.

A Figura 9 apresenta um fluxograma do funcionamento desejado do programa, partindo da aquisição de imagem, passando pelo processamento e resultando nos movimentos desejados.

Figura 9 – Fluxograma



Fonte: Próprio autor

Com o fluxograma é possível entender quais elementos são necessários para o funcionamento do sistema e buscando manter a lógica de processamento, é apresentada inicialmente a luva que deve ser utilizada pelo usuário.

3.1.2 SISTEMA DE TRANSMISSÃO DOS MOVIMENTOS DOS DEDOS: LUVA COLORIDA

O método de reconhecimento de cores foi escolhido para implementação do rastreamento dos movimentos. Portanto se torna necessário desenvolver uma luva capaz de transmitir, de forma simples e confiável, as informações de movimento dos dedos e da palma da mão de forma independente para o sistema de rastreamento que se baseia no reconhecimento de cores com a subtração de fundo. Essa solução visa garantir que o sistema funcione de maneira eficiente, possibilitando a detecção dos movimentos com precisão. As cores utilizadas na confecção da luva podem ser selecionadas de forma arbitrária, contudo, para facilitar o rastreamento e minimizar erros, é crucial optar por cores que apresentem valores significativamente distintos no espaço de cores HSV.

Embora a luva tenha sido adotada como a solução principal, outras alternativas também poderiam ser utilizadas. Por exemplo, seria possível pintar os dedos com cores específicas ou colar papéis coloridos nos mesmos. No entanto, para garantir a praticidade e facilitar o uso do sistema por qualquer usuário, foi optado por um modelo de luva. Essa abordagem oferece maior comodidade, pois a luva pode ser facilmente colocada e retirada da mão do usuário, tornando o processo de teste e operação do sistema mais acessível e ágil.

3.1.3 SISTEMA DE CAPTURA DE IMAGEM

Conforme indicado no fluxograma da Figura 9, é essencial um dispositivo para capturar a imagem que será processada pelo algoritmo. Nesse contexto, uma câmera simples, como as comumente presentes em computadores atuais, é suficiente para realizar essa função. Essas câmeras, embora não possuam alta resolução, oferecem capacidade adequada para capturar as imagens necessárias, desde que o ambiente e as condições de iluminação sejam controlados de forma apropriada.

A captura de imagem é realizada por uma webcam com resolução de 1280 x 720 pixels a 30 frames por segundo, com tecnologia de sensor semiconductor de óxido metálico complementar (CMOS), integrada no notebook Dell Inspiron 3583 (DELL, 2024).

3.1.4 SISTEMA DE PROCESSAMENTO DE IMAGEM

O sistema fornece os comandos necessários para o processamento das informações e coordena a interação com os demais componentes do sistema. Além disso, o computador realiza o armazenamento dos dados gerados e garante que os resultados sejam enviados para os dispositivos de exibição ou outros sistemas de saída.

Neste trabalho foi utilizado um notebook Dell Inspiron 3583, com um processador Intel Core I3 de oitava geração, com 2 núcleos e 4 threads, 64 bits, 16 MB Flash EPROM, 16 GB de Memória RAM 2400 MHz (DELL, 2024). Este modelo foi utilizado por ser o computador pessoal do autor.

3.1.5 SISTEMA DE SOFTWARE

Uma linguagem de programação que consiga oferecer os meios e o suporte necessário para o processamento de imagens, para que seja desenvolvido de uma maneira simples e confiável, em conjunto com um ambiente de programação com as mesmas características.

3.1.6 SISTEMA DE ATUAÇÃO MECÂNICA

O objetivo deste trabalho é realizar o rastreamento dos movimentos da mão, e para validar a rapidez e a assertividade na detecção dos movimentos, é necessário utilizar um atuador com certa complexidade. Esse atuador visa garantir a fidelidade dos movimentos rastreados, além de demonstrar a capacidade do sistema de atender a aplicações mais avançadas e complexas.

Neste trabalho foi utilizado um sistema de braço robótico com quatro graus de liberdade, visando simular uma aplicação em um sistema fabril, sendo possível realizar outros movimentos por meio da combinação de movimentos dos dedos da mão. O único limitador seria a complexidade da ferramenta que pode exigir mais familiaridade e precisão do usuário.

4 DESENVOLVIMENTO

Esta seção descreve o desenvolvimento completo do sistema de rastreamento do movimento da mão, abordando as etapas de montagem, os algoritmos aplicados e as ferramentas de teste utilizadas para a validação do método proposto. A seleção da metodologia ideal foi fundamentada na análise de métodos utilizados em trabalhos anteriores, o que permitiu iniciar o processo de construção e implementação do sistema de forma eficiente e alinhada aos objetivos do projeto.

A implementação teve início com a confecção de uma luva adaptada, etapa crucial para garantir a precisão do sistema. Após essa fase, foi realizada a montagem do sistema mecatrônico, que integra os componentes eletrônicos e os sensores específicos para a captura dos movimentos. A programação do algoritmo de rastreamento foi executada no ambiente Visual Studio Code, utilizando a linguagem Python, devido à familiaridade do autor com a linguagem e à compatibilidade com a biblioteca OpenCV2, amplamente utilizada para processamento de imagens. Além disso, o código necessário para a comunicação com o hardware foi desenvolvido no Arduino IDE, utilizando a linguagem C++.

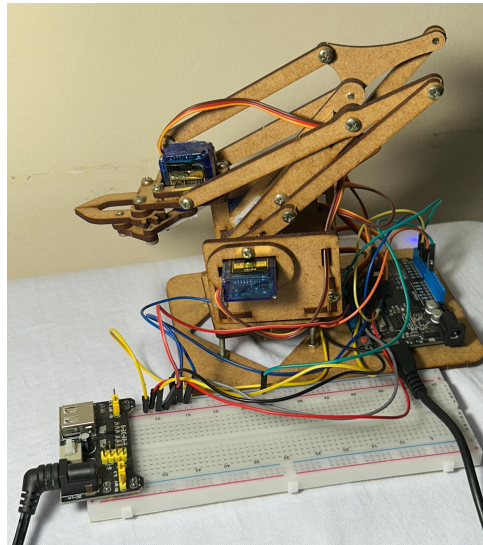
Após a montagem do sistema, foram implementados diversos testes para avaliar a qualidade e o desempenho do rastreamento. Os testes incluíram a medição da velocidade de resposta do sistema, sendo esta uma característica fundamental para garantir a responsividade e a agilidade durante a operação. A capacidade de resposta rápida é essencial, especialmente em aplicações que exigem interações em tempo real, como o rastreamento de movimentos.

Além disso, foram realizados testes de robustez para verificar a estabilidade do sistema sob variações de iluminação incidente na luva, fator crítico para o sucesso do rastreamento. A variabilidade na iluminação pode impactar diretamente na precisão da captura e análise das imagens, tornando esses testes indispensáveis.

4.1 DESENVOLVIMENTO E PROTOTIPAGEM DOS SISTEMAS

O sistema foi projetado com foco em um projeto de baixo custo, utilizando em grande parte elementos de propriedade pessoal do autor. No entanto, todos os valores e detalhes dos componentes utilizados serão devidamente explicitados ao longo da descrição. Essa abordagem visa tornar o sistema acessível, sem comprometer sua funcionalidade, permitindo que seja replicado e testado com recursos limitados. O sistema físico montado está apresentado na Figura 10.

Figura 10 – Protótipo do sistema



Fonte: Próprio autor.

Pode ser dividido em três estágios principais: o desenvolvimento da luva colorida, a montagem mecânica do atuador e a montagem eletrônica do sistema. A integração desses componentes exigiu a implementação de três algoritmos distintos. O primeiro algoritmo foi desenvolvido para obter as informações de cores necessárias para o rastreamento. O segundo, focado no rastreamento da luva, processa as imagens capturadas para identificar os movimentos. Por fim, o terceiro algoritmo foi responsável pela movimentação dos motores do braço robótico, traduzindo as informações de rastreamento em comandos de movimento.

