



DISSERTAÇÃO

Mestrado em Engenharia Eletrotécnica - Eletrónica e Telecomunicações

Sistema de Inspeção Visual de Pasta de Solda

GUILHERME BOITA FERREIRA PAULO

Leiria, Setembro de 2021



MASTER DISSERTATION

Electrical Engineering - Electronics and Telecommunications

Sistema de Inspeção Visual de Pasta de Solda

GUILHERME BOITA FERREIRA PAULO

Dissertação orientada pelos Professores Sérgio Manuel Maciel de Faria e Luís Miguel Ramos Perdigoto, da Escola Superior de Tecnologia e Gestão do Instituto Politécnico de Leiria.

Leiria, September 2021

*All things are difficult before they
are easy.*

(Dr. Thomas Fuller)

Agradecimentos

Esta dissertação simboliza o fim de um percurso de três anos que só foi possível graças ao contributo de várias pessoas e instituições, a quem gostaria de agradecer:

aos meus orientadores, Professor Sérgio Faria e Professor Luís Perdigoto, pela disponibilidade para esclarecer dúvidas, pelos seus conselhos, pelo seu empenho e por todas as correções durante a escrita da dissertação e do artigo de investigação;

ao Instituto Politécnico de Leiria e Digiwest pela oportunidade de poder trabalhar neste projeto;

aos Engenheiros Filipe Perdigoto, António Pires e todos os outros trabalhadores da Digiwest que me auxiliaram e esclareceram dúvidas sempre que foi necessário;

à minha namorada, família e amigos por todo o seu apoio e constante encorajamento ao longo destes três anos;

aos meus colegas de Mestrado em Engenharia Electrotécnica que tanto me ajudaram nas várias etapas deste projeto.

Abstract

The growth of the electronics industry led to a need for efficient methods of testing and validation of printed circuit boards (PCB). It is necessary to identify defects that might appear in a component in early production stages. This task may be performed by automatic inspection systems, showing advantages in speed, accuracy and repeatability, over human inspection.

This thesis describes the working prototype of a visual inspection system that is able to detect solder paste deposition defects on a PCB. The PCB image is analysed to segment the areas with solder paste and those areas are compared to a reference image, obtained through the Gerber file, one of the PCB design files. The detected defects are identified and presented to the user, either due to lack or excess of solder paste. This system is based on low cost components, namely a Raspberry Pi Compute Module, two Raspberry Pi v2 cameras and a structure that holds all the necessary components in order to do a solder paste inspection.

The laboratory and industrial tests performed with the prototype, regarding the PCB defect detection and execution time, allowed to conclude that the system can aid human visual inspection in a production line.

Keywords: Visual Inspection, Solder Paste, Solder Paste Stencil Printing, Solder Deposition, PCB Industry

Resumo

O crescimento da indústria da electrónica levou a um aumento da necessidade de métodos de validação e teste em placas de circuito impresso (PCB) mais eficientes. É necessário identificar defeitos que possam aparecer num componente numa fase inicial de produção. Esta tarefa pode ser executada por um sistema automático de inspeção, que apresenta vantagens na velocidade, exatidão e repetibilidade em relação a uma inspeção manual efetuada por um colaborador.

Esta dissertação descreve um protótipo de um sistema de inspeção visual que torna possível detetar erros que podem ocorrer durante o processo de deposição de pasta de solda. A imagem da PCB é analisada de modo a segmentar as zonas que contêm pasta de solda e essas zonas são comparadas com a imagem de referência, obtida através do ficheiro Gerber, um dos ficheiros de projeto da PCB. Os defeitos detetados são identificados e apresentados ao utilizador, quer tenham origem no excesso ou na ausência de solda. Este sistema foi elaborado com materiais e componentes de baixo custo, nomeadamente um Raspberry Pi Compute Module, duas câmaras Raspberry Pi v2 e toda a estrutura de suporte para facilitar a tarefa de inspeção.

Foram realizados vários testes laboratoriais e em ambiente industrial com o protótipo onde se pode concluir que o sistema permite agilizar a etapa de inspeção em linha de produção.

Palavras chave: Inspeção Visual, Pasta de Solda, Deposição de Pasta de Solda com Stencil, Deposição de Solda, Indústria das PCB

Conteúdo

Agradecimentos	iii
Abstract	v
Conteúdo	x
Lista de Figuras	xi
Lista de Tabelas	xv
1 Introdução	1
1.1 Contexto e Motivação	2
1.2 Objectivos	2
1.3 Estrutura da Dissertação	3
2 Conceitos e Revisão Bibliográfica	5
2.1 Deposição de Solda	5
2.2 Métodos de inspeção de solda em PCB	8
2.3 Aquisição de Imagem	12
2.4 Processamento de Imagem	16
3 Sistema de Inspeção	27
3.1 Descrição do Sistema	27
3.2 Aplicação com o algoritmo de Inspeção	32
3.2.1 Calibração	32
3.2.2 Inspeção	39
3.2.3 <i>Graphical User Interface</i> (GUI)	43
4 Resultados	47

4.1	Tempos de execução	47
4.2	Diferença nos detetores de <i>features</i>	48
4.3	Deteção de falta de solda e solda em excesso	50
4.4	Inspeção de Painel de PCB	52
5	Conclusões e Trabalho Futuro	55
	Bibliografia	57
A	Artigos Publicados	63
B	Manual de utilizador da interface gráfica	65

Lista de Figuras

2.1	Stencil metálico [8].	6
2.2	Defeitos de deposição de solda[10]: (a) bom; (b) excesso; (c) insuficiente; (d) deslocação horizontal; (e) deslocação vertical.	8
2.3	Exemplo de um sistema de inspeção com o método de triangulação laser [11].	9
2.4	Exemplo de uma inspeção a um componente <i>Ball Grid Array</i> (BGA) [19]. .	11
2.5	Tipos de iluminação [21]. As áreas vermelhas indicam as fontes de luz. (a) iluminação direta; (b) iluminação de fundo; (c) iluminação <i>dark field</i> ; (d) iluminação difusa; (e) iluminação direcional.	14
2.6	Equação de transformação do espaço de cores RGB para YCbCr [25]. Y', C'_B e C'_R simbolizam as componentes do espaço YCbCr. R', G' e B' simbolizam as componentes do espaço de cores RGB.	18
2.7	Distorção radial: Grelha da esquerda sem distorção, grelha do meio com distorção de barril, grelha da direita com distorção de almofada [27].	19
2.8	Visualização do método de Otsu[32]. O valor escolhido é o ponto onde existe maior variância entre as duas classes, identificadas a cinzento e vermelho no gráfico.	20
2.9	Visualização do algoritmo k-means em funcionamento [33]. O centro de cada centroide é assinalado com um X e os vários pontos são agrupados em relação à distância para o centroide.	22
2.10	Visualização do algoritmo k-means numa imagem [34]. O número de centroides é definido por K, quanto maior o número de centroides mais cores será possível obter.	23
2.11	Exemplo da aplicação do detetor de arestas de Canny [37].	23
2.12	Exemplo da aplicação do detetor de cantos de Harris [40]. Todos os cantos detetados com a configuração definida são pintados a vermelho.	24
2.13	Exemplo de correspondência de pontos de interesse [42].	25
3.1	Raspberry Pi Compute Module 3+ (a) [44] e placa de IO (b) [45].	27

3.2	Raspberry Pi Camera V2 [46].	28
3.3	Área de trabalho e orientação das câmaras.	29
3.4	Estrutura construída com perfil em alumínio.	29
3.5	Suportes para as câmaras.	30
3.6	Protótipo do sistema de inspeção.	31
3.7	Versão final do protótipo do sistema de inspeção.	32
3.8	Diagrama de blocos da etapa de calibração e inspeção.	34
3.9	Exemplo das posições do xadrez de referência para a calibração. É necessário capturar várias imagens deste padrão em várias posições diferentes.	35
3.10	Relação entre o xadrez de referência e o xadrez no plano da PCB. É calculada a matriz de transformação entre os pontos da imagem da direita e a imagem da esquerda, que será o destino.	36
3.11	Imagem adquirida (esquerda) e imagem transformada (direita), pronta para ser processada, depois da transformação homográfica.	37
3.12	Identificação da região de interesse no espaço de cores HSV (esquerda) e componente S (direita). O contorno detetado é apresentado na imagem da direita a vermelho.	37
3.13	Correspondência de pontos chave com várias correspondências incorretas. É necessário filtrar as correspondências para garantir um melhor alinhamento.	39
3.14	Deteção de excesso (azul) e falta (vermelho) de solda. À esquerda a imagem Gerber; Ao centro a imagem dos <i>pads</i> a analisar; e à direita a deteção de pasta de solda.	40
3.15	Diagrama de blocos do algoritmo de deteção <i>pad</i> a <i>pad</i>	40
3.16	Exemplo do algoritmo de deteção <i>pad</i> a <i>pad</i> . Gerber (em cima) e imagem da PCB a analisar (em baixo). Neste caso, os <i>pads</i> que contêm erro são assinalados a vermelho.	41
3.17	Detalhe da inspeção <i>pad</i> a <i>pad</i> , resultado da inspeção à esquerda e imagem capturada à direita.	42
3.18	Deteção <i>pad</i> a <i>pad</i> . Imagem da PCB a analisar (à esquerda) e Gerber com <i>pads</i> que contêm erro a vermelho (à direita).	42
3.19	Interface gráfica da janela de calibração.	43
3.20	Interface Gráfica do sistema de inspeção.	44
4.1	Combinações de <i>features</i> com ORB (esquerda) e BRISK (direita). Teste ao tempo de execução da correspondência de <i>features</i> detectadas em imagens idênticas, com eliminação de falsas correspondências.	50

4.2	Erros detetados na inspeção: Excesso de solda (azul) e defice de solda (vermelho).	51
4.3	Erro detetado devido à diferença entre Gerber e <i>stencil</i> . Gerber de referência à esquerda, imagem do <i>pad</i> ao centro e erro detetado no <i>pad</i> à direita.	51
4.4	Erro detetado devido à falta de solda. É assinalado a vermelho a falta de solda em relação ao ficheiro Gerber.	52
4.5	Pasta de solda espalhada assinalada a azul quando é fora do <i>pad</i> e a vermelho quando existe falta de solda. Imagem do Gerber à esquerda, imagem do <i>pad</i> ao centro e deteção da falta e excesso de solda à direita.	52
4.6	Gerber disponível (esquerda) e Gerber composto para painel (direita).	53
4.7	GUI de montagem do Gerber final.	54

Lista de Tabelas

2.1	Tipo e tamanho de partículas de solda em μm [9].	7
4.1	Tempos de Execução médios em segundos (em 5 inspeções).	48
4.2	Comparação dos tempos de execução entre os métodos ORB e BRISK. . .	49

Capítulo 1

Introdução

A massificação de sistemas electrónicos, utilizados em cada vez mais objetos no dia a dia, incentivou o desenvolvimento de métodos eficientes de teste e validação do processo de montagem de circuitos electrónicos em placa de circuito impresso, também designada por PCB (*Printed Circuit Board*). É extremamente importante detetar e prevenir problemas que possam existir num circuito numa fase inicial de produção, de modo a que os erros não sejam propagados para uma fase final deste processo.

Em [1] é apresentada uma estimativa do custo originado por um defeito causado pela má deposição de pasta de solda numa PCB. Ao detetar um defeito numa fase inicial de produção, quer seja um defeito na deposição de solda, quer seja no fabrico de uma PCB, o custo para resolver esse problema é cerca de 100 vezes menor que numa fase mais final da sua produção.

Em particular, o processo de deposição de pasta de solda contribui entre 50 a 70% dos erros que podem resultar em todo o processo da colocação de um componente SMD [2], pelo que a inspeção nesta etapa de produção é bastante relevante, não só a nível financeiro como mais eficaz.

Em relação à inspeção realizada por humanos, a inspeção automática apresenta várias vantagens, tais como a velocidade, a precisão e a repetibilidade. A fadiga que esta tarefa causa em humanos leva-os a cometer erros de análise.

Relativamente à inspeção automática (ou semi-automática), existem vários métodos de deteção de erros que têm sido utilizados no processo de deposição de solda. Alguns baseiam-se na análise de informação 3D da placa, utilizando técnicas como a *Fourier Transform Profilometry* [3] ou a triangulação de laser para a reconstrução 3D do modelo de pasta de solda [4, 5]. Outros utilizam equipamentos de raio-X, placas de teste com contacto físicos sobre os componentes, ou apenas imagens 2D da placa de circuito impresso com a solda.

Independentemente do método de aquisição ou do tipo de informação gerada, os sistemas de análise com base em imagens são os mais comuns para inspeção de placas de circuito impresso, com vista a detetar possíveis anomalias na deposição da solda.

Com o avanço da tecnologia, o aumento da capacidade dos processadores para processar imagens e reconhecer padrões os equipamentos computacionais tornaram-se mais acessíveis e com melhores resultados para análise de imagem, permitindo a sua utilização mais generalizada.

1.1 Contexto e Motivação

O processo de inspeção, como já referido anteriormente, é um processo repetitivo e cansativo, uma vez que a mesma tarefa é repetida durante um longo período de tempo. A automatização de processos como este permite um maior controlo no processo de inspeção, sendo mais preciso, rápido e menos propenso a erros.

Em parceria com a Digiwest - Wireless and Embedded Solutions Lda, que pretendia automatizar o processo de inspeção da deposição de pasta de solda em PCB, foi proposto desenvolver um sistema para automatizar o processo de inspeção, tornando-o mais rápido e mais eficiente. O sistema deve receber o circuito impresso com a solda, capturar as imagens, analisar na imagem da placa a posição da solda deposta e, de acordo com os parâmetros da mesma, mostrar ao operador se existem defeitos (falta de solda ou solda fora do local definido).

Devido à elevada flexibilidade e ao custo associado, pretende-se utilizar um *Raspberry Pi Compute Module*, que permite utilizar duas câmaras em simultâneo. Ao utilizar um sistema deste tipo, é possível analisar com maior detalhe (resolução espacial) todos os pontos da PCB sem que seja necessário movimentar a PCB ou qualquer outro componente do sistema, ao contrário de outros sistemas no mercado [6].

1.2 Objectivos

O trabalho desenvolvido no âmbito desta dissertação teve como foco os seguintes objectivos:

- Desenvolvimento de um algoritmo de deteção de pasta de solda na placa de circuito impresso (PCB);
 - Captar as imagens do xadrez de referência com duas câmaras;
 - Calibrar o sistema;

- Criar uma imagem da PCB a analisar a partir da informação das duas câmaras;
- Comparar a imagem com o Gerber de referência;
- Identificar falhas na deposição da solda;
- Mostrar falhas na deposição ao operador com alertas;
- Integração do algoritmo com o hardware (*Raspberry Pi Compute Module*);
 - Algoritmo em C++ com a biblioteca OpenCV;
- Desenvolvimento de uma plataforma protótipo, com estrutura de suporte da placa, iluminação e suporte para o Raspberry Pi;
- Desenvolvimento de uma *Graphical User Interface* (GUI) para configurar, operar e visualizar os resultados;
 - Utilização da *framework* QT em C++;

1.3 Estrutura da Dissertação

Esta dissertação está dividida em 5 capítulos:

No Capítulo 2 é feita uma análise do estado da arte em relação à inspeção de solda em PCB. Além disso, é feito um pequeno resumo de várias técnicas utilizadas na inspeção de uma PCB.

No Capítulo 3 é feita uma descrição do sistema desenvolvido, material e componentes utilizados e o algoritmo desenvolvido.

No Capítulo 4 são apresentados os resultados de vários testes efetuados para avaliar o desempenho do algoritmo, bem como resultados relativos à eficácia do algoritmo na deteção de solda e em relação ao tempo necessário para executar o algoritmo.

No Capítulo 5 são apresentadas as conclusões deste trabalho, realizando-se uma análise do mesmo bem como possíveis melhoramentos que poderiam ser efetuados.

Capítulo 2

Conceitos e Revisão Bibliográfica

Inspeção é o processo que determina se um produto se encontra dentro das especificações, ou seja, se não apresenta erros ou defeitos. Durante o processo de produção e montagem de componentes (assemblagem) de uma PCB, existem várias etapas onde podem ocorrer erros e causar defeitos na mesma. Esses erros devem ser detetados o mais rapidamente possível de modo a que se possam evitar avarias numa fase final do produto em causa.

Esta dissertação foca-se apenas no problema da deteção de defeitos durante o processo de deposição de solda. Assim, a revisão do estado da arte deste capítulo aborda apenas várias tecnologias e técnicas relacionadas com a inspeção de pasta de solda depositada numa PCB. São referidos métodos de inspeção tanto para a deposição de solda, como para a PCB numa fase de produção avançada, com componentes, e são apresentados alguns métodos utilizados para o processamento e análise das imagens da PCB.

2.1 Deposição de Solda

O processo de deposição de solda, como já referido anteriormente no Capítulo 1, é o processo responsável por 50 a 70% dos erros dos defeitos originados numa linha de produção e montagem de PCB [2].

Existem várias técnicas para efetuar a deposição de pasta de solda em PCB como a impressão a jato, através do *stencil* ou manualmente com uma seringa. A deposição de solda por *stencil* [7] é o método mais comum devido à sua eficácia em PCB com um número elevado de *pads* (*pad* é região da PCB onde é colocada a pasta de solda para fixar os pinos do componente) e tamanho reduzido, como é o exemplo dos microprocessadores. O sistema desenvolvido tem como objetivo a análise de deposição da solda aplicada por *stencil*.

Para fazer a deposição de solda utilizando este método é necessário, não só a pasta

de solda, como também um *stencil*. O *stencil* é uma espécie de "folha" de metal ou de *polyimide* que contém apenas furos onde é necessário colocar solda, ou seja, onde estão localizados os *pads*. Na Figura 2.1 é possível observar um dos tipos de *stencil* usado, neste caso um *stencil* metálico.

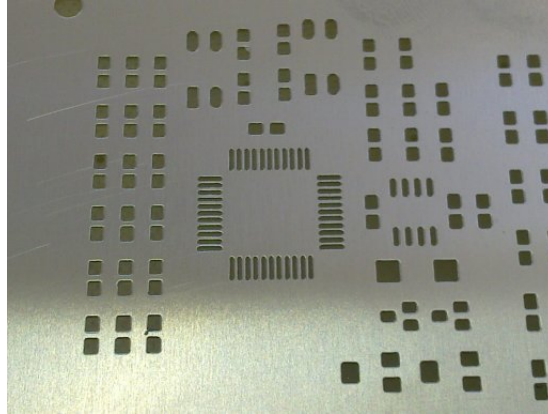


Figura 2.1: Stencil metálico [8].

Para que essa deposição seja feita, um rodo é movido ao longo da superfície do *stencil*, forçando a pasta de solda a mover-se e, com a pressão criada com esse movimento, a solda é espalhada ao longo de todos os *pads*. Quando o varrimento chega ao fim, o *stencil* é levantado e, todos os *pads* devem ter uma camada de pasta de solda correspondente à espessura do *stencil*, com o tamanho pretendido.

Durante o processo de deposição de pasta de solda na PCB existem vários fatores que podem levar a que existam defeitos nessa mesma deposição, sendo que um dos fatores é a própria solda.

A pasta de solda é composta por dois materiais: partículas de solda e fluxo. Essas partículas de solda estão suspensas no fluxo, que age como um adesivo temporário até que a solda seja derretida e unida aos componentes. A pasta de solda é um material tixotrópico, o que significa que a sua viscosidade é alterada ao longo do tempo. Quanto mais for misturada e trabalhada, menor será a viscosidade da mesma. Esta pode ser classificada de duas maneiras diferentes, pelo tipo de fluxo que nela está contido ou pelo tamanho das suas partículas. Na Tabela 2.1 é possível observar os vários tipos de solda existentes e os tamanhos associados aos mesmos.

O tamanho das partículas de solda tem influência na forma como a solda se comporta. Soldas com partículas mais pequenas são mais frágeis e necessitam de mais cuidados de armazenamento do que soldas com partículas maiores. Apesar disso, e segundo a "regra das 5 bolas" devem caber 5 partículas de solda lado a lado no menor *pad* de um *stencil*, o que leva a que cada vez mais sejam utilizadas soldas de tamanhos menores.

Em relação ao fluxo, este pode dividir-se em três tipos: baseado em resina, fluxo

Tipo	Menos de 0.5% maior que	Máximo 10% entre	Minimo 80% entre	Máximo 10% menor que
1	160	150-160	75-150	75
2	80	75-80	45-75	45
3	60	45-60	25-45	25
4	50	38-50	20-38	20
5	40	25-40	15-25	15
6	25	15-25	5-15	5
7	15	11-15	2-11	2

Tabela 2.1: Tipo e tamanho de partículas de solda em μm [9].

solúvel em água e fluxo sem limpeza (*no-clean*). Para além de manter as partículas de solda unidas, o fluxo também remove oxidação das superfícies metálicas, facilitando a adesão da solda nessas superfícies. O tipo de fluxo tem influência na quantidade de oxidação que consegue remover mas, ao remover essa oxidação e depois de derretido, o fluxo pode deixar marcas na PCB. Essas marcas podem causar corrosão em alguns casos e podem ser limpas com maior ou com menor facilidade, dependendo do tipo de fluxo utilizado. No caso de fluxos *no-clean*, essa limpeza não é necessária uma vez que os resíduos são mínimos.

As condições atmosféricas do local onde a solda é armazenada e aplicada também podem influenciar na aplicação da mesma. A pasta de solda deve ser armazenada num ambiente refrigerado, a uma temperatura entre 0 e 10°C, e antes de ser utilizada deve ser colocada à temperatura ambiente, entre duas a quatro horas. Se este procedimento não for cumprido, a viscosidade da solda pode ser afetada e o seu espalhamento não se efetuar corretamente, originando defeitos.

A qualidade do *stencil* é um fator também bastante relevante no processo de colocação da solda. Em [7], os autores concluíram que a variável mais crítica na deposição de solda é o tamanho dos *pads* e a espessura do *stencil*. Além disso, os autores também encontraram uma relação entre o tipo de solda e a velocidade a que a mesma deve ser espalhada pelo *stencil*.

Na Figura 2.2 são apresentados alguns exemplos de deposições de solda numa PCB onde as regiões mais claras são as superfícies metálicas dos *pads*, e as regiões mais escuras a solda depositada. Alguns destes erros, como é o caso do (b) e (c), podem ocorrer devido ao excesso ou falta de solda no *stencil*. No caso do (d) e (e) podem ocorrer devido ao mau alinhamento do *stencil* com a PCB.

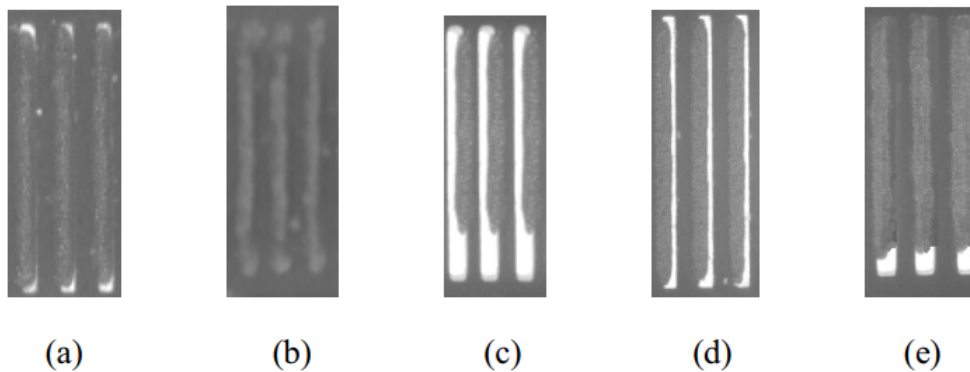


Figura 2.2: Defeitos de deposição de solda[10]: (a) bom; (b) excesso; (c) insuficiente; (d) deslocação horizontal; (e) deslocação vertical.

2.2 Métodos de inspeção de solda em PCB

Durante o processo de deposição de solda numa PCB é possível realizar a inspeção da mesma em duas fases: depois da deposição de pasta de solda e depois de soldado o componente. Para ambos os tipos de inspeção, existem vários métodos que podem ser utilizados sendo que, para inspeções onde o componente já está soldado, podem ser realizadas inspeções físicas em conjunto com inspeções visuais.

Inspeção Visual

A maioria da literatura de inspeção de pasta de solda é baseada em métodos de reconstrução 3D do modelo da pasta de solda depositada. Existem vários métodos para fazer a reconstrução, sendo um deles através da triangulação de laser[4, 5]. Este método utiliza um díodo laser que projeta luz para um objeto (ex. pasta de solda) e a luz refletida pelo objeto é detetada por um sensor, como é o caso da Figura 2.3. Sistemas deste tipo conseguem medir a altura de um bloco de pasta de solda com grande precisão e, de acordo com [4], é possível obter uma resolução de 0.00076 mm. Apesar da grande precisão, sistemas de inspeção deste tipo tipicamente são lentos na tarefa de inspeção devido ao varrimento que tem que efetuar por toda a placa, não sendo comportáveis para incluir numa linha de fabricação de circuitos eletrónicos.

Apesar de bastante eficientes na deteção e medição de pasta de solda em PCB, as técnicas que utilizam este método são bastante complexas e dispendiosas. Sistemas deste tipo podem custar entre 50 e 200 mil euros.

A inspeção usando perfilometria é um dos métodos mais utilizados para realizar inspeções em pasta de solda. Este método projeta, no objeto a analisar, um padrão de luz estruturado que é deformado de acordo com a pasta de solda, apresentando se-



Figura 2.3: Exemplo de um sistema de inspeção com o método de triangulação laser [11].

melhanças em relação ao método da triangulação por laser. Devido à reflexão causada pelo objeto, neste caso, pasta de solda, é formado um padrão de deformação que permite fazer a reconstrução em 3 dimensões do bloco de pasta de solda. Existem vários métodos para fazer a reconstrução utilizando perfilometria, sendo um deles a *Fourier Transform Profilometry* [3] (FTP) ou a *Fringe Projection Profilometry* [12] (FPP).

Em [13], os autores propõem um novo método, *two-step phase shifthing* (TSPS), para detecção do padrão de *fringe*. Este algoritmo é comparado com outros algoritmos, modificações dos algoritmos FTP e FPP onde, o algoritmo desenvolvido apresenta melhores resultados para objetos com formas ou texturas complexas.

Em [10], os autores desenvolveram um protótipo funcional para inspeção de pasta de solda com a utilização de redes neurais para classificação. A rede neuronal é treinada com várias amostras e consegue detetar com alta fidelidade os diferentes casos que podem ocorrer no processo de deposição de solda, como é o caso de excesso, falta ou deslocamentos verticais/horizontais.

Em [14], os autores desenvolveram um protótipo que consegue detetar vários defeitos em pasta de solda através de luz direcional. Para o sistema apresentado são capturadas 5 imagens, uma com luz superior e quatro imagens com luz direcional, uma para cada lado da PCB. Essas imagens são processadas e analisadas por um sistema *fuzzy* para que seja feita a classificação de todos os blocos de solda analisados.

