



VISÃO ARTIFICIAL NO CONTROLO DE QUALIDADE

JOSÉ PEDRO GUEDES BORGES

novembro de 2021

VISÃO ARTIFICIAL NO CONTROLO DE QUALIDADE

José Pedro Guedes Borges



Departamento de Engenharia Eletrotécnica
Mestrado em Engenharia Eletrotécnica – Automação e Sistemas

2021

Relatório elaborado para satisfação parcial dos requisitos da Unidade Curricular de TEDI -
Dissertação do Mestrado em Engenharia Eletrotécnica – Automação e Sistemas

Candidato: José Pedro Guedes Borges, Nº 1151419, 1151419@isep.ipp.pt

Orientação científica: Francisco José Dias Pereira, fdp@isep.ipp.pt

Coorientação científica: Carla Manuela Alves Pinto, cap@isep.ipp.pt

Empresa: Continental Advanced Antenna

Supervisão: Paulo Nogueira, paulo.nogueira@continental-corporation.com



Departamento de Engenharia Eletrotécnica
Mestrado em Engenharia Eletrotécnica – Automação e Sistemas

2021

Dedico este trabalho à minha Família, em especial aos meus Pais.

Agradecimentos

Este trabalho, apesar de individual, é sempre feito com bastante apoio diretamente e indiretamente, contudo apesar de um obrigado ser pouco, deixo aqui alguns dos meus agradecimentos.

Em primeiro lugar, os meus agradecimentos ao orientador da dissertação Professor Francisco José Dias Pereira pelo apoio, estímulo e confiança demonstrada, que foi, sem dúvida, uma chave essencial para a conclusão do meu projeto.

À professora Carla Manuela Alves Pinto, como Coorientadora neste projeto, procedeu à avaliação das ideias desenvolvidas e ajudou na elaboração da tese.

Ao Eng.º Paulo Nogueira representante da Continental Advanced Antenna, como Coorientador pela oportunidade de desenvolver um projeto relacionado com a empresa.

Agradecer a toda a equipa da Continental que me recebeu e me ajudou ao longo destes últimos meses, nomeadamente aos engenheiros Edmundo Soares, Carlos Macedo e Daniel Carvalho dando-me auxílio, exemplos de profissionalismo, bem-estar e trabalho de equipa.

Devo gratular o Instituto Superior de Engenharia do Porto pelo excelente ensino que me foi proporcionado e pelas pessoas com quem partilhei várias experiências ao longo destes anos, tanto amigos como colegas de curso, todos eles contribuíram de alguma forma para o meu sucesso.

Mas, acima de tudo, quero agradecer aos meus pais e a toda a minha família por me terem apoiado e partilhado esta dura caminhada, que sem eles não tinha sido possível. E pelo esforço investido ao longo de toda a minha vida para que eu tivesse as melhores condições, sem nunca me terem faltaram com nada.

Resumo

Atualmente, com o crescimento da indústria, a necessidade de automatização de processos industriais aumenta, como é o caso da inspeção das peças resultantes da produção numa fábrica, pois são exigidos níveis elevados de qualidade no produto, produtividade e capacidade de melhoria contínua.

Um dos setores que mais tem se desenvolvido nos últimos anos é a indústria automóvel, e como consequência a produtividade e a mão de obra. Este desenvolvimento tem levado a uma constante evolução de tecnologias, conceitos e técnicas com o intuito de dar resposta à enorme competitividade e satisfação de procura gerada. Contudo, este desenvolvimento gera uma crescente complexidade e importância de máquinas/robôs nos processos produtivos, visto que as paragens de produção são cada vez mais dispendiosas. Deste modo, as empresas tentam implementar medidas que permitam reduzir esses tempos de paragem e, conseqüentemente, todos os custos a elas associados.

A dissertação apresentada tem como objetivo a melhoria no processo produtivo de alimentação autónoma de parafusos nos robôs de aparafusamento de antenas. Trata-se de um processo que apresenta algumas dificuldades, visto ser preciso os parafusos estarem sempre com uma determinada orientação, caso contrário bloquearia a máquina e conseqüentemente a produção.

Neste sentido, o *software* desenvolvido foi concebido para uma câmara de visão artificial, que garante que os parafusos seguem com a posição correta para que sejam enviados para o robô. Quando a posição não é a correta, e com a ajuda de uma saída de ar comprimido, os parafusos são expulsos novamente para o alimentador. Desta forma, pretendeu-se criar uma solução mais flexível, com menores e menos paragens, e para uma adaptação mais rápida, tanto para novas referências de parafusos como também para outras funções.

Palavras-Chave

Indústria automóvel, Visão artificial, Automação industrial, Aparafusamento, Robótica

Abstract

Nowadays, with the growth of the industry, the need for automation of industrial processes increases, as is the case of the inspection of parts resulting from production in a factory, as high levels of product quality, productivity and capacity for continuous improvement are required.

One of the sectors that has been developing the most in recent years is the automotive industry, and consequently, productivity and labor. In favor of this development, it leads to a constant evolution of technologies, concepts, and techniques to respond to the enormous competitiveness and satisfaction of the demand generated.

However, this development generates an increasing complexity and importance of machines/robots in the production processes, as production stops are increasingly expensive. In this way, companies try to implement measures to reduce these downtimes and, consequently, all costs associated with them.

The dissertation presented aims to improve the productive process of autonomous screw feeding in antenna screwing robots. This is a process that presents certain difficulties, since it is screws must always be in a particular orientation, otherwise block the machine and consequently production.

In this sense, the software developed was designed for an artificial vision camera, which ensures that the screws follow the correct position to be sent to the robot. When the position is not correct, with the help of a compressed air outlet the screws are expelled back to the feeder. In this way, creating a more flexible solution, with fewer and fewer stops, and for faster adaptation, both for new screw references and for other functions.

Keywords

Automotive industry, Artificial vision, Industrial automation, Screwing, Robotic.

Índice

AGRADECIMENTOS	VII
RESUMO	IX
ABSTRACT	XI
ÍNDICE	XIII
ÍNDICE DE FIGURAS	XV
ÍNDICE DE TABELAS	XIX
ACRÓNIMOS	XXI
1. INTRODUÇÃO	1
1.1.CONTEXTUALIZAÇÃO.....	1
1.2.GRUPO CONTINENTAL	3
1.3.OBJETIVOS	5
1.4.CALENDARIZAÇÃO.....	6
1.5.ORGANIZAÇÃO DO RELATÓRIO	6
2. ESTADO DE ARTE	9
2.1.VISÃO ARTIFICIAL.....	9
2.2.SISTEMA DE VISÃO ARTIFICIAL.....	10
2.3.APLICAÇÕES DA VISÃO ARTIFICIAL.....	14
3. CONSTITUIÇÃO DE UM SISTEMA DE VISÃO	21
3.1.SISTEMA ÓTICO	22
3.2.SISTEMA DE ILUMINAÇÃO.....	29
3.3.CÂMARAS DE VISÃO ARTIFICIAL	35
3.4.INTERFACES DE COMUNICAÇÃO.....	40
3.5.FORNECEDORES/SOFTWARE	41
4. PROJETO DESENVOLVIDO	43
4.1.HARDWARE.....	43
4.2.SOFTWARE.....	49
5. OUTRO TIPO DE SOLUÇÕES	81
5.1.VIBRADOR COM SISTEMA DE ORIENTAÇÃO POR PÊNDULO GRAVÍTICO.....	81
5.2.COMPRESSÃO DAS MOLAS.....	82

6. TESTES E RESULTADOS	85
6.1.METODOLOGIA DE TESTE.....	85
6.2.ALTERAÇÕES EFETUADAS	89
6.3.RESULTADOS OBTIDOS	91
7. CONCLUSÕES.....	93
7.1.CONCLUSÕES.....	93
7.2.PROPOSTA DE TRABALHOS FUTUROS	94
REFERÊNCIAS BIBLIOGRÁFICAS	95
ANEXO A. INICIALIZAÇÃO DA SMART CAMERA BANNER PRESENCEPLUS_{P4} OMNI	101
ANEXO B. ESPECIFICAÇÕES DA SMART CAMERA BANNER PRESENCEPLUS_{P4} OMNI ...	111
ANEXO C. CÓDIGO FONTE DA INTERFACE GRÁFICA CRIADA NO VISUAL STUDIO ...	119

Índice de Figuras

Figura 1	Continental <i>Advanced Antenna</i> (esquerda), Continental-Mabor (direita)	4
Figura 2	a) Sensor de visão [9], b) Câmara inteligente [10], c) Sistema de visão artificial [11]	11
Figura 3	Exemplo de resultado obtido por um sensor de visão [12]	12
Figura 4	Exemplo de aplicação da câmara inteligente [16]	13
Figura 5	Exemplo de um Sistema de Visão Integrado [18]	14
Figura 6	Análise da largura das linhas e de marcas da serra de corte [21]	16
Figura 7	Análise de fruta por tamanho e cor [22]	16
Figura 8	Análise de Solda em PCB [23]	17
Figura 9	Visualização do interior da embalagem através de visão Raio-X [22]	17
Figura 10	Inspeção de uma válvula [24]	18
Figura 11	Sistema de visão artificial [16]	22
Figura 12	Distância Focal [30]	23
Figura 13	Distância de trabalho [31]	23
Figura 14	Abertura da Íris [33]	24
Figura 15	Profundidade de campo [34]	24
Figura 16	a) Aberração Esférica; b) Aberração Coma; c) Aberração Cromática; d) Distorção positiva (esquerda) e negativa (direita) [35]	25
Figura 17	Tipos de lentes existentes [38]	26

Figura 18	Filtro passa-faixa (luz verde) [41]	27
Figura 19	Filtro Polarizador [41]	28
Figura 20	Filtro Ultravioleta [42]	28
Figura 21	Resultados da utilização de diferentes técnicas de iluminação [43]	29
Figura 22	Comparação dos diferentes tipos de iluminação [46]	30
Figura 23	Iluminação Campo Escuro (<i>Dark Field</i>) [49]	31
Figura 24	Iluminação Lateral ou em Barra [49]	31
Figura 25	Iluminação de Fundo (<i>Backlight</i>) [49]	32
Figura 26	Iluminação no Eixo (coaxial) difusa [49]	32
Figura 27	Iluminação de Cúpula ou Difusa [49]	33
Figura 28	Iluminação de Anel [49]	33
Figura 29	Iluminação Estruturada [50]	34
Figura 30	Tamanho dos diferentes tipos de sensores [55]	38
Figura 31	<i>Frame Grabber</i> [58]	39
Figura 32	Fluxograma sobre as ligações de <i>hardware</i> utilizado	44
Figura 33	Equipamentos usados no projeto	44
Figura 34	<i>Smart Camera</i> Banner PresencePlus _{p4} OMNI [10]	45
Figura 35	Conectores e indicadores da <i>Smart Camera</i> [59]	46
Figura 36	<i>Pinout</i> da <i>Smart Camera</i>	46
Figura 37	Cabo RJ45 <i>Ethernet</i> STP-MAQDC-806, para sistemas de visão [59][60]	47
Figura 38	Objetiva <i>Computar</i> utilizada [61]	48

Figura 39	Iluminador <i>PresencePlus</i> utilizado [62]	48
Figura 40	Objetivo do projeto – Orientação Correta	49
Figura 41	Ecrã principal do utilizador	50
Figura 42	Ecrã de programação	51
Figura 43	Configurações do endereço IP do PC	52
Figura 44	Configurações do endereço IP da <i>Smart Camera</i>	53
Figura 45	Tempo de exposição e o valor do Ganho utilizado	54
Figura 46	Ferramentas existentes para programação da <i>Smart Camera</i>	54
Figura 47	Fluxograma 1ªTentativa	56
Figura 48	Comando <i>Geometric Find Tool</i> para deteção de padrão	57
Figura 49	Comando <i>Circle Detect Tool</i> para detetar as cabeças dos parafusos (esquerda) e o diâmetro do corpo (direita)	59
Figura 50	Comando <i>Test Tool</i> para definir os intervalos de aceitação	60
Figura 51	Fluxograma 2ªTentativa	61
Figura 52	Comando <i>Geometric Find Tool</i> para deteção de padrão	62
Figura 53	Comando <i>Circle Detect Tool</i> para detetar as cabeças dos parafusos	63
Figura 54	Comando <i>Test Tool</i> para definir intervalo de aceitação	64
Figura 55	Resultado obtido depois da alteração do ganho e do tempo de exposição	65
Figura 56	Fluxograma do programa “Parafuso Deitado”	66
Figura 57	Comando <i>Line Detect Tool</i> para deteção do corpo do parafuso	68
Figura 58	Comando <i>Test Tool</i> para definir intervalo de aceitação do tamanho do corpo do parafuso	69

Figura 59	Configuração das saídas I/O	70
Figura 60	Interface Gráfica	71
Figura 61	Ecrã Principal	72
Figura 62	Parâmetros a preencher e <i>Warnings</i> no Ecrã Config	74
Figura 63	Ecrã Config	79
Figura 64	Início do processo [63]	81
Figura 65	Rotação de 180 de peça mal orientada (esquerda), Queda de peça bem orientada, sem sofrer mudança de orientação. (direita) [63]	82
Figura 66	Sistema de molas à compressão [63]	82
Figura 67	Sistema de molas à compressão desmontado [63]	83
Figura 68	Ação da mola, peça bem orientada (esquerda) e mal orientada (direita) [63]	83
Figura 69	Posicionamento da câmara na linha de produção	86
Figura 70	Alteração do tempo de exposição e do ganho do sensor	87
Figura 71	Ajustes das ferramentas de análise	88
Figura 72	Resultado obtido pelo programa “Deteção das cabeças”	88
Figura 73	Resultado obtido pelo programa “Deteção do corpo”	89
Figura 74	Resultado obtido com e sem o círculo “extra”	90
Figura 75	Condições de análise da Smart Camera	91

Índice de Tabelas

Tabela 1	Calendarização do Projeto	6
Tabela 2	Tipos de iluminação sugerida para várias aplicações [51]	35
Tabela 3	Características das interfaces de comunicação [16]	40
Tabela 4	Comparação dos resultados obtidos por ambos os sistemas	92

Acrónimos

- A/D – *Analog/Digital*
- CCD – *Charge Coupled Device*
- CMOS – *Complementary Metal Oxide Semiconductor*
- DOF – *Depth of field*
- FPA – *Focal Plane Array*
- FPS – *Frames per second*
- GigE – *Giga Ethernet*
- I/O – *Input/Output*
- IP – *Internet Protocol*
- LED – *Light Emitting Diode*
- OCR – *Optical Character Recognition*
- PC – *Personal Computer*
- PCB – *Printed Circuit Board*
- PCBA – *Printed Circuit Board Assembly*
- PLC – *Programmable Logic Controller*
- PoE – *Power Ethernet*
- RAM – *Random Access Memory*
- RCC – *Remote Command Channel*

ROI – *Region of interest*

USB – *Universal Serial Bus*

UV – *Ultra Violet*

WD – *Work Distance*

1. INTRODUÇÃO

O projeto final de curso apresentado foi realizado no âmbito da unidade curricular de TEDI(Tese/Dissertação) do Mestrado de Engenharia Eletrotécnica e Computadores (MEEC), do Departamento de Engenharia Eletrotécnica (DEE), do Instituto Superior de Engenharia do Porto. A empresa onde decorreu o estágio foi na Continental Advanced Antenna, que está centrada no ramo da produção de antenas automóveis.

1.1. CONTEXTUALIZAÇÃO

Este projeto surgiu de uma proposta feita pela empresa em questão, após ter mostrado interesse em realizar o estágio curricular na mesma. A Continental Advanced Antenna, fundada em 2019, tem como missão o desenvolvimento de antenas inteligentes para o ramo automóvel.

O projeto desenvolvido enquadra-se na área de automação industrial, mais propriamente na área de visão artificial. Consiste na programação de uma *Smart Camera* para a deteção da orientação de parafusos, que alimentam um robô aparafusador. Os parafusos têm de ir com uma certa orientação, cabeça do parafuso virada para cima, e a *Smart Camera* irá detetar esta correta orientação e impedir que o robô fique bloqueado e que a produção seja prejudicada.

A aplicação deste projeto, contribuirá significativamente para um aparafusamento mais eficaz das antenas produzidas na fábrica e evitará que seja necessário parar a produção para resolver o problema do robô bloqueado.

No início da dissertação, irá ser feita uma breve introdução à indústria de componentes automóveis e a sua interligação com a visão artificial. A indústria de componentes automóveis é fortemente pressionada para cumprir requisitos de elevada competitividade e qualidade. Esta combinação de fatores exige um elevado grau de automatização e, por outro lado, um forte investimento em controlo de qualidade. Em Portugal, devido ao baixo custo da mão-de-obra não qualificada, é comum recorrer-se a pouca automatização dos processos e à grande intervenção humana. Contudo, torna-se mais difícil garantir a máxima qualidade na produção.

Normalmente, falar em automatização é sinónimo de elevado investimento, o que retrai uma grande parte dos investidores nacionais. No entanto, existem as empresas multinacionais que possuem, na sua maioria, uma abertura de espírito para a automatização de processos, sabendo analisar com rigor qual o período necessário para o retorno do investimento (*Payback*). Assim, estas empresas recorrem cada vez mais a processos automáticos que permitem uma inspeção rigorosa e integral de todos produtos, garantindo um melhor controlo de qualidade. É neste contexto que a visão industrial se insere e se aplica cada vez mais.

Na visão artificial a nível industrial, o princípio baseia-se em utilizar informação visual para distinguir o que é útil do que não é, num determinado processo de fabrico. Uma solução só pode ser implementada em casos onde a aprovação ou rejeição de produtos não se baseia em atributos subjetivos ou não mensuráveis. A utilização da visão industrial permite desenvolver aplicações de elevada complexidade e fiabilidade usando um baixo número de componentes.

Esta área tem evoluído de tal forma que, atualmente, é muito usual encontrar aplicações em vários ramos da indústria, como por exemplo na indústria alimentar (cor e medição de fruta), na indústria da reciclagem (classificação de plásticos), na indústria automóvel (classificação e medida de componentes), entre outras inúmeras aplicações.

1.2. GRUPO CONTINENTAL

A Continental é seguramente uma das mais prestigiadas e reconhecidas marcas mundiais de pneus para automóveis e motos, sendo a criadora do pneu mais seguro do mundo.

A 8 de outubro de 1871, um grupo de nove banqueiros e industriais fundou o Grupo Continental. Inicialmente, a empresa produzia pequenas peças de borracha maleável e pneus maciços para bicicletas. Em outubro de 1882, a empresa afirmou a sua posição no mercado com o registo de uma marca própria preconizada na figura de um símbolo com um “cavalo rampante”. A imagem original registada em 1892 mantém-se até à atualidade, com pequenas variações nas cores e no slogan, mas sempre fiel à imagem original, tendo assim atravessado gerações.

Em 1898, a empresa iniciou a sua produção de pneus lisos para automóveis e deparou-se com problemas e necessidades específicas dos seus clientes e consumidores finais, ao nível da anti derrapagem e da dificuldade na troca de pneus. Com efeito, em 1904 produziu o primeiro pneu com desenho na banda de rodagem e, em 1905, o primeiro pneu antiderrapante. Mais tarde, em 1908 surgiu a roda desmontável, mais fácil de montar e desmontar [1].

Durante os anos 20 e 30 a Continental fundiu-se com diferentes empresas de borracha alemãs, criando-se a Continental Gummi-Werke AG. Esta fusão levou ao fabrico de novos pneus, mais resistentes pois eram reforçados com arames, mais confortáveis e duradouros. Por fim, ainda registaram uma patente para o fabrico de pneus sem câmara-de-ar.

Já nos anos 80 reforçou o seu domínio no continente europeu com a aquisição da marca de pneus austríaca Semperit e reforçou a sua liderança também no continente americano com a compra da fabricante norte-americana de pneus General Tire.

Durante os anos 90, a Continental registou a sua entrada em Portugal, juntamente com a Mabor, permitindo a produção de pneus em Lousado, Portugal, dando o nome de Continental – Mabor (Figura 1), instalando-se assim na península ibérica e na Europa

ocidental. Já nos fins dos anos 90, adquiriu a gigante dos freios e chassis Automotive Brake & Chassis, tornando-se pioneira no sistema de freio anti travagem (ABS).

No século 21, continuou a expandir-se cada vez mais. Conquistando o continente asiático com a aquisição da especialista em eletrónica Temic. Reforçou a sua posição no setor da eletrónica automóvel com uma parceria com a Motorola, permitindo a integração de tecnologias de instrumentação, conectividade, medidores, mostradores e rádio.

Foi considerada o quarto fabricante de pneus no mundo segundo a revista alemã *Neue Reifenzeitung* em 2013. Abriu um novo centro de investigação e desenvolvimento em Silicon Valley, EUA, que contará com especialistas de todas as áreas da empresa a trabalhar para encontrar soluções pioneiras para a mobilidade sustentável do futuro.

Em 2019 fundou a Continental Advanced Antenna (Figura 1), resultante da aquisição da Kathrein Automotive pela Continental em fevereiro deste ano. Sendo esta unidade uma das principais especialistas e fabricantes mundiais de antenas para veículos de marcas de luxo, com a Audi, Daimler, Volvo, Porsche e BMW.

Em 31 de dezembro de 2019, a Continental empregava mais de 240.000 pessoas em 59 países e mercados [1].



Figura 1 **Continental *Advanced Antenna* (esquerda), Continental-Mabor (direita)**

1.3. OBJETIVOS

Este trabalho teve como principal objetivo otimizar um processo de montagem/fabrico de componentes para automóvel já existente. Este, apesar de ter já incorporada alguma automação, apresentava um elevado grau de imprevisibilidade de resultados.

Pretende-se para tal que a câmara de visão artificial seja capaz de fazer um teste a cada parafuso, através da análise da cabeça dos mesmos. Com este processo pretende-se que os parafusos sejam avaliados no que diz respeito à sua orientação e além disso que seja capaz de os rejeitar quando não cumpram os requisitos impostos, nomeadamente, a correta orientação. Isto diminuirá o número de vezes em que a máquina encrava, resultando na paragem do processo de aparafusamento das placas das antenas.

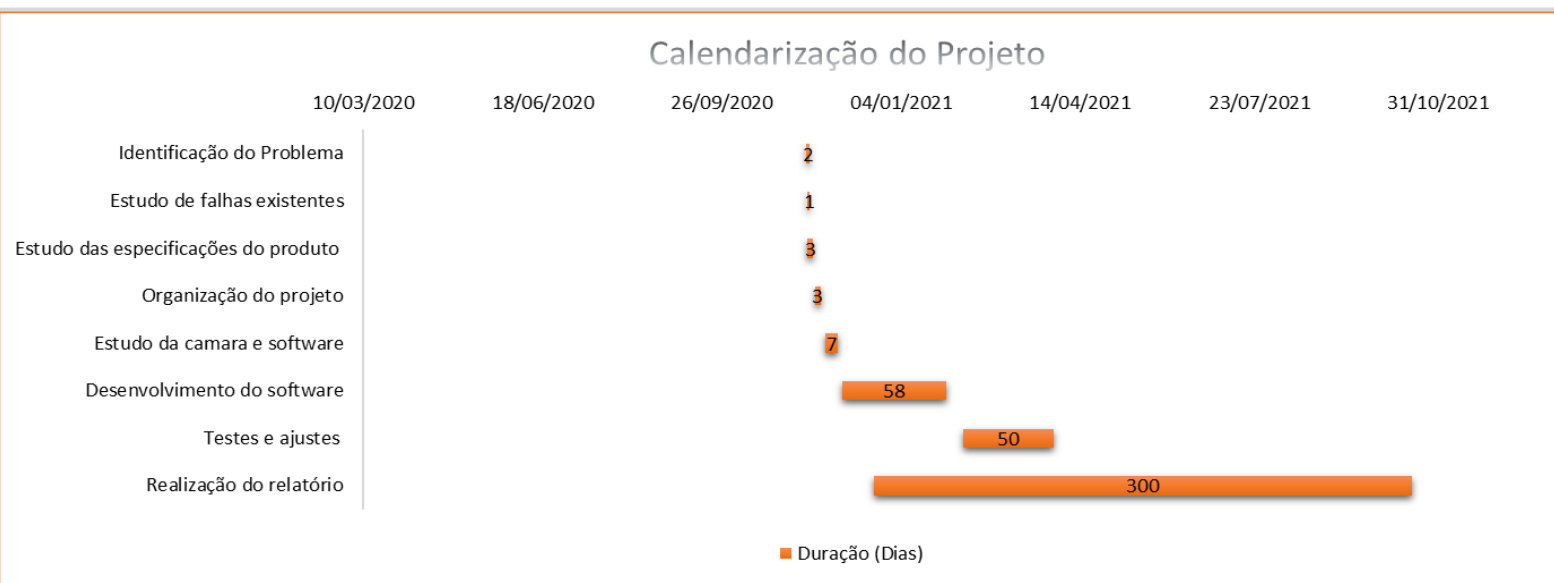
Para a prossecução do objetivo principal anteriormente referido, tornou-se necessário:

- Proceder à identificação do problema;
- Estudar convenientemente o processo já existente, assim como as falhas a ele associadas, tais como:
 - Falha na deteção do parafuso deitado, é dado como posição correta e deixa-o passar;
 - Por vezes o parafuso vai inclinado, de cabeça para baixo, e não é detetado pelo sensor.
- Determinar as propriedades geométricas do produto;
- Pesquisar soluções para problemas idênticos;
- Proceder ao anteprojecto das soluções;
- Projetar o equipamento a utilizar;
- Indicar os componentes de automação necessários ao correto funcionamento do sistema, e respectivas indicações para programação.

1.4. CALENDARIZAÇÃO

Este estágio empresarial teve início oficial no dia 9 de novembro de 2020 e fim no dia 31 de maio de 2021. Ao longo deste período foram cumpridas várias tarefas segundo a calendarização descrita na Tabela 1.

Tabela 1 Calendarização do Projeto



1.5. ORGANIZAÇÃO DO RELATÓRIO

No capítulo 1, introduz-se o projeto, iniciando-se pela sua contextualização, onde é explicada a génese do projeto e os objetivos a atingir. Posteriormente apresenta-se o problema, os objetivos e a calendarização do projeto. Descreve-se, ainda, a organização do relatório.

No capítulo 2, apresenta-se o enquadramento teórico, onde se faz a introdução à visão artificial. Sobre visão artificial, será abordada a sua história, aplicações existentes na indústria e a junção da visão com o controlo de qualidade.

No capítulo 3, continua-se o enquadramento teórico, onde se faz uma introdução aos componentes do sistema de visão industrial.

No capítulo 4, apresenta-se o projeto da câmara, assim como o desenvolvimento da interface gráfica, em termos de software, onde são explicados todos os passos e funções criadas para chegar ao produto final.

No capítulo 5, apresentam-se outros tipos de soluções existentes no mercado que se enquadram de certa forma nesta área e poderiam ser alternativas.

No capítulo 6, apresentam-se os testes e os resultados obtidos durante a realização experimental do projeto na linha de montagem.

No capítulo 7, apresentam-se as conclusões do projeto, onde se apresenta um balanço sobre o que houve de positivo com a realização deste projeto, todos os contratemplos que surgiram ao longo da realização do projeto, bem como possíveis melhorias a fazer.

2. ESTADO DE ARTE

Antes de realizar a parte prática, é necessário estudar os princípios teóricos e o estado da arte. Para isso, nesta secção foram reunidos os principais conceitos existentes que permitam saber o que é a Visão Artificial, como é feita atualmente, como é projetado um sistema deste tipo (*setup*, iluminação, processamento de imagem) e como está interligada com o controlo de qualidade industrial.

2.1. VISÃO ARTIFICIAL

A Inspeção Visual é um conceito que existe desde que o ser humano foi capaz de fabricar os seus próprios objetos, sendo apenas necessário observar a superfície deste para conseguir identificar falhas que pudessem prejudicar o objeto quer na parte estética, quer na parte funcional. Em muitas empresas, o controlo de qualidade ainda é uma operação manual. No entanto, embora a capacidade humana de inspecionar visualmente objetos diferentes seja muito alta, a subjetividade e a fadiga resultantes da execução de tarefas repetitivas podem resultar em erros humanos [2][3].

Nos tempos modernos, grande parte das fábricas já possuíam controlo de qualidade, isto para fazer uma avaliação às peças produzidas, procurando imperfeições ou incoerências geométricas. Quer seja por amostragem ou na totalidade das peças, os

fabricantes começaram a recorrer a sistemas de visão artificial para fazerem a inspeção visual.[2]

O início da utilização desta tecnologia permitiu criar soluções inovadoras dentro da automação industrial, pelo que houve uma enorme quantidade de processos de fabrico que beneficiaram da implementação de técnicas automatizadas de inspeção, tais como, deteção de imperfeições, inspeção geométrica, controlo de embalagens, classificação de produtos, inspeção de acabamento de superfície, análise de cor e textura, entre outros [2][4].

Nos últimos anos, esta tecnologia evoluiu de tal maneira que passou a ser uma componente essencial na área da automação, visto que, com a incorporação da comunicação Ethernet nas câmaras, em paralelo com outros elementos de visão, como por exemplos, iluminação, lentes, fez com que fosse possível a interligação com todo o processo fabril [5].

A visão artificial consiste num conjunto de ferramentas e métodos que nos permitem obter, analisar e processar imagens em tempo real com a ajuda de computadores que comunicam entre si [6]. Com isso, esses sistemas permitem a automação de inúmeras tarefas, enquanto geram uma grande quantidade de dados, e é com esses dados que gestores e operadores conseguem identificar erros durante a fase de processamento e assim tomar medidas de forma atempada.

Em suma, a visão artificial tem um papel decisivo na INDÚSTRIA 4.0, pois é indispensável para qualquer indústria já que está integrada em cada etapa do processo produtivo, desde o controlo de entrada de mercadorias através de leitura de códigos de barras ou *QR Code* até à análise do produto final, verificando se está dentro dos requisitos ou não. E, verificamos que é um elemento fundamental nos processos de automação industrial, visto que proporciona um processo de alta robustez de forma mais económica e um sistema de controlo da qualidade muito mais eficaz, já que tudo é feito automaticamente e sem a influência do cansaço humano na inspeção [4][7].

2.2. SISTEMA DE VISÃO ARTIFICIAL

Os sistemas de visão são dispositivos que utilizam sistemas de processamento de imagem e computadores para analisar uma imagem e, depois, tomar uma decisão, baseada nessa análise [8].

Conforme a topologia utilizada em cada caso, os sistemas de visão podem ser divididos em três categorias distintas (Figura 2):

- Sensores de visão;
- Câmaras inteligentes;
- Sistemas de visão artificial.

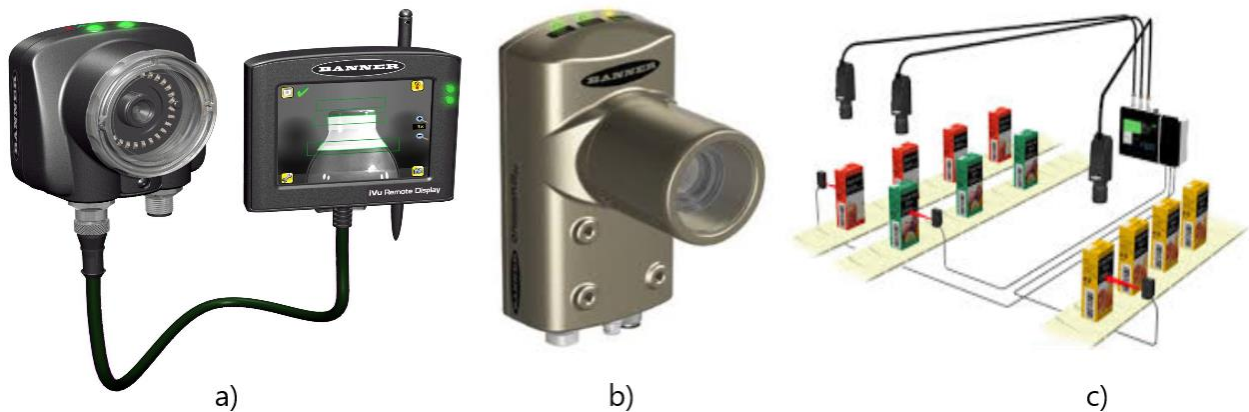


Figura 2 a) Sensor de visão [9], b) Câmara inteligente [10], c) Sistema de visão artificial [11]

2.2.1. SENSORES DE VISÃO

O sensor de visão é composto por células fotoelétricas que possuem a função de capturar uma imagem. Depois de capturar a imagem, o sensor de visão compara as características da imagem com uma imagem pré-definida armazenada na memória do sensor. No entanto, enquanto os sistemas de visão proporcionam dados, os sensores de visão só proporcionam resultados de aprovado/reprovado [12].

Este tipo de sensores está orientado para o mercado de cliente final, e supõe-se que o próprio cliente possa instalá-lo, sem ajuda ou necessidade de ajuste por parte de terceiros. Os sensores de visão estão pensados para resolver problemas concretos e fáceis, e para serem manuseados de forma muito simples, inclusive por pessoal não especializado em visão. Na maior parte das vezes, um botão de aprendizagem é o único mecanismo que o operador tem para controlá-lo, e um sistema de aprovado/reprovado é a única saída possível.

Além do mais, dada a potência limitada do hardware, não são capazes de processar mais do que algumas peças por segundo. Os programas que incluem os sensores de visão

não podem ser modificados, não requerem um PC para processar a imagem ou enviar resultados a um PLC (*Programmable Logic Controller*) ou outro elemento de controlo [13].

Normalmente, os sensores de visão só resolvem um problema concreto, como por exemplo, ler um código de barras ou matriz, verificar uma cor, identificar presença ou ausência, conferir tamanhos (Figura 3), entre outros.

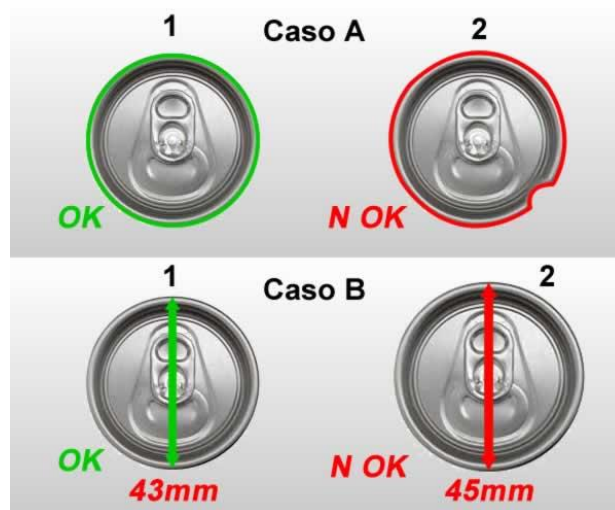


Figura 3 Exemplo de resultado obtido por um sensor de visão [12]

2.2.2. CÂMARAS INTELIGENTES

As câmaras inteligentes já têm uma potência de cálculo maior que os sensores, visto que incluem um sensor de captura, mesmo que seja num espaço muito reduzido, uma memória de armazenamento, um processador e os mecanismos de entrada/saída. Por vezes este tipo de câmara necessita de uma placa de entrada/saída adicional para que se possa conectar ao resto do sistema de automação da fábrica.

Contam com um grande conjunto de ferramentas de inspeção tais como a localização, contagem e medida de peças e ferramentas de deteção de defeitos (Figura 4). Estas ferramentas podem ser combinadas e utilizadas simultaneamente para resolver tarefas simples ou complexas, superando as capacidades de inspeção dos sensores de visão.

O material que constitui as câmaras inteligentes é muito robusto para poder suportar os ambientes tipicamente agressivos dos meios industriais [14][15].