4.1.1 DESENVOLVIMENTO DA LUVA COLORIDA

Para possibilitar a transmissão das informações necessárias ao rastreamento, foi confeccionada uma luva com papéis coloridos posicionados em pontos estratégicos da mão, foram utilizados quatro papéis coloridos de diferentes cores, a cor verde envolveu o polegar, a cor azul envolveu o dedo indicador, a cor amarela envolveu o dedo médio e por fim a cor rosa foi presa na palma da mão, como pode ser observado na Figura 11.

Figura 11 – Luva



Fonte: Próprio autor.

A luva utilizada em questão pode ser encontrada por R\$ 14,90 na internet, no entanto qualquer luva preta que cubra a mão por completo pode ser utilizada. Para os papéis coloridos foi utilizado um papel de cartolina colorido, em que uma folha de 30cm x 65cm custa R\$ 0,50, porém para cada dedo foi utilizada uma pequena área de 7cm x 9 cm, custando pouco menos de R\$ 0,02 por dedo.

A escolha das cores para a luva foi realizada com base na distância entre elas no espaço de cores HSV, no valor de Matiz (Hue), que varia de 0 a 360 graus. As cores foram escolhidas para que estivessem suficientemente distantes no espectro de Matiz, de modo a garantir uma fácil distinção entre elas durante o rastreamento. Assim, as cores inicialmente escolhidas foram vermelho (Matiz = 0), verde (Matiz = 90), azul (Matiz = 180) e rosa (Matiz = 270), que apresentam uma separação adequada no espaço de cores.

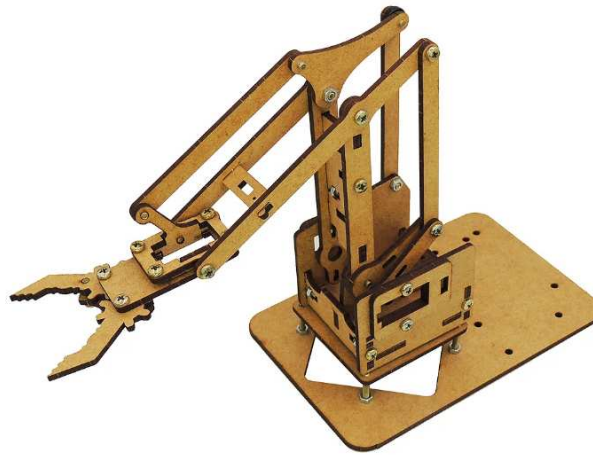
No entanto, durante os testes intermediários, o sistema teve dificuldade em distinguir a cor vermelha da cor da pele, o que comprometeria a precisão do rastreamento. Para resolver esse problema, a cor amarela (Matiz = 45) foi escolhida para substituir o vermelho. Essa alteração garantiu maior confiabilidade no desempenho do sistema.

4.1.2 DESENVOLVIMENTO MECÂNICO DO BRAÇO ROBÓTICO

O braço robótico escolhido para servir como atuador foi um braço robótico de MDF, ele apresenta 15cm de altura e o braço estendido possui aproximadamente 20cm,

encontrado por R\$ 26,99 no site da BR Eletrônica, apresentado na Figura 12.

Figura 12 – Braço Robótico



Fonte: (BR ELETRÔNICA, 2024)

Para o braço robótico há um guia de montagem para compreender a sequência de engates e parafusamentos que deve ser realizada, com uma chave do tipo Phillips é possível montar todas as partes.

4.1.3 DESENVOLVIMENTO ELETRÔNICO DO BRAÇO ROBÓTICO

O braço robótico é capaz de realizar quatro movimentos independentes: girar no próprio eixo, estender o braço para frente, estender o braço para cima e abrir e fechar a garra. Para a execução de cada um desses movimentos, são utilizados quatro servo motores, sendo um dedicado a cada tipo de movimento. A ativação dos motores é controlada por um Arduino Uno, que coordena as operações de movimentação com base nos comandos recebidos do sistema de rastreamento.

Para garantir o fornecimento de energia adequado ao sistema, foi utilizada uma fonte ajustável para protoboard, conectada a uma fonte de tomada. Os cabos de conexão completam o sistema, permitindo a interligação entre os componentes eletrônicos. A Tabela 1 apresenta o nome dos componentes utilizados no sistema, juntamente com seus respectivos valores, detalhando os elementos que compõem a infraestrutura necessária para o funcionamento do braço robótico e a comunicação entre os componentes.

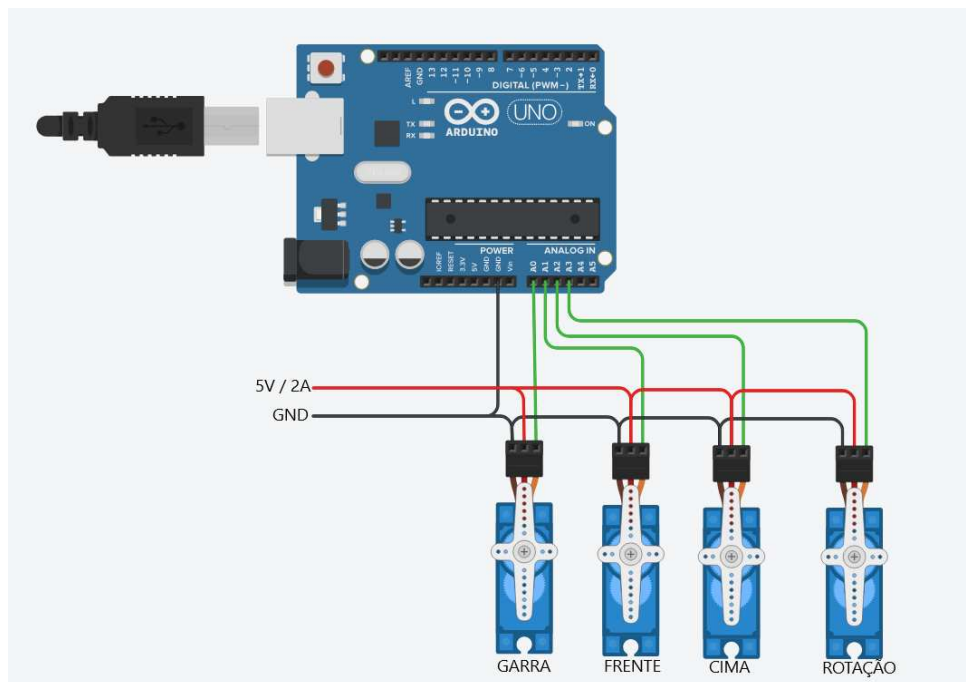
Tabela 1 – Componentes do sistema eletrônico.

Componente	Valor
Arduino Uno R3	R\$40,00
Protoboard 400 Pontos	R\$10,00
Fonte Dc P/ Protoboard In 6v-9v Out 5 Ou 3.3 V	R\$10,00
Fonte Plastica 9V 2A	R\$20,00
4 x Micro Servo Sg90 9g Towerpro	R\$60

Fonte: Próprio autor.

A Figura 13 mostra como foram realizadas as conexões elétricas dos componentes, as conexões de energia de 5V e GND são realizadas com a fonte ajustável conectada na protoboard.

Figura 13 – Circuito Eletrônico



Fonte: Próprio autor.

Conectando todos os componentes mecânicos e eletrônicos o sistema já está pronto para receber comandos e ser movimentado, portanto a próxima etapa é definir e implementar os algoritmos de rastreamento e comunicação entre os componentes.

