



UNIVERSIDADE ESTADUAL PAULISTA “JÚLIO DE MESQUITA FILHO”

Faculdade de Engenharia de Ilha Solteira

ARTHUR SUZINI POLETO

**DESENVOLVIMENTO DE UM SISTEMA DE VISÃO DE MÁQUINA PARA
INSPEÇÃO DE CONFORMIDADE EM UM PRODUTO INDUSTRIAL**

Ilha Solteira

2019



UNIVERSIDADE ESTADUAL PAULISTA “JÚLIO DE MESQUITA FILHO”

Faculdade de Engenharia de Ilha Solteira

ARTHUR SUZINI POLETO

**DESENVOLVIMENTO DE UM SISTEMA DE VISÃO DE MÁQUINA PARA
INSPEÇÃO DE CONFORMIDADE EM UM PRODUTO INDUSTRIAL**

Dissertação apresentada à Faculdade de Engenharia de Ilha Solteira – UNESP – para obtenção do título de Mestre em Engenharia Mecânica.

Prof. Dr. João Antonio Pereira
Orientador

Ilha Solteira

2019

FICHA CATALOGRÁFICA

Desenvolvido pelo Serviço Técnico de Biblioteca e Documentação

P765d **Poleto, Arthur Suzini.**
Desenvolvimento de um sistema de visão de máquina para inspeção de conformidade em um produto industrial / Arthur Suzini Poleto. -- Ilha Solteira: [s.n.], 2019
123 f. : il.

Dissertação (mestrado) - Universidade Estadual Paulista. Faculdade de Engenharia de Ilha Solteira. Área de conhecimento: Mecânica dos Sólidos, 2019

Orientador: João Antonio Pereira
Inclui bibliografia

1. Visão de máquina. 2. Imagem digital. 3. Processamento de imagens digitais. 4. Invariantes de momento. 5. Descritores de forma. 6. Reconhecimento ótico de caracteres.

Rafane da Silva Santos
Rafane da Silva Santos


CERTIFICADO DE APROVAÇÃO

TÍTULO DA DISSERTAÇÃO: DESENVOLVIMENTO DE UM SISTEMA DE VISÃO DE MÁQUINA PARA INSPEÇÃO DE CONFORMIDADE EM UM PRODUTO INDUSTRIAL

AUTOR: ARTHUR SUZINI POLETO

ORIENTADOR: JOAO ANTONIO PEREIRA

Aprovado como parte das exigências para obtenção do Título de Mestre em ENGENHARIA MECÂNICA, área: Mecânica dos Sólidos pela Comissão Examinadora:


Prof. Dr. JOAO ANTONIO PEREIRA
Departamento de Engenharia Mecânica / Faculdade de Engenharia de Ilha Solteira


Prof. Dr. AMARILDO TABONE PASCHOALINI
Departamento de Engenharia Mecânica / Faculdade de Engenharia de Ilha Solteira


Prof. Dr. ADAILTON SILVA BORGES
Departamento de Engenharia Mecânica / Universidade Tecnológica Federal do Paraná

Ilha Solteira, 23 de agosto de 2019

*Para meus filhos,
Isabella Carvalho Poletto e Miguel de Souza Poletto.*

AGRADECIMENTOS

Agradecemos a Deus pela família e pelos amigos, pelo conforto de reconhecer que a nossa felicidade tem o tamanho da felicidade que fizermos para os outros. A meus pais, Arnaldo e Suely, pelo amor, pela educação e exemplo que recebemos durante toda a vida. Somos especialmente gratos ao Professor João Antonio Pereira, pela amizade construída ao longo deste trabalho, pelas orientações criteriosas e sensatas, e por todo o conhecimento compartilhado. Ao Professor Amarildo, pelas palavras de apoio e incentivo, pela contribuição nas bancas examinadoras do Projeto, de Estudos Especiais, e de Defesa. Ao Professor Márcio, pelas palavras de apoio e incentivo, pela contribuição nas bancas examinadoras de Estudos Especiais e do Exame Geral de Qualificação. Ao Professor Douglas, pelas palavras de apoio e incentivo, pela contribuição nas bancas examinadoras do Projeto e do Exame Geral de Qualificação. Ao Professor Adailton, pela contribuição na banca examinadora da Defesa. Somos profundamente gratos a muitas pessoas da Faculdade de Engenharia de Ilha Solteira, dentre professores e funcionários, que colaboraram durante nosso estudo desde a graduação. Ao meu irmão Arnaldo, pela companhia, pelo conhecimento compartilhado. A minha esposa Thaissey pela paciência durante a realização deste trabalho e dedicação com que cuida de nossa família. Agradecemos aos amigos Beto, Cidão, Eduardo, Jacaré, Luciano, e Saldela, que apesar da distância e do tempo, continuamos com nossa amizade. Agradecemos aos amigos de Ilha Solteira, Fábio Kroll de Lima e Fábio Torsoni Ribeiro. Agrademos também ao amigo Marcio Antonio Caldeira, da Flash Cover, por todo o suporte e por todos os materiais disponibilizados para este trabalho.

RESUMO

Visão de máquina é um campo multidisciplinar que vem crescendo na indústria, que está cada vez mais preocupada em reduzir custos, automatizar processos, e atender requisitos de qualidade do produto para atender seus clientes. Processos de montagem realizados de forma manual com inspeção e controle visual são tipicamente processos susceptíveis a erros, à utilização de peças não conformes na montagem do produto final. Este trabalho apresenta uma proposta de desenvolvimento de um sistema de visão de máquina com base no processamento e análise de imagens digitais para a inspeção das características e especificações das peças e componentes utilizados na montagem de capotas marítimas, objetivando verificar e garantir a conformidade do produto final. A inspeção e avaliação da conformidade do produto são feitas por etapas com a utilização de duas câmeras, uma captura a imagem do código de identificação alfanumérico do produto e a outra inspeciona o conjunto de elementos de fixação. As imagens passam por um processo de tratamento que envolve a filtragem espacial utilizando máscara de médias para suavização, alargamento de contraste para expandir a faixa de intensidades e segmentação para formação dos objetos de interesse. Uma função de OCR é utilizada para a extração de caracteres e reconhecimento do código do produto e a extração de características específicas do conjunto de componentes de fixação é feita por descritores de forma representados pelos invariantes de momento. As características específicas dos componentes de fixação são usadas para avaliar a conformidade do produto com o seu respectivo código. A apresentação dos dados e resultados da proposta implementada é feita através de uma interface gráfica que informa os parâmetros e dados de inspeção do produto, emitindo sinal de alerta de produtos não conformes. Um protótipo da proposta foi implementado em uma linha de montagem para testes de avaliação e validação da metodologia proposta e os resultados obtidos nos testes preliminares mostraram-se robustos, com verificação corretas dos produtos conformes e dos produtos não conformes.

Palavras-chave: Visão de máquina. Imagem digital. Processamento de imagens digitais. Momentos Descritores. Descritores de forma. Reconhecimento ótico de caracteres.

ABSTRACT

Machine vision is a growing multidisciplinary field in the industry that is increasingly concerned with reducing costs, automating processes, and meeting product quality requirements to serve its customers. Manual assembly processes with inspection and visual control are typically error-prone processes using non-conforming parts in the final product assembly. This work presents a proposal for the development of a machine vision system based on digital image processing and analysis for the inspection of the characteristics and specifications of the parts and components used in the assembly of marine bonnets, aiming to verify and ensure the conformity of the final product. Inspection and conformity assessment of the product are done in stages using two cameras, one capturing the image of the alphanumeric identification code of the product and the other inspecting the set of fasteners. The images undergo a treatment process that involves spatial filtering using averaging masks for smoothing, contrast widening to expand the range of intensities, and segmentation to form the objects of interest. An OCR function is used for character extraction and product code recognition, and the extraction of specific features of the fastener assembly is done by shape descriptors represented by the moment invariants. The specific characteristics of the fasteners are used to assess the conformity of the product with its respective code. The presentation of data and results of the implemented proposal is made through a graphical interface that informs the parameters and product inspection data, sending warning signal of nonconforming products. A prototype of the proposal was implemented in production line for evaluation and validation tests of the proposed methodology and the results obtained in the preliminary tests were robust, with correct verification of the conforming and nonconforming products.

Keywords: Machine vision. Digital Image. Digital Image Processing. Descriptor Moment. Shape descriptors. Optical Character Recognition.

LISTA DE FIGURAS

Figura 2.1	Sensor de aquisição de imagem	22
Figura 2.2	Processo de formação de uma imagem digital	23
Figura 2.3	Discretização e quantização de uma imagem digital	24
Figura 2.4	Imagem de 8 bits 4x4 pixels em escala de cinza	25
Figura 2.5	Imagem de 24 bits e seus componentes RGB	26
Figura 2.6	Matriz tridimensional de uma imagem de 24 bits	26
Figura 2.7	Imagem de 8 bits de cores indexadas	27
Figura 2.8	Formação de imagem industrial obtida por raios X	28
Figura 2.9	Imagem no espaço de cores RGB, à esquerda, e correspondente imagem em escala de cinza, à direita, obtida após conversão	30
Figura 2.10	Coefficientes da máscara 3x3. A origem do <i>kernel</i> se move juntamente com sua vizinhança para gerar uma imagem de saída	31
Figura 2.11	<i>Layout</i> de uma máscara 3x3 para média aritmética simples	33
Figura 2.12	Função de alargamento de contraste	34
Figura 2.13	Alargamento de contraste por meio de funções de transformação lineares	35
Figura 2.14	Função linear de alargamento de contraste com parte dos níveis de intensidade saturados	35
Figura 2.15	Imagem de baixo contraste à esquerda e correspondente imagem de alto contraste à direita	36
Figura 2.16	Imagens de baixo e alto contraste e correspondentes histogramas	37
Figura 2.17	Funções de distribuição acumulada sobrepostas aos histogramas da Figura 2.16	38
Figura 2.18	Ordenação das intensidades das imagens de entrada e de saída, à esquerda, e função de transformação de intensidade para alarga- mento de contraste, à direita	39

Figura 2.19	Função transformação de intensidade para produção de uma imagem binária	40
Figura 2.20	Imagem em escala de cinza, à esquerda, e correspondente imagem binária	40
Figura 2.21	Imagem binária de objetos segmentados	41
Figura 2.22	Perfil de uma região em imagem de objetos em escala de cinza	43
Figura 2.23	Visualização 3D da região destacada na Figura 2.22	44
Figura 2.24	Visualização das bordas detectadas pelas derivadas de primeira e segunda ordens	44
Figura 2.25	Máscara de filtragem representando o Laplaciano discreto de duas variáveis	46
Figura 2.26	Máscara de filtragem representando o Laplaciano com diagonais incorporadas	46
Figura 2.27	Imagem de entrada, acima, e a saída com as bordas detectadas, abaixo ..	47
Figura 2.28	Elipses sobrepostas aos seus objetos	52
Figura 2.29	Representação de objeto com descritores de orientação, eixos inerciais, e elipse	52
Figura 2.30	Normalização do caractere segmentado para a resolução do <i>template</i> , 24x48 pixels	53
Figura 2.31	<i>Template</i> binário de 24x48 pixels	53
Figura 3.1	Modelo esquemático do sistema de inspeção	57
Figura 3.2	Visão global das etapas do sistema de inspeção	58
Figura 3.3	Imagens dos códigos dos produtos inspecionados	58
Figura 3.4	Imagens dos conjuntos de peças dos produtos inspecionados	59
Figura 3.5	Bancada para experimentação e desenvolvimento do sistema de inspeção	60
Figura 3.6	Sequência de imagens mostrando a reflexão da luz	60

Figura 3.7	Fluxograma das etapas de processamento das imagens do código	61
Figura 3.8	Imagem do código impresso em tecido e destaque de uma das letras	62
Figura 3.9	Filtro de média utilizado para suavizar a imagem de um código	63
Figura 3.10	Intensidades dos pixels da letra “A” antes e após filtragem	63
Figura 3.11	Intensidade dos pixels da letra “A”. Imagem suavizada, à esquerda, e imagem de alto contraste, à direita	64
Figura 3.12	Intensidades das imagens suavizada e de alto contraste, à esquerda, e a função de transformação de intensidade para alargamento de contraste, à direita	65
Figura 3.13	Funções de distribuição acumulada sobrepostas aos histogramas das imagens da letra “A”	66
Figura 3.14	Imagem de entrada seguida pela imagem suavizada e pela imagem com alto contraste	67
Figura 3.15	Diferentes limiares de binarização para a imagem suavizada e para a imagem realçada	68
Figura 3.16	Interface para validação do reconhecimento dos códigos utilizados	70
Figura 3.17	Fluxograma das etapas de processamento das imagens dos compo- nentes	72
Figura 3.18	Peças de aço estampadas usadas no experimento	73
Figura 3.19	Eixos inerciais e correspondentes elipses das peças testadas	74
Figura 3.20	Sequência de <i>frames</i> do componente de fixação com perfil largo	74
Figura 3.21	Sequência de <i>frames</i> do componente de fixação com perfil delgado	76
Figura 3.22	Fluxograma de ramificação para decisão de alerta de conformidade	78
Figura 3.23	Interface simplificada para teste em bancada	80
Figura 4.1	Estrutura do dispositivo de inspeção	81
Figura 4.2	Componentes do dispositivo de inspeção	82
Figura 4.3	Detalhes da placa de barreira de luz	83

Figura 4.4	Controle de menu <i>popup</i> para escolha da câmera	85
Figura 4.5	Imagem <i>live</i> das câmeras em painéis disponibilizados na interface	86
Figura 4.6	Extração da Região de Interesse da imagem original	87
Figura 4.7	Controles para segmentação disponíveis na interface	88
Figura 4.8	Resultados do reconhecimento do código e dos elementos de fixação	89
Figura 4.9	Interface do sistema de inspeção com identificação do modelo	90
Figura 4.10	Interface do sistema de inspeção com os respectivos resultados	91
Figura 4.11	Conjuntos de elementos de fixação	92
Figura 4.12	Diferentes eventos possíveis durante a inspeção dos produtos	93
Figura 4.13	Painel de apresentação da <i>string</i> de reconhecimento. Código insatisfatório	94
Figura 4.14	Controles dos parâmetros de processamento. Imagem do código	95
Figura 4.15	Controle dos parâmetros de processamento. Imagem dos elementos de fixação	96
Figura 4.16	Segmentação incorreta indicando a necessidade de calibração	96
Figura 4.17	Segmentação correta após ajustes nos parâmetros de processamento	97
Figura 4.18	Interface do sistema apontando produto conforme	99
Figura 4.19	Interface do sistema apontando produto não conforme	100
Figura A.1	Criando um objeto de vídeo	107
Figura A.2	Exemplo de pré-processamento	107
Figura A.3	Chamando a função OCR e armazenando os resultados em uma variável	108
Figura A.4	Extraindo características dos objetos	108
Figura A.5	Função para o cálculo e visualização dos descritores	109
Figura B.1	Interface do aplicativo para calcular ângulo de objetos	110
Figura B.2	Etapas para o cálculo dos ângulos	111

Figura B.3	Borda detectada de um traço em ângulo	112
Figura B.4	Retas em ângulo obtidas por regressão linear	113
Figura B.5	Código para computar ângulos em objetos binários	114
Figura B.6	Código para a detecção da borda de um objeto	115
Figura B.7	Código para alteração na direção de detecção d borda	116
Figura B.8	Código para detecção do ponto inicial da borda	116
Figura C.1	Desenho do dispositivo desenvolvido	117

SUMÁRIO

1	INTRODUÇÃO	15
1.1	OBJETIVO	20
1.2	TÓPICOS DO TRABALHO	20
2	CONCEITOS E FUNDAMENTOS BÁSICOS	21
2.1	SENSORES DE VISÃO E TIPOS DE IMAGENS DIGITAIS	21
2.1.1	Amostragem e quantização	23
2.1.2	Imagens de 8 bits em escala de cinza	24
2.1.3	Imagens coloridas de 24 bits	25
2.1.4	Imagens coloridas de 8 bits	27
2.1.5	Fontes de energia e outras formações de imagem	27
2.2	TRANSFORMAÇÕES EM SISTEMAS DE CORES	28
2.2.1	Conversão entre os espaços RGB e YIQ	29
2.2.2	Filtragem espacial	30
<i>2.2.2.1</i>	<i>Filtros lineares de suavização</i>	32
2.2.3	Alargamento de contraste	33
2.2.4	Histograma	36
2.2.5	Limiarização	39
2.3	ATRIBUTOS DE IMAGENS	41
2.3.1	Detecção de borda	42
2.3.2	Detecção de borda por filtragem espacial	44
2.3.3	Características Geométricas e Propriedades dos Objetos	47
<i>2.3.3.1</i>	<i>Momentos descritores</i>	48
<i>2.3.3.2</i>	<i>Área</i>	49
<i>2.3.3.3</i>	<i>Centroide</i>	49
<i>2.3.3.4</i>	<i>Elipse</i>	50
<i>2.3.3.5</i>	<i>Orientação</i>	51
2.3.4	Reconhecimento de caracteres	53

3	DESCRITORES DE FORMA E RECONHECIMENTO ÓTICO DE CARACTERES APLICADOS EM UM SISTEMA DE VISÃO DE MÁQUINA	55
3.1	INSPEÇÃO DE CONFORMIDADE	55
3.2	LAYOUT DO SISTEMA DE INSPEÇÃO	56
3.3	BANCADA DE TESTE UTILIZADA NA IMPLEMENTAÇÃO COMPUTACIONAL DA METODOLOGIA DE IDENTIFICAÇÃO DO SISTEMA DE INSPEÇÃO	59
3.4	IDENTIFICAÇÃO DO CÓDIGO DO PRODUTO	61
3.4.1	Pré-processamento com filtro de média e alargamento de contraste ...	61
3.4.2	Segmentação e reconhecimento ótico dos caracteres	68
3.4.3	Teste de reconhecimento do código usando OCR	69
3.5	IDENTIFICAÇÃO DAS PEÇAS UTILIZADAS NA MONTAGEM DO PRODUTO	71
3.5.1	Pré-processamento das imagens das peças	72
3.5.2	Extração dos momentos descritores	73
3.6	VERIFICAÇÃO E IDENTIFICAÇÃO DA CONFORMIDADE	77
3.6.1	Interface gráfica	78
4	PROTÓTIPO DO DISPOSITIVO DE VISÃO DE MÁQUINA PARA INSPEÇÃO DE CONFORMIDADE DO PRODUTO	81
4.1	SETUP EXPERIMENTAL	81
4.2	INTERFACE DE CONTROLE E GERENCIAMENTO DO SISTEMA DE VISÃO COMPUTACIONAL	84
4.3	PROCESSO DE CALIBRAÇÃO	92
4.4	TESTES E AVALIAÇÃO DO DISPOSITIVO DE VISÃO COMPUTACIONAL INTEGRADO NA LINHA DE PRODUÇÃO	92
4.4.1	Calibração e definição dos parâmetros do sistema de inspeção integrado na linha de montagem	94
4.4.2	Teste de validação - conformidade e não conformidade	97
5	CONCLUSÕES.....	101
5.1	SUGESTÕES PARA TRABALHOS FUTUROS	102

REFERÊNCIAS	103
APÊNDICE A – Sinopse da linguagem de programação empregada	107
APÊNDICE B – Aplicativo para calcular ângulo de objetos	110
APÊNDICE C – Desenho do dispositivo metálico	117
APÊNDICE D – Invariantes de momento	118

1 INTRODUÇÃO

Todo o ambiente ao redor das pessoas, seja a natureza ou aquilo que foi construído pelo homem, pode ser notado, visto, “sentido” e compreendido, através de transdutores/sensores e a posterior análise das informações obtidas pelos mesmos. O grande estímulo que move este projeto é a possibilidade de sensoriamento do ambiente onde vivemos, e partir da análise e processamento dos dados obtidos, identificar, diagnosticar e caracterizar esse ambiente. Ao esquadrihar um sistema na busca de sua identidade, sua “biometria”, é possível entender e descrever seu comportamento e, conseqüentemente, extrair informações que permitem aos engenheiros desenvolver novas ferramentas e melhorias que possam contribuir para o progresso e o bem-estar das pessoas.

Dentre os diversos tipos de sensores existentes destacam-se os sensores óticos presentes nos dispositivos portáteis que registram imagens – as câmeras. A pesquisa que será realizada é fundamentada nestes sensores, que dependem de um efeito fotoelétrico para criar um SINAL elétrico a partir da luz. Fica claro, portanto, que o estudo será embasado em sensores que realizam captura de imagem digitalmente. De fato, são sensores singulares e muito importantes para as pessoas, os responsáveis por “enxergar” o mundo à nossa volta.

O processamento de imagens digitais e a visão computacional desenvolvem uma ciência que permite a extração de características de determinada cena e o reconhecimento de padrões visuais para a aplicação em sistemas específicos. O processamento destas imagens capturadas digitalmente é amplamente utilizado na medicina, astronomia, indústria e em diversas outras áreas específicas. Atualmente, não existe praticamente área de empreendimento técnico que não seja impactada de uma forma ou de outra pelo processamento digital de imagens (GONZALEZ; WOODS, 2009). A análise e classificação de imagens binárias é amplamente utilizada, por exemplo, em reconhecimento de caracteres de texto e para automatizar processos desde a contagem de células sanguíneas até a inspeção de controle de qualidade de produtos manufaturados em linhas de montagem. A análise de regiões nas imagens binárias acaba sendo uma das tarefas mais simples para as quais muitos algoritmos eficientes foram desenvolvidos e usados para implementar aplicativos confiáveis que estão em uso todos os dias. (BURGER; BURGE, 2009).

Como exemplo de aplicações de processamento de imagens, conforme exposto em (ACHARYA; RAY, 2005), pode-se citar sistemas de inspeção visual automáticos que realizam trabalhos de identificação de componentes faltantes e inspeção automática de superfícies de componentes, interpretação de cena remotamente detectada, sistemas de vigilância para defesa, recuperação de imagem baseada em conteúdo, acompanhamento de objetos em movimento, compressão de imagem e vídeo.

O desenvolvimento de sistemas de inspeção visual automatizada baseada em aplicações da tecnologia de visão de máquina desempenha, atualmente, um papel fundamental na busca da competitividade industrial (GOES; MIKOS, 2014). Através de dispositivos dotados de visão artificial, a indústria é capaz realizar tarefas de automação em processos que não permitem erros e que são assistidos pelas normas de qualidade já consagradas pelas organizações e clientes, que estão cada vez mais exigentes.

Neste contexto os sistemas de inspeção visual automatizados são essenciais para melhorar a produtividade e a qualidade do produto nas indústrias. A norma da Associação Brasileira de Normas Técnicas, ABNT NBR ISO 9001, bem difundida e conhecida pelas organizações, trata exatamente dos requisitos que um sistema de gestão da qualidade deve possuir para a organização funcionar de maneira eficaz.

Para uma organização funcionar de maneira eficaz, ela deve determinar e gerenciar diversas atividades interligadas. Uma atividade ou conjunto de atividades que usa recursos e que é gerenciada de forma a possibilitar a transformação de entradas em saídas pode ser considerada um processo. Frequentemente a saída de um processo é a entrada para o processo seguinte (ABNT NBR ISO 9001). A aplicação de um sistema de processos em uma organização, junto com a identificação, interações desses processos e sua gestão para produzir o resultado desejado, pode ser referenciada como “abordagem de processo”. Esta abordagem enfatiza a importância da obtenção de resultados de desempenho e eficácia do processo. Se a saída de um processo for monitorada, a entrada no processo seguinte permite que este novo processo alcance melhores resultados, tornando o produto final mais elaborado e capaz de atender às necessidades dos clientes.

Desde a introdução de sistemas de visão na indústria, as pessoas inspecionam os produtos por possíveis defeitos mecânicos. Diversas são as demandas para estes sistemas, que normalmente são desenvolvidos especialmente para cada problema, pode-se citar como alguns exemplos a inspeção de defeitos de qualidade por contaminação em baterias, inspeção de rótulos na indústria alimentícia, leitura de códigos de barra, inspeção de

presença e localização de componentes a serem embalados, inspeção de placas de circuito integrado, verificação de níveis de enchimento de garrafas entre outros.

A inspeção e avaliação utilizando processamento de imagens encontra enorme campo de aplicação, segundo Georgopoulos *et al.* (1995), o avanço tecnológico em sistemas de inspeção automatizados baseados em imagens proporciona oportuna avaliação das condições de superfícies de pavimentos e da malha rodoviária. A avaliação de defeitos de soldagem em radiografias de gasodutos usando visão computacional é realizada utilizando informações como comprimento, largura, área e perímetro destes defeitos (SHAFEEK *et al.* 2003). Outra aplicação envolvendo solda é discutida por Chu (2016), onde um algoritmo de inspeção de qualidade baseado em visão de máquina inspeciona a solda de tubos metálicos. Stivanello *et al.* (2016), relata o uso de equipamentos de alta tecnologia no processo produtivo de tecidos de malha que permitem a detecção e classificação automática de defeitos nestes tecidos. No trabalho é proposto um sistema de detecção e classificação dos defeitos nos tecidos utilizando técnicas de análise de imagens. No trabalho de Hamdi *et al.* (2017) é discutido um sistema de detecção de defeitos em tecidos utilizando imagens por infravermelho em que são detectados defeitos em diversos tipos de estampa de tecidos.

Na agroindústria a visão computacional ganha cada vez mais espaço. Com o aumento das expectativas de produtos alimentícios de alta qualidade e padrões de segurança, a necessidade de determinação precisa, rápida e objetiva da qualidade dessas características em produtos alimentícios continua a crescer (BROSNAN, 2003). A inspeção de qualidade baseada na cor e classificação de batatas têm sido desenvolvida para atender aos requisitos dos consumidores (NOORDAM *et al.*, 2000). O processamento de imagens para a detecção de hematomas em maçãs tem sido utilizado, visto que é um importante índice de avaliação de qualidade e um fator que acelera a deterioração da fruta (XING *et al.*, 2006).

Sapirsten *et al.* (1987), descreve um sistema de processamento digital de imagens para facilitar a inspeção objetiva e a classificação dos grãos de cereais. Imagens digitalizadas de amostras de grãos inteiros de trigo vermelho, cevada, centeio e aveia foram obtidas usando uma câmera de vídeo com dispositivo de carga acoplada (CCD) conectada a um sistema de aquisição de dados feito sob medida.

Ao fazer uma revisão sobre inspeção e classificação de produtos agrícolas e alimentícios, Brosnan (2002) relata que a visão computacional é uma técnica de inspeção rápida, econômica, consistente e objetiva, que se expandiu em indústrias diversas. Sua velocidade e precisão satisfazem os crescentes requisitos de produção e qualidade,

auxiliando no desenvolvimento de processos totalmente automatizados. Da mesma forma, Zheng *et al.* (2006), revisita o tema nas aplicações e desenvolvimentos recentes de atributos de imagens para avaliação da qualidade de produtos alimentícios.

Kaur *et al.* (2016), utilizou a visão de máquina para a classificação automática de rizoma de cúrcuma usando características morfológicas para a redução de inspeções manuais. Zalkan *et al.* (2016), propõe um sistema para estudar e classificar os diferentes tipos de ovos comerciais, realizando a medição dos diversos tamanhos e diâmetros.

O reconhecimento de padrões visuais independentemente da posição, do tamanho, e da orientação no campo visual, é alvo de muitas pesquisas desde 1961, quando HU (1962) apresentou a teoria dos invariantes de momento bidimensional para figuras geométricas planas utilizando a teoria geral de invariantes algébricos. Dudani *et al.* usou os invariantes de momento de Hu até a terceira ordem no reconhecimento de imagens de aeronaves. Os mesmos invariantes também foram utilizados no reconhecimento de navios (YAP; PARAMESRAN, 2003). Wong e Hall, e Goshtashby, empregaram invariantes de momento em combinação de modelos e registro de imagens de satélite (FLUSSER, 2006).

A identificação e inspeção de objetos 2D usando descritores de forma baseados em momento é tema de um trabalho de Sluzek (1995), onde são utilizados círculos de diferentes raios nos centroides dos objetos para criar as formas que são analisadas por momentos. Saad *et al.* (2017), apresenta um sistema de visão computacional automático para monitoramento e inspeção de qualidade em garrafas com defeitos de forma.

Boa parte do estado da arte foi apresentada nesta introdução, este levantamento serve para uma contextualização da proposta deste trabalho, que utiliza invariantes de momento para caracterização de peças metálicas através de suas formas, que fazem parte de um conjunto fabricado por determinada indústria. Esta caracterização é feita pelos eixos inerciais da elipse que possui o mesmo momento de segunda ordem da forma plana que a peça é reduzida. Isto permite uma verificação de inspeção uma vez que cada peça tem o seu próprio invariante de momento.

O processo de inspeção em pequenas empresas e em empresas familiares, em alguns casos, ainda é muito básico. Em determinados processos industriais que são totalmente executados por colaboradores ou, em outras palavras, processos totalmente manuais ou artesanais, os riscos de não conformidades do produto são maiores se comparados aos processos executados por máquinas. Em tais processos, o monitoramento e inspeção do

produto são vitais se o objetivo da empresa é manter um compromisso com a qualidade e com os requisitos determinados pelas normas e pelos clientes.

Esse problema aumenta significativamente no caso em que o cliente possui requisitos de qualidade específicos, cuja empresa fornecedora deve cumprir para manter o fornecimento. Conseqüentemente, a organização deve garantir a qualidade do produto a partir do monitoramento e da inspeção para fornecer evidências da conformidade do produto com os requisitos determinados.

Nestas circunstâncias, a implementação de um sistema que identifique as não conformidades em determinado processo produtivo é justificada, e diante das vantagens que os sensores de visão aplicados no processo industrial podem proporcionar, a implementação de um sistema de inspeção automático pode levar a uma melhor avaliação da conformidade do produto e resultados mais confiáveis.

Neste trabalho é apresentada uma proposta de metodologia para desenvolvimento de um sistema de inspeção com câmeras digitais utilizando a técnica de descritores de forma por momentos, para a extração de características dos objetos, e reconhecimento ótico de caracteres, para a extração de códigos alfanuméricos de produtos. Uma câmera digital captura o código do produto e uma segunda câmera captura a imagem dos componentes utilizados na montagem. As imagens são processadas visando o reconhecimento do código e extração de características dos componentes utilizados na montagem do produto para posterior classificação e verificação se os componentes utilizados estão de acordo com aquele código específico.

Para o andamento do trabalho, buscou-se numa primeira fase a familiarização com os conceitos e fundamentos envolvidos no processamento de imagens digitais voltados para o desenvolvimento do sistema de visão computacional, mais estritamente, de visão de máquina. O entendimento destes conceitos representa um passo fundamental para a compreensão e o desenvolvimento das várias etapas do trabalho, incluindo a aplicação do sistema de inspeção em um protótipo montado em bancada que será utilizado para validação dos resultados, bem como, avaliação da sua real aplicabilidade em um dispositivo embarcado instalado na linha de montagem do fabricante.

