

**UNIVERSIDADE DE SÃO PAULO
ESCOLA DE ENGENHARIA DE SÃO CARLOS**

Luiz Akyhito Miyazaka

**Desenvolvimento de um sistema de imagem e controle
para deposição a laser na recuperação de peças metálicas.**

São Carlos

2019

Luiz Akyhito Miyazaka

**Desenvolvimento de um sistema de imagem e controle
para deposição a laser na recuperação de peças metálicas.**

Monografia apresentada ao Curso de Engenharia Elétrica com Ênfase em Eletrônica, da Escola de Engenharia de São Carlos da Universidade de São Paulo, como parte dos requisitos para obtenção do título de Engenheiro Eletricista.

Orientador: Prof. Dr. Jarbas Caiado de Castro Neto

Coorientadora: Dra. Fátima Maria Mitsue Yasuoka

São Carlos

2019

AUTORIZO A REPRODUÇÃO TOTAL OU PARCIAL DESTE TRABALHO,
POR QUALQUER MEIO CONVENCIONAL OU ELETRÔNICO, PARA FINS
DE ESTUDO E PESQUISA, DESDE QUE CITADA A FONTE.

Ficha catalográfica elaborada pela Biblioteca Prof. Dr. Sérgio Rodrigues Fontes da
EESC/USP com os dados inseridos pelo(a) autor(a).

M953d Miyazaka, Luiz Akyhito
Desenvolvimento de um sistema de imagem e controle
para deposição a laser na recuperação de peças
metálicas / Luiz Akyhito Miyazaka; orientador Jarbas
Caiado de Castro Neto; coorientadora Fátima Maria
Mitsue Yasuoka. São Carlos, 2019.

Monografia (Graduação em Engenharia Elétrica com
ênfase em Eletrônica) -- Escola de Engenharia de São
Carlos da Universidade de São Paulo, 2019.

1. Deposição a laser. 2. Peça metálica. 3.
Recuperação de superfície. 4. Processamento de imagem.
I. Título.

FOLHA DE APROVAÇÃO

Nome: Luiz Akyhito Miyazaka

Título: “Visão computacional para identificação de trincas de uma placa metálica em um sistema de deposição à laser”

Trabalho de Conclusão de Curso defendido e aprovado
em 7 / 6 / 19,

com NOTA 7,7 (sete, sete), pela Comissão Julgadora:

Prof. Titular Jarbas Caiado de Castro Neto - Orientador - FCM/IFSC/USP

Dra. Fatima Maria Mitsue Yasuoka - Pesquisadora - BR Labs Tecnologia Óptica e Fotônica

Prof. Associado Marcelo Andrade da Costa Vieira - SEL/EESC/USP

Coordenador da CoC-Engenharia Elétrica - EESC/USP:
Prof. Associado Rogério Andrade Flauzino

SUMÁRIO

1	INTRODUÇÃO	13
1.1	Objetivo e motivação	14
2	FUNDAMENTOS TEÓRICOS	15
2.1	Escala de cinza	15
2.2	Binarização	16
2.3	Filtros	16
2.4	Erosão e dilatação	18
3	MATERIAIS E MÉTODOS	19
3.1	Eletrônica	20
3.1.1	Laser de fibra	21
3.1.2	Sensor de distância	21
3.1.3	Motores de passo e fins de curso	21
3.1.4	Raspberry Pi 3	22
3.2	Computação	23
3.2.1	Python	23
3.2.2	Placa de testes	24
4	RESULTADOS DO TRABALHO	25
4.1	Circuito de integração	25
4.2	Programação	27
4.3	Detecção de trincas com processamento de imagem	29
5	CONCLUSÕES	33
	REFERÊNCIAS	35

LISTA DE FIGURAS

Figura 1 – Binarização de uma imagem (imagem de: <i>Mathworks</i>).	16
Figura 2 – Exemplos de filtro passa-alta e filtro passa-baixa, respectivamente. . . .	17
Figura 3 – Aplicação do " <i>non-local means</i> "(imagem de: <i>Mathworks</i>).	18
Figura 4 – Sistema de deposição à laser.	19
Figura 5 – Diagrama de blocos simplificado do sistema.	20
Figura 6 – Laser de fibra SPI.	21
Figura 7 – Microprocessador – Raspberry Pi 3.	22
Figura 8 – Peça de metal danificada utilizada para os testes da visão computacional.	24
Figura 9 – Esquemático de funcionamento da deposição a laser.	25
Figura 10 – Circuito de controle para a ligação na Raspberry Pi 3.	26
Figura 11 – <i>Printed Circuit Board</i> do circuito de integração do sistema à Raspberry Pi 3.	27
Figura 12 – Programação em python para o chaveamento do motor.	28
Figura 13 – Programação em python para o driver do motor.	28
Figura 14 – Programação em python para a inicialização da câmera.	29
Figura 15 – Imagem processada original e após a transformação em tons de cinza, respectivamente.	30
Figura 16 – Imagem processada após a aplicação do NLM e após a binarização, respectivamente.	30
Figura 17 – Imagem final após o processamento para detecção de trincas.	31

RESUMO

MIYAZAKA, L. A. **Desenvolvimento de um sistema de imagem e controle para deposição a laser na recuperação de peças metálicas..** 2019. 35p.
Monografia (Trabalho de Conclusão de Curso) - Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos, 2019.

O presente documento é referente ao desenvolvimento de um sistema de deposição a laser de material sobre peças metálicas danificadas. A deposição é feita para restaurar a superfície danificada, tornando a peça reutilizável. O sistema em desenvolvimento consiste na integração das partes do processo de deposição utilizando a Raspberry Pi 3, capaz de realizar a interface mecânica com o meio digital. O controle de todo o processo é realizado pela implementação de um código de programa em linguagem python. No trabalho, foram criados os circuitos para integração do laser, sensor de distância e motores de passo à Raspberry e a criação de alguns programas de controle para a movimentação dos motores e utilização de uma câmera. Além disso, foram feitos testes preliminares para observar a aplicabilidade do processamento de imagem para automatizar o sistema de deposição. Programas de detecção das trincas da peça danificada e os resultados dos testes estão presentes neste documento. Mesmo que os resultados não tenham sido satisfatórios quanto à detecção das trincas, foi possível concluir que o processamento de imagem pode ser aplicado ao sistema de deposição a laser para automatizar o sistema. O desenvolvimento do sistema ainda está em andamento e ainda não foi concluído para se realizar os testes definitivos para a recuperação de peças metálicas danificadas.

Palavras-chave: Deposição a laser. Peça metálica. Processamento de imagem. Automação.

ABSTRACT

MIYAZAKA, L. A. **Development of an image and control system for laser deposition in the recovery of metal parts..** 2019. 35p. Monografia (Trabalho de Conclusão de Curso) - Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos, 2019.

This document is concerned in the development of a laser deposition system on damaged metal parts. The deposition is performed to restore the damaged surface, creating a reusable piece. The system in development consists on integrating all parts of the deposition process using Raspberry Pi 3, which perform the link between the mechanical system and the digital media. The control of the whole process is accomplished by the implementation of a program code in python language. In this work, the circuits for laser integration, distance sensor and stepper motors control were created to connect to Raspberry and some control programs for the movement of the motors and usage of a camera were created. In addition, preliminary tests were performed to observe the applicability of image processing to automate the deposition system. Programs to detect the damaged area and test results are described in this document. Even though the results of the detection of cracks were not satisfactory, it was possible to conclude that the image processing can be applied to the laser deposition system to automate the system. The development of the system is still ongoing and has not yet been completed to carry out the definitive tests for the recovery of damaged metal parts.

Keywords: Laser deposition. Metal part. Image processing. Automation.

