



UNIVERSIDADE LUTERANA DO BRASIL
PRÓ-REITORIA DE GRADUAÇÃO
DEPARTAMENTO DE ENGENHARIA ELÉTRICA



Rodrigo Soares Bolicó

DETECÇÃO E CLASSIFICAÇÃO DE PONTO DE SOLDA EM
CHAPAS

Canoas, Junho de 2008



Rodrigo Soares Bolico

Detecção e Classificação de Ponto de Solda em Chapas

Trabalho de Conclusão de Curso apresentado ao Departamento de Engenharia Elétrica da ULBRA como um dos requisitos obrigatórios para a obtenção do grau de Engenheiro Eletricista

Departamento:

Engenharia Elétrica

Área de Concentração

Processamento de imagens

Professor Orientador:

MSc. Eng. Eletr. Dalton Luiz Rech Vidor – CREA-RS : 079005-D

Canoas

2008



FOLHA DE APROVAÇÃO

Nome do Autor: Rodrigo Soares Bolico

Matrícula: 011102020-4

Título: Detecção e Classificação de Ponto de Solda em Chapas

Trabalho de Conclusão de Curso apresentado ao Departamento de Engenharia Elétrica da ULBRA como um dos requisitos obrigatórios para a obtenção do grau de Engenheiro Eletricista

Professor Orientador:

MSc. Eng. Eletr. Dalton Luiz Rech Vidor

CREA-RS: 079005-D

Banca Avaliadora:

MSc. Eng. Eletr. Adriane Parraga

CREA: 107196-D

Dr. Eng. Eletr. Valner João Brusamarello

CREA: RS078158-D

Assinaturas:

Autor
Rodrigo Soares Bolico

Orientador
Dalton Luiz Rech Vidor

Avaliador
Adriane Parraga

Avaliador
Valner João Brusamarello

Relatório Aprovado em:



DEDICATÓRIA

Dedico este trabalho a minha mãe e meu pai, a minha esposa, aos meus irmãos, aos meus colegas de trabalho e curso e a meus mestres. Em memória a Daniel Korb.



AGRADECIMENTOS

A todos que colaboraram direta ou indiretamente na elaboração deste trabalho, o meu reconhecimento.

Ao Professor Dalton Vidor pelo estímulo, dedicação e esforço pessoal proporcionado.

Aos colegas de trabalho pelas sugestões e observações valiosas.

Ao Professor Valner João Brussamarello pelas valiosas contribuições.

Aos colegas de curso pelo bom relacionamento e companheirismo durante os semestres.



EPÍGRAFE

“Determinação coragem e autoconfiança são fatores decisivos para o sucesso. Se estamos possuídos por uma inabalável determinação conseguiremos superá-los. Independentemente das circunstâncias, devemos ser sempre humildes, recatados e despidos de orgulho.” (Dalai Lama)



RESUMO

Bolico, Rodrigo. **Detecção e Classificação de Ponto de Solda em Chapas.** Trabalho de Conclusão de Curso em Engenharia Elétrica - Departamento de Engenharia Elétrica. Universidade Luterana do Brasil. Canoas, RS. 2008

Este trabalho consiste em classificar pontos de solda em chapas, em ponto bom ou ponto ruim, o mesmo visa aplicação em fábricas montadoras de automóvel na área de funilaria. Esta detecção é feita utilizando o *toolbox* de *Image Acquisition* e *Image Processing* do Matlab, além de um arquivo executável desenvolvido no software Delphi. Os algoritmos desenvolvidos serão distintos para cada software. Para classificar o ponto de solda será considerado o diâmetro da solda como parâmetro principal da análise. Para executar o trabalho o programa capturará as imagens, processará as mesmas, determinará o diâmetro do ponto de solda e exibirá o resultado do processamento.

Palavras-chaves: Ponto, Solda, Matlab, *Image Acquisition*, *Image Processing*.



ABSTRACT

Bolico, Rodrigo. Detention and Classification of Weld PointTitle in Plates. Work of Conclusion of Course in Electric Engineering - Department of Electric Engineering. Luterana university of Brazil. Canoas, RS. 2008

This work consists of classifying weld points in plates, in good point or bad point. The project intend to be used in plants assembly plants of automobile in the auto body area. This detention is made using *toolbox* of *Image Acquisition* and *Image Processing* of the Matlab. Also an executable archive will be developed in Delphi software. The algorithms developed will be distinct for each software. To classify the weld point, the diameter of the weld will be considered as main parameter of the analysis. To execute the work the program will capture and process the images, to determine the diameter of the weld point and to show the results.

Key-Words: Point, Weld, Matlab, *Image Acquisition*, *Image Processing*.



LISTA DE ILUSTRAÇÕES

Figura 1.1 - Imagem de corpos de prova padrão.....	2
Figura 1.2 - Imagem de um painel interno da porta traseira do Celta	3
Figura 1.3 - Imagem de teste de solda manual em corpo de prova padrão.....	5
Figura 2.1 - Imagem digital com convenção dos eixos.....	11
Figura 2.2 - Imagem da vizinhança-de-4 de um pixel p	12
Figura 2.3 - Imagem da vizinhança-de-8 de um pixel p	13
Figura 2.4 - Etapas do processamento de imagens	15
Figura 2.5 - Imagem original colorida de um ponto de solda	16
Figura 2.6 - (a) Imagem de um ponto de solda em tons de cinza (b) seu Histograma	16
Figura 2.7 - (a) Imagem de um ponto de solda com equalização do histograma (b) Histograma equalizado	17
Figura 2.8 - (a) Imagem de corpo de prova com pontos de solda (b) Região de interesse.....	19
Figura 2.9 - Cores do espectro visível.....	20
Figura 3.1 - Circuito secundário típico de uma solda ponto	25
Figura 3.2 - (a) Imagem do início da solda (b) Imagem da geração de calor	26
Figura 3.3 - (a) Imagem de corpos de prova padrão sobrepostos (b) Imagem do diâmetro do ponto.....	27
Figura 3.4 - Montagem do sistema de testes	28
Figura 3.5 - Imagem de falha na iluminação do ponto.....	29
Figura 3.6 - Diagrama de blocos do código fonte do software.....	30
Figura 3.7 - Fluxograma geral de funcionamento	31
Figura 3.8 - Aquisição da Imagem Original.....	33
Figura 3.9 - Imagem em tons de cinza.....	33
Figura 3.10 - Imagem com filtro Threshold.....	34
Figura 3.11 - Imagem transformada binário.....	35
Figura 3.12 - (a) Imagem de uma imagem A (b) Imagem de um elemento estrutural.....	36
Figura 3.13 - Imagem suavizada com estruturas suprimidas.....	36
Figura 3.14 - Pontos da borda detectados.....	38
Figura 3.15 - Circunferência traçada	39
Figura 3.16 - Imagem original com resultado do processamento	40
Figura 3.17 - Fluxograma de análise dos dados	41
Figura 3.18 - Aquisição da imagem original Delphi 1	42
Figura 3.19 - Imagem em tons de cinza	42
Figura 3.20 - Imagem com ajuste de contraste.....	43
Figura 3.21 - Imagem com filtro Threshold Delphi 1	44
Figura 3.22 - Resultado do processamento	46
Figura 4.1 - Amostra 1 boa.....	49
Figura 4.2 - Amostra 1 boa.....	49
Figura 4.3 - Amostra 2 boa.....	49
Figura 4.4 - Amostra 2 boa.....	49
Figura 4.5 - Amostra 3 boa.....	49
Figura 4.6 - Amostra 3 boa.....	49
Figura 4.7 - Amostra 4 boa.....	49
Figura 4.8 - Amostra 4 boa.....	49



Figura 4.9 – Amostra 5 boa.....	50
Figura 4.10 – Amostra 5 boa.....	50
Figura 4.11 – Amostra 6 boa.....	50
Figura 4.12 – Amostra 6 boa.....	50
Figura 4.13 – (a) Arrancamento aprovado amostra 4 (b) Arrancamento aprovado amostra 5	50
Figura 4.14 – Amostra 21 ruim.....	53
Figura 4.15 – Amostra 21 ruim.....	53
Figura 4.16 – Amostra 22 ruim.....	53
Figura 4.17 – Amostra 22 ruim.....	53
Figura 4.18 – Amostra 23 ruim.....	53
Figura 4.19 – Amostra 23 ruim.....	53
Figura 4.20 – Amostra 24 ruim.....	53
Figura 4.21 – Amostra 24 ruim.....	53
Figura 4.22 – Amostra 25 ruim.....	54
Figura 4.23 – Amostra 25 ruim.....	54
Figura 4.24 – Amostra 26 ruim.....	54
Figura 4.25 – Amostra 26 ruim.....	54
Figura 4.26 – (a) Arrancamento reprovado amostra 21 (b) Arrancamento reprovado amostra 25.....	54



LISTA DE TABELAS

Tabela 1 – Orçamento do projeto	8
Tabela 2 – Cronograma orçamentário	9
Tabela 3 – Tabela de custos	22
Tabela 4 – Relação espessura da chapa e diâmetro do ponto	27
Tabela 5 – Medidas de ponto bom	49
Tabela 6 – Medidas de ponto ruim	52



LISTA DE ABREVIATURAS E SIGLAS

RGB- Imagem formada pela junção de três componentes de cores, vermelha, verde e azul.

PIXEL- Elemento da imagem

ACB- Analise custo-benefício

CCD- *Charge Coupled Device*

BMP- Extensão para arquivo Bitmap

GIF- *Graphics Interchange format*

JPEG- *Joint Photographers Expert group*

PC- Computador



SUMÁRIO

1	INTRODUÇÃO	1
1.1	Visão Geral do Problema	2
1.2	Formulação do Problema de Engenharia	2
1.3	Formulação do Problema Comercial	2
1.4	Estudos de mercado	3
1.1.1	<i>Identificação dos Interessados</i>	3
1.1.2	<i>Oportunidade de Negócios</i>	4
1.5	Definição do Escopo do Projeto	4
1.1.3	<i>Objetivos Gerais</i>	4
1.1.4	<i>Objetivos Específicos</i>	4
1.1.5	<i>Metas</i>	5
1.1.6	<i>Resultados Previstos</i>	5
1.6	Universo das Soluções	5
1.7	Justificativa Técnica e Econômica para a Solução Escolhida	5
1.1.7	<i>Critérios</i>	6
1.1.8	<i>Benchmark</i>	6
1.8	Especificações Técnicas da Solução Escolhida	6
1.9	Localização do Projeto e Escala de Produção	7
1.10	Recursos	7
1.1.9	<i>Orçamento do Projeto</i>	8
1.1.10	<i>Financiamento e cronograma orçamentário</i>	9
1.11	Riscos Internos e Externos da Execução	9
1.12	Índices de Qualidade	9
2	REFERENCIAL TEÓRICO	10
2.1	Histórico do Processamento de Imagens	10
2.2	Processamento de Imagens	11
2.1.1	<i>Pixel e Conectividade</i>	12
2.1.2	<i>Apresentação e Armazenamento das Imagens</i>	13
2.3	Conceitos Básicos	14
2.1.3	<i>Etapas do processamento de imagens</i>	15
2.4	Digitalização de Imagens	20
2.5	Tipos de Imagens	20
2.6	Detecção de Bordas	21
2.7	Limiarização	21
2.8	Tabela de Custos do Sistema	22
3	MATERIAIS E MÉTODOS	24
3.1	Solda por Resistência	24
3.1.1	<i>Princípios Gerais de Solda por Resistência</i>	24
3.1.2	<i>Qualidade de Solda</i>	26
3.2	Descrição Geral do Sistema	27
3.3	Descrição da Funcionalidade do Software	30
3.1.3	<i>Fluxograma</i>	30
3.1.4	<i>Softwares implementados</i>	32



4 APRESENTAÇÃO DOS RESULTADOS OBTIDOS.....	47
5 CONCLUSÃO.....	55
SUGESTÕES PARA TRABALHOS FUTUROS	56
REFERÊNCIAS BIBLIOGRÁFICAS.....	57
APÊNDICE A – CÓDIGO FONTE PROGRAMA MATLAB	59
APÊNDICE B – CÓDIGO FONTE PROGRAMA DELPHI 1	62
ANEXO A- DADOS DO WEBCAM	67



1 INTRODUÇÃO

O processamento de imagens tem sido aplicado em diversas áreas do conhecimento, como a interpretação de fotos aéreas e visualização de imagens para diagnósticos médicos. Na indústria sua aplicação está voltada à orientação e ao posicionamento de objetos, à inspeção automática de circuitos impressos entre outros (BUENO, 2000).

Segundo (PROENÇA et al, 1999), o processamento das imagens vem se apresentando como importantes ferramentas para algumas aplicações industriais. Estas aplicações são reunidas no campo conhecido como visão de máquina. A visão de máquina pode ser subdividida em duas áreas principais: Robótica e inspeção visual automatizada.

A visão de máquina provê inovações para a automação industrial. Como exemplo, pode-se citar o controle de qualidade de componentes eletrônicos, qualidade de produtos têxteis, produtos de vidro, parte específica de máquinas entre outros.

Em um sistema de processamento de imagens, para a aplicação de inspeção de produtos, o sistema deve identificar a falta ou a aceitação ou não de um produto, de acordo com um padrão pré-estabelecido.

Dentre as aplicações industriais a inspeção visual vem se destacando com novas tecnologias que garante a qualidade do produto, aumento da produtividade e confiabilidade dos mesmos.

Atualmente existem no mercado de montadoras de veículos apenas duas ferramentas para detecção e classificação de ponto de solda. O primeiro é o teste de solda ponto manual, o segundo é o teste de solda por ultra som ainda pouco utilizado por se tratar de uma tecnologia nova e de alto custo.

No presente trabalho será desenvolvida uma nova ferramenta para teste de solda, com baixo custo, de simples utilização para o usuário e que garanta a confiabilidade do processo.

1.1 Visão Geral do Problema

Nos dias de hoje, com o consumidor cada vez mais exigente com a qualidade do produto e o mercado automobilístico cada dia mais competitivo, faz-se necessário que as empresas busquem processos cada vez menos suscetíveis a erros e com o menor custo. Uma alternativa é a automação destes processos. Em uma linha de funilaria, o processo de classificação de ponto de solda nas chapas das carrocerias é feito manualmente, com isto é utilizada mão-de-obra em excesso, gerando um custo adicional para a empresa.

1.2 Formulação do Problema de Engenharia

O objetivo é tornar o processo de teste de solda ponto automático, garantindo a qualidade das carrocerias em áreas de funilaria nas montadoras de automóveis, aumentando a produtividade e eliminando custo com testes destrutivos. Este projeto será desenvolvido fora da linha de montagem aplicando testes em corpos de prova padrão com as mesmas características físicas das chapas utilizadas para fabricação de um automóvel. A Figura 1.1 mostra uma imagem de corpos de prova padrão com dois pontos de solda em cada amostra.

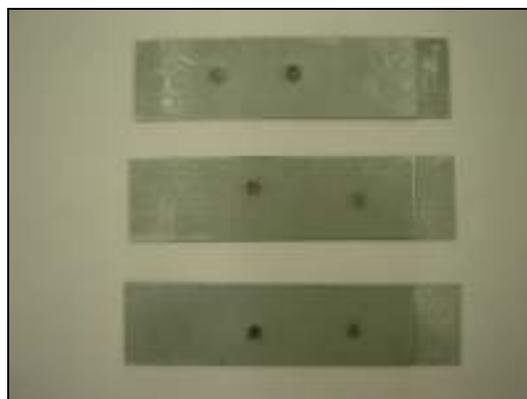


Figura 1.1 - Imagem de corpos de prova padrão

1.3 Formulação do Problema Comercial

Atualmente nas grandes montadoras de automóveis um dos principais problemas na área de funilaria é a classificação de pontos de solda. Para este processo a ferramenta mais utilizada é a mão-de-obra humana efetuando testes manuais com martelo e talhadeira. Com isso é utilizado um grupo de funcionários que necessitam efetuar testes destrutivos, aumentando assim o custo estrutural da

empresa. Além disso, uma falha neste processo pode causar o envio de uma unidade com ponto solto para venda nas concessionárias. Nessa situação faz com que a empresa lance um retrabalho de centenas de veículos. A Figura 1.2 mostra a imagem de um painel interno de porta traseira de um Celta que passou por um teste destrutivo.



Figura 1.2 - Imagem de um painel interno da porta traseira do Celta

1.4 Estudos de mercado

1.1.1 Identificação dos Interessados

Existe no mercado automobilístico brasileiro uma nova tecnologia para testes de solda ponto em chapas, porém essa tecnologia é pouco usada. A mesma tem se expandido mais no mercado Europeu. Trata-se do teste de solda ponto por ultra-som. Este equipamento tem um custo de U\$\$ 45.000,00 e não possui uma interface simples para quem opera, ou seja, necessita de um treinamento específico para um grupo de funcionários. Já no Brasil a ferramenta mais utilizada é a mão-de-obra humana efetuando testes com martelo e talhadeira.

Com este projeto, será possível oferecer às montadoras de automóveis uma solução de baixo custo, prática, e de fácil acesso. A incorporação deste sistema à linha de funilaria se tornará um diferencial, o qual, além de alavancar as vendas do produto, também contribuirá com a melhoria da qualidade do mesmo.



