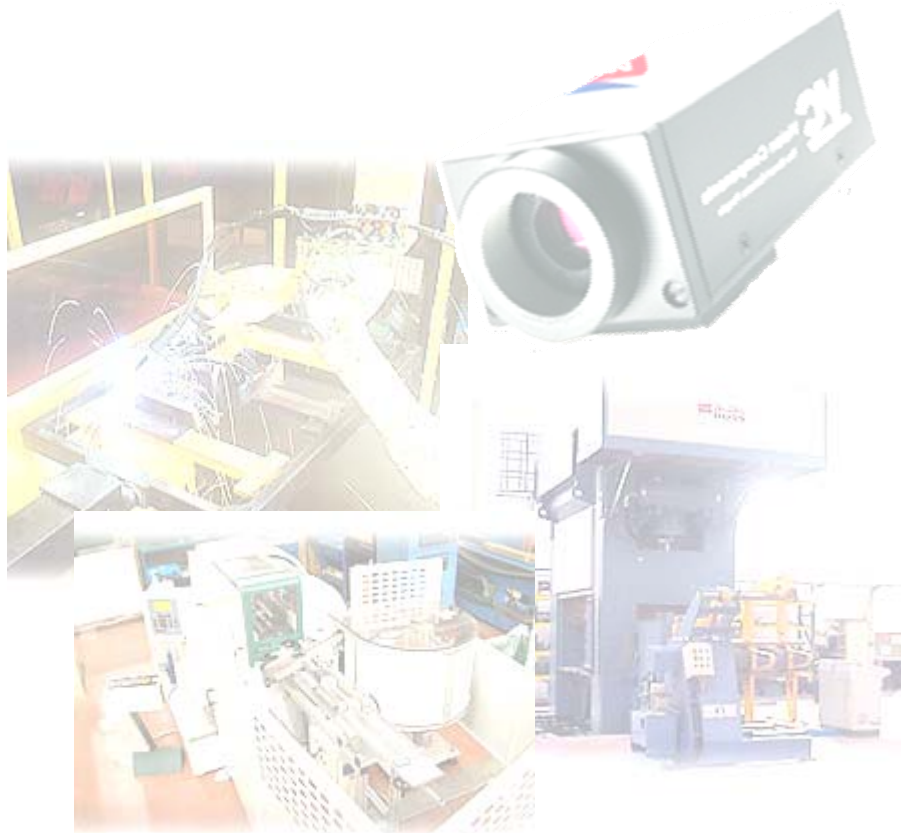




David Manuel Costa Gameiro **Técnicas de Reconhecimento Paramétrico em Sistemas Automáticos de Controlo Qualidade**





David Manuel Costa Gameiro **Técnicas de Reconhecimento Paramétrico em Sistemas Automáticos de Controlo Qualidade**

Dissertação apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Mestre em Engenharia Mecânica, realizada sob a orientação científica do Prof. Dr. Vítor Santos, Professor associado do Departamento de Engenharia Mecânica da Universidade de Aveiro

Dedicatória

Dedico este trabalho ao meu Pai e à minha Irmã, tal como à minha Mãe a qual se estivesse entre nós me teria apoiado de igual forma.

o júri

presidente

Prof. Doutor José Joaquim de Almeida Grácio

professor catedrático do Departamento de Engenharia Mecânica da Universidade de Aveiro

Prof. Doutor Carlos Baptista Carneira

professor auxiliar do Departamento de Engenharia Mecânica do Instituto Superior Técnico

Prof. Doutor. Vítor Manuel Ferreira dos Santos

professor associado do Departamento de Engenharia Mecânica da Universidade de Aveiro

Agradecimentos

Queria agradecer à minha família, ao prof. Dr. Vitor Santos, e colegas por todo o apoio e comentários para a realização desta dissertação. Agradecer a cooperação por parte da Epedal, na medida em que facultou alguma informação relacionada com o tecido industrial em que se encontra inserida.

Palavras-chave

Controlo da Qualidade

Não Conformidades

Percepção sem contacto

Custos da Não Qualidade

Diminuição de tarefas repetitivas e cansativas

Visão Industrial

Técnicas Paramétricas de Inspeção

Resumo

O presente documento pretende divulgar alguns dos problemas existentes nas PME portuguesas, que estão inseridas no mercado automóvel, focando especialmente as metalomecânicas. Este trabalho apresenta uma solução para realizar o controlo e inspeção de qualidade de produtos não conformes, através do reconhecimento paramétrico associado à visão industrial.

O trabalho começa com a análise do tipo de defeitos mais comuns em peças, e dos sistemas implementados para os controlar. Aborda depois a avaliação de soluções existentes no mercado.

O algoritmo desenvolvido, assenta numa análise geométrica e topológica da peça, à semelhança dos métodos utilizados em metrologia avançada, recorrendo a entidades invariantes à rotação e posição da peça na imagem. O sistema desenvolvido permite o controlo de qualidade eficaz de várias peças com cadências elevadas.

Keywords

Quality Control

Non Conformities

Non Contact Perception

Non Quality costs

Reduce repetitive and wearing tasks

Industrial Vision

Parametric techniques

Abstract

The present document intends to reveal some of the existing problems in the Portuguese SM (Small and Medium) companies, which are inserted in the automobile market, focused in metallic components.

This work presents a solution to carry out the parametric recognition associated to industrial vision. The work starts with an analysis of each type of the main common parts defects, and the implemented systems to control them. Later It'll approach the evaluation of existing solutions in the market.

The developed algorithm is based in a geometric and topological study of the part, similar to the methods used in advanced metrology, appealing to invariant features related with the position and rotation of the part in the image.

The developed system is capable of ensure product quality control with a reliable and stable answer at high frequency.

ÍNDICE DE FIGURAS

fig. 1 – Portugal e a Globalização	8
fig. 2 – Inspeção manual a 100%	9
fig. 3 – Custos da Não Qualidade 2006	10
fig. 4 – diagrama de Pareto das NC	11
fig. 5 – NC devido a desvios dimensionais	12
fig. 6 – Exemplo de peças a controlar	14
fig. 7 – Exemplo de entidades referêcia	16
fig. 8 – Calibres de controlo por atributos. (a) tolerâncias geométricas e lineares, (b) tolerância linear ou geométrica	17
fig. 9 – calibre de controlo para controlo por variáveis	18
fig. 10 – calibre de controlo para controlo misto	18
fig. 11 – (a) Máquinas tridimensionais de medição (b) Microscópio com câmara visão	19
fig. 12 – Ambiente de trabalho da aplicação (a) Metrolog XG (b) AxioVision	19
fig. 13 – Evolução do número de autómatos	21
fig. 14 – exemplo de um autómato de inspeção	22
fig. 15 – fluxograma de funcionamento de um autómato	22
fig. 16 – Investimento aplicado na área da visão industrial no período de 1996 a 2000 (Horst Haußecker, 1999)	23
fig. 17 – Tipo de Empresas que investiram em sistemas de visão (Horst Haußecker, 1999)	23
fig. 18 – exemplo de câmaras inteligentes	24
fig. 19- Interfaces do Insigth(a), Sherlock(b) e do Neurocheck(c)	24
fig. 20 – Biblioteca de imagens modelo	25
fig. 21 – Imagens teste simples	25
fig. 22 – Imagem para Pré-Processamento (Gancho 1)	28
fig. 23 – Gancho 1 na escala de cinzentos	28
fig. 24 – Gancho1 Binarizado	29
fig. 25 – Moldura exibida a preto apenas como exemplo	30
fig. 26 – Furos existentes no Gancho 1	30
fig. 27 – Entidades existentes, (a) exemplo 1, (b) Exemplo 1 rodado	30
fig. 28 – Arestas Detectadas pelo detector de Canny	33
fig. 29 – Representação gráfica de três rectas, em que o declive é o mesmo na recta vermelha e azul.	33
fig. 30 – Representação de uma linha recta através de uma parametrização por coordenadas polares ρ, θ	33
fig. 31 - Identificação de linhas existentes na imagem da fig. 28	34
fig. 32 – Representação da recta média que melhor representa a aresta.	34
fig. 33 – Proporção de votos em função do comprimento da linha (menor para y1)	36

fig. 34 – Linhas recta detectadas pelo PPHT, indicadas a verde. _____	36
fig. 35 – diagrama fluxo do algoritmo para o PPHT no OpenCV _____	37
fig. 36 – Exemplo das propriedades retiradas de uma linha, relativamente ao sistema de eixos da imagem _____	39
fig. 37 – Círculos detectados pelo algoritmo do OpenCV _____	40
fig. 38 – (a) ruído deixado pela binarização, potenciais não furos, (b) a vermelho não furos detectados e a verde os furos _____	41
fig. 39 - Ciclo de busca dos furos (CBF) _____	42
fig. 40 – (a) imagem preenchida com o ruído (b) identificação de dois furos por parte do algoritmo (área a rosa e âncora a verde) _____	43
fig. 41 – Ganho 1 com a identificação das propriedades singulares de duas peças idênticas, peça (b) é a peça (a) rodada em 180º _____	44
fig. 42 – Representação de propriedades relativas nas peças da fig. 41 _____	45
fig. 43 – variação da distância com a má escolha do ponto âncora _____	46
fig. 44 – distância normal (pé da perpendicular) _____	47
fig. 45 – (a) Representação de todos os ângulos no Gancho 1, (b) representação do ângulo entre dois vectores directores com a mesma origem _____	47
fig. 46 – Esquema do modelo descritivo de uma peça modelo (n e m podem ser diferentes para cada entidade) _____	48
fig. 47 – (a) Imagem de um Gancho (b) indicação das entidades e dos seus pontos âncora _____	49
fig. 48 – Árvore descritiva da peça da fig. 47 _____	50
fig. 49 – Classificação das propriedades relativas entre entidades _____	51
fig. 50 – AD completa com todas as entidades e as suas propriedades _____	52
fig. 51 – Árvore do MD após memorização do modelo com as entidades relativas. _____	53
fig. 52 – Aspecto gráfico da aplicação desenvolvida em VCE _____	54
fig. 53 – (a) Banca de ensaio utilizada, (b) <i>BirdView</i> _____	55
fig. 54 – Classe <i>ClassImage</i> a qual é a base de toda a aplicação (excerto) _____	58
fig. 55 – Classificação das variáveis e métodos da <i>ClassSpot</i> (excerto) _____	59
fig. 56 - Classificação das propriedades e métodos da <i>ClassLine</i> (excerto) _____	59
fig. 57 - Classificação da <i>ClassFeature</i> (excerto) _____	60
fig. 58 - Classificação da <i>ClassRelation</i> (excerto) _____	61
fig. 59 – Algoritmo das primeiras duas fases do módulo MD _____	61
fig. 60 – Árvore para um limite de binarização de 120 _____	62
fig. 61 – Árvore com as entidades listadas e as suas propriedades _____	63
fig. 62 – Ciclo de classificação das propriedades relativas _____	64
fig. 63 – Matriz com as propriedades relativas entre diversas entidades _____	65
fig. 64 – Árvore após passagem a MD _____	65
fig. 65 – Cascata de decisão implementada no RP _____	66
fig. 66 – MD e identificação das entidades e suas relações _____	67

fig. 67 – Peças usadas nos ensaios _____	68
fig. 68 – Peça 1 (a) sem defeitos (b) falta de um furo (c) aresta deformada _____	68
fig. 69 – Peça 1 (a) entidades escolhidas (b) MD teórico (c) MD definido no PI _____	69
fig. 70 – Resultados da peça 1 _____	70
fig. 71 – Peça 2 (a) sem defeitos (b) falta de um furo (c) aresta deformada _____	71
fig. 72 – Peça 2 (a) entidades escolhidas (b) MD teórico (c) MD real definido no PI _____	71
fig. 73 – Resultados da peça 2 _____	72
fig. 74 – Cantos nas imagens _____	74
fig. 75 – Entidades Virtuais _____	75

ÍNDICE DE TABELAS

Tabela 1 – Tabela de isoestatismo	16
Tabela 2 – Propriedades singulares de algumas entidades	31
Tabela 3 – Exemplos de propriedades relativas, em que $i \neq j$	31
Tabela 4 – propriedades singulares das entidades indicadas no capítulo 3.3	45
Tabela 5 – Matriz de propriedades relativas entre entidades	46
Tabela 6 – Exemplo de propriedades relativas entre entidades	51
Tabela 7 – numeração do tipo das propriedades	53

ACRÓNIMOS E SIGLAS

WPF – *Windows Presentation Foundation*

PI – *Part Inspector*

API – *Application Programming Interface*

VCE – *Visual C++ Express*

NC – Não Conformidade

RP – Reconhecimento Paramétrico

MD – Modelo Descritivo

PPHT – *Progressive Probabilistic Hough Transform* (Transformadas de Hough Probabilísticas Progressivas)

CBF – Ciclo de Busca de Furos

AD – Árvore Descritiva

SNR – *Signal Noise Rate* (percentagem de ruído no sinal)

ROI – Region Of Interest (Zona de Interesse)

ÍNDICE

1	Introdução	8
1.1	Enquadramento	9
1.2	Descrição do Problema	11
1.3	Objectivos	13
2	Sistemas de Controlo de Qualidade	15
2.1	Conceitos de metrologia	15
2.1.1	Desenho Definição	16
2.1.2	Metrologia Industrial	17
2.1.3	Metrologia Avançada	18
2.2	Implementações Existentes	19
2.2.1	Sistemas Semi-Automáticos	21
2.2.2	Visão Industrial	23
3	Solução Proposta	27
3.1	Condições de Funcionamento	27
3.2	Pré-processamento de Imagem	28
3.2.1	Escala de Cinzentos	28
3.2.2	Binarização	29
3.2.3	Preenchimento	29
3.3	Detecção de Entidades na imagem	30
3.3.1	Detecção de Arestas	32
3.3.2	Rectas	33
3.3.3	Furos	39
3.4	Definição de Relações Invariantes	44
3.4.1	Distâncias	46
3.4.2	Ângulos	47
3.5	Construção do Modelo Descritivo	48

3.5.1	Entidades e Propriedades Singulares	49
3.5.2	Classificação das Relações (Propriedades Relativas)	50
3.5.3	Descrição Formatada	52
4	Desenvolvimento da Aplicação de Interface	54
4.1	Hardware	55
4.2	Software	56
4.3	Part Inspector	57
4.3.1	Classes	57
4.3.2	Construção do MD	61
4.4	Reconhecimento Paramétrico	65
4.5	Resultados	68
5	Conclusões e Perspectivas Futuras	73
5.1	Conclusões	73
5.2	Perspectivas Futuras	74
6	Lista de Referências	76

1 INTRODUÇÃO

Analisando as PME portuguesas de perto verificamos que muitas delas desenvolvem produtos para terceiros (mercado interno ou externo), as quais são obrigadas a cumprir com especificações qualitativas e quantitativas dos seus clientes.

Uma parte das PME's portuguesas trabalha para a indústria automóvel, que obriga à produção em série. O principal objectivo da produção em série por parte dos fabricantes de automóveis é baixar os custos de produção.

Com a globalização os produtores de automóveis colocaram todos os seus fornecedores em concorrência directa (ex. China, Alemanha).



fig. 1 – Portugal e a Globalização

Para evitar a saída da produção de componentes as empresas portuguesas necessitaram de se tornar mais competitivas, baixando os seus custos de produção. Para baixar os seus custos de produção é necessário aumentar a produtividade e diminuir o controlo de produção (área crítica). A principal consequência da diminuição de inspecção é as não conformidades (NC), que podem aparecer devido a erro do operador ou devido ao processo produtivo.

1.1 ENQUADRAMENTO

Uma não conformidade é verificada quando algum dos requisitos impostos pelo cliente não é verificado, as NC podem ser dimensionais, visuais (aspecto), mecânicos ou químicos (propriedades da matéria-prima), e resistência estrutural.

As NC's devido ao processo produtivo são resolvidas através de técnicas preventivas de manutenção, as quais permitem detectar anomalias dos meios produtivos (ferramentas, máquinas) antes de elas ocorrerem. As NC devido à falha humana são de maior dificuldade de tratamento devido às inúmeras causas. É possível indicar algumas causas de falha com maior índice ocorrência, como é o caso da incorrecta comparação qualitativa (inspecção), incorrecta ou não execução de uma determinada operação.

Não olhando para o tipo de NC, existe sempre a necessidade de inspecionar ou resolver o actual problema. Havendo uma NC, é iniciado um plano de contenção, e umas das medidas imediatas a executar é a inspecção total do stock existente na empresa e no cliente.



fig. 2 – Inspeção manual a 100%

O controlo do stock realiza-se através da inspecção a 100% de todas as peças. Dependendo da quantidade de peças em stock e do número de pontos a verificar, é definido o número de pessoas necessárias para a execução desta inspecção.

Esta situação é insustentável para as empresas, pois leva ao aumento do custo interno das peças bem como a problemas de produtividade visto que se torna necessário redireccionar recursos para estas inspecções.

O custo da não qualidade é o maior custo das empresas actualmente, este custo aparece devido ao não cumprimento dos parâmetros de qualidade impostos pelos clientes. Olhando para uma PME no ramo da industria automóvel os custos da não qualidade ao fim de um ano permite a compra de um automóvel novo, conforme indicado pelo gráfico da fig. 3.

A avaliação dos custos da não qualidade é realizada através de indicadores das não conformidades (NC internas, externas). A revisão do stock a 100% e a recuperação das peças é uma obrigatoriedade em ambas as situações, mas no caso de uma NC externa implica que houve distúrbios na linha de produção do cliente levando a penalizações monetárias. Todos estes desvios são contabilizados e somados posteriormente levando a gráficos como o da fig. 3.

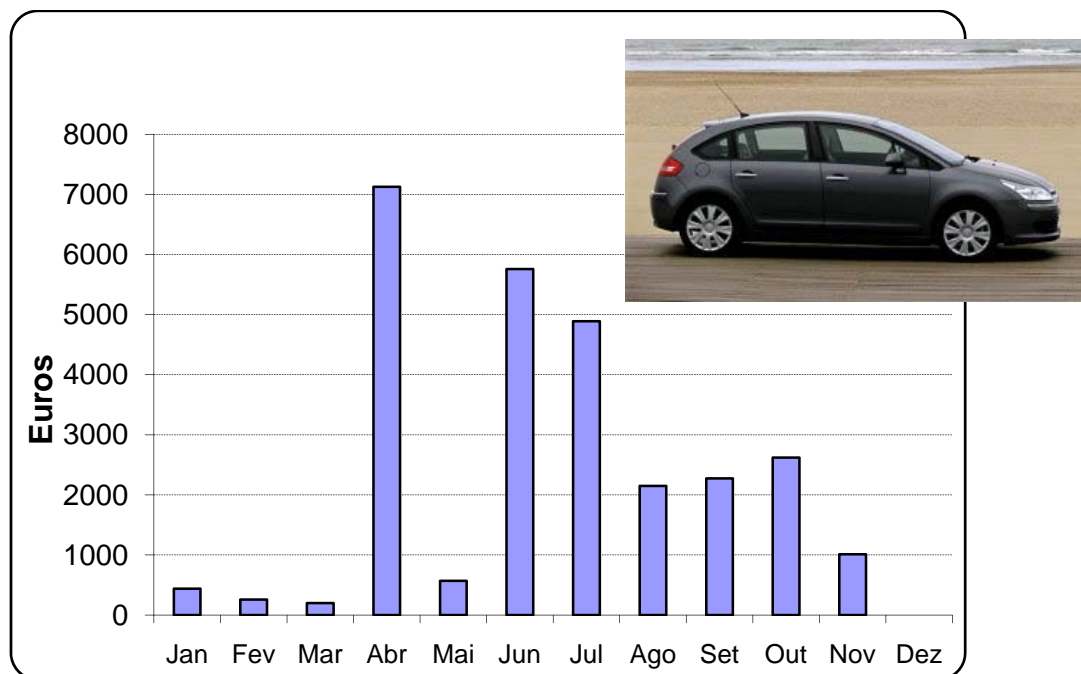


fig. 3 – Custos da Não Qualidade 2006

1.2 DESCRIÇÃO DO PROBLEMA

Com base em alguns dados fornecidos por uma das empresas inseridas nesta área, relativamente ao ano de 2006, verifica-se que o tipo de problemas existentes são sistemáticos e estão associados ao operador (erro humano) na maioria das vezes. Recorrendo-se à análise de Pareto pode-se verificar quais são as falhas mais recorrentes; a análise é realizada com base na regra dos 80-20, ou seja, 80% dos problemas devido a 20 % das causas.

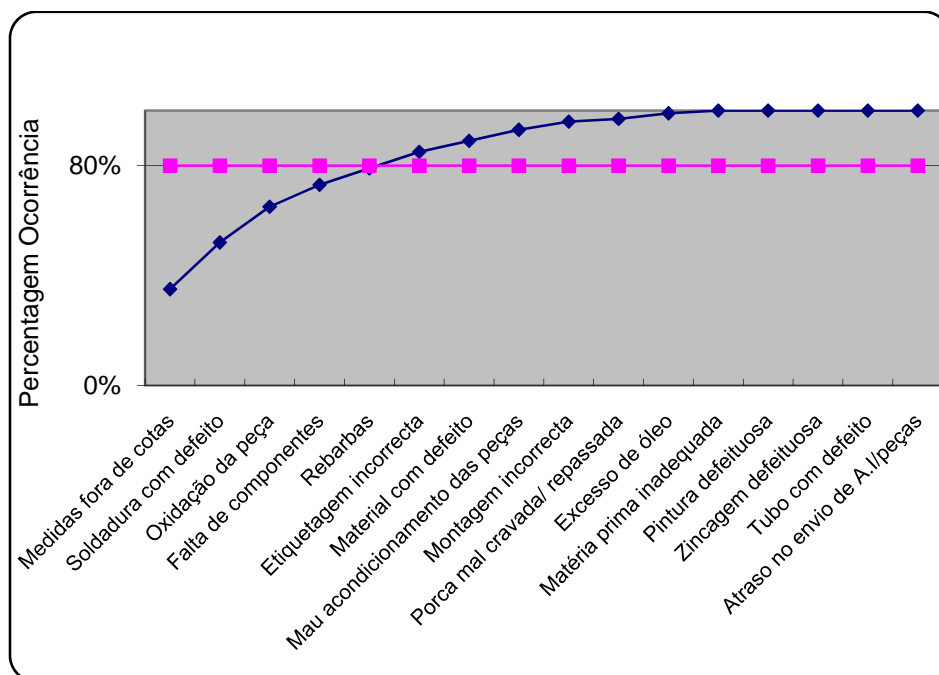


fig. 4 – diagrama de Pareto das NC

O gráfico da fig. 4 indica que na empresa em referência 80% das NC recebidas durante um ano dos seus clientes são problemas dimensionais, soldaduras com defeitos e falta de componentes. Os problemas dimensionais são os maiores problemas porque normalmente surgem devido a má inspeção dimensional ou má execução de uma operação, conforme ilustrado na fig. 5. Os desvios dimensionais aparecem devido à má montagem das peças na sua operação de montagem (fig. 5 a) ou falta de cuidado no manuseamento das peças (fig. 5 b) ou pelo não cumprimento dos procedimentos internos (fig. 5 c, d).

A peça (a) na fig. 5 é aplicada num sistema de escape e apresenta defeitos dimensionais. A causa do defeito dimensional é a posição incorrecta do tubo ou da chapa relativamente um ao outro, neste caso o tubo foi mal colocado na maqueta de soldadura originando uma peça não conforme. As consequências são um escape na linha do cliente danificado e envio de um escape sem um gancho para a linha do construtor do automóvel.

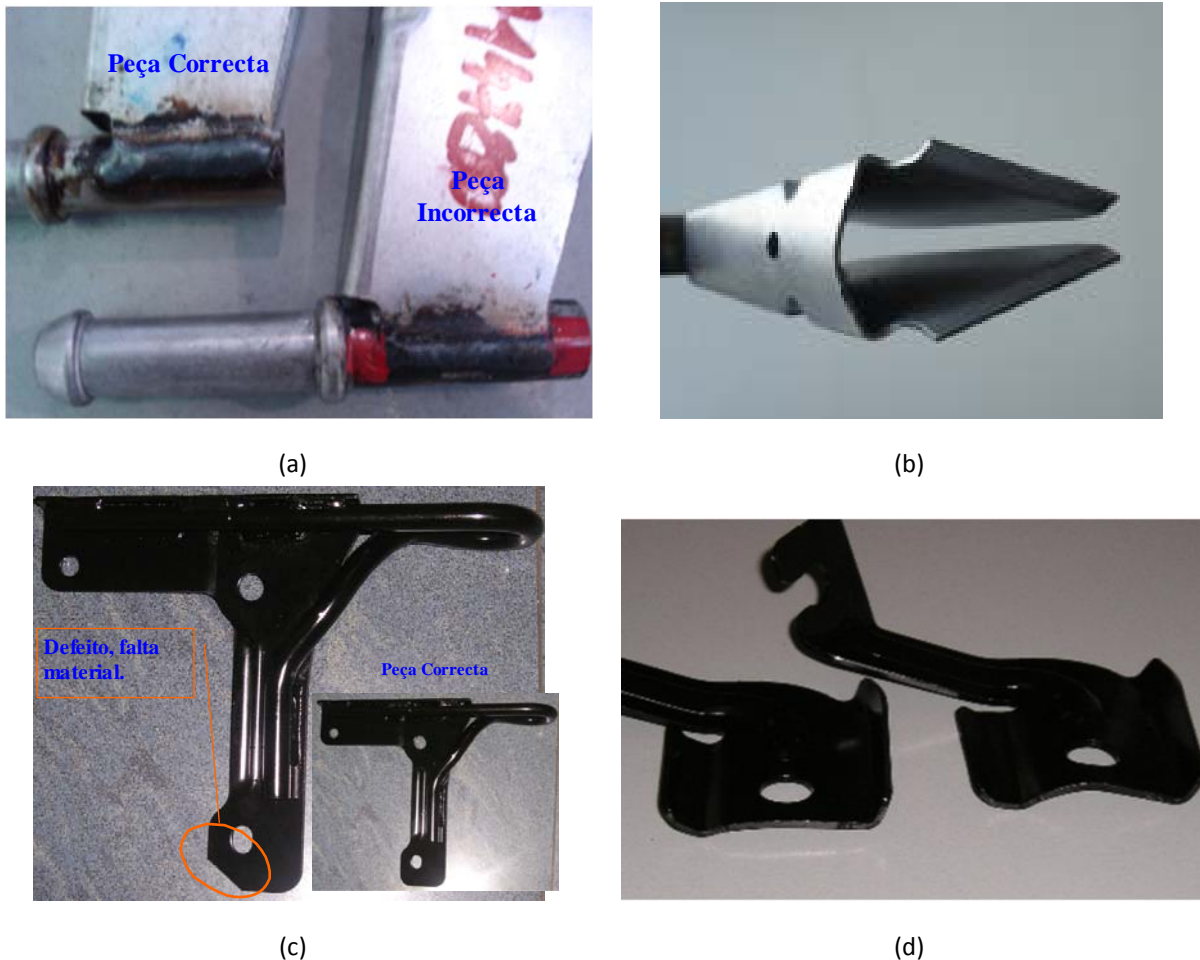


fig. 5 – NC devido a desvios dimensionais

A peça (c) na fig. 5 é um gancho de reboque e é composto por uma chapa e um arame conformado, o problema desta peça é falta de resistência mecânica. A causa poderá ser a ferramenta danificada ou não cumprimento de alguma instrução de produção, neste caso não foi cumprida a indicação para sucatear as primeiras 15 peças as quais podem conter defeitos visuais e ou dimensionais. A peça (d) serve de posicionador e suporte para um tubo com cablagem, e o problema que apresenta é impossibilidade de montagem. A causa poderá ser algum componente da ferramenta danificado ou o não cumprimento de alguma instrução, nesta situação a ferramenta não foi limpa (retirar excessos de óleo e sucata) correctamente nos intervalos previstos levando a que restos de chapa ficassem alocados num dos componentes de estampagem. Estes dois problemas foram detectados antes de chegar ao cliente mas obrigou à inspecção de todo stock existente na empresa, em todas as suas fases de trabalho (estampagem, soldadura, pintura e embalagem).