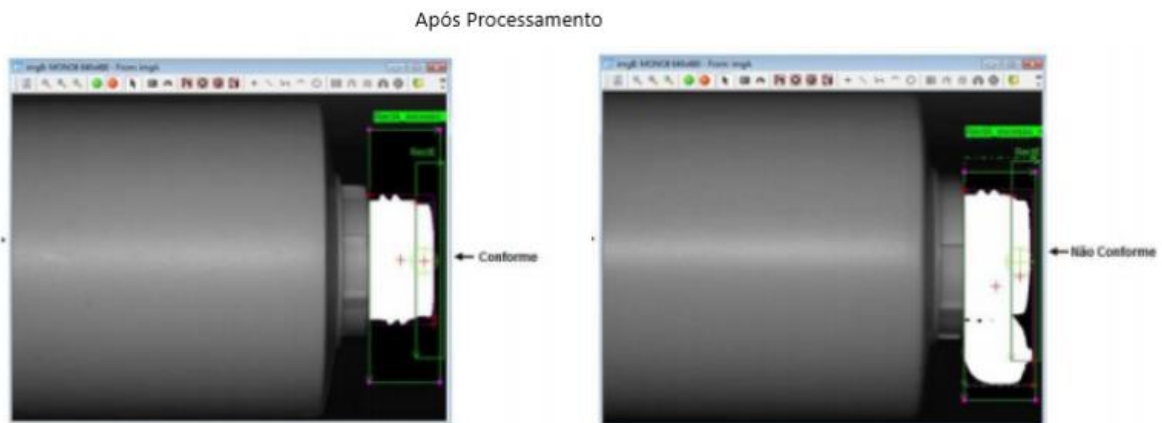


Figura 4 Exemplo de aplicação da câmara inteligente [16]

2.2.3. SISTEMAS DE VISÃO INTEGRADOS

Os sistemas de visão integrados, tal como as câmaras inteligentes, contam com uma potência de cálculo suficientemente elevada para poder resolver a maioria das aplicações de visão industrial.

O que difere os sistemas de visão integrados das câmaras inteligentes é que, estes não possuem os componentes de análise incorporados dentro da câmara. Visto que, tanto o sensor como a memória da câmara, o processador e as entradas/saídas (Figura 5), assim como a conexão com o resto do ecossistema industrial, localizam-se num componente à parte e que pode ser encontrado junto a outros elementos de automação [17].

Por outro lado, apresenta uma vantagem em comparação com as câmaras inteligentes, pois tem só um único elemento de processamento, que se pode conectar a diversas câmaras de visão remotamente. Isso permite reduzir o custo das aplicações quando estas requerem diferentes imagens da mesma peça [14][17].



Figura 5 Exemplo de um Sistema de Visão Integrado [18]

2.3. APLICAÇÕES DA VISÃO ARTIFICIAL

O sistema de visão usa câmaras para visualizar mecanicamente uma imagem específica do processo produtivo. Essa imagem será enviada para um computador com um software que usará algoritmos de visão computacional para processá-la e interpretá-la, instruindo outros elementos do sistema de produção a agir de acordo com esses dados.

De tal modo, que a visão artificial pode ser aplicada numa enorme variedade de necessidades industriais, visto poder ser instalada numa vasta área de setores industriais, entre eles:

- **Automóvel:** No ramo automóvel é usada para a inspeção de produtos durante a fase de fabrico e de montagem, nessa inspeção pode verificar soldas, aparafusamentos, pinturas, rebarbas, presença ou não de componentes, etc. Tornando assim possível identificar defeitos existentes nas peças e conseguir intervir a tempo para se conseguir um produto final de qualidade.
- **Alimentação:** No ramo da alimentação, é usada no controlo de qualidade dos alimentos verificando se estão em condições próprias para consumo, se a embalagem onde estão condicionados está bem lacrada e dentro da validade. Por outro lado, pode ser usada para seleccionar alimentos por cores e tamanhos e desta forma separá-los para diversos fins.
- **Recipientes e embalagens:** No ramo do *packing*, é usada para controlar se estão presentes marcadores obrigatórios nas embalagens dos produtos, tais como

números do lote, datas de validade, códigos de barra/*QR code*, etc. Do mesmo modo, pode verificar se as embalagens se encontram corretamente celadas, com as devidas tampas ou até verificar se o nível de líquido dos produtos está de acordo com o estipulado pela empresa.

- **Eletrónica:** No ramo da eletrónica, é usada para manipular e identificar diferentes componentes nas linhas de montagem ou no fabrico dos PCBAs (*Printed Circuit Board Assembly*) onde verificam a soldagem e a correta montagem dos componentes.
- **Logística:** Na logística, é usada no controlo da movimentação das peças, ou seja, criar uma base de dados onde fica registado quem faz, a hora e se foi aprovada/reprovada e endereçamento das cargas e materiais, verificando a leitura de placas de camiões, assim como contentores que transportam tanto as matérias-primas como qualquer outro tipo de componentes através da leitura de código de barras ou matriz, através do reconhecimento ótico de caracteres (OCR) [19][20].

2.3.1. EXEMPLOS DE APLICAÇÕES DE VISÃO ARTIFICIAL NA INDÚSTRIA

No ambiente industrial, há uma infinidade de aplicações, como a manutenção preditiva, a leitura de códigos de barras, o controlo da qualidade de fabrico, inspeção, orientação, medição e identificação de produtos entre outras.

Já se encontram implementados vários sistemas de visão artificial em diversas áreas da indústria para uma melhoria da qualidade do produto final, como por exemplo:

- **Controlo das *wafers* e células solares** – Permite controlar os defeitos que ocorrem na produção são células solares, entre eles, fendas ou micro-fendas, as marcas da serra de corte (Figura 6), fraturas nas laterais da *wafers*, a textura (inspeção das linhas) e o revestimento [21].

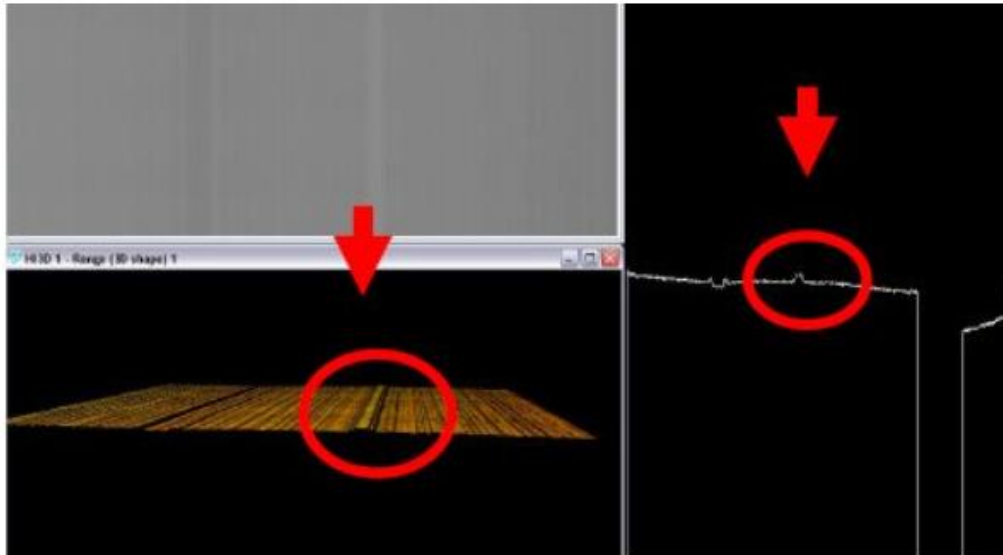


Figura 6 Análise da largura das linhas e de marcas da serra de corte [21]

- **Classificação de produtos por tamanho e cor** – Permite classificar todo o tipo de produto diretamente do tapete rolante. Verificando o tamanho, cor, formato (Figura 7) e comprovar a não existência de corpos estranhos como folhas, pedras, etc [22].

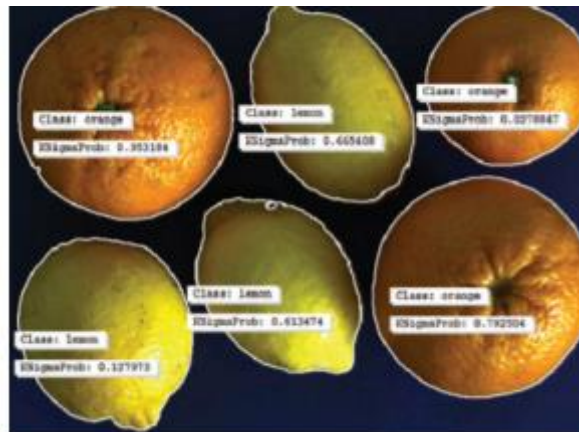


Figura 7 Análise de fruta por tamanho e cor [22]

- **Sistema de inspeção de soldas em PCB (Printed Circuit Board)** - Permite efetuar uma análise tridimensional de cada ponto de solda, sendo assim possível verificar a quantidade e a geometria da solda (Figura 7) e permite também avaliar a ausência de contacto entre as soldas e o PCB [23].

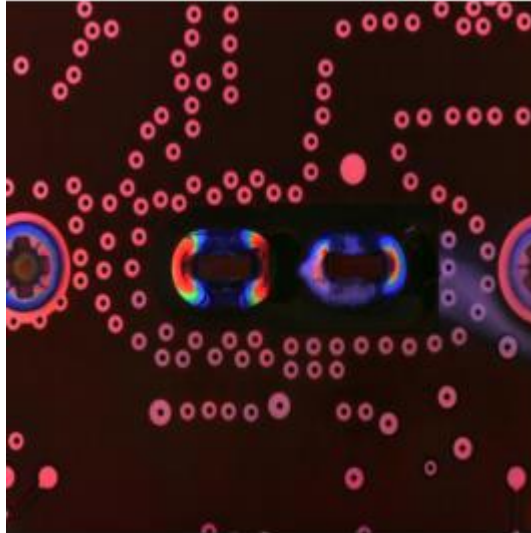


Figura 8 Análise de Solda em PCB [23]

- **Controle de produtos embalados** – Permite após o embalamento verificar a quantidade e posição de cada produto na embalagem quando não é possível a olho nu (Figura 9).



Figura 9 Visulaização do interior da embalagem através de visão Raio-X [22]

- **Inspecção De Válvulas De Automóveis** – Permite através do uso de câmaras lineares criar imagens planas de superfícies cilíndricas, como peças de automóveis para inspeção para garantir que a superfície da válvula está livre de fissuras (Figura 10) e que todas as juntas e filtros estão corretamente instalados e livres de defeitos [24].

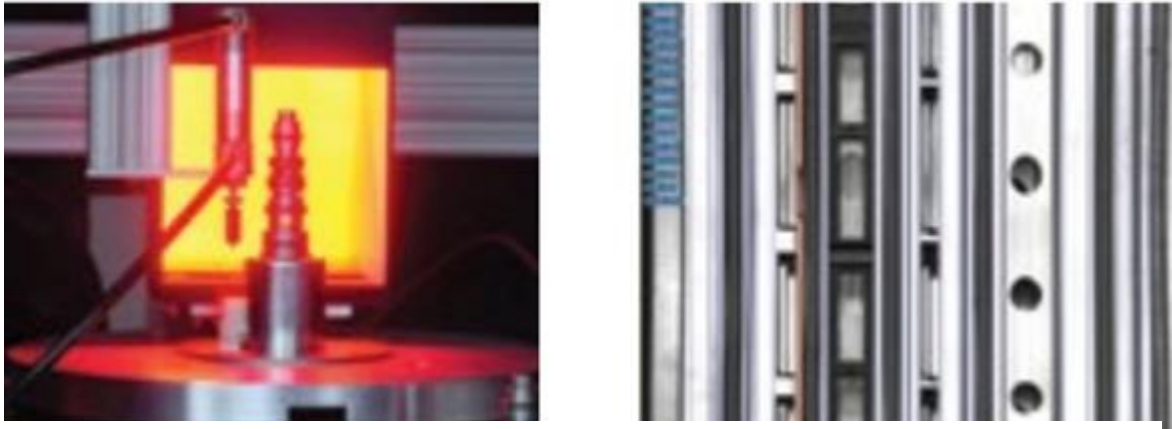


Figura 10 Inspeção de uma válvula [24]

2.3.2. DEEP LEARNING NA INDÚSTRIA

O *Deep Learning* é um dos ramos da Inteligência Artificial capaz de identificar padrões, tomar decisões ou modificar o seu comportamento, de forma autônoma, com base na análise de dados e de experiências, com o mínimo de interferência humana. Funciona através de configurações de parâmetros de dados e de reconhecimento padronizado de informação.

As redes neurais de *Deep Learning* têm como objetivo aumentar o poder computacional das máquinas e aplicações para as quais são desenvolvidas, para serem capazes de processar linguagem natural e entender elementos como voz, imagens e padrões de comportamento [25].

Na atualidade, está a começar a ser muito utilizada na indústria em práticas avançadas de inspeção de qualidade durante as diversas etapas de produção.

O *Deep Learning* pode ser usado na indústria de automação, podendo realizar operações de localização de peças, posição de elementos, classificação e reconhecimento de caracteres através de julgamentos prévios, com mais eficiência e assertividade do que com uma inspeção manual ou até de visão.

Na indústria farmacêutica, mais propriamente no processo de embalagem, visto que, no ciclo de encapsulamento dos medicamentos dentro dos Blisters, acontecem diversos problemas e falhas, tais como, problemas como mal posicionamento do medicamento, quebra, contaminação por algum objeto estranho, entre outros.

Na indústria de vidros, onde seria impossível um sistema de visão convencional identificar defeitos no processo de conformação, morfologia e outras variáveis do vidro, o *Deep Learning* surge como solução eficiente. Além de tornar possível o reconhecimento de defeitos de fabricação, a tecnologia tem a capacidade de identificar e aprender os defeitos causados pelo processo de fabrico, julgando de forma autônoma se o material tem a qualidade esperada ou se deve ser descartado [26].

2.3.3. VISÃO ARTIFICIAL NO CONTROLO DE QUALIDADE

O controlo de qualidade é uma questão primordial nos processos de fabrico, seja no segmento da indústria de alimentos, da farmacêutica, de aparelhos eletrónicos e, até mesmo, na prestação de serviços e no fornecimento de matéria-prima. O fato é que o consumidor final exige que o produto ou serviço esteja de acordo com as especificações técnicas que lhe conferem a qualidade esperada. Atualmente, muitas empresas efetuam o controlo de qualidade por meio de setores específicos para tal, com abordagens elaboradas na gestão de todo o processo produtivo até o estágio final do produto. Ainda que a qualidade seja um conceito exigido antes da revolução industrial, ganha cada vez mais relevância no atual contexto, fazendo com que todos os processos de fabrico sejam submetidos a um intenso controlo de qualidade [27].

A gestão da qualidade pode ser um grande diferencial para o sucesso de uma empresa, pois tem um impacto significativo na relação de compra e venda de um produto.

Para assegurar a qualidade de um produto começou-se a usar sistemas automatizados baseados em câmaras de visão que oferecem uma solução eficaz para o controlo de qualidade e processo. Com isto, os produtos que não cumpram com os requisitos de qualidade mínimos são eliminados do processo [2][28].

A aplicação de sistemas de visão artificial na produção industrial permite uma série de vantagens, tais como:

- **Aumento da produtividade** - Ao aplicar medidas de controlo de qualidade na monitorização de processos irá eliminar a parte defeituosa da mesma, de tal modo que as unidades defeituosas serão eliminadas antes de chegar ao final da linha de produção, o que leva a um aumento quantitativo da produção final.

- **Diminuição de perda de material** - Ao eliminar as unidades defeituosas ao longo do processo produtivo consegue-se reduzir de forma considerável o uso de materiais.
- **Redução dos custos operacionais** - Com o aumento da produtividade e a redução da perda de materiais, obtém-se um melhor aproveitamento dos recursos disponíveis, o que originará uma redução dos custos de produção.
- **Repetibilidade** - Para trabalhos repetitivos, os sistemas de visão são os mais adequados visto realizarem tarefas monótonas com mais eficácia. Ao se implementar um sistema totalmente automatizado aumenta o tempo de produção, pois a máquina não necessita de tempo para pensar como uma pessoa normal, o que leva a que a sua precisão e repetibilidade sejam muito superiores.
- **Precisão** - Garantem um maior grau de precisão dentro da tolerância aceite em todos os processos de fabrico. Mesmo quando os trabalhadores usam equipamentos específicos, como uma lupa, a visão artificial é mais precisa que a humana.
- **Tempo de inatividade reduzido** - Um sistema automatizado de visão é uma ferramenta eficaz para reduzir o tempo de inatividade no controlo de qualidade, visto ser um sistema totalmente automatizado, pode trabalhar 365 dias por ano sem precisar de pausas para descanso.
- **Melhor qualidade final do produto** - Com a utilização da visão artificial os produtos finais são de melhor qualidade, pois ao longo do processo de fabrico foram eliminadas as partes defeituosas e substituídas por novas, e assim consegue-se oferecer um produto final com melhor qualidade, satisfazer as necessidades do consumidor e conseqüentemente uma possível fidelização do mesmo [19][20][29].

3. CONSTITUIÇÃO DE UM SISTEMA DE VISÃO

Um sistema de visão (Figura 11) é constituído normalmente por câmaras, por um sistema ótico, filtros, lentes, por um sistema de iluminação e por um sistema de processamento. Em seguida explica-se mais detalhadamente cada um dos componentes separadamente.

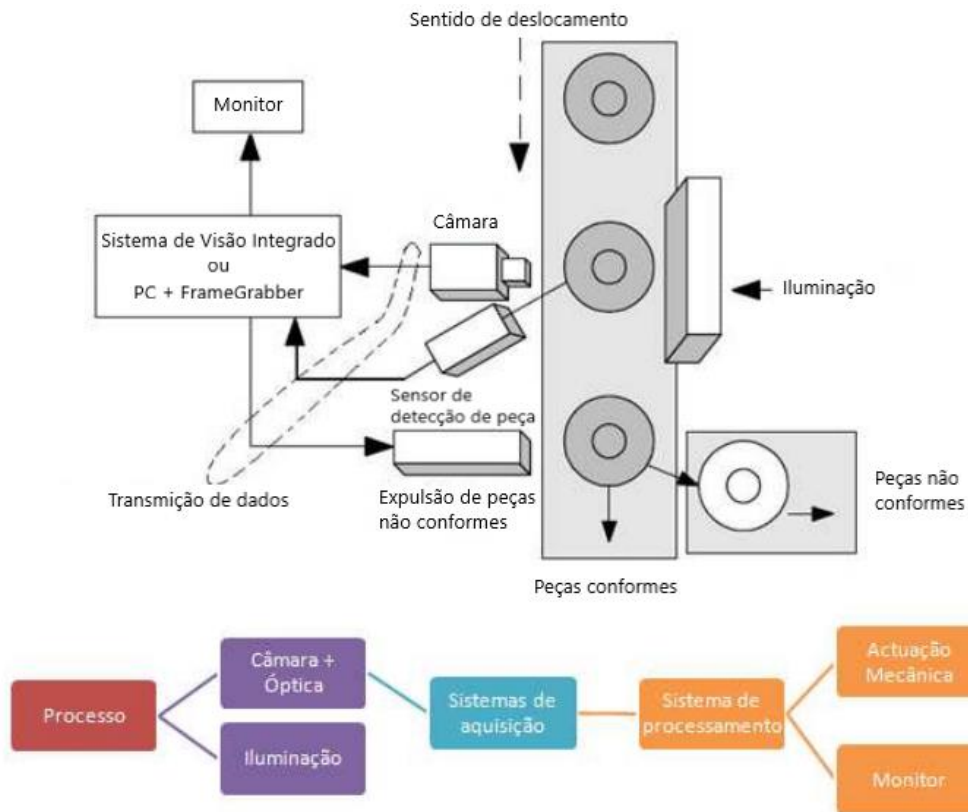


Figura 11 Sistema de visão artificial [16]

3.1. SISTEMA ÓTICO

O sistema ótico de uma câmara é o que permite transmitir a luz ao sensor de uma forma controlada e, assim, conseguir obter uma imagem focada dos objetos em análise. Incorpora um sistema de regulação de abertura, iris ou diafragma, que controla a quantidade de luz que chega ao sensor.

O sistema ótico numa câmara é constituído por uma objetiva onde depois se pode acoplar um filtro para se obter uma imagem melhor para análise. Os parâmetros principais são a abertura máxima e a distância focal, em que a abertura regula a intensidade da luz na imagem e determina a profundidade de campo, já a distância focal é a distância entre o centro ótico (sensor) e os seus pontos de foco.

3.1.1. PRINCÍPIOS BÁSICOS

Distância focal (f) - É a distância ótica entre o ponto onde os raios de luz convergem para formar uma imagem nítida de um objeto no sensor digital, e o sensor. A distância focal diz-nos o ângulo de visão e a ampliação, ou seja, quanto maior for a distância focal, mais estreito o ângulo de visão e maior a ampliação, e vice-versa (Figura 12).

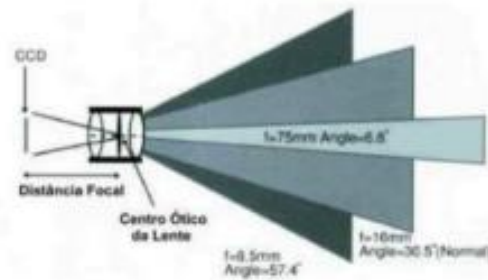


Figura 12 Distância Focal [30]

Distância de trabalho (WD) – É a distância entre o final da ótica e o objeto a observar (Figura 13).



Figura 13 Distância de trabalho [31]

Abertura relativa (F) – Corresponde à medida da máxima quantidade de luz que passa através da lente. Controla-se mediante uma íris ou diafragma que se encontra dentro da lente. A íris ajustável no interior da lente, normalmente, utiliza incrementos *standards* que são: 1.0, 1.4, 2.0, 2.8, 4, 5.6, 8, 11, 16, 22 (Figura 14). Cada incremento representa uma redução da luz para metade, o que vai influenciar a profundidade de campo, pois quanto mais a íris da lente estiver aberta, menor será a profundidade de campo e vice-versa [32].

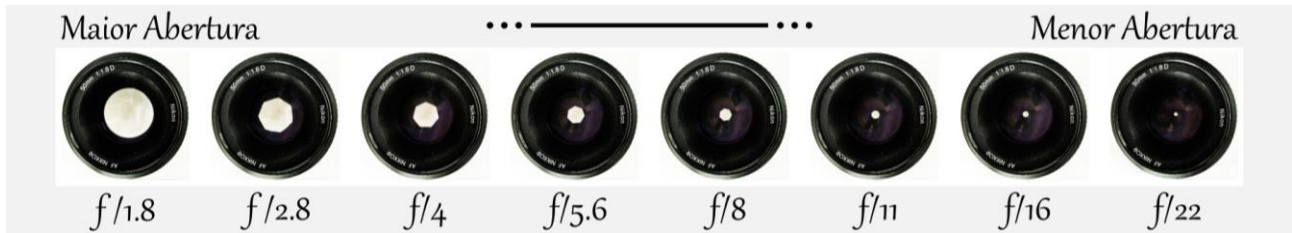


Figura 14 **Abertura da Íris** [33]

Profundidade de campo (DOF) - A profundidade de campo é a distância por diante e por detrás do ponto focado que aparece com nitidez na fotografia. Aproximadamente a distância nítida é o dobro por detrás do ponto focado que por diante, como podemos ver na Figura 15. Este valor é afetado pela:

- **Abertura Relativa** - Quanto maior a abertura, menor o DOF e vice-versa;
- **Distância focal** - Quanto maior a distância focal, menor o DOF e vice-versa;
- **Distancia de trabalho** - Quanto mais próximo do objeto você estiver, menor o DOF e vice-versa [34].

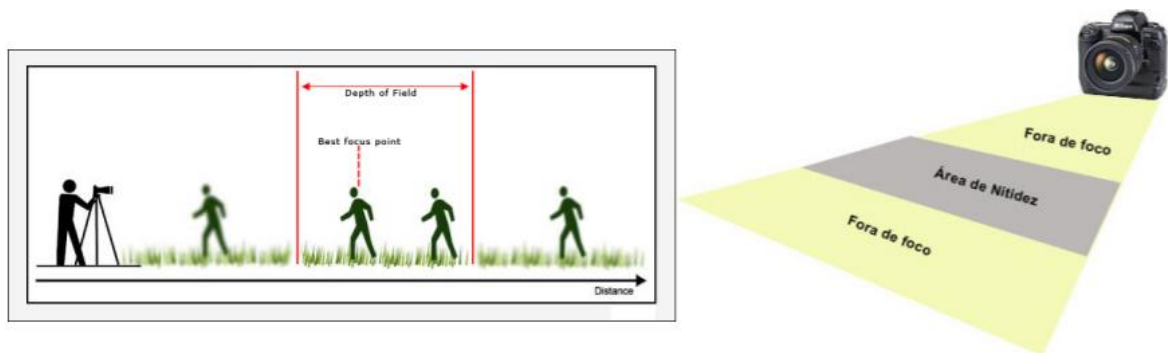


Figura 15 **Profundidade de campo** [34]

Aberrações óticas - Ocorre quando a luz de um ponto de um objeto não converge ou não diverge para um único ponto depois da transmissão através do sistema. As aberrações podem ser cromáticas ou monocromáticas, e estas dividem-se em esféricas, astigmatismo, coma e distorção, como se pode verificar na Figura 16 [35].

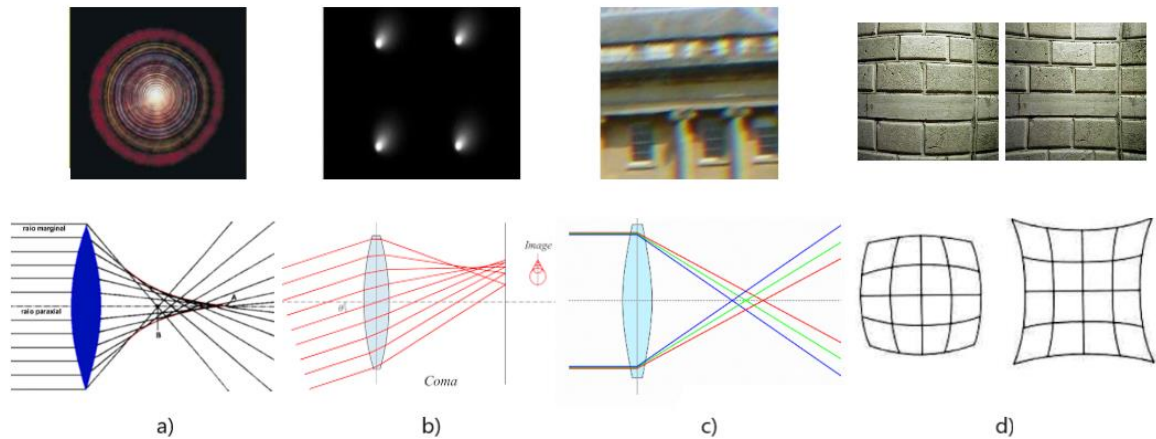


Figura 16 a) Aberração Esférica; b) Aberração Coma; c) Aberração Cromática; d) Distorção positiva (esquerda) e negativa (direita) [35]

3.1.2. OBJETIVAS

As objetivas permitem transmitir a luz ao sensor da câmara de uma forma controlada e, assim, obter uma imagem focada de um ou vários objetos. Incorporam um sistema de regulação da abertura, o diafragma em íris, que permite ajustar a quantidade de luz que se pretende deixar entrar para o sensor.

São constituídas por uma lente ou por uma montagem de lentes, sendo os dois parâmetros principais a abertura máxima e a distância focal. As lentes são responsáveis pela angulação do enquadramento e pela qualidade da imagem. Também podem ser caracterizadas pela sua distância focal e por isso podemos dividi-las em 6 grupos distintos:

Lentes normais – Também conhecida como média angular (25° a 60°), têm o ângulo de visão semelhante ao do olho humano, não aproximam nem ampliam.

Lentes grande angular – Oferecem um maior ângulo de visão (60° a 110°), fazendo com que a área fotografada seja maior que as lentes normais. São indicadas para fotografar locais pequenos com grande área, mas sem muito espaço de distanciamento.

Lentes teleobjetivas - Oferecem um pequeno ângulo de visão (1° a 5°), logo maior distância focal, fazendo com que seja possível fotografar a grandes distâncias, podendo apresentar em alguns casos um efeito de achatamento.

Lentes macro – Podem proporcionar grandes aberturas, tornando possível fotografar pequenos objetos a curtas distâncias.

Lentes olho de peixe – É um tipo de lente grande angular, mas só que com um campo de visão superior, podendo chegar aos 180° .

Lentes zoom – São as lentes mais versáteis, pois possuem a capacidade de variação de *zoom*, podendo variar entre a grande angular até à teleobjetiva [36][37].

Sendo as lentes normais, macro e olho de peixe as mais utilizadas no ramo das câmaras de visão artificial. A Figura17 mostra as diferentes características entre as lentes acima referidas.

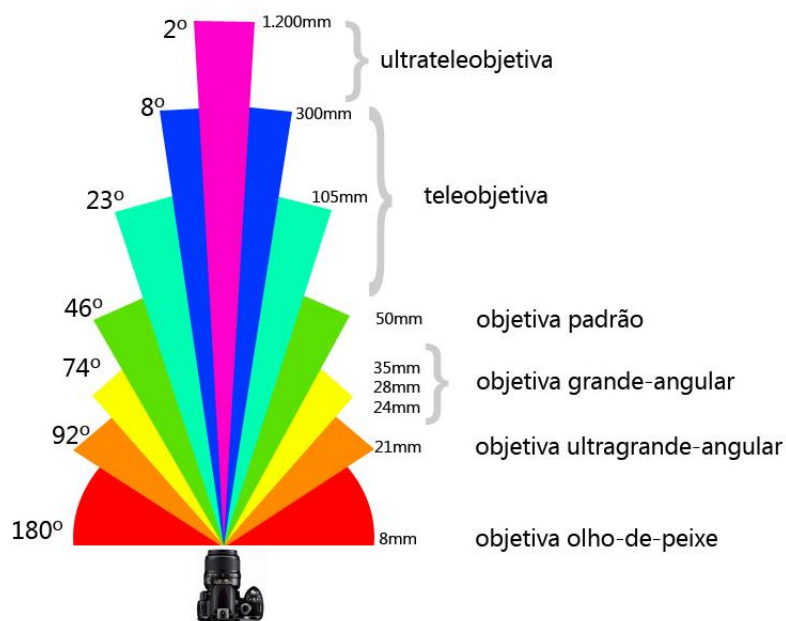


Figura 17 Tipos de lentes existentes [38]

Para escolher qual a objetiva a utilizar para uma dada aplicação, deve-se ter em consideração os seguintes parâmetros:

- Dimensão do sensor da câmara;

- Distância do objeto;
- Campo de visão (*Field of Vision* - representa a área visível que o sensor capta do objeto, ou seja, a região do objeto que preenche e sensibiliza a área do sensor) [16].

Uma vez conhecidos estes parâmetros, a equação (1) relaciona e permite calcular a objetiva a utilizar:

$$\text{Distância focal} = \frac{\text{Tamanho do sensor} \times \text{Distância de trabalho}}{\text{Campo de visão}} \text{ (nm)} \quad (1)$$

3.1.3. FILTROS

Em várias situações onde é aplicada a visão artificial é necessário que chegue à câmara um certo comprimento de onda proveniente do objeto visualizado, para que tal seja possível, é necessário utilizar filtros que vão ser encarregues de filtrar determinados comprimentos de onda [39][40].

Dentro dos filtros óticos, os mais utilizados são os seguintes:

- **Filtro passa-alto** - Transmite a luz acima de certos comprimentos de onda;
- **Filtro passa-baixo** - Transmite a luz abaixo de certos comprimentos de onda;
- **Filtro passa-faixa** - Transmite a luz dentro de uma faixa estreita de comprimentos de onda (Figura 18);

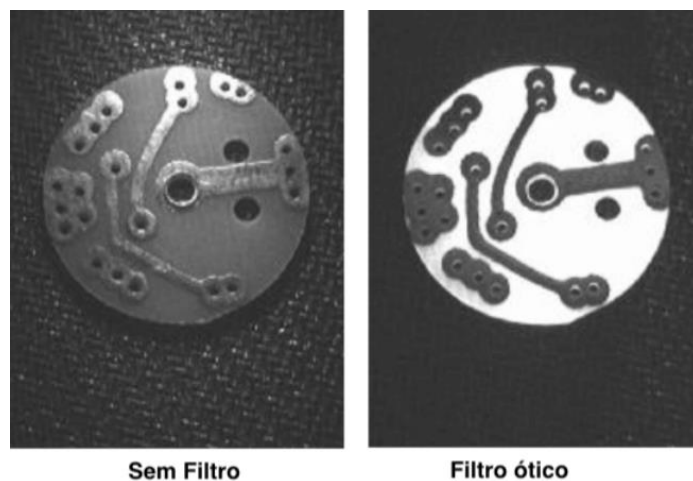


Figura 18 **Filtro passa-faixa (luz verde)** [41]

- **Filtros de corte** - Bloqueiam o passo de certos comprimentos de onda;
- **Filtro polarizador** - permite, assim, obter um efeito muito útil: a eliminação, ou redução, dos reflexos em superfícies não metálicas (Figura 19).

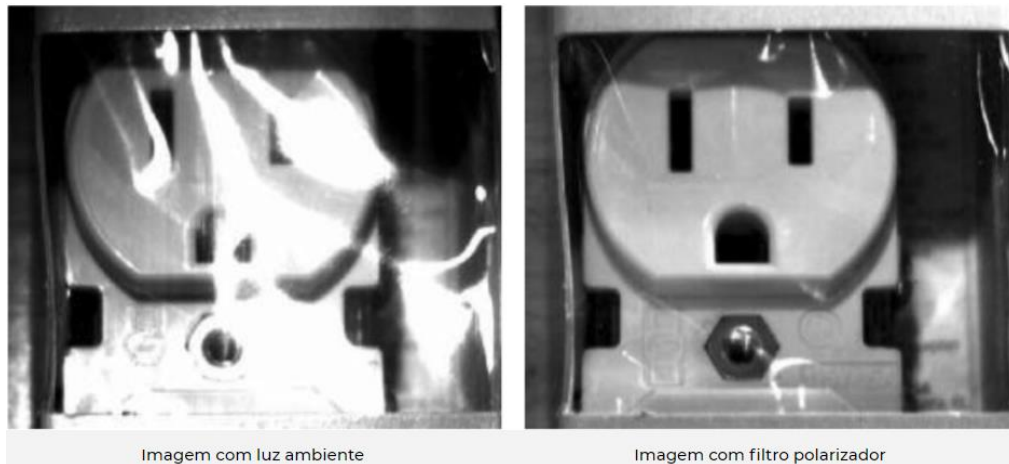


Figura 19 **Filtro Polarizador** [41]

- **Filtro Ultravioleta (UV)** - Tem como função de absorver a luz solar em excesso. Esse filtro fornece mais naturalidade e suavidade às imagens e diminui efeitos de reflexo (Figura 20).

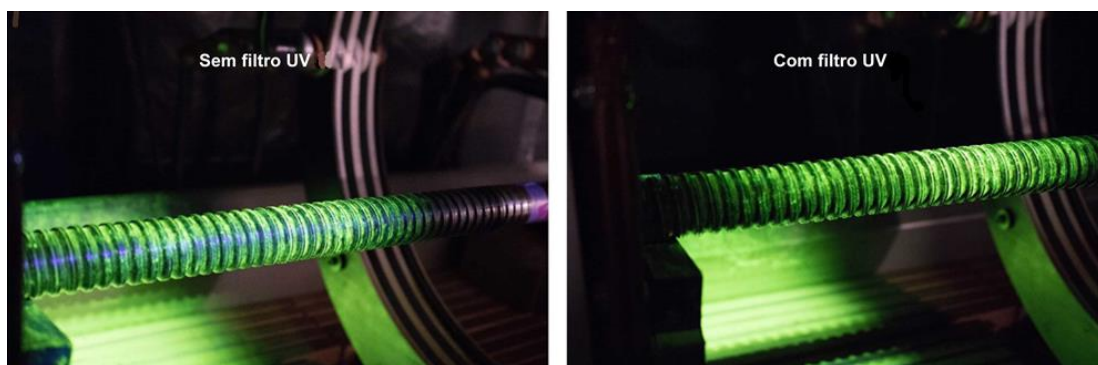


Figura 20 **Filtro Ultravioleta** [42]

Os filtros mais utilizados nos sistemas de visão artificial são os filtros de passo ou de corte de infravermelhos.