4.2 ALGORITMOS PROPOSTOS

O desenvolvimento da programação do sistema foi dividido em três etapas distintas, cada uma abordando uma parte específica do processo de rastreamento e

controle. Na primeira etapa, foi implementado um código para a obtenção da faixa de valores para cada componente do espaço de cores HSV. Esse código foi responsável por definir os limites de Matiz, Saturação e Valor necessários.

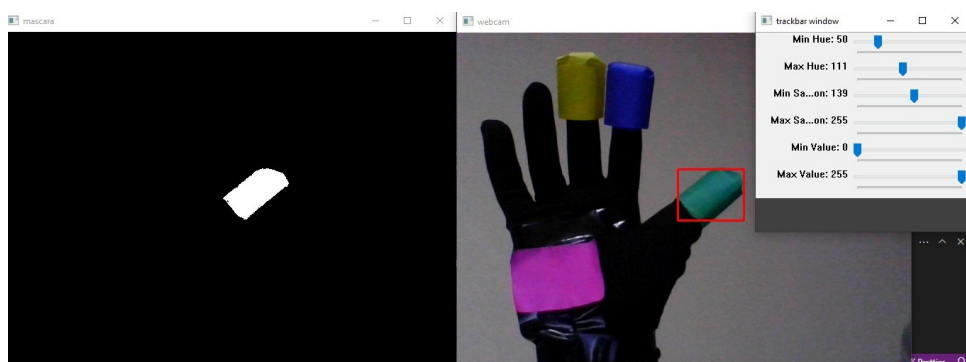
A segunda etapa envolveu o desenvolvimento do código do algoritmo de rastreamento, que processa as imagens capturadas pela câmera e identifica a posição dos papéis coloridos na luva. Além disso, foi implementado o código de comunicação serial com o Arduino, permitindo que as informações de movimento fossem enviadas ao microcontrolador para controlar o atuador robótico.

A terceira e última etapa consistiu no código responsável pela movimentação dos servo motores do braço robótico. Esse código traduz os dados recebidos do rastreamento em comandos específicos para os motores, permitindo a execução dos movimentos desejados.

4.2.1 OBTENÇÃO DE VALORES HSV

A limiarização da imagem requer a definição dos valores dos componentes do espaço HSV para estabelecer os limiares de cada cor utilizada no rastreamento. Para isso, são necessários os limites superiores e inferiores de Matiz, Saturação e Valor para cada cor, de modo a garantir que o algoritmo consiga identificar com precisão as diferentes partes da luva. Esses valores são obtidos por meio de testes empíricos, realizados utilizando o código desenvolvido especificamente para esse propósito, estes testes foram realizados variando iluminação incidente, fundo da imagem e ângulação da mão. A Figura 14 apresenta as janelas que são mostradas no computador ao executar o código.

Figura 14 – Obtenção dos valores HSV



Fonte: Próprio autor.

São abertas três janelas simultaneamente, uma janela com seis controles deslizantes, uma para cada limite superior e inferior dos componentes, em que é possível selecionar o valor desejado para ser utilizado no limiar, a segunda janela mostra a imagem em tempo real capturada com a câmera, possuindo um quadrado ao redor da maior área de pixels brancos e a janela com a limiarização aplicada, mostrando

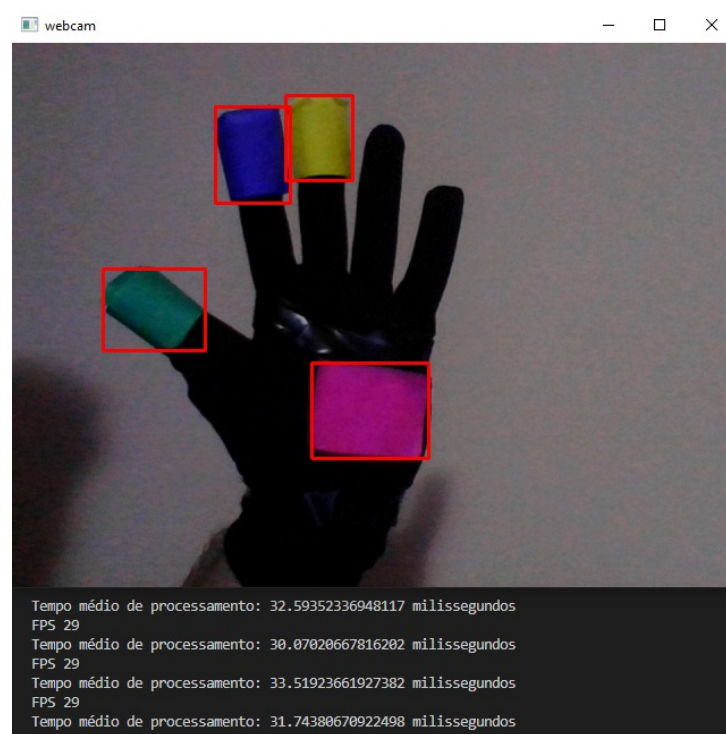
a imagem processada e explicita as áreas rastreadas.

4.2.2 RASTREAMENTO

Com os limiares definidos, o processo de rastreamento da mão pode ser iniciado, seguindo o fluxograma apresentado na Figura 9. O código pode ser dividido em quatro etapas principais, sendo a primeira a captura da imagem. O código solicita à webcam que continue capturando as imagens e as salve em um buffer, permitindo que os quadros sejam utilizados ao longo do programa para o processamento contínuo.

Após a captura, são realizadas as limiarizações das imagens. Como o sistema utiliza quatro cores diferentes, o código executa quatro limiarizações de forma sequencial, gerando quatro imagens em escala de cinza, cada uma representando uma cor específica. Essas imagens são processadas separadamente, as quais passam pela verificação da maior área de pixels brancos. Ao localizar essa área, são obtidas as coordenadas dos pixels que a compõem, e um quadrado é desenhado ao redor da região detectada. Esse quadrado não só serve para fornecer uma visualização clara da área rastreada para o usuário, mas também é utilizado como referência na próxima etapa do processamento. A Figura 15 ilustra a janela do programa em execução, mostrando os quadrados de rastreamento em torno das áreas de interesse.

Figura 15 – Janela de rastreamento



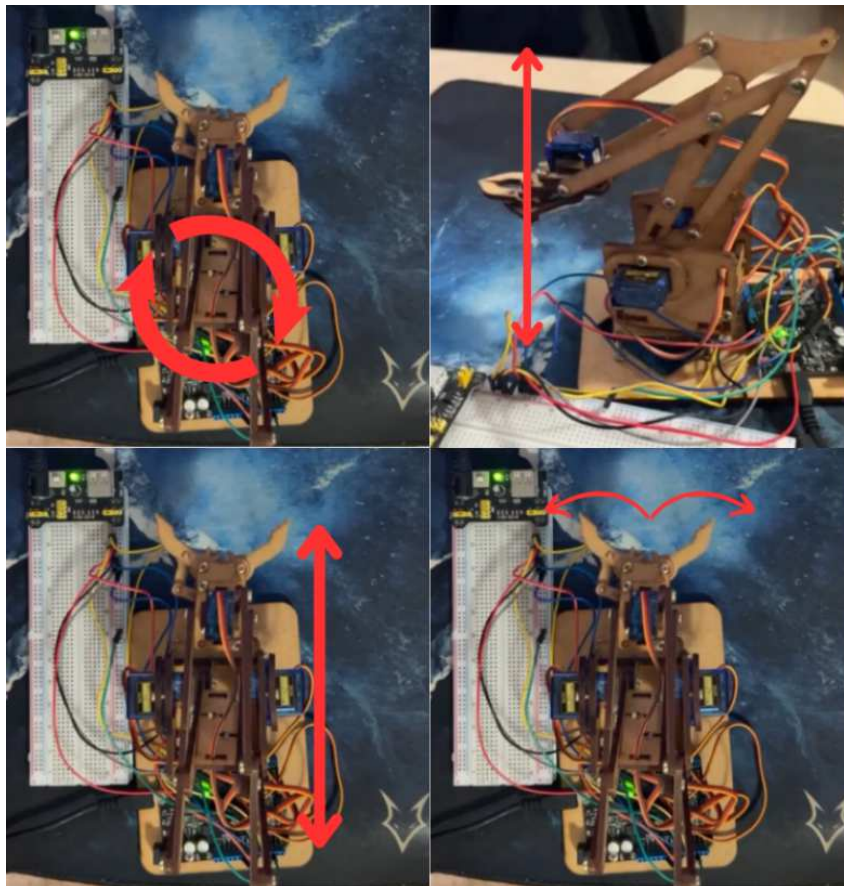
Fonte: Próprio autor.