1.3 OBJECTIVOS

Neste trabalho pretende-se desenvolver um sistema de percepção e inspecção para monitorar e fazer o controlo de qualidade num processo de fabrico de componentes metálicos. Este sistema de inspecção deve operar sem contacto com as peças (visão), permitir a definição de sequências de operações através de uma interface, ser robusto e fiável em particular para o processo de fabrico em causa e para as diversas peças associadas ao problema.

O trabalho será dividido em fases, que encerram os objectivos a cumprir no decorrer do trabalho.

A primeira fase consiste em identificar as diversas soluções de visão existentes para o controlo de qualidade para as diversas peças identificadas (tubos, chapas conformadas e/ou soldadas).

A segunda fase, consiste no desenvolvimento de uma biblioteca de funções reutilizáveis para as diversas configurações da plataforma.

A terceira e última fase pressupõe o desenvolvimento de uma aplicação para servir de interface com o utilizador, e permitir reconfigurar parâmetros, visualizar e analisar dados.

O sistema deverá ser versátil, de forma a possibilitar controlar peças tão distintas como as indicadas na fig. 6 ou na fig. 5 com os seus defeitos.

Além destes objectivos, dever-se-á ter alguma preocupação procurando equipamento de baixo custo e de fácil obtenção.

O sistema pode ou não fazer parte da linha produtiva e dependerá do seu utilizador verificar se este pode ou não cumprir com as cadências onde está a ser inserido.

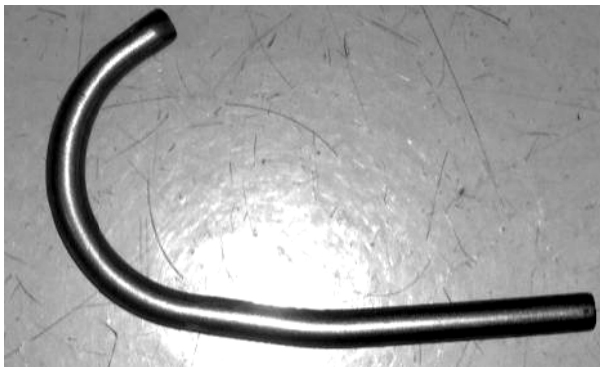


fig. 6 – Exemplo de peças a controlar

2 SISTEMAS DE CONTROLO DE QUALIDADE

Os sistemas de controlo de qualidade apareceram no decorrer da primeira grande guerra, devido ao grande aumento da complexidade das peças e das suas cadências de produção. Daí haver a necessidade de definir níveis de inspeção dependendo da peça e das características a verificar. Os níveis definidos na altura mantiveram-se e ainda são actualmente aplicados, que são a inspeção por amostragem e a 100%. Destes dois níveis existentes, o de amostragem pode ser feito em amostragem por variáveis e por atributos. A diferença entre estes dois tipos de amostragem reside no facto de uma ser baseada em números (variáveis) e a outra ser em valores binários (atributos). Ambas as amostragens podem levar a estudos estatísticos; no entanto, os resultados obtidos na amostragem por atributos não são muito representativos dos desvios existentes no processo produtivo. (Wikipedia)

Antes da primeira guerra mundial já eram produzidos automóveis; no entanto, estes fabricantes de automóveis não passavam de pequenas lojas que produziam os seus produtos manualmente. Com a primeira grande guerra, talvez um pouco antes, aparece a necessidade de aumentar as cadências produtivas para garantir toda a solicitação de produtos. Nesta altura dá-se a primeira revolução na indústria automóvel (USA), através das linhas montagem na Ford (produzir o Ford T). Para garantir que todos os requisitos qualitativos eram cumpridos; os sistemas de qualidade implementados obrigaram à inclusão de técnicas de medição no meio industrial. (Wikipedia)

2.1 CONCEITOS DE METROLOGIA

A Metrologia é a ciência que actualmente abrange todos os aspectos teóricos e práticos das técnicas de medição que asseguram a precisão exigida no processo produtivo, procurando garantir a qualidade de produtos e serviços através da calibração de instrumentos de medição e da realização de ensaios, sendo a base fundamental para a competitividade das empresas. (Wikipedia)

A metrologia e o desenho de definição possuem uma ligação muito estreita, visto que uma indica o que a outra necessita de verificar, ou seja, o desenho de definição especifica as características a cumprir e o desvio admissível relativamente às entidades de referência. A metrologia por sua vez, verifica se as dimensões especificadas se encontram dentro dos limites admissíveis, ou seja, se está dentro de tolerância.

2.1.1 DESENHO DEFINIÇÃO

O desenho de definição especifica todas as condições qualitativas de um determinado produto (peça). As condições ou especificações passam pelos materiais, qualidade e estado superficial da peça e claro das tolerâncias dimensionais da peça.

No desenho de definição todos os pontos são importantes, contudo, um dos pontos mais influentes na execução e medição da peça, são as suas entidades de referência. Estas entidades ou referências podem variar desde um ponto até uma superfície, e no entanto são preferíveis entidades com a menor variabilidade do processo. Em processos de estampagem normalmente, recorre-se a furos e planos como entidades de referência devido à sua baixa variabilidade no processo de fabrico.

O número de entidades de referência varia de peça para peça, mas para o caso indicado na fig. 7 verifica-se o recurso a um plano, um furo para indicar a origem do sistema de coordenadas e por fim um ponto que permite evitar a rotação e assim atribuir a última direcção ao sistema de coordenadas.

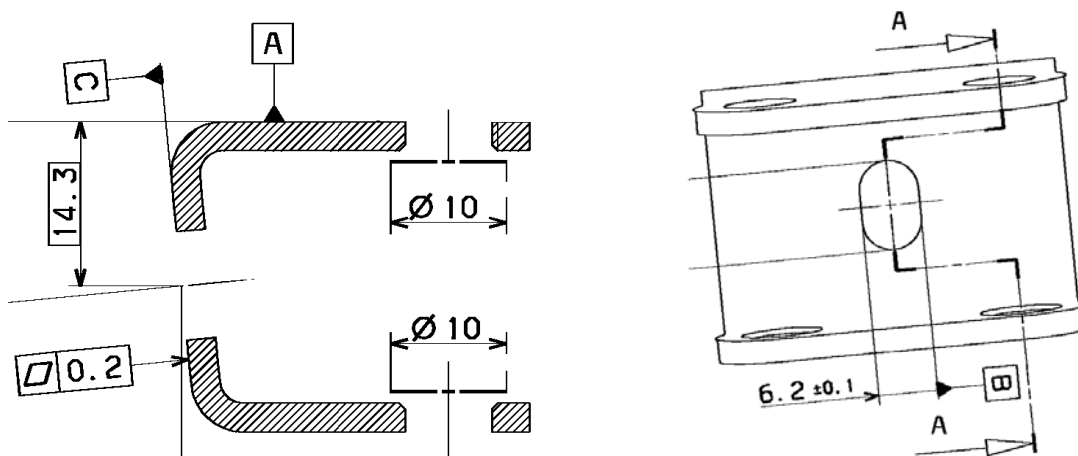


fig. 7 – Exemplo de entidades referência

Através das entidades de referência é possível criar uma tabela de isoestatismo, tabela esta que deverá ser cumprida durante todo o processo produtivo e de medição (calibres de controlo). A Tabela 1 ilustra a tabela de isoestatismo aplicada na peça da fig. 7.

Tabela 1 – Tabela de isoestatismo

Ref.	X	Y	Z
A	xxx		
B			x
C	x	x	

2.1.2 METROLOGIA INDUSTRIAL

A Metrologia Industrial baseia-se em equipamentos de medição para controlar os processos produtivos industriais e são responsáveis pela garantia da qualidade dos produtos acabados.

Estes equipamentos de medição permitem controlar dimensões por atributos ou variáveis.

O **controlo por atributos** consiste em equipamentos de medição, como a fig. 8 denominados por calibres de controlo, “passa” / “não passa”, que apenas indicam ao controlador se a tolerância a ser controlada está dentro dos limites ou não, as tolerâncias a controlar podem ser lineares ou geométricas fig. 8.

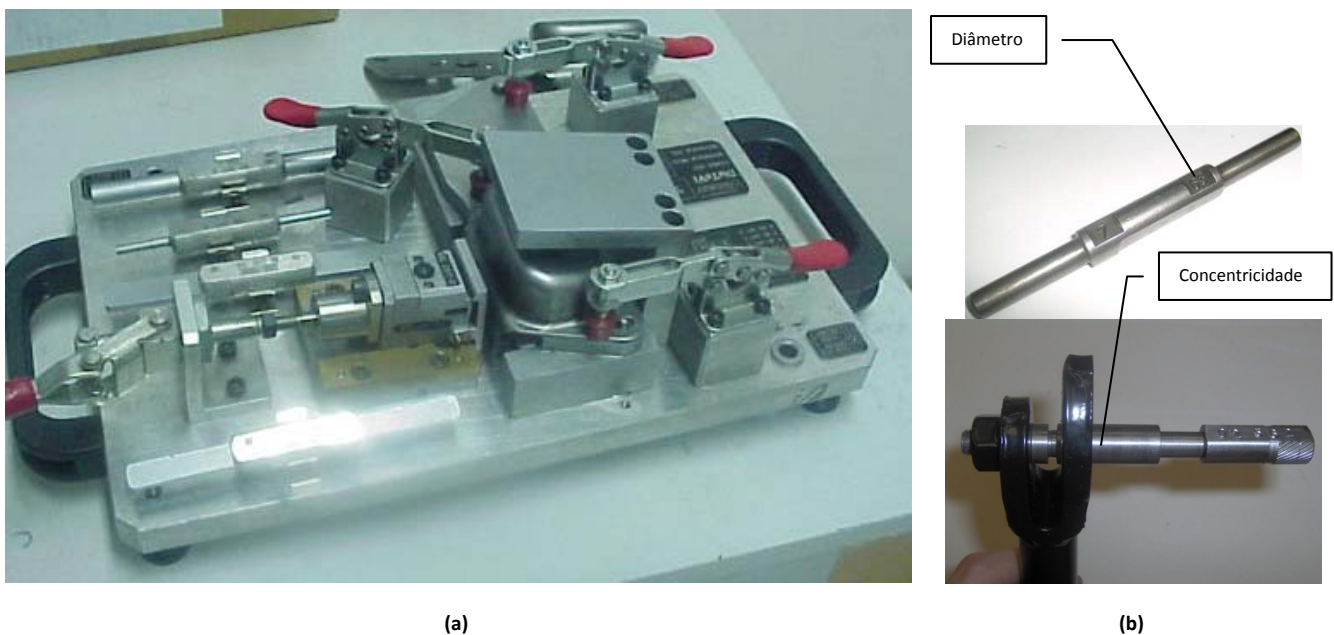


fig. 8 – Calibres de controlo por atributos.
(a) tolerâncias geométricas e lineares, (b) tolerância linear ou geométrica

Equipamentos como o da fig. 8 (b) apenas permitem controlar um tipo de cotas, lineares (diâmetros ou distância) ou geométricas (concentricidade, perpendicularidade, etc) entre entidades. Enquanto equipamentos como do tipo da fig. 8 (a) permite controlar tolerâncias lineares e geométricas (posição furos, tolerâncias de forma, etc.).

Controlo por variáveis é realizado através de calibres de controlo mais complexos do que os da fig. 8 mas são construídos da mesma forma e apenas permitem retirar dados discretos na sua utilização. O recurso a estes equipamentos permite verificar a variação do processo produtivo e assim evitar a saída de uma cota para fora do limite de tolerância antes mesmo de ela ocorrer (estudo estatístico). Os dados discretos são obtidos através do uso de relógios comparadores ou régua calibradas. Equipamentos com relógios comparadores como os da fig. 9 são os mais recorrentes devido à facilidade de afinação do equipamento e verificação.

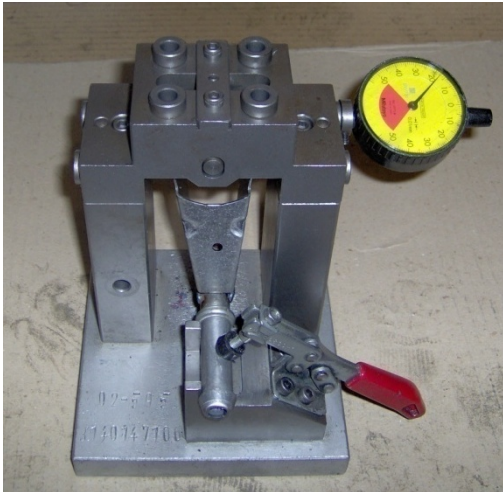


fig. 9 – calibre de controlo para controlo por variáveis

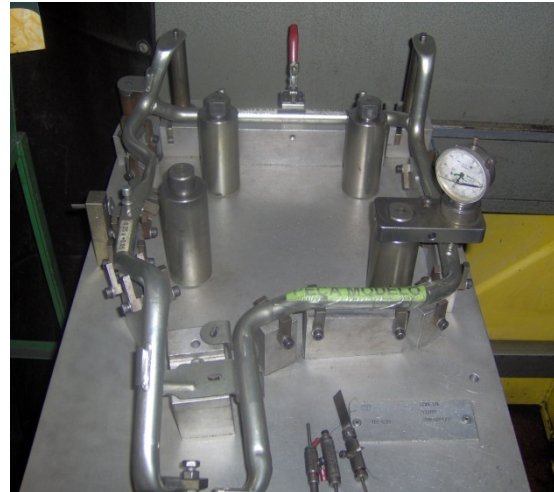


fig. 10 – calibre de controlo para controlo misto

Calibres como o da fig. 10 são aplicados quando é necessário realizar um controlo misto, ou seja, controlo por variáveis e atributos.

2.1.3 METROLOGIA AVANÇADA

A metrologia avançada aparece nas empresas com equipamentos laboratoriais de maior precisão, mas também como sistema de apoio à Metrologia Industrial. Estes equipamentos auxiliam o processo produtivo mas nunca substituem os calibres de controlo, sendo aplicados principalmente para verificar e calibrar os equipamentos metrológicos industriais. Contudo podem ser aplicados para controlar algumas cotas de difícil análise nos Calibres de Controlo.

Máquinas como as exemplificadas na fig. 11 associadas a aplicações de análise como da fig. 12 são alguns dos equipamentos utilizados em metrologia avançada. Cada fabricante destes equipamentos vende soluções de análise numérica, para que seja possível construir um modelo geométrico e comparar o valor nominal com o real.

Com a evolução dos sistemas CAD, aparece a necessidade de comparar o modelo numérico (modelo CAD) com a peça real e analisar os seus desvios, e onde é possível verificar tolerâncias mais complexas. Existem diversas aplicações de análise numérica, sendo as aplicações apresentadas como exemplo com maior sucesso na indústria automóvel europeia, como é o caso do Metrolog XG da Metrologic Group e do AxioVision da CarlZeiss.

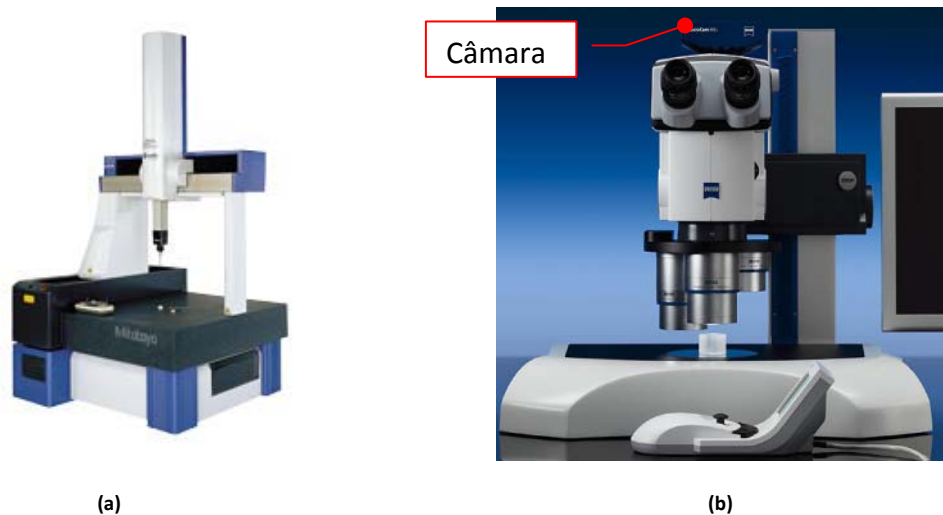


fig. 11 – (a) Máquinas tridimensionais de medição (b) Microscópio com câmara visão

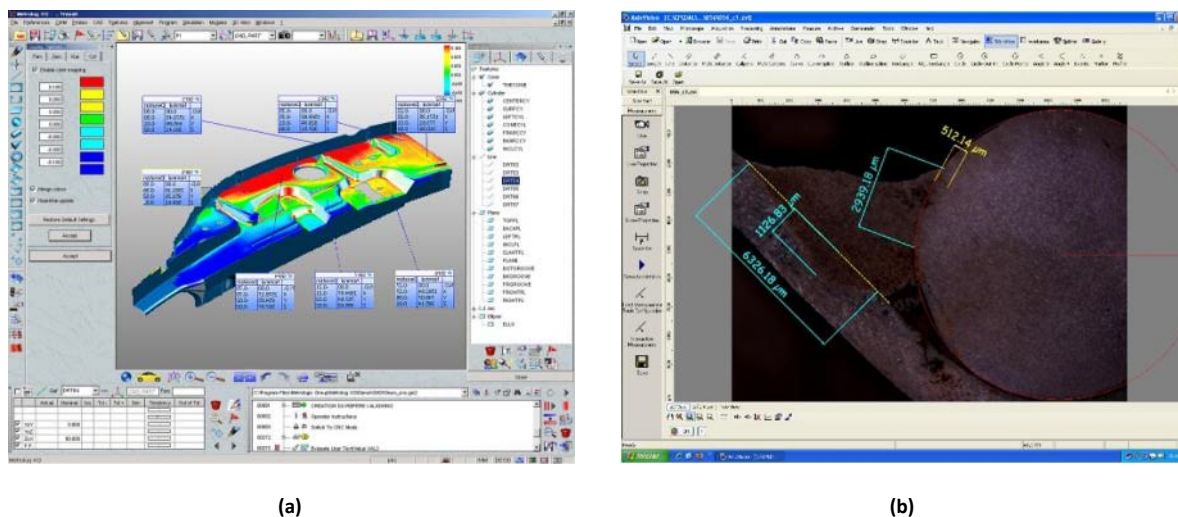


fig. 12 – Ambiente de trabalho da aplicação (a) Metrolog XG (b) AxioVision

Através de destes equipamentos e das suas aplicações torna-se possível realizar análises dimensionais complexas em menor tempo, calcular desvios de superfícies e visualizar qual é o seu desvio relativamente à nominal (Metrolog), e calcular penetrações de soldadura através do AxioVision *online* e *offline*.

2.2 IMPLEMENTAÇÕES EXISTENTES

Ao iniciar a procura de sistemas existentes ou soluções industriais houve a necessidade de definir um conjunto de propriedades que o sistema deve conter, como a cadência.

Numa indústria metalomecânica composta por unidades de estampagem, conformação de tubos, arames (fio trefilado) e soldadura robotizada, verifica-se que a cadência média de um

processo produtivo varia entre as 20 e 30 peças por minuto. Nesta fase, optou-se por considerar que o sistema estaria fora da linha produtiva e que poderia operar a uma cadência mínima de 20 peças por minuto. Com base em dados de uma PME portuguesa, verifica-se que a cadência máxima anual ronda as 800000 peças, o que equivale a produzir 20000 peças por semana (Epedal, 2006). Admitindo que o sistema de inspeção possui uma cadência de 20 pçs/min é possível garantir a análise do stock de uma semana em apenas duas horas. A necessidade de executar um controlo a 100% é pouco incidente levando a que em média não exista mais do que 3 a 4 peças distintas nesta situação.

O sistema a procurar deverá permitir o controlo de peças sem necessidade de contacto, sem pré posicionamento da peça para a sua análise. Além destas duas condições o sistema não deverá depender do fornecedor, ou seja, deverá ser normalizado para uma fácil substituição.

2.2.1 SISTEMAS SEMI-AUTOMÁTICOS

Os sistemas por actuação são as soluções mais utilizadas na indústria automóvel, o recurso a estes sistemas aparece devido à facilidade na obtenção de componentes de substituição em caso de avaria ou má selecção, ou a não dependência de características de nenhum construtor devido à grande uniformização destes produtos no mercado.

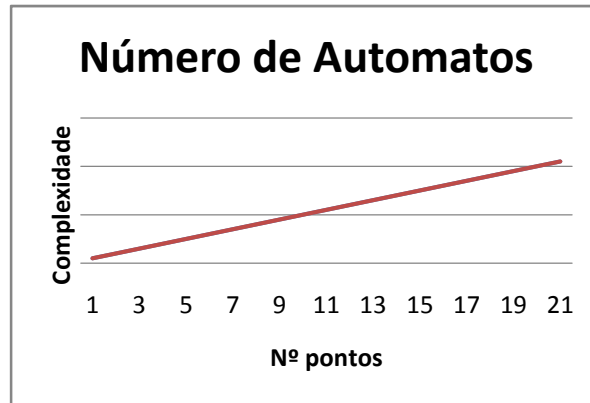


fig. 13 – Evolução do número de autómatos

Actualmente, estes sistemas de inspecção por actuação são os mais usados devido à existência de muita informação sobre a sua utilização e aplicação, e claro, no baixo custo de desenvolvimento. Estes sistemas são normalmente formados por um conjunto de sensores e actuadores. Os sensores mais aplicados nestas soluções são os electromagnéticos, mecânicos e os ópticos.

O número e ou o tipo de sensores é função das características a controlar. No exemplo da fig. 14 verifica-se o recurso a sensores electromagnéticos que estão localizados no cilindro pneumático das hastes. O peso das peças faz com que o cilindro desça e assim que é accionado o sensor inicia-se o ciclo, as hastes são recolhidas e abre a comporta de acesso ao contentor. Este equipamento garante que apenas que sejam colocadas peças com furos dentro dos contentores (evitar a falha humana).

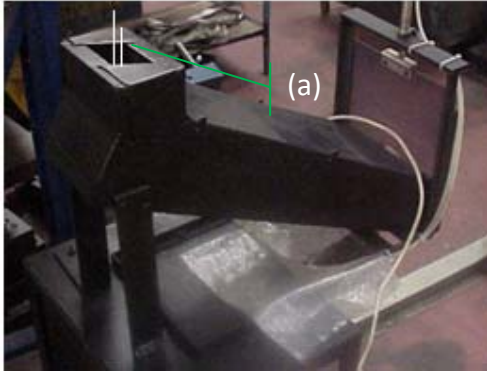


fig. 14 – exemplo de um autómato de inspecção

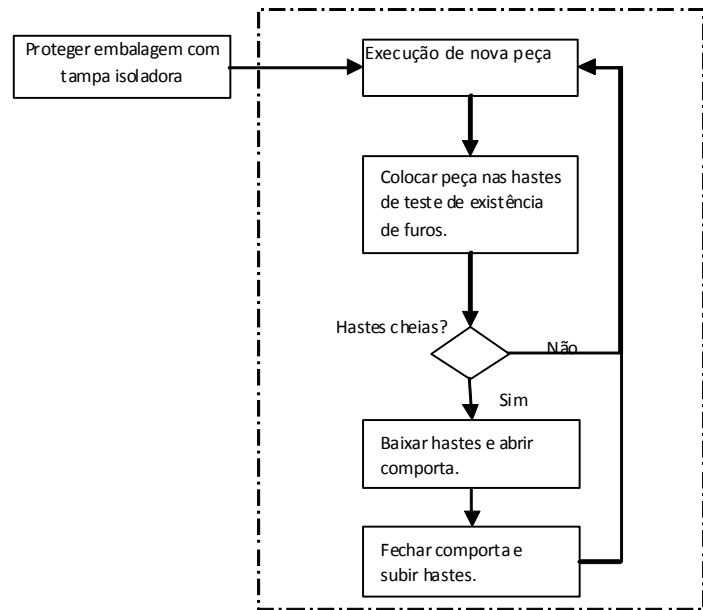


fig. 15 – fluxograma de funcionamento de um autómato

O recurso a estes sistemas, apesar de ser comum, começa a entrar em desuso devido a diversos factores, como o aumento da complexidade com o número de características a analisar fig. 14, e aumento do número destes sistemas com o número de peças ou de características a analisar fig. 13.

