



Avaliação da Qualidade do Fabrico Aditivo por Inspeção Visual Automática

RUI PEDRO CASTRO DO NASCIMENTO

julho de 2021

Avaliação da Qualidade do Fabrico Aditivo por Inspeção Visual Automática

Mestrado em Engenharia Eletrotécnica e de Computadores - Automação e Sistemas

Rui Pedro Castro Nascimento

Orientação
Maria Isabel Martins
Co-Orientação
Luis Carlos Moreira

Ano Letivo: 2020-2021

Instituto Superior de Engenharia do Porto
Departamento de Engenharia Eletrotécnica
Rua Dr. António Bernardino de Almeida, 431, 4200-072 Porto

Agradecimentos

A realização desta dissertação não seria possível sem o apoio incansável da professora Maria Isabel Martins e o engenheiro Luís Moreira. Toda a disponibilidade, conhecimentos e todos os conselhos transmitidos foram determinantes para a conclusão desta etapa e por isso, os meus sinceros agradecimentos.

Um profundo obrigado à minha família, em especial aos meus pais por serem o meu maior exemplo de vida, de dedicação, de amor e carinho. Ao meu irmão, por ser o meu melhor amigo, meu protetor, meu ouvinte e ter sempre uma palavra amiga quando era necessário e por acreditar em mim mesmo quando eu não acreditava. Aos meus avós, por serem os meus anjos da guarda e ao longo das suas vidas sempre me ensinaram a nunca baixar os braços e trabalhar para conseguir concretizar os meus sonhos e objetivos, os seus olhos sempre transmitiram ternura, paciência, força, sabedoria e experiência de uma vida inteira.

Agradecer aos meus amigos, Carla, Eduarda, Francisco e Ricardo, por terem feito parte destes cinco anos de estudantes, pela paciência, pela boa e má disposição, pelas longas noites de estudo onde o desalento nunca se instalou, pelo convívio e pelas memórias que levo para a vida. Agradecer à Teresa, Sofia, João e António porque também sem eles estes cinco anos não seriam o mesmo. Sem eles este percurso seria muito mais complicado.

Por último, mas não menos importante, gostaria de agradecer ao Instituto Superior de Engenharia do Porto e a todos os docentes com quem tive o privilégio de me cruzar ao longo destes anos. Ao INEGI e a todos os seus colaboradores que de algum modo contribuíram para a realização desta dissertação.

Resumo

Este trabalho apresenta um sistema de inspeção visual automática para a classificação de superfícies impressas por processos de fabrico aditivo. São também localizados os defeitos, caso existam, nessas mesmas superfícies. A análise e classificação é realizada por comparação com uma superfície de referência. O sistema está dividido em quatro etapas principais: pré-processamento, análise da cor, análise da superfície e localização de defeitos.

A análise de cor é realizada no espaço de cor CIEL*a*b*, sendo calculada a distância de cor entre a superfície em análise e a de referência, utilizando a fórmula da diferença de cor ΔE_{2000} . As superfícies que não se encontram dentro dos parâmetros predefinidos são rejeitadas de imediato, e as restantes passam para a próxima etapa onde é feita a análise à superfície da peça.

A análise à superfície consiste na binarização da imagem utilizando o detetor de contornos *Canny* sendo, de seguida, calculados os momentos de Hu da imagem da superfície a ser analisada que são comparados com os da superfície de referência. As superfícies que se encontrem dentro dos parâmetros estabelecidos como aceitáveis, são classificadas como superfícies de boa qualidade, as restantes passam para a próxima etapa, onde é realizada a localização dos defeitos. Para a localização de defeitos, a imagem da superfície a analisar é primeiro filtrada com um filtro de mediana. A imagem original e a filtrada são subtraídas, e a imagem resultante da subtração é então binarizada, sendo os defeitos localizados através de um detetor de *blobs*. As superfícies onde sejam detetados erros, são consideradas de má qualidade.

Numa fase de treino, foi utilizado um subconjunto de peças para avaliar o desempenho dos diferentes métodos considerados e estabelecer os valores de alguns parâmetros. Posteriormente, numa fase de teste e validação, o desempenho do sistema foi avaliado recorrendo a um conjunto distinto de peças.

Palavras-Chave: Inspeção Automática, Fabrico Aditivo, Cor, Análise de Superfície, Defeitos.

Abstract

This work presents an automatic visual inspection system for the classification of printed surfaces by additive manufacturing processes. Defects, if any, are also located on those same surfaces. The system is divided into four main steps: pre-processing, color analysis, surface analysis and defect location.

The color analysis is performed in CIEL*a*b* colour space, and the color distance between the surface under analysis and the reference surface is calculated using the color difference formula ΔE_{2000} . The surfaces that are not within the predefined parameters are rejected immediately, and the remaining ones are transferred to the next step where the surface analysis is performed.

The surface analysis consists in the binarization of the image using the Canny edge detection operator and, then, the Hu moments of the image of the surface under analysis are calculated and compared with those of the reference surface. The surfaces that are within the parameters established as acceptable are classified as good quality surfaces, the remaining ones go on to the next stage, where the location of defects is performed. To locate the defects, the image of the surface to be analysed is first filtered with a median filter. The original image and the filtered one are subtracted, the resulting image is then binarized, and the defects are located through a blob detector. Surfaces where errors are detected are considered to be of poor quality.

In a training phase, a subset of parts was used to evaluate the performance of different methods and to establish the values of some parameters. Later, in a testing and validation phase, the performance of the system was evaluated using a different set of parts.

Keywords: Automatic Inspection, Additive Manufacturing, Color, Surface Analysis, Defects.

Índice

Agradecimentos	iii
Índice	i
Lista de Figuras	v
Lista de Tabelas	vii
Acrónimos	ix
1 Introdução	1
1.1 Contextualização	1
1.2 Objetivos	2
1.3 Plano	3
1.4 Estrutura	3
2 Revisão da Literatura	5
2.1 Sistemas de Inspeção Automática	5
2.2 Extração de Características	6
2.2.1 Características de cor	7
2.2.1.1 Histogramas	7
2.2.1.2 Momentos de Cor	7
2.2.1.3 Correlograma de cores	8
2.2.1.4 Vetor de Coerência de Cor	8
2.2.2 Cálculo da distância de cor	8
2.2.3 Características de Textura	10
2.2.4 Características de Forma	12
2.2.5 Medição de similaridade	14
2.2.5.1 Distância Minkowski-Form	14
2.2.5.2 Distância Euclidiana	15
2.3 Classificação de Defeitos	15

2.4	Exemplo Detecção de Defeitos em Superfícies de Cimento	16
2.5	Fabrico Aditivo	16
2.5.1	Métodos de Fabrico Aditivo	17
2.5.1.1	Fused Deposition Modelling	17
2.5.2	Tipos de Erros no Fabrico Aditivo	18
3	Arquitetura	21
3.1	Hardware	21
3.1.1	Câmara	22
3.1.2	Iluminação	23
3.1.3	PC BECKHOFF C6930	24
3.2	Software	25
3.2.1	ADS Image Watch	27
4	Sistema Desenvolvido	29
4.1	Sistema Para a Classificação de Peça	29
4.2	Aquisição de Imagens	30
4.2.1	Calibração	31
4.3	Processamento de Imagem	34
4.3.1	Recorte da Imagem	34
4.4	Binarização	34
4.5	Análise de Cor	36
4.5.1	Conversão Espaço de Cor	36
4.5.2	Cálculo da Distância de Cor	38
4.6	Análise de Superfície	38
4.6.1	Extração de Características	39
4.6.1.1	Detetor de Contornos de <i>Canny</i>	39
4.6.1.2	Momentos de Hu	40
4.7	Localização de Defeitos	41
4.7.1	Filtro de Mediana e Subtração	42
4.7.2	Binarização	44
4.7.3	Operações Morfológicas	44
4.7.4	Localização de Defeitos	45
4.8	Interface Gráfica	46
5	Testes e Discussão dos Resultados	49
5.1	Peças	49
5.2	Calibração	50
5.3	Binarização	51
5.3.1	Detecção de Objeto	52
5.4	Avaliação do sistema de classificação	53
5.5	Fase de Treino	55

5.5.1	Análise de cor	55
5.5.2	Análise de forma	56
5.6	Fase de Testes	58
5.7	Localização de Defeitos	62
6	Conclusão	65
	Referências Bibliográficas	67

Lista de Figuras

1.1	Calendarização	3
2.1	Espaço de cores $L^*a^*b^*$, [fonte: [7], 1970].	9
2.2	Classificação de técnicas para representação e descrição de forma, [fonte: [21], 2].	13
2.3	Primeira etapa de pré-processamento, [fonte: [30], 247].	16
2.4	Fused Deposition Modelling, [fonte: [35], 2021].	18
3.1	Montagem do sistema	22
3.2	SVCAM exo264CGE, [fonte: [38], 2020].	23
3.3	Exemplo de iluminação direcional	23
3.4	Iluminadores LED, JYLED-500s, [fonte: [39], 2020].	24
3.5	PC industrial C6930, [fonte: [40], 2020].	25
3.6	Protocolo de comunicação ADS, [fonte: [42], 2020].	26
3.7	<i>Insert vision device</i> , [fonte: [44], 2020].	26
3.8	<i>Camera configuration assistant</i>	27
3.9	ADS Image Watch	27
4.1	Sequência de operações para a classificação de uma peça	30
4.2	Padrão de calibração - círculos assimétricos, [fonte: [47], 2020].	33
4.3	Módulo vision	33
4.4	Resultado do recorte da imagem	34
4.5	Imagem binária	36
4.6	Resultado da aplicação da máscara	39
4.7	Imagem de entrada, à esquerda, e resultado do detetor de contornos de Canny, à direita	40
4.8	Resultado da aplicação da máscara	42
4.9	Resultados da subtração para imagens em tons de cinza e imagens RGB	43
4.10	Binarização seguida de dilatação	45
4.11	Localização de defeitos na superfície	46
4.12	Interface gráfica HMI	46

4.13	Classificação da superfície - cor errada	47
4.14	Classificação da superfície - erros na superfície	47
4.15	Classificação da superfície - boa qualidade	47
5.1	Exemplos de peças utilizadas	50
5.2	Resultado da medição de um objeto	51
5.3	Resultados para os diferentes métodos de binarização	52
5.4	Resultado da detecção de objeto	53
5.5	Curvas ROC para ΔE e ΔE_{2000}	56
5.6	Curvas ROC- <i>Match Hu Moments</i>	58
5.7	Peça exemplo 1	60
5.8	Peça exemplo 2	61
5.9	Exemplo do suporte visível nas peças exemplo	62
5.10	Exemplo do suporte a ser detetado pelo operador Canny na peça exemplo 2	62
5.11	Localização dos defeitos nas diferentes peças	63

Lista de Tabelas

2.1	Tipos de defeitos comuns no fabrico aditivo e correspondentes causas e consequências [37]	19
5.1	Resultados da análise de cor baseados no ΔE - fase treino	55
5.2	Resultados da análise de cor baseados no ΔE_{2000} - fase treino	56
5.3	Resultados da análise da forma baseados no método 1 - fase treino	57
5.4	Resultados da análise da forma baseados no método 3 - fase treino	57
5.5	Resultados da análise baseada em cor	58
5.6	Resultados da análise baseada na forma	59
5.7	Desempenho global do sistema	59
5.8	Desempenho global do sistema - peça exemplo 1	61
5.9	Desempenho global do sistema - peça exemplo 2	61

Acrónimos

Acrónimo	Descrição
3DP	<i>Three-Dimensional Printing</i>
ADS	<i>Automation Device Specification</i>
AUC	<i>Area Under The Curve</i>
CAD	<i>Computer Aided Design</i>
CBIR	<i>Content-based image retrieval</i>
DED	<i>Deposição de Energia Direta</i>
DoFs	<i>Degrees of Freedom</i>
FDM	<i>Fused Deposition Modeling</i>
FN	<i>False Negative</i>
FOL	<i>Fabricação de Objeto Laminado</i>
FOV	<i>Field of View</i>
FP	<i>False Positive</i>
FPR	<i>False Positive Rate</i>
fps	<i>frames per second</i>
GB	<i>Gigabyte</i>
GDDM	<i>GigE Device Discovery Mechanism</i>
GHz	<i>GigaHertz</i>
GLCM	<i>Gray-Level Co-Occurrence Matrix</i>
GVCP	<i>GigE Vision Control Protocol</i>
GVSP	<i>GigE Vision Stream Protocol</i>
HMI	<i>Human Machine Interface</i>
IEC	<i>International Electrotechnical Commission</i>
INEGI	<i>Instituto de Ciência e Inovação em Engenharia Mecânica e Engenharia Industrial</i>
IP	<i>Internet Protocol</i>
ISO	<i>International Organization for Standardization</i>
LBP	<i>Local Binary Patterns</i>
LED	<i>Light-Emitting Diode</i>
PC	<i>Personal Computer</i>
PLA	<i>Polylactic Acid</i>
PLC	<i>Programmable Logic Controller</i>

Acrónimo	Descrição
RAM	<i>Random Access Memory</i>
RGB	<i>Red, Green, Blue</i>
ROC	<i>Receiver Operating Characteristic</i>
SLM	<i>Selective Laser Melting</i>
SLS	<i>Selective Laser Sintering</i>
SVM	<i>Support Vector Machine</i>
TN	<i>True Negative</i>
TP	<i>True Positive</i>
TPR	<i>True Positive Rate</i>
TwinCAT	<i>The Windows Control and Automation Technology</i>
UDP	<i>User Datagram Protocol</i>

Capítulo 1

Introdução

Baseado na premissa da adição sucessiva de material para a criação de geometrias sólidas, o fabrico aditivo permite a execução dos componentes de forma gradual, camada-a-camada, permitindo assim a criação de peças, consideradas complexas ou impossíveis de fabricar, como vazios interiores, sem qualquer custo acrescido, com alta precisão e eficiência de recursos. A intervenção do operador é também reduzida, uma vez que as etapas de desenvolvimento são condensadas, originando assim um percurso mais direto, desde a idealização, modelo de um *computer aided design* (CAD) 3D, até à materialização da ideia.

Na indústria, sistemas de visão computacional têm sido utilizados no guiamento de robôs ou veículos móveis e na inspeção visual automática. A inspeção visual automática é utilizada para aferir medidas geométricas, texturas e cores de produtos em áreas tão diversas como, por exemplo, a indústria automóvel e a indústria têxtil.

Os sistemas de visão computacional são compostos por câmaras, sistemas de iluminação e computadores. Estes utilizam técnicas de processamento de imagem para extração da informação, apoiadas em técnicas de decisão automática da área de inteligência artificial.

1.1 Contextualização

Inseridas num mercado altamente competitivo, as empresas fazem uso de novos métodos com o objetivo de se destacarem das suas concorrentes. O uso de tecnologias como sistemas de visão computacional e fabrico aditivo, em determinadas aplicações, é cada vez mais uma realidade nos dias de hoje. Estas são utilizadas pelas empresas de modo a aumentar a sua produtividade e a qualidade dos seus produtos, diminuindo os custos de produção.

O fabrico aditivo, ou impressão 3D, é uma das áreas da indústria 4.0 que têm vindo a sofrer uma evolução extraordinária nos últimos tempos permitindo às empresas uma resposta mais rápida, precisa e com boa qualidade aos requisitos dos clientes. Por isso, são cada vez mais as que optam por introduzir o fabrico aditivo nos seus modelos de negócio.

As indústrias exigem cada vez mais um elevado padrão e índice de uniformidade, e é por isso necessário que o operador esteja concentrado, quando está a realizar o processo de controlo de qualidade. No entanto, quando esse controlo é realizado por operadores humanos existem algumas desvantagens, tal como o tempo na inspeção, quando comparado com um sistema automático e, derivado a fatores externos, ambientais ou pessoais, a eficiência nem sempre é a mesma. Resulta assim um custo mais elevado para a empresa e sem uma total garantia do resultado final da inspeção.

Um sistema de inspeção automática, quando inserido em ambientes industriais, deve ser capaz de realizar diversos objetivos, como a deteção de vários tipos de defeitos, tendo em conta o tempo de execução, o custo computacional e efetuar uma aquisição de imagens adequada de modo a que a análise feita seja o mais precisa possível.

O INEGI, foi criado em 1986, com o intuito de realizar atividades de investigação e de inovação de base tecnológica, transferência de tecnologia, consultoria e serviços tecnológicos, orientados para o desenvolvimento da indústria e da economia em geral. O desenvolvimento de tecnologias de fabrico aditivo tem sido um dos principais focos de desenvolvimento do INEGI, surgindo assim a necessidade de desenvolver um sistema de inspeção automática para avaliar as peças produzidas por fabrico aditivo. O sistema desenvolvido neste trabalho, será depois integrado com um robô colaborativo e um transportador de alta velocidade, de modo a criar um sistema de automação industrial, para avaliação da qualidade do fabrico aditivo.

1.2 Objetivos

Este projeto teve dois objetivos principais, a realização do estudo do *software*, TwinCAT Vision, analisando as suas potencialidades e limitações no processamento de imagens e o desenvolvimento de um sistema de inspeção visual automática para a deteção e localização de defeitos em peças fabricadas em impressão 3D. De modo a ser possível cumprir os objetivos propostos, é possível distinguir os seguintes objetivos, mais específicos, para este projeto:

- Levantamento das funções existentes na biblioteca disponível no *software* da Beckhoff TwinCAT Vision, para o processamento de imagem;

- Aplicar técnicas de processamento de imagem com vista à deteção de defeitos em peças fabricadas por fabrico aditivo;
- Detetar e localizar defeitos em peças fabricadas por fabrico aditivo;
- Decidir a rejeição ou aceitação do produto em questão.

1.3 Plano

O *Gantt Chart* ilustrado na Figura 1.1 resume o plano de trabalhos para este projeto.

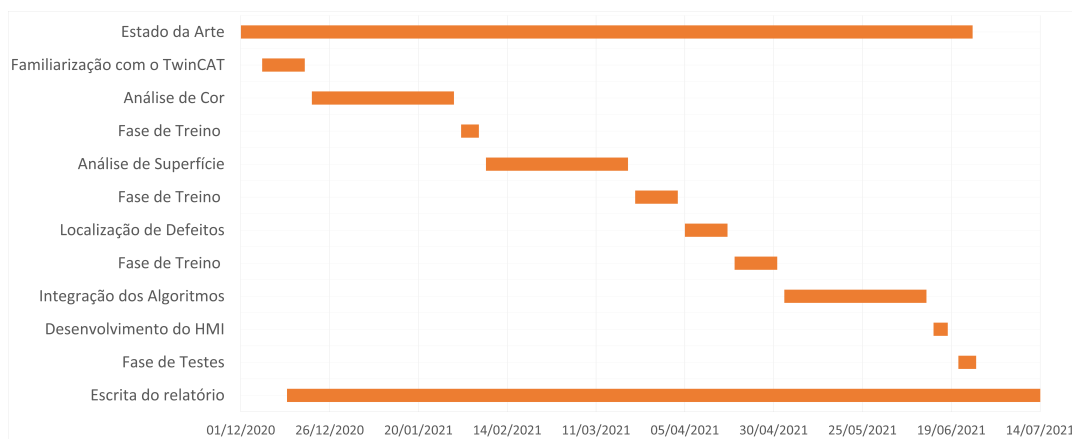


Figura 1.1: Calendarização

1.4 Estrutura

Este relatório encontra-se dividido em 6 capítulos.

No presente capítulo, é feita uma contextualização dos sistemas de inspeção automática. É feita uma introdução ao problema, e são apresentados os objetivos gerais e específicos deste projeto.

No segundo capítulo são apresentadas diferentes técnicas de análises relacionadas com o trabalho, análise de cor, análise textural e de forma e técnicas de localização de defeitos em superfícies. São ainda apresentados trabalhos que fundamentam essas mesmas análises e técnicas.

O terceiro capítulo descreve a arquitetura do sistema desenvolvido, onde é apresentado o sistema proposto para resolver o problema em causa.

O quarto capítulo, apresenta o algoritmo desenvolvido para a inspeção das peças. São explicadas as diferentes fases do algoritmo, desde a aquisição de imagens até à localização de defeitos, dando ainda uma explicação teórica dos diferentes métodos utilizados.

No quinto capítulo são apresentados os resultados obtidos, desde a correta calibração da câmara, às diferentes técnicas de binarização e detecção de objeto. São ainda apresentados os resultados obtidos durante a fase de treino do algoritmo, apresentando uma justificação dos métodos e dos valores de limiar de decisão escolhidos. Por fim, são apresentados os resultados obtidos na fase de testes e validação do algoritmo, apresentando a sua precisão na classificação de diferentes superfícies.

