

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
DEPARTAMENTO ACADÊMICO DE MECÂNICA
CURSO DE TECNOLOGIA EM MECATRÔNICA INDUSTRIAL

DAGOBERTO OKASAKI
DENIS CHRISTOPH DÜCK
EDUARDO DA SILVA RODRIGUES ALVES

**SOFTWARE DE PROCESSAMENTO DE IMAGENS PARA UM SISTEMA DE
VISÃO DE MÁQUINA**

TRABALHO DE CONCLUSÃO DE CURSO

CURITIBA

2011

DAGOBERTO OKASAKI
DENIS CHRISTOPH DÜCK
EDUARDO DA SILVA RODRIGUES ALVES

**SOFTWARE DE PROCESSAMENTO DE IMAGENS PARA UM SISTEMA DE
VISÃO DE MÁQUINA**

Trabalho de Conclusão de Curso de graduação, apresentado à disciplina de Trabalho de Diplomação, do Curso Superior de Tecnologia em Mecatrônica Industrial do Departamento Acadêmico de Mecânica – DAMEC -da Universidade Tecnológica Federal do Paraná – UTFPR, como requisito parcial para obtenção do título de Tecnólogo.

Orientador: Prof. Dr. Marcelo Victor Wüst Zibetti

CURITIBA

2011

DAGOBERTO OKASAKI
DENIS CHRISTOPH DÜCK
EDUARDO DA SILVA RODRIGUES ALVES

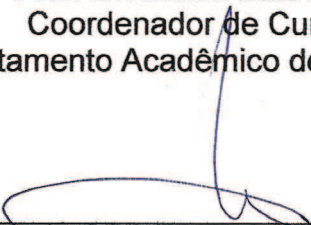
**SOFTWARE DE PROCESSAMENTO DE IMAGENS PARA UM SISTEMA DE
VISÃO DE MÁQUINA**

Esta Monografia foi julgada e aprovada como requisito parcial para obtenção do título de Tecnólogo em Mecatrônica Industrial pela Universidade Tecnológica Federal do Paraná.

Curitiba, 25 de março de 2011.

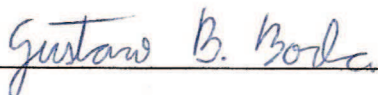


Prof. Dr. Milton Luiz Polli
Coordenador de Curso
Departamento Acadêmico de Mecânica



Prof. Dr. Décio Estevão do Nascimento
Responsável pelo Trabalho de Conclusão de Curso
Departamento Acadêmico de Eletrônica

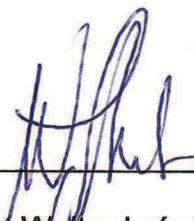
BANCA EXAMINADORA



Prof. Dr. Gustavo Benvenuto Borba



Prof. Dr. Marcelo Victor Wüst Zibetti
Orientador



Prof. Dr. Walter Luís Mikos

Aos nossos pais, que nos acompanharam durante o processo e nos apoiaram com palavras e petiscos em dias em que precisamos gastar horas nos reunindo para elaboração do projeto.

Às nossas namoradas e esposas, que sacrificaram momentos que poderiam ter passado conosco para que pudéssemos fazer reuniões com a equipe ou com nosso orientador.

Ao nosso orientador, que sempre se mostrou prestativo e que nos ajudou a transpor algumas barreiras de falta de conhecimento que tínhamos em alguns momentos do projeto.

À Deus, que nos deu a vida e que nos guiou até aqui.

RESUMO

ALVES, Eduardo da Silva Rodrigues; DUCK, Denis Christoph; OKASAKI, Dagoberto.

Software de processamento de imagens para um sistema de visão de máquina.

Trabalho de conclusão de curso – Curso Superior de Tecnologia em Mecatrônica Industrial, Universidade Tecnológica Federal do Paraná. Curitiba, 2011.

A crescente demanda por bens duráveis vem obrigando as empresas a se preocuparem com a qualidade de seus produtos. Como exemplo, diversas empresas estão oferecendo garantias mais longas para seus produtos, o que, de fato, passou a se tornar uma exigência do mercado. A qualidade de um produto está diretamente ligada à qualidade de sua fabricação e à conformidade de seus componentes. Para a verificação de qualidade e conformidade é necessária uma inspeção eficiente. A inspeção de qualidade, anteriormente feita por amostragem, agora muitas vezes precisa ser feita em 100% das peças produzidas. Os sistemas de inspeção através de imagens, ou sistemas de visão, atingiram um grau de maturidade para serem utilizadas na indústria. Este trabalho tem como objetivo desenvolver um programa de inspeção por visão de máquina capaz de obter dados de objetos através de imagens capturadas por câmeras digitais e usá-las para controle de qualidade em uma produção flexível automatizada.

Palavras-chave: Inspeção, Visão de Máquina, Imagem.

ABSTRACT

ALVES, Eduardo da Silva Rodrigues; DUCK, Denis Christoph; OKASAKI, Dagoberto. **Image processing software for a machine vision system.** Trabalho de conclusão de curso – Curso Superior de Tecnologia em Mecatrônica Industrial, Universidade Tecnológica Federal do Paraná. Curitiba, 2011.

The increasing demand for durable goods is forcing the companies to take care about the quality of their products. For example, some companies are offering longer warranties for their products, and this is becoming a requirement by the market. The product quality is directly related to its manufacturing quality and its components. An efficient inspection is needed to verify the quality and accordance of the products. The quality inspection, usually based on statistical sampling, nowadays is commonly performed on 100% of the parts. The image inspection systems, or vision systems, present performance and reliability which attend the industry requirements. This work aims to develop a machine vision inspection software, able to acquire image data from a digital camera and use them to control the quality of an automatic flexible production.

Keywords: Inspection, Machine Vision, Image.

LISTA DE FIGURAS

Figura 1 – Componentes de uma câmera industrial.....	18
Figura 2 – Fluxograma do <i>software</i> de inspeção.....	25
Figura 3 – Tela principal do programa de inspeção	26
Figura 4 – Imagem carregada no programa.....	27
Figura 5 – Imagem pré-processada	28
Figura 6 – Cálculo da área de um CD.....	29
Figura 7 – Cálculo da borda	30
Figura 8 – Contagem de furos ou cavidades.....	31
Figura 9 – Tela para seleção do furo para ser medido.....	32
Figura 10 – Exibição do diâmetro do furo medido.....	32

SUMÁRIO

1	Introdução	10
1.1	Histórico	10
1.2	FMS	11
1.3	Inspeção	11
1.4	Problema.....	12
1.5	Justificativa.....	13
1.6	Objetivos	14
1.6.1	Objetivo Geral	14
1.6.2	Objetivos Específicos	15
1.7	Método de pesquisa.....	15
1.8	Estrutura do trabalho.....	16
2	Fundamentação teórica	17
2.1	Novas tecnologias de inspeção.....	17
2.2	Inspeção por Imagens - Visão de Máquina.....	17
2.2.1	Câmeras Industriais	18
2.2.2	Hardware de Processamento.....	19
2.2.3	Software de Processamento	19
2.3	Processamento Digital de Imagens.....	20
2.3.1	Aquisição da imagem	22
2.3.2	Pré-processamento	22
2.3.3	Segmentação	22
2.3.4	Detecção de bordas.....	23
2.3.5	Reconhecimento	23
2.3.6	Interpretação	24
3	Desenvolvimento	25
3.1	Interface Gráfica.....	26
3.2	Aquisição.....	26
3.3	Pré-processamento.....	27
3.4	Tarefas de Inspeção	28
3.4.1	Cálculo da área	29
3.4.2	Cálculo de borda	29
3.4.3	Número de furos.....	30

3.4.4 Medição de furos.....	31
4 Conclusões	33
Referências	34
Apêndice A – Código MATLAB por bloco funcional	36

1 INTRODUÇÃO

1.1 HISTÓRICO

A história da manufatura teve uma longa evolução, iniciada no final do século XVIII. O movimento conhecido como a primeira revolução industrial ficou marcado pelo deslocamento da população rural para as cidades. Suas principais características foram a criação das fábricas e o desenvolvimento de máquinas que substituíram o serviço manual do homem, as máquinas-ferramenta, inicialmente movidas à vapor.

A segunda revolução industrial, a partir dos anos 1900, mudou novamente o rumo da manufatura, introduzindo a produção em massa através das linhas de fabricação e montagem dedicadas. Estas estratégias de produção foram implementadas principalmente por Henry Ford (fordismo) e Frederick Taylor (taylorismo), conforme Groover (2002). Mais tarde, surgiram linhas especializadas e automatizadas, denominadas linhas *transfer*.

Hoje, vivemos num período conhecido como terceira revolução industrial, que se iniciou após a segunda guerra mundial. Neste período, foi observado que havia uma grande lacuna que os sistemas dedicados de alta produção não conseguiam preencher – variedade de produtos produzidos por uma linha. Caso houvesse alguma mudança do produto que a linha produzia, eram necessários grandes ajustes na mesma para produzir o novo produto. Estes ajustes geralmente necessitavam de grandes períodos de tempo para serem feitos, o que acabava se tornando impraticável neste tipo de produção, pois o tempo de máquina parada causava grandes prejuízos.

Nessa época, o Japão, arrasado pela segunda guerra mundial, precisava retomar o seu crescimento. Para tal, comprava grandes quantidades de máquinas do ocidente para desmontá-las com o objetivo de entender e reproduzir as tecnologias empregadas. As empresas japonesas passavam por sérias crises financeiras e para reverter isto foram empregadas estratégias de produção focadas no mínimo desperdício, tanto de material quanto de tempo, denominadas hoje de estratégias de produção enxuta, ou *lean line*. Estas estratégias começaram a ser aprimoradas, com destaque para a montadora de carros japonesa Toyota, que criou o sistema de manufatura hoje conhecido como TPS ou *Toyota Production System*

(Sistema de Produção Toyota). Esse tipo de estratégia de produção reduziu os custos de manufatura dos produtos e aumentou o lucro das empresas, possibilitando a recuperação do Japão naquela época.