2.2.2 VISÃO INDUSTRIAL

A visão é uma tecnologia que têm vindo a ter um grande desenvolvimento desde os anos 70. No entanto só durante o fim dos anos 80 inicio dos 90 é que a sua aplicação começou a ser mais ampla, com a sua aplicação nos satélites de observação(Horst Haußecker, 1999).

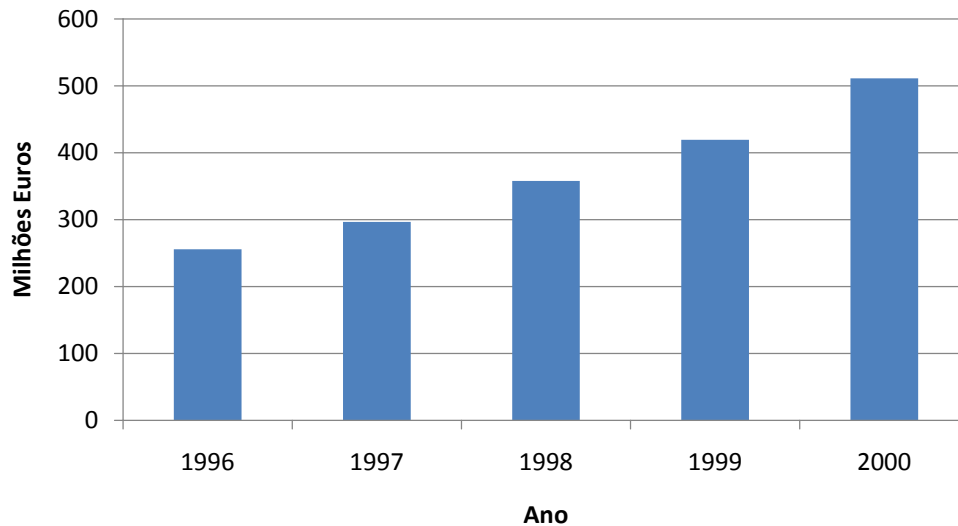


fig. 16 – Investimento aplicado na área da visão industrial no período de 1996 a 2000 (Horst Haußecker, 1999)

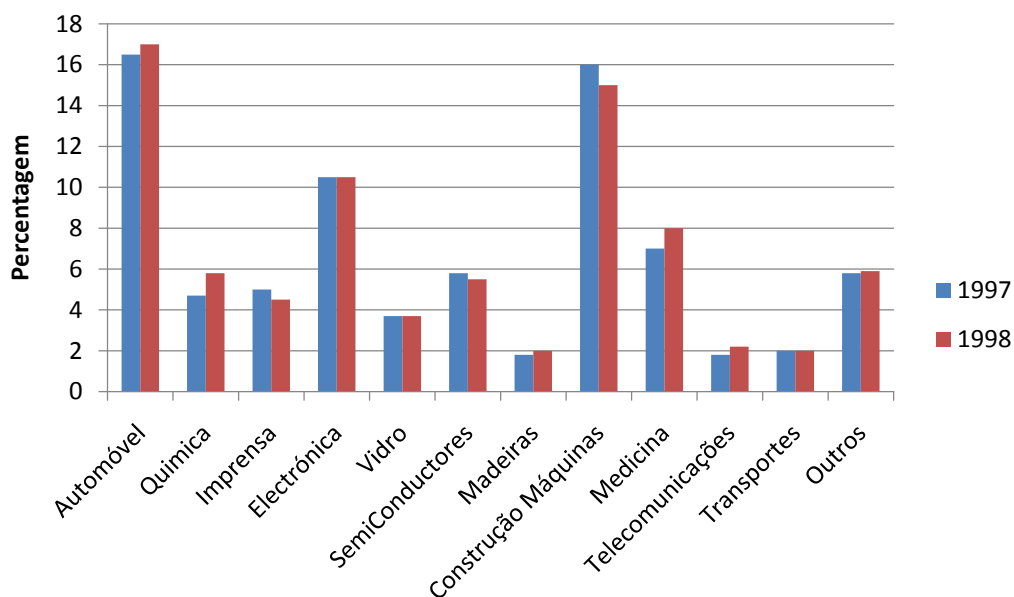


fig. 17 – Tipo de Empresas que investiram em sistemas de visão (Horst Haußecker, 1999)

No gráfico da fig. 17 verifica-se que no mercado alemão entre 1997 e 1998 houve um investimento médio de 7% na área da visão industrial, mas a industria que mais procurou soluções nesta área foi a indústria automóvel.

Sendo a indústria Alemã uma das que apresenta mais sucesso na Europa, verifica-se que o investimento em sistemas de visão teve um aumento constante no período de 1996 a 2000 (Horst Haußecker, 1999) (The Machine Vision Market: 1997 results and forecasts through 2002, 1998).

Nos últimos anos, os grandes saltos tecnológicos a nível de hardware permitiram que os resultados obtidos na área da visão artificial aumentassem significativamente. Principalmente devido ao PC comum (com unidades de processamentos com velocidades de relógio na ordem do 3GHz) tal como às câmaras “inteligentes” (unidades de processamento local com unidades I/O).



fig. 18 – exemplo de câmaras inteligentes

Para além das evoluções significativas a nível de hardware as facilidades para o utilizador aumentaram drasticamente ao nível da sua programação, permitindo assim a sua aplicação em áreas dispersas recorrendo a plataformas normalizadas com soluções amplas. Alguns desses exemplos estão indicados na fig. 19 com as suas *interfaces* gráficas. As aplicações com mais impacto neste mercado são o Insigth (Cognex), NeuroCheck e Sherlock. Estas aplicações são recorrentemente usadas em sistemas de Controlo Qualidade e são de fácil aprendizagem e implementação, todas são *Windows-based* para uma melhor interação com o utilizador.

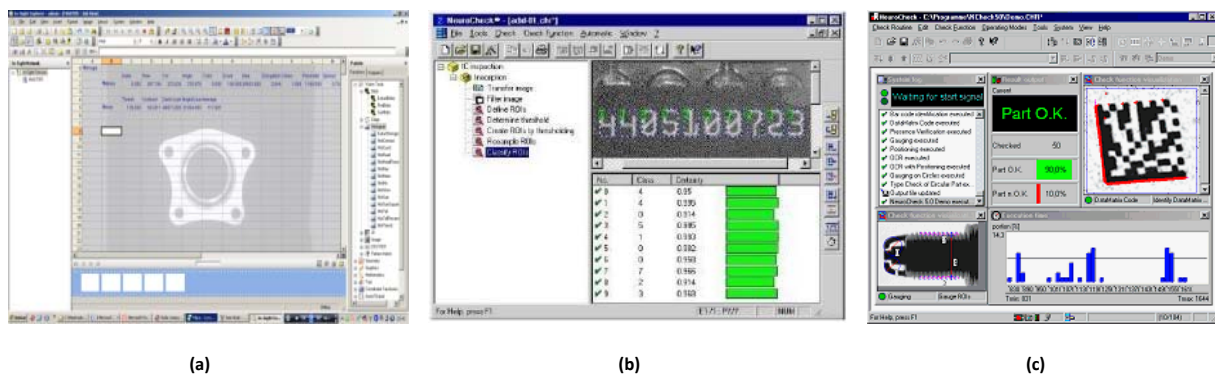


fig. 19- Interfaces do Insigth(a), Sherlock(b) e do Neurocheck(c)

As três aplicações proporcionam o mesmo tipo de ferramentas, definição de zonas de interesse (ROI), detecção de manchas e arestas, detecção de caracteres e reconhecimento de formas, tal como permitem definir um conjunto de operações a executar a cada ciclo de inspeção. O conjunto de operações deve ser realizado com a seguinte ordem; aquisição da imagem, seguida de todas as outras instruções até atingir o resultado final (decisão).

A definição dos ROI's durante o percurso de inspeção tem o intuito de diminuir as áreas de busca por parte dos algoritmos de detecção de manchas, arestas e formas.

A detecção de arestas e manchas funciona apenas quando é fornecido um ROI que deve ser inferior ao tamanho da imagem original. (NeuroCheck) (Sherlock) (Cognex)

O detector de formas ou *Template Match* é aplicado na maioria das soluções industriais como um complemento de decisão. O funcionamento deste modelo (*template match*) ocorre em dois passos básicos (Horst Haußecker, 1999).

- 1- Definição do modelo;
- 2- Busca e detecção do modelo na imagem de teste.

É exemplo a detecção de caracteres, em que o *Template Match* pode ser evidenciado pela passagem pelos seguintes passos; primeiro criar uma biblioteca de modelos, como o exemplo da fig. 20, e depois iniciar o processo de teste.

Este processo de reconhecimento apresenta bons resultados se as imagens a testar forem do tipo da fig. 21. Este método por si só tem algumas desvantagens, como a sensibilidade a distorções e a ruídos, sensibilidade à escala, e sensibilidade à rotação. Mas permite indicar qual é o grau de certeza que traz para a decisão.

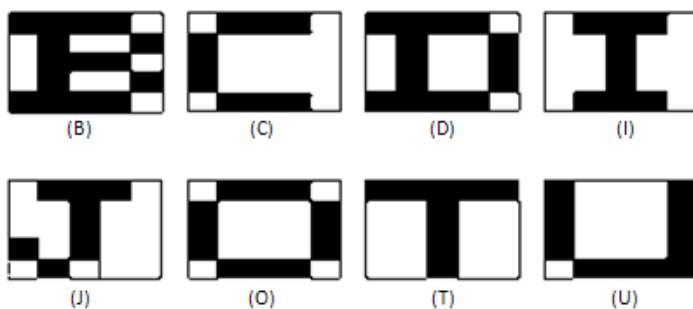


fig. 20 – Biblioteca de imagens modelo



fig. 21 – Imagens teste simples

Cada uma das aplicações tem as suas próprias funções de detecção de formas mas todas elas se baseiam na ideologia apresentada, apresentando as mesmas desvantagens.

No Sherlock e no Neurocheck a ordem das operações é importante e deve ser sequencial (desenvolvidas com base numa árvore sequencial de operações.), visto que o resultado na instrução anterior será um dado de entrada na instrução seguinte.

A ideologia implementada no Insigth é muito diferente das aplicações anteriores e permite que o utilizador não tenha de se preocupar com a ordem das operações, ou seja, o Insigth é composto por uma folha de cálculo (tipo Excel) e em cada célula pode ser colocado um valor de qualquer tipo (imagem, mancha ou outro) podendo ser um valor estático ou o resultado de uma função.

Todas as aplicações anteriores obrigam a que o utilizador identifique o sistema de eixos, ou seja, reorientar das coordenadas câmara para as coordenadas peça. A definição do sistema de eixos é realizada através da detecção de entidades (furos, arestas, etc), contudo a detecção das entidades possui um conjunto de restrições, as quais obrigam a que a peça a testar apareça com uma orientação que nunca varie mais de 10º e numa posição circunscrita na ROI de busca, em relação à imagem modelo. Esta situação leva a restrições tanto ao nível de execução das macros de inspecção como ao nível do equipamento complementar onde a câmara é colocada e na alimentação das peças a controlar.

A visão industrial apresenta ser a solução mais adequada devido às características indicadas neste capítulo e o exemplo da indústria Alemã, a qual necessita de diminuir também os seus custos. No entanto, as aplicações de desenvolvimento actuais apresentam algumas vantagens e desvantagens.

O Sherlock e o Neurocheck funcionam com um conjunto de câmaras, enquanto o Insigth apenas funciona com câmaras Cognex. Das três aplicações o Insigth apresenta ser a mais promissora devido a facilidade de utilização e ausência de regras na ordem das operações, contudo todas obrigam a que as peças a inspeccionar apareçam sempre numa posição semelhante ao modelo. O problema principal destas três soluções é a obrigatoriedade em usar câmaras de um conjunto limitado de fornecedores e a quantidade de alimentadores distintos.

A limitação das câmaras depende da aplicação, o Insigth apenas funciona com câmaras Cognex porque a aplicação precisa de validar o código de identificação da câmara, as quais permitem a execução de código local, o NeuroCheck e o Sherlock funcionam com quase todo o tipo de câmaras no entanto os fabricantes aconselham um conjunto de fornecedores.

A diversidade de alimentadores necessária leva a uma situação semelhante ao dos autómatos de inspecção, torna-se necessário desenvolver uma solução que proporcione a redução dos alimentadores.

3 SOLUÇÃO PROPOSTA

Conforme constatado no capítulo 2, verifica-se que as maiorias das soluções de visão existentes necessitam de diversos sistemas de alimentação para garantirem o posicionamento das peças, de modo a facilitar a aplicação do *Template Match*.

Para minimizar ou contornar estas desvantagens propõe-se o Reconhecimento Paramétrico (RP), que é um método que pretende diminuir algumas dificuldades existentes no método anterior, nomeadamente o problema da rotação, e sensibilidade a distorções. Mas também poderá permitir integrar o *Template Match* como sendo um dos passos do RP. Reduzindo ou evitando-se o problema da rotação é possível reduzir o número de equipamentos de alimentação.

Este método passa por uma abordagem geométrica e topológica da peça, ou seja, tal como em Metrologia devem ser seleccionadas como referências um conjunto de entidades geométricas que trazem menores desvios para o sistema de alinhamento, também neste método deverão ser seleccionadas as entidades geométricas com menor variabilidade possível. Neste caso, a variabilidade não está associada ao processo de fabrico mas sim aos efeitos ópticos da imagem, como é o caso da escala e da rotação. No decorrer deste trabalho o termo entidade irá aparecer como uma simplificação de entidade geométrica.

3.1 CONDIÇÕES DE FUNCIONAMENTO

Nem todas as falhas do *Template Match* são possíveis de evitar e dessa forma foi necessário definir quais as condições necessárias a ter para diminuir essas falhas.

As condições de funcionamento obrigatórias a definir são:

1. A câmara deverá estar numa posição fixa, durante todo o processo de inspecção a posição da câmara deverá ser mantida;
2. A câmara deverá estar numa posição paralela ao tapete de alimentação e à distância necessária para verificar todas as características pretendidas;
3. O contraste entre o objecto e o fundo deve ser elevado, este contraste deve ser realizado através da iluminação e da cor do fundo.

3.2 PRÉ-PROCESSAMENTO DE IMAGEM

Este capítulo irá reflectir o pré-processamento necessário a realizar nas imagens. Este pré-processamento é fundamental para a análise futura a desencadear, sendo óbvio que um mau pré-processamento irá invalidar toda a posterior análise mais complexa. Todos os métodos de seguida indicados tanto serão úteis nesta área como a industria metalomecânica como para a medicina entre outras(Davies, 1997).



fig. 22 – Imagem para Pré-Processamento (Gancho 1)

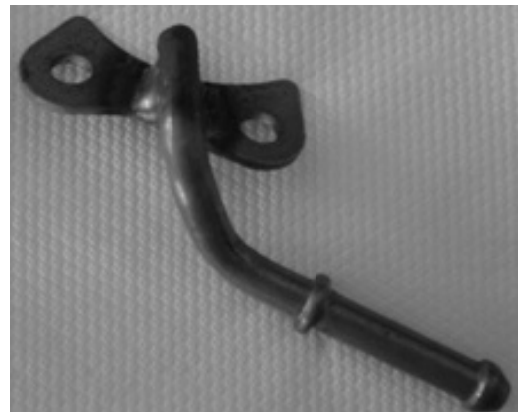


fig. 23 – Gancho 1 na escala de cinzentos

3.2.1 ESCALA DE CINZENTOS

A imagem da fig. 22 encontra-se na escala RGB de cor, a qual está representada em três canais (vermelho –R, Verde – G e Azul – B) terminologia que advém da literatura inglesa.

Quando é necessário que um determinado objecto tenha peso numa determinada cena a cor desse objecto torna-se um factor enorme a ter em conta(Davies, 1997). Na maioria das aplicações de visão os custos de processamento adicionais não são relevantes.

Observando a imagem da fig. 22 e da fig. 23 verifica-se que o gancho continua visível e perfeitamente definido.

3.2.2 BINARIZAÇÃO

Uma imagem é sempre representada por uma matriz na memória do computador. Para uma imagem de 8 bits (escala de cinzentos) como a imagem da fig. 23 a matriz contém valores que variam de 0 a 255. Encontrar o gancho na fig. 23 para um ser humano é simples, mas para o computador torna-se mais difícil. Contudo apenas se pode tomar decisões depois de encontrar o objecto, para isso é necessário separar o fundo do objecto. Com base na fig. 23 verifica-se que o gancho é mais escuro que o resto da imagem, dessa forma pode-se supor que tudo o que é mais claro é o fundo.

Dessa forma, selecciona-se na matriz (imagem) o menor valor que representa o objecto (gancho). Este é o passo que realiza a primeira distinção do que poderá ser o objecto ou fundo, em que os pixels brancos são considerados o fundo da imagem e os pretos o objecto.

Apesar das inúmeras técnicas para a selecção e escolha do limite de binarização, neste trabalho, numa primeira fase, optou-se pelo método da selecção do limite. Nesta situação é atribuído um valor ao limite de binarização o qual decide se o pixel x_i é fundo ou objecto (parâmetro válido para toda a imagem). A separação do objecto do fundo exemplificado pela fig. 24.



fig. 24 – Gancho1 Binarizado

3.2.3 PREENCHIMENTO

O algoritmo de preenchimento permite eliminar informação da imagem possibilitando detectar outros dados, como por exemplo os furos do gancho da fig. 24, e também permite diminuir ruído após a binarização da imagem.

Para detectar os furos, será necessário procurar uma passagem de preto a branco, esta busca seria feita pixel a pixel e não seriam encontrados apenas dois furos mas centenas deles devido ao ruído. Para isso, é necessário diminuir o ruído e fundir o fundo com a peça mas conservando os furos. Esta operação é efectuada através do preenchimento da

imagem. Este preenchimento é realizado pela propagação de uma semente, o pixel semente escolhido é o pixel P_0 , este pixel encontra-se na linha e coluna 0 da imagem. Para evitar problemas na propagação do preenchimento, ou seja, para garantir que toda a imagem é preenchida coloca-se uma moldura de espessura de um pixel à volta de toda a imagem e inicia-se a propagação com base no pixel semente (fig. 25).



fig. 25 – Moldura exibida a preto apenas como exemplo



fig. 26 – Furos existentes no Gancho 1

3.3 DETECÇÃO DE ENTIDADES NA IMAGEM

O RP baseia-se na definição de um modelo (objecto) através de entidades invariantes (básicas). Tendo como exemplo a imagem da (fig. 27 (a)) as entidades básicas a detectar são todas aquelas que independentemente da sua posição ou ângulo de visualização estão sempre presentes.

Uma entidade é um elemento que apenas existe no objecto da imagem, desde que o nível de ruído seja baixo.

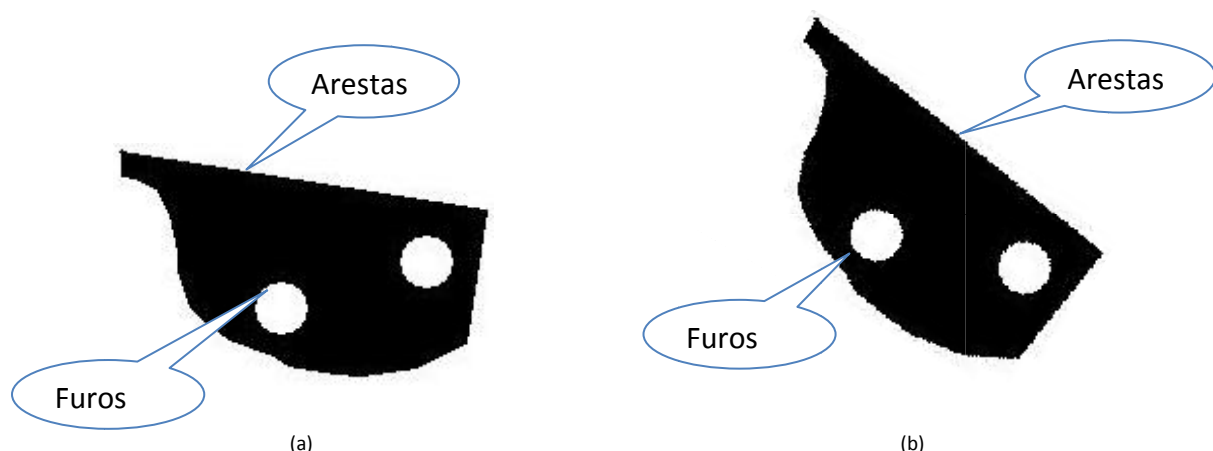


fig. 27 – Entidades existentes¹, (a) exemplo 1, (b) Exemplo 1 rodado

¹ Esta imagem encontra-se já pré-processada.

Apenas são enunciadas as arestas e os furos na fig. 27, porque estas são as entidades mais básicas detectadas, poderíamos enunciar por exemplo os cantos, mas um canto não é nada mais do que um caso concreto da união de dois segmentos de recta, além de que a detecção cantos é uma entidade de fácil implementação posteriormente.

Todos os passos de detecção de entidades são realizados após as operações de pré-processamento.

Para todas as entidades é necessário definir propriedades. As propriedades foram divididas em dois tipos, **singulares** e **relativas**. As propriedades são elementos com as quais é possível descrever as entidades.

As propriedades singulares são todas aquelas que apenas dependem da própria entidade como exemplificado na Tabela 2.

Tabela 2 – Propriedades singulares de algumas entidades

Entidade	Propriedade 1	Propriedade 2	Propriedade 3
Circulo	Diâmetro	Centro geométrico	n/a
Segmento Recta	Comprimento	Coordenadas	Ângulo

As propriedades relativas, como o nome indica, são propriedades construídas com base em propriedades singulares das entidades, ou seja, relações entre entidades, como os exemplos dados na Tabela 3.

Tabela 3 – Exemplos de propriedades relativas, em que $i \neq j$

Entidade 1	Entidade 2	Propriedade Relativa
Circulo _i	Circulo _j	Distância
Linha _i	Linha _j	Ângulo
Circulo	Linha	Distância
Rectângulo	Linha	Ângulo
Rectângulo	Linha	Distância
Rectângulo	Circulo	Distância

Todas as entidades e propriedades indicadas na Tabela 2 e Tabela 3 não implicam a sua utilização no decorrer deste trabalho. Mas são enunciadas como meio de facilitar a percepção do método.

3.3.1 DETECÇÃO DE ARESTAS

Dependendo do fim a que se destina, a detecção de arestas pode ser tida como o fim ou pré-processamento para passos subsequentes. Neste trabalho, a detecção de arestas pode-se considerar num nível intermédio de processamento. De qualquer forma, independentemente dos resultados desejados, é necessário que a estratégia de detecção de arestas seja eficiente. Porém, a imagem a usar para a detecção de arestas deve estar na escala de cinzentos, imagens na escala de cinzentos trazem algum ruído levando à detecção de más arestas (arestas erradas).

Para que as más arestas, provenientes de ruído ou textura da imagem, não sejam detectadas, deve-se suavizar (uniformizar) a imagem antes da detecção. Contudo, existem efeitos inoportunos ligados à uniformização, i. e., perda de informação. Além disso, existem diferenças entre as propriedades dos operadores diferenciais utilizados na generalidade, ocasionando arestas distintas. Logo, é difícil formular um algoritmo de detecção de arestas que possua um bom desempenho em diferenciados contextos e capture os requisitos necessários aos estágios subsequentes de processamento (ZIOU, et al., 1998).

Relativamente ao processamento de imagem, uma variedade de detectores de arestas tem sido desenvolvidos com diferentes objectivos, com formulações matemáticas diferenciadas e com propriedades algorítmicas distintas (VALE, et al., 2002)

Com base nos problemas acima mencionados, Canny desenvolveu um processo de detecção de arestas a partir de detectores de arestas conhecidos como os trabalhos de Marr e Hildreth (Canny, 1986).

A vantagem que este método oferece é a identificação e redução de informação sem perder as características identificativas do objecto (Green, 2002). A conclusão fundamental do trabalho de Canny foi a definição de um conjunto de critérios a ter em conta para melhorar os métodos existentes de detecção de arestas tal como a passagem por 6 etapas (Green, 2002). O algoritmo de Canny, é indicado por muitos como sendo o melhor método para a detecção de arestas (Green, 2002) (Fisher, et al., 1994).

Na fig. 28 estão representadas as arestas detectadas pela técnica de Canny, tendo-se recorrido ao algoritmo existente no OpenCV.

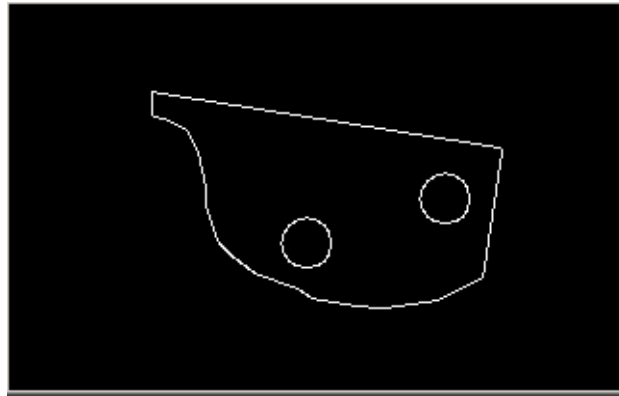


fig. 28 – Arestas Detectadas pelo detector de Canny

3.3.2 RECTAS

A detecção de arestas por si só não terá grande utilidade para a definição do objecto, porque o método de detecção de arestas indica todos os contornos do objecto (cantos, linhas recta e curvas).

A imagem da fig. 28 exemplifica a dificuldade de usar as arestas indicadas para a definição do modelo, mas recuando aos aspectos de metrologia mencionados no capítulo 2.1, o alinhamento de medição de uma peça, regra geral, é definido através de furos (círculos), planos e rectas, então pode-se considerar uma boa ideia reduzir as arestas apenas às linhas rectas como arestas importantes para este método.

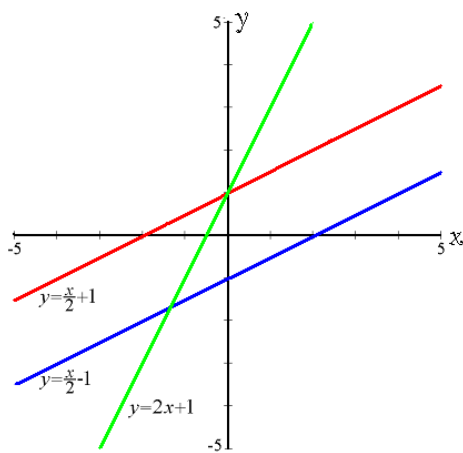


fig. 29 – Representação gráfica de três rectas, em que o declive é o mesmo na recta vermelha e azul.