1.1.2 Oportunidade de Negócios

Este projeto prevê tornar o processo de inspeção de solda ponto mais preciso, rápido e econômico através de um processo não destrutivo, reduzindo os testes manuais. Com uma proposta mais simples e economicamente mais acessível este projeto pode ser implementado em grandes montadoras como General Motors, Mercedes, Daimler-Chrysler, GME (Nearly every plant), BMW, Audi, Hyundai, Fiat, Renault, Ford, Jaguar, Porsche, Seat.

1.5 Definição do Escopo do Projeto

1.1.3 Objetivos Gerais

Este projeto tem por objetivo detectar e classificar pontos de solda em chapas. O mesmo será de fácil acesso, baixo custo e pouca manutenção. A aquisição da imagem será feita através de um webcam de 1.4 Megapixels. O sistema será desenvolvido no Matlab e Delphi para detecção do diâmetro do botão de solda e uma comparação dos padrões de solda. A iluminação será feita com uma lâmpada dicróica de cinquenta watts. O suporte do webcam será feito em aço 1020 pintado de preto e preso na base da mesa de análise.

Neste trabalho, o uso do Matlab será apenas para validar o método de aquisição e manipulação de imagens. Para implementar em uma linha de produção, deve-se construir o código do programa em um software dedicado, pois o custo da licença do Matlab é muito elevado para uso comercial.

1.1.4 Objetivos Específicos

Este projeto está sendo desenvolvido para uma futura instalação na montadora de automóvel General Motors planta Gravataí. Esta ferramenta será utilizada pelo grupo de teste de solda ponto manual, visando à redução de teste manual. A Figura 1.3 mostra como é realizado o teste manual.



Figura 1.3 - Imagem de teste de solda manual em corpo de prova padrão

1.1.5 Metas

Tornar o processo de inspeção de solda ponto mais confiável, rápido e econômico através de um processo não destrutivo reduzindo os testes manuais.

1.1.6 Resultados Previstos

Se este projeto atender as necessidades da General Motors, local onde será implementado, será iniciado um estudo para viabilizá-lo comercialmente entre as demais montadoras de veículos.

1.6 Universo das Soluções

Atualmente existem no mercado de montadoras de veículos apenas dois tipos de ferramenta para detecção e classificação de ponto de solda. O primeiro é o teste de solda ponto manual utilizado em todas as montadoras, o segundo é o teste de solda por ultra som utilizado somente nas grandes montadoras por ser uma tecnologia nova e de alto custo.

1.7 Justificativa Técnica e Econômica para a Solução Escolhida

O projeto requer um estudo aprofundado do Matlab e conhecimento em programação Pascal para o software Delphi, também se faz necessário o entendimento de aquisição de imagens e definição das mesmas. Para tal solução foi



feito estudo análise custo benefício (ACB) levando em consideração as ferramentas utilizadas, equipe de projeto, instalações, *up grade* de equipamentos a cada três anos, manutenção, treinamento e assistência técnica. O resultado comparado com as demais soluções mostrou uma economia de 60% do custo total de instalação.

1.1.7 Critérios

A viabilidade técnica, econômica e a maturidade da solução foram os critérios utilizados para desenvolvimento deste projeto.

1.1.8 Benchmark

Tecnicamente, o sistema de teste de solda em chapas através do processamento de imagens se torna mais viável por requerer conhecimentos de programação em Delphi e Matlab, quando comparado ao sistema de teste de solda por ultra som. O ultra som é uma tecnologia de alto custo e atualmente utilizada apenas no mercado Europeu. A análise do ponto de solda é feita através de um software que interpreta a vibração de uma bolha de água quando colocada sobre o ponto. Para análise desta oscilação necessita de uma modelagem matemática que decompõe formas de ondas em séries de Fourier e compara com determinados padrões. Tratando-se do custo benefício o teste de solda por imagem apresenta uma redução de custo de 60% para a empresa.

1.8 Especificações Técnicas da Solução Escolhida

Este projeto prevê a construção de um software para gerenciamento de imagens de pontos de solda, realizando a detecção de imperfeições e ainda criando uma interface gráfica com o usuário. Para execução do mesmo será utilizado um PC como ferramenta para execução do software, licença Matlab e Delphi estudante, um Webcam de 1.4 megapixel e suporte para iluminação.



1.9 Localização do Projeto e Escala de Produção

O desenvolvimento deste projeto ocorrerá nas dependências da empresa General Motors do Brasil Ltda. localizada na cidade de Gravataí, que incorporará este sistema fora da linha de produção, sendo aplicados os testes somente em peças que abastecem a linha e podem ser retiradas temporariamente. O detector de chapas estará localizado na área da qualidade que efetua os testes manuais.

Dentre as restrições para uma boa detecção, pode-se citar: o ponto deve estar concêntrico com a câmera, ou seja, o centro da imagem da câmera deve estar alinhado com o centro do ponto e, a distância do ponto para a câmera deve ser sempre a mesma.

1.10 Recursos

Para execução deste projeto os recursos materiais são webcam, software Matlab e Delphi, suporte para câmera, um PC e corpos de prova para teste. Para o desenvolvimento do software é necessário conhecimento em programação Pascal, linguagem Matlab, aquisição e tratamento de imagens.



1.1.9 Orçamento do Projeto

Tabela 1 – Orçamento do projeto

Gastos Planejados (R\$)						
	20__ Trim. 1	20__ Trim. 2	20__ Trim. 3	20__ Trim. 4	Total	Comentários
<i>Staff Interno</i>	3000,00	2000,00	1000,0	1000,00	7000,00	Desenvolvimento
<i>Serviços</i>	300,00	200,00	200,00	200,00	900,00	Pintura
<i>Ferramentas de Software</i>	2473,00				2473,00	Aquisição de um Pc, licença software
<i>Hardware</i>	180,00				180,00	Dispositivo
<i>Materiais e acessórios</i>	250,00				250,00	Iluminação
<i>Instalações</i>	216,00				216,00	Mão-de-obra
<i>Telecomunicações</i>	80,00	100,00	100,00	150,00	430,00	
<i>Treinamento</i>	800,00	200,00	200,00	200,00	1400,00	20hs (3 pessoas)
<i>Contingência (Risco)</i>	1000,00				1000,00	
Total	8299,00	2500,00	1500,00	1550,00	13849,00	
Explicação: os valores acima estimados são referentes ao projeto completo.						
Fontes de Recursos (R\$)						
	20__ Trim. 1	20__ Trim. 2	20__ Trim. 3	20__ Trim. 4	Total	Comentários
<i>Fundos Gerais</i>	1000,00	500,00	300,00	200,00	2000,00	Impostos
<i>Fundos não gerais</i>	500,00	500,00	500,00	500,00	2000,00	Gastos Internos
<i>Federais</i>	400,00	400,00	400,00	400,00	1600,00	Impostos
<i>Outros</i>	200,00	200,00	200,00	200,00	800,00	Risco Interno
Total	2100,00	1600,00	1400,00	1300,00	6400,00	
Explicação: os valores acima estimados são referentes ao projeto completo.						



1.1.10 Financiamento e cronograma orçamentário

Tabela 2 – Cronograma orçamentário

1º Trimestre	2º Trimestre	3º Trimestre	4º Trimestre
R\$ 4.000,00	R\$ 2.000,00	R\$ 2.000,00	R\$ 8.000,00

1.11 Riscos Internos e Externos da Execução

O risco interno na execução é o projeto não atender as especificações esperadas, como grau de confiabilidade da imagem medida.

1.12 Índices de Qualidade

O resultado final será amostrado através de testes em pontos de solda de corpo de prova padrão, verificando a confiabilidade do produto, incerteza de medição e comparando os resultados com testes destrutivos das peças.



2 REFERENCIAL TEÓRICO

2.1 Histórico do Processamento de Imagens

Segundo (GONZALES; WOODS, 2000), as primeiras referências a aplicações de processamento de imagens surgiram no início do século passado, mais precisamente nos anos vinte, com a utilização da transmissão de imagens digitalizadas, via cabo submarino, entre Londres e Nova York. As técnicas mais modernas de processamento de imagens só ocorreram a partir da terceira geração dos computadores, no início dos anos sessenta. Atribui-se a Larry Roberts, que em 1962 era estudante de doutorado no MIT (*Massachusetts Institute of Technology*), o laurel de precursor dessas técnicas. Na sua tese de doutorado, Roberts fez uma série de estudos sobre processamento de imagens e inventou ferramentas até hoje usadas, tais como a detecção de bordas pelo operador que levou seu nome – *Robert edges detector*.

Outros eventos que merecem destaque:

- Pesquisadores do Jet Propulsion Laboratory da NASA em Pasadena – Califórnia utilizou processamento de imagens por computador, para corrigir vários tipos de distorções em imagens da lua enviadas por uma sonda Ranger, em 1964.
- O desenvolvimento da abordagem *bottom-up* para processamento de imagens (um ponto de vista diferenciado a respeito das pesquisas até então dominado pela abordagem *top-down*) levado a termo por David Marr, um cientista multidisciplinar com formação em matemática e neurociência. Marr faleceu antes de ver publicado seu livro mais importante “*Vision: A Computational Investigation into the Human Representation and Processing of Visual Information*”, o que ocorreu em 1982.
- Criação da IAPR – International Association for Pattern Recognition em 1978. A IAPR é uma associação sem fins lucrativos voltada ao estudo

e aplicações de técnicas de reconhecimento de padrões, visão computacional, processamento de imagens e áreas afins.

2.2 Processamento de Imagens

A imagem digital é a matéria prima para desenvolvimento de projetos de processamento de imagens. A mesma refere-se à função bidimensional de intensidade da luz $f(x,y)$, onde x e y denotam as coordenadas espaciais e o valor de f em qualquer ponto (x,y) é proporcional ao brilho (ou níveis de cinza) da imagem naquele ponto.

Uma imagem digital é uma imagem $f(x,y)$ discretizada tanto em coordenadas espaciais quanto em brilho. Uma imagem digital pode ser considerada como sendo uma matriz cujos índices de linhas e de colunas identificam um ponto na imagem, e o correspondente valor do elemento da matriz identifica o nível de cinza naquele ponto. Os elementos dessa matriz digital são chamados de *elementos da imagem*, *elementos da figura*, “*pixels*” ou “*pels*”. Estes dois últimos, abreviações de “Picture elements” (elemento da figura). A Figura 2.1 ilustra a convenção dos eixos para representação de imagens digitais.



Figura 2.1 – Imagem digital com convenção dos eixos
Fonte: <http://atlas.ucpel.tche.br/vbastos/>

A Figura 2.1 é uma imagem de 8 bits, o que significa que cada *pixel* da imagem tem um valor referente ao nível de cinza naquele ponto que está entre 0 (preto) e 255 (branco). Já em uma imagem digital colorida de 24 bits, por exemplo, cada *pixel* da imagem teria seu valor dividido em 3 valores de 8 bits que seriam referentes ao modelo de cores daquela imagem. Caso esse modelo de cores seja o

RGB (Vermelho, Verde e Azul), esses valores indicariam o nível de vermelho entre 0 e 255, o nível de verde entre 0 e 255 e o nível de azul entre 0 e 255.

A transformação de uma imagem digital colorida de 24 bits com modelo de cores RGB em uma imagem digital monocromática de 8 bits, é necessário somar os valores dos níveis de intensidade de cada uma das cores de cada *pixel*, dividi-los por 3 e atribuir esse valor ao *pixel* novamente.

2.1.1 Pixel e Conectividade

Um pixel p nas coordenadas (x,y) possui quatro vizinhos horizontais e verticais, cujas coordenadas são dadas por

$$(x+1,y), (x-1,y), (x,y+1), (x,y-1).$$

Esse conjunto de pixel, chamado de vizinhança-de-4 de p , é representado por $N_4(p)$. Cada pixel está a uma unidade de distância de (x,y) , sendo que alguns vizinhos de p ficarão fora da imagem digital se (x,y) estiver na borda da imagem.

Os quatro vizinhos diagonais de p possuem como coordenadas

$$(x+1,y+1), (x+1,y-1), (x-1,y+1), (x-1,y-1)$$

e são denotados por $N_D(p)$. Esses pontos, juntos com a vizinhança-de-4, são chamados de vizinhança-de-8 de p , representada por $N_8(p)$. Como antes alguns dos pontos de $N_8(p)$ cairão fora da imagem quando (x,y) se encontrar na borda da imagem.

A vizinhança-de-4 de um pixel p é definida pelo conjunto dos pixels adjacentes a p , não levando em conta os pixels localizados nas diagonais passando por p , como mostra a Figura 2.2.

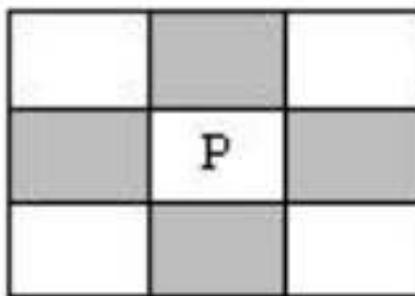


Figura 2.2 – Imagem da vizinhança-de-4 de um pixel p

A vizinhança-de-8 de um pixel p é formada pelo conjunto de todos os pixels que são adjacentes a p , como mostra a Figura 2.3.

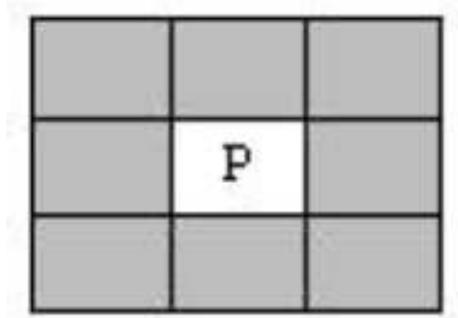


Figura 2.3 – Imagem da vizinhança-de-8 de um pixel p

A conectividade entre pixels é um conceito importante usado no estabelecimento das bordas de objeto e componentes de região de uma imagem. Para estabelecer se dois pixels estão conectados, é preciso determinar se eles são de alguma forma adjacente (digamos, se são vizinhos-de-4) e se seus níveis de cinza satisfazem certo critério de similaridade (digamos, se eles são iguais). Em uma imagem binária com valores 0 e 1, dois pixels podem ser vizinhos-de-4, mas eles não são ditos conectados a menos que tenham o mesmo valor.

2.1.2 Apresentação e Armazenamento das Imagens

Segundo Tessaro, 2004 as imagens podem ser apresentadas e armazenadas em diferentes formas. Na imagem monocromática preta e branca cada pixel é representado por um bit (0 ou 1). Uma imagem monocromática 640 x 480 ocupa 37.5 KB em memória.

Na imagem monocromática em tons de cinza, cada pixel é representado por um byte (valor entre 0 e 255). Uma imagem em escala de cinza 640 x 480 ocupa 300 KB em memória.

Para imagem a cores de 8 bits (256 cores), cada pixel é representado por um byte. Neste caso suporta 256 cores. Uma imagem a cores de 8-bits com resolução 640 x 480 ocupa 300 KB em memória.

Em uma imagem a cores de 24 bits, cada pixel é representado por três bytes (RGB). Suportam 256 x 256 x 256 cores (16.777.216). Uma imagem a cores de 24-bits com resolução 640 x 480 ocupa 900 KB em memória.

Com relação aos formatos de arquivo tem-se a seguir alguns dos mais utilizados:



BMP – (*Bitmap*), que permite que o *Windows* mostre a imagem em qualquer dispositivo de visualização. É chamado independente do dispositivo porque a cor do pixel é especificada de forma independente do método usado pela característica de representação de cor. O tamanho do arquivo é diretamente associado às características da imagem (tamanho e resolução da imagem). O trabalho com imagens do tipo bitmap pode ser extremamente complexo e muito dependentes de softwares profissionais (ex: *Adobe Photoshop*) que disponibilizam ferramentas poderosas para a seleção de áreas precisas da imagem.

GIF- *Graphics Interchange Format*: este formato é bastante comum nas transmissões de rede e limitado a 256 cores. É um formato de alta qualidade, que possibilita a visualização de informação gráfica de alta resolução numa grande variedade de dispositivos. Definido em termos de blocos e sub-blocos que contêm parâmetros e informação relevante para a reprodução da imagem.

JPEG: *Joint Photographers Expert Group* – (nome do comitê ISO que criou a norma). Este formato foi criado para comprimir imagens coloridas ou pretas e brancas de cenas reais e naturais. Funciona bem em fotografia. Segundo (Ramos, 2000) o formato JPEG é o mais utilizado no armazenamento e na transmissão de imagens estáticas em multimídia e na *internet*. A passagem suave entre cores é representada de forma mais fiel em menor espaço em JPEG do que em GIF. Como vantagem do JPEG, pode-se citar a capacidade de compressão, por exemplo, 20:1, onde um arquivo é compactado de 2 MB para 100 KB.

2.3 Conceitos Básicos

Segundo Gonzáles, 1992, o processamento de imagens digitais abrange uma ampla escala de hardware, software e fundamentos teóricos. Na Figura 2.4 estão representados os passos que desenvolvem todo estudo de processamento digital de imagens.