Esse novo sistema de manufatura fez com que alguns antigos conceitos de manufatura fossem mudados, dentre eles justamente o conceito de flexibilidade. Ao ouvirem falar do TPS, várias outras indústrias começaram a se preocupar mais com a distribuição de suas máquinas e a movimentação de materiais em suas fábricas, o que, segundo o TPS, eram fatores que traziam perdas as empresas. Assim, a preocupação com o sistema de produção tornou-se importante. Um dos sistemas de manufatura criado com o objetivo de ser flexível foi o FMS.

1.2 FMS

Conforme Groover (2002), um FMS, sigla que traduzida significa Sistema Flexível de Manufatura (do inglês, *Flexible Manufacturing System*), é um sistema altamente automatizado composto por máquinas flexíveis voltadas à produção de um grupo de tecnologia, isto é, peças com características similares. Este sistema consiste em: estações de processamento (normalmente máquinas CNC), sistemas automáticos de manuseio de material, troca de ferramenta e armazenamento, todos integrados e controlados por computadores.

Além destes, um importante elemento que compõem um FMS é o sistema de inspeção, responsável por garantir a conformidade e a qualidade dos produtos manufaturados. Neste projeto, o objeto de estudo é o sistema de inspeção.

1.3 INSPEÇÃO

De acordo com Groover (2008), inspeção é um processo no qual uma característica de uma peça ou produto, como a dimensão de um furo, por exemplo, é examinada para determinar se está de acordo com a especificação do projeto. Este processo é freqüentemente manual e além de ser repetitivo e demorado, faz com que os custos de produção sejam elevados. Em muitos casos, para redução de tempo e custo, as inspeções são feitas por amostragem em lotes de peças. O que vai contra o TPS que tem como característica a aplicação da inspeção em 100% dos produtos.

Com a responsabilidade das empresas da atualidade de garantir a qualidade de seus produtos ao consumidor, observa-se um abandono crescente da inspeção por amostragem e uma adoção cada vez maior da inspeção de todas as peças manufaturadas. Sendo assim, os sistemas de inspeção automatizados são cada vez mais requisitados, pois fazem a inspeção com maior precisão e rapidez que aquela realizada manualmente. Várias tecnologias desenvolvidas nos últimos anos na área de medições têm sido empregadas com sucesso, cada uma com suas vantagens e desvantagens.

Recentemente, surgiram os sistemas de inspeção através de imagens. Esta nova tecnologia está sendo cada vez mais explorada e tem mostrado grande potencial na indústria.

1.4 PROBLEMA

Atualmente a Universidade Tecnológica Federal do Paraná (UTFPR) possui, para estudo e pesquisa, um FMS (que pode ser mais corretamente denominado de FMC – *Flexible Manufacturing Cell* – Célula de Manufatura Flexível), montado para fins didáticos na sala A-007, sobre responsabilidade do Departamento Acadêmico de Mecânica (DAMEC). Este sistema realiza as seguintes funções:

- Transporte;
- Manipulação de peças;
- Armazenamento;
- Manufatura;
- Controle.

Entretanto, o sistema ainda não possui uma estação de medição funcional. No *lay-out* do FMS está previsto um espaço e uma estrutura para a implantação do mesmo. O tipo de estrutura presente é favorável para a instalação de um sistema de inspeção por imagem, pois possui espaço para uma câmera a qual será utilizada para a aquisição de imagens das peças. O sistema em questão pode ser completamente automatizado e trabalhar com alta velocidade de inspeção, ou seja, não representaria um atraso para o sistema.

Uma alternativa para preencher a falta do sistema de inspeção seria adotar uma solução já existente no mercado. Porém, a mesma apresenta a desvantagem

do elevado custo e falta do controle e conhecimento sobre o funcionamento da tecnologia empregada, impossibilitando o estudo e implementação de melhorias para o conceito empregado. Desta forma, o desenvolvimento completo de um sistema de inspeção de imagens permitirá que professores, pesquisadores e alunos tenham acesso completo a esta tecnologia. Uma vez pronto o sistema poderá ser usado para ensino e pesquisa. Diversos projetos poderão ser realizados posteriormente com a finalidade de aprimoramento e ampliação dos conhecimentos sobre inspeção por imagens.

1.5 JUSTIFICATIVA

Os equipamentos de inspeção por imagens são capazes de realizar a inspeção de forma totalmente automática, através da aquisição de imagens por uma câmera e um posterior processamento. O *software* de processamento de imagens é geralmente o principal componente do equipamento. O processamento que pode ser realizado pelo *software* varia desde uma comparação com um padrão pré-existente até medições complexas.

O tipo de inspeção realizada por este equipamento é sem contato, isto é, isenta a peça inspecionada de operações que possam causar desgaste ou pequenos danos. Além disso, existe a possibilidade da calibração do sistema ser feita de forma automática, preparando o equipamento para operação durante a sua inicialização.

O *software* de processamento de imagens não terá como requisito para funcionamento nenhum equipamento de difícil acesso, apenas uma câmera digital compacta e um computador, os quais são facilmente encontrados e não possuem custo elevado se comparados às soluções já existentes no mercado.

O desenvolvimento do *software* de processamento de imagens é uma parte necessária para implementação do sistema de inspeção por imagens no FMS existente na UTFPR. O foco do projeto como um todo é disponibilizar este equipamento para discentes e docentes. Por se tratar de uma tecnologia recente, existe a necessidade de estudos e pesquisas envolvendo processamento de imagens.

Os conhecimentos técnicos envolvidos no desenvolvimento deste *software* incluem a manipulação e processamento de imagens digitais, fundamentos de

inspeção, experimentos em MATLAB e programação em C/C++ em plataforma Windows.

O desenvolvimento deste projeto é extremamente motivante por se tratar de uma tecnologia nova e que pode ser expandida para várias aplicações. O processamento de imagens pode ser utilizado em sistemas de segurança, reconhecimento de objetos e pessoas, controle de processos automatizados, além de robótica e inteligência artificial. A implementação desta proposta no FMS, juntamente com os outros projetos envolvidos, poderá dar início a criação de um sistema aberto, mais flexível que as soluções de visão de máquina existentes no mercado, e permitirá a realização de pesquisas e geração patentes e registros de *software*. O projeto é economicamente viável por não necessitar de equipamentos específicos e caros, porém tem alto valor agregado no que diz respeito à lógica do *software* desenvolvido.

1.6 OBJETIVOS

Atualmente, o domínio sobre a tecnologia de processamento de imagens ainda não é suficientemente amplo. Tanto que esse tópico não foi abordado no curso de graduação do qual os alunos participaram. Mesmo assim, foi dado ao grupo o desafio de compreender essa nova tecnologia e, com o conhecimento e o apoio do professor orientador, criar um programa capaz de fazer a verificação de peças produzidas na planta do FMS, comparando com os dados dos desenhos. Isto supre a necessidade do FMS da UTFPR de ter um sistema de medição funcional e aberto, onde a instituição detenha o conhecimento da tecnologia para possibilitar futuros aprimoramentos do processo, sendo esta a sua principal vantagem.

1.6.1 Objetivo Geral

Desenvolver para a planta do FMS um programa que terá a função de processar imagens captadas por uma câmera digital através da utilização de técnicas de processamento de imagens e de dados sobre a peça inspecionada.

1.6.2 Objetivos Específicos

Aprender sobre processamento de imagens, através de leituras técnicas, aulas de processamento de imagens e por meio de testes e simulações no *software* MATLAB, o qual possui uma biblioteca de processamentos de imagens incorporada. Além disso, serão pesquisadas algumas bibliotecas para este processamento em linguagem “C” ou “C++”.

Identificar, em ambiente MATLAB, quais os melhores métodos de processamento para a aplicação em inspeção de peças e, em seguida, definir as melhores seqüências de filtros e tarefas que serão utilizados no *software* desenvolvido para se obter o melhor resultado em tarefas específicas de inspeção.

O software desenvolvido tem o objetivo de realizar as seguintes tarefas:

- Calibração automática do tamanho do pixel, relacionando a dimensão do mesmo à uma dimensão física;
- Inspeção por comparação com imagem padrão;
- Estimativa de dimensões máximas horizontais e verticais da peça;
- Contagem de furos ou cavidades na peça.

Criar uma interface homem-máquina (IHM), para facilitar o uso do programa pelos operadores do FMS. O programa possuirá dentro de sua programação a seqüência de processamento da imagem capturada e a interpretação dos dados da peça de forma que seja possível comparar as medidas obtidas da peça física com aquelas adquiridas do desenho.

Compilação de toda a documentação técnica do programa, juntamente com os relatórios do projeto, contendo todas as informações do mesmo, incluindo a teoria dos processos envolvidos e os códigos do programa.

1.7 MÉTODO DE PESQUISA

Para atender as necessidades existentes no FMS, deve-se levar em conta que, primeiramente, temos que trabalhar em uma solução simples e eficaz para o sistema de inspeção. O foco do trabalho é desenvolver um software de processamento que seja didático e esteja pronto para receber aprimoramentos.