No último capítulo, capítulo 6, são retiradas conclusões e são perspetivados futuros desenvolvimentos.

Capítulo 2

Revisão da Literatura

Os sistemas de visão computacional são o resultado da integração de diferentes tecnologias, podendo ser aplicados em diversos tipos de situações. Neste capítulo são apresentadas algumas das análises possíveis para a realização do controle de qualidade e detecção de defeitos por inspeção visual automática.

2.1 Sistemas de Inspeção Automática

Na indústria, os sistemas de visão computacional podem ser utilizados para a inspeção do produto fabricado, de modo a verificar se este se encontra com defeito ou não. Os sistemas de inspeção automática, na indústria, são classificados de acordo com o tipo de inspeção que é feita ao produto, a classificação é dividida nomeadamente em quatro categorias. Inspeção da qualidade dimensional, onde são analisadas as dimensões, formas, posicionamento, orientação, entre outras, de um produto. Inspeção da qualidade da superfície, analisa características como fissuras, desgaste, acabamentos e texturas. Inspeção da correta montagem do produto, qualidade estrutural, onde são analisados os furos, ranhuras e parafusos. Inspeção da qualidade operacional, onde o produto é analisado para verificar a sua compatibilidade de operação de acordo com os padrões e especificações definidos. Apesar de os quatro modos serem diferentes, estes têm todos o mesmo objetivo, confirmar se o produto cumpre os padrões de satisfação especificados [1].

Quando aplicados à indústria, os sistemas podem ainda ser classificados com base em características cujas medições não afetam o processo de inspeção, permitindo, assim, que o sistema seja independente dessas características. Ao conjunto dessas características dá-se o nome de "Degrees of Freedom" (DoFs) do processo de inspeção. Na indústria, os DoFs mais comuns dizem respeito à

forma, dimensões geométricas, textura, cor, entre outros. Quanto menos DoFs, mais forte é a dependência do sistema, contudo estes sistemas são menos propensos a ser expandidos. Um maior número de DoFs permite que o sistema possa ser aprimorado para detetar novos tipos de defeitos caso sejam introduzidas novas funções de processamento e análise de imagem [1].

O primeiro problema a resolver quando se está a desenvolver um sistema deste género é entender o tipo de informação que o sistema deve recuperar e como isso é traduzido em medições ou características extraídas da imagem. Por exemplo, é importante especificar, com antecedência, a definição de "defeito" em termos de medições e regras e implementar essas tarefas em *software* e *hardware*. Então, uma decisão deve ser tomada sobre o tipo de medições a serem adquiridas e a localização para obtenção das mesmas [1].

Um sistema de inspeção automática deve ser confiável e robusto. Para ser considerado confiável, este deve gerar o menor número possível de falsos positivos e falsos negativos. A responsabilidade de manter o sistema como um sistema confiável, recai nas unidades de processamento e classificação. No entanto, a eficiência da classificação depende também da qualidade das imagens adquiridas. Um sistema diz-se robusto, quando este é capaz de se adaptar e alcançar um bom desempenho de forma consistente apesar das irregularidades na iluminação ou condições de fundo e, acomodar as incertezas nos ângulos, posições, entre outros. É difícil alcançar um desempenho robusto, uma vez que altas taxas de reconhecimento e classificação de defeitos são obtidas sob certas condições de boa iluminação e baixo ruído. Por fim, um sistema de inspeção automática deve ser rápido e económico [1].

2.2 Extração de Características

Com a extração de características pretende-se extrair dos pixels da imagem as informações que descrevem as características da mesma. Essas informações são, de seguida, mapeadas para um espaço de características de dimensão inferior, de modo a comprimir a quantidade de informação e melhorar assim a eficiência do reconhecimento de diferenças entre diversas imagens. As características dos defeitos habitualmente usados na inspeção visual incluem características na escala de cinza, características de forma e tamanho e características de textura e características de cor. As características na escala de cinza são as mais intuitivas da imagem, como características estatísticas da escala de cinza e características estatísticas de diferença de escala de cinza. As características relativas à forma e tamanho são informações relevantes para a identificação de vários defeitos. Defeitos comuns podem ser detetados pela informação da forma, como linhas, curvas, elipses e rectângulos ou pela informação do tamanho como área e perímetro. A textura, é uma característica importante da imagem. Não faz uso

da cor ou do brilho para reflectir homogeneidade das imagens. Representa informação importante sobre a disposição estrutural da superfície e a sua relação com o meio envolvente.

2.2.1 Características de cor

Uma das mais difíceis análises a ser feita a uma peça é a análise da cor. A cor é uma quantidade psicofísica, que atua como uma impressão durante a estimulação do sistema visual humano. Existem diversos fatores que podem influenciar a percepção e comparação de cores. Estes podem ser fatores externos, como a iluminação, mas também características individuais da pessoa que está a fazer a respetiva análise. É, por isso, importante definir e caracterizar a cor de uma peça de forma precisa, quando esta é um parâmetro principal. Devido à existência de três tipos de recetores no olho humano, o modelo mais comum para a caracterização de uma cor é o modelo tridimensional. Este cria uma cor sólida com parâmetros independentes [2].

Existem vários métodos para representar a cor. Histogramas, momentos de cor, correlograma de cores, vetor de coerência de cor. São alguns desses métodos.

2.2.1.1 Histogramas

O principal método para representar a informação de cor nas imagens em sistemas *Content-based image retrieval* (CBIR) é através de histogramas. Estes são obtidos discretizando as cores da imagem e contando o número de vezes que cada cor discreta ocorre na imagem. Estes são invariantes a translações e rotações, podendo ser afetados pela mudança de ângulo, escala e oclusão. Uma das principais desvantagens do histograma é a falta de consideração a informação espacial de pixels, imagens podem ser consideradas semelhantes apenas por terem distribuições de cores semelhantes [3].

Em Stricker *et. al.* [4] é proposta uma melhoria do método do histograma de cores que inclui o histograma de cores acumulado. Os resultados demonstraram ser mais vantajoso que a abordagem convencional. Contudo, apresenta a desvantagem de que, no caso de existir mais do que um histograma, não há maneira clara de ordenar as classes.

2.2.1.2 Momentos de Cor

Stricker *et. al.* [4] propuseram utilizar o método dos momentos de cor, evitando assim as desvantagens da quantização. Os momentos de cor são momentos estatísticos da probabilidade de distribuições de cores e têm sido utilizadas com sucesso em vários sistemas especialmente quando a imagem contém apenas o objeto. A primeira ordem, média, a segunda, variância, e a terceira ordem,

assimetria, dos momentos de cor provaram ser eficazes na representação de distribuição de cores. Uma das desvantagens do uso dos momentos de cor é que não conseguem codificar informações espaciais na vizinhança da cor dentro da imagem.

2.2.1.3 Correlograma de cores

Huang *et. al.* [5] propõem o correlograma de cores como um meio de codificar as informações de cor de uma imagem. O método incorpora dados espaciais nas informações de cor codificadas. O correlograma de cores tem como vantagens a inclusão de correlação espacial de cores, permitindo assim descrever a distribuição de correlação espacial local de cores e é também um método fácil de implementar a nível computacional.

2.2.1.4 Vetor de Coerência de Cor

Um vetor de coerência de cores é um histograma dividido que separa os pixels de acordo com a sua coerência espacial. Cada pixel da imagem é dividido em dois tipos, coerente ou incoerente conforme seja parte de uma maior região de cor uniforme. Os histogramas, separados, podem então ser processados para pixels coerentes e incoerentes incluindo alguma informação espacial no vetor de características. Devido ao facto de o vetor de coerência de cor conter informações espaciais, este produz melhores resultados que o histograma de cores, especialmente para imagens que tem maior número de regiões cores uniformes ou maior regiões de textura [6].

2.2.2 Cálculo da distância de cor

Para descrever matematicamente a cor, foram criados diferentes espaços de cor. Para determinar os espaços de cor apropriados para a análise da diferença de cor, é necessário alterar a percepção desses espaços, e dos seus componentes tricromáticos, X, Y, Z, estimulando recetores visuais humanos. Esta alteração foi feita pela Comissão Internacional de Iluminação (CIE), calculando o estímulo de referência, por exemplo, L^* , a^* , b^* . A ideia era criar um espaço de cor linear, onde a distância entre pontos que definem cores individuais fosse proporcional à diferença perceptual entre elas e apresentar as cores com as coordenadas que descrevem um dos seus principais atributos [2].

Quando se pretende utilizar um sistema de visão computacional para a inspeção da cor de uma peça, é necessário calcular a diferença de cor entre a peça que está a ser analisada e uma peça que esteja dentro dos parâmetros estabelecidos como aceitáveis. A diferença de cor, ΔE , entre dois estímulos é calculada como a distância entre as coordenadas que as representam. A medida mais comum é a distância de Euclidiana.

No espaço de cores CIELAB, a cor é descrita por três componentes: L^* , luminosidade, que pode tomar valores entre 0 e 100, sendo que 100 significa que se encontra com o máximo de intensidade luminosa; a^* , representa a cor no campo verde e vermelho, e pode tomar valores entre -128 e +127, sendo que quando $a > 0$, a cor é mais avermelhada e quando $a < 0$, a cor é mais esverdeada; b^* , representa a cor no campo azul e amarelo, e pode tomar os mesmo valores que a^* , quando $b < 0$ a cor é mais azulada e quando $b > 0$ a cor é mais amarelada. Para $a^* = 0$ e $b^* = 0$, apenas existem tons de cinza [2].

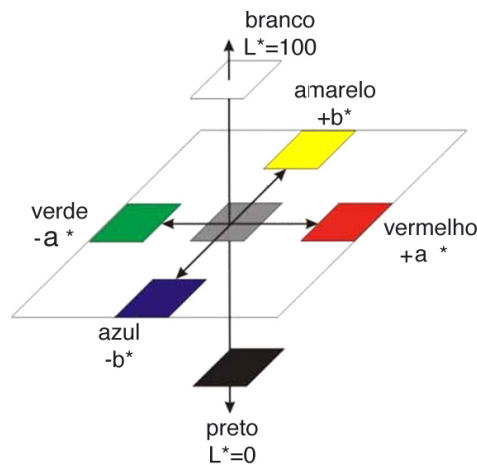


Figura 2.1: Espaço de cores $L^*a^*b^*$, [fonte: [7], 1970].

Este espaço é baseado no modelo de cor oposta, ou seja, as cores que sejam opostas ao longo do plano L^* , a^* e b^* não podem ser vistas simultaneamente. Apenas se vê cores escuras ou claras, vermelho ou verde, amarelo ou azul [2].

O cálculo da diferença de cor no espaço $L^*a^*b^*$, ΔE_{lab} , é dado pela Equação 2.1.

$$\Delta E_{lab} = \sqrt{(\Delta L^*)^2 + (\Delta a^*)^2 + (\Delta b^*)^2} \quad (2.1)$$

Para um observador *standard*, quando:

- $0 < \Delta E < 1$ - o observador não nota diferença na cor;
- $1 < \Delta E < 2$ - apenas um observador experiente nota diferença na cor;
- $2 < \Delta E < 3.5$ - um observador normal consegue notar diferença na cor;
- $3.5 < \Delta E < 5$ - a diferença entre as cores é notável;
- $5 < \Delta E$ - o observador distingue duas cores diferentes;

Contudo, o espaço CIELAB não é uniforme. É, por isso, recomendado pela ISO a substituição da fórmula do ΔE pela fórmula ΔE_{2000} , a qual é utilizada especialmente para a análise de cor em aplicações industriais [2].

A medida da diferença de cor é dada pela Equação 2.2 [2].

$$\Delta E_{2000} = \sqrt{\left(\frac{\Delta'_L}{K_L S_L}\right)^2 + \left(\frac{\Delta'_C}{K_C S_C}\right)^2 + \left(\frac{\Delta'_H}{K_H S_H}\right)^2} \quad (2.2)$$

No entanto, ainda que a fórmula tenha sido desenvolvida especificamente para o cálculo de distância de cor em aplicações industriais, esta apresenta uma maior complexidade na sua aplicação.

É também possível calcular a distância entre cores quando estas se encontram no espaço de cor RGB , usando a equação 2.3. No entanto, como este não é um espaço uniforme, os resultados não são tão precisos [2].

$$\Delta_{RGB} = \sqrt{3(\Delta_R)^2 + 4(\Delta_G)^2 + 2(\Delta_B)^2} \quad (2.3)$$

De notar, na equação os valores de Δ_R , Δ_G e Δ_B , são multiplicados por 3, 4 e 2, respetivamente, de modo a compensar a diferente sensibilidade do olho humano às alterações na intensidade das componentes R, G e B [2].

Em Pereira *et. al.* [8], é proposta uma nova estratégia que melhora a métrica do ΔE_{2000} , equação 2.2, quando é possível impor um valor de distância máxima. A estratégia proposta reduz o número de vezes que é necessário calcular a distância da cor, mantendo os mesmos resultados. Para tal, são combinadas diferentes propriedades que, juntamente com o limite de distância de cor, pode ser usado para reduzir o número de vezes que é calculada a ΔE_{2000} . As propriedades utilizadas são módulo, M , distância euclidiana entre dois vetores RGB , ângulo entre os dois vetores RGB , A , e a máxima diferença entre a componente de luminosidade entre duas cores no espaço $CIEL^*a^*b$, L . Os módulos são calculados em sequência e o ΔE_{2000} é apenas calculado em último recurso. Duas das sequências que apresentaram melhores resultados foram a MAL e a MA. O método proposto apresenta ganhos significativos, mostrando a dependência do limite de decisão no resultado do ΔE_{2000} .

2.2.3 Características de Textura

Devido ao seu potencial em extrair características proeminentes, a análise de textura tem sido amplamente utilizada em visão computacional e reconhecimento de padrões. Contudo, a extração das características de textura é um dos principais problemas quando se está a efetuar a análise de textura, e é por isso um tópico de pesquisa de longa data, em que se pretende compreender como

funciona o processo de reconhecimento de textura em humanos e qual a sua importância numa ampla variedade de aplicações baseadas em visão computacional e análise de imagem. Aplicações como detecção e reconhecimento facial, reconhecimento de objetos e cenários, detecção de peões, inspeção de alimentos, são alguns dos exemplos onde se pode explorar a análise de textura [9].

Grande parte das abordagens para a análise de textura quantifica as medidas de textura por valores únicos, como médias e variâncias. Esses valores são depois utilizados como elementos dos vetores de características na realização da classificação. Com este método de análise, muita informação importante contida nas distribuições de valores de características pode ser perdida. Alguns estudos sugerem que distribuições de ocorrências conjuntas de pares de características oferecem melhores resultados quando comparados com os resultados de distribuições de características únicas [10].

Em Ojala *et. al.* [10] são avaliadas algumas medidas de textura que foram usadas com sucessos em diversas aplicações. O método utilizado para a classificação é baseado na discriminação de Kullback da amostra e são utilizados protótipos de distribuições. A classificação do desempenho é avaliada para características únicas com valor de distribuição de característica unidimensional e para pares de características complementares com valores de distribuição de características bidimensionais. Para a experiência são utilizados dois tipos de dados diferentes, um conjunto de imagens Brodatz's e imagens utilizadas num estudo comparativo por Ohanian e Dubes. Neste estudos, as medidas de textura utilizadas são *Gray-level difference method*, *Laws' texture measures*, *Center-symmetric covariance measures*, *Local binary patterns* e *Complementary feature pairs*.

Dos métodos avaliados, o que teve um melhor desempenho, no geral, foi o *Gray-level difference method*, este discriminou a maioria das texturas de forma bastante eficaz, sendo também um método fácil de implementar. Outro método que apresentou uma boa performance foi a medida de textura baseada em *Local binary patterns* (LBP), sendo este também fácil de implementar a nível computacional. Os LBP são invariantes em escala de cinza e podem ser combinados com uma medida de contraste, simples, para ser ainda mais eficaz. O facto de ser invariante em escala de cinza é bastante importante, quando aplicados em ambientes industriais, devido à iluminação irregular.

A matriz de co-ocorrência de escala de cinza, é um método comum para descrever a textura, estudando as propriedades de correlação espacial da escala de cinza. Reflete as informações abrangentes dos níveis de cinza da imagem sobre a direção. Estas informações podem ser usadas para analisar as primitivas e o arranjo da estrutura da imagem [11] [12] [13].

Os LBPs expressam a relação entre o ponto da vizinhança local e o ponto central através de bits binários [14]. É um método bastante robusto a mudanças

na imagem em escala de cinza causadas pela iluminação [15] [16] [17]. Jafar J. Abukhait [18] propôs um sistema automatizado para a identificação de diferentes defeitos de superfícies. A classificação é feita utilizando um *Support Vector Machine* (SVM), de acordo com as características da textura da superfície. Para a extração de características, é apresentada uma nova técnica que faz a extração de características locais e globais. A combinação de LBP com GLCM, pretende lidar com diferentes tipos de defeitos de superfícies. O sistema proposto apresentou uma precisão de 96.9% para imagens de telhas cerâmicas e 94% para superfícies de madeira.

2.2.4 Características de Forma

A forma é uma característica visual importante, e uma das características base para descrever o conteúdo de uma imagem. Contudo, a representação e descrição da forma não é fácil, pois quando um objeto real 3D é projetado para um plano da imagem 2D, uma dimensão de informação do objeto é perdida. Assim é possível afirmar que a forma extraída da imagem, apenas representa parcialmente o objeto projetado. A forma é também facilmente corrompida por ruído, defeitos, distorção arbitrária e oclusão [19].

A representação da forma, por norma, procura características eficazes e perceptivamente importantes com base nas informações do contorno ou contorno mais interior. Têm sido utilizadas várias características como assinatura de forma, histograma de assinatura, invariantes de forma, momentos, curvatura, contexto de forma, matriz de forma e características espetrais. A maneira mais comum para avaliar as características de forma em recuperar formas semelhantes de uma base de dados, e a precisão do seu desempenho, porém este método de avaliação não é suficiente, isto porque são ignoradas outras características consideradas importantes na representação de forma. MPEG-7, definiu vários parâmetros para medir um descritor de forma, são este, boa precisão de recuperação, características compactas, aplicação geral, baixa complexidade computacional, desempenho de recuperação robusto e representação hierárquica de grosso para fino [19] [20].

Quanto à classificação das técnicas para a representação e descrição da forma, geralmente estas são classificadas em duas classes de métodos, métodos baseados no contorno e métodos baseados na região. Em cada classe, estes métodos podem ainda ser divididos em abordagens estruturais ou globais, sendo ainda possível dividir estas abordagens no domínio espacial ou no domínio das transformadas, Figura 2.2 [19].

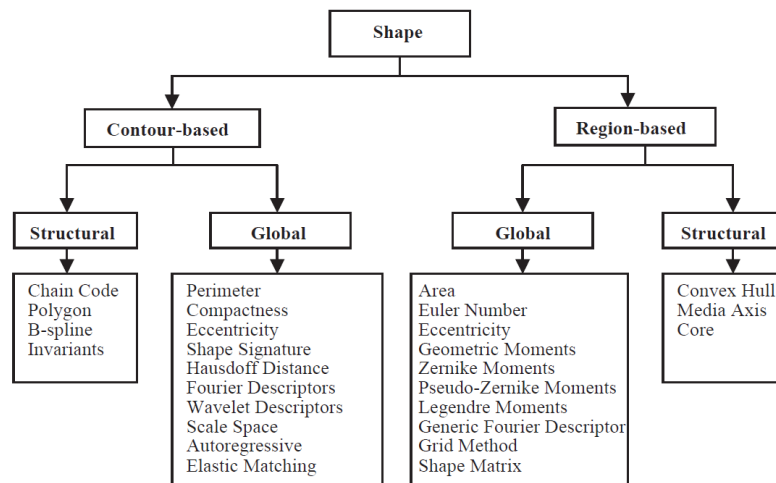


Figura 2.2: Classificação de técnicas para representação e descrição de forma, [fonte: [21], 2].

Nas técnicas baseadas na região, todos os pixels da imagem são contabilizados para obter a representação da forma. Normalmente os métodos que têm por base a região usam descritores de momento para descrever a forma. Os métodos globais tratam a forma como um todo, resultando num vector numérico característico da forma que pode ser utilizado para descrever a forma. Para verificar a dissimilaridade entre formas, é calculada a distância entre os seus vectores de características.