1.1 OBJETIVO

Este trabalho tem como objetivo principal o desenvolvimento de um sistema de visão de máquina com base no processamento e análise de imagens digitais para a inspeção das características e especificações das peças e componentes utilizados na montagem de um produto industrial, objetivando verificar e alertar a conformidade do produto final.

1.2 TÓPICOS DESTE TRABALHO

No capítulo 1, apresenta-se uma revisão bibliográfica do assunto abordado neste trabalho, inserida em uma introdução onde busca-se o entendimento do problema a ser resolvido bem como da proposta de solução para ele. São relacionados trabalhos e pesquisas na área onde ferramentas de processamento de imagens são utilizadas para a solução de questões similares às apresentadas neste texto. No capítulo 2, são apresentados os fundamentos e conceitos básicos das formações de alguns tipos de imagem, das transformações no domínio espacial, e dos processamentos para segmentação e extração de atributos de imagem. No capítulo 3 são apresentadas as metodologias utilizadas no desenvolvimento deste trabalho, empregando as ferramentas discutidas no capítulo anterior. É apresentado um sistema protótipo desenvolvido em bancada onde são validadas todas as ferramentas construídas para o sistema de visão. No capítulo 4 apresenta-se o sistema de visão desenvolvido em um dispositivo metálico para inspeção de componentes de um produto industrial. No capítulo 5 são apresentadas as principais conclusões obtidas na pesquisa e no desenvolvimento do sistema de visão, em seguida, as propostas para trabalhos futuros e possíveis pesquisas decorrentes deste trabalho. No apêndice A, é apresentada uma sinopse da linguagem de programação com algumas das *functions* utilizadas. No apêndice B, mostra-se um aplicativo utilizando a técnica de detecção de bordas que realiza uma medição sem contato do ângulo formado por objetos presentes na imagem. No apêndice C, é mostrado o desenho para confecção do dispositivo metálico utilizado para instalação do *hardware*. No apêndice D, apresenta-se uma melhor fundamentação do conceito de invariantes de momento.

2 CONCEITOS E FUNDAMENTOS BÁSICOS

Neste capítulo são apresentados a fundamentação e os conceitos básicos de formação e processamento de imagens digitais. O princípio físico que conduz à formação das imagens através dos sensores, as formas de amostragem e quantização, bem como a história dos dispositivos para captura digital das imagens, compõem um contexto para a melhor compreensão do trabalho. As técnicas de processamento para transformação no domínio espacial e para extração de atributos de imagem são discutidas e fornecem as ferramentas empregadas para o desenvolvimento e implementação do sistema proposto.

2.1 SENSORES DE VISÃO E TIPOS DE IMAGENS DIGITAIS

A história da câmera digital começou nos laboratórios Jet Propulsion Laboratory da California Institute of Technology em 1961 com Eugene F. Lally. A ideia era utilizar um fotosensor em forma de mosaico para capturar imagens de planetas e estrelas durante viagens espaciais para dar informações sobre a posição dos astronautas. Em 1972, Willis Adcock's, que trabalhava na Texas Instruments, patenteou uma câmera que não utilizava filmes e o engenheiro Steven Sasson, da Kodak, inventou e construiu a primeira camera eletrônica usando a tecnologia CCD em 1975.

Duas tecnologias estão presentes nos sensores das câmeras, CMOS e CCD, e não há uma conclusão definitiva sobre qual tecnologia é a melhor. Não somente o que é tecnicamente aplicável nos sensores, mas também o que é comercialmente viável afeta a escolha do equipamento ou sensores utilizados. Algumas aplicações funcionam melhor com CMOS e outras com CCD. A tecnologia de fabricação dos sensores CCD é um tanto dispendiosa financeiramente e, além disso, não permite a inclusão de circuitos de controle na mesma pastilha semicondutora ao contrário da tecnologia CMOS.

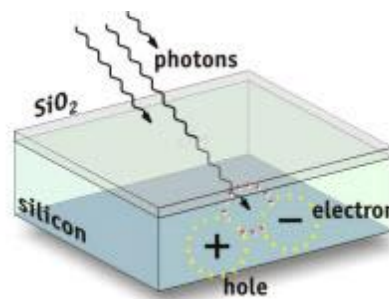
Outra vantagem da tecnologia CMOS é o consumo de energia, que é reduzido a aproximadamente um quinto do consumo da tecnologia CCD (Vellacott 1994, citado por FILHO, 1999). CCD (Charge Coupled Device) e CMOS (complementary metal oxide semiconductor) são duas tecnologias diferentes para capturar imagens digitalmente, ambos convertem a luz em sinais eletrônicos. A tecnologia CCD baseia-se em pastilhas

semicondutoras com determinado número de recipientes capazes de armazenar carga elétrica, que possuem um determinado mecanismo de transferência entre elas. A utilização de materiais fotossensíveis permite a construção de dispositivos CCD cujos recipientes de carga constituem também elementos sensíveis à luz. Assim, a carga armazenada em cada recipiente é proporcional à intensidade luminosa incidente sobre o mesmo. Posteriormente, a leitura sequencial dos recipientes pode gerar um sinal elétrico variante no tempo nos moldes do padrão gerado por tubos de câmera (FILHO, O. M.; NETO, H. V. 1999).

No sensor CMOS, semicondutor de oxido metálico complementar, cada pixel tem a própria conversão carga/voltagem e muitas vezes condicionamento, correção de ruído e circuitos de digitalização, de tal forma que a saída do chip é digital. Com cada pixel fazendo sua própria conversão, a uniformidade é menor, mas também é massivamente paralela, permitindo alta largura de banda para alta velocidade. Esta abordagem permite que sejam incluídos na mesma pastilha circuitos de controle, conversores analógico/digital, memórias digitais e circuitos com funções específicas para o processamento de imagens, além da matriz sensora propriamente dita (FILHO, O. M.; NETO, H. V. 1999).

A Fig. 2.1 ilustra o elemento sensor de formação de imagem digital, que é alimentado e a energia recebida provoca a saída do sinal analógico. O arranjo de elementos sensores determina a resolução que um dispositivo de captura possui.

Figura 2.1 – Sensor de aquisição de imagem.



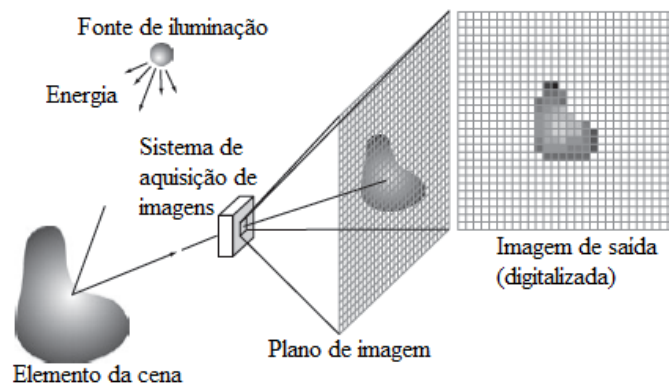
Fonte: CCD vs. CMOS, website da empresa TELEDYNE DALSA.

A maioria das imagens nas quais estamos interessados é gerada pela combinação de uma fonte de “iluminação” e a reflexão ou absorção de energia dessa fonte pelos elementos da “cena” cuja imagem está sendo gerada. Neste contexto, iluminação e cena estão entre aspas para enfatizar o fato de que elas são consideravelmente mais genéricas que a situação

habitual na qual uma fonte de luz visível ilumina uma cena 3-D comum (GONZALEZ; WOODS, 2009).

O arranjo de sensores predominante em câmeras digitais são os dispostos na forma matricial. Nesta configuração a energia é coletada e projetada em um plano contendo os sensores, o plano da matriz de sensores coincide com o plano focal do conjunto de lentes que normalmente acompanham os dispositivos. Este processo é ilustrado na Figura 2.2.

Figura 2.2 – Processo de formação de uma imagem digital.



Fonte: Adaptado de Gonzalez (2009).

A eletrônica embarcada juntamente com a pastilha de sensores realiza aquisição dos sinais analógicos de cada elemento sensor, e um componente específico do sistema de aquisição digitaliza estes sinais. A saída é uma imagem digitalizada.

2.1.1 Amostragem e quantização

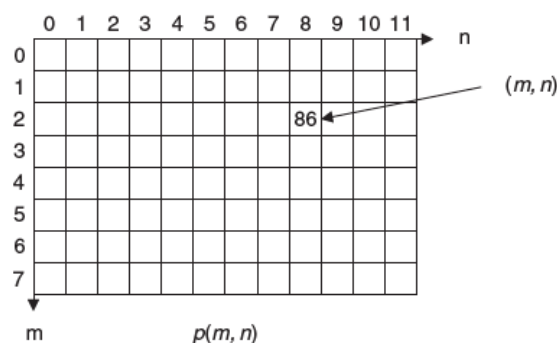
A imagem digital é resultado da amostragem no plano da matriz de sensores e na amplitude, que nada mais é do que a intensidade da energia captada por cada elemento. A digitalização nas coordenadas do plano é chamada de amostragem e é diretamente relacionada à resolução espacial própria de cada dispositivo; e a digitalização dos valores de amplitude é chamada de quantização.

A qualidade da imagem digital é claramente determinada, em grande parte, pelo número de amostras e de níveis discretos de intensidade utilizados na amostragem e na quantização (GONZALEZ, 2009). Uma imagem digital é composta por pixels e a posição de cada pixel é especificada através de um índice para o número de colunas, m , e outro para

o número de linhas, n . A Figura 2.3 mostra que o pixel na posição $[2, 8]$ tem uma intensidade de 86 e está localizado na segunda linha e oitava coluna. (TAN, 2008).

O número de pixels apresentados em uma imagem digital é a sua resolução espacial $m \times n$ que, como citado anteriormente, é diretamente relacionada à qualidade da imagem. Quanto maior é sua resolução espacial, mais qualidade a imagem apresenta. A resolução espacial pode ser alta, por exemplo, 1600×1200 (1.920.000 pixels, ou seja, 1,92 megapixels), ou baixa, 320×200 (64.000 pixels, 64 kilopixels).

Figura 2.3 – Discretização e quantização de uma imagem digital.



Fonte: Tan (2008).

Na notação o primeiro número representa a largura, e o número à direita, a altura da imagem. A qualidade da imagem também depende do número de *bits* usados na codificação da intensidade, ou nível, de cada pixel (TAN, 2008).

2.1.2 Imagens de 8 bits em escala de cinza

A quantidade de amostras utilizadas para gerar uma imagem no plano de formação, ou seja, a resolução espacial, é uma variável presente nos dispositivos e oferta uma grande variedade de opções no mercado. Já o número de níveis de intensidade é restrito a algumas poucas opções.

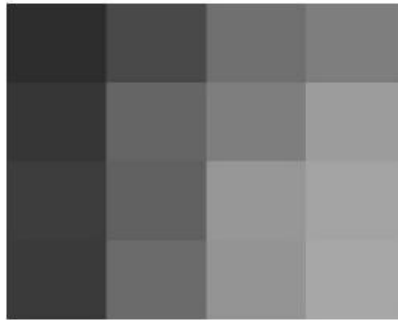
O número mais comum é 8 bits, ou seja, $2^8 = 256$ níveis, com 16 bits sendo utilizados em algumas aplicações nas quais o realce em determinadas faixas de intensidade é necessária. A quantização de intensidade utilizando 32 bits é rara. A quantização, também chamada de resolução de intensidade, refere-se à menor variação discernível dos níveis de intensidade de uma imagem (GONZALEZ, 2009).

Em uma imagem na escala de cinza de 8 bits, um pixel é processado em um intervalo de 0 a 255, onde o zero significa “preto” e 255 corresponde à cor “branca”, os números que ficam neste intervalo correspondem a níveis de cinza. Para uma imagem 640x480, são necessários 307,2 Kilobytes para realizar o armazenamento. A matriz abaixo, Equação 2.1, representa uma imagem de 8 bits e resolução espacial 4x4.

$$Imagem_{8\ bits} = \begin{bmatrix} 46 & 74 & 112 & 126 \\ 54 & 101 & 127 & 156 \\ 62 & 97 & 151 & 164 \\ 58 & 107 & 149 & 166 \end{bmatrix} \quad (2.1)$$

É possível notar na Figura 2.4 os diferentes níveis de cinza correspondentes às intensidades representadas pelos números da matriz.

Figura 2.4 – Imagem de 8 bits 4x4 pixels em escala de cinza.



Fonte: Elaborado pelo autor.

2.1.3 Imagens coloridas de 24 bits

Em uma imagem colorida de 24 bits, também chamada “true color”, cada pixel é registrado com um componente vermelho, um componente verde e um componente azul (RGB), conforme Figura 2.5.

Figura 2.5 – Imagem de 24 bits e seus componentes RGB.

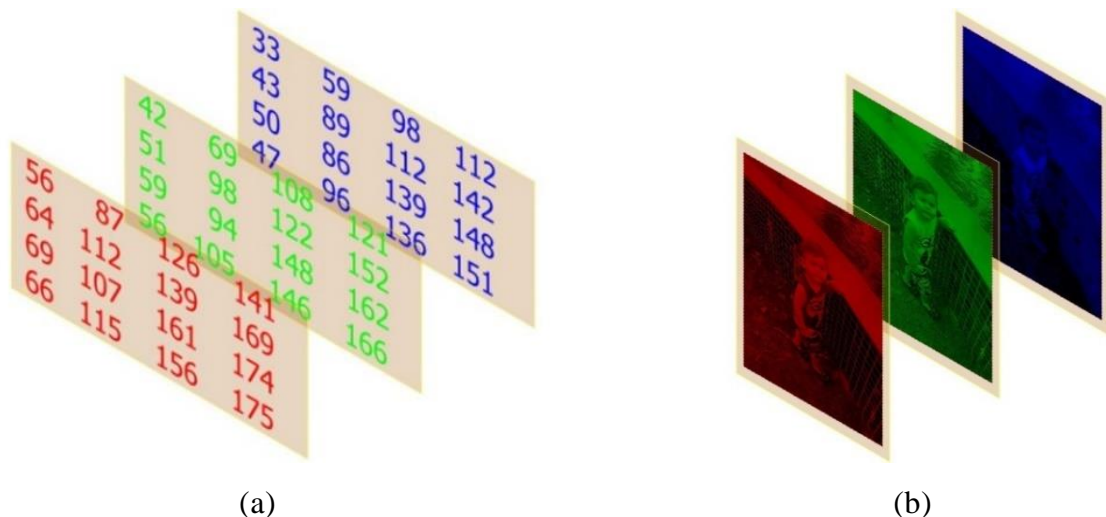


Fonte: Elaborado pelo autor.

Os modelos cromáticos são combinados de várias maneiras para reproduzir outras cores. Nesta imagem a resolução de intensidade atinge 2^{24} níveis porque cada componente é processado e registrado com 8 bits, ou seja, como há três canais separados de 8 bits o resultado é um total de 24 bits e 16,777216 milhões de cores. Em uma imagem colorida de 24 bits 640×480 são necessários 921,6 Kilobytes para realizar o armazenamento.

A imagem é armazenada como uma matriz tridimensional com dimensão $(m \times n \times 3)$ de valores inteiros no intervalo de zero a 255, conforme exemplificado na Fig. 2.6, que mostra os três canais em forma de matrizes, representando uma imagem 4×4 , Fig. 2.6a.

Figura 2.6 – Matriz tridimensional de uma imagem de 24 bits.



Fonte: Elaborado pelo autor.

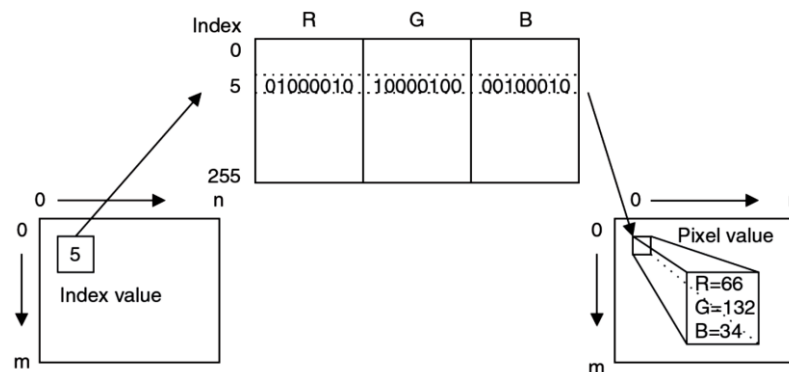
Este é o sistema de cores presente em monitores de TV e computador, retroprojetores, scanners e nas câmeras digitais. Para ilustrar a composição dos três canais

monocromáticos, a Fig. 2.6b ilustra a imagem da Fig. 2.5 da mesma maneira que a representação da matriz tridimensional.

2.1.4 Imagens coloridas de 8 bits

Na imagem colorida de 8 bits, chamada de imagem de cores indexadas, os valores dos pixels são índices que apontam para uma tabela de cores contendo componentes RGB, Fig. 2.7. Cada componente de cores é registrado com 8 bits, desta forma há somente 256 cores diferentes na imagem. Para armazenamento de dados, a imagem colorida de 8 bits de 640x480 pixels precisa de uma memória de 307,2 Kilobytes sendo necessários outros 768 bytes (3×256) para armazenamento da tabela de cores – também chamada de mapa de cores.

Figura 2.7 – Imagem de 8 bits de cores indexadas.



Fonte: Tan (2008).

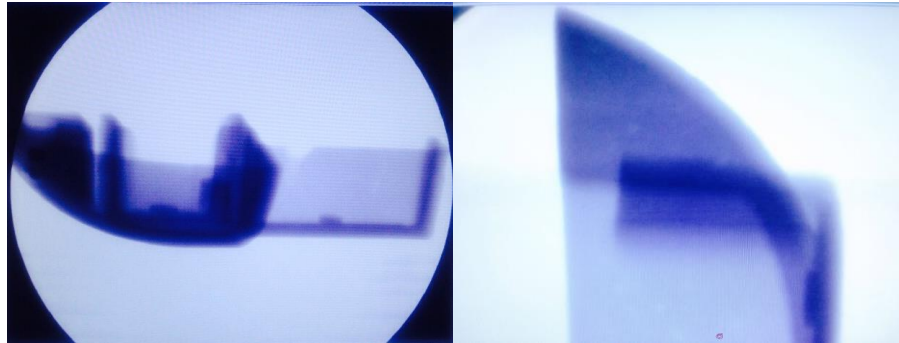
A imagem é armazenada como uma matriz bidimensional ($m \times n$) de valores inteiros (índices) em um intervalo de zero a 255. O mapa de cores é uma matriz $m \times 3$ com valores *floating point* em um intervalo de zero a 1. Na Fig. 2.7 é mostrada uma imagem que possui um pixel de valor 5, que é o índice para a tabela de cores. No mapa de cores há três componentes RGB associados ao índice 5 e desta forma é mostrada apenas uma cor para cada índice da imagem.

2.1.5 Fontes de energia e outras formações de imagem

A fonte de energia para formação de imagens mais utilizada é o espectro eletromagnético nas suas bandas visuais. No entanto, as fontes de energia podem ser as mais

variadas possíveis, como as imagens formadas por raios gama, por raios X, Fig. 2.8, pelo espectro não visível das ondas eletromagnéticas tais como na banda ultravioleta e na banda infravermelho, formação na banda de micro-ondas, banda de rádio, e por fim a utilização de energia acústica, o ultrassom, e a formação eletrônica de imagens.

Figura 2.8 – Formação de imagem industrial obtida por raios X.



Fonte: Autor.

Vê-se na Fig. 2.8 um exemplo de radiografia industrial usada como exame não destrutivo para verificação da integridade interna de peças. A imagem neste exemplo mostra um dos diversos componentes utilizados em uma capota marítima. Os raios X são muito utilizados para a detecção de descontinuidades internas dos materiais, que de outra forma não poderiam ser encontradas.

2.2 TRANSFORMAÇÕES EM SISTEMAS DE CORES

Os métodos de processamento de imagens podem ser divididos em duas categorias: métodos cuja entrada e saída são imagens e métodos cujas entradas podem ser imagens, mas as saídas são atributos extraídos dessas imagens. A *filtragem* e o *realce* de imagens são processos de manipulação de uma imagem de forma que o resultado, uma imagem de saída, seja mais adequado do que o original para uma aplicação específica (GONZALEZ, 2009). As técnicas para realçar as imagens são direcionadas especialmente para o problema a ser resolvido.

2.2.1 Conversão entre os espaços RGB e YIQ

Rotineiramente são necessárias conversões entre os espaços de cores, também chamados de sistemas de cores ou modelos de cores, para fins de processamento ou até mesmo de armazenamento. Como exemplo, pode-se citar as imagens de impressões digitais, que são armazenadas na memória e processadas no formato de níveis de cinza. Outro exemplo é a compressão de imagens coloridas utilizando o espaço de cores chamado YIQ (*luminance, in-phase, quadrature*), sistema que possui um canal para a luminância, que representa a intensidade de luz, e dois canais para de crominância, que representam as particularidades das cores.

Neste sistema, a luminância carrega toda a informação necessária para a formação de uma imagem na escala de cinza, e após a conversão da imagem original no espaço RGB (três matrizes $m \times n$) para o espaço YIQ, onde são formadas novas matrizes $m \times n$, apenas a informação $Y(m \times n)$ é utilizada como uma nova imagem na escala de cinza. Para esta conversão é utilizada uma matriz de transformação definida pela matriz padrão abaixo:

$$\begin{bmatrix} Y(m,n) \\ I(m,n) \\ Q(m,n) \end{bmatrix} = \begin{bmatrix} 0,299 & 0,587 & 0,114 \\ 0,596 & -0,274 & -0,322 \\ 0,212 & -0,523 & 0,311 \end{bmatrix} \begin{bmatrix} R(m,n) \\ G(m,n) \\ B(m,n) \end{bmatrix} \quad (2.2)$$

A transformação inversa, onde os componentes RGB são obtidos a partir do espaço YIQ, é dada por:

$$\begin{bmatrix} R(m,n) \\ G(m,n) \\ B(m,n) \end{bmatrix} = \begin{bmatrix} 1,000 & 0,956 & 0,621 \\ 1,000 & -0,272 & -0,647 \\ 1,000 & -1,106 & 1,703 \end{bmatrix} \begin{bmatrix} Y(m,n) \\ I(m,n) \\ Q(m,n) \end{bmatrix} \quad (2.3)$$

Como a imagem na escala de cinza é formada apenas pela parcela de luminância, dada uma imagem RGB a conversão fica:

$$Y(m,n) = 0,299 [R(m,n)] + 0,587 [G(m,n)] + 0,114 [B(m,n)] \quad (2.4)$$

Desta forma, encontramos a correspondente imagem em escala de cinza a partir da imagem RGB mantendo apenas o canal de luminância e descartando os outros dois canais IQ de crominância, conforme mostrado na Figura 2.10.

A conversão para o sistema NTSC (National Television System Committee), sistema de TV analógica, também utiliza o espaço YIQ , onde uma imagem RGB de 24 bits é convertida para uma imagem YIQ também de 24 bits.

Figura 2.9 – Imagem no modelo de cores RGB, à esquerda, e correspondente imagem em escala de cinza, à direita, obtida após conversão.



Fonte: Elaborado pelo autor.

Em muitos casos onde ambientes de processamento utilizam operações com *floating-point* a imagem em escala de cinza, cujos valores de intensidade estão no intervalo 0-255, têm seus valores normalizados para um intervalo entre 0 e 1 e são as chamadas imagens de intensidade.

2.2.2 Filtragem espacial

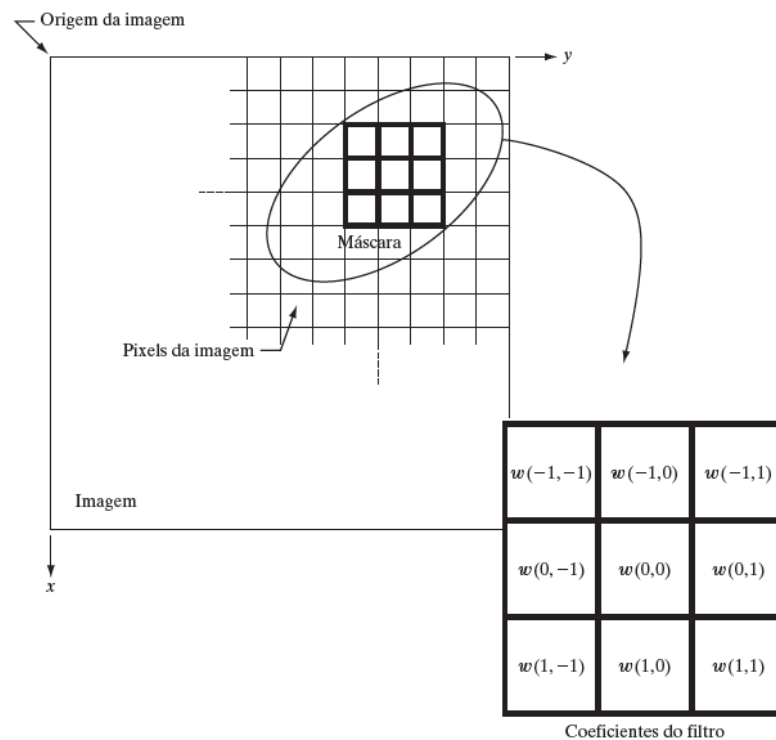
O processo de transformação no domínio espacial que será abordado neste trabalho pode ser definido pela equação abaixo (2.5), onde f é a imagem de entrada, g é a imagem de saída, T é a transformação aplicada, e consiste em mover um ponto (posição do pixel) na imagem aplicando um operador T na vizinhança do respectivo ponto.

$$g(x, y) = T[f(x, y)] \quad (2.5)$$

Normalmente o processo tem início no canto superior esquerdo da imagem de entrada e avança pixel a pixel em uma varredura horizontal da imagem, linha por linha, o ponto ou origem ocupa o centro da vizinhança que tem tamanho muito mais inferior que a própria imagem. Na Fig. 2.10, é mostrado um ponto arbitrário (x, y) e sua respectiva vizinhança (*kernel*), neste caso é uma vizinhança 3×3 com oito pixels adjacentes à origem.

O processamento das imagens no plano dos pixels, ou seja, no domínio espacial, é utilizado para a realização de transformações nas intensidades dos pixels e filtragens espaciais, que são técnicas que operam diretamente nos pixels de uma imagem, diferentemente das operações no domínio da frequência onde o processamento é realizado na transformada de Fourier de uma imagem.

Figura 2.10 – Coeficientes da máscara 3×3 . A origem do *kernel* se move juntamente com sua vizinhança para gerar uma imagem de saída.



Fonte: Adaptado de Gonzalez (2009).

A transformação T aplicada nas intensidades da vizinhança do ponto (x, y) na imagem f produz uma intensidade $g(x, y)$ na imagem de saída g . Neste trabalho o operador T é definido como a média das intensidades da vizinhança. Quando a origem se encontra nas bordas da imagem parte da vizinhança fica fora dela, isto cria um efeito nas bordas da

imagem g pelo fato de se ignorar os vizinhos externos nos cálculos de T ou preencher as bordas com zeros ou outros valores de intensidade predefinidos. Este procedimento é chamado de filtragem espacial, no qual a vizinhança, acompanhada de uma operação predefinida, é chamada de *filtro espacial*. Também possui outras denominações como: *máscara*, *kernel*, *janela*. O tipo de operação realizada na vizinhança determina a natureza do processo de filtragem (GONZALEZ, 2009).

As operações que calculam a média dos pixels contidos na máscara, são comumente chamadas de filtros de suavização, também chamadas de filtros de média, e são usadas para o borramento e redução de ruídos. O borramento é aplicado em tarefas de pré-processamento, como remoção de pequenos detalhes da imagem antes da extração de objetos (grandes) e conexão de pequenas discontinuidades em linhas ou curvas (GONZALEZ, 2009).

2.2.2.1 Filtros lineares de suavização

O conceito por trás dos filtros de suavização é que ao substituir o valor das intensidades de cada pixel de uma imagem pela média das intensidades dos pixels da sua vizinhança, a saída tem como resultado uma imagem com perda de nitidez, que significa uma suavização das “transições abruptas” nos níveis de intensidade. Uma utilização importante do processo de suavização ou borramento é a redução de detalhes considerados irrelevantes visto o tamanho muito pequeno destes detalhes diante do tamanho da imagem e do filtro utilizado.

A imagem formada g em qualquer ponto $g(x, y)$ é calculada utilizando os coeficientes do filtro ou máscara e os níveis de intensidade dos pixels da imagem de entrada como na Fig. 2.11. Como o filtro adotado neste caso gera uma média aritmética simples dos pixels cobertos pela máscara, o *layout* do filtro é composto por coeficientes 1 e uma constante de multiplicação $1/9$, necessários para esta média.

Figura 2.11 – Layout de uma máscara 3×3 para média aritmética simples.

$$\frac{1}{9} \times$$

1	1	1
1	1	1
1	1	1

Fonte: Adaptado de Gonzalez (2009).

Em qualquer ponto (x, y) da imagem a resposta $g(x, y)$ é dada pela equação 2.6 na forma a seguir. Computacionalmente é mais eficiente ter os coeficientes do filtro todos 1s, como mostrado na Fig. 2.12, e após o processo de filtragem toda a imagem é dividida por 9.

$$g(x, y) = \frac{1}{9} [w(-1, -1)f(x - 1, y - 1) + w(-1, 0)f(x - 1, y) + \dots + w(0, 0)f(x, y) + \dots + w(1, 1)f(x + 1, y + 1)] \quad (2.6)$$

2.2.3 Alargamento de contraste

Como estudado anteriormente, a transformação representada pela equação 2.5 realiza a filtragem espacial que é caracterizada por uma operação pré-definida em uma vizinhança da origem o *kernel*. Quando esta vizinhança possui o tamanho 1×1 , a menor vizinhança possível, a imagem g depende apenas do valor de f em um único ponto (x, y) , e a transformação T é chamada de *função de transformação de intensidade*.

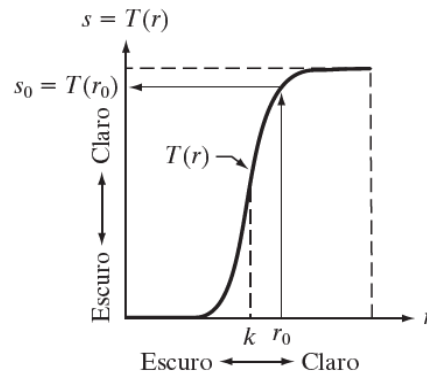
$$s = T[r] \quad (2.7)$$

Onde, s e r são variáveis que indicam, respectivamente, a intensidade de g e f em qualquer ponto (x, y) . Considerando que a transformação $T(r)$ possui a forma da Fig. 2.12, o resultado da operação em cada pixel de f para gerar os pixels em g produz uma imagem

de maior *contraste* do que a original, escurecendo os níveis de intensidade abaixo de um determinado valor de intensidade k e clareando os níveis acima de k .