Em [15], os autores utilizam um método similar ao [14]. Neste caso, são usados 3 padrões de iluminação, luz superior (branca), luz direcional na parte de cima e de baixo do bloco de solda e luz direcional em ambos os lados do bloco de solda. As luzes direcionais utilizadas são vermelha e azul, em direções opostas e, depois de processadas, é possível reconstruir o bloco de solda com a informação das sombras criadas pela luz. São utilizadas duas cores diferentes para ser possível distinguir a orientação dos defeitos detetados, sendo possível detetar casos onde existe falta de solda ou solda em excesso.

Em ambos [14] e [15], o sistema desenvolvido consegue analisar apenas uma pequena

parte da PCB e necessita de dois motores para movimentar a câmara nos eixos horizontal e vertical para que toda a PCB seja analisada. Este sistema, apesar de mais preciso, uma vez todos os *pads* são analisados com grande detalhe, é lento na inspeção de uma PCB com um número de *pads* elevado, o que leva a que não seja indicado para uma linha de produção.

Inspeção por Raio-X (BGA)

Depois de depositada a pasta de solda e garantir que está correta com as respetivas inspeções, o próximo passo é colocar e soldar os componentes numa PCB. Depois de realizada a soldadura a todos os componentes, é necessário inspecionar novamente a PCB, agora para garantir que todos os componentes estão soldados e não existem circuitos abertos ou curto-circuitos.

A *Surface-Mount Technology* (SMT) levou ao desenvolvimento de PCB com componentes muito pequenos, os *surface-mount devices* (SMDs) e, como estes componentes têm tamanhos muito reduzidos, é possível incorporar vários deste tipo numa pequena área.

A evolução desta tecnologia levou também ao desenvolvimento de circuitos integrados onde as componentes a serem soldadas não ficam visíveis após a colocação do componente, como é o caso do empacotamento BGA (Ball Grid Array). As PCB com componentes deste tipo não podem ser inspecionadas usando uma inspeção visual tradicional e por isso é usada a inspeção por raio-X[16, 17, 18]. Este tipo de inspeção consegue detetar soldaduras que não são visíveis bem como defeitos que possam ocorrer durante a fase de soldadura. Além disto, a inspeção por raio-X também consegue inspecionar PCB com várias camadas sendo possível avaliar se todas as camadas estão corretas e alinhadas.

Estes sistemas permitem realizar uma inspeção semelhante aos métodos usando perfilometria e permitem também que, depois dos componentes estarem soldados na PCB, seja detetado se essas soldaduras foram realizadas de forma correta sem que seja necessário realizar um teste físico à PCB.

Um dos maiores problemas desta tecnologia é o custo associado à mesma. Como é uma tecnologia complexa e dispendiosa, apenas se justifica o uso da mesma quando existem componentes onde a solda não é visível ou quando a densidade de componentes na PCB é tão elevada que a inspeção visual não consegue funcionar corretamente. Como em muitas linhas de produção não são produzidas placas com essas características, este tipo de inspeção é menos comum, sendo apenas usado por empresas especializadas.

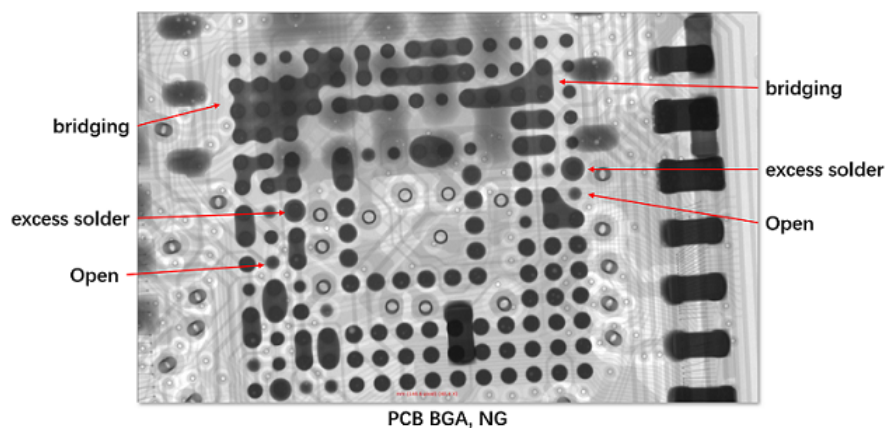


Figura 2.4: Exemplo de uma inspeção a um componente *Ball Grid Array*(BGA) [19].

Inspeção por contactos

Além das inspeções utilizando visão computacional, uma das formas de encontrar defeitos é através de testes físicos. Por teste físico entende-se testar as ligações da PCB e as suas funcionalidades de modo a apurar se a PCB se encontra a funcionar de forma correta. Apesar da inspeção visual ser bastante precisa, podem ocorrer erros que não são detetados, sendo por vezes necessário e recomendado o teste físico.

Uma inspeção por contactos, como o nome sugere, serve para testar a PCB através de contactos. São testados os vários componentes da PCB e os vários circuitos de modo a determinar se todos os componentes da PCB estão ligados e a funcionar de forma correta e prevista.

Este teste pode ser efetuado de duas formas: uma "cama de agulhas" ou um dispositivo de teste móvel. A cama de agulhas é composta por uma matriz de agulhas onde cada agulha faz contacto com uma parte do circuito a ser testado. Estas agulhas têm molas dentro delas de modo a "amortecer" quando o contacto é realizado. De referir ainda que podem ser efetuados centenas de contactos simultaneamente, sendo possível testar todo o circuito de uma só vez.

Um dos maiores problemas deste tipo de teste é o seu custo para desenvolver e a dificuldade de mudança do setup de agulhas. Além disso, para PCB com uma grande densidade de componentes, este tipo de teste apresenta alguns problemas, que podem ser resolvidos usando um dispositivo de teste móvel.

Os dispositivos de teste móvel, funcionam da mesma forma que uma "cama de agulhas", mas não necessitam de ter uma densidade de agulhas tão elevada como no outro caso. Como o dispositivo é móvel e controlado por software, é relativamente fácil, em comparação com o caso anterior, de testar vários circuitos diferentes, apenas fazendo contacto nos pontos onde é necessário. De referir ainda que este tipo de teste apenas serve

para testar as ligações entre os componentes e não a funcionalidade da PCB. Este teste parte do pressuposto que a PCB foi desenhada de forma correta e a funcionalidade da mesma é garantida com o funcionamento de todos os seus componentes.

Teste Funcional

Depois de realizadas as várias inspeções referidas anteriormente, o próximo passo é testar a funcionalidade da PCB. Apesar de não serem detetados quaisquer erros ou defeitos durante o processo de formação da PCB, podem ocorrer erros que afetem a funcionalidade da mesma. Esses erros podem ocorrer devido a ruídos do circuito, mau dimensionamento dos componentes, ou até por deficiente projeto da PCB.

Para testar a funcionalidade de uma PCB é necessário colocá-la no mesmo ambiente que irá ser usada. Estes testes muitas vezes são únicos e diferem de PCB para PCB, pois o ambiente pode variar. Na sua essência, este teste é composto por uma interface que irá ligar todas as entradas e saídas da PCB e simular o seu funcionamento. Como os testes são muitas vezes únicos, este método é complexo uma vez que é necessário mudar o *setup* de teste para cada PCB, de maneira semelhante aos testes físicos.

Apesar de testar a funcionalidade da PCB, este teste também não garante o completo funcionamento da PCB a não ser que sejam testados todos os casos e todas as variáveis que possam existir. É também nesta fase que podem ser determinados vários parâmetros do seu funcionamento, como é o caso dos consumos gerais de funcionamento.

Detetar erros nesta fase de inspeção significa que todos os processos que foram realizados anteriormente foram desperdiçados, tanto os componentes que estão na PCB assim como o tempo que foi necessário para criar a PCB em causa.

2.3 Aquisição de Imagem

Nesta secção descrevem-se as características principais de um sistema de visão computacional e todas as suas componentes, que permitem que seja feita a aquisição de imagens. A forma como é efetuada a aquisição de imagens, que torna possível analisar computacionalmente uma PCB, tem uma grande influência na análise da imagem, devendo por isso ser devidamente estudado e especificado.

Iluminação

A iluminação é um dos principais fatores para que uma inspeção seja realizada com sucesso[20, 21]. Quando não existe uma iluminação uniforme todo o produto a ser ins-

peccionado, ou existem sombras no mesmo, podem ser detetados defeitos que não existem ou não ser possível detetar um defeito que seria facilmente detetado com uma iluminação correta. É importante dimensionar um sistema de iluminação para, de acordo com os objetivos, iluminar de forma correta todo o produto a ser inspecionado.

Sendo o sistema de iluminação um dos principais componentes num sistema de visão artificial, o mesmo deve ser planeado de acordo com as necessidades. Nesse sentido, podem ser usados vários tipos de iluminação sendo que as tecnologias fluorescente, halogéneo e LED são as mais comuns em sistemas de menor dimensão. Em sistemas de dimensões maiores são mais comuns iluminações de xénon, vapor de sódio de alta pressão e mercúrio [22].

O processo de iluminação num sistema de visão artificial não depende apenas do tipo de iluminação usada, mas também a forma como a iluminação incide sobre o objeto e como está posicionada em relação à câmara. A posição da iluminação em relação à câmara e em relação ao objeto irá depender do que é pretendido pela inspeção. Pode ser necessário evidenciar relevos, sendo necessário sombras ou pode ser necessário uma iluminação uniforme em todo o objeto, sem sombras e sem reflexos.

• Iluminação direta

A principal função da iluminação direta é iluminar o objeto e revelar a sua forma e os seus contornos sendo que, muitas vezes são criadas sombras distintas. Este tipo de iluminação é utilizado em objetos que exigem um alto contraste, causando muitas vezes reflexões quando o objeto a analisar tem uma superfície refletora, como é o caso das PCB. Na Figura 2.5 (a) pode-se observar um exemplo deste tipo de iluminação que, caso o objeto fosse uma PCB, iria resultar em grandes reflexões que iriam ser observadas na imagem captada.

• Iluminação de fundo ou *backlight*

A iluminação de fundo ou *backlight* tem como função realçar o contorno dos objetos a analisar uma vez que consegue criar um contraste entre a iluminação e o objeto. É bastante utilizada para identificar buracos ou a ausência deles, a orientação de objetos ou para efetuar medições, uma vez que é bastante eficiente para efetuar deteções de contorno. Na Figura 2.5 (b) pode-se observar um exemplo deste tipo de iluminação. Caso o objeto a observar fosse uma PCB, o resultado seria o contorno da mesma bem como quaisquer furos que ela tivesse.

• Iluminação *Dark Field*

No tipo de iluminação *Dark Field*, a luz é projetada no objeto através de um ângulo

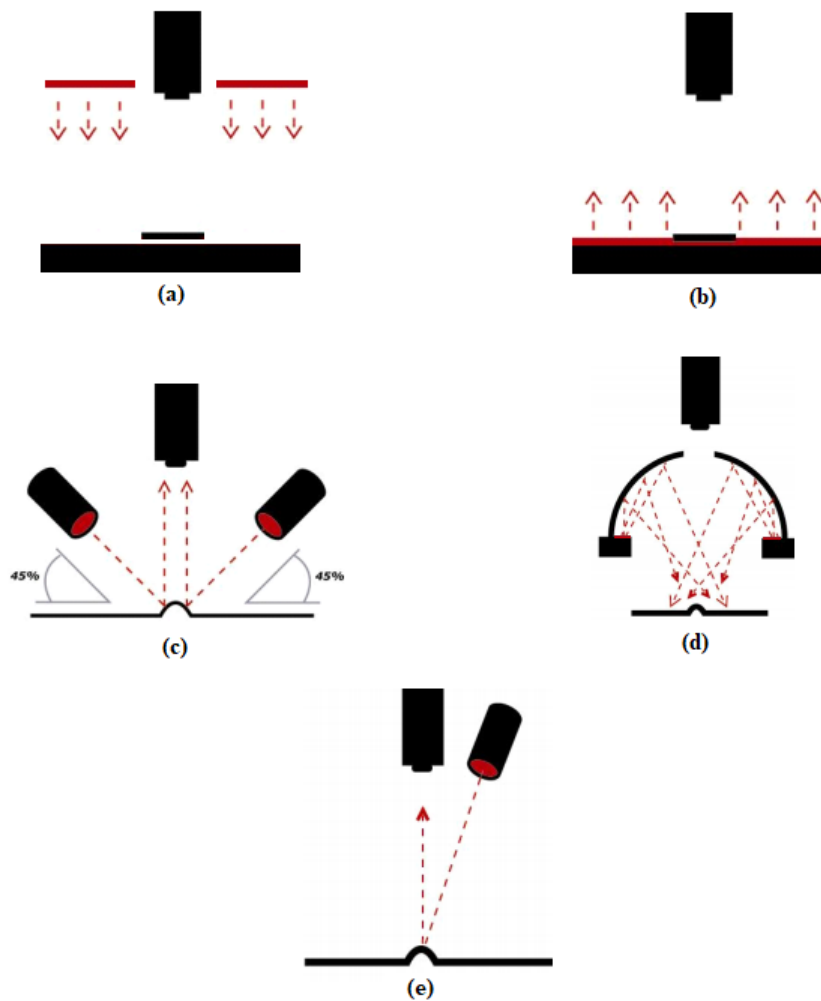


Figura 2.5: Tipos de iluminação [21]. As áreas vermelhas indicam as fontes de luz. (a) iluminação direta; (b) iluminação de fundo; (c) iluminação *dark field*; (d) iluminação difusa; (e) iluminação direcional.

reduzido e, deste modo, sobressaem quaisquer relevos que existam no objeto. Este tipo de iluminação permite também que não existam reflexos uma vez que a maior parte da luz é refletida para longe da câmara. Na Figura 2.5 (c) pode-se observar um exemplo deste tipo de iluminação e da forma como a luz é refletida.

• Iluminação Difusa

A iluminação difusa tem como objetivo iluminar todo o objeto e não apenas uma superfície. Como a iluminação não é direta, e é feita através de várias reflexões do meio, é possível iluminar várias partes do objeto, com a mesma intensidade, sendo possível obter reflexões em vários ângulos. Como várias partes do objeto são iluminadas uniformemente, quaisquer sombras produzidas pela iluminação são reduzidas. Na Figura 2.5 (d) é possível observar um exemplos deste tipo de iluminação.

• Iluminação Direcional

A iluminação direcional é tipo de iluminação mais utilizada, tanto para inspeção como no dia a dia, como é o caso da luz solar. Este tipo de iluminação é semelhante à iluminação direta, uma vez que permite obter um grande contraste entre o objeto a analisar e a superfície onde este se encontra, de modo a detetar com mais facilidade quaisquer relevos que existam no objeto. Uma vez que é direcionada, a reflexão da mesma é captada pela câmara, produzindo uma zona de brilho na zona pretendida, o que pode ser necessário dependendo dos casos. Na Figura 2.5 (e) pode-se observar um exemplo deste tipo de iluminação.

Câmaras

A câmara é uma componente tão importante como a iluminação para que a inspeção seja efetuada com sucesso. Apesar de ter a simples tarefa de capturar o objeto a ser inspecionado, existem vários fatores que podem influenciar a imagem capturada como o foco, a resolução do sensor, o tipo e qualidade da lente ou a profundidade de campo.

Na escolha de uma câmara para aquisição de imagem existem vários parâmetros que, dependendo da aplicação são relevantes. Em relação ao tipo de sensores, existem dois tipos comuns: complementary metal-oxide semiconductor (CMOS) e charge-coupled device (CCD). Nos sensores CMOS cada píxel contém um conversor faz a conversão luz-tensão e o sensor inclui muitas vezes amplificadores, corretores de ruído e circuitos digitais, sendo este sinal digital na sua saída. Estes sensores são mais rápidos que os CCD apesar de resultarem numa imagem com menor qualidade e menos detalhe.

No sensor CCD, a carga de cada pixel é transferida ao longo da mesma linha e coluna e os sinais resultantes são multiplexados, convertidos em tensão e o sinal analógico é enviado. Estes sensores conseguem capturar imagens de alta qualidade com pouco ruído, porém, o custo de fabrico destes sensores é mais elevado e consomem mais energia [23].

Outro dos parâmetros é o modo como a imagem é adquirida, se é matricial, se é linear ou 3D. O tipo de câmara mais conhecido é a câmara matricial, sendo este o tipo de câmara mais utilizado para as aplicações do dia a dia. O sensor da câmara captura uma determinada área e replica a informação numa matriz de pixéis a duas dimensões.

No caso das câmaras lineares, a imagem é capturada apenas por uma linha de pixéis, em vez de uma matriz como na câmara matricial. Neste tipo de câmara, é necessário que exista movimento para que o objeto seja capturado com sucesso, tendo vantagens no registo de objetos em movimento. A utilização deste tipo de câmaras é vantajoso para inspeções de objetos cilíndricos pois torna possível a inspeção de todo o objeto, fazendo-os apenas rodar sobre si próprios [24].

Quanto às câmaras 3D, como o nome indica, permitem captar informações de um objeto a três dimensões. Este tipo de aquisição também pode ser realizado com recurso a duas câmaras (câmara stereo), simulando a visão humana. Este setup deve ser usado para inspeções de objetos onde existam defeitos com características reconhecíveis em 3D.

2.4 Processamento de Imagem

Depois de realizada a captura do objeto a analisar, o próximo passo da inspeção é identificar um possível defeito. De modo a que seja possível realizar essa identificação, é necessário recorrer a alguns métodos de processamento e análise de imagem, que são cruciais para a identificação de defeitos.

O processamento de imagem pode ter diversos objetivos, tais como melhorar a imagem, remover ruído ou interpretar a informação da imagem. A maioria das técnicas de processamento trata a imagem como uma matriz de valores e, com recurso a diversos métodos é possível efetuar a sua análise.

Nesta secção são revistas algumas das técnicas de processamento de imagem que foram utilizadas no desenvolvimento do trabalho.

Representação de Imagens

De modo a obter uma imagem, como já referido anteriormente na Secção 2.3, é necessário capturar o objecto de estudo através de uma câmara, que transforma a luz capturada numa matriz com valores, ou seja, a imagem. Cada elemento na imagem (tradicionalmente retangular) é designado por "pixel" (*Picture Element*). Cada pixel pode ser constituído por várias componentes ou apenas por uma componente, no caso de ser uma imagem apenas com níveis de cinza. Cada amostra de uma componente será representada numa gama de valores dependendo do número de bits utilizado para a representar. Por exemplo, no caso de se utilizar 8 bits para representar cada amostra de uma componente do pixel, a gama de valores varia entre 0 e 255, que representam a intensidade luminosa captada.

É necessário referir também que, na maioria das câmaras comuns, as imagens são capturadas no espaço de cor RGB (*Red, Green, Blue*), contendo três componentes (ou canais) distintos, o canal vermelho, canal verde e canal azul. Isto leva a que, em vez de apenas existir um valor por pixel, tipicamente existem três valores para cada pixel numa imagem, correspondendo aos três canais.

Além das imagens a cor, que apresentam três canais por pixel, existem mais dois tipos de imagem comuns: imagens na escala de cinza e imagens binárias. As imagens na escala de cinza apresentam apenas um valor para cada pixel, tipicamente uma gama entre 0 e

255 (8 bits) onde 0 é a cor preta (menor intensidade de luz) e 255 é a cor branca (maior intensidade de luz).

As imagens binárias, como o próprio nome indica, são imagens que apenas apresentam um valor por cada pixel, sendo este valor 0 ou 1. Esta imagem apenas necessita de 1 bit por cada pixel, onde 0 é considerado como preto e 1 é considerado branco.

Espaços de Cores

Um espaço de cores é a forma como os vários canais de cor são organizados para representar a imagem de píxeis. A cor pode ser representada com várias componentes, dependendo do espaço de cor utilizado, e a combinação linear de todas as componentes cria a cor. De seguida são apresentados alguns dos espaços de cores mais utilizados sendo eles o RGB, YCbCr e HSV.

• RGB

Como já referido anteriormente, tradicionalmente, uma imagem é captada e armazenada no espaço de cores RGB, ou seja, cada pixel da imagem é representado através de 3 valores, um para o vermelho (*Red*), um para o verde (*Green*) e outro para o azul (*Blue*). Geralmente, este espaço de cores é representado com 8 bits por canal, ou seja, como cada canal pode ter 256 valores diferentes, podem ser representadas 16.8 milhões de cores diferentes.

Neste espaço de cor as três componentes estão relacionadas umas com as outras, ou seja, cada componentes apresenta a informação relativa à sua cor para cada pixel da imagem. As imagens relativas a cada componente diferenciam-se pela maior ou menor preponderância dessa componente de cor na imagem adquirida.

• YCbCr

Enquanto que no espaço de cores RGB as componentes estão relacionadas, existem vários espaços de cores onde as várias componentes não estão relacionadas como é o exemplo do YCbCr.

No espaço de cores YCbCr existem dois tipos de componente, a luminância (Y) e as crominâncias (Cr e Cb). A componente Y é a componente onde está contida a maior parte da informação da imagem (relacionada com a intensidade luminosa). Dependendo da aplicação, pode ser necessário apenas analisar esta componente da imagem, poupando tempo de processamento e a memória utilizada para armazenamento. A informação da cor está presente nas outras duas componentes (Cr e Cb), que corresponde às diferenças de cor relativamente à cor R e B. Como a cor é menos importante no sistema de visão

humano, muitas vezes as crominâncias são representadas com menor resolução espacial que a componente Y. De modo a obter uma imagem com este espaço de cores, é necessário fazer a conversão a partir do RGB. Para fazer essa conversão podem ser utilizados vários standards que dependem do número de cores que é possível apresentar, como é o caso do ITU-R BT.2020 [25, 26] ou ITU-R BT.709 [26].

Na Figura 2.6 é apresentada uma das equações de transformação do espaço de cores RGB para YCbCr de acordo com o standard ITU-R BT.2020.

$$Y' = 0.2627R' + 0.6780G' + 0.0593B'$$

$$C'_B = \frac{B' - Y'}{1.8814}$$

$$C'_R = \frac{R' - Y'}{1.4746}$$

Figura 2.6: Equação de transformação do espaço de cores RGB para YCbCr [25]. Y' , C'_B e C'_R simbolizam as componentes do espaço YCbCr. R' , G' e B' simbolizam as componentes do espaço de cores RGB.

- **HSV**

Outro dos espaços de cores bastante utilizados é o HSV, que é definido por três componentes: a matiz (Hue), a saturação (Saturation) e o valor (Value). Na componente H é apresentada a cor. Esta componente está relacionada com o comprimento de onda da cor original em RGB. A componente S representa a saturação dessa cor, quanto maior for o valor, maior será a vividez da mesma. Quanto menor for, menor será a vividez, resultando numa cor mais acinzentada. Quanto à componente V, representa a luminosidade, que é bastante semelhante à componente Y do espaço de cores YCbCr.

Distorção radial

Durante o processo de aquisição de uma imagem podem ser gerados vários tipos de distorção, que podem ou não ocorrer dependendo dos vários parâmetros da câmara e da imagem a ser capturada. A distorção radial, causada principalmente pela lente da câmara, leva a que linhas que deveriam aparecer retas na imagem, apresentem uma curvatura. Na Figura 2.7 são apresentados dois tipos de distorção mais comuns, a distorção de barril e a distorção de almofada (*pincushion*). Na distorção de barril, a grelha fica com efeito de como fosse envolvida num barril, dando a ilusão de que um ponto no centro da grelha está mais próximo da câmara que um ponto numa das extremidades da grelha. Este tipo de distorção é bastante comum em lentes com campos de visão maiores. Quanto à distorção

de almofada, acontece o inverso da distorção da barril. Este tipo de distorção dá a ilusão de que um ponto no centro está mais longe do que um ponto numa extremidade da grelha.

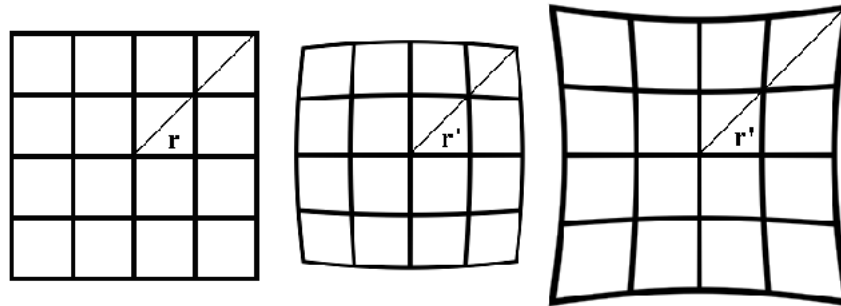


Figura 2.7: Distorção radial: Grelha da esquerda sem distorção, grelha do meio com distorção de barril, grelha da direita com distorção de almofada [27].

Para estes tipos de distorções foram desenvolvidos algoritmos para remover a distorção, como é o caso do modelo de Brown [28, 29]. Com a utilização de bibliotecas como a OpenCV (*Open Source Computer Vision Library*) é possível determinar os parâmetros intrínsecos da câmara que são:

- distância focal (em pixel);
- ponto principal da imagem (correspondente ao centro da lente);
- coeficientes de distorção radial e tangenciais.

Através do procedimento de calibração, com base nestes parâmetros, a imagem pode ser transformada/corrigida de forma a reduzir os efeitos da distorção radial.

Homografia

A homografia é a relação entre duas imagens da mesma superfície plana. Pontos numa superfície plana de uma imagem estão relacionados com pontos da mesma superfície noutra imagem através de uma relação de homografia[30]. Esta relação permite realizar transformações geométricas, ou seja, ao aplicar a homografia, os pontos de uma das imagens ficam alinhados com os mesmos pontos noutra imagem.

Segmentação

A segmentação consiste na divisão dos píxeis da imagem em vários grupos com características diferentes, onde dentro de cada grupo os píxeis possuem características semelhantes. Esta segmentação pode ser realizada através de regiões ou de uma característica

comum entre vários pixels. Existem vários tipos de filtragens e operações que podem ser efetuadas para que seja possível dividir uma imagem em várias regiões.

- **Binarização**

A binarização é o método utilizado para transformar uma imagem em tons de cinza numa imagem a preto e branco, ou seja, com apenas dois valores por pixel. Este tipo de operação é muito utilizado e permite a identificação de forma simples de objetos. O método de Otsu é um exemplo conhecido desta técnica que permite efetuar a binarização de uma imagem com definição do *threshold* de forma automática. O *threshold* é o valor limite que faz a distinção entre o que apresenta valor 0 (preto) ou valor 1 (branco). Num cenário ideal, ao analisar o histograma de uma imagem em tons de cinza eram detetados dois picos com um vale entre eles e a divisão ideal seria feita nesse mesmo vale. Na realidade não é tão simples identificar esse vale com precisão em grande parte das imagens, especialmente em casos onde o vale contém ruído. O algoritmo desenvolvido por Otsu[31] procura por um valor (onde é feita a divisão) que maximize a variância entre duas classes (neste caso os picos). Na Figura 2.8 é apresentado o exemplo de um histograma onde é detetado o valor que apresenta uma maior variância entre classes e esse mesmo ponto serve como referência para a binarização.

