

Bárbara Nicolý Menezes de Oliveira

**Alinhamento automático no empacotamento
de chips fotônicos baseado em visão
computacional**

Campina Grande, Brasil

13 de dezembro de 2019

Bárbara Nicolý Menezes de Oliveira

Alinhamento automático no empacotamento de chips fotônicos baseado em visão computacional

Trabalho de Conclusão de Curso submetido à Coordenadoria do Curso de Graduação em Engenharia Elétrica da Universidade Federal de Campina Grande, Campus Campina Grande, como parte dos requisitos necessários para obtenção do título de Engenheiro Eletricista.

Universidade Federal de Campina Grande - UFCG
Centro de Engenharia Elétrica e Informática - CEEI
Departamento de Engenharia Elétrica - DEE

Orientador: Antonio Marcus Nogueira Lima, Dr.

Campina Grande, Brasil

13 de dezembro de 2019



Bárbara Nicolý Menezes de Oliveira

Alinhamento automático no empacotamento de chips fotônicos baseado em visão computacional

Trabalho de Conclusão de Curso submetido à Coordenadoria do Curso de Graduação em Engenharia Elétrica da Universidade Federal de Campina Grande, Campus Campina Grande, como parte dos requisitos necessários para obtenção do título de Engenheiro Eletricista.

Trabalho aprovado em: 06/12/2019

Antonio Marcus Nogueira Lima, Dr.
Orientador

Adolfo Fernandes Herbster, D.Sc.
Convidado

Campina Grande, Brasil
13 de dezembro de 2019

Dedico este trabalho aos meus amados pais, Michelline de Menezes Oliveira e José Erivan Gomes de Oliveira, à minha família, em especial, ao meu querido e falecido avô Antônio Ribeiro de Oliveira; quem me inspirou na escolha pela carreira em engenharia.

Agradecimentos

Agradeço primeiramente à Deus, que me permite o dom da vida diariamente e me capacita a seguir meus prósitos com fé, por meio de Seu grandioso amor.

Agradeço aos meus pais, Sra. Michelline e Sr. Erivan, que de maneira insubstituível são detentores de minha grande admiração. Sou grata por atuarem diariamente em minha formação como humana, através do exemplo de amor, gentileza, sabedoria e fé; que me inspiram a viver a importância do olhar fraterno e justo, sobre todas as circunstâncias em que a vida se abre nos meus caminhos. As conquistas são sempre prazerosas por tê-los comigo à todo momento.

Agradeço aos meus familiares, pelo apoio amigo durante toda essa jornada, pelas orações de incentivo e palavras de cuidado.

Aos amigos que dividiram grandes momentos de aprendizado e foram suporte em momentos de dúvidas e aflições. Em especial ao Dimitre e ao Idris, amigos que me trouxeram calma e paz, dentro de suas diferentes personalidades, mas em equivalente oferta de ótima companhia.

Aos meus mestres, professores que fizeram parte de minha jornada acadêmica na busca pela formação em engenharia, pela confiança em meu trabalho e pela honra do aprendizado, tanto profissional quanto humano, por meio de seus exemplos de excelência e admirável personalidade.

Ao meu cachorro, Aquiles, companheiro fiel de bons momentos. Que mesmo não sendo humano, me ensina diariamente a nobreza em doar-se ao outrem, sem métricas ou julgamentos.

"It's fine to celebrate success but it is more important to heed the lessons of failure. "
Bill Gates

Resumo

Este trabalho trata de um processo de automação no posicionamento de componentes fotônicos para a montagem de circuitos integrados, a partir do reconhecimento por imagem, em uma máquina posicionadora industrial. A estação *Pick&Place* foi o sistema alvo da aplicação, sendo este concedido pela empresa *Idea! Electronic Systems*, como parte do processo de capacitação no projeto de Fotônica em Silício.

Na estação *Pick and Place* há um controlador lógico programável (CLP) atuando em um sistema mecânico de deslocamento por motores-de-passo, limitados por sensores de movimento do tipo indutivos. O sistema é integrado com uma plataforma digital por meio de um canal de comunicação serial vinculado à uma arquitetura cliente-servidor via protocolo OPC. Uma câmera CMOS, monocromática de 10,55 MP é utilizada para aquisição de imagens da base de operações. A etapa de calibração da câmera do sistema de aquisição de imagens foi realizada com o emprego do método *Zhang*, e o reconhecimento automático dos objetos na base de trabalho foi feito pelo processo de segmentação de imagem, com o uso de técnicas de processamento digital de imagens como o operador Sobel, coeficiente de correlação normalizado, filtros lineares de suavização, dentre outras.

Os resultados obtidos indicaram uma correta correlação do mundo adimensional dos pixels com o do eixo mecânico dos motores-de-passo do equipamento, mapeado em escala milimétrica. Um erro de 0.8mm foi computado no primeiro deslocamento de cada sequência. O tempo de uma operação padrão completa de "Pick" e "Place" para a solução guiada por imagem, é de 01min46s.

Palavras-chaves: Visão de Máquina, Visão omputacional, Indústria 4.0, empacotamento fotônico, automação, fotônica de silício

Abstract

This work deals with an automation process in the positioning of photonic components, based on image recognition, in an industrial positioning machine. The Pick & Place station was the target system of the application, which was granted by the company Idea! Electronic Systems as part of the training process in the Silicon Photonics project.

In the Pick and Place station there is a programmable logic controller (PLC) operating in a mechanical stepper displacement system, limited by inductive motion sensors. The system is integrated with a digital platform through a serial communication channel linked to a client-server architecture via OPC protocol. A 10.55 MP monochrome CMOS camera is used for base-image acquisition. The camera calibration step of the image acquisition system was performed using the Zhang method, and the automatic recognition of the objects in the workbench was done by the image segmentation process, using the techniques of digital image processing such as Sobel operator, normalized correlation coefficient, smoothing linear filters, among others.

The final results indicated a correct correlation of the dimensionless world of the pixels with the mechanical axis of the equipment stepper motors, mapped in millimeter scale. A 0.8mm error was computed at the first offset of each sequence. The time for a full standard "Pick" and "Place" operation for the image guided solution is 1min: 46s.

Key-words: Machine Vision, Computer Vision, Industry 4.0, Photonic Packaging, Automation, Silicon Photonics.

Lista de ilustrações

Figura 1 – Diagrama de blocos do modelo funcional da máquina <i>Pick and Place</i>	2
Figura 2 – Representações tridimensional e bidimensional da estrutura mecânica da máquina posicionadora.	3
Figura 3 – Imagens reais da estrutura mecânica do equipamento posicionador.	4
Figura 4 – Esquema de mapeamento de I/Os do controlador lógico programável P7C302.	5
Figura 5 – Sensores de fim de curso do tipo indutivo para limite de deslocamento ao longo dos eixos mecânicos. <i>Fonte: Banco de imagens da Idea! Electronic Systems</i>	6
Figura 6 – Diagrama de blocos da comunicação OPC e APIs necessárias para sua correta implementação com os produtos Hi Tecnologia.	7
Figura 7 – Campo de visão do ambiente de trabalho capturado pela câmera.	10
Figura 8 – Referência de catálogo da câmera CMOS UI-1490LE-M-GL utilizada no sistema de visão computacional da máquina <i>Pick and Place</i>	11
Figura 9 – Exemplo gráfico da representação digital de uma função imagem $f(x, y)$ de tamanho $M \times N$, em uma estrutura matricial.	12
Figura 10 – Ilustração do processo de aquisição de uma imagem digital (a) e posterior geração de uma imagem digital (b) através de amostragem e quantização, a partir de uma imagem contínua (elemento em escala de cinza).	13
Figura 11 – Exemplo ilustrativo de escala de níveis de cinza para $K=8$	13
Figura 12 – Ilustração das definições de distâncias: (a) vizinhança D_4 e (b) vizinhança D_8	14
Figura 13 – Representação gráfica da implementação matemática da função gradiente	15
Figura 14 – Representação gráfica da binarização de uma imagem digital pelo método de segmentação por <i>thresholding</i>	16
Figura 15 – Ilustração do modelo <i>Pin Hole</i> (b) para formação de imagens em câmeras baseado na formação de imagens do olho humano (a). <i>Fonte: Elaborada pela própria autora</i>	17
Figura 16 – O modelo da câmera pinhole. O ponto da imagem \mathbf{u} está na interseção do plano da imagem e a linha que une o ponto 3D, \mathbf{X} e o centro da câmera com distância focal f . A linha contínua é o eixo óptico da câmera. <i>Fonte:(SONKA; HLAVAC; BOYLE, 2015), pg.12</i>	18
Figura 17 – Exemplo ilustrativo da transformação de coordenadas de mundo $3D \Rightarrow 2D$. <i>Fonte: Elaborada pela própria autora</i>	18

Figura 18 – Ilustração com medidas reais do sistema de condicionamento de imagem da estação <i>Pick and Place</i>	19
Figura 19 – Conjunto de 10 fotos do padrão de um tabuleiro de xadrez com diferentes perspectivas para a calibração da câmera. <i>Fonte: Elaborada pela própria autora</i>	22
Figura 20 – Rotina inicial de calibração da câmera para aquisição da matriz de parâmetros intrínsecos.	23
Figura 21 – Padrão "dots" para reforço de precisão e confiabilidade do procedimento de calibração.	24
Figura 22 – Dispositivos fotônicos com diferentes padrões de litografia, bem como materiais de fabricação.	25
Figura 23 – Conjunto de elementos utilizados e base de referência para realização das operações.	26
Figura 24 – Ciclo de diretrizes para correta execução da função de reconhecimento automático de dispositivos fotônicos. <i>Fonte: Autoria Própria</i>	26
Figura 25 – Ilustração do fluxo de operações empregadas no reconhecimento do objeto a ser capturado pela pinça no procedimento de Pick.	28
Figura 26 – Capturas da câmera: (a) <i>Background</i> e (b) Imagem de saída após rotina de detecção com emprego da técnica <i>Edge Detection</i>	29
Figura 27 – Ilustração do método <i>template matching</i> para localização da característica desejada nos dispositivos fotônicos.	30
Figura 28 – Exemplo de imagem de saída após realização completa do algoritmo de reconhecimento de dispositivos na área de trabalho.	32
Figura 29 – Diagrama ilustrativo do fluxo de execução do processo de calibração da câmera e mapeamento dos dispositivos fotônicos em uma máquina posicionadora do tipo <i>Pick and Place</i>	32
Figura 30 – Exemplo de um caso de exceção " Limitação de deslocamento dentro da área de trabalho ".	33
Figura 31 – Exemplo de um caso de exceção " Mismatch entre o número de componentes para o "Pick" e o número de bases para o "Place" ".	34
Figura 32 – Exemplo de erro para a falta de objetos no espaço de trabalho ou falha no reconhecimento.	35
Figura 33 – Linguagem de programação em diagrama de bloco para desenvolvimento no software Labview.	36
Figura 34 – DataSocket para acesso, por parte do Labview, aos itens(I/Os) publicados na camada servidora OPC(SOS), a partir de um arquivo do tipo CSV.	36
Figura 35 – Interface de interação com o usuário para operações manual e automática da estação <i>Pick and Place</i>	37

Lista de abreviaturas e siglas

PIC	Photonic Integrated Circuit
CMOS	Complementary Metal Oxide Semiconductor
IHM	Interface Homem-máquina
CLP	Controlador Lógico Programável
OPC	Object Linking and Embedding for Process Control
CPU	Unidade Central de Processamento
PA	Process Automation
FOV	Field of View
ROI	Region of Interest
MTF	Modulation Transfer Function

Lista de símbolos

θ	Teta
μ	Mi
Δ	Delta
*	Operador de convolução
X,Y,Z	Coordenadas de mundo em milímetros
x,y,z	Coordenadas em pixels

Sumário

1	INTRODUÇÃO	1
2	SISTEMA DE POSICIONAMENTO	2
2.1	Estrutura mecânica	2
2.2	Sensores e unidade de controle	3
2.2.1	CLP - P7C302	5
2.2.2	Sensores indutivos	6
2.3	Rede de comunicação	6
2.3.1	Comunicação OPC	6
2.4	Considerações Finais	8
3	SISTEMA DE VISÃO COMPUTACIONAL	9
3.1	Visão geral e aspectos ópticos	9
3.1.1	Digitalização e Segmentação de imagens	11
3.1.1.1	Correlação espacial e vizinhança	12
3.1.1.2	Gradiente	14
3.1.1.3	Magnitude e <i>Thresholding</i>	16
3.1.2	Calibração da Câmera	17
3.1.2.1	Matriz de Transformação e mapeamento 2D \Leftrightarrow 3D	19
3.1.2.2	Procedimento	21
3.2	Considerações Finais	24
4	PROCEDIMENTO EXPERIMENTAL	25
4.1	Reconhecimento e mapeamento dos dispositivos	25
4.2	Tratamento de exceções e erros	33
4.2.1	Interface Homem-Máquina	35
4.2.1.1	LabVIEW	35
4.2.1.2	Funcionalidades	35
4.3	Considerações Finais	37
5	CONCLUSÕES	38
5.1	Perspectivas de Trabalhos Futuros	39
	REFERÊNCIAS	40

1 Introdução

Os volumes de produção têm um grande impacto no desenvolvimento e na evolução de uma máquina. Na indústria de circuitos fotônicos integrados, as máquinas atuais devem priorizar a flexibilidade, permitindo que os usuários finais adaptem rapidamente seus equipamentos às mudanças de processo.

Neste trabalho, uma máquina industrial do tipo posicionadora, comumente referida como estação *Pick and Place*, foi o alvo de um processo de automação através do desenvolvimento de um sistema de visão computacional, objetivando o alinhamento automático de dispositivos fotônicos durante uma das etapas do processo de empacotamento de *chips* híbridos.

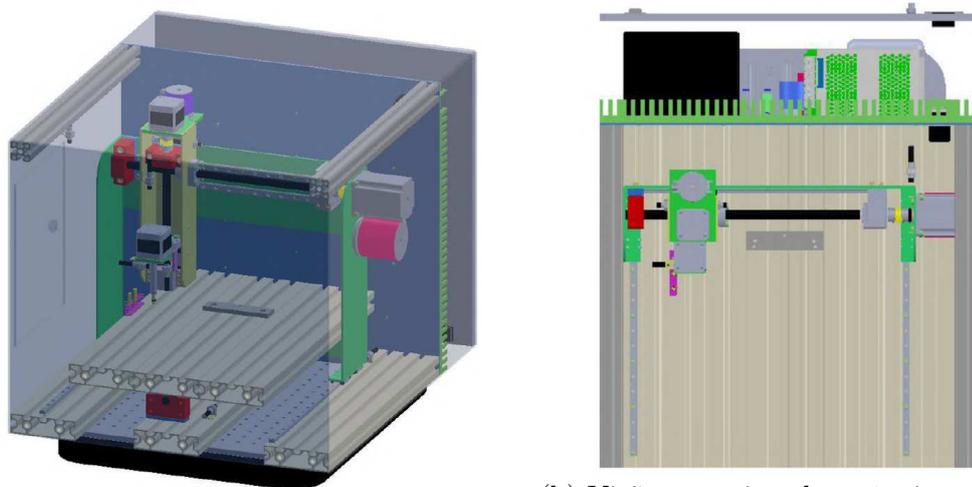
A máquina, concedida pela empresa *Idea! Electronic Systems*, realiza o deslocamento de uma pinça à vácuo, condicionado por um conjunto de três motores-de-passo, dispostos em orientação tridimensional e interconectados com parafusos castanha de 5mm de diâmetro e cerca de 30cm de comprimento, que deslizam nas direções dos eixos X, Y e Z, com um grau de liberdade adicional para a rotação do eixo Z, referido como c. Até a data de recebimento, o equipamento não dispunha de um sistema automático de identificação e localização dos dispositivos fotônicos dentro do perímetro de trabalho, apesar de possuir um sistema óptico com uma câmera CMOS monocromática de 10,55 Mp.