Nessa técnica, algumas vezes chamada de alargamento de contraste, os valores de r menores que k são comprimidos pela função de transformação em uma faixa estreita de s , na direção do nível mais escuro. O oposto se aplica a valores de r maiores que k (GONZALEZ, 2009).

Figura 2.12 – Função de alargamento de contraste.



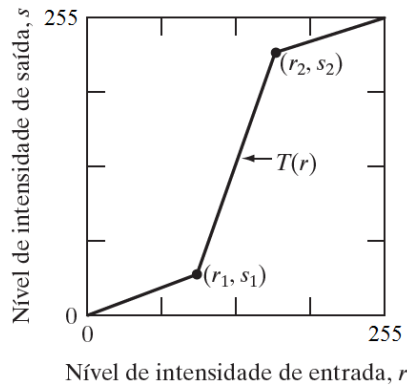
Fonte: Adaptado de Gonzalez (2009).

O contraste de uma imagem é entendido como a medida da dispersão dos valores das intensidades de cada pixel em relação à média destas intensidades. Desta forma, medidas como a variância e o desvio padrão das intensidades de toda a matriz da imagem são medidas úteis do contraste da imagem.

Como as unidades de variância são em valores de intensidade elevados ao quadrado, ao comparar valores de contraste, normalmente é utilizado o desvio padrão porque suas dimensões são diretamente em termos de valores de intensidade. Considerando que o intervalo de valores de intensidade em uma imagem possuindo quantização em escala de cinza de 8 bits é entre zero e 255, os valores de desvio padrão se relacionam com este intervalo de forma mais intuitiva do que a variância.

Uma abordagem mais simples consiste na utilização de funções lineares para a transformação de alargamento de contraste, Fig. 2.13. As posições dos pontos (r_1, s_1) e (r_2, s_2) controlam o formato da função de transformação. Isto permite, de forma mais vantajosa, que a função seja facilmente arbitrada expandindo a faixa de níveis de intensidade de uma imagem de modo a incluir todo o intervalo de intensidades dos dispositivos de formação e exibição, característica do alargamento de contraste.

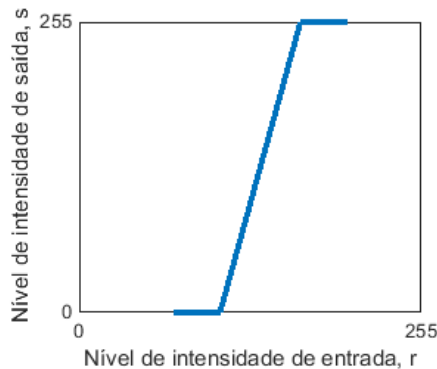
Figura 2.13 – Alargamento de contraste por meio de funções de transformação lineares.



Fonte: Adaptado de Gonzalez (2009).

É comum a utilização da função linear de alargamento de contraste, com parte dos pixels de menor intensidade da imagem de entrada saturados a zero na imagem de saída, e parte dos pixels de maior intensidade saturados a 255. A Fig. 2.14 ilustra esta função, desta forma, o ponto (r_1, s_1) é definido arbitrariamente com $s_1 = 0$, e o ponto (r_2, s_2) com $s_2 = 255$.

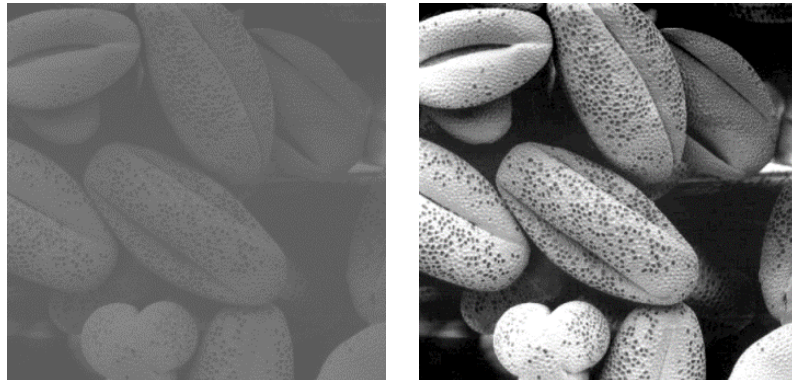
Figura 2.14 – Função linear de alargamento de contraste com parte dos níveis de intensidade saturados.



Fonte: Elaborado pelo autor.

Um exemplo da operação de alargamento de contraste aplicado em uma imagem de baixo contraste pode ser visualizado na Fig. 2.15. A função de transformação de intensidade utiliza parte dos pixels com as menores intensidades (1% das menores intensidades) para saturar a zero estas intensidades na imagem de saída e parte das maiores intensidades para saturar a 255 (1% das maiores intensidades).

Figura 2.15 – Imagem de baixo contraste à esquerda e correspondente imagem de alto contraste à direita.



Fonte: Imagem esquerda de <http://www.imageprocessingplace.com>.

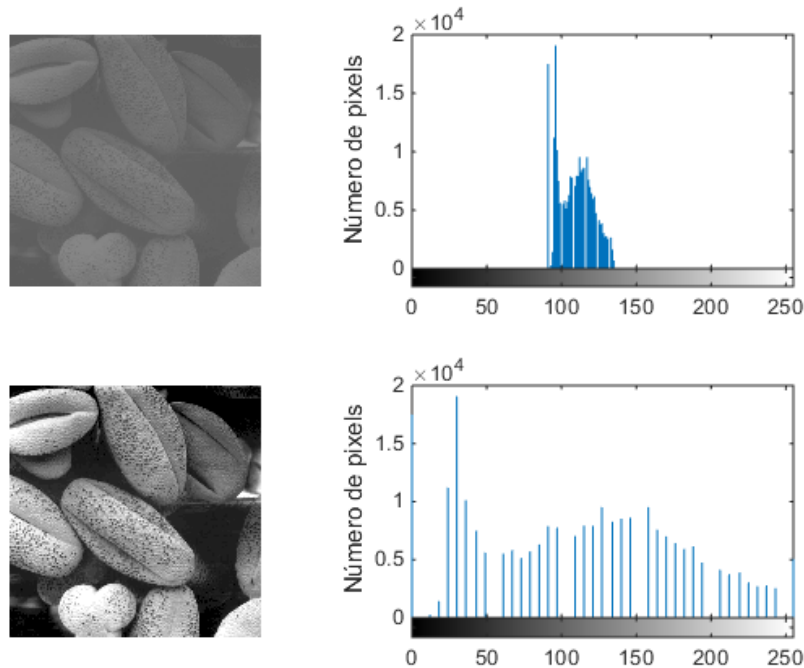
Ao se realizar o ajuste de contraste de uma imagem é possível notar a variação do histograma da imagem de entrada para a imagem de saída, mostrando que o processamento de histogramas como transformação de intensidade busca cobrir uma faixa muito ampla da escala de intensidades e realizar uma distribuição de pixels com tendência à uniformidade.

2.2.4 Histograma

A manipulação de imagens no domínio espacial pode ser analisada através de histogramas. O histograma de uma imagem digital com níveis de intensidade no intervalo $[0, 255]$, por exemplo, é uma função discreta onde cada valor de intensidade é um dado de entrada da função que retorna como resposta a quantidade de pixels que possui aquele valor de intensidade.

Intuitivamente, é razoável concluir que uma imagem cujos pixels tendem a ocupar todo o intervalo de níveis possíveis de intensidade e, além disso, tendem a ser distribuídos uniformemente terá uma aparência de alto contraste e exibirá uma grande variedade de tons de cinza. O resultado final será uma imagem que mostra boa correspondência em relação aos detalhes de nível de cinza e tem uma ampla faixa dinâmica (GONZALEZ, 2009).

Figura 2.16 – Imagens de baixo e alto contraste e correspondentes histogramas.



Fonte: Elaborado pelo autor.

A Fig. 2.16 mostra os histogramas das imagens de baixo e alto contraste, no histograma da imagem de baixo contraste uma faixa estreita de níveis de intensidade, que vai de zero a 255, é usada por todos os pixels, e no histograma da imagem de alto contraste toda a faixa de níveis é utilizada.

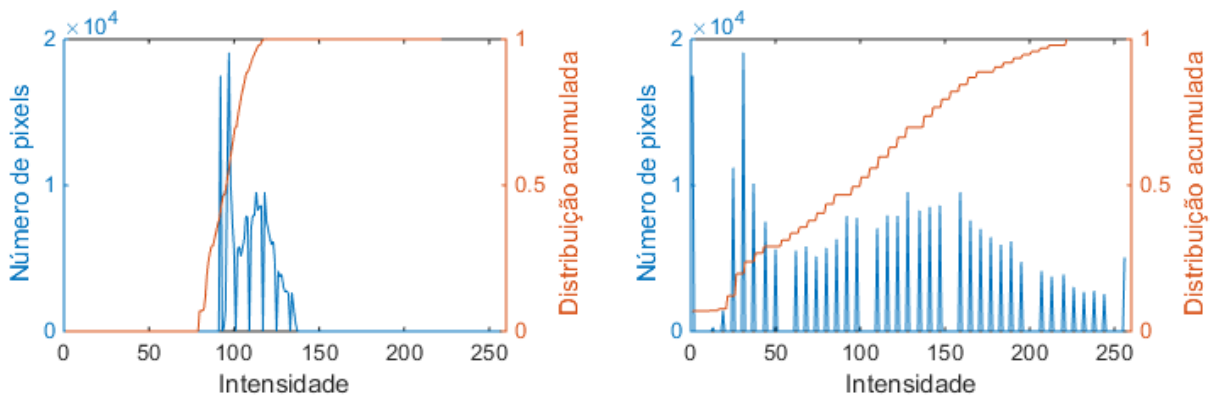
Um conceito muito utilizado em processamento de imagens é a função de distribuição acumulada para cálculo de equalização de histogramas. A equalização de histogramas ou linearização de histogramas não será abordada neste texto, porém a função de distribuição acumulada (*CDF*, *cumulative distribution function*) é utilizada para encontrar a porcentagem desejada de intensidades na imagem de entrada que serão saturadas tanto a zero quanto a 255 na imagem de saída. É calculada pela soma cumulativa de pixels do histograma dividida pela soma total de pixels da imagem:

$$CDF_k = \frac{1}{\sum n_k} \cdot \sum_{j=0}^k n_j \quad (2.8)$$

onde tem-se o valor de intensidade $k = 0, 1, 2, \dots, 255$, n_j é o número de pixels com intensidade r_k , o somatório $\sum n_k$ corresponde à soma das quantidades de pixels de cada valor de intensidade k , ou seja, é igual ao número total de pixels da imagem, $m \times n$.

A função de distribuição acumulada sobreposta ao respectivo histograma é mostrada na Fig. 2.17. Com a construção desta distribuição, a partir do histograma da imagem de entrada, é possível arbitrar qual a porcentagem de pixels que serão saturados na imagem de saída. Quando é arbitrado, por exemplo, 1% de saturação a zero e 1% de saturação a 255, busca-se a posição limite no vetor CDF que possui valor 0.01 e a posição limite de valor 0.99, e as intensidades relativas a estas posições são saturadas. Entre estes limites, conforme Fig. 2.14, a função de transformação é linear.

Figura 2.17 – Funções de distribuição acumulada sobrepostas aos histogramas da Figura 2.16.



Fonte: Elaborado pelo autor.

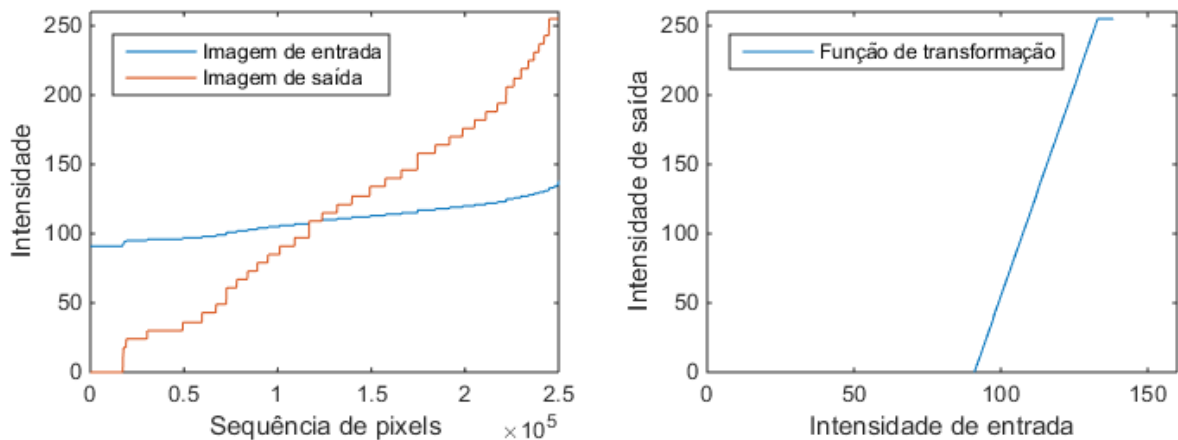
É interessante notar que a quantidade de níveis de intensidade utilizados pelos pixels é maior na imagem de entrada apesar da imagem ocupar um pequeno intervalo em toda a escala de 256 níveis possíveis. Neste caso específico, a imagem de entrada utilizou 42 níveis de intensidade, ao passo que a imagem de saída, após a transformação, utilizou 37 níveis dos 256 possíveis. Isto ocorre devido ao fato da imagem de saída ter parcela dos pixels saturados em dois níveis, zero e 255.

Uma outra forma de se visualizar o alargamento de contraste é realizar uma ordenação simples dos valores dos pixels das imagens. A imagem à esquerda da Fig. 2.18, mostra este resultado, a matriz da imagem de entrada é remodelada para ter apenas uma dimensão, ou seja, formar um vetor, e após isto é feita uma ordenação da menor para a maior

intensidade. Obviamente, a ordenação das intensidades da imagem de entrada ocupa uma faixa estreita da escala de intensidades e da imagem de saída ocupa toda a escala.

Quando colocamos em um gráfico a curva da imagem de saída pela curva da imagem de entrada o resultado é função de transformação relativa a estas imagens. Lembrando que a função de transformação já havia sido arbitrada quando determinamos os limites de saturação.

Figura 2.18 – Ordenação das intensidades das imagens de entrada e de saída, à esquerda, e função de transformação de intensidade para alargamento de contraste, à direita.



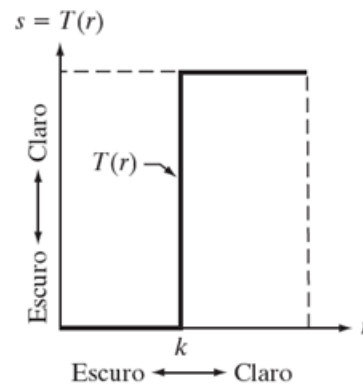
Fonte: Elaborado pelo autor.

O ajuste de contraste pode ser utilizado também em imagens RGB, neste caso, o mesmo mapeamento é realizado nos três planos da imagem, vermelho, verde e azul. A equalização de histograma de uma imagem RGB de 24 bits pode ser feita utilizando a conversão para o espaço YIQ, o canal Y é equalizado e recolocado no formato YIQ para transformação de volta ao espaço RGB. Na imagem colorida de 8 bits, de cores indexadas, os valores dos pixels ou de índices originais não são alterados, mas apenas o conteúdo do mapa de cores.

2.2.5 Limiarização

No caso limite da função mostrada na Fig. 2.12, $T(r)$ produz uma imagem de dois níveis (binária), esta nova função limite é ilustrada na Fig. 2.19. Esta conversão amplamente utilizada a partir da imagem em escala de cinza é chamada de limiarização (*thresholding*).

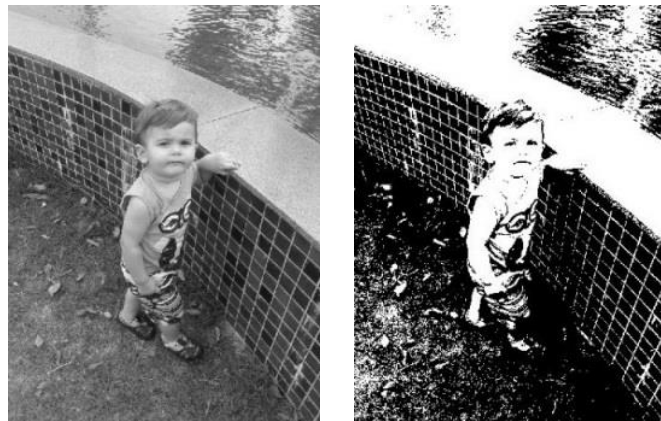
Figura 2.19 – Função transformação de intensidade para produção de uma imagem binária.



Fonte: Adaptado de GONZALEZ, 2009.

Na limiarização, ou *thresholding*, a intensidade de cada pixel é normalizada para os valores *zero* ou *um*, preto ou branco, onde o limiar, que rotineiramente é a média das intensidades, determina para qual valor a intensidade do pixel será normalizada. A Fig. 2.20 exemplifica esta transformação a partir de uma imagem na escala de cinza.

Figura 2.20 – Imagem em escala de cinza, à esquerda, e correspondente imagem binária.



Fonte: Elaborado pelo autor.

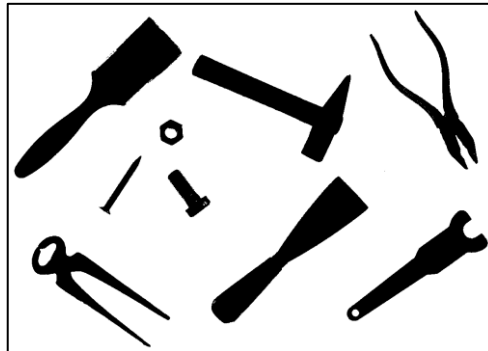
Em uma imagem de 8 bits todas as intensidades entre 0 e 127 (limiar) são normalizadas para 0, e todas as intensidades entre 128 e 255 são normalizadas para 1. A determinação do limiar pode ser feita arbitrariamente ou ser um resultado teórico de uma análise estatística.

2.3 ATRIBUTOS DE IMAGENS

O processamento morfológico trabalha com as ferramentas de extração de componentes presentes nas imagens que descrevem as mais variadas formas e o que elas representam. Neste ponto as saídas dos processos já não são mais imagens, mas sim atributos de imagens. A *segmentação* separa da imagem partes dela ou objetos individuais, como exemplificado na Fig. 2.21, para posterior identificação e é um dos processos mais difíceis no processamento de imagens.

A *representação* e a *descrição* na maioria das vezes se utilizam dos processos de segmentação previamente. A representação se refere às regiões de interesse onde são extraídas as fronteiras ou o conjunto de pixels presente no interior delas, definindo completamente o objeto a ser analisado. Quando o interesse se concentra em vértices e pontos de inflexão a determinação da fronteira é o mais apropriado ao passo que quando o interesse é a textura ou a forma do objeto a determinação da região aumentam as chances do processamento correto.

Figura 2.21 – Imagem binária de objetos segmentados.



Fonte: Burger e Burger (2009).

Na descrição ou seleção de características, a extração de atributos resulta em informação quantitativa e diferenciação de um tipo de objeto em relação a outro. O *reconhecimento* é a atribuição de um nome ou um rótulo aos objetos com base em suas características. Com o reconhecimento das características e descritores, um objeto pode, por exemplo, ter a atribuição de “torquês” ou “alicate” associada a ele. Neste trabalho a

descrição e reconhecimento serão feitos pelo uso dos descritores de forma e invariantes de momento, estudados mais à frente.

2.3.1 Detecção de borda

Na categoria dos métodos de processamento de imagens em que as entradas de um processo são imagens, mas as saídas são atributos de imagem, pode-se identificar nos objetos individuais suas bordas. Mudanças ou descontinuidades no valor de amplitude de uma imagem são importantes características primitivas porque fornecem uma indicação da extensão física dos objetos dentro da imagem. Estas mudanças locais bruscas na intensidade podem ser detectadas usando derivadas de primeira e segunda ordem. As derivadas de funções digitais são definidas em termos de diferenças, como será visto a seguir, dado que $\Delta x = 1$.

Dada a derivada de uma função contínua pode-se, a partir desta, estabelecer a derivada primeira de uma função digital. É evidente que a derivada de uma função pode ser analisada como sendo uma representação da inclinação de uma reta em determinado ponto desta função. Expandindo esta ideia para a função digital, a derivada primeira é abordada de tal maneira que possua valor *zero* nas áreas onde a intensidade se mantém constante (não há “inclinação”), seja diferente de zero no início de um degrau ou rampa de intensidade, e que seja diferente de zero nos pontos ao longo de uma rampa de intensidade.

$$\frac{\partial f}{\partial x} \cong \frac{f(x + \Delta x) - f(x)}{\Delta x} \quad (2.9)$$

$$\frac{\partial f}{\partial x} = f(x + 1) - f(x) \quad (2.10)$$

Para a segunda derivada tem-se:

$$\frac{\partial^2 f}{\partial x^2} = \frac{\partial' f(x)}{\partial x} = f'(x + 1) - f'(x) \quad (2.11)$$

$$= f(x + 2) - f(x + 1) - f(x + 1) + f(x) \quad (2.12)$$

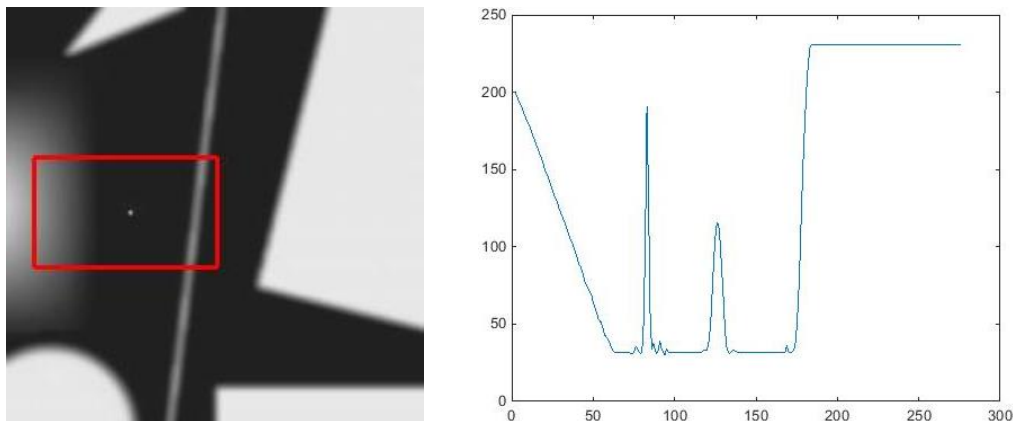
$$= f(x + 2) - 2f(x + 1) + f(x) \quad (2.13)$$

Como a expressão acima é sobre o ponto $x+1$ e nosso interesse é sobre o ponto x deve-se subtrair dos argumentos e obter a seguinte expressão:

$$\frac{\partial^2 f}{\partial x^2} = f''(x) = f(x + 1) + f(x - 1) - 2f(x) \quad (2.14)$$

De modo a ilustrar o efeito das derivadas primeira e segunda, aplica-se as equações 2.8 e 2.11 em um perfil de intensidades, mostrado na Fig. 2.23. A figura apresenta uma imagem em escala de cinza contendo objetos de diferentes formas, propícias para se analisar bordas. Tal perfil é criado a partir de uma linha horizontal exatamente na altura do ponto de intensidade alta destacado pelo retângulo em vermelho.

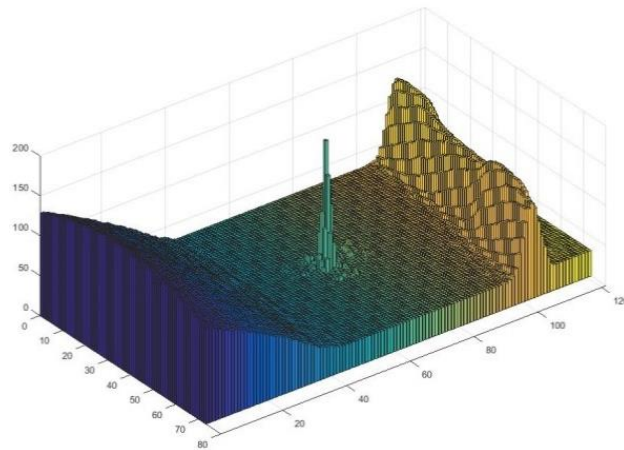
Figura 2.22 – Perfil de uma região em imagem de objetos em escala de cinza.



Fonte: Elaborado pelo autor.

Toda a região contida no retângulo vermelho é mostrada em três dimensões na Fig. 2.23. Características como rampa, mudanças bruscas e discontinuidades nos valores de amplitude são mostradas.

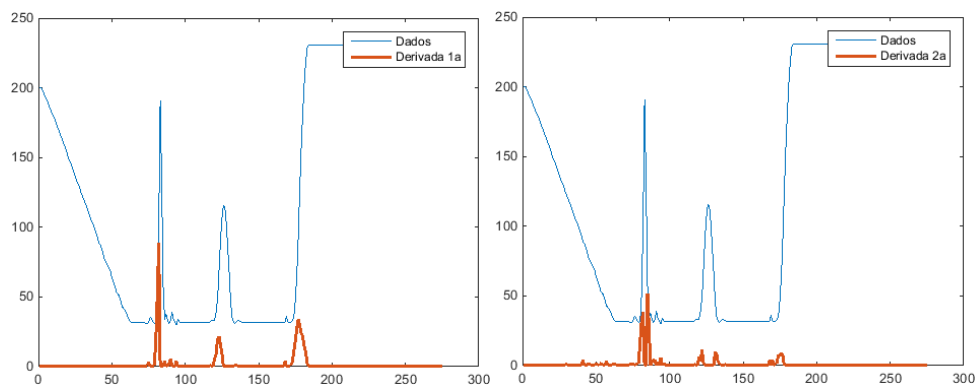
Figura 2.23 – Visualização 3D da região destacada da Figura 2.23.



Fonte: Elaborado pelo autor.

Após a aplicação da derivada primeira na função perfil, as bordas são detectadas no início dos degraus e podem ser observadas no gráfico à esquerda da Fig. 2.24. No caso da aplicação da segunda derivada, as bordas são destacadas no início e no final das variações abruptas de intensidade, conforme observado no gráfico à direita da Fig. 2.24.

Figura 2.24 – Visualização das bordas detectadas pelas derivadas de primeira e segunda ordens.



Fonte: Elaborado pelo autor.

2.3.2 Detecção de borda por filtragem espacial

A metodologia consiste basicamente em definir uma fórmula discreta da derivada de segunda ordem e construir uma máscara de filtragem com base nessa formulação. Estamos interessados em filtros isotrópicos, cuja resposta independe da direção das descontinuidades da imagem à qual o filtro é aplicado. Em outras palavras, os filtros isotrópicos são

invariantes em rotação, no sentido de que rotacionar a imagem e depois aplicar o filtro fornece o mesmo resultado que aplicar o filtro à imagem primeiro e depois rotacionar o resultado. (GONZALES, 2009).

Pode ser demonstrado (Rosenfeld e Kak, 1982) que o operador derivativo isotrópico mais simples é o laplaciano, que, para uma função (imagem) $f(x, y)$ de duas variáveis, é definido como:

$$\nabla^2 f(x, y) = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} \quad (2.15)$$

Como o laplaciano é um operador diferencial, sua utilização destaca as descontinuidades de intensidade em uma imagem e atenua as regiões com níveis de intensidade de variação mais suave.

Na direção x tem-se aplicado o definido para derivada segunda:

$$\frac{\partial^2 f(x, y)}{\partial x^2} = f(x + 1, y) + f(x - 1, y) - 2f(x, y) \quad (2.16)$$

E na direção y , da mesma forma, tem-se que:

$$\frac{\partial^2 f(x, y)}{\partial y^2} = f(x, y + 1) + f(x, y - 1) - 2f(x, y) \quad (2.17)$$

Desta forma, o Laplaciano discreto para duas variáveis é:

$$\nabla^2 f = f(x + 1, y) + f(x - 1, y) + f(x, y + 1) + f(x, y - 1) - 4f(x, y) \quad (2.18)$$

Esta equação pode ser implementada utilizando a máscara de filtragem da Fig. 2.25 que gera resultados isotrópicos em incrementos de 90 graus. A implementação é similar ao da filtragem linear de suavização.

Figura 2.25 – Máscara de filtragem representando o Laplaciano discreto de duas variáveis.

0	1	0
1	-4	1
0	1	0

Fonte: Adaptado de Gonzalez (2009).

As direções diagonais podem ser incorporadas à definição do laplaciano digital com o acréscimo de mais dois termos à Equação 2.16, um para cada direção diagonal. A forma de cada termo é a mesma das Equações 2.14 ou 2.15, mas as coordenadas estão ao longo das diagonais. Como cada termo de diagonal também contém $-2f(x, y)$, o total subtraído dos termos da diferença agora é $-8f(x, y)$. O resultado pode ser implementado utilizando a máscara da Fig. 2.26, que é isotrópica para incrementos de 45 graus.

Figura 2.26 – Máscara de filtragem representando o Laplaciano com diagonais incorporadas.

1	1	1
1	-8	1
1	1	1

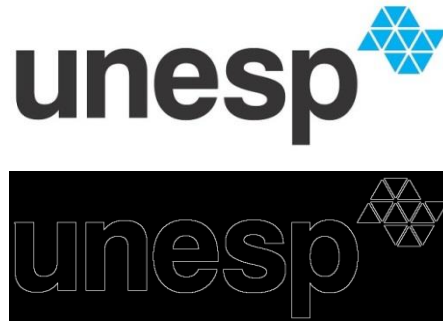
Fonte: Adaptado de Gonzalez (2009).

Um exemplo de aplicação do *kernel* da Figura 2.26 está apresentado na Fig. 2.27 através da Equação 2.17. A máscara percorre toda a extensão da imagem de entrada detectando todas as bordas presentes nos objetos. É importante observar que conforme a definição de derivada feita na seção anterior, onde o resultado é obtido pela diferença entre os pontos discretos, aqui a máscara 3x3 possui resultado *zero* no somatório de seus elementos.

$$g(x, y) = w(-1, -1)f(x - 1, y - 1) + w(-1, 0)f(x - 1, y) + \dots + w(0, 0)f(x, y) + \dots + w(1, 1)f(x + 1, y + 1) \quad (2.19)$$

É fácil notar que quando a imagem presente por “baixo” da máscara possui valores de intensidade constantes, o resultado na imagem de saída após a passagem da máscara é zero para seu centro. Normalmente estes valores *zerados* compõem o *background* da imagem de saída.