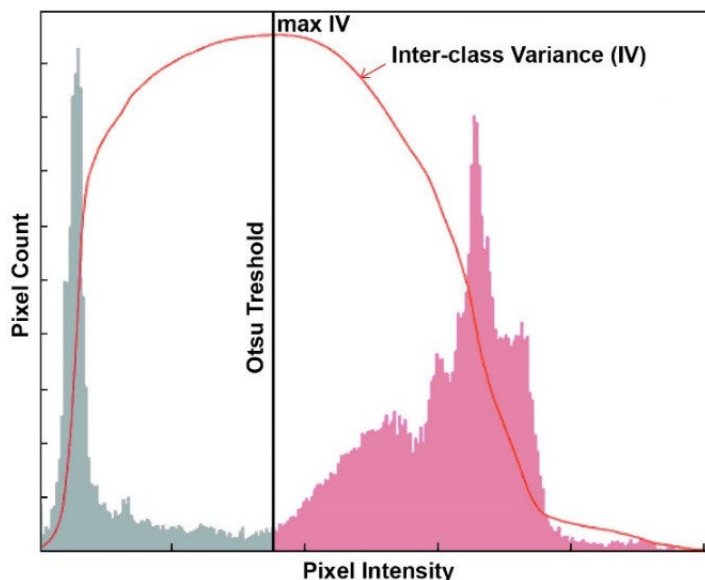


Figura 2.8: Visualização do método de Otsu[32]. O valor escolhido é o ponto onde existe maior variância entre as duas classes, identificadas a cinzento e vermelho no gráfico.

Apesar de bastante reconhecido, este algoritmo apresenta falhas. Como já referido anteriormente, quando não existem dois picos bastante definidos, quando existe ruído entre os dois picos ou mesmo quando um objeto tem um tamanho reduzido em comparação com o fundo, existindo a possibilidade de encontrar um falso máximo e, eventualmente,

realizar uma má segmentação. Foram realizados vários melhoramentos a este método, para colmatar os defeitos que ele apresenta, sendo um deles a utilização de pontos vizinhos para calcular a variância máxima. Quando se pretende calcular a variância entre dois conjuntos de valores, ao utilizar-se os valores vizinhos para o cálculo são atenuados ruídos existentes, sendo possível calcular com mais precisão a variância máxima numa imagem com bastante ruído.

• K-means Clustering

Outro método para fazer a segmentação de uma imagem é através do agrupamento de píxeis, ou seja, dividir uma imagem em vários grupos dependendo das suas características. Este algoritmo permite escolher o número de grupos e, no caso de processamento de imagem, divide a imagem nesse mesmo número de grupos definidos. Existem várias configurações, mas a mais comum utiliza uma técnica de aperfeiçoamento iterativo. Este algoritmo foi utilizado para fazer a deteção automática da cor da pasta de solda. Este algoritmo começa por definir o centro dos 'y' grupos definidos, sem qualquer critério. Depois disso são efetuadas duas tarefas iterativamente até que essas duas tarefas não obtenham melhores resultados. Uma dessas tarefas é o cálculo do quadrado da distância Euclidiana para todos os pontos, neste caso píxeis, em relação ao centroide mais próximo (centro dos grupos definidos inicialmente). Depois desse cálculo é efetuada a próxima tarefa que é o recálculo dos centroides. O objetivo é reduzir a distância Euclidiana dentro de cada centroide e, à medida que são recalculados os valores do centroide, este move-se de modo a reduzir a distância Euclidiana média.

O algoritmo é finalizado quando, ao recalculando o centroide, a distância Euclidiana total dentro do mesmo apenas aumenta, em vez de diminuir. Quando isto acontece para todos os centroides, o algoritmo está otimizado para o caso em específico. Na Figura 2.9 é apresentado um exemplo do funcionamento deste algoritmo com um grupo de dados e é possível observar as diferentes aproximações que são feitas de modo a dividir o grupo de dados em 3 grupos diferentes, de acordo com as suas características.

Na Figura 2.10 é possível observar o resultado do algoritmo numa imagem. Para o caso de imagens, cada centroide será uma cor e cada pixel pertence ao centroide do qual a cor está mais próxima. Ao adicionar mais centroides, é possível adicionar mais cores e obter uma segmentação melhorada. É necessário definir o número de centroides de acordo com a tarefa a executar.

• Deteção de características chave

A deteção de características chave ou *feature detection* é usada para identificar regiões de uma imagem que apresentam certas características que podem ser, por exemplo, pontos,

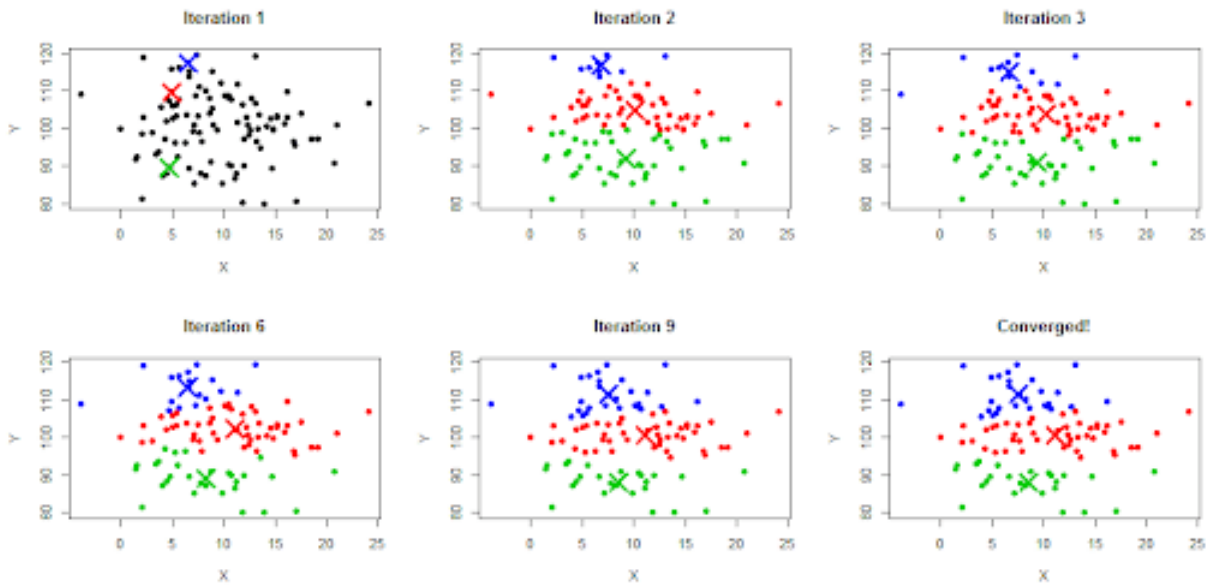


Figura 2.9: Visualização do algoritmo k-means em funcionamento [33]. O centro de cada centroide é assinalado com um X e os vários pontos são agrupados em relação à distância para o centroide.

arestas ou texturas. O conceito de *feature* depende do tipo de aplicação e da tarefa a executar sendo muitas vezes, na visão computacional, uma parte importante do algoritmo.

Como já referido anteriormente, existem vários tipos de características que podem definir uma imagem e as arestas são uma dessas características. Arestas são pontos de fronteira entre duas regiões diferentes e geralmente são caracterizadas por uma diferença significativa de magnitude em relação aos pixels vizinhos. Os métodos Canny [35] ou Sobel [36] são dois exemplos de detetores de arestas bastante reconhecidos e utilizados devido aos seus bons resultados. Na Figura 2.11 é apresentado um exemplo simples do detetor de arestas de Canny.

Um ponto de interesse pode ser detetado com facilidade por vários algoritmos e pode ser um canto, um ponto isolado com uma intensidade diferente da vizinhança ou até um ponto onde uma curva atinge curvatura máxima. Muitos dos algoritmos de deteção de cantos na realidade são detetores de pontos de interesse. Estes algoritmos necessitam de avaliar cada ponto detetado e, caso sejam realmente cantos, apenas esses são contabilizados, descartando os restantes. Os detetores de Harris [38] e Shi-Tomasi [39] são dois dos detetores mais utilizados na deteção de cantos, devido aos seus resultados. Na Figura 2.12 é apresentado um exemplo do funcionamento do detetor de cantos de Harris onde é possível observar a deteção de vários cantos na imagem.

Os detetores de bolha (*blob detector*) detetam regiões de interesse em vez de pontos de interesse, como os detetores de cantos. Uma bolha é uma região de uma imagem com propriedades semelhantes, onde todos os pontos dentro dessa bolha podem ser considera-



Figura 2.10: Visualização do algoritmo k-means numa imagem [34]. O número de centroides é definido por K , quanto maior o número de centroides mais cores será possível obter.

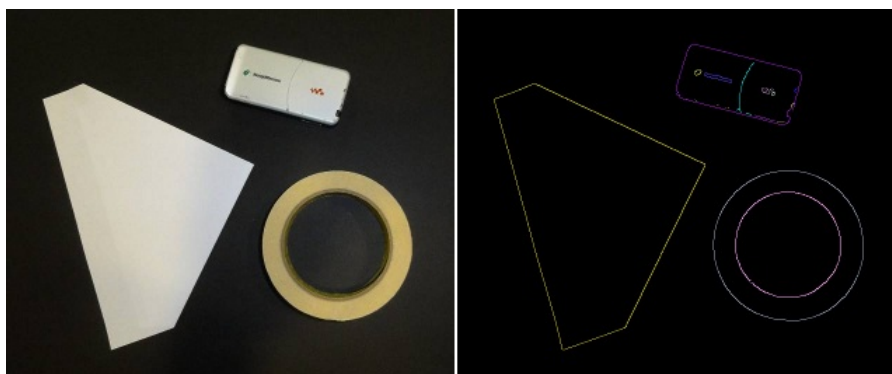


Figura 2.11: Exemplo da aplicação do detetor de arestas de Canny [37].

dos iguais ou muito parecidos. Este tipo de detetor pode ser usado como complemento aos detetores de arestas e cantos, uma vez que pode conter informações que não é possível obter com os anteriores. Existem vários detetores de bolha reconhecidos, como é o caso do Laplaciano da Gaussiana (LoG) [41], a Diferença de Gaussianas (DoG) [41] e o Determinante de Hessiano (DoH) [41].

Os detetores e descritores permitem não só identificar regiões e pontos de interesse mas também gerar uma "descrição" desses mesmos pontos ou regiões. É possível, ao usar este tipo de detetor, obter uma descrição do objeto detetado. Esta descrição pode depois ser usada para identificar um objeto ou região numa imagem que contenha esse mesmo objeto ou região. Na Figura 2.13 é apresentado um exemplo do que é possível fazer com um detetor e descritor.

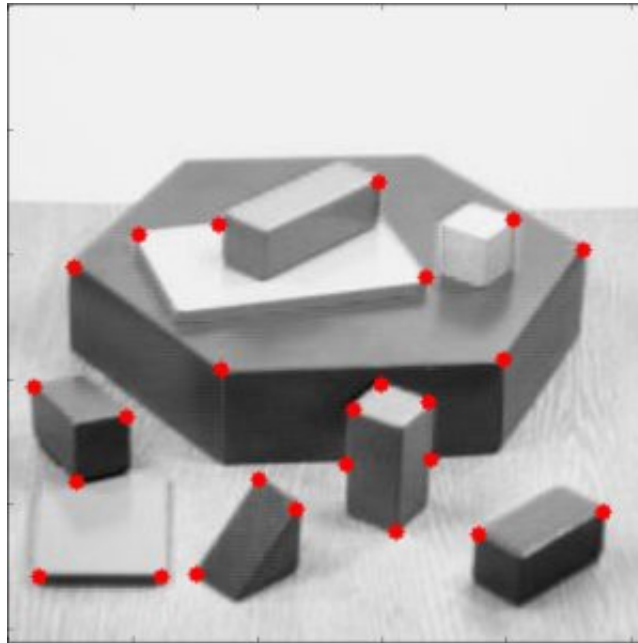


Figura 2.12: Exemplo da aplicação do detetor de cantos de Harris [40]. Todos os cantos detetados com a configuração definida são pintados a vermelho.

Existem alguns algoritmos bastante populares que são utilizados como detetores e descritores, como é o caso do SIFT (*Scale-Invariant Feature Transform*) [41] e do SURF (*Speeded-Up Robust Features*) [43]. O SURF é uma versão otimizada do SIFT que, como o nome sugere, é uma versão mais rápida e mais robusta. Estes algoritmos consistem em duas partes, a deteção e a descrição. Em ambos os algoritmos, são usadas variações de algoritmos de deteção de bolha. No caso do SIFT é usada uma adaptação do DoG e no caso do SURF é usada uma adaptação do DoH. Quanto à descrição, o objetivo é apresentar uma descrição única e robusta de um ponto de interesse com base na intensidade e distribuição de valores de pixels da sua vizinhança.



Figura 2.13: Exemplo de correspondência de pontos de interesse [42].

Capítulo 3

Sistema de Inspeção

Neste capítulo é apresentado o sistema de inspeção desenvolvido, composto pela estrutura de suporte, todos os componentes necessários para que a inspeção seja realizada e, por fim, o algoritmo de detecção e visualização de falhas.

3.1 Descrição do Sistema

Este projeto tinha como objetivo o desenvolvimento de um protótipo funcional de um sistema de inspeção de pasta de solda em PCB para ser usado em ambiente real na linha de produção de uma empresa. Neste âmbito pretendia-se desenvolver e implementar um algoritmo de detecção de falhas num sistema computacional de baixo custo, neste caso, um Raspberry Pi Compute Module 3+, como se pode ver na Figura 3.1a.

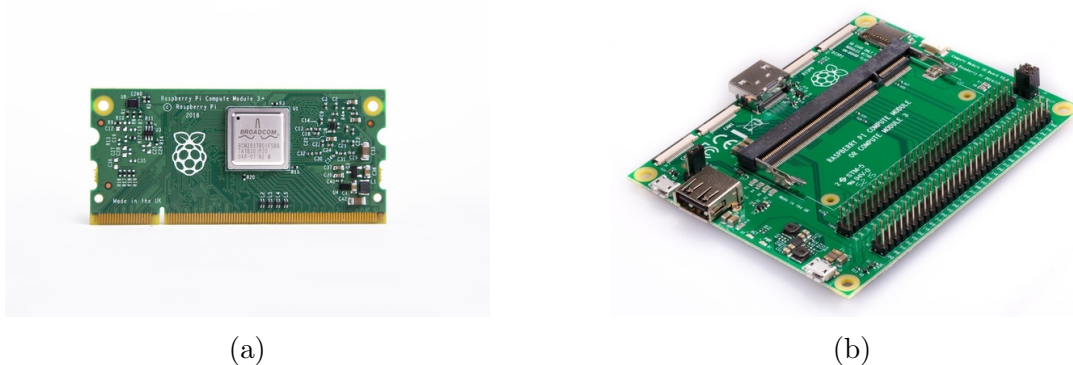


Figura 3.1: Raspberry Pi Compute Module 3+ (a) [44] e placa de IO (b) [45].

Este módulo contém um processador (ARMv8 de 64 bits a 1.2GHz), memória RAM (1GB), uma eMMC (64GB) e os circuitos de alimentação necessários. Isto permite que o módulo seja inserido numa placa de IO customizável utilizando o *hardware* do Raspberry Pi. Para implementar o algoritmo, foi utilizado uma placa de IO, apresentada na

Figura 3.1b que permite que o módulo seja alimentado e tenha acesso aos portos IO (Input/Output). Esta placa contém 120 pinos de GPIO (general purpose input/output), uma porta HDMI, uma porta USB, duas portas para câmaras e duas portas para *display*.

Foram utilizadas duas câmaras Raspberry Pi Camera Module v2, que se pode ver na Figura 3.2, da Sony, que contém o sensor IMX219 (CMOS) de 8 Megapixel para aquisição das imagens das PCB a analisar.

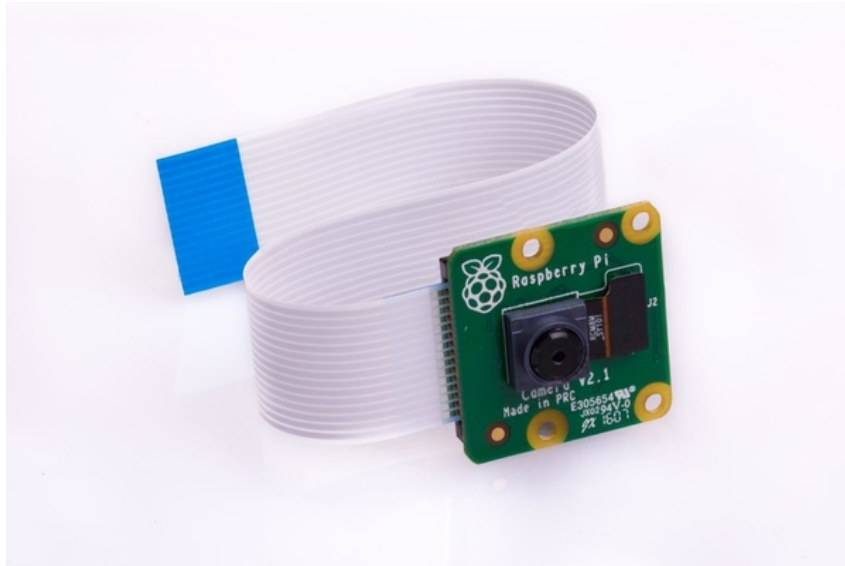


Figura 3.2: Raspberry Pi Camera V2 [46].

As câmaras foram instaladas nos extremos do plano de trabalho, onde a PCB é colocada, com uma inclinação relativamente a este plano, como se pode ver na Figura 3.3.

Esse plano de trabalho é visto pelas duas câmaras e a imagem analisada é construída aproveitando a zona do plano de trabalho mais próxima de cada câmara, de modo a obter uma imagem final com maior resolução espacial. Caso a PCB a analisar seja pequena, apenas será necessário utilizar uma das câmaras, neste caso, a câmara mais próxima da PCB. A calibração e alinhamento com a referência são feitas individualmente para cada câmara.

De modo a suportar o sistema de câmaras e iluminação, foi projetada e construída uma estrutura através de barras de perfil em alumínio de 40mm por 40mm, como se mostra na Figura 3.4.

Esta estrutura tem uma área útil de inspeção de 240mm por 350mm, permitindo inspeções de PCB até essa dimensão. Contém também suportes para as câmaras, que foram desenhados e impressos com uma impressora 3D para acomodar as câmaras na estrutura, apresentados na Figura 3.5. Estes suportes são ajustáveis, permitindo alterar a distância em relação à PCB, assim como o ângulo de visualização.

O suporte onde são colocadas as PCB a analisar também é ajustável em ambos os eixos

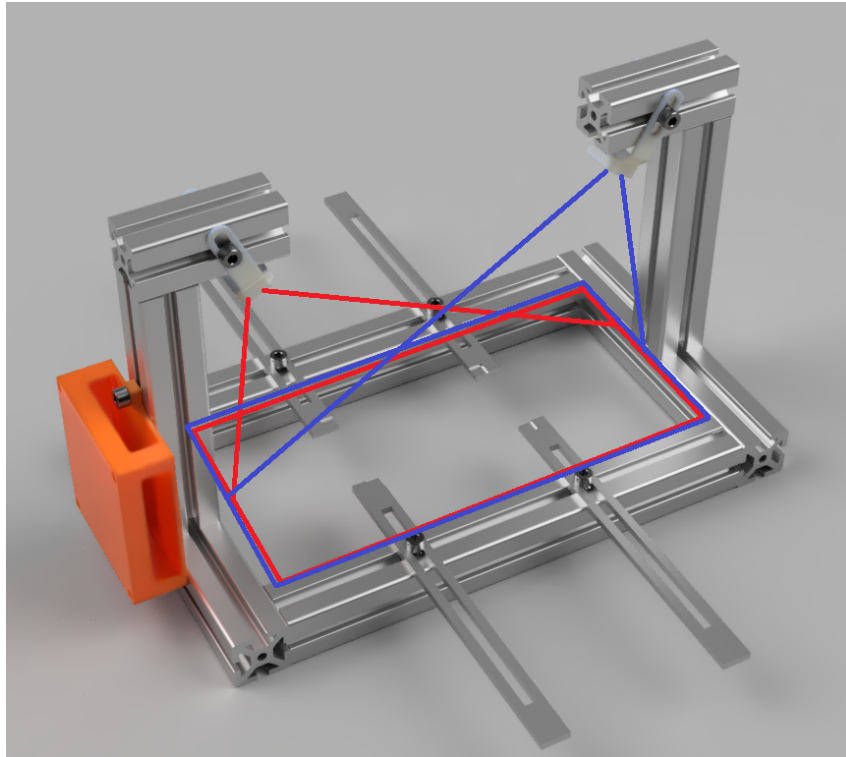


Figura 3.3: Área de trabalho e orientação das câmaras.

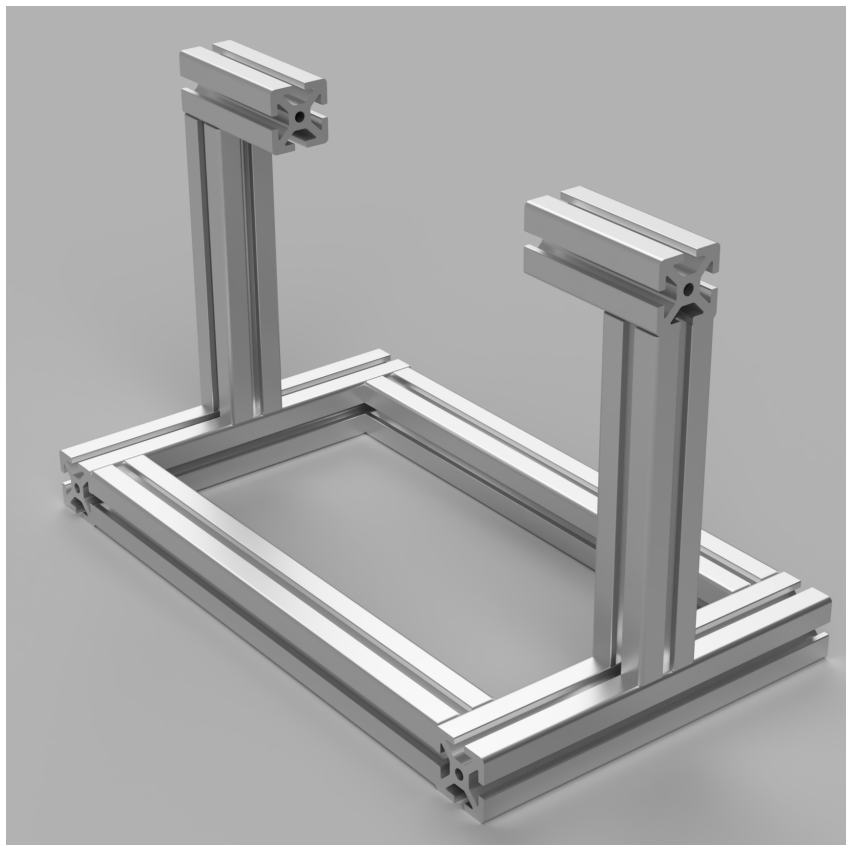


Figura 3.4: Estrutura construída com perfil em alumínio.



Figura 3.5: Suportes para as câmaras.

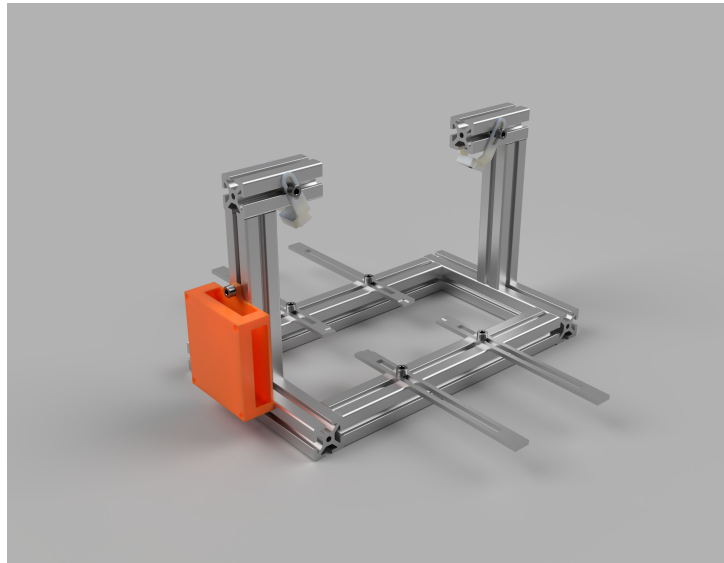
permitindo assim utilizar PCB de vários tamanhos. Estes suportes foram projetados e construídos a partir de uma barra de alumínio, o que dá robustez para que ao longo do processo de inspeção esse mesmo suporte não mude de posição.

Na Figura 3.6 é possível observar a versão inicial do protótipo funcional do sistema de inspeção de pasta de solda, composta por todas as partes necessárias na Figura 3.6a e o protótipo funcional com todas as partes pronto a operar na Figura 3.6b.

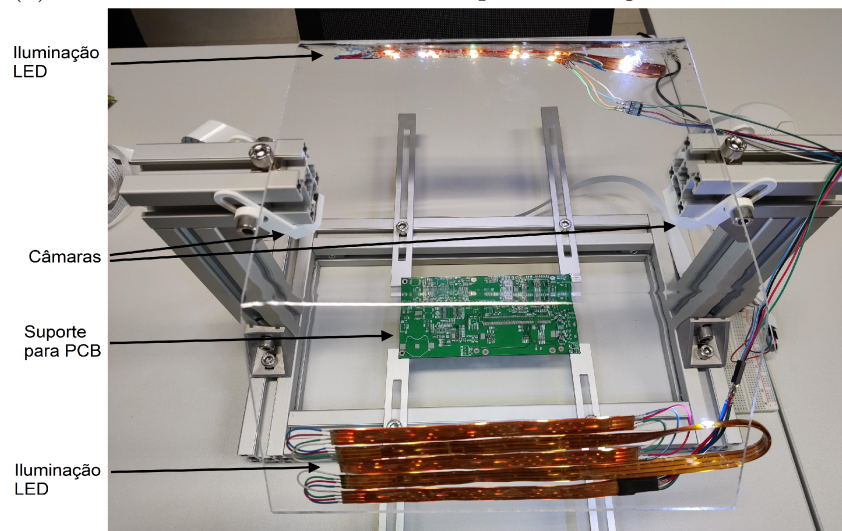
Em relação à iluminação, esta é do tipo "Dark Field" e são utilizados dois painéis de LED branco, em vez das fitas de LED RGB patentes na Figura 3.6b, uma vez que não se revelou vantajoso a utilização de iluminação de várias cores, mas sim iluminação de cor branca com maior intensidade. Na Figura 3.7 é possível ver o protótipo com todas as suas componentes:

- Raspberry Pi na caixa laranja;
- Iluminação em ambos os lados da estrutura;
- Monitor com a interface gráfica;
- Câmaras nos seus suportes nas barras verticais;
- PCB na posição de inspeção.

Estes painéis estão ao lado da PCB, perpendiculares às câmaras. Esta disposição permite que exista uma iluminação uniforme ao longo da PCB sem que sejam capturados



(a) Estrutura mecânica com a caixa para colocação da eletrônica.