3.2. SISTEMA DE ILUMINAÇÃO

A iluminação é uma das variáveis mais importantes de um sistema de visão, e influencia de forma indubitável a qualidade da imagem obtida. Uma correta escolha do tipo iluminação permite realçar características, e desta forma diminuir o trabalho necessário ao nível de algoritmos de visão a utilizar (Figura 21).

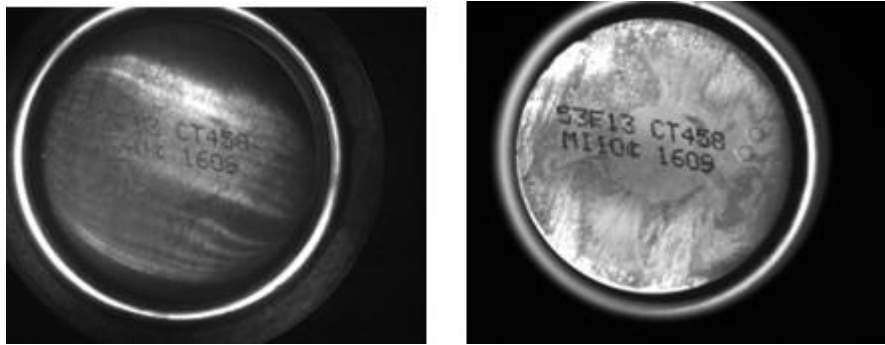


Figura 21 Resultados da utilização de diferentes técnicas de iluminação [43]

3.2.1. TIPOS DE FONTES

Atualmente, existem no mercado sistemas de iluminação baseados em iluminação por *LED* (*Light Emitting Diode*), lâmpadas halogéneas, laser, fibra ótica e fluorescente [44]. A escolha entre as diversas possibilidades assenta na análise das características pretendidas para a aplicação. Depende da intensidade de luz necessária, se deve ser direcional ou difusa, luz branca ou monocromática e, ainda, se queremos iluminação constante ou apenas em instantes determinados, como podemos ver na Figura 21 [8][16].

Seguidamente, apresentam-se as principais características dos vários tipos de fonte de luz.

- **LED** – Utiliza-se em aplicações que necessitam de grandes intensidades de luz e de forma contínua, podendo ser branca ou monocromática e permite uma iluminação direcional ou difusa. Por outro lado, também é a mais económica e mais duradoura;
- **Halogénio** – Produz um tipo de iluminação mais semelhante à luz natural, baixo custo, tempo de vida média baixo, não são direcionais e necessitam de alimentação regulada para as aplicações de visão industrial;

- **Laser** – Utiliza-se para iluminar um objeto, com um padrão de luz específico, e desta forma observar como se altera o padrão em relação ao objeto, podendo assim obter informações sobre a terceira dimensão do objeto iluminado;
- **Fibra ótica** – A que pode proporcionar a luz mais intensa, pois conduz a luz de uma lâmpada de halogéneo, que se encontra numa fonte de iluminação, através de um feixe de fibras óticas que termina num adaptador específico.
- **Fluorescentes** - Tempo de vida média razoável, eficiência energética, não são direcionais, nas aplicações de visão precisa-se de frequências superiores a 25kHz o que faz aumentar o custo destas lâmpadas, [8][45].

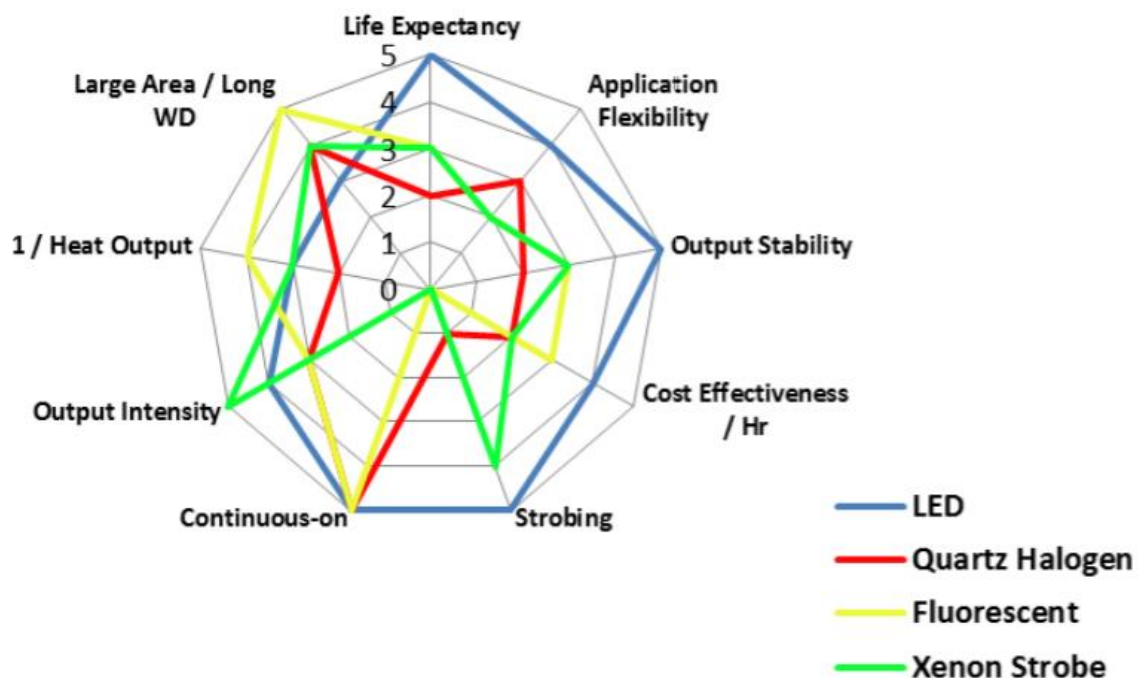


Figura 22 Comparação dos diferentes tipos de iluminação [46]

3.2.2. TÉCNICAS DE ILUMINAÇÃO

Um dos parâmetros fundamentais da iluminação é a técnica de iluminação, que quando selecionada corretamente permite simplificar a extração de propriedades. Entre as técnicas principais e mais utilizadas [47][48], destacam-se as seguintes:

- **Iluminação Campo Escuro (*Dark Field*)** - Projeta a luz na perpendicular ao objeto, destacando características da superfície, tais como arranhões, bordas ou marcas (Figura 23);

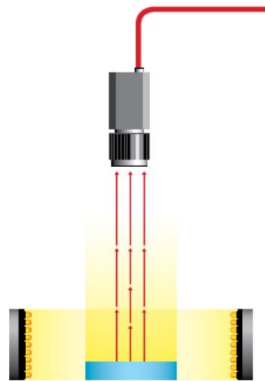


Figura 23 Iluminação Campo Escuro (*Dark Field*) [49]

- **Iluminação Lateral ou em Barra** - Projeta uma faixa de luz no alvo, para obter uma iluminação constante ao longo de uma área. Utilizada para adicionar contraste a superfícies foscas ou definir bordas (Figura 24);

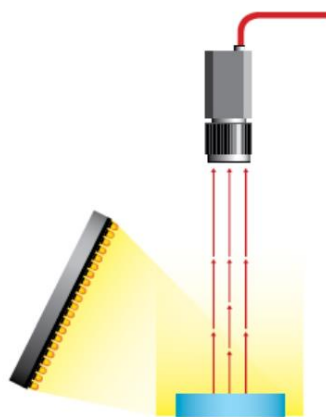


Figura 24 Iluminação Lateral ou em Barra [49]

- **Iluminação de Fundo (*Backlight*)** - Projeta a luz na parte traseira do objeto, destacando a forma do mesmo. Usada para detetar orifícios, medição e contorno (Figura 25);

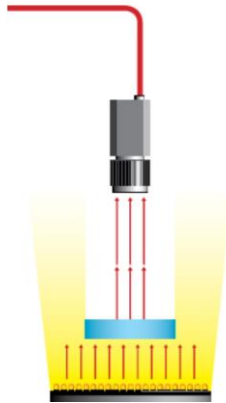


Figura 25 Iluminação de Fundo (*Backlight*) [49]

- **Iluminação no Eixo (coaxial) difusa** - Projeta a luz na perpendicular ao objeto, mas para isso utiliza um espelho para transmitir a luz num ângulo de 90 graus. Utilizada para detetar falhas em superfícies brilhantes e medições (Figura 26).

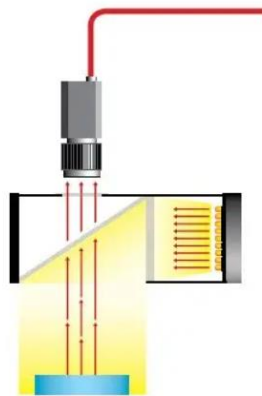


Figura 26 Iluminação no Eixo (coaxial) difusa [49]

- **Iluminação de Cúpula ou Difusa** - Projeta a luz a partir de vários ângulos, distribuindo a iluminação de forma uniforme por todo o objeto, eliminando sombras e brilhos. Utilizada para inspecionar superfícies brilhantes, curvas ou irregularidades (Figura 27).

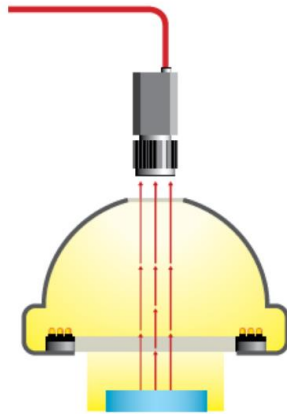


Figura 27 **Iluminação de Cúpula ou Difusa** [49]

- **Iluminação de Anel** - Projeta a luz através de um anel brilhante, fornecendo uma iluminação sem sombras e com bons contrastes. Utilizada para a inspeção de peças altamente refletivas ou brilhantes (Figura 28).

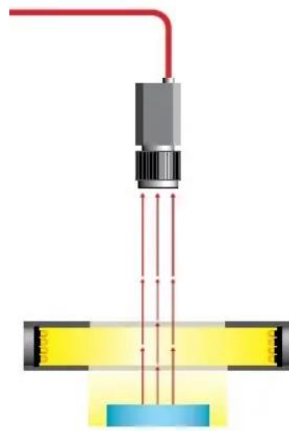


Figura 28 **Iluminação de Anel** [49]

- **Iluminação Estruturada** - Projeta pontos, linhas ou grelhas com um determinado ângulo na superfície de trabalho com o objetivo de obter uma percepção tridimensional do objeto (Figura 29). Pode ser obtida utilizando diferentes fontes de luz, sendo a mais utilizada a proveniente de um feixe de laser.

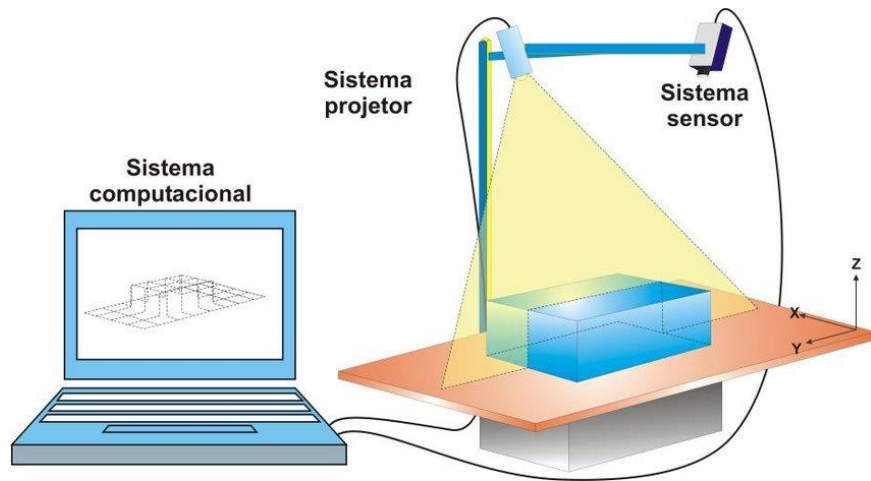


Figura 29 Iluminação Estruturada [50]

Na Tabela 2 é apresentado um resumo dos diferentes tipos de iluminação e que, possivelmente, melhores resultados terão em alguns tipos de aplicações [51].

Tabela 2 Tipos de iluminação sugerida para várias aplicações [51]

<i>Aplicação</i>	<i>Tipos de objeto a inspeccionar</i>	<i>Iluminação sugerida</i>
<i>Redução de especulações</i>	Objetos brilhantes	Iluminação difusa frontal ou lateral
<i>Iluminação uniforme</i>	Qualquer tipo de objeto	Iluminação difusa frontal, ou axial
<i>Realce de defeitos de superfície</i>	Objetos planos (2D)	Iluminação direcional simples, Iluminação estruturada
<i>Realce de textura de objetos com sombra</i>	Qualquer tipo de objeto	Iluminação direcional ou Iluminação estruturada
<i>Redução de sombras</i>	Objetos com irregularidades 3D	Iluminação difusa frontal, axial, ou em anel.
<i>Realce de defeitos de objetos</i>	Objetos transparentes	<i>Backlight</i>
<i>Contornos do objeto</i>	Qualquer tipo de objeto	<i>Backlight</i>
<i>Perfil de objetos 3D</i>	Objetos com irregularidades	Luz estruturada

3.3. CÂMARAS DE VISÃO ARTIFICIAL

É o componente que permite capturar uma imagem projetada no sensor e em seguida enviá-la para um sistema eletrônico externo, um computador, por exemplo. Uma câmara digitaliza diretamente a luz original refletida pelo objeto, decompondo

imediatamente esse padrão de luz numa numa matriz de pontos (pixéis) alocados em memória [16][52].

A aquisição de imagem, materializa-se, então, por meios eletrônicos, podendo se utilizar um sensor CCD (*charge coupled device*) ou um sensor CMOS (*complementary metal oxide semiconductor*). Sendo que a conversão mais comum é a obtida pelas células CCD na qual cada ponto da célula representa um ponto de imagem projetada sobre ela [53].

3.3.1. TIPOS DE CÂMARAS DE VISÃO ARTIFICIAL

- **Câmaras matriciais** – Formada por um sensor que cobre a área formada por uma matriz de pixéis. Uma câmara matricial produz a imagem de uma área, normalmente, com uma proporção de 4:3.
- **Câmaras lineares** - A construção da imagem é feita linha-a-linha utilizando um sensor linear, ou seja, ou a câmara desloca-se em relação ao objeto a captar, ou é o objeto que se desloca em relação à câmara. Usadas na inspeção de materiais em processos contínuos, papel, tecidos, etc.
- **Câmaras IP** - Contêm uma unidade de processo que permite transferir imagens comprimidas de forma autónoma mediante protocolo IP-Ethernet. As câmaras IP-Ethernet permitem capturar vídeo/imagem em tempo real, com possibilidade de *trigger* externo para ativar a gravação. Junto ao envio de imagens permite uma série de funções adicionais que facilitam o uso dessas câmaras em aplicações de supervisão e segurança.
- **Câmaras inteligentes** – Estas câmaras incorporam um processador, uma memória e sistema de comunicações com o exterior (Porto Série, I/O, *Ethernet*), formando assim um sistema completo de visão artificial compacto.
- **Câmaras térmicas/infravermelhas** - São câmaras capazes de determinar a temperatura dos corpos a partir da sua radiação infravermelha [53].

3.3.2. TIPOS DE SENSORES DE IMAGEM

Um sensor de imagem capta a luminosidade das imagens que são projetadas nele continuamente e dá início ao processo de captura de uma imagem ou de uma sequência de imagens consecutivamente. Dentro dos sensores existem quatro tipos, os sensores CCD, os CMOS (*Complementary Metal-Oxide Semiconductor*), os Microbolómetros e os FPA (*Focal Plane Array*). Em seguida apresentamos mais detalhadamente cada um deles.

- **Sensor CCD** – O espectro de inspeção encontra-se na zona da luz visível, têm uma grande sensibilidade de ofuscamento e apresenta ruído variável através da temperatura. Podem ser usados em *scanners*, monitorização de linhas de acondicionamento e de embalagens.
- **Sensor CMOS** - Superior aos sensores CCD, visto que, têm resistência ao ofuscamento e conseguem inspecionar na zona dos infravermelhos. São ideais para operar em ambientes com fraca luminosidade, têm uma velocidade de leitura superior à CCD e um baixo consumo energético.
- **Sensor Microbolómetros** - São sensores de infravermelhos utilizados exclusivamente em câmaras termográficas. Têm grande rapidez de processamento.
- **Sensor FPA** – Apresentam grande sensibilidade e uma grande qualidade de imagem. Podem ser usados para orientação de armas teleguiadas, câmaras de visão e para captura de imagens médicas e espaciais [53][54].

3.3.3. TAMANHO DOS SENSORES DE IMAGEM

Todas as câmaras digitais possuem sensores de imagem de diferentes tamanhos, como podemos verificar na Figura 30.

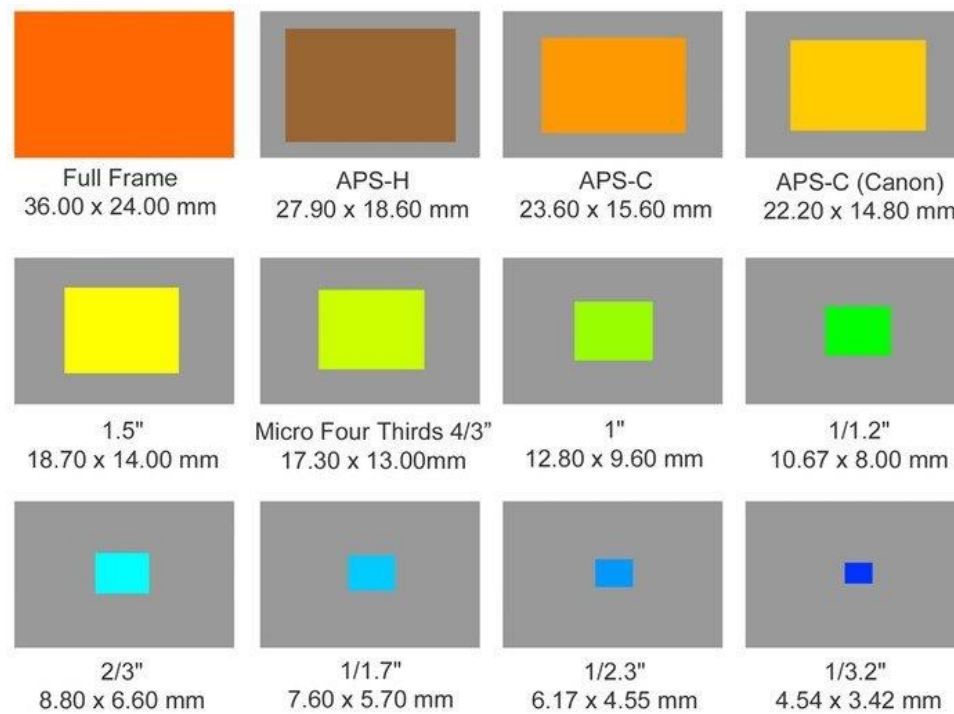


Figura 30 Tamanho dos diferentes tipos de sensores [55]

O tamanho do sensor de imagem determina a qualidade da imagem e o ângulo de visão que se pode captar quando se usa lentes de diferentes tamanhos. A ocorrência de diferença de tamanhos dos sensores leva-nos ao fator de corte, que consiste na área que um sensor mais pequeno perde comparando a mesma imagem, mas capturada com um sensor maior. Quanto maior for o sensor, mais quantidade de pixéis tem e melhor será a qualidade da imagem.

3.3.4. UNIDADE DE PROCESSAMENTO

A unidade de processamento contempla vários módulos, um computador “tradicional”, entradas e saídas para permitir a interação com o processo, *framegrabber* para aquisição de imagens através das câmaras (caso estas não sejam câmaras *Firewire*, ou *Gigabit Ethernet*, ou o sistema não suporte estes formatos de comunicação) e um software de visão de forma a processar os dados recebidos da câmara [16].

O processamento é realizado por *software* e consiste em várias etapas. Primeiro, uma imagem é adquirida a partir do sensor. Em alguns casos, o pré-processamento pode ser necessário para otimizar a imagem e garantir que todas as características necessárias se destacam. Em seguida, o *software* localiza as características específicas, executa as

medidas e as compara com as especificações. Finalmente, uma decisão é tomada e os resultados são comunicados [56]. De notar que, caso se utilizem câmaras inteligentes, a unidade de processamento torna-se desnecessária pois todo o trabalho é realizado pelas próprias câmaras, tendo inclusive entradas e saídas para permitir a interação com o processo.

Muitas das câmaras utilizadas nos sistemas de visão artificial e análise de imagem conectam-se diretamente ao computador através de diferentes portas: *Firewire*, *USB* e *Gigabit Ethernet*. Entretanto algumas câmaras ainda requerem uma conexão com o computador através de uma placa de captura. Estas placas de captura que frequentemente incluem componentes imprescindíveis para as aplicações de visão são denominados *Frame Grabbers* (Figura 31).

Estas placas têm a função de converter, montar e armazenar uma imagem vinda de uma câmara analógica ou digital na memória RAM. A imagem é representada por uma matriz numérica, correspondente ao número de pixel da célula CCD da câmara. Cada pixel é armazenado num *byte* da memória e tem o comprimento de 8 *bits*.

Nas câmaras digitais, não é dispensável o uso do *Frame Grabber*, porém o modelo da placa a ser utilizado é diferente do modelo analógico, pelo fato da câmara analógica não possuir um conversor A/D. Na realidade, a própria câmara digital faz a conversão do valor de pixel analógico em um valor digital. Este processo permite a construção de câmaras de alta velocidade [57].

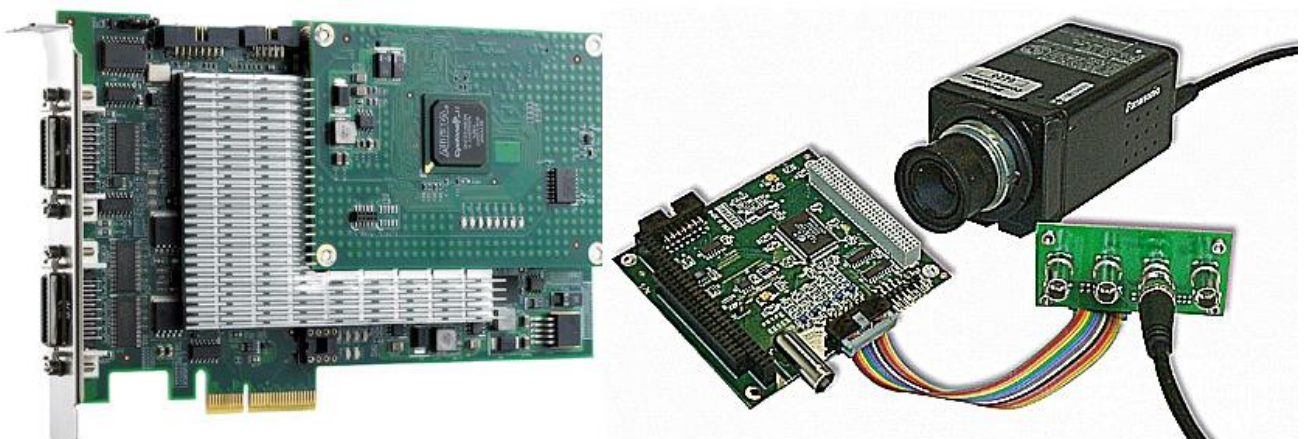


Figura 31 *Frame Grabber* [58]

3.4. INTERFACES DE COMUNICAÇÃO

É através das interfaces que se torna possível a transmissão de informações e de dados entre a câmara e o computador. Na Tabela 3 estão algumas das interfaces mais usadas no campo da visão, bem como algumas das características principais.

Tabela 3 Características das interfaces de comunicação [16]

	<i>USB 2.0</i>	<i>USB 3.0</i>	<i>CamaraLink</i>	<i>GigaE</i>	<i>PoE</i>	<i>Fire Wire</i>
<i>Topologia</i>						
<i>Tipo de Standard</i>	Consumo	Consumo	Comercial	Comercial		Consumo
<i>Tipo de conexão</i>	<i>Master/ Slave</i>	<i>Master/ Slave</i>	<i>Bus Partilhado</i>	<i>Peer- to-peer</i>	<i>Master/ Slave</i>	<i>Peer- to-peer</i>
<i>Taxa de transmissão</i>	480 Mb/s	4800 Mb/s	6000 Mb/s	1000 Mb/s	1000 Mb/s	400 Mb/s
<i>Distância</i>	5 m	5 m	10 m	100 m	100 m	10 m
<i>Distância c/ repetidores</i>	30 m	-	40 m	Ilimitada	-	72 m
<i>Distância com fibra ótica</i>	-	-	1000 m	Ilimitada	-	200 m
<i>Número de dispositivos</i>	127	-	4	Ilimitado	-	63
<i>Interface PC</i>	Placa Base Placa PCI	Placa Base Placa PCI	<i>Frame Grabber</i>	Placa Base Placa PCI	Placa Base Placa PCI	Placa Base Placa PCI

3.5. FORNECEDORES/SOFTWARE

Existem vários fornecedores de equipamento de visão e de software, sendo os mais comuns os seguintes:

- Banner Engineering;
- Cognex;
- Omron Corporation;
- National Instruments Corporation;
- NeuroCheck GmbH;
- Dalsa Corporation.

4. PROJETO DESENVOLVIDO

Este capítulo apresenta o projeto desenvolvido, desde as opções escolhidas em termos de *hardware* ao *software* e aos testes efetuados.

4.1. HARDWARE

A Figura 32 ilustra os equipamentos a que a câmara de visão artificial se encontra conectada, nomeadamente o PC para se proceder à programação da mesma, o compressor que vai ser utilizado para expulsar os parafusos mal orientados e por fim o iluminador para uma melhor captura das fotografias em tempo real.

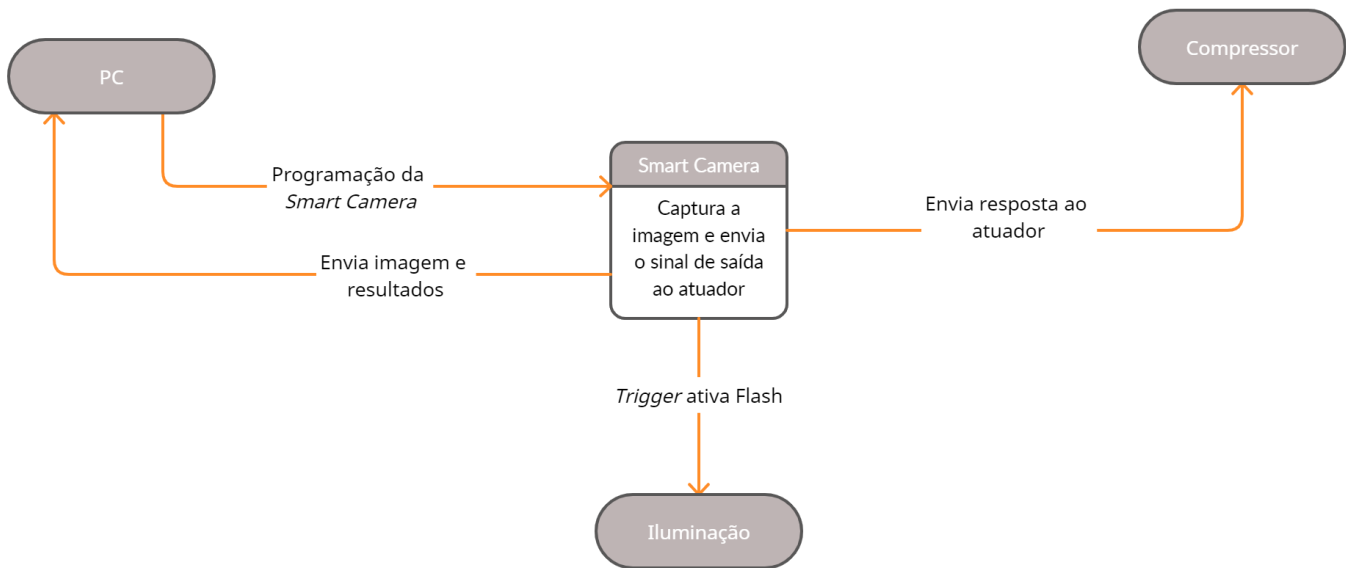


Figura 32 Fluxograma sobre as ligações de *hardware* utilizado

Os equipamentos usados neste projeto foram, um iluminador da Banner, próprio para acoplar a câmara, e de uma *Smart Camera* de visão artificial também da Banner, que posteriormente estará ligada a um compressor (Figura 33).



Figura 33 Equipamentos usados no projeto

4.1.1. BANNER PRESENCEPLUSP4 OMNI

A *Smart Camera* utilizada neste projeto foi a PresencePlusP4 OMNI da Banner (Figura 34), este modelo possui ferramentas poderosas de análise de imagem, tais como:

- Localização de peças;
- Verificação e medição;
- Reconhecimento de Caractere Ótico (OCR);
- Leitura de códigos de barras e texto.

Usa um sensor do tipo 1/3-inch CCD, tem uma resolução de 640 x 480 pixéis, 32 MB de memória interna que proporciona um armazenamento de até 85 ficheiros.



Figura 34 *Smart Camera* Banner PresencePlusP4 OMNI [10]

A Figura 35 apresenta os conectores e indicadores da *Smart Camera* utilizada, e as suas especificações estão presentes no Anexo B.

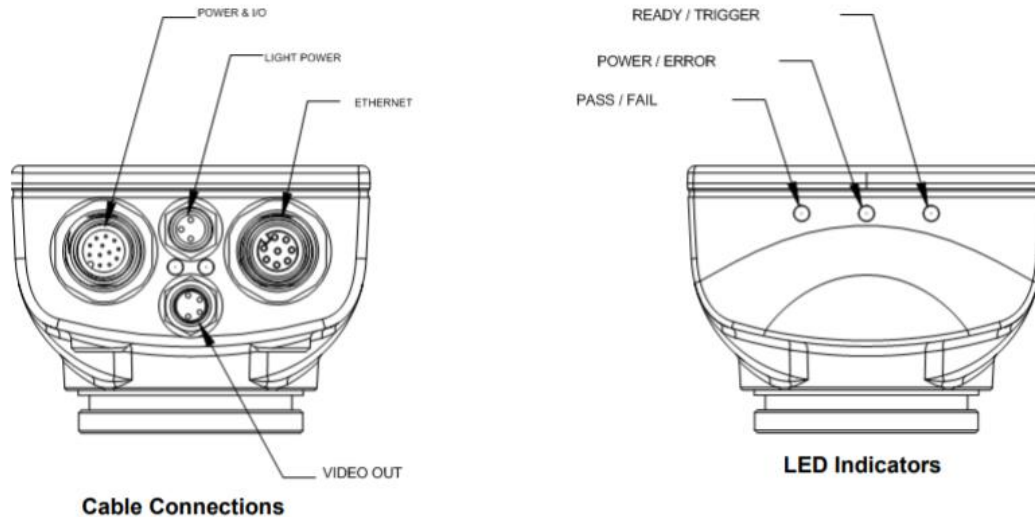


Figura 35 Conectores e indicadores da *Smart Camera* [59]

Power & I/O – É onde se conecta o cabo *BreakoutI*, M12 de 12 pinos, que fornece alimentação à câmara, entrada do acionador de aquisição (*trigger*), as 4 saídas I/O e a comunicação RS-232 (Figura 36).

12-Wire Cable Pin Assignments			
Pin #	Wire Color	Description	Direction
1	Yellow	RS-232 TX	Output
2	Gray	Remote Teach	Input
3	Orange	Product Change	Input
4	Pink	External Trigger	Input
5	Black	Discrete I/O #1	In/Out
6	Red	Discrete I/O #2	In/Out
7	White	Discrete I/O #3	In/Out
8	Light Blue	Discrete I/O #4	In/Out
9	Violet	RS-232 RX	Input
10	Green	RS-232 Signal Ground	Output
11	Blue	Common (Signal Ground)	Input
12	Brown	10-30V dc	Input

Figura 36 *Pinout da Smart Camera*

Light Power – É onde se conecta o cabo que interliga a câmara ao sistema de iluminação e lhe fornece alimentação.

Ethernet – É onde se conecta a câmara ao PC, por exemplo. Esta porta é usada para se conectar a câmara à rede, este conector fornece a ligação *Ethernet* aos dispositivos de rede externos.

Video Out – É onde se conecta a câmara a um monitor e é usado um cabo específico, BNC de 4 pinos M8.

Este modelo de câmara suporta dois tipos de ligação diferente, suporta a ligação NPN e a ligação PNP, e acabou-se por escolher a ligação PNP. Um sensor com saída PNP é comumente chamado de fonte ou “*sourcing*” pois quando o mesmo deteta um objeto, ele fornecerá para a carga o polo positivo (+12V, +24V, etc.).

A ligação entre a câmara e o PC é mostrada na Figura 37. Esta ligação é feita usando um cabo de Ethernet específico, o cabo industrial *Ethernet* M12 de 8 pinos (Anexo B). Este cabo permite a conexão a um único dispositivo ou pode fornecer conexões a vários dispositivos por meio de um *router* ou *network switch*.

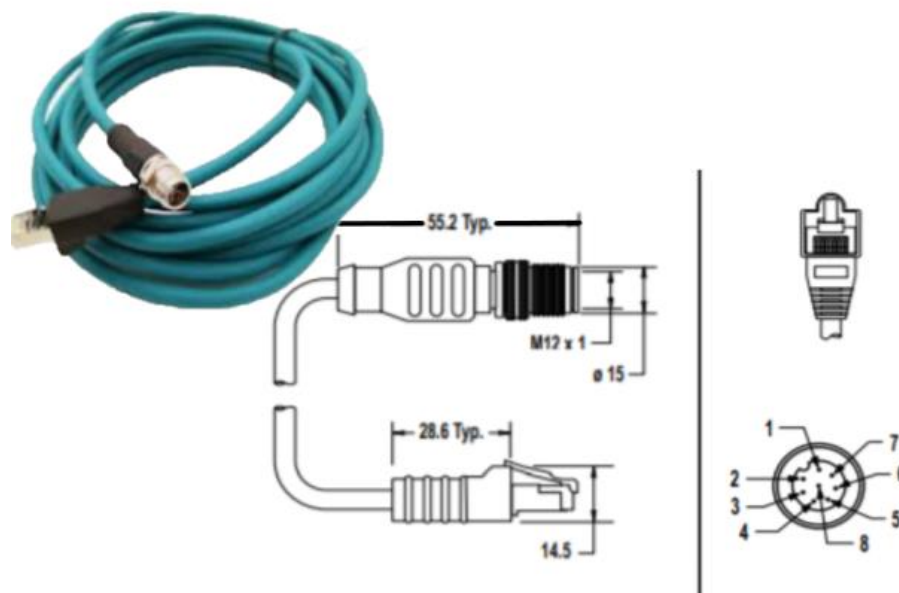


Figura 37 Cabo RJ45 *Ethernet* STP-MAQDC-806, para sistemas de visão [59][60]

Posteriormente teve-se de acoplar à *Smart Camera* uma objetiva para que fosse possível capturar as imagens com qualidade. Para tal, usou-se uma objetiva, da marca **Computar** (Figura 38), com uma distância focal de 8 mm e com uma abertura de f1.4.



Figura 38 **Objetiva *Computar* utilizada [61]**

4.1.2. ILUMINADOR PRESENCEPLUS

O iluminador utilizado foi o ilustrado na Figura 39, devido ao fato de se poder acoplar diretamente à *Smart Camera* e assim diminuir o espaço ocupado pelo mesmo no local onde foi implementada a câmara.



Figura 39 **Iluminador *PresencePlus* utilizado [62]**

Dado o exposto, é possível perceber como foi feita a interligação entre a *Smart Camera*, o PC e o atuador. De seguida passa-se então a apresentar o projeto desenvolvido no *software* utilizado.