Figura 2.27 – Imagem de entrada acima e a saída com as bordas detectadas.



Fonte: Elaborado pelo autor.

Uma aplicação utilizando a detecção de bordas é apresentada no Apêndice B deste trabalho. Esta implementação detecta bordas de objetos segmentados em uma imagem com o objetivo de calcular o ângulo formado pelas mesmas.

2.3.3 Características Geométricas e Propriedades dos Objetos

Uma determinada região presente em uma imagem binária pode ser vista como pontos (x, y) , conectados ou não, distribuídos em um espaço discreto. Pode-se entender que uma região é composta por um ou mais objetos. Sabe-se que o valor que cada um destes pontos pode tomar é *um* ou *zero*, compondo uma matriz de valores que produz estruturas que são interpretadas como de primeiro plano ou de segundo plano.

Pela característica de uma determinada região, realiza-se medições quantitativas e qualitativas que são calculadas a partir do valor e da posição de cada pixel que compõe a região. A maioria das propriedades geométricas são definidas de tal forma que a região é considerada como sendo um arranjo de pixels (W. Burger, M.J. Burge. 2009).

Como exemplo de característica simples pode-se citar o tamanho ou *área* de uma região, que é entendida como a soma de todos os pixels desta região. Diferentes características são combinadas para marcar uma “assinatura” da região que pode ser usada para classificação ou comparação com outras regiões.

2.3.3.1 Momentos descritores

Os descritores de forma, ou *shape*, de um objeto ou região de uma imagem possuem uma abordagem estatística que é capaz de entregar um conjunto de características que são independentes da posição, da orientação, ou até mesmo do tamanho que o objeto se apresenta na matriz de imagem. Na estatística e na mecânica clássica, o conceito de momento é usado extensivamente como propriedades geométricas de figuras planas, normalmente são denominados como momentos estáticos, momentos de inércia, juntamente com o conceito de momento central. Eixos principais e centro de gravidade também são denominações comuns que decorrem do estudo destes momentos.

O teorema geral sobre momentos pode ser usado como ponto de partida para os invariantes que este trabalho irá tratar.

$$m_{p,q} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x^p y^q f(x,y) dx dy \quad (2.20)$$

Onde $(p + q)$ é a ordem do momento, $f(x, y)$ é a função distribuição de densidade, que representa a intensidade da região da imagem que contém os pontos x e y .

Como neste trabalho os invariantes de momento serão sempre em relação a uma imagem binária, tem-se que $f(x, y)$ sempre assumirá valores *um*.

$$f(x, y) = \begin{cases} 1 & \text{Objeto} \\ 0 & \text{Background} \end{cases} \quad (2.21)$$

No apêndice D deste trabalho é feito um detalhamento em relação aos momentos, são mostradas as ordens destes momentos assim como a definição de momento central e momento central normalizado.

O momento de qualquer ordem em um espaço discreto de uma imagem binária é mostrado abaixo.

$$m_{p,q} = \sum_x \sum_y x^p y^q \quad (2.22)$$

A partir deste momento, $m_{p,q}$, o momento central de qualquer ordem pode ser obtido definindo previamente o momento de ordem zero e o momento de primeira ordem.

$$\mu_{p,q} = \frac{m_{p,q}}{m_{0,0}} - \left(\left(\frac{m_{1,0}}{m_{0,0}} \right)^p \left(\frac{m_{0,1}}{m_{0,0}} \right)^q \right) \quad (2.23)$$

2.3.3.2 Área

A área de uma região é descrita pelo momento de ordem zero, obtido a partir da forma geral dos momentos, considerando uma imagem binária onde $f(x,y) = 1$ para os pontos da região, desta maneira pode ser calculado pelo somatório abaixo. A área representa o número de pixels com valor *um* que compõem as regiões, que podem ser entendidas como o *shape* dos objetos.

$$m_{0,0} = A = \sum_x \sum_y x^0 y^0 \quad (2.24)$$

2.3.3.3 Centroide

A forma de um objeto ou região assume um padrão geométrico, e todo padrão geométrico sempre pode ser representado por uma função $f(x,y)$. O centroide de uma região é definido a partir do momento de primeira ordem, e da mesma forma é considerada uma função de imagem binária para escrever o somatório de momento.

$$m_{1,0} = \sum_x \sum_y x^1 y^0 \quad m_{0,1} = \sum_x \sum_y x^0 y^1 \quad (2.25)$$

O centroide de uma região é calculado dividindo o momento de primeira ordem pela área da região.

$$\begin{aligned}x_c &= \frac{m_{1,0}}{m_{0,0}} = \frac{1}{A} \sum_x \sum_y x^1 y^0 \\y_c &= \frac{m_{0,1}}{m_{0,0}} = \frac{1}{A} \sum_x \sum_y x^0 y^1\end{aligned}\tag{2.26}$$

2.3.3.4 *Ellipse*

Sabe-se da teoria dos momentos que a forma geométrica possui um par de eixos inerciais que se cruzam no centroide e podem ser fixados sobre a forma plana da imagem. Os eixos inerciais são encontrados calculando os autovalores do tensor inercial, este por sua vez, é calculado a partir dos três momentos centrais de segunda ordem.

Utilizando os eixos inerciais é sobrescrita uma elipse que se aproxima da forma ou objeto em questão e é identificada como uma característica determinante para seu reconhecimento.

Os momentos de segunda ordem são obtidos a partir da forma geral 2.22, da mesma forma que os momentos de primeira ordem anteriormente mostrados.

$$\begin{aligned}m_{2,0} &= \sum_x \sum_y x^2 y^0 \\m_{0,2} &= \sum_x \sum_y x^0 y^2 \\m_{1,1} &= \sum_x \sum_y x^1 y^1\end{aligned}\tag{2.27}$$

Em seguida são determinados os três momentos centrais de segunda ordem. Aqui, mostra-se os momentos diretamente, que são demonstrados no apêndice D deste trabalho.

$$\begin{aligned}\mu_{2,0} &= \frac{m_{2,0}}{m_{0,0}} - x_c^2 \\ \mu_{0,2} &= \frac{m_{0,2}}{m_{0,0}} - y_c^2 \\ \mu_{1,1} &= \frac{m_{1,1}}{m_{0,0}} - x_c y_c\end{aligned}\tag{2.28}$$

Estes momentos compõem o tensor inercial mostrado abaixo.

$$J = \begin{bmatrix} \mu_{2,0} & -\mu_{1,1} \\ -\mu_{1,1} & \mu_{0,2} \end{bmatrix}\tag{2.29}$$

Os eixos inerciais são encontrados calculando os autovalores do tensor inercial.

$$\lambda_{1,2} = \sqrt{\frac{1}{2}(\mu_{2,0} + \mu_{0,2}) \pm \sqrt{4\mu_{1,1}^2 - (\mu_{2,0} - \mu_{0,2})^2}}\tag{2.30}$$

Os semieixos são definidos como $a = \text{semieixo maior}$ e $b = \text{semieixo menor}$ e definem a excentricidade da elipse.

$$\varepsilon = \frac{\sqrt{a^2 - b^2}}{a}\tag{2.31}$$

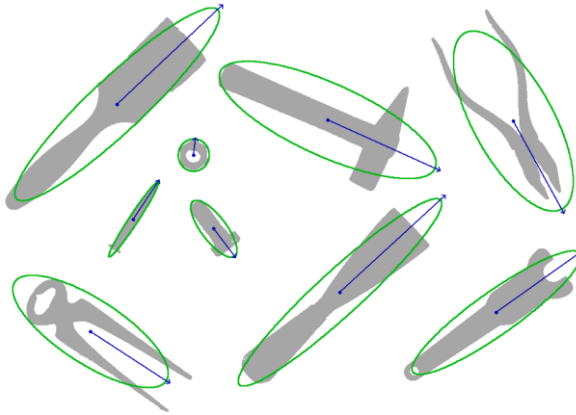
2.3.3.5 Orientação

A orientação do objeto é definida como o ângulo de inclinação entre o eixo x e o eixo principal do objeto, corresponde ao autovetor com o menor autovalor definido anteriormente. Nesta direção, o objeto tem a sua maior “extensão”.

$$\theta = \frac{1}{2} \arctan\left(\frac{2\mu_{1,1}}{\mu_{2,0} - \mu_{0,2}}\right)\tag{2.32}$$

Após a extração destes descritores, é possível desenhá-los computacionalmente sobrepondo-os aos seus respectivos objetos, como ilustrado na Fig. 2.28.

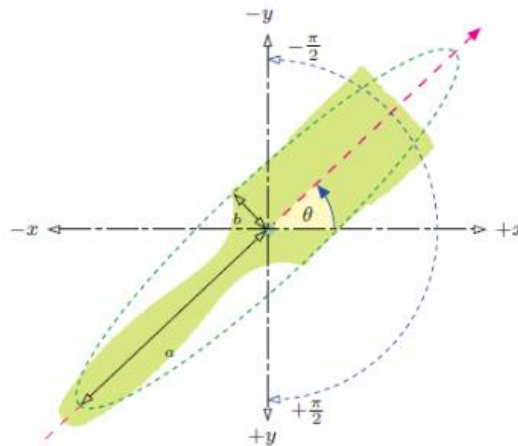
Figura 2.28 – Elipses sobrepostas aos seus objetos.



Fonte: Burger e Burge (2009).

A Fig. 2.29 destaca um destes objetos para fins de definição quanto ao semieixo maior, semieixo menor e orientação de objetos segmentados em uma imagem.

Figura 2.29 – Representação de objeto com descritores de orientação, eixos inerciais, e elipse.

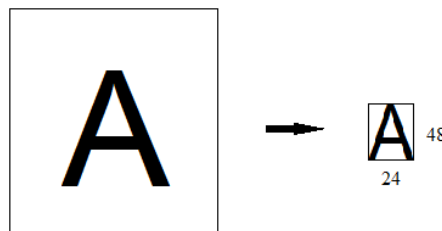


Fonte: Adaptado de W. Burger, M.J. Burge. 2009.

2.3.4 Reconhecimento de caracteres

A maioria dos processos de reconhecimento de texto passa pela segmentação que separa cada caractere. Os caracteres segmentados são normalizados até ficarem com a mesma resolução espacial de um *template* previamente armazenado, Fig. 2.30. Neste caso específico o *template* armazenado possui uma resolução espacial de 24x48 pixels.

Figura 2.30 – Normalização do caractere segmentado para a resolução do *template*, 24x48 pixels.



Fonte: Elaborado pelo autor.

Em cada caractere segmentado é realizada uma comparação pixel a pixel com modelos de caracteres previamente carregados, processo que recebe o nome de *matrix matching*. O *template* que atingir a maior taxa de correspondência com o objeto segmentado é escolhido como resultado do reconhecimento. Um exemplo de *template* 24x48 é apresentado na Fig. 2.31. Os modelos carregados previamente podem incluir, além das letras do alfabeto, números e símbolos.

Figura 2.31 – *Template* binário de 24x48 pixels.



Fonte: Elaborado pelo autor.

O procedimento *matrix matching* mencionado, realiza uma implementação de correlação entre matrizes. Normalmente busca-se um coeficiente de correlação entre

imagens, e o maior coeficiente traduz a maior correspondência entre a imagem do objeto e a imagem do *template*.

$$coef = \frac{\sum_m \sum_n (A_{m \times n} - \bar{A})(B_{m \times n} - \bar{B})}{\sqrt{(\sum_m \sum_n (A_{m \times n} - \bar{A})^2) (\sum_m \sum_n (B_{m \times n} - \bar{B})^2)}} \quad (2.33)$$

A equação 2.33 mostra como implementar um algoritmo para o cálculo deste coeficiente, onde A e B são matrizes $m \times n$, e \bar{A} é a matriz com a média dos elementos de A e \bar{B} é a matriz com a média dos elementos de B .

Para o algoritmo de reconhecimento óticos de caracteres, OCR, entregar os resultados de forma satisfatória é fundamental que alguns requisitos sejam cumpridos. O pré-processamento é quase obrigatório, por exemplo, para imagens naturais capturadas sem qualquer forma de controle e a iluminação é o fator preponderante para que o sistema de OCR reconheça o texto. Neste trabalho os pré-processamentos de filtragem de média e alargamento de contraste foram utilizados em uma etapa anterior ao chamamento da função de OCR.

Funções de reconhecimento aplicadas em caracteres que utilizam o mesmo tipo de fonte dos modelos armazenados costumam responder com uma taxa de reconhecimento de quase 100% de sucesso.

Neste capítulo apresentou-se a fundamentação teórica envolvida na formação e tratamento de imagens digitais, as transformações no domínio espacial, e as técnicas de processamento e extração de atributos destas imagens. Investigando-se os conceitos e fundamentos básicos necessários para o desenvolvimento do trabalho, busca-se a construção de ferramentas simples para a composição do sistema de visão para inspeção de produto industrial.

3 DESCRITORES DE FORMA E RECONHECIMENTO ÓTICO DE CARACTERES APLICADOS EM UM SISTEMA DE VISÃO DE MÁQUINA

Neste capítulo é apresentada a metodologia utilizada para o desenvolvimento e implementação do sistema de visão computacional proposto para a inspeção da conformidade do produto. A metodologia envolve um sistema de captura e inspeção de imagens digitais que busca identificar eventual não conformidade ou troca de componentes que podem ocorrer no processo de montagem do produto. Isso garante que o produto final atenda aos requisitos de qualidade especificados. O sistema é formado por um conjunto de duas câmeras digitais, uma para a identificação do código do produto e outra para a inspeção e identificação da conformidade do produto.

3.1 INSPEÇÃO DE CONFORMIDADE

Em alguns processos industriais os riscos de não conformidades do produto, defeito ou erros de montagem, são maiores quando executados de forma manual, se comparados aos processos executados por máquinas. Em tais processos, a inspeção do produto final é vital se o objetivo da empresa é manter um compromisso com a qualidade e, principalmente, para atender os requisitos das normas e aqueles especificados pelos clientes. Esse problema aumenta significativamente quando os clientes são montadoras do setor automobilístico, essas empresas exigem de seus fornecedores padrões específicos de qualidade para que o fornecimento seja mantido.

A entrega de produtos com alguma não conformidade e as respectivas consequências que recaem sobre a empresa fornecedora do produto são enormes. Isso pode ser mensurado, considerando que, uma vez identificada uma não conformidade na linha de montagem do cliente (montadoras do setor automobilístico) a responsabilidade e os custos deste problema são diretamente imputados à empresa fornecedor do produto. Os custos envolvem gastos com transportadora, custos por hora nos *boxes* de análise, custos de programas de identificação da causa raiz e correção na produção e outros. Caso ocorra alguma não

conformidade do produto, a empresa fornecedora é responsabilizada podendo até perder os certificados obtidos junto às certificadoras acreditadas.

Nestas circunstâncias, um sistema que identifique eventual não conformidade do produto é de fundamental importância para o departamento de qualidade da empresa e poderia representar uma significativa redução de custo na linha de montagem do produto.

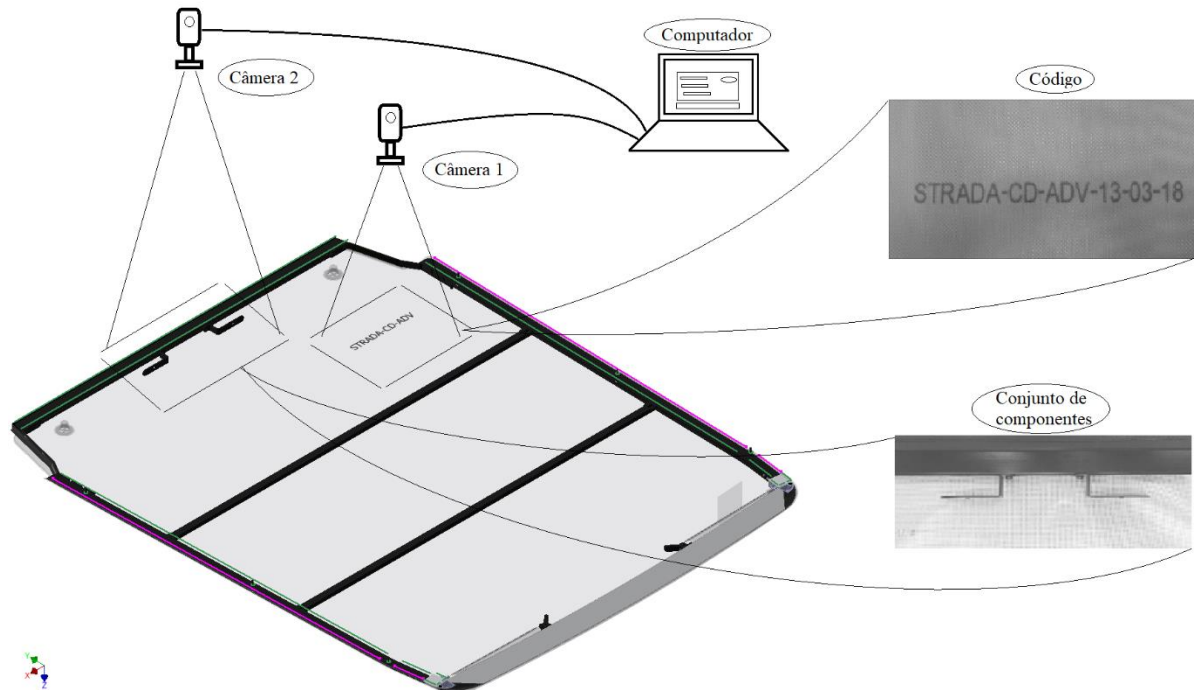
3.2 LAYOUT DO SISTEMA DE INSPEÇÃO

O sistema de inspeção proposto é composto de duas câmeras digitais, uma para captura do código de identificação do produto e outra para a captura e identificação dos componentes. As câmeras são operadas e controladas através de uma interface gráfica implementada em ambiente MATLAB®.

O sistema tem como propósito a inspeção final de conformidade de um produto automotivo, mais especificamente, o acessório dos veículos utilitários do tipo *pick-up* utilizado para cobrir e proteger o vão de carga, popularmente conhecido como lona ou capota marítima. O processo de montagem das capotas marítimas é o mesmo para todos os modelos, entretanto, cada carroceria (código do produto) possui características específicas, ou seja, para cada tipo de veículo os correspondentes componentes de fixação da capota na carroceria possuem dimensões e geometrias diferentes (características das peças de fixação). A inspeção é feita para garantir que não houve falha na linha de montagem do produto tais como montagem de componentes de fixação invertidos ou mesmo a troca de componentes de modelos diferentes.

Na Fig. 3.1 é mostrado o *layout* simplificado do sistema de inspeção proposto. As imagens são capturadas isoladamente e processadas, inicialmente, para identificação do código do produto e, posteriormente, para a identificação das características do conjunto dos componentes de fixação utilizado na montagem. No desenho é mostrado a região da posição do código e da posição das peças de fixação na capota marítima as quais vão definir as regiões de inspeção do produto.

Figura 3.1 – Modelo esquemático do sistema de inspeção.

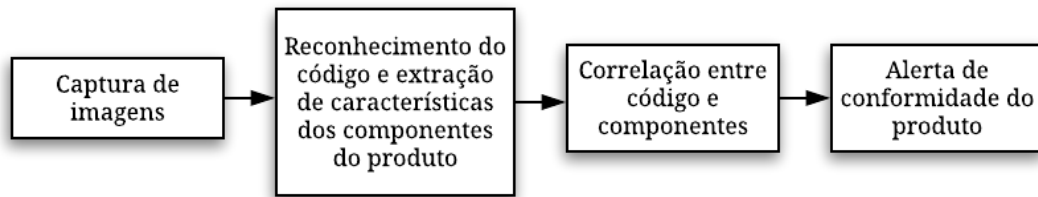


Fonte: Elaborado pelo autor.

Para a identificação do código do produto é utilizada a técnica OCR (*optical character recognition*) e a identificação dos componentes (peças) é feita com base no processamento e extração dos invariantes de momento das formas geométricas das peças. Posteriormente, os descritores dos objetos (peças) são associados ao código do produto e, desta forma, é possível verificar a conformidade dos componentes utilizados na montagem.

Na Fig. 3.2 é ilustrada esquematicamente a sequência das etapas do sistema de visão computacional, captura das imagens, processamento e identificação do produto e das características das peças utilizadas na montagem, verificação da conformidade do produto através da associação do código com os correspondentes componentes e, finalmente, a etapa de tomada de decisão, ou seja, o produto está ou não conforme.

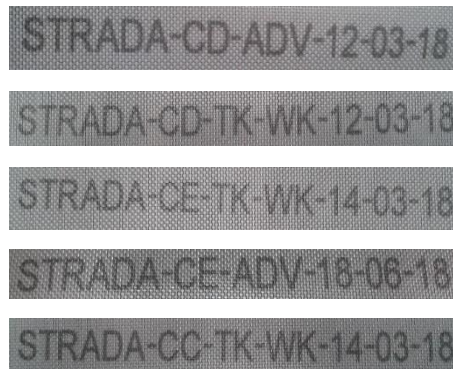
Figura 3.2 – Visão global das etapas do sistema de inspeção.



Fonte: Elaborado pelo autor.

Neste projeto em específico, os itens inspecionados são o código do produto e o conjunto de componentes de fixação, sendo que a proposta poderia ser expandida para outros componentes. Na Fig. 3.3 são mostrados os diferentes códigos e na Fig. 3.4 os dois conjuntos de componentes de fixação.

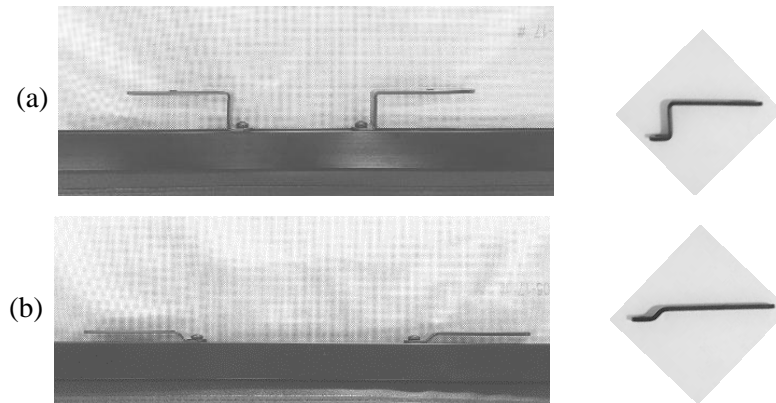
Figura 3.3 – Imagens dos códigos dos produtos inspecionados.



Fonte: Elaborado pelo autor.

Os componentes de fixação são produzidos em aço carbono e passam por corte e conformação através de ferramenta de estampagem acoplada em uma prensa excêntrica. A Fig. 3.4a mostra o conjunto cujos componentes de fixação possuem perfis mais largos, enquanto a Fig. 3.4b apresenta o conjunto de componentes mais delgados, cada qual projetado e conformado para uma carroceria específica.

Figura 3.4 – Imagens dos conjuntos de peças dos produtos inspecionados.



Fonte: Elaborado pelo autor.

Um aspecto importante no processo de captura da imagem dos itens inspecionados é o posicionamento da câmera, a item inspecionado deve ser corretamente enquadrados na cena para que o processamento e extração de características do objeto sejam capaz de identificar corretamente os descritores daquele componente.

3.3 BANCADA DE TESTE UTILIZADA NA IMPLEMENTAÇÃO COMPUTACIONAL DA METODOLOGIA DE IDENTIFICAÇÃO DO SISTEMA DE INSPEÇÃO

A bancada de teste utilizada para a implementação e validação da proposta de inspeção é composta basicamente de um computador com duas câmeras montadas sobre uma mesa de fundo branco onde serão posicionados o código de identificação do produto e as respectivas peças de fixação a serem identificadas, Fig. 3.5. Foram utilizadas duas câmeras digitais de 8 bits com resolução espacial de 640x480 pixels controlados via *software*. A câmera 1 é apontada para a região contendo a amostra do código alfanumérico do produto e a câmera 2 é direcionada para a região contendo a peça de fixação.

Figura 3.5 – Bancada para experimentação e desenvolvimento do sistema de inspeção.



Fonte: Elaborado pelo autor.

No processo de captura das imagens, além dos parâmetros de controle digitais, o controle dos fatores ambientais de luz e sombra são fundamentais, pois os mesmos vão interferir nos resultados da extração das características das imagens. Durante a preparação dos testes foi possível observar a importância da luz como um parâmetro que deve ser controlado no ambiente. A Fig. 3.6 mostra a reflexão da luz que chega no sensor dependendo da posição da câmera, no caso (a) é possível ler o código do produto facilmente enquanto no caso (d) a dificuldade é maior. Esse efeito afeta o processamento das imagens e dificulta a identificação correta dos parâmetros.

Figura 3.6 – Sequência de imagens mostrando a reflexão da luz.



Fonte: Elaborado pelo autor.

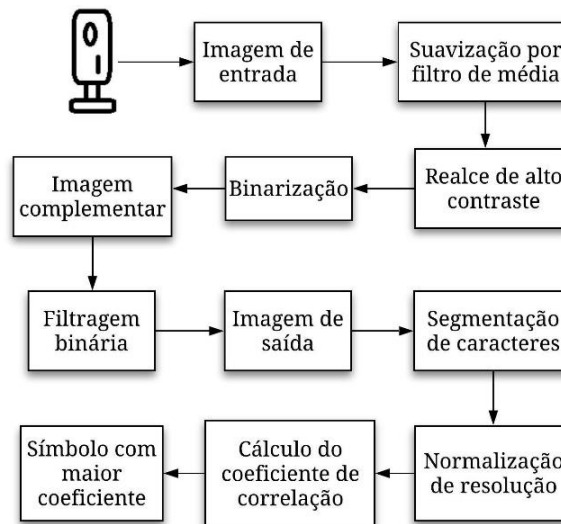
As etapas do processo identificação de características passam pela captura das imagens (código e componentes de fixação), filtragem espacial utilizando máscara de médias para suavização, alargamento de contraste para realce, conversão para imagem binária utilizando um nível de *threshold*, filtragem de objetos binários estranhos aos objetos

de interesse e segmentação. Uma vez obtidos os parâmetros e características do código e dos componentes de fixação é verificada a conformidade do produto, ou seja, se os componentes de fixação identificados são os mesmos associados a aquele respectivo código.

3.4 IDENTIFICAÇÃO DO CÓDIGO DO PRODUTO

O fluxograma das etapas de processamento para o reconhecimento do código é mostrado na Fig. 3.7. Inicialmente é feita a filtragem espacial para suavização das imagens e o alargamento de contraste para realçar as intensidades como pré-processamentos para a segmentação dos objetos (letras). Isso visa melhorar a qualidade da imagem para um melhor desempenho da técnica OCR.

Figura 3.7 – Fluxograma das etapas de processamento das imagens do código.



Fonte: Elaborado pelo autor.

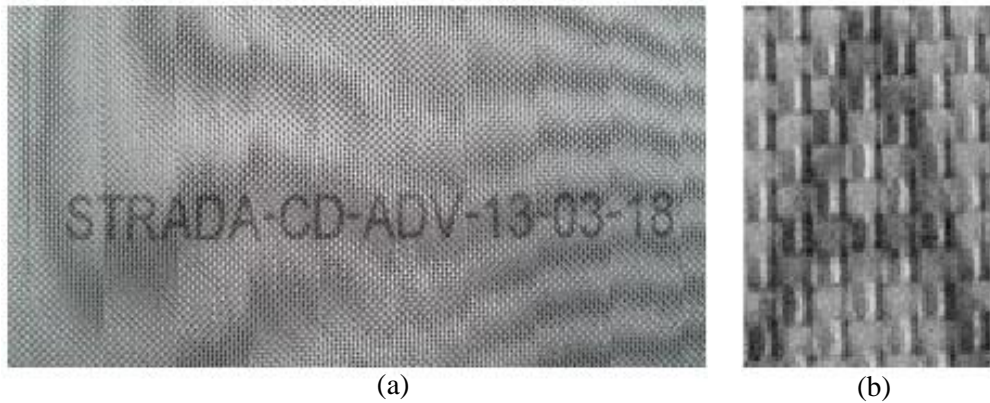
A implementação e o detalhamento das etapas de pré-processamento para identificação do código do produto são apresentados nos próximos itens.

3.4.1 Pré-processamento com filtro de média e alargamento de contraste

A Fig. 3.8a mostra a imagem do código impresso e suas particularidades. É possível notar que o tecido possui uma trama (poliéster) bem característica espalha por todo o espaço (*background*) da imagem, também apresenta níveis de reflexão da luz diferentes e isso vai

formar detalhes marcantes devido às bruscas mudanças nas intensidades. Para ilustrar esse problema foi feito um recorte da imagem separando uma letra código, a letra A em uma nova imagem, Fig. 3.8b. Analisando a imagem é possível notar claramente as variações bruscas de intensidade.

Figura 3.8 – Imagem do código impresso em tecido (a) e destaque de uma das letras (b).



Fonte: Elaborado pelo autor.

No processo de segmentação a limiarização binária baseia-se na diferença dos níveis de cinza que compõe os objetos da imagem. A partir de um limiar estabelecido, a imagem é segmentada em dois níveis, buscando assim isolar o objeto de interesse. Estas variações nos níveis de intensidade, ao final do processo de formação dos objetos por limiarização, produzem um mosaico de pixels conectados quando não deveriam, e desconexos, quando deveriam estar agregados. As transições abruptas nas intensidades são indicadores da necessidade de filtragem espacial para a suavização das variações de intensidade e consequentemente, facilitar a segmentação dos objetos.

Em determinadas situações a suavização da imagem e o alargamento de contraste são ferramentas importantes no pré-processamento da imagem. Neste trabalho a imagem do código do produto impresso em tecido, inicialmente passa por um processo de suavização e de realce de contraste antes da extração dos caracteres, visto que somente a limiarização direta (transformação binária) não é capaz de fazer a segmentação correta dos objetos, apesar da diferença de tonalidade entre o texto do código e o *background*.

Para a suavização da imagem foi utilizado um filtro de média 11x11, visualizado na Fig. 3.9, que produz uma uniformização nas intensidades dos pixels. Como discutido no

capítulo anterior, esta máscara percorre toda a matriz da imagem realizando o processamento espacial.