Os motores do atuador são controlados conforme as distâncias relativas entre as cores da luva, sendo utilizadas as informações dos quadrados resultantes do

rastreamento para determinar as movimentações dos servo motores. As coordenadas dos centros de cada quadrado permitem a identificação das ações a serem executadas pelo braço robótico.

Para que o braço rotacione no próprio eixo, é verificada a distância entre o quadrado rosa e o limite esquerdo da imagem capturada. Para os movimentos de subida e descida do braço, é calculada a distância entre o quadrado rosa e o limite superior da imagem, enquanto para os movimentos de avanço e recuo do braço, é verificada a distância entre os quadrados amarelo e rosa. Por fim, para o controle da pinça do braço robótico, é medida a distância entre os quadrados azul e verde, imitando o movimento de pinça com os dedos da mão. Essas distâncias são calculadas, ajustadas para a escala desejada e armazenadas em buffers. A figura 16 apresenta todos os movimentos que o braço robótico pode realizar.

Figura 16 – Movimentos do braço robótico



Fonte: Próprio autor.

Com as distâncias de interesse definidas, é realizada a comunicação serial com o Arduino, transmitindo o número do motor que deve ser acionado, seguido pelo valor do ângulo de rotação. Essa comunicação é feita de forma sequencial, começando pelo motor da base e finalizando no motor da garra. O código completo, que detalha a lógica e o funcionamento desse processo, está apresentado no Apêndice A.

4.2.3 MOVIMENTAÇÃO DO BRAÇO ROBÓTICO

Ao receber a mensagem contendo as informações de movimentação enviadas pelo código de rastreamento, a função do código do Arduino é direcionar os valores corretos de ângulo para cada motor. Após definir as saídas analógicas do Arduino, conforme o circuito apresentado na Figura 13, o código do Arduino processa a mensagem recebida, interpretando a informação sobre qual motor deve ser acionado. O valor do ângulo, que foi calculado pelo código de rastreamento, é então enviado ao servo motor especificado, que realiza o movimento correspondente. Esse processo é realizado de forma sequencial, garantindo que cada motor execute seu movimento de maneira precisa e coordenada, permitindo o controle adequado do braço robótico.

5 RESULTADOS

Esta seção apresenta os resultados obtidos a partir dos testes realizados, seguidos de uma discussão sobre suas implicações.

5.1 RESULTADOS DOS TESTES

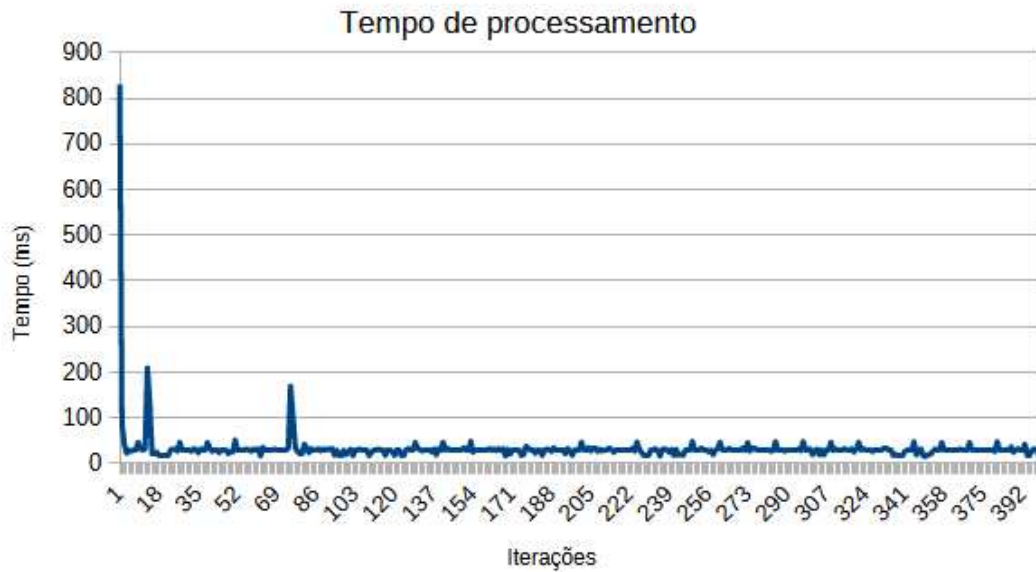
Os resultados obtidos por meio dos testes práticos são fundamentais para avaliar o sucesso do projeto e identificar possíveis melhorias na implementação do método ou nas escolhas realizadas. Para essa análise, foram conduzidos dois testes principais. O primeiro teste focou na velocidade de processamento, com o objetivo de medir o atraso entre o movimento da mão e a resposta correspondente do braço robótico, avaliando a responsividade do sistema.

O segundo teste foi voltado para a robustez à luz, buscando identificar os limites de operação do sistema sob diferentes condições de iluminação. Esse teste permitiu verificar a eficácia do rastreamento em situações adversas, como variações na intensidade e direção da luz incidente sobre a luva, além de avaliar a assertividade do sistema em distinguir as cores utilizadas. Os resultados desses testes forneceram informações essenciais para validar a viabilidade do sistema, bem como para apontar ajustes necessários na metodologia ou nos componentes utilizados.

5.1.1 AVALIAÇÃO DA VELOCIDADE DE PROCESSAMENTO

O teste foi conduzido para analisar a velocidade de processamento do sistema como um todo, abrangendo todas as etapas, desde a captura da imagem até a execução do movimento pelo braço robótico. Para isso, foi medida a diferença de tempo entre cada ação, permitindo avaliar o desempenho integrado do sistema, incluindo o rastreamento, a comunicação serial e o controle dos motores. A Figura 17 apresenta o tempo de rastreamento em quatrocentos ciclos do programa.

Figura 17 – Tempo de processamento



Fonte: Próprio autor.