1 INTRODUÇÃO

O avanço tecnológico cada vez mais presente no nosso cotidiano desperta o interesse de muitas pessoas para a ciência e a pesquisa. A descoberta de novos métodos e novas informações testam a capacidade criativa dos seres humanos no desenvolvimento de propostas inovadoras aos problemas que atingem a sociedade atual. O laser, por exemplo, encontrou muitas aplicabilidades em vários ramos da ciência após a sua descoberta; aplicações estas que são possíveis graças às características peculiares do laser quanto fonte emissora de luz (Bagnato, 2008).

Em vista disso, o presente projeto teve como base um processo de deposição a laser sobre superfícies danificadas utilizando um material em pó. O laser tem o importante papel de fundir o material de deposição para que este, ao entrar em contato com a superfície, seja incorporado ao material (Schmidt et al., 2017). Tal processo é possível graças à alta potência fornecida pelo feixe de luz em curtos intervalos de tempo. O laser pulsado pode atingir potências de ordem de grandeza de $10^{12}W$, emitindo sua energia acumulada ao longo de períodos com intervalos de pico-segundos (Bagnato, 2008). Esse sistema se valoriza nas suas aplicações em indústrias. Diariamente peças de metal podem ser danificadas devido ao seu mal uso ou desgaste com o passar do tempo. A reposição desses materiais se torna dispensável quando comparada às vantagens que a deposição a laser oferece, já que menos material é gasto no reparo, minimizando-se as perdas. Como todo esse processo deve ser preciso e sutil, o uso de visão computacional aplicado à deposição a laser pode atuar diminuindo os riscos de erros do operador.

Depositar pó demais, e em locais não danificados, inutilizará a peça para seu antigo uso. A visão computacional atua detectando as trincas na peça metálica e desenvolve a automação do processo, colocando a subjetividade do operador em segundo plano. Ela nada mais é do que a tentativa de emulação da visão humana no computador. Dessa maneira, é possível que a máquina tome decisões do que será feito no sistema através da codificação em linguagem de máquina (Marengoni & Stringhini, 2009). Tudo isso é executável graças à interface entre o *mundo real* e o *digital* possibilitada pelo uso de um computador de placa única. Este computador de placa única é um integrado de circuitos que possui um sistema de memória, microprocessador, uma unidade de processamento gráfico e conjuntos de portas de entrada e saída. Ele é capaz de gerir informações provenientes do exterior, através do uso de periféricos ligados às suas portas de entrada e saída; seu sistema é bastante conveniente e pode controlar inúmeras funções, desde leds e portas lógicas até câmeras e motores com uma codificação em linguagem de máquina. Além disso, seu núcleo de processamento nos possibilita acessar a internet e realizar as mesmas funções de um computador (Richardson & Wallace, 2013).

Em relação à detecção de trincas para a automação do sistema, existe um método de processamento de imagem o qual é responsável pela manipulação dos dados coletados da imagem. Tal manipulação se dá através da codificação em linguagem de máquina, utilizando-se para isso um conjunto de funções matemáticas que alteram o estado dos elementos de uma matriz imagem (Jensen & Lulla, 1987). As imagens são tratadas como grandes matrizes em meio digital, alterando o estado destas matrizes é possível alterar características da imagem final.

Já existem, no campo acadêmico, alguns estudos relacionados à detecção de irregularidades em superfícies lisas. Grandes exemplos dessas aplicações são a detecção de rachaduras em paredes de concreto (Dinh, 2016) e a detecção de falhas nos asfaltos (Zou et al., 2012), onde são estudados alguns métodos possíveis para a aplicação do processamento de imagem nesses casos. O uso desses métodos no processo de detecção de trincas em superfícies metálicas se torna viável graças à similaridade desses três tipos de metodologias (Mohan & Poobal, 2018). Há inicialmente a aquisição da imagem do objeto para a detecção da irregularidade em meio digital, transformando a imagem em uma matriz de dados. O processamento se dá principalmente aplicando-se métodos de filtragem com filtros passa-baixa para a retirada de ruídos na superfície da imagem, e uso de alguns operadores de abertura e fechamento de objetos destacados para uma melhora na qualidade do resultado. A imagem final deverá ser uma imagem binarizada, com apenas pixels pretos e brancos, com um deles destacando a trinca na imagem original e o outro sendo a base para a superfície não destacada.

1.1 Objetivo e motivação

O principal objetivo de todo o projeto foi a criação de um sistema de controle para comandar todo o processo de deposição a laser. Esse controle foi feito utilizando um computador de placa única, que fará a interface do conjunto mecânico com o meio digital. Dentro desse sistema, a criação de uma programação para a detecção de trincas na peça de recuperação possibilitou alguns testes preliminares no desenvolvimento do projeto para observar a viabilidade do uso do processamento de imagem para aplicação no sistema.

A motivação do projeto está na diminuição dos gastos com a recuperação da peça que foi inutilizada com o aparecimento das irregularidades na superfície. O uso de material para a recuperação dessa superfície é muito menor se comparado com o gasto para a criação de uma nova placa. Com isso, tem-se um material reutilizável, aumentando o seu tempo de vida útil. Em relação ao processamento de imagem, os testes para a automação garante maior precisão e regularidade na deposição de material, comparando a ação lógica das máquinas com a ação imprecisa do homem, e a questão da segurança do operador da máquina, já que o laser de deposição emite raios de alta potência capazes de derreter o metal de deposição e podem causar queimaduras ao entrar em contato com a pele humana.

2 FUNDAMENTOS TEÓRICOS

Nesta seção se encontram alguns fundamentos sobre o processamento de imagem que foi essencial para a realização dos testes na detecção das trincas sobre uma imagem da peça de deposição.

O processamento de imagem define o conjunto de funções codificadas para alterar a imagem de entrada. Dependendo da aplicação, esse processo permite alterar a imagem digitalizada a fim de melhorar a sua qualidade, tanto para restaurá-la por completo quanto para destacar algumas partes importantes – por exemplo, destacar determinados tipos de células em uma imagem microscópica – (Marques Filho & Neto, 1999). Essas mudanças variam desde aumento do brilho e contraste da imagem, retirada de ruídos, detecção de formas e binarização.

Para o computador, as imagens digitalizadas nada mais são do que um conjunto de números dispostos em matrizes. Cada elemento que compõe essa matriz, define um píxel diferente na imagem original, e seu valor numérico representa a intensidade desse píxel. Uma imagem tratada com elementos inteiros de *8bits*, por exemplo, tem seus valores que variam no conjunto $[0, 255]$ – valores próximos de zero tem intensidades mais escuras até o preto, e próximos de 255 tem intensidades mais claras até o branco. Nesta matriz é possível realizar qualquer tipo de operação matemática que resultará em alterações nas disposições numéricas e, com isso, provocará alterações nas imagens. Existem portanto, algumas operações específicas que realizam mudanças oportunas para cada tipo de finalidade.

2.1 Escala de cinza

Caso a imagem digitalizada seja colorida, há a criação de mais de uma matriz imagem na programação. Geralmente, utiliza-se o sistema de RGB determinada por três matrizes separadas: *red*, *green* e *blue*; a união dessas três matrizes resultam na imagem colorida original – pode ocorrer a criação de números diferentes de matrizes dependendo do sistema de cores adotados na programação.

Trabalhar com sistemas matemáticos em conjuntos de mais de uma matriz necessita de funções mais complexas e difíceis de entendimento imediato. Portanto, em projetos onde as cores não influenciam na pesquisa, é comum transformar a imagem colorida em tons de cinza. As intensidades de cinza em cada píxel de imagem variam de acordo com o tom da cor original, num alcance de valores dependendo da classe numérica adotada (quantidade de *bits*), seja ele *uint8* $[0,255]$, *uint16* $[0, 65535]$, *int16* $[-32768, 32767]$, *single* $[0,1]$ ou *double* $[0,1]$ (Kovesi, 2000).

Para um sistema de detecção de objetos sob uma superfície lisa, essa etapa de

processamento de imagem se torna útil já que as cores do objeto não interferem em sua detecção por completo. Utilizando apenas os tons de cinza que diferem o objeto da superfície é possível destacar somente este na imagem.