Iniciou-se o projeto através do estudo dos algoritmos de processamento de imagem através do livro Gonzalez e Woods (2008). A partir deste estudo foram

selecionados os tipos de filtros e algoritmos mais eficientes para cada tipo de inspeção. A seleção dos filtros e das técnicas mais eficientes de processamento de imagens foi feita com a ajuda do software MATLAB, da Mathworks Inc.. Este programa serviu de base para testar a eficiência dos algoritmos implementados no projeto.

Após a conclusão dessa etapa e já tendo em mãos os métodos corretos para o processamento de imagens, foi desenvolvida uma interface para o programa principal. A interface homem-máquina (IHM) para o FMS deve ser enxuta, prática e oferecer recursos para a manipulação de arquivos e *presets* necessários para a aquisição e processamento de imagem, buscando disponibilizar ao usuário o melhor método possível para obter um bom resultado. A IHM possui uma tela principal na qual o usuário tem acesso fácil aos comandos através de botões e campos com informações obtidas através do processamento da imagem conforme pode ser visto na Figura 2.

Como etapa final, foram testadas as velocidades e tempos de resposta do programa principal em diferentes configurações de computadores a fim de aprimorar o desempenho dos algoritmos de processamento de imagem utilizados.

1.8 ESTRUTURA DO TRABALHO

Este trabalho está organizado da seguinte forma: introdução (este capítulo), desenvolvimento e considerações finais. A introdução situa o leitor a respeito do tema escolhido como alvo de estudo deste trabalho, o problema a ser resolvido e a sua justificativa. O desenvolvimento apresenta os conceitos e os métodos utilizados durante a execução do projeto bem como os resultados obtidos. No capítulo final é feita a análise dos resultados e a partir dos mesmos relaciona as aplicações para as quais o programa desenvolvido pode ser utilizado e também possibilidades de expansão do mesmo.

2 FUNDAMENTAÇÃO TEÓRICA

A crescente evolução dos processos produtivos em termos de velocidade e variedade de produtos necessita que novos métodos de inspeção sejam criados para garantir que os produtos entregues ao cliente estejam de acordo com as especificações de projeto. Segundo Groover (2002), tecnologias avançadas estão substituindo medições e calibrações manuais em plantas de manufatura moderna.

2.1 NOVAS TECNOLOGIAS DE INSPEÇÃO

Existem quatro grandes áreas de novos métodos de inspeção conforme Groover (2008): máquinas de medição por coordenadas, lasers, visão de máquina e outras técnicas sem contato. A visão de máquina, ou sistema de visão, envolve a aquisição, processamento e interpretação dos dados por um computador para diversas aplicações. Entre estas aplicações a empresa Sick, fornecedora de instrumentação industrial, destaca: verificação de dimensões de componentes, verificação da presença de furos, porcas, inspeção de superfícies a procura de defeitos, controle de nível de fluidos.

Os sistemas de visão existentes em máquinas hoje são capazes de processar informações a taxas extremamente elevadas e possuem um poder muito grande para armazenamento de dados, porém, são inflexíveis sob diversas condições de trabalho, realizando de forma eficiente somente a tarefa para a qual foram projetados, conforme Pedrino (2008). A flexibilidade dos sistemas de visão depende de três elementos: dispositivo de aquisição, *software* de processamento e interpretação.

2.2 INSPEÇÃO POR IMAGENS - VISÃO DE MÁQUINA

O recurso de inspeção por imagem tem tornado-se cada vez mais confiável e robusto devido tanto à evolução tecnológica tanto das câmeras utilizadas para inspeção, quanto às melhorias nos meios de processamento. Dentro da indústria, os recursos de visão de máquina substituem com facilidade as inspeções visuais, as quais eram realizadas por mão-de-obra humana, garantindo maior robustez, confiabilidade e velocidade.

A seguir, estão detalhados os componentes da inspeção por imagens: as câmeras industriais, o *hardware* de processamento e o *software* de processamento,

2.2.1 Câmeras Industriais

Os sistemas de visão têm como equipamento básico uma câmera para a captação das imagens para análise do sistema de processamento de dados, Gonzalez e Woods (2008). Em um sistema de visão industrial, as câmeras são, em geral, mais robustas que as convencionais e preparadas para o chão de fábrica. Contudo, o princípio de funcionamento é o mesmo das câmeras digitais convencionais.

Os principais componentes da câmera são: lentes, sensor de imagem, interface de comunicação e a eletrônica embarcada. As lentes são responsáveis pela formação da imagem no sensor. O sensor de imagem, por sua vez, converte a luz em um sinal digital que irá compor a imagem digital. A eletrônica de suporte controla o processo de conversão da imagem e a armazena em uma memória interna. Por último, a imagem é transferida até o hardware de processamento através da interface da câmera. As interfaces mais comuns são a Gigabit Ethernet, USB e FireWire (IEEE-1394a/b). A Figura ilustra estes componentes.

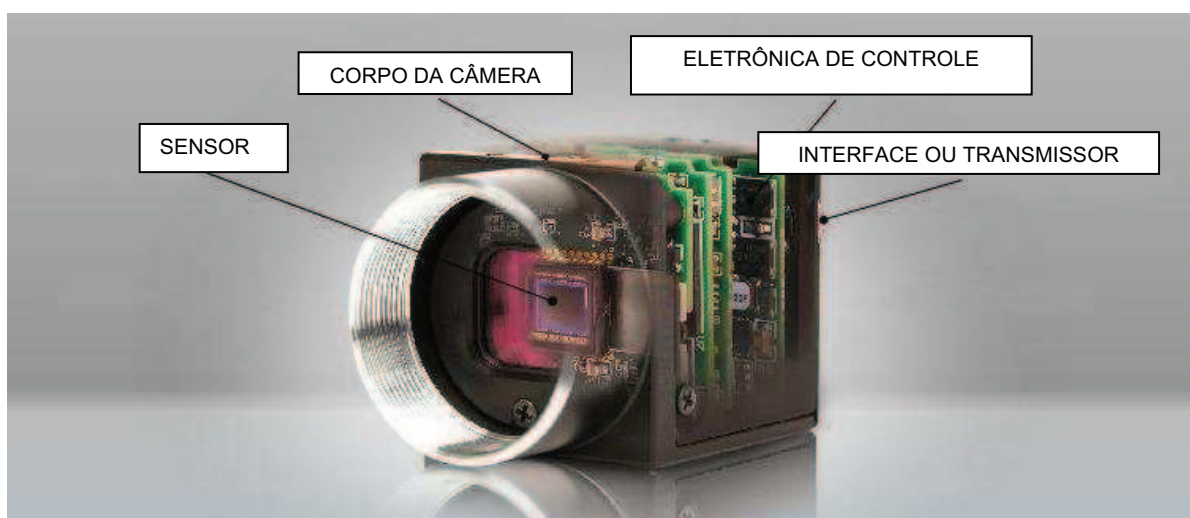


Figura 1 – Componentes de uma câmera industrial
Fonte: www.flea3.com

2.2.2 Hardware de Processamento

O *hardware* de processamento é responsável por executar as tarefas de processamento de imagens. Em geral, um computador pessoal, ou industrial, é adequado para esta tarefa. Além disto, este equipamento é flexível o suficiente para executar outras tarefas, como as tarefas de controle do sistema.

Além dos computadores, hardwares específicos também estão disponíveis, como as câmeras inteligentes e os sistemas compactos da empresa National Instruments. As câmeras inteligentes incorporam soluções tudo em um: alta qualidade de imagem e processamento em um equipamento robusto suficiente para operar no chão de fábrica. Os sistemas compactos consistem em um equipamento robusto e amigável onde o usuário tem a possibilidade de operar simultaneamente mais de uma câmera inteligente além de oferecer suporte às redes industriais e programação remota.

2.2.3 Software de Processamento

O software de processamento é responsável por realizar as tarefas de inspeção. O presente trabalho prevê exatamente a criação deste componente do sistema de visão. Em geral, as tarefas de inspeção devem ser programadas para serem executadas no hardware previamente especificado. O tipo de programação a ser realizada depende do conhecimento na área e da habilidade do programador.

A programação de sistemas de visão de máquina pode ser tão genérica quanto possível, realizada em C/C++, por exemplo, ou mais específica, usando sistemas pré-prontos, como o Vision Builder AI, da National Instruments.

Na programação genérica, em C/C++, tudo deve ser programado, desde a interface gráfica, até as funções de visão de máquina. O uso de bibliotecas pré-prontas, como o OpenCV, oferece uma possibilidade de realizar a programação mais rapidamente, já que as funções de visão de máquina estão prontas. Contudo, esta forma de programação ainda é a mais trabalhosa.

O uso de sistemas específicos pré-programados, como o Vision Builder AI da National Instruments, é interessante para o "integrador", que não tem conhecimentos de programação de sistemas de visão, mas conhece os problemas

da aplicação específica. Assim o desenvolvimento do programa de inspeção torna-se rápido, contudo pouco flexível e restrito ao uso de uma tecnologia fechada.

A alternativa intermediária, adotada neste trabalho, é o desenvolvimento em MATLAB. Este oferece ao usuário uma série de comandos estruturados que permitem uma possibilidade de personalização aceitável para grande parte dos tipos de inspeções visuais mais simples como detecção de bordas, limiarização e binarização sem perder a flexibilidade. Com o MATLAB, a programação é mais simples que a realizada em C/C++ com OpenCV. Além disso, a forma de programação é suficientemente genérica para se obter uma flexibilidade. Frente aos tipos de programação disponíveis para a realização do trabalho, optou-se por desenvolver a programação com o software MATLAB pela sua facilidade de manipulação e personalização e o seu grande número de comandos de processamento de imagens disponíveis, Eddins (2008).