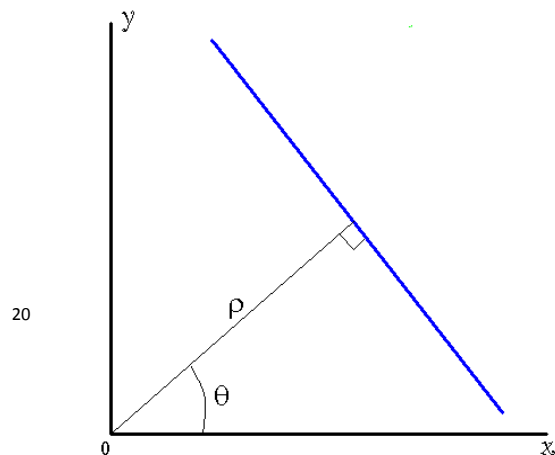


fig. 30 – Representação de uma linha recta através de uma parametrização por coordenadas polares (ρ, θ)

Uma recta num plano cartesiano pode ser definida algebricamente através de:

$$y = mx + b \quad \text{Eq. 1}$$

em que m é o declive da recta e b o ponto de intersecção com o eixo das ordenadas.

Com base na definição anterior e no seu formalismo matemático, é evidente a possibilidade de descrever rectas com base nos pontos das arestas pelo detector de Canny.

Em 1962, dá-se um grande passo para a detecção de arestas rectas com a apresentação do método mais popular para a detecção de rectas em arestas; as transformações de Hough (Davies, 1997). Actualmente, o método original de Hough é tido como inadequado à computação, visto que princípios mais rápidos foram desenvolvidos (Davies, 1997).

A forma como o método de Hough foi inicialmente apresentado envolvia linhas parametrizáveis de acordo com a Eq. 1. Duda e Hart em 1972, indicam que a substituição da Eq. 1 por:

$$\rho = x \cos \theta + y \sin \theta \quad \text{Eq. 2}$$

facilita a representação de uma linha no espaço Normal (ρ, θ) e permite ganhar tempo computacional (Davies, 1997). Este princípio assenta na representação das linhas rectas que passam por cada ponto P_i , no sistema de eixos (ρ, θ) através das curvas seno. Os diversos pontos no espaço (ρ, θ) , indicam através de ρ e θ a presença de linhas na imagem original.

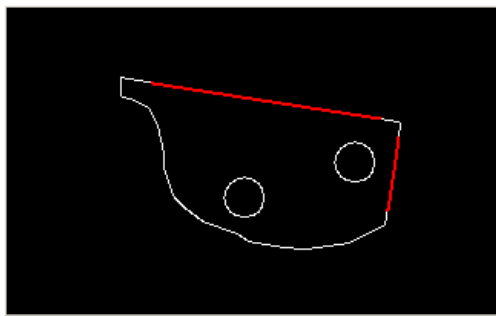


fig. 31 - Identificação de linhas existentes na imagem da fig. 28

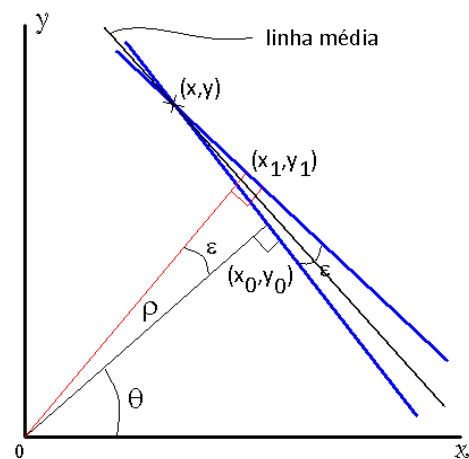


fig. 32 - Representação da recta média que melhor representa a aresta.

Contudo o grande desenvolvimento das transformações de Hough deu-se devido ao sistema de votação implementado para acumular evidências das diversas instâncias (Davies, 1997).

O método *Progressive Probabilistic Hough Transform* (PPHT), que será aplicado neste trabalho, minimiza os recursos de processamento necessários para detectar linhas, explorando a diferença entre o número de votos necessários para suportar uma correcta decisão na detecção de linhas com um número diferente de pontos (pixels inclusos). O número de pontos da linha a exercer o voto é função do comprimento da linha, é intuitivo que para uma linha longa a proporção de votos seja inferior, do que para uma linha pequena fig. 33. Para contornar esta situação, o limite de votos aceites deve ser dinâmico (Galambos, et al., 1999).

O PPHT, algoritmo proposto por (Galambos, et al., 1999), baseia-se na execução repetida dos seguintes passos. Selecção de um ponto aleatório da lista para votar. Após a votação, é testada a seguinte hipótese “Sou oriundo do ruído? O número máximo de votos na urna foi atingido?”. O número ℓ máximo de votos aceites por urna altera com o número de votos previstos, em cada actualização da urna deverá ser testado se o número de votos atingiu ℓ . Com a detecção de uma linha, os pontos que a definiram retiram o seu voto e os restantes pontos, que também definem a linha, são retirados da lista de pontos escolhidos mesmo que ainda não tenham votado.

Este algoritmo, permite a definição de outras propriedades muito atractivas (Galambos, et al., 1999). Uma linha é detectada assim que o contador da urna permitir a decisão. Permite retirar dados importantes mesmo que seja interrompido o seu ciclo. Não é necessário definir nenhum critério de paragem. Contudo, o ciclo de busca termina quando todos os pontos da lista votaram ou foram associados a uma linha. Como critérios de paragem, pode-se recorrer a restrições como o comprimento mínimo da linha, a qual pode ser verificada antes de seleccionar um novo ponto para votar.

Conforme indicado anteriormente, o número máximo de votações aceites por urna deve ser dinâmico. Por defeito, o limite ℓ é calculado pressupondo que todos os pontos são ruído (Galambos, et al., 1999).

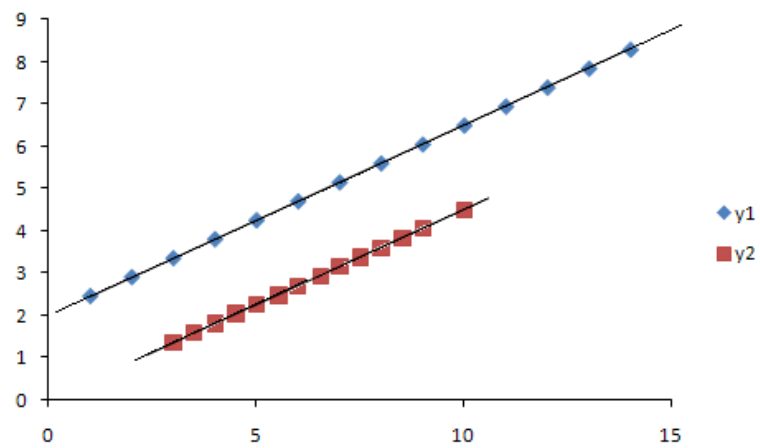


fig. 33 – Proporção de votos em função do comprimento da linha (menor para y1)

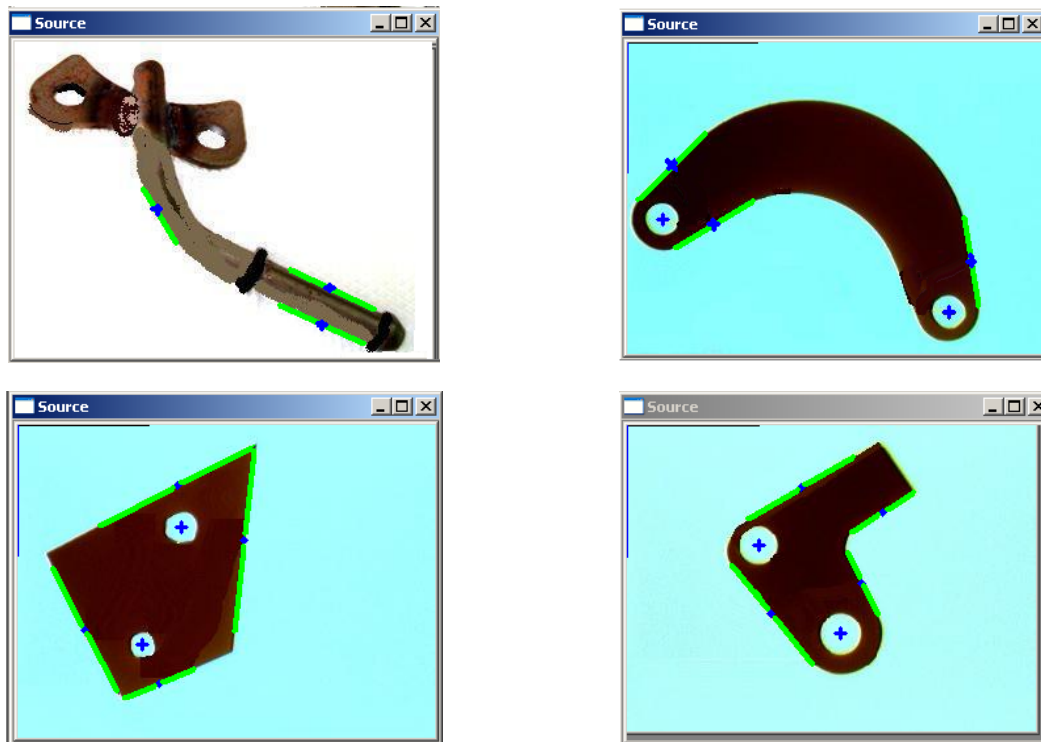


fig. 34 – Linhas recta detectadas pelo PPHT, indicadas a verde.

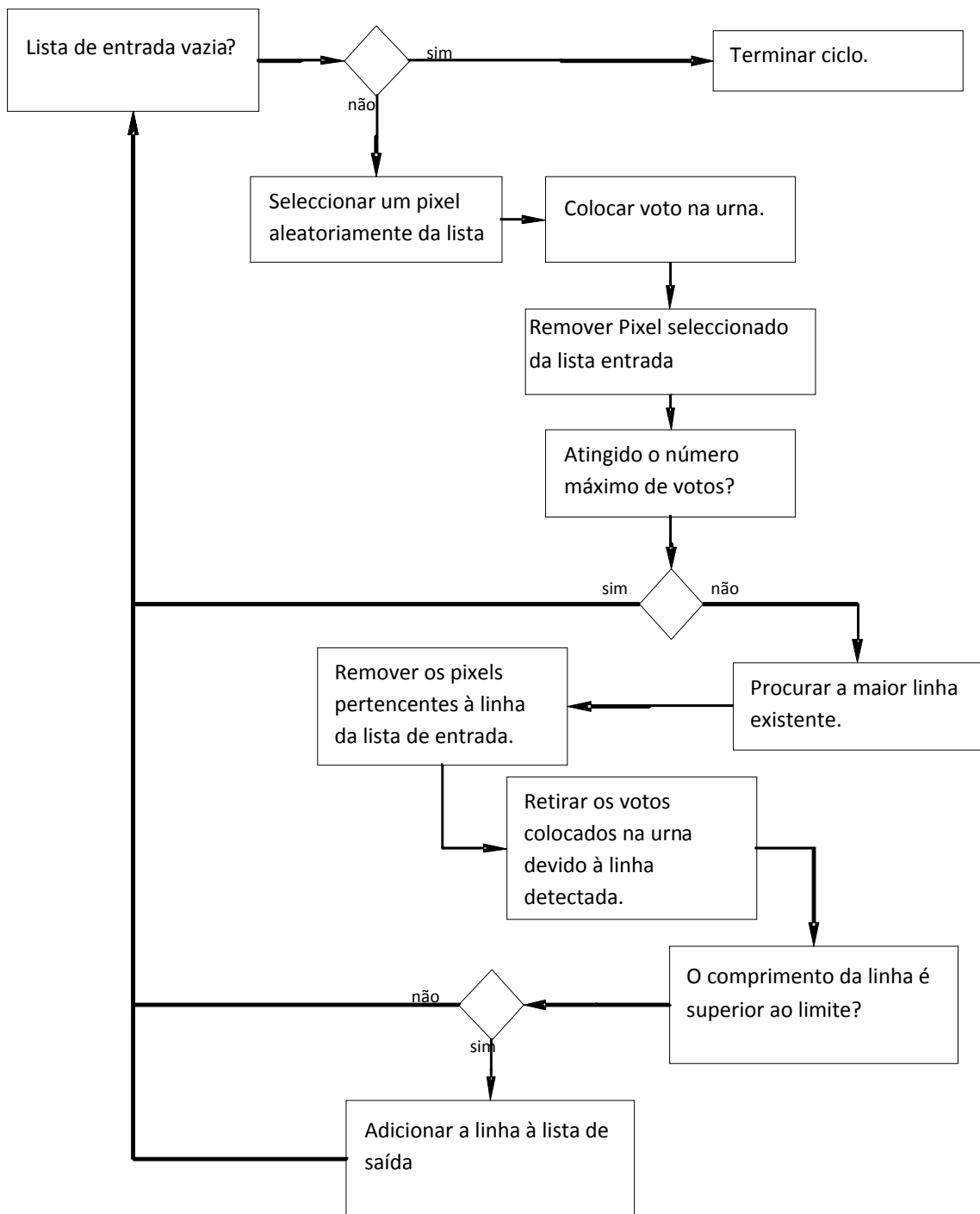


fig. 35 – diagrama fluxo do algoritmo para o PPHT no OpenCV

Este método, encontra-se especialmente adequado a aplicações em que existe necessidade de decisões em tempo-real, e em que o tempo de processamento se encontra previamente pré estabelecido (Galambos, et al., 1999).

Apesar da necessidade do estudo do PPHT não foi necessária a sua implementação, tendo-se recorrido ao algoritmo existente no OpenCV, o qual apresenta resultados semelhantes aos indicados na fig. 34.

O algoritmo implementado no OpenCV retira um Vector (**SeqLinhas**) com o conjunto de linhas detectadas. Cada posição da SeqLinhas é composta por dois dados, as coordenadas dos dois pontos que definem uma linha recta, conforme a Eq. 3.

$$[L] = \begin{bmatrix} L_0 \\ L_1 \\ L_2 \\ L_3 \\ L_4 \\ L_5 \\ \vdots \\ L_{n-1} \\ L_n \end{bmatrix}; \text{ sendo } L_i = [(x_0, y_0), (x_f, y_f)]; \text{ em que } i = \{0, \dots, n\} \quad \text{Eq. 3}$$

Com base nas coordenadas de cada L_i é possível retirar um conjunto de dados, as propriedades singulares de maior relevância para cada linha: é o caso do ângulo (declive), comprimento e o ponto âncora da entidade, fig. 36.

O ângulo é calculado relativamente ao sistema de eixos da imagem, através da Eq. 4. O comprimento por sua vez, é dado pela Eq. 5.

$$\alpha = \tan^{-1} \left(\frac{x_1 - x_0}{y_1 - y_0} \right) \quad \text{Eq. 4}$$

$$d = \sqrt{(x_f - x_0)^2 + (y_f - y_0)^2} \quad \text{Eq. 5}$$

$$(x_m, y_m) = \left(x_0 + \frac{x_f - x_0}{2}, y_0 + \frac{y_f - y_0}{2} \right) \quad \text{Eq. 6}$$

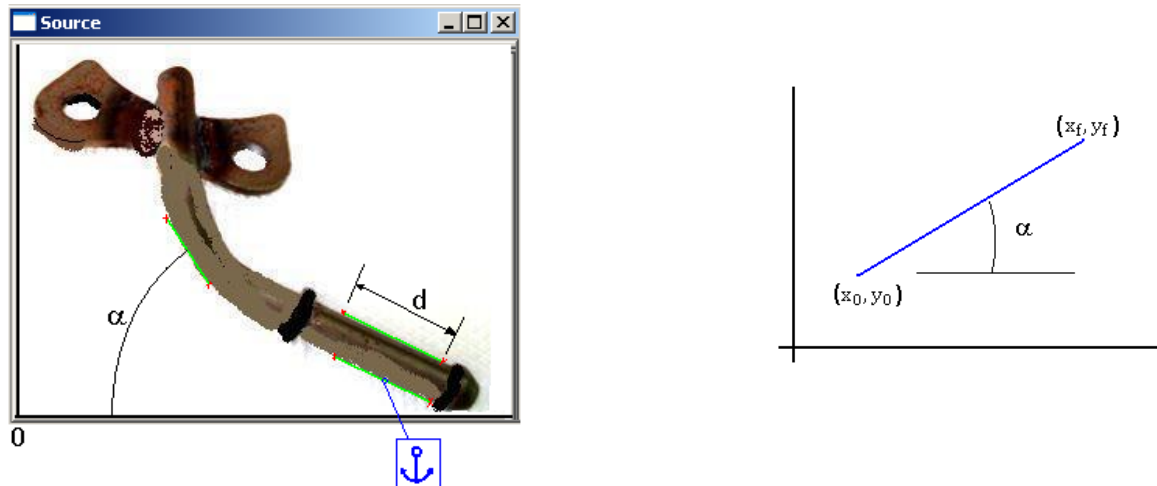


fig. 36 – Exemplo das propriedades retiradas de uma linha, relativamente ao sistema de eixos da imagem

O ponto âncora (⚓), aparece como a propriedade que permite localizar uma determinada entidade na imagem. O ponto âncora é definido como sendo o ponto médio definido pelas coordenadas da linha, de acordo com a Eq. 6.

3.3.3 FUROS

O processo de detecção de furos pode ser considerado como um processo de segmentação. A segmentação é normalmente encarada como a divisão de uma imagem em regiões menores nas quais existe uniformidade de informação (Davies, 1997). A uniformidade pode passar por diversos aspectos como a cor ou a textura, mas neste caso passa pela cor branca dos furos (fundo).

Como indicado no capítulo 3.3.1 existe um conjunto de informação retirada pelo operador de Canny. O processo de detecção a implementar, pode passar pelo recurso às transformações de Hough (HT), os quais podem ser adaptadas a qualquer forma pretendida (Wikipedia), bastando existir um prévio conhecimento das arestas (contorno) da peça a analisar (fig. 28). As arestas são obtidas pelo detector de Canny, conforme demonstrado no capítulo 3.3.1. Posto isto, verifica-se que a implementação para o reconhecimento de círculos através das transformações de Hough terá todo o sentido.

Sendo um círculo representado matematicamente através da Eq. 7.

$$(x - a)^2 + (y - b)^2 - r^2 = 0 \quad \text{Eq. 7}$$

Este modelo depende de três variáveis, duas delas definem o centro do círculo e a outra o raio. O gradiente dos ângulos de cada aresta proporciona a redução de graus de liberdade e o tamanho do espaço de busca. A direcção do vector é determinada através do gradiente do ângulo, ficando apenas o raio como uma variável desconhecida (Davies, 1997).

Convertendo para a representação polar, obtêm-se as seguintes equações.

$$x = a + r \cos q \Leftrightarrow a = x - r \cos q \quad \text{Eq. 8}$$

$$y = b + r \sin q \Leftrightarrow b = y - r \sin q \quad \text{Eq. 9}$$

Com base no gradiente do ângulo, é retirado $\cos q$ e $\sin q$ para cada ponto da aresta. Simplificando as equações, é possível deixar o cálculo do raio para depois.

$$b = a \tan q - x \tan q + y \quad \text{Eq. 10}$$

Através da Eq. 10 e do gradiente dos ângulos são encontradas as coordenadas dos centros dos círculos, calculando-se posteriormente os raios.

Através do algoritmo existente no OpenCV, adaptado a círculos são obtidos resultados como os indicados na fig. 37.

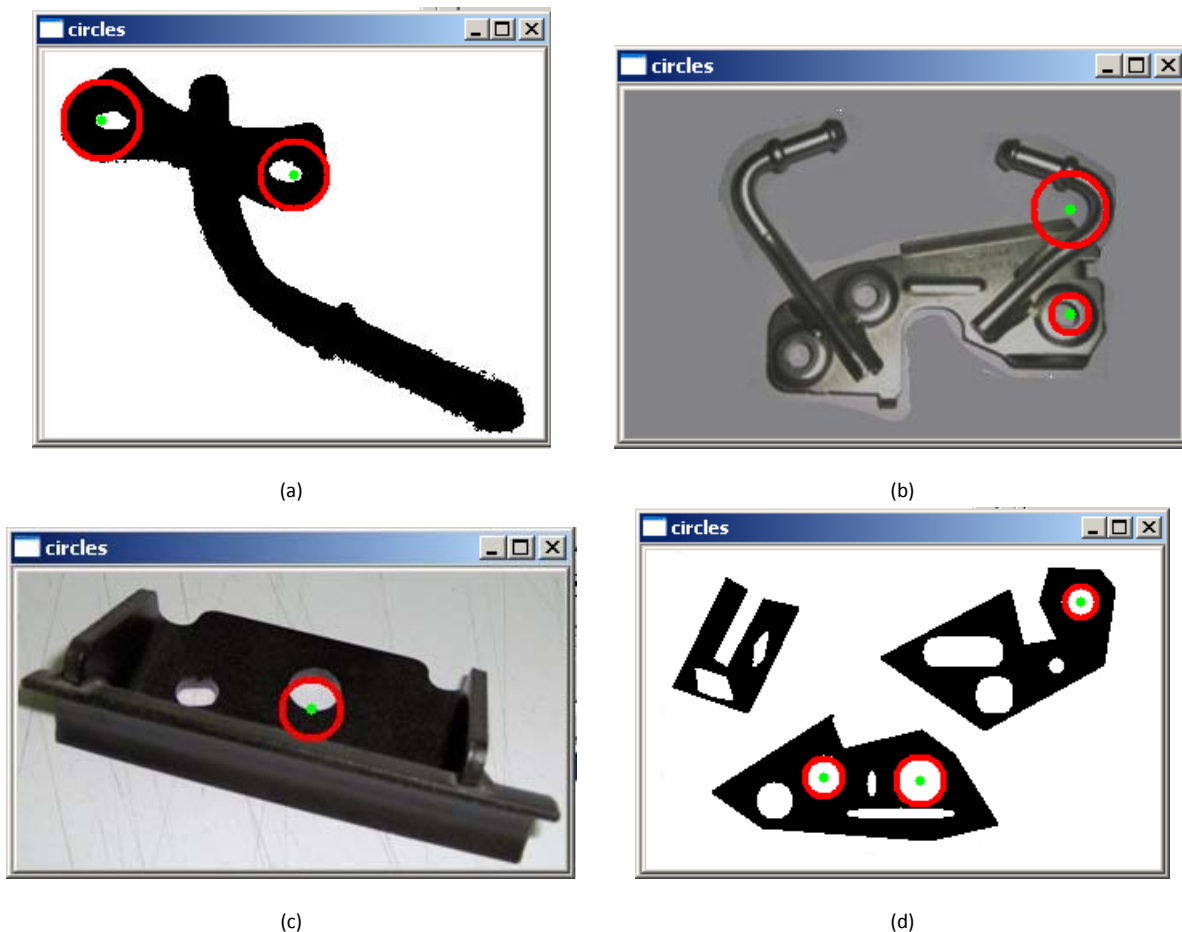
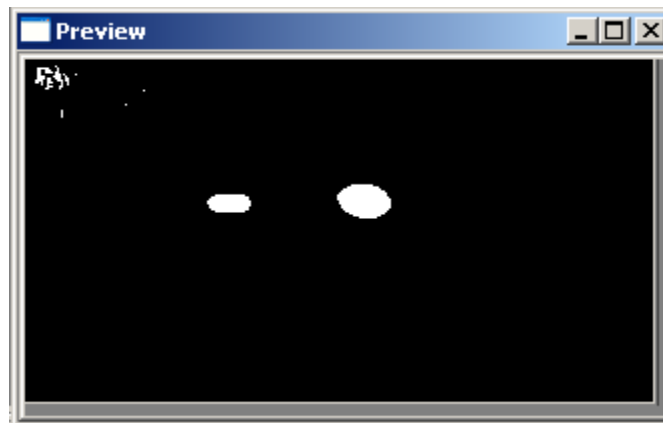


fig. 37 – Círculos detectados pelo algoritmo do OpenCV

No entanto verifica-se que nem todos os furos são detectados, além deste facto, a grande diversidade de formas existentes fig. 37(d), (como oblongos, semi-circunferência, quadrados, etc.), leva a que a seja necessário desenvolver algoritmos direccionados a cada forma.

Devido a estas circunstâncias, recorre-se a um algoritmo simples de busca de furos. Este algoritmo baseia-se em imagens com a operação de preenchimento já realizada. Desta forma, e observando a fig. 26 é simples a procura e localização dos furos. Contudo se a imagem for semelhante à da fig. 38(a) o número de furos detectados aumenta significativamente, visto que o ruído deixado pela binarização aumenta o número de furos detectados. Estes não furos, no entanto, não serão problemáticos visto que apenas irão aumentar o número de entidades detectadas, afectando apenas o tempo de processamento do processo, para evitar os não furos apenas serão aceites furos em que a sua área for superior ao valor pré-definido.