(b) Versão inicial do sistema de iluminação.

Figura 3.6: Protótipo do sistema de inspeção.

quaisquer reflexos pelas câmaras. Embora no protótipo desenvolvido não seja possível, a disposição das câmaras e dos painéis de iluminação permite manter acesso livre ao plano de inspeção por cima.

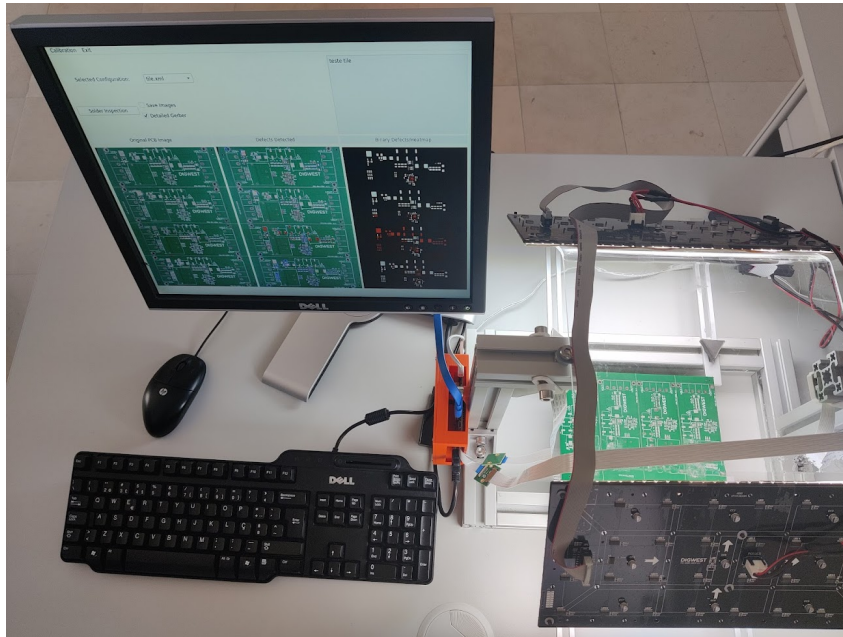


Figura 3.7: Versão final do protótipo do sistema de inspeção.

3.2 Aplicação com o algoritmo de Inspeção

Em relação ao algoritmo de processamento de imagem, este está dividido em duas etapas distintas, a etapa da calibração e a etapa da inspeção. A calibração permite calcular os parâmetros intrínsecos da câmara e os parâmetros extrínsecos, que relacionam a posição do plano de inspeção com a posição da câmara. É feita também a calibração do posicionamento da placa no plano de inspeção e a sua correspondência com a imagem de referência (Gerber). Esta etapa de calibração apenas é realizada uma vez por cada tipo de PCB a inspecionar.

3.2.1 Calibração

Nesta secção será exposto o algoritmo de calibração do sistema.

• Parâmetros de Entrada

Para realizar a calibração do sistema, é necessário configurar vários parâmetros de entrada.

O formato Gerber [47] é um *standard* usado na indústria das PCB onde são descritas, através de imagens, todas as informações acerca da PCB, tais como as camadas de cobre, a máscara da solda, as legendas, a informação dos furos, entre outras. Para este processo de calibração são utilizados 3 ficheiros Gerber, um apenas com o contorno da PCB, um com vários detalhes:

- legendas;
- buracos;
- pads;

e por fim, um ficheiro com os *pads* onde é depositada a pasta de solda. O ficheiro de Gerber com o contorno é utilizado apenas numa fase inicial para pré-processamento, na definição do rácio entre altura e largura da PCB e define o tamanho dos outros dois *Gerbers*, removendo qualquer informação presente fora do contorno, caso exista. O ficheiro Gerber com os vários detalhes da PCB é utilizado numa fase mais avançada da calibração para alinhamento com a PCB que está a ser analisada. Por fim, o ficheiro Gerber apenas com os *pads* é utilizado na fase de inspeção como referência de onde deve estar e onde não deve estar colocada pasta de solda.

Além dos ficheiros Gerber, é necessário também um padrão xadrez de calibração. Este padrão é utilizado para detetar o plano de referência onde se encontra a PCB. Esta operação permite realizar uma transformação homográfica da imagem capturada para o plano de trabalho.

Por fim, é necessário capturar imagens do xadrez de calibração e da PCB colocados no plano da de trabalho, ou seja nas barras de apoio.

• Fluxograma do sistema

Como já referido anteriormente, a etapa da calibração é onde são calculadas as matrizes de homografia necessárias para transformação da imagem da PCB na imagem a ser analisada, para além de outros parâmetros essenciais para o sistema, como é o caso da posição geométrica da PCB no plano de trabalho e as suas dimensões. Na Figura 3.8 é apresentado o fluxograma da etapa de calibração e inspeção, onde são apresentadas todas as operações efetuadas.

• Remoção de distorção

A primeira operação a ser efetuada é a remoção de distorções da lente através da calibração da câmara. Como já referido anteriormente na Secção 2.3, a imagem pode conter uma distorção causada pela lente da câmara, que é necessário corrigir. Todas as imagens capturadas pela câmara podem conter a distorção calculada anteriormente e, através do cálculo dos parâmetros intrínsecos da câmara, é obtida uma matriz de transformação que é aplicada a todas as imagens de modo a corrigir esse problema. O cálculo desses parâmetros é feito através do OpenCV onde são dadas várias imagens do xadrez de calibração em várias posições diferentes. Na Figura 3.9 são apresentadas duas das imagens que são utilizadas para o cálculo dos parâmetros intrínsecos da câmara.

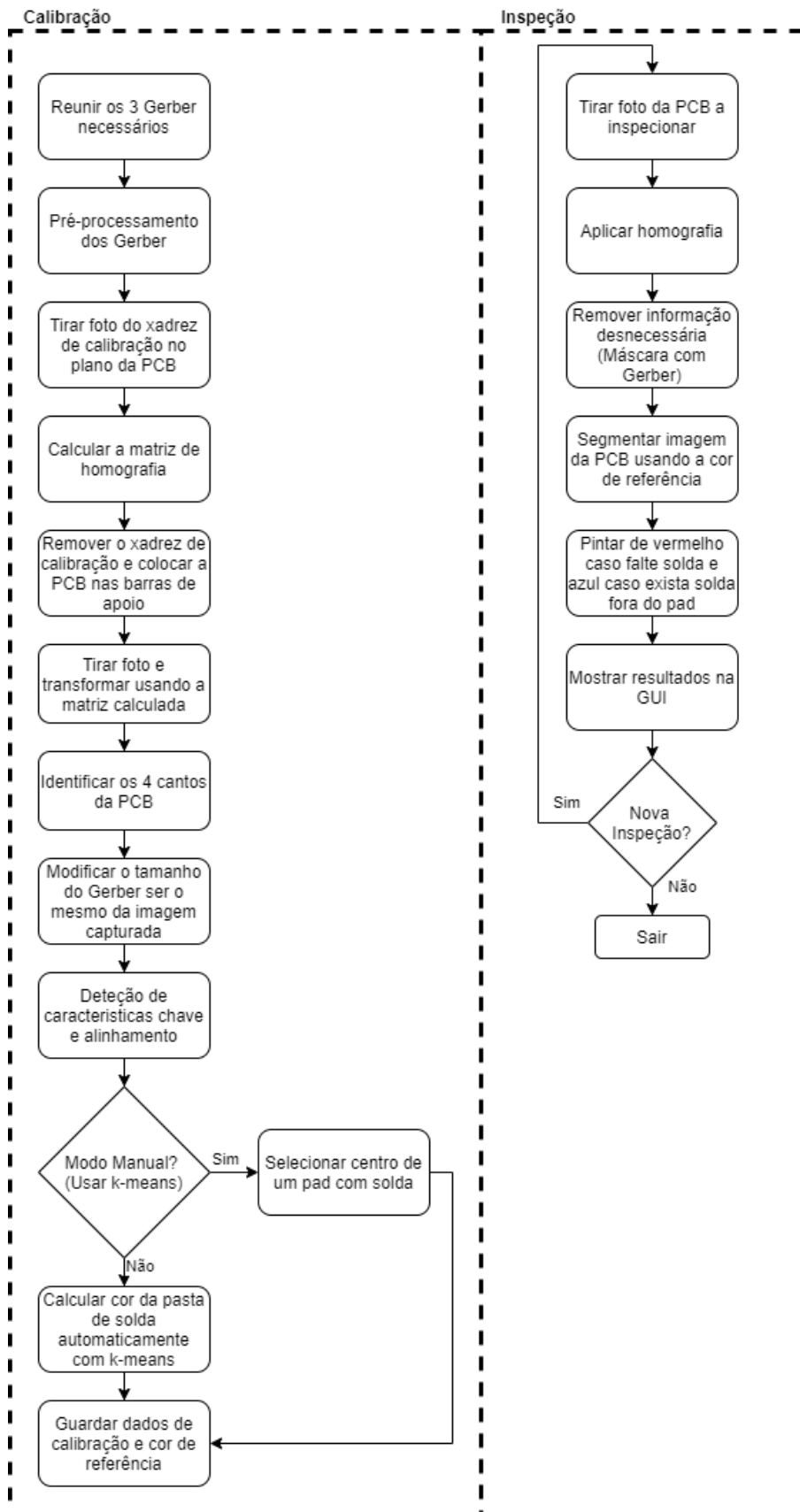


Figura 3.8: Diagrama de blocos da etapa de calibração e inspeção.



Figura 3.9: Exemplo das posições do xadrez de referência para a calibração. É necessário capturar várias imagens deste padrão em várias posições diferentes.

- **Pré-processamento dos Gerber**

Seguidamente é feito um pré-processamento dos ficheiros Gerber para garantir que todos têm o mesmo tamanho, assim como calcular e guardar vários valores que são utilizados ao longo da calibração como é o caso de tamanhos.

- **Cálculo da matriz de homografia**

Depois de efetuados todos os passos de pré-processamento, é possível realizar a primeira transformação. Esta transformação tem como objetivo calcular o plano onde está a PCB a inspecionar. Para isso é colocado um xadrez de calibração no mesmo plano da PCB. Dado que os pontos do xadrez de calibração são conhecidos, é calculada a posição desses mesmos pontos na imagem, como é apresentado na Figura 3.10.

É possível estabelecer uma relação de homografia entre os dois conjuntos de pontos através da matriz de homografia definida pela Eq. 3.1,

$$s \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = H \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad (3.1)$$

O ponto (x, y) corresponde ao ponto na imagem adquirida enquanto que (x', y') pertence ao plano de calibração (referência). O parâmetro s representa um fator de escala arbitrário. A homografia H é estimada através de um conjunto de pontos no plano de referência e na imagem real através da minimização do erro de projeção dado pela Eq. 3.2,

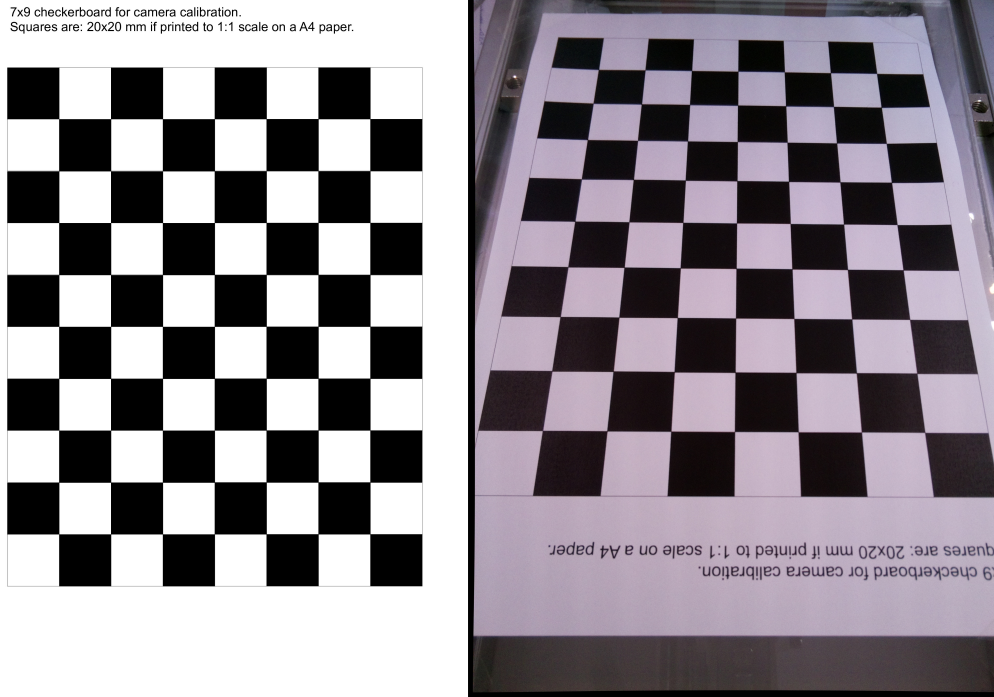


Figura 3.10: Relação entre o xadrez de referência e o xadrez no plano da PCB. É calculada a matriz de transformação entre os pontos da imagem da direita e a imagem da esquerda, que será o destino.

$$\sum_i \left(x'_i - \frac{h_{11}x_i + h_{12}y_i + h_{13}}{h_{31}x_i + h_{32}y_i + h_{33}} \right)^2 + \left(y'_i - \frac{h_{21}x_i + h_{22}y_i + h_{23}}{h_{31}x_i + h_{32}y_i + h_{33}} \right)^2 \quad (3.2)$$

onde o subscrito i denota número do ponto do conjunto de calibração.

Na Figura 3.11 é apresentada a transformação descrita, a imagem na esquerda é a capturada pela câmara, a imagem da direita é obtida através da aplicação da matriz de homografia calculada anteriormente. Cada pixel da imagem original é reprojetoado, usando a homografia, para o plano da PCB, contendo as dimensões reais, tornando possível a sua análise.

A imagem que se pretende analisar é sempre transformada de modo a ser comparada com a PCB de referência (proveniente do ficheiro Gerber). Desta maneira, a orientação da câmara em relação à PCB não é relevante, sendo possível ser ajustada sem que comprometa a calibração, apesar de que será necessário realizar uma nova calibração caso a orientação seja alterada.

- **Identificação da região de interesse**

Depois de efetuada a homografia, o próximo passo é a identificação da região de interesse, neste caso, onde está localizada a PCB. A PCB apresenta uma cor com bastante intensidade e, ao utilizar o espaço de cores HSV, a componente S (Saturação) evidencia

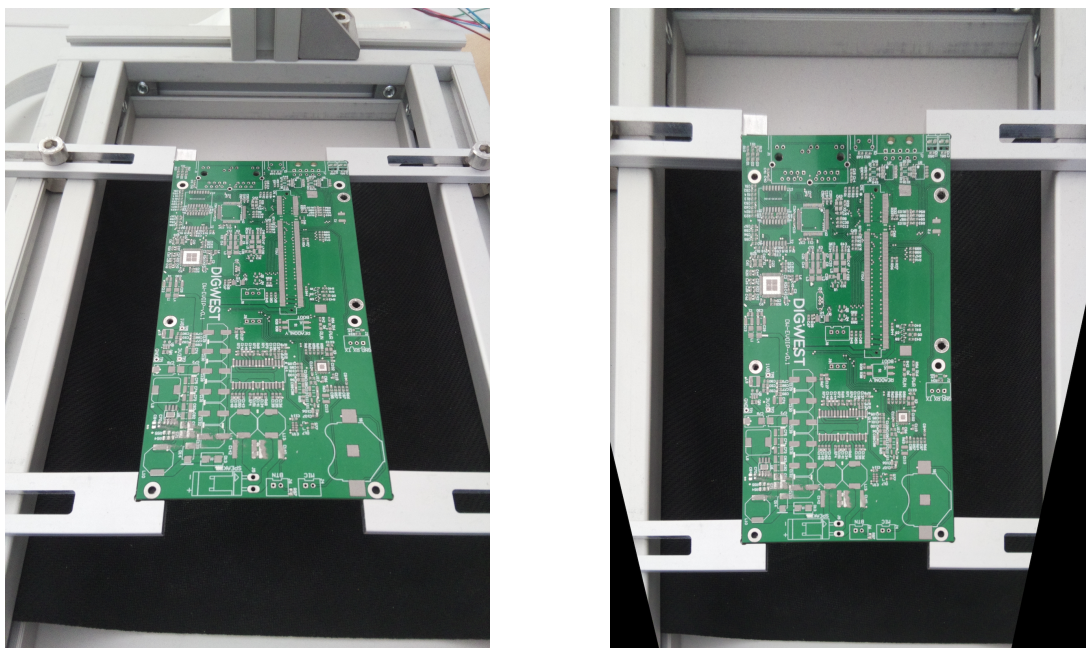


Figura 3.11: Imagem adquirida (esquerda) e imagem transformada (direita), pronta para ser processada, depois da transformação homográfica.

o contorno da PCB. Ao binarizar essa componente é possível fazer a detecção do contorno da PCB, como se pode observar na Figura 3.12 a vermelho.

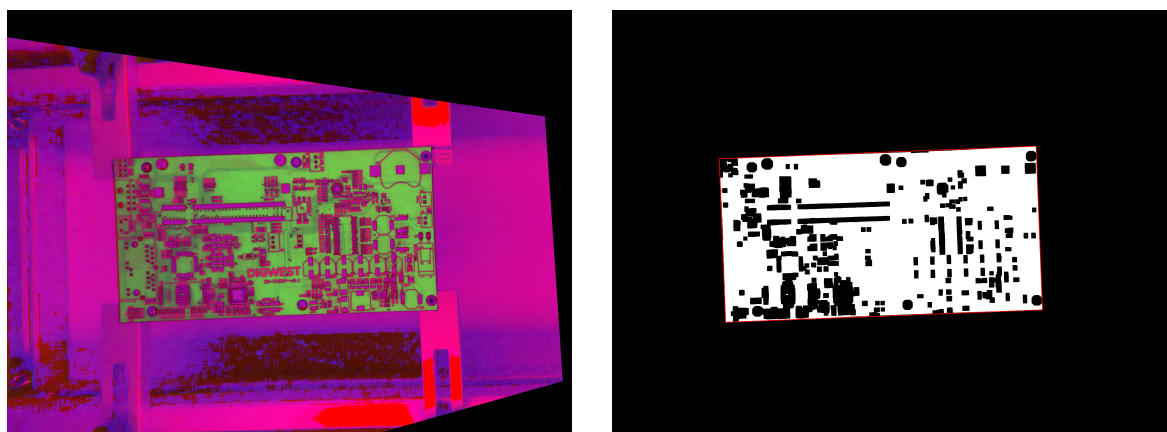


Figura 3.12: Identificação da região de interesse no espaço de cores HSV (esquerda) e componente S (direita). O contorno detetado é apresentado na imagem da direita a vermelho.

Apesar de ser evidente para o sistema visual humano qual a localização da PCB, para o computador, essa localização não é tão evidente, sendo necessário especificar qual é a região de interesse. Como a maior parte das PCB são retangulares, assumiu-se essa forma, sendo usada uma função da biblioteca OpenCV que encontra um quadrado fechado utilizando o mínimo de área possível. Esta função permite que o retângulo encontrado esteja rodado, o que facilita a deteção, uma vez que as barras de suporte da PCB não

necessitam de estar em simetria com as câmaras.

- **Identificação da região de interesse**

A detecção de erros de pasta de solda assenta na comparação entre a imagem e a referência do Gerber, pelo que a determinação do alinhamento entre os dois é essencial.

De modo a corrigir e garantir um melhor alinhamento, são usados dois algoritmos para detecção e correspondência de pontos chave, o detetor de pontos chave ORB [48] e o algoritmo de correspondência BFMatcher [49]. No artigo [49] são comparados vários detetores e vários algoritmos de correspondência comparando não só o número de pontos chave detetados, como quantos desses são correspondidos e o tempo de execução desse mesmo algoritmo. Este conjunto de algoritmos apresenta bons resultados temporais, ou seja, o seu tempo de execução é curto em comparação com outros conjuntos de algoritmos. Para além disso, o algoritmo está implementado em código *open source*.

Numa PCB podem existir centenas de *pads* iguais com vizinhanças semelhantes, o que pode causar falsos positivos na correspondência de pontos chave. Sabendo que isso pode ser um problema, devido às características da PCB, todas as correspondências são filtradas de forma a minimizar correspondências erradas. Como já referido anteriormente, a PCB é isolada de modo a ser comparada com o Gerber, ficando numa posição semelhante, servindo o processo de *features* para fazer um alinhamento detalhado. São analisadas todas as correspondências e as mesmas só são guardadas caso ambos os pontos chave estejam na mesma região. Na Figura 3.13 é apresentado um exemplo de correspondências, onde é possível observar que várias correspondências são feitas na diagonal, quando todas as correspondências deveriam ser feitas paralelamente, para a mesma região. Desta maneira é possível remover falsas correspondências, melhorando posteriormente a transformação que irá sobrepor a PCB real com o Gerber de referência.

- **Deteção da cor da pasta de solda**

Após as transformações, todos os *pads* da PCB real encontram-se na mesma posição na imagem e no Gerber de referência. É possível agora passar à etapa final da calibração. A segmentação das áreas de solda e metálico é feita recorrendo à cor apresentada na imagem. Esta cor pode ser definida manualmente pelo utilizador ou de forma automática. Quando o utilizador pretende escolher a cor manualmente, é apresentada a imagem da PCB real onde o utilizador escolhe (prime com o rato) na zona onde está o *pad*. No seguimento, é calculada uma média da cor dos pixels envolventes, sendo esse valor posteriormente usado como referência. Quando o utilizador opta pela seleção automática de cor, o algoritmo *k-means* é utilizado e a cor dos *pads* é calculada automaticamente. Para isso é utilizado o Gerber com a informação dos *pads* e, ao usar esse mesmo Gerber como máscara, o

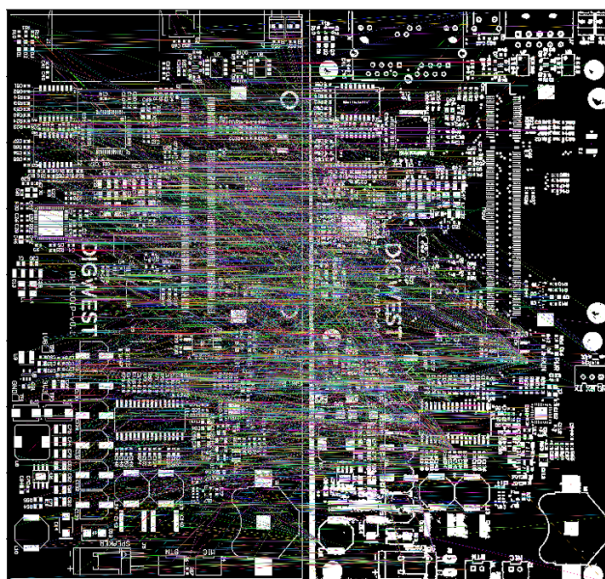


Figura 3.13: Correspondência de pontos chave com várias correspondências incorretas. É necessário filtrar as correspondências para garantir um melhor alinhamento.

resultado final é a imagem real com apenas os *pads* visíveis. Todo o remanescente da PCB (legendas, caminhos, buracos, etc) é descartado ficando cada pixel com o valor 0, ou seja, preto. O algoritmo *k-means* divide a imagem em várias cores, neste caso duas, o preto e a cor da solda.

Depois de obtidas todas as informações necessárias, os dados são gravados e, sempre que seja necessário realizar uma inspeção, os dados da calibração estão disponíveis.

3.2.2 Inspeção

Depois do processo de calibração, o sistema está pronto a realizar inspeções de pasta de solda. Como descrito anteriormente, a imagem real da PCB é alinhada com o Gerber de referência usando os parâmetros estimados na etapa de calibração. A cor de referência da pasta de solda é usada para identificar as áreas onde existe pasta de solda e áreas com as zonas metálicas dos *pads* expostos. Ao comparar a imagem real com a referência, as áreas onde falta pasta de solda nos *pads* e as áreas com excesso de solda são identificadas. Note-se, que apenas é necessário realizar a calibração uma vez para cada tipo de placa, com uma placa onde não existem quaisquer erros e depois é feita a inspeção repetidamente para todas as placas.

Caso a calibração não seja efetuada corretamente, a PCB pode conter um deslocamento em relação ao Gerber, o que resulta numa deteção de solda fora do *pad*, a azul, e falta de solda na extremidade oposta, a vermelho, como é possível observar na Figura 3.14.

Este é um problema de robustez do algoritmo, uma vez que existe a possibilidade de

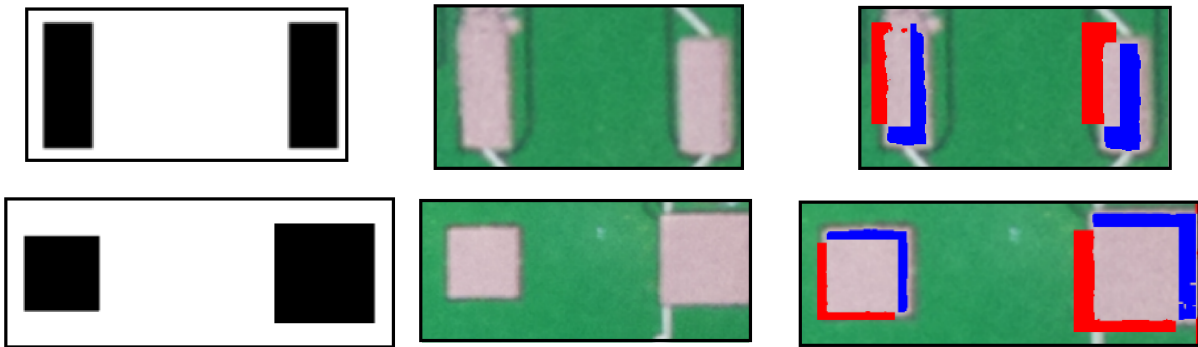


Figura 3.14: Detecção de excesso (azul) e falta (vermelho) de solda. À esquerda a imagem Gerber; Ao centro a imagem dos *pads* a analisar; e à direita a detecção de pasta de solda.

o alinhamento não ser calculado corretamente e resultar neste desalinhamento. Este desalinhamento ocorre em zonas da PCB menos populosas, ou seja, com menos informação, onde é detetado um número inferior de características chave.

Em vez do alinhamento e comparação pixel a pixel, cada *pad* é segmentado e realinhado com a referência, de forma a corrigir pequenos desalinhamentos. Na Figura 3.15 é possível observar o diagrama de blocos deste algoritmo.