2.3 PROCESSAMENTO DIGITAL DE IMAGENS

O objetivo do software de processamento é realizar a tarefa de inspeção por imagens. Como tal, os conhecimentos de processamento de imagens são essenciais para o desenvolvimento deste trabalho. A seguir, serão apresentados alguns fundamentos do processamento de imagens utilizado em visão de máquina.

Segundo Gonzalez e Woods (2008), o processamento de imagens digitais parte de duas áreas de aplicação:

- Melhoria da informação fotográfica para a interpretação humana e;
- Processamento de dados em forma de imagem para armazenamento, transmissão e representação para máquinas com percepção autônoma.

Isso pode ser visto claramente quando se observa os diversos campos de aplicação desse tipo de tecnologia. O desenvolvimento de telas para exibição de imagens que comportam cada vez resoluções maiores e a criação de máquinas automotoras, capazes de reconhecer caminhos, objetos e obstáculos, exemplifica claramente essas duas áreas supridas por essa tecnologia.

Embora tenha potencial para aplicação em diversas áreas, o processamento de imagens ainda é uma tecnologia nova e em muitos aspectos inexplorada. O processamento de imagens digitais é um campo extremamente amplo e intimamente ligado com outros, como análise de imagens e inteligência artificial. Devido a isto,

não existe até hoje uma definição clara de processamento de imagens. Em seu livro, Gonzalez e Woods (2008) sugerem uma definição:

“O que definimos como Processamento de Imagens Digitais neste livro são os processos cujas entradas e saídas são ambas imagens e, além disso, aqueles que extraem informações de imagens...” .

Mesmo com essa dificuldade para achar uma definição, uma classificação interessante é de considerar três níveis de processos computadorizados referentes ao processamento de imagens digitais: níveis baixo, médio e alto.

Processos de nível baixo envolvem operações primitivas, como pré-processamento para reduzir o ruído, aumento do contraste e aumento do brilho da imagem. É caracterizado por possuir como entrada e saída uma imagem, contudo, muitas vezes a entrada pode ser um sinal que não é uma imagem, como na tomografia ou ultrassonografia.

Processamentos de nível médio em imagens envolvem tarefas como a segmentação (partição de uma imagem em regiões ou objetos distintos), a descrição desses objetos para reduzi-los para uma forma adequada para o processamento pelo computador e a classificação (reconhecimento) de objetos individuais. Este nível é caracterizado como tendo imagens como entrada, porém possuindo informações retiradas daquela imagem em sua saída, conforme Gonzalez e Woods (2008).

Finalmente, o processamento de nível alto envolve o “fazer sentido” de um conjunto de objetos reconhecidos e simulando as funções cognitivas geralmente associadas à visão.

Para entender melhor esses aspectos, Gonzalez e Woods (2008) definem como um sistema computadorizado interpreta uma imagem. No processamento digital, a imagem é vista como uma função de duas dimensões, $f(x,y)$, onde x e y são coordenadas espaciais, e a amplitude de f em qualquer par de coordenadas (x,y) é chamada de intensidade ou *Gray Level* (nível de cinza) da imagem naquele ponto. Quando os valores de x , y e da amplitude são quantidades finitas e discretas chamamos a imagem de uma imagem digital. Cada ponto corresponde a uma unidade mínima na imagem, a qual é chamada de pixel. O tamanho físico do pixel determina a resolução final da imagem. Quanto mais pixels uma imagem possuir dentro da mesma área, maior será resolução e, portanto, melhor será sua nitidez.

Com o objetivo de extrair informações de uma imagem digital são necessários alguns passos fundamentais para executar esta tarefa. No campo da inspeção, a imagem é normalmente obtida em um ambiente controlado, no qual serão capturadas somente imagens do produto a ser inspecionado. Define-se assim o domínio do problema, conforme Gonzalez e Woods (2008). O sistema deve estar preparado para avaliar as imagens dos produtos para os quais foi projetado produzindo resultados que sirvam como parâmetros de aprovação ou reprovação.

Após definido o domínio do problema, os passos necessários para obter informações da imagem digital são: aquisição da imagem, pré-processamento, segmentação, reconhecimento e interpretação.

2.3.1 Aquisição da imagem

A aquisição da imagem digital depende de dois elementos: dispositivo de sensoriamento físico e digitalizador. O dispositivo de sensoriamento tem a função de converter um espectro de energia eletromagnética – raios X, ultravioleta, visível ou infravermelho – em um sinal elétrico. A intensidade deste sinal elétrico deve ser proporcional à intensidade da energia eletromagnética captada segundo Gonzalez e Woods (2008).

O sinal elétrico é então convertido pelo digitalizador – conversor analógico-digital - para a forma digital.

A aquisição de imagens através de uma câmera digital compacta utiliza este processo. No seu interior, ela possui um sensor de imageamento chamado CCD – dispositivo de carga acoplada.

2.3.2 Pré-processamento

Gonzalez e Woods (2008) definem como pré-processamento as técnicas utilizadas para melhorar a qualidade da imagem melhorando os resultados das etapas posteriores. Estas técnicas envolvem o realce o balanceamento do contraste, redução de ruídos.

2.3.3 Segmentação

A segmentação é um importante e muitas vezes o mais difícil passo durante o processo de extração de informações de uma imagem. É através da segmentação que os objetos de uma imagem são isolados para que seja possível analisá-los individualmente. Segundo Gonzalez e Woods (2008), uma segmentação feita da forma correta pode garantir o sucesso da análise de um objeto.

Os algoritmos de segmentação utilizados em imagens monocromáticas podem ser divididos em duas categorias. A primeira categoria utiliza como base a descontinuidade no valor do nível de cinza de pixels vizinhos enquanto que a segunda categoria utiliza a similaridade.

Os algoritmos de descontinuidade podem ser utilizados para detectar pontos isolados, linhas e bordas em uma imagem, sendo esta última uma das técnicas com mais aplicações práticas.

Através da similaridade, obtém-se a discriminação em regiões com características similares e informações quantitativas das mesmas. Uma importante característica obtida através desta categoria é a quantidade de pixels em uma região. Esta medida está relacionada diretamente com a área do objeto.

2.3.4 Detecção de bordas.

A detecção de bordas é uma das tarefas mais usuais no campo do processamento de imagens. Gonzalez e Woods (2008) definem uma borda como sendo o limite entre duas regiões que possuem propriedades relativamente distintas de nível de cinza. A detecção da borda de um objeto permite avaliar as suas arestas em busca de imperfeições e variações. Por exemplo, através da borda pode-se verificar a concentricidade de um furo circular.

2.3.5 Reconhecimento

A tarefa de reconhecimento através de imagens via software tem como objetivo aproximar-se da capacidade dos seres humanos (Gonzalez e Woods, 2008). Porém, esta tarefa está entre as mais complexas e os sistemas existentes são altamente especializados, possuindo pouca flexibilidade para executar mais de um tipo de análise. As soluções desenvolvidas são altamente dependentes dos problemas para as quais foram projetadas.

2.3.6 Interpretação

O último passo para se obter informações através do processamento de imagens digitais é a tarefa da interpretação. Segundo Gonzalez e Woods (2008), a interpretação consiste na atribuição de um significado a uma imagem. A intervenção humana nesta etapa é significativa, limitando o escopo das características a serem analisadas em uma imagem com o objetivo de simplificar o problema.

Os passos anteriores do processamento digital de imagens oferecem os dados a serem interpretados – a matriz de pixels que compõem a imagem – e as características obtidas através das ferramentas de processamento – borda, número e área das regiões. O sucesso da interpretação é altamente dependente das etapas anteriores. Durante a aquisição da imagem, é importante que o ambiente seja controlado e apenas objetos para os quais o sistema foi projetado tenham sua imagem capturada. A segmentação separa os objetos de interesse para obtenção das informações desejadas e a interpretação fornece os rótulos para que cada objeto selecionado forneça a informação pertinente.

3 DESENVOLVIMENTO

Neste capítulo, serão apresentados os componentes implementados no programa de inspeção por imagens. Estes componentes são: interface gráfica, programação de aquisição, pré-processamento e tarefas de inspeção. Nas tarefas de inspeção, quatro funções foram implementadas: cálculo da área, cálculo de borda, contagem do número de furos e inspeção dimensional dos furos. Estas funções são freqüentemente encontradas nas aplicações de inspeção de peças utilizando imagem. A figura seguinte ilustra a seqüência funcional do programa de inspeção. No Apêndice A encontra-se o código MATLAB utilizado em cada um dos blocos.

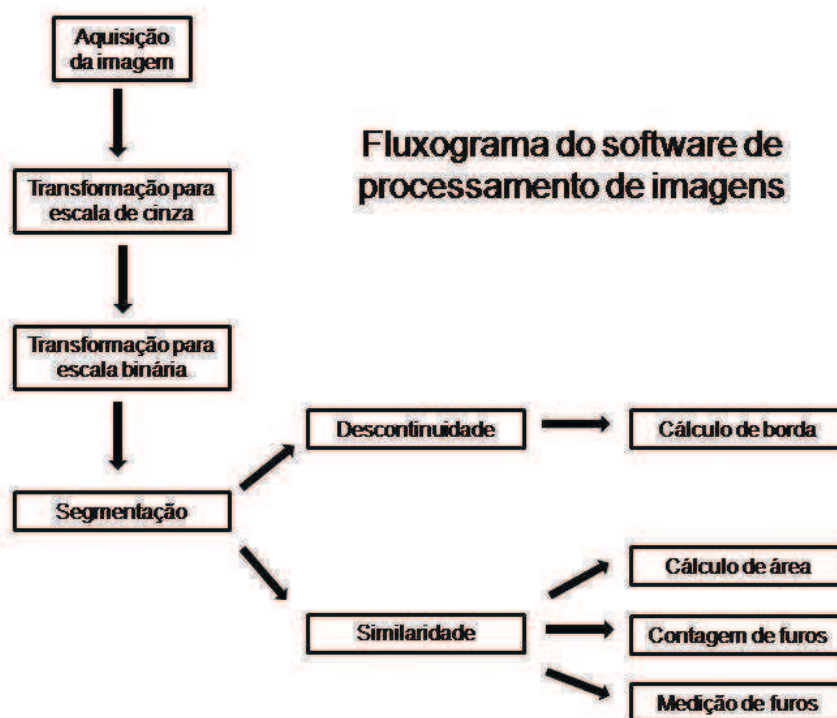


Figura 2 – Fluxograma do *software* de inspeção

Fonte: Autoria própria

3.1 INTERFACE GRÁFICA

Para facilitar a manipulação do programa pelo usuário, foi desenvolvida uma interface homem-máquina (IHM). Nesta interface estão disponíveis botões para realizar as operações de processamento de imagens, campos para exibir a imagem original e a imagem processada e campos com os dados dimensionais obtidos pelos processos. A figura abaixo exibe o *lay-out* da interface gráfica.