Um método global, para descrever a forma, consiste no cálculo de momentos geométricos invariantes. Hu publicou o primeiro artigo relevante acerca do uso dos momentos invariantes da imagem para aplicações de reconhecimento de padrões de duas dimensões [22]. Os momentos de Hu, ou momentos invariantes de Hu, são um conjunto de sete números calculados através dos momentos centrais, os quais são invariantes às transformações da imagem. Os primeiros seis momentos são invariantes à translação, escala, rotação e reflexão, enquanto que o sétimo momento muda para a reflexão da imagem [23]. Estes momentos são calculados através das seguintes fórmulas:

$$\begin{aligned}
h_1 &= \eta_{20} + \eta_{02} \\
h_2 &= (\eta_{20} - \eta_{02})^2 + 4\eta_{11}^2 \\
h_3 &= (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} - \eta_{03})^2 \\
h_4 &= (\eta_{30} + 3\eta_{12})^2 + (\eta_{21} + \eta_{03})^2 \\
h_5 &= (\eta_{30} - 3\eta_{12})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] \\
&\quad + (3\eta_{21} - \eta_{03})[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] \\
h_6 &= (\eta_{20} - \eta_{02})[(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2 + 4\eta_{11}(\eta_{30} + \eta_{12})(\eta_{21} + \eta_{03})] \\
h_7 &= (3\eta_{21} - \eta_{03})(\eta_{30} - \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] \\
&\quad - (\eta_{30} - 3\eta_{12})(\eta_{21} + \eta_{03})[(3\eta_{30} - \eta_{12})^2 - (\eta_{21} + \eta_{03})^2]
\end{aligned} \tag{2.4}$$

em que η_{ij} representam os momentos centrais normalizados.

Os momentos de Hu são considerados um método numérico, global e sem preservação de reconhecimento de forma, pois usa todos os pixels da imagem e não é possível recuperar a imagem original [24] [25].

Wei Jie Chen *et. al.* [26], apresentam um método para detecção e classificação de defeitos durante o processo de impressão 3D. Os momentos de Hu são utilizados para a extração de características na imagem binária, sendo depois calculada a distância de forma para a detecção dos erros. O método mostrou ter uma taxa de sucesso de 68.88% e 17.39% de taxa de falsos alarmes. Em Hongbo Mu *et. al.* [27], os momentos de Hu são utilizados para a extração de características das imagens com defeito, que são depois inseridos numa rede neuronal para a identificação do defeito. Os resultados mostraram que era possível detetar e classificar erros de madeira tendo por base os momentos invariantes de Hu.

2.2.5 Medição de similaridade

A medição de similaridade é o processo de encontrar a semelhança ou diferenças entre duas ou mais imagens recorrendo às suas características. O uso de medidas de distância é dos métodos mais utilizados para realizar a medição de similaridade.

Das medidas de distância existentes é possível destacar algumas, uma vez que são as mais utilizadas. Por exemplo, a distância Minkowski-Form, Euclidiana e Manhattan.

2.2.5.1 Distância Minkowski-Form

A distância Minkowski-Form é a métrica mais usada para recuperação de imagem. Dados dois vetores de características f_1 e f_2 de N classes, o cálculo da distância Minkowski-Form é dado pela equação 2.5.

$$D(f_1, f_2) = \left(\sum_1^N |f_1(i) - f_2(i)|^p \right)^{1/p} \quad (2.5)$$

Nesta métrica, cada dimensão do vetor de características de imagem é independente das restantes e de igual importância. A cada valor parâmetro, p , é possível atribuir uma distância. Quando $p=1$, a métrica Minkowski-Form corresponde à distância de Manhattan, $p=2$, corresponde à métrica da distância Euclidiana, para $p=\infty$ é denominada de distância Chebyshev.

2.2.5.2 Distância Euclidiana

A distância Euclidiana é a métrica mais comum para medir distâncias entre dois vetores. Esta métrica pode ser utilizada quando os elementos do vetor de características da imagem são igualmente importantes e independentes. É possível descrever esta métrica como a distância normal entre dois valores. O cálculo da distância Euclidiana é dado pela equação 2.6.

$$D = \sqrt{\sum_{k=1}^n (p_k - q_k)^2} \quad (2.6)$$

onde n é o comprimento do vetor de características e D é a distância entre os dois vetores. A distância Euclidiana é a métrica mais fácil para calcular e simples de implementar com baixo nível de complexidade.

2.3 Classificação de Defeitos

Para identificar as categorias de defeito numa imagem, é necessário que as características não descrevam apenas a imagem, mas também distingam diferentes categorias de imagem. O principal objetivo da classificação de defeitos, é treinar o classificador de acordo com as características extraídas e, de seguida, identificar o tipo de erro na superfície corretamente, tendo por base métodos de reconhecimento de padrões supervisionados em não supervisionados.

O *Support Vector Machine* (SVM), é um classificador para reconhecimento de padrões supervisionado. Este é adequado quando o número de dados não é muito elevado e para classificações não lineares de alta dimensão, sendo bastante utilizado para sistemas de inspeção automática em ambientes industriais. Em Jia *et. al.*, [28], é descrito um sistema de visão em tempo real que usa SVM para automaticamente, aprender padrões de defeitos complicados. Li *et. al.* [29], propôs um método de classificação para padrões de defeitos binários, utilizam SVM para classificar e identificar defeitos. Este juntou dois métodos, um supervisionado, SVM e um não supervisionado, *Self-Organizing Maps* (SOM). Os resultados apresentados mostram uma precisão de 90%.

2.4 Exemplo Detecção de Defeitos em Superfícies de Cimento

Este método foi proposto por Yusuke Fujita *et. al.* [30] para a detecção de fissuras em superfícies de cimento. O método inclui duas etapas de pré-processamento e duas etapas de detecção.

Na primeira etapa de pré-processamento, a imagem adquirida é filtrada, através de um filtro mediana, de forma a reduzir pequenas variações como sombras. Se, por sua vez, o tamanho do filtro for muito superior à largura da fenda, estas são removidas da imagem filtrada. De seguida, é realizada a subtração da imagem original com a imagem filtrada, sendo o resultado uma imagem onde apenas as fissuras da superfície são visíveis. Na Figura 2.3, é possível visualizar os resultados da primeira etapa de pré-processamento. Na Figura 2.3c, a imagem subtraída foi invertida a nível de intensidade [30].

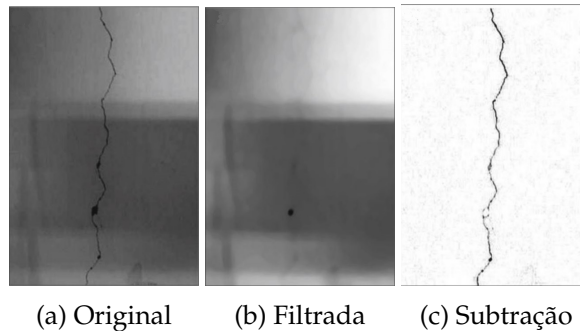


Figura 2.3: Primeira etapa de pré-processamento, [fonte: [30], 247].

A segunda etapa de pré-processamento, tem como objetivo, realçar as fissuras. Para tal, é utilizado um filtro de linha multi-escala com a matriz de Hessian. A matriz de Hessian é utilizada para a medição de similaridade de uma linha clara que descreve estruturas de segunda ordem de variações de intensidade local, na vizinhança de cada píxel, em imagens bidimensionais [30]. Após as etapas de pré-processamento, é utilizado o relaxamento probabilístico para a detecção de fissuras evitando ruídos. Aos resultados do processo de relaxamento, é aplicado um limiar para uma detecção mais precisa das fissuras [30].

2.5 Fabrico Aditivo

Impressão 3D é o processo de criar objetos físicos de três dimensões de um modelo CAD, adicionando material camada-a-camada. Este processo é denominado de fabrico aditivo [31]. Por outro lado, o método de fabrico denominado de fabrico subtrativo é o processo de remoção de material de um bloco sólido

para a criação do objeto 3D [32]. Existem diversos materiais que podem ser utilizados no fabrico aditivo, tais como, papel, filamento em pó, metal, líquidos e termoplásticos, como o *polylactic acid* (PLA). A escolha do material depende do método de fabrico aditivo a ser usado.

2.5.1 Métodos de Fabrico Aditivo

Têm sido desenvolvidos novos métodos de fabrico aditivo, de modo a conseguir acompanhar a procura de impressão de estruturas complexas, sendo capazes de imprimir grandes estruturas, reduzindo defeitos de impressão e melhorando propriedades mecânicas. O método mais comum na impressão 3D é o *fused deposition modelling* (FDM), existindo outros métodos como o *selective laser sintering* (SLS), *selective laser melting* (SLM), que utilizam pós para a impressão, *liquid binding in three-dimensional printing* (3DP), impressão a jato de tinta, elaboração de contorno, estereolitografia, deposição de energia direta (DED) e fabricação de objeto laminado (FOL) [33].

2.5.1.1 Fused Deposition Modelling

FDM é um método de fabrico aditivo que consiste na extrusão de material, para a criação de várias formas e estruturas complexas. Este método tem por base o conceito de derreter a matéria prima e moldá-la para construir novas formas. O material mais utilizado neste método são filamentos termoplásticos, como o PLA. No FDM, o filamento é colocado num rolo, puxado por uma roda motriz, e depois colocado na cabeça do bico de extrusão, a uma temperatura controlada, sendo então o filamento aquecido para semilíquido, Figura 2.4. A cabeça de extrusão está ligada a um sistema de três ou mais eixos, que permite que esta se movimente nas direções X, Y e Z e suas rotações, permitindo que o material extruído seja posicionado camada por camada, nas localizações predeterminadas onde arrefece e solidifica. O processo de arrefecimento de material pode ser acelerado por ventiladores de arrefecimento acoplados à cabeça de extrusão. Este processo segue os contornos da camada especificado pelo programa, normalmente CAD, que é inserido no sistema de trabalho do FDM [34].

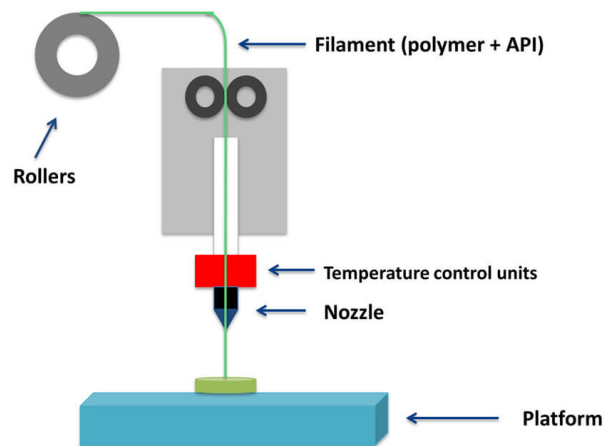


Figura 2.4: Fused Deposition Modelling, [fonte: [35], 2021].

2.5.2 Tipos de Erros no Fabrico Aditivo

Apesar das diversas vantagens do fabrico aditivo, tais como a redução da restrição geométrica, redução do *lead time* de produção, nos protótipos, flexibilidade do material utilizado, conforme o produto que se pretende imprimir, redução de desperdício e flexibilidade de personalização, este método é limitado para atingir ou garantir uniformidade das propriedades mecânicas no entanto formas das peças a serem impressas, especialmente quando estas são produzidas em massa.

Na Tabela 2.1, estão apresentados alguns dos defeitos mais comuns na impressão 3D de peças, que podem comprometer a qualidade das peças produzidas utilizando esta tecnologia [36].

Tabela 2.1: Tipos de defeitos comuns no fabrico aditivo e correspondentes causas e consequências [37]

Tipos de Defeitos	Causas	Consequências
Subextrusão	A impressora não extrude plástico suficiente	Lacunas entre perímetros e preenchimento
Excesso de extrusão	A impressora expulsa demasiado material	Excesso de material, dimensões externas incorretas
Superaquecimento	Processo lento de arrefecimento do filamento	Deformações na peça
Camadas desalinhadas	Fatores externos que podem alterar a posição da cabeça de extrusão	Camadas desalinhadas
Delaminagem	A camada a imprimir não fixa à camada imediatamente abaixo	Separação de camadas
Bolhas ou espinhas	Constante liga e desliga da impressora para se movimentar para novas posições	Marcas na superfície da peça
Precisão dimensional	Subextrusão, superextrusão, contração térmica, qualidade do filamento	Peças com dimensões erradas

Capítulo 3

Arquitetura

Um sistema inspeção visual automática, convencional, é constituído por três módulos, iluminação, aquisição de imagem e o algoritmo para processamento de imagem e deteção de defeitos. Este trabalho tem como principais objetivos, o desenvolvimento de um sistema de inspeção automática para avaliação da qualidade do fabrico aditivo, recorrendo ao software da Beckhoff, TwinCAT Vision. Neste capítulo, será apresentada a arquitetura do sistema para a aquisição de imagens, o tipo de iluminação utilizada no mesmo e por fim, apresentado o software, TwinCAT Vision e o modo como é estabelecida a ligação entre a câmara e o software.

3.1 Hardware

O sistema de aquisição utilizado neste projeto seguiu os parâmetros mais convencionais de um sistema de inspeção automática. O sistema é composto por um braço totalmente ajustável, ao qual a câmara foi acoplada, de modo a ser possível ajustar da melhor forma o *Field of View* (FOV). As peças encontram-se ainda apoiadas num suporte, de modo a que o ângulo entre elas e a câmara seja 90°, para uma melhor visualização da superfície da peça. A Figura 3.1 apresenta a montagem que permitiu a aquisição das imagens.

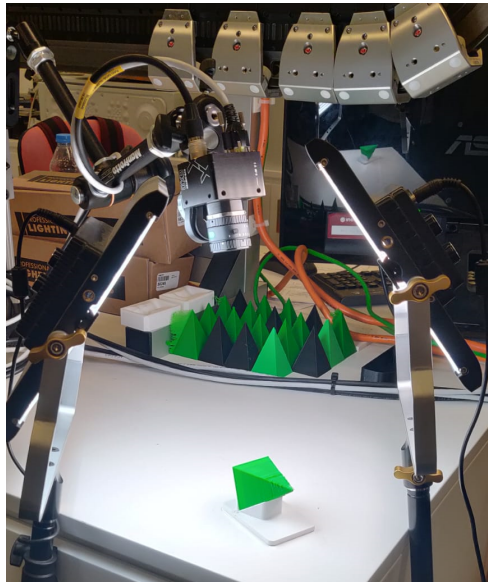


Figura 3.1: Montagem do sistema

3.1.1 Câmera

A câmera utilizada para a captura foi fornecida pelo INEGI, denominando-se SVCAM exo264CGE, e está representada na Figura 3.2. Esta possui uma resolução de 2448x2048 pixels, e o sensor existente na câmera possui uma taxa máxima de 24.5 fps . Possui, ainda, uma interface GigE Vision, interface *standard* para processamento industrial de imagens, facilitando a configuração e a operação de câmeras industriais. Algumas das vantagens da interface *GigE Vision* são, a taxa de transmissão, *default*, de 1 Gigabits/s; a possibilidade de utilizar cabos com comprimento até 100 metros, sem ser necessário extensão e o uso de interfaces *standard* para *Gigabit Ethernet*, quer de *hardware* quer de *software*. A comunicação com dispositivos compatíveis é feita via *User Datagram Protocol/Internet Protocol (UDP/IP)* , consistindo em quatro elementos [38]:

- O *GigE Vision Protocol (GVCP)* , que define como os dispositivos GigE Vision devem ser endereçados. Especifica também os canais de dados e os mecanismos para a transferência de imagens e configuração de dados entre o PC e a câmera.
- O *GigE Vision Stream Protocol (GVSP)* , que especifica o tipo de dados e métodos de transmissão usados para a transferência de imagens da câmera para o PC, estando disponível um *Packet Resend* para a correção de erros de transmissão.
- O *GigE Device Discovery Mechanism (GDDM)* , que permite encontrar câmeras na rede utilizando *search queries*.

- Um ficheiro XML que contém uma descrição que define todas as funções públicas da câmara.



Figura 3.2: SVCAM exo264CGE, [fonte: [38], 2020].

3.1.2 Iluminação

A iluminação é um dos factores mais importantes para ser possível adquirir uma imagem com boa qualidade para análise. A principal função da iluminação é ultrapassar a interferência da luz ambiente, garantir a estabilidade da imagem e obter imagens com alto contraste. Em suma, o seu principal objetivo é salientar as principais características da peça e reduzir as características indesejadas, tais como sombras.

Existem diversos tipos de iluminação, tais como luz direcional, luz difusa, iluminação traseira e iluminação estroboscópica. A luz direcional é um tipo de luz que é utilizado quando é necessário especificar uma direção, assim os objetos são iluminados de acordo com a mesma, como se ilustra na Figura 3.3

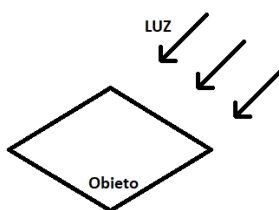


Figura 3.3: Exemplo de iluminação direcional

Para a luz difusa, apenas é necessário fazer com que a luz não incida diretamente no objeto. Uma das maneiras de utilizar a luz difusa num ambiente controlado é apontar a luz para um objeto branco, para que incida nele e seja refletida para o objeto a fotografar.

Apesar de a iluminação direcional ser mais propensa a causar reflexos nas imagens das superfícies esta foi a iluminação escolhida e utilizada para o trabalho. De modo a combater esses reflexos optou-se por utilizar dois iluminadores colocados em lados opostos da peça, este método ajudou na redução dos reflexos, apesar de não os eliminar na sua totalidade. Os iluminadores utilizados foram os LED, JYLED-500s, da NEEWER (Figura 3.4). Estes encontram-se posicionados em lados opostos da câmara, como se mostra na Figura 3.1. Não houve processo de ponderação e comparação com outros diferentes tipos de iluminadores. Uma vez que estes se encontravam disponíveis e possuem as características necessárias para o sistema, tais como altura ajustável, regulação de luz e controlo do sentido de iluminação, estes foram os iluminadores escolhidos. Ainda que a sua localização não seja a mais favorável, pois existem fatores externos que afetam a qualidade da iluminação da peça, estas foram colocadas em lados opostos da peça.



Figura 3.4: Iluminadores LED, JYLED-500s, [fonte: [39], 2020].

3.1.3 PC BECKHOFF C6930

Para este projeto foi utilizado o PC industrial fornecido pelo INEGI e denominado C6930, apresentado na Figura 3.5, desenvolvido pela BECKHOFF. O C6930 é uma plataforma de alto desempenho, próprio para aplicações altamente complexas como o uso do *software* de automação TwinCAT. Equipado com o processador Intel® Core™ i7-4700EQ 2.4 GHz, 4 cores, memória de 2 GB DDR3L RAM, duas entradas *ethernet* 100/1000BASE-T, onde é estabelecida a ligação com a rede local e ligação com I/O, neste caso com as câmaras [40].



Figura 3.5: PC industrial C6930, [fonte: [40], 2020].

3.2 Software

O *software* utilizado foi o TwinCAT permite que quase todos os sistemas baseados em PC se tornem num sistema de controlo em tempo real, com vários sistemas a serem executados em paralelo. O TwinCAT 3 é o resultado do desenvolvimento sistemático do TwinCAT 2. Deste *software* é possível destacar várias características como o uso de apenas um *software* para a programação e configuração, a integração do Visual Studio[®], maior liberdade na escolha da linguagem de programação, suporte para a extensão orientada a objetos da IEC 61131-3, uso das linguagens de programação C/C++ para aplicações em tempo real e a integração com o MATLAB[®]/Simulink[®], permitindo assim o uso de uma ferramenta padrão em aplicações científicas e de medição, desenvolvimento, simulação e otimização de malhas de controlo complexas e geração automática de código [41].

O sistema TwinCAT é constituído por uma camada de transporte que permite a troca de dados entre os diferentes módulos do *software*, o *Automation Device Specification* (ADS), como se mostra na Figura 3.6. Caso seja necessário comunicar com outro PC ou dispositivo, o protocolo ADS é executado em cima do protocolo TCP/IP, e é ainda capaz de ser executado em cima do protocolo UDP/IP [42].