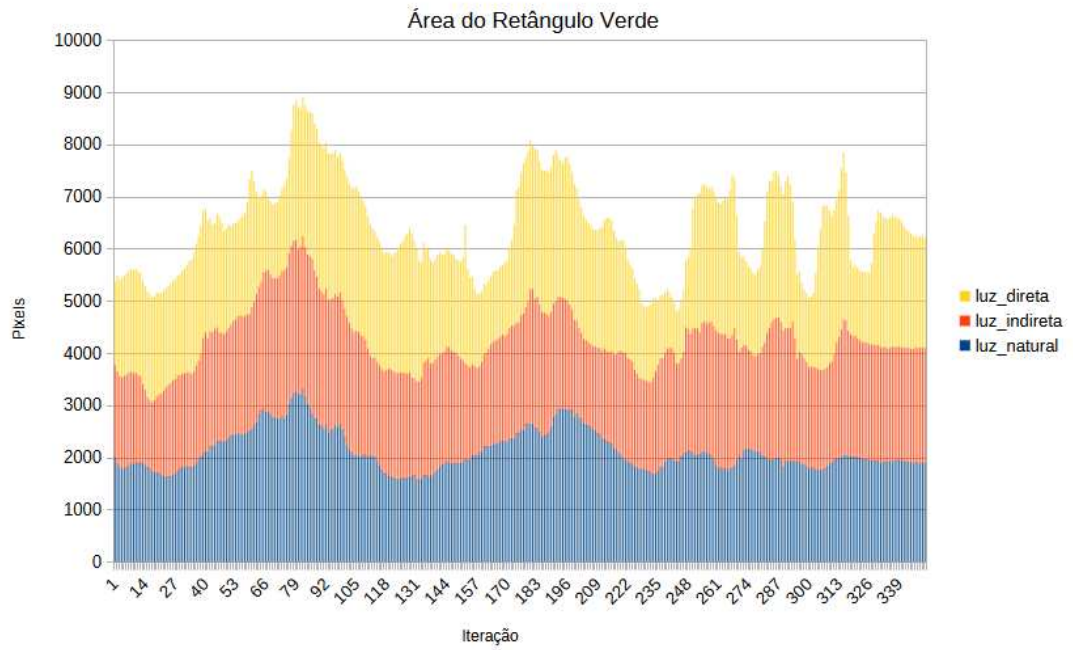
Os resultados foram obtidos por meio de testes repetidos sob diferentes condições de iluminação e variações nos movimentos da mão. Esse procedimento teve como objetivo simular cenários variados e identificar possíveis oscilações críticas que pudessem impactar o desempenho do projeto.

5.1.2 AVALIAÇÃO DA ROBUSTEZ À VARIAÇÃO DA LUZ

Este teste é essencial para determinar o ambiente ideal para a operação do sistema, além de avaliar a confiabilidade do rastreamento das cores na luva. Foram realizadas três variações de luminosidade, mantendo o mesmo padrão de movimento da mão em todos os casos. O primeiro cenário foi testado com apenas luz ambiente, sem iluminação direta na luva. No segundo, foi utilizada uma luz indireta, direcionada para a mesa em frente ao sistema. Por fim, no terceiro teste, uma luz direta foi projetada sobre a luva.

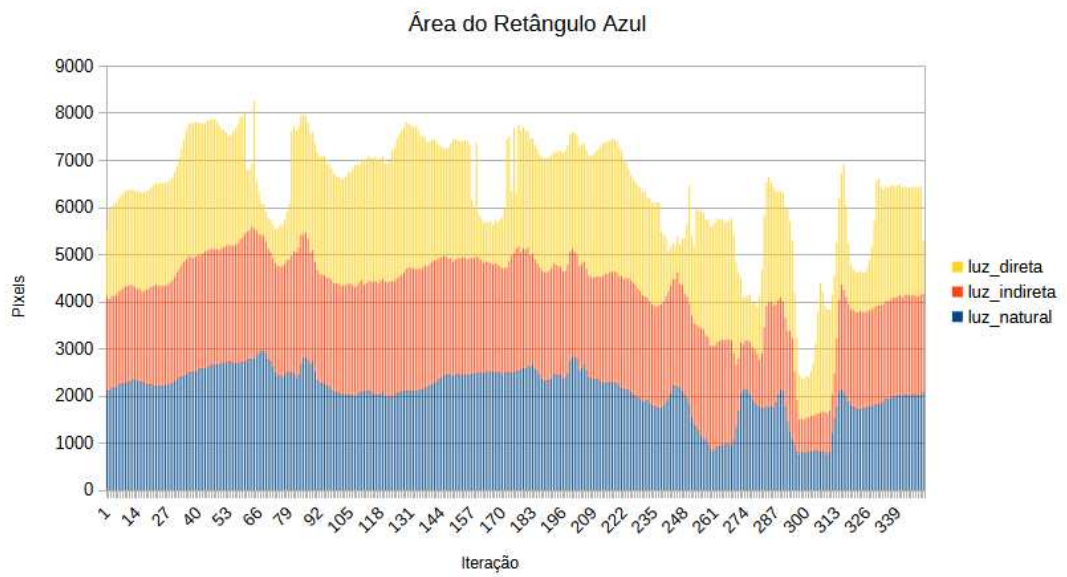
Os resultados obtidos para cada cor nas diferentes condições de iluminação estão apresentados nos gráficos correspondentes, nas Figuras 18, 19, 20 e 21. Essas análises permitem observar a influência das variações de luminosidade na precisão e estabilidade do rastreamento.

Figura 18 – Rastreamento da cor verde



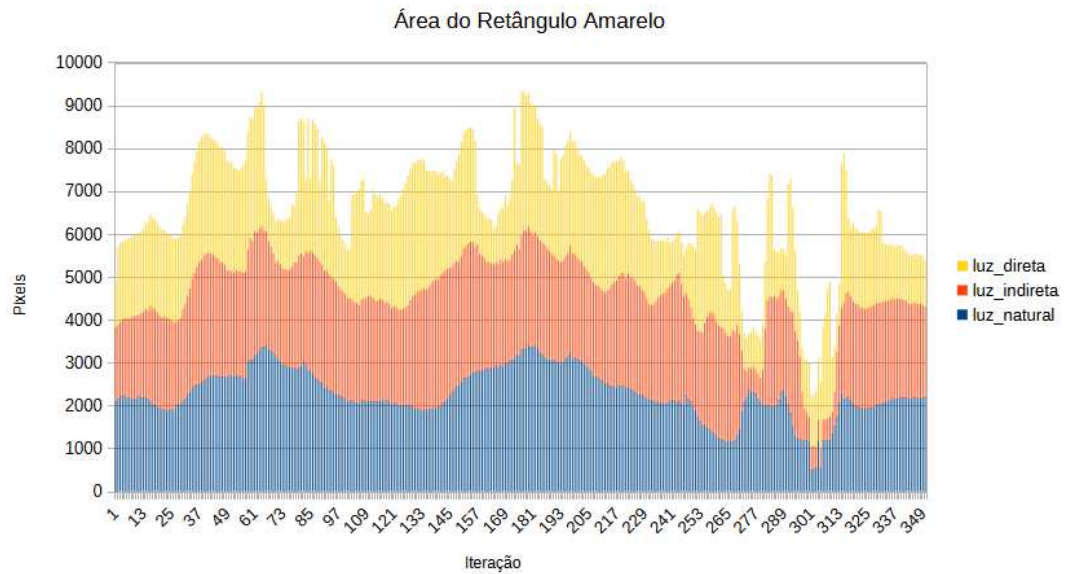
Fonte: Próprio autor.

Figura 19 – Rastreamento da cor azul



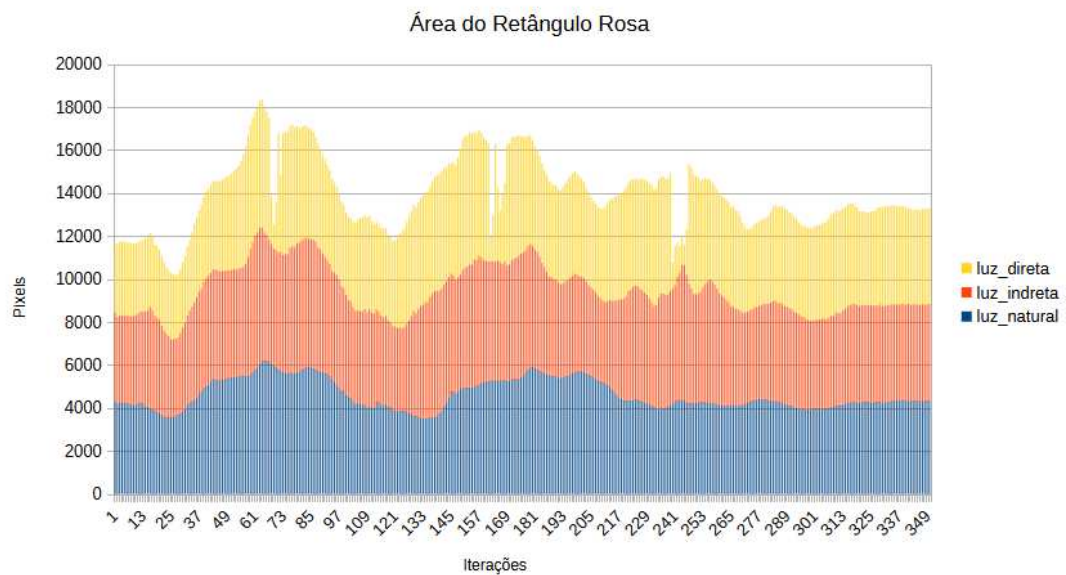
Fonte: Próprio autor.