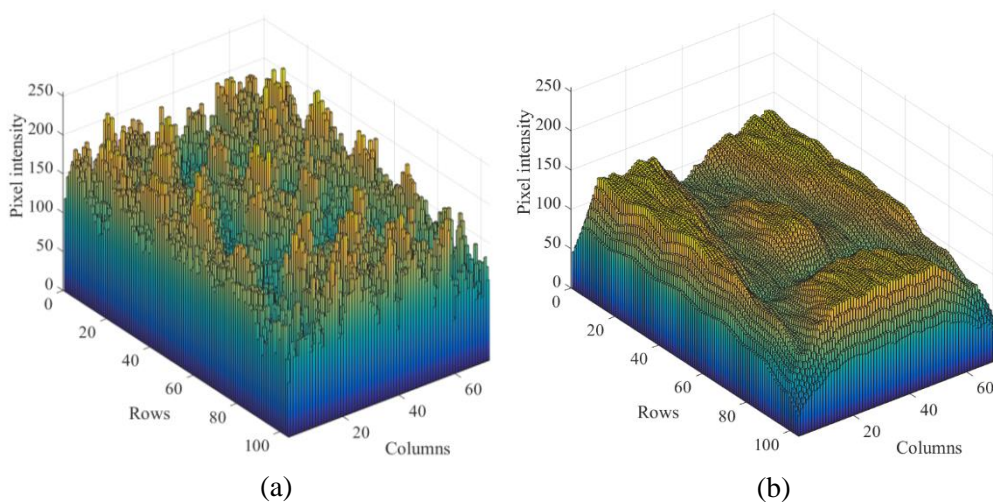
Figura 3.9 – Filtro de média utilizado para suavizar a imagem do código.

$\frac{1}{121}$	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Fonte: Elaborado pelo autor.

Neste caso, as intensidades dos pixels da imagem original, letra A, é mostrada no gráfico da Fig. 3.10.a, onde é possível notar as variações abruptas e, após o processo de suavização a imagem passa a destacar muito mais eficientemente os dois patamares de intensidade, *background* e texto impresso, Fig. 3.10b. É possível observar que após a filtragem espacial os pixels relativos ao caracteres ficam destacados dos pixels de *background*, sendo possível até mesmo “ver” o desenho da letra A.

Figura 3.10 – Intensidades dos pixels da letra “A” antes e após filtragem.

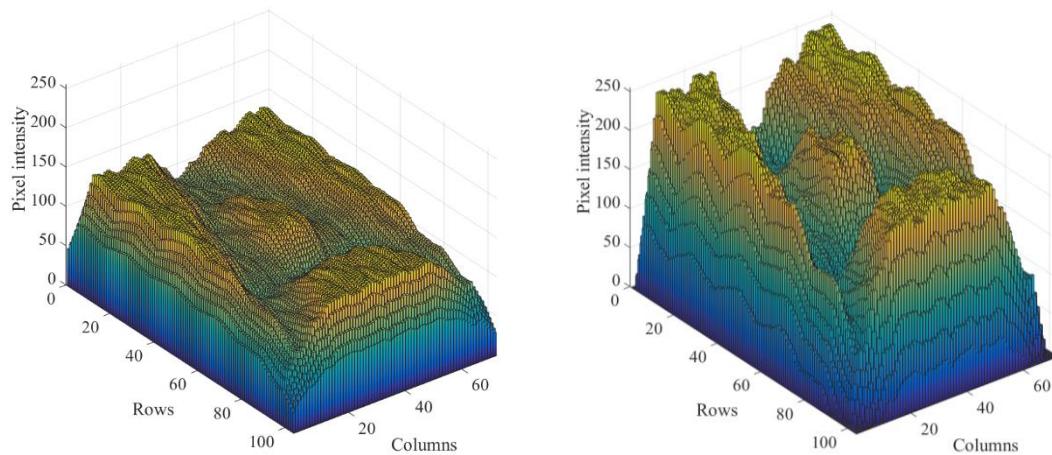


Fonte: Elaborado pelo autor.

O próximo passo é o alargamento de contraste, onde imagem suavizada é agora a imagem tratada e o resultado é uma imagem de alto contraste, conforme discutido anteriormente, em que 1% das menores intensidades e das maiores intensidades são

normalizadas, respectivamente aos níveis zero e 255. Na Fig. 3.11 são mostrados os níveis de intensidade da imagem, imagem apenas suavizada (Fig. 3.11a) e imagem de alto contraste (Fig. 3.11.b). Na imagem de baixo contraste as intensidades dos dois patamares de destaque (*background* e letra) possuem baixo desvio padrão e na imagem de alto contraste possuem alto desvio padrão em relação à média das intensidades.

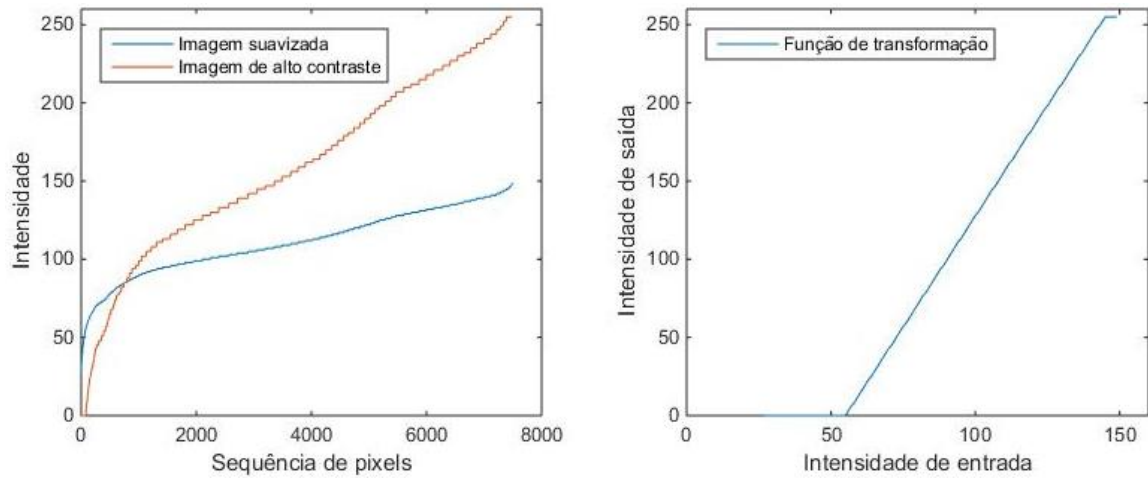
Figura 3.11 – Intensidade dos pixels da letra “A”. Imagem suavizada, à esquerda, e imagem de alto contraste, à direita.



Fonte: Elaborado pelo autor.

Na Fig. 3.12 é mostrada ainda a ordenação crescente das intensidades dos pixels da imagem apenas suavizada e da imagem de alto contraste. Estas curvas são utilizadas para traçar a função de transformação de intensidade utilizada para alargamento de contraste da imagem.

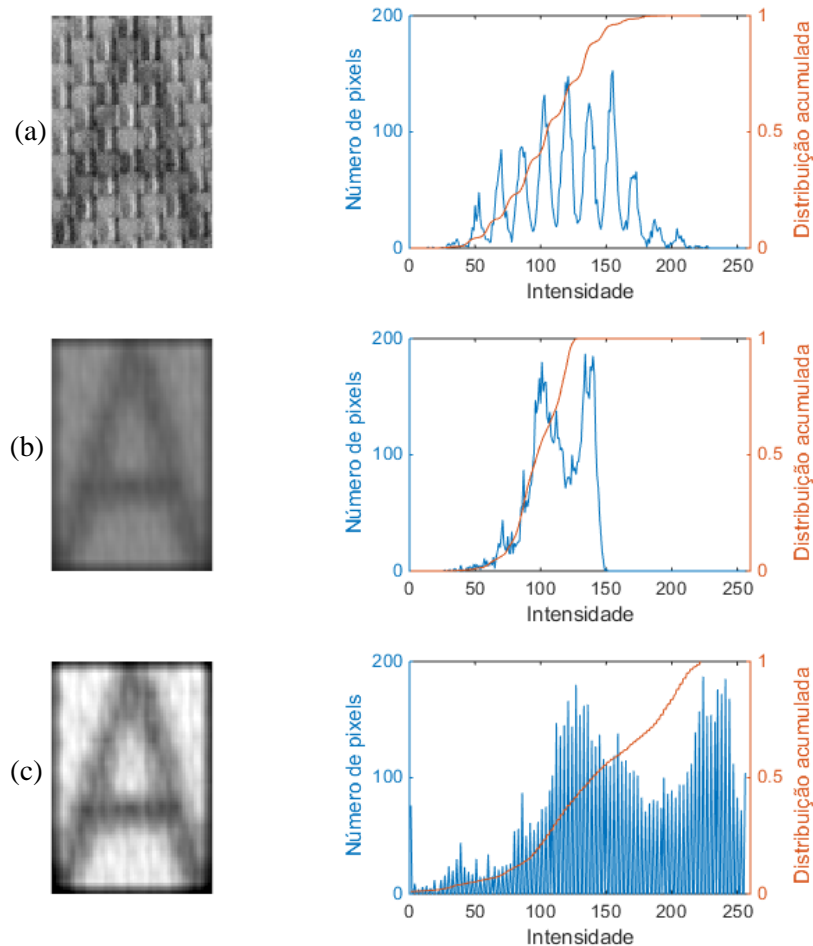
Figura 3.12 – Intensidades das imagens suavizada e de alto contraste à esquerda, e a função de transformação de intensidade para alargamento de contraste, à direita.



Fonte: Elaborado pelo autor.

Os histogramas das imagens original, suavizada e de alto contraste são mostrados na Fig. 3.13. Como pode ser observada, a filtragem espacial estreitou a escala de intensidades, no entanto, como ponto positivo, definiu muito bem dois patamares de níveis de intensidade, o nível de background e o nível da letra impressa, isso facilita a separação entre objeto e fundo, Fig. 3.13b.

Figura 3.13 – Funções de distribuição acumulada sobrepostas aos histogramas das imagens da letra A.



Fonte: Elaborado pelo autor.

Na imagem original os níveis de intensidade são bastante variáveis, já na imagem suavizada os níveis de intensidade foram separados em dois patamares distintos, da letra impressa em torno de 100 e os níveis de fundo em torno de 140. Neste caso houve uma separação da intensidade do objeto em relação à do fundo da imagem, entretanto, os patamares estão próximos um do outro. Isso pode comprometer a segmentação do objeto no processo de limiarização para gerar a imagem binária, visto que o ponto de limiar fica restrito a uma escala muito pequena (entre os dois níveis). A etapa alargamento de contraste da imagem melhora essa separação, como pode ser visto na Figura 3.13c. Os dois novos patamares estão respectivamente em torno de 120 e de 230, ou seja, distantes um do outro.

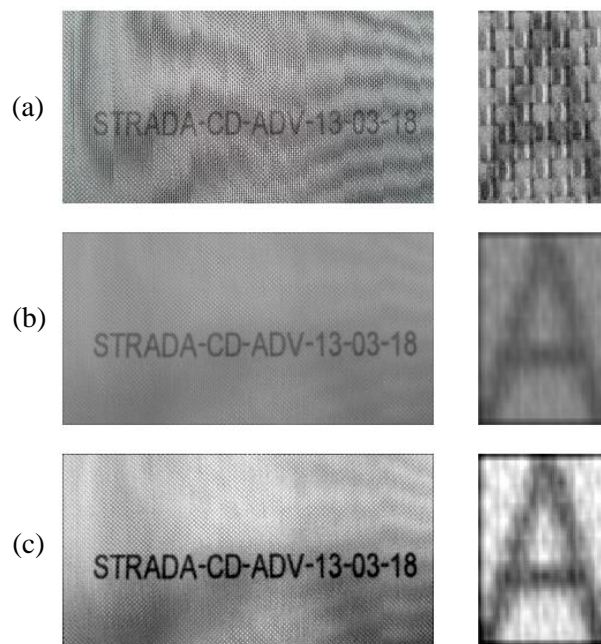
Esta diferença entre os níveis de intensidade permite uma liberdade maior para a escolha do ponto de limiar (*threshold*) no processo de segmentação, que pode ser arbitrado

dentro de uma escala maior. Em um sistema automático sujeito a variações nos parâmetros ambientais, esta liberdade é bem-vinda para o funcionamento do sistema.

De forma geral, os efeitos e dificuldades para separar o objeto de interesse em relação ao fundo da imagem (*background*) discutidos, são os mesmos quando a imagem processada contém o código do produto como um todo e não apenas uma letra como ilustrado acima.

Na Fig. 3.14 é mostrado o resultado de cada uma das etapas de pré-processamento, agora aplicadas para a imagem com código do produto. São apresentadas as imagens original, Fig. 3.14a, a imagem suavizada, Fig. 3.14b e, finalmente, a imagem com alargamento de contraste, Fig. 3.14c, sempre destacando a imagem da letra A.

Figura 3.14 – Imagem de entrada, seguida pela imagem suavizada e pela imagem com alto contraste.



Fonte: Elaborado pelo autor.

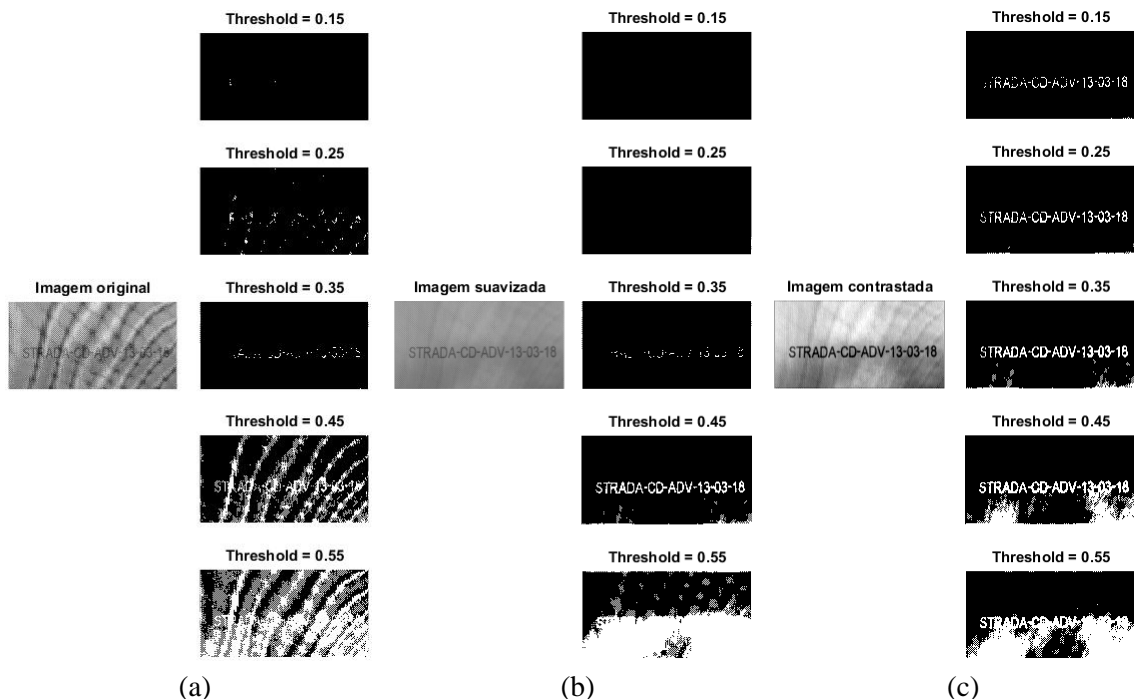
A próxima etapa do processamento busca separar o objeto de interesse do fundo da imagem, o que vai permitir extrair as letras e a numeração que identifica o modelo da capota inspecionado.

3.4.2 Segmentação e reconhecimento ótico dos caracteres

Nesta fase de segmentação os parâmetros de formação da imagem binária são o limiar de *threshold* e a filtragem de objetos binários desconexos estranhos aos objetos de interesse (letras impressas). Esta filtragem, que se diferencia da filtragem espacial para suavização estudada anteriormente, verifica o tamanho ou área de objetos conexos, eliminando aqueles que proporcionalmente possuem área muito diferente dos objetos de interesse.

No processo de segmentação, a influência da suavização e do alargamento de contraste pode interferir nos resultados levando a uma maior quantidade de objetos separados nas imagens processadas. Isso pode ser observado nas imagens da Fig. 3.15, que apresenta a binarização realizada nas imagens original, suavizada, e de alto contraste, com cinco diferentes níveis de *threshold*.

Figura 3.15 – Diferentes limiares de binarização para a imagem suavizada e para a imagem realçada.



Fonte: Elaborado pelo autor.

As cinco imagens formadas a partir da imagem original, Fig. 3.15a, mostram que não foi possível a separação do código, demonstrando a necessidade de um pré-processamento. Na imagem suavizada e possível a separação, entretanto, conforme

discutido anteriormente, a faixa de valores de *threshold* onde se consegue a segmentação é muito pequena. Nota-se que dos cinco níveis de *threshold* adotados, em apenas um a separação do código é minimamente alcançada, Fig. 3.15b. Já na imagem de alto contraste, Fig. 3.15c, nota-se a grande liberdade de pontos de limiarização, ou seja, praticamente em todo níveis de *threshold* adotados é possível observar minimamente a formação do código.

Um aspecto importante ainda no reconhecimento dos quando se utiliza a função OCR é a uniformidade dos objetos, sua forma e solidez. Portanto, após as etapas de suavização, alargamento de contraste e transformação binária, ainda é necessário uma última etapa de filtragem de objetos da imagem binária. Esta etapa realiza a remoção de pequenos objetos da imagem binária que dificultam o reconhecimento do código.

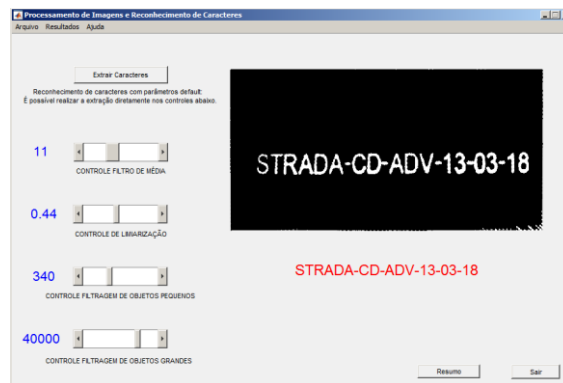
3.4.3 Teste de reconhecimento do código usando OCR

A técnica de reconhecimento de caracteres *Optical Character Recognition* (OCR) utiliza é baseada nas técnicas de correlação de imagens (*Image Correlations*) que compara o caractere segmentado com tamanho normalizado com a matriz *template* em busca da correspondência de padrões (*Pattern Matching*). Essa comparação é realizada pixel a pixel.

Nas técnicas de OCR o ajuste dos sensores e parâmetros ambientais são fundamentais para a obtenção das imagens. Fatores como luz e o posicionamento do sensor devem ser controlados de forma adequada, outra característica importante para a execução da função OCR é o posicionamento angular da câmera. Os caracteres devem permanecer sempre alinhados, de tal forma que as linhas estejam na horizontal.

Na definição dos parâmetros utilizados no processamento foi utilizada uma interface gráfica em que o usuário pode interagir facilmente com o programa, selecionando os parâmetros e observando os resultados. A Fig. 3.16 mostra esta interface implementada.

Figura 3.16 – Teste para validação do reconhecimento dos códigos utilizados.



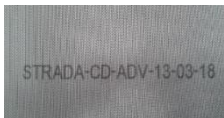

Fonte: Elaborado pelo autor.

Inicialmente o usuário carrega a imagem e define os valores dos parâmetros de controle, dimensão do filtro de média, borramento (suavização) e o controle de limiarização, ou seja, determina o nível de intensidade em que os pixels abaixo normalizados para zero e os acima em um. Há ainda dois controles da filtragem de objetos binários, o primeiro realiza a “limpeza” de objetos com áreas de tamanhos pequenos em relação às áreas dos caracteres (letras). Já o segundo controle filtra os objetos com áreas maiores que os caracteres, entretanto, com um bom ajuste da luminosidade esta etapa é praticamente desnecessária. O alargamento de contraste com a saturação de 1% dos pixels de maiores intensidades e 1% dos de menores intensidades e realizado automaticamente.

Os resultados são mostrados diretamente na tela, acima a imagem do condigo e logo abaixo, em vermelho, a *string* do código identificada pelo programa.

Na tabela 3.1 é mostrado o resultado da identificação do condigo sem o pré-processamento.

Tabela 3.1 – Resultados de OCR com entrada sem pré-processamento.


<i>Imagem original</i>	<i>Imagem binária</i>	<i>Texto extraído</i>
		'JJk'tfi>r'ri:' 'll}JTJI'

Fonte: Elaborado pelo autor.

Neste caso, o controle da dimensão da máscara de suavização é colocado na posição inicial, ou seja, a máscara do filtro passa a ter tamanho *um* e desta forma, não executa nenhuma transformação na imagem original. A *string* identifica não apresenta nenhuma correspondência com o código.

Na tabela 3.2 é mostrado o resultado utilizando o controle de suavização em conjunto com o alargamento de contraste. Neste caso, a dimensão da máscara de suavização é ajustada juntamente com os controles de limiarização e filtragem de objetos binários e a imagem mostrada no *display* é instantaneamente atualizada.

Tabela 3.2 – Resultados de OCR com entrada pré-processada.

<i>Imagem pré-processada</i>	<i>Imagem binária filtrada</i>	<i>Texto extraído</i>
		'STRADA-CD-ADV-13-03-18' 'STRADA-CD-ADV-'13-03-18'

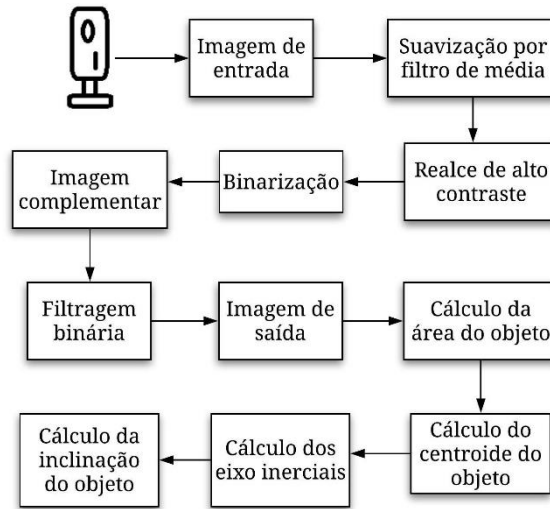
Fonte: Elaborado pelo autor.

Com a configuração de parâmetros ajustados de acordo com a Figura 3.16 a *string* identifica corresponde exatamente ao código do produto capturado na imagem.

3.5 IDENTIFICAÇÃO DAS PEÇAS UTILIZADAS NA MONTAGEM DO PRODUTO

O fluxograma das etapas de processamento para extração de características dos componentes de fixação é mostrado na Fig. 3.17. Assim como no item anterior, inicialmente é feito o pré-processamento com a filtragem espacial para suavização das imagens e o alargamento de contraste para realçar as intensidades e posterior segmentação dos objetos, neste caso, às formas geométricas das peças.

Figura 3.17 – Fluxograma das etapas de processamento das imagens dos componentes.



Fonte: Elaborado pelo autor.

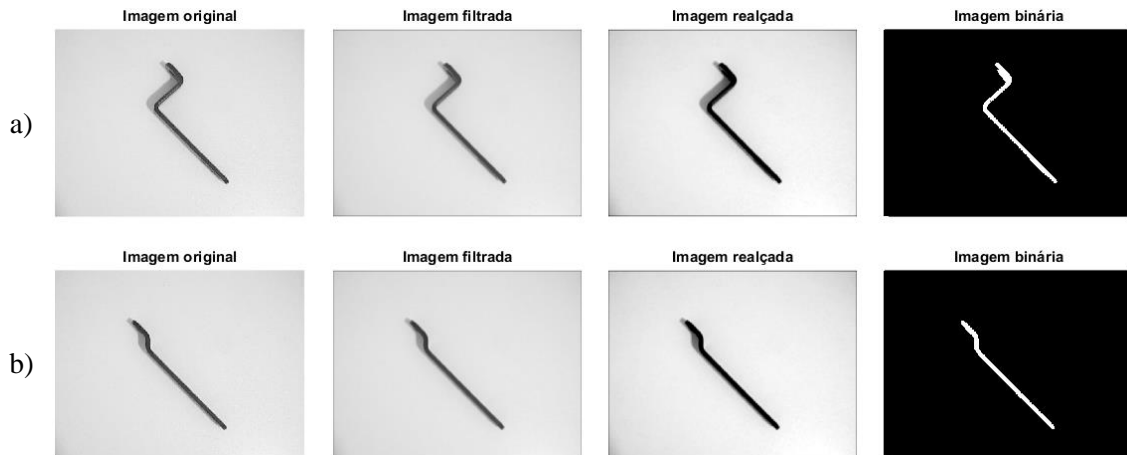
Uma vez realizado o processamento, são identificados os descritores de área, eixos inerciais, razão dos eixos, centroide, elipse, e orientação dos componentes de fixação daquele modelo específico.

3.5.1 Pré-processamento das imagens das peças

Neste caso, o pré-processamento das imagens não exige o nível de controle dedicado às imagens do código do produto, importante salientar que segmentar formas planas e perfis para extração de invariantes de momento é significativamente mais espontâneo que a segmentação dos caracteres impressos no tecido para reconhecimento ótico dos mesmos. Isso ocorre devido ao contraste natural entre as peças em preto e a bancada que tem a superfície branca.

As peças possuem um perfil bem definido e as imagens resultam em objetos cujas propriedades são igualmente bem definidas pelos invariantes de momento, possibilitando assim determinar claramente qual é a peça. A Fig. 3.18 ilustra as etapas de processamento utilizadas para a binarização da imagem e segmentação dos componentes de fixação. A Fig. 3.18a apresenta a imagem original, a imagem suavizada, imagem com realce de contraste e finalmente a imagem binária, relativas ao componente de perfil mais largo. Já na Fig. 3.18b, a mesma sequência de imagens é apresentada, relativas à peça de perfil mais delgado.

Figura 3.18 – Peças de aço estampadas usadas no experimento.



Fonte: Elaborado pelo autor.

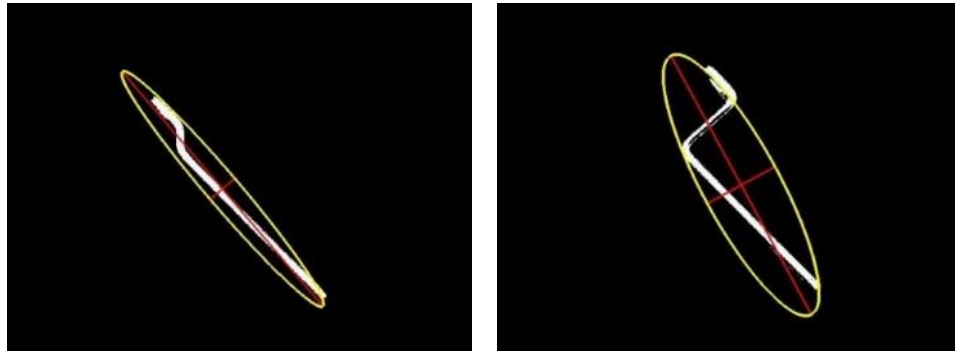
Para gerar as imagens dos componentes de fixação, eles são colocados na região de captura da segunda câmera de tal maneira que seus perfis caracterizantes estejam no plano da cena.

3.5.2 Extração dos momentos descritores

Uma vez obtida a imagem binária, com a separação do objeto (pixels um) do *background* (pixels zero) os momentos de ordem zero, momentos de primeira e segunda ordem além dos momentos centrais de segunda ordem são calculados visando definir os eixos principais de inercia da peça.

Os descritores mais importantes nesta aplicação são os relativos aos eixos dos objetos, mostrados na Fig. 3.19, juntamente com sua elipse característica. Nesta figura são mostradas as duas peças utilizadas no experimento já segmentadas e com a representação dos seus respectivos descritores, eixos inerciais e elipse, sobrepostos a sua forma.

Figura 3.19 – Eixos inerciais e correspondentes elipses das peças testadas.

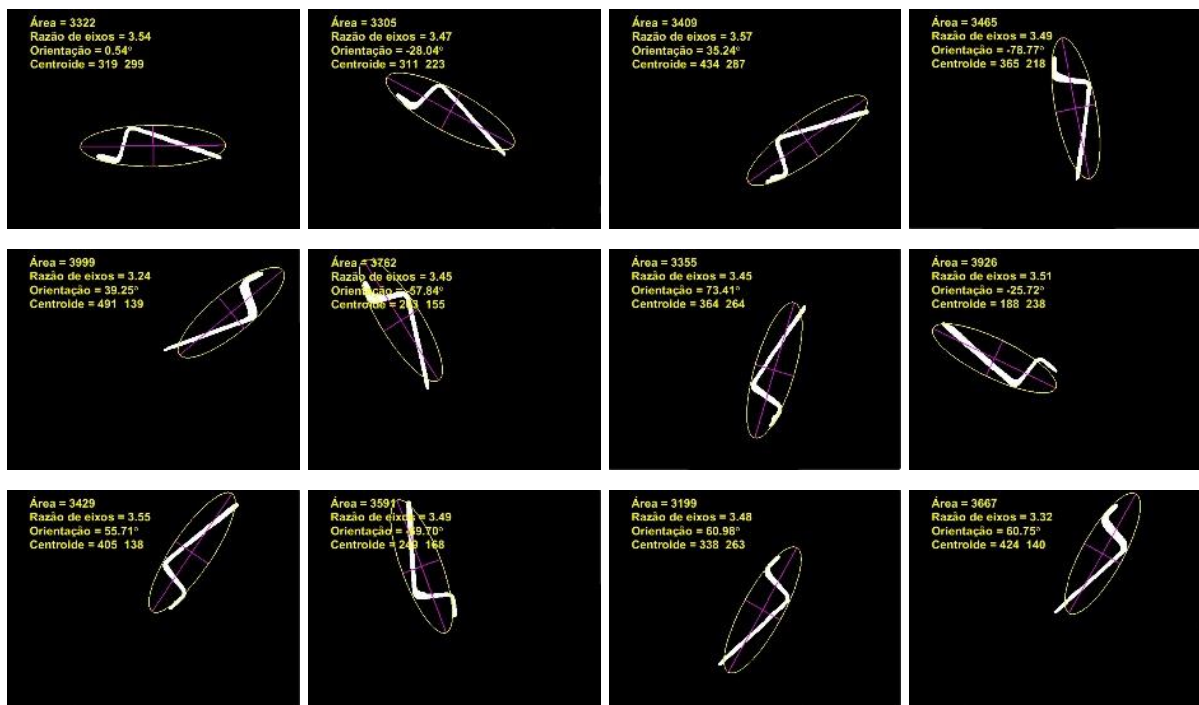


Fonte: Elaborado pelo autor.

Os eixos das elipses são elementos que representam bem as características das peças, de forma que, a identificação de cada componente será feita com base nas medidas dos eixos inerciais, mais especificamente, será empregada a razão entre os eixos da elipse. Vários testes foram realizados capturando a imagem com a peça posicionada na cena de diferentes formas e esta relação ser mostrou como um parâmetro bem estável.

Na Fig. 3.20, são mostrados os valores dos descritores de área, eixos inerciais, orientação e centroide, obtidos a partir de uma sequência de imagens do componente com o perfil mais largo. A sequência de 12 imagens (*frames*) foi gerada movimentando a peça na bancada e capturando a imagem a cada movimento.