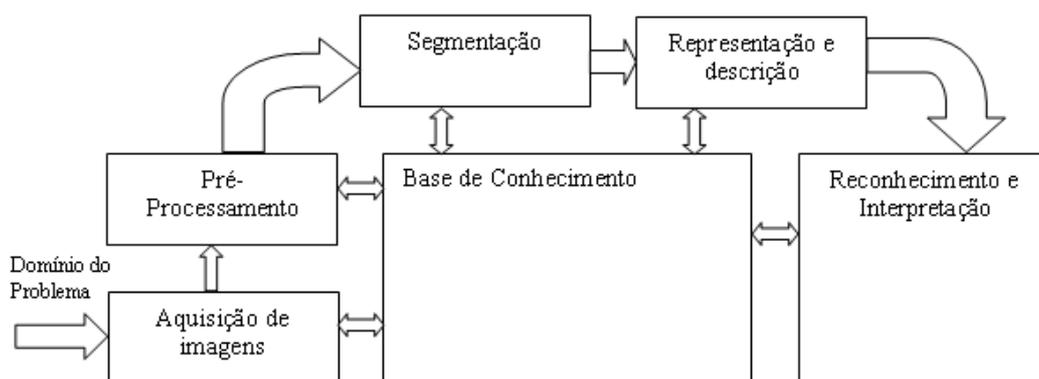


Figura 2.4 – Etapas do processamento de imagens

Fonte: GONZALES; WOODS, 2000

2.1.3 Etapas do processamento de imagens

As etapas básicas do processamento de imagens são descritas a seguir:

2.3.3.1 Aquisição da imagem

Segundo Gonzáles, 2000, para a aquisição de imagens digitais são necessários dois dispositivos. O primeiro é um dispositivo físico que seja sensível a uma banda do espectro de energia eletromagnética (como raios X, ultravioleta, visível, ou banda infravermelha) e que produza um sinal elétrico de saída proporcional a um nível de energia percebida. O segundo, chamado de *digitalizador*, é um dispositivo para a conversão da saída elétrica de um dispositivo de sensoriamento físico para a forma digital. Um exemplo de dispositivos de aquisição de imagens seria os Scanners, Webcams, Câmeras Digitais.

As imagens são capturadas e convertidas em uma imagem digital. Um exemplo de dispositivo bastante utilizado para este fim é a câmera CCD (*Charge Couple Device* – Dispositivo de Carga Acoplado). Este tipo de dispositivo utiliza uma matriz de células fotossensíveis que atuam como capacitores, armazenando carga elétrica na proporção da energia luminosa incidente.

2.3.3.2 Pré-processamento

A função do pré-processamento é melhorar a imagem (GONZALES; WOODS, 2000). Nesta etapa, são utilizadas técnicas para aumento de contraste, remoção de ruídos, realce, normalização, com o objetivo de converter os padrões para uma forma que possibilite uma simplificação do posterior processo de reconhecimento (RODRIGUES, 2002).

Podem-se citar algumas etapas do pré-processamento: transformação em níveis de cinza, histogramas, ajuste da imagem ou contraste e filtragem. A Figura 2.5 mostra uma imagem original colorida de um ponto de solda em um corpo de prova padrão.

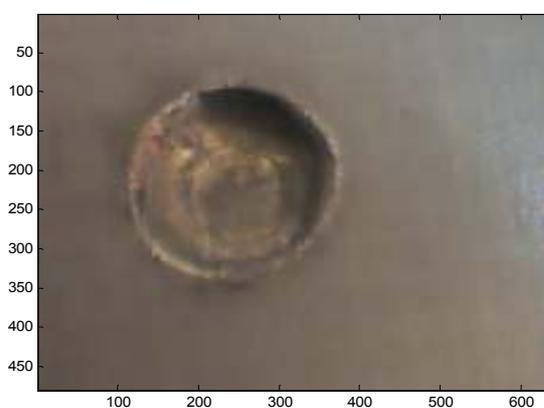


Figura 2.5 – Imagem original colorida de um ponto de solda

A seguir, a imagem do ponto de solda em um corpo de prova padrão é transformada em escalas de cinza como mostra a Figura 2.6.

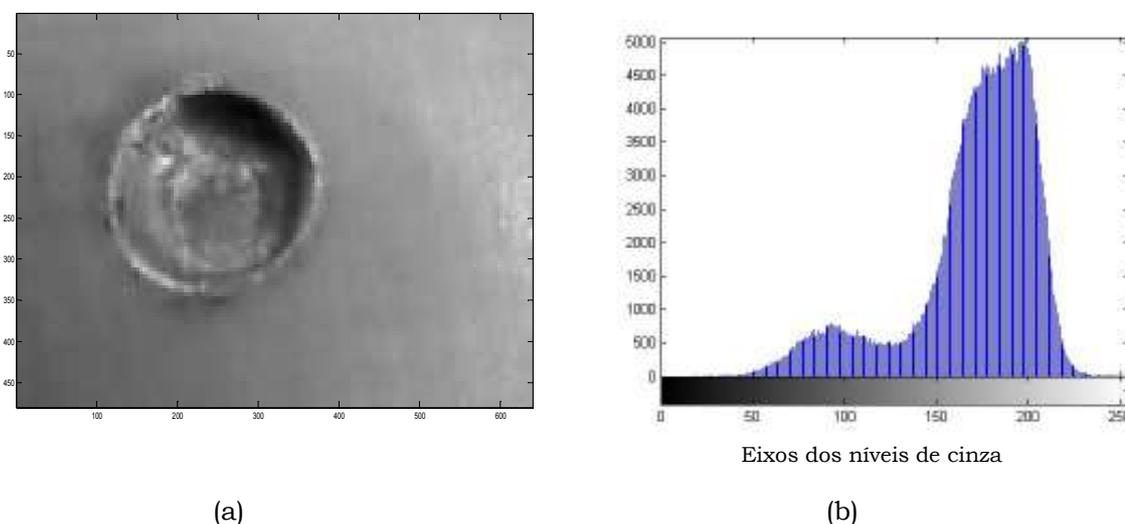


Figura 2.6 – (a) Imagem de um ponto de solda em tons de cinza (b) seu Histograma

Na mesma figura tem-se o histograma da imagem. O histograma de uma imagem consiste em um gráfico de barras que fornece a ocorrência para cada nível de cinza o número (ou o percentual) de pixels correspondentes na imagem. Através da utilização do histograma de uma imagem obtém-se uma indicação de sua qualidade quanto ao nível de contraste e quanto ao seu brilho médio (se a imagem é predominantemente clara ou escura).

Outra etapa possível é a equalização do histograma, que modifica o histograma da imagem original de tal forma que se obtenha um histograma com distribuição de níveis de cinza uniforme, aparecendo na imagem com mesma frequência. Tem-se na Figura 2.7(a) a imagem referente ao histograma equalizado e na Figura 2.7(b) o histograma representativo de tal operação.

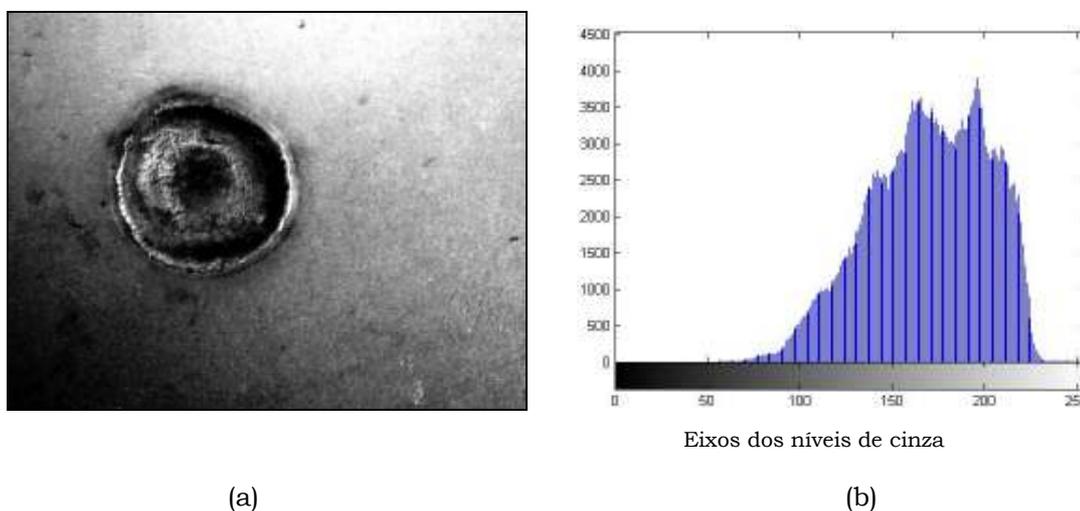


Figura 2.7 – (a) Imagem de um ponto de solda com equalização do histograma (b) Histograma equalizado

Na prática a obtenção de um histograma perfeitamente uniforme não é possível devido à natureza das imagens, cujos *pixels* podem assumir somente valores discretos. O processo de equalização pode levar a falsos contornos na imagem, pois o número de níveis de cinza pode ser reduzido. Segundo (RODRIGUES, 2002) a equalização pode ter um efeito de realce notável da imagem, mas o contraste pode ser muito forte e, por isto a equalização de histograma deve ser usada e comparada com outros métodos para verificar sua eficácia.

2.3.3.3 Segmentação

A segmentação particiona uma imagem de entrada em suas partes constituintes ou objetos. Cada uma destas partes é uniforme e homogênea com respeito a algumas propriedades da imagem, como por exemplo, cor e textura.

Segundo (GONZALES; WOODS, 2000), algoritmos de segmentação são geralmente baseados em duas propriedades básicas: similaridade e descontinuidade.

Na similaridade, a partição é efetuada com base na semelhança entre pixel, seguindo um determinado critério. Por exemplo, pode-se citar a binarização ou limiarização da imagem, que consta da transformação desta em dois tons (branco e preto) e separa os objetos do fundo da imagem através da transformada da mesma para binário, sendo a etapa inicial antes da interpretação de contornos de imagem.

Na descontinuidade a participação da imagem é efetuada com base nas alterações bruscas de intensidade. Um exemplo de descontinuidade é a detecção de contornos. Uma borda é a fronteira entre duas regiões com propriedades distintas de níveis de cinza, ou seja, o contorno de um objeto que indica seus limites na imagem. A detecção de bordas é a técnica mais comum usada na detecção de descontinuidades em imagens tons de cinza. A detecção de bordas é um processo de localização e realce dos pixels de borda, aumentando o contraste entre a borda e fundo utilizando a variação de luminosidade (nível de cinza) entre os pixels.

As técnicas de segmentação nesta categoria buscam evidenciar os limites entre os objetos, através da detecção de pontos isolados e da detecção de linhas na imagem (RODRIGUES, 2002).

Esta etapa do processo será de extrema importância, pois se trata de limitar a área de trabalho na figura, ou seja, segmentar a imagem na região de interesse. No caso da imagem da chapa, a área de interesse será o ponto de solda, em especial, os pontos críticos, onde há necessidade de garantir uma solda de qualidade.

2.3.3.4 Representação e descrição

Geralmente, a saída do estágio de segmentação produz dados brutos de pixels. Neste caso pode ser necessário converter os dados para uma forma conveniente, possibilitando o processamento por computador.

A descrição, também chamada de seleção de características, lida com a extração de características que resultam em algumas informações quantitativas de interesse ou que são básicas para diferenciar uma classe de objetos de outra (GONZALES, WOODS, 2000).

Tem-se ainda a Região de Interesse (ROI), que é a região definida automaticamente a partir de parâmetros obtidos na própria imagem (ou por um usuário) onde o processamento estará totalmente concentrado. Pode-se, por exemplo, definir uma região de interesse onde se sabe por antecedência qual o local

de análise para extração de características. Na Figura 2.8, tem-se uma imagem original e uma imagem com região de interesse.

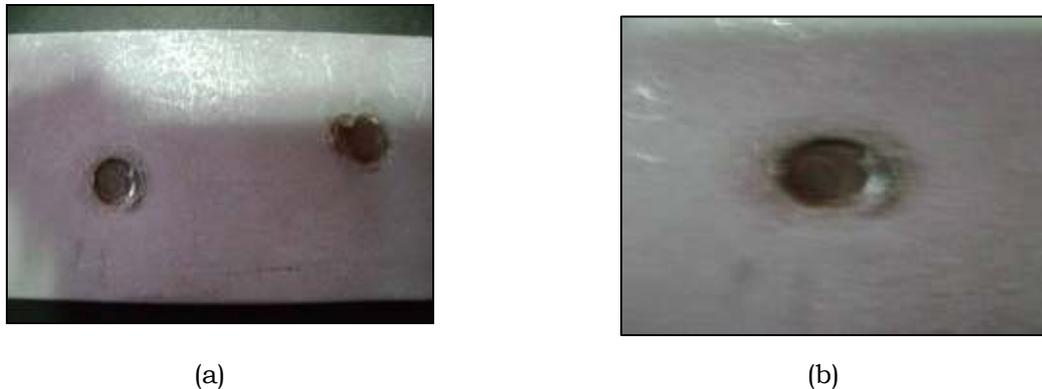


Figura 2.8 – (a) Imagem de corpo de prova com pontos de solda (b) Região de interesse

2.3.3.5 Reconhecimento e classificação

Consiste em atribuir um significado aos objetos da imagem. Reconhecimento é a parte do processamento que vai classificar os objetos a partir de informações encontradas na imagem, geralmente tendo como apoio uma base de conhecimentos previamente estabelecida como forma, textura, entre outras características. Nesta etapa, serão reconhecidos os diâmetros dos pontos e sua coloração, através de algoritmos montados no Matlab e Delphi para que em seguida, seja efetuada a interpretação e classificação do ponto de solda na chapa.

2.3.3.6 Base de conhecimento

Em um sistema de análise de imagens, o conhecimento sobre o domínio do problema pode ser codificado na forma de uma base de conhecimentos. Este conhecimento pode ser representado tanto na forma de regiões simples quanto em regiões detalhadas de uma imagem, onde a informação de interesse a ser localizada é conhecida. Desta forma, a busca pela informação pode ser reduzida.

A base de conhecimento também pode ser bastante complexa, tal como uma base de dados contendo imagens de alta resolução de uma determinada região, obtidas a partir de um satélite, em conexão com aplicação de detecção de mudanças. Além disso, para guiar a operação e cada módulo de processamento, a base de conhecimento também controla a interação entre eles.

2.4 Digitalização de Imagens

Segundo (Stringhini, 2006) é possível afirmar que toda e qualquer grandeza do mundo real pode ser codificada e representada sob forma de um conjunto de números, sendo isto chamado de digitalização. A aquisição de imagens consiste em obter imagens eletrônicas a partir de documentos em papel, microfilme, etc. Cada imagem eletrônica é um conjunto de pontos chamados pixels e que de uma forma discreta representam os componentes do documento original.

Neste processo pode-se definir que uma imagem nada mais é do que uma montagem feita a partir de pequenos quadrados coloridos, ou células, cada uma com uma cor homogênea. Para reproduzir uma imagem basta desenhar célula por célula, cada uma na sua posição, e atribuir a cada célula a cor correspondente a imagem original. Como se consegue exprimir as cores através de números, a imagem pode ser inteiramente codificada, célula a célula, e reproduzida usando estes códigos.

Para reproduzir as cores em números, deve-se utilizar uma paleta de cores como a que segue na Figura 2.9.

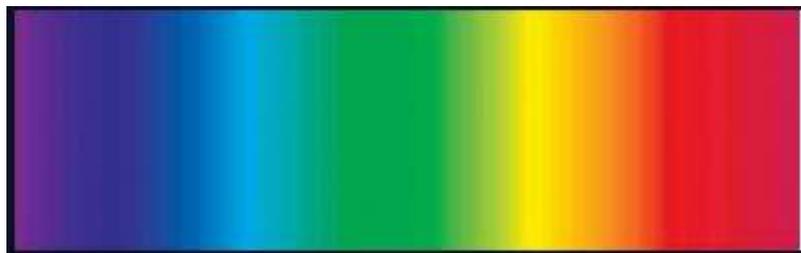


Figura 2.9 – Cores do espectro visível

Nesta paleta de cores, existem todas as cores percebidas pelo olho humano, em uma imagem suave cujas cores variam gradualmente. Cada faixa corresponde a uma cor homogênea, podendo dividir em 256 faixas verticais, numerando de 0 a 255, sendo que zero corresponde à faixa vertical preta e 255 corresponde à faixa vertical branca.

2.5 Tipos de Imagens

No sistema apresentado, serão utilizadas imagens do tipo RGB, escala de cinza (*grayscale*) e imagens binárias, que para um melhor resultado final do processo, passarão por filtros de suavização e limpeza da imagem.



Uma imagem do tipo RGB é formada pela junção de três componentes de cores, vermelha, verde e azul. A cor de cada pixel é determinada pela combinação das intensidades de cada uma destas cores. No Matlab, este tipo de imagem é armazenado como um arranjo de dados $m \times n \times 3$, onde m é o número de colunas, n é o número de linhas e 3 é a representação de cada escala de cor, ou seja, $m \times n \times 1$ é a representação na intensidade vermelha. Já $m \times n \times 2$ é na intensidade verde e $m \times n \times 3$ na intensidade azul.

Já no caso de uma imagem binária, cada pixel assume um de dois valores discretos, ou seja, zero ou um. Neste caso, a imagem será armazenada como uma combinação de zeros e uns, que representam a cor branca e preta respectivamente.