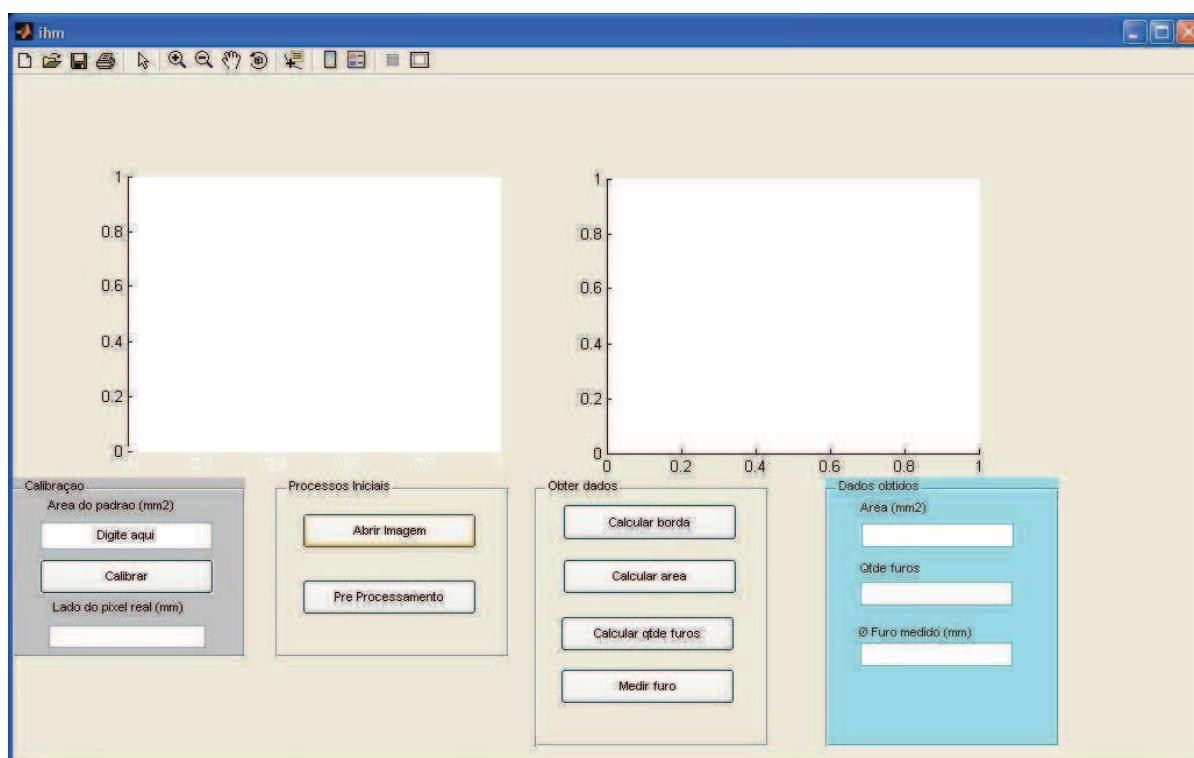


Figura 3 – Tela principal do programa de inspeção
Fonte: Autoria própria

3.2 AQUISIÇÃO

A aquisição da imagem para processamento é realizada através da leitura de um arquivo de imagem previamente gravado no computador. A captura da imagem é feita externamente através do *software* de interface com a câmera digital.

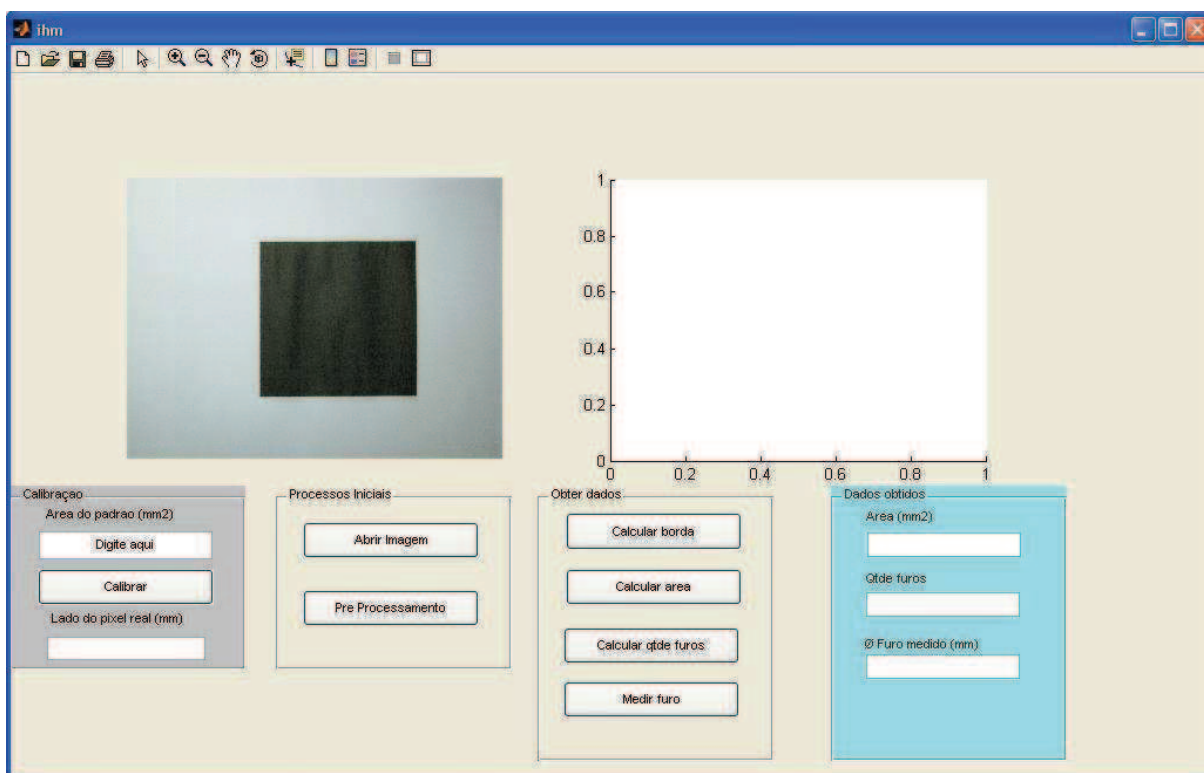


Figura 4 – Imagem carregada no programa
Fonte: Autoria própria

Para uma boa captura é importante que alguns cuidados sejam tomados com respeito à iluminação e ao plano de fundo. Isso é necessário para que o processamento identifique facilmente a imagem, tendo esta um bom contraste em relação ao fundo. A Figura 4 ilustra um exemplo.

No software MATLAB a leitura da imagem da memória é feita pelo comando “imread”. Após a leitura, a mesma é gravada como uma variável para posterior processamento.

3.3 PRÉ-PROCESSAMENTO

Após a aquisição, a imagem recebe uma série de transformações de preparação para a inspeção. Esta série de transformações é chamada de pré-processamento. Inicialmente, a imagem é transformada de colorida para tons de cinza. A etapa seguinte é a binarização, a qual analisa o histograma da imagem e define o limiar ou limite entre o plano de fundo e o objeto de interesse. Normalmente o fundo é mais claro que o objeto, embora isso não seja uma regra. Esse cálculo é

realizado pelo comando "graytresh", o qual realiza uma média entre o valor da escala de cinza superior e inferior, os valores acima dessa média serão adotados como "1" e os valores abaixo como "0". Após a binarização da imagem, faz-se necessário a correção da limiarização utilizando operadores morfológicos. A etapa final de pré-processamento é a calibração da medida do pixel. Considerando a distância entre o objeto e a câmera como um valor fixo, a imagem capturada será sempre da mesma região e portando terá a mesma quantidade de pixels. Assim, podemos obter uma imagem de um objeto padrão, cuja área tem valor conhecido. Em seguida é feita a binarização possibilitando a contagem dos pixels deste objeto e obtenção da relação da dimensão física pela dimensão em pixels. Com o sistema calibrado, podem-se carregar no programa as imagens dos objetos a serem inspecionados. Um exemplo do processo completo é apresentado na Figura 4.