Contudo, este equipamento posicionador requer identificação e localização de dispositivos fotônicos passivos e seus respectivos pontos alvos, pré-determinados, afim de habilitar a coleta e o subsequente transporte destes dentro do perímetro de uma base quadrada de 40,6cm de lado, em rotinas cíclicas de deslocamento.

O sistema de visão deve identificar elementos com 1mm^2 de área e, no máximo 22mm de comprimento. O sistema também deve ser capaz de diferenciar estes elementos - concebidos em materiais com diferentes índices de refletividade, transparência e níveis de contraste - que são posicionados em um mesmo campo de visão.

Com este trabalho implementou-se um sistema de visão computacional com o uso da ferramenta de desenvolvimento *Visual Studio Code*, para o reconhecimento e mapeamento automáticos por imagem, dos elementos fotônicos, afim de habilitar a operação de alinhamento automático. No experimento elaborado, a estudante entra em contato com diversos conceitos abordados nas áreas de processamento digital de imagens, óptica e instrumentação eletrônica, como: a calibração de uma câmera CMOS, o processo de formação, aquisição e segmentação de imagens, e a utilização de redes industriais de sensores para realizar a comunicação entre os componentes do sistema.

Figura 2 – Representações tridimensional e bidimensional da estrutura mecânica da máquina posicionadora.



(a) Visão em perspectiva da máquina *Pick and Place* com estrutura de proteção. (b) Visão superior da máquina *Pick and Place*.

Fonte: Banco de imagens da Idea! Electronic Systems

A interpolação dos múltiplos eixos e o controle de trajetória são baseados na computação em tempo real executada por um computador incorporado, a interface do usuário e a programação no nível do aplicativo de *software*. Esta última é facilitada pela presença do firmware que interpreta e executa os comandos de alto nível. Entradas analógicas diretas e conversores analógico-digitais facilitam o alinhamento ativo e recebem sinais de tensão direta de drivers dedicados aos motores-de-passo. O esquema mecânico físico, descrito, pode ser visto na figura 3a. O sistema completo, que integra a eletrônica do controle de posicionamento com a eletrônica de potência que aciona os motores pode ser visto na figura 3b.

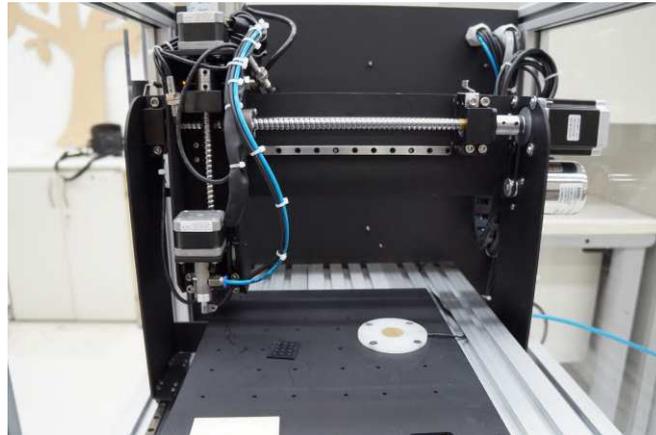
Para prover as funcionalidades requeridas por operações convencionais do alinhamento de PICs, a resolução nanométrica dos sensores de indutivos, o poder de processamento e a robustez do controlador lógico programável junto à ferramenta de *software Labview* constituem os principais módulos para a automação das sequências de operações de posicionamento.

2.2 Sensores e unidade de controle

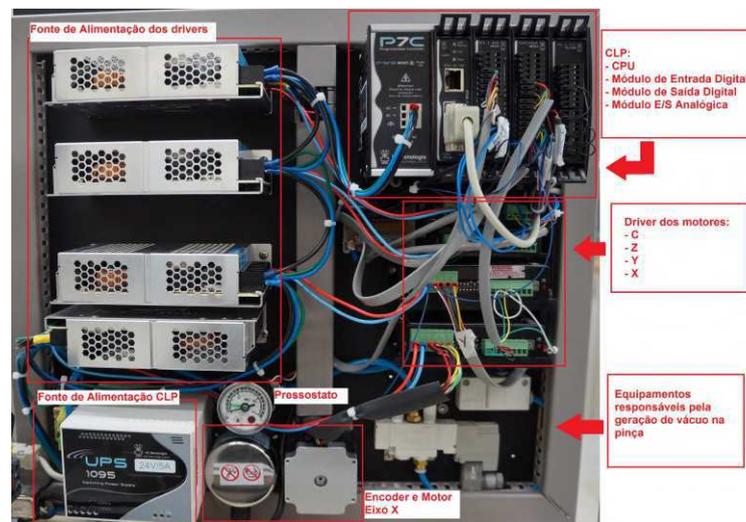
Referir-se a sistemas automáticos ou com determinando grau de automação, relaciona diretamente a utilização de transdutores; usualmente definidos como um circuito capaz de transformar um sinal de entrada, de natureza mecânica ou eletromagnética, em um sinal de saída elétrico ou óptico.

Sensores compõem uma extensa categoria de transdutores, ao passo que, por meio

Figura 3 – Imagens reais da estrutura mecânica do equipamento posicionador.



(a) Estrutura mecânica da máquina Pick and Place, desenvolvida nas dependências da Idea! Electronic Systems para o processo de empacotamento automático de chips fotônicos.



(b) Sistema de controle do equipamento: eletrônica de controle e eletrônica de potência.

Fonte: Banco de imagens da Idea! Electronic Systems

de princípios físicos, mecânicos ou químicos podem atuar na interface entre as grandezas de mundo e o condicionamento de sinais elétricos. Neste trabalho consta a utilização de sensores indutivos, empregados com objetivo de reagir à proximidade de objetos metálicos, por meio de variações do fluxo eletromagnético em seu campo eletromagnético.

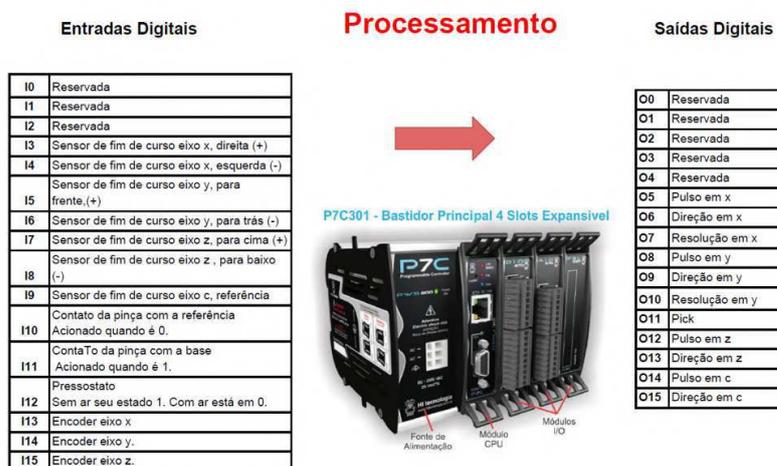
Mas, condicionar grandezas físicas em elétricas não completa o processo de automação de um sistema sem que haja um módulo ativo que manipule, altere e distribua essas grandezas de forma adequada entre os subsistemas do processo. A unidade do Programador Lógico Programável também será descrita de maneira breve.

2.2.1 CLP - P7C302

O CLP é um dispositivo eletrônico que possui uma memória interna programável capaz de armazenar sequências de instruções lógicas, que controlam a entrada e a saída dados para instrumentos conectados a ele, sendo os sensores e atuadores, respectivamente. Sendo assim, em sua unidade central de processamento (CPU), há a execução contínua de um firmware, software desenvolvido pelo seu desenvolvedor que monitora constantemente as entradas de dados e com o auxílio de um programa implementado por um usuário, em software compatível com o do fabricante do CLP, verifica-se a lógica desenvolvida para realizar o controle das saídas, podendo contar com implementações de portas lógicas simples, como também de estruturas temporizadoras.

A máquina utilizada neste trabalho opera com o CLP P7C302 da **Hi Tecnologia**, que pode ser configurado para acesso a até 368 pontos de I/O. Sua arquitetura é baseada em bastidores expansíveis com capacidade de 4 módulos de hardware por bastidor. Ele foi instalado em uma estrutura de suporte, o *rack*, que oferece interligamento eletromecânico entre os seus demais componentes, que são os módulos. O primeiro a ser conectado no rack foi uma fonte de alimentação, UPS 1095 24V/5A, que fornece tensão para o bastidor principal (a CPU) e para os módulos de entrada e saída. Em seguida, a CPU302 - Processador G3S, que realiza o processamento e o armazenamento de todas as informações no controle de um processo, devido à presença de um processador e de memórias flash e RAM na sua estrutura interna. Também são utilizados, um módulo de comunicação WCM610 e três módulos de I/O: DIM 401, DOM 451 e AIO 574, compondo a interface de conexão de entradas e saídas analógicas. O CLP pode ser visto já integrado ao sistema de controle na figura 3a, e o esquema de reservas dos módulos de I/O é apresentado na figura 4.

Figura 4 – Esquema de mapeamento de I/Os do controlador lógico programável P7C302.



Fonte: Banco de imagens da

Idea! Electronic Systems

2.2.2 Sensores indutivos

Seis unidades do sensor digital de deslocamento indutivo - Série EX-305V, da empresa *Keyence* são utilizadas com a finalidade de delimitar o movimento mecânico da unidade que acopla a pinça de coleta com a câmera do sistema de visão computacional, dentro de limites de segurança para o correto funcionamento do equipamento, bem como a prevenção de danos aos módulos integrados no sistema. A cabeça sensora cilíndrica de diâmetro 5,4mm, possui resolução de $0,4\mu\text{m}$, faixa de 0 a 1mm e uma precisão de $\pm 0,3\%$.

Na figura 5 pode ser visto o sensor indutivo descrito nesta seção.

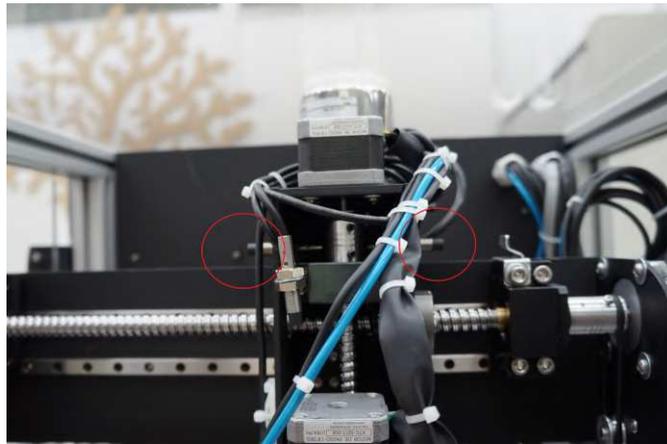


Figura 5 – Sensores de fim de curso do tipo indutivo para limite de deslocamento ao longo dos eixos mecânicos. *Fonte: Banco de imagens da Idea! Electronic Systems*

2.3 Rede de comunicação

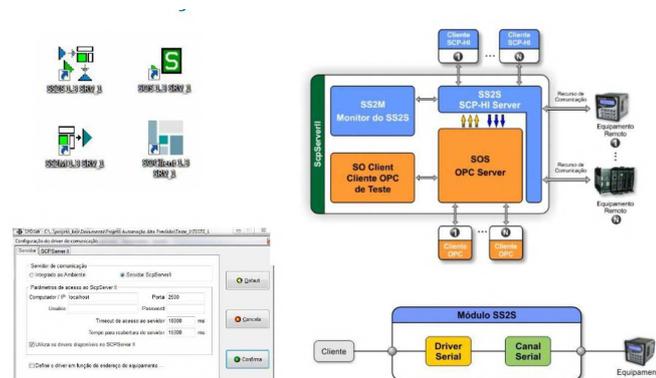
Esta seção é destinada a exposição técnica da rede de comunicação e os protocolos para sincronização dos dados que operam na estação *Pick and Place*, conforme funcionamento definido pelos desenvolvedores do equipamento.

2.3.1 Comunicação OPC

A tecnologia OPC (*Object Linking and Embedding for Process Control*) é um padrão de aquisição e transmissão de dados em tempo real entre *softwares* e *hardwares*, gerando um servidor com as informações adquiridas, um programa, e clientes – outros *softwares* que as utilizam, como as Interfaces Homem Máquina (IHM). Uma grande vantagem do uso da tecnologia OPC é a fácil integração entre dispositivos de diferentes fabricantes (SCHLEIPEN et al., 2016), como é o caso do sistema de posicionamento utilizado neste trabalho, no qual os dados recebidos do CLP P7C302 são convertidos para o protocolo OPC, por meio do padrão COM (*Component Object Model*) que possibilita a criação de objetos no servidor que armazenam os dados recebidos e os que serão enviados para o cliente.

Dessa forma, os computadores que possuem os drivers dos equipamentos de campo, que constituirão os “servidores OPC”, reconhecem os dados provenientes da rede de comunicação dos equipamentos da planta industrial e os traduzem para o padrão OPC. As aplicações que recebem esses dados são os “clientes OPC” e podem estar em quaisquer computadores conectados à rede do servidor OPC, fechando o ciclo do fluxo de dados provenientes da planta (motores de passo e sensores de fim de curso, neste caso). A figura 6 ilustra o diagrama de blocos da arquitetura cliente-servidor necessária ao correto endereçamento de dados entre o hardware de controle e a API de software.

Figura 6 – Diagrama de blocos da comunicação OPC e APIs necessárias para sua correta implementação com os produtos Hi Tecnologia.



Fonte:

<https://www.hitecnologia.com.br/produtos/software/conectividade/servidor-opc-hi-tecnologia/>

2.4 Considerações Finais

O Capítulo 2 destacou os principais módulos operacionais do sistema em discussão, pontuando informações úteis para o levantamento de requisitos operacionais da seção seguinte, que abordará o sistema de visão computacional empregado como ferramenta de automação no processo gerido pela máquina posicionadora. Portanto, ter em mente as limitações mecânicas, o tipo de grandezas elétricas em manipulação, bem como a maneira e grau de acesso as mesmas, constitui uma abordagem inicial condicionadora das implementações subsequentes.

3 Sistema de Visão Computacional

Este capítulo aborda conceitos da fundamentação teórica do processo de formação de imagem por um modelo matemático bem difundido no campo da visão computacional, além de revisar definições relacionadas à manipulação de imagens digitais. Na sequência, na seção 3.2 será exposta a metodologia empregada na estratégia de reconhecimento de imagem conforme uma seleção de algoritmos adequados à natureza do problema.

3.1 Visão geral e aspectos ópticos

O nível de complexidade inerente ao processo de *packaging* de circuitos fotônicos integrados endereça desafios que vão além das requisições de mercado por escalabilidade e custo benefício. Na verdade, atingir essa necessidade e atender as inúmeras áreas que estão apostando na fotônica de silício para as tecnologias do futuro, são demandas que estão condicionadas ao nível de automação que a indústria é capaz de ofertar ao ecossistema de fabricação dos dispositivos com suas inúmeras variáveis de caracterização e concepção.