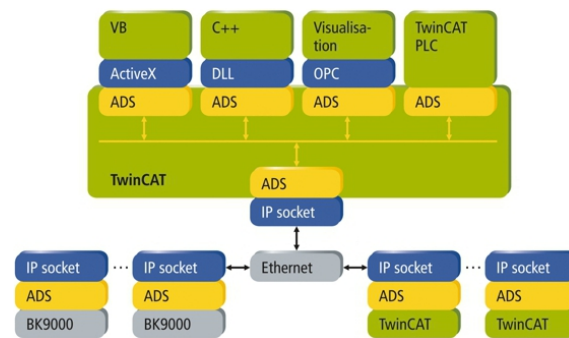


Figura 3.6: Protocolo de comunicação ADS, [fonte: [42], 2020].

O TwinCAT Vision, veio incorporar processamento de imagem, a uma plataforma de controlo universal que incorpora PLC, controlos de movimento, robótica, IoT, HMI e muitos mais. O sistema permite que algoritmos de processamento de imagem sejam executados em tempo real e sincronizados com outros tipos de tarefas como tarefas de robótica, eliminando as latências e interfaces adicionais, o que leva a uma optimização significativa dos processos e a um aumento na eficiência [43].

A ligação entre a câmara e o PLC é feita através do protocolo UDP/IP. Para que seja possível adicionar a câmara ao sistema, é necessário que ambos se encontrem na mesma rede. Quando é feita a seleção da câmara a adicionar, devem ser habilitados o comunicador ADS, para ser possível guardar *streams* e o módulo *Image Provider*, pois este é a interface para as funções correspondentes do PLC, Figura 3.7. Com a ligação estabelecida corretamente, será criado o módulo da câmara, no qual é possível verificar informações gerais acerca da câmara e fazer a configuração da mesma, como, por exemplo, a calibração como se mostra na Figura 3.8.

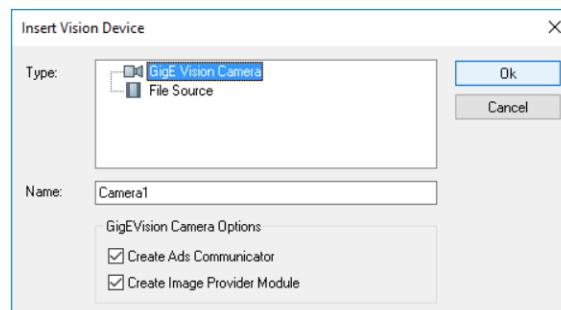


Figura 3.7: *Insert vision device*, [fonte: [44], 2020].

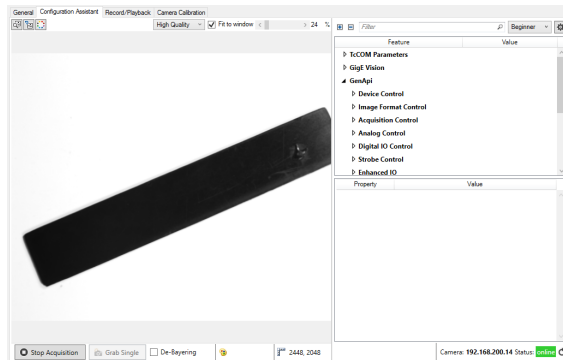


Figura 3.8: Camera configuration assistant

3.2.1 ADS Image Watch

O *ADS Image Watch* permite que as imagens guardadas no PLC, sejam exibidas no ambiente de desenvolvimento. Para selecionar a imagem que se pretende exibir deve ser especificado o sistema alvo, o porto do PLC, o nome da variável, do tipo *ITcVnDisplayableImage* e o dimensionamento da imagem que por *default* está definido o *Fit to Window* [45], na Figura 3.9, está representada o exemplo de uma imagem a ser exibida no *ADS Image Watch*.



Figura 3.9: ADS Image Watch

Capítulo 4

Sistema Desenvolvido

Neste capítulo será apresentado o sistema desenvolvido. Numa primeira parte será apresentado o sistema global para a classificação da peça. De seguida, serão explicadas detalhadamente as diferentes etapas, desde a aquisição de imagens até à localização de defeitos. Serão especificados os diferentes métodos de processamento de imagem utilizados, assim como uma explicação teórica dos mesmos.

4.1 Sistema Para a Classificação de Peça

Na Figura 4.1 estão representadas as diferentes etapas para a classificação de uma peça pelo sistema de inspeção desenvolvido. Numa primeira fase, a imagem adquirida é pré-processada, realizando-se o recorte da mesma e, de seguida, é feita a sua binarização. Em paralelo, a imagem de referência é também binarizada, separando-se o fundo da superfície. Após o pré-processamento da imagem, é efetuada a análise da cor da superfície. Esta é realizada convertendo as imagens da superfície da peça do espaço de cor RGB para o espaço de cor CIEL*a*b*, calculando-se de seguida a distância de cor entre as superfícies, recorrendo à fórmula do ΔE_{2000} . Caso o valor da distância seja superior a $T_{\Delta E_{2000}}$, a peça é rejeitada de imediato. Por outro lado, se a distância for inferior a $T_{\Delta E_{2000}}$ a peça passa para a próxima etapa. Numa segunda fase, é avaliada a qualidade da superfície para determinar se esta apresenta defeitos. Para a análise de superfície, é utilizado o detetor de contornos de *Canny*, para ser possível obter uma imagem binária característica da superfície. Este operador é também aplicado à imagem da peça de referência. Segue-se o cálculo dos momentos de Hu para ambas as imagens binárias. Estes são comparados entre si através do cálculo da dissimilaridade, usando apenas a diferença máxima entre as características individuais. Se o valor de dissimilaridade for menor que T_{Diss} , então a peça é

considerada de boa qualidade, se o valor for superior então a peça passa para a próxima fase, onde são localizados os defeitos, antes da peça ser rejeitada.

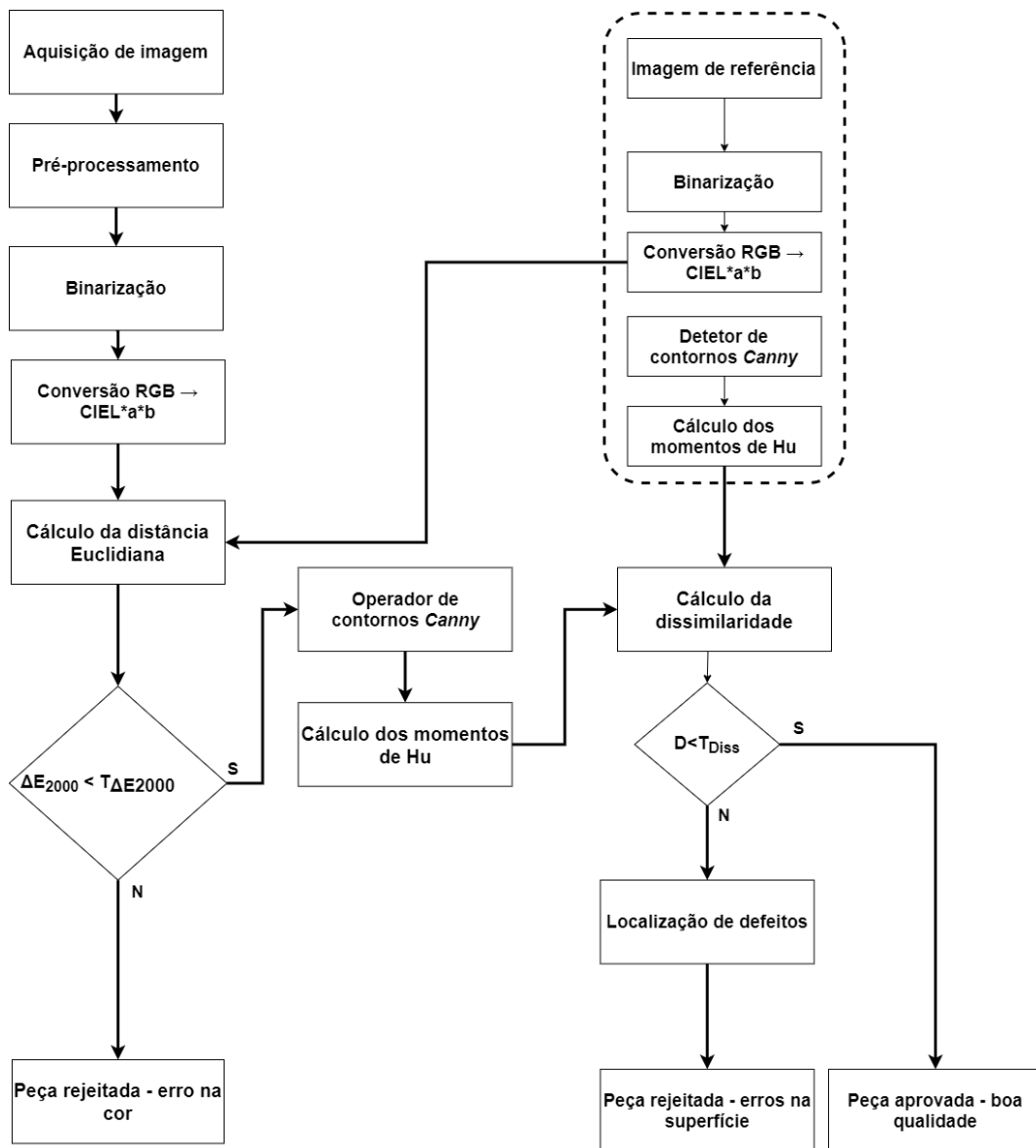


Figura 4.1: Sequência de operações para a classificação de uma peça

4.2 Aquisição de Imagens

A fase de aquisição de imagens é uma fase bastante importante em qualquer trabalho relacionado com a inspeção visual automática. Nesta fase, os dados de entrada são fornecidos ao sistema computacional, através de um sistema físico, para serem processados. A correta calibração da câmara é importante para ser

possível obter uma imagem com a melhor qualidade possível.

4.2.1 Calibração

A calibração é o processo de determinação dos parâmetros que caracterizam o sistema, o ambiente em que este está inserido e o modo como vai ser utilizado. A determinação incorreta destes parâmetros pode levar a maus resultados [46].

A calibração permite fazer uma correspondência precisa entre os pontos do espaço tridimensional e os pixels da imagem. Esta é composta por duas fases, a determinação dos parâmetros intrínsecos e extrínsecos [46].

Os parâmetros intrínsecos são calculados de modo a ser possível relacionar as coordenadas do pixel na imagem, com as coordenadas de pontos do espaço, medidos no sistema referencial com a origem no centro da câmara. São parâmetros que dependem exclusivamente das características físicas da câmara, como a sua geometria interna ou o tipo de lente [46].

Os parâmetros extrínsecos representam a posição e orientação do referencial de coordenadas da câmara relativamente a um outro sistema de coordenadas. Por vezes, a determinação dos parâmetros extrínsecos não é necessária, caso seja possível trabalhar unicamente com as coordenadas da câmara [46].

Para se fazer a conversão de coordenadas tridimensionais de um sistema de eixos coordenados para outro é necessário realizar uma transformação linear. Na transformação das coordenadas de um ponto no espaço, $XX=(X;Y;Z)$, para um sistema referencial do mundo de coordenadas do referencial da câmara, $XX_c=(X_c;Y_c;Z_c)$, é apenas necessário conhecer a matriz de rotação R e o vetor de translação T , responsáveis pela transformação do referencial do mundo no referencial da câmara [46].

$$XX_c = R * XX + T \quad (4.1)$$

A matriz R , uma matriz 3x3, pode ser determinada a partir do conhecimento dos ângulos que o referencial do mundo tem de girar em torno de cada um dos seus eixos para se transformar num referencial idêntico ao da câmara a menos de uma deslocação, T . T é um vetor coluna de 3 elementos [46].

Para determinar os parâmetros intrínsecos, é necessário relacionar as coordenadas 3D do espaço com a sua projeção no plano da imagem. Com as coordenadas da câmara, $XX_c=(X_c;Y_c;Z_c)$, de uma série de pontos do espaço, é possível determinar as coordenadas normalizadas desses mesmos pontos. As coordenadas normalizadas são obtidas dividindo as duas primeiras componentes pela terceira, tendo em conta que o eixo Z_c é perpendicular à superfície da lente, equação 4.2. Deste modo, todos os pontos de um raio projetante, com origem no centro da lente, terão as mesmas coordenadas normalizadas [46].

$$x_n = \begin{bmatrix} X_c/Z_c \\ Y_c/Z_c \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix} \quad (4.2)$$

Com os resultados obtidos da calibração, é possível estimar a distorção da lente que irá afetar estes pontos normalizados. Por norma, a distorção radial é superior à distorção tangencial. Quanto maior for a distorção normalizada dos pixels ao ponto principal normalizado, $x_n = (0;0)$, maior será a distorção [46].

O vector da distorção tangencial, dx , é dado pela equação 4.3.

$$dx = \begin{bmatrix} 2 * kc_t(1) * x * y + kc_t(2) * (r^2 + 2 * x^2) \\ kc_t(1) * (r^2 + 2 * y^2) + 2 * kc_t(2) * x * y \end{bmatrix} \quad (4.3)$$

Onde, $kc_t(1)$ e $kc_t(2)$, são os coeficientes da distorção tangencial e $r^2 = x^2 + y^2$.

Para o cálculo das coordenadas normalizadas com distorção x_{dn} é utilizada a equação 4.4.

$$x_{dn} = \begin{bmatrix} x_d \\ y_d \end{bmatrix} = (1 + kc_r(1) * r^2 + kc_r(2) * r^4) * x_n + dx \quad (4.4)$$

Onde, $kc_r(1)$ e $kc_r(2)$, são coeficientes da distorção radial.

Com as coordenadas normalizadas com distorção calculadas, é possível obter as coordenadas do pixel, através da equação 4.5.

$$\begin{bmatrix} x_p \\ y_p \\ 1 \end{bmatrix} = KK * \begin{bmatrix} x_d \\ y_d \\ 1 \end{bmatrix} \quad (4.5)$$

KK , é a matriz de transformação da câmara. Esta, permite converter coordenadas normalizadas, em coordenadas pixel. Depende dos valores estimados para a distância focal da câmara, fc , e do ponto principal, cc , ambos expressos em pixels [46]. É possível calcular a matriz de transformação da câmara, através da equação 4.6.

$$KK = \begin{bmatrix} fc_x & 0 & cc_x \\ 0 & fc_y & cc_y \\ 0 & 0 & 1 \end{bmatrix} \quad (4.6)$$

O software da Beckhoff, TwinCAT3 Vision, disponibiliza aos seus utilizadores uma variedade de padrões de calibração, sendo possível realizar a calibração da câmara com ajuda dos padrões de calibração e da função *Camera Calibration*. Para este projeto foi utilizado o padrão de círculos assimétrico, apresentado na

Figura 4.2, sendo que qualquer outro dos padrões disponíveis teria o mesmo resultado na calibração.

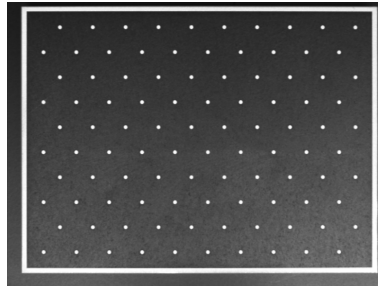


Figura 4.2: Padrão de calibração - círculos assimétricos, [fonte: [47], 2020].

Os dados necessários para a função *Camera Calibration* são: tipo de padrão; comprimento, que diz respeito ao número de círculos que existem numa linha; largura, número de linhas existentes; distancia X, especifica a distância mais curta entre dois círculos na horizontal; Y, distância entre duas linhas; se a imagem apresenta objetos pretos num fundo branco ou vice-versa; e, também, o ponto zero da origem extrínseca. Com esses dados o *software* faz o cálculo de forma autónoma, retornando depois a matriz de transformação da câmara, os coeficientes de distorção, a matriz de rotação e o vetor de translação e, ainda, um valor de erro de projeção, indicando o resultado da calibração dos parâmetros intrínsecos. Esses valores são guardados no *GigE Vision Image Provider* da câmara que se encontra no módulo de *Vision* do *Solution Explorer*, Figura 4.3.

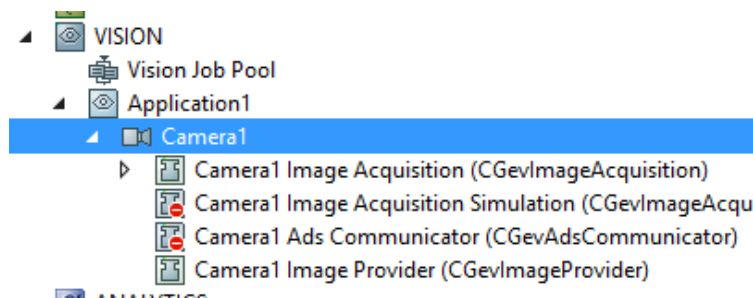


Figura 4.3: Módulo vision

De modo a ser possível aceder aos parâmetros de calibração, foi criada uma função, *FBCalibrationParams*, que tem o objetivo de retornar os valores guardados no *GigE Vision Image Provider*, antes de ser feita a aquisição de imagem, de modo a que as imagens adquiridas tenham a correta calibração. Para tal, é preciso inicializar a câmara com a função *FB_VN_GevCameraControl*, pois esta permite as funcionalidades básicas para controlar a *GigE Camera* e aceder aos dados de calibração.

4.3 Processamento de Imagem

4.3.1 Recorte da Imagem

Antes de se iniciar o processo de análise da peça, é necessário realizar um recorte da imagem com o objetivo de eliminar, o máximo possível, o fundo da imagem a analisar — este processo é denominado de *cropping*. Este passo é importante, uma vez que o fundo pode interferir com a fase de comparação entre as peças, ao apresentar diferenças de contraste ou ruído. Assim, apenas a peça será, de facto, analisada. O recorte da imagem foi realizado com a função *F_VN_SetRoi* que define uma região de interesse. Esta função tem como parâmetros de entrada: a posição inicial, x , y , de onde será definida a região de interesse; o comprimento e largura dessa região; e, a imagem de destino. Na Figura 4.4 é possível visualizar o resultado do *cropping*.

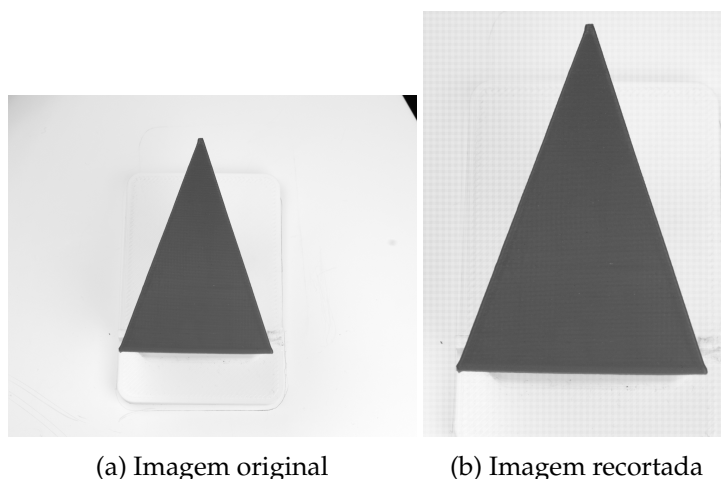


Figura 4.4: Resultado do recorte da imagem

4.4 Binarização

A etapa de binarização, tem como objetivo separar o fundo da imagem da peça. Neste projeto, foi utilizado o método de binarização automática através da comparação com um valor de limiar, recorrendo-se ao método de Otsu para determinar automaticamente o valor ideal do limiar.

O método de binarização por limiar, consiste na comparação entre o valor do pixel da imagem e o limiar de decisão. Neste método, é atribuído um valor de limiar, compreendido entre 0 e 255, uma vez que se trata de uma imagem com apenas um canal. Por norma, esta encontra-se em escala de cinzas. Quando o valor do pixel a ser analisado for superior ao valor do limiar, este toma o valor

máximo de 255, e caso seja inferior irá tomar o valor 0. Este processo pode sofrer alterações de acordo com o objetivo final para o qual está a ser aplicado [48].

Sendo que este método, tem como base um valor de limiar, é importante encontrar o valor mais adequado, uma vez que o valor definido irá afetar o resultado final da binarização. Um dos métodos possíveis para encontrar automaticamente o valor de limiar mais indicado, é o método de Otsu.