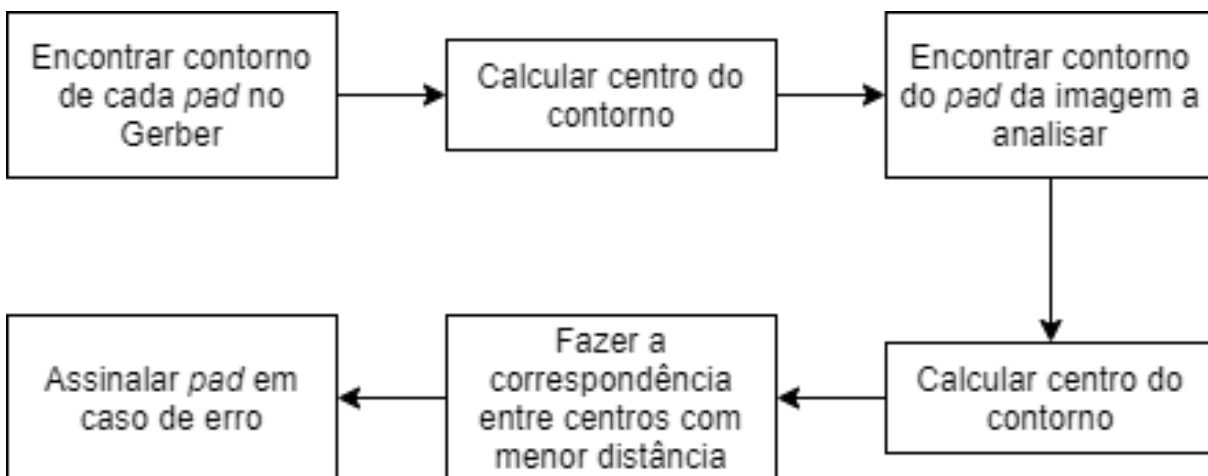


Figura 3.15: Diagrama de blocos do algoritmo de detecção *pad* a *pad*.

É necessário identificar todos os contornos de *pads* no Gerber, bem como a sua localização, que será usado como referência. É necessário realizar a mesma operação para a imagem a analisar, neste caso podem ser encontrados mais ou menos contornos de *pads* uma vez que podem não ser detetados todos os *pads*.

Como cada *pad* está sobreposto ou muito próximo, para efetuar a comparação entre Gerber e PCB, apenas é necessário fazer a correspondência entre o *pad* da PCB que está mais próximo da localização do *pad* no Gerber. Desta forma o Gerber é sempre comparado com o *pad* correspondente na imagem original, eliminando assim o problema da não sobreposição entre *pads*. Caso as dimensões do *pad* do Gerber sejam muito diferentes em

relação ao *pad* ao qual foi feita a correspondência, significa que o *pad* tem falta de solda, caso seja muito menor, ou excesso de solda, caso seja muito maior.

Também são assinalados *pads* onde são feitas múltiplas correspondências. É possível que ocorra esta situação quando existe uma falha ou onde existe um *pad* e são detetados dois.

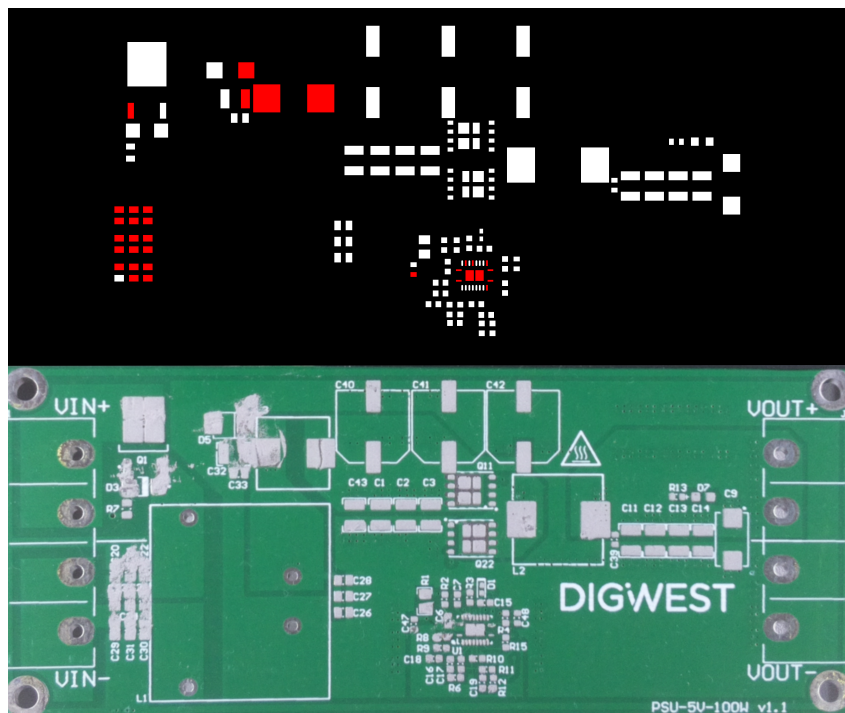


Figura 3.16: Exemplo do algoritmo de detecção *pad a pad*. Gerber (em cima) e imagem da PCB a analisar (em baixo). Neste caso, os *pads* que contêm erro são assinalados a vermelho.

Na Figura 3.16 é apresentado o resultado da solução referida, neste caso, a imagem que é apresentada ao utilizador é mais "limpa" e é mais fácil para o utilizador descobrir se existe erro ou não na PCB. Nesta imagem, os *pads* a branco significam que não existe erro nesse *pad*. Os *pads* a vermelho significam que existe um erro detetado nesse *pad*. Na Figura 3.17 é possível observar com mais detalhe a detecção dos *pads* que estão com erro. Neste caso, os *pads* a vermelho estão esborratados, criando curto circuito com os *pads* vizinhos.

Na Figura 3.18 é apresentado outro exemplo desta detecção, neste caso para um painel que contém várias PCB. Existem vários falsos positivos mas todos os *pads* que não contêm solda são assinalados corretamente.

Apesar disto, este método não mostra solda fora dos *pads*, caso exista uma zona onde a solda se encontra fora do *pad* essa mesma zona não é detetada nem apresentada. Este método pode ser utilizado em conjunto com a inspeção e, ao usar os dois métodos, é possível detetar a falta de solda com ambos os métodos e a solda fora dos *pads* com o

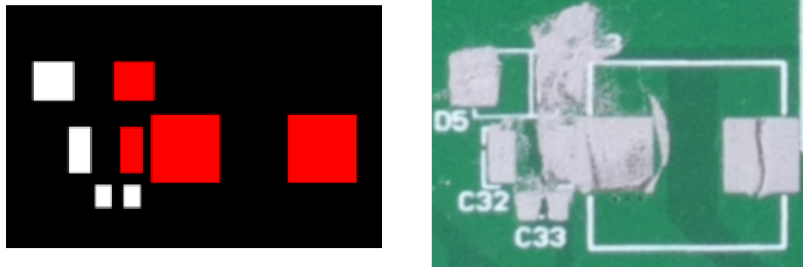


Figura 3.17: Detalhe da inspeção *pad a pad*, resultado da inspeção à esquerda e imagem capturada à direita.

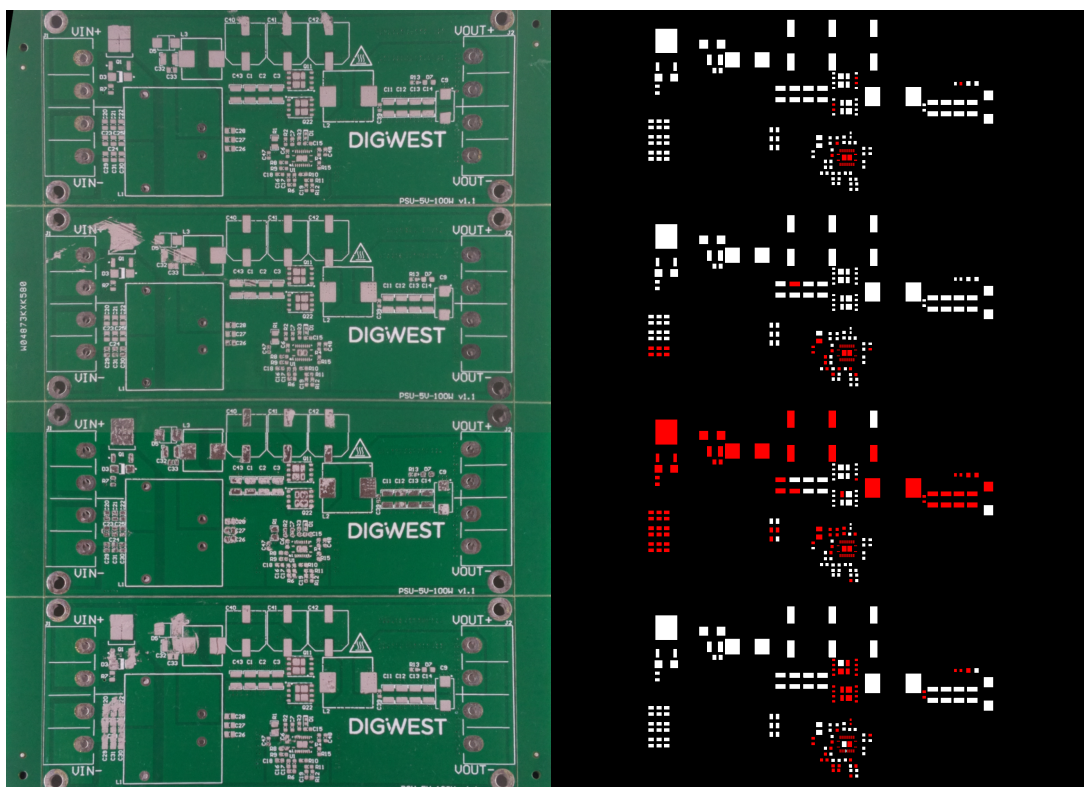


Figura 3.18: Detecção *pad a pad*. Imagem da PCB a analisar (à esquerda) e Gerber com *pads* que contêm erro a vermelho (à direita).

método original.

3.2.3 *Graphical User Interface (GUI)*

Foi desenvolvida uma interface gráfica (*graphical user interface*), para que o utilizador possa controlar de um modo fácil o sistema. Esta interface permite realizar o processo de calibração e visualizar a inspeção.

Os dados de calibração de vários tipos de PCB são gravados para que possam ser utilizados na próxima inspeção. Depois de cada inspeção são apresentadas várias imagens ao utilizador, a imagem da PCB original, uma imagem onde estão desenhados os defeitos detetados e uma imagem com a acumulação de erros por zona. Todos os dados em relação à inspeção são guardados para que possam ser utilizados no futuro.

Na Figura 3.19 é apresentado o menu de calibração da GUI onde são solicitados ao utilizador todos os parâmetros necessários para executar a calibração. Também através deste menu é possível adquirir as imagens necessárias do xadrez de calibração e da PCB. Contém também uma janela onde é apresentada uma breve explicação de todos os parâmetros e todas as imagens que são necessárias para realizar a inspeção. Também disponibiliza uma caixa de texto que permite ao utilizador escrever observações ou detalhar com mais pormenor a calibração que está a ser realizada e uma janela de pré-visualização, de modo a confirmar que todas as imagens são bem capturadas.

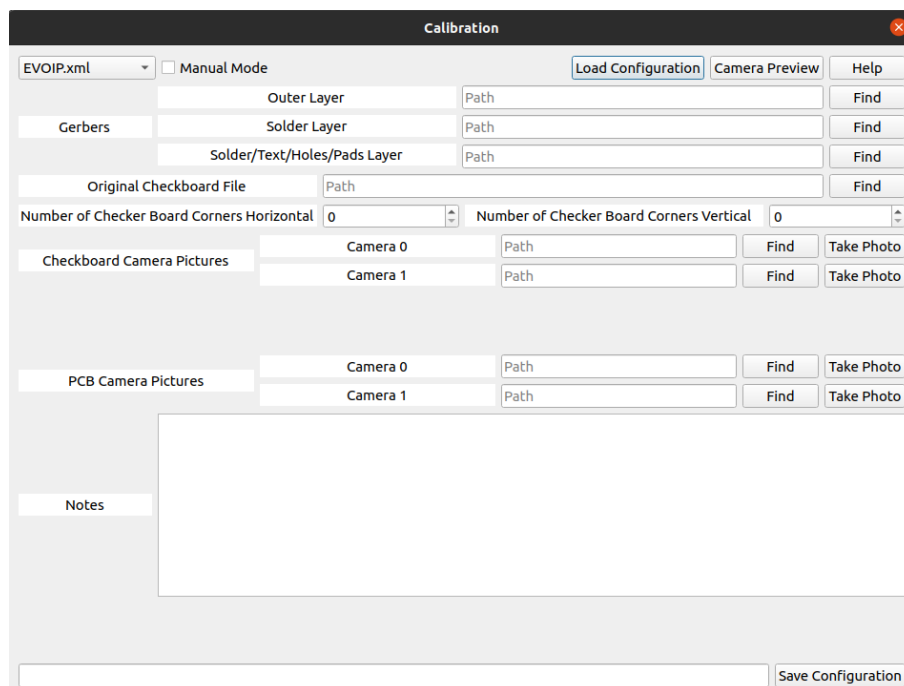


Figura 3.19: Interface gráfica da janela de calibração.

Na Figura 3.20 é apresentado o menu principal da GUI, onde é possível seleccionar qual

a configuração da PCB que irá ser analisada. São apresentadas 3 imagens ao utilizador, a imagem original da PCB, a imagem com os erros detetados e um *heatmap* onde são apresentadas zonas com possíveis erros.

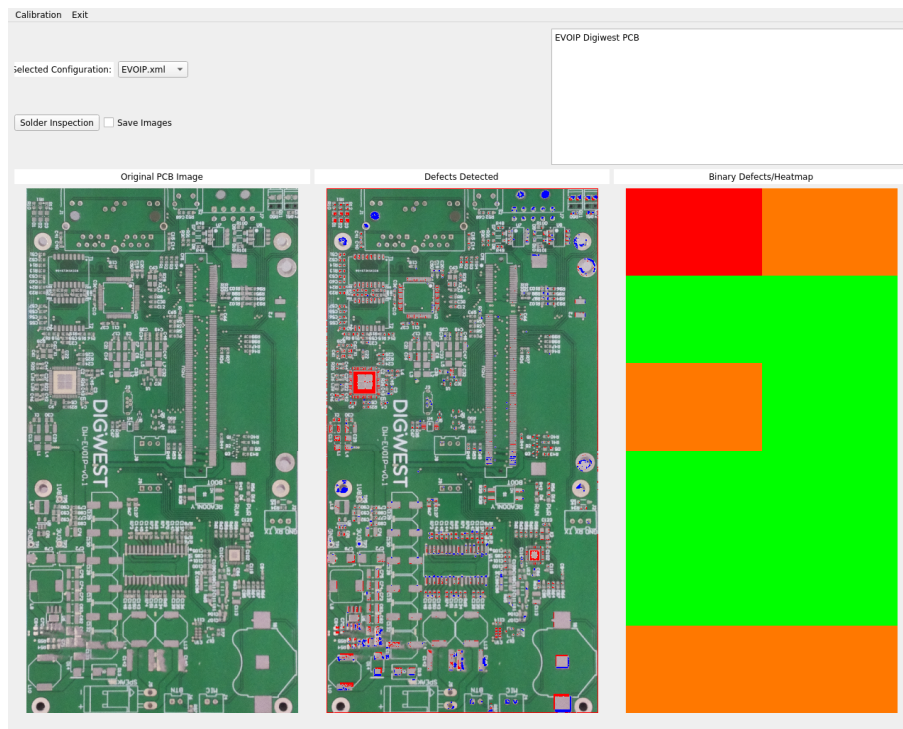


Figura 3.20: Interface Gráfica do sistema de inspeção.

É a partir desta janela que é possível abrir a janela de calibração, para que a mesma possa ser efetuada, bem como a janela de criação do ficheiro Gerber para matrizes de PCB, apresentado no capítulo 4.4.

Para realizar a inspeção o operador necessita de especificar a configuração referente à calibração na interface gráfica do utilizador. Todas as calibrações podem ser guardadas com nomes diferentes sendo apenas necessário escolher a configuração pretendida. A informação da inspeção é apresentada ao operador e o mesmo pode depois efetuar a inspeção visual das áreas afetadas e determinar se existe realmente um problema, que pode ou não afetar o funcionamento da PCB.

Na lista 3.1 são listados todas as funcionalidades disponíveis na GUI.

Lista 3.1: Funcionalidades disponíveis na GUI.

- Calibração
 - Capturar imagem do xadrez de calibração;
 - Capturar imagem da PCB;
 - Calcular calibração;
 - Guardar parâmetros de calibração;
 - Pré-visualização do plano de trabalho;
 - Janela de ajuda para o operador.
- Inspeção
 - Escolher configuração de calibração;
 - Realizar inspeção.
- Criação de painel de PCB

Capítulo 4

Resultados

Neste capítulo são apresentados resultados do protótipo em funcionamento e são feitas várias comparações entre algoritmos que foram testados. É ainda feita uma comparação do tempo de execução de vários algoritmos no Raspberry Pi e num computador com processador Intel i7.

4.1 Tempos de execução

Um dos fatores que é necessário ter em atenção é o tempo de processamento do algoritmo, uma vez que o mesmo irá funcionar num Raspberry Pi. É importante que os tempos de execução sejam compatíveis com os tempos de produção das placas num ambiente real. No processo de produção das placas, apenas uma placa pode ser barrada com pasta de solda de cada vez. Enquanto o operador está a barrar uma placa com pasta de solda, uma placa pode estar a ser inspeccionada em paralelo. Para isso é necessário que o tempo de inspeção seja inferior ao tempo de deposição de solda, que é cerca de 30 segundos no menor caso, para que o operador não precise de esperar para colocar uma nova PCB na inspeção. De modo a testar o tempo de execução dos algoritmos, os mesmos foram testados num Raspberry Pi Compute Module 3+, com um processador ARMv8 1.2GHz e 1GB de RAM, referido no capítulo 3, e num PC com um processador Intel i7-3632QM a 2.2GHz e 8Gb de RAM. Desta maneira é possível observar as diferenças de processamento e os compromissos que devem ser feitos para que possam ser feitas inspeções sucessivas sem que o utilizador necessite de esperar pelo fim da inspeção.

Foram testados vários algoritmos utilizados ao longo do processo de inspeção, sempre com as mesmas imagens de modo a que seja possível fazer uma comparação entre o tempo de processamento do Raspberry e de um PC. Na Tabela 4.1 são apresentados os tempos de execução de vários algoritmos, assim como os tempos totais de inspeção e calibração. Os algoritmos apresentados apresentam dois valores distintos, uma vez que são utilizadas

duas câmaras e duas imagens diferentes.

Algoritmo	Raspberry		PC	
	Câmara 0	Câmara 1	Câmara 0	Câmara 1
Encontrar cantos no xadrez	4,660	4,299	0,244	0,242
Encontrar relação de homografia	0,076	0,066	0,002	0,002
Encontrar contorno da PCB (ROI)	0,387	0,410	0,028	0,031
Encontrar pontos chave (<i>features</i>)	32,621	32,588	1,828	1,786
Algoritmo Kmeans	6,680		0,038	
Tempo total de calibração	125,124		10,669	
Tempo total de inspeção	20,143		1,988	

Tabela 4.1: Tempos de Execução médios em segundos (em 5 inspeções).

É possível observar a diferença substancial no tempo de execução do mesmo algoritmo em diferentes dispositivos. Esta diferença deve-se à capacidade de processamento de ambos os dispositivos dado que enquanto o Raspberry tem apenas 4 *cores* (núcleos) de processamento e 1Gb de RAM, o PC tem o dobro dos cores, com uma velocidade superior, e 8 vezes mais memória RAM. Estes fatores levam a que em certos algoritmos, o processamento seja 10 vezes mais rápido num PC que no Raspberry.

Apesar desta diferença, o tempo total de inspeção é aceitável uma vez que é o tempo necessário para que seja colocada a pasta de solda numa nova PCB e, enquanto o utilizador está a fazer essa deposição de pasta de solda, pode estar a ser feita a inspeção à PCB anterior, tornando o processo de deposição de pasta de solda mais rápido.

Existem também diferenças no tempo de execução dos algoritmos entre câmaras. Esta diferença deve-se à quantidade de informação que é processada, ou seja, se uma câmara tiver um número superior de *pads*, o processamento dessa imagem será maior relativamente a uma imagem com um menor número de *pads*. Quanto maior for o número de *pads*, maior será o tempo de execução.

O tempo de execução do algoritmo de calibração é bastante elevado em relação ao de inspeção mas, como a calibração apenas é feita uma vez por tipo de PCB, esse tempo não é muito relevante.

4.2 Diferença nos detetores de *features*

No capítulo 3 é referido uma publicação [49] onde são feitas várias comparações entre detetores e combinadores de *features* utilizadas para fazer a correspondência entre pontos chave. Nesse artigo é referido que a combinação do detetor BRISK com o combinador *BFMatcher* é a combinação ideal, uma vez que existe uma boa relação entre *features* combinados e tempo de execução. De acordo com este estudo, embora não tenha um

desempenho na detecção tão bom, também pode ser utilizada a combinação do detetor ORB e do combinador *BFMatcher*.

Foram testadas essas duas combinações de modo a perceber a diferença entre as mesmas para o caso em estudo. Foram obtidos os tempos de execução dos algoritmos, assim como o número de *features* encontrados e combinados. O tempo de execução apresentado é o tempo total do algoritmo para efetuar a detecção da orientação da PCB e o alinhamento final dos *pads* da imagem em relação ao Gerber. São efetuadas 4 detecções e combinações de *features*, uma para cada orientação e só são guardados *features* que se encontram na mesma coordenada ou próxima em relação ao Gerber.

	Algoritmo	Número de <i>Features</i>	Tempo de execução
Raspberry	ORB	100	64s
	BRISK	2100	6000s
PC	ORB	100	3,5s
	BRISK	2100	300s

Tabela 4.2: Comparação dos tempos de execução entre os métodos ORB e BRISK.

Na tabela 4.2 é possível observar o número de *features* para ambos os algoritmos bem como o tempo de execução de ambos para os diferentes dispositivos. O número de *features* detetados é o mesmo para os dois dispositivos uma vez que a imagem é a mesma. Em relação ao tempo de execução já não é possível concluir o mesmo uma vez que a diferença é considerável. Para ambos os dispositivos, o algoritmo BRISK demora mais tempo que o algoritmo ORB mas, esse tempo de execução resulta numa quantidade de *features* muito maior.

Na tabela 4.1, os tempos de calibração medidos utilizaram o algoritmo ORB para a detecção de *features*, caso fosse utilizado o algoritmo BRISK, para ambos os dispositivos, o tempo total de calibração seria muito maior que o tempo observado, em vez de demorar 2 minutos no Raspberry (125 segundos), demoraria cerca de 1 hora e 40 minutos (6100 segundos), o que não seria prático em ambiente industrial. No PC também este tempo total de calibração seria afetado, demorando cerca de 5 minutos (300 segundos), em vez dos 10 segundos medidos com o algoritmo ORB. Na Figura 4.1 pode observar-se a diferença no número de *features* que são encontrados e combinados.

Apesar de não serem desenhadas todas as combinações de *features* encontradas, a diferença no número de *features* entre as duas imagens é considerável. Embora um maior número de *features* tenham sido encontradas e combinadas em relação ao algoritmo ORB, na PCB em estudo, o aumento do número de pontos combinados não afetou o resultado da transformação sendo que, para ambos os casos, o resultado é idêntico. Posto isto é preferencial a utilização de um algoritmo mais rápido uma vez que o tempo de execução é um fator importante e os resultados finais são semelhantes.

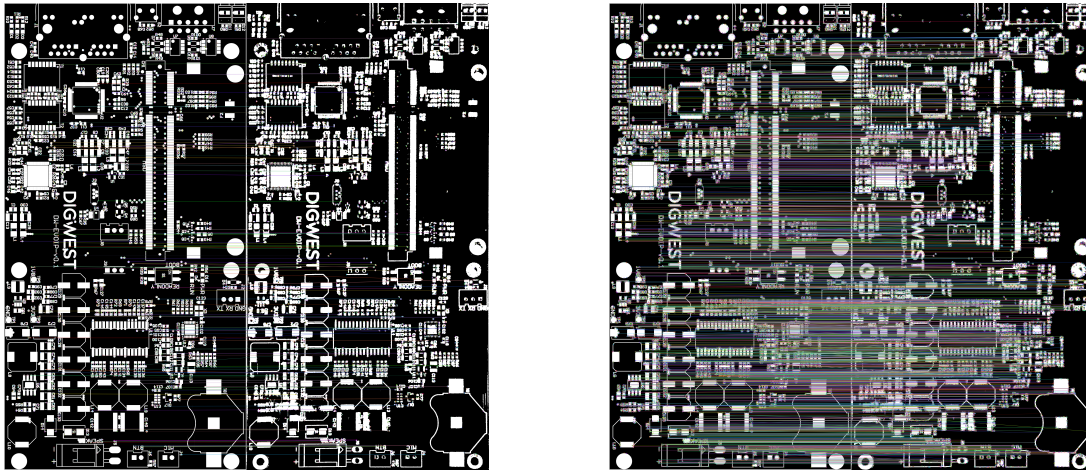


Figura 4.1: Combinações de *features* com ORB (esquerda) e BRISK (direita). Teste ao tempo de execução da correspondência de *features* detectadas em imagens idênticas, com eliminação de falsas correspondências.

4.3 Detecção de falta de solda e solda em excesso

Este sistema tem o objetivo de apoiar a tarefa da inspeção de pasta de solda através da identificação de áreas na PCB onde falta ou existe pasta de solda em excesso (ex. solda fora dos *pads*). Esta informação é apresentada ao operador através da GUI, sendo este que determina se a PCB deve ser descartada ou não, ou se existe um problema no processo de deposição.

Na Figura 4.2 é mostra-se a imagem que é apresentada na GUI ao utilizador. A imagem da PCB é marcada sobre as falhas com duas cores diferentes caso sejam detetados dois tipos de erros. No caso da existência de solda fora dos *pads* (ou excesso de solda), essa zona é pintada a azul. Caso exista um défice de solda num *pad*, essa zona é pintada a vermelho. Desta maneira o utilizador consegue facilmente identificar as zonas e os *pads* que contém erros.

Na Figura 4.3 é apresentado um caso que ocorreu numa das PCB em estudo onde existe uma diferença entre o Gerber e o *stencil* usado para aplicar a solda. Devido a essa diferença, uma vez que a deteção da pasta de solda utiliza o *Gerber* como referência, é detetado um erro que é assinalado a vermelho, indicando falta de pasta de solda no *pad*.

Segundo o Gerber, imagens da esquerda na Figura 4.3, o *pad* central deveria estar coberto com solda. Nas imagens centrais é possível ver que não é isso que acontece uma vez que o *stencil* tem um design diferente, composto por quatro quadrados pequenos centrados num dos *pads* e um quadrado centrado no caso do outro *pad*. Devido a esta diferença, é pintado a vermelho toda a área onde não existe solda (imagens da direita na Figura 4.3).