Figura 3.20 – Sequência de *frames* do componente de fixação de perfil largo.



Fonte: Elaborado pelo autor.

Os descritores área e razão de eixos independem do posicionamento da peça, permanecem vinculados ao objeto e não variam com a posição da peça, ou seja, são os mesmos a cada *frame*. Já os descritores de centroide e de orientação são dependentes de como as peças são posicionadas na bancada (posição e/ou orientação) e apresentam valores diferentes a cada *frame* e, para esta aplicação específica, não são úteis para a identificação do componente.

Os valores dos descritores de área e eixos inerciais não variam, entretanto, estão sujeitos a oscilações que ocorrem por se tratar de objetos tridimensionais capturados no plano e devido ao ruído intrínseco ao processo de digitalização. Na Tabela 3.3 são mostrados os valores médios obtidos para os descritores e os respectivos desvios, considerando as 12 posições (*frames*) analisadas.

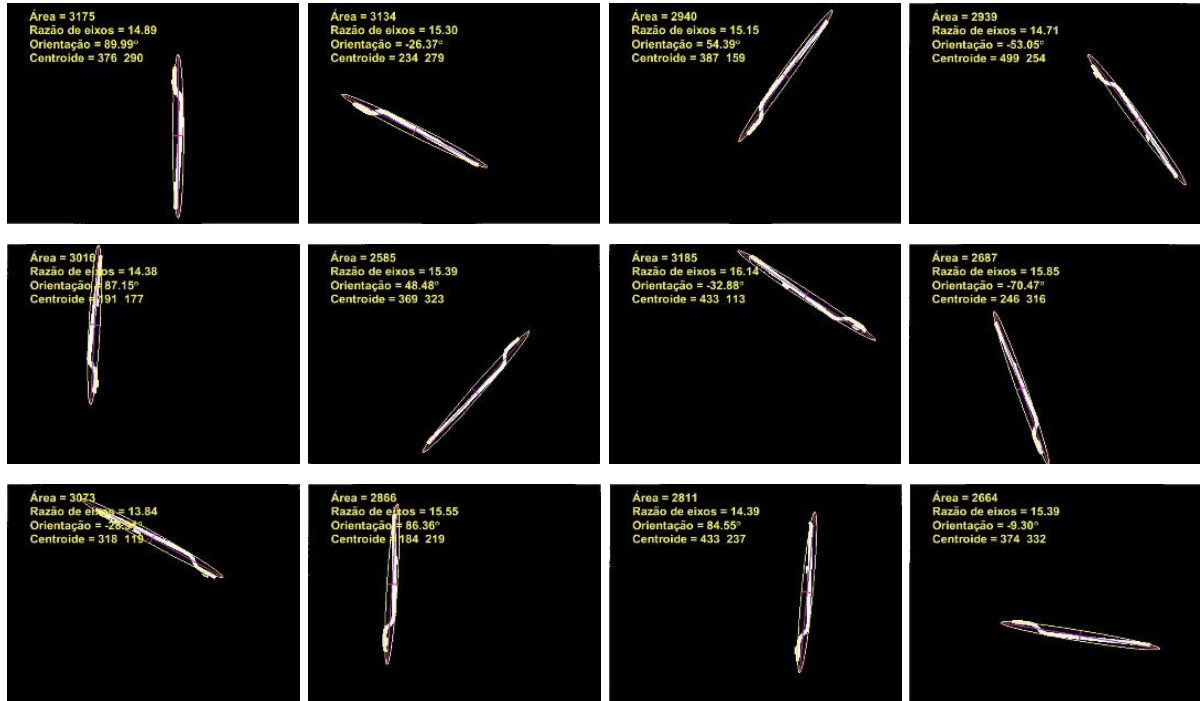
Tabela 3.3 – Medidas de área e razão de eixos para a peça perfil largo.

	Média	Desvio Padrão	Desvio %
Área	3535,75	255,10	3535,75 ± 7,21%
Razão de eixos	3,46	0,10	3,46 ± 2,89%

Fonte: Elaborado pelo autor.

O mesmo teste discutido acima foi realizado para o componente com perfil delgado. Na Fig. 3.21, é mostrado os valores dos descritores obtidos a partir da sequência de imagens do componente.

Figura 3.21 – Sequência de *frames* do componente de fixação de perfil delgado.



Fonte: Elaborado pelo autor.

Os valores médios obtidos para os descritores e os respectivos desvios são mostrados na Tabela 3.4.

Tabela 3.4 – Medidas de área e razão de eixos para uma das peças.

	Média	Desvio Padrão	Desvio %
Área	2922,92	204,76	2922,92 ± 7,01%
Razão de eixos	15,08	0,66	15,08 ± 4,41%

Fonte: Elaborado pelo autor.

Os resultados obtidos nos testes mostraram que a razão entre eixos praticamente independe de como uma peça é posicionada na cena, mas a diferença da razão entre eixos de uma peça e a razão dos eixos da outra peça é muito grande. Na peça perfil largo a média deste descritor (razão de eixos) é 3,46 com um desvio de 0.10, enquanto na peça perfil delgado é de 15,08 com uma variação de 0,66. Consequentemente, esse é um descritor bastante adequado para identificar o componente de fixação que está sendo usado na montagem do produto.

3.6 VERIFICAÇÃO E IDENTIFICAÇÃO DA CONFORMIDADE

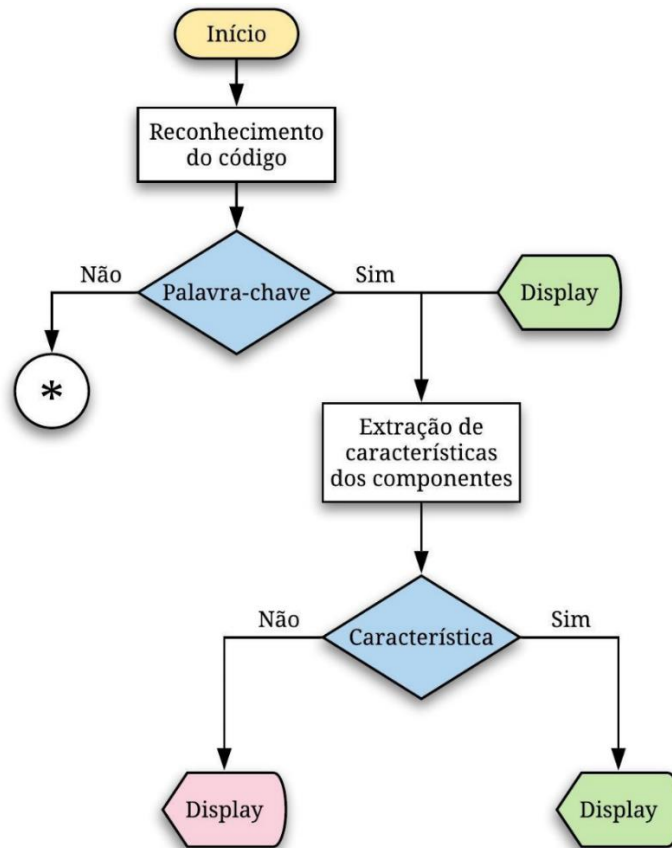
Nesta etapa do processamento, após identificação do código do produto e os descritores dos componentes de fixação, é verificada a conformidade do produto. Isso é feito correlacionando os descritores dos componentes de fixação com o código do produto.

Nos testes realizados foram avaliados cinco diferentes códigos, ou seja, cinco diferentes modelos do produto. Cada código é identificado por um conjunto de caracteres alfanuméricos: “STRADA-CE-ADV-DD-MM-AA”, “STRADA-CE-TK-WK-DD-MM-AA”, “STRADA-CC-TK-WK-DD-MM-AA”, “STRADA-CD-TK-WK-DD-MM-AA” e “STRADA-CD-ADV-DD-MM-AA”.

O texto dos códigos, contém vários caracteres que são os mesmos de um modelo para o outro, portanto, para identificação de cada modelo basta localizar aqueles caracteres que são diferentes, denominados aqui de palavras-chave. Neste caso, as palavras-chave a serem identificadas no texto do código são “CE”, “CC”, “CD”, “TK”, “WK” e “ADV”. Uma vez identificadas as palavras-chave, ou seja, o código do produto, então é verificado se os descritores do componente de fixação são compatíveis com aquele modelo. Caso afirmativo o produto é conforme, caso contrário o produto é defeituoso.

Para exemplificar este processo, a Fig. 3.22 ilustra fluxograma de processamento que permite a realização da inspeção de conformidade. Após o reconhecimento de um dos códigos, o algoritmo verifica se a medida da razão de eixos do objeto segmentado relativo ao componente está dentro de um intervalo específico, compatível com os desvios resumidos nas tabelas anteriores e ajustado na implementação do código.

Figura 3.22 – Fluxograma de ramificação para decisão de alerta de conformidade.



Fonte: Elaborado pelo autor.

No fluxograma, o conector com “asterisco” ilustra a busca por todas as palavras-chave definidas previamente, quando o processamento não encontra as palavras-chave, o algoritmo retorna ao início para nova captura de imagem e reconhecimento de caracteres.

3.6.1 Teste de bancada

O teste implementado para verificação da conformidade do produto permite a visualização gráfica, através de alertas verde e vermelho, verificar se o produto se apresenta em boas condições ou apresenta alguma não conformidade. Uma vez calibrado o sistema, o mesmo não oferece opções de controle e definição de parâmetros pelo usuário. O sistema possui botões para a ativação das câmeras, e um botão específico para ativação do *loop* de processamento. Todas as definições dos parâmetros de suavização, contraste, limiarização e limpeza de objetos binários, são implementadas via código.

Para ilustrar o funcionamento da interface e a apresentação dos resultados gráficos do processo de identificação, correlação e associação dos componentes de fixação com o código do produto é testada uma situação onde são testados os códigos STRADA-CD-ADV-DD-MM-AA e STRADA-CE-TK-WK-DD-MM-AA com os dois componentes de fixação associados.

No teste, o sistema de reconhecimento é acionado e passa a operar de forma contínua e, tanto as duas amostras de lona com os códigos, como os dois componentes de fixação são trocados indistintamente ao longo do teste.

A Fig. 3.23a mostra a situação onde o código “CD ADV” foi reconhecido e o componente de fixação presente sobre a bancada naquele momento era o componente correto para aquele produto. Já a Fig. 3.23b mostra a situação onde o código “CD ADV” foi reconhecido e o componente de fixação foi trocado, portanto, o sistema ativa na interface o sinal de alerta vermelho.

Ainda com o sistema rodando, a amostra do código foi trocada pela outra contendo o código “CE TK WK”. Neste momento, os três alertas mudam de *status*, o alerta de informação do modelo é ativado para o código recém reconhecido, o alerta do modelo “CE ADV” é apagado, e finalmente, o alerta de conformidade é ativado em verde, visto que o componente é o correto, Fig. 3.23c. Já na Fig. 3.23d é mostrada a situação em que o componente de fixação é trocado pelo outro, e conseqüentemente o sistema ativa na interface o alerta vermelho.

Figura 3.23 – Interface simplificada para teste em bancada.



Fonte: Elaborado pelo autor.

Os testes realizados e discutidos neste capítulo foram realizados em uma bancada de teste. Os resultados obtidos mostraram que é possível realizar a inspeção dos componentes isoladamente e que o cuidado em relação à extração do código impresso em tecido é fundamental para o sistema desempenhar sua função de forma satisfatória. Foi necessário um controle criterioso de luz incidente sobre o tecido, e também do posicionamento do sensor em relação ao plano do tecido.

Durante os testes foi também contabilizado o tempo entre as leituras dos códigos e a maior velocidade de reconhecimento foi de 1,2 segundos por extração. Este tempo elevado decorre, principalmente, da abordagem utilizada na programação, que executa a ativação e desativação da câmera durante a captura das imagens. Entretanto, esse tempo quando comparado com o tempo gasto no processo em que o produto fica fixado ao dispositivo é muito pequeno, ou seja, o tempo de reconhecimento e verificação da conformidade do produto atende claramente os propósitos do processo de inspeção.

4 PROTÓTIPO DO DISPOSITIVO DE VISÃO DE MÁQUINA PARA INSPEÇÃO DE CONFORMIDADE DO PRODUTO

Neste capítulo é apresentada a proposta de implementação do protótipo do sistema de visão de máquina na linha de montagem juntamente com os testes de avaliação de desempenho do sistema. O protótipo consiste basicamente em um aparato estrutural utilizado para a montagem das capotas (produto), de duas câmeras, e do sistema de *hardware/software* utilizado para a aquisição, processamento e extração de características das imagens e verificação da conformidade de montagem do produto.

4.1 SETUP EXPERIMENTAL

O Sistema de Visão de Máquina consiste em um dispositivo metálico com suportes para fixação das câmeras, uma plataforma para a CPU juntamente com o monitor e periféricos e uma base com grampos de fixação para a montagem do produto (capota marítima). O protótipo do Sistema foi projetado e construído de forma que o mesmo pudesse ser integrado na linha de produção do produto. Na Fig. 4.1 é mostrada uma foto do dispositivo metálico utilizado na montagem do Protótipo do Sistema de Visão Computacional.

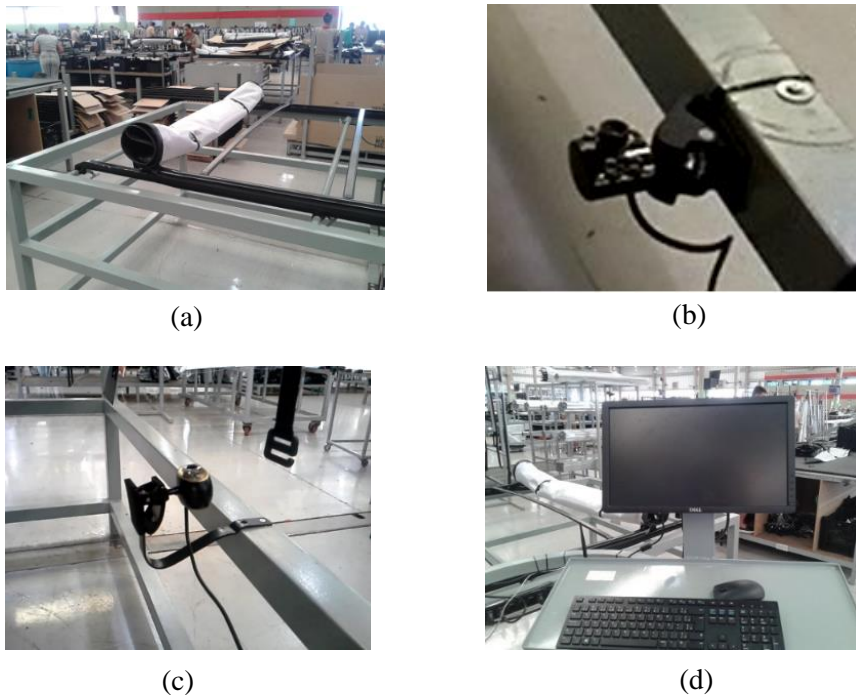
Figura 4.1 – Estrutura do dispositivo de inspeção.



Fonte: Elaborado pelo autor.

O Protótipo foi posicionado no fluxo da linha de montagem em uma etapa anterior à etapa de embalagem e expedição, dado que o objetivo seria a verificação final da conformidade da montagem do produto evitando assim a expedição de produtos não conformes. Na Fig. 4.2 são mostradas algumas imagens do Protótipo ilustrando como a capota seria estendida na estrutura, como as duas câmeras estão posicionadas, inclusive o computador que vai gerenciar o sistema de inspeção.

Figura 4.2 – Componentes do dispositivo de inspeção.



Fonte: Elaborado pelo autor.

Na Fig. 4.2a é mostrado a estrutura de instalação e fixação da capota, na Fig. 4.2b é mostrada a posição de fixação da câmera utilizada para captura das imagens dos componentes de fixação, na Fig. 4.2c é mostrada a câmera para a captura do código do produto e na Fig. 4.2d o computador utilizado para gerenciamento do sistema. O desenho detalhado da estrutura do dispositivo é apresentado no Apêndice C.

Um dos problemas encontrados na montagem e posicionamento das câmeras foi a questão da incidência de luz no sensor contra a direção de formação da cena, esta luz excita o silício presente no sensor de forma desproporcional, “apagando” elementos de interesse na cena.

Na disposição construtiva do dispositivo as câmeras são posicionadas em direções que possuem sentido de baixo para cima, devido a localização dos alvos, que estão posicionados na parte inferior do produto. Portanto, essa exposição do elemento sensor da câmera deve ser evitada de tal forma que a luz não incida diretamente e, tão pouco, após ser refletida no poliéster. Esse problema foi resolvido com uma placa fixada na parte frontal do dispositivo que funciona como barreira da luz incidente no sensor da câmera que faz a captura dos elementos de fixação, Fig. 4.3.

Figura 4.3 – Detalhe da placa de barreira de luz.



Fonte: Elaborado pelo autor.

A proposta é uma inspeção total em que toda capota produzida é inspecionada ao final da linha de montagem, antes da etapa de embalagem e expedição. Para isso, a capota é montada na estrutura metálica e as câmeras capturam a imagem do código do produto e a imagem dos respectivos elementos de fixação da capota na carroceria. Como na bancada de teste, a primeira câmera é apontada para a região da lona que contém as informações do código do produto (código impresso) e a segunda câmera é direcionada para o conjunto de elementos de fixação.

A área de captura da câmera, no caso do código do produto, contempla pequenas variações na posição da impressão do código na lona. Essas variações não comprometem a identificação correta do código quando a câmera é posicionada de forma correta, assim, a instalação da câmera junto à estrutura do dispositivo deve considerar a correta formação da área de captura. Essas variações ocorrem pelo fato de a impressão do código ser feita de forma manual.

Para a captura da imagem do conjunto de elementos de fixação o processo é ligeiramente diferente daquele utilizado na bancada de testes discutida no capítulo 3. Na bancada, os elementos de fixação foram posicionados livremente sobre ela, enquanto que no protótipo do dispositivo de visão computacional, o produto já está pronto e a imagem do conjunto de elementos de fixação é capturada sem a variação de posicionamento.

O sistema de câmeras e *hardware* de aquisição é controlado pelo computador, o operador inicialmente faz o ajuste e a calibração do sistema, definindo o posicionamento das câmeras, o ajuste de foco e dos parâmetros de processamento a serem utilizados no processo das imagens, e a partir desse momento, o sistema está pronto operar de forma autônoma.

O processo de inspeção como um todo, envolve várias etapas, desde a aquisição e processamento das imagens até decisão final da conformidade ou não produto. O gerenciamento do processo de inspeção é feito através de um *software* que contém várias interfaces gráficas que permitem ao usuário fazer o acompanhamento de cada etapa processo.

4.2 INTERFACE DE CONTROLE E GERENCIAMENTO DO SISTEMA DE VISÃO COMPUTACIONAL

O gerenciamento das várias etapas do processo de inspeção é feito computacionalmente através de uma interface gráfica com várias janelas que permitem ao usuário acompanhar as etapas do processo de inspeção, desde a captura das imagens até a verificação final da conformidade do produto.

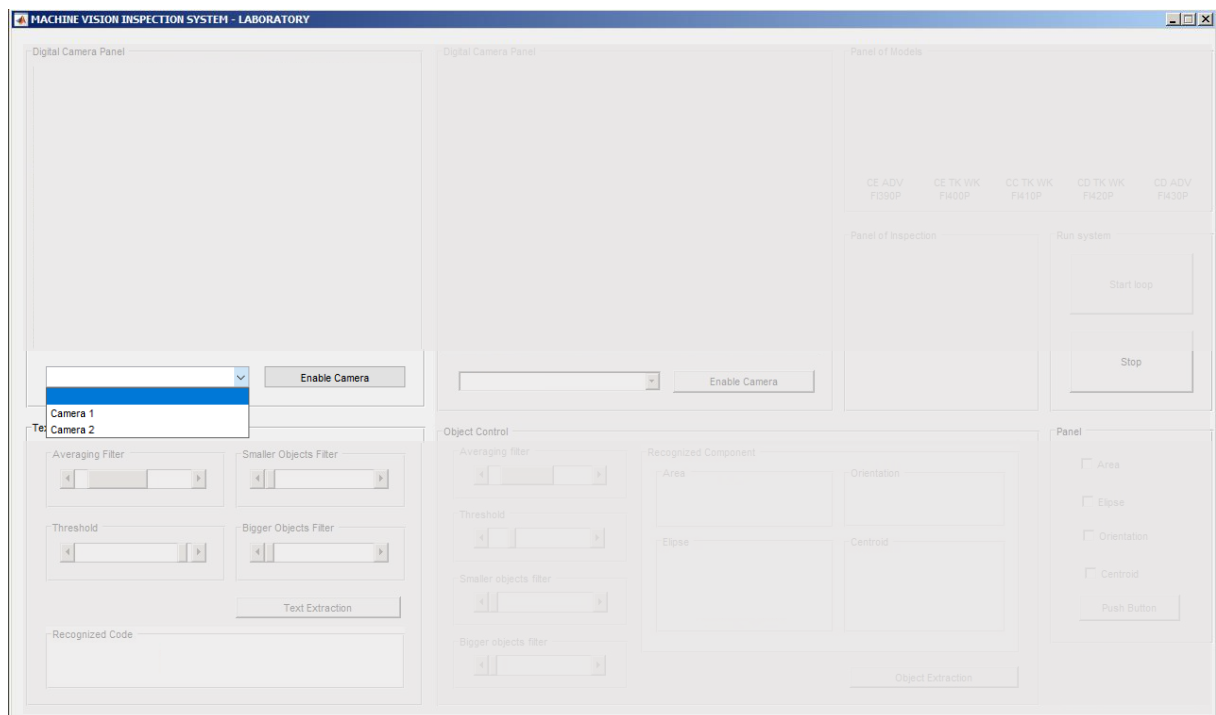
O fluxograma das etapas de controle do processo é praticamente o mesmo discutido no capítulo anterior. Inicialmente é feita a captura das imagens, o reconhecimento do código, a extração das características das peças de fixação, posteriormente, é estabelecida correlação do código com o respectivo conjunto de elementos de fixação e, finalmente, a emissão de um sinal de alerta. No caso, sinal verde para o produto conforme e sinal vermelho para o produto não conforme, ou seja, quando ocorreu algum erro ou troca na montagem do produto.

A interface gráfica está separada basicamente em três módulos, no primeiro módulo é feita a seleção e os testes de verificação e especificação dos parâmetros das câmeras para

a aquisição das imagens, no segundo módulo é feito o processamento e extração dos parâmetros e finalmente no terceiro módulo é verificada a conformidade ou não do produto.

Nas próximas figuras são apresentadas e discutidas as principais telas gráficas utilizadas na inspeção. Inicialmente o usuário seleciona e ativa as câmeras para a captura e avaliação das imagens, conforme ilustrado na Fig. 4.4.

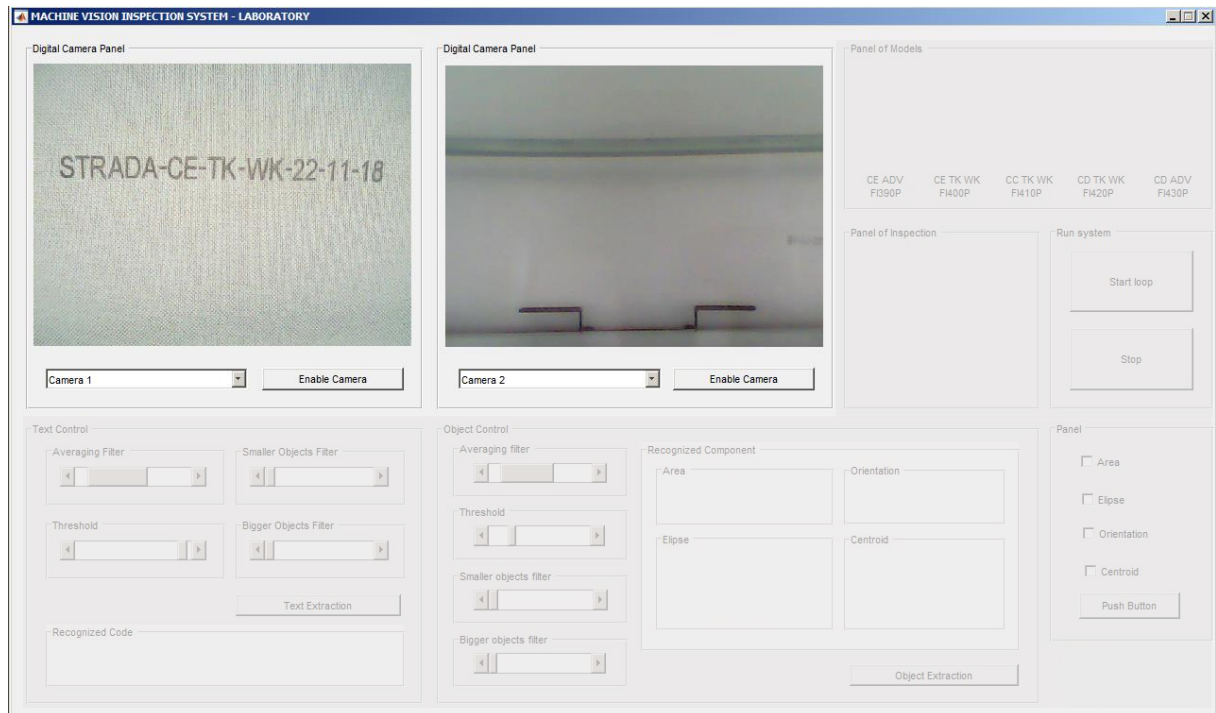
Figura 4.4 – Controle de menu *popup* para escolha da câmera.



Fonte: Elaborado pelo autor.

Quando o usuário escolhe a câmera 1 ou câmera 2 pressionando o botão *Enable Camera*, imediatamente o *preview* da imagem (imagem *live*) é mostrada na interface gráfica, conforme ilustrado na Fig. 4.5. Neste caso ambas as câmeras foram selecionadas, câmera 1 para captura do código e câmera dois para captura dos elementos de fixação.

Figura 4.5 – Imagens *live* das câmeras em painéis disponibilizados na interface.



Fonte: Elaborado pelo autor.

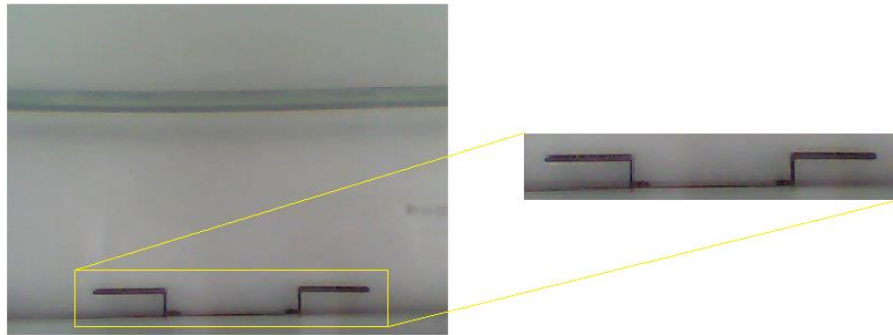
Agora, com *preview* disponível, o usuário pode avaliar a qualidade das imagens e ajustar o foco e o direcionamento das câmeras para a obtenção da melhor imagem. Neste sistema, esse processo é feito manualmente.

Uma vez definida a qualidade das imagens é a feita aquisição e o pré-processamento das imagens e, posteriormente, o processamento e reconhecimento/extração de informações das imagens.

Na etapa de pré-processamento, os elementos de fixação são capturados com o produto já montado na estrutura do dispositivo e, conseqüentemente, a posição do objeto (elementos de fixação) na cena da imagem é sempre a mesma, isso possibilitou um processamento prévio que reduz o tamanho da imagem original. Desta forma, apenas a região de interesse da imagem, *ROI (region of interest)* é processada.

Neste caso, a imagem original de tamanho 640x480 pixels é transformada em uma imagem de 480x150 pixels através de uma operação simples, onde apenas parte das linhas e colunas da matriz original da imagem é utilizada para formar a nova imagem, Fig. 4.6. A vantagem deste pré-processamento é que a região que possui relevância é separada da imagem original, e apenas esta região, que contém os elementos de fixação, é utilizada.

Figura 4.6 – Extração da Região de Interesse da imagem original.

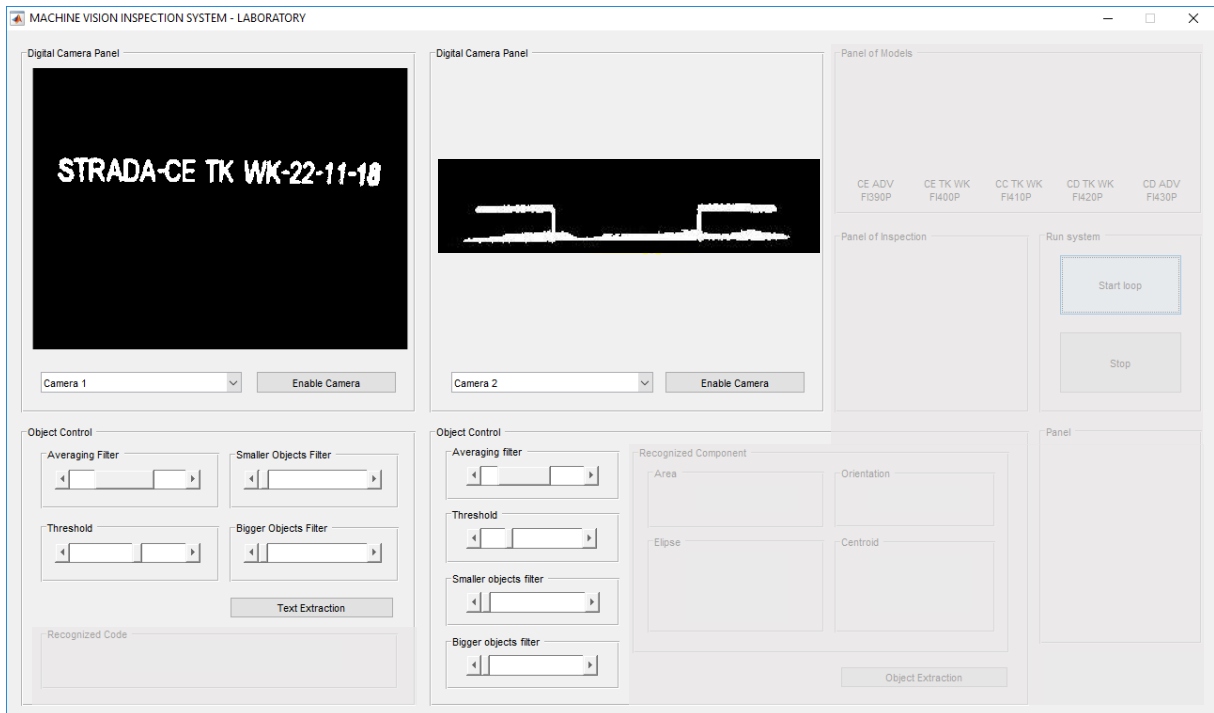


Fonte: Elaborado pelo autor.