2.2 Binarização

Uma imagem binarizada é um tipo especial de escala de cinza cujas intensidades estabelecem apenas os valores extremos: branco ou preto (Kovesi, 2000). O processo de binarização ocorre com a escolha de um limiar adequado – chamado de *threshold* – que fará a separação das intensidades originais de cinza: valores menores que o *threshold* se tornam pretos, enquanto maiores se tornam brancos. A escolha adequada deste limitante é importante para que apenas as partes relevantes da imagem sejam destacadas.

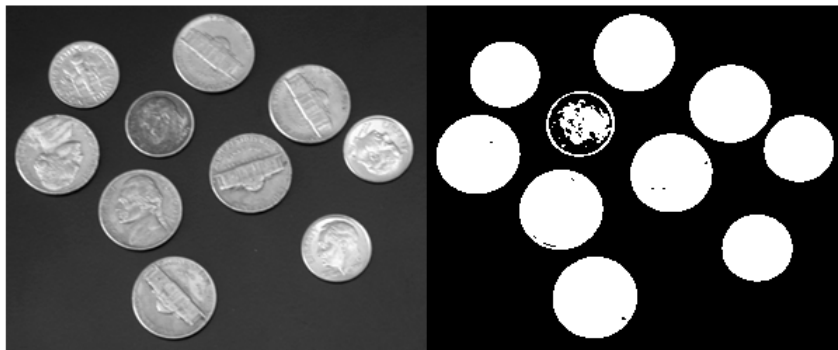


Figura 1: Binarização de uma imagem (imagem de: *Mathworks*).

Dentro dessa técnica, alguns estudos promoveram a criação de algoritmos para uma binarização com *threshold* adaptativo (Sauvola & Pietikäinen, 2000). O procedimento acontece similarmente à binarização normal, com a diferença de que mais de um limiar pode ser escolhido numa mesma imagem. A matriz de intensidades é segmentada em partes, e cada parte pode possuir um *threshold*. Isso diminui as interferências de luzes e ruídos presentes na imagem e otimiza essa separação de píxeis entre preto e branco.

Essa tarefa se mostra interessante em aplicações de detecção de objetos em imagens. Apenas as partes importantes dela são destacadas para localizar e processar imagens com maior facilidade.

2.3 Filtros

Como todo processamento da imagem ocorre em meio digital, é inevitável que apareçam ruídos indesejados ao final do processo. Os filtros aparecem como funções mais complexas com a finalidade de melhorar a qualidade da imagem final. Podem ser aplicados em qualquer etapa do processo dependendo do objetivo adotado, através de operações matriciais, transformações de Fourier e convoluções (Sonka, Hlavac & Boyle, 2014).

Filtros como o passa-baixa e o passa-alta são aplicados no começo do processo para atenuar ou ampliar determinadas frequências na imagem. Geralmente são realizadas pela convolução com matrizes especiais menores que a matriz imagem, como no exemplo da figura 2. Nela, a primeira matriz de números, ao realizar a convolução com a matriz imagem, resulta na atenuação das porções da imagem onde há pouca alteração na escala de cinza. Sua característica definida pelo maior peso no centro da matriz (dado pelo valor "8") faz com que quando esta passe por um local com uma grande variação de tom de cinza, ocorra um ganho na região; esta situação é muito aplicada para a detecção das bordas e pequenos detalhes dos objetos nas imagens. Já a segunda matriz é característica de um filtro passa-baixa. Com valores iguais em toda a matriz, ao realizar a convolução com a matriz imagem fornece uma média de valores nos locais em que o filtro passa, resultando em uma imagem mais "borrada". A convolução entre duas matrizes se dá pela passagem da matriz menor (filtro) em todos os elementos da matriz maior (imagem), realizando-se multiplicações entre os elementos de mesma linha e coluna e somando cada uma dessas multiplicações. Na matriz convoluída, a soma geralmente é colocada no local onde o centro da matriz filtro coincide com a matriz imagem.

-1	-1	-1
-1	8	-1
-1	-1	-1

1/9	1/9	1/9
1/9	1/9	1/9
1/9	1/9	1/9

Figura 2: Exemplos de filtro passa-alta e filtro passa-baixa, respectivamente.

Existe também o filtro denominado "*non-local means*" cuja funcionalidade é similar ao filtro passa-baixa, mas diferentemente dos filtros em geral, esse não se utiliza de uma matriz característica. É um tipo de filtragem mais avançado atualmente, utilizado para diminuir as perdas da qualidade da imagem, retirando alguns ruídos em sua superfície e destacando os detalhes com áreas mais homogêneas (vide figura 3 com um exemplo de aplicação do filtro).

O *non-local means* atua em todos os elementos da matriz imagem singularmente, procurando no resto da matriz por áreas similares àquela definida no local do píxel alvo. Ao definir todos os elementos com características similares, é realizado uma média entre todos os píxels, utilizando para isso uma média ponderada que define pesos maiores sobre os elementos mais semelhantes ao dado elemento alvo, e pesos menores sobre os elementos que o diferem, tanto pelo seu tom de cinza quanto pela sua vizinhança de tons ao redor (Buades & Morel, 2011).



Figura 3: Aplicação do "non-local means"(imagem de: *Mathworks*).

2.4 Erosão e dilatação

Erosão e dilatação são exemplos de transformações morfológicas que mudam forma do objeto destacado na imagem. A topologia desconhecida da imagem é alterada com o uso de elementos estruturantes conhecidos. Dependendo da forma do elemento estruturante, diferentes resultados de transformação do formato do objeto podem ser obtidos.

A erosão é aplicada por uma operação matemática entre a matriz imagem e a matriz do elemento estruturante, definida pela equação $A \ominus B = \{x \mid B_x \subseteq A\}$.

Em sua aplicação, o elemento estruturante B deve estar posicionado e centrado em um píxel x da imagem A. Quando B está totalmente contido em A, o píxel x se torna relevante e é passado para a imagem final A erodida. Caso contrário, esse píxel se torna não relevante na imagem final. Esse tipo de operação é feita principalmente para diminuir partículas e eliminar elementos menores que o elemento estruturante (Marques Filho & Neto, 1999).

No caso da dilatação, a função matemática que a define é dada pela aplicação $A \oplus B = \{x \mid B_x \cap A \neq \emptyset\}$.

Agora, o elemento estruturante B posicionado e centrado no píxel x da imagem A deve ter interseção com A para que o píxel seja relevante para a imagem final. A dilatação é realizada para aumentar partículas e objetos destacados na imagem original (Marques Filho & Neto, 1999).

A combinação dessas duas operações geram funções de "fechamento"(dilatação seguida de erosão) ou "abertura"(erosão seguida de dilatação) sobre os objetos destacados na imagem.

3 MATERIAIS E MÉTODOS

A montagem do sistema de deposição está ilustrada na figura 4, desenvolvido no Laboratório de Instrumentação Óptica para Oftalmologia e Agricultura (LIO2A), no Instituto de Física de São Carlos – Universidade de São Paulo. Tal montagem engloba um grupo de 4 pessoas envolvidas nas áreas abaixo. Este documento se enfoca nas duas últimas áreas onde o presente projeto se fez parte.

- Mecânica – desenho e usinagem das peças mecânicas da mesa de deposição, em software adequado.
- Eletrônica – Criação de circuitos e interface de todos os elementos mecânicos no microcontrolador.
- Computação – programação da lógica de funcionamento de todo o sistema, incluindo o processo de detecção das trincas na peça danificada.



Figura 4: Sistema de deposição à laser.