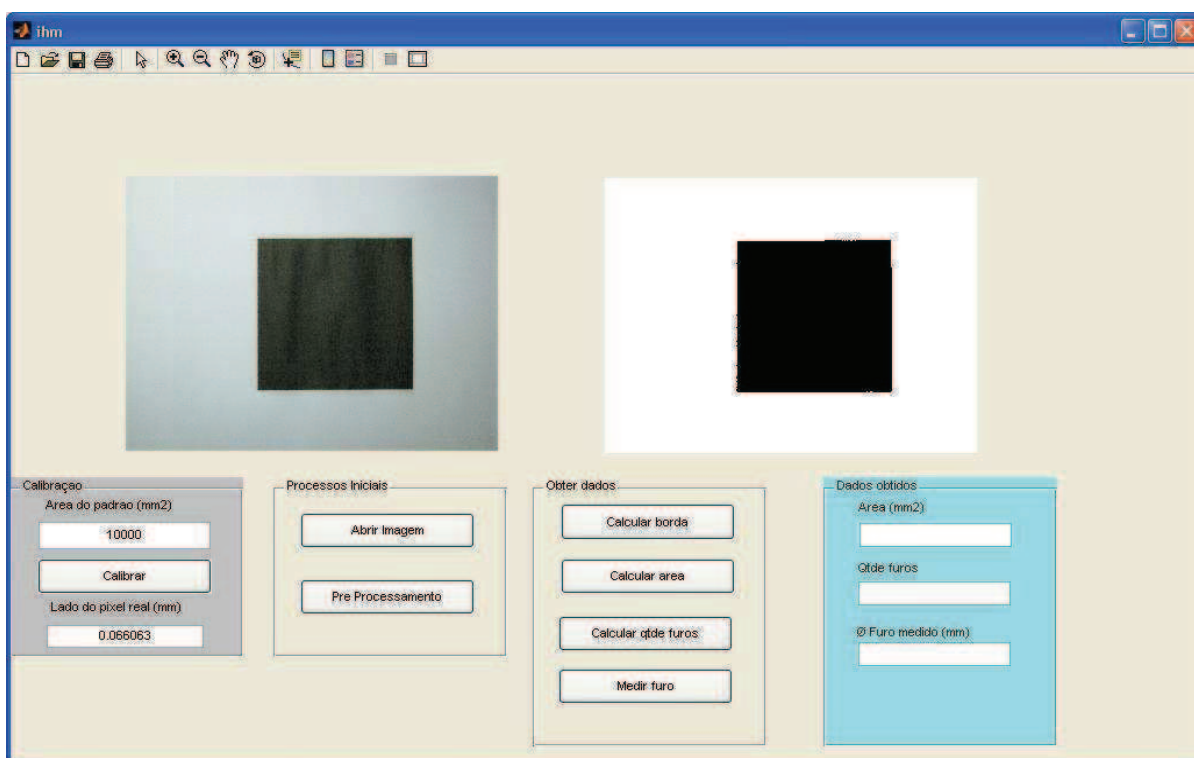


Figura 5 – Imagem pré-processada

Fonte: Autoria própria

3.4 TAREFAS DE INSPEÇÃO

Foram implementadas quatro tarefas de inspeção por imagens: cálculo da área do objeto, cálculo da borda, contagem do número de furos de um objeto e inspeção dimensional de furos.

3.4.1 Cálculo da área

Para o cálculo de área é utilizado o comando "regionsprops". Através desse comando, podemos extrair dados de uma região da imagem como área, em número de pixels e centróide. Após a extração da área do objeto, em pixels, basta realizar a multiplicação pela taxa de milímetros por pixel, obtida na calibração da imagem. O produto dessa operação é a área do objeto medido. Em seguida, esse resultado é exibido na interface do programa. No exemplo abaixo, calcula-se a área de um CD.

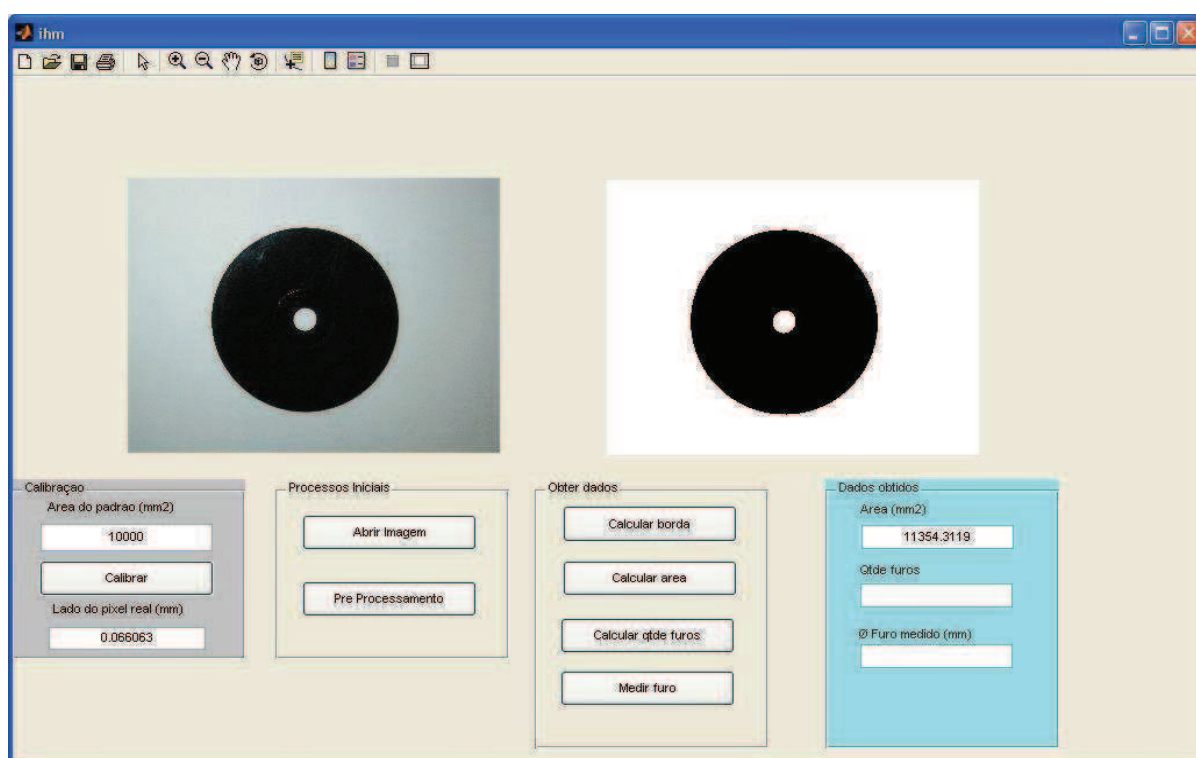


Figura 6– Cálculo da área de um CD
Fonte: Autoria própria

3.4.2 Cálculo de borda

O cálculo de borda consiste na contagem do número de pixels que fazem o contorno do objeto processado. Para isso é necessário extrair da imagem binarizada os pixels que pertencem à parte de dentro do objeto. Nesta tarefa é utilizado o comando "imerode" com a função de fazer o objeto original "aumentar" em questão de contorno em um pixel, gerando uma nova imagem com um acréscimo de um pixel em torno de todo o objeto. Em seguida é realizada a subtração da imagem que foi "aumentada" com a imagem binarizada original, o resultado dessa operação é o

contorno do objeto. Após a realização da operação, o usuário pode utilizar as ferramentas de zoom disponíveis na IHM para avaliar a borda do objeto.