A variedade de soluções de *packaging*, combinada com a falta de padrões, exige que as máquinas atuais priorizem a flexibilidade em detrimento de velocidade, permitindo que os usuários finais adaptem rapidamente seus equipamentos às mudanças de processo, passando de protótipos iniciais/séries limitadas, para produção otimizada de dispositivos.

Há uma diferença substancial entre precisão, repetibilidade e resolução da mecânica multi-eixo usada para posicionar e alinhar os componentes dos dispositivos fotônicos. Embora a combinação de mecânicas convencionais acionadas por motor e elementos acionados por sensores possa fornecer resoluções de algumas dezenas de nanômetros, alcançar a repetibilidade bidirecional ao nível submícron em equipamentos industriais, robustos e com preços razoáveis ainda é um desafio.

É nesse contexto que o emprego da visão computacional, como ferramenta de precisão, ganha espaço relevante no processo de *packaging* fotônico.

A visão de máquina representa uma parte essencial de uma máquina de montagem. Está presente nas operações de *pick and place*, posicionamento e alinhamento de peças, inspeção óptica automatizada, identificação de peças via OCR (reconhecimento óptico de caracteres) e leitura de código, bem como na caracterização de feixes, tanto no espectro visível quanto na faixa infravermelha.

A visão de máquina refere-se ao uso de algoritmos de processamento de imagem para extrair automaticamente recursos e informações de uma imagem adquirida. Na maioria dos casos, essas informações são correlacionadas aos dados posicionais do equipamento

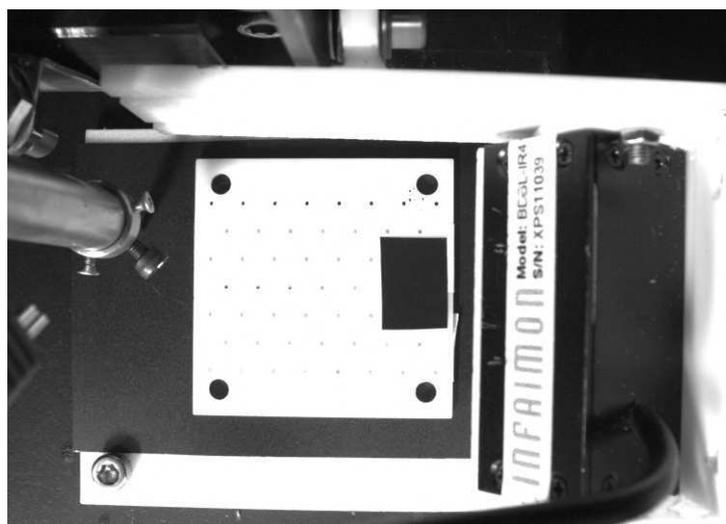
de movimento e exploradas para executar operações automáticas, o que configura a principal motivação para a realização deste trabalho.

Os materiais utilizados na fotônica nem sempre são fáceis de visualizar, exibindo frequentemente um baixo contraste e superfícies altamente refletivas; O projetista do sistema de visão computacional geralmente enfrenta requisitos contraditórios, como ter um FOV (campo de visão) grande, para gerar uma imagem de todos os componentes, além de ter uma resolução muito alta para obter precisão geométrica suficiente. Quando a resolução é muito menor que $1\mu\text{m}/\text{pixel}$ (em outras palavras: um único pixel da imagem é dimensionalmente menor que o comprimento de onda da luz), o efeito do limite de difração se torna perceptível. A imagem exibirá bordas 'embaçadas', bem como artefatos devido à difração, convergindo em perda de nitidez.

Combinar imagem e movimento também requer a correlação do mundo adimensional dos pixels com o do eixo mecânico e seus codificadores. Isso pode incluir também a calibração relativa de uma câmera, usada no tipo flip-chip de posicionamento de componentes e outras rotinas de alinhamento.

Um outro fator crítico que merece atenção na escolha do sistema óptico apropriado ao estágio de *packaging* que se deseja realizar é a iluminação do ambiente de trabalho. Os melhores resultados foram obtidos combinando a iluminação através do ajuste da abertura do obturador com luz monocromática difusa baseada em LED. Na figura 7 é possível observar o campo de visão do ambiente de trabalho, capturado pela câmera operando com parâmetros ópticos pré-definidos (*default*) e sob condições de luz semi-controladas.

Figura 7 – Campo de visão do ambiente de trabalho capturado pela câmera.



Fonte: Elaborada pela própria autora

Na prática, combinar a qualidade geral das lentes, a iluminação que pode ser implementada e a morfologia real dos marcadores e características que podemos encontrar em um chip fotônico com a necessidade de ter FOVs (*Field of View*) razoavelmente grandes,

dificultam ter uma resolução abaixo de $1\mu m$ ou $0,5\mu m/pixel$.

O sistema ótico empregado neste trabalho atende as especificações básicas consideradas anteriormente e é composto pelo seguinte conjunto de elementos:

- 01 Câmera USB UI-1490LE-M-GL CMOS, 3,2fps, 3840x2748, 10,5MPx e 1/2";
- 01 lente standard alta resolução 8mm;
- 01 lente standard 25mm;
- 01 ponto de iluminação LED monocromático branco com difusor;
- Resolução de $2,5\mu/pixel$.

Figura 8 – Referência de catálogo da câmera CMOS UI-1490LE-M-GL utilizada no sistema de visão computacional da máquina *Pick and Place*.

UI-1490LE-M-GL																																			
	<table border="1"> <thead> <tr> <th colspan="2">Sensor</th> </tr> </thead> <tbody> <tr> <td>Sensor Technology</td> <td>CMOS Mono</td> </tr> <tr> <td>Manufacturer</td> <td>Aptina</td> </tr> <tr> <td>Resolution (MPix)</td> <td>10.55</td> </tr> <tr> <td>Resolution (h x v)</td> <td>3840 x 2748</td> </tr> <tr> <td>ADC</td> <td>12 bit</td> </tr> <tr> <td>Color depth (camera)</td> <td>8 bit</td> </tr> <tr> <td>Pixel Class</td> <td>UHD+</td> </tr> <tr> <td>Sensor Size</td> <td>1/2</td> </tr> <tr> <td>Shutter</td> <td>Rolling shutter</td> </tr> <tr> <td>max. fps in Freerun Mode</td> <td>3.2</td> </tr> <tr> <td>Binning Modes</td> <td>Color</td> </tr> <tr> <td>Subsampling Modes</td> <td>Color</td> </tr> <tr> <td>Sensor Model</td> <td>MT9J003STM</td> </tr> <tr> <td>Pixel size</td> <td>1.67 μm</td> </tr> <tr> <td>Optical Size</td> <td>6.413 mm x 4.589 mm</td> </tr> <tr> <td>Optical sensor diagonal</td> <td>7.89 mm (1/2.03")</td> </tr> </tbody> </table>	Sensor		Sensor Technology	CMOS Mono	Manufacturer	Aptina	Resolution (MPix)	10.55	Resolution (h x v)	3840 x 2748	ADC	12 bit	Color depth (camera)	8 bit	Pixel Class	UHD+	Sensor Size	1/2	Shutter	Rolling shutter	max. fps in Freerun Mode	3.2	Binning Modes	Color	Subsampling Modes	Color	Sensor Model	MT9J003STM	Pixel size	1.67 μm	Optical Size	6.413 mm x 4.589 mm	Optical sensor diagonal	7.89 mm (1/2.03")
Sensor																																			
Sensor Technology	CMOS Mono																																		
Manufacturer	Aptina																																		
Resolution (MPix)	10.55																																		
Resolution (h x v)	3840 x 2748																																		
ADC	12 bit																																		
Color depth (camera)	8 bit																																		
Pixel Class	UHD+																																		
Sensor Size	1/2																																		
Shutter	Rolling shutter																																		
max. fps in Freerun Mode	3.2																																		
Binning Modes	Color																																		
Subsampling Modes	Color																																		
Sensor Model	MT9J003STM																																		
Pixel size	1.67 μm																																		
Optical Size	6.413 mm x 4.589 mm																																		
Optical sensor diagonal	7.89 mm (1/2.03")																																		

Fonte: Manual do fabricante, IDS: UI-1490LE-M-GL

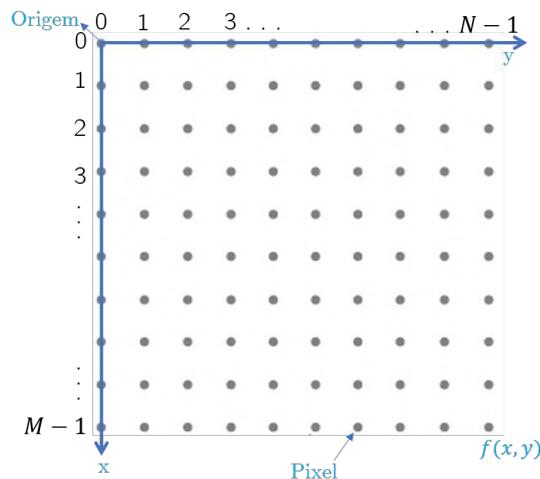
3.1.1 Digitalização e Segmentação de imagens

Uma imagem a ser processada pelo computador deve ser representada usando uma estrutura de dados discreta apropriada, por exemplo, uma matriz. Uma imagem capturada por um sensor é expressa como uma função contínua $f(x, y)$ de duas coordenadas no plano. **Digitalização** de imagem significa que a função $f(x, y)$ é amostrada em uma matriz com M – linhas e N – colunas (figura z9). A **quantização** de imagem atribui a cada amostra contínua um valor inteiro - o intervalo contínuo da função de imagem $f(x, y)$ é dividido em K intervalos.

Quanto mais fina for a amostragem (ou seja, maior M e N) e quantização (maior K), melhor será a aproximação da função de imagem contínua $f(x, y)$ (SONKA; HLAVAC; BOYLE, 2015). A figura 10 combinada duas ilustrações que bem definem esse processo de maneira clara e intuitiva.

É possível notar a correspondência direta do efeito de quantização e digitalização na imagem adquirida pelo sensor e posteriormente processada através por meio de sistemas

Figura 9 – Exemplo gráfico da representação digital de uma função imagem $f(x, y)$ de tamanho $M \times N$, em uma estrutura matricial.



Fonte: Elaborada pela própria autora.

matemáticos bem definidos. Em resumo, uma imagem digital pode ser tratada como uma coleção de elementos de imagem com tamanho finito - pixels que carregam informações sobre o brilho de um local específico na imagem (figura 11).

No que compete ao processamento digital de imagens, há dois fatores importantes de se ter em mente para uma aplicação mais eficaz dos métodos de transformações de imagens e identificação de características: **Amostragem** e **quantização**.

A resolução espacial de uma imagem, basicamente consiste no menor detalhe discernível na mesma, e é determinada, principalmente pela **amostragem**. De forma que, um valor da imagem amostrada $f_s(j\Delta x, k\Delta y)$ é expresso como um valor digital no processamento de imagem. A transição entre valores contínuos da função de imagem (brilho) e seu equivalente digital é a **quantização**.

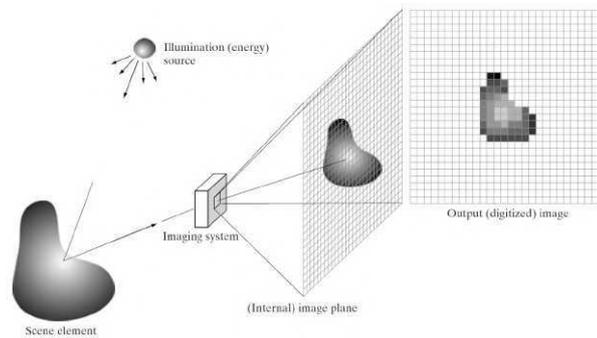
É importante também destacar o conceito complementar da resolução no *nível de cinza*, que refere-se à menor alteração discernível no nível de cinza. Ou seja, para uma quantidade de níveis $N = 8$ tem-se valores da escala de cinza distribuídos linearmente em intensidade de preto (0) e branco ($K - 1$).

3.1.1.1 Correlação espacial e vizinhança

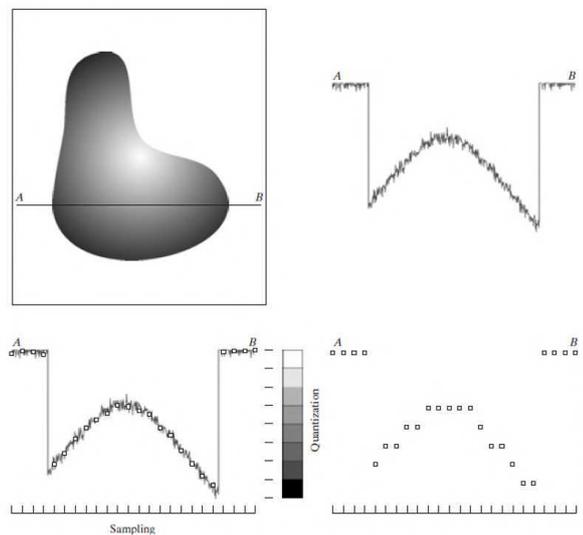
Em segmentação de imagens (HARALICK; SHAPIRO, 1985), o conceito de correlação espacial refere-se a proximidade de amostras (*pixels*) de imagens no plano da imagem. Por exemplo, uma "alta" correlação pode significar que *pixels* em uma vizinhança (adjacentes) têm cores (ou intensidade de brilho) semelhantes. De maneira análoga, uma correlação zero pode significar que *pixels* adjacentes não têm relação de cores.

Ainda mais, correlação não necessariamente é uma medida, do grau em que duas

Figura 10 – Ilustração do processo de aquisição de uma imagem digital (a) e posterior geração de uma imagem digital (b) através de amostragem e quantização, a partir de uma imagem contínua (elemento em escala de cinza).



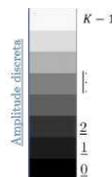
(a)



(b)

Fonte: (GONZALEZ; WOODS, 2002), pg.50 e pg 53

Figura 11 – Exemplo ilustrativo de escala de níveis de cinza para $K=8$.



Fonte: Elaborada pela própria autora.

variáveis concordam no valor real, mas no comportamento geral. Um exemplo está na reação global de um conjunto de pixels que, em uma mesma região sejam similares com referência a alguma característica ou propriedade computacional, tais como cor, intensidade, textura ou continuidade. Regiões adjacentes devem possuir diferenças significativas com respeito a(s) mesma(s) característica(s).

A operação matemática que descreve este indicador, é da forma:

$$R(x, y) = \sum_{m, n} l(m, n)l(m - x, n - y) \quad (3.1)$$

Onde R indica o nível de correlação entre dois *pixels* quaisquer.

Um conceito complementar, mas não menos importante, que deve ser considerado antes qde qualquer oeração global aplicada a um conjunto de elementos da imagem, é o conceito de **vizinhança**. A distância entre dois pontos pode ser expressa como o número mínimo de etapas elementares, na grade digital, necessárias para se mover do ponto inicial ao ponto final. Se apenas movimentos horizontais e verticais são permitidos, é obtida a distância D_4 do "*city-block*" (também chamada de métrica L_1 ou distância de *Manhattan*, devido à analogia com a distância entre dois locais em uma cidade com uma grade retangular de ruas):

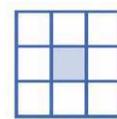
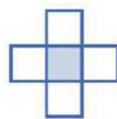
$$D_4((x, y), (m, n)) = |x - m| + |y - n| \quad (3.2)$$

Se os movimentos na direção diagonal são permitidos na grade de digitalização, obtemos a distância D_8 ou a distância do "tabuleiro de xadrez". D_8 é igual ao número mínimo de movimentos do rei no tabuleiro de xadrez de uma parte para outra:

$$D_8 = ((x, y), (m, n)) = \max|x - m|, |y - n| \quad (3.3)$$

As definições de distância das equações 3.2 e 3.3, são definidas na figura 12.