O diagrama de blocos simplificado (figura 5) mostra o sistema completo de funcionamento da deposição. Começando com uma peça danificada, é adquirida a imagem da peça através do uso de uma câmera conectada ao microcontrolador em I. Após essa aquisição de imagem, o processo desenrola na manipulação e processamento da imagem até a visão computacional, onde são feitas as detecções das trincas da peça, em II. Por fim, o sistema de motores e sensores realizam a deposição de material, em III, nas trincas detectadas para que a peça seja devidamente restaurada. Dentro desse sistema, a etapa destacada em II (visão computacional) é onde ocorreram os principais testes de aquisição de dados sobre a imagem para que esta seja usada no sistema automatizado de deposição.

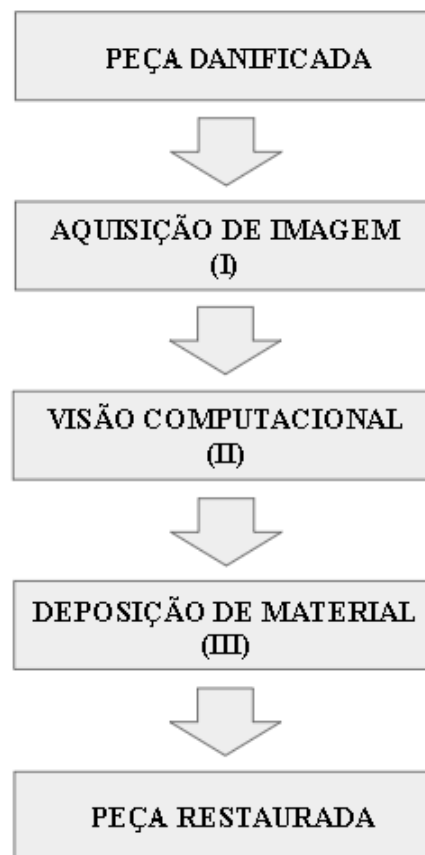


Figura 5: Diagrama de blocos simplificado do sistema.

3.1 Eletrônica

A eletrônica é definida pelo uso do computador de placa única *Raspberry Pi 3*, capaz de relacionar todos os componentes envolvidos na deposição. Essa integração dos componentes deve ser feita com a criação de circuitos elétricos específicos para cada tipo de componente envolvido, ligando todos eles nas entradas da *Raspberry Pi 3*.

Os componentes utilizados no processo de deposição envolvem:

- Laser de fibra;
- Sensor de distância;
- Motores de passo;
- Fins de curso dos motores de passo.

3.1.1 Laser de fibra

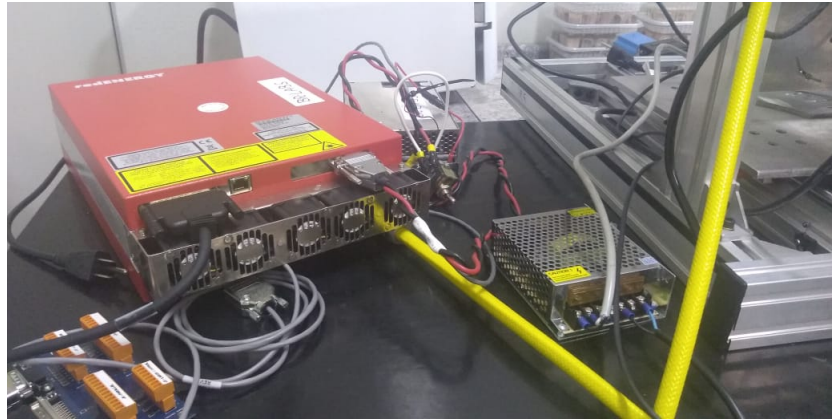


Figura 6: Laser de fibra SPI.

O laser utilizado é um laser de fibra SPI (figura 6), utilizado como emissor de laser pulsado, capaz de atingir altas potências. Esse laser pode ser acionado utilizando um comando binário (0 ou 1) para ligar e desligar o equipamento. Em relação ao comando do modo como este pode trabalhar, ele possui uma de comunicação serial Ethernet. De acordo com a sequência de bits transmitida nessa comunicação é possível alterar o modo de emissão, a sua potência e a frequência do laser emitido.

3.1.2 Sensor de distância

O sensoramento da distância de deposição, entre o laser de fibra e a base de deposição, utiliza um sensor a laser da série BGS-HL. Este sensor trabalha com valores analógicos de tensão que variam de 0 – 5V. Para a sua conexão à Raspberry Pi 3 é necessário o uso de um conversor AD (analógico para digital), para tomar as distâncias em meio digital; este conversor utiliza-se de 16bits, portanto o valor lido pelo sensor cai em uma faixa de valores no intervalo $[-32767, 32767]$.

O sensor é importante para que o laser trabalhe sempre em sua distância focal, onde a emissão de laser possui sua maior potência de dissipação.

3.1.3 Motores de passo e fins de curso

Os motores de passo são motores já instalados na mesa de deposição. Estes possuem um sistema de drivers que comandam os motores e entradas de tensão de trabalho de 12V.

Eles realizam o movimento da base nos seus eixos x, y e z. Para comandar cada motor, é necessário enviar uma sequência de pulsos para cada porta de entrada do driver. Cada pulso define a movimentação de um passo do motor.

Cada motor possui o seu fim de curso, definido por um botão. O botão do fim de curso é acionado quando o motor atinge os limites da mesa de deposição. Este não deve ultrapassar esse limite da mesa para que não haja comprometimento dos componentes mecânicos do sistema. Por programação, os fins de curso devem ser monitorados para garantir tal proteção.

3.1.4 Raspberry Pi 3

Os computadores de placa única são circuitos integrados que possuem as mesmas funcionalidades de um computador – como acessar a internet, criar arquivos e instalar softwares – com a vantagem desse possuir um tamanho muito reduzido. Também, é possível realizar o controle de robôs e máquinas industriais, dependendo da codificação realizada em sua memória e utilizando-se das portas de entrada e saída do microprocessador (Mehl, 2013).



Figura 7: Microprocessador – Raspberry Pi 3.

A Raspberry Pi é um tipo de computador de placa única muito usado em sistemas industriais. A sua facilidade de programação e uso promove a sua versatilidade em inúmeros projetos de automação. O circuito em questão possui um processador interno de $700MHz$ de $32bits$, construído sobre a arquitetura ARM II (Richardson & Wallace, 2013). Além disso, possui entradas para:

- Cartão de memória SD (*Secure Digital*);
- Portas USB (*Universal Serial Bus*);
- Porta ethernet;
- Conector HDMI (*High-Definition Multimedia Interface*);

- Fonte de tensão;
- *Flat cable* para câmera.

Ademais, encontram-se em sua arquitetura 40 pinos de ligação que variam entre GPIO's (*General Purpose Input/Output*), saídas de tensões de 5V e 3,3V, aterramento (GND) e algumas funções específicas a determinados componentes que ali são conectados. Esse conjunto de pinos é responsável por realizar as ligações de outros componentes, como sensores de temperatura e pressão, leitores de radiofrequência, células de carga, motores, etc. Através de uma programação adequada, muitos destes componentes são controlados de acordo com a sua necessidade.

Para o projeto, foram reservados alguns GPIO's para o acionamento do laser, controle do sensor de distância, movimentação dos motores de passo e monitoramento dos fins de curso de cada motor. Além disso, a porta ethernet para enviar os comandos para o laser, as portas USB para conectar o mouse e teclado, bem como o conector HDMI para um monitor e a entrada de *flat cable* para a câmera.

3.2 Computação

Com o estabelecimento da mesa de deposição, o próximo passo foi realizar os testes de programas computacionais para controle dos dados adquiridos com os componentes utilizados, e processamento de imagem para a detecção das trincas. Esses programas foram criados utilizando a linguagem de programação de alto nível: Python.

3.2.1 Python

O python é uma linguagem de script de alto nível que possui um sistema de OpenCV (*Open Source Computer Vision Library*). A característica de *script* faz com que a execução de tarefas seja automatizada e realizada passo-a-passo, sem a necessidade de uma pré-compilação de todas as tarefas. Isso faz com que o código se torne mais intuitivo e facilita tanto a lógica de escrita quando a detecção de erros no sistema codificado, além de ter um tempo de execução relativamente rápido dependendo da aplicação dessa linguagem (Van Rossum, 2007).