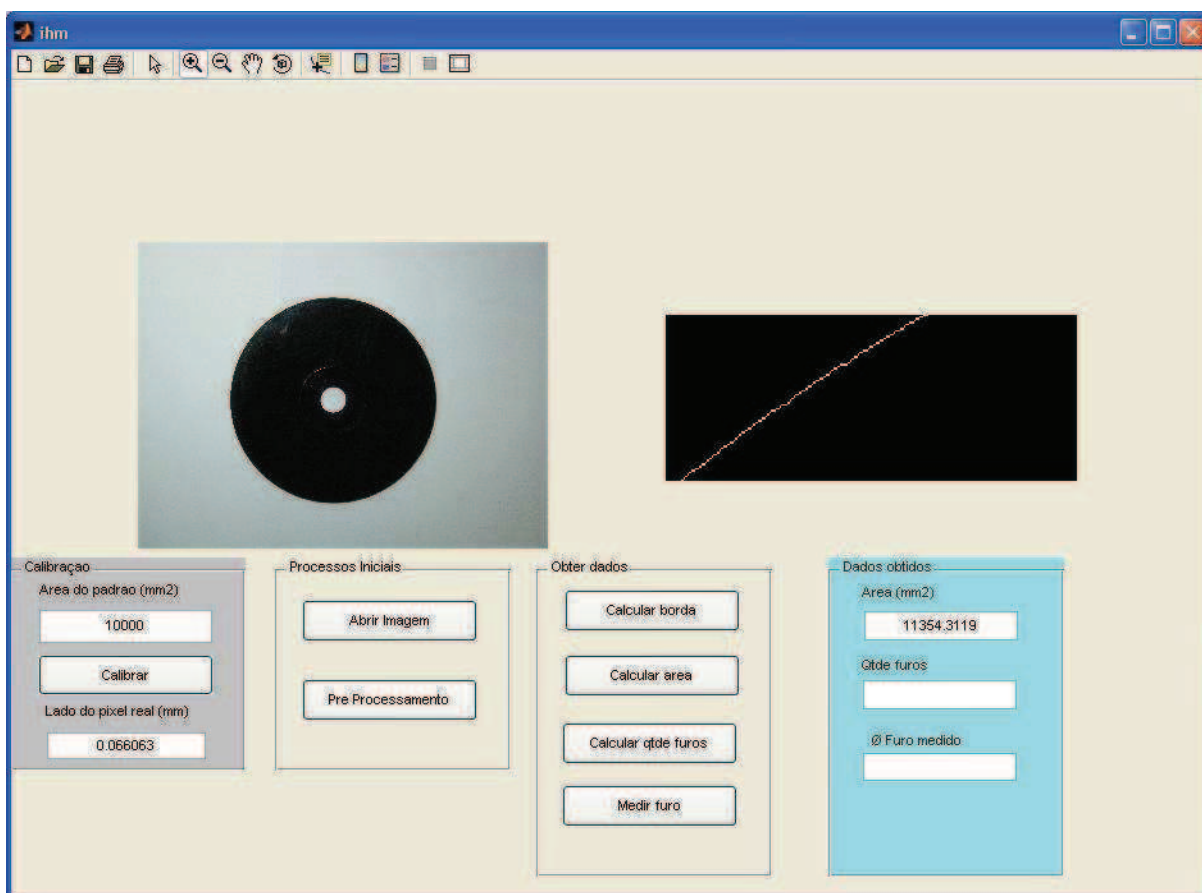


Figura 7 – Cálculo de borda
Fonte: Autoria própria

3.4.3 Número de furos

Para a medição do número de furos também foi utilizado o comando "regionsprops". O comando determina quantas regiões existem dentro da imagem, em seguida é subtraído do número de regiões existentes duas regiões que não interessam para essa utilidade, o plano de fundo e o objeto medido, ou seja, é realizada a subtração do número de regiões em duas unidades. Finalmente, o valor final é exibido na interface gráfica conforme mostrado na imagem a seguir.

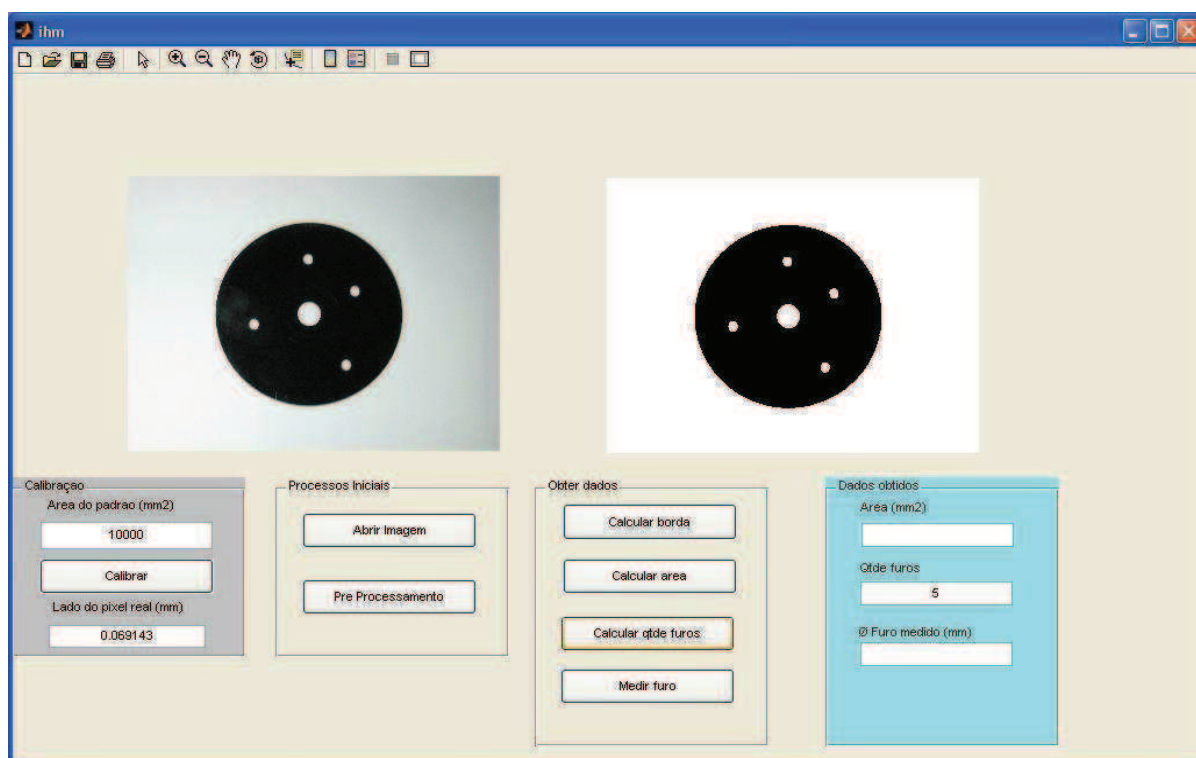


Figura 8– Contagem de furos ou cavidades
Fonte: Autoria própria

3.4.4 Medição de furos

Para a medição dos furos o programa solicita ao usuário que selecione o furo para ser medido como pode ser visto na Figura 9. O número de pixels do furo é obtido por meio do “regionsprops” sendo o mesmo multiplicado pelo valor em milímetros do pixel obtido na calibração e esse valor de área é utilizado então para o cálculo do diâmetro. O resultado pode ser visto na Figura 10.

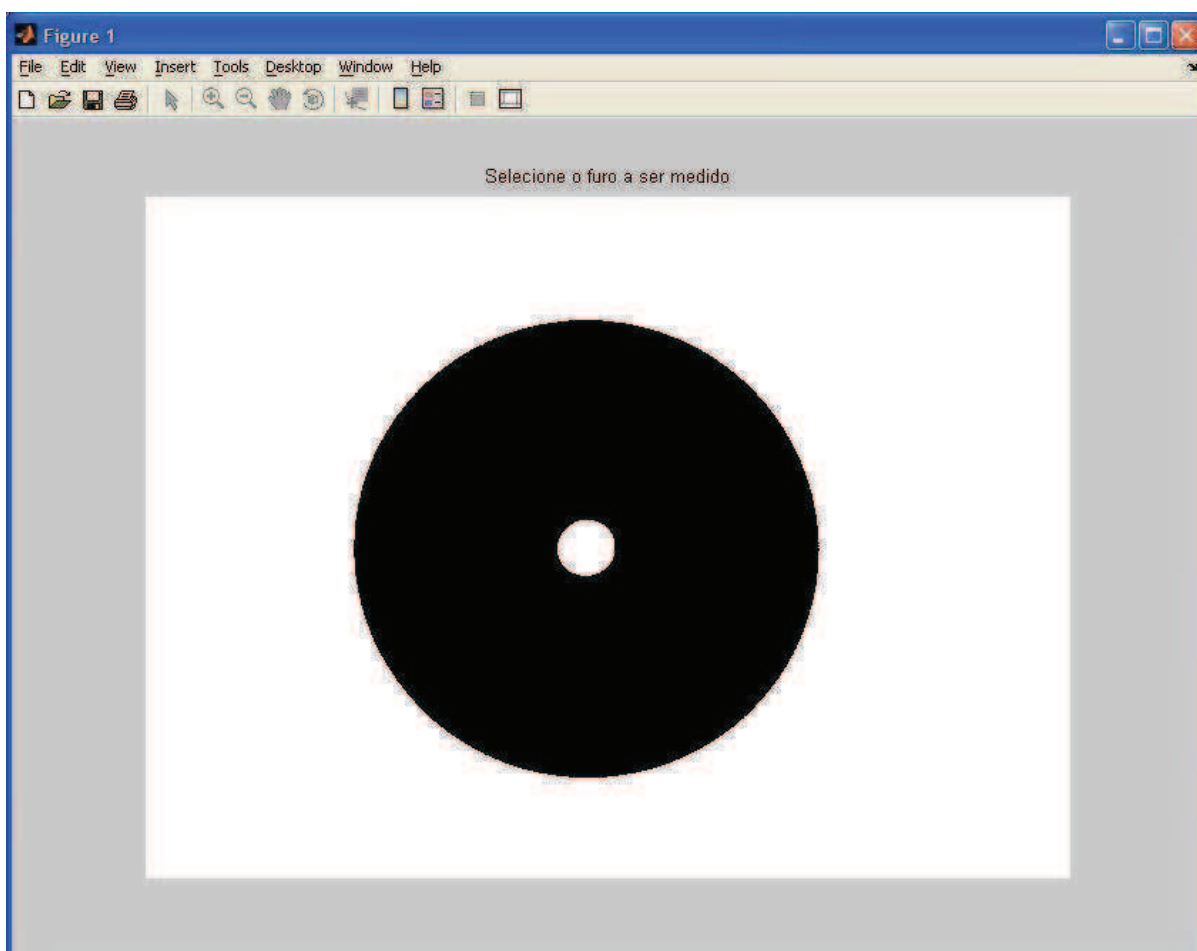
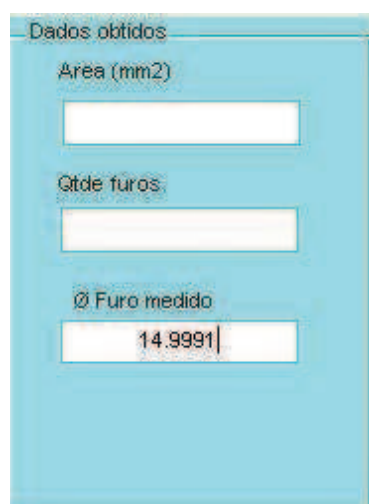


Figura 9– Tela para seleção do furo para ser medido
Fonte: Autoria própria



Dados obtidos	
Area (mm2)	<input type="text"/>
Qtde furos	<input type="text"/>
Ø Furo medido	<input type="text" value="14.9991"/>

Figura 10 – Exibição do diâmetro do furo medido
Fonte: Autoria própria

4 CONCLUSÕES

O uso de sistemas de visão para inspeção industrial tem se tornado cada vez mais comum. Hoje, câmeras, softwares e hardwares de processamento são quase tão acessíveis quanto controladores lógicos programáveis e motores. Dentro dos tipos de softwares e sistemas encontrados no mercado, tem-se desde os mais simples, e naturalmente restritos, até os mais complexos, que permitem uma variedade maior de possibilidades e personalizações. O domínio sobre os sistemas de visão, seu funcionamento, configuração e programação, é importante para os tecnólogos em mecatrônica. Por este motivo foi realizado este projeto.