A cor que uma determinada superfície possui depende da luz que ilumina a mesma e depende também de como ela reage a essa luz. As luzes possuem diferentes tonalidades (temperaturas) de cor: a luz incandescente tende para o vermelho, a fluorescente comum para o verde. Neste caso, através de testes percebe-se que o uso da luz emitida pela lâmpada do tipo dicróica, proporciona um melhor resultado para aquisição e análise das imagens.

2.6 Detecção de Bordas

Segundo Gonzáles, a detecção de bordas é a abordagem mais comum da detecção de descontinuidades significantes nos níveis de cinza. Uma borda é um limite entre regiões com propriedades bastante distintas, neste caso, por se tratar de uma imagem binária, a borda é o limite entre as regiões que saltam de 0 para 1. A detecção de bordas é um processo de localização e realce dos pixels de borda, aumentando o contraste entre a borda e fundo utilizando a variação de luminosidade (nível de cinza) entre os pixels.

O resultado obtido consiste em *pixels* localizados apenas sobre a fronteira entre regiões, facilitando a determinação de medidas como raio, diâmetro, perímetro, etc. Estes pixels servem também para delimitar a região de interesse onde serão analisados os aspectos que determinam se a imagem está aprovada ou reprovada.

2.7 Limiarização

É um método de extração de objetos onde um limiar T é selecionado, determinando assim um limite de separação entre os dois grupos. Segundo



Gonzáles, cada ponto (x, y) tal que $f(x, y) > T$ é denominado um ponto do objeto, caso contrário, faz parte do fundo.

Para esta etapa, é escolhido um limiar T , o qual será aplicado ao histograma da imagem ou diretamente na imagem, separando os pixels em dois grupos distintos: objeto e fundo. Todos os pixels que estiverem valores de níveis de cinza menor que o limiar T se tornam 0 (preto), e todos os pixels que estiverem os valores de níveis de cinza maior que o limiar T , se tornam 1 (branco). Assim separa o objeto (0) e o fundo (1), ou vice versa. Esse processo gera uma imagem binária, e a escolha do limiar pode ser feita de forma manual, com o usuário escolhendo o melhor limiar, ou também pode ser feito de forma automática, com o sistema escolhendo o limiar automaticamente.

Neste caso, o método de limiarização foi utilizado para determinar, na transformação da imagem para binária, o ponto no qual os *pixels* deveriam ser acesos. Assim, após a imagem ser transformada em binária, a detecção da borda se torna mais fácil, pois a diferença entre os *pixels* 0 e 1 é facilmente percebida.

2.8 Tabela de Custos do Sistema

Conforme tabela 3, o custo será baixo se comparado aos sistemas existentes atualmente, o que torna bastante viável para as empresas que necessitam reduzir custos incorporarem esta solução a suas linhas de produção, podendo assim garantir a qualidade do ponto de uma forma rápida, automática e eficaz em 100% dos pontos antes que a o carro seja enviado para as concessionárias.

Tabela 3 – Tabela de custos

Item	Custo
Web cam 1.4 Mega pixel	R\$ 90,00
Licença Matlab Estudante com 2 toolbox	R\$ 473,04
Licença Delph Estudante com 2 toolbox	R\$ 680,00
Iluminação	R\$ 30,00
Suporte para câmera	R\$ 400,00



Chave on-off	R\$ 4,00
TOTAL	R\$ 1677,00

Para que o sistema se torne viável, deve-se desenvolver um código fonte em um aplicativo específico, onde o processamento da imagem se torne mais rápido.

3 MATERIAIS E MÉTODOS

Os principais temas abordados neste capítulo são: as características do processo de solda a resistência e o desenvolvimento do *software* para análise da solda.

3.1 Solda por Resistência

O processo de soldagem consiste em unir duas ou mais peças de metal através da aplicação de calor e pressão.

O processo de solda por resistência compreende um ramo da soldagem no qual o calor de solda necessário ao processo é gerado pela resistência oferecida pelas partes a serem fundidas à passagem da corrente elétrica. Este fato torna este processo diferente dos demais já que nenhum material externo, como fluidos ou arames é utilizado. Além disso, o processo de soldagem por resistência difere dos processos de soldagem por fusão ao utilizar força mecânica para forjar e unir as peças.

3.1.1 Princípios Gerais de Solda por Resistência

Segundo manual interno de treinamento GMB (Fundamentos da soldagem por resistência volume 1) uma solda ponto por resistência é efetuada ao se passar uma corrente durante um intervalo de tempo determinado pelas peças a partir dos eletrodos que aplicam, por sua vez, uma força de soldagem. A aplicação da corrente é contínua até que a fusão na superfície sobreposta das peças ocorra. A força do eletrodo é aplicada antes, durante e depois do intervalo do tempo.

A Figura 3.1 ilustra um esquema simplificado do equipamento mínimo necessário para se efetuar uma solda ponto por resistência. Cada um dos porta-eletrodos ou mesmo ambos estão anexados a um sistema de força que é capaz de

produzir a força necessária ao processo e movimentar os porta-eletrodos em dois sentidos, sendo um em direção à peça de trabalho e o outro afastando da mesma.

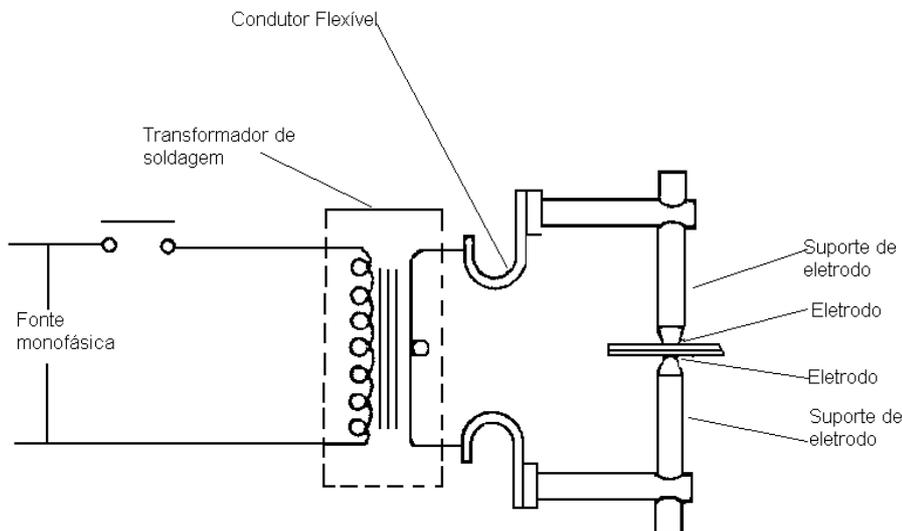


Figura 3.1 – Circuito secundário típico de uma solda ponto

3.1.1.1 Geração de Calor

Uma solda ponto é efetuada ao pressionar duas ou mais peças sobrepostas enquanto uma corrente elétrica passa através de uma área de contato localizada até aquecer o metal formando uma lente de solda até a temperatura de soldagem.

Um dos princípios da soldagem por resistência é gerar o calor na região da solda de forma rápida para que a quantidade mínima de calor seja dissipada através da condução para o material adjacente mais frio. Para que isto ocorra, é necessária uma alta taxa de geração de calor e isto é obtido através da passagem de alta corrente pela região de solda durante um curto intervalo de tempo. Outro princípio é gerar muito mais calor na região de solda do que em qualquer outra região do circuito secundário da máquina de solda.

A teoria e prática para se atingir estes dois princípios pode ser resumida da seguinte forma: calor é gerado sempre que uma corrente elétrica passa por uma resistência elétrica. A taxa na qual o calor é gerado é dada por:

$$P = I^2 \cdot R \quad \text{Equação 3.1}$$

onde:

- P = potência elétrica em watts

- I = corrente em ampères
- R = resistência em ohms

Se a corrente for aplicada em um intervalo de tempo **t** em segundos, a energia desenvolvida na resistência é:

$$E = I^2 \cdot R \cdot t \quad \text{Equação 3.2}$$

onde:

- E = energia elétrica

Percebe-se a partir da equação 3.2 que a amplitude da energia gerada pode variar de acordo com as variáveis de corrente, resistência e tempo de soldagem. A Figura 3.2 ilustra a geração de calor para efetuar a solda por resistência.

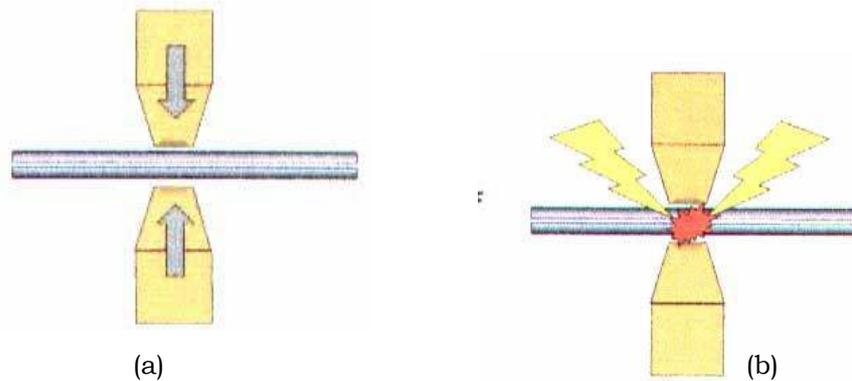


Figura 3.2 – (a) Imagem do início da solda (b) Imagem da geração de calor

3.1.2 Qualidade de Solda

Segundo manual interno de treinamento GMB (Fundamentos da soldagem por resistência volume 2) a qualidade de uma solda ponto é usualmente determinada pelo grau de controle dos fatores que fazem parte da formação da solda. Os principais fatores são: diâmetro, simetria e penetração da zona de fusão.

Para validação da qualidade de solda neste projeto será utilizado o diâmetro do ponto como fator determinante. Os pontos de solda serão aplicados em corpos de prova padrão, conforme ilustração da Figura 3.3.

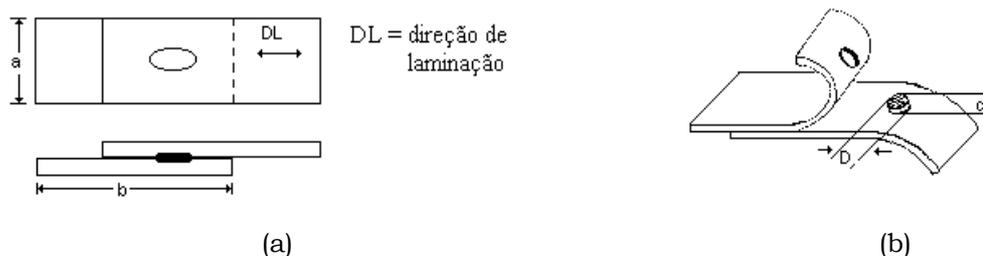


Figura 3.3 – (a) Imagem de corpos de prova padrão sobrepostos (b) Imagem do diâmetro do ponto

Na tabela 4 pode-se verificar os padrões de diâmetro do ponto de solda em relação à espessura da chapa.

Tabela 4 - Relação espessura da chapa e diâmetro do ponto

Espessura do material (mm)	Diâmetro mínimo do ponto de solda
Até 0,5	4 mm
0,5 a 1,0	5 mm
1,0 a 2,0	6 mm
2,0 a 3,10	6,5 mm

3.2 Descrição Geral do Sistema

Na Figura 3.4, está representado o sistema de testes de detecção e classificação de ponto de solda em chapas. Neste sistema, a posição da câmera é essencial para o enquadramento da imagem capturada. Para que o reconhecimento das bordas determine se o ponto está bom ou não, o mesmo deve estar concêntrico com a câmera. Após definido o ângulo, será determinado qual o espaço de leitura do software. Assim, com a captura da imagem, o software entenderá que a mesma é uma matriz de 640x480x3, ou seja, uma imagem do tipo RGB conforme explicado no capítulo 2.5.

A câmera está posicionada a uma distância de 35 mm da base, com regulagem de altura através do suporte lateral ou via software (zoom) que acompanha a câmera. A regulagem da altura e da posição do ponto deve ser feito manualmente ao iniciar o sistema de testes. Estas regulagens ficam salvas para todos os testes seguintes. As configurações de hardware, tipo brilho, contraste,

saturação, definição, equilíbrio de branco, são configuradas previamente pelo software da câmera, que está em modo automático.

A iluminação é feita através de uma lâmpada dicróica localizada lateralmente a câmera, esta lâmpada tem potência de 50 W e está direcionada para o suporte de alumínio pintado com tinta na cor branca que produzirá uma iluminação indireta e homogênea no ponto, iluminando o máximo possível sem deixar sombras que atrapalham o reconhecimento das bordas. Esta lâmpada está montada sobre um sistema basculante conforme mostrado na Figura 3.4, onde é possível regular a distância e o ângulo para que não haja sombras sobre o ponto de solda. Através de testes de detecção e classificação de pontos de solda em chapas encontrou-se a melhor posição para a câmera. O sistema de iluminação está ligado em 220 V e é acionado através de uma chave liga e desliga com possibilidade de regular a intensidade de brilho da lâmpada através de um dimer que está fixado em uma caixa de passagem junto com a chave. Este acionamento é feito manualmente conforme necessidade de testes.

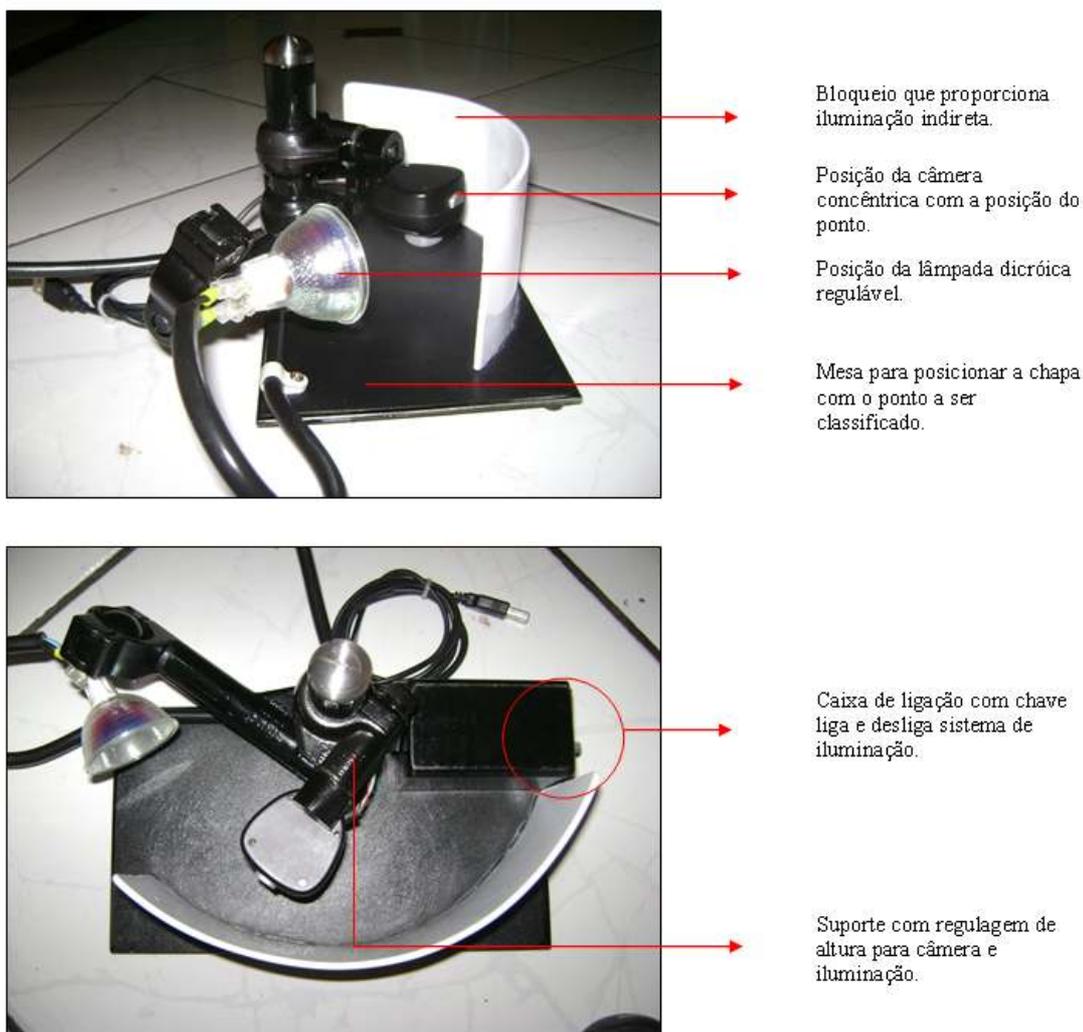


Figura 3.4 – Montagem do sistema de testes

Por meio destes testes, chegou-se a conclusão que o ponto a ser analisado deve estar a uma altura de 35 mm da câmera. Esta é a posição correta para se trabalhar com o ponto de solda em escala real. A iluminação foi testada com lâmpadas incandescentes e fluorescentes, porém o resultado não foi satisfatório. Este tipo de iluminação não elimina as sombras e interfere diretamente na detecção da imagem. A partir destes testes foi inserido o suporte de alumínio pintado na cor branca para proporcionar uma iluminação indireta e homogênea, depois de instalado o suporte e efetuado novos testes verificou-se a existência de sombras sobre a chapa dificultando a detecção das bordas. Devido à chapa refletir a iluminação a melhor solução foi utilizar uma lâmpada dicrômica que elimina as sombras e deixa a imagem homogênea refletindo no suporte branco.