(a)



(b)

fig. 38 – (a) ruído deixado pela binarização, potenciais não furos, (b) a vermelho não furos detectados e a verde os furos

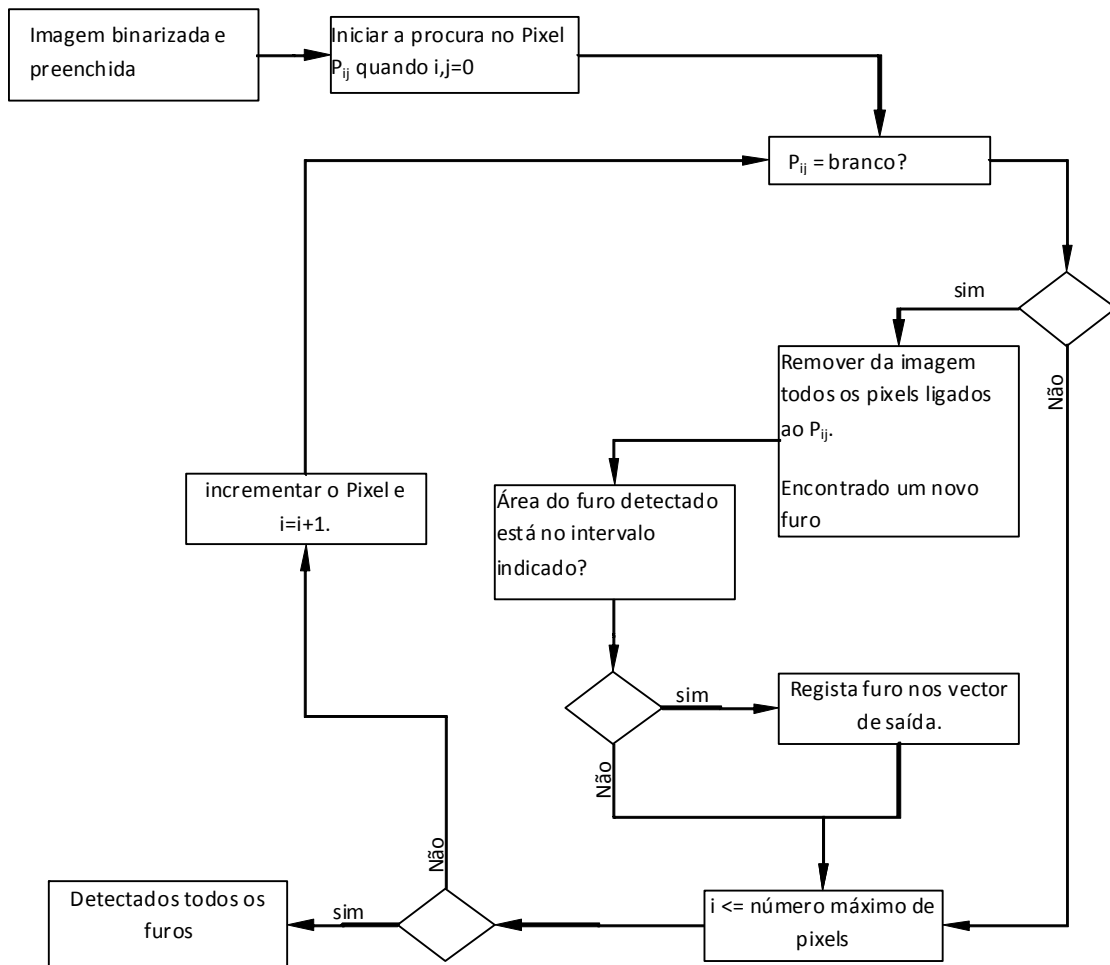


fig. 39 - Ciclo de busca dos furos (CBF)

A segmentação, detecção dos furos é realizada conforme o fluxograma da fig. 39. O algoritmo desenvolvido passa pela execução repetida dos seguintes passos, escolha sequencial do Pixel P_{ij} (i representa a linha e j a coluna da imagem (matriz)), caso seja encontrado um pixel branco, uma imagem temporária é criada e efectua-se um preenchimento com base no P_{ij} encontrado (ponto semente). Com base no passo anterior, é conhecida a área ocupada por esse furo e, caso esteja no intervalo previsto de aceitação é registada um novo furo no vector de saída (Eq. 11) e são retirados os pixels associados ao furo detectado. O ciclo termina quando todos os pixels forem inspeccionados.

$$[F] = \begin{bmatrix} L_0 \\ L_1 \\ L_2 \\ L_3 \\ L_4 \\ L_5 \\ \vdots \\ L_{n-1} \\ L_n \end{bmatrix} \text{ sendo } L_i = \{\text{Área}, (x_c, y_c)\}, \text{ em que } i = 0, 1, \dots, n - 1, n \quad \text{Eq. 11}$$

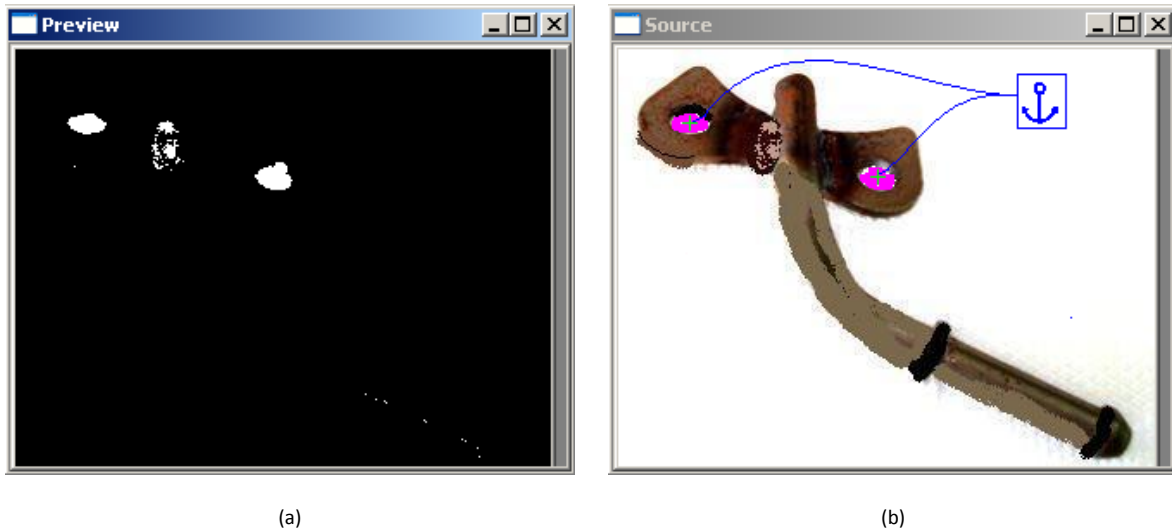


fig. 40 – (a) imagem preenchida com o ruído (b) identificação de dois furos por parte do algoritmo (área a rosa e âncora a verde)

O ruído da fig. 40 (a) é tratado de acordo com o fluxograma da fig. 39. O retorno deste algoritmo é o ponto âncora (coordenadas do centro geométrico) do furo e a área (nº pixels).

3.4 DEFINIÇÃO DE RELAÇÕES INVARIANTES

Conforme indicado no capítulo 3.3 todas as entidades são identificadas através das suas propriedades, possibilitando assim retirar conclusões sobre a imagem.

A peça a inspeccionar pode ser colocada em qualquer posição e orientação, desta forma, é possível que a peça (imagem) a inspeccionar esteja numa das situações da fig. 41, em que nenhuma delas é semelhante à imagem referênciada.

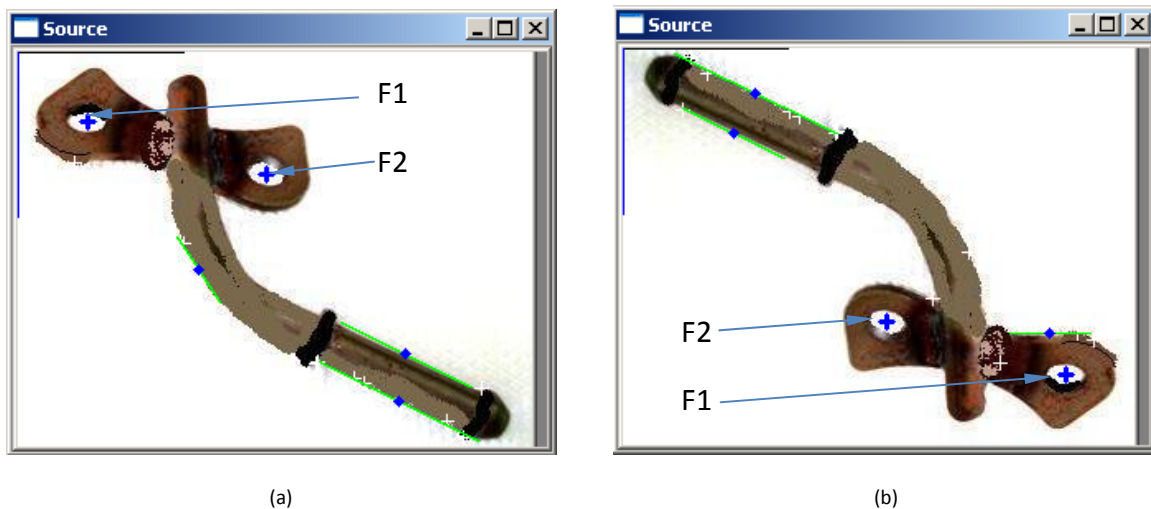


fig. 41 – Ganho 1 com a identificação das propriedades singulares de duas peças idênticas, peça (b) é a peça (a) rodada em 180°

O recurso a propriedades singulares deverá permitir identificar a peça seja qual a posição na qual ela apareça e, claro, concluir se a peça está correcta ou não. No entanto, não existe certeza absoluta na decisão dada com base apenas nas propriedades singulares.

Considere-se que a peça da imagem (a) da fig. 41 é a peça de referência a procurar, olhando para a peça (b) da fig. 41 verifica-se que a peça poderá ser a mesma. Através das propriedades singulares, dos furos na peça fig. 41 (b) e as linhas, neste caso, tanto os furos como as linhas estão correctos. Contudo, a peça como a da fig. 41 (b) está errada, porque os furos estão deslocados.

Pelo exposto, verifica-se que as propriedades singulares não permitem ter certezas sobre a peça a inspeccionar. Desta forma, e para resolver os problemas trazidos pela rotação da peça, houve a necessidade de definir propriedades invariantes na imagem.

Olhando novamente para os conceitos de metrologia, é indicado que as propriedades de referência não indicam nada, mas essas sim permitem verificar dimensionalmente propriedades que são suas dependentes. Assim, definir propriedades relativas entre entidades permitirá complementar a decisão. No entanto, estas propriedades relativas, para serem invariantes, é necessário garantir que o efeito de escala não aparece, ou seja, após a

definição de todo o sistema de inspeção a câmara não pode ser modificada, sendo necessário garantir que a câmara está localizada sempre na mesma posição, e com as mesmas propriedades do sensor.

Com base na peça (a) da fig. 41, realiza-se uma busca por propriedades relativas invariantes, conforme indicado no capítulo 3.3 todas as entidades utilizadas são definidas pelas propriedades singulares da Tabela 4 as quais deverão permitir criar entidades relativas.

Tabela 4 – propriedades singulares das entidades indicadas no capítulo 3.3

Entidade	Propriedade 1	Propriedade 2	Propriedade 3
Furo	Área	Âncora (Centro)	n/a
Linha	Comprimento	Âncora (Ponto médio)	Ângulo

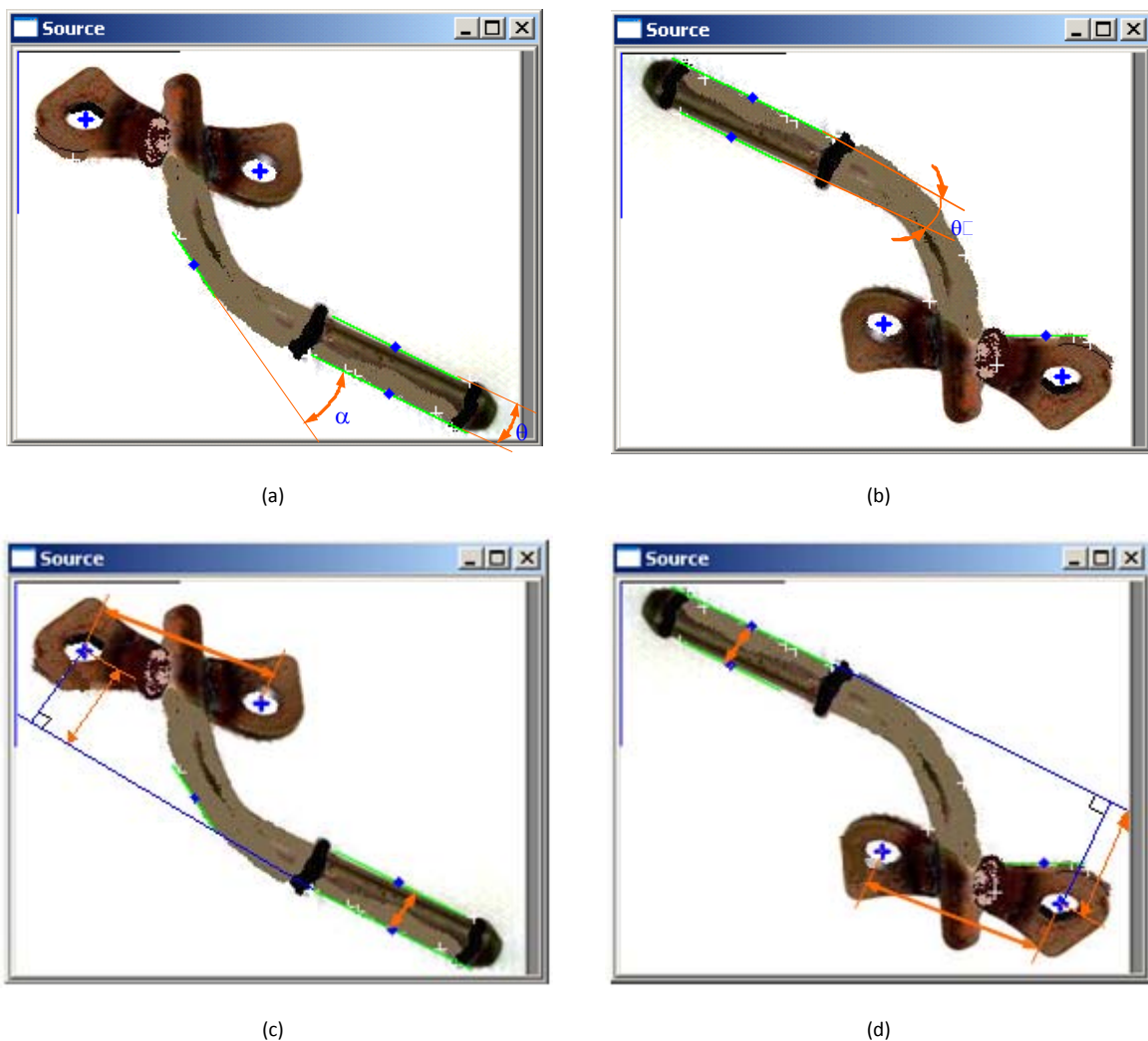


fig. 42 – Representação de propriedades relativas nas peças da fig. 41

3.4.1 DISTÂNCIAS

Uma das propriedades relativas de fácil visualização é a distância entre entidades. Esta propriedade é bastante simples, e apenas requer conhecimento de pontos fixos para que seja possível retirar a sua relação. Conforme indicado nos capítulos 3.3.2 e 3.3.3 os pontos âncora das entidades permitem que seja retirada a distância entre si.

Conforme indicado nos capítulos 3.3.2 e 3.3.3 o ponto âncora está associado sempre ao centro da entidade, a razão da escolha deste ponto âncora deve-se ao facto que a variação que este ponto irá trazer ser a menor possível, no caso dos furos. No entanto, para o par Linha/Furo, o ponto médio poderá trazer alguma variabilidade, por isso, recorre-se à distância normal (pé da perpendicular) entre o furo e a linha.

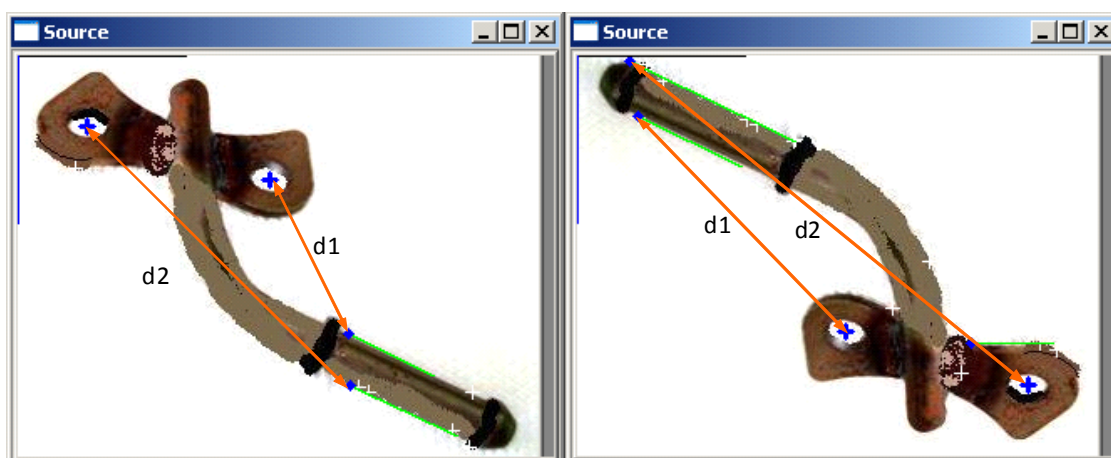


fig. 43 – variação da distância com a má escolha do ponto âncora

A imagem da fig. 43 ilustra a variação que a escolha de um dos diversos pontos poderá originar, enquanto a distância normal fig. 42 (c e d) traz uma baixa variação.

Para a Tabela 4 é possível calcular uma tabela com as propriedades relativas do tipo distância.

Tabela 5 – Matriz de propriedades relativas entre entidades

Entidade i	Entidade j
Furo	Furo
Furo	Linha
Linha	Linha

Todas as distâncias são calculadas através da Eq. 5.

$$d = \sqrt{(x_j - x_i)^2 + (y_j - y_i)^2} \quad \text{Eq. 12}$$

Em que x_i e y_i representa o ponto de intersecção entre as duas rectas e x_j e y_j representa o centro geométrico do círculo, conforme demonstra a fig. 44.

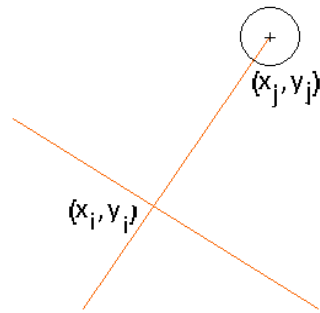


fig. 44 – distância normal (pé da perpendicular)

3.4.2 ÂNGULOS

O ângulo entre entidades é outra propriedade relativa a ter em conta, conforme ilustra a fig. 42 (a e b).

Esta propriedade relativa apenas poderá ser aplicada entre **entidades do tipo Linha**. Duas linhas que sejam concorrentes são definidas através dos seus vectores directores e o ângulo que formam entre si é a relação entre elas. Neste caso e para que uma peça esteja correcta o ângulo entre linhas deverá manter-se constante independentemente da direcção do vector director da linha, conforme ilustrado na fig. 45.

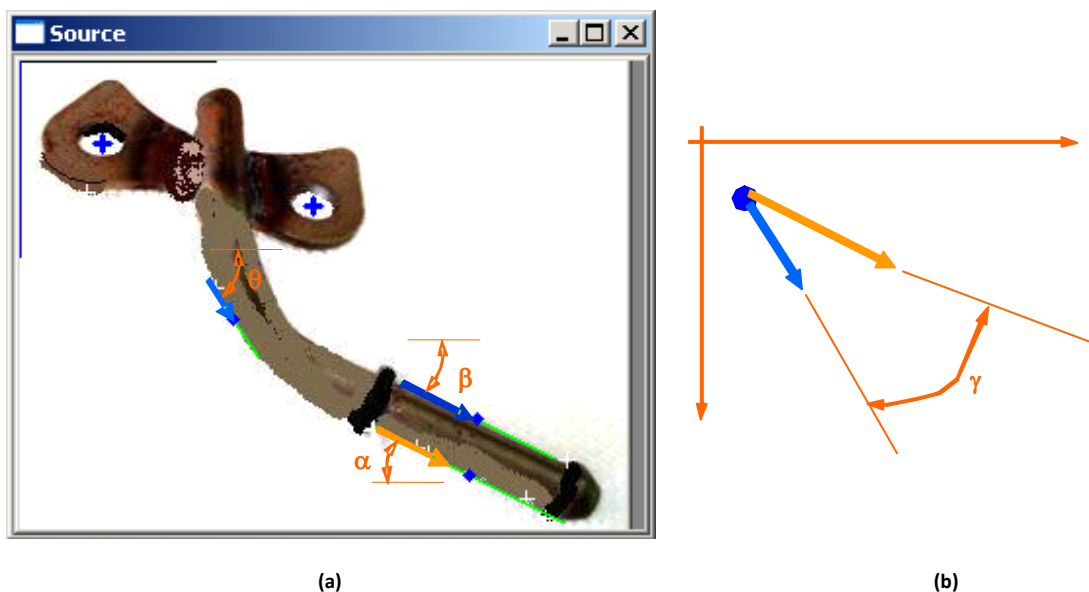


fig. 45 – (a) Representação de todos os ângulos no Gancho 1, (b) representação do ângulo entre dois vectores directores com a mesma origem

O ângulo entre linhas é calculado através da Eq. 13, em que i representa a primeira entidade e j a segunda entidade.

$$\theta = \theta_j - \theta_i \quad \text{Eq. 13}$$

Para duas linhas quaisquer, como as indicadas na fig. 45 (a) o cálculo dos ângulos entre linhas realiza-se recorrendo à Eq. 13. O ângulo γ na fig. 45 (b) representa a relação entre duas linhas.

$$\gamma = \theta - \alpha \quad \text{Eq. 14}$$

3.5 CONSTRUÇÃO DO MODELO DESCRITIVO

A construção do modelo descritivo (MD) é a definição de regras abstractas que garantam uma maior eficiência na representação de um modelo com base num conjunto de entidades e propriedades desse mesmo modelo.

O MD é dividido em três nós distintos, o primeiro está associado às entidades, o segundo às propriedades singulares, e o terceiro está associado às propriedades relativas. A fig. 46 representa o MD com origem na peça através da sua árvore descritiva (AD).

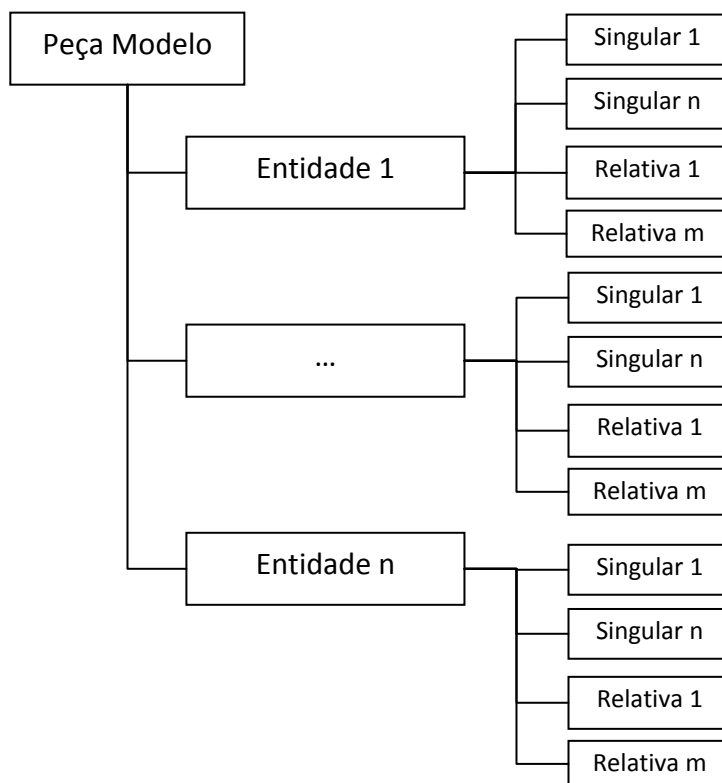


fig. 46 – Esquema do modelo descritivo de uma peça modelo (n e m podem ser diferentes para cada entidade)

Durante todo o processo de classificação apenas existe uma AD que no seu estado final é transformada no MD, nesta fase o modelo é memorizado e algumas das propriedades

singulares são retiradas. Estas propriedades são eliminadas por dependerem da imagem, ou seja, do referencial da imagem logo não sendo invariantes.

3.5.1 ENTIDADES E PROPRIEDADES SINGULARES

A AD permite facilmente verificar e analisar as interligações existentes na definição do modelo. De acordo com o capítulo 3.5 os primeiros passos a tomar para atingir a definição do MD passa por obter os nós com as entidades e as suas propriedades singulares as quais dependem directamente das entidades.

As entidades serão fornecidas pelos algoritmos de detecção aplicados (PPHT capítulo 3.3.2 e o CBF capítulo 3.3.3), através dos valores de saída de cada um destes ciclos é construído uma matriz de entidades.

Com base na peça da fig. 47 a AD obtida para a construção do MD é de acordo com a fig. 48.

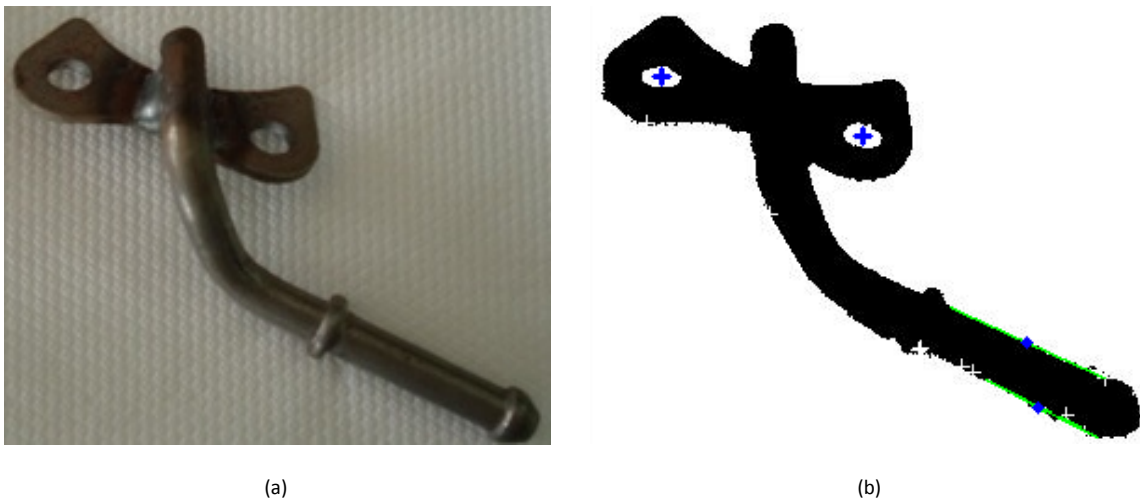


fig. 47 – (a) Imagem de um Gancho (b) indicação das entidades e dos seus pontos âncora

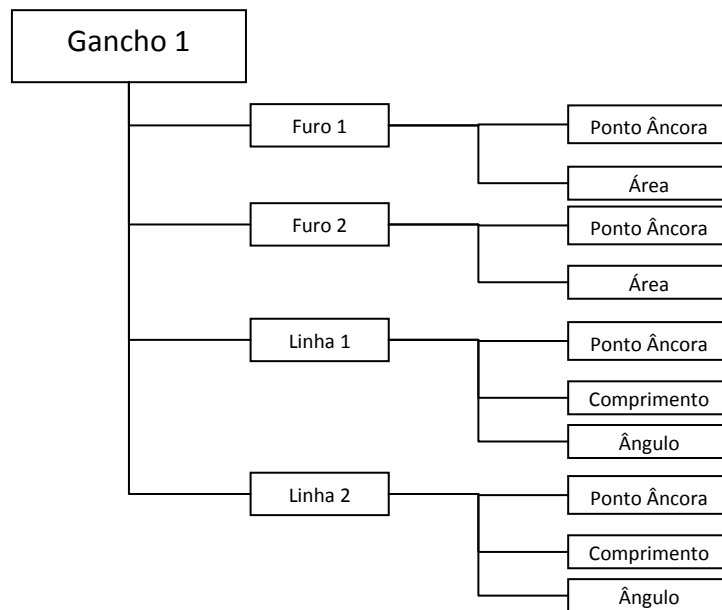


fig. 48 – Árvore descritiva da peça da fig. 47

3.5.2 CLASSIFICAÇÃO DAS RELAÇÕES (PROPRIEDADES RELATIVAS)

Conforme indicado no início deste capítulo, a última fase da construção do MD é a classificação das propriedades relativas entre entidades. Estas propriedades dependem das propriedades singulares das entidades, como a distância entre furos, calculada com as suas âncoras.