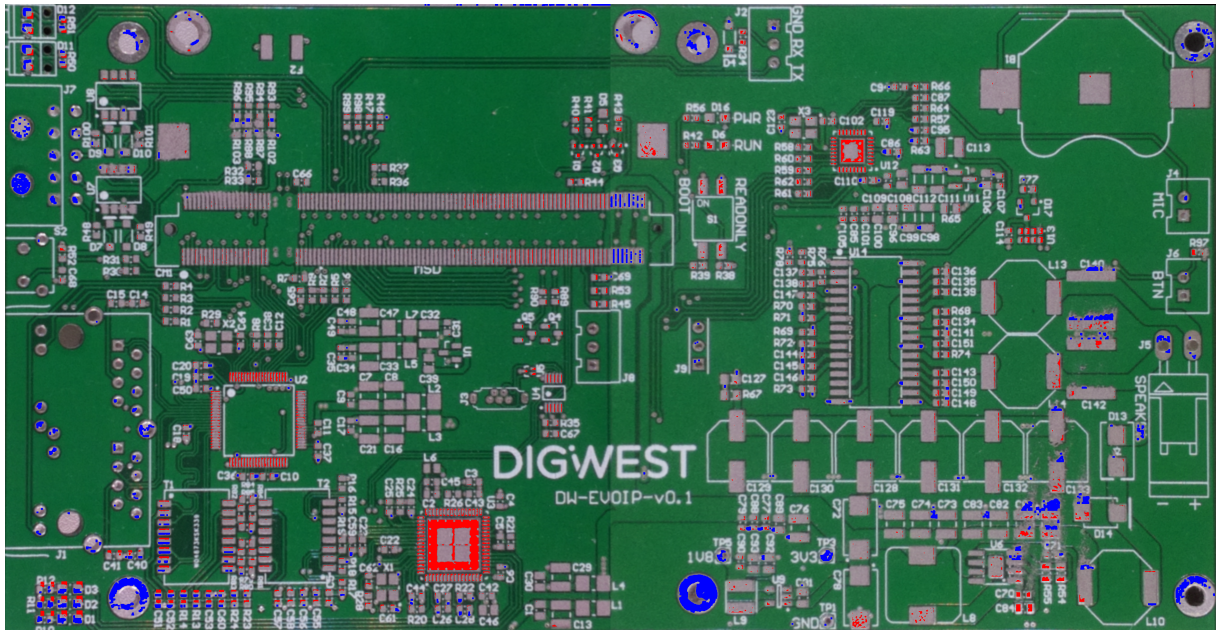


Figura 4.2: Erros detetados na inspeção: Excesso de solda (azul) e defice de solda (vermelho).

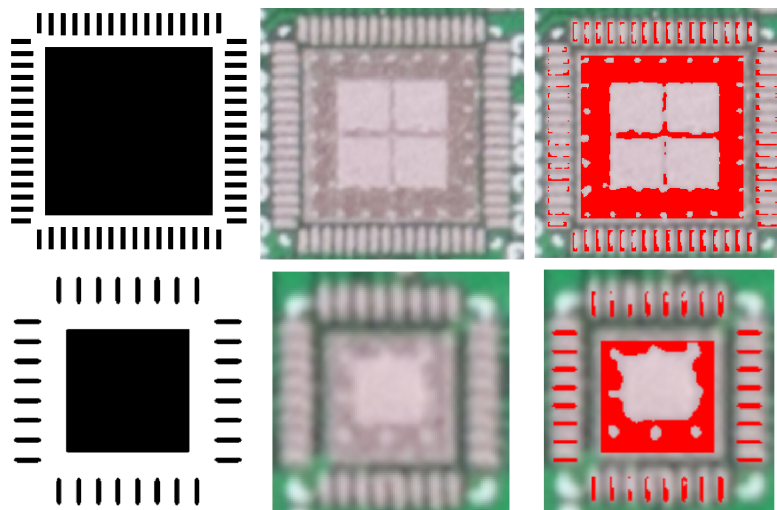


Figura 4.3: Erro detetado devido à diferença entre Gerber e *stencil*. Gerber de referência à esquerda, imagem do *pad* ao centro e erro detetado no *pad* à direita.

Outro exemplo da detecção de falta de solda é apresentado na Figura 4.4 onde não foi depositada solda em dois *pads* e alguns *pads* encontram-se com falta de solda.

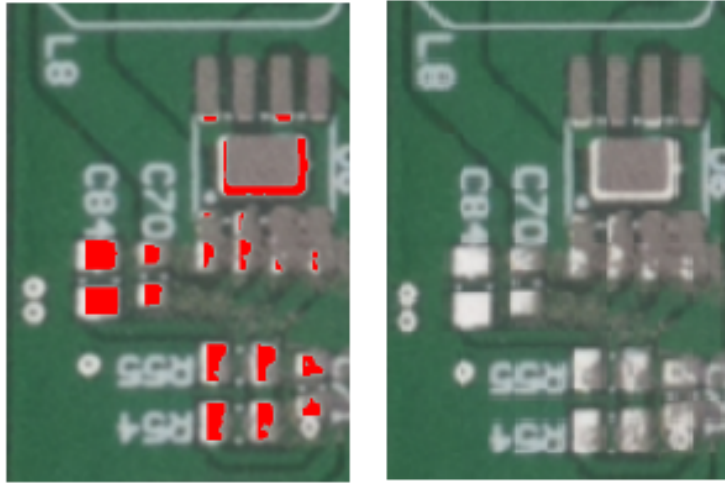


Figura 4.4: Erro detetado devido à falta de solda. É assinalado a vermelho a falta de solda em relação ao ficheiro Gerber.

Na Figura 4.5 é apresentado um exemplo não só da falta de solda, como da existência de solda em "excesso". Neste caso a pasta de solda foi espalhada fora do *pad* devido a um toque acidental na mesma. Nos *pads* centrais é visível não só a falta de solda como o excesso de solda que é espalhado para o lado direito.



Figura 4.5: Pasta de solda espalhada assinalada a azul quando é fora do *pad* e a vermelho quando existe falta de solda. Imagem do Gerber à esquerda, imagem do *pad* ao centro e deteção da falta e excesso de solda à direita.

4.4 Inspeção de Painel de PCB

De modo a agilizar o processo de deposição de pasta de solda e colocação de componentes, para grandes quantidades de PCB, um dos métodos utilizados na indústria é a utilização de várias PCB num único painel, ou seja, em vez de ser analisada apenas uma PCB é analisada uma matriz com várias PCB iguais.

O *stencil* que é utilizado na deposição de solda é fabricado pela mesma empresa que fabrica as PCB e na maior parte dos casos, a empresa que irá fazer a inspeção não tem acesso aos Gerber utilizados no fabrico do *stencil*, apenas têm disponível o ficheiro

Gerber original de cada PCB. Na Figura 4.6 é apresentado um exemplo do sucedido, onde o ficheiro Gerber disponível (esquerda) é diferente do ficheiro Gerber (direita) que seria necessário para realizar a inspeção da PCB. Para este caso, o painel a ser analisado contém 4 PCB que estão dispostas lado a lado.

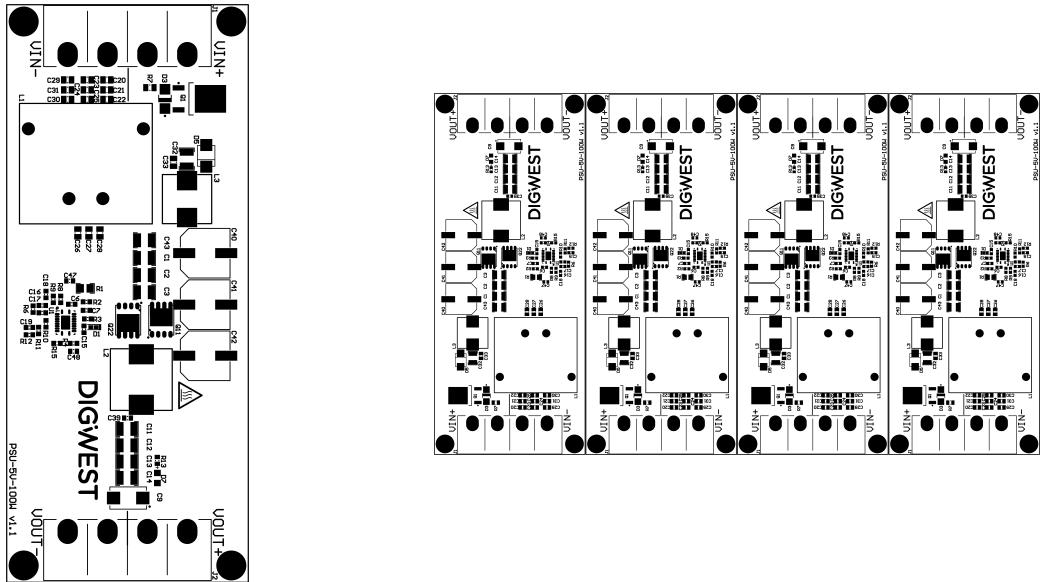


Figura 4.6: Gerber disponível (esquerda) e Gerber composto para painel (direita).

De modo a resolver este problema foi criada uma funcionalidade na aplicação, que através de uma janela na GUI seja efetuada a composição da imagem correspondente ao ficheiro Gerber necessário com várias PCB num painel. Na Figura 4.7 é apresentada a referida janela que necessita das 3 imagens referentes aos ficheiros Gerber necessários para a realização da inspeção, o *outline*, o Gerber com a solda e um Gerber com vários detalhes.

É necessário também definir os tamanhos do Gerber, do painel e do espaçamento ente PCB, em milímetros. Estas medidas permitem definir a composição de PCB em painel, nomeadamente o espaçamento entre PCB ou espaçamento entre a aresta e a PCB.

Gerber to Matrix Conversion

Outer Gerber

Solder Gerber

Full Detail Gerber

Number of Lines (Gerber)

Number of Columns (Gerber)

Horizontal spacing between Gerbers Vertical spacing between Gerbers

Board Width (mm)

Board Height (mm)

Gerber Width (mm)

Gerber Height (mm)

Figura 4.7: GUI de montagem do Gerber final.

Capítulo 5

Conclusões e Trabalho Futuro

Este trabalho, apresenta um sistema de inspeção visual que é capaz de detetar defeitos causados pela deposição de pasta de solda em PCB. Este sistema foi desenvolvido, em parceria com a Digiwest - Wireless and Embedded Solutions Lda, utilizando componentes de baixo custo para ser usado numa linha de produção de PCB, servindo como auxílio ao operador dessa mesma linha. Foram realizados testes em ambiente real na empresa onde foi possível comprovar o funcionamento do sistema em ambiente industrial.

Para que o sistema possa realizar a inspeção de uma PCB é necessário realizar a calibração. A calibração consiste no cálculo de várias homografias, identificação de tamanhos e posições da PCB no plano de trabalho. Ao guardar várias informações acerca da PCB, torna possível que a inspeção tenha um tempo de execução curto, uma vez que muitas variáveis e informações auxiliares necessárias já foram calculadas anteriormente. É possível realizar todas essas operações numa interface gráfica que foi criada para facilitar a tarefa. Nesta interface é possível realizar a calibração, inspeções e criar matrizes de Gerber a partir de um Gerber singular. Esta interface apresenta ao operador o resultado da deteção de pasta de solda, onde são pintados os pixeis onde é detetada solda em excesso ou falta de solda no *pad*. O operador decide com base nessa deteção, se a PCB deve ser descartada ou se está apta para a próxima fase do processo de criação da PCB.

O sistema desenvolvido no Raspberry Pi é capaz de realizar uma inspeção entre 15 a 30 segundos. Uma vez que o processo de deposição de solda demora o mesmo tempo, caso a inspeção seja feita em simultâneo com o processo de deposição, o tempo é adequado para uso numa linha de produção.

Apesar do bom funcionamento em vários casos, o sistema não é totalmente robusto no processo de alinhamento da PCB com o Gerber. Por isso, foi proposto um novo sistema de realinhamento onde é feito um realinhamento *pad a pad*, o que resultou numa melhor deteção da falta de pasta de solda nos *pads*.

Este sistema, apesar de funcional, pode ser melhorado. Uma das melhorias seria a

otimização do tempo de execução dos algoritmos, poderia ser resolvido com um dispositivo com maior capacidade de processamento mas também envolveria um custo mais elevado.

Esta dissertação originou a publicação de um artigo de investigação, com o nome "*Automatic visual inspection system for solder paste deposition in PCBs*" [50] apresentado na "2021 Telecoms Conference (ConfTELE)".

Bibliografia

- [1] D. Burr. “Solder paste inspection: process control for defect reduction”. Em: *Proceedings International Test Conference 1997*. Nov. de 1997, pp. 1036–.
- [2] R. S. Clouthier. “The complete solder paste printing processes”. Em: *Surf. Mount Technol.* 3.1 (1999), pp. 6–8.
- [3] Xianyu Su e Wenjing Chen. “Fourier transform profilometry:: a review”. Em: *Optics and Lasers in Engineering* 35.5 (2001), pp. 263–284. ISSN: 0143-8166. DOI: [https://doi.org/10.1016/S0143-8166\(01\)00023-9](https://doi.org/10.1016/S0143-8166(01)00023-9). URL: <https://www.sciencedirect.com/science/article/pii/S0143816601000239>.
- [4] R. R. Lathrop. “Solder paste print qualification using laser triangulation”. Em: *IEEE Transactions on Components, Packaging, and Manufacturing Technology: Part C* 20.3 (1997), pp. 174–182.
- [5] Jef L. Horijon et al. “Optical system of an industrial 3D laser scanner for solder paste inspection”. Em: *Three-Dimensional and Unconventional Imaging for Industrial Inspection and Metrology*. Ed. por Michael R. Descour, Kevin G. Harding e Donald J. Svetkoff. Vol. 2599. International Society for Optics e Photonics. SPIE, 1996, pp. 162–170.
- [6] Lu Shenglin, Zhang Xianmin e Kuang Yongcong. “An integrated inspection method based on machine vision for solder paste depositing”. Em: *2007 IEEE International Conference on Control and Automation, ICCA*. Institute of Electrical e Electronics Engineers Inc., 2007, pp. 137–141. ISBN: 1424408180. DOI: 10.1109/ICCA.2007.4376334.
- [7] Jianbiao Pan et al. “Critical variables of solder paste stencil printing for micro-BGA and fine-pitch QFP”. Em: *IEEE Transactions on Electronics Packaging Manufacturing* 27.2 (abr. de 2004), pp. 125–132.
- [8] *Solder paste stencil*. Fev. de 2020. URL: https://www.itmconsulting.com/blog/wp-content/uploads/2015/03/solder/_paste/_stencil.jpg.
- [9] *Requirements for Soldering Pastes*. Standard. Bannockburn, Illinois: IPC, abr. de 1996.

- [10] Huihui Wu et al. “A real-time machine vision system for solder paste inspection”. Em: *2008 IEEE/ASME International Conference on Advanced Intelligent Mechatronics*. 2008, pp. 205–210.
- [11] *Solder paste inspection system*. Fev. de 2020. URL: <https://caltexsystems.com/products/solder-paste-inspection-systems/spi-200-3d/>.
- [12] Chao Zuo et al. “Phase shifting algorithms for fringe projection profilometry: A review”. Em: *Optics and Lasers in Engineering* 109 (2018), pp. 23–59. ISSN: 0143-8166. DOI: <https://doi.org/10.1016/j.optlaseng.2018.04.019>. URL: <https://www.sciencedirect.com/science/article/pii/S0143816618302203>.
- [13] Yongkai Yin et al. “A two-step phase-shifting algorithm dedicated to fringe projection profilometry”. Em: *Optics and Lasers in Engineering* 137 (2021), p. 106372. ISSN: 0143-8166. URL: <https://www.sciencedirect.com/science/article/pii/S0143816620308290>.
- [14] Grantham K.H. Pang e Ming-Hei Chu. “Automated inspection of solder paste by directional LED lighting”. Em: *2008 IEEE International Conference on Automation and Logistics*. 2008, pp. 232–237.
- [15] Ming-Hei Chu e Grantham Kwok-Hung Pang. “3D Shape Reconstruction based on Directional LED lights”. Em: *2019 IEEE 10th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)*. 2019, pp. 0436–0441.
- [16] C. Neubauer. “Intelligent X-ray inspection for quality control of solder joints”. Em: *IEEE Transactions on Components, Packaging, and Manufacturing Technology: Part C* 20.2 (abr. de 1997), pp. 111–120.
- [17] V. Sankaran, A. R. Kalukin e R. P. Kraft. “Improvements to X-ray laminography for automated inspection of solder joints”. Em: *IEEE Transactions on Components, Packaging, and Manufacturing Technology: Part C* 21.2 (abr. de 1998), pp. 148–154.
- [18] Young Jun Roh, Won Shik Park e Hyungsuck Cho. “Correcting image distortion in the X-ray digital tomosynthesis system for PCB solder joint inspection”. Em: *Image and Vision Computing* 21.12 (2003), pp. 1063–1075.
- [19] *What are the main defects that the Wellman X-ray inspection system can find on the BGA of PCB?* Mar. de 2020. URL: <https://www.wmxray.com/what-are-the-main-defects-that-the-wellman-x-ray-inspection-system-can-find-on-the-bga-of-pcb/>.
- [20] E.D. Megaw. “Factors affecting visual inspection accuracy”. Em: *Applied Ergonomics* 10.1 (1979), pp. 27–32.

- [21] D Martin. “A practical guide to machine vision lighting”. Em: *Advanced Illumination* (jan. de 2007), pp. 1–3.
- [22] *Guia prático sobre iluminação para visão de máquina – Parte I - NI*. URL: <https://www.ni.com/pt-pt/innovations/white-papers/12/a-practical-guide-to-machine-vision-lighting.html> (acedido em 26/08/2020).
- [23] *Difference Between CMOS & CCD and Why CMOS Sensors Are Preferred for Machine Vision Cameras*. URL: <https://www.phase1vision.com/blog/difference-between-cmos-and-ccd>.
- [24] POMEAS. *Difference Between CMOS & CCD and Why CMOS Sensors Are Preferred for Machine Vision Cameras*. URL: <https://www.phase1vision.com/blog/difference-between-cmos-and-ccd>.
- [25] Masayuki Sugawara, Seo-Young Choi e David Wood. “Ultra-High-Definition Television (Rec. ITU-R BT.2020): A Generational Leap in the Evolution of Television [Standards in a Nutshell]”. Em: *IEEE Signal Processing Magazine* 31.3 (2014), pp. 170–174. DOI: 10.1109/MSP.2014.2302331.
- [26] BT Series. “Colour gamut conversion from Recommendation ITU-R BT. 2020 to Recommendation ITU-R BT. 709”. Em: (2017).
- [27] Kai San Choi, Edmund Y. Lam e Kenneth K. Y. Wong. “Automatic source camera identification using the intrinsic lens radial distortion”. Em: *Opt. Express* 14.24 (nov. de 2006), pp. 11551–11565. DOI: 10.1364/OE.14.011551. URL: <http://www.osapublishing.org/oe/abstract.cfm?URI=oe-14-24-11551>.
- [28] C Brown Duane. “Close-range camera calibration”. Em: *Photogramm. Eng* 37.8 (1971), pp. 855–866.
- [29] Artur Nowakowski e Władysław Skarbek. “Analysis of Brown camera distortion model”. Em: *Photonics Applications in Astronomy, Communications, Industry, and High-Energy Physics Experiments 2013*. Vol. 8903. International Society for Optics e Photonics. 2013, p. 89030X.
- [30] Anubhav Agarwal, CV Jawahar e PJ Narayanan. “A survey of planar homography estimation techniques”. Em: *Centre for Visual Information Technology, Tech. Rep. IIT/TR/2005/12* (2005).
- [31] Nobuyuki Otsu. “A threshold selection method from gray-level histograms”. Em: *IEEE transactions on systems, man, and cybernetics* 9.1 (1979), pp. 62–66.
- [32] Maciej Szelag. “Evaluation of Cracking Patterns in Cement Composites-From Basics to Advances: A Review”. Em: *Materials* 13 (mai. de 2020), p. 2490. DOI: 10.3390/ma13112490.

- [33] Divyosmi Goswami. *K-mean: is he really mean?* Fev. de 2021. URL: <https://divyosmi.medium.com/k-mean-is-he-really-mean-b739758adc01>.
- [34] *K-Means Clustering in OpenCV*. URL: https://opencv24-python-tutorials.readthedocs.io/en/latest/py_tutorials/py_ml/py_kmeans/py_kmeans_opencv/py_kmeans_opencv.html.
- [35] John Canny. “A computational approach to edge detection”. Em: *IEEE Transactions on pattern analysis and machine intelligence* 6 (1986), pp. 679–698.
- [36] Wenshuo Gao et al. “An improved Sobel edge detection”. Em: *2010 3rd Int. Conf. on Computer Science and Information Technology*. Vol. 5. 2010, pp. 67–71.
- [37] *Example Canny Edge*. URL: https://boofcv.org/index.php?title=Example_Canny_Edge.
- [38] Christopher G Harris, Mike Stephens et al. “A combined corner and edge detector.” Em: *Alvey vision conference*. Vol. 15. 50. Citeseer. 1988, pp. 10–5244.
- [39] Jianbo Shi et al. “Good features to track”. Em: *1994 Proceedings of IEEE conference on computer vision and pattern recognition*. IEEE. 1994, pp. 593–600.
- [40] Piyush Khanna. *Harris Corner Detector-an Overview of the Original Paper*. Nov. de 2019. URL: <https://medium.com/swlh/harris-corner-detector-an-overview-of-the-original-paper-cf20c502ab0f>.
- [41] David G Lowe. “Distinctive image features from scale-invariant keypoints”. Em: *International journal of computer vision* 60.2 (2004), pp. 91–110.
- [42] *Feature Matching*. URL: https://opencv24-python-tutorials.readthedocs.io/en/stable/py_tutorials/py_feature2d/py_matcher/py_matcher.html.
- [43] Herbert Bay, Tinne Tuytelaars e Luc Van Gool. “Surf: Speeded up robust features”. Em: *European conference on computer vision*. 2006, pp. 404–417.
- [44] Raspberry Pi. *Buy a Compute Module 3*. URL: <https://www.raspberrypi.org/products/compute-module-3-plus/>.
- [45] Raspberry Pi. *Buy a Compute Module 1/3/3 IO Board*. URL: <https://www.raspberrypi.org/products/compute-module-io-board-v3/>.
- [46] Raspberry Pi. *Buy a Camera Module 2*. URL: <https://www.raspberrypi.org/products/camera-module-v2/>.
- [47] *Official Gerber Format Website*. URL: <https://www.ucamco.com/en/gerber>.
- [48] E. Rublee et al. “ORB: An efficient alternative to SIFT or SURF”. Em: *2011 International Conference on Computer Vision*. 2011, pp. 2564–2571.

-
- [49] F. K. Noble. “Comparison of OpenCV’s feature detectors and feature matchers”. Em: *2016 23rd International Conference on Mechatronics and Machine Vision in Practice (M2VIP)*. 2016, pp. 1–6.
- [50] Guilherme B. F. Paulo, Luís M. R. Perdigoto e Sergio M. M. Faria. “Automatic visual inspection system for solder paste deposition in PCBs”. Em: *2021 Telecoms Conference (ConfTELE)*. 2021, pp. 1–6. DOI: 10 . 1109 / ConfTELE50222 . 2021 . 9435522.

Apêndice A

Artigos Publicados

G. B. F. Paulo, L. M. R. Perdigoto and S. M. M. Faria, "Automatic visual inspection system for solder paste deposition in PCBs", 2021 Telecoms Conference (ConfTELE), 2021, pp. 1-6, doi: 10.1109/ConfTELE50222.2021.9435522.

Apêndice B

Manual de utilizador da interface gráfica

DIGWEST

Manual de utilização do sistema de inspeção de pasta de solda

29 de setembro de 2021

Conteúdo

1	Introdução	2
2	Janela Principal/Inspeção	2
2.1	Efetuar Inspeção	3
2.2	Aceder às janelas de calibração e criação de gerber	3
3	Janela de Calibração	4
3.1	Parâmetros de calibração	4
3.2	Calibração	7
4	Janela de criação de painel de gerber	9

1 Introdução

Este documento tem como objetivo servir de manual de utilização do sistema de inspeção de pasta de solda. São apresentadas todas as funcionalidades do sistema e de que forma podem ser utilizadas para que seja realizada uma inspeção.

Em primeiro lugar é apresentada a página inicial onde é possível realizar inspeções e alterar as configurações da PCB que está a ser inspecionada. Em seguida é apresentada a janela de calibração onde é possível calibrar o sistema para efetuar a inspeção. São apresentados todos os ficheiros que são necessários bem como todas as variáveis necessárias para efetuar a calibração.

2 Janela Principal/Inspeção

A janela principal tem como função permitir ao operador realizar uma inspeção, aceder à janela de calibração, aceder à janela de criação do gerber em matriz e alterar entre configurações de PCB guardadas anteriormente. Essa janela é apresentada na Figura 1.

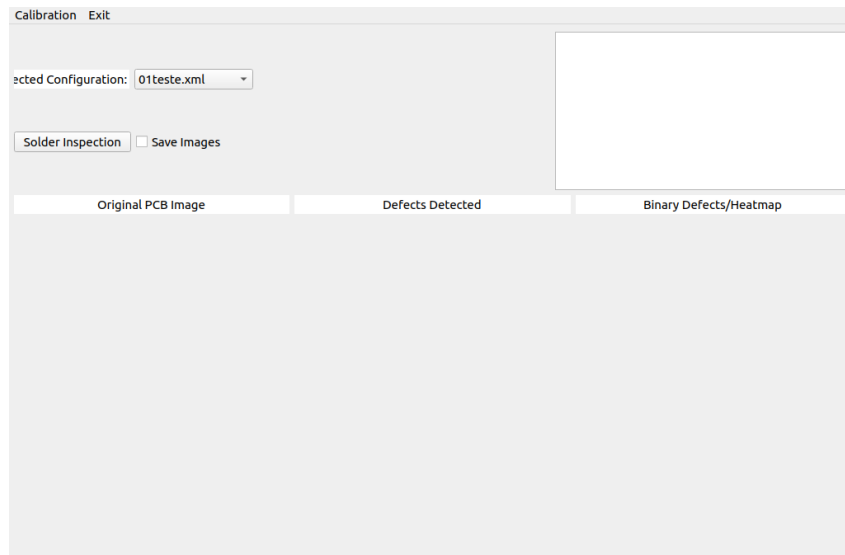


Figura 1: Janela Principal.

2.1 Efetuar Inspeção

Para efetuar uma inspeção de uma PCB apenas é necessário escolher qual a configuração a ser utilizada clicando no botão realçado na Figura 2. São apresentadas todas as configurações de calibrações que foram efetuadas anteriormente e se encontram em memória, como é o caso da Figura 3.

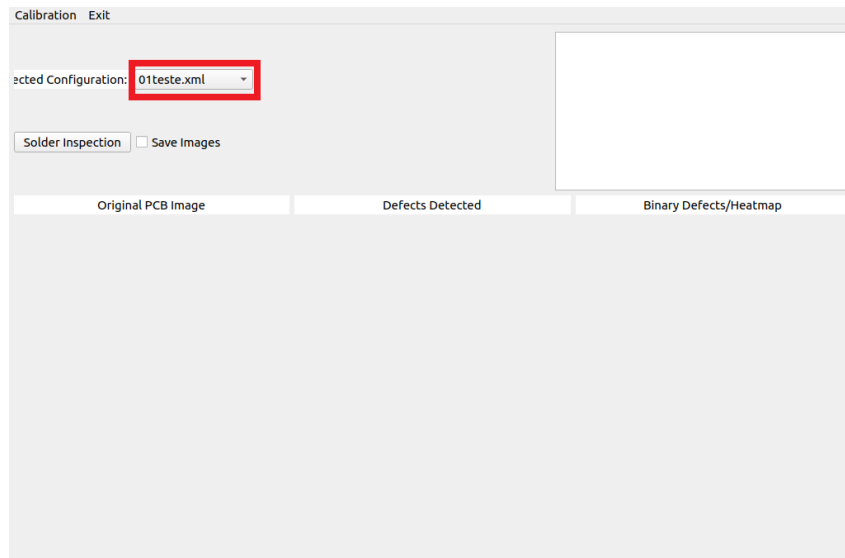


Figura 2: Alterar configuração de inspeção.