4.2. SOFTWARE

4.2.1. OBJETIVO DO SOFTWARE

O *software* desenvolvido, tem como objetivo a captura de fotografias de parafusos em tempo real, e posterior análise das mesmas. Nessa análise, o *software* irá analisar os parafusos, um a um, através das ferramentas que possui e retornar se estão com a orientação correta ou não, através da localização das cabeças dos parafusos (Figura 40) e posteriormente expulsar os que estiverem mal orientados para o alimentador.



Figura 40 Objetivo do projeto – Orientação Correta

4.2.2. GERAL

O desenvolvimento do projeto exigiu o estudo inicial do software a ser utilizado no projeto e, principalmente, o estudo e a aprendizagem do *PresencePlus Vision Sensor*, utilizado para a programação da *Smart Camera*.

Em seguida explica-se grande parte da programação desenvolvida (Anexo C) para a *Smart Camera* de visão usada.

Para que a explicação se torne mais clara, foram elaborados fluxogramas das várias funcionalidades do sistema. Apresentam-se também todas as funções usadas para melhor se entender o procedimento para a criação do programa.

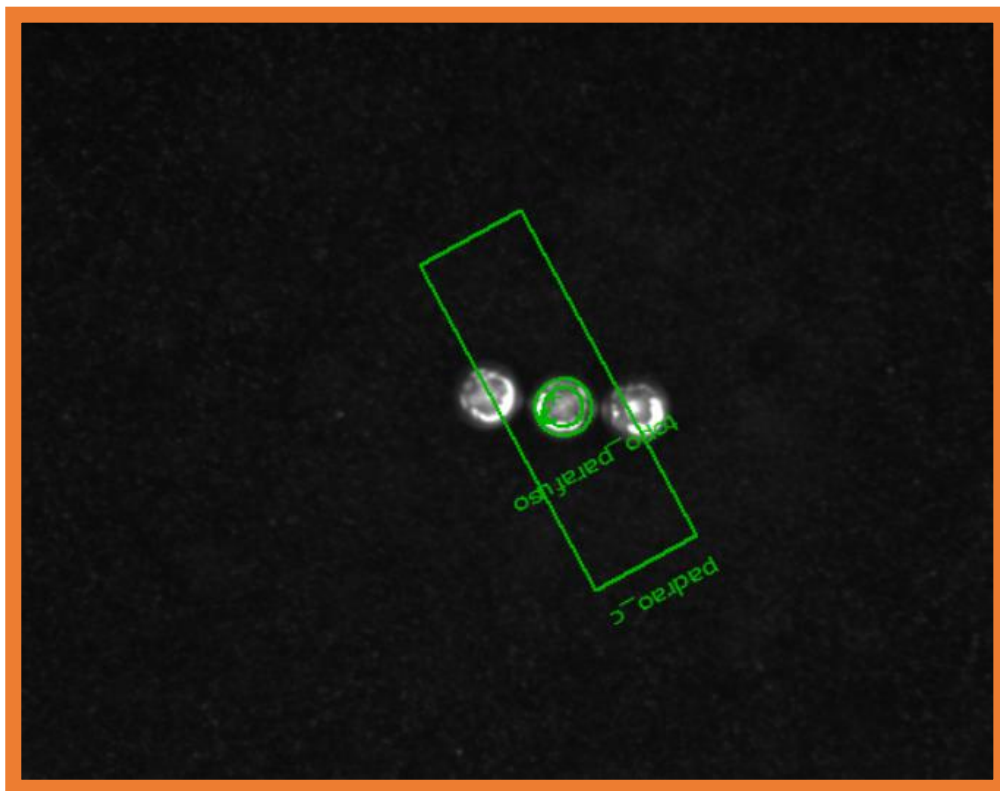


Figura 41 Ecrã principal do utilizador

O ecrã principal do utilizador é o ilustrado na Figura 41, este é o ecrã que encontra quando liga o sistema. Neste ecrã o utilizador tem permissão para ver o histórico da análise às fotografias capturadas e assim verificar se a produção está a correr conforme planeado.

Para proceder à programação da *Smart Camera*, o utilizador acede ao ecrã da Figura 42, onde tem a seu dispor um grande leque de ferramentas de análise. No entanto, a tarefa inicial consiste em configurar a câmara e, este tema será abordado seguidamente.

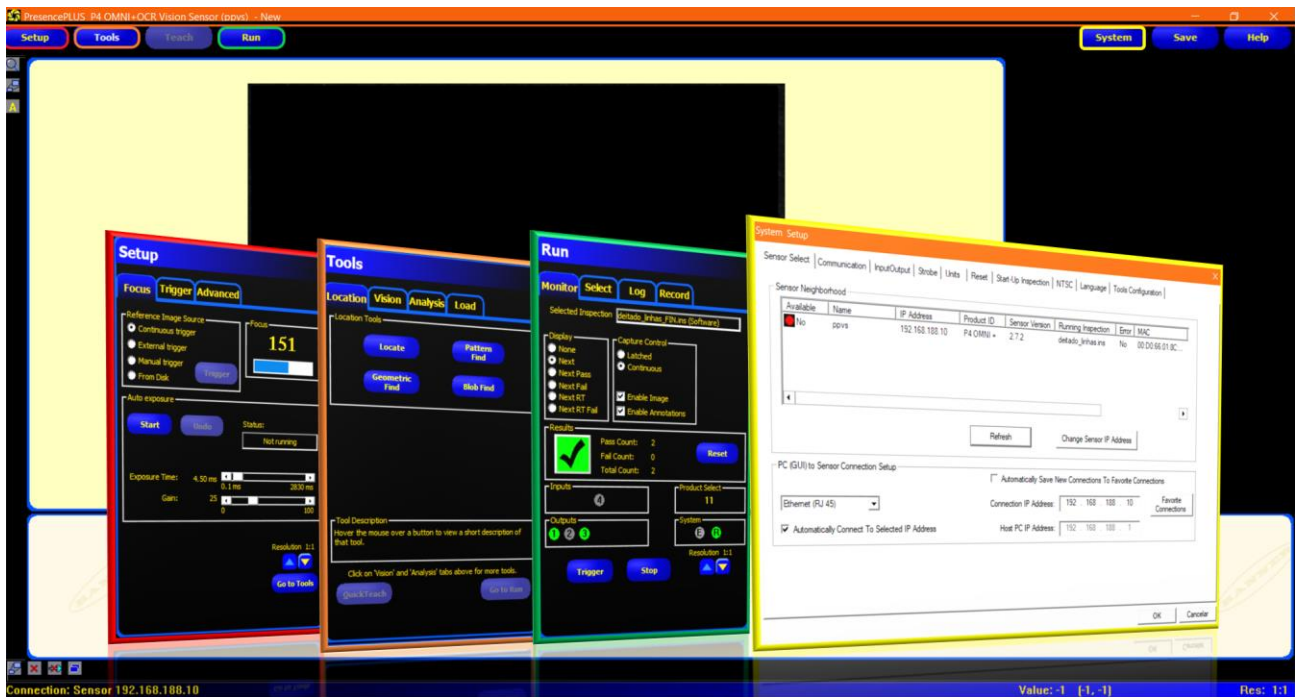


Figura 42 Ecrã de programação

O *PresencePlus Vision Sensor* é constituído por vários submenus:

- O retângulo vermelho, Submenu **Setup**, é onde o utilizador configura a imagem a analisar (tempo de exposição e ganho) e o *Trigger*;
- O retângulo laranja, Submenu **Tools**, é onde se encontram as ferramentas de visão que o utilizador poderá utilizar e onde se encontra a lista de todos os programas gravados na câmara;
- O retângulo verde, Submenu **Run**, é neste submenu que se visualiza os *outputs* do programa que está a ser executado (número de peças OK, NOK, saídas I/O e o programa escolhido),
- O retângulo amarelo, Submenu **System Setup**, é onde o utilizador faz as configurações da câmara, entre elas, o tipo de comunicação a utilizar (*Ethernet*

Socket, Serial, Industrial Ethernet), configurar as saídas I/O, unidades de medida e o Protocolo RCC (*Remote Command Channel*) que pode ser ASCII ou Kawasaki.

4.2.3. INICIALIZAÇÃO DA SMART CAMERA

A fim de estabelecer a comunicação entre a *Smart Camera* e o computador (Anexo A) foi preciso proceder à alteração do endereço IP do PC para o escolhido, que neste caso foi o **192.168.188.1** e em seguida o da máscara de sub-rede para **255.255.255.0**, como ilustrado na Figura 43.

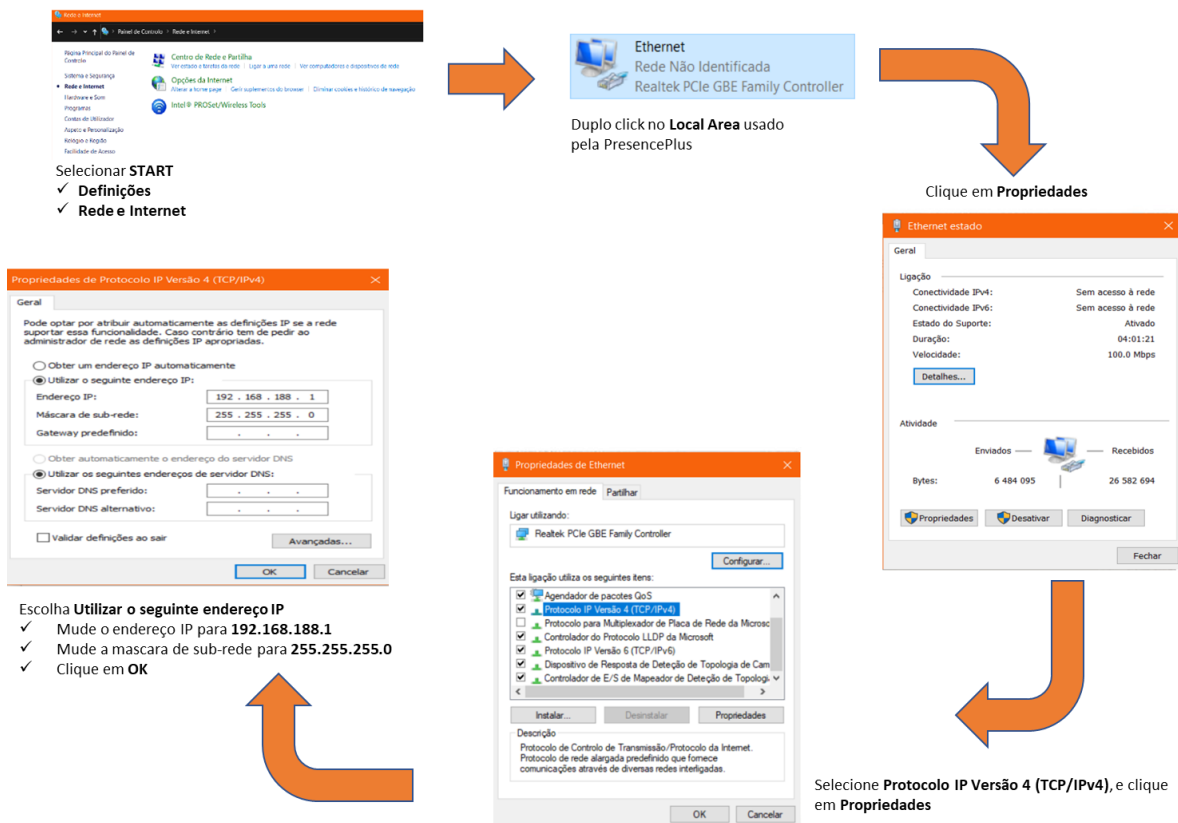


Figura 43 Configurações do endereço IP do PC

Em seguida, teve-se de definir um endereço IP da mesma família que o anterior para a *Smart Camera*. Esta configuração é feita no próprio software da *Smart Camera* (Figura 44), o *PresencePlus Vison*, e o IP escolhido para o efeito foi **192.168.188.10**.

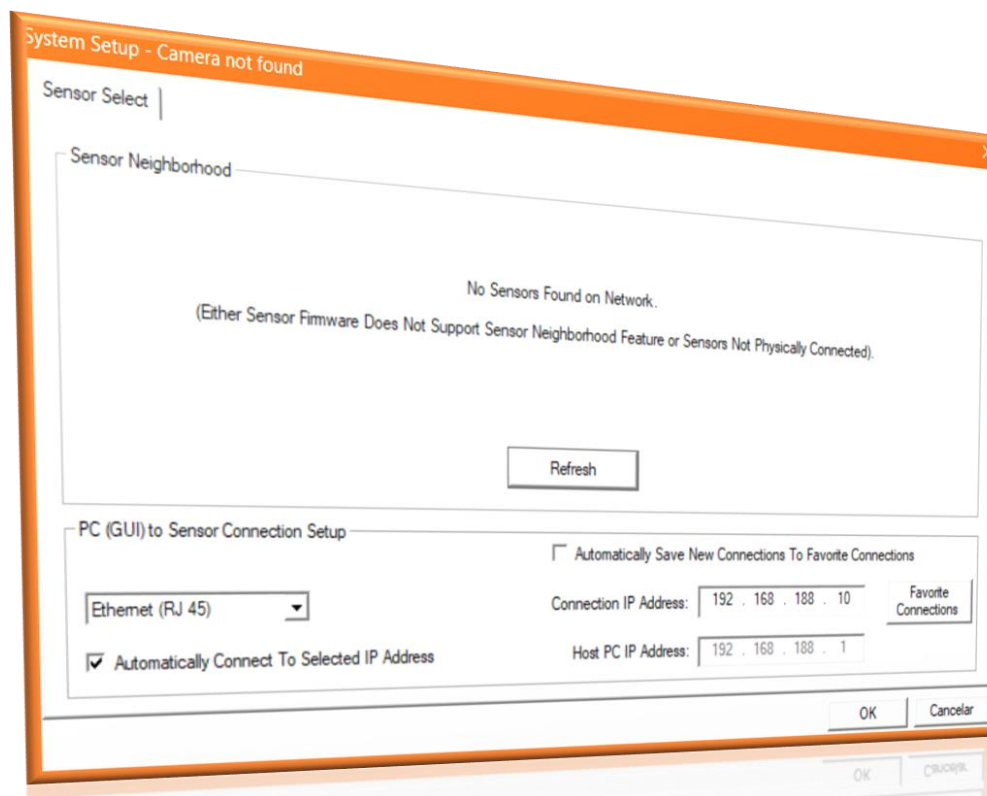


Figura 44 Configurações do endereço IP da *Smart Camera*

Depois de estabelecida a ligação, iniciou-se o programa *PresencePlus Vison* para iniciar a programação da câmara. No entanto, foi preciso alterar alguns campos no submenu *Setup*, nomeadamente o tempo de exposição e o ganho, para se obter uma imagem “limpa” e com uma qualidade suficientemente boa para utilizar na programação.

Após alguns testes, obteve-se os valores ideais para o tempo de exposição e para o ganho, 5,90 ms e 26 respetivamente. Obtendo-se a imagem da Figura 45.



Figura 45 Tempo de exposição e o valor do Ganho utilizado

4.2.4. PROGRAMAÇÃO DA SMART CAMERA

Posteriormente à inicialização da *Smart Camera*, deu-se início à programação da mesma. Para tal, teve-se de entrar no submenu *Tools*, local onde se encontram todas as ferramentas disponíveis para analisar as imagens (Figura 46).

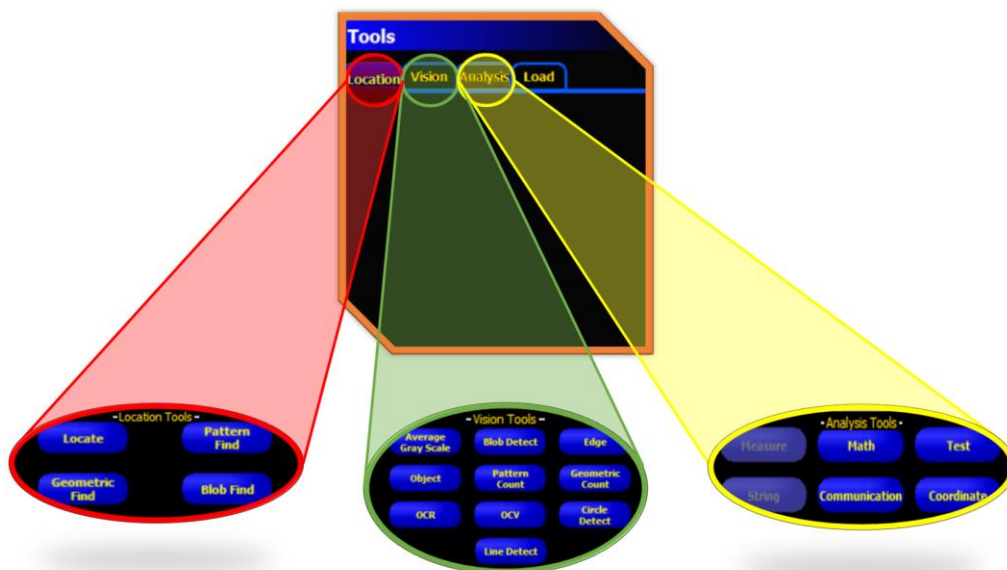


Figura 46 Ferramentas existentes para programação da *Smart Camera*

Dentro das ferramentas, existem três tipos diferentes:

- **Location** – contém ferramentas que são utilizadas para localizar padrões, objetos, *blobs* (“buracos”) e localizar a presença ou não de uma peça na imagem;
- **Vision** – contém uma vasta quantidade de ferramentas para diversas situações que possam aparecer. Dentro do leque de ferramentas para análise tem disponível a escala de cinzentos, detecção de *blobs*, de padrões, de formas geométricas, de limites e a detecção de caracteres (OCR);
- **Analysis** - contém as ferramentas de análise matemática, de teste (AND, OR, XOR), de medição, de comunicação e de localização cartesiana (x, y, ângulo).

Para uma melhor compreensão da programação utilizou-se fluxogramas que mostram as ferramentas, que se encontram no submenu *Tools* (Figura 46), utilizadas ao longo da elaboração do projeto até se ter atingido o resultado pretendido.

A Figura 47 mostra as primeiras ferramentas utilizadas para resolver o problema proposto, mais propriamente o **Geometric Find Tool** que vai ser utilizado para detetar a forma geométrica, neste caso o da cabeça do parafuso (círculo). O **Circle Detect Tool** que vai ser usado para encontrar as cabeças dos parafusos e o círculo do corpo do parafuso, complementando a outra ferramenta já utilizada, o que permitirá ver se estão presentes e na posição correta, visto que se encontrar o da cabeça do parafuso este está na posição correta caso encontre o círculo do corpo, está mal posicionado.

Por fim, também se usou o **Pattern Find Tool**, que consiste em encontrar um padrão, que neste caso foi o padrão de um parafuso deitado, para que sempre que fosse detetado expulsar o parafuso, e o **Test Tool**, que vai ser usado para medir o diâmetro da cabeça do parafuso dentro de um intervalo e o diâmetro do corpo do parafuso através dos resultados obtidos na ferramenta *Circle Detect Tool*, neste caso foi [13, 16] e [5,7] mm, respetivamente. Como a cabeça do parafuso mede 14 mm, e se se encontrar dentro do intervalo estabelecido, prova que o parafuso está na posição correta, caso o valor do diâmetro do corpo esteja dentro do intervalo prova que está mal orientado.

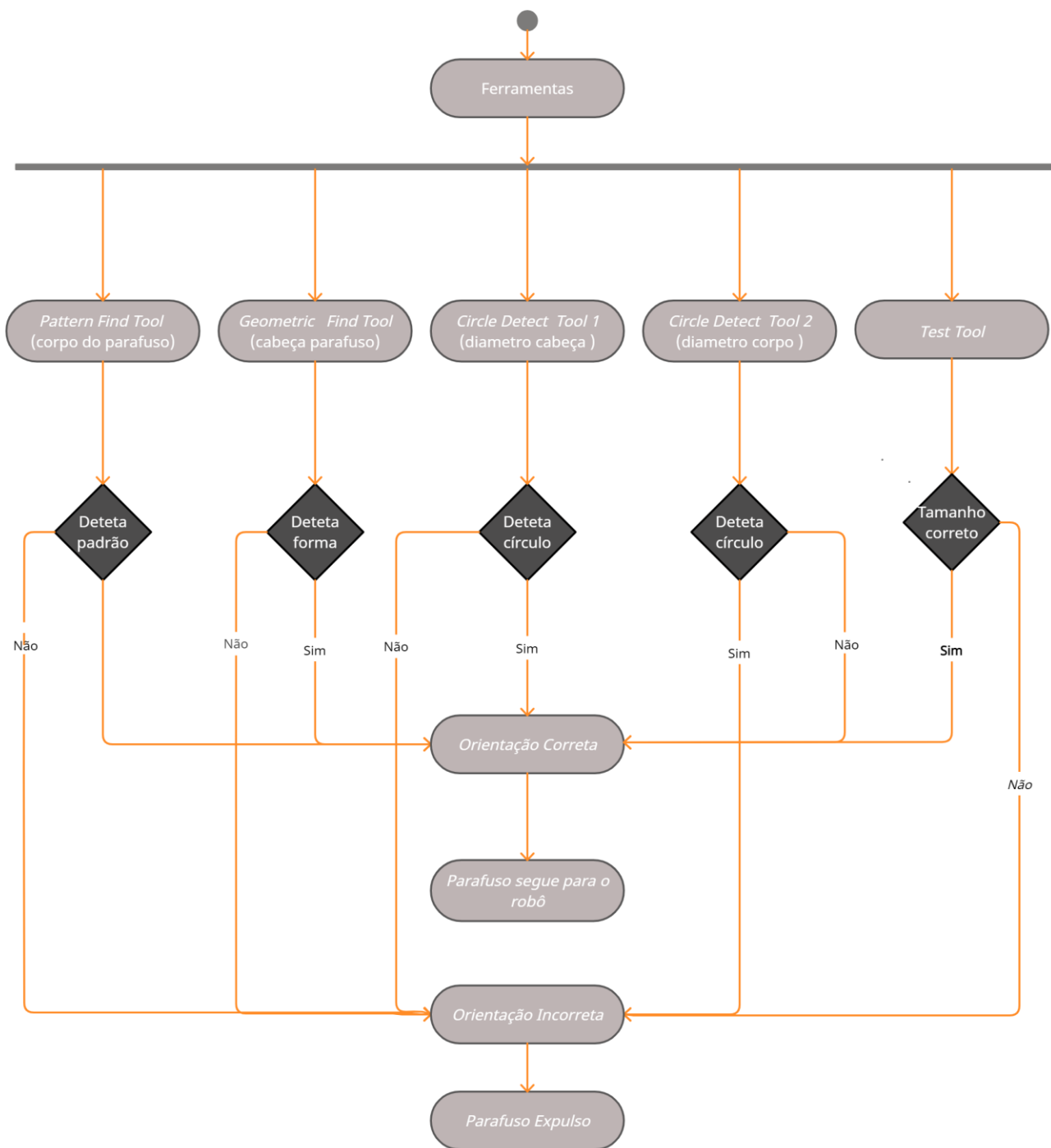


Figura 47 Fluxograma 1ª Tentativa

Depois de escolhidas as ferramentas a utilizar, deu-se início à programação da *Smart Camera*. Começou-se por tentar encontrar o padrão das cabeças dos parafusos, orientação correta, através do uso da ferramenta *Geometric Find Tool* (Figura 48).

Para tal, teve-se de configurar alguns parâmetros antes de a utilizar, como a **ROI**, que consiste em definir a região de interesse onde se quer encontrar o padrão, o **Threshold** que tem duas alternativas, o **Adaptive** (O sensor define automaticamente o limite e tenta ajustar as variações de contraste) e o **Edge strength** (Define-se o limite num valor na escala de cinza entre 0 e 127, fornecendo mais controlo sobre o que pode ser considerado um pixel de limite), sendo o **Edge strength** o escolhido por se obter melhores resultados com o valor de 50 na escala de cinza. Em seguida configurou-se o **Edge Length** com o valor 12, este campo ignora os padrões alvo com limites iguais ou inferiores ao número especificado de pixels para o valor limite do comprimento.

Por fim alterou-se o **Rotation Range**, para +180 e -180, de forma a definir a rotação máxima permitida (0 ° a 180 °) que um padrão possa ter e ainda ser identificável, e o **Minimum Acceptance Level** para 80%, pois é neste campo que se define o nível de aceitação para correspondências de padrão.

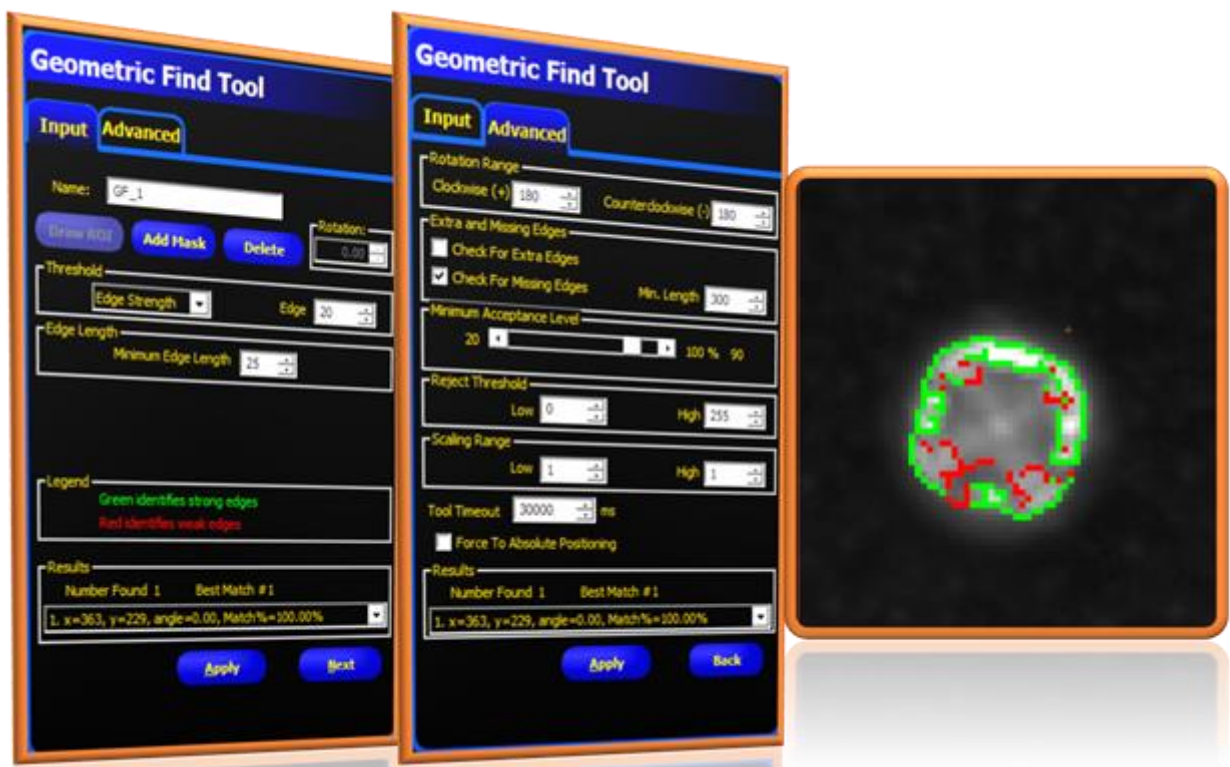


Figura 48 Comando *Geometric Find Tool* para deteção de padrão

Em seguida, para complementar a função de encontrar os padrões adicionou-se uma nova ferramenta o **Circle Detect Tool** (Figura 49), que consiste em encontrar um círculo

numa zona definida na imagem, neste caso um dos círculos corresponde à cabeça dos parafusos e o outro ao diâmetro do corpo do parafuso.

Caso o círculo da “cabeça” do parafuso seja detetado, quer dizer que o parafuso está na posição correta, caso seja o do diâmetro do corpo, quer dizer que se encontra mal posicionado.

Da mesma forma que na ferramenta anterior, nesta também foi necessário efetuar algumas configurações no campo *Edges*, nomeadamente no ***Threshold***, este campo varia entre 5 e 255, sendo o valor 20 o escolhido, visto ser o valor padrão dito pelo fornecedor, a ***Polarity***, que é onde se determina se a procura dos limites do círculo vai ser feita na direção do claro para o escuro (*Bright to Dark Edge*), do escuro para o claro (*Dark to Bright Edge*) ou em ambos os sentidos (*Any Edgde*). Neste projeto escolheu-se a ***Polarity Bright to Dark Edge***, devido a oferecer melhores resultados que as outras opções.

A ***Sample Rate***, define o espaçamento entre as linhas de varredura invisíveis no ponto médio entre os anéis interno e externo da ROI. Quando a taxa de amostragem é aumentada, menos arestas serão encontradas, quando a taxa de amostragem é reduzida, mais borda será encontrada e melhor resultados serão obtidos, por isso atribuiu-se-lhe o valor 1.

Por fim configurou-se o ***Search For Field***, que oferece três opções de ajustar um círculo aos pontos de borda que a ferramenta encontra:

- ***"Circle, Best Points"***, é a escolha mais comum. Descobre automaticamente quais pontos da borda formam o melhor círculo e ignora o resto. Este modo é muito tolerante a ruído ou a bordas extras na ROI, mas em algumas circunstâncias pode ignorar as bordas que se deseja incluir.
- ***"Circle, All Points"***, encontra um círculo que une todas as arestas da melhor forma possível, sem ignorar nada. Essa configuração é a mais rápida, mas não funciona bem se houver bordas extras ou ruído na ROI. Este modo é útil se o círculo que está sendo examinado for irregular ou elíptico.
- ***"Circular Arc"***, encontra só um pedaço de um círculo. Esta configuração é usada para encontrar o ângulo inicial ou final de um arco ou uma dobra.

Depois de serem testadas as três opções, concluiu-se que a opção que mais se enquadrava e obteve melhores resultados era a *Circle, Best Points*, acabando por se escolher esta opção.

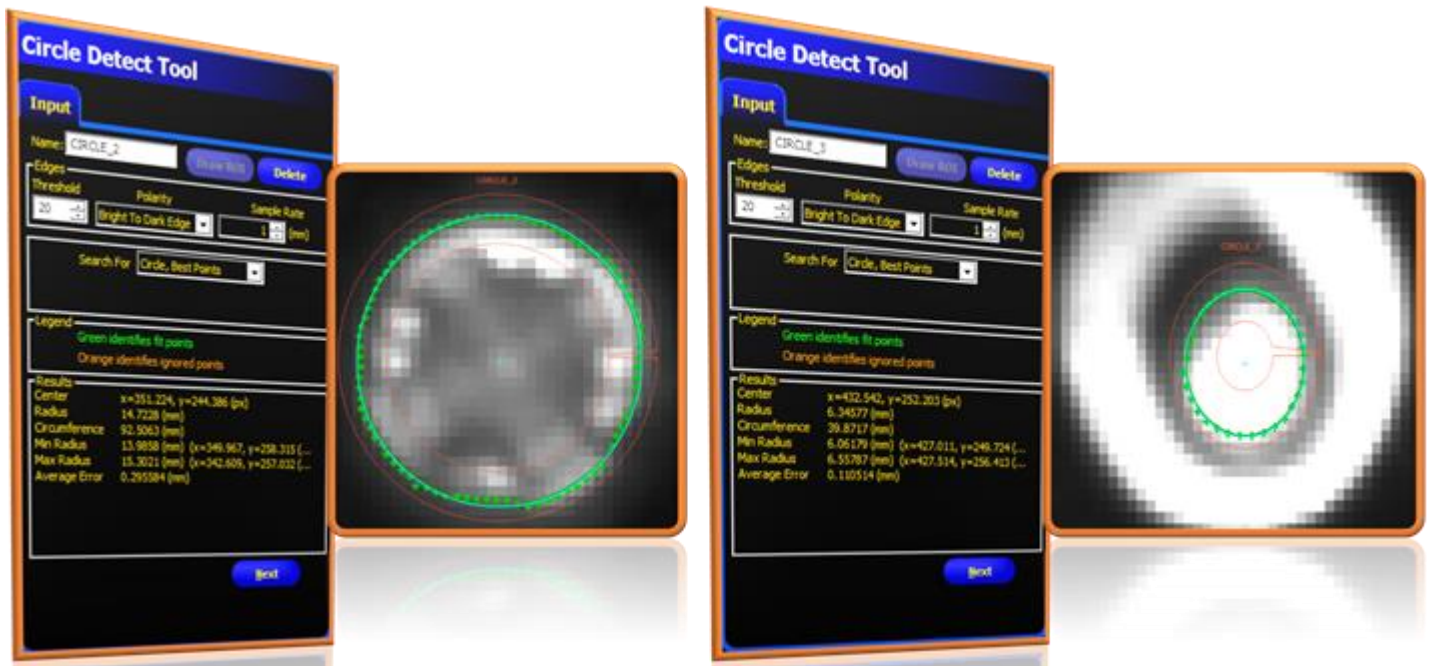


Figura 49 Comando *Circle Detect Tool* para detetar as cabeças dos parafusos (esquerda) e o diâmetro do corpo (direita)

Posteriormente, adicionou-se uma ferramenta de teste, o *Test Tool*, que consiste em usar lógica booleana para combinar ou converter os resultados das ferramentas inseridas nos *Inputs* (Figura 50). Esta ferramenta exibe os intervalos de resultados obtidos, normalmente esses intervalos são definidos automaticamente pelo *Quick Teach* ou podem ser definidos ou modificados manualmente antes de executar a inspeção.

Neste caso, nos *Inputs* foi inserida a ferramenta *Circle Detect Tool*, mais propriamente a que deteta a cabeça e a do diâmetro do corpo, de modo a definir um intervalo de valores aceitáveis para o diâmetro da cabeça e do corpo do parafuso. Os valores foram inseridos manualmente, criando o intervalo [13,16] mm para a cabeça, usando a ferramenta *Circle Detect Tool 1*, e de [5,7] mm para o corpo, usando a ferramenta *Circle Detect Tool 2*, de maneira a excluir valores fora deste intervalo.

E ainda se ativou a opção *Pass/Fail Contribution*, para que o resultado obtido da operação descrita anteriormente contribui-se para a saída I/O correspondente.



Figura 50 Comando *Test Tool* para definir os intervalos de aceitação

Depois de implementadas as três ferramentas passou-se à realização de testes para ver se tudo estava a funcionar corretamente. Foi nessa fase que se detetou que não se estava a obter os melhores resultados com a utilização do *Pattern Find Tool*, visto esta não encontrar os padrões dos parafusos deitados quando estes não vinham bem alinhados e conseguinte a ferramenta *Circle Detect Tool* (diâmetro do corpo) também não apresentava os melhores resultados.

Deste modo, teve-se de descartar esta hipótese de análise e começar a pensar em novas hipóteses.