A selecção e classificação das propriedades relativas são realizadas após a conclusão das duas primeiras fases (preenchimento dos dois primeiros nós).

A classificação relativa é executada por selecção de pares de entidades e indicando o tipo de relação entre elas (de acordo com a Tabela 6). Esta classificação deve ser repetida até que o utilizador considere que o modelo está correctamente definido de forma a obter uma AD conforme a fig. 50.

Tabela 6 – Exemplo de propriedades relativas entre entidades

Entidade 1	Propriedade	Entidade 2
Furo	Distância	Furo Linha
Linha	Ângulo	Linha
Linha	Distância	Linha Furo

Na fig. 49 estão listadas todas as entidades existentes de acordo com a fig. 48, existem duas listas uma à direita que representa a entidade de referência e a da esquerda representa as entidades relativas.

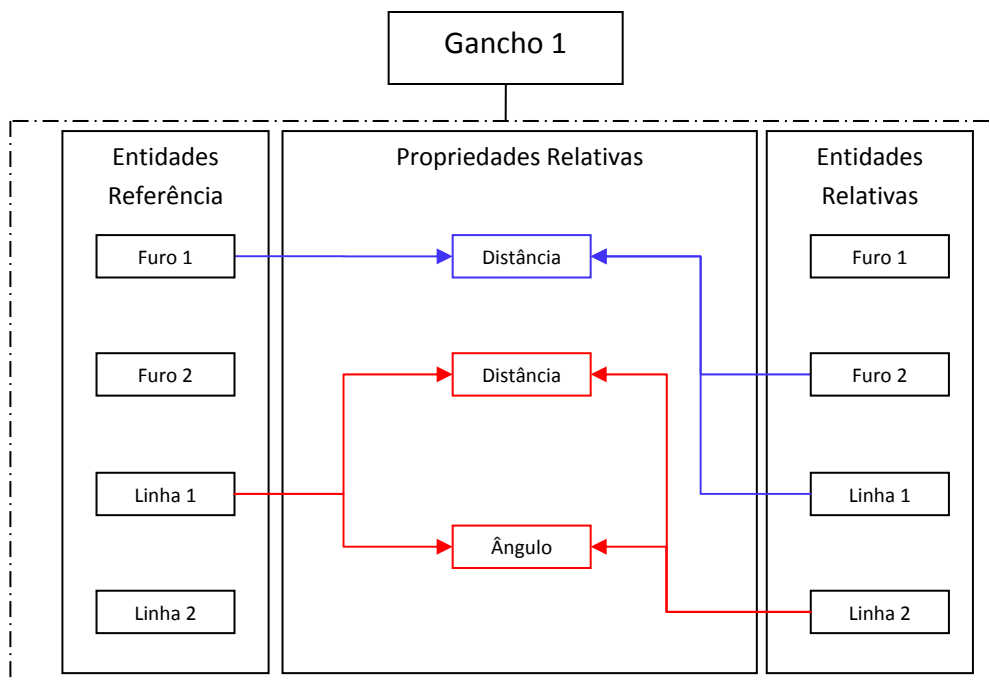


fig. 49 – Classificação das propriedades relativas entre entidades

A AD resultante desta fase é a árvore no seu estado mais completo, e encontra-se de acordo com a fig. 50.

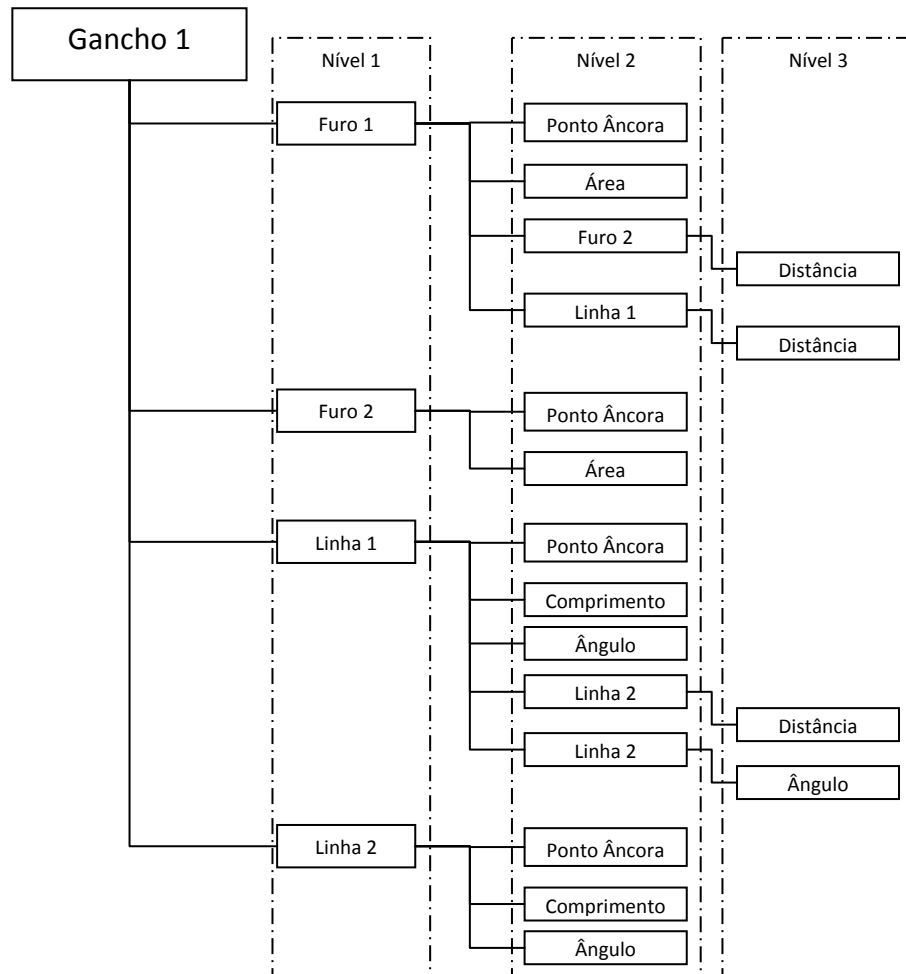


fig. 50 – AD completa com todas as entidades e as suas propriedades

3.5.3 DESCRIÇÃO FORMATADA

A árvore indicada na fig. 50 é a árvore mais completa da peça modelo com todas as suas propriedades. Após o utilizador concluir a classificação, a árvore deverá ser memorizada, ou seja, neste passo deverá ser registada a AD como o MD.

Neste passo é realizada a construção do MD, através de uma reanálise da AD e mantendo apenas as propriedades de relevância. Esta reanálise passa por todos os nós, mas tendo apenas impacto nos nós dos subníveis 2 e 3, este impacto advém do facto do subnível 3 ser eliminado e restando apenas o subnível dois com todas as propriedades.

O primeiro subnível contém a lista de entidades consideradas válidas, a qual é denominada pelo tipo de entidade e número sequencial da entidade, podendo ser do tipo Furo ou Linha.

O segundo contém a lista das propriedades singulares e propriedades relativas, as quais deveram indicar qual é o seu tipo (natureza).

A natureza das propriedades é realizada por intermédio de uma numeração de acordo com a Tabela 7. Dado que existe distâncias entre entidades de tipos diferentes será necessário distingui-las uma das outras daí haver dois tipos de distâncias. A Distancia1 representa a distância entre entidades do mesmo tipo, por sua vez, a Distancia2 representa a distância entre duas entidades de tipos diferentes.

Tabela 7 – numeração do tipo das propriedades

Propriedade	Tipo	Natureza
Área	1	Singular
Comprimento	2	Singular
Distancia1	3	Relativa
Distancia2	4	Relativa
Ângulo	5	Relativa

O exemplo da árvore do MD é a indicada na fig. 51 a qual apresenta apenas as propriedades relevantes (singulares e relativas) para a definição da peça modelo, com a ilustração das propriedades relativas entre entidades de tipos idênticos e diferentes.

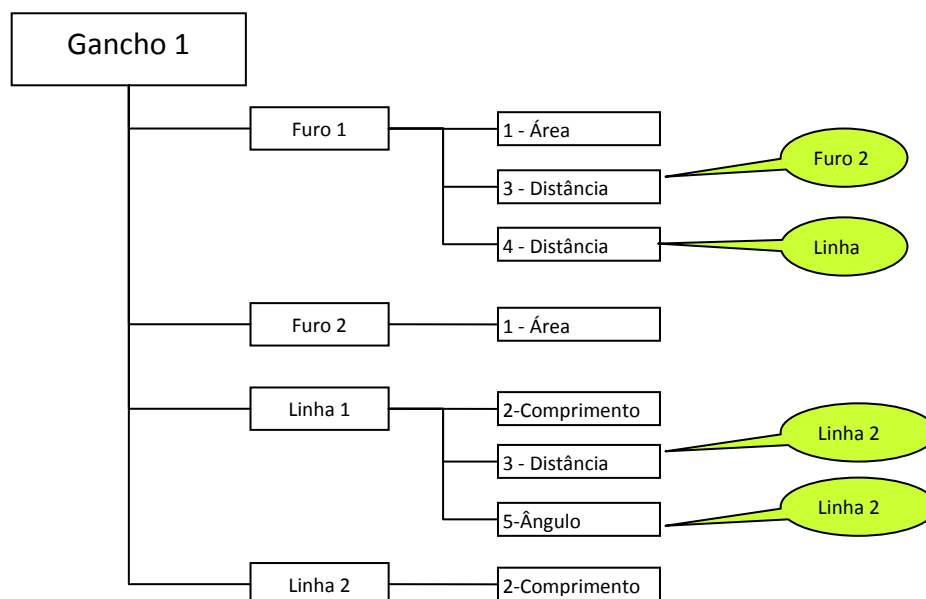


fig. 51 – Árvore do MD após memorização do modelo com as entidades relativas.

Os critérios que apoiam a selecção das propriedades relevantes foram definidos anteriormente nos capítulos 3.3 e 3.4 quando se referiu as características das propriedades singulares e relativas. As propriedades relativas têm de ser mantidas na construção do MD, dado que são invariantes. As propriedades singulares por sua vez podem ser eliminadas, caso seja uma propriedade que apresente variância (varia de imagem para imagem), como é o caso dos pontos âncora, ângulos das linhas.

4 DESENVOLVIMENTO DA APLICAÇÃO DE INTERFACE

Para comprovação do método exposto, é necessário verificar a sua eficácia, para isso recorreu-se às definições de parâmetros de ensaio bem como quais as ferramentas a usar.

Para a construção do modelo descritivo foi desenvolvida uma aplicação em C++. Nesta etapa foi necessário implementar todos os anteriores sub capítulos pertencentes ao capítulo 3.

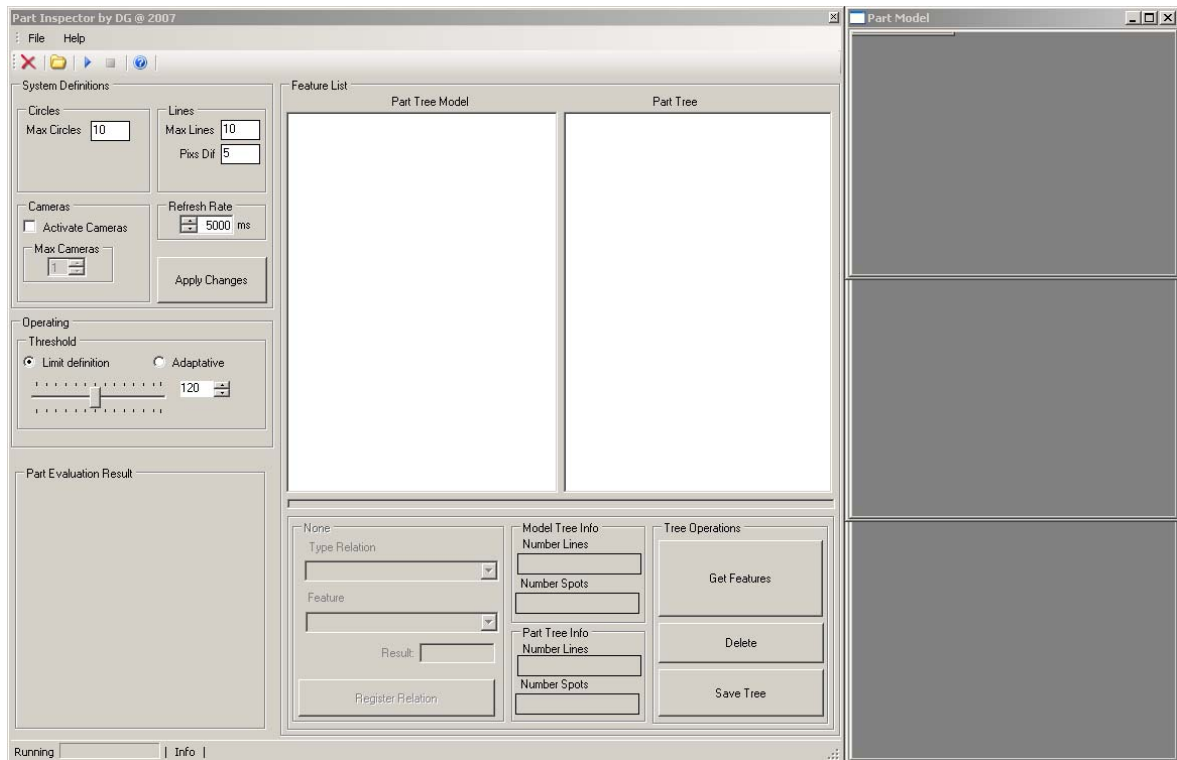


fig. 52 – Aspecto gráfico da aplicação desenvolvida em VCE

4.1 HARDWARE

Para a realização da análise experimental recorreu-se equipamentos de fácil acesso e a preços acessíveis.

Desta forma, utilizou-se um PC portátil Hp Pavilion dv5000 com um processador duo core centrino 1.83Ghz, 1GB de memória RAM, e uma placa gráfica nVidia GeForce Go 7400.

Como equipamento periférico (câmara digital), recorreu-se a uma câmara digital UniBrain Fire-i 2040.

Para realizar os ensaios, foi utilizada uma pequena banca de ensaio, que consiste num suporte para a câmara com uma lâmpada e uma base branca para colocar a peça.



(a)



(b)

fig. 53 – (a) Banca de ensaio utilizada, (b) *BirdView*

4.2 SOFTWARE

Para interagir com o utilizador e periféricos, foi necessário desenvolver uma aplicação com um ambiente gráfico simples que permitisse interagir com a câmara ou imagem a fim de testar todos os métodos indicados anteriormente.

A selecção da linguagem de desenvolvimento para a biblioteca de funções recai sobre o C++, principalmente por permitir o desenvolvimento orientado para o objecto (encapsulamento de dados, esconder informação e polimorfismo das funções). O encapsulamento e, a inacessibilidade à informação é realizada através de tipos de dados criados pelo utilizador: as Classes.

A escolha do sistema operativo recai sobre o MS Windows devido ao geral conhecimento deste sistema. Para desenvolvimento da interface recorre-se ao *Visual C++ Express* (VCE) da Microsoft, para permitir a integração da biblioteca desenvolvida e do OpenCV 1.0 para MS Windows.

4.3 PART INSPECTOR

O *Part Inspector* (PI) é uma aplicação desenvolvida recorrendo às facilidades do VCE e do OpenCV.

O recurso ao VCE aparece devido à necessidade de desenvolver uma *interface* gráfica intuitiva, e que facilitasse a sua utilização e evitasse a repetição de operações desnecessariamente. Apesar de muitas funcionalidades oferecidas pelo VCE, recorreu-se ao uso do WPF apenas para o desenvolvimento da API sendo todo o resto do código (classes) inerente aos métodos indicados desenvolvido usando Ansi C++, para que se torne possível a compilação destas classes noutros sistemas (desde que seja compatível com Ansi C++).

O OpenCV é uma livreria mundialmente difundida e com aplicação em inúmeras áreas com visão (investigação ou industrial). Esta livreria suporta um conjunto enorme de potencialidades tal como funções de operações matriciais, matemáticas e tratamento de imagem.

O PI foi desenvolvido em fases distintas como a aquisição de imagem, obtenção de dados, e exibição dessa informação.

4.3.1 CLASSES

Inicialmente foi necessário definir o tipo de dados, ou seja, o OpenCV disponibiliza vários tipos de dados como é o caso das **IplImage**, **CvPoint**, etc. Mas aproveitando as benesses do C++ optou-se pelo recurso à criação de uma classe que não só permitisse incluir a imagem como incluir todos os métodos e outros dados.

As Classes são compostas por membros privados e públicos podendo ser variáveis ou métodos. Os membros públicos encontram-se acessíveis exteriormente ao objecto enquanto os privados apenas são de acesso interno da classe. A aplicação desenvolvida é composta por diversas classes, podendo estar associadas umas às outras, ou serem paralelas. A **ClassImage** é composta pela **ClassSpot** e **ClassLine**, enquanto a **ClassFeature** é obtida pela **ClassImage** mas é composta pela **ClassRelation**.

ClassImage

Foi idealizada uma classe denominada por *ClassImage* (fig. 54) a qual incluiria toda a informação associada a essa imagem.

Os membros privados existentes estão relacionados com a imagem fonte (peça) denominada **Src**. A **Src** é obtida a após a leitura de um ficheiro de dados (disco) ou da câmara enquanto todas as outras **IplImage**'s são obtidas com a execução dos métodos públicos e apenas se encontram acessíveis para leitura. As variáveis públicas **NumImages**,

NumSpot e **NumLines** servem como contadores para que seja conhecido o número de objectos de cada tipo.

```

class ClassImage
{
public:
    /*procedures or functions related with Class Image that can also
    change Objects of type Line or Spot*/
    ClassImage(void);
    int LoadFromFile(char *FileName);//Acquire image from a diskfile
    int GetFromDevice(char node);//Acquire an image from hardware device, and grab it at src
    int DoThreshold(char Kind,int value);//decides what kind of Threshold shall be made
    int FillThreshold(bool refresh=FALSE);//do the fill on image
    int DoCanny(bool refresh=FALSE);//applies canny filter
    int GetImage(IplImage *Img,int Index=0);//functions to get out one of the images
    /*Functions or Procedures related to Spot Analysis*/
    int Segmentation(void);
    /*Functions or Procedures related to Line Object*/
    int GetSrcLines(void);
    //Variables and pointers declaration
    ClassSpot *(Spot[CL_IM_NUM_SPOTS]); //Classe de todas as manchas detectadas
    ClassLine *(Line[CL_IM_NUM_LINE])
    int NumSpots; //number of objects of type ClassSpot
    static int NumImages; //Number of objects of type ClassImage
    int NumLines; //Number of objects of type Classline
    CvSize Size; //Image Dimension, necessary to do several operations without need calculus
private:
    IplImage *Src; //contains the original image
    IplImage *Threshold; //contains the thresholded image
    IplImage *Fill; //contains the filled image
    IplImage *Canny; //contains the image with canny filter
    IplImage *TmpImg;//this IplImage just needs to be create one time
    //private functions
    int SetImage(IplImage **src1); //create at memory the requested image after Src exists
    int ReleaseImage(IplImage **src2);//remove from memory the requested image
};

```

fig. 54 – Classe ClassImage a qual é a base de toda a aplicação (excerto)

Falando sobre os métodos existentes na classe **ClassImage** verifica-se quase todos os métodos indicados no capítulo 3 encontram-se implementados nesta classe. Sendo de realçar o método *Segmentation* e *GetSrcLines* os quais implementam os métodos CBF e PPHT, estes métodos interagem com os vectores *Spot* e *Line*. Nesta aplicação o comprimento máximo dos vectores considerado foi 100, ou seja, apenas pode existir 100 furos ou 100 linhas.

ClassSpot e ClassLine

Estas duas classes aparecem devido à necessidade de guardar informação das entidades da imagem original. Implementam os dois métodos associados aos furos e às linhas da imagem (peça).

```
class ClassSpot
{
public:
    ClassSpot(void);
    int SetSpot(IpImage *Src); //set the image with all information of Spot
    struct Properties
    {
        CvPoint CenterPoint;
        int Area;
        int Perimeter;
    }Property;
    int GetAnchor(CvPoint* Anchor);
private:
    IpImage *Spot;
    IpImage *SpotCanny;
    IpImage *TmpImg;
};
```

fig. 55 – Classificação das variáveis e métodos da ClassSpot (excerto)

A classe *ClassSpot*(fig. 55) é composta por variáveis privadas e públicas, nesta classe as imagens são consideradas mais uma vez variáveis de leitura apenas acessíveis pelos métodos internos da classe. Todas as outras variáveis são calculadas quando é executado o método *SetSpot*. Este método não é o construtor da classe, mas funciona como tal, visto que a sua execução requisita todo o espaço que necessita ao sistema operativo para alocar toda a informação que contém.

```
class ClassLine
{
public:
    ClassLine(void);
    int SetLine(CvPoint * Point); //creates all information related with the line, based on the line points
    float LineLength(void){return Length;}; //returns the line length
    float LineAngle(void){return Angle;}; //returns the line angle with image axes
    int GetAnchor(CvPoint* Anchor); //returns the anchor point
private:
    CvPoint Ancora;
    float Length;
    float Angle;
};
```

fig. 56 - Classificação das propriedades e métodos da ClassLine (excerto)

A Classe *ClassLine* (fig. 56) apenas é formada por variáveis privadas e métodos públicos. As variáveis privadas são apenas todas as propriedades associadas a este tipo de entidade.

ClassFeature

Esta classe (fig. 57) aparece para realizar a definição do MD através da sua AD. A definição desta árvore apenas ocorre após a definição de todos os objectos anteriores (*ClassImage*). A distinção entre esta a *ClassFeature* e a *ClassImage* ocorre devido a serem paralelas, ou seja, após a execução a definição de todo o objecto do tipo *ClassImage*, essa informação é copiada para a *ClassFeature* de forma a existir um nível de abstracção necessário. Daí é possível iniciar a selecção das entidades a manter e a eliminar.

Um membro desta classe é a classe *ClassRelation*, a qual permite conter informação entre entidades.

ClassRelation

A *ClassRelation* (fig. 58) conforme o nome indica, apenas serve para conter o valor da relação entre duas entidades, ou seja, este objecto irá conter propriedades relativas.

```
class ClassFeature
{
public:
    ClassFeature(int Num);
    //specify variables here
    CvPoint Anchor;
    int Type; // 0 - Not Classified, 1 - Circle, 2 - Linha
    double Dimension1,Dimension2; //values depends of Type
    int NumRelations; //specifies number of relations that the feature have
    ClassRelation *Relation[CL_F_NUM_RELATIONS]; //relation information
    //configure the feature with type and properties
    int SetFeature(int type, double value1, double value2=0);
    //configure the relation between features
    int SetRelation(int Type, int Feature, double Val);
    ...
};
```

fig. 57 - Classificação da *ClassFeature* (excerto)

```

class ClassRelation
{
public://specifies variables
    int Type; //0- Distance, 1 - Angle
    double Dimension; //value of relation
    int Feature; //number of related feature
    ...
};

```

fig. 58 - Classificação da ClassRelation (excerto)

4.3.2 CONSTRUÇÃO DO MD

Esta construção é dividida em três fases distintas. A primeira é a procura das entidades existentes na peça, a segunda classifica as propriedades singulares de cada entidade e a terceira permite ao utilizador a classificar e salvar as propriedades relativas.

A primeira e segunda fase são fases totalmente automáticas em que a aplicação não necessita da interação do utilizador. Enquanto a terceira devido à existência de diversas entidades é realizada por intermédio do utilizador, para que apenas sejam definidas as relações necessárias para caracterizar o modelo.

Classificação das Entidades

A classificação das entidades é a execução das duas primeiras fases, no entanto a eficiência da classificação depende da selecção do limite de binarização.