Figura 20 – Rastreamento da cor amarelo



Fonte: Próprio autor.

Figura 21 – Rastreamento da cor rosa



Fonte: Próprio autor.

Para que seja possível analisar as variações de área conforme a luminosidade os dados obtidos foram empilhados em cada um dos gráficos, apresentando assim os diferentes comportamentos seguindo os diferentes movimentos realizados.

5.2 DISCUSSÃO DOS RESULTADOS

Neste capítulo são discutidos os resultados obtidos por meio dos testes realizados, e a partir desta análise são elencados os fatores limitantes para o sistema.

5.2.1 INTERPRETAÇÃO DOS DADOS OBTIDOS NOS TESTES

Os testes de velocidade de processamento indicaram que o sistema é capaz de realizar o rastreamento e concluir todas as etapas com um atraso adequado. O tempo de processamento médio se manteve aproximadamente em 32,6 milissegundos para o ciclo completo, desde a captura da imagem até a execução do movimento pelo braço robótico, mesmo considerando os picos no início do rastreamento, que são ocasionados pela escolha do hardware. Em ações que necessitam de fluidez esse fator se torna um dos mais importantes, pois pode ser o diferencial entre o sucesso e o fracasso do trabalho que se deseja realizar.

Os resultados dos testes de robustez apresentam a sequência de dados obtidos durante diferentes movimentos, permitindo identificar a iluminação ideal para o funcionamento do sistema. A análise foi realizada com base nos picos positivos e negativos das curvas correspondentes, em que a menor variação na área rastreada indica a condição de iluminação mais favorável para o rastreamento eficaz.

Os dados mostram que a utilização apenas da iluminação natural do ambiente resultou em um desempenho superior. A iluminação artificial direta na luva causou reflexos nos papéis coloridos, comprometendo a precisão do reconhecimento das cores. Esse comportamento evidencia a sensibilidade do sistema a condições específicas de luz.

5.2.2 LIMITAÇÕES DO SISTEMA DESENVOLVIDO

A análise dos resultados permite compreender e estabelecer os limites do sistema de rastreamento, sendo o primeiro fator limitante relacionado ao hardware. Nesse contexto, é fundamental realizar uma avaliação do equipamento utilizado no computador responsável pelo processamento. A proximidade das especificações de hardware com aquelas do computador utilizado no desenvolvimento deste trabalho garante maior similaridade nos valores de processamento apresentados.

No aspecto de iluminação, observou-se que a redução na incidência luminosa direta sobre a luva contribui significativamente para o desempenho do rastreamento. Dessa forma, a utilização de ambientes com iluminação controlada e indireta direcionada à luva melhora a precisão do sistema, reduzindo reflexos indesejados que dificultam o reconhecimento das cores. Essas condições otimizadas asseguram maior estabilidade e assertividade no rastreamento, especialmente em aplicações práticas.

6 CONCLUSÕES

Diante do aumento da adoção de tecnologias no ambiente empresarial e da crescente incidência de acidentes envolvendo as mãos no Brasil, este trabalho teve como objetivo desenvolver um sistema de rastreamento de movimentos das mãos e transmissão a um braço robótico, com aplicação direcionada a ambientes industriais. A proposta visa oferecer uma solução tecnológica que possa contribuir para a segurança e a automação nos processos industriais, alinhando inovação e funcionalidade.

O sistema proposto foi embasado em uma revisão da literatura, analisando métodos utilizados e resultados obtidos em trabalhos anteriores. Com base nessa análise, foi possível selecionar o método mais adequado para resolver o problema proposto. Os testes conduzidos tiveram como finalidade validar as decisões tomadas durante o desenvolvimento, confirmando a funcionalidade do sistema e sua capacidade de atender às exigências práticas e operacionais do contexto industrial.

Os objetivos estabelecidos foram alcançados com sucesso, considerando as limitações identificadas. O sistema mostrou boa estabilidade no rastreamento e na transmissão dos movimentos, operando em um intervalo de tempo aceitável para interação humana. Essa característica viabiliza sua implementação em diversas áreas da indústria e da segurança, com o intuito de preservar a integridade dos usuários e aumentar a eficiência operacional. Apesar dos resultados satisfatórios, as limitações observadas desempenham um papel crucial na confiabilidade do sistema, podendo ocasionar variações que impactem seu desempenho.

Em síntese, este trabalho apresentou o desenvolvimento de um sistema completo de rastreamento e transmissão dos movimentos, baseado na interação com um braço robótico, utilizando materiais de baixo custo. A proposta visa atender às demandas de segurança operacional, ao mesmo tempo em que proporciona uma alternativa economicamente viável para empresas, contribuindo para a redução de custos.

6.1 POSSÍVEIS MELHORIAS

Embora o trabalho tenha alcançado seus objetivos ao desenvolver um sistema de rastreamento do movimento da mão, existem aspectos do projeto que podem ser aprimorados para otimizar a usabilidade e a precisão do sistema. A escolha das cores, por exemplo, desempenha um papel fundamental no desempenho do rastreamento, impactando diretamente o sistema em diferentes condições de iluminação e fundo.

Futuras pesquisas poderiam incluir uma análise mais aprofundada das cores e dos materiais utilizados para compor os marcadores, considerando sua interação com diferentes tipos de fundo e iluminação. Esse estudo permitiria identificar combinações

de cores e materiais que apresentem menores variações e maior estabilidade no rastreamento, resultando em um sistema ainda mais robusto e confiável para diferentes cenários de uso.

Outra melhoria relevante seria a análise da variação do hardware utilizado e do sistema de aplicação para qual são transmitidos os movimentos da mão. A realização de um estudo detalhado sobre o impacto do hardware na velocidade e no desempenho geral do sistema pode resultar em melhorias tanto no custo quanto na agilidade do processamento, tornando o sistema acessível e funcional. Além disso, a variação do sistema de aplicação pode revelar pontos adicionais de melhoria no método, considerando diferentes arquiteturas de software ou ajustes na interface de usuário.

REFERÊNCIAS

AGOSTINHO, I. A. **Combinação de dispositivos de baixo custo para rastreamento de gestos**. 2014. Dissertação (Mestrado em Ciência da Computação) — Faculdade de Ciências e Tecnologia, Universidade Estadual Paulista Júlio de Mesquita Filho, Bauru, 2014.

ARAÚJO, L. d. L. **Colorização Automática de Imagens em Escala de Cinza Utilizando Deep Learning**. 2019. Dissertação (Bacharel em Engenharia Elétrica) — Centro de Engenharia Elétrica e Informática - Universidade Federal de Campina Grande, Campina Grande - Paraíba, 2019.

BERTHOLDO, F. A. R. **Técnicas de Limiarização para Melhorar a Qualidade Visual de Documentos Históricos**. 2007. Dissertação (Mestrado em Ciência da Computação) — Instituto de Ciências Exatas - Universidade Federal de Minas Gerais, Belo Horizonte - Minas Gerais, 2007.