Figura 12 – Ilustração das definições de distâncias: (a) vizinhança D_4 e (b) vizinhança D_8 .



(a) vizinhança-"*Manhattan*" (b) vizinhança-"*chessboard*"

Fonte: Elaborada pela própria autora.

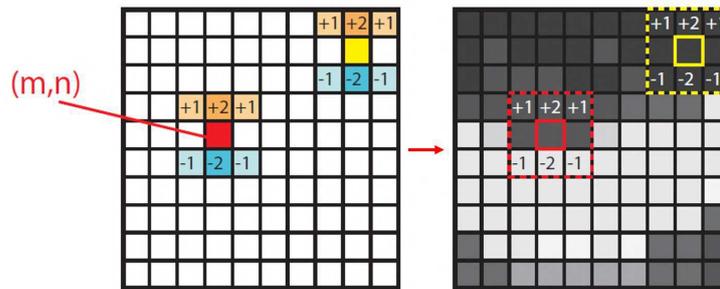
3.1.1.2 Gradiente

Uma "borda" ou aresta, é um outro conceito importante usado na análise de imagens e será empregado nas seções subsequentes. Este conceito é uma propriedade local de um *pixel* e sua vizinhança - é um vetor dado por uma magnitude e direção que nos diz com que rapidez a intensidade da imagem varia em uma pequena vizinhança de um *pixel*.

Imagens com muitos níveis de brilho são usadas para o cálculo das bordas e o gradiente da função de imagem é usado para calcular as bordas. De forma que, a direção da borda é perpendicular à direção do gradiente, que aponta na direção do crescimento mais rápido da função de imagem. O **filtro Sobel** calcula o gradiente da intensidade da imagem em cada ponto, dando a direção da maior variação de claro para escuro e a quantidade de variação nessa direção. Assim, obtém-se uma noção de como varia a luminosidade em cada ponto, de forma mais suave ou abrupta.

Assim, é possível estimar a presença de uma transição claro-escuro e qual a orientação desta. Como as variações claro-escuro intensas correspondem a fronteiras bem definidas entre objectos (ou ruídos), o (operador Sobel) é muito utilizado para a detecção de contornos.

Figura 13 – Representação gráfica da implementação matemática da função gradiente



Fonte: Elaborada pela própria autora.

Assumindo uma imagem monocromática representada em níveis de cinza, dois gradientes são necessários: um gradiente na direção horizontal G_y e outro na direção vertical G_x . Sejam eles:

$$G_x(m, n) = M_x(m, n) * I_k(m, n) \tag{3.4}$$

$$G_y(m, n) = M_y(m, n) * I_k(m, n) \tag{3.5}$$

Assim, calcula-se $G(x, y)$ para a direção vertical, por exemplo, como:

$$G_x(m, n) = M_x(m, n) * I_k(m, n) = \sum_{j=-1}^{+1} \sum_{i=-1}^{+1} M_x(i, j) * I_k(m - i, n - k) \tag{3.6}$$

onde,

$$M_x = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix} \tag{3.7}$$

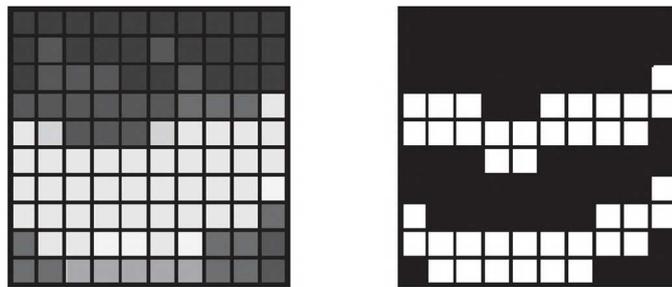
$$M_y = \begin{bmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \quad (3.8)$$

A figura 14 ilustra o processo de convolução das máscaras M_x (Eq. 3.7) e M_y (Eq. 3.8) com a função discreta imagem, *pixel-a-pixel*.

3.1.1.3 Magnitude e *Thresholding*

Um método simples de segmentação de imagens, é o *thresholding*; um algoritmo que define uma métrica constante de intensidade, e, ao analisar cada *pixel* da imagem, altera o mesmo para branco se a intensidade do pixel for maior que o valor da métrica, mas se for menor será preto, gerando assim uma imagem binária (figura ??).

Figura 14 – Representação gráfica da binarização de uma imagem digital pelo método de segmentação por *thresholding*.



Fonte: Elaborada pela própria autora.

Combinar o método de binarização por *thresholding* com o operador *Sobel* é uma estratégia utilizada para encontrar bordas em imagens, ao comparar-se o nível de *thresholding* T com o módulo de uma função gradiente G , dado por:

$$|G(m, n)| = \sqrt{G_x^2(m, n) + G_y^2(m, n)} \quad (3.9)$$

O módulo de G define o conceito de magnitude, assim os seguintes casos são considerados:

- $|G_x(m, n)|$ grande se (m, n) estiver no limite de uma aresta vertical.
- $|G_y(m, n)|$ grande se (m, n) estiver no limite de uma aresta horizontal.
- $|G_x(m, n)| > T$ se (m, n) estiver no limite (borda) da aresta.

Obs: As referências de orientações horizontal e vertical estão conforme a figura 9.

Todos os conceitos definidos na seção 3.1.1, serão utilizados durante todo este trabalho, considerando a manipulação direta na matriz de pixels de uma função imagem $f(x, y)$.

3.1.2 Calibração da Câmera

Quando falamos sobre calibração da câmera, estamos falando sobre o que acontece quando uma câmera, captura em seu sensor, objetos 3D no mundo real e os transforma em uma imagem 2D. Essa transformação não é perfeita e distorções são passíveis de ocorrer, gerando uma correspondência equivocada com um desvio, muitas vezes, não tolerável pela aplicação. Compreender os fundamentos que culminam nessas distorções é fundamental para escolher estratégias que amenizem o erro de transformação de coordenadas de um sistema de mapeamento por visão computacional.

Quando uma câmera "olha" para um objeto, está visualizando o mundo de maneira semelhante à nossa visão. No exemplo da figura 15, é possível observar uma situação em que o foco da câmera está na luz refletida pelos objetos no mundo. Nesse caso, embora seja por um pequeno orifício, a câmera focaliza a luz refletida em um sinal de trânsito 3D e forma uma imagem 2D em sua parte traseira. O modelo que descreve essa simples formação de imagem na parte de trás da câmera é conhecido como **modelo *Pinhole***.

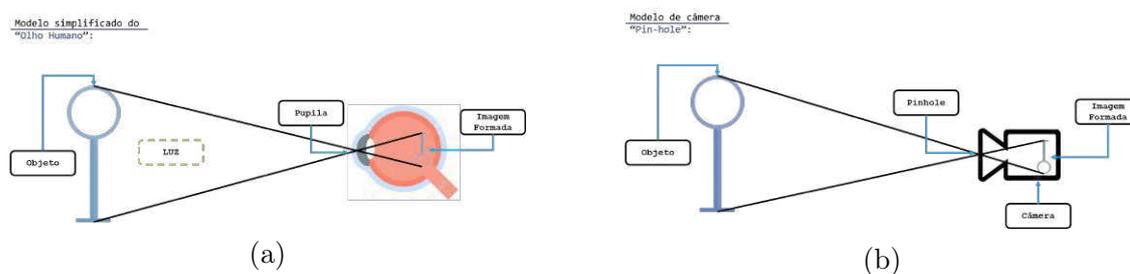


Figura 15 – Ilustração do modelo *Pin Hole* (b) para formação de imagens em câmeras baseado na formação de imagens do olho humano (a). *Fonte: Elaborada pela própria autora.*

O modelo de câmera *pinhole* (ou modelo de câmera projetiva) é um modelo de câmera amplamente utilizado em visão computacional. É simples e preciso o suficiente para a maioria das aplicações. O nome vem do tipo de câmera, como uma câmera obscura, que coleta luz através de um pequeno orifício para o interior de uma caixa ou sala escura. No modelo de câmera *pin hole*, a luz passa por um único ponto, o centro da câmera, C , antes de ser projetada no plano da imagem. A imagem de intensidade 2D é o resultado de uma projeção em perspectiva da cena 3D, que é modelada pela imagem capturada por uma câmera *pin-hole*, como ilustrado na figura 16. As quantidades x , y e z são coordenadas do ponto X em uma cena 3D, e f é a distância focal do orifício até o plano da imagem (SONKA; HLAVAC; BOYLE, 2015).

A recuperação de informações perdidas pela projeção em perspectiva é apenas um problema, principalmente geométrico, da visão computacional - outro é entender o brilho da imagem. As únicas informações disponíveis em uma imagem de intensidade são o brilho do pixel, que depende de vários fatores independentes, como: propriedades de

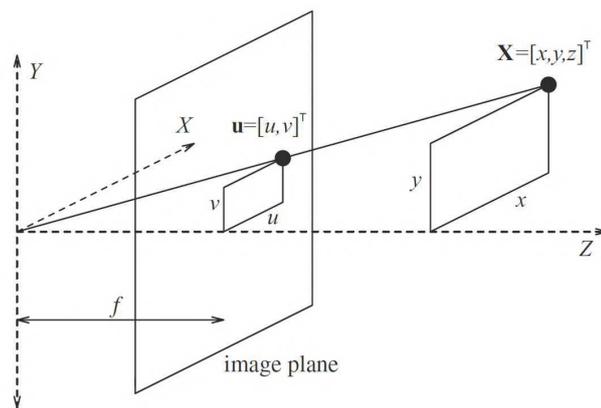


Figura 16 – O modelo da câmera pinhole. O ponto da imagem \mathbf{u} está na interseção do plano da imagem e a linha que une o ponto 3D, \mathbf{X} e o centro da câmera com distância focal f . A linha contínua é o eixo óptico da câmera. Fonte: (SONKA; HLAVAC; BOYLE, 2015), pg.12

refletância da superfície do objeto (fornecidas pelo material da superfície, microestrutura e marcação), propriedades de iluminação da superfície do objeto e a sua orientação em relação ao visualizador e à fonte de luz.

Matematicamente, a transformação dos pontos de objeto 3D, $\mathbf{P}(\mathbf{X}, \mathbf{Y}, \mathbf{Z})$ em $\mathbf{p}(x, y)$ é feita por uma matriz de transformação chamada *matriz da câmera*, utilizada como principal ferramenta para calibrar a câmera.

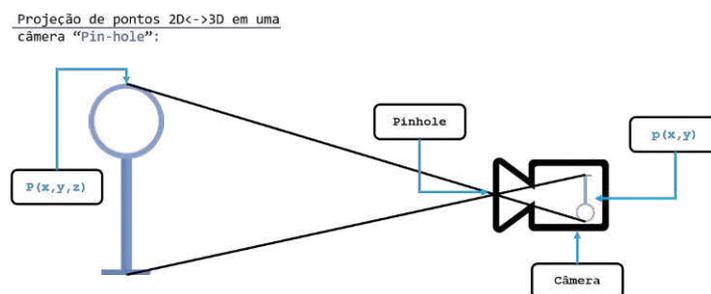
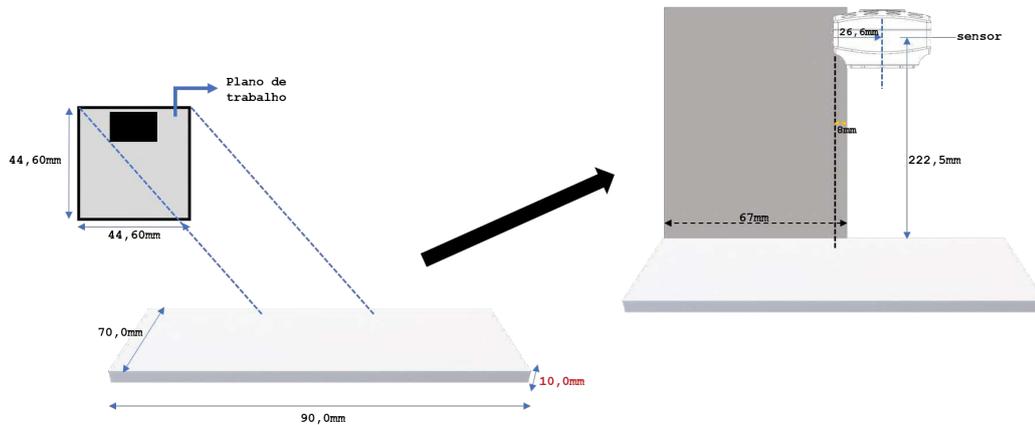


Figura 17 – Exemplo ilustrativo da transformação de coordenadas de mundo 3D \Rightarrow 2D. Fonte: Elaborada pela própria autora.

No entanto, câmeras reais não usam pequenos furos; eles usam lentes para focar em vários raios de luz por vez, o que lhes permite formar imagens rapidamente. Mas, as lentes também podem causar distorção. Sendo estas classificadas em dois tipos: radial e tangencial.

Distorção radial é o tipo mais comum que afeta as imagens, no qual, quando uma câmera captura imagens, linhas retas aparentam estar ligeiramente curvadas. Já a distorção tangencial decorre, principalmente, da lente não está paralela ao plano de imagem, o que faz com que a imagem seja estendida ou inclinada, fazendo com que os objetos pareçam mais afastados ou mais próximos do que realmente são.

Figura 18 – Ilustração com medidas reais do sistema de condicionamento de imagem da estação *Pick and Place*.



Fonte: Elaborada pela própria autora.

Na literatura (BRADSKI; KAEBLER, 2008; SOLEM, 2012), essas distorções podem ser capturada por cinco parâmetros afim de amenizar seu efeito na transformação de perspectiva, estes são conhecidos por **Coefficientes de Distorção**, cujos valores refletem a quantidade de distorção radial e tangencial em uma imagem.

Para este trabalho, as condições do ambiente são consideradas bem definidas e os objetos utilizados são similares em material e tamanho, de forma a padronizar, parte dos fatores que influenciam na intensidade dos pixels. E as imagens são todas consideradas estáticas e monocromáticas, representadas por uma função de imagem contínua $f(x, y)$ cujos argumentos são coordenadas dos pixels.

3.1.2.1 Matriz de Transformação e mapeamento 2D \leftrightarrow 3D

As funções nesta seção usam o modelo de câmera pin-hole, no qual, a vista de uma cena tridimensional é formada projetando-se esses pontos 3D no plano 2D da imagem usando uma transformação de perspectiva, segundo:

$$s \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \quad (3.10)$$

Onde,

- (X, Y, Z) são as coordenadas de um ponto 3D no espaço de coordenadas de mundo;
- (x, y) são as coordenadas do ponto de projeção em pixels;
- \mathbf{A} é a matriz da câmera, ou a *matriz de parâmetros intrínsecos*;
- (cx, cy) é o ponto principal, usualmente o cento da imagem;
- f_x e f_y são as distâncias focais expressas em unidades de pixels.