O acesso a uma gama variedade de informação da linguagem permite criar códigos de programação com facilidade utilizando poucas linhas de programação. A biblioteca multiplataforma de OpenCV é um sistema de compartilhamento de informações de livre uso acadêmico e comercial, que possui funções e tutoriais referentes à linguagem de programação.

A codificação em python realiza toda a interface de comandos para os componentes da deposição (laser, sensor de distância, motores e câmera), bem como a lógica de

programação para o funcionamento do sistema implantado, incluindo o processamento de imagem na detecção das trincas na imagem da placa de testes.

3.2.2 Placa de testes

Como o desenvolvimento da mesa de deposição ainda não foi totalmente concluído no laboratório (LIO2A), os testes preliminares dos programas foram feitos principalmente para definir a viabilidade da aplicação de um processamento de imagem na detecção de trincas a fim de automatizar o sistema por completo. No processamento de imagem, foram utilizadas as funções de manipulação de imagem destacadas no capítulo 2 de fundamentos teóricos. A sequência de funções para o processamento foi a transformação em escalas de cinza, seguido da aplicação do filtro "non-local means"(NLM), binarização adaptativa e finalizando com a função de abertura (erosão + dilatação).

Para a aquisição da imagem da peça a ser restaurada, foi utilizada a câmera conectada à Raspberry Pi 3 com o *flat cable*, com a criação de um código de programação para controle da câmera. A imagem tomada da peça de testes é a figura 8. Como se pode observar, é uma chapa de metal gasta, com 3 grandes trincas visíveis e algumas mínimas irregularidades em sua superfície.



Figura 8: Peça de metal danificada utilizada para os testes da visão computacional.

4 RESULTADOS DO TRABALHO

O esquemático da figura 9 mostra o funcionamento geral do sistema de deposição. Nele é possível observar a disposição das partes envolvidas no processo em bancada desenvolvida, ilustrada pela figura 4, para um maior entendimento do fluxo de dados do sistema.

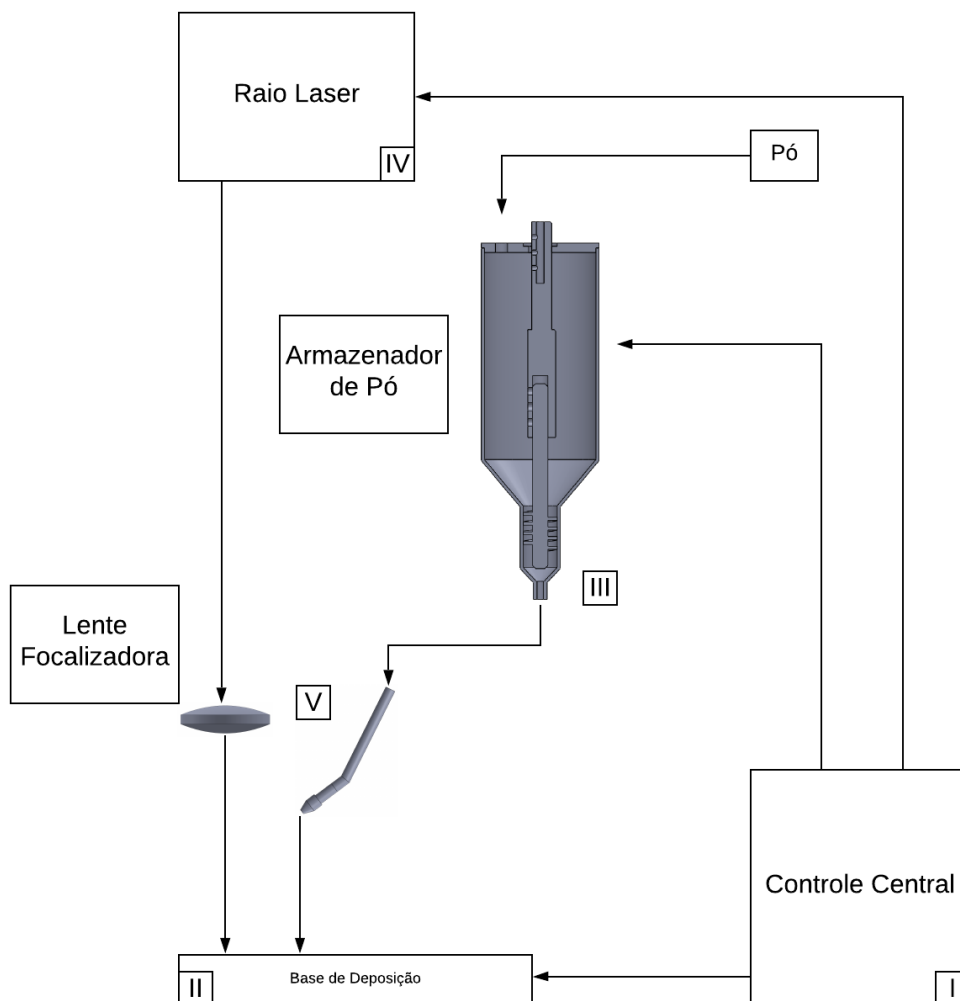


Figura 9: Esquemático de funcionamento da deposição a laser.

4.1 Circuito de integração

O circuito de integração do sistema para a Raspberry Pi 3 foi desenhado no *software Eagle*, com o mapa de ligações do esquemático e o plano de criação da PCB (*Printed Circuit Board*). O circuito é mostrado na figura 10, possuindo as ligações dos motores, sensor de distância, válvula e laser e fins de curso nos pinos de GPIO da Raspberry Pi