O *software* MATLAB oferece uma série de bibliotecas para desenvolvimento de algoritmos. Nele, estão disponíveis ferramentas e instruções que auxiliam a implementação das tarefas básicas de processamento de imagens como o carregamento e importação de imagens, operações como a transformação para a escala de cinza e binarização.

Desta forma, optou-se em desenvolver os algoritmos necessários para o software de inspeção em plataforma MATLAB. Em seguida foi desenvolvida uma interface homem-máquina, utilizando o assistente de desenvolvimento de interfaces gráficas do próprio MATLAB, a qual permitiu a integração das tarefas de processamento de imagens e da inspeção do objeto em si.

Apesar da idéia inicial do software ser para a aplicação no sistema de FMS da UTFPR, o software desenvolvido pode ser usado em outras aplicações. Futuramente, de acordo com a necessidade e tipos de máquina que forem acrescentadas ao FMS, novas rotinas de inspeção por imagem podem ser agregadas ao programa, como por exemplo, diferenciação de peças por cor, reconhecimento de padrões e caracteres. O programa também pode ser integrado ao sistema de imagem do LabVIEW, *software* da National Instruments disponível no FMS, pois esse é compatível com código do MATLAB.

Em resumo, pode-se concluir que a inspeção por imagens é uma realidade. Sua popularização se deve à grande variedade tanto de *hardware*, com a popularização das câmeras e sistemas de processamento, quanto de *software*, com diversos aplicativos disponíveis. Contudo, o conhecimento nesta área ainda precisa avançar, tanto na popularização do ensino quanto nas pesquisas na área.

REFERÊNCIAS

EDDINS, Steven L.; GONZALEZ, Rafael C.; WOODS, Richard E. **Digital Image Processing Using MATLAB®**. 2ª edição. Gatesmark Publishing, 2008.

FERREIRA, Mauricio; MORAES, Aurélio. **Tutorial OpenCV**. Disponível em: <<http://www.tecgraf.puc-rio.br/~malf/opencv/>>. Acesso em: 19 nov. 2010.

GONZALEZ, Rafael C.; WOODS, Richard E. **Digital Image Processing**. 3ª edição. New Jersey: Prentice Hall, 2008.

GROOVER, Mikell P. **Automation, Production Systems, and Computer-Integrated Manufacturing**. 3 edição. Pearson-Prentice Hall, 2008.

GROOVER, Mikell P. **Fundamentals of Modern Manufacturing: materials, processes and systems**. 2ª edição. [S.I.]: John Wiley & Sons Inc, 2002.

PEDRINO, Emerson C. INTRODUÇÃO À VISÃO DE MÁQUINAS. **Mecatrônica Atual**. Editora Saber. Disponível em: <<http://www.mecatronicaatual.com.br/secoes/leitura/135>>. Acesso em: 19 nov. 2010.

National Instruments. **Embedded Vision System**. Disponível em: <<http://sine.ni.com/nips/cds/view/p/lang/en/nid/207111>>. Acesso em: 19 nov. 2010.

National Instruments. **NI Machine Vision Software**. Disponível em: <<http://www.ni.com/vision/software/>>. Acesso em: 19 nov. 2010.

National Instruments. **NI Smart Cameras**. Disponível em: <<http://sine.ni.com/nips/cds/view/p/lang/en/nid/204077>>. Acesso em: 19 nov. 2010.

Point Grey Research. **Flea®3 (IEEE 1394b) CCD Camera**. Disponível em: <<http://www.flea3.com/>>. Acesso em: 06 abr. 2011.

SICK AG. **Vision Sensors Inspector**. Disponível em: <http://www.sick-automation.ru/images/File/pdf/DIV06/Inspector_I20UV_I40.pdf>. Acesso em: 19 nov. 2010.

The MathWorks, Inc. **Arquivos de ajuda MATLAB 7.0**. Maio, 2004.

APÊNDICE A – CÓDIGO MATLAB POR BLOCO FUNCIONAL

Neste apêndice encontra-se o código MATLAB dos blocos funcionais conforme a Figura 2.

a) Aquisição da imagem

```
% =====BOTAO ABRIR IMAGEM=====
% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% mostra uma tela para que o usuário escolha um arquivo de imagem:
[filename,pathname] = uigetfile({'*.jpg','*.jpeg'},'Selecione a imagem');

% guarda o nome do arquivo
handles.arquivo = filename;

% abre a imagem
handles.imagem = imread(filename);

% seleciona o componente axes1
axes(handles.axes1);

% exibe a figura
imshow(handles.imagem);

% salva a variável handles
guidata(hObject, handles);
```

b) Transformação para escala de cinza

```
%passar para escala de cinza
IG = double(.2989*ioriginal(:, :, 1)+.5870*ioriginal(:, :, 2)+.1140*ioriginal(:, :, 3));
```

```
%Calcular amplitudes extremas
if((max(IG)>255)|(min(IG)<0))
    disp(sprintf('\n Amplitude com saturação !'));
end
```

```
%mostra a imagem em Grayscale
%figure, imshow(IG,[]), title('Imagem em Grayscale');
```

c) Transformação para escala binária

```
%binarização da imagem grayscale
TIG = graythresh(IG/max(IG(:)));
```

```
%BW = ~im2bw(IG/max(IG(:)),TIG);
BW = im2bw(IG/max(IG(:)),TIG);
```

```
%mostra a imagem em binário
%figure, imshow(BW,[]), title('Imagem em Binário');
```

```
%correção da limiarização usando operadores morfológicos
MM=[0 0 1 1 1 0 0 ; 0 ones(1,5) 0; ones(3,7); 0 ones(1,5) 0; 0 0 1 1 1 0 0 ];
%MM equivale a se3 = strel('disk',4)% disk, radius 4
%Ajustando os parametros de MM, modifica-se a intensidade da correção
```

```
BW1=imclose(BW,MM);
%figure, imshow(BW1,[]), title ('Imagem pré processada');
BWL=imopen(BW1,MM);
```

d) Cálculo de borda

```
function [borda] = calcularborda(BWL)
```

```

%Calculo das bordas binárias
%Processa a imagem de forma a obter a borda do objeto
%Variáveis:
% "BWL": imagem em binário com correção da limiarização
% "BWL1": variável auxiliar para calcular a borda
% "borda": imagem contendo a borda do objeto

%calculo das bordas binárias (vide slides)
BWL1=imerode(BWL,[0 1 0; 1 1 1; 0 1 0]);
borda=BWL-BWL1;
% figure, imshow(borda,[]), title('Detecção de borda');

```

e) Cálculo de área

```

% =====BOTAO CALCULAR AREA=====
% --- Executes on button press in calcular_area.
function calcular_area_Callback(hObject, eventdata, handles)
% hObject    handle to calcular_area (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

%Exibe a imagem para o usuário selecionar a área
imageminvertida=~handles.imagemproc;
figure, imshow(imageminvertida,[]), title ('Selecione a area para medir');

selecaoarea = bwselect;
figure, imshow(selecaoarea,[]), title ('Area selecionada');

% Obtém as regiões da imagem pré-processada
[B,L] = bwboundaries(selecaoarea);

% Obtém as informações das regiões
stats = regionprops(L,'Area','Centroid');

```

```

% calculo da área real
areaimagem = stats(1).Area;
areareal = areaimagem * handles.areapixelreal;

% Copia a área padrão da imagem para handles
handles.areareal = areareal;

% transforma o valor da área em string para exibir no campo
areareal_string = num2str(handles.areareal);
set(handles.edit6,'String',areareal_string);

% salva a variável handles
guidata(hObject, handles);

```

f) Contagem de furos

```

% =====BOTAO CALCULAR QTDE FUROS=====
% --- Executes on button press in contar_furos.
function contar_furos_Callback(hObject, eventdata, handles)
% hObject    handle to contar_furos (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Obtém as regiões da imagem pré-processada
[B,L] = bwboundaries(handles.imagemproc);

% Obtém as informações das regiões
stats = regionprops(L,'Area','Centroid');

% O numero de furos é a quantidade de regiões – (peça + fundo)
handles.qtdefuros = length(B) - 2;

% transforma o valor em string para exibir no campo

```

```
qtdefuros_string = num2str(handles.qtdefuros);
set(handles.exibir_qtde_furos,'String',qtdefuros_string);
```

```
% salva a variável handles
guidata(hObject, handles);
```

g) Medição de furos

```
function [dfuro] = medirfuro(imagem , areapixelreal)
%Programa para medir 1 furo
% Entrada: imagem, valor da área do pixel real
% saída: diâmetro do furo real

% Exibe a imagem
%imagem=~imagem; comando que inverte a imagem
figure, imshow(imagem,[]), title ('Selecione o furo a ser medido');

% O usuário seleciona a região a ser medida
furo=bwselect;
figure, imshow(furo,[]), title ('furo selecionado');

% Obter informações da região
[B,L] = bwboundaries(furo);
stats = regionprops(L,'Area','Centroid');

% Calcular diâmetro
areafuro = stats(1).Area*areapixelreal;
raiofuro = sqrt((areafuro/pi))
dfuro = raiofuro*2;
```