Na etapa de processamento existem várias ferramentas que o usuário pode utilizar inicialmente durante a calibração e definição dos parâmetros de processamento e extração de características do produto. Na calibração o usuário pode controlar a dimensão do *kernel* que regula a quantidade de suavização ou borramento a ser executado na imagem através da filtragem de média, pode definir o limiar de binarização, assim como a área de objetos binários a serem filtrados.

O filtro de objetos binários possui dois controles pelos quais objetos com áreas menores que um determinado valor preestabelecido são removidos da imagem. Da mesma forma, objetos com áreas maiores que o valor preestabelecido também são eliminados. O controle e seleção dos parâmetros de processamento são feitos a partir barras de rolagem (*Slides*), onde o usuário facilmente altera os valores dos parâmetros e o resultado é instantaneamente atualizado na tela. Na fig. 4.7 é mostrado o resultado da etapa de binarização das imagens, tanto para o condigo do produto, como dos elementos de fixação.

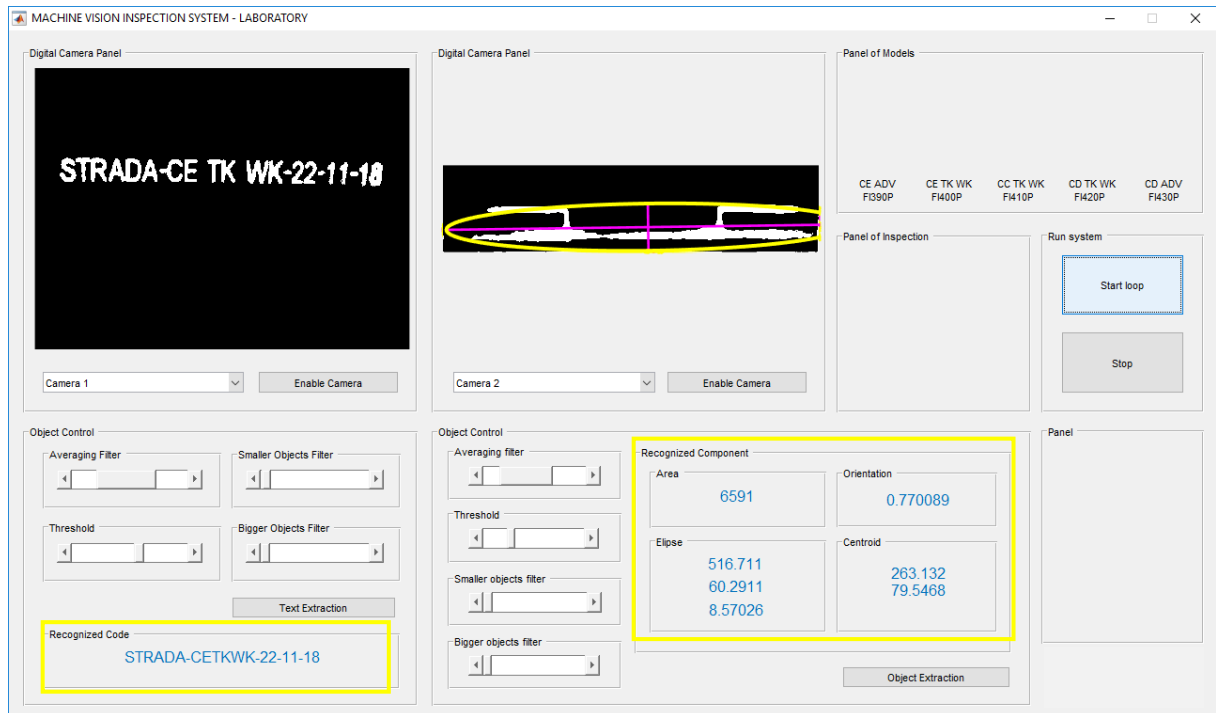
Figura 4.7 – Controles para segmentação disponíveis na interface.



Fonte: Elaborado pelo autor.

O reconhecimento ótico de caracteres e os valores dos descritores também são mostrados na interface em painéis específicos, um para o código do produto e outro para os descritores dos elementos de fixação. Assim como na etapa anterior, o usuário pode controlar os valores dos parâmetros por barras de rolagem e acompanhar os resultados que são imediatamente atualizados na tela. Na Fig. 4.8 é mostrado os resultados destacados em amarelo, tanto do reconhecimento do código, painel *Recognized Code*, como dos descritores do elemento de fixação, painel *Recognized Component*.

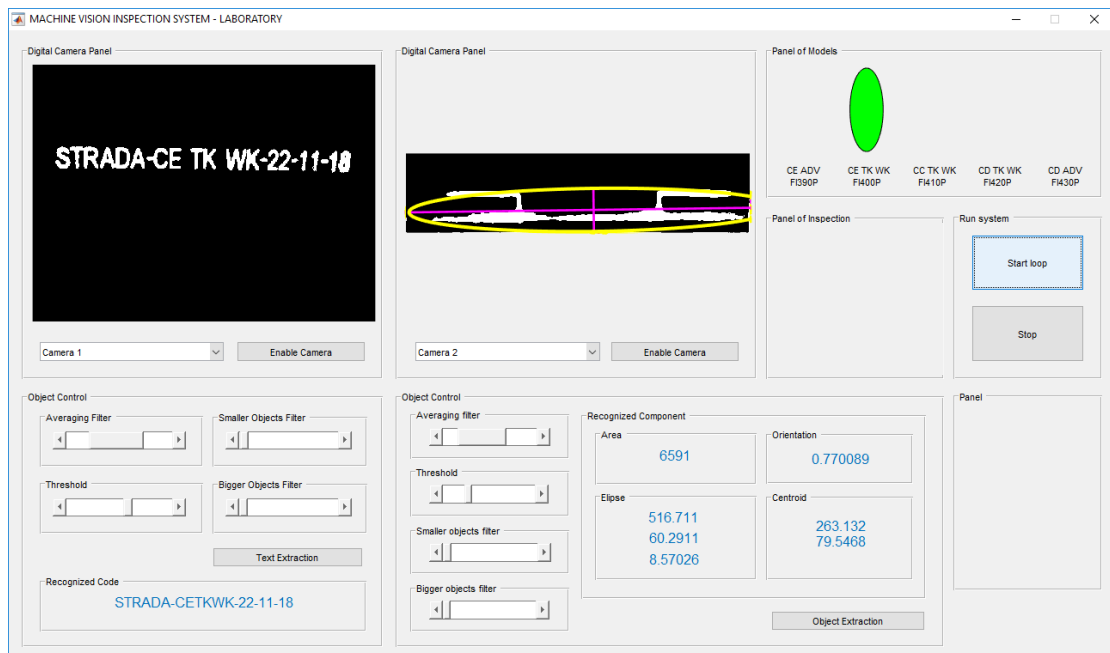
Figura 4.8 – Resultados do reconhecimento do código e dos elementos de fixação.



Fonte: Elaborado pelo autor.

Uma vez segmentados os objetos referentes ao código, os mesmos são definidos como a entrada da função de OCR utilizada para a extração da *string* contendo o texto do código. O texto do código contém todas as informações do produto e, juntamente com os descritores dos elementos de fixação (tamanho), permite verificar a conformidade ou não do produto. No processo como um todo são identificados cinco diferentes tipos de produtos (códigos), o sistema primeiro verifica se o código identificado é o código de um dos cinco produtos e ativa no painel, *panel of models*, na cor verde àquele produto em específico, Fig. 4.9.

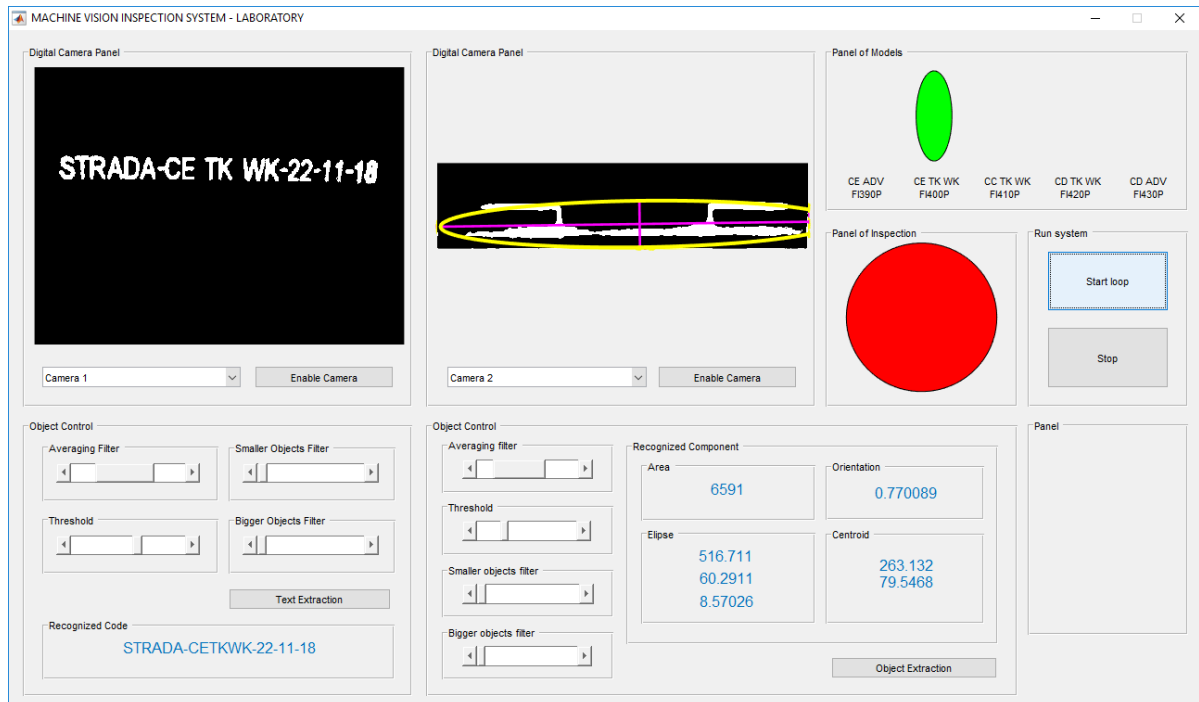
Figura 4.9 – Interface do sistema de inspeção com identificação do modelo.



Fonte: Elaborado pelo autor.

Em seguida, o sistema verifica se os elementos de fixação identificados com base nos valores dos seus respectivos descritores são os elementos de fixação correspondentes a aquele produto em específico e ativa no painel, *panel of inspection*, um sinal de conformidade (verde) ou não conformidade (vermelho). Na Fig. 4.10 é ilustrado o caso em que os elementos de fixação não correspondem a aquele código de produto, e, portanto, é ativado o sinal de alerta (vermelho). Em conjunto com o alerta vermelho, o sistema dispara um alerta sonoro utilizando o próprio *speaker buzzer* da CPU

Figura 4.10 – Interface do sistema de inspeção com os respectivos resultados.



Fonte: Elaborado pelo autor.

O acionamento do sistema de inspeção é controlado a partir do painel, *Run System*, lado direito, acima na interface gráfica. Para execução do processo basta acionar a opção *start loop* que o sistema vai fazer a aquisição e o processamento das imagens, identificando o código do produto e os respectivos os elementos de fixação e informar se o produto inspecionado atende ou não as condições conformidade.

Uma vez finalizado e desativado o processo de inspeção ao longo do turno, para reinicializar o processo novamente são necessários o ajuste e a calibração do sistema. A etapa de calibração envolve pequenos acertos do posicionamento das câmeras, o ajuste de foco e a definição dos parâmetros de processamento a serem utilizados no processo de inspeção. Uma vez realizada à calibração do sistema, o processo de inspeção é feito de forma automática e controlado pelo computador que gerencia o sistema de câmeras e *hardware* de aquisição, o processamento e extração de características das imagens e verificação final da conformidade do produto.

4.3 PROCESSO DE CALIBRAÇÃO

A calibração do Sistema de Visão Computacional é uma etapa importante no sistema de inspeção e deve ser realizada previamente antes de se iniciar o processo inspeção do produto. No processo de calibração é utilizada uma amostra do próprio produto para definição dos parâmetros a serem utilizados na captura e o processamento das imagens. Nessa etapa o usuário tem total controle dos parâmetros do sistema, definição da dimensão do kernel que regula a quantidade de suavização ou borramento, do limiar de binarização, bem como definição da área de objetos binários a serem filtrados.

Uma vez definidos esses parâmetros, o aplicativo é municiado com essas informações e todo o processo de inspeção e verificação da conformidade do produto é feito pelo próprio aplicativo, de forma automática, sem a intervenção do usuário.

4.4 TESTES E AVALIAÇÃO DO DISPOSITIVO DE VISÃO COMPUTACIONAL INTEGRADO NA LINHA DE PRODUÇÃO

Os testes de avaliação e validação dos algoritmos e da metodologia do Sistema de Visão Computacional para inspeção dos diferentes tipos de capotas marítimas produzidas na fábrica foram realizados na bancada de testes discutida no capítulo 3. Neste capítulo a proposta é simular a implementação e operação do Protótipo do Dispositivo de Visão de Máquina já na própria linha de montagem.

Na linha de montagem são produzidos cinco diferentes modelos de capota marítima, mas o conjunto de elementos de fixação utilizados na ancoragem da capota na carroceria dos veículos é o mesmo para quatro modelos de capotas. São utilizados apenas dois tipos de elementos de fixação distintos. Um conjunto de elementos de fixação com um perfil mais delgado, mostrado na Fig. 4.11a e um conjunto de elementos de fixação com perfil mais largo, Fig. 4.11b, denominados respectivamente por elementos tipo I e elementos tipo II.

Figura 4.11 – Conjuntos de elementos de fixação.

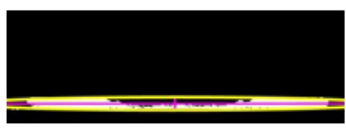



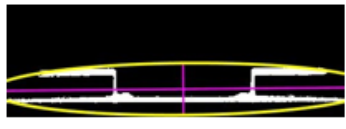





Fonte: Elaborado pelo autor.

Os elementos do tipo I são utilizados em 4 diferentes modelos de capota, modelos “CE ADV”, “CE TK WK”, “CD TK WK” e “CD ADV”. Já os elementos do tipo II são usados apenas no modelo “CC TK WK”. Para verificação da conformidade do produto, o sistema de inspeção inicialmente identifica o código do produto e verificar a compatibilidade com um dos cinco tipos de modelos, tendo identificado o produto, o sistema irá identificar o conjunto de fixação, elementos do tipo I ou elementos do tipo II e, finalmente, correlacionar o produto identificado (modelo de capota) com o seu respectivo conjunto de fixação para verificar a conformidade ou não conformidade do produto.

Desta forma, durante a inspeção do produto é possível identificar dez possíveis combinações, ilustradas na Fig. 4.12. Os códigos CE TK WK, CE ADV, CDE TK WK e CD ADV quando vinculados ao conjunto de componentes de fixação do tipo I indica uma montagem correta (elementos conformes), Fig. 4.12a, quando vinculados ao conjunto de fixação do tipo II indica uma montagem incorreta (elementos não conformes), Fig. 4.11b. Já o código CC TK WK quando vinculado ao conjunto de fixação do tipo II indica a montagem correta, Fig. 4.12c e quando vinculados ao conjunto do tipo I indica montagem incorreta, Fig. 4.12d. Nos casos de montagem correta ou incorreta os alertas de conformidade são exibidos em verde ou vermelho respectivamente.

Figura 4.12 – Diferentes eventos possíveis durante inspeção nos produtos.

(a)		CE TK WK CE ADV CD TK WK CD ADV	
(b)		CE TK WK CE ADV CD TK WK CD ADV	
(c)		CC TK WK	
(d)		CC TK WK	

Fonte: Elaborado pelo autor.

A aplicação do sistema de visão para a inspeção dos produtos na fábrica tem o objetivo de identificar montagens eventualmente invertidas, possibilidades (b) e (d) e não permitir expedição de produtos com problemas de qualidade.

4.4.1 Calibração e Definição dos Parâmetros do Sistema de Inspeção Integrado na linha de Montagem

Nos testes de avaliação proposto o Sistema de Inspeção está integrado na linha de montagem e a calibração e definição dos parâmetros de aquisição e processamento das imagens foi realizada diretamente na linha de montagem utilizado o modelo de capota “CD ADV” e o seu respectivo conjunto de elementos de fixação, elementos do tipo I.

O procedimento foi iniciado com a avaliação previa das imagens. Para isso, a opção *preview* das câmeras foi ativada e, a partir da visualização da imagem no *display*, foi feita a avaliação da qualidade da imagem e os ajustes de foco e do posicionamento da câmera para o melhor enquadramento do objeto na cena.

Uma vez ajustado o posicionamento e foco das câmeras, foi feita a captura da imagem do código pela câmera 1 e a captura dos elementos de fixação pela câmera 2. Capturadas as imagens, foram iniciados os respectivos processos de processamento e extração de caracteres das imagens.

No processo de extração do código, foram utilizados inicialmente os parâmetros de controle definidos por *default* do sistema. Os parâmetros de *default* são definidos para uma condição padrão, entretanto, podem, eventualmente, não levar o reconhecimento correto devido a pequenas alterações ambientais e ou mesmo movimentações de posicionamentos do sistema. Na Fig. 4.13 é mostrado o resultado inicial da identificação do código durante o processo de calibração com os parâmetros *default*, como pode ser observada, a identificação da *string* contendo a especificação do código foi parcialmente identificada e a identificação de algumas letras ficou comprometida.

Figura 4.13 – Painel de apresentação da *string* de reconhecimento. Código insatisfatório.



Fonte: Elaborado pelo autor.

Esse problema durante o processo de calibração foi corrigido com a definição dos parâmetros de processamento, Fig. 4.14. Neste caso, a intensidade de suavização e contraste através do controle *Averaging Filter*, o limiar de binarização através do controle *Threshold* e a filtragem de objetos binários através dos controles *Smaler Objects Filter* e *Bigger Objects Filter*, foram redefinidos. Estes controles adequam a maneira como os objetos binários são segmentados ao final do processamento, de forma que a função OCR possa ser utilizada corretamente. Para o caso em questão, foram aplicados um *kernel* de dimensão 5x5, um alargamento de contraste onde 1% das maiores e das menores intensidades foram saturadas, um limiar de binarização de 0,96, e objetos binários com áreas menores que 160 pixels foram filtrados.

Redefinidos os parâmetros, a atualização dos valores e resultados obtidos no processo de identificação do código são imediatamente mostrados na tela *Recognized Code*.

Figura 4.14 – Controles dos parâmetros de processamento. Imagem do código.



Fonte: Elaborado pelo autor.

Para a identificação do conjunto de elementos de fixação o procedimento foi similar, agora com a segunda câmera sendo ativada. A ativação é feita através do menu *popup* e o modo *live* é mostrado no *display*, possibilitando o enquadramento e ajuste correto da imagem. O botão *Object Extraction* é acionado e a extração dos descritores do conjunto de fixadores é realizada imediatamente. Neste caso não foi necessária a redefinição dos parâmetros de controle de processamento. Os valores de suavização, controle *Averaging Filter* que opera de maneira concomitante com o alargamento de contraste, do limiar de binarização, controle *Threshold*, e dos parâmetros de filtragem de objetos binários, controles *Smaler Objects Filter* e *Bigger Objects Filter*, foram utilizados por *default*. Foi utilizado um *kernel* de dimensão 3x3, um alargamento de contraste onde 1% das maiores e das menores intensidades foram saturadas, um limiar de binarização de 0,2, e objetos binários com áreas menores que 80 pixels foram filtrados. Os valores dos descritores identificados são exatamente os valores dos elementos de fixação do tipo I, Fig. 4.15.

Figura 4.15 – Controles dos parâmetros de processamento. Imagem dos elementos de fixação.



Fonte: Elaborado pelo autor.

Nesse ponto é importante salientar e ilustrar como seria feito o ajuste e a definição dos parâmetros utilizados para a identificação do conjunto de elementos de fixação quando os parâmetros *default* não levam a identificação correta dos descritores do conjunto de elementos de fixação. Na Fig. 4.16 são ilustradas situações de segmentações onde a formação do objeto binário não foi adequada para a caracterização do conjunto de componentes, sendo necessário o ajuste dos parâmetros utilizados.

Figura 4.16 – Segmentação incorreta indicando a necessidade de calibração.



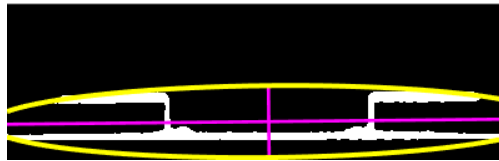
Fonte: Elaborado pelo autor.

Na Fig. 4.16a, apenas parte do objeto binário que representa o conjunto de fixação foi segmentado, lado direito, e as *features* extraídas não identificam os elementos de fixação. Na Fig.4.16b, a imagem ilustra a mesma ocorrência, desta vez segmentando uma região menor ainda. Na Fig. 4.16c, apenas as características da região inferior foram

extraídas. Já na Fig. 4.16d a segmentação dos elementos de fixação foi melhor, mas a parte superior, lado esquerdo, não foi segmentada.

Na Fig. 4.17 é mostrado o resultado da segmentação obtida a partir do ajuste dos parâmetros de controles. Neste caso, a segmentação obtida a partir do ajuste nos controles produz um objeto sólido e com o perfil do conjunto de peças.

Figura 4.17 – Segmentação correta após ajustes nos parâmetros de processamento.



Fonte: Elaborado pelo autor.

Com todos os controles ajustados tanto para o reconhecimento ótico de caracteres (código) quanto para a extração dos descritores de forma (conjunto de componentes de fixação), o sistema está pronto para sua ativação e início do processo de inspeção no produto.

4.4.2 Teste de validação - conformidade e não conformidade

Os testes de verificação e validação dos algoritmos do Sistema de Visão de Máquina foram realizados no capítulo 3. Neste capítulo a proposta é avaliar a implementação do Protótipo do Dispositivo de Visão de Máquina na própria linha de produção, simular a operação de inspeção e verificar a capacidade do Sistema em identificar a conformidade ou não do produto.

Para simular a operação de inspeção na linha de produção, o Protótipo do Sistema de Visão de Máquina foi instalado no final da linha, antes da etapa de embalagem do produto, e um conjunto de amostras de capotas pré-selecionadas foram utilizadas para verificação do funcionamento do dispositivo (Protótipo).

O conjunto de capotas utilizadas nos testes é formado por dez amostras, oito amostras montadas corretamente e duas amostras montadas com o conjunto de elementos de fixação trocados. O conjunto de amostras foi definido a partir de dois modelos de capotas, modelo CE TK WK (código STRADA-CE-TK-WK-DD-MM-AA) e modelo CC TK WK (código STRADA-CC-TK-WK-DD-MM-AA), sendo quatro amostras do modelo CE TK

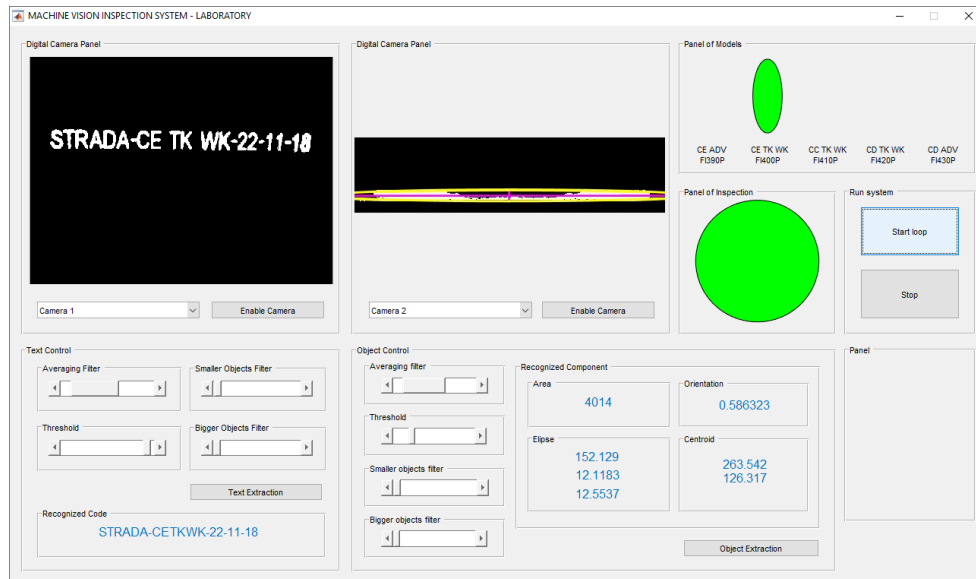
WK montada com o seu respectivo conjunto de elemento de fixação (tipo I), quatro amostras do modelo CC TK WK montada com o seu respectivo conjunto de elemento de fixação (tipo II) e duas amostras montadas com o conjunto de elementos de fixação, propositalmente, invertido.

A estrutura lógica do sistema de inspeção é a mesma apresentada anteriormente, após a identificação do código do produto e extração dos descritores do conjunto de componentes, é feita a correlação dos descritores com o código do produto para identificar a conformidade ou não do produto.

Na linha de produção real cada capota produzida é inspecionada visualmente ao final da linha pelo inspetor de produção e liberada para expedição ou reprovada devido a qualquer eventual não conformidade do produto. Para simular uma condição similar ao processo contínuo de produção, o Sistema de Inspeção instalado ao final da linha de produção foi ativado (após calibração) e cada amostra do conjunto de capotas previamente definido foi inspecionada. Para isso, cada uma das amostras era instalada na estrutura do dispositivo e o Sistema de Visão de Máquina capturava automaticamente as imagens do código e do conjunto de elemento de fixação da amostra e realizava o correspondente processamento das imagens para extração dos parâmetros de interesse, utilizados para a verificação da conformidade (ou não) do produto.

Na Fig. 4.18 são ilustrados os parâmetros e o resultado da inspeção de uma amostra com o correspondente conjunto de elementos de fixação. A amostra é o modelo CE TK WK (código) montado com o conjunto de elementos de fixação tipo I, isto é, produto conforme. Após a captura das imagens e a correspondente transformação para imagens binárias (*Digital Camera Panel*), o código do produto (*Recognized Code*) é identificado “CE TK WK” e os descritores do conjunto de elementos de fixação (*Recognised Component*) são extraídos, i.e., os valores dos eixos principais fornecidos em unidades pixels “152.129 e 12.118”. Os valores obtidos correspondem ao conjunto de elementos do tipo I. Identificado o código e os descritores, o Sistema de Inspeção verifica a conformidade do produto. Neste teste, como esperado, o modelo de capota identificada, *Panel of Models* (verde), é o modelo CE TK WK e o conjunto de elementos de fixação é o do tipo I, consequentemente, indicando a aprovação do produto *Panel of Inspection* (verde).

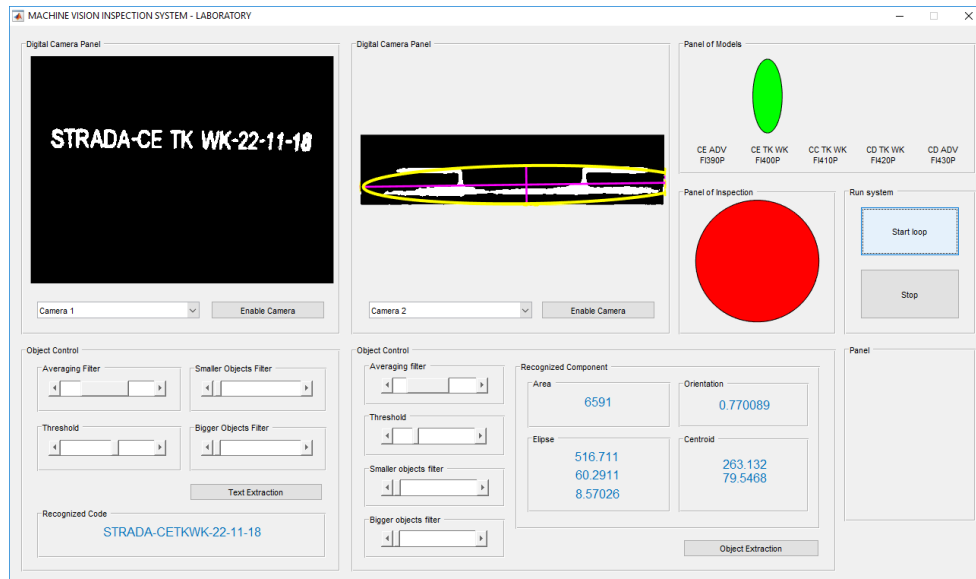
Figura 4.18 – Interface do sistema apontando produto conforme.



Fonte: Elaborado pelo autor.

Na Fig. 4.19 é ilustrado o resultado da inspeção de uma amostra não conforme, mais especificamente, a amostra do modelo CE TK WK (código) montado com o conjunto de elementos de fixação trocado (tipo II). Inicialmente é identificado o código do produto (CE TK WK) e posteriormente, extraídos os descritores do conjunto de elementos de fixação, i.e., os valores dos eixos principais em pixels (516.711 e 60.2911), valores que correspondem ao conjunto de elementos do tipo II. Neste teste, a amostra inspecionada foi efetivamente identificada, modelo CE TK WK, *Panel of Models* (verde) e o conjunto de elementos de fixação também foi identificado, elementos do tipo II, entretanto, esse conjunto de elementos de fixação não é utilizado nesse modelo. Consequentemente, o alerta é ativado *Panel of Inspection* (vermelho), indicando a não aprovação da amostra (produto não conforme).

Figura 4.19 – Interface do sistema acusando produto não conforme.



Fonte: Elaborado pelo autor.

Nos testes realizados a conformidade das dez amostras analisadas foram todas identificadas corretamente, ou seja, oito foram classificadas como produtos conformes e duas como produtos não conformes. Ressalta-se que nos testes a sequência de amostras analisadas foi definida de forma aleatória combinando, indistintamente, tanto as capotas montadas corretamente como as montadas com o conjunto de fixação invertido. Em todos os casos, o sistema foi capaz de identificar tanto a mudança do modelo, quanto as mudanças dos descritores sem dificuldades e, conseqüentemente, classificar corretamente cada amostra.

Isso demonstra que metodologia proposta se mostrou adequada para este tipo de inspeção e que os descritores de eixos inerciais são adequados para a identificação dos componentes de fixação, levando à inspeção correta de todas as amostras. Possibilitando assim, construir um sistema de visão de máquina que identifica o produto e verifica automaticamente se seus componentes estão em conformidade com os requisitos exigidos.