3. O circuito se encontra na caixa de comandos determinada por I (controle central) na figura 9.

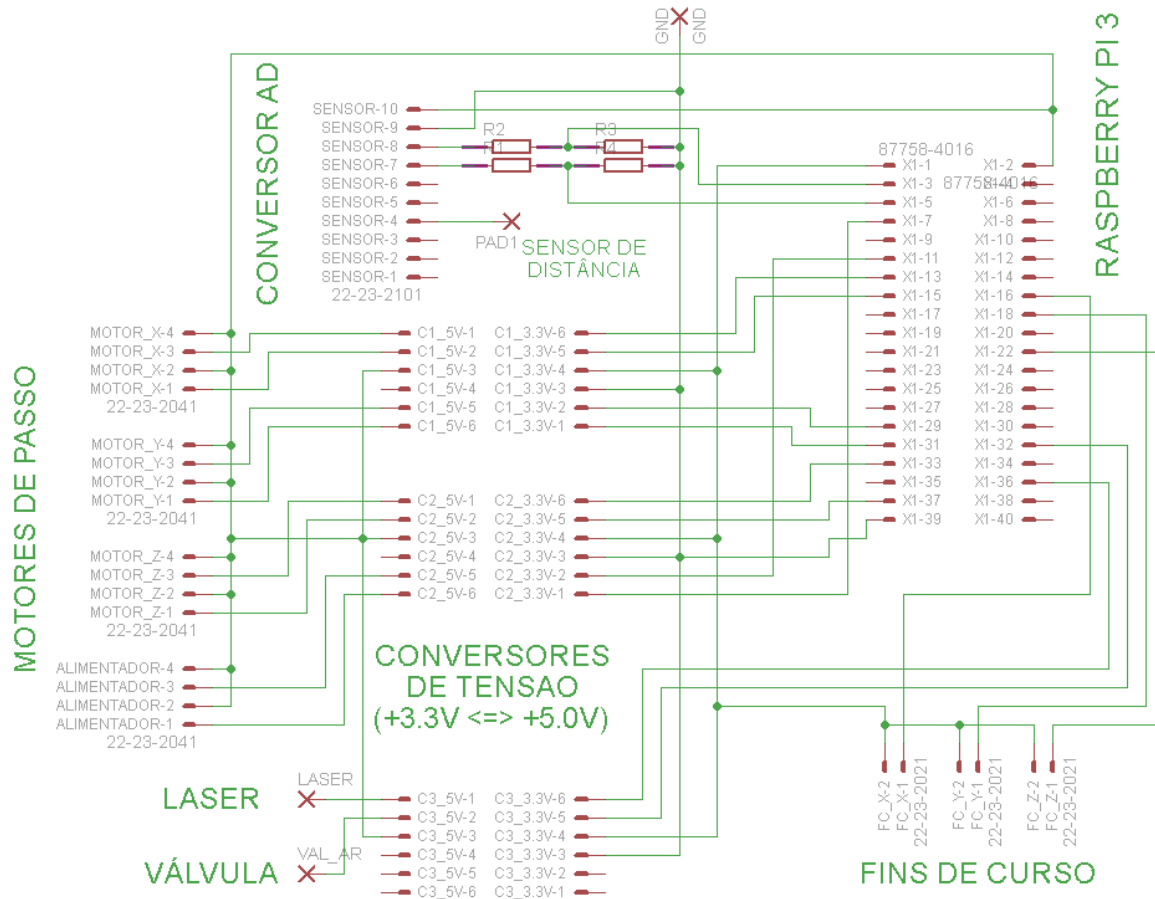


Figura 10: Circuito de controle para a ligação na Raspberry Pi 3.

Existem três motores de passo no circuito que realizam a movimentação da base de deposição (em II da figura 9) nos eixos x, y e z. A movimentação é dada pelo controle de 2 GPIO's em cada motor, um determinando o sentido de rotação e o outro a sequência de pulsos (quantidade de passos).

Cada um dos três motores possui um sistema de segurança com fins de curso para que o motor não ultrapasse os limites da mesa. Os fins de curso funcionam como chaves que, quando acionadas, interferem em uma das entradas de GPIO para barrar a movimentação dos seus respectivos motores de acordo com a programação. Um quarto motor foi previsto no circuito e usado para realizar a alimentação do pó de material depositado, entre o armazenador de pó e o dispositivo de deposição (entre III e V da figura 9). À medida que o motor gira, mais pó é transportado para o sistema.

O laser e a válvula do circuito são entradas que ligam a GPIO's para o controle

de habilitação tanto do laser quanto da válvula de pressão. De acordo com a tensão de saída da GPIO da Raspberry, o laser é ligado e desligado para fundir o pó metálico, e a válvula é acionada ou não para injetar o pó. É possível observar o posicionamento de tais componentes destacados na figura 9 em IV e V. Tanto estes dois pinos de acionamento do laser e válvula quanto cada uma das entradas dos quatro motores de passo possuem uma tensão de trabalho de $+5,0V$; e a tensão de alimentação das portas da Raspberry injetam apenas $+3,3V$ ao circuito. Portanto, se fez necessário o uso de 3 conversores de tensão $+3,3V$ para $+5,0V$ entre as GPIO's e a entrada desses componentes.

No topo do circuito, o conversor AD faz a interface entre o sensor de distância e a central de comando. Os dados analógicos advindos do sensor passam pelo conversor e a sua saída, agora digital, é fornecida para os GPIO's da Raspberry realizarem a interpretação. Nesse caso há também o problema da tensão de trabalho fornecida pelo sensor ser de $+5,0V$. As quatro resistências fazem a divisão de tensão para $+3,3V$. O sensor é acoplado juntamente ao laser em IV (figura 9).

Com o mapeamento do circuito, o Eagle proporciona a criação do desenho da PCB do circuito. A figura 11 mostra a placa impressa nesse mesmo laboratório.

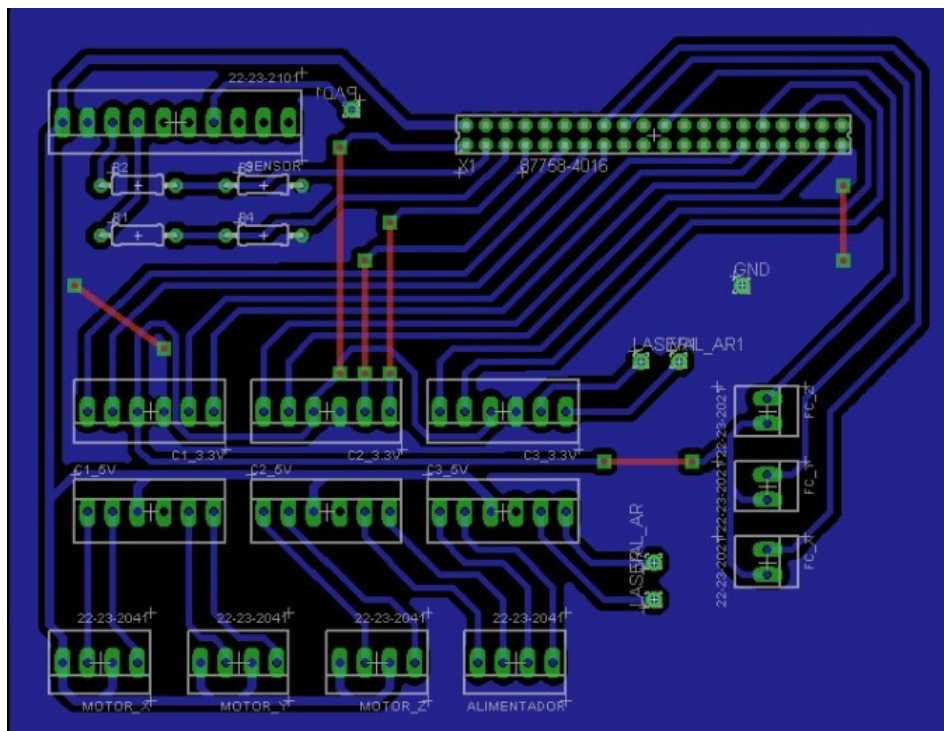


Figura 11: *Printed Circuit Board* do circuito de integração do sistema à Raspberry Pi 3.

4.2 Programação

A programação para controle do laser e da válvula utilizam portas de GPIO configuradas como *output*. O estado *HIGH* define acionamento e o *LOW*, o desativamento de cada componente.

A programação dos motores se dá pelo uso de duas funções: "chaveamento"(figura 12) e "driver"(figura 13). O primeiro cria os pulsos a serem enviados para uma das portas do motor, enquanto o segundo realiza o envio dos pulsos com a sua quantidade juntamente com a escolha do sentido de rotação do motor.