BOWMAN, D. A. et al. **3D User Interfaces: Theory and practice**. Boston: Addison-Wesley, 2004.

BR ELETRÔNICA. **Kit Braço Robótico em MDF + Parafusos**. 2024. Disponível em: https://www.breletronica.com.br/kit-braco-robotico-em-mdf-parafusos?utm_source=Site&utm_medium=GoogleShopping&utm_campaign=IntegracaoGoogle&gad_source=1&gclid=Cj0KCQiA_qG5BhDTARIsAA0UHSIN4Zk-t5EFHqtoga7_fMFegOP7EC3HnoRAm0CH8Hq_oM1DE3TGiRoaAhRZEALw_wcB. Acesso em: 04 novembro. 2024.

CONFEDERAÇÃO NACIONAL DA INDÚSTRIA. **Indústria 4.0: 5 anos depois**. Brasília, 2022. Disponível em: https://static.portaldaindustria.com.br/media/filer_public/cd/a2/cda22223-5c33-4a5f-af4e-f5a5d64b3d85/sondespecial_industria40_cincoanosdepois_abril2022.pdf. Acesso em: 08 mai. 2024.

CONTI, C.; GONZAGA, A. Rastreamento do movimento dos dedos da mão humana em imagens de vídeo. **Learning and Nonlinear Models**, v. 9, n. 2, p. 104–119, 2011.

DELL. **Especificações do Inspiron 3583**. 2024. Disponível em: <https://www.dell.com/support/manuals/pt-br/inspiron-15-3583-laptop/inspiron-3583-1sp-setup-and-specifications/especificaÃ§Ã—es-do-inspiron-3583?guid=guid-7c9f07ce-626e-44ca-be3a-a1fb036413f9&lang=pt-br>. Acesso em: 03 novembro. 2024.

FARIAS, R. D. **Espaço de cores, morfologia matemática e extração de feições**. 2016. Dissertação (Mestrado em Ciências Geodésicas e Tecnologias da Geoinformação) — Centro de tecnologia e geociências - Universidade Federal de Pernambuco, Recife, 2016.

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

FREITAS, R. F. **Sistema de rastreamento da mão humana utilizando visão artificial para aplicações embarcadas**. 2011. Dissertação (Mestrado em Engenharia de

Teleinformática) — Departamento de Engenharia de Teleinformática - Universidade Federal do Ceará, Fortaleza, 2011.

GONZALES, R. C.; WOODS, R. E. **Digital Image Processing**. 3. ed. Nova Jersey, EUA: Pearson Education, Inc., 2008.

KOIKE, H.; SATO, Y.; KOBAYASHI, Y. Integrating paper and digital information on enhanceddesk: a method for realtime finger tracking on an augmented desk system. **ACM Transactions on Computer-Human Interaction**, v. 8, n. 4, p. 307–322, 2001.

LOAIZA, M. E. et al. Sistemas de rastreamento de baixo custo para desktops. **XXV Congresso da Sociedade Brasileira de Computação**, Rio de Janeiro, p. 1676–1690, jul. 2005.

MINISTÉRIO DA PREVIDÊNCIA SOCIAL. **Quantidade de acidentes do trabalho, por situação do registro e motivo, segundo os 200 códigos da Classificação Internacional de Doenças - CID-10 mais incidentes, no Brasil - 2022**. 2022. Disponível em: https://www.gov.br/previdencia/pt-br/assuntos/previdencia-social/saude-e-seguranca-do-trabalhador/acidente_trabalho_incapacidade/arquivos/copy2_of_AEAT_2022/secao-i-estatisticas-de-acidentes-do-trabalho/subsecao-c-acidentes-do-trabalho-segundo-a-cid/capitulo-57-brasil-e-grande-regioes. Acesso em: 08 mai. 2024.

MOURA, J. C. d. C. **Técnicas de limiarização em imagens de retinografia**. 2018. Dissertação (Bacharel em Engenharia Biomédica) — Centro de Tecnologia - Universidade Federal do Rio Grande do Norte, Natal - Rio Grande do Norte, 2018.

SANTOS, T. N. **Detecção e rastreamento da mão utilizando dados de profundidade**. 2015. Dissertação (Mestrado em Engenharia Mecatrônica) — Escola Politécnica - Universidade Federal da Bahia, Salvador, 2015.

SILVA, S. R. **Reconhecimento de gestos customizados da mão em tempo real usando aprendizado de máquina e grafos de ação**. 2017. Dissertação (Mestrado em Matemática) — Instituto de Matemática - Universidade Federal de Alagoas, Maceió, 2017.

TORI, R.; KIRNER, C.; SISCOOTTO, R. **Fundamentos e tecnologia de realidade virtual e aumentada**. Belém: Sociedade Brasileira de Computação, 2006.

TRIBUNAL SUPERIOR DO TRABALHO - JUSTIÇA DO TRABALHO. **O que é acidente de trabalho?** 2022. Disponível em: <https://tst.jus.br/web/trabalhoseguro/o-que-e-acidente-de-trabalho>. Acesso em: 09 out. 2024.

APÊNDICE A - CÓDIGO DE RASTREAMENTO EM PYTHON

```

1 import numpy as np
2 import cv2
3 import math
4 import serial
5 import time
6 import csv
7
8 #===== Tratamento da Imagem =====
9 #Seleciona a camera para capturar a imagem
10 cap = cv2.VideoCapture(0)
11
12 #Cria as coordenadas para os dedos
13 coordXcentros = [0,0,0,0]
14 coordYcentros = [0,0,0,0]
15
16 cores = {
17     #'Nome' : {'lowerColor': np.array([hueMin, satMin, valMin]), '
18     upperColor': np.array([hueMax, satMax, valMax])}
19     'Verde' : {'lowerColor': np.array([50,140,0]), 'upperColor': np.
20     array([110,255,255])},
21     'Azul' : {'lowerColor': np.array([115,135,35]), 'upperColor': np.
22     array([135,255,255])},
23     'Amarelo' : {'lowerColor': np.array([20,100,0]), 'upperColor': np.
24     array([60,255,255])},
25     'Rosa' : {'lowerColor': np.array([145,35,100]), 'upperColor': np.
26     array([175,255,255])},
27 }
28
29 #Realiza o tratamento da imagem
30 def computeTracking(frame):
31     aux = 0
32
33     #Transforma a imagem de RGB para HSV
34     hsvImage = cv2.cvtColor(frame, cv2.COLOR_BGR2HSV)
35
36     #Cria as mscaras para cada cor
37     mascaraVerde = cv2.inRange(hsvImage, cores['Verde']['lowerColor'],
38     cores['Verde']['upperColor'])
39     mascaraAzul = cv2.inRange(hsvImage, cores['Azul']['lowerColor'],
40     cores['Azul']['upperColor'])
41     mascaraAmarelo = cv2.inRange(hsvImage, cores['Amarelo']['lowerColor']
42     ], cores['Amarelo']['upperColor'])
43     mascaraRosa = cv2.inRange(hsvImage, cores['Rosa']['lowerColor'],
44     cores['Rosa']['upperColor'])