O método de Otsu processa o histograma de uma imagem, em tons de cinza, selecionando como valor do limiar aquele que minimiza a soma da variância intraclases. O cálculo da variância intraclases é feito através da equação 4.7, onde W corresponde ao peso de cada classe, que é a probabilidade que um pixel possui de pertencer à classe b , *background* e f , *foreground* [49] [50].

$$\sigma_W^2 = W_b\sigma_b^2 + W_f\sigma_f^2 \quad (4.7)$$

No entanto, o cálculo da variância, é muito dispendioso a nível computacional, uma vez que exige também o cálculo dos valores das médias e dos pesos correspondentes para cada classe, sendo este realizado para todos os valores de limiar possíveis. Otsu demonstrou que é possível substituir a fórmula mencionada acima pelo cálculo da variância interclasse, diminuindo assim o custo do algoritmo. A equação 4.8, representa o novo método implementado por Otsu [50],

$$\sigma_B^2 = \sigma^2 - \sigma_W^2 = W_b(\mu_b - \mu)^2 + W_f(\mu_f - \mu)^2 = W_bW_f(\mu_b - \mu_f)^2 \quad (4.8)$$

onde,

$$\mu = W_b\mu_b + W_f\mu_f. \quad (4.9)$$

Com este método, o algoritmo procura o valor que maximiza a variância interclasse, pois esse valor é o mesmo que minimiza a variância intraclasse [50].

Para a realização da binarização recorreu-se à função *F_VN_Threshold*. Esta tem como parâmetros de entrada: a imagem a ser binarizada; a imagem de destino; o valor do limiar, o qual não é utilizado se for selecionado um limiar dinâmico; e, o valor que será dado aos pixels que cumprem os critérios do limiar, neste caso, 255, correspondente a branco. Na Figura 4.5, é possível visualizar o resultado da binarização de uma peça.

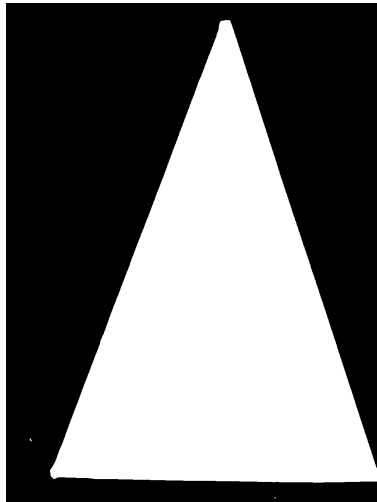


Figura 4.5: Imagem binária

4.5 Análise de Cor

A primeira análise a ser feita à superfície da peça é a análise da cor. Nesta fase, é calculada a distância entre a cor da superfície a analisar e a cor da superfície de referência, de modo a determinar se esta se encontra dentro dos valores para os quais a peça é considerada aceitável. Para tal, é necessário convertê-las para um espaço de cor que permita realizar esse cálculo de modo preciso, tendo em conta atributos chaves, tais como a luminosidade, saturação e cor.

Para este projeto, o cálculo da distância de cor foi realizado com a imagem no espaço de cor, $CIE\ L^*a^*b^*$.

4.5.1 Conversão Espaço de Cor

O primeiro passo desta etapa, é a conversão da imagem para o espaço de cor $CIE\ L^*a^*b^*$ [2]. O *software* da BECKHOFF, TwinCAT3 Vision, tem disponível na sua biblioteca a função que permite realizar a conversão entre espaços de cor, *F_VN_ConvertColorSpace*. Esta é adequada para realizar a conversão para espaços como RGB ou para escala de cinza, no entanto, não é eficaz na conversão para espaços de cor como o $CIE\ L^*a^*b^*$, uma vez que os valores são redimensionados para um intervalo de 0 a 255.

A imagem adquirida, foi convertida para o espaço RGB, através da função *F_VN_ConvertColorSpace*. Na imagem binarizada, é feita uma análise ao valor dos pixels, através da função *F_VN_GetPixel*, também disponível na biblioteca. Quando o pixel toma o valor de 255, significa que este faz parte da peça e, então, nessa mesma posição, e usando a mesma função, é lido o valor do pixel na imagem em RGB, para depois se fazer a sua conversão. Primeiro, é feita a conversão

do espaço RGB para o espaço XYZ, e só depois, do espaço XYZ para o espaço CIEL*a*b*.

Para a conversão do espaço de cor RGB para o espaço de cor XYZ, é necessário, em primeiro lugar, redimensionar os valores RGB e só depois é efetuada a conversão para XYZ. Esta conversão é implementada através das equações 4.10, 4.11, 4.12 e 4.13.

$$\begin{cases} R' = R/255 \\ G' = G/255 \\ B' = B/255 \end{cases} \quad (4.10)$$

$$\begin{cases} R' = \left(\frac{R'+0.055}{1.055}\right)^{2.4}, R' > 0.04045 \\ R' = \frac{R'}{12.92}, R' < 0.04045 \\ G' = \left(\frac{G'+0.055}{1.055}\right)^{2.4}, G' > 0.04045 \\ G' = \frac{G'}{12.92}, G' < 0.04045 \\ B' = \left(\frac{B'+0.055}{1.055}\right)^{2.4}, B' > 0.04045 \\ B' = \frac{B'}{12.92}, B' < 0.04045 \end{cases} \quad (4.11)$$

$$\begin{cases} R' = R' * 100 \\ G' = G' * 100 \\ B' = B' * 100 \end{cases} \quad (4.12)$$

$$\begin{cases} X = R' * 0.4121 + G' * 0.3576 + B' * 0.1805 \\ Y = R' * 0.2126 + G' * 0.7152 + B' * 0.0722 \\ Z = R' * 0.0193 + G' * 0.1192 + B' * 0.9505 \end{cases} \quad (4.13)$$

As formulas utilizadas para a conversão do espaço XYZ para CIEL*a*b* são:

$$\begin{cases} L^* = 116(X/X_{ref})^{1/3} - 16 \\ a^* = 500((X/X_{ref})^{1/3} - (Y/Y_{ref})^{1/3}) \\ b^* = 200((Y/Y_{ref}) - (Z/Z_{ref})^{1/3}) \end{cases} \quad (4.14)$$

onde, $X_{ref}=94.81$, $Y_{ref}=100.0$, $Z_{ref}=107.3$, são as coordenadas da cor do corpo branco para o iluminante CIE D65 [2].

4.5.2 Cálculo da Distância de Cor

Com os pixels convertidos para o espaço CIEL*a*b*, são somados os valores de cada canal, L*, a* e b*, quer da imagem a analisar, quer da imagem de referência. É depois calculado o valor médio de cada canal em ambas as imagens. Esse valor médio será o valor utilizado para o cálculo da distância de cor.

O cálculo da distância de cor, é feito através da equação 2.2. Para verificar se a peça se encontra dentro dos parâmetros aceitáveis, foi definido que o valor máximo da distancia de cor seria 1.6. A escolha deste método, assim como do valor do limiar de decisão, tem por base os resultados obtidos por experimentação e que serão apresentados em 5.5.1.

Caso o valor da distância de cor seja superior a 1.6, a peça é rejeitada de imediato, sem ser necessário passar pelas próximas etapas de inspeção. Caso o valor seja inferior, a peça segue para a próxima etapa, em que é efetuada uma análise à sua superfície para determinar se esta tem defeitos.

4.6 Análise de Superfície

Nesta etapa é feita uma análise à superfície da peça para verificar se esta tem defeitos ou não. A análise consiste na extração das características da forma da imagem da superfície da peça, calculando os respetivos momentos de Hu, sendo feito o mesmo à imagem da peça de referência. Os valores obtidos são então comparados, e é calculado um valor de dissimilaridade entre ambas. Caso o valor de dissimilaridade seja superior ao valor estabelecido como máximo admissível, significa que existem defeitos na superfície que está a ser analisada.

Para eliminar algum ruído que ainda possa existir após o processo de recorte da imagem, é aplicada uma máscara à imagem a analisar, colocando todos os pixels considerados de fundo no processo de binarização a preto, como mostra a Figura 4.6.

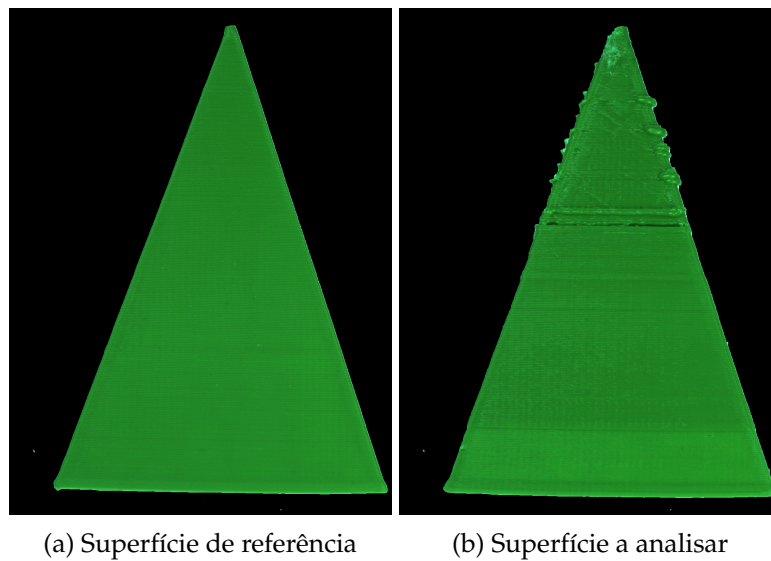


Figura 4.6: Resultado da aplicação da máscara

4.6.1 Extração de Características

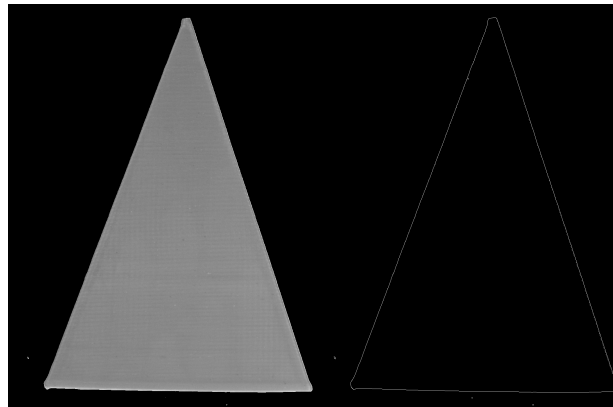
A imagem da superfície da peça a ser analisada, assim como a imagem da superfície da peça de referência, são binarizadas, utilizando um algoritmo de detecção de contornos, detetor de contornos de *Canny*. Este processo é feito para ser possível obter uma imagem binária descritiva da superfície da peça, para uma melhor extração das suas características.

4.6.1.1 Detetor de Contornos de *Canny*

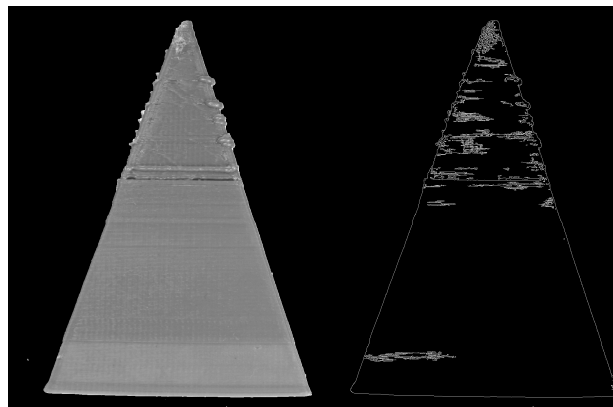
O algoritmo de detecção de contornos de *Canny* é um algoritmo para detecção de arestas ou contornos que consiste em 4 passos [51]:

1. Redução de ruído através de um filtro gaussiano.
2. Detecção de arestas aplicando uma matriz de Sobel, na direção vertical e horizontal, para ser obtida a primeira derivada da intensidade; a partir dessas duas imagens podemos encontrar o gradiente e a direção do contorno para cada pixel.
3. Remoção dos pixels que não pertencem ao contorno, por meio de supressão de não-máximos;
4. Binarização por limiar através de histerese, para decidir quais os contornos válidos.

A função utilizada foi *F_VN_CannyEdgeDetection*. Esta tem três parâmetros de entrada: imagem à qual se vai aplicar o detetor de contornos de Canny e os valores máximo e mínimo de limiar. A imagem de entrada tem que ser uma imagem de um canal, com 8 bits de intensidade, sendo possível utilizar uma imagem em escala de cinzas ou utilizar um canal individual de uma imagem a cores. Para este trabalho as imagens foram convertidas para escala de cinza. Na Figura 4.7 apresenta-se um exemplo da aplicação deste método.



(a) Imagem da superfície de referência



(b) Imagem da superfície a analisar

Figura 4.7: Imagem de entrada, à esquerda, e resultado do detetor de contornos de Canny, à direita

4.6.1.2 Momentos de Hu

Para a extração de características, são calculados os 7 momentos de Hu, para ambas as imagens, referência e adquirida. Os momentos são então comparados e é calculada a dissimilaridade entre ambas as imagens.

A função utilizada para o cálculo e comparação dos momentos de Hu foi *F_VN_MatchImageHuMoments*, a qual dispõe de três métodos possíveis para a

comparação das imagens e respetivo cálculo de dissimilaridade. O primeiro método consiste no cálculo da soma das diferenças entre o inverso das características individuais, conforme a equação 4.15. O segundo, consiste no cálculo da soma das diferenças entre as características individuais, conforme a equação 4.16. O terceiro e último método calcula a dissimilaridade usando apenas a diferença máxima entre as características individuais, de acordo com a equação 4.17,

$$D(A, B) = \sum_{i=1}^7 \left| \frac{1}{m_i^A} - \frac{1}{m_i^B} \right| \quad (4.15)$$

$$D(A, B) = \sum_{i=1}^7 | m_i^A - m_i^B | \quad (4.16)$$

$$D(A, B) = \max_{i=1..7} \frac{| m_i^A - m_i^B |}{| m_i^A |} \quad (4.17)$$

onde,

$$m_i^A = \text{sign}(h_i^A) \cdot \log(h_i^A) \quad (4.18)$$

$$m_i^B = \text{sign}(h_i^B) \cdot \log(h_i^B) \quad (4.19)$$

sendo que, h_i^A e h_i^B , representam cada um dos momentos de Hu das duas imagens.

Com base nos resultados obtidos durante a fase de teste, o método escolhido foi o método 3, sendo este o que apresentou melhor discriminação entre as peças com e sem defeito. O valor estabelecido como máximo de dissimilaridade foi 0.1. Este foi o valor escolhido por experimentação. Caso o valor de dissimilaridade seja inferior a 0.1, a peça é considerada de boa qualidade, uma vez que tem a cor correta e não apresenta defeitos na sua superfície. Se o valor de dissimilaridade for superior, então esta irá passar à próxima fase onde serão detetados os defeitos presentes na sua superfície.

4.7 Localização de Defeitos

Nesta etapa são encontrados os defeitos existentes na superfície da peça. Para a deteção dos defeitos, a imagem a ser analisada é primeiro filtrada através de um filtro de mediana e, de seguida, é feita a subtração da imagem original pela imagem filtrada. A imagem resultante da subtração é depois invertida e binarizada, de modo a ser possível localizar os defeitos.

Semelhante à etapa de análise de superfície, é aplicada uma máscara à imagem, no entanto, os pixels de fundo irão ser colocados a branco e não a preto, como mostra a Figura 4.8.

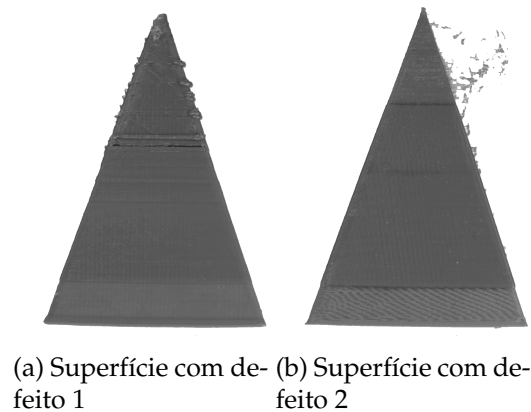


Figura 4.8: Resultado da aplicação da máscara

4.7.1 Filtro de Mediana e Subtração

O primeiro passo desta etapa é a aplicação de um filtro de mediana. O tamanho da matriz do filtro utilizada foi de 21x21. Este valor foi o escolhido por ser o valor para o qual os defeitos da superfície eram eliminados na totalidade. Para a aplicação do filtro, a função utilizada foi $F_VN_MedianFilter$, que tem como parâmetros de entrada a imagem a ser filtrada, a imagem de destino, onde será guardada a imagem filtrada, e o tamanho da matriz do filtro.

O filtro de mediana é um filtro não linear, que se baseia em estatísticas, este filtra cada pixel da imagem e compara-o com os pixels vizinhos de modo a decidir se este é representativo da sua vizinhança. Por norma, o valor ruidoso de cada pixel é substituído pelo valor da mediana dos valores dos pixels na sua vizinhança, sendo esta definida pela máscara. Ou seja, os pixels da máscara são ordenados por valor crescente, e o valor na posição central do grupo é usado para substituir o valor do pixel ruidoso. A saída do filtro de mediana é $g(x,y) = med\{f(x-i, y-j), i, j \in W\}$, onde $f(x,y)$, $g(x,y)$ são a imagem original e a imagem de saída, filtrada, respetivamente, e W é uma máscara bidimensional de tamanho $n \times n$ sendo, por norma, n um número ímpar [52] [53].

O próximo passo é a subtração da imagem original pela imagem filtrada, e para tal recorreu-se à função $F_VN_SubtractImages$, que tem como parâmetros de entrada as duas imagens a serem subtraídas e a imagem de destino. Como o tamanho da matriz do filtro é maior que a largura dos defeitos, estes são eliminados da imagem filtrada, mas continuam presentes na imagem original. O

resultado desta operação será uma imagem onde apenas os defeitos estão presentes.

De modo a destacar os defeitos, resultantes da subtração das imagens, recorreu-se à função *F_VN_InvertImageColor*, a qual permite inverter a cor da imagem. Caso só existam pixels com valores positivos, estes são subtraídos do valor máximo do pixel disponível. Na imagem invertida foi feita uma análise a todos os pixels, colocando todos os pixels com valor inferior a 255, cor branca, a 0. A função tem como parâmetros a imagem a ser invertida e a imagem de destino.

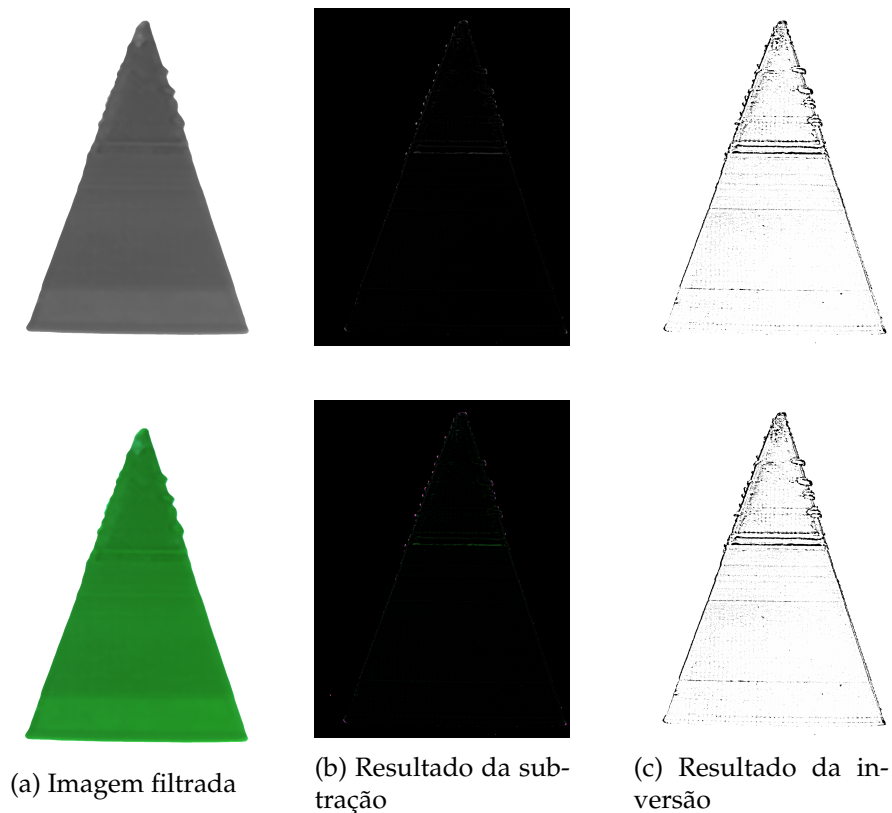


Figura 4.9: Resultados da subtração para imagens em tons de cinza e imagens RGB

Este método foi testado para dois tipos de imagens, em escala de cinza e em RGB. Como é possível ver na Figura 4.9, os resultados obtidos com os dois métodos são semelhantes e, por isso, qualquer um deles é apropriado para detecção de erros. Neste projeto foi utilizado o primeiro método, imagem em escala de cinza, pois o tempo de processamento para realizar a filtragem da imagem é menor.