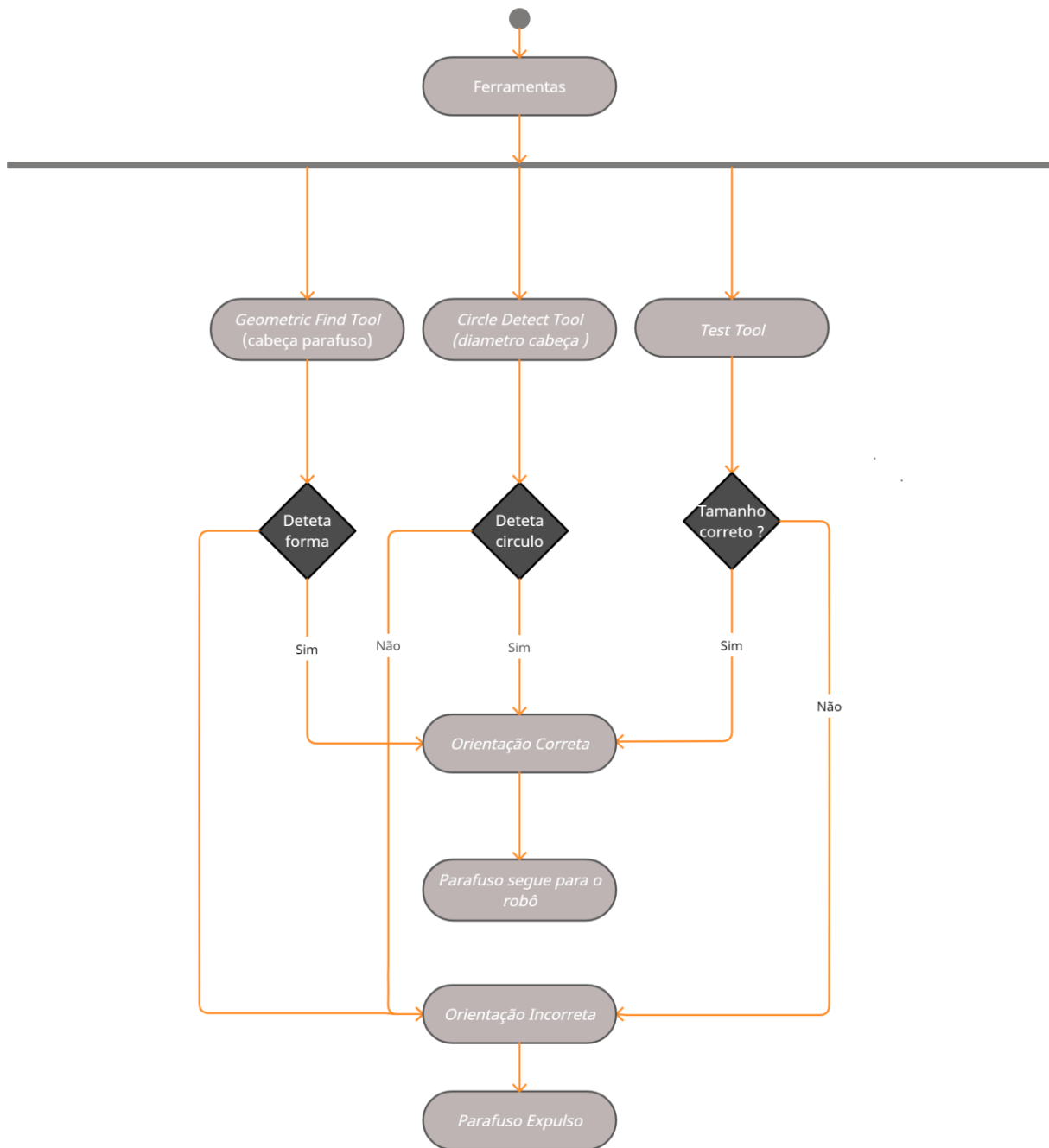


Figura 51 Fluxograma 2ª Tentativa

A Figura 51 mostra as novas ferramentas utilizadas para tentar resolver os problemas encontrados anteriormente. Retirou-se a ferramenta do *Pattern Find Tool* e o *Circle Detect Tool* (diâmetro do corpo) e optou-se por fazer umas alterações no *Circle Detect Tool*, aumentando um pouco o tamanho da ROI para obter um melhor resultado dos limites das cabeças dos parafusos, já no *Geometric Find Tool* diminui-se o tamanho da

ROI para o mesmo objetivo e por fim alterou-se também o intervalo de aceitação da ferramenta *Test Tool*, para [16, 18] mm.

Para implementar esta alternativa, foi necessário fazer novamente uma configuração nas ferramentas a utilizar. Na *Geometric Find Tool*, diminuiu-se a ROI, ou seja, a área de procura foi reduzida (Figura 52) de maneira a se obter melhores resultados, e alterou-se o valor do *Edge Length* para 12 e no campo *Threshold* o valor do *Edge* para 50.

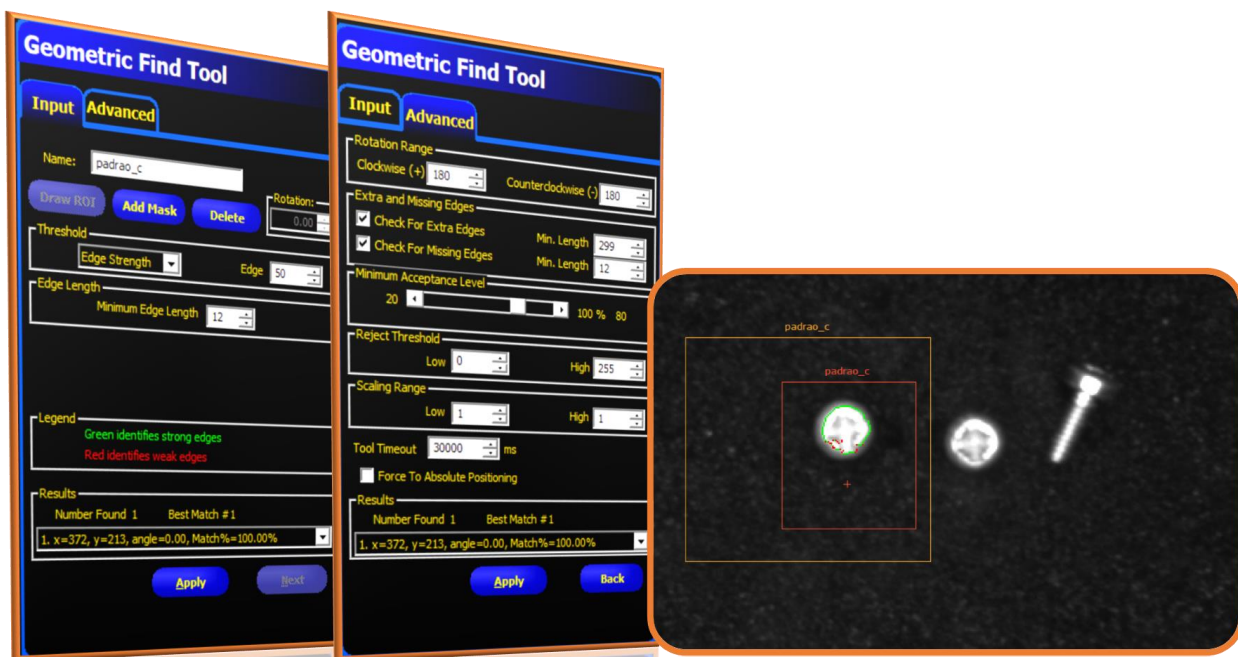


Figura 52 Comando *Geometric Find Tool* para detecção de padrão

Posteriormente procedeu-se à configuração da ferramenta *Circle Detect Tool*, nomeadamente à alteração do *Threshold* de 20 para o valor mínimo 5 e o desenho de uma nova ROI significativamente maior que a anterior (Figura 53), de maneira a encontrar mais facilmente o círculo que representa a cabeça do parafuso e assim obter melhores resultados.

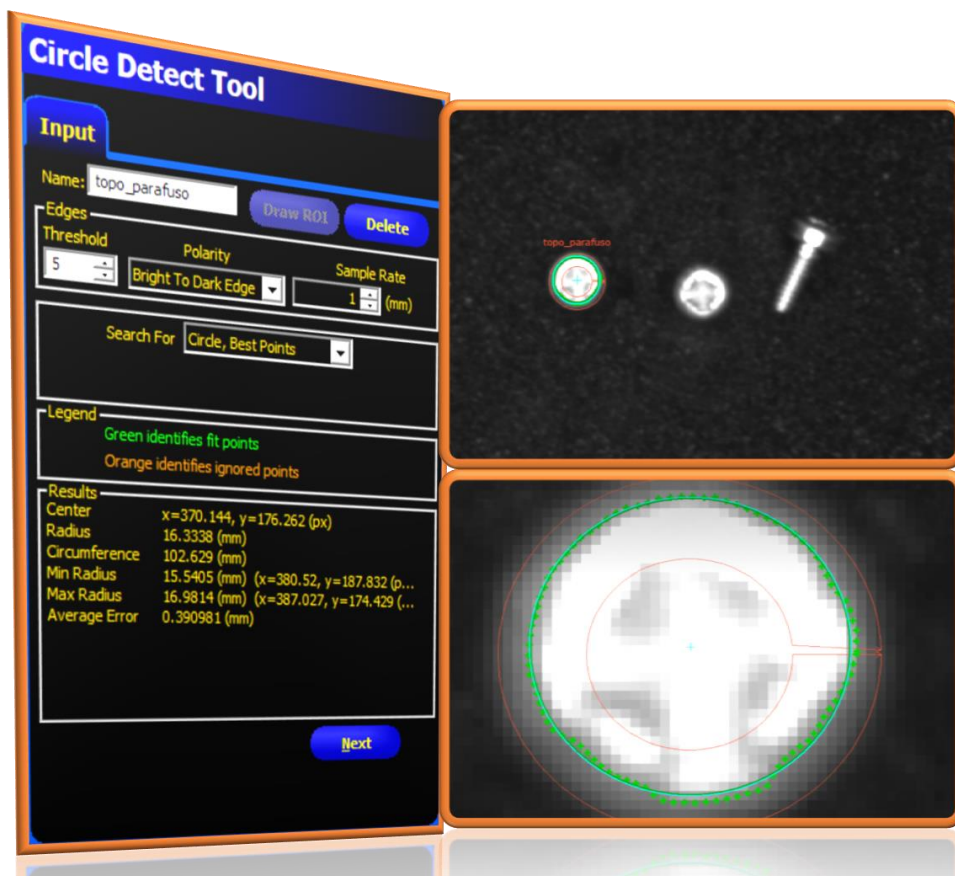


Figura 53 Comando *Circle Detect Tool* para detetar as cabeças dos parafusos

Em seguida, teve-se de alterar a ferramenta *Test Tool*, devido a se ter aumentado a área de análise do *Circle Detect Tool*. De tal forma, que se teve de aumentar o intervalo de aceitação do tamanho dos círculos para [16,18] mm e assim conseguimos obter os resultados da Figura 54.

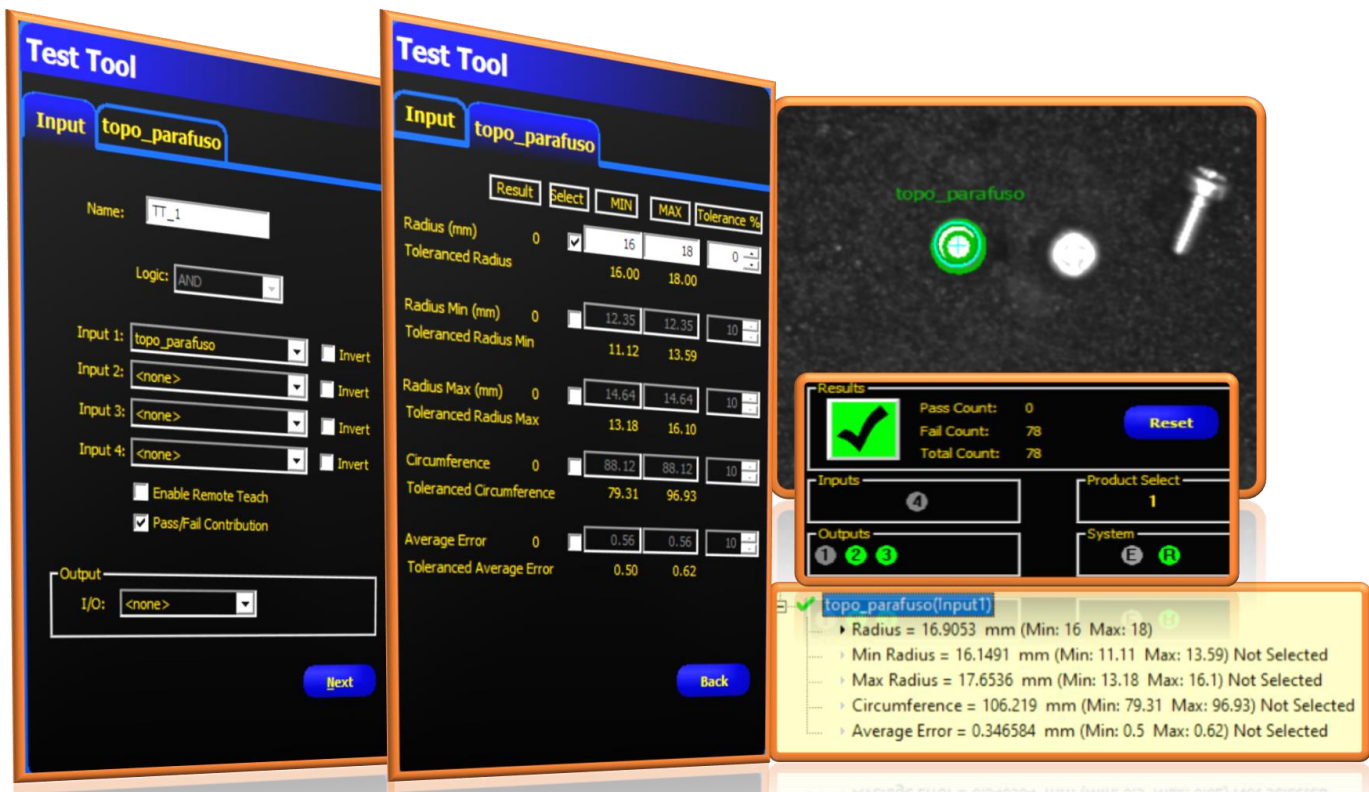


Figura 54 Comando *Test Tool* para definir intervalo de aceitação

Posteriormente, fez-se um pequeno ajuste que originou uma melhoria nos resultados obtidos. Alterou-se o valor do ganho e do tempo de exposição para se obter uma melhor imagem das cabeças dos parafusos e desta maneira conseguiu-se obter resultados muito mais satisfatórios com as mesmas ferramentas.

A alteração consistiu em passar o tempo de exposição de 5,90 para 30,10ms e o valor do ganho de 26 para 20, o que ajudou a obter uma imagem mais nítida das cabeças dos parafusos como se pode verifica na Figura 55.

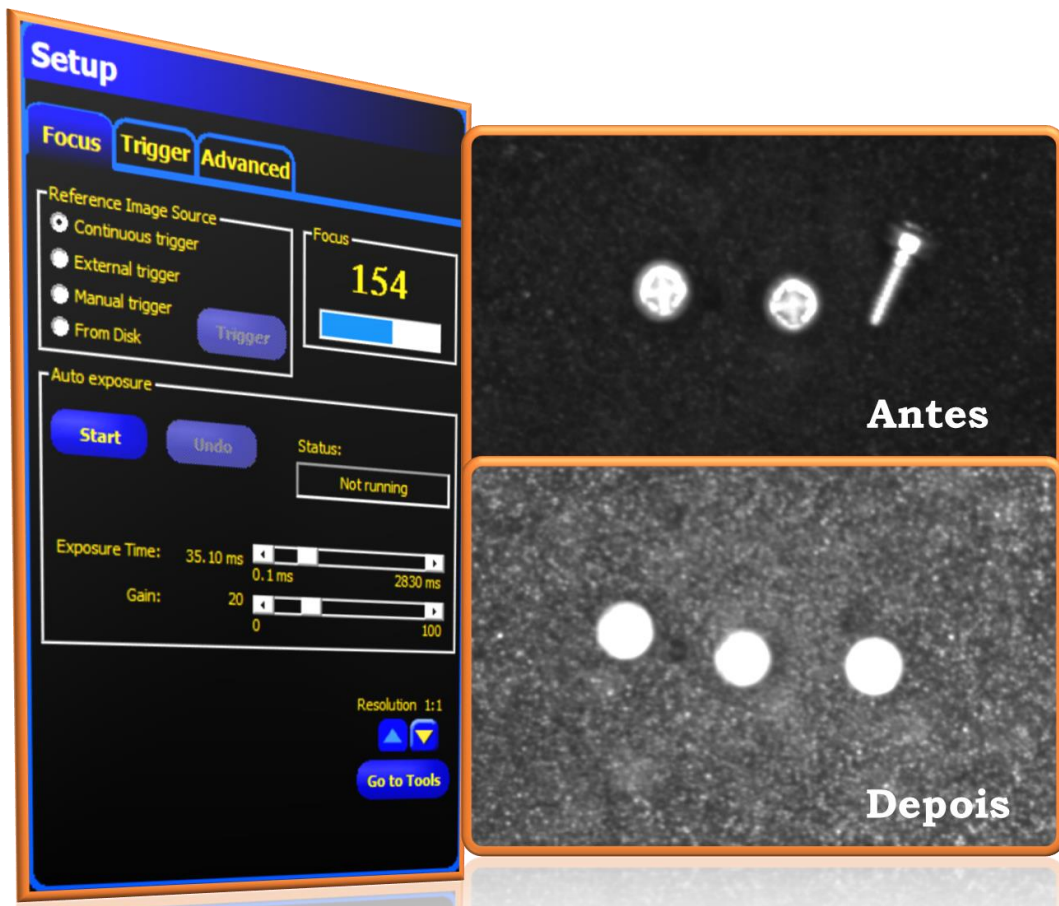


Figura 55 Resultado obtido depois da alteração do ganho e do tempo de exposição

Com estas mudanças foi possível obter o resultado que se pretendia desde o início e desta maneira conseguir analisar as cabeças dos parafusos mesmo que estas estivessem um pouco desalinhadas ou houvesse a ausência de algum parafuso.

Posteriormente, após algum estudo, optou-se por fazer um outro programa (Figura 56), onde se iria analisar a posição incorreta do parafuso, ou seja, detetar se o parafuso vinha deitado ou não. Para tal e como já estávamos mais à vontade com o programa foi mais fácil e optou-se logo pela ferramenta *Line Detect Tool* que iria encontrar uma ou até 4 linhas (uma na horizontal, uma na vertical e duas na diagonal) de modo a detetar qualquer posicionamento do parafuso, essa linha representa o corpo do parafuso, desta forma conseguir detetar se o parafuso vem deitado ou não.

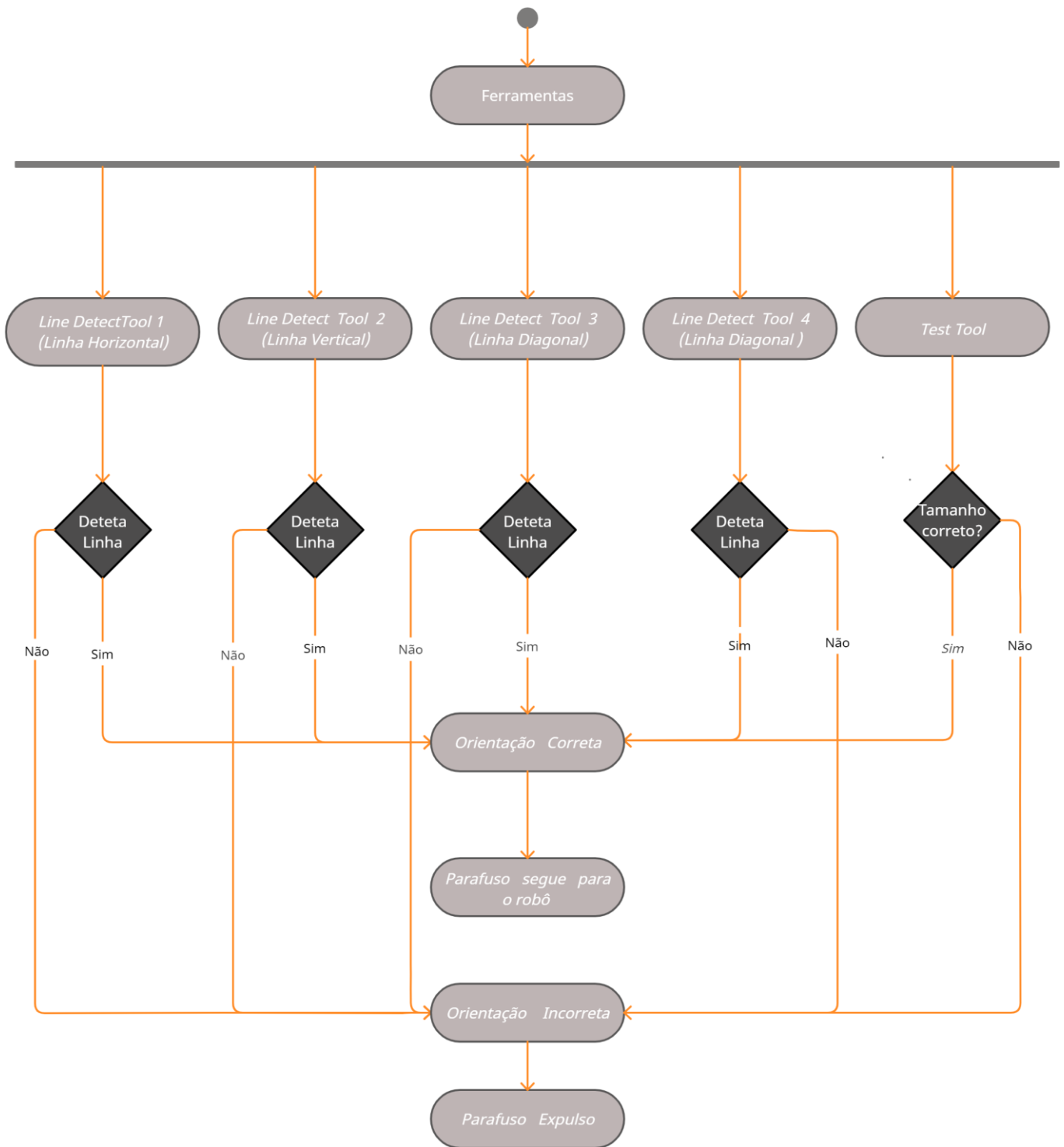


Figura 56 Fluxograma do programa “Parafuso Deitado”

A Figura 56 mostra as ferramentas utilizadas na programação do novo programa. Depois de escolhidas as ferramentas a utilizar, deu-se início à programação da *Smart Camera*.

Para tal, teve-se de configurar alguns parâmetros antes de a utilizar, como se tem vindo a fazer desde o início. Neste caso, teve de se configurar a ferramenta *Line Detect Tool* (Figura 57), mais propriamente o *Threshold* que para este problema atribui-se o valor 35 de maneira a encontrar melhor os limites do corpo do parafuso. Em seguida definiu-se a *Polarity* para *Bright to Dark* e ao *Sample Rate* atribui-se o valor 1, de modo a obter resultados mais precisos e até mesmo falhas.

Como o *Search For* tem duas alternativas:

- "*Line, Best Points*" é a escolha mais comum. Pois deteta automaticamente quais os pontos de borda que formam a melhor linha e ignora o resto. Este modo é tolerante a ruído ou bordas extras na ROI.
- "*Line, All Points*" encontra uma linha que junta todas as arestas da melhor forma possível, sem ignorar nada. Esta configuração é a mais rápida, mas funciona mal se houver bordas extras ou ruído na ROI. Este modo pode ser útil se a linha que está sendo examinada for irregular ou curva.

Com o exposto, configurou-se e escolheu-se a "*Line, Best Points*", pois apresentava melhores resultados na deteção do corpo do parafuso em comparação com a "*Line, All Points*", visto que esta às vezes associava a cabeça do parafuso como linha e dava como posição incorreta quando na realidade não era isso que acontecia.

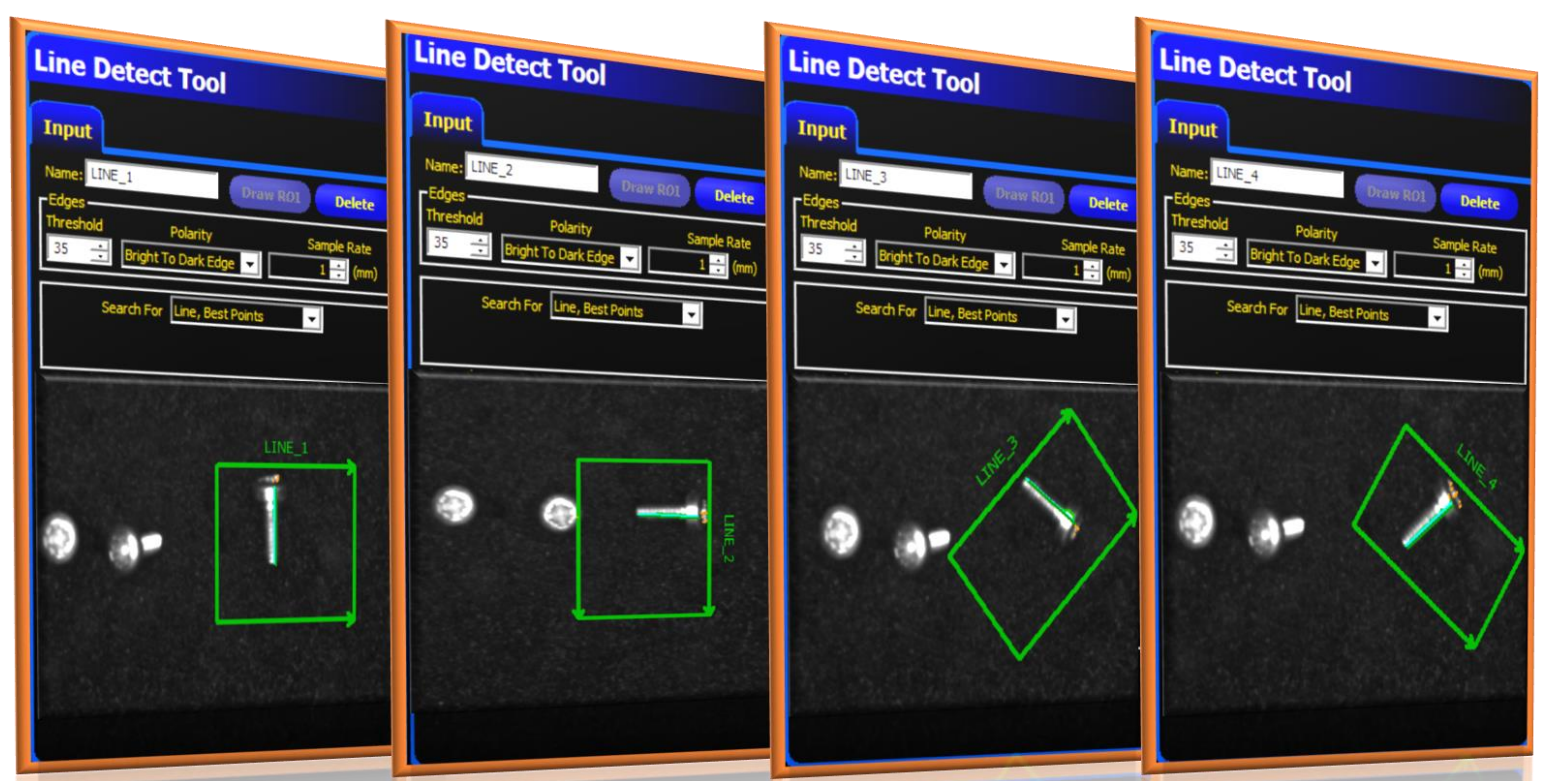


Figura 57 Comando *Line Detect Tool* para deteção do corpo do parafuso

Seguidamente, configurou-se a ferramenta *Test Tool*, onde se inseriu cada uma das ferramentas *Line Detect Tool* nos *Inputs* correspondentes (Figura 58). No campo *Logic* escolheu-se a operação logica OR, de modo que basta obter um resultado positivo de uma das ferramentas que o parafuso e logo dado como deitado e posteriormente expulso para o alimentador vibratório. Por fim, ativou-se a opção *Pass/Fail Contribution*, para que o resultado obtido da operação descrita anteriormente contribui-se para a saída I/O correspondente.

Neste caso, o intervalo de aceitação escolhido para o comprimento do corpo do parafuso foi [46,62]mm, deste modo sempre que o valor obtido, na análise feita pela ferramenta *Line Detect Tool*, estivesse dentro deste intervalo era reportado como “Parafuso Deitado” e expulso para o alimentador.



Figura 58 Comando *Test Tool* para definir intervalo de aceitação do tamanho do corpo do parafuso

Após a conclusão da programação dos dois programas, um para a detecção das cabeças dos parafusos (Posição Correta) e outro para a detecção do corpo dos parafusos (Posição Incorreta), procedeu-se à configuração das saídas I/O.

Para tal, foi necessário entrar no menu *System* e em seguida em *InputOutput* (Figura 59) para programar as saídas. A *Smart Camera* possui 4 saídas I/O, mas para os dois programas em questão só foram utilizadas as três primeiras saídas, que correspondem aos Pin 5, 6 e 7 respetivamente (Figura 36), em que a 1ª saída foi configurada para ser ativada sempre que o resultado obtido pela ferramenta *Test Tool* fosse “Pass”, ou seja, sempre o que resultado obtido fosse o correto. A 2ª saída foi configurada para ser ativada sempre que o resultado obtido fosse “Fail”, ou seja, sempre o que resultado obtido fosse o incorreto. A 3ª saída foi configurada para ser ativada quando a *Smart Camera* estivesse pronta a funcionar e desta maneira acender uma luz que correspondente ao seu estado de funcionamento (verde ou vermelha).

Em seguida, todas as saídas foram configuradas como *Normally Open*, em que a saída fica ativa quando a condição lógica que controla a saída é verdadeira. E o *Output Duration* como *Latched*, em que a saída fica ativa até que a condição mude (Figura 58).

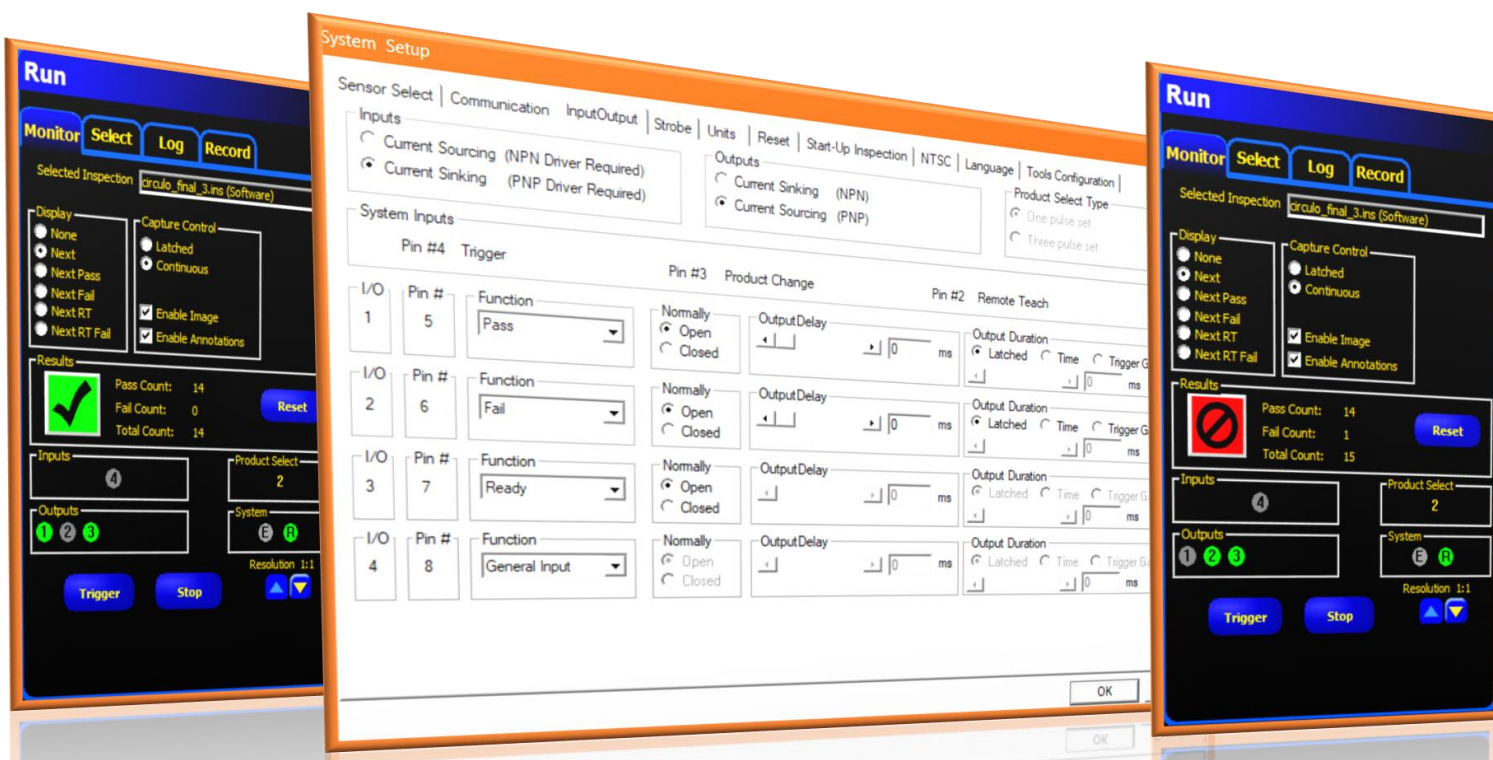


Figura 59 Configuração das saídas I/O

4.2.5. PROGRAMAÇÃO DA INTERFACE GRÁFICA

Posteriormente à programação da *Smart Camera*, iniciou-se a elaboração de uma interface gráfica (Figura 60) para uma melhor interpretação e visualização dos dados obtidos nas capturas das imagens dos parafusos.

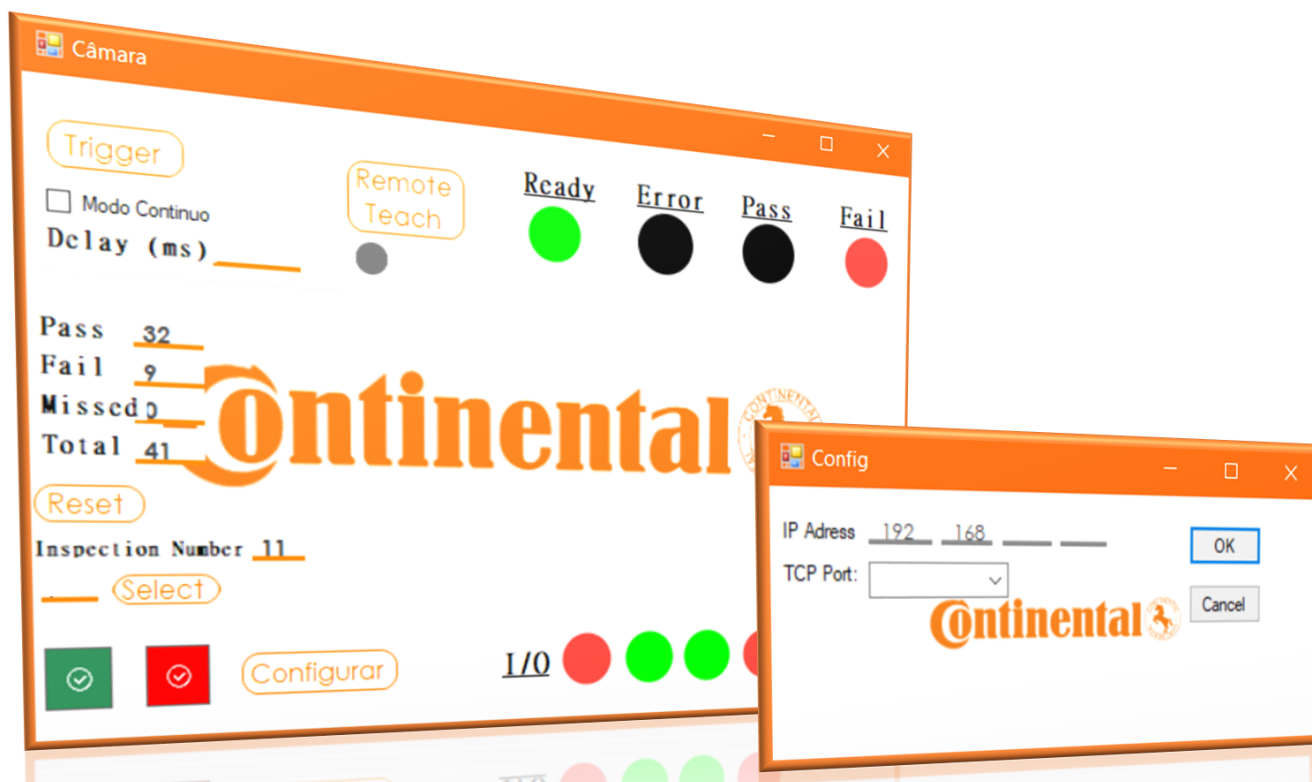


Figura 60 Interface Gráfica

O ecrã principal do utilizador é o ilustrado na Figura 61, este é o ecrã que encontra quando se abre a aplicação. Neste ecrã o utilizador para ter acesso à *Smart Camera*, tem de configurar o endereço IP e o da Porta TCP. Depois de configurada, o utilizador consegue-a colocar em modo contínuo, com o *delay* que pretender, ou em modo manual e tirar fotos só quando pretender.