Depois de selecionada qual a configuração de inspeção apenas é necessário clicar no botão de inspeção, mostrado na Figura 4.

As imagens da inspeção podem ser guardadas fazendo um *check* na *checkbox* *Save images*, apresentada na Figura 5.

Os resultados da inspeção são apresentados ao operador nas 3 caixas, a primeira para a imagem da PCB original, a segunda mostra os defeitos detetados durante a inspeção e a terceira mostra um *heatmap* de zonas onde ocorrem mais ou menos erros. Na Figura 6 são apresentadas as imagens.

No caso dos resultados da inspeção, a imagem resultante pode estar pintada de duas cores diferentes, **vermelho**, quando existe **falta** de solda e **azul** quando existe solda em **excesso**. Na Figura 7 é apresentado um exemplo de uma PCB onde existe solda em excesso a azul e falta de solda assinalada a vermelho.

2.2 Aceder às janelas de calibração e criação de gerber

Para aceder à janela de calibração e janela de criação de matrizes de gerber é necessário clicar no topo da janela na opção "Calibration" e depois escolher

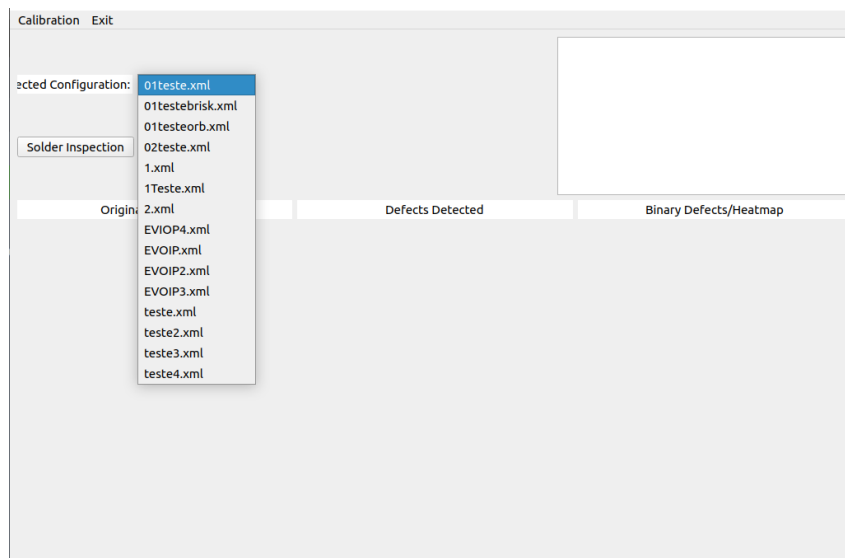


Figura 3: Configurações de inspeção guardadas.

através do clique qual a janela que pretende abrir. Na Figura 8 é possível observar a escolha das janelas, onde "Calibrate Matrices" leva à janela de calibração e "Transform to Matrix" leva à janela de criação de gerbers em matriz.

3 Janela de Calibração

Após abrir a janela de calibração (botões Calibrate->Calibrate Matrices na janela principal), é aberta a janela de calibração, apresentada na Figura 9.

Nesta janela são pedidos todas as imagens e variáveis necessárias para realizar a calibração do sistema. Em primeiro lugar é possível carregar informações de calibrações guardadas em memória. Basta apenas escolher a configuração no menu e clicar no botão "Load Configuration" como é possível observar na Figura 10. Esta opção irá preencher todos os campos e desta maneira facilita a recalibração do sistema.

3.1 Parâmetros de calibração

Para realizar a operação de calibração são necessários vários parâmetros. Em primeiro lugar, são necessários 3 imagens dos ficheiros de gerber da PCB a inspecionar. É necessário uma imagem do gerber apenas com o contorno da PCB, presente na Figura 11a, uma imagem do gerber apenas com a informação da solda, que irá servir de referência para a inspeção, presente

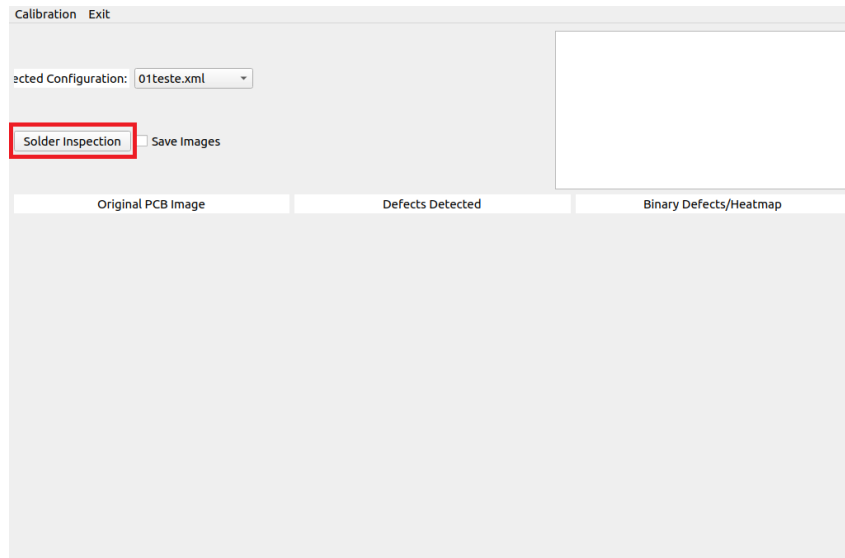


Figura 4: Botão de realizar inspeção.

na Figura 11b e uma imagem do gerber com vários detalhes (solda, buracos, texto), presente na Figura 11c.

Para adicionar estes ficheiros basta clicar no botão ao lado de cada caminho e será aberta uma janela onde apenas é necessário procurar na directoria do sistema como é possível observar na Figura 12.

A próxima imagem necessária é a do xadrez de calibração, a imagem original. Esta imagem está presente na Figura 13 e o método de adicionar a imagem é igual aos métodos de adicionar os imagens dos gerber, basta clicar no botão "Find" ao lado do caminho referente à imagem pretendida.

O próximo campo é o número de cantos do xadrez na vertical e na horizontal. Neste espaço basta apenas escrever o valor nos campos correspondentes e na Figura 14 é possível observar o número de cantos do xadrez, tanto na horizontal como na vertical. O número de cantos é definido pela interceção de quadrados, ou seja, todas as interceções entre quadrados pretos e brancos irão ser identificadas, na imagem só são identificados a primeira linha e a primeira coluna. Para este xadrez de calibração, são detetados 20 cantos na horizontal e 36 cantos na vertical, o que irá resultar numa deteção de 720 cantos no total.

É possível alterar o xadrez de calibração desde que sejam colocados os valores corretos no número de cantos.

Depois de adicionado a imagem do xadrez original, é necessário colocar esse xadrez no plano de trabalho, ou seja, no mesmo plano onde será inspeccionada a PCB. De maneira a garantir que o xadrez de calibração e a PCB se encontra no sitio correto no plano de trabalho e é visível nas duas imagens, é

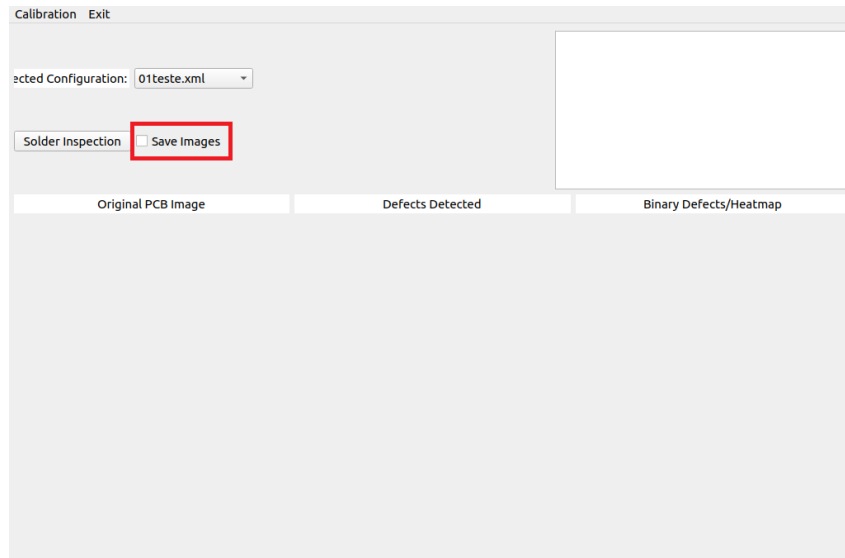


Figura 5: *Checkbox* para guardar imagens de inspeção.

possível abrir uma janela de visualização do plano de trabalho, apresentada na Figura 15. Apenas é necessário clicar no botão "Start Preview" para que seja possível observar o plano de trabalho e garantir que ambas as câmaras contêm toda a informação necessária.

Na Figura 16 é possível observar o xadrez de calibração no plano de trabalho. É necessário que as duas câmaras consigam capturar todos os pontos de intercepção do xadrez, não sendo necessário que os quadrados na periferia fiquem completamente visíveis.

Para tirar foto ao xadrez no plano de trabalho basta apenas clicar no botão "Take Photo" como é possível observar na Figura 17, sublinhado a azul. É necessário tirar fotos com as duas câmaras, com o xadrez na mesma posição por isso basta apenas clicar nos dois botões para tirar foto.

Depois de capturadas as imagens do xadrez de calibração, o próximo passo é colocar as barras de apoio, onde será colocada a PCB. A posição da PCB tem de estar no plano de trabalho, de preferência na mesma posição do xadrez de calibração. Na Figura 18 é possível observar que a PCB e o xadrez de calibração se encontram no mesmo plano e, caso a imagem da PCB fosse sobreposta com a imagem do xadrez, a PCB estaria inserida dentro do xadrez.

À semelhança do xadrez de calibração, também é necessário tirar duas fotos à PCB no local onde será inspecionada. Para isso basta clicar nos botões "Take photo", à semelhança do xadrez de calibração, botões esses presentes na Figura 19, no quadrado azul. Também é possível procurar a imagem caso já tenha sido tirada a fotografia anteriormente. Nesse caso

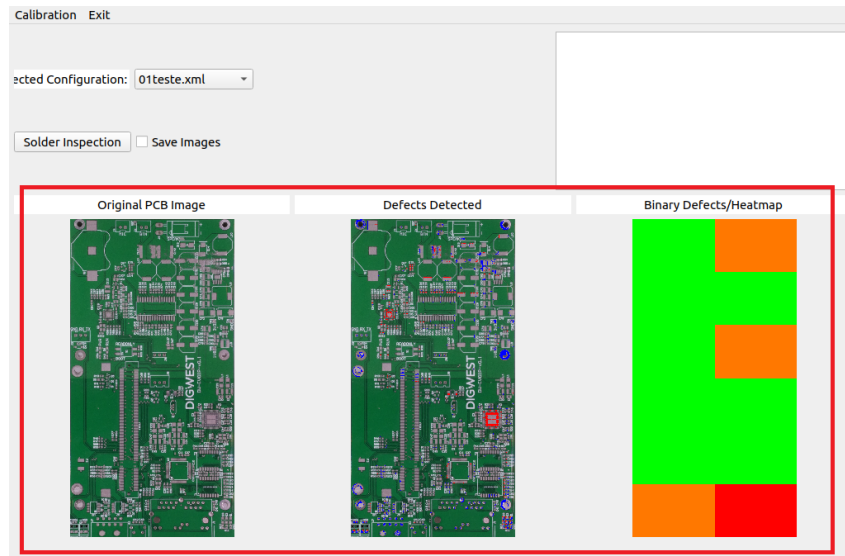


Figura 6: Imagens mostradas ao operador.

apenas é necessário clicar no botão "Find" e o procedimento é igual ao das imagens dos gerber.

Na Figura 20 é possível observar as imagens da PCB em ambas as câmaras na posição final onde serão inspecionadas. Caso sejam alterada a posição das barras de apoio, será necessário realizar uma nova calibração.

O último campo da janela de calibração é o das notas, apresentado na Figura 21. Este campo é opcional e permite ao operador escrever notas acerca da calibração, da PCB ou de qualquer outra situação que seja relevante para o processo de inspeção. Estas notas são apresentadas na janela principal quando é escolhida a configuração da inspeção.

Por último, é apenas necessário escrever qual o nome da configuração de calibração e clicar no botão "Save Configuration", presentes na Figura 22.

3.2 Calibração

Depois de clicado o botão "Save Configuration", será começada a calibração do sistema. Caso o nome da inspeção seja o mesmo de um que está guardado em memória, será aberta uma janela onde é dito que, ao seleccionar aquele nome, a calibração guardada em memória será substituída pela nova calibração.

Podem ocorrer dois tipos de calibração, a calibração manual ou a calibração automática. É possível alterar modos na *checkbox* "Manual Mode" no topo da janela.

Na calibração automática a cor da solda é detetada automaticamente,

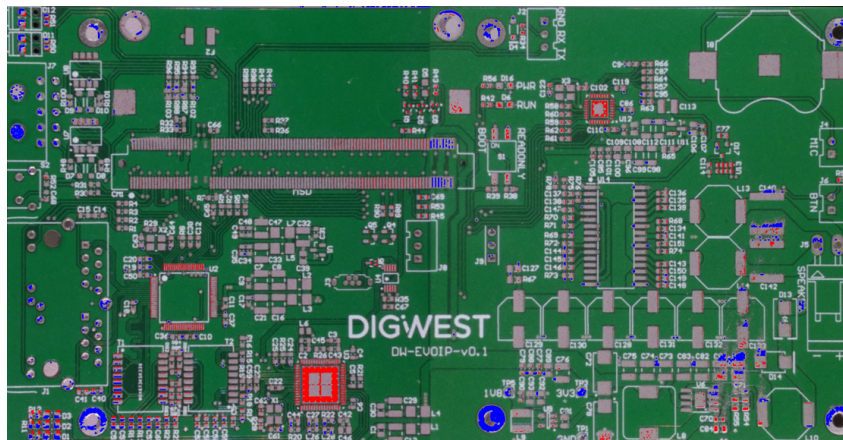


Figura 7: Resultado da inspeção. Vermelho significa falta de solda, azul significa solda em excesso.

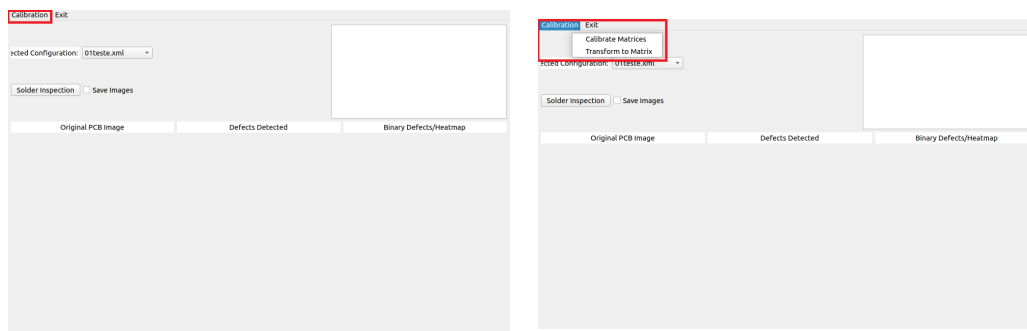


Figura 8: Escolher janela de calibração ou criação de matrizes de gerber.

não sendo necessário realizar qualquer outra tarefa. Na inspeção manual, será aberta uma nova janela onde é necessário escolher a cor da solda. Esta janela é apresentada na Figura 23. É necessário carregar no botão "Load Image" para que a imagem da PCB seja carregada.

Depois de carregada a imagem é necessário clicar num *pad* com a solda visível, de preferência um *pad* com maiores dimensões. Depois de clicado no *pad*, é apresentada a zona num pequeno quadrado no topo da janela para confirmar a seleção. Depois de selecionada a zona correta apenas é necessário clicar no "Ok" para confirmar e continuar a calibração.

Esta seleção de cor da solda é realizada duas vezes, uma para cada parte da PCB, caso sejam utilizadas as duas câmaras. Se apenas for utilizada uma câmara, o algoritmo faz essa deteção automática e apenas é pedida a escolha de cor uma vez.

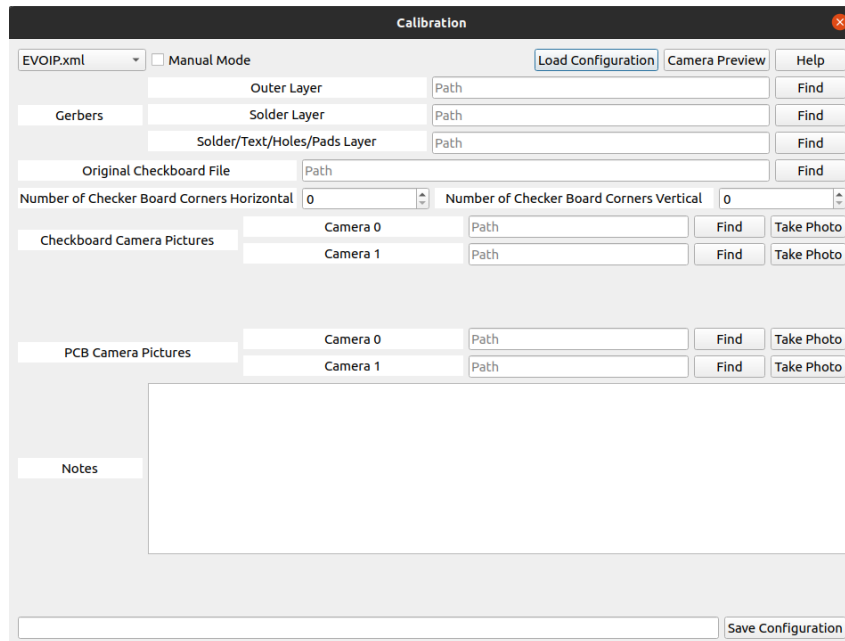


Figura 9: Janela de Calibração.

4 Janela de criação de painel de gerber

A janela de criação de painel de gerber é acedida através do menu principal através do botão no topo da janela principal "Calibration->Transform to Matrix". A janela de criação de painel é apresentada na Figura 25.

São necessárias as apenas imagens dos gerbers além de várias medidas. São necessários os mesmos 3 tipos de imagens que são necessários para a calibração. Na Figura 26 são apresentadas as imagens dos gerber necessários. É necessária a imagem do contorno da PCB, Figura 26a, a imagem do gerber apenas com a componente de solda, Figura 26b e a imagem do gerber com vários detalhes (solda, buracos e legendas), apresentada na Figura 26c.

Em seguida é necessário definir como é a matriz, ou seja, quantos gerber estão no painel. O número de linhas e o número de colunas. Permite que os gerber fiquem todos numa linha ou numa coluna, dependendo da sua orientação.

O próximo passo é indicar o tamanho tanto do gerber, como do painel. São necessárias as medidas em milímetros, do comprimento e largura do painel e o comprimento de largura da PCB que está inserida no painel, que faz referência ao gerber.

Por fim, existem dois botões, o botão "Show Preview" e o botão "Transform". O botão "Show Preview" mostra um dos gerber que será gerado com aquelas configurações para que o operador possa confirmar que será aquela

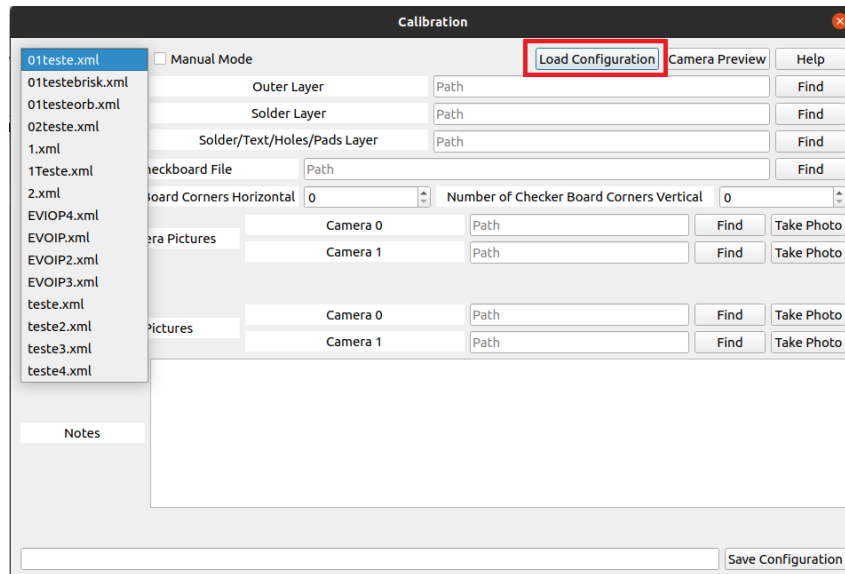
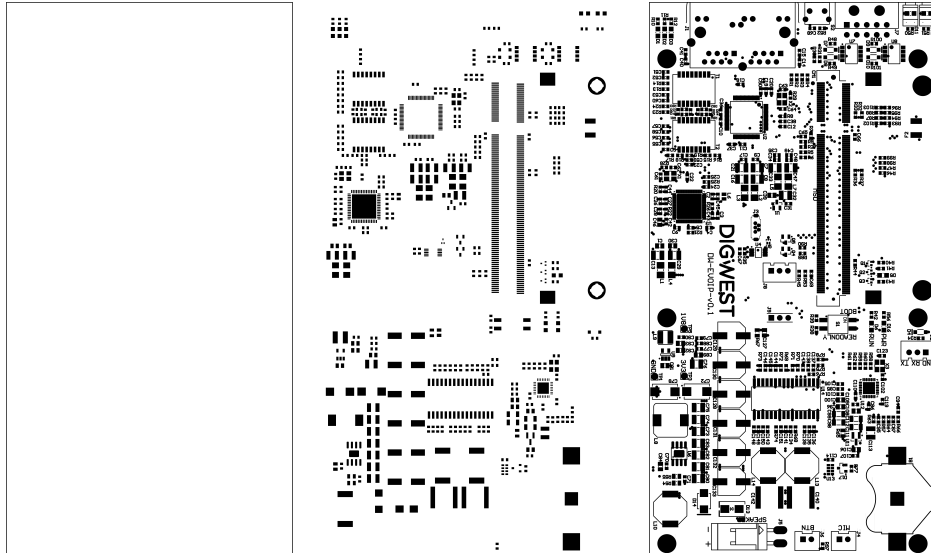


Figura 10: Carregar informações de calibração anteriores.

a PCB que irá ser inspecionada. Caso o gerber esteja correto, basta clicar no botão "Transform" e os 3 gerbers serão gerados na mesma pasta onde estão inseridos os gerbers originais.



(a) Gerber com contorno (b) Gerber com informação da PCB. (c) Gerber com vários de-
da PCB. ção da pasta de solda. talhes.

Figura 11: Gerbers necessários para a calibração.

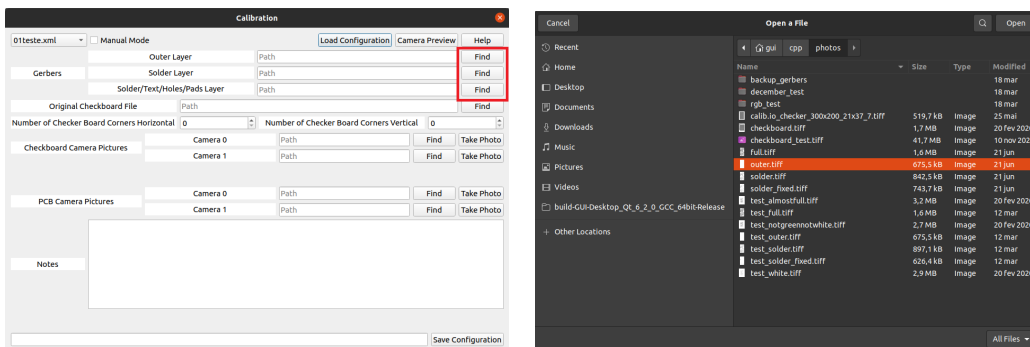


Figura 12: Escolher imagens dos gerber no sistema.

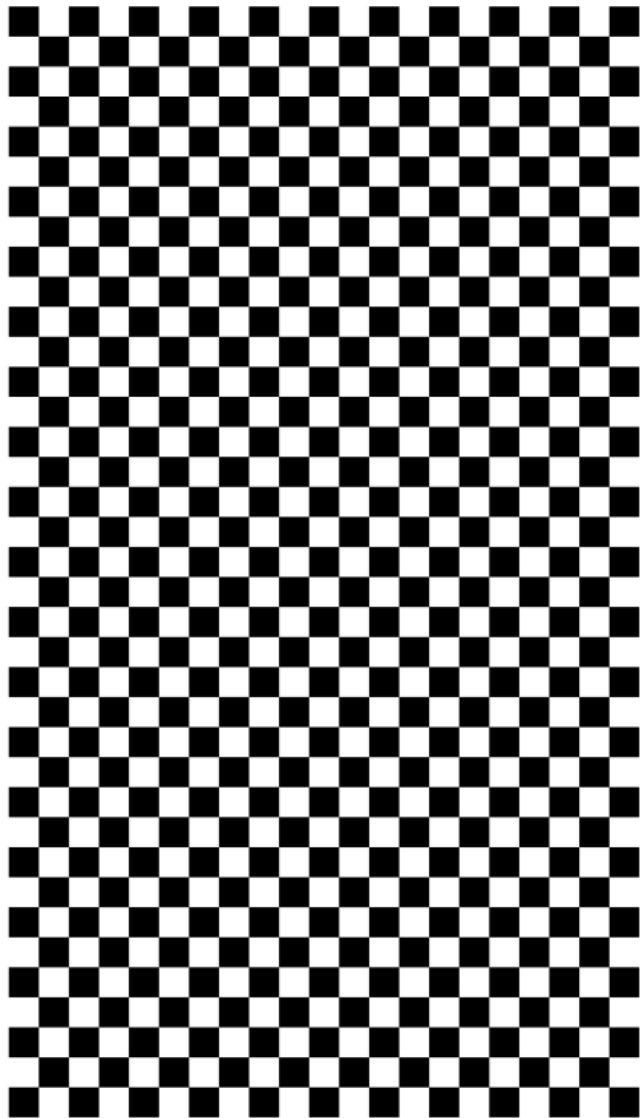


Figura 13: Xadrez de calibração original.

www.calib.io | 21x37 | Checker Size: 7 mm.