4.7.2 Binarização

A imagem invertida é depois binarizada pelo mesmo método utilizado anteriormente. De modo a eliminar alguns ruídos que possam existir e destacar os erros, é aplicado à imagem binarizada um operador morfológico, dilatação.

4.7.3 Operações Morfológicas

Os operadores morfológicos, em processamento de imagem, modificam as imagens baseados em formas. Têm muitas aplicações, tais como na análise de textura, eliminação de ruído e extração de contornos. O objetivo das operações morfológicas é eliminar possíveis imperfeições mantendo a estrutura da imagem [54].

As técnicas morfológicas têm como base a teoria dos conjuntos [54]. Utilizam uma pequena estrutura matricial, designada *template* ou elemento estruturante, que é aplicada a todos os pixels da imagem, e a sua forma e tamanho tem um impacto significativo no resultado final. Nestas técnicas, os valores dos pixels de saída são baseados na semelhança dos pixels da imagem de entrada com os seus vizinhos. A maioria das operações morfológicas têm por base a dilatação e a erosão, expressas na notação lógica *OR* e *AND*, ou uma combinação destas, como é o caso da operação de abertura e de fecho.

Neste trabalho, a operação utilizada foi a dilatação. A dilatação é aplicada à imagem binarizada, fazendo com que o objeto expanda. Esta operação aumenta gradualmente os limites dos pixels de primeiro plano, provocando um aumento dessa área e a diminuição de buracos nessa região. A nível matemático, podemos descrever o processo de dilatação da seguinte maneira. Seja A um conjunto de coordenadas da imagem de entrada e B um conjunto de coordenadas do elemento estruturante e B_x a tradução de B de modo a que a sua origem esteja em x . A dilatação de A por B é um conjunto de todos os pontos de x onde a interseção de B_x com A não é nula, equação 4.20 [54].

$$A \oplus B = \{x | (B_x \cap A) \neq \emptyset\} \quad (4.20)$$

A dilatação necessita de dois dados de entrada, a imagem que se pretende dilatar e o elemento estruturante. Este elemento determina o quanto a imagem deve ser dilatada [54]. A função utilizada para a aplicação do operador morfológico é a *F_VN_MorphologicalOperator*, cujos parâmetros de entrada são a imagem à qual se vai aplicar o operador morfológico, a imagem de destino onde será guardada a imagem após a operação, o tipo de operador e o elemento estruturante a ser aplicado. O elemento estruturante utilizado foi um rectângulo com tamanho 3x2 pixels, o qual foi criado com a função *F_VN_CreateStructuringElement*.

Na Figura 4.10, é possível ver a imagem binarizada, antes e após ser aplicada a operação de dilatação.

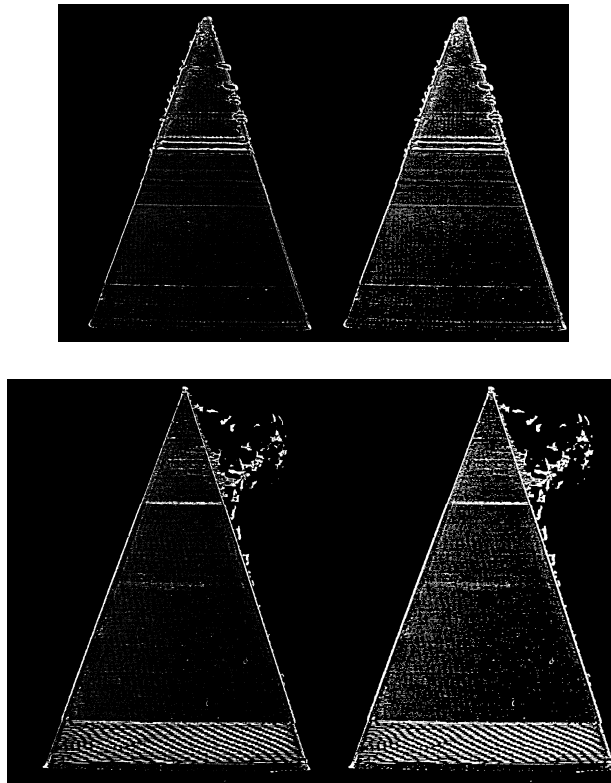


Figura 4.10: Binarização seguida de dilatação

4.7.4 Localização de Defeitos

Para a localização de defeitos, é utilizada a função $F_VN_DetectBlobs$, a qual procura por contornos fechados, sendo possível filtrá-los, de acordo com os parâmetros estabelecidos. Neste caso, este filtra os contornos de acordo com a sua área. Para o projeto apenas se limitou a área mínima para ser considerado um defeito, sendo que, para tal, a área tem de ser igual ou superior a 50 pixels. A função tem como parâmetros de entrada a imagem na qual será feita a procura de contornos, o *container* na qual serão guardados os contornos encontrados e os parâmetros para filtrar os contornos. Recorrendo à função $F_VN_DrawContours$, que tem como parâmetros de entrada o *container* no qual estão guardados os contornos detetados anteriormente, o indicador do contorno a ser desenhado (neste caso o indicador tem o valor de -1, para indicar que todos os contornos guardados no *container* são para ser desenhados), a imagem na qual serão desenhados, a cor e a espessura da linha. Os contornos são desenhados na imagem da superfície da peça, que se encontra no espaço de cor RGB, como se pode ver na Figura 4.11.

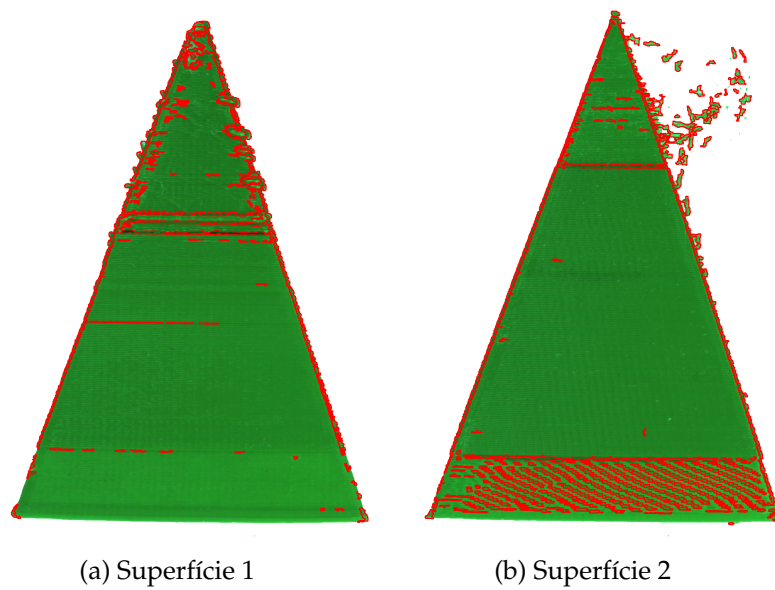


Figura 4.11: Localização de defeitos na superfície

4.8 Interface Gráfica

De modo a ser possível visualizar os resultados da análise feita pelo sistema, foi desenvolvida uma interface HMI, representada na Figura 4.12. A interface é constituída por um objeto visual, na qual é apresentada, em tempo real, a superfície que está a ser analisada e três LEDs, que ajudam o utilizador a ter uma melhor interpretação da classificação da peça por parte do sistema. Estes são de três cores distintas, vermelho, caso a cor da peça esteja errada, amarelo, caso a peça tenha na sua superfície erros e verde caso a peça seja classificada como tendo boa qualidade.



Figura 4.12: Interface gráfica HMI

Nas seguintes figuras é possível ver exemplos dos três tipos de classificação, cor errada 4.13, erros detetados na superfície 4.14 e superfície em boa quali-

dade 4.15.

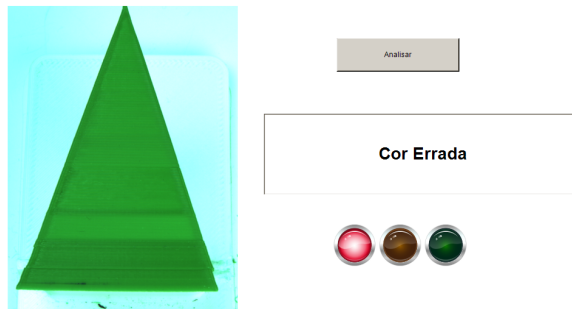


Figura 4.13: Classificação da superfície - cor errada

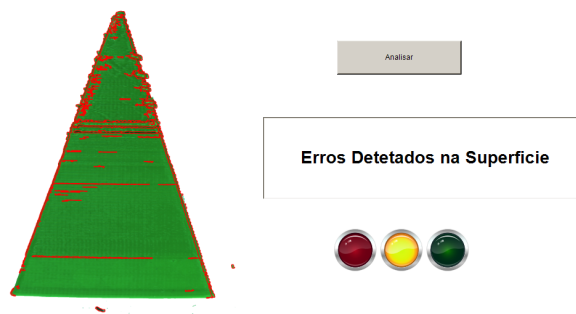


Figura 4.14: Classificação da superfície - erros na superfície



Figura 4.15: Classificação da superfície - boa qualidade

Capítulo 5

Testes e Discussão dos Resultados

Neste capítulo serão apresentados os testes realizados ao longo do projeto, incluindo a calibração da câmara e as diferentes alternativas de binarização. Os resultados obtidos foram avaliados recorrendo a diferentes métricas, quer para análise de cor, quer para a análise de superfície. Por fim, será avaliado e discutido o desempenho global do sistema.

5.1 Peças

Para o desenvolvimento do sistema, as peças disponibilizadas têm o formato de uma pirâmide quadrangular. Estas apresentavam nas suas superfícies vários defeitos, desde erros na cor até excesso de extrusão. Na Figura 5.1, é possível visualizar algumas das peças utilizadas para a realização da tarefa de inspeção automática. As peças foram disponibilizadas pelo INEGI, sendo que o número de peças disponíveis para o desenvolvimento do sistema foi muito limitado.

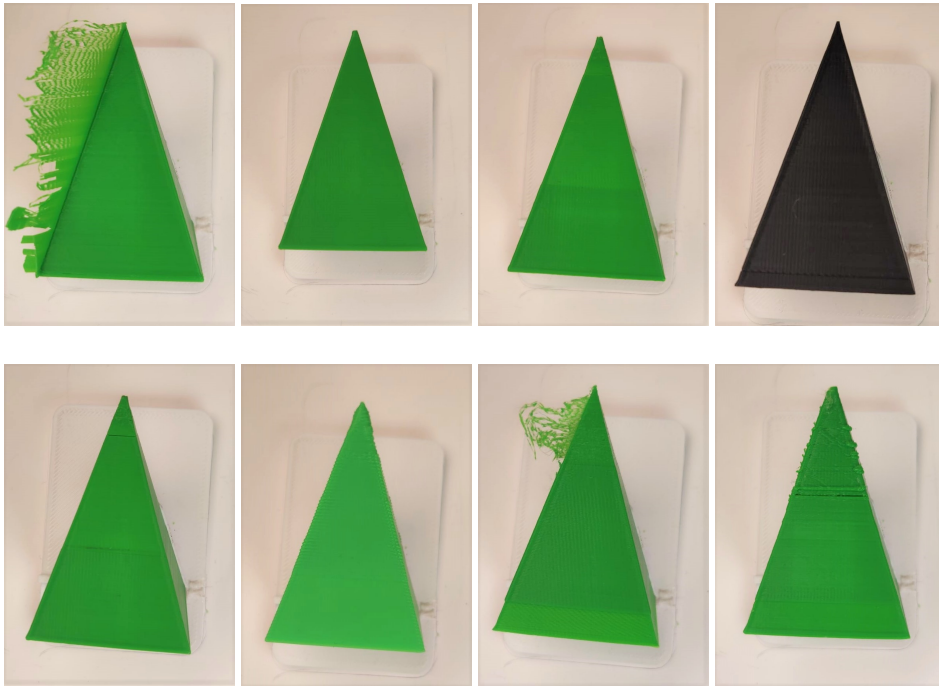


Figura 5.1: Exemplos de peças utilizadas

5.2 Calibração

De modo a verificar se a calibração da câmara foi feita corretamente, foi criada uma função com o intuito de medir objectos. A função denominada, *fMedicao*, faz uso de algumas funções disponíveis na biblioteca do TwinCAT Vision. A função *F_VN_LocateEdge* é usada para procurar as bordas do objeto, na imagem já binarizada. Esta função é utilizada duas vezes para encontrar o limite superior e o limite inferior, e os pontos pertencentes às bordas dos objetos, são armazenados nos *containers*. A função *F_VN_GetAt_TcVnPoint2_REAL* é utilizada para se obter o primeiro ponto de cada *container* e, por fim, a função *F_VN_ImagePointsWorldDistance* calcula a distância entre dois pontos no sistema de coordenadas real.



Figura 5.2: Resultado da medição de um objeto

Na Figura 5.2, está representada a medição de um objeto. Este tem 4 centímetros de largura e, como é possível verificar, o resultado da medição foi de 4.30 centímetros, aproximadamente. O resultado da medição apresentou um erro de 7.5%, e este valor deve-se ao facto do objeto não estar posicionado corretamente, resultando que o primeiro ponto da primeira borda e o segundo ponto da segunda borda não se encontram exatamente um sobre o outro, ou seja, a medição não é feita por uma linha vertical, mas sim por uma linha oblíqua.

Para o sistema desenvolvido a calibração não é essencial, uma vez que não é feita inspeção às dimensões da peça. No entanto, a calibração será importante quando o sistema for integrado com o robô colaborativo e o transportador pois o *gripper*, acoplado na *flange* do robô, terá de realizar com sucesso o *pick and place* da peça. O erro de calibração não irá comprometer o desempenho do sistema uma vez que a abertura do *gripper*, terá uma folga de abertura muito maior.

5.3 Binarização

Para a binarização do objeto foram utilizados vários métodos, estes disponíveis na função $F_VN_Threshold$. Os métodos testados para a binarização foram: a binarização por limiar invertida, onde é dado o valor máximo, 255, ao pixel caso este tenha um valor inferior ao limiar; truncatura, caso o valor do pixel seja superior ao valor do limiar este fica com o valor do limiar; o limiar para zero, em que é dado o valor 0 ao pixel, caso este seja inferior ao valor do limiar; o limiar a zero invertido, onde é dado o valor de 0 ao pixel cujo valor seja superior ao limiar [55]; e o método de Otsu, que determina o valor do limiar de decisão automaticamente.

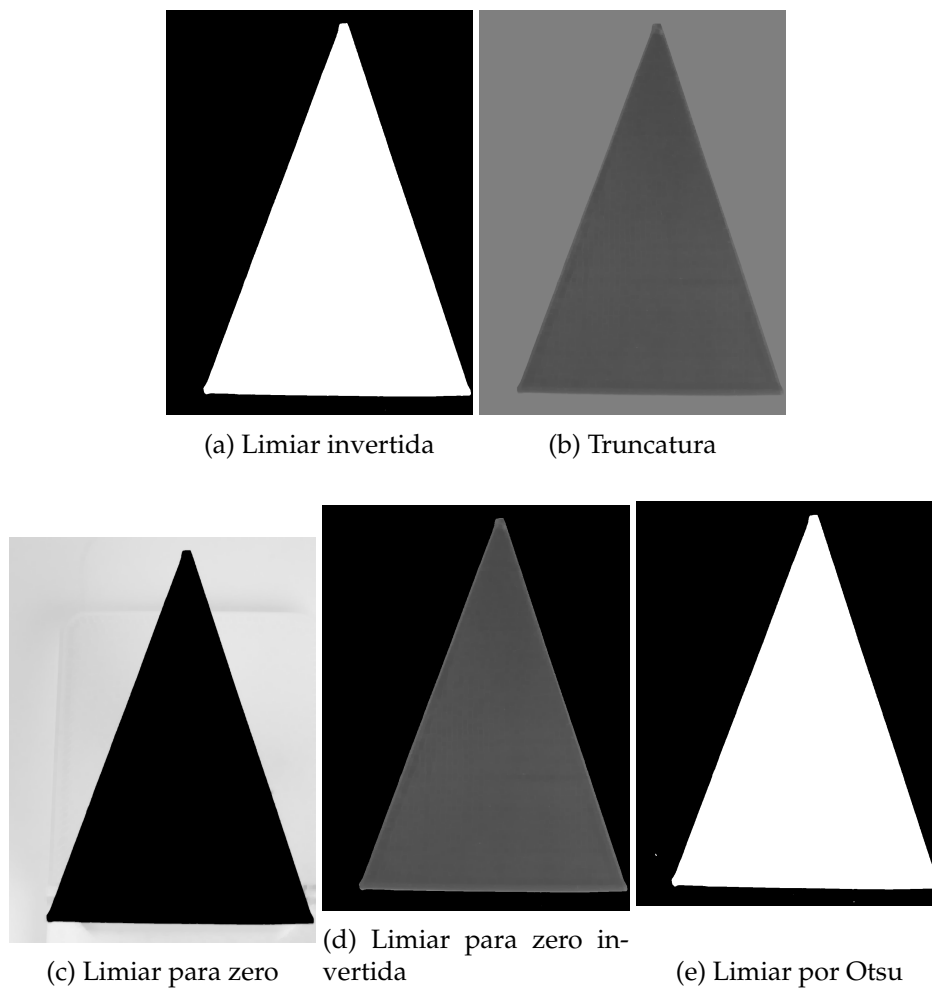


Figura 5.3: Resultados para os diferentes métodos de binarização

Com a análise Figura 5.3, podemos concluir que os métodos que obtiveram melhores resultados foi o método de binarização invertida e a binarização por método de Otsu. Dos dois, a binarização por Otsu foi o método escolhido para implementar no sistema pois este não precisa de um valor de limiar pré definido, adaptando-se automaticamente às características da imagem.

5.3.1 Detecção de Objeto

A verificação da binarização, foi feita através da função $F_VN_FindContours$, a qual procura os contornos do objeto na imagem binária. Os contornos são depois desenhados na imagem original, através da função $F_VN_DrawContours$. Caso a binarização tenha sido feita corretamente, os contornos da peça irão aparecer sobrepostos a vermelho na imagem original, como mostra a Figura 5.4

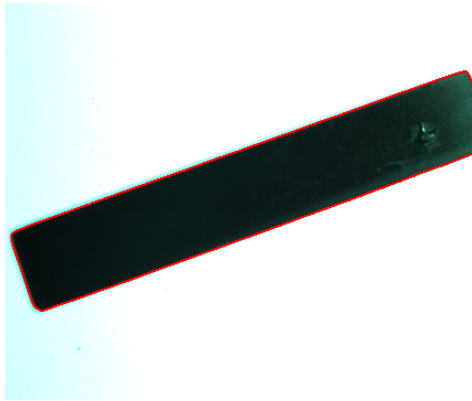


Figura 5.4: Resultado da detecção de objeto

A correta detecção do objeto será importante para a futura integração do sistema com o robô pois este irá necessitar de localizar corretamente a peça, para poder realizar o *pick and place* para o transportador.

5.4 Avaliação do sistema de classificação

Durante o desenvolvimento do sistema, foram estudados e testados vários métodos para realizar a classificação da superfície. Para esses métodos foi alterado o valor de limiar de decisão, de modo a ser possível determinar com qual dos métodos e qual o valor de limiar que obtém melhores resultados na classificação das superfícies. Para isso, procedeu-se à análise gráfica da *Receiver Operating Characteristic* (ROC) e, conseqüentemente, da *Area Under the Curve* (AUC).