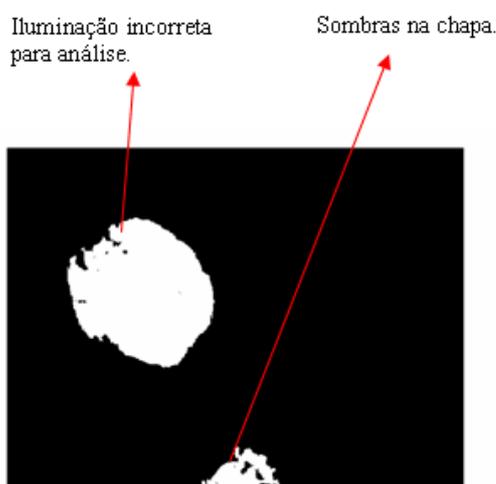


Figura 3.5 – Imagem de falha na iluminação do ponto

Utilizando este sistema, através de testes, foi possível obter imagens que não estavam aptas para a detecção e classificação dos pontos. Durante estes testes, foi-se modificando o tipo de iluminação, a posição e a superfície onde deveria estar o ponto de solda e chegou-se a conclusão que, a posição do mesmo deve ser concêntrica com a câmera. A iluminação não pode deixar sombras na superfície da chapa. Caso a chapa esteja com sombras serão reconhecidos todos os objetos escuros como sendo fronteiras para detecção das bordas.

Nas etapas de determinação do raio, a imagem tem que estar na forma binária, o que ocorre no processo de filtragem da imagem.

No diagrama de blocos, representado na Figura 3.6, estão representados as etapas do programa.

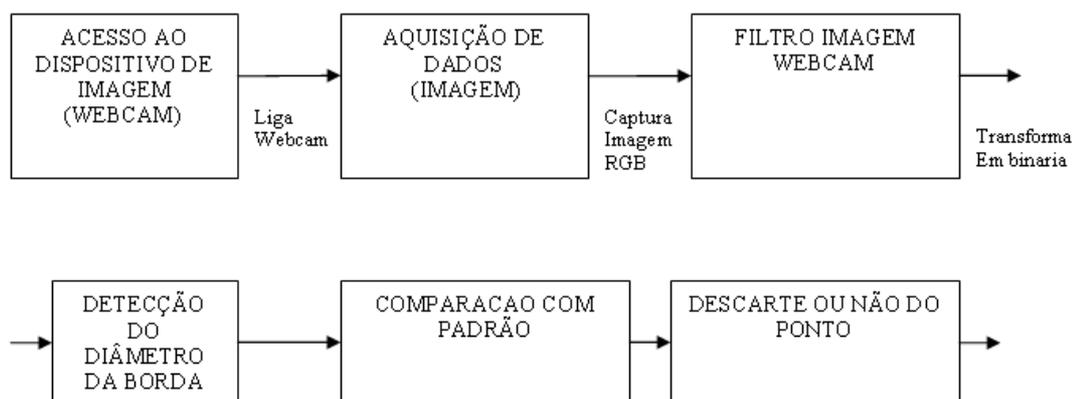


Figura 3.6 – Diagrama de blocos do código fonte do software

3.3 Descrição da Funcionalidade do Software

Para validação do método foram utilizados dois softwares distintos, *Matlab* e outro software específico desenvolvido em *Delphi*.

3.1.3 Fluxograma

No fluxograma da Figura 3.7, está representado o sistema geral de aquisição e processamento de imagem. A aquisição da imagem deve ser feita através de um comando do operador sempre que houver presença de uma chapa, em seqüência a imagem será processada e analisada para detectar a presença e classificação do ponto de solda.

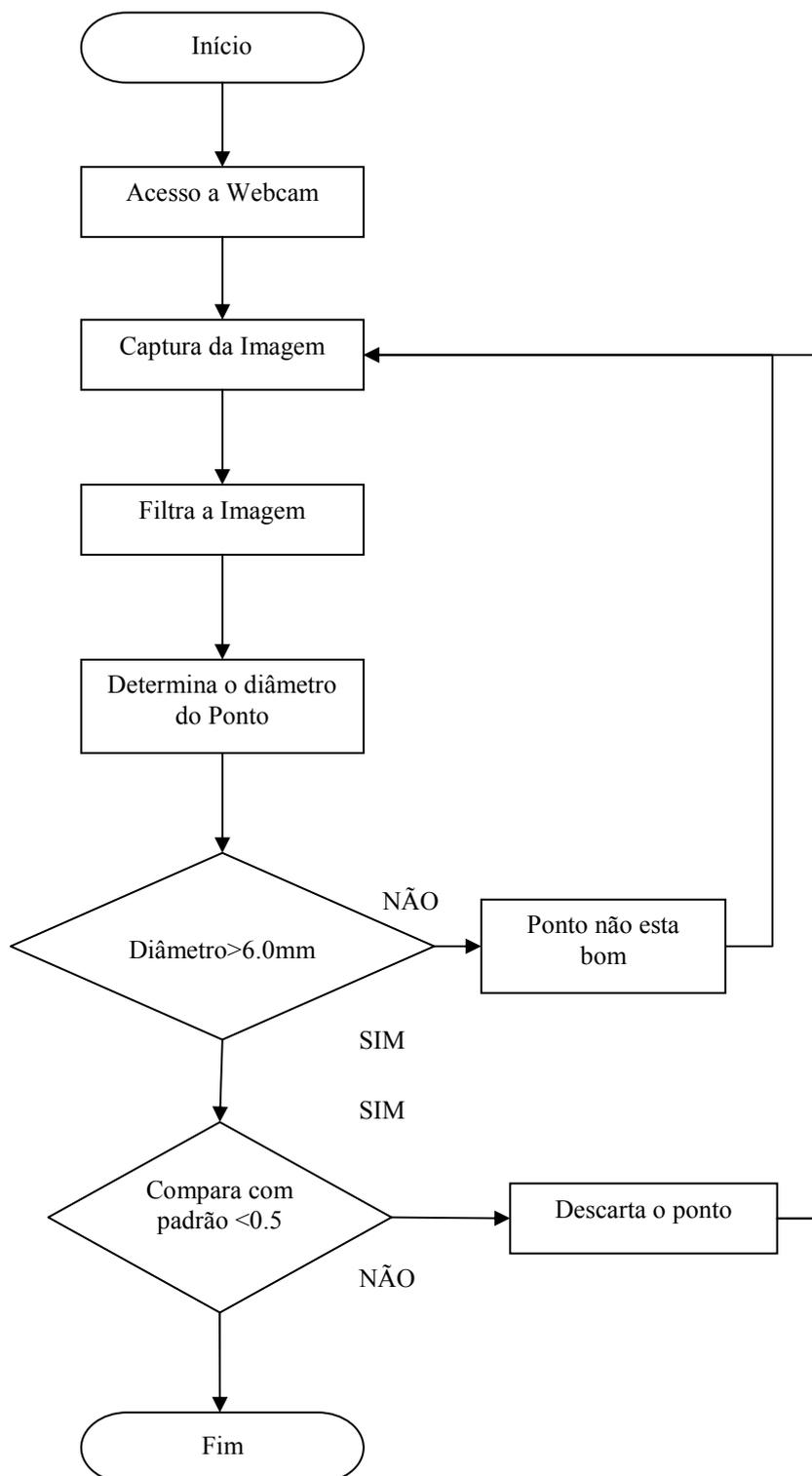


Figura 3.7 – Fluxograma geral de funcionamento

Após a aquisição e processamento da imagem, serão feitas as análises dos resultados obtidos. Assim, se o diâmetro do ponto de solda for menor que 6.0 mm, o mesmo será classificado como ruim. Caso o ponto esteja bom o mesmo passa por um processo de comparação com a imagem padrão.



3.1.4 Softwares implementados

O primeiro software do projeto foi desenvolvido na plataforma Matlab, com o auxílio do *Image Processing Toolbox*, que é uma coleção de funções que possibilitam a manipulação de imagens, dentre elas estão: aquisição de imagens, eliminação de ruídos, segmentação, entre outros. O segundo software foi desenvolvido em Delphi que assim como o Matlab possui componentes para manipulação de imagens.

3.3.4.1 Desenvolvimento Matlab

3.3.4.1.1 Sistema e Metodologia da Aquisição de Dados

A aquisição de dados é feita através do acesso a um dispositivo de vídeo pelo programa Matlab versão 7.2. No software desenvolvido, é possível alterar algumas características acessando a fonte deste dispositivo. Foi definido que o tamanho da imagem a ser trabalhada é de 640x480 *pixels*. Um vídeo é formado por um conjunto de imagens, que rodadas seqüencialmente com um curto espaço de tempo entre elas, produzem uma continuidade de formatos que dão a impressão de movimento. Cada uma destas imagens é conhecida como *frames*, conforme Figura 3.8. A sintaxe desta aquisição pode ser:

Acessar um dispositivo de aquisição de imagem:

```
wcam = videoinput('winvideo', 1, 'RGB24_640x480')
```

Tirando uma foto:

```
wfoto = getsnapshot(wcam);
```

Mostar a foto em uma janela nova:

```
imagesc(wfoto); →
```

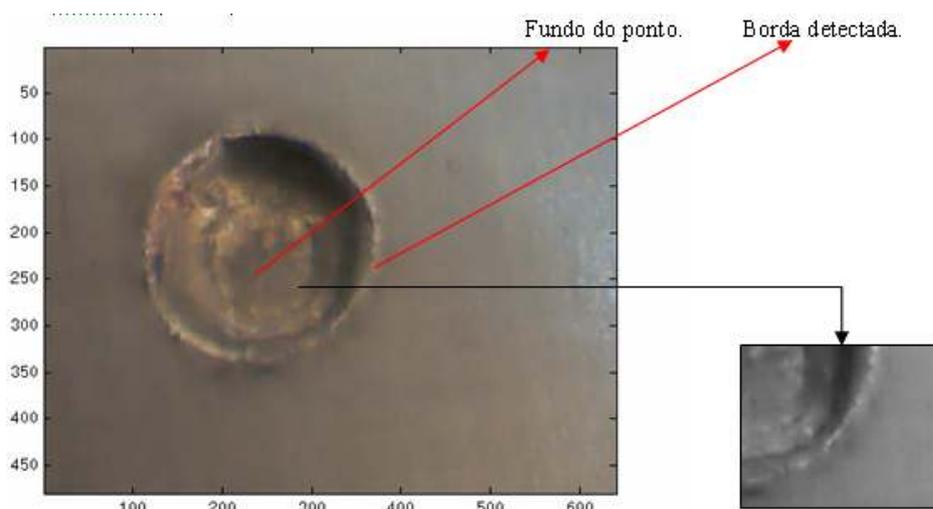


Figura 3.8 – Aquisição da Imagem Original

3.3.4.1.2 Tratamento da Imagem

Nesta etapa do processo, a imagem passa por alguns filtros que a tornarão ideal para verificação da borda. O primeiro filtro pelo qual a imagem passa é a transformação de RGB para tons de cinza, conforme mostra a Figura 3.9. Foi utilizado este filtro pelas suas características de exibir um alto valor para todos os níveis de cinza dentro de uma faixa de interesse e um baixo valor para todos os outros níveis de cinza, assim pode-se determinar as áreas de interesse. Nesta etapa, o software converte a imagem eliminando as informações de matiz e saturação enquanto retém a iluminação. A sintaxe desta transformação pode ser:

$$I = \text{rgb2gray}(wfoto);$$



Figura 3.9 – Imagem em tons de cinza

Na segunda etapa do filtro, ocorre a conversão para preto e branco. Para uma melhor transformação, utiliza-se o método limiar (*threshold*) que determina um nível de cinza, a partir do qual o ponto deve ser aceso, ou seja, arbitra um valor de intensidade normalizado, que fica entre o alcance de 0 e 1, que serve de limite para transformação, conforme Figura 3.10. Este filtro aplica a binarização como função e limitará as áreas de interesse separando o ponto de solda a ser analisado da chapa.

O filtro *threshold* de parâmetro t aplicado a uma imagem f é definido como

$$T(f,t)(x) = \begin{cases} 1, & \text{se } f(x) \geq t \\ 0 & \text{c.c} \end{cases}$$

Este método tem por finalidade minimizar a discrepância entre *pixels* preto e branco. Pode-se utilizar a seguinte sintaxe para este filtro:

Threshold=graythresh(I);

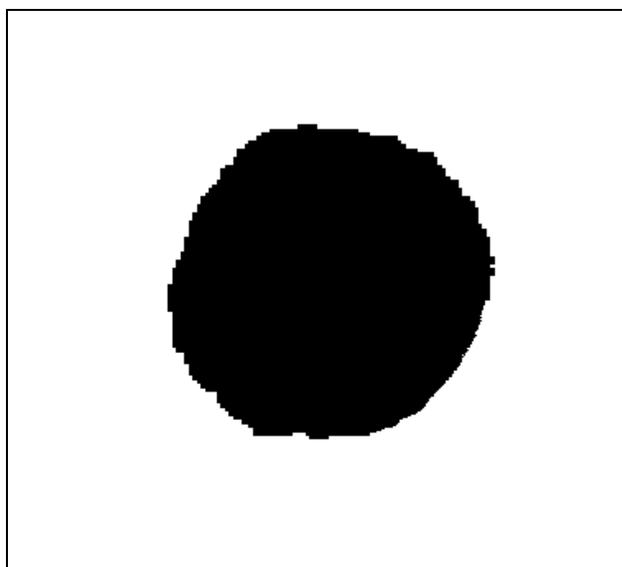


Figura 3.10 – Imagem com filtro Threshold

A próxima etapa de filtragem, inverte a forma binária da imagem para melhor detecção dos pontos onde a imagem muda seu valor de 1 para 0, onde $f(x)$ é definido como

$$f(x) = \begin{cases} 1, & \text{se } f(x) = 0 \\ 0, & \text{se } f(x) = 1 \end{cases}$$

e conforme figura 23. Sua sintaxe pode ser:

```
BW1 = ~im2bw(I,threshold);
```

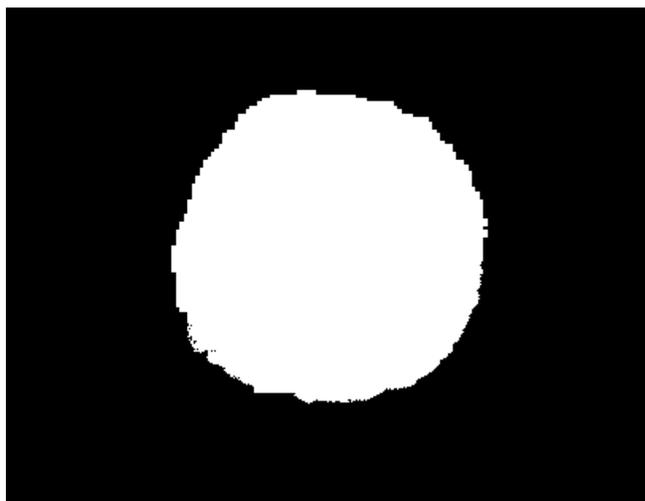


Figura 3.11 – Imagem transformada binário

No filtro seguinte, a imagem passa por um processo de suavização, onde é feito uma limpeza na borda do ponto. Este processo consiste em remover todos os pequenos objetos que estão conectados nas fronteiras de uma vizinhança binária, suprimindo as estruturas mais claras que estão conectadas a borda do ponto, delimitado pelo número de *pixels* que estão em contato, conforme mostra Figura 3.13. Para esta etapa do processo será utilizada dilatação de imagens como um operador básico da morfologia digital. A dilatação é uma transformação morfológica que utiliza dois conjuntos usando adição vetorial, onde uma pequena área relacionada a um pixel é alterada para um dado padrão. Seu símbolo é \oplus . Como o nome diz, o resultado será uma imagem “engordada”.