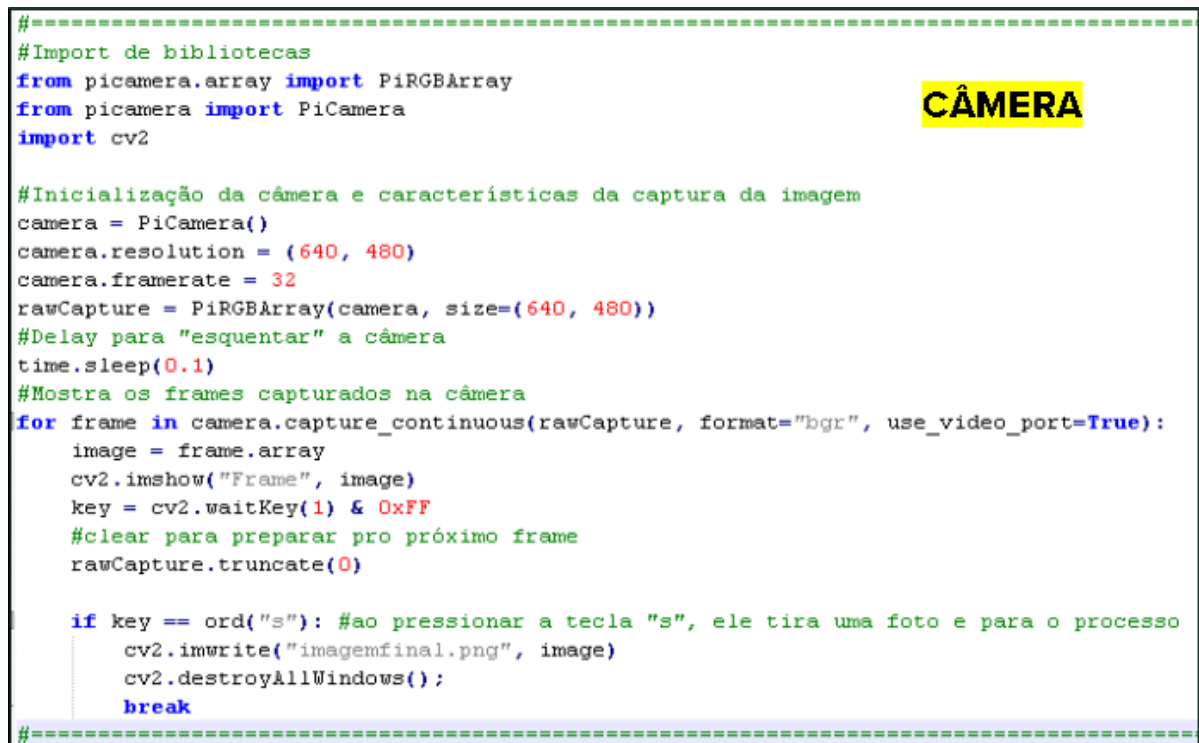
```
def chaveamento(passos, pino):
    pulso = 0.0001 #define a largura do pulso.
    step_cont = 0
    x_pin = 1 #começa em alto.
    for step_cont in range (passos):
        if (x_pin == 1):
            GPIO.output(pino,True)
            x_pin = x_pin-1
            time.sleep(pulso)
        else:
            GPIO.output(pino,False)
            time.sleep(pulso)
            x_pin = x_pin+1
    GPIO.output(pino,False)
#-----
```

Figura 12: Programação em python para o chaveamento do motor.

```
def driver(passos, eixo, dir):
    #determina os eixos x,y e z e direções de e
    if (eixo == 1):
        pino = 13;
        if (dir == 1):
            GPIO.output(15,True);
        elif (dir == -1):
            GPIO.output(15,False);
    elif (eixo == 2):
        pino = 29;
        if (dir == 1):
            GPIO.output(31,True);
        elif (dir == -1):
            GPIO.output(31,False);
    elif (eixo == 3):
        pino = 33;
        if (dir == 1):
            GPIO.output(37,True);
        elif (dir == -1):
            GPIO.output(37,False);
    #função do chaveamento de passos
    chaveamento(passos, pino)
#GPIO.cleanup() #limpa todos os canais de GPIO
```

Figura 13: Programação em python para o driver do motor.

Para a câmera, a programação foi feita para que esta funcione inicialmente como filmadora, com uma resolução de 640x480 e *frame rate* igual a 32. Quando um botão do teclado é acionado – tecla "s" para a programação da figura 14 – a câmera tira uma foto e a salva na memória da Raspberry para uso posterior no processamento de imagem.



```

#-----
#Import de bibliotecas
from picamera.array import PiRGBArray
from picamera import PiCamera
import cv2

#Inicialização da câmera e características da captura da imagem
camera = PiCamera()
camera.resolution = (640, 480)
camera.framerate = 32
rawCapture = PiRGBArray(camera, size=(640, 480))
#Delay para "esquentar" a câmera
time.sleep(0.1)
#Mostra os frames capturados na câmera
for frame in camera.capture_continuous(rawCapture, format="bgr", use_video_port=True):
    image = frame.array
    cv2.imshow("Frame", image)
    key = cv2.waitKey(1) & 0xFF
    #clear para preparar pro próximo frame
    rawCapture.truncate(0)

    if key == ord("s"): #ao pressionar a tecla "s", ele tira uma foto e para o processo
        cv2.imwrite("imagemfinal.png", image)
        cv2.destroyAllWindows()
        break
#-----

```

CÂMERA

Figura 14: Programação em python para a inicialização da câmera.

4.3 Detecção de trincas com processamento de imagem

O resultado da detecção de trincas na imagem tomada da placa de testes utilizada pode ser vista na figura 17.

No processamento de imagem, fez-se uso de funções encontradas no OpenCV encontrados para a linguagem de programação utilizada. A transformação para tons de cinza foi realizada pela função "cv2.cvtColor()". Para a filtragem com o NLM, existe a função "cv2.fastNlMeansDenoisingColored()". Seguida da binarização adaptativa utilizando a função "cv2.adaptiveThreshold()" e a aplicação da função de abertura com "cv2.erode()" e "cv2.dilate". Nestes últimos, utilizou-se matrizes de elemento estruturante básicos, com todos os elementos tendo valor unitário e com dimensões 3x3.

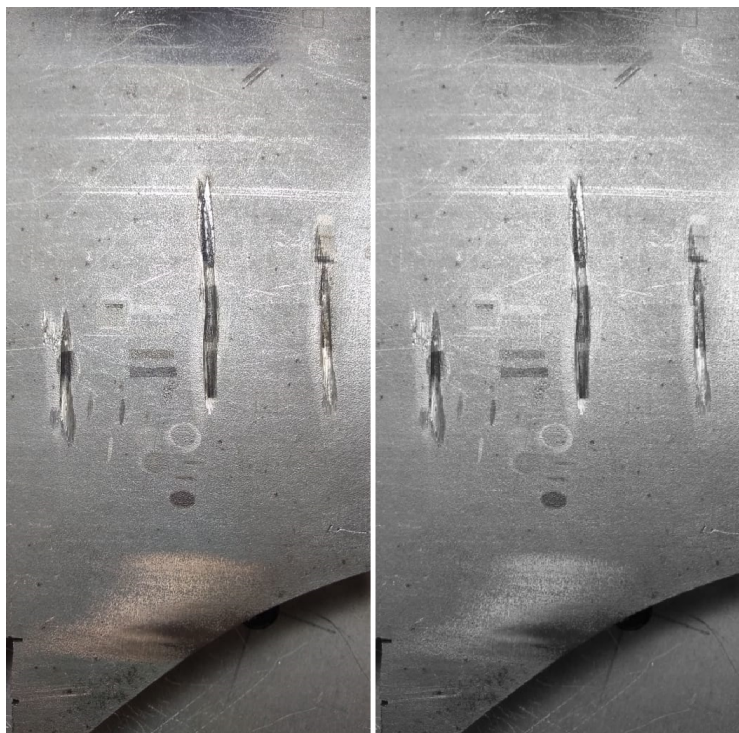


Figura 15: Imagem processada original e após a transformação em tons de cinza, respectivamente.

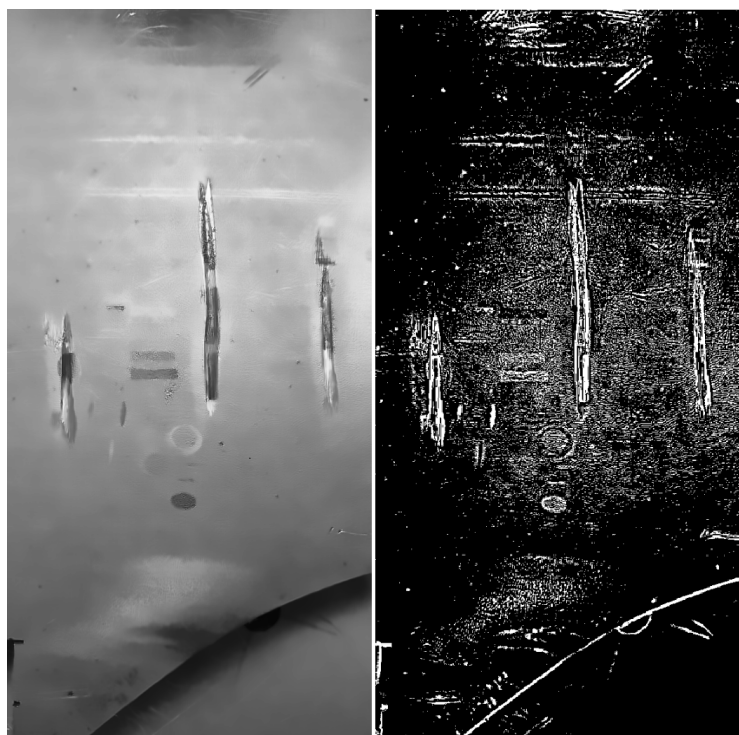


Figura 16: Imagem processada após a aplicação do NLM e após a binarização, respectivamente.