Uma curva ROC é um gráfico que mostra o desempenho de um modelo de classificação para diferentes valores de um parâmetro, neste caso, um conjunto de valores do limiar de decisão, permitindo estudar a variação da sensibilidade e especificidade do classificador. Esta curva apresenta dois parâmetros, a Taxa de Positivos Verdadeiros (*True Positive Rate*, TPR) e a Taxa de Falsos Positivos (*False Positive Rate*, FPR), calculados através das equações 5.1 e 5.2, respectivamente.

$$TPR = \frac{TP}{TP + FN} \quad (5.1)$$

$$FPR = \frac{FP}{FP + TN} \quad (5.2)$$

Para este trabalho, as superfícies sem defeito foram classificadas como positivas e as superfícies com defeito foram consideradas como negativas. Assim,

nas equações 5.1 e 5.2, TP (*True Positive*) representa as peças sem defeitos corretamente classificadas, FP (*False Positive*) as peças com defeitos erradamente classificadas, TN (*True Negative*) as peças com defeitos corretamente classificadas, FN (*False Negative*) as peças sem defeitos erradamente classificadas. O objectivo é maximizar TPR e minimizar FPR mas, em geral, existe um compromisso entre elas. Este compromisso pode ser representado pela ROC. Assim, pela análise da curva ROC é possível determinar qual o valor de limiar de decisão mais adequado para o modelo.

A AUC representa o grau ou a medida de separabilidade, ou seja, ajuda a distinguir a capacidade de discriminação de diferentes métodos. Quanto maior a AUC, maior a capacidade do modelo distinguir entre duas classes.

Também na fase de treino foi calculada a *Percentage of Wrong Classifications* (PWC), para cada valor de limiar. PWC é a percentagem de classificações erradas e é calculada através da equação 5.3.

$$PWC = 100 \frac{FN + FP}{TP + FN + FP + TN} \quad (5.3)$$

Na fase de testes, de modo a avaliar os diferentes métodos de análise e o desempenho global do sistema, foram calculadas também a sensibilidade, a precisão, e a exatidão, para além do PWC.

A sensibilidade ou TPR equação 5.1, representa a percentagem de superfícies positivas corretamente classificadas do total de superfícies positivas analisadas.

A precisão é a percentagem de superfícies positivas corretamente classificadas do total de superfícies classificadas como positivas. Esta é calculada através da equação 5.4.

$$\text{Precisão} = \frac{TP}{TP + FP} \quad (5.4)$$

A exatidão é a percentagem de superfícies corretamente classificadas do total de superfícies analisadas, calculada através da equação 5.5. Onde n representa o número total de peças.

$$\text{Exatidão} = \frac{TP + TN}{n} \quad (5.5)$$

Nesta avaliação, é assumido que o custo de classificar incorretamente uma peça de boa qualidade é igual ao custo de classificar incorretamente uma peça de má qualidade. Numa aplicação real, estes erros podem ter associados custos diferentes. Nesse caso, será necessário utilizar uma ponderação diferente para os dois tipos de erro.

5.5 Fase de Treino

Nesta fase, é utilizada uma amostra das peças, correspondente a cerca de 60% do total de peças disponíveis. Estas serão avaliadas por diferentes métodos e com diferentes valores de limiar de decisão, de modo a encontrar o modelo e o valor de limiar mais adequado para a implementação do sistema. Para a fase de testes, apenas estava disponível um tipo de peça, quer em termos de cor quer de geometria.

Tanto para a fase de treino como, posteriormente, para a fase de testes, as peças foram previamente classificadas de forma independente por três pessoas, sendo a classificação final da peça obtida por maioria. Esta foi a classificação de referência utilizada para avaliar o desempenho do sistema.

5.5.1 Análise de cor

Para este projeto foram testados dois métodos para análise de cor, já apresentados no capítulo 2: ΔE , dado pela Equação 2.1 e ΔE_{2000} , dado pela Equação 2.2. Na fase de treino, foram utilizadas 68 superfícies, das quais 20 se encontravam com cor incorreta e 48 com cor correta.

Para o primeiro método, o cálculo da distância de cor através da equação 2.1, foram testados cinco valores diferentes de limiar de decisão $T_{\Delta E}$, 0.5, 1, 1.5, 2, 2.5. Para o segundo método, o cálculo da distância de cor através da equação 2.2, foram também testados cinco valores diferentes de limiar de decisão $T_{\Delta E_{2000}}$, 1, 1.2, 1.4, 1.6, 1.8. Estes valores foram escolhidos de acordo com os valores analisados numa primeira fase.

Na Tabela 5.1, é possível verificar os resultados obtidos usando a primeira medida de distância de cor na fase de treino, para os diferentes valores de limiar.

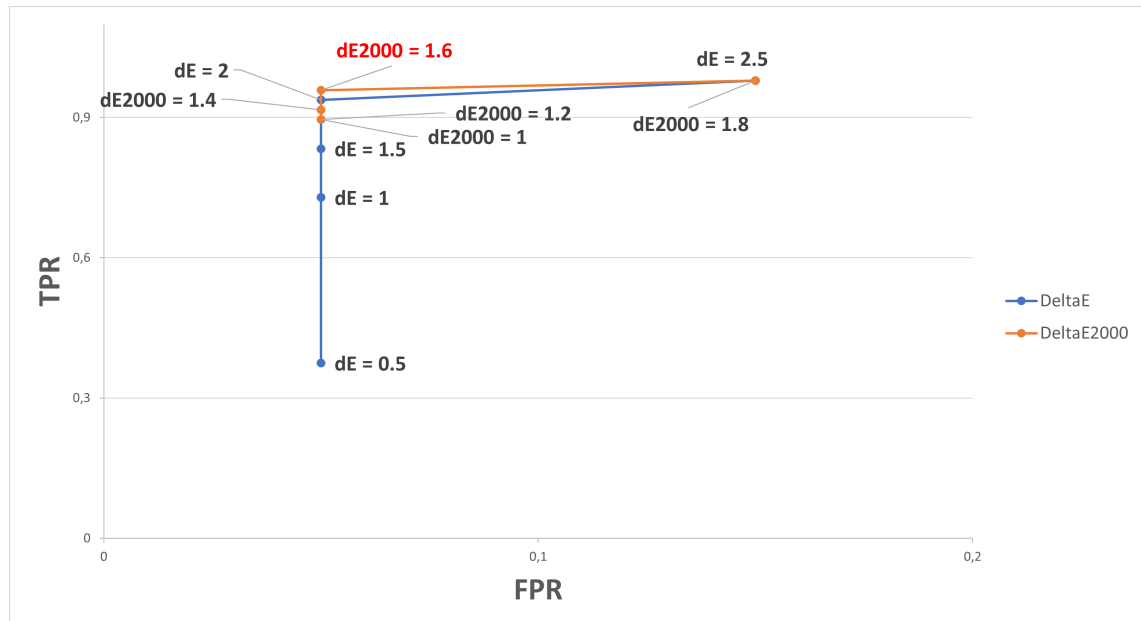
Tabela 5.1: Resultados da análise de cor baseados no ΔE - fase treino

$T_{\Delta E}$	TP	FN	TN	FP	TPR	FPR	PWC
0.5	18	30	19	1	0.375	0.05	45.588
1	35	13	19	1	0.729	0.05	20.588
1.5	40	8	19	1	0.833	0.05	13.235
2	45	3	19	1	0.9375	0.05	5.882
2.5	47	1	17	3	0.979	0.15	5.882

Na Tabela 5.2, é possível verificar os resultados obtidos para os diferentes valores de limiar, usando a segunda medida, ΔE_{2000} , na fase de treino.

Tabela 5.2: Resultados da análise de cor baseados no ΔE_{2000} - fase treino

$T_{\Delta E_{2000}}$	TP	FN	TN	FP	TPR	FPR	PWC
1	43	5	19	1	0.896	0.05	8.823
1.2	43	5	19	1	0.896	0.05	8.823
1.4	44	4	19	1	0.917	0.05	7.353
1.6	46	2	19	1	0.958	0.05	4.412
1.8	47	1	17	3	0.979	0.15	5.882

Figura 5.5: Curvas ROC para ΔE e ΔE_{2000}

Na Figura 5.5, estão representadas as curvas ROC para os classificadores de cor baseados nas duas medidas propostas, ΔE e ΔE_{2000} , assim como os valores de limiar testados para cada uma. Pela análise do gráfico é possível observar que o segundo método, ΔE_{2000} , apresenta uma maior AUC e o valor de limiar que resulta num menor PWC é 1.6. Assim, podemos concluir que estes são os mais indicados para implementar no sistema.

5.5.2 Análise de forma

A análise da forma foi realizada recorrendo aos momentos de Hu, apresentados no capítulo 2. A função $F_VN_MatchImageHuMoments$ dispõe de três métodos para calcular a dissimilaridade entre duas imagens, baseada nos momentos de Hu. Para o projeto, apenas o primeiro método, equação (4.15) e o terceiro método, equação (4.17), foram analisados, pois os valores de dissimilaridade obtidos com o segundo método não permitem uma classificação precisa das su-

perfícies. De modo a verificar qual dos métodos é o mais adequado e qual o valor de limiar de decisão ideal foi, à semelhança do que foi feito para a análise de cor, realizada uma análise às curvas ROC, e respetivas AUC, dos dois métodos. Na fase de treino foram usadas 40 superfícies, onde 26 apresentavam erros e 14 não tinham qualquer tipo de erro na superfície.

Uma vez que, em ambos os métodos, os valores de dissimilaridade eram próximos para a mesma superfície, os valores de limiar de decisão T_{Diss} , foram os mesmos para ambos os métodos, sendo estes 0.05, 0.1, 0.15, 0.2, 0.25 e 0.3.

Na Tabela 5.3, é possível verificar os resultados obtidos pelo primeiro método na fase de treino, para os diferentes valores de limiar.

Tabela 5.3: Resultados da análise da forma baseados no método 1 - fase treino

T_{Diss}	TP	FN	TN	FP	TPR	FPR	PWC
0.05	0	14	26	0	0	0	0
0.1	6	8	23	3	0.429	0.115	35.0
0.15	7	7	23	3	0.5	0.115	27.5
0.2	9	5	21	5	0.643	0.192	25.0
0.25	11	3	20	6	0.786	0.231	22.5
0.3	11	3	19	7	0.786	0.269	25.0

Na Tabela 5.4, é possível verificar os resultados obtidos pelo terceiro método na fase de treino, para os diferentes valores de limiar.

Tabela 5.4: Resultados da análise da forma baseados no método 3 - fase treino

T_{Diss}	TP	FN	TN	FP	TPR	FPR	PWC
0.05	10	4	25	1	0.714	0.04	12.5
0.1	12	2	23	3	0.857	0.115	12.5
0.15	12	2	16	10	0.857	0.385	30.0
0.2	12	2	10	16	0.643	0.615	45.0
0.25	12	2	9	17	0.786	0.654	47.5
0.3	13	1	9	17	0.929	0.654	45.0

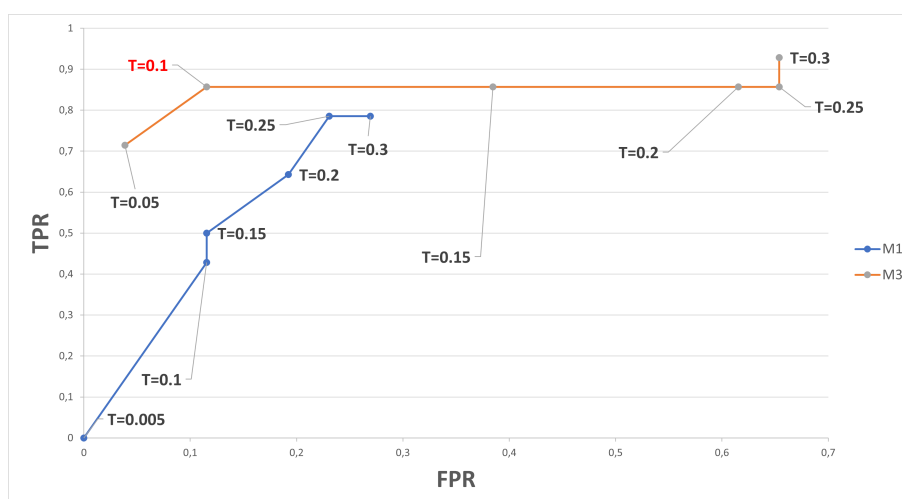


Figura 5.6: Curvas ROC- Match Hu Moments

Na Figura 5.6, estão representadas as curvas ROC para as duas medidas propostas, assim como os valores de limiar testados para cada uma. Pela análise da figura, percebe-se que o método a que corresponde uma maior AUC é o terceiro, podendo concluir-se que este tem um melhor desempenho na classificação das superfícies da peça. Para este, os valores de limiar de decisão que resultam numa menor percentagem de erro de classificação são 0.05 e 0.1. Para implementação no sistema, foi escolhido o valor de 0.1 pois é o que apresenta maior sensibilidade.

5.6 Fase de Testes

Para a fase de testes foram utilizadas 52 superfícies e, destas, 24 apresentavam erro na cor, 14 apresentavam erros na superfície e as restantes tinham boa qualidade. Na fase de testes foi calculada a sensibilidade, equação 5.1, a precisão, equação 5.4, a exatidão, equação 5.5, e o PWC, equação 5.3, de cada uma das etapas de classificação individualmente e em conjunto, avaliando assim o desempenho do sistema na classificação de superfícies. Também nesta fase, apenas um tipo de peça estava disponível.

Para a análise de cor foram utilizadas 52 superfícies, 24 com erro na cor e 28 com a cor correta. Na Tabela 5.5, estão representados os resultados obtidos na fase de testes na análise baseada em cor.

Tabela 5.5: Resultados da análise baseada em cor

Cor	TP	FN	TN	FP	Sensibilidade	Precisão	Exatidão	PWC
Detetados	27	1	23	1	0.964	0.964	0.962	3.8

Com os resultados da Tabela 5.5, é possível concluir que o método para a análise de cor classificou 28 superfícies como não tendo erro na cor, dessas 28 apenas 27 não apresentavam erro, obtendo-se assim uma sensibilidade de 96.4%. Do total de superfícies positivas, 27 foram classificadas corretamente, obtendo-se assim uma precisão de 96.4%. Por fim, o método foi capaz de classificar corretamente 50 superfícies das 52 analisadas, apresentando por isso uma exatidão de 96.2%. O método apresenta uma percentagem de erro de 3.8%, sendo possível afirmar que o método implementado é capaz de classificar a peça tendo por base a cor.

Para a análise de superfície foram utilizadas 28 superfícies, 14 sem defeitos e 14 com defeitos. Na Tabela 5.6, estão representados os resultados obtidos na fase de testes na análise de superfície tendo por base a forma.

Tabela 5.6: Resultados da análise baseada na forma

Superfície	TP	FN	TN	FP	Sensibilidade	Precisão	Exatidão	PWC
Detetados	11	3	12	2	0.786	0.846	0.821	17.8

Na Tabela 5.6 estão demonstrados os resultados obtidos para a análise baseada na forma. O método desenvolvido classificou 13 superfícies como não tendo defeitos, dessas apenas 11 não apresentavam qualquer tipo de defeito, concluído assim que o método tem um sensibilidade de 78.6%. Das 14 superfícies, apenas 11 foram corretamente classificadas, obtendo assim uma precisão de 84.6%. Das 28 superfícies analisadas o método classificou corretamente 23 tendo por isso uma exatidão de 82.1%. O método apresenta um PWC de 17.857%, um valor não desprezável, pois algumas das imagens das superfícies apresentavam reflexos, como mencionado em 3.1.2, que eram por sua vez destacados pelo detetor de contornos de *Canny*, sendo por isso considerados como defeitos pelo sistema ainda que não sejam. Uma iluminação mais adequada, onde as imagens não sejam adquiridas com reflexos nem haja interferência de fatores externos como por exemplo, a movimentação dos iluminadores, irá diminuir o valor de PWC.

Por último foram combinadas as duas análises de modo a avaliar o sistema desenvolvido na classificação de superfícies. Para tal, foram utilizadas 52 amostras, onde 14 não apresentavam qualquer erro ou defeito e 38 apresentavam erros na cor ou defeitos na forma. Na Tabela 5.7 estão representados os resultados obtidos na fase de testes para a avaliação global do sistema desenvolvido.

Tabela 5.7: Desempenho global do sistema

Completo	TP	FN	TN	FP	Sensibilidade	Precisão	Exatidão	PWC
Detetados	10	4	35	3	0.714	0.714	0.865	13.4

Pela observação dos resultados da Tabela 5.7 é possível afirmar que o sistema classificou corretamente 45 das 52 superfícies analisadas, tendo uma exatidão de 86.5%. O sistema obteve um PWC de 13.4%. Durante a avaliação global do sistema foi possível observar que 5 das peças mal classificadas, foram mal classificadas quando foi realizada a análise à sua forma. Como foi dito anteriormente, a análise à forma é influenciada pela iluminação e, não sendo esta a mais adequada, provoca reflexos na superfície das peças que são classificados pelo sistema como defeitos.

Como foi dito anteriormente, foi realizada a classificação prévia das peças por três pessoas, de forma independente. Verificou-se que não houve concordância em todas as classificações, mais especificamente na classificação das peças que se encontravam na fronteira entre boa e má qualidade. Algumas dessas peças foram também incorretamente classificadas pelo sistema. Por este motivo, poderá fazer sentido considerar uma nova classe de classificação, de peças com qualidade intermédia, para as quais poderá ser necessária uma análise adicional, de modo a obter uma classificação definitiva.

Foram ainda analisadas peças com geometrias diferentes, das quais se mostram alguns exemplos, Figura 5.7 e Figura 5.8. De salientar que quando as peças foram analisadas o *setup* já não se encontrava nas mesmas condições, quer de iluminação quer da posição da câmara. Os resultados obtidos não são por isso, os mais indicados para realizar a avaliação do sistema. Contudo, são utilizados para analisar as limitações do mesmo.

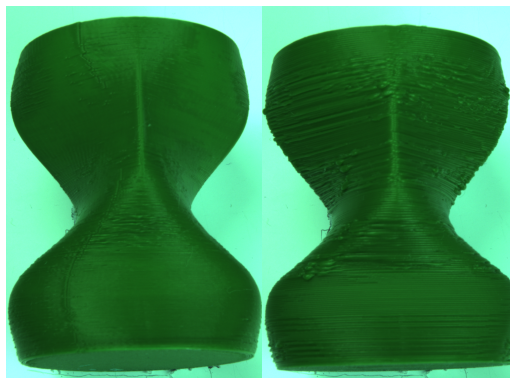


Figura 5.7: Peça exemplo 1

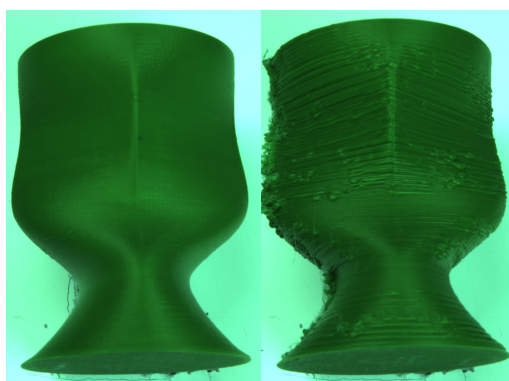


Figura 5.8: Peça exemplo 2

Foi feita a análise de 18 peças do tipo apresentado na Figura 5.7, estando representados na Tabela 5.8 os resultados obtidos pelo sistema para a sua classificação.

Tabela 5.8: Desempenho global do sistema - peça exemplo 1

Completo	TP	FN	TN	FP	Sensibilidade	Precisão	Exatidão	PWC
Detetados	6	3	8	1	0.666	0.855	0.777	22.2

Na Figura 5.8, é possível visualizar alguns exemplos de outra peça utilizada para a análise. Foi feita a análise de 18 peças desta geometria e, na Tabela 5.9, estão representados os resultados obtidos pelo sistema para a sua classificação.

Tabela 5.9: Desempenho global do sistema - peça exemplo 2

Completo	TP	FN	TN	FP	Sensibilidade	Precisão	Exatidão	PWC
Detetados	5	4	7	2	0.556	0.714	0.667	33.3

Comparando as duas tabelas, 5.8 e 5.9, é possível observar que o PWC da peça exemplo 2 é maior que o PWC da peça exemplo 1. Isto acontece pois, na peça exemplo 2, o suporte é detetado em dois sítios diferentes enquanto que na peça exemplo 1 é apenas detetado num sítio, como demonstrado na Figura 5.9. Ainda pela análise dos resultados nas tabelas 5.8 e 5.9, podemos afirmar que o desempenho global do sistema piorou, como é possível comprovar com o aumento do PWC em ambas as tabelas. Durante a análise das peças foi possível verificar que a incorreta classificação das peças ocorreu quando era realizada a análise tendo por base a forma. Uma vez que as peças foram testadas nos últimos dias, não foi possível arranjar um suporte adequado para mesmas, sendo este detetado pelo operador de contornos *Canny* em algumas delas. Na Figura 5.10, está representada a imagem resultante do operador de contornos

Canny, onde se pode verificar que o suporte é detetado como parte da peça. Outra razão para o valor elevado de PWC é o facto de as peças testadas serem de geometrias mais complexas. Para este tipo de geometrias, deverá ser admitido um valor de dissimilaridade maior entre a peça em análise e a referência, uma vez que não se consegue atingir uma qualidade de impressão igual àquela que se obtém com geometrias mais simples. No entanto, os resultados apresentados usam o valor que tinha sido previamente estabelecido.