A dilatação de um conjunto A pelo conjunto B e definida por:

$$A \oplus B = \{ c \mid c = a + b, a \in A, b \in B \}$$

Onde A representa a imagem sendo operada e B é um segundo conjunto onde é chamado *elemento estrutural* e sua composição define a natureza específica da dilatação, sendo assim a dilatação expande uma imagem. A Figura 3.12 demonstra a seqüência de passos na dilatação de A por B de uma imagem.

$$A + \{0,0\} = \{1,1\},\{1,2\},\{2,1\},\{2,2\} =$$

$$\{1,1\} + \{0,0\} = \{1,1\}$$

$$\{1,2\} + \{0,0\} = \{1,2\}$$

$$\{2,1\} + \{0,0\} = \{2,1\}$$

$$\{2,2\} + \{0,0\} = \{2,2\}$$

$$A + \{1,0\} = \{2,1\},\{2,2\},\{3,1\},\{3,2\} =$$

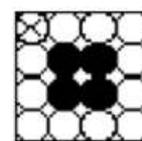
$$\{1,1\} + \{1,0\} = \{2,1\}$$

$$\{1,2\} + \{1,0\} = \{2,2\}$$

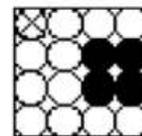
$$\{2,1\} + \{1,0\} = \{3,1\}$$

$$\{2,2\} + \{1,0\} = \{3,2\}$$

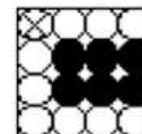
$$A \oplus B = \{1,1\},\{1,2\},\{2,1\},\{2,2\},\{3,1\},\{3,2\} =$$



a)



b)



c)

Figura 3.12 – (a) Imagem de uma imagem A (b) Imagem de um elemento estrutural
(c) Imagem da dilatação de A por B

Para executar este processo foi utilizado um comando do Matlab com sintaxe:

```
BW = bwareaopen(BW1,2000);
```

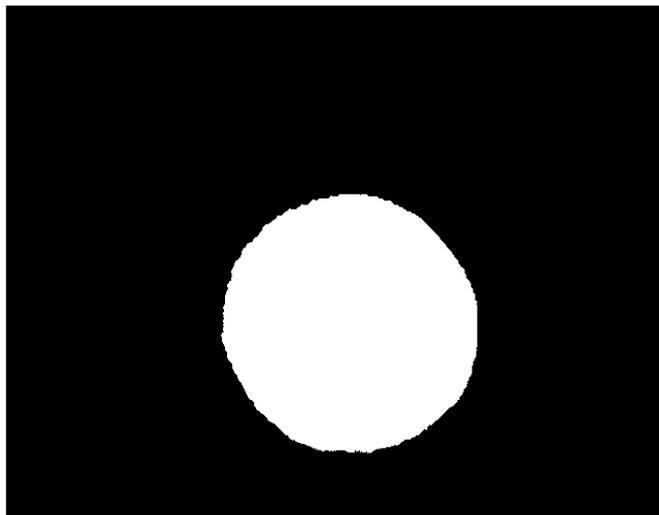


Figura 3.13 – Imagem suavizada com estruturas suprimidas

Através de testes executados chegou-se a conclusão que todos os objetos na imagem BW1 menores que 2000 pixels devem ser removidos da mesma proporcionando uma imagem limpa e preparada para o reconhecimento da borda.



3.3.4.1.3 Reconhecimento da Borda

Nesta etapa o objetivo é reconhecer a borda do ponto e extrair o raio do mesmo, para verificar se o ponto está pronto para seguir o processo. Primeiramente especifica-se um ponto no limite da borda. Este ponto será utilizado para referenciar o começo da localização da borda. A sintaxe que exerce esta função é a seguinte:

```
dim = size(BW);  
  
col = round(dim(2)/2)-90;  
  
row = min(find(BW(:,col)));
```

O próximo passo é achar a localização dos pontos de limites. Para maximizar a precisão do cálculo do raio, é importante achar o maior número de pontos possíveis. Neste caso, através de testes, foram encontrados 500 pontos que percorrem a borda do ponto. A conectividade foi determinada com valor oito para o programa encontrar as vizinhanças-de-8 e obter o traçado mais próximo da circunferência, pois assim, a partir do primeiro ponto encontrado o programa verifica as direções na diagonal, vertical e horizontal dos pixels vizinhos conforme Figura 3.14. A rotina para determinar esta etapa é:

```
conectividade = 8;  
  
num_pontos = 500;  
  
contador=bwtraceboundary(BW[row,col], 'N', conectividade, num_pontos);
```



Figura 3.14 – Pontos da borda detectados

Na figura 3.14 a linha traçada na cor verde é referente aos pontos detectados na borda da figura do ponto de solda.

Em seguida, é ajustada uma circunferência sobre os pontos encontrados. Nesta parte do processo, também são calculados o centro do ponto e a posição do raio. A sintaxe desta etapa é:

Determinando a Circunferência de Contorno:

```
x = contador(:,2);  
y = contador(:,1);  
abc=[x y ones(length(x),1)]/[-(x.^2+y.^2)];  
a = abc(1); b = abc(2); c = abc(3);
```

Cálculo da localização do centro e do raio:

```
xc = -a/2;  
yc = -b/2;  
raio = sqrt((xc^2+yc^2)-c)
```

Plotar a circunferência inteira

```
theta = 0:0.01:2*pi;
```

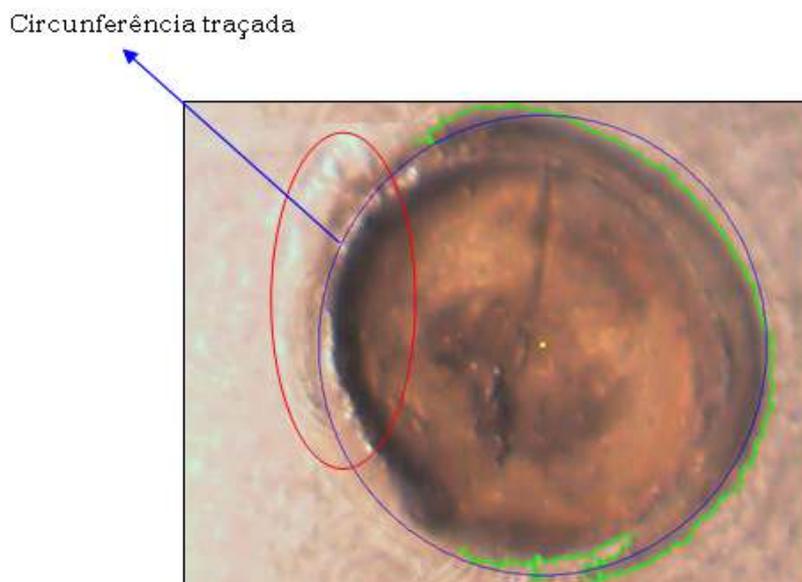


Figura 3.15 – Circunferência traçada

Na figura 3.15 a linha traçada na cor azul é referente a aproximação da circunferência a partir do raio da figura do ponto de solda.

Usar representação paramétrica da circunferência para obter coordenada de pontos na imagem:

$$X_{fit} = raio * \cos(\theta) + x_c;$$

$$Y_{fit} = raio * \sin(\theta) + y_c;$$

Cálculo do diâmetro em cm:

$$diâmetro = raio * 0.0515;$$

Depois de calculado o diâmetro do ponto a imagem é exibida no seu formato original com a mensagem do diâmetro calculado e se o mesmo está bom ou ruim.



Figura 3.16 – Imagem original com resultado do processamento

3.3.4.1.4 Análise dos Dados

A análise dos dados é feita através de um algoritmo que compara os valores encontrados com padrões de qualidade dos pontos. Estes valores são:

Diâmetro da Borda: Maior que 6.0mm

Comparação com Padrão: Maior que 80%

Nesta etapa, é mostrada a imagem original e será escrito sobre esta imagem se o ponto está aprovado ou reprovado, o diâmetro do ponto e o resultado comparado com uma imagem padrão. A sintaxe está representada no apêndice A. O fluxograma da Figura 3.17 representa o processo para análise dos dados.

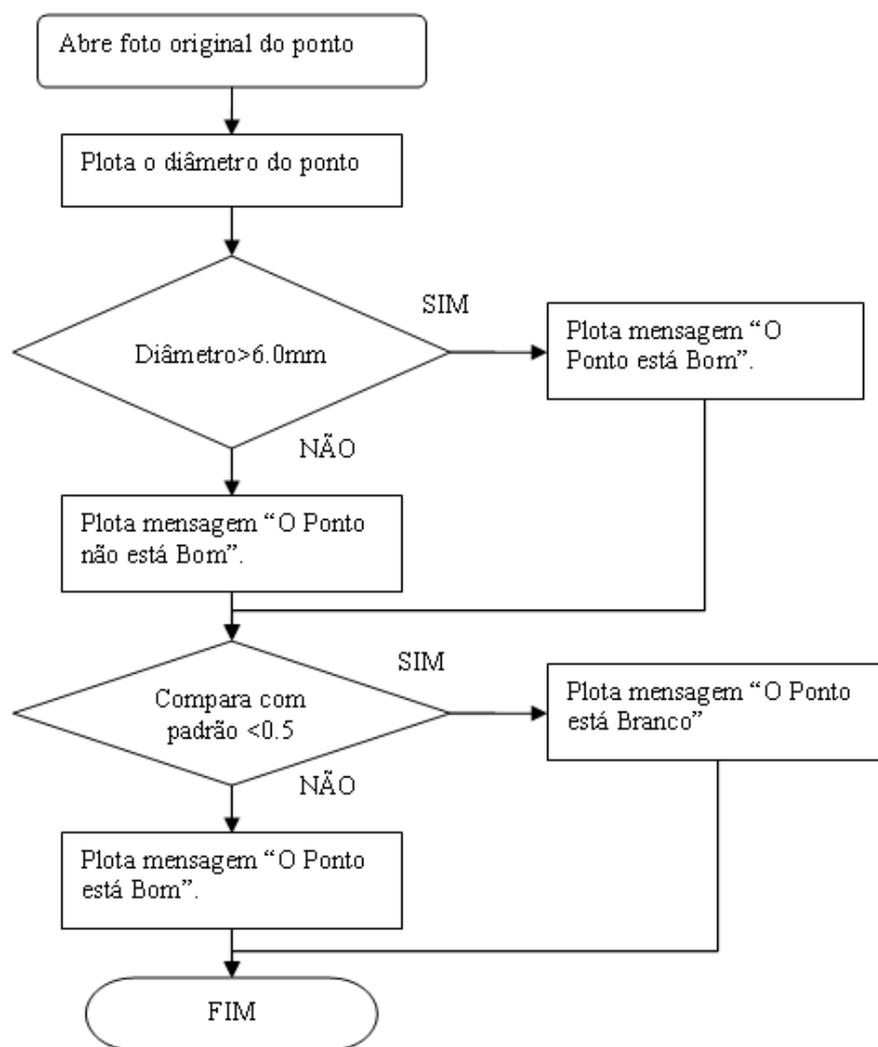


Figura 3.17 – Fluxograma de análise dos dados

3.3.4.2 Desenvolvimento Delphi 1

3.3.4.2.1 Sistema e Metodologia da Aquisição de Dados

A aquisição de dados é feita através do acesso a um dispositivo de vídeo pelo programa Delphi. No software desenvolvido, é possível alterar algumas características acessando a fonte deste dispositivo. Foi definido que o tamanho da imagem a ser trabalhada é de 640x480 *pixels*, conforme Figura 3.18. A sintaxe desta aquisição pode ser:

Video.ShowVideo := True;



Figura 3.18 – Aquisição da imagem original Delphi 1

3.3.4.2.2 Tratamento da Imagem

O primeiro filtro pelo qual a imagem passa após ser capturada é a transformação de RGB para tons de cinza, conforme Figura 3.19. Nesta etapa, o software converte a imagem eliminando as informações de matiz e saturação enquanto retém a iluminação. A sintaxe desta transformação pode ser:

ImageEffects.GrayScale(Image.Bitmap);



Figura 3.19 – Imagem em tons de cinza

Na segunda etapa do filtro, é feito um ajuste do contraste da imagem para melhorar a diferença entre os níveis de cinza médio do objeto. Para uma melhor transformação, a manipulação do contraste é realizada em cada pixel aplicando uma transformação T em cada pixel com o objetivo de melhorar a discriminação visual entre os objetos presentes na imagem. Esta operação é realizada ponto a ponto e reproduzem diferentes tons do objeto de imagem com grandes diferenças de densidade. Para esta função utiliza-se o realce linear que aumenta o contraste de uma imagem expandindo o intervalo original de níveis de cinza do objeto, onde $fg(x)$ é definido como

$$Fg(x) = G * f(x) + D$$

- $fg(x)$ é o novo valor do nível de cinza.
- $f(x)$ é o valor original do nível de cinza.
- D é um fator de incremento a todos os níveis de cinza da imagem – associado diretamente com o brilho da imagem.
- G é o ganho desejado – associado diretamente com o contraste da imagem.

Para definição do ganho foi inserido no programa um ajuste manual de nível para determinar o melhor contraste da imagem, conforme Figura 3.20

ImageEffects.Threshold(Image.Bitmap, EdtLevel.Position);



Figura 3.20 – Imagem com ajuste de contraste

Na última etapa do filtro, é a conversão para binário. Para uma melhor transformação, utiliza-se o método limiar (*threshold*), conforme Figura 3.21. Pode-se utilizar a seguinte sintaxe para este filtro:

```
ImageEffects.Threshold(Image.Bitmap, EdtLevel.Position);
```

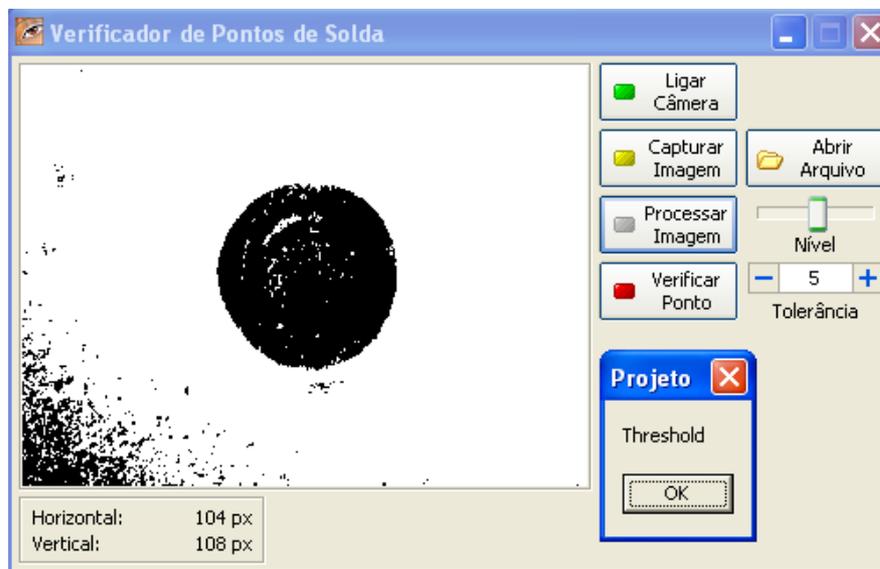


Figura 3.21 – Imagem com filtro Threshold Delphi 1

3.3.4.2.3 Reconhecimento da Borda

Para o reconhecimento da borda foi desenvolvido um algoritmo no software que através de uma varredura dos pixels nos eixos (x,y) determina as fronteiras do ponto de solda. Para minimizar o processamento do programa a análise do ponto de solda foi dividida em quatro quadrantes a partir do centro da imagem. Considerando que a área de verificação do ponto está em uma janela de 640 x 480 e o ponto está posicionado no centro da mesma, o software executa quatro varreduras partindo do ponto central e analisando os pixels, ou seja, no eixo x verifica todos os pixels da posição 320 até 1 e depois 320 até 640, do mesmo modo no eixo y porém da posição 240 até 1 e depois 240 até 480. A cada quadrante analisado, enquanto o pixel for zero continua o teste, caso encontre algum pixel com valor um, verifica se realmente é a fronteira ou apenas uma impureza ou sombra da imagem. Esta verificação ocorre através da tolerância pré-estabelecida no programa, ou seja, se esta estiver com seu valor fixo em cinco a varredura ocorre enquanto o programa não encontrar cinco pixels com valor um. Quando encontrar os cinco pixels com valor um o programa subtrai cinco retornando para posição referente ao resultado e traça a fronteira. Se o pixel analisado for a borda passa



para o próximo quadrante. Após encontrar todas as fronteiras, são traçadas quatro linhas formando um quadrado com os diâmetros correspondentes ao ponto conforme mostra a Figura 3.22. A sintaxe para esta função é:

```
for X1 := 320 downto 1 do
begin
if Image.Bitmap.Canvas.Pixels[X1, 240] = clWhite then I := I + 1 else I := 0;
if I >= EdtLimite.Value then Break;
end;
for X2 := 320 to 640 do
begin
if Image.Bitmap.Canvas.Pixels[X2, 240] = clWhite then I := I + 1 else I := 0;
if I >= EdtLimite.Value then Break;
end;

for Y1 := 240 downto 1 do
begin
if Image.Bitmap.Canvas.Pixels[320, Y1] = clWhite then I := I + 1 else I := 0;
if I >= EdtLimite.Value then Break;
end;

for Y2 := 240 to 480 do
begin
if Image.Bitmap.Canvas.Pixels[320, Y2] = clWhite then I := I + 1 else I := 0;
if I >= EdtLimite.Value then Break;
end;
```

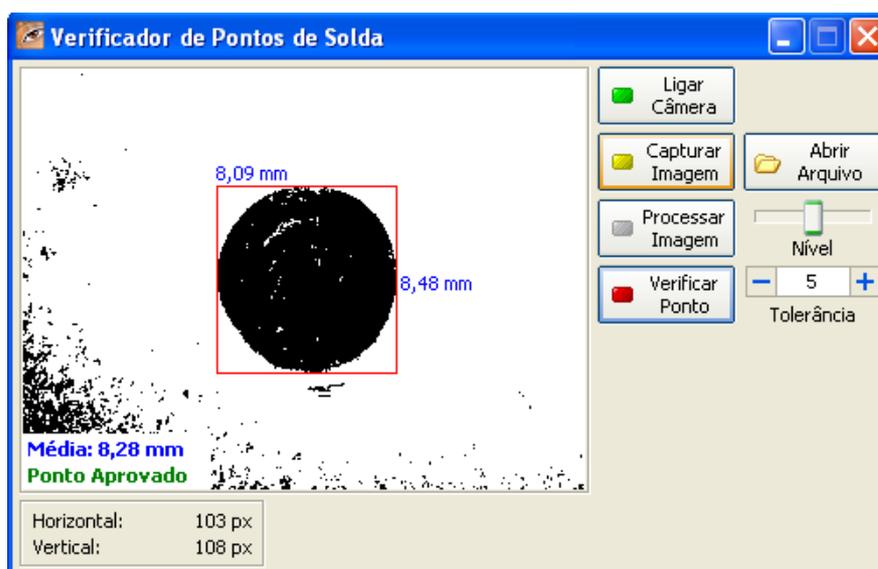


Figura 3.22 – Resultado do processamento

3.3.4.2.4 Análise dos Dados

A análise dos resultados segue os mesmos padrões determinados no software do Matlab apresentado na secção 3.3.2.1.5.