A equação 3.10 pode ser referida por $sp = A[R|t]P$. Onde R é a matriz de rotação, t a matriz de translação, p os pontos da imagem em *pixels*, P os pontos de mundo em milímetros e s o fator de escalonamento para homogeneização entre coordenadas.

Portanto, se uma imagem da câmera é dimensionada por um fator s , todos esses parâmetros devem ser dimensionados (multiplicados/divididos, respectivamente) pelo mesmo fator. A *matriz de parâmetros intrínsecos* não depende da cena visualizada. Assim, uma vez estimada, pode ser reutilizada enquanto a **distância focal for fixa**. A *matriz de rotação-translação* conjunta $[R|t]$ é chamada de *matriz de parâmetros extrínsecos*. É usada para descrever o movimento da câmera em torno de uma cena estática, ou vice-versa, e o movimento rígido de um objeto na frente de uma câmera estática. Ou seja, $[R|t]$ converte coordenadas de um ponto (X, Y, Z) em um sistema de coordenadas, fixo, em relação à câmera.

Neste trabalho, a transformação acima é válida quando $z \neq 0$, segundo:

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = R \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} + t \quad (3.11)$$

Assim,

$$x' = \frac{x}{z} \quad (3.12)$$

$$y' = \frac{y}{z} \quad (3.13)$$

$$x'' = x' \frac{1 + k_1 r^2 + k_2 r^2 + k_3 r^6}{1 + k_4 r^2 + k_5 r^2 + k_6 r^6} + 2p_1 x' y' + p_2 (r^2 + 2x'^2) \quad (3.14)$$

$$y'' = y' \frac{1 + k_1 r^2 + k_2 r^2 + k_3 r^6}{1 + k_4 r^2 + k_5 r^2 + k_6 r^6} + p_1 (r^2 + 2y'^2) + 2p_2 x' y' \quad (3.15)$$

$$(3.16)$$

onde, $r^2 = x'^2 + y'^2$

$$x = f_x x'' + c_x \quad (3.17)$$

$$y = f_y y'' + c_y \quad (3.18)$$

Nota-se que as equações 3.16 contemplam os coeficientes de distorção mencionados na seção anterior, sendo estes, isolados como:

$$\text{coef. distorção radial} = k_1, k_2, k_3, k_4, k_5 \text{ e } k_6$$

$$\text{coef. distorção tangencial} = p_1 \text{ e } p_2$$

Nas funções da biblioteca **OpenCV**, os coeficientes são passados ou retornados como: $(k_1, k_2, p_1, p_2, k_3, k_4, k_5, k_6)$

Portanto, se o vetor contiver quatro elementos, isso significa que $k_3 = 0$. Os coeficientes de distorção não dependem da cena visualizada. Assim, eles também pertencem aos **parâmetros intrínsecos da câmera** e permanecem os mesmos, independentemente da resolução da imagem capturada. Por exemplo, se uma câmera foi calibrada em imagens com resolução de 320×240 , os mesmos *coeficientes de distorção* podem ser usados para imagens de 640×480 da mesma câmera, enquanto f_x, f_y, c_x, c_y precisam ser redimensionados adequadamente.

3.1.2.2 Procedimento

De posse do modelo descrito na seção anterior, implementou-se em código, todas as operações necessárias para a obtenção dos parâmetros extrínsecos e intrínsecos da câmera utilizada neste trabalho.

Um procedimento comum de calibração, indicado pela literatura ([GONZALEZ; WOODS, 2002](#)), consiste em:

1. Obtenção de $m \geq 6$ pontos de mundo (duas equações) com coordenadas conhecidas $(X_i, Y_i, Z_i), i = 1, 2, \dots, m$;
2. Imageamento desses pontos com a câmera em uma dada posição para obter os correspondentes pontos da imagem $(x_i, y_i), i = 1, 2, \dots, m$;
3. aplicação dos resultados nas equações do modelo pin-hole.

Existem muitas técnicas numéricas para encontrar uma solução ótima para o sistema de equações lineares. Neste trabalho foi implementada a função `calibrateCamera()`, que recebe como entrada os pontos de objeto, pontos de imagem (fornecidos pela saída da função `findChessboardCorners()`) e o formato da imagem, então, calcula e retorna:

- **mtx**: A Matriz da câmera, que ajuda a transformar pontos de objetos 3D em pontos de imagem 2D;
- **dist**: coeficiente de distorção;
- **rvecs, tvecs**: Vetores com os valores de rotação e translação com relação à posição da câmera no mundo

O procedimento de calibração adotado neste trabalho foi o método *Zhang*, devido sua extensiva utilização para calibração de câmeras no contexto industrial ([D'EMILIA; GASBARRO, 2017](#)). O método de *Zhang* contempla a calibração através da utilização de padrões de imagens conhecidos para detecção e correção de erros de distorção.

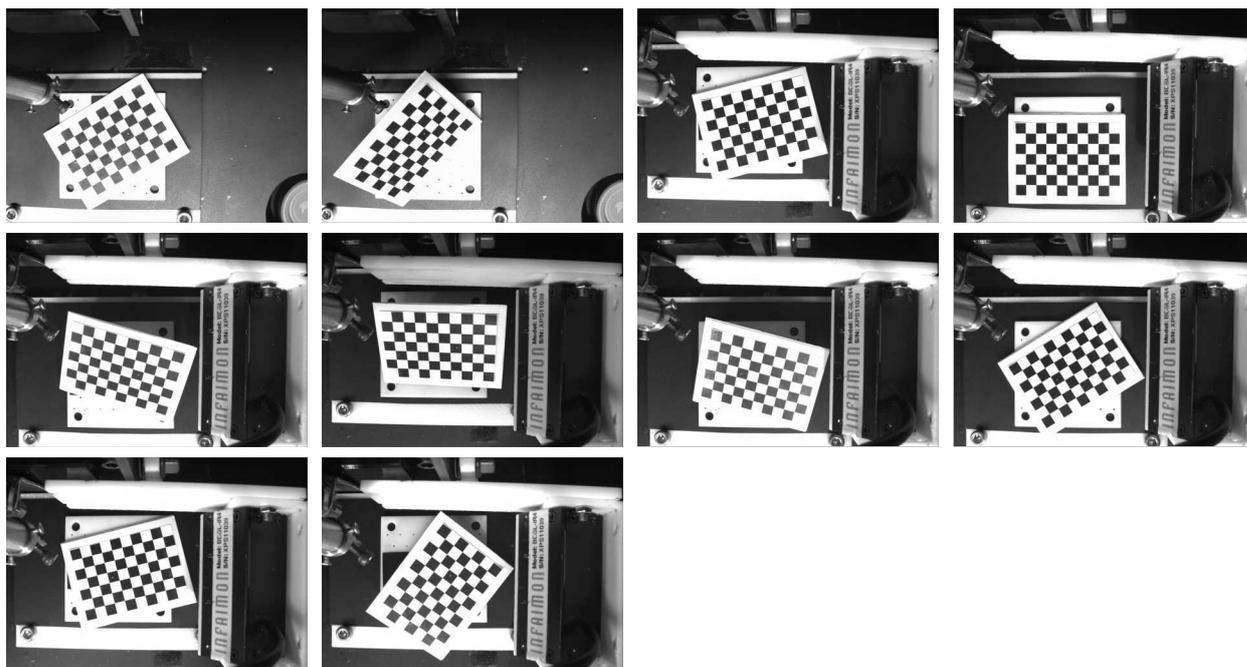


Figura 19 – Conjunto de 10 fotos do padrão de um tabuleiro de xadrez com diferentes perspectivas para a calibração da câmera. *Fonte: Elaborada pela própria autora.*

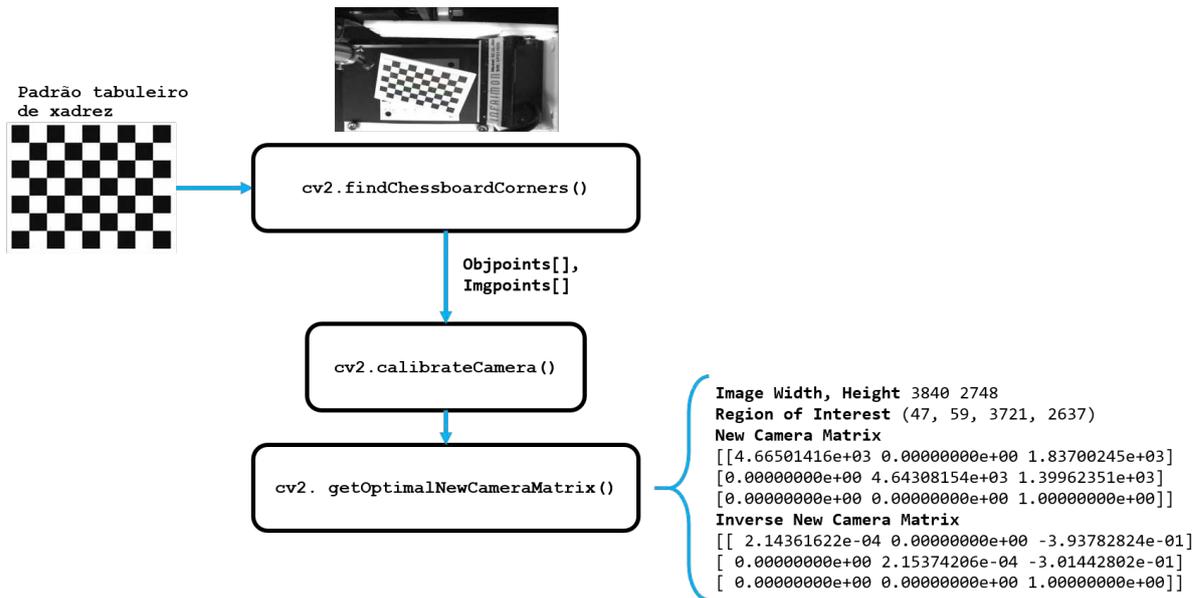
Um tabuleiro de xadrez é indicado como escolha confiável para calibração, porque seu padrão regular de alto contraste facilita a detecção automática, além de ser plano e sem distorções. Um conjunto de 10 imagens (figura 19) com diferentes ângulos do tabuleiro, foram postas paralelas ao plano de imagem da câmera, uma por vez, e então capturadas.

No arquivo `machine_vision_calibration.py` é possível encontrar a função `get_images()` que retorna um `array` de pontos de imagens e outro com pontos de objetos, considerando as dimensões do padrão do tabuleiro utilizado para o mapeamento das coordenadas dos "cantos" da imagem exibida em 2D (`imagepoints`) para as coordenadas 3D dos "cantos" reais e sem distorção do tabuleiro de xadrez, chamados de `objectpoints`, para um tabuleiro 8x6. Dessa forma todos os pontos passam a ser homogeneizados por um fator comum.

Na sequência, esses `arrays` de dados são passados como as entrada da função `cv2.calibrateCamera()`, a qual retorna dados que serão usados como entradas na função `cv2.getOptimalNewCameraMatrix()` encadeada no mesmo ciclo. Dessa forma, o procedimento inicial é finalizado com o retorno da matriz da câmera (*matriz de parâmetros intrínsecos*), no formato da matriz \mathbf{A} da equação 3.10. A figura 20 ilustra esse procedimento, bem como os dados aferidos.

De posse de informações como das coordenadas do centro ótico da câmera (centro da imagem), \mathbf{c}_x e \mathbf{c}_y , foi possível realizar um procedimento de calibração complementar, afim de determinar com melhor precisão o fator de escalonamento s e a *matriz de parâmetros extrínsecos* $[\mathbf{R}|\mathbf{t}]$ do modelo *pin-hole*, definido na equação 3.10.

Figura 20 – Rotina inicial de calibração da câmera para aquisição da matriz de parâmetros intrínsecos.

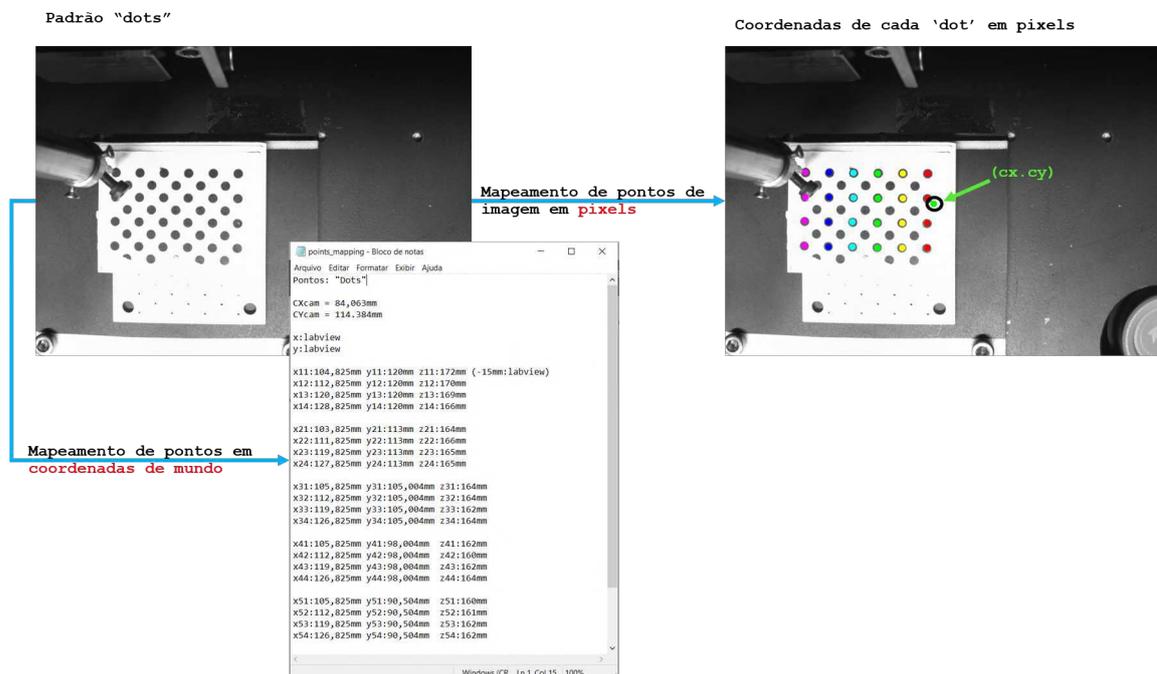


Fonte: Elaborada pela própria autora

Foi usado um novo padrão com círculos de mesmo raio e igualmente espaçados, de forma que as coordenadas de seus respectivos centros puderam ser facilmente mapeadas em coordenadas de *pixels*, como ilustrado na figura 21. Neste mesmo mapeamento foi computado o centro da imagem (cx, cy) e localizado na mesma.

Com o auxílio da própria pinça de "pick" ativada pelo controle manual da interface no *Labview*, foram adquiridas as coordenadas de mundo de cada centro $c(x_i, y_i)$ para os i círculos. A partir deste procedimento conclui-se a aquisição de todos os dados que compõem o modelo da equação 3.10, de forma que o fator de escalonamento s foi facilmente extraído e salvo para a realização das futuras transformações de coordenadas que farão a correspondência entre a localização de objetos de interesse identificados na imagem e sua posição real na base de operações da máquina. Em termos dos scripts, a função que implementa essas operações é a *evaluating_calibration_accuracy()*, encapsulada na classe **machine_mapping**, a qual também possui a função *imagePoints_to_worldPoints()* que implementa a transformação de coordenadas em pixels para coordenadas reais de mundo.