Figura 17: Imagem final após o processamento para detecção de trincas.

Observa-se pelo resultado obtido, que o método aplicado para os testes preliminares para a aplicação do processamento de imagem no sistema de deposição teve algumas falhas, com o aparecimento de vários ruídos na superfície da imagem. Os falsos positivos em pixels brancos onde não deveria haver a detecção da trinca, e os falsos negativos em pixels pretos onde deveria haver a detecção da trinca. Essas falhas se devem principalmente à não homogeneidade da peça, já que esta possui algumas irregularidades em sua superfície, tanto em profundidade quanto em questão de cores. O problema se eleva com a presença de um jogo de luz e sombra na imagem no momento em que a foto foi tirada pela câmera.

Para amenizar as influências dos ruídos na placa seriam necessários estudos mais aprofundados sobre as funções que regem o processamento da imagem para detectar com mais clareza apenas as trincas das placas de deposição.

Entretanto, em questões de aplicação para o projeto de deposição, foi possível notar que é possível aplicar o processamento de imagem para automatizar o sistema por completo. O local das três grandes trincas da peça de teste foram detectadas, mesmo que com o aparecimento dos ruídos citados. É possível contornar estes problemas através do uso de uma programação adequada, com a ajuda do sensor de distância.

Ao percorrer os locais detectados pelo processamento, quando há variação no valor lido pelo sensor, implica na presença real de trinca no local, devendo-se portanto realizar a deposição de material. Similarmente, quando não há variação na distância lida pelo sensor no local detectado, não há presença de trincas. Além disso, o sistema de deposição se utiliza

de material fundido. Este material, no momento da deposição, preenche os locais mais profundos da trinca antes de chegar na superfície da peça. Isso garante que a deposição de material seja feita em toda a superfície da trinca, mesmo que a sua área não seja detectada por completo.

5 CONCLUSÕES

Em questões de pesquisa, a detecção das trincas ocorreu de forma pouco satisfatória. Não se verificaram resultados que mostrem exclusivamente as trincas detectadas, nem resultados com toda a extensão da trinca. Entretanto pôde-se concluir que a aplicação do processamento de imagem é uma realidade para o sistema de deposição a laser. É possível utilizar essas imagens, mesmo que com alguns ruídos na superfície, para automatizar o sistema em questão por completo com uma programação adequada.

Como o sistema ainda está sendo desenvolvido no LIO2A e ainda não fora concluído no presente momento da escrita deste documento, alguns passos posteriores envolvem a implementação dessa programação adequada para o controle da deposição de material. Atualmente existem apenas algumas programações que controlam partes do sistema individualmente. É preciso realizar a integração desses códigos, juntando em uma única lógica de programação, levando em consideração todos os componentes do projeto de deposição. Após essa integração, será possível realizar os testes definitivos para a recuperação da superfície metálica danificada.

Ademais, para melhorar a qualidade da detecção com o processamento de imagem, existem outros parâmetros que podem ser adicionados ou alterados no programa. Existem métodos para correção de iluminação não uniforme em uma imagem, outros tipos de funções que auxiliam na detecção (como os filtros de detecção de borda e funções de crescimento de região), uso de outras formas para os elementos estruturantes na erosão e dilatação, além dos estudos de *machine learning* para a detecção de trincas, aplicando a área de inteligência artificial. Fora as mudanças em software, pode-se também melhorar a qualidade da iluminação no local onde a imagem da peça é tomada para amenizar os efeitos de sombreamento, e realizar uma pré preparação da superfície da peça (com um polimento) para tornar a superfície mais homogênea.

A implementação da linguagem de máquina para trabalhos a princípio de humanos, se mostrou um procedimento complexo. Emular nossa visão e transferir a lógica de pensamento do nosso cérebro ao meio digital depende de muitas variáveis e métodos de transição. Muitas vezes, essa transição não é direta e deve-se introduzir outros parâmetros de interpretação e manipulação de dados (como o uso dos filtros e sensores, nesse caso). Entretanto, a automação dos processos com o uso de máquinas é uma realidade para a nossa sociedade e deve ser tomada na prática para incrementar o ambiente à nossa volta.

REFERÊNCIAS

- BAGNATO, V. S. **Laser e suas aplicações em ciência e tecnologia**. 1. ed. [S.l.]: E. Livraria da Física, 2008.
- BUADES, A.; COLL, B.; MOREL, J.-M. Non-local means denoising. **Image Processing On Line**, v. 1, p. 208–212, 2011.
- DINH, T. H.; HA, Q.; LA, H. M. Computer vision-based method for concrete crack detection. In: IEEE. **2016 14th International Conference on Control, Automation, Robotics and Vision (ICARCV)**. [S.l.], 2016. p. 1–6.
- FILHO, O. M.; NETO, H. V. **Processamento digital de imagens**. [S.l.]: Brasport, 1999.
- JENSEN, J. R.; LULLA, K. **Introductory digital image processing: a remote sensing perspective**. Taylor & Francis, 1987.
- KOVESI, P. D. Matlab and octave functions for computer vision and image processing. **Centre for Exploration Targeting, School of Earth and Environment, The University of Western Australia, available from: <http://www.csse.uwa.edu.au/pk/research/matlabfns>**, v. 147, p. 230, 2000.
- MARENGONI, M.; STRINGHINI, S. Tutorial: Introdução à visão computacional usando opencv. **Revista de Informática Teórica e Aplicada**, v. 16, n. 1, p. 125–160, 2009.
- MEHL, E. L. Do transistor ao microprocessador. **S/d. Coletânea online de artigos acadêmicos, prof. Mehl.**, 2013.
- MOHAN, A.; POOBAL, S. Crack detection using image processing: A critical review and analysis. **Alexandria Engineering Journal**, Elsevier, v. 57, n. 2, p. 787–798, 2018.
- RICHARDSON, M.; WALLACE, S. Primeiros passos com o raspberry pi. **Primeira Edição. Novatec Editora Ltda**, p. 20, 2013.
- ROSSUM, G. V. et al. Python programming language. In: **USENIX annual technical conference**. [S.l.: s.n.], 2007. v. 41, p. 36.
- SAUVOLA, J.; PIETIKÄINEN, M. Adaptive document image binarization. **Pattern recognition**, Elsevier, v. 33, n. 2, p. 225–236, 2000.
- SCHMIDT, M. et al. Laser based additive manufacturing in industry and academia. **CIRP Annals**, Elsevier, v. 66, n. 2, p. 561–583, 2017.
- SONKA, M.; HLAVAC, V.; BOYLE, R. **Image processing, analysis, and machine vision**. [S.l.]: Cengage Learning, 2014.
- TOYSERKANI, E.; KHAJEPOUR, A.; CORBIN, S. F. **Laser cladding**. [S.l.]: CRC press, 2004.
- ZOU, Q. et al. Cracktree: Automatic crack detection from pavement images. **Pattern Recognition Letters**, Elsevier, v. 33, n. 3, p. 227–238, 2012.