4 APRESENTAÇÃO DOS RESULTADOS OBTIDOS

Para análise de desempenho dos métodos propostos foram utilizadas 40 amostras disponibilizadas pela empresa General Motors. Sendo 20 contendo pontos bons e 20 pontos ruins. Constatou-se que no software Matlab para determinação do ponto bom, houve o reconhecimento de 95% dos pontos que estavam dentro das condições exigidas pelo trabalho. No reconhecimento de ponto solto, a detecção não permitiu a medição correta de todos os pontos, pois alguns pontos não proporcionavam a determinação da borda, o que gera um erro na determinação do diâmetro do ponto analisado, para este teste a eficiência do projeto limitou-se a 50% das chapas analisadas. Já no software Delphi para a determinação de ponto bom, obteve-se o reconhecimento de 100% dos pontos que estavam dentro das condições exigidas. No reconhecimento de ponto solto, alguns pontos não proporcionavam a determinação da borda, o que gera um erro na determinação do diâmetro do ponto analisado, para este teste a eficiência foi de 70% das chapas. Têm-se na tabela de classificação das peças boas as medições automáticas e manuais para as especificações, em seqüência as imagens correspondentes.

Tabela 5 - Medidas de ponto bom

Peça	Medidas do Programa Matlab	Medidas do Programa Delphi	Medidas manual (paquímetro)	Diferença entre medidas Matlab	Diferença entre medidas Delphi
1	7.02 mm	6.95 mm	7.0 mm	0.02 mm	0.05 mm
2	6.40 mm	6.79 mm	6.6 mm	0.2 mm	0.19 mm
3	6.73 mm	6.63 mm	6.6 mm	0.13 mm	0.03 mm
4	7.83 mm	7.89 mm	7.8 mm	0.03 mm	0.09 mm
5	7.69 mm	7.77 mm	7.8 mm	0.11 mm	0.03 mm
6	6.25 mm	6.48 mm	6.4 mm	0.15 mm	0.08 mm
7	8.05 mm	8.09 mm	8.0 mm	0.05 mm	0.09 mm
8	6.60 mm	6.24 mm	6.4 mm	0.20 mm	0.16 mm
9	6.92 mm	7.07 mm	7.0 mm	0.08 mm	0.07 mm
10	7.01 mm	7.26 mm	7.2 mm	0.19 mm	0.06 mm
11	7.26 mm	7.42 mm	7.4 mm	0.14 mm	0.02 mm
12	6.40 mm	6.16 mm	6.2 mm	0.20 mm	0.04 mm
13	7.88 mm	8.09 mm	8.0 mm	0.12 mm	0.09 mm
14	7.35 mm	7.26 mm	7.3 mm	0.05 mm	0.04 mm
15	6.44 mm	6.91 mm	6.9 mm	0.46 mm	0.01 mm
16	6.12 mm	6.12 mm	6.2 mm	0.08 mm	0.08 mm
17	6.76 mm	7.38 mm	7.0 mm	0.24 mm	0.38 mm
18	6.38 mm	6.32 mm	6.5 mm	0.12 mm	0.18 mm
19	7.61 mm	7.58 mm	7.5 mm	0.11 mm	0.08 mm
20	10.55 mm	7.54 mm	7.5 mm	Medida descartada	0.04 mm
	Acertos 95%	Acertos 100%		Media 0.14 mm	Media 0.09 mm

A seguir o gráfico com comparação de medidas de pontos bons dos softwares e paquímetro. O mesmo mostra a medida manual com paquímetro como referência de medição e as comparações com os softwares. No eixo das coordenadas estão as peças em seqüência de medidas e no eixo das abscissas os valores correspondentes ao diâmetro do ponto.

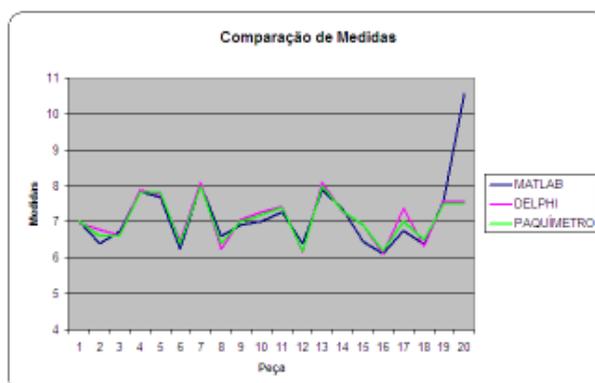


Gráfico 1 – Comparação de medidas ponto bom

As 12 imagens a seguir ilustram os resultados das 6 primeiras peças de pontos bons analisadas pelos softwares distintos.

Resultado Matlab

Resultado Delphi



Figura 4.1 – Amostra 1 boa

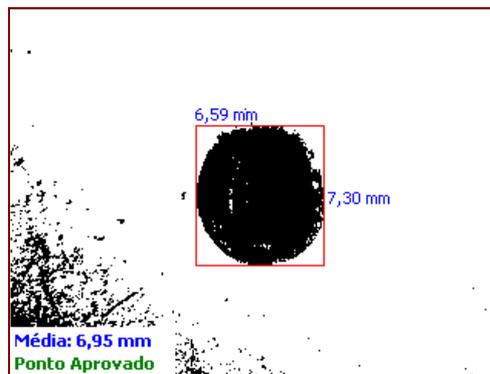


Figura 4.2 – Amostra 1 boa



Figura 4.3 – Amostra 2 boa

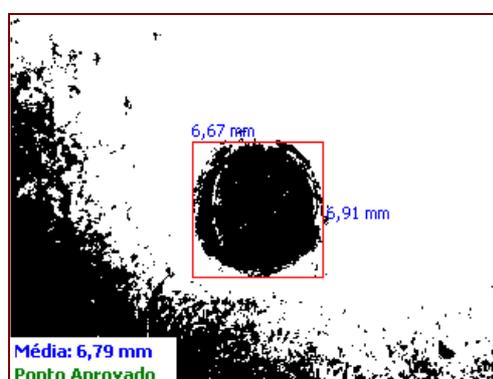


Figura 4.4 – Amostra 2 boa



Figura 4.5 – Amostra 3 boa

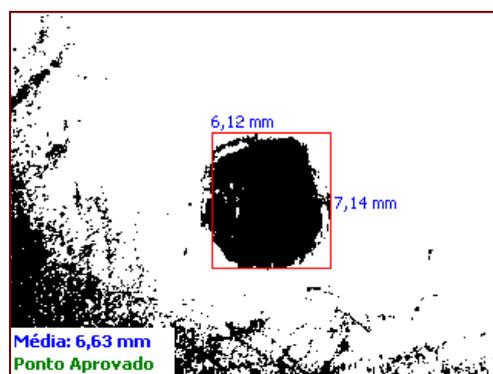


Figura 4.6 – Amostra 3 boa



Figura 4.7 – Amostra 4 boa

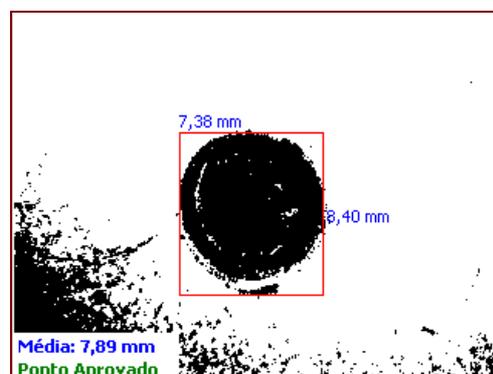


Figura 4.8 – Amostra 4 boa

Resultado Matlab



Figura 4.9 – Amostra 5 boa



Figura 4.11 – Amostra 6 boa

Resultado Delphi

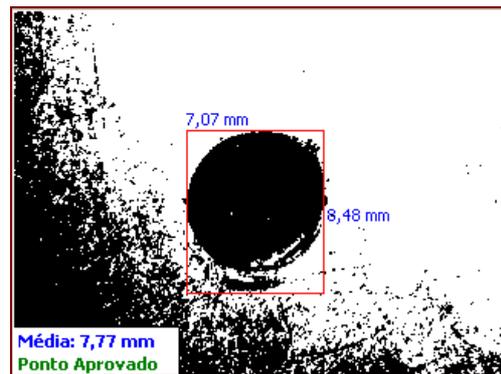


Figura 4.10 – Amostra 5 boa

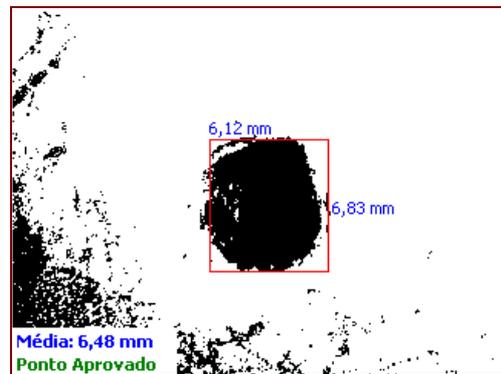
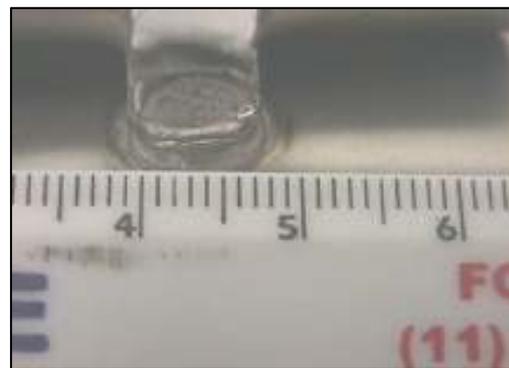


Figura 4.12 – Amostra 6 boa

Para comprovação dos resultados foi realizado testes de arrancamento de chapas em todos os corpos de prova. Através deste teste foi comprovado que todos os pontos aprovados pelo programa estavam bons. A Figura 4.13 a e b apresenta os resultados do teste de arrancamento das amostras 4 e 5.



(a)



(b)

Figura 4.13 – (a) Arrancamento aprovado amostra 4 (b) Arrancamento aprovado amostra 5

Têm-se na tabela de classificação das peças ruins as medições automáticas e manuais para as especificações, em seqüência as imagens correspondentes.

Tabela 6 - Medidas de ponto ruim

Peça	Medidas do Programa Matlab	Medidas do Programa Delphi	Medidas manual (paquímetro)	Diferença entre medidas Matlab	Diferença entre medidas Delphi
21	4.19 mm	4.44 mm	4.0 mm	0.19 mm	0.44 mm
22	5.03 mm	5.10 mm	5.0 mm	0.03 mm	0.10 mm
23	5.34 mm	5.34 mm	5.5 mm	0.16 mm	0.16 mm
34	5.54 mm	5.69 mm	5.5 mm	0.04 mm	0.19 mm
25	3.32 mm	4.59 mm	4.9 mm	Medida descartada	0.31 mm
26	3.03 mm	5.10 mm	5.0 mm	Medida descartada	0.10 mm
27	4.68 mm	4.5 mm	4.8 mm	0.12 mm	0.30 mm
28	4.69 mm	5.1 mm	4.8 mm	0.11 mm	0.30 mm
29	5.46 mm	5.21 mm	5.5 mm	0.04 mm	0.29 mm
30	4.97 mm	5.13 mm	4.9 mm	0.07 mm	0.23 mm
31	3.53 mm	2.45 mm	5.0 mm	Medida descartada	Medida descartada
32	4.70 mm	5.00 mm	4.8 mm	0.1 mm	0.20 mm
33	4.46 mm	4.8 mm	4.5 mm	0.04 mm	0.30 mm
34	4.65 mm	3.12 mm	5.0 mm	Medida descartada	Medida descartada
35	3.93 mm	2.33 mm	5.0 mm	Medida descartada	Medida descartada
36	3.72 mm	5.32 mm	5.0 mm	Medida descartada	0.32 mm
37	3.65 mm	4.32 mm	4.4 mm	Medida descartada	0.08 mm
38	3.36 mm	3.33 mm	5.0 mm	Medida descartada	Medida descartada
39	7.41 mm	3.24 mm	4.3 mm	Medida descartada	Medida descartada
40	7.59 mm	7.23 mm	5.5 mm	Medida descartada	Medida descartada
	Acertos 50%	Acertos 70%		Media 0.09 mm	Media 0.23 mm

A seguir o gráfico com comparação de medidas de pontos ruins dos softwares e paquímetro.

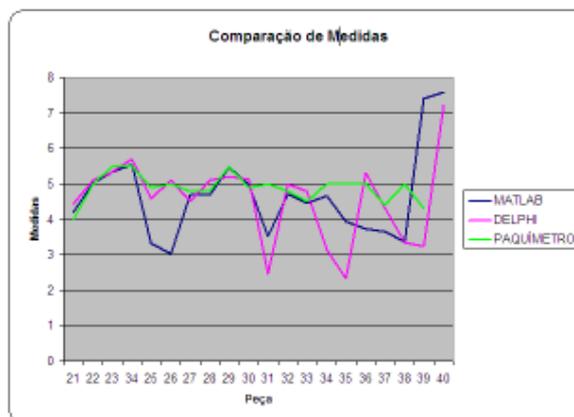


Gráfico 2 – Comparação de medidas ponto ruim

O Gráfico 2 mostra a medida manual com paquímetro como referência de medição e as comparações com os softwares. No eixo das coordenadas estão as peças em seqüência de medidas e no eixo das abscissas os valores correspondentes ao diâmetro do ponto.

Para a detecção de ponto solto algumas medidas não foram validadas. Na tabela 6 as medidas em cor vermelha foram descartadas, pois as mesmas foram detectadas fora da área de interesse do processamento.

As 12 imagens a seguir ilustram os resultados das 6 primeiras peças analisadas de pontos ruins pelos softwares distintos.

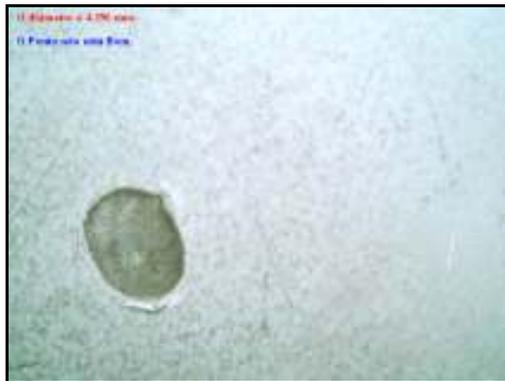
Resultado Matlab

Figura 4.14 – Amostra 21 ruim



Figura 4.16 – Amostra 22 ruim



Figura 4.18 – Amostra 23 ruim



Figura 4.20 – Amostra 24 ruim

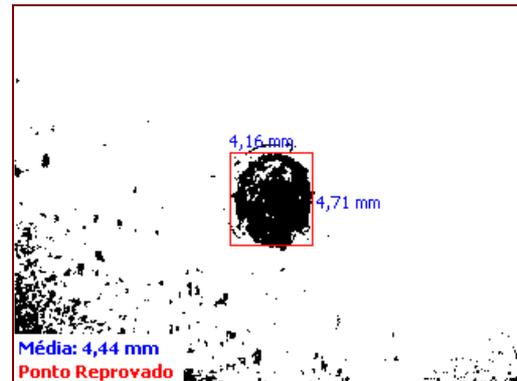
Resultado Delphi

Figura 4.15 – Amostra 21 ruim

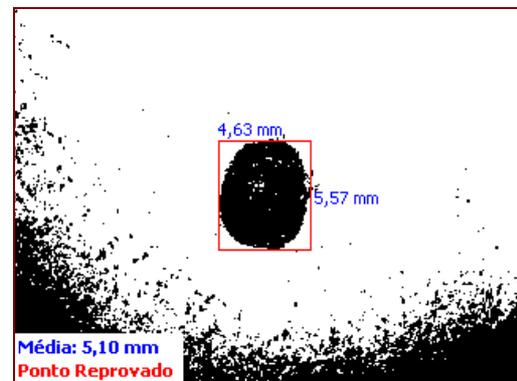


Figura 4.17 – Amostra 22 ruim

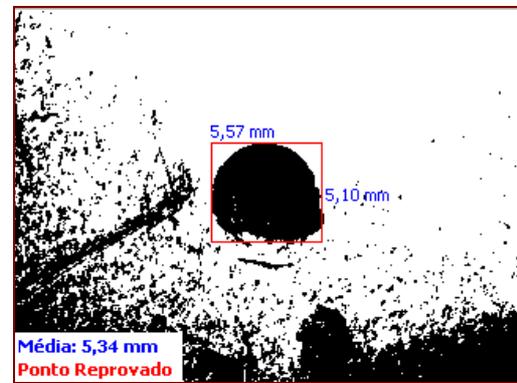


Figura 4.19 – Amostra 23 ruim

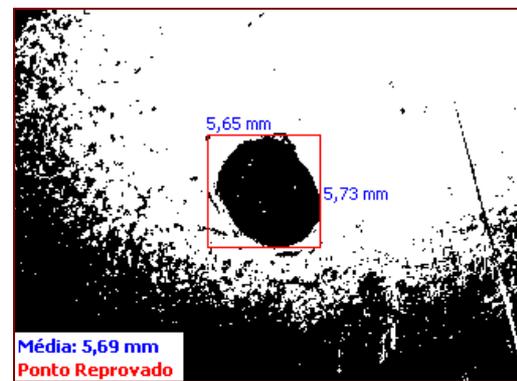


Figura 4.21 – Amostra 24 ruim

Resultado Matlab



Figura 4.22 – Amostra 25 ruim



Figura 4.24 – Amostra 26 ruim

Resultado Delphi

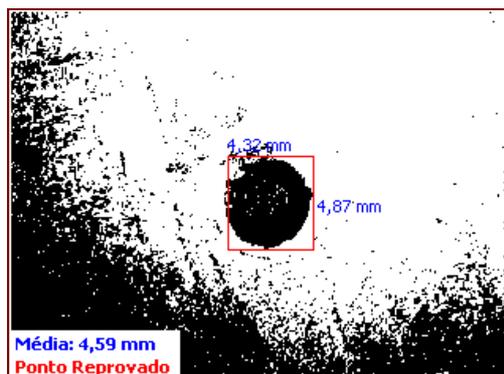


Figura 4.23 – Amostra 25 ruim

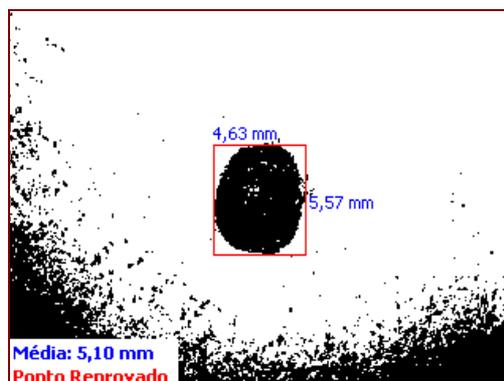


Figura 4.25 – Amostra 26 ruim

Para comprovação dos resultados foram realizados testes de arrancamento de chapas em todos os corpos de prova. Através deste teste foi comprovado que todos os pontos reprovados pelo programa estavam ruins. A Figura 4.26 a e b apresenta os resultados do teste de arrancamento das amostras 21 e 25.