Figura 21 – Padrão "dots" para reforço de precisão e confiabilidade do procedimento de calibração.



Fonte: Elaborada pela própria autora.

3.2 Considerações Finais

O Capítulo 3 resumizou os principais conceitos e métodos da fundamentação teórica de processamento de imagens digitais que serão implementados nos algoritmos utilizados no capítulo 5. Além de destacar a metodologia de calibração da câmera como uma operação fundamental e inerente ao processo de correta correlação de coordenadas entre o plano da imagem e as coordenadas cartesianas do plano de trabalho da máquina.

4 Procedimento experimental

Esta seção trata do procedimento realizado para o reconhecimento automático de dispositivos fotônicos sobre a área de trabalho (base quadrada de 44,60mm de lado) da estação *Pick and Place*.

4.1 Reconhecimento e mapeamento dos dispositivos

Os materiais mais utilizados na fabricação de dispositivos fotônicos - fosfeto, silício sobre isolador, niobato de lítio e etc., para citar alguns - apresenta altos índices de refletividade com diferentes escalas, em contrapartida aos baixos níveis de contraste, além de muitas vezes, possuir elevada transparência. Esses fatores combinados, convergem em um grande desafio para o tratamento de imagens desses elementos. Para este trabalho foi selecionado um grupo de elementos semelhantes, onde tem-se por semelhante, aqueles que foram fabricados na mesma *wafer*, com tamanho e marcas litográficas iguais, figura 22b.

Figura 22 – Dispositivos fotônicos com diferentes padrões de litografia, bem como materiais de fabricação.



(a) Gel *box* com todos os padrões de dispositivos fotônicos disponibilizados pela empresa *Idea! Electronic Systems*.

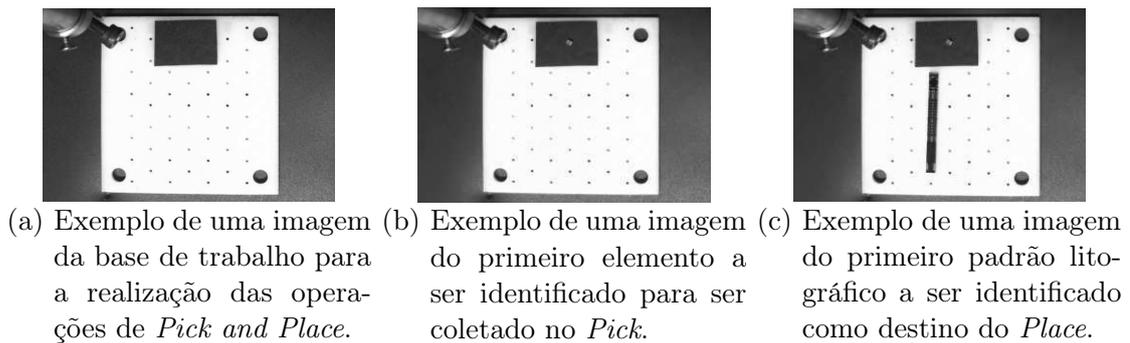


(b) Padrão litográfico escolhido para a primeira versão do sistema de visão computacional com reconhecimento automático.

Fonte: Elaborada pela própria autora.

Após uma vasta investigação na literatura (J.R.PARKER, 2011), foram selecionados alguns algoritmos para a realização de manipulações morfológicas, segmentação de imagem e de operações lógicas com a finalidade de melhoria da imagem no domínio espacial. Esses algoritmos foram combinados com o objetivo de prover um cenário favorável ao reconhecimento individual dos componentes no espaço de trabalho, previamente delimitado, como pode ser visto na figura 23a.

Figura 23 – Conjunto de elementos utilizados e base de referência para realização das operações.

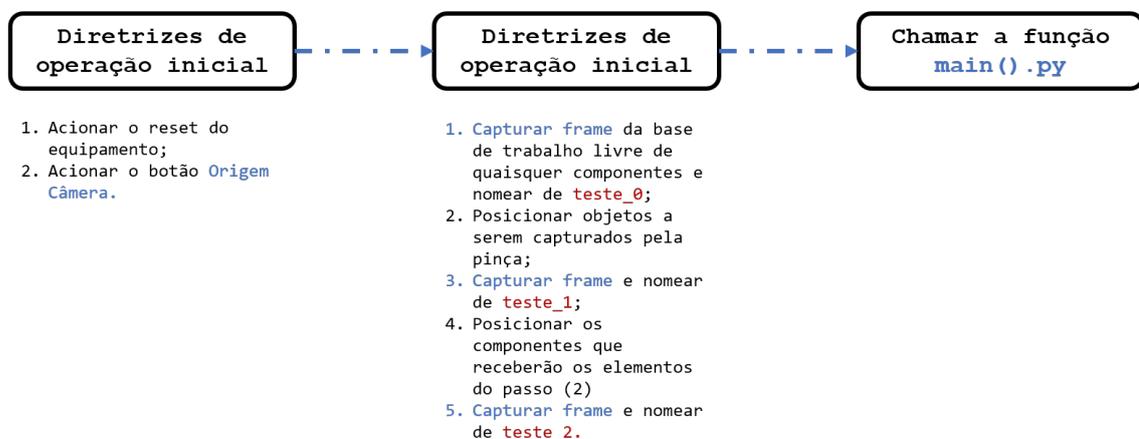


Fonte: Elaborada pela própria autora.

O processo de reconhecimento de dispositivos dar-se-á com a chamada da função *main.py*, que quando inicializada, interage com o usuário por meio de diretrizes de orientação operacionais, conforme ilustrado no diagrama da figura 24.

Figura 24 – Ciclo de diretrizes para correta execução da função de reconhecimento automático de dispositivos fotônicos. Fonte: Autoria Própria.

Estágio 3 -> Operação



Fonte: Elaborada pela própria autora.

A *main.py* implementa um ciclo de execução que divide o sistema de reconhecimento de objetos em duas etapas, através da chamada de objetos da classe *main_loop.py*.

A **Etapa 1** está implementada no objeto **detection** da classe *main_loop()*. De forma que este instancia a função **run_detection()** pertencente a classe de *machine_vision_detection.py*. Esta, finalmente, endereça um conjunto de funções subsequentes para o retorno de uma imagem com os objetos dispostos base de trabalho, reconhecidos e mapeados.

A função *run_detection()* implementa o algoritmo de detecção de bordas (*Edge detection*) baseado em gradiente. Funções disponibilizadas pela biblioteca OpenCV, em Python, foram utilizadas para as transformações e operações matemáticas nas imagens.

O ciclo da etapa 1 segue sumarizado pela sequência abaixo:

1. Diferença absoluta entre as duas primeiras imagens passadas na entrada da *main.py*;
2. Aplica-se um filtro de suavização linear utilizando a função *cv2.Sobel()*;
3. Encontra-se um valor ótimo de *threshold* para binarização com a função *cv2.threshold()*, que usa uma combinação dos métodos de binarização simples com o método Otsu.
4. Por fim aplica-se um filtro gaussiano de suavização com a função *cv2.GaussianBlur*, para a remoção de ruído
5. A saída é uma imagem suavizada e binarizada da diferença absoluta das duas primeiras entradas, a qual é endereçada a função *cv2.findContours()*
6. Os contornos validados, internamente, são evidenciados na imagem original (segunda entrada da função *main.py*), e o centro de massa do perímetro interno da conexão destes contornos, é computado.

O procedimento acima pode ser visualizado graficamente pela ilustração na figura [25](#)

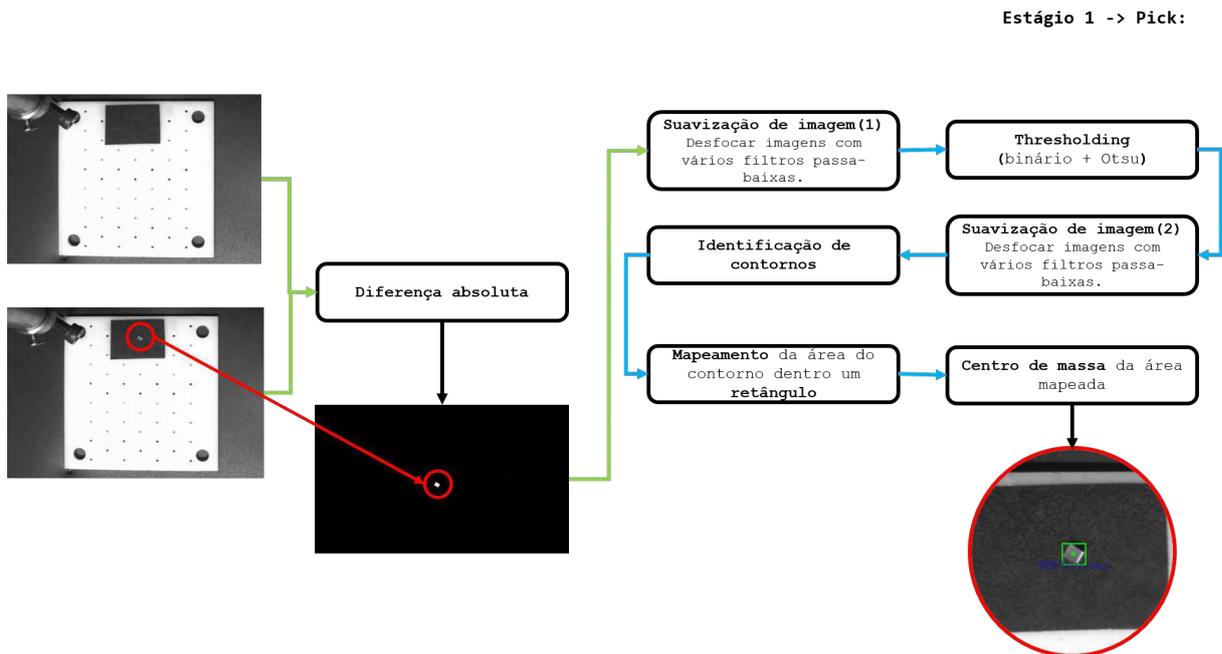
A escolha dos métodos justifica-se na fundamentação matemática das operações implementadas (SOLEM, 2012) por cada um deles, na função imagem $f(x, y)$.

A principal técnica utilizada na **Etapa 1** de reconhecimento é conhecida na literatura por *Edge Detection*: processo de identificação de mudanças "nítidas" no brilho (valores contínuos da intensidade dos pixels) da função imagem $f(x, y)$. Neste trabalho destacaram-se os métodos de *thresholding*, operado Sobel e média gaussiana. Sendo estes próprios de uma *Detecção de Borda Baseada em Gradiente*.

Considerando as funções utilizadas da biblioteca OpenCV e sua maneira de implementação, as técnicas podem ser resumidamente definidas como:

- **Thresholding**: Para este trabalho foi utilizada a segmentação por *thresholding* simples, com métrica pré-definida. O algoritmo atua globalmente para todos os pixels, de

Figura 25 – Ilustração do fluxo de operações empregadas no reconhecimento do objeto a ser capturado pela pinça no procedimento de Pick.



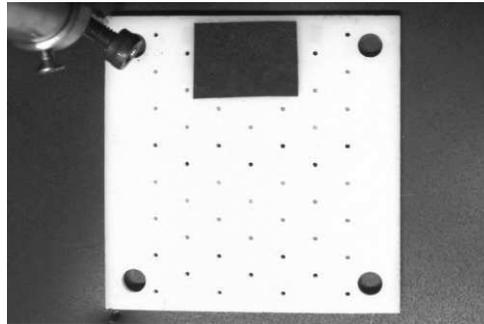
Fonte: Elaborada pela própria autora.

forma que o mesmo valor limite é aplicado. Se o valor do pixel for menor que o limite, ele será definido como 0, caso contrário, será definido como um valor máximo. O primeiro argumento da função é a imagem de origem, que deve ser uma imagem em escala de cinza. O segundo argumento é o valor limite usado para classificar os valores de pixel. O terceiro argumento é o valor máximo atribuído aos valores de pixel que excedem o limite (255 foi usado neste trabalho).

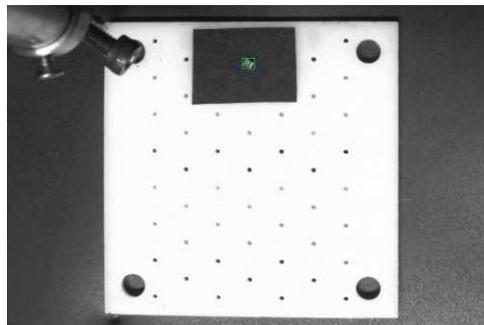
- **Image Blurring (Image Smoothing):** Em OpenCV a suavização da imagem pode ser obtida por diferentes técnicas. Neste trabalho foi utilizada a técnica do filtro gaussiano (`cv2.GaussianBlur()`), o qual utiliza um filtro passa-baixas como um *kernel* (máscara) gaussiano e opera com uma convolução linear deste com a imagem, conforme o princípio da equação 3.6, porém sem a utilização de gradiente. O operador Sobel (`cv2.Sobel()`) é outra técnica utilizada e este implementa as equações da seção 3.1.1.2.
- **Contour Detection:** No OpenCV, encontrar contornos é como encontrar objetos brancos de fundo preto. Portanto, o objeto a ser encontrado deve ser branco e o fundo, preto. Cada contorno individual é uma matriz *Numpy* de (u, v) coordenadas dos pontos de contorno do objeto.

O resultado da rotina implementada, ilustrada na figura 25, pode ser observado na figura 26b.

Figura 26 – Capturas da câmera: (a) *Background* e (b) Imagem de saída após rotina de detecção com emprego da técnica *Edge Detection*.



(a)



(b)

Fonte: Elaborada pela própria autora.

O trabalho de reconhecimento e identificação dos objetos é finalizado em uma segunda etapa, que faz uso do método *Template Matching* para reconhecer o padrão da base de deposição escolhida para ser realizado o procedimento de "*Place*".

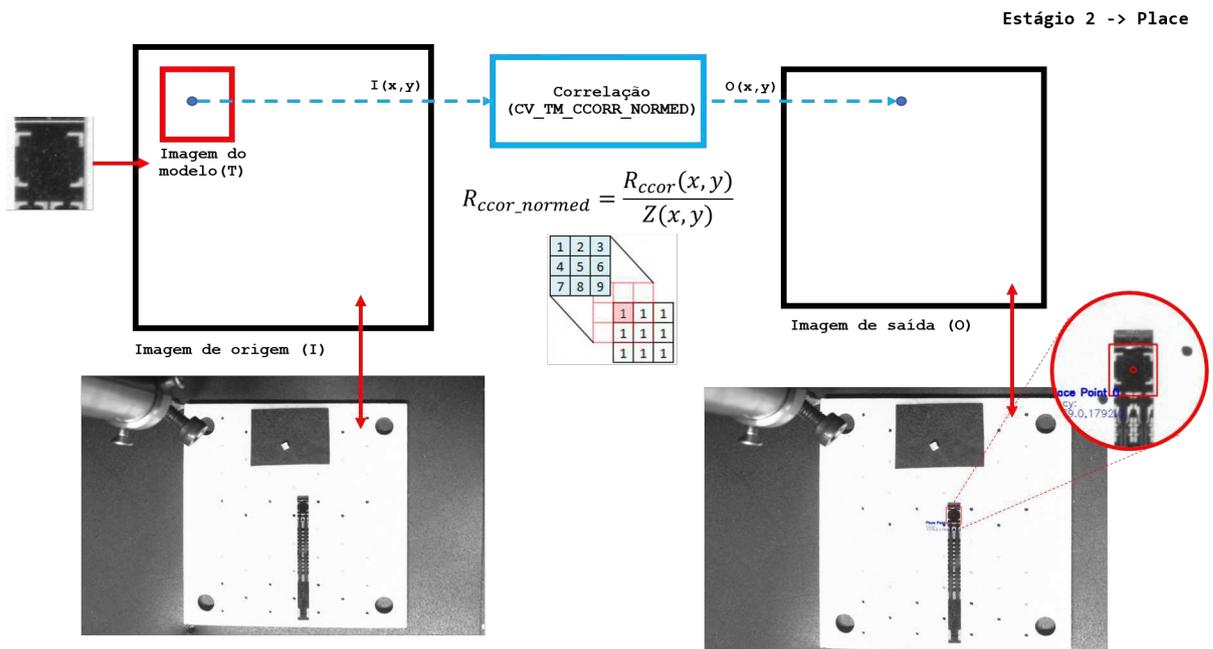
A correspondência de modelo (*template matching*) é uma técnica para localizar áreas de uma imagem que correspondam (são semelhantes) a uma imagem de modelo (*patch*). Utilizar este método envolve dois componentes principais:

- A imagem de origem (I): Aquela na qual espera-se encontrar uma correspondência com a imagem do modelo
- A imagem do modelo/*template* (T): A imagem do *patch* que será comparada à imagem do modelo de referência.