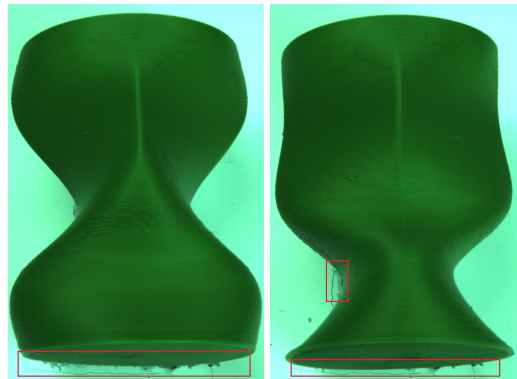


Figura 5.9: Exemplo do suporte visível nas peças exemplo



Figura 5.10: Exemplo do suporte a ser detetado pelo operador Canny na peça exemplo 2

5.7 Localização de Defeitos

Na Figura 5.11, estão representados alguns exemplos dos resultados obtidos na etapa de localização de defeitos.

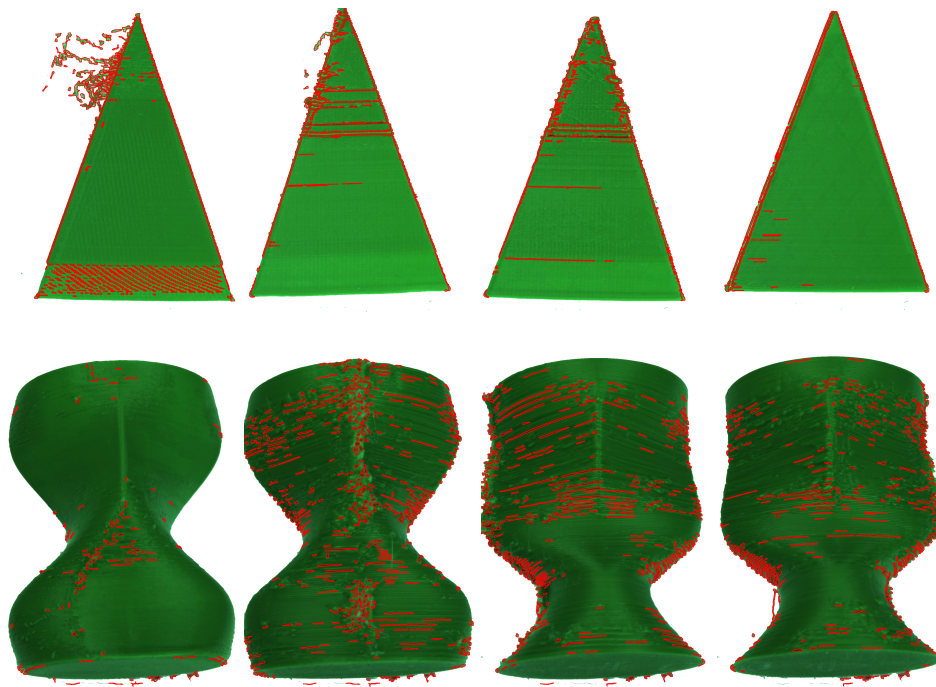


Figura 5.11: Localização dos defeitos nas diferentes peças

Como é possível observar na Figura 5.11 o método utilizado para a localização de defeitos consegue localizar a maioria dos defeitos existentes nas superfícies. Alguns defeitos não são localizados, uma vez que não são eliminados pelo filtro de mediana. Quando é realizada a subtração entre a imagem filtrada e a imagem original, esse defeitos são também eliminados. Sendo estas peças fabricadas por processos de fabrico aditivo, FDM, a divisão entre camadas é destacada em alguns casos pela incidência da luz, e essa divisão é depois localizada como defeito, ainda que não o seja. A melhoria do *setup* do sistema, como a iluminação, irá reduzir a localização de ruídos.

Capítulo 6

Conclusão

Este trabalho teve como principais objetivos o desenvolvimento de um sistema de inspeção automática, recorrendo ao *software* TwinCAT Vision da Beckhoff. O sistema desenvolvido, recorre ao processamento de imagem para a classificação de peças, fabricadas através de impressão 3D e localização de defeitos, caso estes existam. Numa primeira fase é feito o pré-processamento da imagem de modo a obter uma imagem com o mínimo ruído possível para uma melhor análise da superfície da peça. De seguida, é feita uma análise à cor da superfície, calculando a distância entre a cor da peça de referência e a cor da peça que está a ser analisada. Este cálculo é efetuado recorrendo à equação 2.2, para ver se esta se encontra dentro dos parâmetros aceitáveis e, caso não se encontre, a peça é rejeitada de imediato. Se, por outro lado, a peça tiver a cor correta, esta passa à próxima fase, onde é feita a análise à sua superfície, utilizando o detetor de contornos *Canny*, para se obter uma imagem descritiva da superfície da peça a ser analisada para ser comparada com a da peça de referência. Em seguida, são calculados os momentos de Hu destas imagens e comparados através do cálculo de dissimilaridade, usando apenas a diferença máxima entre as características individuais. Esta etapa irá permitir concluir se a superfície da peça se encontra com ou sem defeitos. As peças classificadas como tendo erros na sua superfície passam para a terceira e última fase onde, através de processos como a filtragem, subtração e inversão, é possível localizar os erros detetados na segunda fase.

No decorrer do projeto, existiram alguns obstáculos que levaram a um atraso no desenvolvimento do mesmo. Entre eles, é possível destacar a falta de documentação, por parte da Beckhoff, das funções disponíveis na biblioteca de visão do *software*, a falta de espaço físico de modo a ser possível montar um *setup* apropriado, que não fosse afetado por fatores externos como, por exemplo, a iluminação, sendo esta bastante importante para qualquer sistema de inspeção

automática.

Outra limitação do trabalho foi o número limitado e a falta de variedade das peças disponíveis, quer na cor quer na geometria, para testar e avaliar o desempenho do sistema. De referir que durante toda a fase de desenvolvimento e treino do sistema apenas um tipo de peça estava disponível.

Contudo foi possível desenvolver um sistema experimental de inspeção automática, capaz de classificar e localizar defeitos, caso existam, em peças fabricadas por processos de fabrico aditivo. Quando implementado num sistema real, existirão condições favoráveis para que haja um aumento da exatidão global do sistema. Condições como melhor iluminação e suporte para as peças adequados com as mesmas que não interfiram na análise da peça.

O TwinCAT Vision, ainda que seja um *software* recente de processamento de imagem, tem disponível na sua biblioteca diversas funções que permitem desenvolver um sistema de inspeção automática, contudo, existe ainda pouca informação sobre essas funções.

Como futuros desenvolvimentos, é sugerido:

- Um melhoramento do *setup*, como as condições de iluminação, que permita que a aquisição da imagem das peças sujeitas a inspeção seja feita em melhores condições, não estando sujeitas à influência de fatores externos.
- Uma análise mais completa dos limiares de decisão usados na classificação, que resulte em valores que se adaptem melhor a peças com diferentes geometrias, mais complexas, e com diferentes cores. Eventualmente, o desenvolvimento de um método para estabelecimento automático destes limiares.
- A implementação de um algoritmo para análise da textura das peças, como os *Local Binary Pattern* (LBP) que, justamente como a análise de forma irá permitir uma melhor análise e consequentemente uma melhor classificação das superfícies das peças a serem analisadas.
- O desenvolvimento de um classificador com mais de dois níveis. Por exemplo a classificação ser dividida em três ou quatro categorias, boa, moderadamente boa, moderadamente má e má.
- Por fim, é sugerido também avaliar a possibilidade de implementação de uma rede neuronal que, através dos defeitos localizados, possa classificá-los corretamente.

Referências Bibliográficas

- [1] E. Malamas, E. Petrakis, M. Zervakis, L. Petit, and J.-D. Legat, “A survey on industrial vision systems, applications and tools,” *Image and Vision Computing*, vol. 21, pp. 171–188, 02 2003. [Quoted on p. 5, 6]
- [2] W. Mokrzycki and M. Tatol, “Color difference delta e - a survey,” *Machine Graphics and Vision*, vol. 20, pp. 383–411, 04 2011. [Quoted on p. 7, 8, 9, 10, 36, 37]
- [3] M. J. Swain and D. H. Ballard, “Color indexing,” *International Journal of Computer Vision*, vol. 7, no. 1, pp. 11–32, 1991. [Quoted on p. 7]
- [4] M. Stricker and M. Orengo, “Similarity of color images,” *Proceedings of SPIE - The International Society for Optical Engineering*, vol. 2420, pp. 381–392, 03 1995. [Quoted on p. 7]
- [5] J. Huang, S. Kumar, M. Mitra, W.-j. Zhu, and R. Zabih, “Image indexing using color correlograms,” *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 762–768, 05 1997. [Quoted on p. 8]
- [6] G. Pass and R. Zabih, “Histogram refinement for content-based image retrieval,” *Proceedings Third IEEE Workshop on Applications of Computer Vision. WACV’96*, pp. 96–102, 1996. [Quoted on p. 8]
- [7] f, “L*a*b* color.” <http://a-cordosdias.blogspot.com/2010/12/lab-color.html>. Accessed:2021-07-08. [Quoted on p. 9]
- [8] A. Pereira, P. Carvalho, G. Coelho, and L. Côrte-Real, “Efficient ciede2000-based color similarity decision for computer vision,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 30, no. 7, pp. 2141–2154, 2020. [Quoted on p. 10]
- [9] I. E. Khadiri, A. Chahi, Y. E. Merabet, Y. Ruichek, and R. Touahni, “Local directional ternary pattern: A new texture descriptor for texture classification,” *Comput. Vis. Image Underst.*, vol. 169, pp. 14–27, 2018. [Quoted on p. 11]

- [10] T. Ojala, M. Pietikäinen, and D. Harwood, "A comparative study of texture measures with classification based on featured distributions," *Pattern Recognition*, vol. 29, no. 1, pp. 51–59, 1996. [Quoted on p. 11]
- [11] P. Mohanaiah, P. Sathyanarayana, and L. GuruKumar, "Image texture feature extraction using glcm approach," 2013. [Quoted on p. 11]
- [12] K. Pushpalatha, A. Karegowda, and D. Ramesh, "Identification of similar looking bulk split grams using glcm and cglcm texture features," *International Journal of Computer Applications*, vol. 167, pp. 30–36, 2017. [Quoted on p. 11]
- [13] D. Zhu, R. Pan, W. Gao, and J. Zhang, "Yarn-dyed fabric defect detection based on autocorrelation function and glcm," *Autex Research Journal*, vol. 15, no. 3, pp. 226–232, 2015. [Quoted on p. 11]
- [14] T. Ojala, M. Pietikainen, and T. Maenpaa, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 7, pp. 971–987, 2002. [Quoted on p. 11]
- [15] X. Tan and B. Triggs, "Fusing gabor and lbp feature sets for kernel-based face recognition," pp. 235–249, 11 2007. [Quoted on p. 12]
- [16] R. Nosaka, Y. Ohkawa, and K. Fukui, "Feature extraction based on co-occurrence of adjacent local binary patterns," pp. 82–91, 11 2011. [Quoted on p. 12]
- [17] C. Shan, "Learning local binary patterns for gender classification on real-world face images. pattern recogn. lett. 33(4), 431-437," *Pattern Recognition Letters*, vol. 33, pp. 431–437, 03 2012. [Quoted on p. 12]
- [18] J. Abukhait, "An automated surface defect inspection system using local binary patterns and co-occurrence matrix textures based on svm classifier," 2018. [Quoted on p. 12]
- [19] D. Zhang and G. Lu, "Review of shape representation and description techniques," *Pattern Recognition*, vol. 37, no. 1, pp. 1–19, 2004. [Quoted on p. 12]
- [20] I. T. Young, J. E. Walker, and J. E. Bowie, "An analysis technique for biological shape. i," *Information and Control*, vol. 25, no. 4, pp. 357–370, 1974. [Quoted on p. 12]
- [21] N. Salih and D. Ngo, "A novel method for shape representation," 01 2005. [Quoted on p. 13]
- [22] M.-K. Hu, "Visual pattern recognition by moment invariants," *IRE Transactions on Information Theory*, vol. 8, no. 2, pp. 179–187, 1962. [Quoted on p. 13]

- [23] S. Mallick, "Shape matching using hu moments (c++/python)." <https://learnopencv.com/shape-matching-using-hu-moments-c-python/>. Accessed:2021-07-08. [Quoted on p. 13]
- [24] T. Pavlidis, "A review of algorithms for shape analysis," *Computer Graphics and Image Processing*, vol. 7, no. 2, pp. 243–258, 1978. [Quoted on p. 14]
- [25] S. Loncaric, "A survey of shape analysis techniques," *Pattern Recognition*, vol. 31, no. 8, pp. 983–1001, 1998. [Quoted on p. 14]
- [26] W. J. Chen, J.-H. Ho, K. B. Mustapha, and T.-Y. Chai, "A vision based system for anomaly detection and classification in additive manufacturing," in *2019 IEEE Conference on Sustainable Utilization and Development in Engineering and Technologies (CSUDET)*, pp. 87–92, 2019. [Quoted on p. 14]
- [27] H. Mu and D. Qi, "Pattern recognition of wood defects types based on hu invariant moments," in *2009 2nd International Congress on Image and Signal Processing*, pp. 1–5, 2009. [Quoted on p. 14]
- [28] H. Jia, Y. Murphey, J. Shi, and T.-S. Chang, "An intelligent real-time vision system for surface defect detection," in *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.*, vol. 3, pp. 239–242 Vol.3, 2004. [Quoted on p. 15]
- [29] T.-S. Li and C.-L. Huang, "Defect spatial pattern recognition using a hybrid som–svm approach in semiconductor manufacturing," *Expert Systems with Applications*, vol. 36, no. 1, pp. 374–385, 2009. [Quoted on p. 15]
- [30] Y. Fujita and Y. Hamamoto *Machine Vision and Applications*, vol. 22, no. 2, pp. 245–254, 2011. [Quoted on p. 16]
- [31] R. Dancel, "Case study paper on additive manufacturing (3d printing technology)," 04 2019. [Quoted on p. 16]
- [32] J. Beaman, C. Atwood, T. Bergman, D. Bourell, S. Hollister, and D. Rosen, "Additive/subtractive manufacturing research and development in europe," p. 155, 12 2004. [Quoted on p. 17]
- [33] T. D. Ngo, A. Kashani, G. Imbalzano, K. T. Nguyen, and D. Hui, "Additive manufacturing (3d printing): A review of materials, methods, applications and challenges," *Composites Part B: Engineering*, vol. 143, pp. 172–196, 2018. [Quoted on p. 17]
- [34] R. B. Kristiawan, F. Imaduddin, D. Ariawan, Ubaidillah, and Z. Arifin, "A review on the fused deposition modeling (fdm) 3d printing: Filament processing, materials, and printing parameters," *Open Engineering*, vol. 11, no. 1, pp. 639–649, 2021. [Quoted on p. 17]

- [35] P. D. Printer, "Fused deposition modeling (fdm) 3d printing – simply explained." <https://pick3dprinter.com/fdm-3d-printing/>. Accessed:2021-07-08. [Quoted on p. 18]
- [36] O. Bowoto, B. Oladapo, A. Zahedi, F. Omigbodun, and O. Emenuvwe, "Analytical modelling of in situ layer-wise defect detection in 3d-printed parts: additive manufacturing," *The International Journal of Advanced Manufacturing Technology*, vol. 111, pp. 1–11, 12 2020. [Quoted on p. 18]
- [37] Simplify3D, "Print quality troubleshooting guide." <https://www.simplify3d.com/support/print-quality-troubleshooting/>. Accessed: 2021-07-08. [Quoted on p. 19]
- [38] SVS-VISTEK, "exo264cge exo gige." <https://www.svs-vistek.com/en/industrial-cameras/svs-camera-detail.php?id=exo264CGE>. Accessed: 2021-07-08. [Quoted on p. 22, 23]
- [39] JaraStudio, "Newer dimmable bi-color." <http://www.jarastudio.es/en/rental/image/newer-dimmable-bi-color-led-professional-video-light-190-centimeters-adjustable-light-stand/>. Accessed:2021-07-08. [Quoted on p. 24]
- [40] BECKHOFF, "C6930-0050 | control cabinet industrial pc." <https://www.beckhoff.com/en-en/products/ipc/pcs/c69xx-compact-industrial-pcs/c6930-0050.html>. Accessed: 2021-07-08. [Quoted on p. 24, 25]
- [41] BECKHOFF, "Twincat automation software." https://www.beckhoff.com/en-en/products/automation/twincat/#text_bild_2. Accessed: 2021-07-08. [Quoted on p. 25]
- [42] BECKHOFF, "Ads-communication." https://infosys.beckhoff.com/english.php?content=../content/1033/cx8190_hw/5091854987.html&id=. Accessed: 2021-07-08. [Quoted on p. 25, 26]
- [43] BECKHOFF, "Tf7xxx |twincat 3 vision." <https://www.beckhoff.com/en-en/products/automation/twincat/tfxxxx-twincat-3-functions/tf7xxx-tc3-vision/>. Accessed: 2021-07-08. [Quoted on p. 26]
- [44] BECKHOFF, "Creating a gige vision camera." https://infosys.beckhoff.com/english.php?content=../content/1033/tf7xxx_tc3_vision/18014403990062091.html&id=. Accessed: 2021-07-08. [Quoted on p. 26]
- [45] BECKHOFF, "Image display in ads image watch." https://infosys.beckhoff.com/english.php?content=../content/1033/tf7xxx_tc3_vision/18014403990062091.html&id=. Accessed: 2021-07-08. [Quoted on p. 27]

- [46] R. Meireles, "Interface com computador por controlo visual de cursores." https://web.fe.up.pt/~ee97107/Relatorio_de_Projecto_FINAL_PARTE_2.pdf. Accessed:2021-07-08. [Quoted on p. 31, 32]
- [47] BECKHOFF, "Calibration assistant." https://infosys.beckhoff.com/english.php?content=../content/1033/tf7xxx_tc3_vision/27021603042171531.html&id=. Accessed:2021-07-08. [Quoted on p. 33]
- [48] G. B. e Adrian Kaehler, "Learning opencv." <https://www.bogotobogo.com/cplusplus/files/OReilly%20Learning%20openCV.pdf>. Accessed:2021-07-08. [Quoted on p. 35]
- [49] A. M. e Sakshi Seth, "Otsu's thresholding with opencv." <https://learnopencv.com/otsu-thresholding-with-opencv/>. Accessed:2021-07-08. [Quoted on p. 35]
- [50] L. Torok, "Método de otsu." <https://learnopencv.com/otsu-thresholding-with-opencv/>. Accessed:2021-07-08. [Quoted on p. 35]
- [51] BeckhOff, "F_vn_cannyedgedetection." https://infosys.beckhoff.com/english.php?content=../content/1033/tf7xxx_tc3_vision/9007207517370763.html&id=1898124676303280624. Accessed:2021-07-08. [Quoted on p. 39]
- [52] Y. Zhu and C. Huang, "An improved median filtering algorithm for image noise reduction," *Physics Procedia*, vol. 25, pp. 609–616, 2012. International Conference on Solid State Devices and Materials Science, April 1-2, 2012, Macao. [Quoted on p. 42]
- [53] K. Boateng, B. Weyori, and D. Laar, "Improving the effectiveness of the median filter," *International Journal of Electronics and Communication Engineering*, vol. 5, pp. 85–97, 01 2012. [Quoted on p. 42]
- [54] D. Chudasama, T. Patel, S. Joshi, and G. Prajapati, "Image segmentation using morphological operations," *International Journal of Computer Applications*, vol. 117, pp. 16–19, 05 2015. [Quoted on p. 44]
- [55] OpenCV, "Basic thresholding operations." https://docs.opencv.org/3.4/db/d8e/tutorial_threshold.html. Accessed:2021-07-08. [Quoted on p. 51]