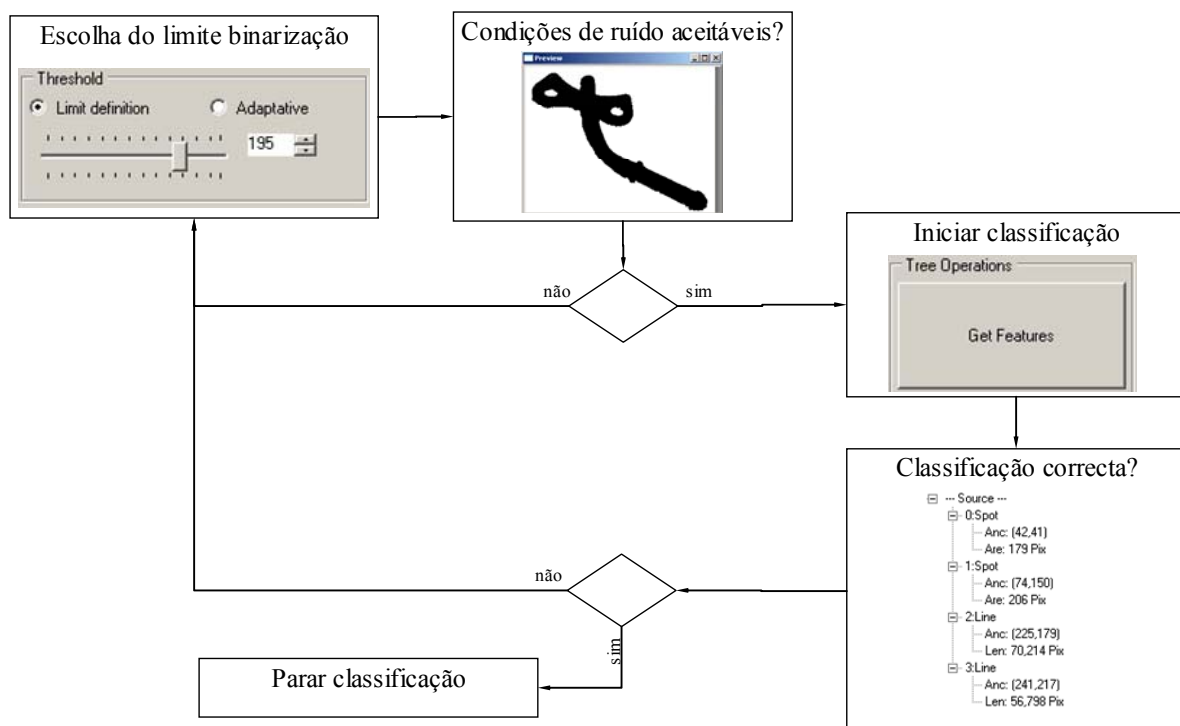


fig. 59 – Algoritmo das primeiras duas fases do módulo MD

A lista de entidades é tanto menor e com menor necessidade de tratamento quanto menor for a percentagem de ruído existente na imagem após a binarização. Quanto mais adequado for o limite de binarização escolhido melhor será o desempenho do método. A fig. 60 representa a lista de entidades obtida para um limite de binarização de 120. Apesar de estar correcta, ou seja, o número de furos existe tal como de linhas, aparecem devido a ruído deixado pela binarização (grupo de pixels soltos).

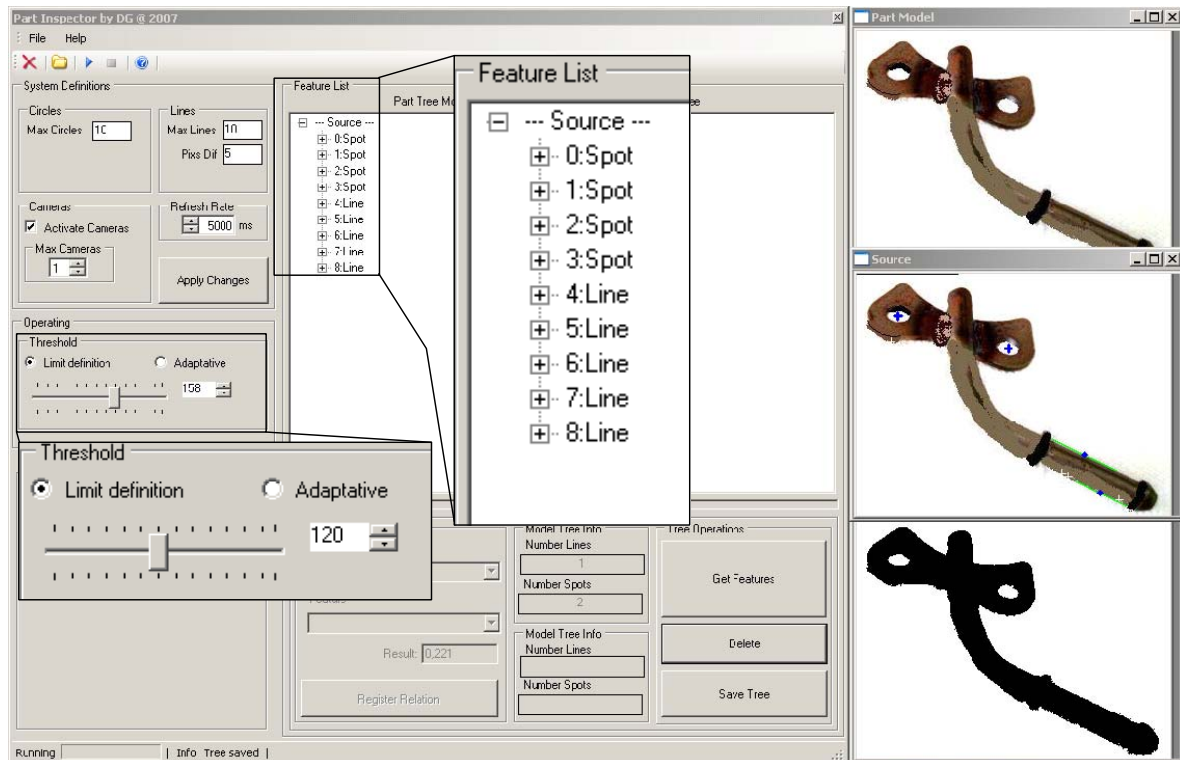


fig. 60 – Árvore para um limite de binarização de 120

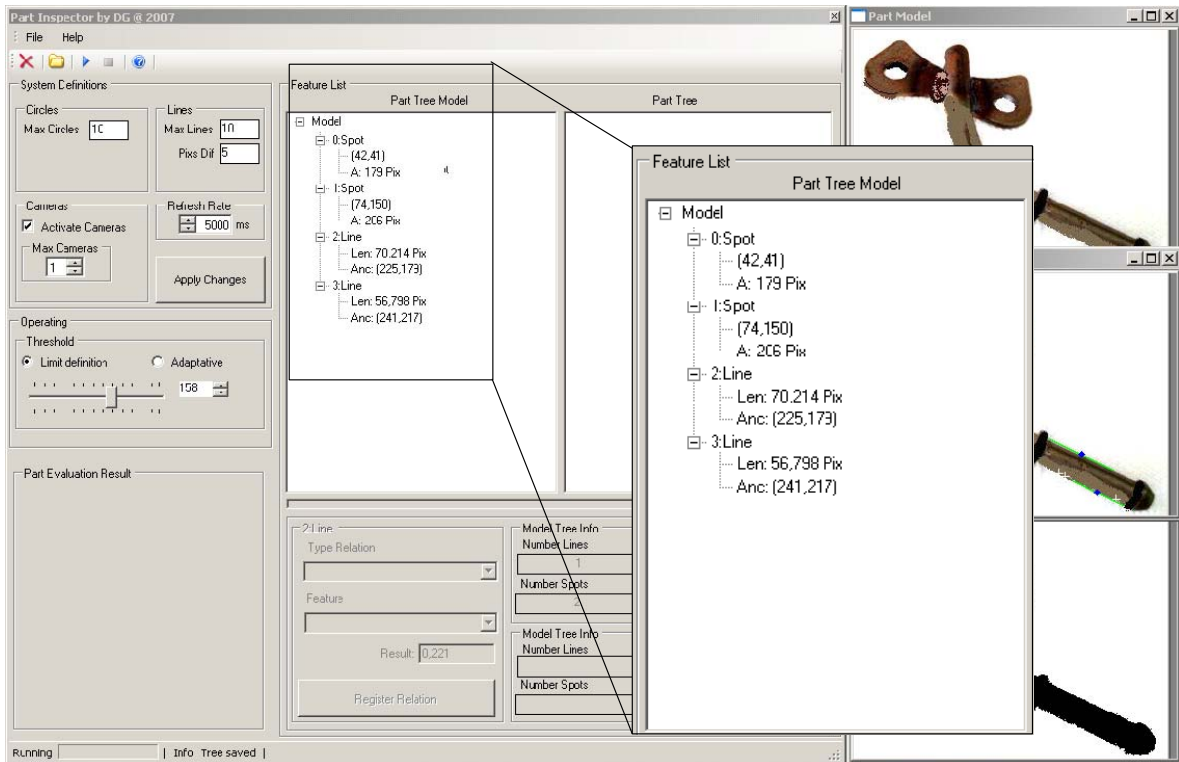


fig. 61 – Árvore com as entidades listadas e as suas propriedades

A fig. 61 representa a árvore ou listagem das entidades pretendida, duas linhas e dois furos. A árvore tem as propriedades singulares para futura análise, classificação das propriedades relativas.

Classificação das Relações

Conforme indicado no início deste subcapítulo, a última fase do algoritmo do MD é a classificação das propriedades relativas entre entidades.

As propriedades relativas são indicadas pelo utilizador. A selecção e classificação das propriedades é realizada após a conclusão das duas primeiras fases, a matriz de entidades e propriedades singulares é exibida (fig. 61) através da qual se pode iniciar a classificação.

A classificação relativa é executada seleccionando pares de entidades, ou seja, o utilizador escolhe uma entidade da lista, seguidamente deverá escolher o tipo de relação entre entidades e por fim com qual entidade irá existir essa relação. O ciclo indicado na fig. 62 deverá ser repetido até que o utilizador considere que o modelo está correctamente definido.

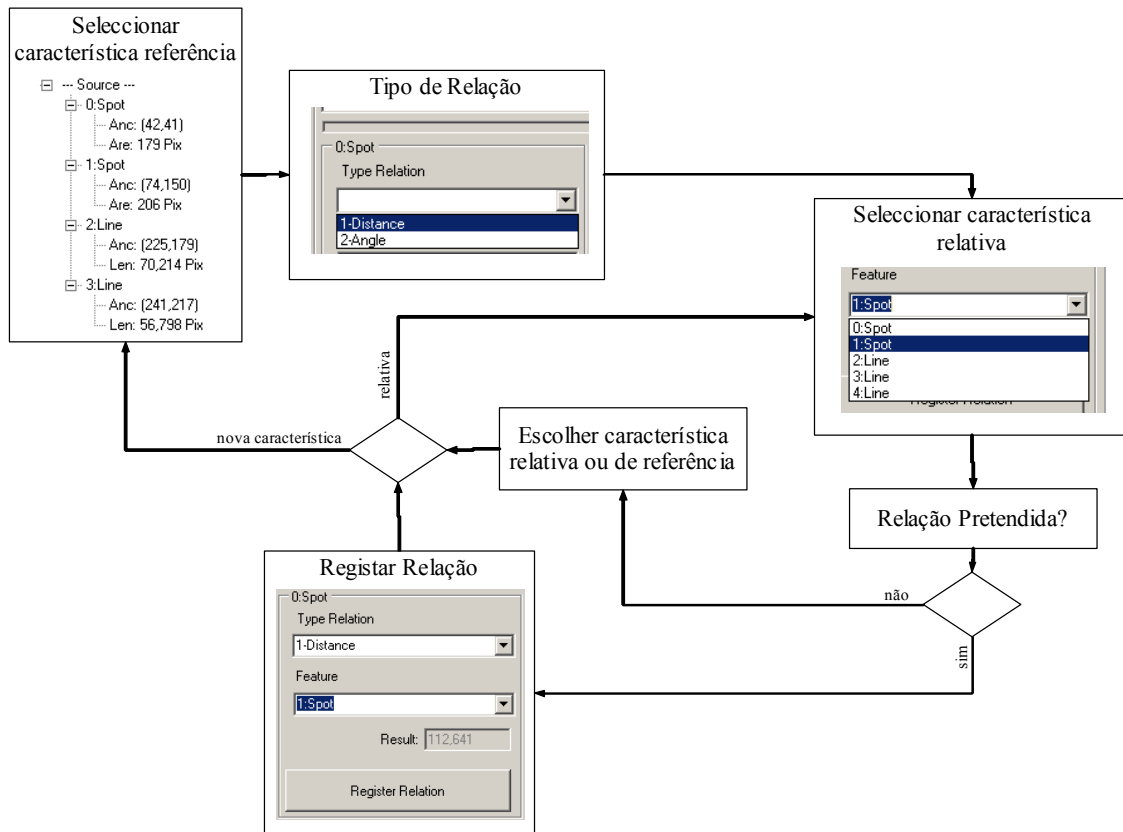


fig. 62 – Ciclo de classificação das propriedades relativas

Após o utilizador considerar que a AD está completa deverá guardar-se a árvore, passando a ser o MD da peça. A árvore indicada na fig. 63 é AD para a peça da fig. 22 a qual contém todas as propriedades singulares e relativas para a definição da peça.

A fig. 64 é obtida pela transformação da AD da fig. 63 de acordo com os passos indicados no capítulo 3.5. A identificação das propriedades é realizada sempre por intermédio de três caracteres alfanuméricos, e da Tabela 7 que indica os tipos de propriedades relativas entre das diversas entidades. Estes caracteres facilitam a percepção e interação entre o utilizador e a aplicação.

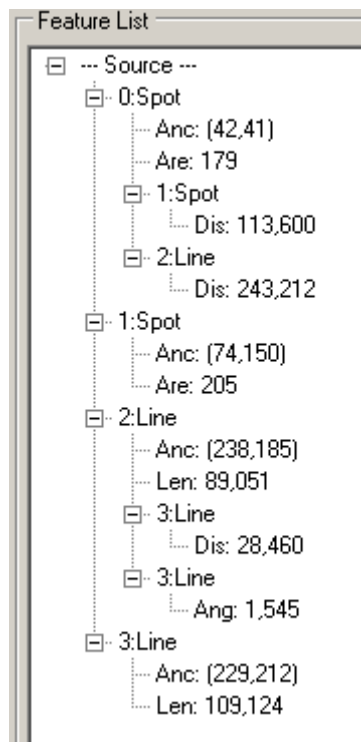


fig. 63 – Matriz com as propriedades relativas entre diversas entidades

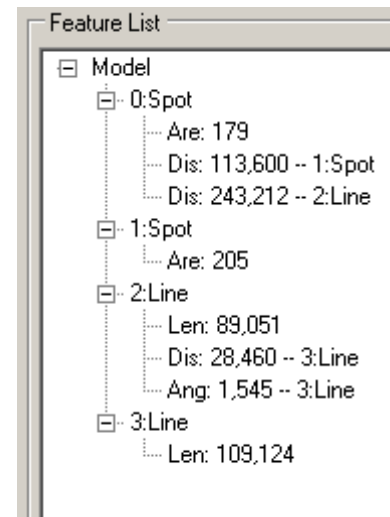


fig. 64 – Árvore após passagem a MD

4.4 RECONHECIMENTO PARAMÉTRICO

Reconhecer uma peça é algo que o ser humano faz de forma intuitiva, ou seja, após visualizar uma peça pela primeira vez o cérebro humano é capaz de a identificar posteriormente entre muitas outras. Este reconhecimento é realizado através de entidades armazenadas na memória humana (memória descritiva) as quais foram consideradas as mais importantes aquando a primeira visualização.

Por analogia com o sistema anterior o reconhecimento paramétrico (RP) baseia-se na busca de entidades registadas no MD, em que o MD é a memória descritiva de uma determinada peça modelo conforme indicado no capítulo 4.3. Com base nestas entidades e propriedades torna-se possível indicar se a peça a ser visualizada é conhecida ou não.

O RP é realizado em duas etapas; a primeira etapa consiste na definição da AD da peça através da detecção das entidades existentes na imagem, as entidades são detectadas através dos mesmos algoritmos utilizados na definição da AD do capítulo 4.3.

A segunda etapa passa pela comparação da AD da imagem da peça a reconhecer com a peça modelo, através da verificação da existência das entidades e suas relações. Esta fase é executada em cascata (fig. 65) e em cada nível da cascata o resultado obtido indica se a imagem a testar contém a MD a procurar ou não. A passagem de níveis da cascata ocorre quando é validado o nível anterior. Diferentes condições e propriedades das entidades são

validadas em cada nível, estando previsto que a propriedade a verificar não seja exactamente idêntica à do MD, assim, é necessário atribuir tolerâncias em cada nível. A tolerância é calculada de acordo com a Eq. 15. A Eq. 15 é obtida pela diferença entre a propriedade da AD com a do MD em relação a do MD, sendo assim uma tolerância percentual.

$$Tol = \left| \frac{Prop da AD - Prop do MD}{Prop do MD} \right| \quad \text{Eq. 15}$$

O nível A aparece para verificar se o número de entidades descritas no MD se verifica; o número de furos e de linhas. Neste nível, as condições que a AD deve verificar são número de manchas existentes de ser igual ao número de manchas do MD, e o número de mínimo de linhas existentes deve ser igual ao número de linhas usadas na construção do MD. O nível B verifica as propriedades singulares das entidades do tipo Spot. Apenas são verificadas as entidades do tipo Spot em relação à sua propriedade singular (área) porque esta é invariante à rotação. No nível C são verificadas as propriedades relativas entre entidades do tipo linha, ou seja, o ângulo entre linhas. No nível D verifica-se a distância entre entidades do tipo Spot, distância entre furos.

O comprimento das linhas apesar de ter sido considerado na caracterização do MD, não foi incluído na cascata de decisão, porque se verificou pouco eficaz e com alguma variabilidade.

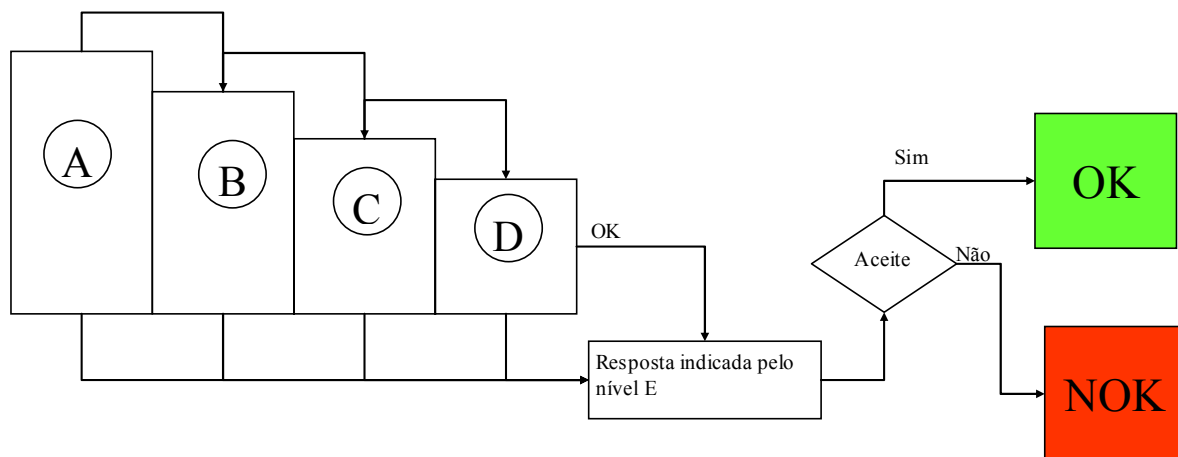
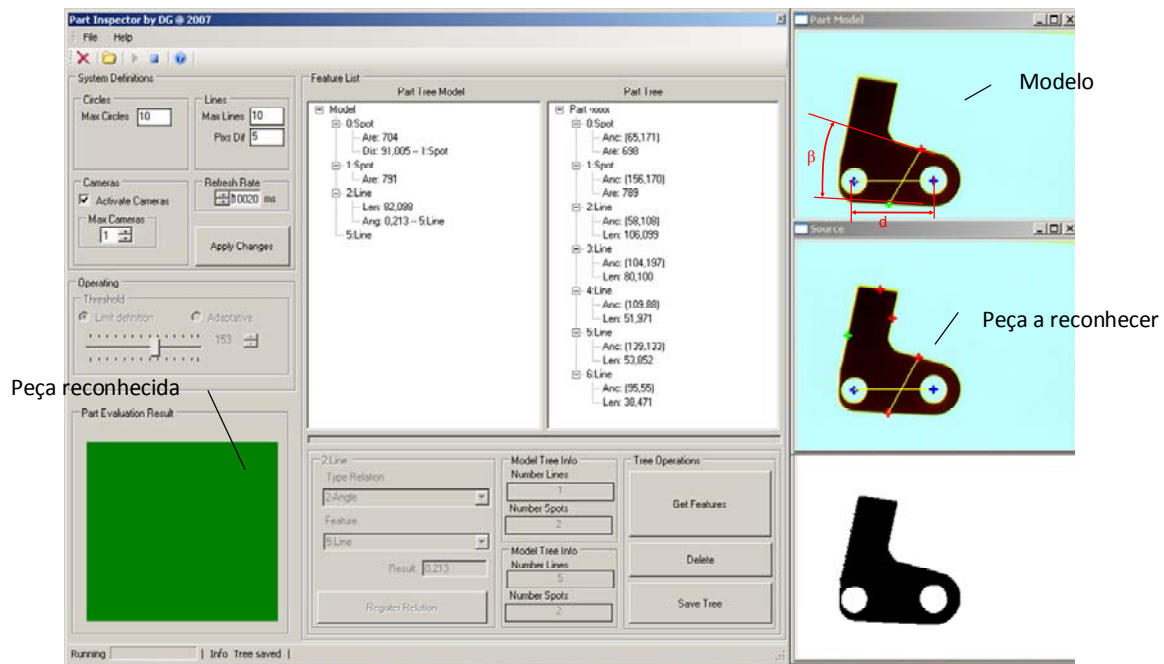


fig. 65 – Cascata de decisão implementada no RP

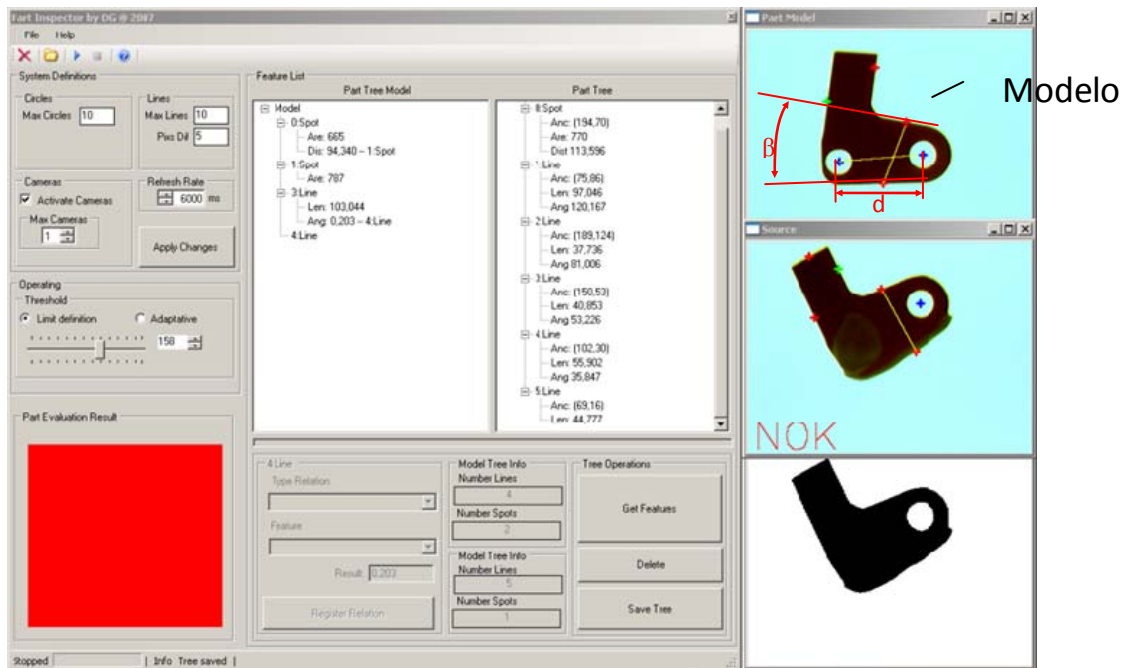
A escolha de um sistema de decisão em cascata (fig. 65) aparece devido à necessidade de todos os critérios serem comprovados para que a validação de reconhecimento de uma peça ocorra, bastando que um critério falhe para a rejeição no reconhecimento. Este sistema de decisão em cascata permite facilmente a introdução de novos critérios de aceitação, estes critérios que podem estar associados a entidades existentes ou novas.

A imagem da fig. 66 mostra uma imagem modelo com quatro entidades e suas relações, este modelo é composto por dois furos e duas linhas (4 entidades), estando definida uma

distância entre os furos e o ângulo entre as linhas. Esta imagem indica que as entidades e suas propriedades foram identificadas com sucesso para outra peça numa posição diferente à do modelo.



(a)



(b)

fig. 66 – MD e identificação das entidades e suas relações

A fig. 66 (a) representa um caso em que a peça é considerada correcta, enquanto a imagem (b) representa uma peça incorrecta.

4.5 RESULTADOS

Todos os ensaios foram realizados em situações específicas, ou seja, com retro iluminação sobre a peça numa base branca para aumentar o nível de contraste. Esta técnica simplifica o problema na sua vertente de pré-processamento (filtrar ruído, normalizar brilhos, encontrar limite binarização) que não é a vertente que este trabalho pretendia aprofundar.

A câmara (CCD) está colocada de forma paralela com a banca, de forma a obter uma vista com a planta da peça (2D), conforme ilustrado na fig. 53.

Foram realizados ensaios com peças distintas de acordo com a fig. 67, as peças tinham uma MD semelhante, ou seja, furos e a distância entre eles, arestas e ângulos entre elas.



fig. 67 – Peças usadas nos ensaios

Os resultados obtidos encontram-se ilustrados nas fig. 70, e fig. 73 demonstram a decisão da peça em cada leitura. As linhas amarelas que ligam as diversas entidades aparecem apenas para indicar quais as propriedades definidas ou detectadas. Neste caso, e para facilitar a discussão de resultados as imagens tem todas as propriedades relativas identificadas através de uma linha amarela entre os pontos âncoras das entidades.

A fig. fig. 68 (a) ilustra a peça 1 sem defeitos e usada para classificar o MD, enquanto (b e c) ilustram a peça 1 com defeitos, como é o caso da falta de furos, ou aresta incorrecta. A fig. 70 representa o teste realizado na peça 1 em diversas posições e orientações.