De forma que o objetivo seja detectar a área de correspondência mais alta.

O método consiste em realizar uma convolução 2D entre o *patch* e a imagem fonte, isto significa que move-se o *patch* por um pixel de cada vez (da esquerda para a direita, de cima para baixo) através da imagem original. Em cada local de T sobre I, uma métrica é calculada para representar quão o quão bem o modelo corresponde à imagem naquela posição (ou quão semelhante o *patch* é àquela área específica da imagem de origem). A

Figura 27 – Ilustração do método *template matching* para localização da característica desejada nos dispositivos fotônicos.



Fonte: Elaborada pela própria autora.

figura 27 ilustra o modelo implementado na função `cv2.matchTemplate()` do OpenCV, com seis métodos de correlação disponíveis. Onde entende-se por correlação o exposto na seção 3.1.1.1. Neste trabalho, o método utilizado foi o de **Cros Correlação Normalizada** (`img_gray,template,cv2.TM_CCOEFF_NORMED`).

O procedimento acima pode ser visualizado graficamente pela ilustração na figura 27

Matematicamente, o método **Correspondência do coeficiente de correlação** (método = **CV_TM_CCOEFF**), endereça a correspondência da média de um modelo (T) relativamente à média da imagem de origem (I). Portanto, uma correspondência perfeita será 1 e uma incompatibilidade perfeita será -1; um valor 0 significa que não há correlação (alinhamentos aleatórios). Sendo, na função do OpenCV, implementadas as seguintes equações:

$$R_{ccoeff}(x, y) = \sum_{x', y'} [T'(x', y') \cdot I'(x + x', y + y')]^2 \quad (4.1)$$

$$T'(x', y') = T(x', y') - \frac{1}{(w \cdot h) \sum_{x'', y''} T(x'', y'')} \quad (4.2)$$

$$I'(x + x', y + y') = I(x + x', y + y') - \frac{1}{(w \cdot h) \sum_{x'', y''} I(x + x'', y + y'')} \quad (4.3)$$

Com a normalização, é possível reduzir os efeitos das diferenças de iluminação entre o modelo e a imagem, neste caso, o coeficiente de normalização é definido conforme:

$$Z(x, y) = \sqrt{\sum_{x', y'} T(x', y')^2 \cdot \sum_{x', y'} I(x + x', y + y')^2} \quad (4.4)$$

Assim, o valor para o método que fornece os cálculos normalizados será,

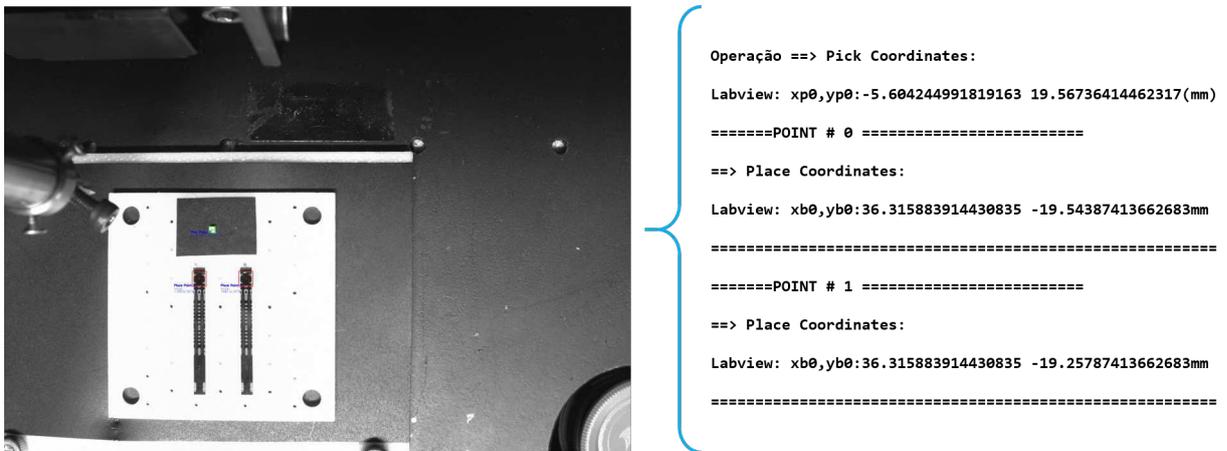
$$R_{ccoeff_normed}(x, y) = \frac{R_{ccoeff}(x, y)}{Z(x, y)} \quad (4.5)$$

Depois de usar o **cv2.MatchTemplate()** para obter uma imagem de resultado correspondente, pode-se então aplicar o resultado a função **cv2.MinMaxLoc()** para encontrar a localização da melhor correspondência. Uma boa correspondência deve ter boas correspondências por perto, pois pequenos desalinhamentos do modelo não devem variar muito os resultados para correspondências reais. A procura da melhor "colina" correspondente pode ser feita suavizando ligeiramente a imagem resultante antes de procurar os métodos de correspondência máxima (para correlação ou coeficiente de correlação) ou mínima (para diferença quadrada). Os operadores morfológicos também são empregados nesse contexto, para manipulações úteis (DEEN et al., 2010).

Portanto, ao combinar os dois estágio de reconhecimento, deve-se obter uma imagem de saída rotulada como é exposto na figura 28, bem como os dados de indicação da posição de cada rótulo, ou seja, de cada elemento identificado na área de trabalho da máquina.

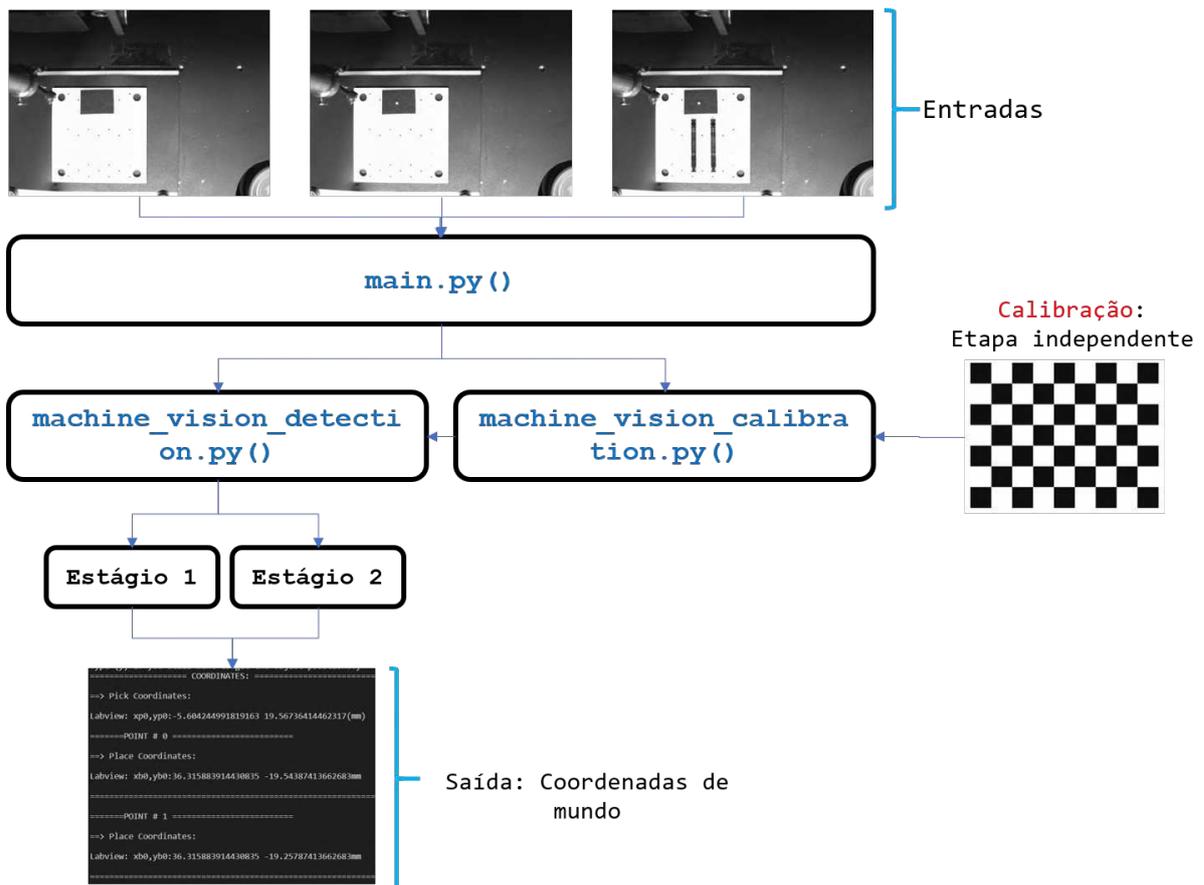
Uma visão geral do sistema de reconhecimento de dispositivos por imagem, no que diz respeito ao fluxo de operações e ciclos de execução do processo completo, é ilustrada na figura 29.

Figura 28 – Exemplo de imagem de saída após realização completa do algoritmo de reconhecimento de dispositivos na área de trabalho.



Fonte: Elaborada pela própria autora.

Figura 29 – Diagrama ilustrativo do fluxo de execução do processo de calibração da câmera e mapeamento dos dispositivos fotônicos em uma máquina posicionadora do tipo *Pick and Place*.



Fonte: Elaborada pela própria autora.

4.2 Tratamento de exceções e erros

Apesar de contemplar o funcionamento automático de um equipamento, o tratamento de erros em dados configura uma medida prudente e preventiva que contribui com o aumento de confiabilidade do sistema, uma vez que inúmeras operações lógicas e matemáticas estão sendo processadas e alterações em leituras podem ocorrer em alguma transação do fluxo de dados do sistema.

Este tratamento foi empregado em laços de exceções à nível de código, e os mais relevantes podem ser listados a seguir:

- **Limitação de deslocamento dentro da área de trabalho:** As posições relativas, resultados das transformações de coordenadas entre os pontos identificados pelo algoritmo de reconhecimento na imagem e sua correspondência em coordenadas de mundo, é restrita ao tamanho da base de operação. Quantidades de deslocamento que somem valores fora desse perímetro impedem que o arquivo com coordenadas seja gerado, impedindo assim, a atualização de movimento na máquina.

Um exemplo, no qual o deslocamento na direção y foi intencionalmente alterado para a condição de exceção, é contemplado na figura 30.

Figura 30 – Exemplo de um caso de exceção "**Limitação de deslocamento dentro da área de trabalho**".

```

deltaxPick = (WpPick[0,1]-WPreference[1])/1000
#deltayPick = (WpPick[0,0]-WPreference[0])/1000

→ deltaxPick = -45

if (deltaxPick>44.6 or deltaxPick<-44.6):
    print("Error: Invalid data")
    exit()
elif(deltayPick>44.6 or deltaxPick<-44.6):
    print("Error: Invalid data")
    exit()
else:pass

print("Movement Amount:\n")
print("==> Pick Point (x,y):\n")
print("Labview(x,y):" + str(deltaxPick),str(-deltayPick)+"mm\n")