```

```

36     mascaras = np.array([mascaraVerde, mascaraAzul, mascaraAmarelo,
37     mascaraRosa])
38
39     cntFiltro = 0
40     areaVerde = 0
41     areaAzul = 0
42     areaVermelha = 0
43     areaRosa = 0
44
45     for filtro in mascaras:
46         #Aplica mascara na imagem
47         result = cv2.bitwise_and(frame, frame, mask = filtro)
48
49         #Aplica limiariza o para o cinza
50         gray = cv2.cvtColor(result, cv2.COLOR_BGR2GRAY)
51         _,gray = cv2.threshold(gray, 0, 255, cv2.THRESH_BINARY | cv2.
52         THRESH_OTSU)
53
54         #Encontra pontos que circundam regi es conexas
55         contours, hierarchy = cv2.findContours(gray, cv2.RETR_LIST, cv2.
56         CHAIN_APPROX_SIMPLE)
57
58         if contours:
59             #Retornando a rea do primeiro grupo de pixels brancos
60             maxArea = cv2.contourArea(contours[0])
61             contourMaxAreaId = 0
62             i = 0
63
64             #Para cada grupo de pixels branco
65             for cnt in contours:
66                 #Procura o grupo com a maior rea
67                 if maxArea < cv2.contourArea(cnt):
68                     maxArea = cv2.contourArea(cnt)
69                     contourMaxAreaId = i
70                 i += 1
71
72             #Define qual a rea de cada cor
73             match cntFiltro:
74                 case 0 :
75                     areaVerde = maxArea
76                 case 1:
77                     areaAzul = maxArea
78                 case 2:
79                     areaVermelha = maxArea
80                 case 3:
81                     areaRosa = maxArea

```



```

80         cntFiltro += 1
81
82         #Acha o contorno com maior área
83         cntMaxArea = contours[contourMaxAreaId]
84
85         #Retorna um retângulo que envolve o contorno em questão
86         xRect, yRect, wRect, hRect = cv2.boundingRect(cntMaxArea)
87
88         #Desenha retângulo envolvendo a área
89         cv2.rectangle(frame, (xRect, yRect), (xRect + wRect, yRect +
90             hRect), (0, 0, 255), 2)
91
92         #Calcula as coordenadas dos centros dos retângulos
93         coordXcentros[aux] = int(xRect+wRect/2)
94         coordYcentros[aux] = int(yRect+hRect/2)
95
96         aux += 1
97
98         #Calcula a distância para a pinça
99         pinca = math.hypot(coordXcentros[1]-coordXcentros[0], coordYcentros
100             [1]-coordYcentros[0])
101
102         #Calcula a distância para frente
103         frente = math.hypot(coordXcentros[2]-coordXcentros[3], coordYcentros
104             [2]-coordYcentros[3])
105
106         #Calcula a distância para a parte de cima
107         cima = coordYcentros[3]
108
109         #Calcula a distância para o lado esquerdo
110         esquerda = coordXcentros[3]
111
112         #Salva os dados das áreas rastreadas
113         with open('./dados.csv', 'a', newline='') as arquivo:
114             writer = csv.writer(arquivo)
115             writer.writerow([areaVerde, areaAzul, areaVermelha, areaRosa])
116
117         return frame, pinca, frente, cima, esquerda
118
119 #===== Movimentos do Servo =====
120 #Define o delimitador
121 delim = bytes(';','utf-8')
122 #Realiza a comunicação Serial com o Arduino
123 arduino = serial.Serial(port = 'COM6', baudrate = 115200, timeout=1)

```

```

124 #Recebe o n emro do motor e valor entre 0 e 180 para movimentar o Servo
125 def write_dat(motor, valor):
126     match motor:
127         case 1:
128             arduino.write(bytes('1' + str(int(valor)), "utf-8"))
129             arduino.write(delim)
130         case 2:
131             arduino.write(bytes('2' + str(int(valor)), "utf-8"))
132             arduino.write(delim)
133         case 3:
134             arduino.write(bytes('3' + str(int(valor)), "utf-8"))
135             arduino.write(delim)
136         case 4:
137             arduino.write(bytes('4' + str(int(valor)), "utf-8"))
138             arduino.write(delim)
139     return 1
140
141
142 #===== Main =====
143 while True:
144     #Tempo para inicial do processamento
145     tempoProcessamento = time.perf_counter()
146
147     #Captura imagem de c mera
148     success, frame = cap.read()
149
150     #Inverte imagem verticalmente
151     frame = cv2.flip(frame, 1)
152
153     #Realiza tratamento da imagem
154     frame, pinca, frente, cima, esquerda = computeTracking(frame)
155
156     #Mostra a imagem tratada
157     cv2.imshow("webcam", frame)
158
159     #Interpola as dist ncias dos pontos de interesse para os movimentos
160     valorPinca = int(np.interp(pinca, [30,155], [40,0]))
161     valorFrente = int(np.interp(frente, [50,160], [50,0]))
162     valorCima = int(np.interp(cima, [200,450], [170,110]))
163     valorEsquerda = int(np.interp(esquerda, [100,550], [180,0]))
164
165     #Envia o valor de movimento para o Servo
166     write_dat(1, valorPinca)
167     write_dat(2, valorFrente)
168     write_dat(3, valorCima)
169     write_dat(4, valorEsquerda)
170

```

```
171     #Limpa a comunica o com o Arduino
172     arduino.flush()
173
174     #Calcula o tempo de processamento
175     processamento = time.perf_counter() - tempoProcessamento
176     with open('./tempos.csv', 'a', newline='') as arquivo:
177         writer = csv.writer(arquivo)
178         writer.writerow([processamento])
179
180     #Para parar a captura, pressionar q
181     if cv2.waitKey(1) & 0xFF == ord('q'):
182         break
183
184 #Fecha a captura e telas abertas
185 arduino.close()
186 cap.release()
187 cv2.destroyAllWindows()
```

ANEXO A - USO DE TECNOLOGIAS DIGITAIS

Tabela 2 - Uso de tecnologias digitais

Percentual de respostas por intensidade tecnológica (%)

Foco	Tecnologia	Intensidade tecnológica			
		alta	média-alta	média-baixa	baixa
Desenvolvimento	Sistemas integrados de engenharia para desenvolvimento e manufatura de produtos	29	44	25	22
	Prototipagem rápida, impressão 3D e similares	27	20	11	11
	Design assistido por inteligência artificial	2	5	2	3
	Simulações/ análise de modelos virtuais para projeto e comissionamento (Elementos Finitos, Fluidodinâmica Computacional etc.)	15	19	9	5
Processo	Automação digital sem sensores, uso de Controlador Lógico Programável (CLP) sem sensores	27	38	31	28
	Automação digital com sensores para controle de processo	37	39	35	36
	Automação digital com sensores com identificação de produtos e condições operacionais, linhas flexíveis	27	20	17	22
	Monitoramento e controle remoto da produção com sistemas do tipo MES e SCADA*	17	15	11	10
	Sistemas integrados de manufatura, como comunicação M2M (máquina-máquina)	15	14	9	14
	Aplicações de Inteligência Artificial para soluções na fábrica	8	9	5	6
	Simulação de processos e gêmeos digitais (<i>Digital Twins</i>)	0	5	1	1
	Manufatura aditiva, robôs colaborativos (cobots)	12	14	8	6
	Coleta, processamento e análise de grandes quantidades de dados (<i>big data</i>) do processo produtivo	21	18	16	14
	Inspeção da qualidade automatizada ou avançada	25	17	11	12
Produto	Ferramentas digitais que aumentam as capacidades dos trabalhadores (<i>smart glasses, smart watches, etc.</i>)	6	12	9	8
	Incorporação de serviços digitais nos produtos (<i>Internet das Coisas</i> ou <i>Product Service Systems</i>)	15	15	10	10
	Coleta, processamento e análise de grandes quantidades de dados (<i>big data</i>) sobre o mercado; monitoramento do uso dos produtos pelos consumidores	21	13	8	9
	Ferramentas digitais de relacionamento com o cliente (<i>chatbots, atendimento ao cliente interativo, etc.</i>)	31	28	18	22

Nota: A soma dos percentuais supera 100% devido a possibilidade de múltiplas escolhas.

*MES - *Manufacturing Execution Systems*; SCADA - *Supervisory Control and Data Acquisition*.

Fonte: CNI (2022, p. 13).