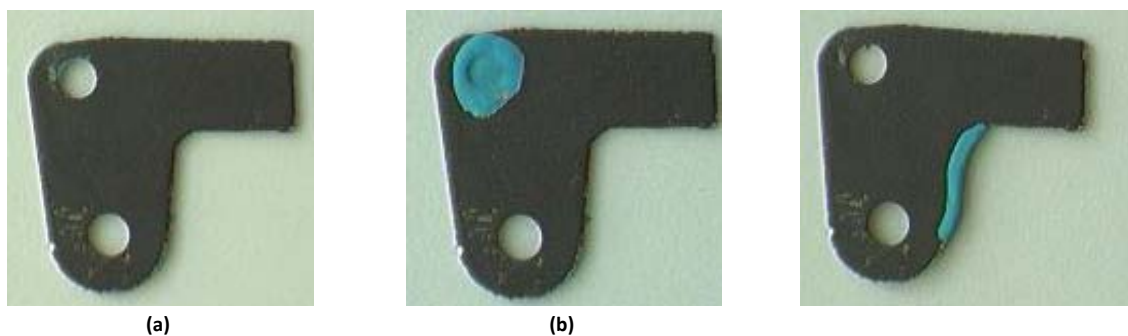


fig. 68 – Peça 1 (a) sem defeitos (b) falta de um furo (c) aresta deformada

O MD depende do utilizador e dos defeitos detectados, no caso da peça 1 (fig. 68 - a), o utilizador sabe que a peça possui defeitos num dos furos (fig. 68 - b) e numa das suas arestas (fig. 68 - c). Com base na topologia da peça e dos seus defeitos o utilizador entende ser suficiente a árvore da fig. 69 (b) para caracterizar a peça 1, definindo assim o MD desta peça, no qual existe furos e a distância entre eles, e duas linhas (arestas) com um ângulo entre elas, a fig. 69 (c) representa o MD registado pelo utilizador. Após a definição do MD pode ser iniciado o processo de reconhecimento das peças nas imagens.

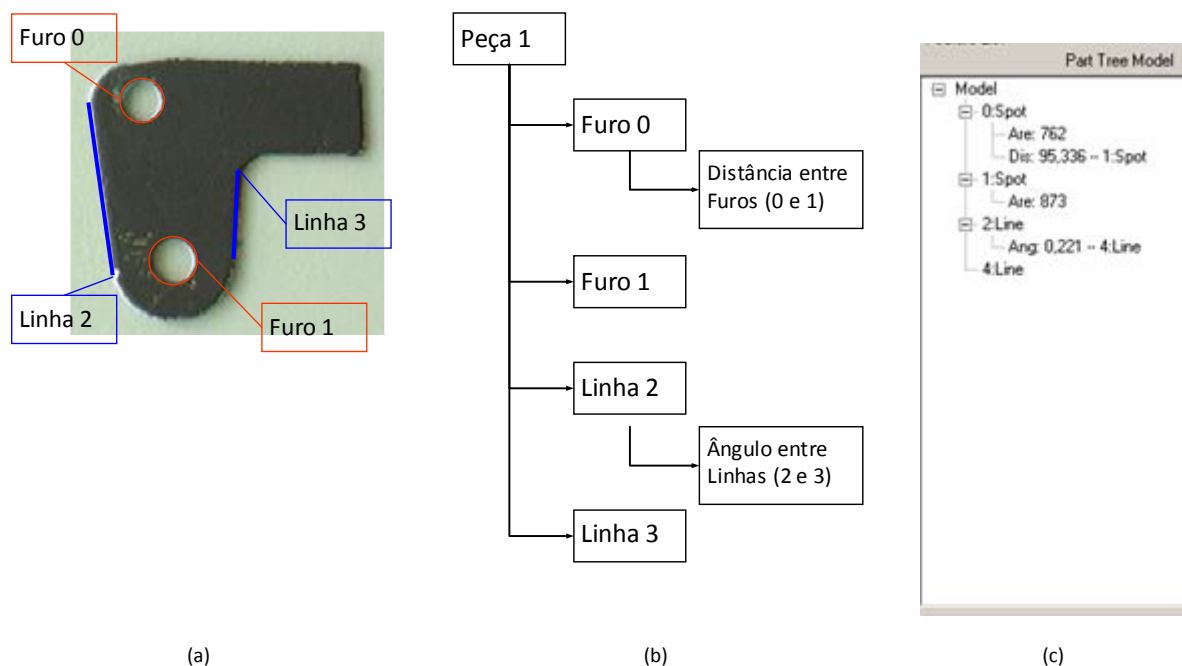


fig. 69 – Peça 1 (a) entidades escolhidas (b) MD teórico (c) MD definido no PI

A fig. 70 representa um conjunto de casos testados pelo PI para o exemplo da fig. 68, conjunto composto por peças boas e defeituosas, as peças foram colocadas de forma aleatória e com qualquer orientação e posição.

Os resultados para a peça 1 indicam que para o caso **a, b, c, d, e, j** (fig. 70) as peças estão correctas, ou seja, em todas existe dois furos, e as suas áreas não variam mais de 10% com o MD. A distância entre os furos tem uma variação inferior a 10% do MD. Existe também duas linhas que fazem entre si um ângulo que não difere mais de 10% do MD.

As imagens **f** e **i** da fig. 70 representam uma peça defeituosa, em que uma das arestas (linhas) está com uma inclinação diferente comparativamente à imagem modelo, não é possível encontrar a relação angular entre duas arestas, no entanto, conforme a figura são encontrados os furos e a distância entre eles de acordo com o MD.

A imagem **g** representa uma situação em que a peça não tem um dos furos que a identifica, mas detecta duas arestas que fazem entre si o ângulo conforme o MD. Enquanto o caso **h**

apresenta uma situação com os três defeitos, ou seja, falta um furo e uma das arestas está incorrecta, não sendo detectada nenhuma propriedade relativa.

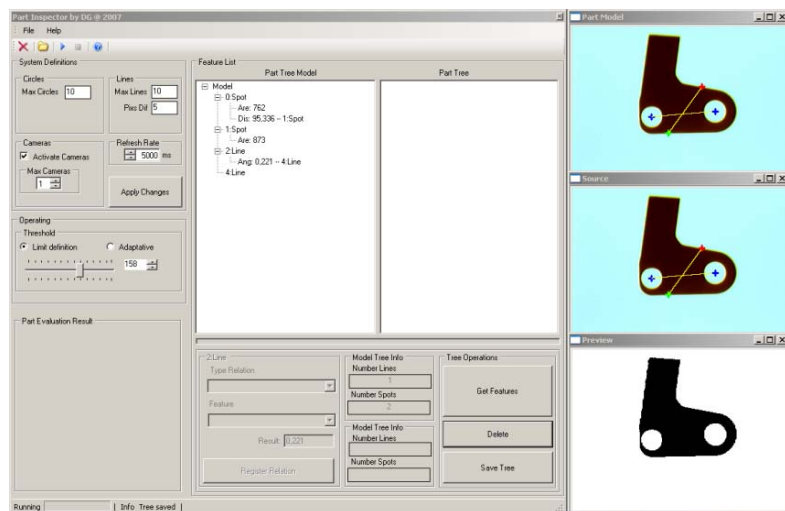


fig. 70 – Resultados da peça 1

Para o caso da peça 2 (fig. 71), as entidades consideradas mais relevantes pelo utilizador para identificar a peça são os furos, arestas, e as relações entre elas (distância e ângulo). O MD teórico previsto pelo utilizador encontra-se definido na fig. 72 (b) enquanto a fig. 72 (c) mostra o MD obtido pelo utilizador no PI.

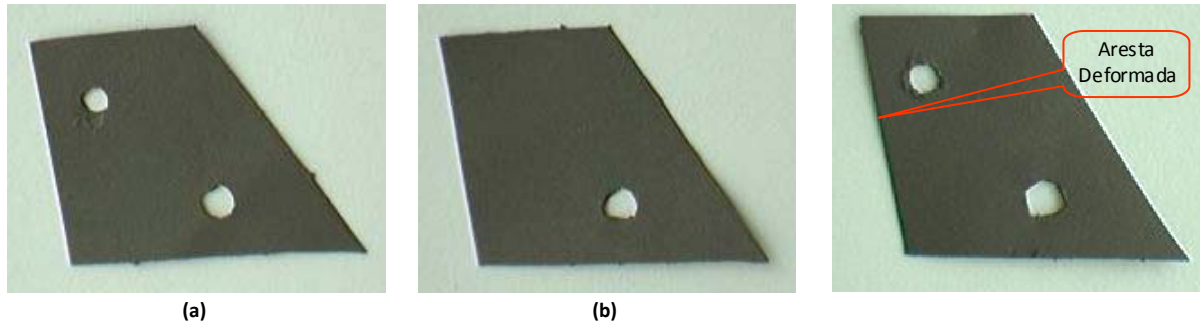


fig. 71 – Peça 2 (a) sem defeitos (b) falta de um furo (c) aresta deformada

A fig. 73 representa todos os casos testados pelo PI com as peças boas e defeituosas da peça 2, as peças foram colocadas de forma aleatória e em qualquer posição e orientação.

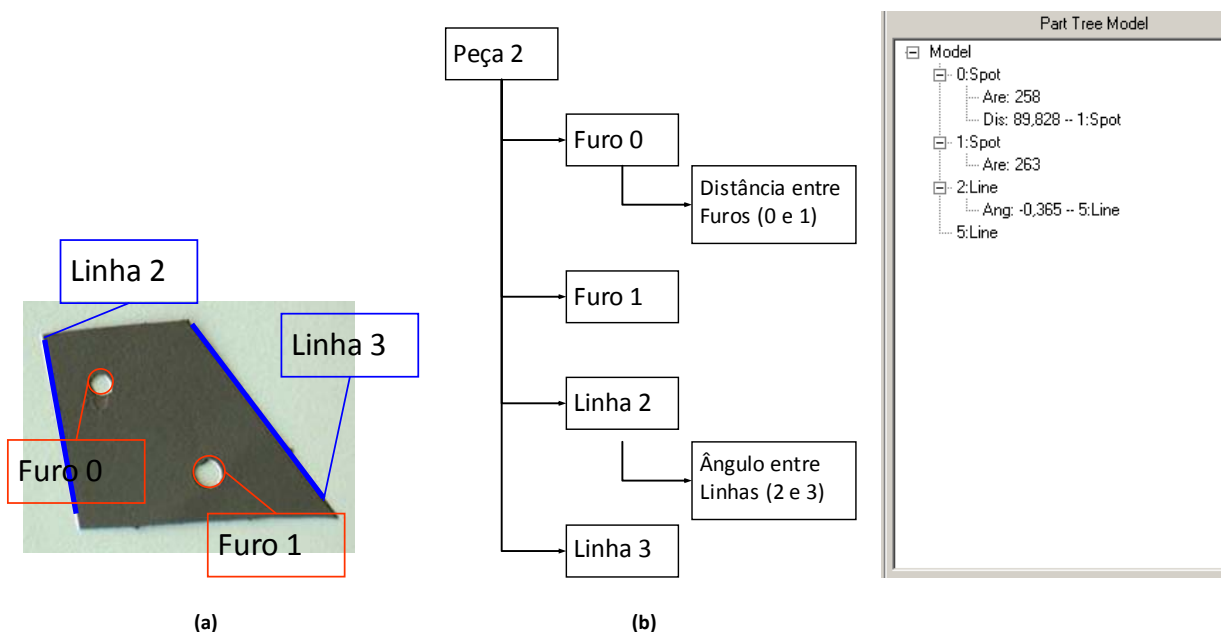


fig. 72 – Peça 2 (a) entidades escolhidas (b) MD teórico (c) MD real definido no PI

Os resultados para a peça 2 indicam que para o caso **a, b, c, d, e, j** (fig. 73) as peças estão correctas, ou seja, em todas existe dois furos, e as suas áreas não variam mais de 10% do que o MD. A distância entre os furos tem uma variação inferior a 10% do MD. Existe também duas linhas que fazem entre si um ângulo que não difere mais de 10% do MD.

As imagens **g** e **i** da fig. 73 representam uma peça defeituosa, em que uma das arestas (linhas) está com uma inclinação diferente comparativamente à imagem modelo, não é possível encontrar a relação angular entre duas arestas, no entanto, conforme evidencia a figura são encontrados os furos e a distância entre eles de acordo com o MD.

As imagens **e**, **f** e **h**, representam uma situação em que a peça não tem um dos furos que a identifica, mas detecta duas arestas que fazem entre si o ângulo conforme o MD.

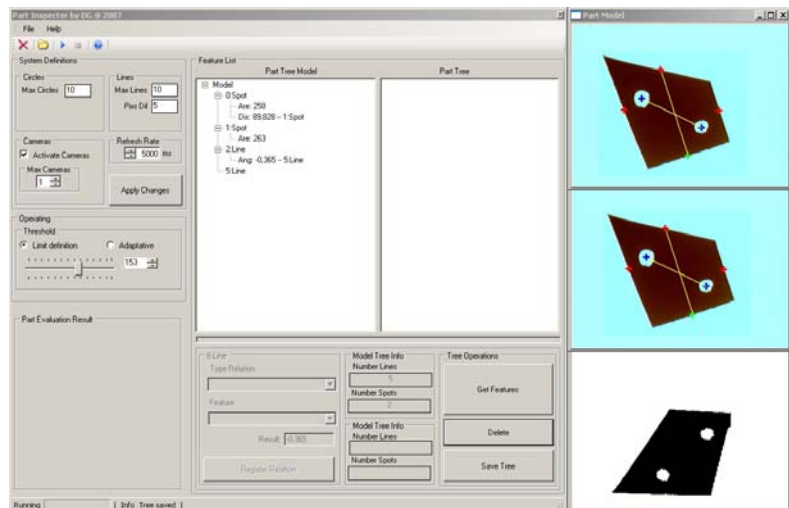
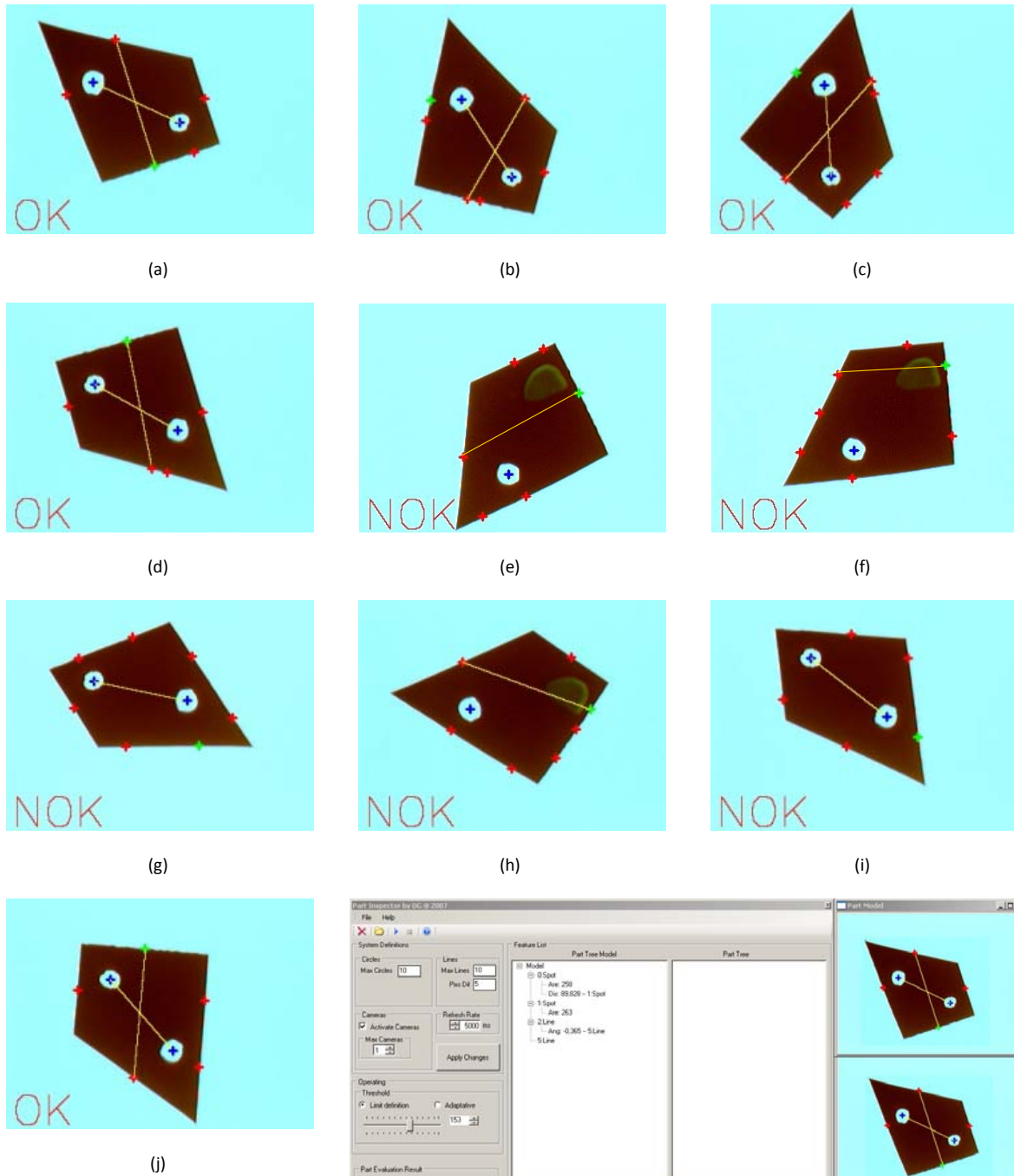


fig. 73 – Resultados da peça 2

5 CONCLUSÕES E PERSPECTIVAS FUTURAS

Nos capítulos anteriores foram detalhados todos os passos executados para levar a cabo a realização deste trabalho. Neste capítulo, são indicadas as conclusões principais tal como alguns aspectos que poderão contribuir para o aumento da eficiência desta técnica.

5.1 CONCLUSÕES

Neste trabalho foram apresentados os fundamentos teóricos e algoritmos para o desenvolvimento do Reconhecimento Paramétrico. Foram também apresentados os resultados obtidos relativamente à identificação das entidades e das suas propriedades e da comparação entre a árvore modelo e a imagem (peça) real, esta solução apresenta para as peças testadas robustez e fiabilidade. A duração do ciclo de Reconhecimento Paramétrico depende do número de entidades detectadas e da complexidade das relações, mas para as peças testadas obteve-se uma duração de 100ms por ciclo (busca de 2 furos e 5 arestas). Este tempo de ciclo permite inspeccionar 20000 peças em apenas 30 minutos, esta situação permitiria controlar o stock de um ano num dia. Mas tendo em consideração a parte física (maquinaria) de todo o sistema de inspeção, pode-se indicar que o sistema poderia tomar uma decisão a cada meio segundo e assim inspeccionar o stock de uma semana apenas em 3 horas.

As entidades escolhidas apresentaram bons resultados, dado que possuem propriedades singulares e relativas que são invariantes à rotação e posição, no entanto, as entidades do tipo Furos são mais simples de identificar enquanto as Linhas (arestas) são mais complexas para análise, daí apenas ter sido utilizado o critério dos ângulos que fazem entre si. Contudo, verifica-se que para imagens (peças) mais complexas e para facilitar a identificação das arestas, será necessário adicionar novos tipos de entidades como é o caso dos cantos (fig. 74) os quais são invariáveis à rotação e posição e até à escala, nomeadamente as relações possíveis de registar.

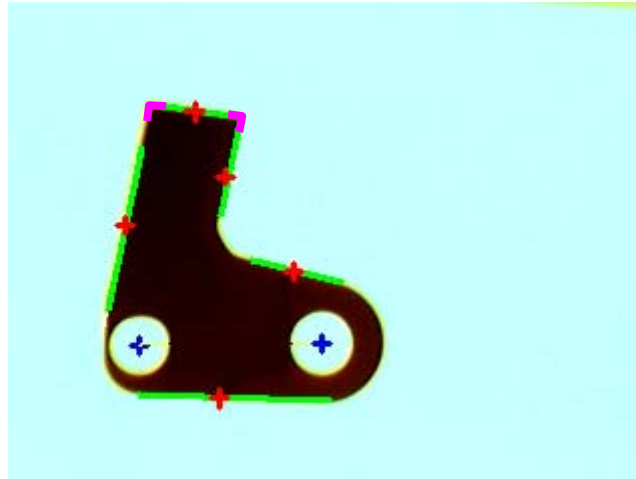


fig. 74 – Cantos nas imagens

Este tipo de entidades, conforme indicado na fig. 74, permitem auxiliar a identificação das arestas reconhecidas e a localização dos furos relativamente às arestas.

Apesar de ter sido desenvolvida uma aplicação que permitiu validar o Reconhecimento Paramétrico, este método poderia ser facilmente implementado nas soluções industriais apresentadas no capítulo 2.2.2, das quais, o Insignth parece oferecer melhores possibilidades de interação, permitindo a construção de funções e bibliotecas.

Este trabalho assentou no desenvolvimento de uma técnica não intrusiva baseada em visão que permite realizar o reconhecimento de peças para controlo e inspeção de qualidade. Esta técnica difere das habituais abordagens industriais definindo propriedades invariantes à posição e rotação, criando com base nestas o Modelo Descritivo da peça que será testado posteriormente para verificação da conformidade da mesma.

5.2 PERSPECTIVAS FUTURAS

Para uma implementação mais ampla desta técnica numa aplicação industrial deverá haver algumas revisões bem como o desenvolvimento de algoritmos adicionais. No decorrer deste trabalho foram identificados aspectos que trariam melhorias para a eficiência desta técnica, como novas entidades, mesmo do tipo virtual, e algoritmos de reconhecimento paramétrico que possam lidar com situações de redundâncias.

As entidades virtuais são criadas a partir de outras entidades já detectadas, ou seja, as entidades virtuais poderão ser deduzidas do conjunto de entidades reais detectadas. Consultando a fig. 75 verifica-se que é possível construir um canto virtual (3) com base nas arestas (C, E). A aresta A da fig. 75 não é detectada em todas as situações, no entanto os seus cantos (1,2) podem ser detectados em todas as situações sendo por isso possível criar uma linha virtual que represente esta aresta.

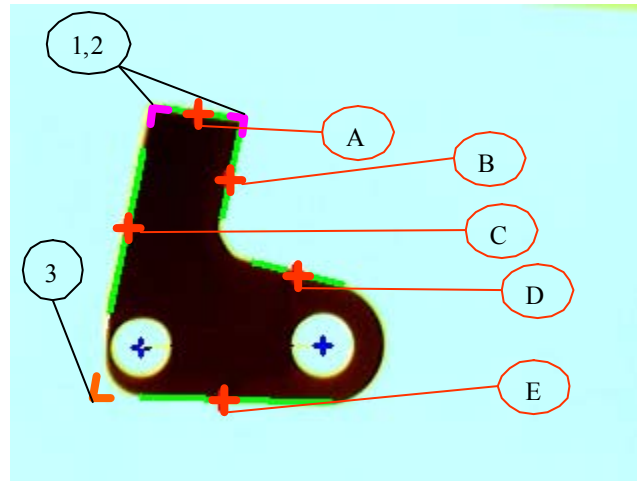


fig. 75 – Entidades Virtuais

A redundância aparece quando existem múltiplas soluções para verificar um Modelo Descritivo definido com base em algumas relações entre entidades, por exemplo, entre arestas. As arestas C e E fazem entre si o ângulo α mas as arestas B e E fazem entre si o ângulo β . Apesar de estes ângulos serem diferentes, estão contidos no intervalo de tolerância originando uma incorrecta decisão.

A adição de algumas melhorias no Reconhecimento Paramétrico, como as indicadas anteriormente, apenas possibilitará a aplicação deste método a tipologia de peças como as indicadas na fig. 6, no entanto, os princípios básicos do Reconhecimento Paramétrico são válidos e podem ser implementados e afinados para um processo produtivo.

6 LISTA DE REFERÊNCIAS

- Braggins, Don. 1996.** *Industrial Vision Applications Sectors*. Royston Herts SG8 6LB, s.l. : 1996.
- Canny, J. 1986.** *A Computational Approach to Edge Detection*. 1986.
- Cognex. *Insigth Tutorials and Samples*.**
- Davies, E. R. 1997.** *Machine Vision - Theory and Algorithms Practicalities V2*. London : Academic Press, 1997.
- Epedal SA. 2006.** Indicadores de Cadencias Mensais. Sangalhos : s.n., 2006. Cadencias mensais reais versus teóricos.
- Fisher, Bob, Perkins, Simon e Walker, Ashley. 1994.** Hiper Media Imagem Processing Reference. *University of Edinburgh*. [Online] Department of Artificial Intelligence, 1994. [Citação: 19 de 06 de 2007.] http://www.cee.hw.ac.uk/hipr/html/hipr_top.html.
- Galambos, C., Matas, J. e Kittler, J. 1999.** Progressive Probabilistic Hough Transform for line detection. *IEEE*. 1999, p. 554 a 560.
- Green, William E. 2002.** *Canny Edge Tutorial*. 2002.
- Horst Haußecker, Peter Geißler. 1999.** *Handbook of Computer Vision and Applications Volume 3*. s.l. : Academic Press, 1999.
- Mamede, Carlos e Pinheiro, Aureliano. 1999/2000.** LUMINOTECNIA. *laseeb.isr.ist.utl.pt*. [Online] 1999/2000. <http://laseeb.isr.ist.utl.pt/publications/tfc/luminotecnia/luminotecnia.htm>.
- NeuroCheck. *Application Notes*.** www.neurocheck.com : s.n.
- Sherlock. *Sherlock Specifications*.** www.sherlock.com : s.n.
- The Machine Vision Market: 1997 results and forecasts through 2002*. **Arbor, Ann. 1998.** MI, USA : (AIA), Automed Imaging Association,, 1998.
- VALE, GIOVANE MAIA DO e POZ, ALUIR PORFÍRIO DAL. 2002.** PROCESSO DE DETECÇÃO DE BORDAS DE CANNY. 2002, Vol. v8, pp. 67-78.
- Wikipedia.** Wikipedia. *Wikipedia*. [Online] [Citação: 25 de Junho de 2007.] www.wikipedia.org.
- XG, Metrolog.** www.metrologxg.fr : s.n.
- ZIOU, D. and TABBONE, S. 1998.** Edge Detection Techniques - An Overview. [ed.] Université de Sherbrooke. 1998, Vol. 8, pp. 537-559.