Neste ecrã, é possível visualizar o número de parafusos que passam (*Pass*), que reprovam (*Fail*), que são perdidos (*Missed*) e o total de análises feitas (*Total*) e por outro lado, escolher o programa que se quer que a *Smart Camera* use para a deteção do defeito em questão. Por fim, consegue-se visualizar as saídas que são ativas dependendo do resultado obtido e se a camara está pronta a funcionar ou se existe algum erro que o impeça.

O software utilizado para desenvolver a aplicação (Figura 61) foi o Visual Studio 2019, usando o Windows Forms App (.Net).

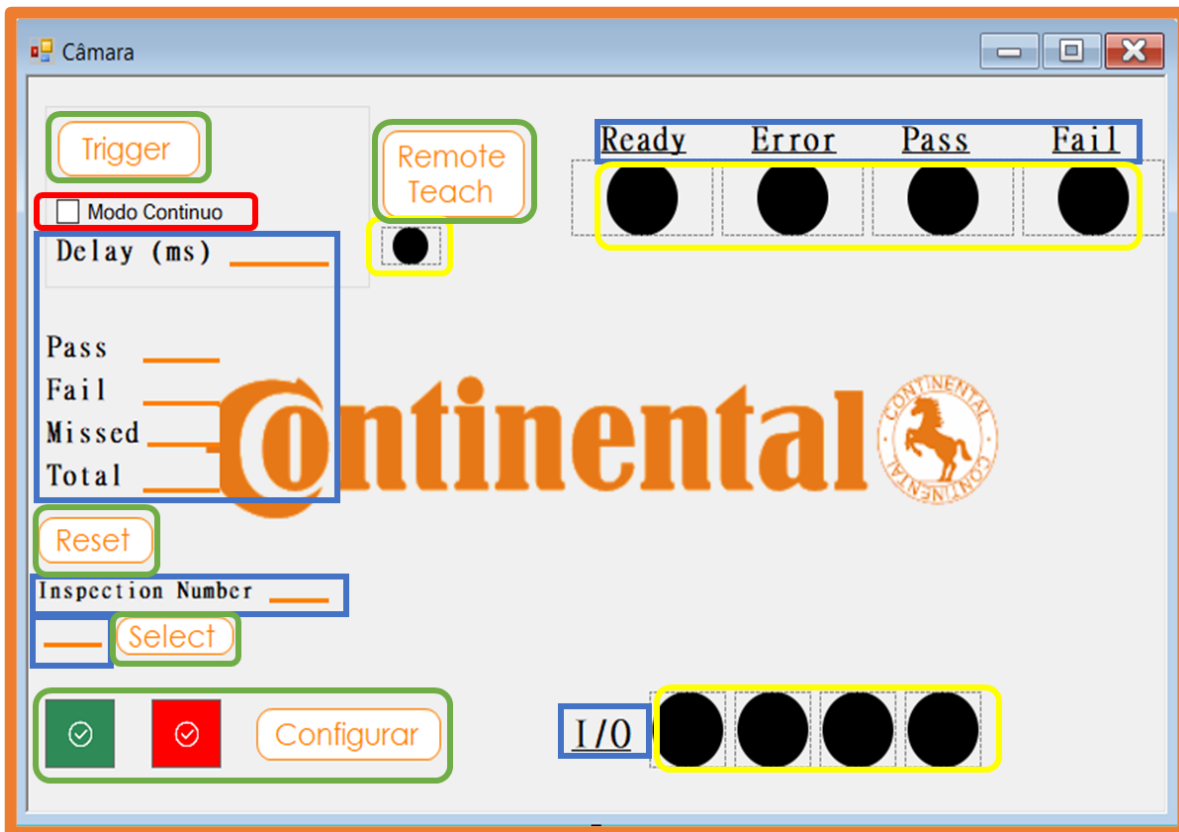


Figura 61 Ecrã Principal

Foi usado o Windows Forms App (.Net), pois é muito intuitivo e fácil de utilizar para a construção de aplicativos graficamente ricos. Estes podem funcionar quando estão conectados ou desconectados da Internet e podem aceder a recursos no computador local de maneira mais segura.

Contudo apresenta várias vantagens em relação a outros tipos de *templates* de projetos existentes no Visual Studio, entre elas:

- A possibilidade de utilização da funcionalidade *Drag & Drop* de ferramentas permitem uma configuração simplificada do sistema;
- As funções, opções e operações dedicadas permitem a resolução de aplicações complexas;
- Exibir e manipular dados;

- Criar interfaces de usuário de forma robusta, flexível e eficiente tornando-as sofisticadas e interativas.

O ecrã principal foi desenhado com ajuda de algumas ferramentas existentes no Windows Forms entre elas:

- Os retângulos verdes, correspondem aos botões criados que desempenham diferentes funções, como acionar *trigger*, ligar e desligar, fazer *Reset* e selecionar;
- Os retângulos azuis, correspondem às caixas de texto/ *Labels* utilizadas para escrever algum texto ou apresentar os valores obtidos;
- O retângulo vermelho, corresponde à *CheckBox* criada, para quando for selecionada ativar o modo contínuo da *Smart Camera*;
- Os retângulos amarelos, correspondem às *Pictures* (imagens), que vão alterar de cor para verde ou vermelho se o resultado for o esperado ou não.

Em seguida explica-se grande parte da programação desenvolvida para a interface gráfica do ecrã principal. Apresentam-se também todas as funções usadas para melhor se entender o procedimento para a criação da interface, encontrando-se o código completo no Anexo C.

Primeiramente, foi necessário importar a *Socket Class* e o *Thread Class* para que fosse possível estabelecer a conexão TCP entre a *Smart Camera* e a aplicação (PC) e para criar e controlar um *thread*, definindo a sua prioridade e obter o seu estado, respetivamente.

```
Imports System.Net.Sockets
Imports System.Threading
```

Seguidamente, deu-se início à programação da aplicação em questão começando pela função **Form1_Load**, que consiste em desativar todos os botões à exceção do botão “Configurar” quando se inicia a aplicação e desta forma o utilizador é “obrigado” a

configurar o endereço IP e a Porta TCP, no **Ecrã Config** (Figura 62), para conseguir iniciar a comunicação com a *Smart Camera*.

```
Trigger.Enabled = False
ModoContinuo.Enabled = False
Remote_Teach.Enabled = False
Selectprograma.Enabled = False
Reset.Enabled = False
Desconectar.Enabled = False
Ligar.Enabled = False
```

É neste ecrã, que o utilizador insere o endereço IP da *Smart Camera* que quer ligar e o número da porta TCP. Caso não preencha corretamente os campos pretendidos aparecerá uma mensagem de erro a avisar que não preencheu corretamente os campos.

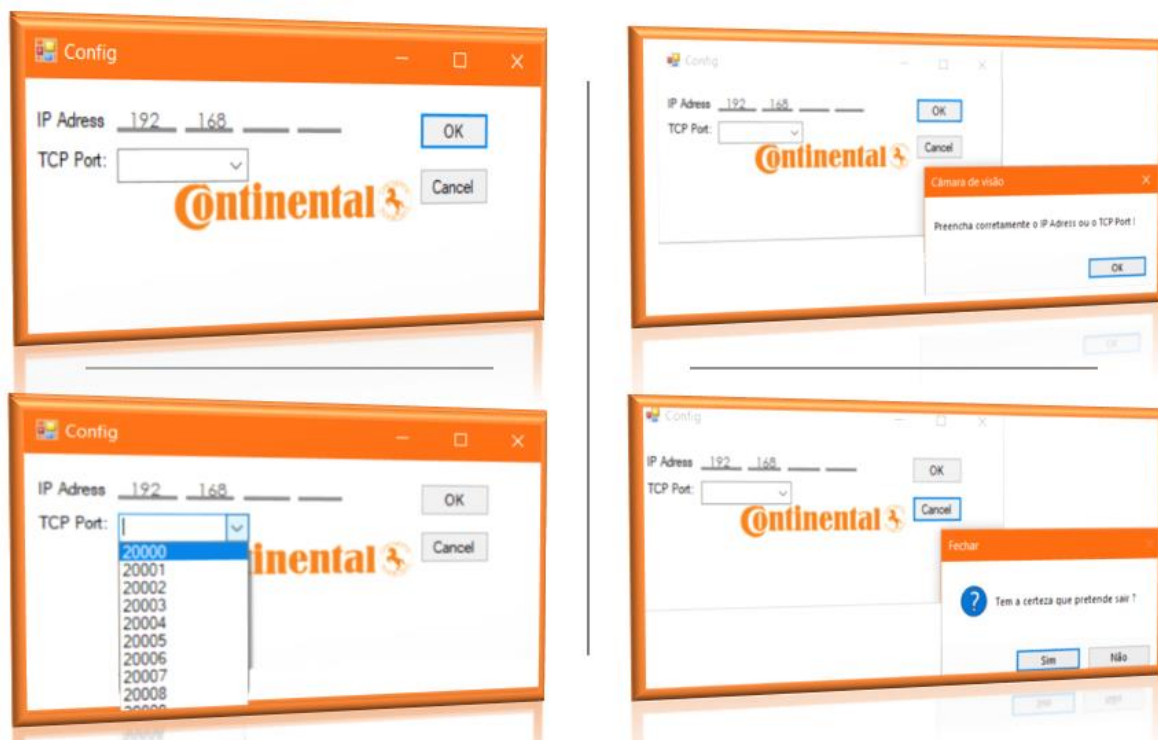


Figura 62 **Parâmetros a preencher e Warnings no Ecrã Config**

A função **Ligar_Click**, é ativada quando o utilizador clica no botão “verde” após já ter configurado os endereços através do botão “Configurar”, senão não é ativada. Quando é ativada, vai ler o endereço IP e o TCP *Port* que foram atribuídos no **Ecrã Config** e se forem os corretos será estabelecida a conexão com a *Smart Camera*.

Após estabelecida a comunicação, inicia a task que chama a função **valores_entrada**, e espera que termine, para retornar os dados relativos ao número de *Pass*, *Fail*, *Missed* e *Total*, bem como o programa que está a ser executado, as saídas I/O que se encontram ativas e se a *Smart Camera* está pronta a funcionar ou se há algum erro.

```
IPAddress = Config.IP1.Text & Chr 46) &
Config.IP2.Text & Chr (46) & Config.IP3.Text &
Chr (46) & Config.IP4.Text
Porta = Config.TCP_port.Text

If OpenETH1(IPAddress, Porta) = True Then

client.SendTimeout = 0
stream = client.GetStream()

Dim tf3 As New Task (Sub()valores_entrada())

tf3.RunSynchronously()
tf3.Wait()

Pass.Text = passcnt
Fail.Text = failcnt
Missed.Text = misscnt
inspection_number.Text = programa
Total.Text = totalcnt
```

É através da função **OpenETH1** que consegue estabelecer a ligação, pois esta é que faz o ping à *Smart Camera* e se tiver sucesso em seguida estabelece a ligação, caso contrário informa que não foi possível estabelecer a ligação.

```
If Not My.Computer.Network.Ping(IPAddress, Porta)
Then

Console.WriteLine(" Scanner with IP: " & IPAddress
& " fail")
Return False

End If

client.Connect(IPAddress, Porta)
Return True
```

Posteriormente, a estabelecer a conexão, o utilizador tem a seu dispor o botão “*Trigger*” que serve para mandar um comando à *Smart Camera* e esta tirar uma fotografia, analisá-la e retornar para a aplicação os resultados obtidos.

Quando se clica no botão, este inicia uma Task que vai chamar a função **trigger_valores** e espera que esta termine para retornar os valores obtidos e apresentá-los na interface.

```
Dim tg As New Task(Sub() trigger_valores())
tg.Start()
tg.Wait()

Pass.Text = passcnt
Fail.Text = failcnt
Missed.Text = misscnt
Total.Text = totalcnt
```

Contudo, se o utilizador optar por um modo “automático” só tem de seleccionar o “**Modo Contínuo**” e inserir o *delay* (intervalo de tempo entre fotografias) que pretende em milissegundos, este valor por defeito é de 10ms. Ao seleccionar este modo, desativa o botão “*Trigger*” e inicia o *Timer1*, que irá repetir a função **trigger_valores** no intervalo de tempo igual ao definido no *delay*.

```
If ModoContinuo.Checked = True Then

If timer.Text = "" Or timer.Text = "0" Then
timer.Text = 10
End If

Timer1.Interval = timer.Text
timer.Enabled = False
Trigger.Enabled = False
Timer1.Start()
```

Caso o utilizador pretenda mudar o programa que está a ser utilizado pela *Smart Camera* pode-o fazer inserindo o número do programa que pretende, este número corresponde a posição em que está guardado na memória da câmara (1-99), e em seguida clicar no botão “*Select*”.

```

programa = sensor.Text

Dim prog = Readincomigdata("exec prodchange " &
programa)

        Readincomigdata("get inspnum")
Dim insp = Readincomigdata("get inspnum")

If insp.Length > 1 Then
inspection_number.Text = insp.Substring(3, 2)

Else
inspection_number.Text = insp.Substring(0, 1)

```

Porém o utilizador ainda dispõe de mais duas funções na interface, a de *Remote Teach* através do botão “Remote Teach”, que serve para atualizar remotamente os parâmetros de inspeção no modo *Run*, onde a(s) ferramentas(s) de visão e de teste podem ser selecionadas para serem ensinadas.

Ao clicar no botão será enviada uma instrução à *Smart Camera* e esta irá tirar uma nova fotografia e “aprender” os novos parâmetros e, desta maneira adaptar-se as novas circunstâncias apresentadas, retornando o resultado da aprendizagem através de um sinal verde ou vermelho consoante o resultado obtido.

```

Dim teach = Readincomigdata("exec rteach
trigger")

If teach.Substring(0, 1) = 0 Then
rt_picture.Image = My.Resources.circulo_verde
Else
rt_picture.Image = My.Resources.circulo_vermelho

```

A outra função é a de *reset*, através do botão “Reset”, que serve para limpar as contagens de *Pass*, *Fail*, *Missed* e *Total*. Ao clicar no botão, envia duas instruções à *Smart Camera*, uma para fazer a contagem *Pass/Fail* e outra para a contagem *Missed*, para que ambas as contagens voltem a começar do zero.

```

Readincomigdata("exec reset PASSFAILCNT")
Readincomigdata("exec reset MISSTRIGCNT")

Dim p = Readincomigdata("get passcnt")
Pass.Text = p.Substring(0, 1)
Dim f = Readincomigdata("get failcnt")
Fail.Text = f.Substring(0, 1)
Dim m = Readincomigdata("get misstrigcnt")
Missed.Text = m.Substring(0, 1)
Dim t = Readincomigdata("get intercnt")
Total.Text = t.Substring(0, 1)

```

Se o utilizador quiser terminar a conexão entre a *Smart Camera* e a aplicação, clica no botão vermelho. Ao clicar ativa a função **Desconectar_Click** que encerra o *socket* criado aquando do início da ligação através do botão “verde” e cria em seguida um novo *socket* para o caso de o utilizador pretender estabelecer uma nova ligação com os mesmos endereços que estava a usar.

```

Private Sub Desconectar_Click(sender As Object, e
As EventArgs) Handles Desconectar.Click

stream.Close()
client = New System.Net.Sockets.TcpClient()

```

O ecrã secundário do utilizador é o ilustrado na Figura 63, este é o ecrã que encontra quando se clica no botão “Configurar”. Neste ecrã é onde o utilizador insere o endereço IP que está associado à *Smart Camera* e a porta TCP também a ela associada.



Figura 63 Ecrã Config

O ecrã secundário foi desenhado com ajuda de algumas ferramentas existentes no Windows Forms entre elas:

- Os retângulos verdes, correspondem aos botões criados que desempenham diferentes funções, como confirmar (OK) e cancelar;
- Os retângulos azuis, correspondem às caixas de texto/ *Labels* utilizadas para escrever algum texto ou apresentar os valores obtidos;
- O retângulo vermelho, corresponde à *ComboBox* criada, para quando for selecionada mostrar os números das portas TCP disponíveis (20000-20009) e selecionar o que está associado à *Smart Camera* que se quer ligar.

Em seguida explica-se grande parte da programação desenvolvida para a interface gráfica do ecrã secundário.

Primeiramente, foi necessário criar uma função, **OK_Click**, que enviasse o endereço IP e a porta TCP escolhida pelo utilizador para a função que permite estabelecer a ligação com a *Smart Camera*, mais propriamente com a função **Ligar_Click**. Esta vai ler os endereços recebidos e verificar se são os corretos, caso sejam em seguida irá estabelecer a ligação.

Esta função, **OK_Click**, é ativada quando o utilizador clica no botão “OK”, e caso não preencha corretamente um dos campos será enviada uma mensagem a reportar que um dos campos preenchido não está correto.

```
If IP1.Text = "" Or IP2.Text = "" Or IP3.Text =
"" Or IP4.Text = "" Or TCP_port.Text = "" Then

MsgBox("Preencha corretamente o IP Adress ou o
TCP Port !")
Else
Me.Hide()
Câmara.Show()
End If
```

Posteriormente foi adicionada uma outra função com o intuito de fechar o ecrã secundário, questionando se pretende cancelar a configuração da *Smart Camera* e voltar ao ecrã principal sem ter definido nenhum endereço IP ou porta TCP (Figura 61).

```
Dim result As DialogResult = MessageBox.Show("Tem
a certeza que pretende sair? ", "Fechar",
MessageBoxButtons.YesNo, MessageBoxIcon.Question)

If result = Windows.Forms.DialogResult.Yes Then
Me.Hide()
Câmara.Show()
ElseIf result = Windows.Forms.DialogResult.No
Then
Me.Show()
End If
```

5. OUTRO TIPO DE SOLUÇÕES

5.1. VIBRADOR COM SISTEMA DE ORIENTAÇÃO POR PÊNDULO GRAVÍTICO

Um possível sistema a adotar é um sistema desenvolvido por uma equipa de fabricantes de taças de vibradores chinesa. Este sistema usa um pêndulo gravítico para que antes das peças saírem do vibrador estejam com a orientação correta [63].

O pêndulo é colocado a seguir a um degrau (Figura 64), para que, as peças sejam revistas uma-a-uma, fazendo uso das diferenças de diâmetro nas duas extremidades que compõem as peças, ou seja, irá apenas fixar a extremidade com o diâmetro correspondente ao do pêndulo.

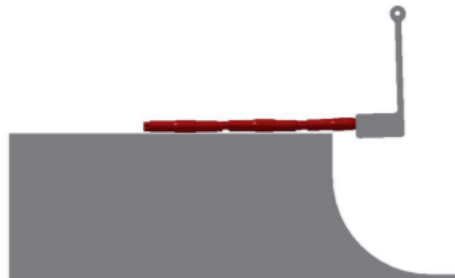


Figura 64 Início do processo [63]

As peças vão subir o vibrador até chegarem perto do pêndulo, Figura 65, estas ao continuar a subir vão se empurrar mutuamente, até que uma das extremidades fique fixa ao pêndulo, se mal orientadas, ou deixadas cair se a peça se encontrar bem orientada, Figura 65 (esquerda) e Figura 65 (direita) respetivamente.



Figura 65 **Rotação de 180 de peça mal orientada (esquerda), Queda de peça bem orientada, sem sofrer mudança de orientação. (direita) [63]**

No entanto, este vibrador não tem nenhum tipo de sistema de escolha de peças não conforme ou de peças que se tenham misturado, podendo ocorrer encravamentos, que levará a cadências limitadas. Este sistema também poderá servir como vibrador para outro tipo de referências de peças aumentando a sua versatilidade e rentabilidade.

5.2. COMPRESSÃO DAS MOLAS

Este sistema de orientação de peças é um sistema mecânico composto por duas molas helicoidais de compressão de igual comprimento, fixas a um calçador, através de um parafuso que serve também de veio de orientação, Figura 66 [63].

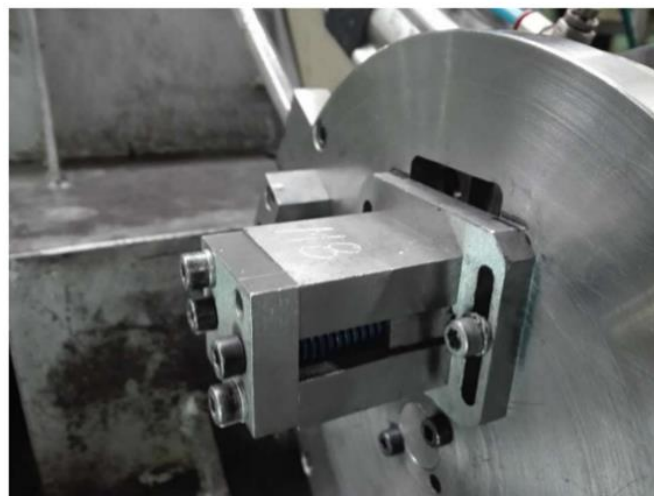


Figura 66 **Sistema de molas à compressão [63]**

Este sistema de orientação, acoplado aos pratos da mesa rotativa, tira partido da diferença de diâmetros das peças. Esta variação de diâmetro vai permitir que as molas sejam comprimidas (Figura 67), se tiverem com a orientação incorreta, ou então não atua se a orientação da peça for a correta.



Figura 67 Sistema de molas à compressão desmontado [63]

As molas ao serem contraídas devido ao contacto com as peças, vão dar origem a uma força normal que se forma durante o contacto das peças com os pratos do sistema. Essas forças vão permitir fixar as peças mal orientadas para não caírem através da força de atrito gerada nos contactos que se formam, Figura 68, e desta maneira assegurar o correto posicionamento das peças ao longo do processo.

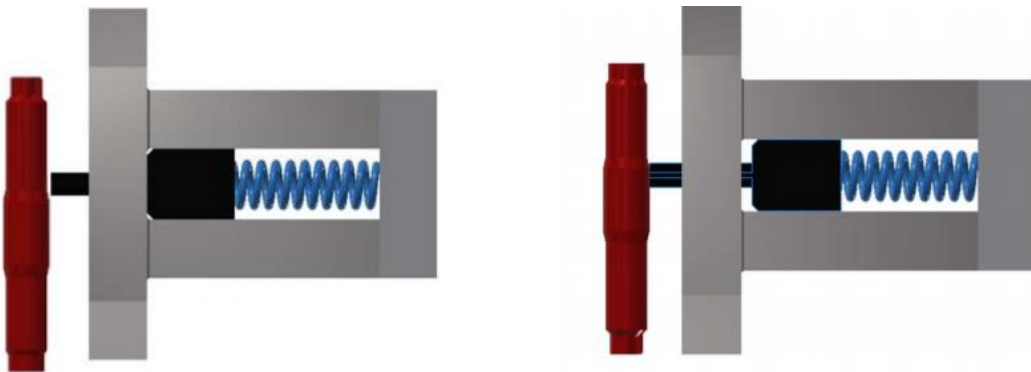


Figura 68 Ação da mola, peça bem orientada (esquerda) e mal orientada (direita) [63]

6. TESTES E RESULTADOS

Neste capítulo será apresentado e analisado o ensaio realizado com equipamento industrial, que foi descrito anteriormente, na linha de produção.

6.1. METODOLOGIA DE TESTE

Este projeto de visão, foi projetado e implementado numa tentativa de conseguir encontrar os defeitos na orientação dos parafusos do qual o projeto partiu. E, por conseguinte, melhorar a qualidade da linha de aparafusamento em questão.

A metodologia de teste teve as seguintes fases:

- Ajuste do posicionamento da câmara em relação ao objeto;
- Calibração da câmara;
- Ajuste dos parâmetros dos programas elaborados;
- Captura de imagens dos diferentes programas;
- Escolha do programa com melhor taxa de resultados.

6.1.1. AJUSTE DO POSICIONAMENTO E CALIBRAÇÃO DA CÂMARA

O ponto de partida para a implementação do projeto foi a escolha do posicionamento da câmera em relação aos parafusos. Isto inclui a distância que fica a câmera dos parafusos e o alinhamento horizontal e vertical da câmera em relação ao centro da calha.

Desta forma, a *Smart Camera* foi posicionada de maneira a conseguir cumprir todos os requisitos, ou seja, ficar o mais centrada possível em relação aos parafusos e conseguir capturar uma imagem do início ao fim da calha (Figura 69).

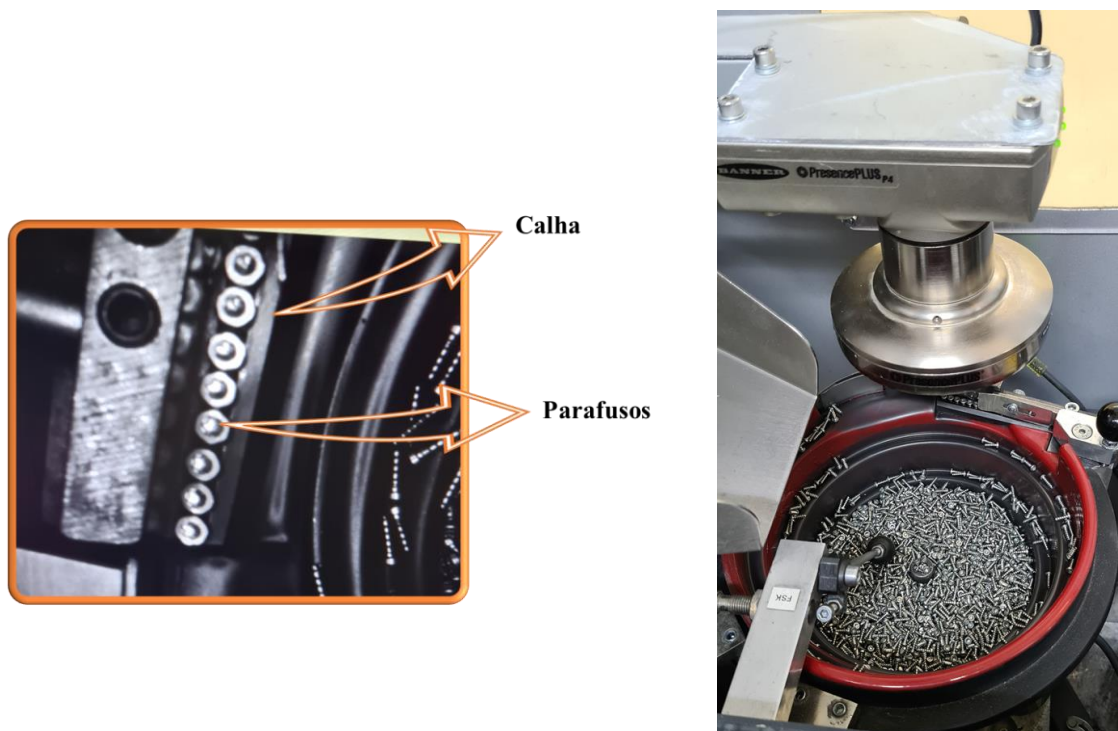


Figura 69 Posicionamento da câmera na linha de produção

6.1.2. AJUSTE DOS PARÂMETROS DOS PROGRAMAS

6.1.2.1. Tempo de exposição e Ganho do sensor

Como já foi dito anteriormente a qualidade da imagem capturada depende do tempo de exposição e do ganho do sensor, em que o ganho do sensor depende dos processamentos feitos à imagem original, enquanto o tempo de exposição altera efetivamente a fotografia capturada.

Visto que, a programação da *Smart Camera* foi feita fora da linha de produção e com condições de iluminação diferentes às que se encontram na linha, foi preciso ajustar estes dois valores, aquando da sua instalação na linha, para que fosse possível obter imagens com qualidade suficiente para posterior análise. Optando por um tempo de exposição de 8,3 ms e um ganho de 2 (Figura 70). Estes valores foram obtidos através da realização de alguns ensaios até se obter a imagem com a qualidade desejada para a *Smart Camera*.

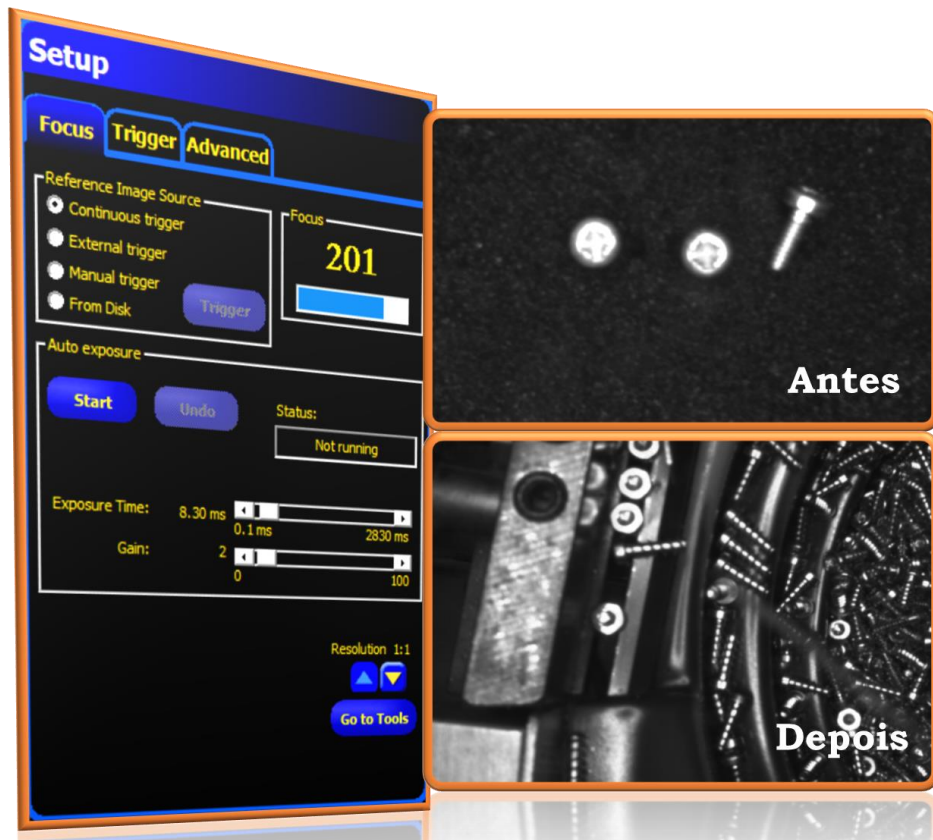


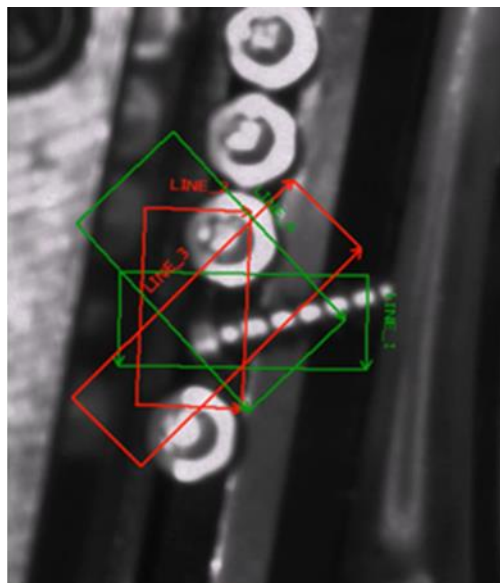
Figura 70 Alteração do tempo de exposição e do ganho do sensor

6.1.2.2. Ajuste das ferramentas de análise

Devido ao fato de se ter programado a câmara num local diferente ao de onde ia ser implementada, teve-se que fazer uns ajustes em ambos os programas (Figura 71). Visto que, as *ROI's* de ambos os programas não estavam nos locais corretos e não davam os resultados pretendidos, daí a necessidade destes pequenos ajustes.



Deteção das cabeça



Deteção do corpo

Figura 71 Ajustes das ferramentas de análise

6.1.3. CAPTURA DE IMAGENS DOS DIFERENTES PROGRAMAS

Posteriormente ao ajuste do posicionamento e dos parâmetros da câmara, deu-se início à aquisição de imagem provenientes de cada um dos programas, com o intuito de no fim dos testes optar-se pelo que obtivesse uma maior taxa de êxito.

Começou-se pelo programa que deteta as cabeças dos parafusos e obteve-se os seguintes resultados (Figura 72).



Figura 72 Resultado obtido pelo programa “Deteção das cabeças”

Em seguida, usou-se o programa que deteta o corpo dos parafusos, ou seja, quando vêm deitados na calha e obteve-se os seguintes resultados (Figura 73).

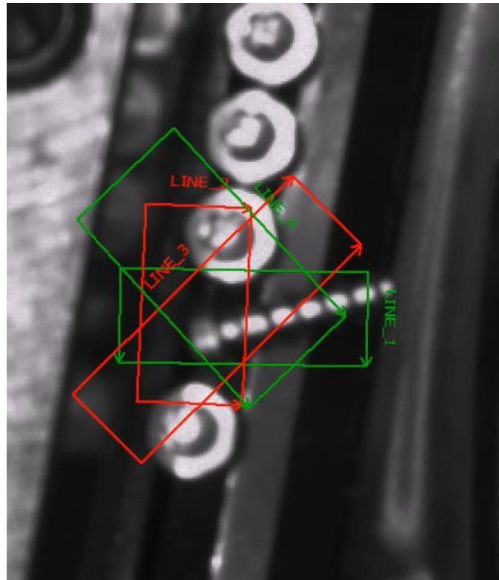


Figura 73 Resultado obtido pelo programa “Detecção do corpo”

6.1.4. ESCOLHA DO PROGRAMA A UTILIZAR

Após vários testes a ambos os programas, chegou-se à conclusão de que o mais indicado para este tipo de projeto, é o programa que deteta as cabeças dos parafusos.

Chegou-se a esta conclusão, visto que este programa era mais consistente em encontrar as cabeças dos parafusos do que o outro programa a detetar o corpo. Isto deve-se ao fato de quando vêm deitados a posição nunca é a mesma e daí existirem problemas na detecção do corpo, enquanto no que deteta as cabeças, como os parafusos vêm sempre na mesma posição consegue detetar muito melhor.

6.2. ALTERAÇÕES EFETUADAS

Posteriormente à escolha do programa a utilizar, foi necessário efetuar algumas alterações de modo a cumprir as normas e condições impostas pela empresa. De tal modo, que se teve de acrescentar novas condições e ferramentas de análise, nomeadamente um detetor de círculo, um de linha e um de blob.

O detetor de círculo é utilizado para ajudar a detecção nas situações em que a câmara não consegue captar o parafuso ou capta entre dois parafusos, metade de cada um

(Figura 74), e com isto se detetar este círculo quererá dizer que está presente um parafuso com orientação correta e não o irá eliminar.

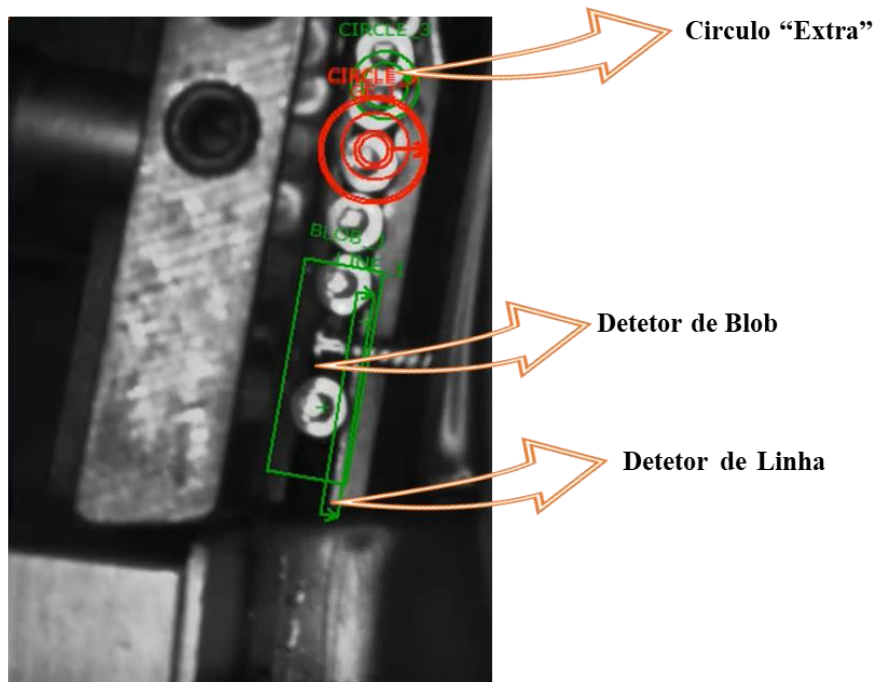


Figura 74 Resultado obtido com e sem o círculo “extra”

O detetor de linha é para detetar a calha vazia ou cheia, dependendo se a linha é detetada ou não, respetivamente. Foi-lhe adicionada uma condição para tal, sempre que o valor do comprimento da linha fosse superior a 135 era dada como calha vazia.