```

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
libpng warning: sBIT: invalid
OSu defined threshold value is 76
valid contours 1
d:\Docs\github\machinevisionIdea\Vision System\MachineVisionSystem\machine_vision_detection.py:187: RuntimeWarning: coroutine 'sleep' was never awaited
  sleep(0.1)
RuntimeWarning: Enable tracemalloc to get the object allocation traceback
----- COORDINATES: -----
→ Error: Invalid data
(opencv-env) 0:\Docs\github\machinevisionIdea\Vision System\MachineVisionSystem\

```

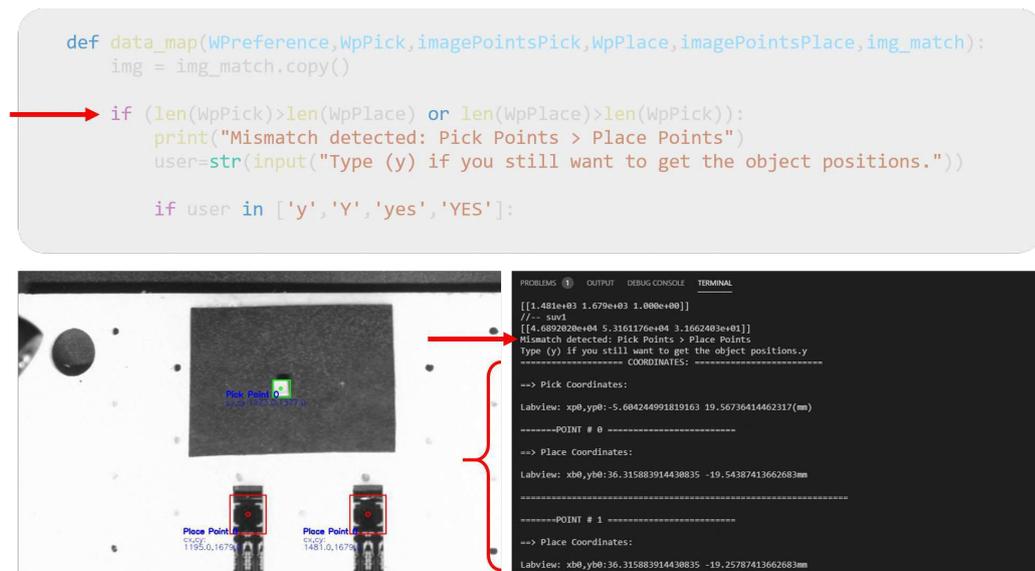
Fonte: Elaborada pela própria autora.

- **Mismatch** entre o número de componentes para o "Pick" e o número de bases para o "Place": Se o número, reconhecido, de componentes que serão manipulados pela pinça diferir da quantidade de bases identificadas para a deposição

destes, é exibida uma *warning* para o usuário de forma que ele pode decidir se a operação continua e "pula" a dupla em *mismatch* ou se o processo é abortado.

Um exemplo, no qual o número de bases para a deposição de componentes é diferente (maior) do que a quantidade de componentes, é contemplado na figura 31.

Figura 31 – Exemplo de um caso de exceção "*Mismatch* entre o número de componentes para o "Pick" e o número de bases para o "Place".

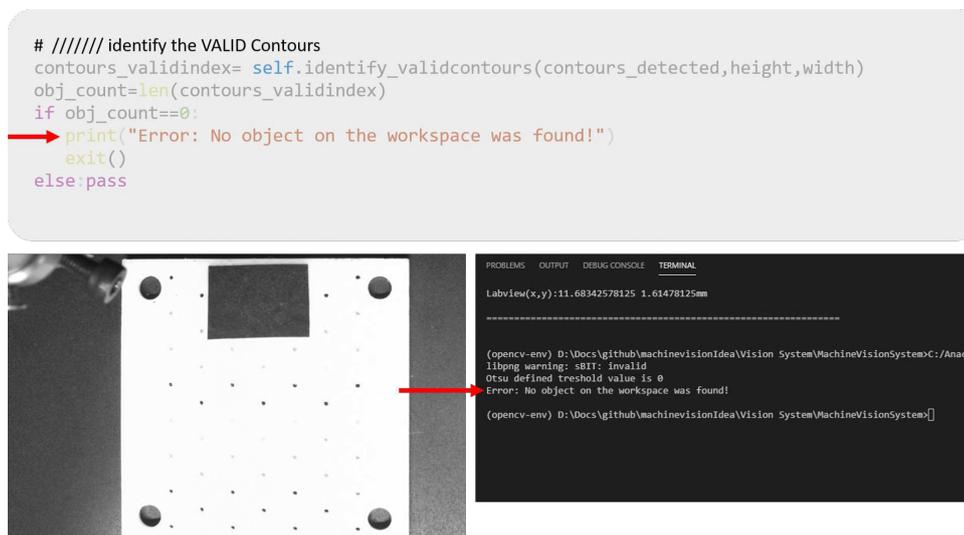


Fonte: Elaborada pela própria autora.

- **Obstrução do campo de visão da câmera ocasionando falha na visualização dos objetos:** Se nenhum objeto para o "Pick" for identificado, o processo é abortado e uma mensagem de erro informa ao usuário o motivo;

Para testar esse erro foram utilizadas duas imagens do espaço de trabalho sem objetos, de forma que, quando o algoritmo realiza a diferença entre as duas primeiras imagens não encontra contornos válidos, o que também pode acontecer para o caso de imagens ruidosas, com superexposição ou sombras. O algoritmo faz um tratamento para os falsos positivos no que diz respeito as bordas. Em todo caso, não encontrar elementos para o primeiro estágio (Pick), aborta a operação por completo.

Figura 32 – Exemplo de erro para a falta de objetos no espaço de trabalho ou falha no reconhecimento.



Fonte: Elaborada pela própria autora.

- **Restrição do tipo de dados na saída da função de reconhecimento para o arquivo TXT:** Considerando as estritas requisições de precisão para a aplicação em foco, os dados de saída devem ser, unicamente, do tipo float, contidos no conjunto de números reais.

4.2.1 Interface Homem-Máquina

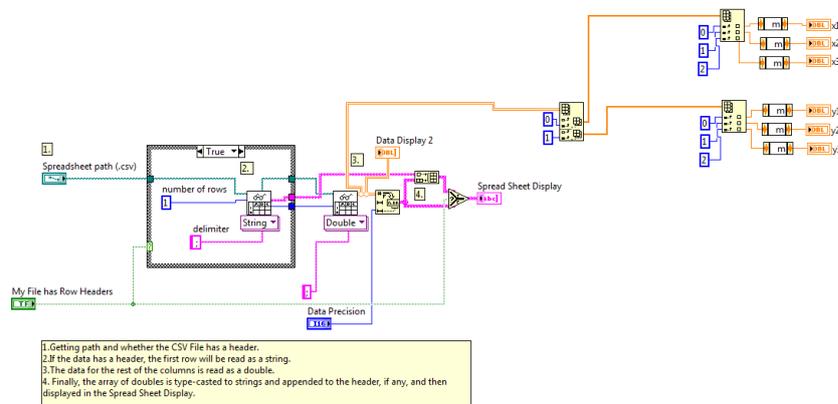
4.2.1.1 LabVIEW

O LabVIEW é um *software* de programação gráfica da empresa *National Instruments* que dispõe de linguagem de programação baseada em diagrama de blocos. Um projeto elaborado no LabVIEW contém programas chamados de VIs (*Virtual Instruments* – Instrumentos Virtuais), que são compostos por um painel frontal, em que se tem a interface gráfica com o usuário, e por um diagrama de blocos, que é o ambiente de programação em si, assim como se pode visualizar na figura 33. Esse *software* é extensamente empregado em aplicações de instrumentação eletrônica e processos industriais. (ELLIOTT et al., 2007). É possível realizar a conexão com um servidor OPC, conectando-se assim com os endereços de memória do CLP P7C302, por meio da criação de um servidor de entrada/saída (*I/O server*) que possibilita a utilização das etiquetas dos dispositivos no LabVIEW, possuindo seu valor atualizado a cada 100 ms.

4.2.1.2 Funcionalidades

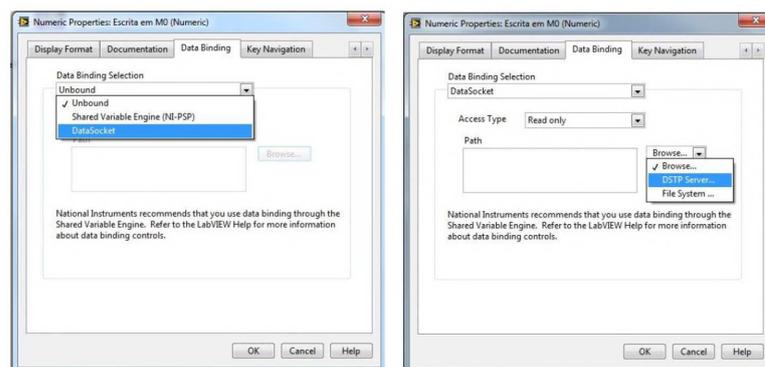
No que diz respeito a interação do usuário com a máquina posicionadora, este dispõe de dois tipos de controle para operação: manual e automático. O foco deste trabalho é otimizar as funcionalidades automáticas do equipamento, no contexto de evitar o seu

Figura 33 – Linguagem de programação em diagrama de bloco para desenvolvimento no software Labview.



Fonte: Banco de imagens do P&D em fotônica de silício.

Figura 34 – DataSocket para acesso, por parte do Labview, aos itens(I/Os) publicados na camada servidora OPC(SOS), a partir de um arquivo do tipo CSV.



Fonte: Banco de imagens do P&D em fotônica de silício.

manuseio indevido ao endereçar medidas de prevenção de erros operacionais manuais que possam comprometer o fluxo das operações em curso. Portanto, o sistema de visão computacional em desenvolvimento está direcionado ao modo automático de funcionamento, não havendo funções que sejam realiccionadas com a operação manual do mesmo.

Neste contexto, a seção semi-automática da interface com o usuário foi utilizada para a execução dos testes e receberá os dados de saída da função *main.py*, os quais devem ser inseridos nos campos (X,Y,Z) correspondentes às direções de movimento dos carros mecânicos e da pinça de vácuo.

A figura 35 destaca o conjunto de "abas" que compõe a seção de operação semi-automática da IHM.

Figura 35 – Interface de interação com o usuário para operações manual e automática da estação *Pick and Place*.



Fonte: Banco de imagens do P&D em fotônica de silício.

4.3 Considerações Finais

Este capítulo contemplou, inicialmente, os métodos, técnicas e processos utilizados para o reconhecimento por imagem dos dispositivos fotônicos em uma máquina posicionadores. As estratégias de mitigação de erros provenientes das próprias saídas das funções do sistema, bem como de condições inadequadas de formação de imagens foram abordadas como uma ação preditiva do processo. Considerando as centenas de operações matemáticas envolvidas em todo o processo de transformação e manipulação das funções imagens, é possível que a unidade de processamento emita eventuais erros como: tipos inconsistentes de dados, confusão numérica por atrasos, *bugs* no sistema ou simplesmente uma passagem errada de parâmetros; o que, apesar de bastante raro, não está livre de ocorrer. Por fim, foram abordadas as possibilidades de operação do usuário sobre o sistema de posicionamento uma vez que este detém os dados de posição dos elementos após a utilização da função principal de reconhecimento e mapeamento automáticos.

Os dados gerados com o sistema descrito neste capítulo influenciam diretamente a interação do usuário do equipamento, uma vez que habilita a operação automática ou semi-automática de posicionamento, contribuindo com a escalabilidade e confiança do processo.

Todos os textos de códigos descritos e implementados nesta seção podem ser encontrados no endereço do repositório GitHub: <<https://github.com/barbaranicolu/PickandPlaceCV.git>>.

5 Conclusões

A realização deste trabalho atendeu os requisitos funcionais da proposta de desenvolvimento de um sistema para a correta correlação do mundo adimensional dos pixels com o do eixo mecânico dos motores-de-passo do equipamento, o que também incluiu a calibração relativa da câmera, possibilitando a execução da tarefa final de reconhecimento automático de dispositivos fotônicos e mapeamento em coordenadas de mundo destes.

O sistema apresentado e descrito contemplou o objetivo principal de mapeamento dos dispositivos fotônicos para a realização da funcionalidade de *Pick and Place*, que compõe uma das operações na indústria de *packaging fotônico* na tendência da concepção de circuitos integrados em fotônica de silício. Foi possível reconhecer, de maneira generalista, os padrões litográficos dos dispositivos conforme o objetivo da operação desejada, assim como endereçar as peças posicionadas para a coleta, com um nível satisfatório de generalidade, considerando os problemas inerentes às diferenças de refletividade das mesmas, em decorrência dos diferentes materiais desses componentes. Testes de validação dos dados de saída foram implementados para majorar a confiabilidade ao sistema, bem como testes de erro, afim de impedir um funcionamento indesejado dentro dos limites da área de trabalho. Além disso, diferentes possibilidade de atuar com um grupo de dispositivos foram consideradas; ao permitir a manipulação de componentes fora do padrão usual de rotinas de posicionamento, no caso de uma necessidade de testes rápidos pelo usuário. Como por exemplo, ter na bancada um número divergente entre bases receptoras e componentes a serem coletados.

Durante o desenvolvimento da solução de mapeamento automático, eventuais limitações foram identificadas, como: As limitações mecânicas da máquina, que não possui todos os sensores bem fixados, podendo ocasionar uma impressão errada dos limites de precisão no eixo z; uma vez que a referência pode ser alterada com a mudança de posição do sensor. Um outro problema foi a implementação das rotinas automáticas geradas pelo *script*. Apesar de receber posições corretas e em sequência pré-definida para um ciclo de atividades, a execução da rotina, por parte do *Labview*, nem sempre executa todas as entradas que recebe, alguns eixos ficam travados quando sequências de comandos são enviadas. De forma que faz-se necessário uma investigação no código fonte embarcado no controlador, afim de identificar como esses sinais são tratados e enviados aos motores-de-passo.

5.1 Perspectivas de Trabalhos Futuros

Mediante a experiência com as ferramentas e visão computacional, fica como sugestão um investimento em um sistema controlado de iluminação para a máquina, dado os diferentes materiais dos dispositivos, que dificulta um equilíbrio ótimo de contraste em uma bancada de trabalho muito diversificada. Outra sugestão diz respeito ao desenvolvimento de um algoritmo que faça o reconhecimento em tempo real da localização da pinça de vácuo, e por meio do *tracking* oferecer *feedback* de localização; permitindo assim, maior flexibilidade mediante a ocorrência de exceções. Dessa forma é possível endereçar questões como: *Se o vácuo falhar e um dispositivo cair da pinça, qual a última posição em que esta estava? É possível recuperar o dispositivo e continuar o procedimento ?*

Por fim, a ultima sugestão diz respeito ao investimento em uma investigação teórica para a implementação de um banco de dados contemplando as centenas de possibilidades de alinhamento de dispositivos dentro de uma *golden box*, de forma a favorecer o equilíbrio térmico, mecânico e elétrico dos circuitos embarcados, haja vista sua natureza milimétrica ou nanométrica e o nível de precisão requeridos para o adequado funcionamento. Um banco de dados de imagens e dados de desempenho térmico, por exemplo, poderia treinar uma rede neural com a correlação destes últimos de maneira a "aprender" potenciais regiões de posicionamento de um "x-tipo" de componente, que favoreça o acoplamento óptico e término naquele *setup* de uma aplicação específica.

Referências

- BRADSKI, G.; KAEBLER, A. *Learning OpenCV: Computer Vision with the OpenCV Library*. 1005 Gravenstein Highway North, Sebastopol: O’reilly, 2008. First Edition. ISBN 978-0-596-51613-0. Citado na página 19.
- DEEN, N. et al. *On image pre-processing for PIV of single- and two-phase flows over reflecting objects. Experiments in Fluids*, v. 49, p. 525–530, 08 2010. Citado na página 31.
- D’EMILIA, G.; GASBARRO, D. D. Review of techniques for 2d camera calibration suitable for industrial vision systems. *Journal of Physics: Conference Series*, IOP Publishing, v. 841, p. 012030, may 2017. Disponível em: <<https://doi.org/10.1088/2F1742-6596/2F841/2F1/2F012030>>. Citado na página 21.
- ELLIOTT, C. et al. National instruments labview: A programming environment for laboratory automation and measurement. *JALA: Journal of the Association for Laboratory Automation*, v. 12, n. 1, p. 17–24, 2007. Disponível em: <<https://doi.org/10.1016/j.jala.2006.07.012>>. Citado na página 35.
- GONZALEZ, R.; WOODS, R. *Digital Image Processing*. [S.l.]: Prentice Hall, Inc, 2002. 34-38,66-69 p. ISBN 0-201-18075-8. Citado 2 vezes nas páginas 13 e 21.
- HARALICK, R. M.; SHAPIRO, L. G. Image segmentation techniques. *Computer Vision, Graphics, and Image Processing*, v. 29, n. 1, p. 100 – 132, 1985. ISSN 0734-189X. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0734189X85901537>>. Citado na página 12.
- J.R.PARKER. *Algorithms for Image Processing and Computer Vision*. [S.l.]: Wiley Publishing, Inc, 2011. Second Edition. 21-38,87-100 p. ISBN 978-0-470-64385-3. Citado na página 26.
- SCHLEIPEN, M. et al. Opc ua industrie 4.0 - enabling technology with high diversity and variability. *Procedia CIRP*, v. 57, p. 315 – 320, 2016. ISSN 2212-8271. Factories of the Future in the digital environment - Proceedings of the 49th CIRP Conference on Manufacturing Systems. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S2212827116312094>>. Citado na página 6.
- SOLEM, J. E. *Programming Computer Vision with Python*. 1005 Gravenstein Highway North, Sebastopol: Creative Commons, 2012. First Edition. ISBN 978-0-596-51613-0. Citado 2 vezes nas páginas 19 e 27.
- SONKA, M.; HLAVAC, V.; BOYLE, R. *Image Processing, Analysis, and Machine Vision*. United States: Cengage Learning, 2015. Fourth Edition. ISBN 978-1-133-59360-7. Citado 4 vezes nas páginas 8, 11, 17 e 18.
- SVEDA, M.; VRBA, R. *Actuator-sensor- interface interconnectivity. Control Engineering Practice*, Pergamon, v. 7, p. 95–100, 1999. Citado na página 2.