(a)



(b)

Figura 4.26 – (a) Arrancamento reprovado amostra 21 (b) Arrancamento reprovado amostra 25

5 CONCLUSÃO

Neste trabalho foram desenvolvidos dois programas para análise de ponto de solda em softwares distintos, Matlab e Delphi, a fim de classificar as partes submetidas em dois parâmetros qualitativos: solda boa e solda ruim.

As soldas foram executadas em corpos de prova padrão sob diferente parametrização da máquina de solda, de modo a gerar pontos bons e ruins nas amostras. Para cada amostra submetida a teste foram executados em ambos os programas as mesmas técnicas de processamento de imagem, porém a determinação da borda do ponto é diferente para cada software. Por se tratar de algoritmos diferentes pode-se compará-los e determinar a eficiência dos mesmos.

Para um lote de vinte amostras de ponto bom constatou-se que no software Matlab, houve o reconhecimento de 95% dos pontos que estavam dentro das condições exigidas pelo. Já no software Delphi, obteve-se o reconhecimento de 100% dos pontos que estavam dentro das condições exigidas.

Para um lote de vinte amostras de ponto ruim no software Matlab a detecção não permitiu a medição correta de todos os pontos, pois alguns pontos não proporcionavam a determinação da borda, o que gera um erro na determinação do diâmetro do ponto analisado, para este teste a eficiência do projeto limitou-se a 50% das chapas analisadas. No software Delphi alguns pontos não proporcionavam a determinação da borda, o que gera um erro na determinação do diâmetro do ponto analisado, para este teste a eficiência foi de 70% das chapas.

Observou-se que o programa executável desenvolvido em software Delphi apresenta uma eficiência melhor que o programa desenvolvido em Matlab, maior velocidade de análise e interface amigável com o operador, porém em ambos os algoritmos utilizados a eficiência para pontos soltos tem seu resultado comprometido de acordo com o tipo de coloração da solda.

Em trabalhos futuros pretende-se aprimorar as técnicas de processamento de imagens para produzir um resultado melhor nas peças com ponto solto e livre de falsos positivos.



SUGESTÕES PARA TRABALHOS FUTUROS

Pontos a serem melhorados no software Matlab:

- Utilizar resoluções menores para o processamento, pois 640 x 480 tornam esta etapa muito lenta.
- Utilizar mais filtros para melhorar o processamento da imagem.
- Desenvolver uma interface para o usuário.

Pontos a serem melhorados no software Delphi

- Utilizar mais filtros para melhorar o processamento da imagem.
- Desenvolver um algoritmo para ajuste automático de contraste conforme a iluminação do ambiente.



REFERÊNCIAS BIBLIOGRÁFICAS

Gonzales, Rafael C.; Woods, Richard E. - *Processamento de Imagens Digitais*- São Paulo: 2000

Gonzales, Rafael C.; Woods, Richard E. - *Processamento de Imagens Digitais*- São Paulo: 1992

PROENÇA, Claudia Belmiro;CONCI,Aura.Investigação para Detecção Automática de Falhas Têxteis, Revista Brasileira de Ciências Mecânicas , São Paulo, v.21, n.3, p.493-508, 1999

BUENO, Marcelo Lemes;STEMMER, Marcelo Ricardo;BORGES, Paulo Sergio da Silva. Inspeção Visual Automática de peças Cerâmicas via Inteligência Artificial. Cerâmica Industrial, v.5, n.5, p.29-37, 2000

RAMOS, O. T. H. Análise Comparativa entre os principais formatos de armazenamento de imagens. Dissertação de mestrado pela Universidade Presbiteriana Mackenzie. São Paulo, Dezembro de 2000

RODRIGUES, Fabricio Augusto; Localização e Reconhecimento de Placas de Sinalização utilizando Redes Neurais Artificiais. Campina Grande: UFPB, 2002. Dissertação de Mestrado, Ciências da Computação, São Paulo 2002

Stringhini, André M. - *Detecção de Impurezas em Latas de Alumínio* - TCC apresentado em Dezembro de 2006 - Canoas:Universidade Luterana do Brasil, 2006

Tessaro, Antonio P. - *Inspeção de um Processo de Soldagem Robotizada por Análise de Imagens* - Dissertação Mestrado apresentado em 2004 - Canoas:Universidade Luterana do Brasil, 2004

<http://www.atlas.ucepel.br/ubastos/> Acessado em Janeiro de 2008.



GMB, Manual para treinamento interno ,Fundamentos da soldagem por resistência volume 1, General Motors planta Gravataí, 2001

GMB, Manual para treinamento interno ,Fundamentos da soldagem por resistência volume 2, General Motors planta Gravataí, 2001



APÊNDICE A – CÓDIGO FONTE PROGRAMA MATLAB

-ACESSANDO O DISPOSITIVO DE VIDEO

```
close all
```

```
% Acessar um dispositivo de aquisição de imagem.  
wcam = videoinput('winvideo', 1, 'RGB24_640x480')
```

```
% Acessar a fonte de vídeo do dispositivo.  
wfonte = wcam.Source
```

```
% Fonte de vídeo selecionada para aquisição  
set(wcam, 'SelectedSourceName', 'input1');
```

```
% Obter o objeto de fonte vídeo que é selecionado atualmente  
wfontes = getselectedsource(wcam);
```

```
% Listar a propriedade da fonte de vídeo do objeto e seus valores atuais.  
get(wfontes);
```

```
% Pula as primeira Frames ante de pegar a foto.  
wcam.TriggerFrameDelay = 100;
```

```
% Numero de frames por segundo.  
actualRate = 15.15;  
src.FrameRate = num2str(actualRate);
```

**- CAPTURANDO A IMAGEM**

```
% Tirando uma foto
wfoto = getsnapshot(wcam);

% Mostrar a foto em uma janela nova.
imagesc(wfoto);
```

- FILTRANDO A IMAGEM

```
% Transformando em cinza
I = rgb2gray(wfoto);

% Conversão da imagem Preto e Branco
Threshold=graythresh(I);

% Transformando em Binário
BW1=~im2bw(I,threshold);

% Suaviza imagem
BW=bwareaopen(BW1,2000);
```

- DETERMINANDO O DIÂMETRO

```
% Extraindo o local de inicio da borda
dim = size(BW);
col = round(dim(2)/2)-90;
row = min(find(BW(:,col)));

% Traçando as Bordas
conectividade = 8;
num_pontos = 500;
contador = bwtraceboundary(BW, [row, col], 'N', conectividade,
num_pontos);

% Determinando o Circulo de Contorno
x = contador(:,2);
y = contador(:,1);
```



```
% resolver por parâmetros a, b e c usando o operador backslash
abc=[x y ones(length(x),1)]\[-(x.^2+y.^2)];
a = abc(1); b = abc(2); c = abc(3);

% calculo do centro e do raio
xc = -a/2;
yc = -b/2;
raio = sqrt((xc^2+yc^2)-c)

% plotar o circulo inteiro
theta = 0:0.01:2*pi;

% usar representação paramétrica do círculo para obter coordenada de
pontos
% no círculo
Xfit = raio*cos(theta) + xc;
Yfit = raio*sin(theta) + yc;
diametro=raio*0.0515;
```

- AMOSTRA DOS RESULTADOS

```
figure: imshow(wfoto);
hold on;
plot(xc,yc,'yx','LineWidth',2);
mensagem = sprintf('O diâmetro é %2.3f mm', diametro);
text(15,15,mensagem,'Color','y','FontWeight','bold');

if diametro>6.0
    mensagem = sprintf('O Ponto está Bom. ');
    text(15,45,mensagem,'Color','r','FontWeight','bold');

else
    mensagem = sprintf('O Ponto não está Bom . ');
    text(15,45,mensagem,'Color','w','FontWeight','bold');

end
```



APÊNDICE B – CÓDIGO FONTE PROGRAMA DELPHI 1

```
unit uMain;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls,
Forms,
  Dialogs, ievview, imageenvview, ievect, videocap, RzButton, RzTrkBar,
  StdCtrls, RzLabel, RzSpnEdt, hyiedefs, hyieutils, RzBorder, uImageEffects;

var
  frmMain: TfrmMain;

implementation

{$R *.dfm}

- ACESSANDO O DISPOSITIVO DE VIDEO
procedure TfrmMain.BtnCameraOnClick(Sender: TObject);
begin
  Video.ShowVideo := True;
end;

- CAPTURANDO A IMAGEM
procedure TfrmMain.BtnCapturarClick(Sender: TObject);
begin
  Video.Freeze;
  Video.IEBitmap.CopyToTBitmap(Image.Bitmap);
  Video.ShowVideo := False;
  Image.Proc.Resample(640, 480);
  Image.Update;
```



end;

-SEGMENTANDO A IMAGEM

```
procedure TfrmMain.BtnBinarizarClick(Sender: TObject);
begin
  ImageEffects.GrayScale(Image.Bitmap);
  Image.Update;
  ShowMessage('GrayScale - Imagem convertida em tons de cinza');
  ImageEffects.Contrast(Image.Bitmap, 255);
  Image.Update;
  ShowMessage('Contrast - Aumentado o contraste da imagem');
  ImageEffects.Threshold(Image.Bitmap, EdtLevel.Position);
  Image.Update;
  ShowMessage('Threshold - Realizada a binarização da imagem');
end;
```

-DECLARANDO AS VARIÁVEIS

```
procedure TfrmMain.BtnLimitesClick(Sender: TObject);
var X1: Integer;
    X2: Integer;
    Y1: Integer;
    Y2: Integer;
    I : Integer;
    AX: Integer;
    AY: Integer;
    MX: Double;
    MY: Double;
begin
  Image.Proc.ConvertTo24Bit;
  Image.Bitmap.Canvas.Pen.Color := clRed;
  Image.Bitmap.Canvas.Brush.Style := bsClear;
```

-ALGORITMO PARA DETECÇÃO DAS BORDAS

```
for X1 := 320 downto 1 do
```



```
begin
if Image.Bitmap.Canvas.Pixels[X1, 240] = clWhite then I := I + 1 else I := 0;
  if I >= EdtLimite.Value then Break;
end;

for X2 := 320 to 640 do
begin
if Image.Bitmap.Canvas.Pixels[X2, 240] = clWhite then I := I + 1 else I := 0;
  if I >= EdtLimite.Value then Break;
end;

for Y1 := 240 downto 1 do
begin
if Image.Bitmap.Canvas.Pixels[320, Y1] = clWhite then I := I + 1 else I := 0;
  if I >= EdtLimite.Value then Break;
end;

for Y2 := 240 to 480 do
begin
if Image.Bitmap.Canvas.Pixels[320, Y2] = clWhite then I := I + 1 else I := 0;
  if I >= EdtLimite.Value then Break;
end;

X1 := X1 + EdtLimite.Value - 1;
X2 := X2 - EdtLimite.Value + 2;
Y1 := Y1 + EdtLimite.Value - 1;
Y2 := Y2 - EdtLimite.Value + 2;
```

-ALGORITMO PARA CALCULAR O DIÂMETRO

```
Image.Bitmap.Canvas.Rectangle(X1, Y1, X2, Y2);
AX := X2 - X1;
AY := Y2 - Y1;
MX := (8.4 * AX) / 214;
MY := (8.4 * AY) / 214;

Image.Bitmap.Canvas.Font.Style := [];
Image.Bitmap.Canvas.Font.Name := 'Tahoma';
```



```
Image.Bitmap.Canvas.Font.Size := 8;
Image.Bitmap.Canvas.Font.Color := clBlue;
Image.Bitmap.Canvas.TextOut(X1, (Y1 - 14), FormatFloat('0.00 mm', MX));
Image.Bitmap.Canvas.TextOut((X2 + 2), ((Y2 + Y1) div 2) - 5,
FormatFloat('0.00 mm', MY));
```

```
Image.Bitmap.Canvas.Pen.Color := clWhite;
Image.Bitmap.Canvas.Brush.Color := clWhite;
Image.Bitmap.Canvas.Rectangle(0, 448, 107, 480);
```

```
Image.Bitmap.Canvas.Font.Style := [fsBold];
Image.Bitmap.Canvas.Font.Color := clBlue;
Image.Bitmap.Canvas.TextOut(3, 450, 'Média: ' + FormatFloat('0.00 mm',
((MX + MY) / 2)));
if ((MX + MY) / 2) > 6 then
begin
  Image.Bitmap.Canvas.Font.Color := clGreen;
  Image.Bitmap.Canvas.TextOut(3, 465, 'Ponto Aprovado');
end
else
begin
  Image.Bitmap.Canvas.Font.Color := clRed;
  Image.Bitmap.Canvas.TextOut(3, 465, 'Ponto Reprovado');
end;

end;
```

```
procedure TfrmMain.BtnOpenFileClick(Sender: TObject);
begin
  if OpenFileDialog.Execute then
  begin
    Image.IO.LoadFromFileAuto(OpenDialog.FileName);
    Image.Proc.Resample(640, 480);
  end;
end;
```



```
procedure TfrmMain.VideoVideoFrame(Sender: TObject; Bitmap:
TImage; Canvas: TCanvas; ARect: TRect)
begin
  Bitmap.CopyToTBitmap(Image.Bitmap);
  Image.Proc.Resample(640, 480);
  Image.Bitmap.Canvas.Brush.Style := bsClear;
  Image.Bitmap.Canvas.Pen.Color := clRed;
  Image.Bitmap.Canvas.Rectangle(200, 120, 440, 360);
  Image.Bitmap.Canvas.Pen.Color := clWhite;
  Image.Bitmap.Canvas.MoveTo(320, 0);
  Image.Bitmap.Canvas.LineTo(320, 480);
  Image.Bitmap.Canvas.MoveTo(0, 240);
  Image.Bitmap.Canvas.LineTo(640, 240);
  Image.Update;
end;

procedure TfrmMain.RzBitBtn1Click(Sender: TObject);
begin
  Video.DoConfigureFormat;
end;

end.
```



ANEXO A- DADOS DO WEBCAM

- Microsoft LifeCam Versão 1.40.164.0
- Captura fotografias e vídeos com resolução de até 1.4 Megapixel (1280 x 960 pixel).
- Pode, através do software que acompanha, aumentar a imagem em até quatro Megapixels.
- Desempenho de baixa-luz: Tecnologia Microsoft, com sensor de alto-desempenho e processamento de imagem avançada para corrigir a iluminação.
- Requerimentos de sistema: *Windows®* 2000, XP, Vista, *Pentium®* 4 1.4GHz ou AMD *Athlon®* processador 1GHz (2.4GHz ou mais rápido), 128MB RAM (256MB recomendado), 200MB de espaço livre no HD, CD-ROM *drive*, 16-bit adaptador de exibição de cor, porta 1.1 ou 2.0 USB (USB 2.0 requerida para aquisição de imagens Megapixel).