O detetor de blob (buracos/manchas) é para ajudar a ferramenta anterior, e sempre que o valor de blobs detetados fosse inferior a 2 era dado como calha vazia.

Com estas alterações pretendia-se que a câmara fosse capaz de fazer o que está representado na Figura 75.

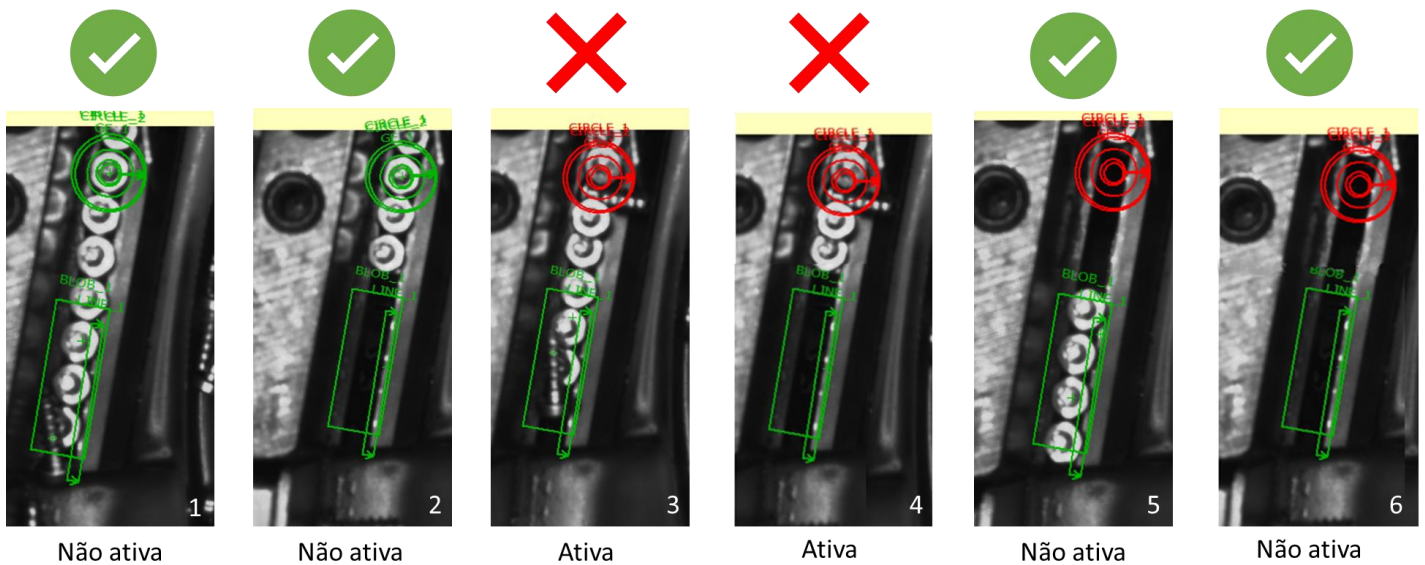


Figura 75 Condições de análise da Smart Camera

Depois de se implementar estas novas alterações, chegou-se à conclusão de que havia duas condições que não podiam ser aplicadas, condição 4 e 5 da Figura 75, visto que a câmara só possui uma ferramenta de teste onde só utiliza os operadores lógicos AND e OR.

Estas duas condições iriam servir para que o expulsador de parafusos não fosse ativado quando a calha estivesse totalmente vazia ou a linha ter sido parada e, no local onde se deteta os parafusos não ter lá nenhum.

Dado o caso, optou-se por inserir as condições 1, 2, 3 e 6 e descartar as condições 4 e 5. Isto devido à pouca ocorrência destas duas condições, 4 e 5, durante os testes efetuados, visto que os parafusos ao deslizarem pela calha poucas vezes ficam numa destas situações (Figura 75).

6.3. RESULTADOS OBTIDOS

Depois de se ter efetuado as alterações expostas anteriormente, iniciou-se a fase de testes com a linha a funcionar em paralelo, com o sensor já existente na linha e com a *Smart Camera*, para se ver se funcionava corretamente e se se obtida os resultados pretendidos.

Os testes foram iniciados com a *Smart Camera* em paralelo com o sensor utilizado pela empresa, para se comparar o resultado obtido por ambos os sistemas, sensor e *Smart Camera*. E obteve-se os resultados da Tabela 4.

Tabela 4 **Comparação dos resultados obtidos por ambos os sistemas**

<i>Método</i>	<i>Sensor</i>	<i>Smart Camera</i>
<i>Nº de ativações</i>	27	63
<i>Ativações Corretas</i>	27	39
<i>Ativações Incorretas</i>	-	“Repetidas” - 12 Mal detetado - 12 Condição 5 - 14

Através da análise da Tabela 4, podemos verificar que a *Smart Camera* foi ativada o dobro das vezes do Sensor. Porém, no parâmetro das ativações corretas o valor obtido pela *Smart Camera* foi de 39, mas entre elas, 12 ativações foram “repetidas”. Ou seja, como o sensor demorou a expulsar os parafusos a *Smart Camera* continuou a tirar fotos a esses parafusos enquanto não eram expulsos e daí ter dado um número maior de ativações que o Sensor.

Todavia, devido à baixa taxa de refrescamento da *Smart Camera* e não se conseguir implementar a condição 5 (Figura 75), não ativar o expulsador quando há parafusos no fim da calha, mas não no início da mesma. Devido à taxa de refrescamento da *Smart Camera* ser baixa não se consegue tirar fotografias muito rápido e, por isso às vezes capta metades de cabeças de parafusos e consequentemente ativa o expulsador quando na verdade não devia.

Contudo, mesmo com estes dois pequenos problemas a linha de aparafusamento ficou a funcionar devidamente e com uma taxa de encravamento muito inferior há que existia antes da implementação da *Smart Camera*, aumentado assim a taxa de produtividade da linha em questão, pois diminuíram os tempos de paragem devido ao encravamento da alimentação do robô.

7. CONCLUSÕES

7.1. CONCLUSÕES

A realização deste projeto envolveu conhecimentos de visão artificial e o modo como esta pode ser utilizada na otimização de processos fabris. Ao longo da realização do projeto foram aparecendo algumas adversidades que o tornaram mais desafiante. Entre elas o fabrico dos suportes que iam ser usados para suportar a *Smart Camera* e o iluminador, o que atrasou a ida para a linha de produção e posteriormente a realização de testes para se verificar se funcionava tudo corretamente como projetado.

Após a fabricação dos suportes estar concluída e a *Smart Camera* instalada na linha de produção deu-se um ponto de viragem no projeto. A partir deste momento já se pode começar a realizar os testes dos programas elaborados, bem como obter resultados e proceder a melhorias, visto que até este momento não se sabia se o projeto funcionava corretamente como o delineado.

Na linha de montagem, na fase de testes deparou-se com a existências de alguns problemas, o que originou algumas alterações na programação da *Smart Camera* para que conseguisse cumprir todos os objetivos propostos pela empresa. Todavia deparou-se que a *Smart Camera* apresentava algumas limitações, o que impossibilitou de se obter o melhor

resultado possível. As limitações encontradas foram a baixa taxa de refrescamento e só a possibilidade de usar operações lógicas de AND e OR.

Apesar das limitações detetadas, o projeto ficou a funcionar corretamente e de acordo com a ideia inicial, ou seja, expulsar os parafusos mal orientados e deixar passar os bem orientados para o robô, para este proceder ao aparafusamento das antenas.

Em suma, os resultados obtidos após a instalação da *Smart Camera* na linha de produção foram muito satisfatórios. De uma forma geral, todos os objetivos do trabalho foram atingidos, contribuindo claramente para adquirir novos conhecimentos, consolidar outros já adquiridos, e alcançar noções que de outra forma só seriam alcançadas em contexto extraescolar. Todos esses factos fizeram com que este trabalho se tornasse extremamente enriquecedor para a minha formação.

7.2. PROPOSTA DE TRABALHOS FUTUROS

Trabalhos que poderão ser desenvolvidos para futuro, poderia passar pela otimização do equipamento, no sentido de ter uma resposta mais rápida, a nível de velocidade sem pôr em causa os requisitos do cliente, conseguindo com isto um aumento da taxa de refrescamento, ou seja, aumentando o valor dos fps . A ideia poderia passar pela alteração da *Smart Camera* por outra de melhor resposta existente no mercado.

Uma alternativa seria uma *Smart Camera* da Cognex, mais propriamente a *In-Sight* 7600. Pois apresenta uma resolução de imagem igual à utilizada, mas apresenta uma taxa de 217 *fps* (frames por segundo), também temos outra da marca Allied Vison, do modelo Manta que apresenta uma taxa de até 286 *fps* ou então da marca Teledyne DALSA e com o modelo Falcon4-CLHS M4400 apresenta uma taxa de 330 *fps*. Todas elas superiores à utilizada que só apresenta uma taxa de 48 *fps*.

O pensamento será sempre a melhoria contínua, no sentido de a empresa tirar o máximo de rentabilidade do equipamento nem que implique investimento, com o objetivo de aumentar a produtividade das linhas e conseqüentemente aumentar o lucro.

Referências Bibliográficas

- [1] “História da empresa” <https://www.continental.com/pt-pt/empresa/hist%C3%B3ria>.
Acedido: 2021-3-20. [citado na p. 3, 4]
- [2] “Computer vision for quality control.” <https://softengi.com/blog/computer-vision-for-quality-control/>. Acedido: 2020-11-20. [citado na p. 9, 10, 19]
- [3] Z. A. Jaffery and A. K. Dubey, “Scope and prospects of non-invasive visual inspection systems for industrial applications” *Indian Journal of Science and Technology* 9(4):1-11, 2016
- [4] J. D. A. Martins, “Visão artificial na inspeção e caracterização de defeitos superficiais em peças fundidas.”, Tese do Mestrado Integrado em Engenharia Mecânica, Faculdade de Engenharia da Universidade Do Porto, pp. 3-4, 2018
- [5] “A visão artificial na indústria 4.0.” <https://blog.infaimon.com/pt/visao-artificial-industria-4-0/>. Acedido: 2020-11-22. [citado na p. 10]
- [6] “A importância do sistema de visão artificial na automação industrial.” <https://www.rtiautomacao.com.br/importancia-do-sistema-de-visao-artificial-na-automacao-industrial/>. Acedido: 2020-11-22. [citado na p. 10]
- [7] O que é visão artificial e onde essa tecnologia pode ser aplicada na indústria e manufatura? <https://ilk.com.br/o-que-e-visao-artificial-e-onde-essa-tecnologia-pode-ser-aplicada-na-industria-e-manufatura/>. Acedido: 2020-11-22. [citado na p. 10]
- [8] “Sistemas de Visão Industrial” https://paginas.fe.up.pt/~ee97031/relatorio_de_pesquisa.pdf. Acedido: 2020-11-22. [citado na p. 10, 29, 30]
- [9] “Banner vision sensors: Ivu series.” <https://www.bachofen.ch/en-us/banner-vision-sensors-ivu-series-at030>. Acedido: 2020-11-24. [citado na p. 11]

- [10] "Presenceplus p4." <https://www.bannerengineering.com/br/pt/products/machine-vision/smart-cameras/presenceplus-p4.html>. Acedido: 2020-11-24. [citado na p. 11, 44]
- [11] "Sensor industrial." <https://ballast.com.br/sistemas-de-visao-industrial-funcionamento-e-aplicacao/> Acedido: 2020-11-24. [citado na p. 11]
- [12] "Sensor de visão: Você conhece?" <https://www.citisystems.com.br/sensor-de-visao/>. Acedido: 2020-11-26. [citado na p. 11, 12]
- [13] "Sensores de visão." <https://www.infaimon.com/pt-pt/enciclopedia-de-la-visao/sensores-de-visao/>. Acedido: 2020-11-29. [citado na p. 12]
- [14] "Sistemas de visão artificial: tipos e aplicações." <https://blog.infaimon.com/pt/sistemas-de-visao-artificial-tipos-aplicacoes/>. Acedido: 2020-11-29. [citado na p. 12, 13]
- [15] "Câmaras inteligentes." <https://www.infaimon.com/pt-pt/categoria-producto/sistemas-visao-integrados-pt/camaras-inteligentes-pt-sistemas-visao-integrados-pt/>. Acedido:2020-11-29. [citado na p. 12]
- [16] I. A. A. Ribeiro, "Controlo de qualidade de embalagens usando visão industrial", Tese do Mestrado em Engenharia de Automação industrial, Universidade de Aveiro, pp:10-82, 2010.
- [17] "Sistemas visão integrados." <https://www.infaimon.com/pt-pt/categoria-producto/sistemas-visao-integrados-pt/>. Acedido: 2020-11-29. [citado na p. 13]
- [18] "Sistemas de visão para uma produção moderna." <https://www.balluff.com/local/br/industries-and-solutions/solutions-and-technologies/machine-vision/>. Acedido: 2020-11-29. [citado na p. 13]
- [19] "Visão artificial industrial: Aplicações e setores." <https://blog.infaimon.com/pt/visao-artificial-industrial-aplicacoes-setores/>. Acedido: 2020-11-29. [citado na p. 15, 20]
- [20] "Como a visão computacional está revolucionando a gestão da qualidade na indústria." <https://www.industria40.ind.br/artigo/18753-como-a-visao->

computacional-esta-revolucionando-a-gestao-de-qualidade-na-industria. Acedido: 2020-11-29. [citado na p. 15, 20]

- [21] “Visão artificial - um método muito eficaz no controlo de qualidade das células e painéis solares.” <https://blog.infaimon.com/pt/visao-artificial-controlo-qualidade-celulas-paineis-solares/>. Acedido:2020-11-30. [citado na p. 15]
- [22] “Soluções de visão artificial para alimentação e embalagens.” <https://www.infaimon.com/pt-pt/publicaciones/solucoes-de-visao-artificial-para-alimentacao-e-embalagens/>. Acedido:2020-11-30. [citado na p. 15, 16, 17]
- [23] “Automatic quality control beamweld.” <https://neadvance.com/automatic-quality-control/beamweld/>. Acedido: 2021-11-30. [citado na p. 16]
- [24] Infaimon, “Soluções de câmara única e de múltiplas câmaras para aplicações de visão artificial automatizada,” [citado na p. 17]
- [25] “O que é deep learning e quais são suas aplicações na indústria.” <https://toshyro.com.br/deep-learning-na-industria/>. Acedido: 2020-12-10. [citado na p. 18]
- [26] “Deep learning: O que a indústria ganha com essa tecnologia?”<https://www.pollux.com.br/blog/deep-learning-o-que-a-industria-ganha-com-essa-tecnologia/>. Acedido:2020-12-10. [citado na p. 18]
- [27] “Controle de qualidade: o que é e qual a importância de realizar na sua empresa?”<https://www.voitto.com.br/blog/artigo/controle-de-qualidade>. Acedido:2020-12-10. [citado na p. 19]
- [28] E. A. S. N. Luana Nascimento de Paula, Adriano Rosa Alves, “A importância do controle de qualidade em indústria do segmento alimentício,” vol. 2, 2017. [citado na p. 19]
- [29] “Processo de fabricação: O controlo de qualidade é imprescindível.” <https://blog.infaimon.com/pt/processo-fabricacao/>. Acedido: 2020-12-10. [citado na p. 20]

- [30] “Conhecendo lentes.” <http://www.tucanobrasil.com.br/oritec/lentes/lentes.htm>.
Acedido: 2021-1-10. [citado na p. 23]
- [31] M. L. R. F. D. A. X. P. MARCELO RICARDO STEMMER, ALEXANDRE ORTH,
“Apostila de sistemas de visão,” 2005. [citado na p. 23]
- [32] “Abertura relativa.” <https://www.infaimon.com/pt-pt/enciclopedia-de-la-visao/abertura-relativa/>. Acedido: 2021-1-10. [citado na p. 23]
- [33] “Fotografia para iniciantes - exposição parte 1 de 4: Abertura.”
<https://alugueumfotografo.wordpress.com/tag/abertura/>. Acedido: 2021-1-10. [citado na p. 24]
- [34] “Foco e profundidade de campo.” <http://arte-digital.org/fotografia/>. Acedido:2021-1-10. [citado na p. 24]
- [35] “Aberrações Ópticas.” <https://www.infoescola.com/fisica/aberracoes-opticas/>.
Acedido: 2021-1-11. [citado na p. 25]
- [36] “Tipos de lentes fotográficas.” <https://www.infoescola.com/fotografia/tipos-de-lentes-fotograficas/>. Acedido: 2021-1-15. [citado na p. 26]
- [37] “Tipos de lentes fotográficas, suas funções e aplicações.”
<http://cameraneon.com/acessorios/tipos-de-lentes-fotograficas-suas-funcoes-e-aplicacoes/>. Acedido: 2021-1-15. [citado na p. 26]
- [38] “Entendendo o foco e distância focal.”
<https://diegorodrigophotography.wordpress.com/2014/11/25/foco-e-distancia-focal/>.
Acedido: 2021-1-15. [citado na p. 26]
- [39] “Deve utilizar um filtro polarizador.” <https://www.ipf.pt/site/deve-utilizar-um-filtro-polarizador/>. Acedido: 2021-1-18. [citado na p. 27]
- [40] “Filtros.” <https://www.infaimon.com/pt-pt/categoria-producto/oticaspt/filtros-pt/>.
Acedido: 2021-1-17. [citado na p. 27]
- [41] “Capturando imagens - Filtros Óticos.” <https://industria4-0.com/capturando-imagens-filtros-oticos-parte-1/>. Acedido: 2021-2-17. [citado na p. 27, 28]

- [42] “Aprenda sobre a função e a importância dos filtros uv-a e por que a astm e3022 exige que todas as luminárias led uv os tenham.” <https://magnaflux.com.br/br/blog/porque-astm3022-requer-filtro-luminaria-ultravioleta/>. Acedido: 2021-1-17. [citado na p. 28]
- [43] “Practical guide to machine vision.” <https://machinevisiondirect.com/pages/practical-guide-to-machine-vision>. Acedido: 2021-1-17. [citado na p. 29]
- [44] “Tipo de Iluminação” <https://www.infaimon.com/pt-pt/enciclopedia-de-la-visao/tipos-de-iluminacao>. Acedido: 2021-1-17. [citado na p. 29]
- [45] “Iluminação por fluorescente” <https://www.infaimon.com/pt-pt/enciclopedia-de-la-visao/iluminacao-por-fluorescente/>. Acedido: 2021-1-17. [citado na p. 30]
- [46] D. Martin, “Basic lighting techniques for machine vision,” [citado na p. 30]
- [47] “Técnicas de iluminação no setor industrial” <https://blog.infaimon.com/pt/tecnicas-de-iluminacao-setor-industrial/>. Acedido: 2021-1-24. [citado na p. 31]
- [48] “ILUMINAÇÃO DE VISÃO INDUSTRIAL” <https://www.cognex.com/pt-pt/what-is/machine-vision/components/lighting>. Acedido: 2021-1-24. [citado na p. 31]
- [49] “Técnicas de Iluminação” <https://www.infaimon.com/pt-pt/enciclopedia-de-la-visao/tecnicas-de-iluminacao/>. Acedido: 2021-1-24. [citado na p. 31, 32, 33]
- [50] M. L. L. Reiss, “O potencial da fotogrametria na ufrgs,” vol. 2, 2008. [citado na p. 34]
- [51] “Common Illumination Types” <https://www.edmundoptics.com/knowledge-center/application-notes/illumination/choose-the-correct-illumination/>. Acedido: 2021-1-24. [citado na p. 34, 35]
- [52] “Como funciona uma câmera digital?” <https://www.oficinadanet.com.br/post/12125-como-funciona-uma-camera-digital>. Acedido: 2021-1-31. [citado na p. 36]
- [53] “Que câmara de visão industrial escolher?” <http://guide.directindustry.com/pt/que-camara-de-visao-industrial-escolher/>. Acedido: 2021-1-31. [citado na p. 36, 37]

- [54] “Tecnologia cameras.” <https://www.infaimon.com/pt-pt/enciclopedia-de-la-vision/Tecnologia-cameras>. Acedido: 2021-1-31. [citado na p. 37]
- [55] “Os tamanhos dos sensores de imagem.” <https://blog.emania.com.br/os-tamanhos-dos-sensores-de-imagem/>. Acedido: 2021-1-31. [citado na p. 38]
- [56] COGNEX, “Introdução ao sistema de visão,” 2018. [citado na p. 39]
- [57] “Tecnologia frame grabbers.” <https://www.infaimon.com/enciclopedia-de-la-vision/tecnologia-frame-grabbers/>. Acedido: 2021-2-7. [citado na p. 39]
- [58] “4-input frame grabber.” <http://www.sensoray.com/products/311.htm>. Acedido: 2021-2-7. [citado na p. 39]
- [59] “Presenceplus p4 sealed omni.” <https://info.bannerengineering.com/cs/groups/public/documents/literature/145259.pdf>. Acedido: 2021-2-14. [citado na p. 45, 46]
- [60] “M12 to rj45 ethernet cable.” <http://www.fleconn-china.com/productinfo-153-M12-to-RJ45-Ethernet-cable,-Cat-6A-Shielded-X-coded-8-Pole-Connector.html>. Acedido: 2021-2-14. [citado na p. 46]
- [61] “Mp2 machine vision series m0814-mp2.” <https://computar.com/product/552/M0814-MP2>. Acedido: 2020-2-14. [citado na p. 47]
- [62] “Ledrr90s-p.” <https://www.bannerengineering.com/br/pt/products/part.75892.html>. Acedido: 2021-2-14. [citado na p. 47]
- [63] F. M. S. Pereira, “Alimentação automática de máquinas de roscar parafusos” Tese de Mestrado em Engenharia Mecânica na Especialidade de Produção e Projeto, Universidade de Coimbra, pp:38-55, 2018.

*Anexo A. Inicialização da Smart Camera Banner
PresencePlus_{P4} OMNI*

PresencePLUS[®] P4

Models included:

- GEO/GEO 1.3
- EDGE/EDGE 1.3
- BCR/BCR 1.3
- AREA/AREA 1.3
- OMNI/OMNI 1.3

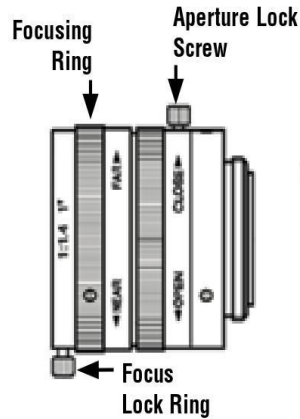


BANNER[®]

more sensors, more solutions

step 1

1. Install the lens (and filters, if used).
For EDGE, GEO, BCR, AREA, and OMNI any lens may be used. For EDGE 1.3, GEO 1.3, AREA 1.3, BCR 1.3, and OMNI 1.3 use only Megapixel lenses. For non-Banner lenses, follow the lens manufacturer's unpacking and installation instructions.



NOTE: If the lens has a focus lock screw, loosen the screw before focusing the lens. Some lenses also have an aperture lock screw.

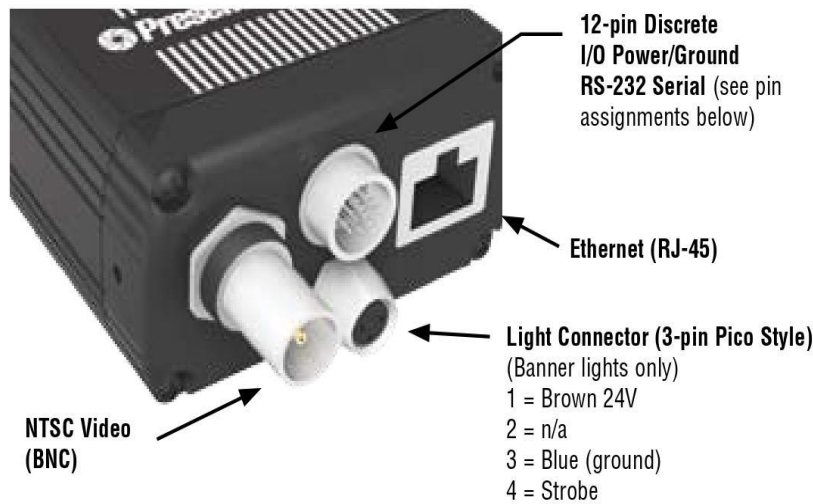
Cable Connections

1. If a light will be powered by the Sensor, connect it to the Light connector.

step 2

- a. Caution: If the light is powered by the Sensor, the Sensor power source must be 24V dc.
- b. Caution: This connection is for Banner lights only!
2. If an NTSC monitor is used, connect it to the Sensor via a BNC-to-BNC cable to the NTSC Video connector.
3. Connect the Ethernet cable from your PC to the Sensor at the RJ-45 connector.
 - a. If connecting directly from the PC, use a crossover cable (such as Banner Model No. STPX07).
 - b. If connecting the Sensor to a hub or router, use straight cables (such as Banner Model No. STP07) to the Sensor as well as to the PC.
4. Push the connector end of the supplied 12-wire cable onto the 12-pin connector on the Sensor.

5. Connect the leads on the 12-wire cable to the appropriate locations (see pin assignments below).



12-Wire Cable Pin Assignments

Pin #	Wire Color	Description	Direction
1	Yellow	RS-232 TX	Output
2	Gray	Remote Teach	Input
3	Orange	Product Change	Input
4	Pink	External Trigger	Input
5	Black	Discrete I/O #1	In/Out
6	Red	Discrete I/O #2	In/Out
7	White	Discrete I/O #3	In/Out
8	Light Blue	Discrete I/O #4	In/Out
9	Violet	RS-232 RX	Input
10	Green	RS-232 Signal Ground	Output
11	Blue	Common (Signal Ground)	Input
12	Brown	10-30V dc	Input

12-Wire Cable Models

- P4C06** — 2 m (6')
- P4C23** — 7 m (23')
- P4C32** — 10 m (32')
- P4C50** — 16 m (50')
- P4C75** — 23 m (75')

Crossover Ethernet Cable (to PC Ethernet Port)

- STPX07** — 2.1 m (7')
- STPX25** — 7.6 m (25')

or

Standard Ethernet Cable (to PC via Network Hub or Switch)

- STP07** — 2.1 m (7')
- STP25** — 7.6 m (25')

Monitor Cable (to Video Monitor, optional)

- BNC06** — 2 m (6')
- BNC15** — 5 m (15')
- BNC30** — 9 m (30')

TIPS

The trigger device can be any 10-30V dc photoelectric sensor, or a device with a similar output.

step 3

PC Configuration

1. Ethernet communication, configure the IP address of your computer.

Typical IP Address:
192.168.0.2



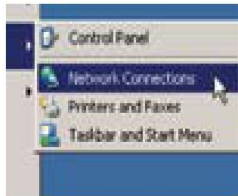
Sensor

Default IP Address
192.168.0.1

Write down the existing address of your PC before changing it: ____ . ____ . ____ . ____

To change the IP address of your PC, do the following:

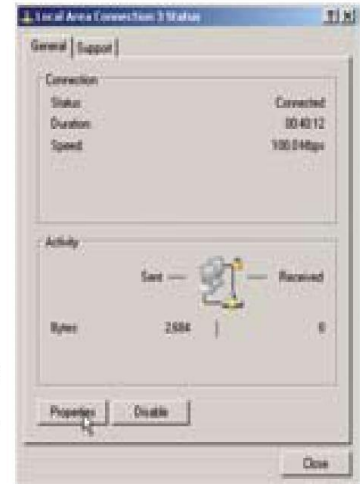
The following screen captures are from Windows XP. For earlier Windows versions, see Banner's Supplemental Information at http://info.bannersalesforce.com/xpedio/groups/public/documents/trainingjobaid/vr_01_00_e.pdf.pdf



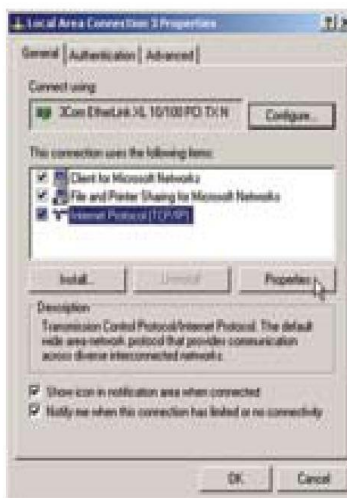
- a. Select **Start** > **Settings** > **Network Connections**.



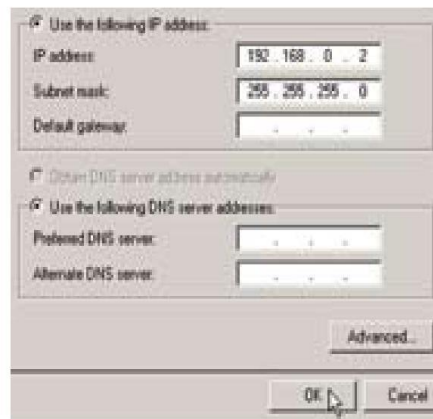
- b. Double-click the Local Area Connection used by *PresencePLUS*.



- c. Click the **Properties** button.



- d. Highlight **Internet Protocol TCP/IP**, and click the **Properties** button.



- e. Choose **Use the following IP address**, and:
 - Change the IP address to 192.168.0.2
 - Change the Subnet mask to 255.255.255.0
 - Click the **OK** button.

2. Install the *PresencePLUS* software.
 - a. Insert the Installation CD.
 - b. Click **Install PresencePLUS PC Software**.



step 4

Starting the *PresencePLUS P4*

1. Power up the hardware and verify that the Power/Error light turns Green. This may take up to 20 seconds.
2. Verify that the yellow LED on the Ethernet port is ON. If it is not ON, see Cable Connections on page 4.



step 5

Launching Software

1. Start the *PresencePLUS* program by clicking **Start > Program Files > PresencePLUS**.
2. At start-up, *PresencePLUS* will try to communicate with the Sensor.

If communication with the Sensor is successful, the application will launch and display the Setup or Run screen. If communication was not successful:

 - Verify that the Ethernet cable is the correct type (see Cable Connections on page 4).
 - Verify that the TCP/IP settings are correct (see PC Configuration on page 6).
3. If using an optional NTSC video monitor, verify that the monitor is displaying an image. You may not see an image until the camera is given the first trigger.
4. When the software launches, create an inspection, configure the discrete I/O, and begin running inspections.

NOTE: Initially, all discrete I/O are configured as inputs. Go to the System window to change the discrete I/O. For detailed configuration information, refer to the User's Manual (under the **Help** button in the GUI; see back cover for part number).


step 6

Software Setup

Use the Main Menu toolbar to navigate the *PresencePLUS P4* options. Proceeding from left to right, the buttons in the Menu toolbar step through the process of creating an inspection file.




Inspection Work Flow

1.  Set up the Sensor, lens, and lighting, to acquire a reference image.
 - a. Set up the Sensor lens and lighting.
 - b. Choose Trigger option **Continuous** for a live image.
 - c. Click **Auto Exposure** to adjust the brightness.
 - d. Focus the Sensor lens by turning the lens until the Focus Number is maximized.

NOTE: While still in the Setup screen, verify that the trigger works by selecting **External** in the Trigger Options. When in Run mode, the Sensor uses only the external trigger.

 - e. When you have the desired image, click **Next** to proceed to the Tools screen to acquire the reference image.

2.  Add tools to the inspection. Build the tools from scratch or add tools from a previous inspection file saved on the PC or the Sensor. To add a Vision tool, click the Tool button. To remove a tool, click the "X" in the lower left corner of the screen.
 - a. Add **Location tool(s)** to find the target to adjust the following Regions of Interest (ROI) for transitional and rotational changes.
 - Required** b. Add **Vision or Bar Code tool(s)** to inspect the part.
 - c. Add **Measure tool(s)** to create distance measurements from points found.
 - Required** d. Add **Test tool(s)** to set the Pass/Fail criteria. (The Vision, Bar Code, and Measure tools are inputs to the Test tool.)

NOTES:

 - Click **Quick Teach** to automatically set all the selected parameters in the Test tool and proceed to the Run screen, or click **Next** to proceed to the Teach screen, to teach a sample set of good products.
 - To manually set min/max parameters in a Test tool, skip **Teach** and go directly to **Run**.

TIPS

Before creating an inspection file, set up the electrical configuration of the external trigger. (Click **System** button, select **Trigger** tab.)

Tool Options

NOTE: Tool availability depends on P4 model; see back page.

Tool Name		Function	Description
Location Tools	GEO Find	Translation and rotation	Locates the target by searching for a taught pattern and compensates for translation and rotation.
	Locate	Translation and rotation	Finds the edge of the part and compensates for translation and rotation (if selected).
Vision Tools	GEO Count	Finds one or more patterns	Locates and counts a taught pattern.
	Edge	Counts and locates edges	Detects and counts transitions between bright and dark pixels. The total number of edges can be counted, and the position of each edge can be found.
	Object	Locates and counts objects, determines midpoints, and measures widths	Detects the edges of dark and bright objects, locates their midpoints, counts dark and bright objects, and measures the width of each dark and bright object.
	Average Gray Scale	Determines presence, absence, and shades of gray	Determines average gray scale value in the Region of Interest (ROI)
	BLOB	Counts/measures areas and counts, sizes, and locates objects	Detects groups of connected light or dark pixels within the ROI and designates them as BLOBs. After BLOBs are found, they can be counted, sized, and located.
Bar Code Reader	Bar Code	Decodes bar code markings	Finds and decodes DataMatrix, PDF-417, and Linear bar code types in user-selectable decoder modes, color schemes, and viewer schemes.
Analysis Tools	Measure	Measures between points	Measures distance between two prescribed points.
	Test	Logic input/output	Evaluates results of selected Vision and Analysis tools to determine whether an inspection passes or fails. It also performs logical operations and activates outputs.

TIPS

- Each inspection must contain at least one Vision tool and one Test tool.
- Save a backup copy of your inspection to the host PC.

3.



This screen automatically configures the parameters chosen in the Tools screen.

- a. Choose the sample size
- b. Click **Start**
- c. Trigger the controller with the external trigger device
- d. Click **Stop**
- e. Click **Next** to proceed to **Run**

Before entering **Run**, save the inspection file to one of the 12 memory locations on the Sensor.

4.



Select an inspection to run, and view the results of the inspection.

To select an inspection, (in the Select tab) enable **Software Override** and select the inspection file from the list of stored inspections.

Alternate method: Use **Hardware Input** to select an inspection via Product Change and Product Select lines.

Viewing Results

Display Options

Next Pass	Display only the next passing inspection.
Next Fail	Display only the next failing inspection.
Next RT	Display the next remote teach.
Next RT Fail	Display the next unsuccessful remote teach.
Next	Continuously display inspections.
None	Don't display any inspections.

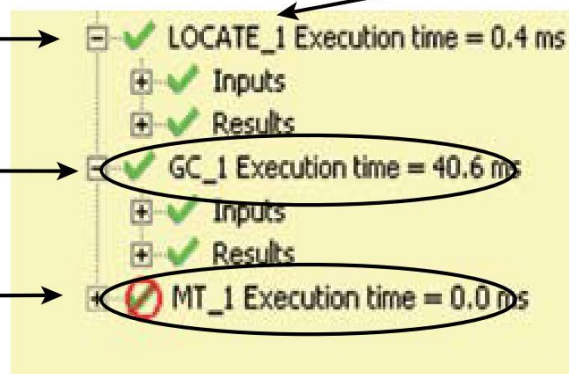
TIPS

When using the Hardware input, pulse the **Product Change** and **Product Select** inputs to initiate an inspection change. See the User's Manual in the Help menu for complete information.