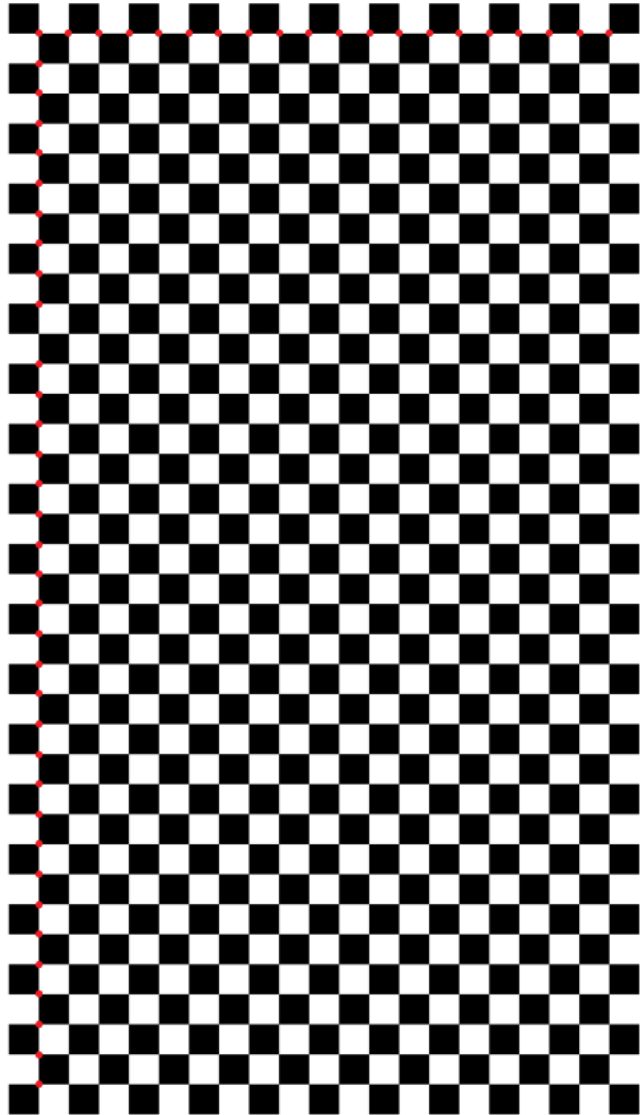


Figura 14: Número de cantos do xadrez de calibração.

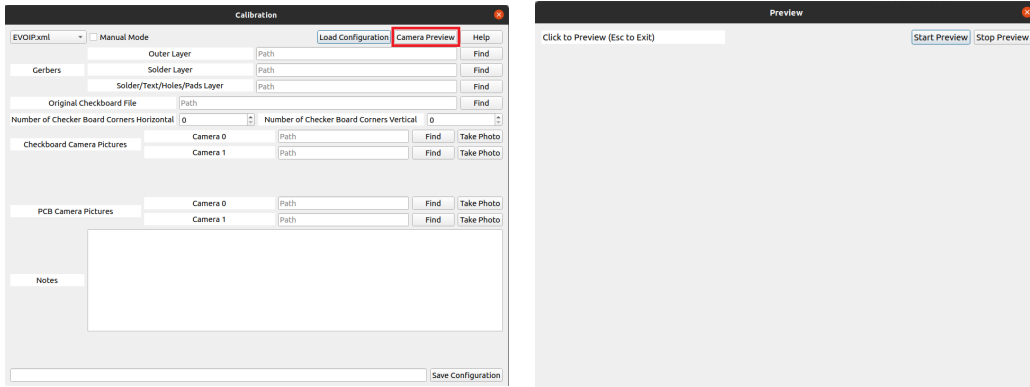


Figura 15: Xadrez de calibração no plano de trabalho.

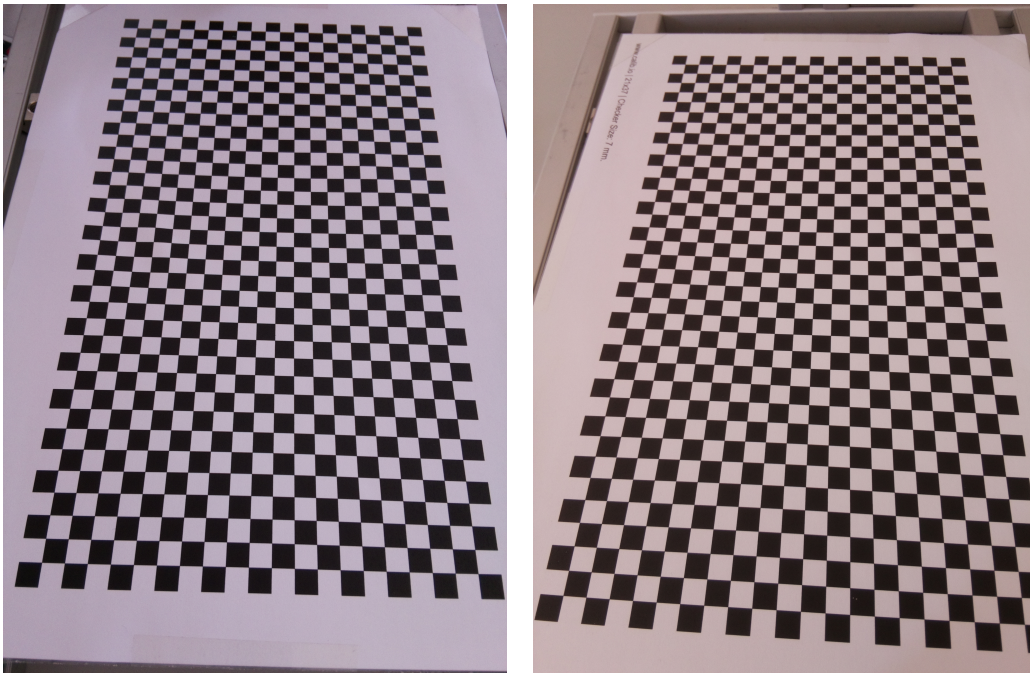


Figura 16: Xadrez de calibração no plano de trabalho.

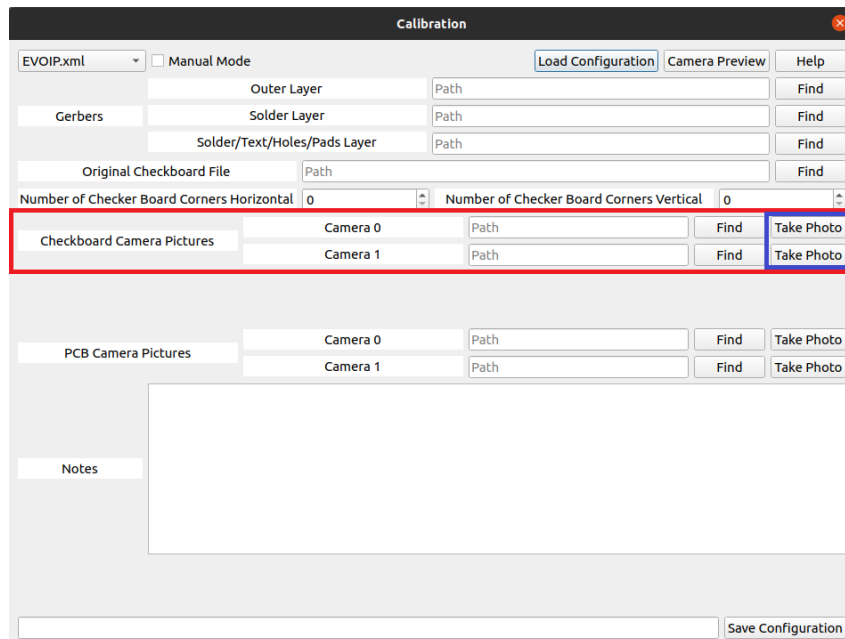


Figura 17: Botões de tirar foto ao xadrez de calibração.

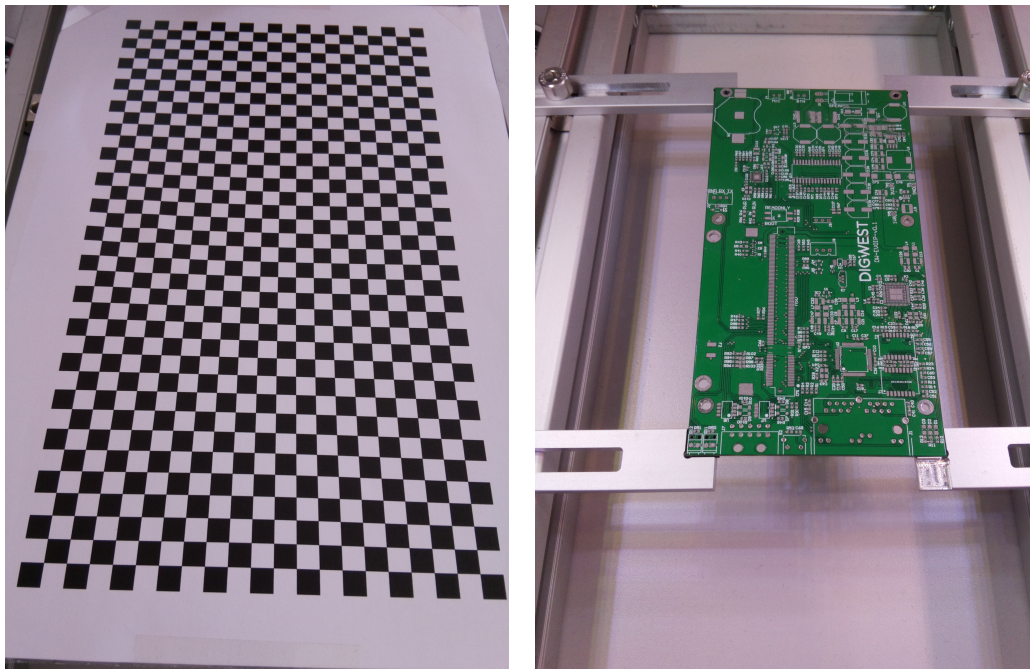


Figura 18: Posição da PCB em relação ao xadrez.

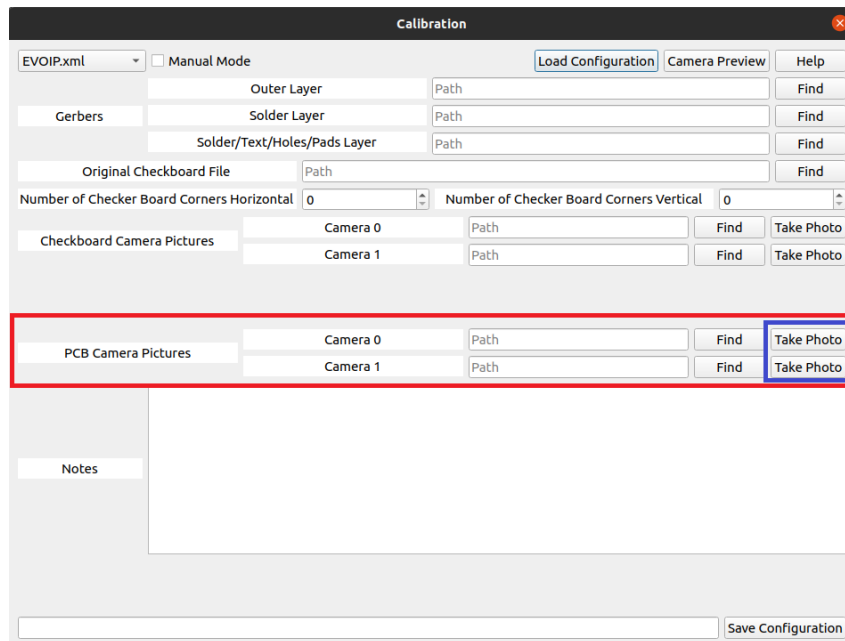


Figura 19: Botões de tirar foto à PCB.

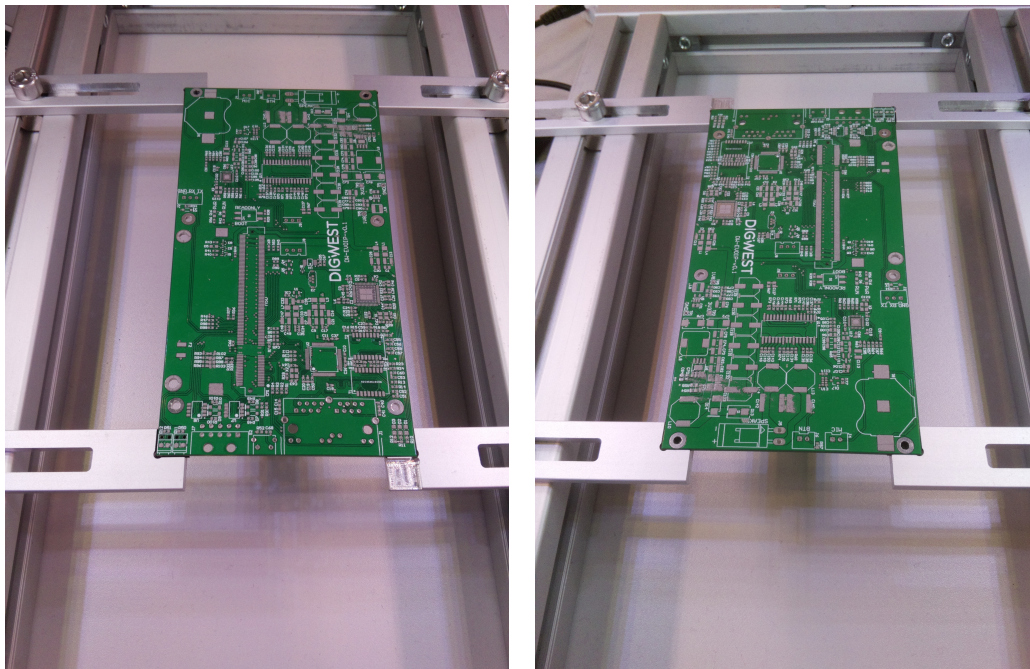


Figura 20: Imagens da PCB nas duas câmaras.

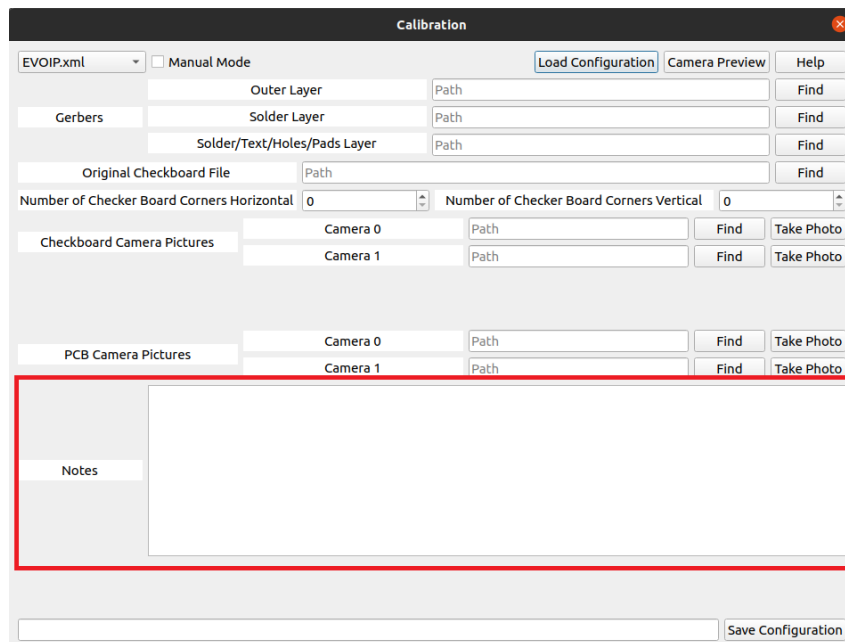


Figura 21: Espaço para notas.

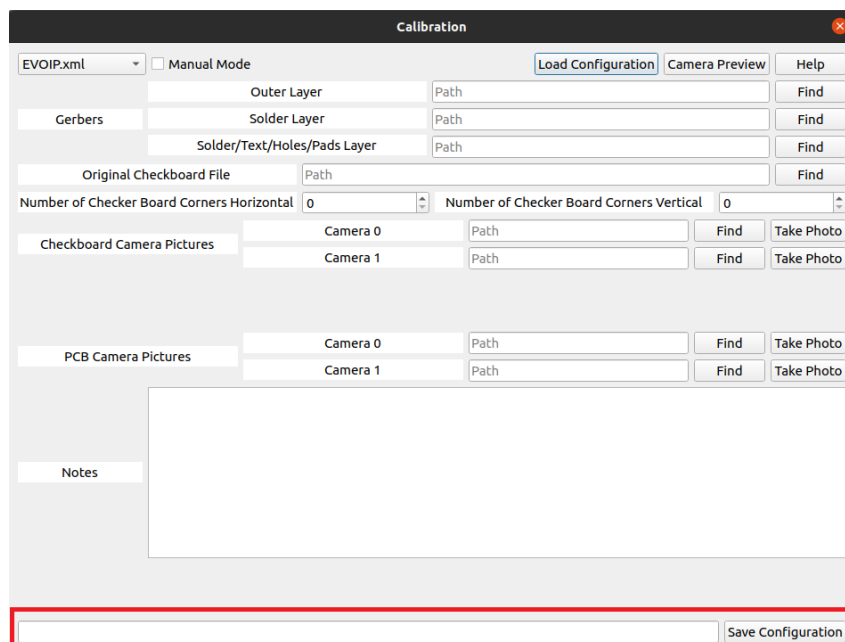


Figura 22: Guardar calibração.

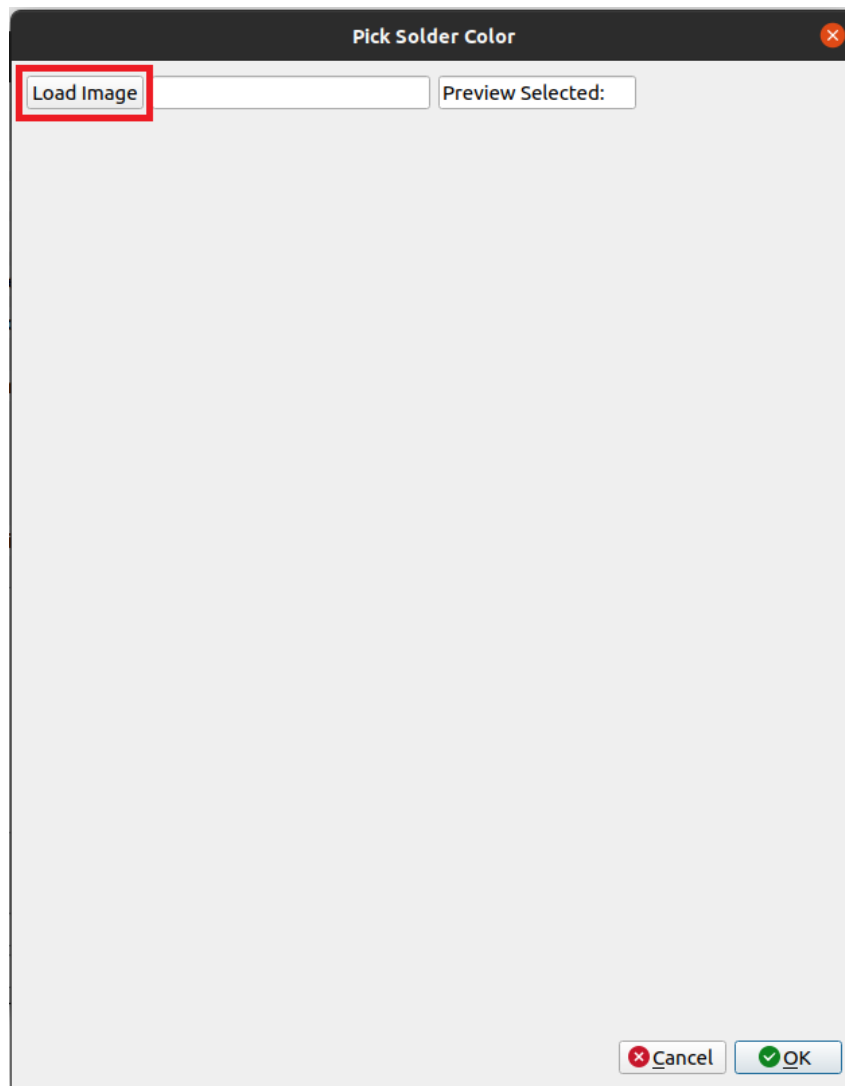


Figura 23: Carregar imagem para escolher cor da solda.

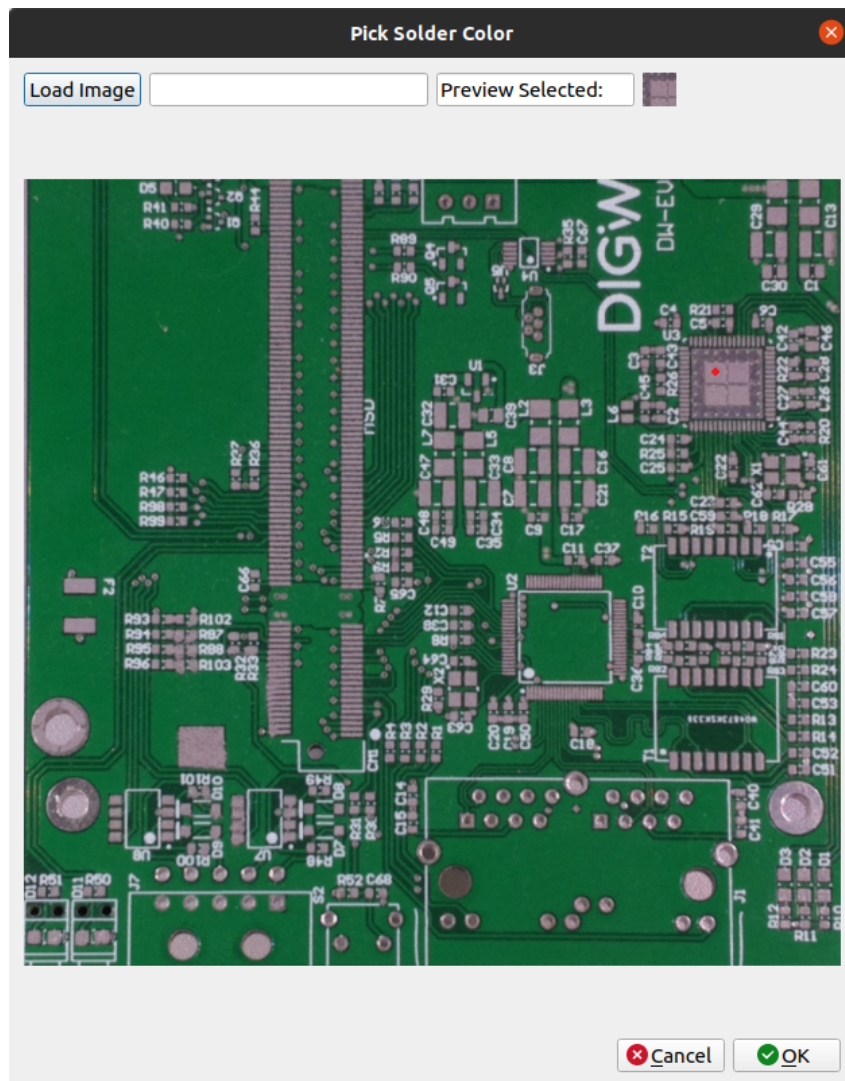


Figura 24: Escolher a cor da solda através do clique na imagem.

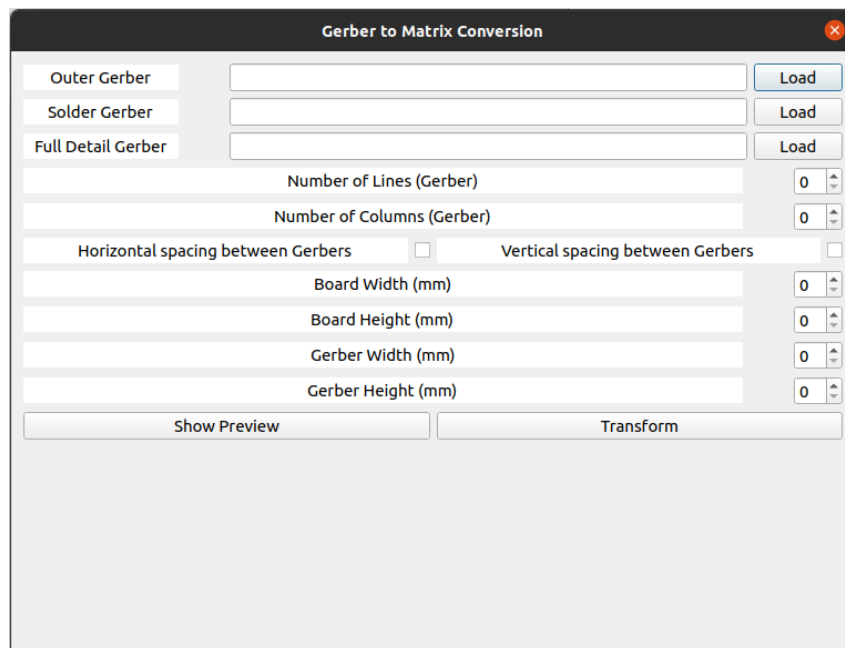
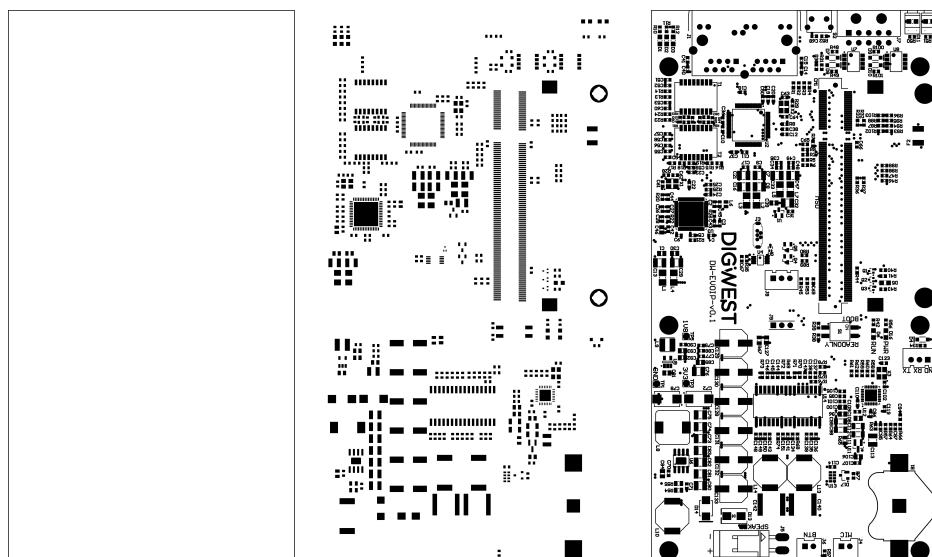


Figura 25: Janela de criação de painel de gerber.



(a) Gerber com contorno da PCB. (b) Gerber com informação da pasta de solda. (c) Gerber com vários detalhes.

Figura 26: Gerbers necessários para a transformação em matriz.