Click + to expand and - to contract category

Passing Tool

Failing Tool



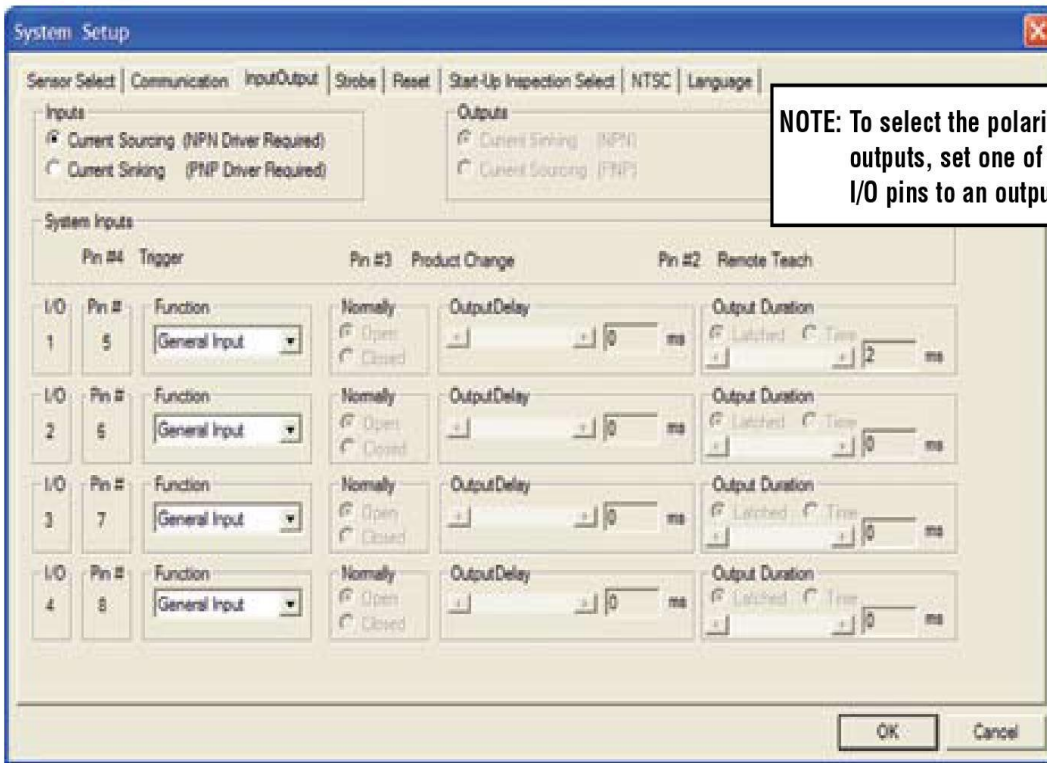
Click on tool name to show the ROI

To begin inspecting, click the **Start** button in the Run screen.

step 7

System Setup

Use the System Setup screen to change discrete I/O, the communication port, the product change, the strobe output (for external lighting control), the trigger input, and to view diagnostic information.



NOTE: To select the polarity of the outputs, set one of the four I/O pins to an output.

InputOutput Configuration Tab



Save inspections to the Sensor or to a folder on the PC.



Provides: Operation Help
Installation Help
Lens Selection Help
Lighting Help
Error Codes
About (GUI Software version, Sensor Firmware version, Hardware version, Sensor's physical address)

TIPS
Use **Help > About** in the GUI to look up the software, firmware, and hardware versions. The information will be helpful if you call Banner for support.

*Anexo B. Especificações da Smart Camera Banner
PresencePlus_{P4} OMNI*

PresencePLUS P4 Sealed OMNI



OMNI, OMNI 1.3, and Color OMNI



Optional Lens Enclosure shown mounted on camera

- Ruggedized housing and connectors
- Washdown rating of IEC IP68
- Available for standard, high-resolution 1.3 megapixel, and Color OMNI
- Supports a wide range of lighting
- Supports standard C-mount lenses
- Choice of lens enclosure or integrated IP68 Ring Light LED..R90S Series
- Supports Banner Engineering's standard *PresencePLUS* user interface, image processing, and communications

Models

Model	Description
P4ORS	P4 Omni Right Angle—IP68 640 × 480 gray scale
P4O1.3RS	P4 Omni 1.3 Megapixel Right Angle—IP68 1280 × 1024 gray scale
P4CORS	P4 Color Omni Right Angle—IP68 752 × 480 color and gray scale



NOTE: Models require a Lens Enclosure or Integrated Ring Light, which are available separately.

Specifications

Power

Voltage: 10 to 30V dc (24V dc \pm 10% if a light source is powered by the Sensor)

Current:

P4 OMNI: 650 mA maximum (exclusive of I/O load and lights)

P4 Color OMNI: 550 mA maximum (exclusive of I/O load and lights)

P4 OMNI 1.3: 550 mA maximum (exclusive of I/O load and lights)

Discrete I/O

- 1 Trigger IN
- 1 Strobe OUT
- 4 Programmable I/O
- 1 Product Change
- 1 Remote Teach

Input/Output Configuration

NPN or PNP software selectable

Output Rating

150 mA

ON-State Saturation Voltage: <1V at 150 mA max.

NpN

OFF-State Leakage Current: <100 microamps NPN or PNP

Exposure Time

P4 OMNI: 0.1 to 2830 ms

P4 OMNI 1.3: 0.1 to 1670 ms

P4 Color OMNI: 0.1 to 1000 ms

Imager

P4 OMNI: 4.736 × 3.552 mm, 5.900 mm diagonal (1/3 inch CCD)

P4 OMNI 1.3: 8.576 × 6.861 mm, 10.9829 mm diagonal (2/3 inch CMOS)

P4 Color OMNI: 4.5 × 2.9 mm, 5.4 mm diagonal (1/3 inch CMOS)

Lens Mount

C-mount

Pixel Size

P4 OMNI: 7.4 × 7.4 microns

P4 OMNI 1.3: 6.7 × 6.7 microns

P4 Color OMNI: 6.0 × 6.0 microns

Construction

Die-cast nickel-plated aluminum sensor housing, glass or acrylic window

Weight: Approximately 0.43 kg (0.95 lb)

Environmental Rating

IEC IP68



Communication

10/100 Ethernet connection for running PresencePLUS P4 software and/or output inspection results. M12, 8-pin, female connector
RS-232 flying leads

Display Options

PC or NTSC video; 9 m (30 ft) max. cable length

Memory

INS Storage Memory: 32 MB
Number of INS Files: 999

Operating Conditions

Stable Ambient Temperature: 0° to +50° C (+32° to +122° F)

Stable Ambient Lighting: No large, quick changes in light level; no direct or reflected sunlight

Certifications

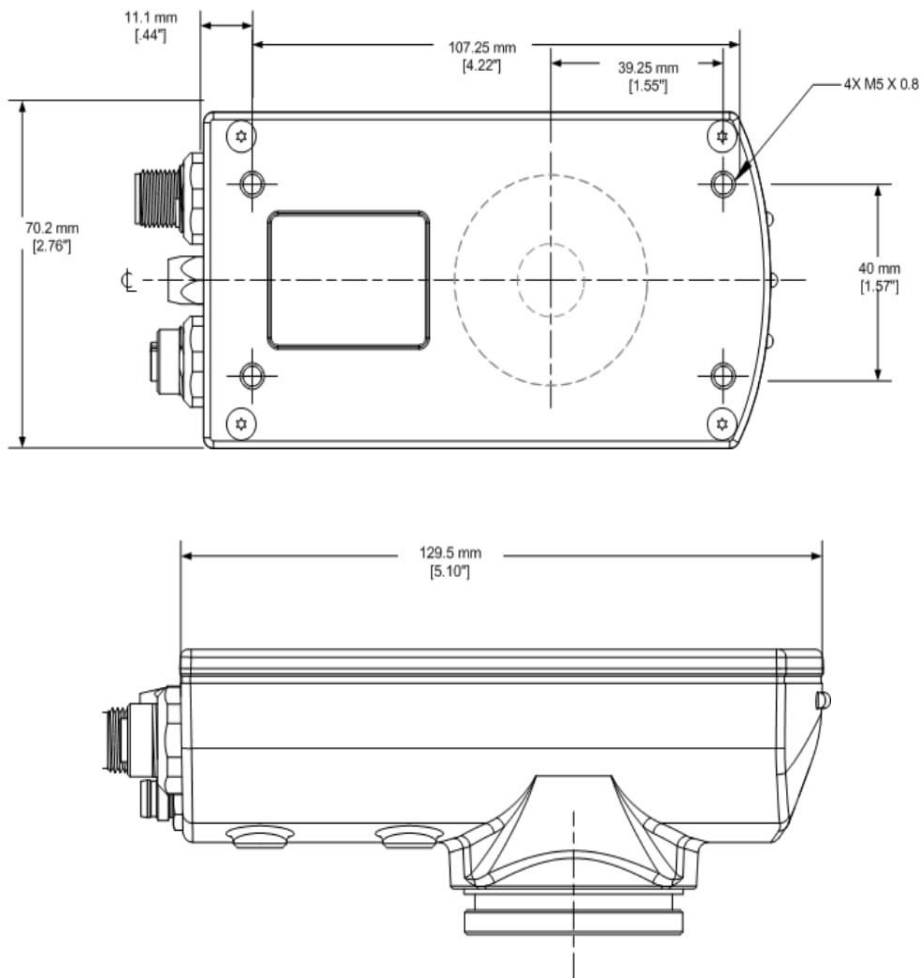


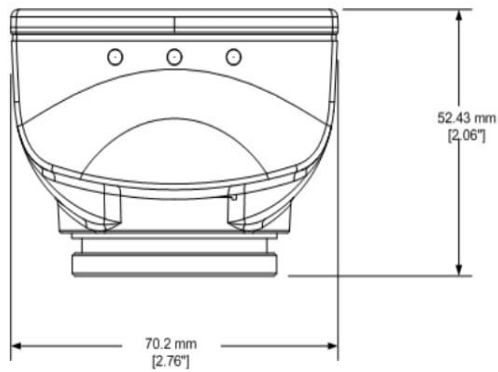
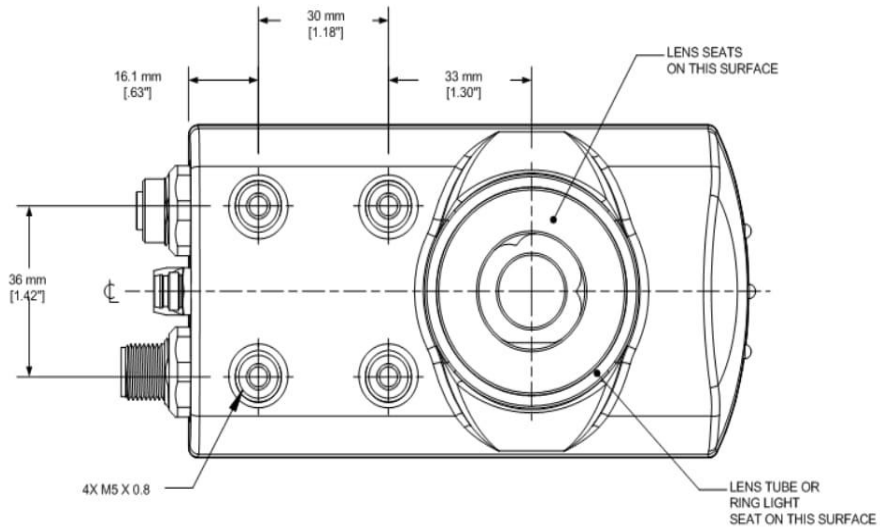
Acquisition

Levels of Gray Scale: 256
Color OMNI: 256 Red, Green, and Blue

Camera	Frames Per Second	Image Size (pixels)
P4 OMNI	48 max.	640 × 480
P4 OMNI 1.3	26.8 max.	1280 × 1024
P4 Color OMNI	17 max.	752 × 480

Dimensions





Cable Connections

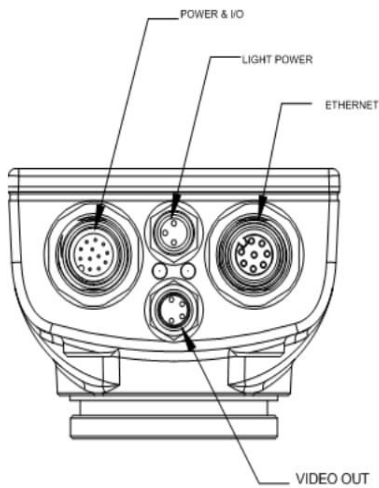


Figure 1. Cable Connections

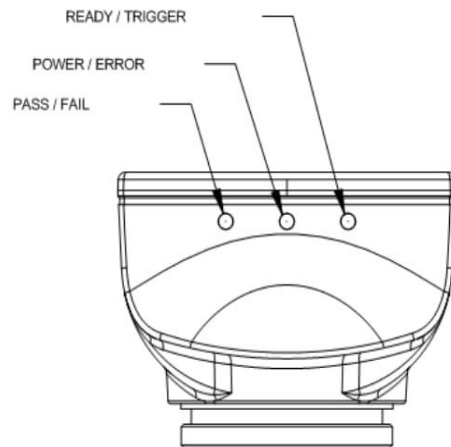


Figure 2. LED Indicators

Power and I/O Pinouts

Pin #	Wire Color	Description	Direction
1	White	Discrete I/O #3	In/Out
2	Brown	+10 to 30V dc	Input

Pin #	Wire Color	Description	Direction
3	Green	RS-232 Ground ¹	Output
4	Yellow	RS-232 Transmit ¹	Output
5	Gray	Remote Teach	Input
6	Pink	Trigger	Output
7	Blue	-10 to 30V dc	Input
8	Red	Discrete I/O #2	In/Out
9	Orange	Product Change	Input
10	Light Blue	Discrete I/O #4	In/Out
11	Black	Discrete I/O #1	In/Out
12	Violet	RS-232 Receive ¹	Input
Shield	Bare Metal	Chassis Ground	

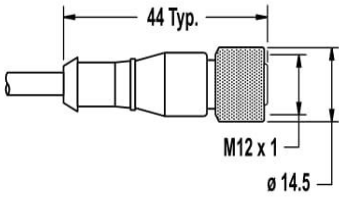
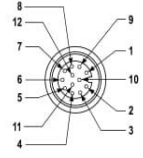


NOTE: All unused inputs and outputs should be connected to ground if configured as PNP, and connected to +24V dc if configured as NPN. Serial input pins should be connected to ground.

Cordsets

Power Cordsets

I/O sealed high-flex cordsets 12 flying leads

12-Pin M12/Euro-Style Cordsets with Open Shield				
Model	Length	Style	Dimensions	Pinout
MQDC2S-1206	1.83 m (6 ft)	Straight		 <p>1 = White 2 = Brown 3 = Green 4 = Yellow 5 = Gray 6 = Pink 7 = Blue 8 = Red 9 = Orange 10 = Lt. Blue 11 = Black 12 = Violet</p>
MQDC2S-1215	4.57 m (15 ft)			
MQDC2S-1230	9.14 m (30 ft)			
MQDC2S-1250	15.2 m (50 ft)			
MQDC2S-1275	22.9 m (75 ft)			

¹ These three wires make up the RS-232 serial connection.

Camera to IP68 Ring Light Cordsets

PKG3M-35-PSG3M is a cordset from the Camera to the IP68 Ring Light

PKG3M-2-PSG3M is a Cordset from the Camera to the external light

3-Pin Threaded/Snap M8/Pico-Style Cordsets—Double Ended				
Model	Length	Style	Dimensions	Pinout
PKG3M-.35-PSG3M	0.35 m (1.1 ft)	Straight		<p>Female</p>
PKG3M-2-PSG3M	2.00 m (6.6 ft)			<p>Male</p> <p>1 = Brown 3 = Blue 4 = Black</p>

Ethernet Cordsets

Straight Ethernet Cordset (for use with a hub)

RJ45 Ethernet to 8-Pin Threaded M12/Euro-Style Cordsets				
Model	Length	Style	Dimensions	Pinout
STP-MAQDC-806	1.83 m (6 ft)	Cat5e Shielded		<p>1 = Wh/Bl 2 = Wh/Br 3 = Brown 4 = Orange 5 = Wh/Gr 6 = Wh/Or 7 = Blue 8 = Green</p>
STP-MAQDC-815	4.57 m (15 ft)			
STP-MAQDC-830	9.14 m (30 ft)			

Video Cordsets

Video Cordset for use with PPM8

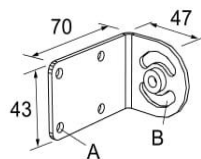
BNC to 4-Pin Threaded M8/Pico-Style Cordsets with Shield				
Model	Length	Style	Dimensions	Pinout
PKG4M-2/CS	2.00 m (6.6 ft)	BNC/Pico QD Straight		<p>Female</p> <p>1 = Brown 2 = Not Used 3 = Blue 4 = Drain</p> <p>Male</p>
PKG4M-5/CS	5.00 m (16.4 ft)			
PKG4M-9/CS	9.00 m (29.5 ft)			

Mounting Bracket

SMBP4SRAF

Bracket includes:

- 4 stainless steel M5 mounting screws
- Allen wrench



A = 4 × ø 5.5; B = 4 × ø 7

Banner Engineering Corp Limited Warranty

Banner Engineering Corp. warrants its products to be free from defects in material and workmanship for one year following the date of shipment. Banner Engineering Corp. will repair or replace, free of charge, any product of its manufacture which, at the time it is returned to the factory, is found to have been defective during the warranty period. This warranty does not cover damage or liability for misuse, abuse, or the improper application or installation of the Banner product.

THIS LIMITED WARRANTY IS EXCLUSIVE AND IN LIEU OF ALL OTHER WARRANTIES WHETHER EXPRESS OR IMPLIED (INCLUDING, WITHOUT LIMITATION, ANY WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE), AND WHETHER ARISING UNDER COURSE OF PERFORMANCE, COURSE OF DEALING OR TRADE USAGE.

This Warranty is exclusive and limited to repair or, at the discretion of Banner Engineering Corp., replacement. **IN NO EVENT SHALL BANNER ENGINEERING CORP. BE LIABLE TO BUYER OR ANY OTHER PERSON OR ENTITY FOR ANY EXTRA COSTS, EXPENSES, LOSSES, LOSS OF PROFITS, OR ANY INCIDENTAL, CONSEQUENTIAL OR SPECIAL DAMAGES RESULTING FROM ANY PRODUCT DEFECT OR FROM THE USE OR INABILITY TO USE THE PRODUCT, WHETHER ARISING IN CONTRACT OR WARRANTY, STATUTE, TORT, STRICT LIABILITY, NEGLIGENCE, OR OTHERWISE.**

Banner Engineering Corp. reserves the right to change, modify or improve the design of the product without assuming any obligations or liabilities relating to any product previously manufactured by Banner Engineering Corp.

Anexo C. Código Fonte da Interface Gráfica criada no Visual Studio

Ecrã Principal

```
1. Imports System.Net.Sockets
2. Imports System.Threading
3.
4. Public Class Câmara
5.     Public Shared client As New TcpClient()
6.     Public Shared stream As NetworkStream
7.
8.     Public Shared valores(2) As String
9.     Public Shared passcnt As Integer
10.    Public Shared failcnt As Integer
11.    Public Shared misscnt As Integer
12.    Public Shared totalcnt As Integer
13.    Public Shared programa As Integer
14.
15.    Private Sub Form1_Load(sender As Object, e As
    EventArgs) Handles MyBase.Load
16.
17.        Trigger.Enabled = False
18.        ModoContínuo.Enabled = False
19.        Remote_Teach.Enabled = False
20.        Selectprograma.Enabled = False
21.        Reset.Enabled = False
22.        Desconectar.Enabled = False
23.        Ligar.Enabled = False
24.        timer.Enabled = False
25.        sensor.Enabled = False
26.
27.
28.    End Sub
29.
30.    Private Sub Ligar_Click(sender As Object, e As
    EventArgs) Handles Ligar.Click
31.
32.        IPAdress = Config.IP1.Text & Chr(46) &
    Config.IP2.Text & Chr(46) & Config.IP3.Text & Chr(46) &
    Config.IP4.Text
33.        Porta = Config.TCP_port.Text
34.
35.
36.        If OpenETH1(IPAdress, Porta) = True Then
37.
38.            client.SendTimeout = 0
39.            stream = client.GetStream()
40.
41.            Trigger.Enabled = True
42.            ModoContínuo.Enabled = True
43.            Remote_Teach.Enabled = True
44.            Selectprograma.Enabled = True
45.            Reset.Enabled = True
46.            Desconectar.Enabled = True
47.            Ligar.Enabled = False
48.            timer.Enabled = True
49.            timer.Enabled = True
```

```

50.
51.         Dim tf3 As New Task(Sub() valores_entrada())
52.         tf3.RunSynchronously()
53.         tf3.Wait()
54.
55.         Pass.Text = passcnt
56.         Fail.Text = failcnt
57.         Missed.Text = misscnt
58.         inspection_number.Text = programa
59.         Total.Text = totalcnt
60.
61.     Else
62.         MsgBox("Conexão Falhada")
63.
64.     End If
65.
66. End Sub
67.
68. Private Sub Desconectar_Click(sender As Object, e As
EventArgs) Handles Desconectar.Click
69.
70.     stream.Close()
71.     client = New System.Net.Sockets.TcpClient()
72.     rt_picture.Image = My.Resources.circulo_preto
73.
74.     Trigger.Enabled = False
75.     ModoContínuo.Enabled = False
76.     Remote_Teach.Enabled = False
77.     Selectprograma.Enabled = False
78.     Ligar.Enabled = True
79.     Reset.Enabled = False
80.
81. End Sub
82.
83. Private Sub Configurarbutton_Click(sender As Object, e
As EventArgs) Handles Configurarbutton.Click
84.     Config.Show()
85.     Ligar.Enabled = True
86.
87. End Sub
88.
89. Private Sub Trigger_Click(sender As Object, e As
EventArgs) Handles Trigger.Click
90.
91.     Dim tg As New Task(Sub() trigger_valores())
92.     tg.Start()
93.     tg.Wait()
94.
95.     Pass.Text = passcnt
96.     Fail.Text = failcnt
97.     Missed.Text = misscnt
98.     Total.Text = totalcnt
99.
100. End Sub
101.

```

```

102.     Private Sub Remote_Teach_Click(sender As Object, e As
        EventArgs) Handles Remote_Teach.Click
103.
104.
105.         Dim teach = Readincomigdata("exec rteach trigger")
106.
107.         If teach.Substring(0, 1) = 0 Then
108.             rt_picture.Image = My.Resources.circulo_verde
109.         Else
110.             rt_picture.Image =
                My.Resources.circulo_vermelho
111.         End If
112.
113.     End Sub
114.
115.     Private Sub Selectprograma_Click(sender As Object, e As
        EventArgs) Handles Selectprograma.Click
116.
117.         Dim programa As Integer
118.         programa = sensor.Text
119.
120.         If sensor.Text = "0" Then
121.             programa = inspection_number.Text
122.         End If
123.
124.         Dim prog = Readincomigdata("exec prodchange " &
            programa)
125.
126.
127.         Dim insp = Readincomigdata("get inspnum")
128.         If insp.Length > 1 Then
129.             inspection_number.Text = insp.Substring(3, 2)
130.         Else
131.             inspection_number.Text = insp.Substring(0, 1)
132.         End If
133.
134.         sensor.Text = ""
135.
136.     End Sub
137.
138.     Public Shared Function OpenETH1(ByVal IPAdress As
        String, ByVal Porta As Integer)
139.
140.         If Not My.Computer.Network.Ping(IPAdress, Porta)
            Then
141.             Console.WriteLine(" Scanner with IP: " &
                IPAdress & " fail")
142.             Return False
143.         End If
144.
145.         client.Connect(IPAdress, Porta)
146.
147.         Return True
148.     End Function
149.
150. Private Function Readincomigdata(ByVal sb As String)

```

```

151.
152.     sb = sb & vbCrLf
153.     Dim responseData As [String] = [String].Empty
154.     Dim data As [Byte]()
155.     data =
        System.Text.Encoding.ASCII.GetBytes(sb.ToString())
156.     stream.Write(data, 0, data.Length)
157.     Dim bytes As Int32 = stream.Read(data, 0,
        data.Length)
158.
159.     Thread.Sleep(10)
160.
161.     responseData &=
        System.Text.Encoding.ASCII.GetString(data, 0, bytes)
162.
163.     Dim i As Integer
164.     For i = 0 To 1
165.
166.         valores(i) = responseData.Replace(vbCrLf, "")
167.
168.     Next i
169.
170.     Return responseData
171. End Function
172.
173. Private Sub ModoContinuo_CheckedChanged(sender As
        Object, e As EventArgs) Handles ModoContinuo.CheckedChanged
174.
175.     If ModoContinuo.Checked = True Then
176.
177.         If timer.Text = "" Or timer.Text = "0" Then
178.             timer.Text = 10
179.         End If
180.
181.         Timer1.Interval = timer.Text
182.         timer.Enabled = False
183.         Trigger.Enabled = False
184.         Timer1.Start()
185.     Else
186.         Trigger.Enabled = True
187.         timer.Enabled = True
188.         Timer1.Stop()
189.     End If
190. End Sub
191.
192. Private Sub Timer1_Tick(sender As Object, e As
        EventArgs) Handles Timer1.Tick
193.
194.     Dim ct As New Task(Sub() trigger_valores())
195.     ct.RunSynchronously()
196.     ct.Wait()
197.
198.     Pass.Text = passcnt
199.     Fail.Text = failcnt
200.     Missed.Text = misscnt
201.     Total.Text = totalcnt

```

```

202.
203.     End Sub
204.
205.     Private Sub Reset_Click(sender As Object, e As
        EventArgs) Handles Reset.Click
206.
207.         Readincomigdata("exec reset PASSFAILCNT")
208.         Readincomigdata("exec reset MISSTRIGCNT")
209.
210.
211.         Dim p = Readincomigdata("get passcnt")
212.         Pass.Text = p.Substring(0, 1)
213.         Dim f = Readincomigdata("get failcnt")
214.         Fail.Text = f.Substring(0, 1)
215.         Dim m = Readincomigdata("get misstrigcnt")
216.         Missed.Text = m.Substring(0, 1)
217.         Dim t = Readincomigdata("get intercnt")
218.         Total.Text = t.Substring(0, 1)
219.
220.     End Sub
221.
222.     Sub valores_entrada()
223.
224.
225.         Dim q = Readincomigdata("get passcnt")
226.         Dim pl = valores(1).Length
227.         If pl >= 2 Then '2
228.             passcnt = valores(1).Substring(0, 2)
229.             If pl >= 3 Then '>=3
230.                 passcnt = valores(1).Substring(0, 3)
231.             End If
232.         Else
233.             passcnt = valores(1).Substring(0, 1)
234.         End If
235.
236.
237.         Dim b = Readincomigdata("get failcnt")
238.         Dim fl = valores(1).Length
239.         If fl >= 3 Then
240.             failcnt = valores(1).Substring(1, 2)
241.             If fl > 4 Then
242.                 failcnt = valores(1).Substring(1, 3)
243.             End If
244.         Else
245.             failcnt = valores(1).Substring(1, 1)
246.         End If
247.
248.
249.         Dim c = Readincomigdata("get misstrigcnt")
250.         Dim ml = valores(1).Length
251.         If ml >= 3 Then
252.             misscnt = valores(1).Substring(1, 2)
253.             If ml > 4 Then
254.                 misscnt = valores(1).Substring(1, 3)
255.             End If
256.         Else

```

```

257.         misscnt = valores(1).Substring(1, 1)
258.     End If
259.
260.
261.     Dim k = Readincomigdata("get itercnt")
262.     Dim tt = valores(1).Length
263.     If tt >= 3 Then
264.         totalcnt = valores(1).Substring(1, 2)
265.         If tt > 3 Then
266.             totalcnt = valores(1).Substring(1, 3)
267.         End If
268.     Else
269.         totalcnt = valores(1).Substring(1, 1)
270.     End If
271.
272.
273.     Dim d = Readincomigdata("get inspnum")
274.     Dim il = valores(1).Length
275.     If il >= 3 Then
276.         programa = valores(1).Substring(1, 2)
277.     Else
278.         programa = valores(1).Substring(1, 1)
279.     End If
280.
281.
282.     Dim f = Readincomigdata("get readyflg")
283.
284.     If valores(1).Substring(1, 1) = 1 Then
285.         Ready_led.Image = My.Resources.circulo_verde
286.     Else
287.         Ready_led.Image = My.Resources.circulo_vermelho
288.     End If
289.
290.
291.     Dim g = Readincomigdata("get SYSERRFLG")
292.
293.     If valores(1).Substring(1, 1) = 1 Then
294.         Error_led.Image = My.Resources.circulo_vermelho
295.     Else
296.         Error_led.Image = My.Resources.circulo_preto
297.     End If
298.
299.
300.     For i As Integer = 1 To 4
301.
302.         Readincomigdata("get IOSTS " & i)
303.
304.         Select Case i
305.             Case 1
306.                 If valores(1).Substring(1, 1) = 1 Then
307.                     io1.Image =
My.Resources.circulo_verde
308.                 Else
309.                     io1.Image =
My.Resources.circulo_vermelho
310.                 End If

```

```

311.             Case 2
312.                 If valores(1).Substring(1, 1) = 1 Then
313.                     io2.Image =
314.                         My.Resources.circulo_verde
315.                     Else
316.                         io2.Image =
317.                             My.Resources.circulo_vermelho
318.                     End If
319.             Case 3
320.                 If valores(1).Substring(1, 1) = 1 Then
321.                     io3.Image =
322.                         My.Resources.circulo_verde
323.                     Else
324.                         io3.Image =
325.                             My.Resources.circulo_vermelho
326.                     End If
327.             Case 4
328.                 If valores(1).Substring(1, 1) = 1 Then
329.                     io4.Image =
330.                         My.Resources.circulo_verde
331.                     Else
332.                         io4.Image =
333.                             My.Resources.circulo_vermelho
334.                     End If
335.             End Select
336.         Next
337.     End Sub
338.
339.     Sub trigger_valores()
340.         Readincomigdata("exec trigger")
341.
342.         Dim w = Readincomigdata("get passcnt")
343.         Dim pl = valores(1).Length
344.         If pl >= 3 Then
345.             passcnt = valores(1).Substring(1, 2)
346.             If pl > 3 Then
347.                 passcnt = valores(1).Substring(1, 3)
348.             End If
349.         Else
350.             passcnt = valores(1).Substring(1, 1)
351.         End If
352.
353.         Dim q = Readincomigdata("get failcnt")
354.         Dim fl = valores(1).Length
355.         If fl >= 3 Then
356.             failcnt = valores(1).Substring(1, 2)
357.             If fl > 3 Then
358.                 failcnt = valores(1).Substring(1, 3)
359.             End If
360.         Else
361.             failcnt = valores(1).Substring(1, 1)

```

```

361.         End If
362.
363.
364.         Dim k = Readincomigdata("get misstrigcnt")
365.         Dim ml = valores(1).Length
366.         If ml >= 3 Then
367.             misscnt = valores(1).Substring(1, 2)
368.             If ml > 3 Then
369.                 misscnt = valores(1).Substring(1, 3)
370.             End If
371.         Else
372.             misscnt = valores(1).Substring(1, 1)
373.         End If
374.
375.
376.         Dim g = Readincomigdata("get itercnt")
377.         Dim tt = valores(1).Length
378.         If tt >= 3 Then
379.             totalcnt = valores(1).Substring(1, 2)
380.             If tt > 3 Then
381.                 totalcnt = valores(1).Substring(1, 3)
382.             End If
383.         Else
384.             totalcnt = valores(1).Substring(1, 1)
385.         End If
386.
387.
388.         Readincomigdata("get PASSFAILFLG")
389.
390.         If valores(1).Substring(1, 1) = 1 Then
391.             Pass_led.Image = My.Resources.circulo_verde
392.             Fail_led.Image = My.Resources.circulo_preto
393.
394.             For i As Integer = 1 To 4
395.
396.                 Readincomigdata("get IOSTS " & i)
397.
398.                 Select Case i
399.                     Case 1
400.                         If valores(1).Substring(1, 1) = 1
401.                             Then
402.                                 io1.Image =
403.                                     My.Resources.circulo_verde
404.                                 Else
405.                                     io1.Image =
406.                                         My.Resources.circulo_vermelho
407.                                 End If
408.                             Case 2
409.                                 If valores(1).Substring(1, 1) = 1
410.                                     Then
411.                                         io2.Image =
412.                                             My.Resources.circulo_verde
413.                                         Else
414.                                             io2.Image =
415.                                                 My.Resources.circulo_vermelho
416.                                         End If

```

```

411.             Case 3
412.             If valores(1).Substring(1, 1) = 1
    Then
413.                 io3.Image =
    My.Resources.circulo_verde
414.             Else
415.                 io3.Image =
    My.Resources.circulo_vermelho
416.             End If
417.             Case 4
418.             If valores(1).Substring(1, 1) = 1
    Then
419.                 io4.Image =
    My.Resources.circulo_verde
420.             Else
421.                 io4.Image =
    My.Resources.circulo_vermelho
422.             End If
423.             End Select
424.
425.         Next
426.
427.     Else
428.
429.         Fail_led.Image = My.Resources.circulo_vermelho
430.         Pass_led.Image = My.Resources.circulo_preto
431.
432.         For i As Integer = 1 To 4
433.
434.             Readincomigdata("get IOSTS " & i)
435.
436.             Select Case i
437.                 Case 1
438.                     If valores(1).Substring(1, 1) = 1
    Then
439.                         io1.Image =
    My.Resources.circulo_verde
440.                     Else
441.                         io1.Image =
    My.Resources.circulo_vermelho
442.                     End If
443.                 Case 2
444.                     If valores(1).Substring(1, 1) = 1
    Then
445.                         io2.Image =
    My.Resources.circulo_verde
446.                     Else
447.                         io2.Image =
    My.Resources.circulo_vermelho
448.                     End If
449.                 Case 3
450.                     If valores(1).Substring(1, 1) = 1
    Then
451.                         io3.Image =
    My.Resources.circulo_verde
452.                     Else

```

```

453.             io3.Image =
My.Resources.circulo_vermelho
454.             End If
455.             Case 4
456.                 If valores(1).Substring(1, 1) = 1
Then
457.                     io4.Image =
My.Resources.circulo_verde
458.                 Else
459.                     io4.Image =
My.Resources.circulo_vermelho
460.                 End If
461.             End Select
462.
463.         Next
464.
465.     End If
466. End Sub
467. End Class

```

Ecrã Secundário

```

1. Public Class Config
2.     Private Sub Config_Closing(sender As Object, e As
FormClosingEventArgs) Handles MyBase.Closing
3.         Me.Hide()
4.         Câmara.Show()
5.
6.     End Sub
7.
8.     Private Sub OK_Click(sender As Object, e As EventArgs)
Handles OK.Click
9.
10.        If IP1.Text = "" Or IP2.Text = "" Or IP3.Text = ""
Or IP4.Text = "" Or TCP_port.Text = "" Then
11.            MsgBox("Preencha corretamente o IP Address ou o
TCP Port !")
12.        Else
13.            Me.Hide()
14.            Câmara.Show()
15.        End If
16.
17.    End Sub
18.
19.    Private Sub Cancel_Click(sender As Object, e As
EventArgs) Handles Cancel.Click
20.
21.        Dim result As DialogResult = MessageBox.Show("Tem a
certeza que pretende sair ? ", "Fechar",
MessageBoxButtons.YesNo, MessageBoxIcon.Question)
22.
23.        If result = Windows.Forms.DialogResult.Yes Then
24.            Me.Hide()
25.            Câmara.Show()

```

```
26.         ElseIf result = Windows.Forms.DialogResult.No Then
27.             Me.Show()
28.         End If
29.
30.     End Sub
31. End Class
```