Outro aspecto importante é que a operação do sistema e os alertas de conformidade integrados em uma interface gerenciadora facilita a utilização do sistema de inspeção que, numa etapa posterior, poderia ser integrado a todo o sistema de produção, permitindo a criação de um banco de dados de todo o histórico de produção.

5 CONCLUSÕES

Neste trabalho é discutido o desenvolvimento de um Sistema de Visão de Máquina para a realização de inspeção de um produto industrial, mais especificamente, a inspeção de capotas marítimas objetivando a verificação da conformidade ou não do conjunto de elementos de fixação com cada modelo de capota.

O sistema se utiliza de duas câmeras digitais, sem nenhum recurso adicional, para a captura de imagens do produto. Uma das câmeras é apontada para a região onde o código do produto é impresso, e a outra é apontada para um conjunto de elementos de fixação, peças características de cada modelo de produto. Através da captura das imagens dos códigos e dos conjuntos de peças que tais produtos utilizam, realiza-se a vinculação código/componente de fixação para cada modelo é feita a partir do processamento e extração de atributos das imagens.

O processamento das imagens envolve aplicação de técnicas de filtragem espacial com máscara de média, alargamento de contraste, binarização e filtragem de objetos binários que não compõem os elementos de interesse e extração dos descritores do conjunto de elementos de fixação.

Para avaliação da metodologia e dos algoritmos implementados foi utilizada uma bancada de testes em que vários componentes foram avaliados alterando as condições do ambiente, posicionamento das câmeras, assim como as trocas dos componentes e das amostras de código sobre a bancada.

A correta segmentação de objetos binários tanto dos códigos dos produtos quanto dos seus componentes revelou-se um problema. Sem a segmentação adequada, a extração dos descritores e o reconhecimento dos códigos não seria possível. Este obstáculo foi superado através da definição criteriosa dos parâmetros mecânicos, como o foco da câmera e posicionamento, ambientais, como a incidência de luz direta ou difusa, e digitais, como tamanho de kernel e nível de limiarização

Um protótipo da proposta foi implementado em uma linha de montagem para testes de avaliação e validação da metodologia proposta e os resultados obtidos nos testes preliminares mostraram-se robustos, com verificação corretas dos produtos conformes e dos produtos não conformes.

Testes adicionais, envolvendo principalmente variações ambientes ainda são necessários para definir dos limites e a robustez do Sistema de Visão Computacional proposto, de forma que o mesmo poderia ser implementado em chão de fábrica.

5.1 SUGESTÕES PARA TRABALHOS FUTUROS

Como sugestões para trabalhos que podem ser produzidos como desdobramentos desta pesquisa, cita-se:

A interface desenvolvida pode ser aprimorada com a inclusão de alguns campos para controle e contagem de amostras e geração de relatórios, de tal forma que ao final do processo de inspeção do lote o usuário tenha à sua disposição o resultado de todos os produtos inspecionados.

Os descritores vinculados às coordenadas espaciais de imagem podem ser utilizados quando se pretende mensurar a variação de posicionamento de objetos que devem permanecer fixados em determinada posição em uma montagem ou mesmo acompanhar o posicionamento de objetos na cena. Com a vinculação ao posicionamento, seria possível determinar não conformidades em componentes que foram montados em posição incorreta ou deslocada.

REFERÊNCIAS

ACHARYA, T.; RAY, A. K. **Image processing principles and applications**. New Jersey: John Wiley & Sons, 2005.

AGIN, G. J. **Computer vision systems for industrial inspection and assembly**. New York: IEEE, 1980.

ARAUJO, S. A.; PIRES, A. C.; SANTANA, J. C. C. Classificação Automática de Sementes de Feijão Usando Técnicas de Visão Computacional. *In*: ENCONTRO NACIONAL DE ENGENHARIA DE PRODUÇÃO, 31., Belo Horizonte, 2011. **Anais [...]** Belo Horizonte, 2011.

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS – ABNT. **NBR ISO 9001**: Sistemas de gestão da qualidade: requisitos. Rio de Janeiro: ABNT, 2008.

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS – ABNT. **ISSO/TS 16949**: Sistemas de gestão da qualidade: requisitos particulares para aplicação da ABNT NBR ISO 9001:2008 para organizações de produção automotiva e peças de reposição pertinentes. Rio de Janeiro: ABNT, 2010.

BROSNAN, T.; SUN, D. W. Inspection and grading of agricultural and food products by computer vision systems – a review. **Computers and Electronics in Agriculture**, Amsterdam, v. 36, n. 2-3, p. 193-213, 2002.

BROSNAN, T.; SUN, D. W. Improving quality inspection of food products by computer vision: a review. **Journal of Food Engineering**, Amsterdam, v. 61, n. 1, p. 3-16, 2004.

BURGER, W.; BURGE, M. J. **Principles of digital image processing**: core algorithms. London: Springer-Verlag, 2009. v. 2.

BURGER, W.; BURGE, M. J. **Principles of digital image processing**: advanced methods. Nova York. Springer-Verlag London, 2009. v. 3.

CHU, H. H.; WANG, Z. Y.; LUO, X. Quality inspection algorithm based on machine vision for tube-sheet welding. *In*: IEEE INTERNATIONAL WORKSHOP ON ADVANCED ROBOTICS AND ITS SOCIAL IMPACTS (ARSO), 2016, Shanghai. **Proceedings [...]** Shanghai: [s. n.], 2016.

COSTA, L. A. **Identificação automática de caracteres em antibiogramas com uso de momentos invariantes**. 2015. Dissertação (Mestrado em Engenharia Biomédica) - Faculdade Gama, Universidade de Brasília, 2015.

CUBERO, S.; ALEIXOS, N.; MOLTÓ, E.; SANCHIS, J. G.; BLASCO, J. Advances in machine vision applications for automatic inspection and quality evaluation of fruits and vegetables. **Food Bioprocess Technol.**, New York, v. 4, n. 4, p. 487-504, 2011.

ERHARDT, A. **Theory and applications of digital image processing**. Offenburg: University of Applied Sciences Offenburg, 2000.

ERPEN, L. R. **Reconhecimento de padrões em imagens por descritores de forma.** Dissertação (Mestrado) - Universidade Federal do Rio Grande do Sul, Porto Alegre, 2004.

FILHO, O. M.; NETO, H. V. **Processamento digital de imagens.** Rio de Janeiro: Brasport, 1999. 307 p.

FLUSSER, J. On the independence of rotation moment invariants. **Pattern Recognition**, Amsterdam, v. 33, n. 9, p. 1405-1410, 2000.

FLUSSER, J. Moment invariants in image analysis. In: WORLD ACADEMY OF SCIENCE, ENGINEERING AND TECHNOLOGY, 11., [S. l.]. **Proceedings [...]** [S. l.: s. n.], 2006. p. 197. 2006.

GEROGOPOULOS, A.; LOIZOS, A.; FIOUDA, A. Digital image processing as a tool for pavement distress evaluation. **ISPRS Journal of Photogrammetry and Remote Sensing**, [S. l.], v. 50, n. 1, 1995.

GOES, S; MIKOS, W, L. Desenvolvimento de sistema de inspeção automatizada de componentes manufaturados baseada em visão de máquinas. *In: ENCONTRO NACIONAL DE ENGENHARIA DE PRODUCAO*, 34., 2014, [S. l.]. **Anais [...]** [S. l.: s. n.], 2014.

GONZALEZ, R. C.; WOODS, R. E. **Processamento digital de imagens.** 3. ed. São Paulo: Pearson Prentice Hall, 2010.

HAMDI, A. A.; FOUAD, M. M.; SAYED, M. S.; HADHOUD, M. M. Patterned Fabric Defect Detection System Using Near Infrared Imaging. *In: IEEE INTERNATIONAL CONFERENCE ON INTELLIGENT COMPUTING AND INFORMATION SYSTEMS (ICICIS)*, 2017, Cairo. **Proceedings [...]** New York: IEEE, 2017. p. 111-117.

HU, M. K. Visual pattern recognition by moment invariants. **IRE Transactions on Information Theory**, New York, v. 8, n. 2, p. 179-187, 1962.

KAUR, A.; SAINI, N.; KAUR, R.; DAS, A. Automatic classification of turmeric rhizomes using the external morphological characteristics. *In: CONFERENCE ON ADVANCES IN COMPUTING, COMMUNICATIONS AND INFORMATICS (ICACCI)*, Jaipur. **Proceedings [...]** [S. l.: s. n.], 2016.

KILIAN, J. **Simple image analysis by moments.** Version 0.2. Freely distributable. [S. l.], 2001.

KIVIORG, A. **Development of a low-cost vision system for finding contour and surgace defects on cast iron engine components.** Master of Science (Thesis) - School of Industrial Engineering and Management. Royal Institute of Technology. Stockholm: Sweden, 2014.

LOAIZA, M. E.; RAPOSO, A. B.; GATTASS, M. Multi-camera calibration based on an invariant pattern. **Computers & Graphics Journal**, Amsterdam, p. 198-207, 2010.

MALAMAS, E. N.; PETRAKIS, E. G. M.; ZERVAKIS, M.; PETIT, L. LEGAT, J. D. A survey on Industrial vision systems, applications and tools. **Image and Vision Computing**, [S. l.], v. 21, n. 2, p. 171-188, 2003.

MACHADO, D. S. **Sistema de Inspeção Visual Automática Aplicado ao Controle de Qualidade de Ovos em Linhas de Produção**. Dissertação (Mestrado) - Centro Federal de Educação Tecnológica de Minas Gerais, Belo Horizonte, 2009.

NOORDAM, J. C.; OTTEN, G. W.; TIMMERMANS, T. J. M.; ZWOL, B. H. V. High-speed potato grading and quality inspection based on a color vision system. **Proceedings of spie. SPIED Digital Library**. San Jose, CA. 2000.

ONITA, D.; VARTAN, N.; KADAR, M.; BIRLUTIU, A. Quality Control in Porcelain Industry based on Computer Vision Techniques. *In: IEEE INTERNATIONAL YOUNG ENGINEERS FORUM, 2018, Costa da Caparica. Proceedings [...]* New York: IEEE, 2018.

PAWLAK, M. **Image analysis by moments: reconstruction and computational aspects**. Wrocław: Oficyna Wydawnicza Politechniki Wrocławskiej, 2006.

SAAD, N. M.; ABDUL, N. N. S.; WAHAB, F. A. Shape Defect Detection for Product Quality Inspection and Monitoring System. *In: INTERNATIONAL CONFERENCE ON ELECTRICAL ENGINEERING, COMPUTER SCIENCE AND INFORMATICS, 2017. Yogyakarta. Proceedings [...]* New York: IEEE, 2017. p. 19-21.

SAPIRSTEIN, H. D.; NEWMAN, M.; WRIGHT, E. H.; SHWEDYK, E.; BUSHUK, W. An instrumental system for cereal grain classification using digital image analysis. **Journal of Cereal Science**, London, v. 6, n. 1, p. 3-14, 1987.

SATPUTE, M. R.; JAGDALE, S. M. **Automatic fruit quality inspection system**. *In: INTERNATIONAL CONFERENCE ON INVENTIVE COMPUTATION TECHNOLOGIES (ICICT), Coimbatore, 2016. Proceedings [...]* [S. l.: s. n.], 2016.

SHAFEEK, H. I.; GADELMAWLA, E. S.; SHAFY, A. A. A.; ELEWA, I. M. Assessment of welding defects for gas pipeline radiographs using computer vision. **NDT&E International**, Amsterdam, 2004. p. 291-299.

SLUZEK, A. Identification and inspection of 2-D objects using new moment-based descriptors. **Pattern Recognition Letters**, Amsterdam, p. 687-697, 1995.

SMITH, S. W. **The Scientist and Engineer's Guide to Digital Signal Processing**. 2. ed. San Diego: California Technical Publishing, 1999. p. 650.

STIVANELLO, M. E.; VARGAS, S.; ROLOFF, M. L.; STEMMER, M. R. Automatic Detection and Classification of Defects in Knitted Fabrics. **IEEE Latin America Transaction**, New York, v. 14, 2016.

TAN, L. **Digital signal processing fundamentals and applications**. Burlington: Elsevier, 2008. 820 p.

THOMAS, A. D. H.; RODD, M. G.; HOLT, J. D.; NEILL, C. J. Real-time industrial visual inspection: a review. [s. l.]: Real-Time Imaging, 1995. p. 139-158.

VARGAS, S. **Desenvolvimento de um sistema de inspeção automática de tecido de malha**. Dissertação (Mestrado) - Instituto Federal de Educação, Ciência e Tecnologia de Santa Catarina. Florianópolis, 2016.

WATANABE, Y.; KOMURO, T.; ISHIKAWA, M. A High-Speed Vision System for Moment-Based Analysis of Numerous Object. *In: IEEE INTERNATIONAL CONFERENCE ON IMAGE PROCESSING, 2007, Proceedings [...]* [S. l.: s. n.], 2007.

WU, H.; LUO, H.; ZHU, W.; WANG, Y.; ZHANG, Q.; MA, B.; YANG, Y.; FAN, H.; XU, H. **Surface defect detection of plaster coating based on machine vision**. New York: IEEE, 2017.

XING, J.; SAEYS, W.; BAERDEMAEKER, J. D. Combination of chemometric tools and image processing for bruise detection on apples. **Computers and Electronics in Agriculture**, Amsterdam, 2007. p. 1-13.

XU, M.; MA, Y.; CHEN, S. Research on Real-Time Quality Inspection of PET Bottle Caps. *In: IEEE INTERNATIONAL CONFERENCE ON INFORMATION AND AUTOMATION (ICIA), Macau, 2017. Proceedings [...]* [S. l.: s. n.], 2017. p. 1023-1026.

ZALHAN, M. Z.; SYARMILA, S. S.; NAZRI, I. M.; TAHA, I. M. Vision-based egg grade classifier. *In: INTERNATIONAL CONFERENCE ON INFORMATION AND COMMUNICATION TECHNOLOGY (ICICTM), 2016, Kuala Lumpur. Proceedings [...]* [S. l.: s. n.], 2016. p. 31-35.

ZAMBRANO, C. E. P.; CÁCERES, J. C. G.; CASTAÑÓN, C. A. B. Computer vision grading system for physical quality evaluation of green coffee beans. *In: LATIN AMERICAN COMPUTING CONFERENCE (CLEI), XLII, Valparaiso. Proceedings [...]* [S. l.: s. n.], 2016.

ZHANG, Y.; ZHANG, Y.; Wen, C. A new focus measure method using moments. **Image and Vision Computing**, Amsterdam, v. 18, n. 12, p. 959-965, 2000.

ZHENG, C.; SUN, D. W.; ZHENG, L. Recent developments and applications of image feature for food quality evaluation and inspection: a review. **Trends in Food Science & Technology**, Amsterdam, v. 17, n. 12, p. 642-655, 2006.

APÊNDICE A - SINOPSE DA LINGUAGEM DE PROGRAMAÇÃO EMPREGADA

São mostradas neste apêndice as principais funções para a operação de uma câmera USB através do MATLAB®. O *hardware support package*, chamado *Generic Vídeo Interface*, necessário para a instalação do adaptador de vídeo *winvideo*, permite a comunicação com a câmera USB que é estabelecida quando se cria um objeto de vídeo conforme Figura A.1.

Figura A.1 – Criando um objeto de vídeo.

```
% Construct a video input object.
obj = videoinput('winvideo', ID);
% Acquire a single image frame.
frame = getsnapshot(obj);
% Display a frame acquired
imshow(frame)
```

Fonte: Elaborado pelo autor.

No exemplo a seguir, uma imagem capturada em escala de cinza é processada com filtro de média, limiarização, e filtragem de objetos. A Figura A.2 mostra também a criação do filtro espacial com uma máscara 5x5.

Figura A.2 – Exemplo de pré-processamento.

```
% Create a normalized, 5-by-5, averaging filter.
h = ones(5,5)/25;
% Apply the averaging filter to the grayscale image.
frameFilt = imfilter(frameGray, h);
% Apply the contrast adjust to the filtered image.
frameAdj = imadjust(frameFilt);
% Convert image to binary image, based on threshold.
frameBW = im2bw(frameAdj, 0.5);
% Complement image.
frameBW = ~frameBW;
% Filter, retaining only those objects with areas between 40 and 5000.
frameBWObjFilt = bwareafilt(frameBW, [40 5000]);
% Display a frame acquired
imshow(frameBWObjFilt)
```

Fonte: Elaborado pelo autor.

A função de extração de caracteres nativa do *software* é usada para o reconhecimento de textos impressos conforme Figura A.3. Existem algumas opções quando o resultado do

reconhecimento é pobre, é possível incorporar à função a particularidade de ‘entender’ o texto na forma de linha, na forma de um bloco, ou na forma de uma palavra.

Figura A.3 – Chamando a função OCR e armazenando os resultados em uma variável.

```
% Call OCR function
structure = ocr(frameBWObjFilt);
% storing the result into a variable
textResult = structure.Words;
```

Fonte: Elaborado pelo autor.

Alguns descritores podem ser obtidos pela função abaixo, como mostra a Figura A.4. Uma programação extra é usada para desenhar as elipses em tempo real a partir dos eixos inerciais e da orientação.

Figura A.4 – Extraindo características dos objetos.

```
% Area of objects (structure)
objArea=regionprops(frameBWObjFilt, 'area');
% Major axis of ellipse
objMajorAxis=regionprops(frameBWObjFilt, 'MajorAxisLength');
% Minor axis of ellipse
objMinorAxis=regionprops(frameBWObjFilt, 'MinorAxisLength');
% Ellipse orientation
objOrient=regionprops(frameBWObjFilt, 'Orientation');
% Ellipse centroid
objCentro=regionprops(frameBWObjFilt, 'Centroid');
```

Fonte: Elaborado pelo autor.

Figura A.5 – Função para cálculo e visualização dos descritores.

```

1 function descritores(hObject, eventdata, handles)
2
3 objArea=regionprops(handles.frameBWFilt2, 'area');
4 objArea=cell2mat(struct2cell(objArea));
5 objAreaN=max(objArea);
6 set(handles.dispArea, 'String', objAreaN)
7 posObj=find(objArea==objAreaN);
8 posObjM=posObj;
9 bool isempty(objAreaN);
10 if bool == 1
11     msg=msgbox('Operation does not return object. ');
12     pause(0.5); delete(msg);
13
14 else
15     objMajorAxis=regionprops(handles.frameBWFilt2, 'MajorAxisLength');
16     objMajorAxis=objMajorAxis(posObjM).MajorAxisLength;
17     set(handles.dispEixoMaior, 'String', objMajorAxis)
18     objMinorAxis=regionprops(handles.frameBWFilt2, 'MinorAxisLength');
19     objMinorAxis=objMinorAxis(posObjM).MinorAxisLength;
20     set(handles.dispEixoMenor, 'String', objMinorAxis)
21     razao=objMajorAxis/objMinorAxis;
22     set(handles.razao, 'String', razao);
23     objOrient=regionprops(handles.frameBWFilt2, 'Orientation');
24     objOrient=objOrient(posObjM).Orientation;
25     set(handles.dispOrient, 'String', objOrient)
26     objCentro=regionprops(handles.frameBWFilt2, 'Centroid');
27     objCentro=objCentro(posObjM).Centroid;
28     set(handles.dispCent, 'String', objCentro)
29     xMaior = objCentro(1) + [ -1 +1 ] * objMajorAxis*cosd(-objOrient)/2;
30     yMaior = objCentro(2) + [ -1 +1 ] * objMajorAxis*sind(-objOrient)/2;
31     line(xMaior,yMaior,'color', 'm', 'linewidth', 2, 'Parent',
handles.axes2);
32     xMenor = objCentro(1) + [ -1 +1 ] * objMinorAxis*cosd(-objOrient+90)/2;
33     yMenor = objCentro(2) + [ -1 +1 ] * objMinorAxis*sind(-objOrient+90)/2;
34     line(xMenor,yMenor,'color', 'm', 'linewidth', 2, 'Parent',
handles.axes2);
35     x1=xMaior(1);
36     y1=yMaior(1);
37     x2=xMaior(2);
38     y2=yMaior(2);
39     e=sqrt((objMajorAxis/2)^2 - (objMinorAxis/2)^2)/(objMajorAxis/2);
40     a = 1/2*sqrt((x2-x1)^2+(y2-y1)^2);
41     b = a*sqrt(1-e^2);
42     t = linspace(0,2*pi);
43     X = a*cos(t);
44     Y = b*sin(t);
45     w = atan2(y2-y1,x2-x1);
46     x = (x1+x2)/2 + X*cos(w) - Y*sin(w);
47     y = (y1+y2)/2 + X*sin(w) + Y*cos(w);
48     plot(x,y,'y-', 'linewidth', 3, 'Parent', handles.axes2)
49 end
50 end

```

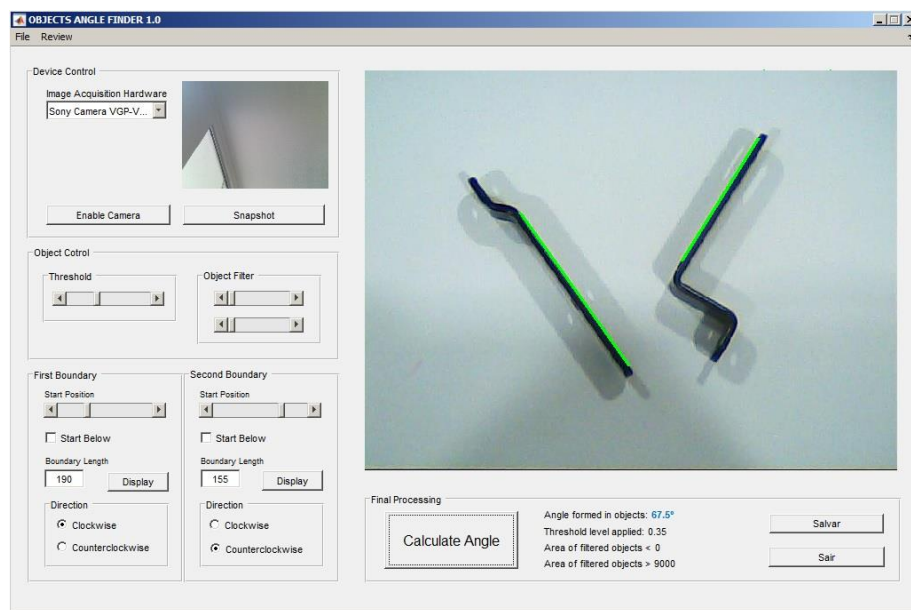
Fonte: Elaborado pelo autor.

APÊNDICE B - APLICATIVO PARA CALCULAR ÂNGULO DE OBJETOS

Demonstra-se neste apêndice uma das possíveis aplicações da detecção de bordas de objetos presentes nas imagens. Um aplicativo que realiza a medição sem contato de ângulos presentes nos objetos foi desenvolvido utilizando a técnica.

A detecção das bordas é feita em uma imagem binária processada a partir da imagem de entrada capturada por uma câmera digital, por exemplo, a imagem capturada das peças metálicas conformadas estudadas neste trabalho. A Figura B.1 mostra a interface do programa desenvolvido para a medição de ângulos.

Figura B.1 – Interface do aplicativo para calcular ângulos de objetos.



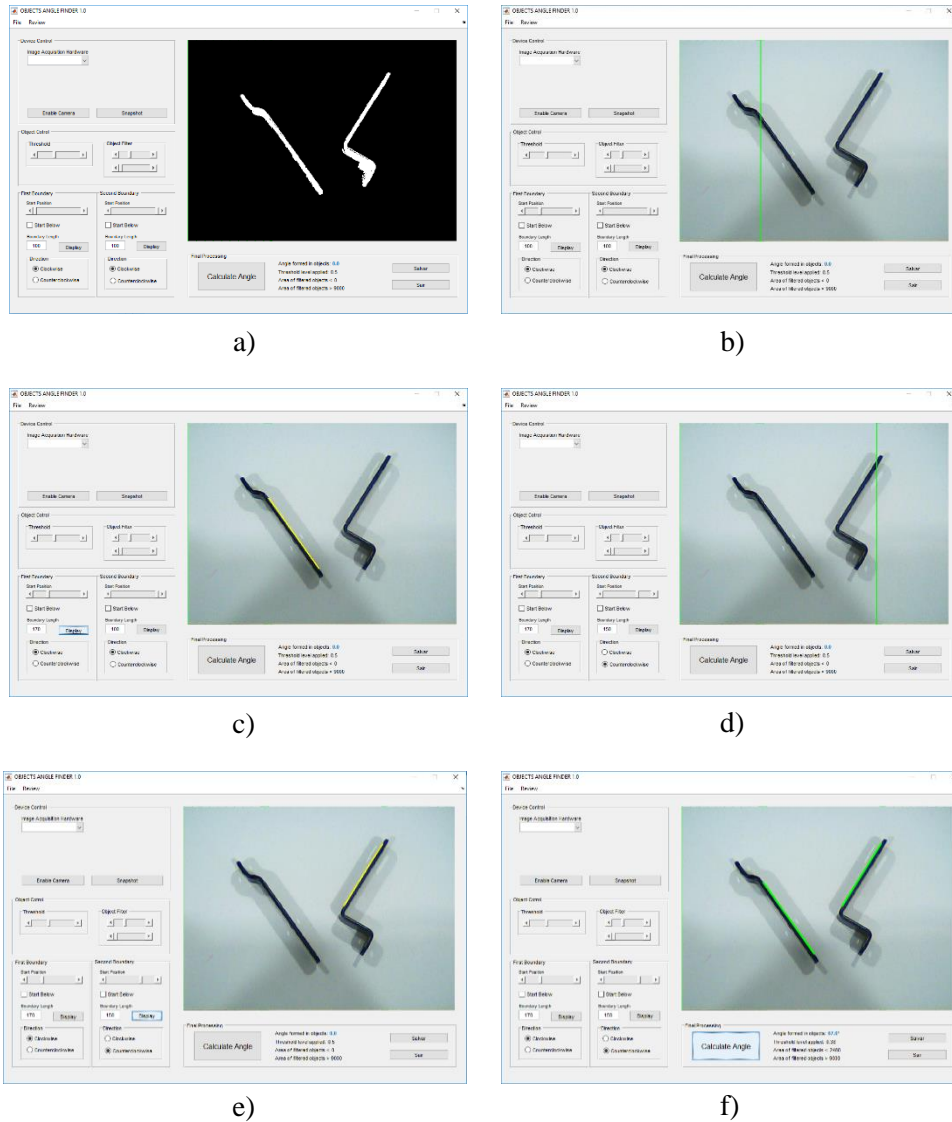
Fonte: Elaborado pelo autor.

A interface disponibiliza todos os controles de interação com o usuário, além de *displays* para visualização das imagens da câmera e das etapas de processamento para obtenção das medidas de ângulos nos objetos presentes na cena.

Após a escolha da câmera a ser utilizada para a captura das imagens através de um menu *popup*, o botão de habilitação da câmera é pressionado ativando a visualização de um *preview* das imagens. Isto permite a escolha do melhor posicionamento da câmera para a captura da imagem, que ocorre quando o botão *snapshot* é apertado.

A imagem capturada passa a ser visualizada em outro display, que mostra, além disto, as operações para o cálculo dos ângulos presentes nos objetos segmentados na imagem. A Figura B.2 mostra como estas operações são realizadas.

Figura B.2 – Etapas para o cálculo dos ângulos.



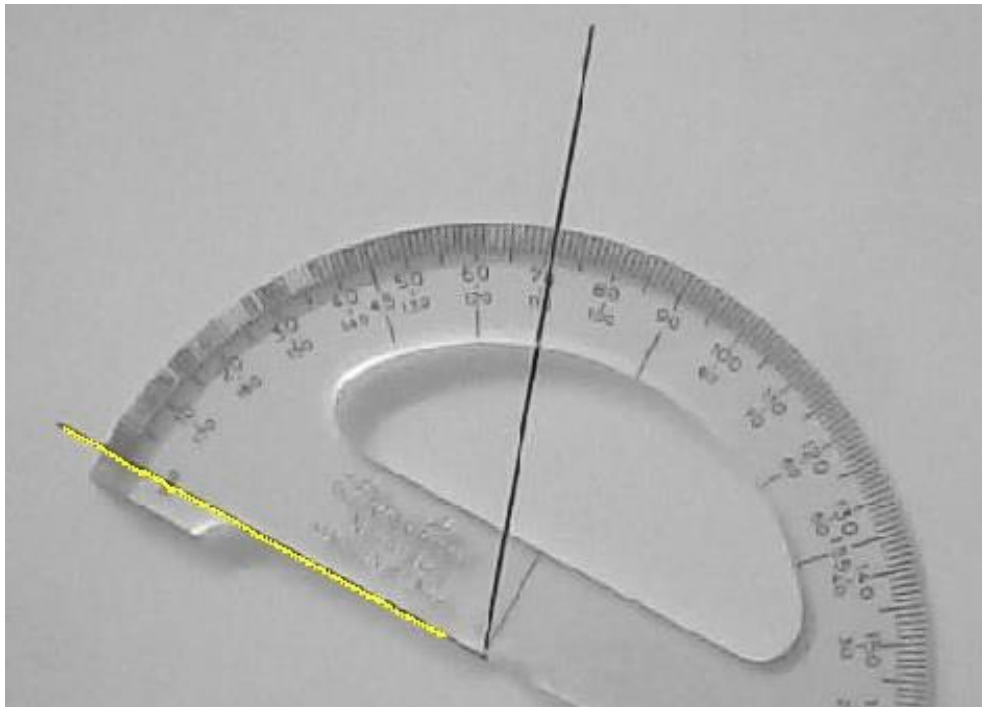
Fonte: Elaborado pelo autor.

Utilizando controles de limiarização e de filtragem de objetos binários, a imagem é segmentada, Figura B.2a. É através da imagem binária que a localização das bordas é realizada. Após a segmentação, uma linha vertical desliza sobre a imagem original através da manipulação de um controle para o usuário. Esta linha opera para a localização do ponto inicial da borda, Figura B.2b.

A quantidade de pontos da borda que serão destacados é arbitrariamente determinada e a direção que estes pontos deverão ser mostrados também é escolhida, desta forma a borda dimensionada é desenhada sobreposta à imagem original, Figura B.2c. Os mesmos passos são repetidos para uma segunda borda, Figuras B.2d e B.2e. Destacadas as bordas, o ângulo é calculado através de um botão na interface, Figura B.2f.

Os pontos da borda seguem uma tendência de linha. Tais pontos, no entanto, possuem uma dispersão que pode ser observada também na Figura B.3, em amarelo. A partir destes pontos é realizada uma regressão linear pelo método dos mínimos quadrados para obtenção da reta de melhor ajuste dos pontos, que o *software* desenha, em verde.

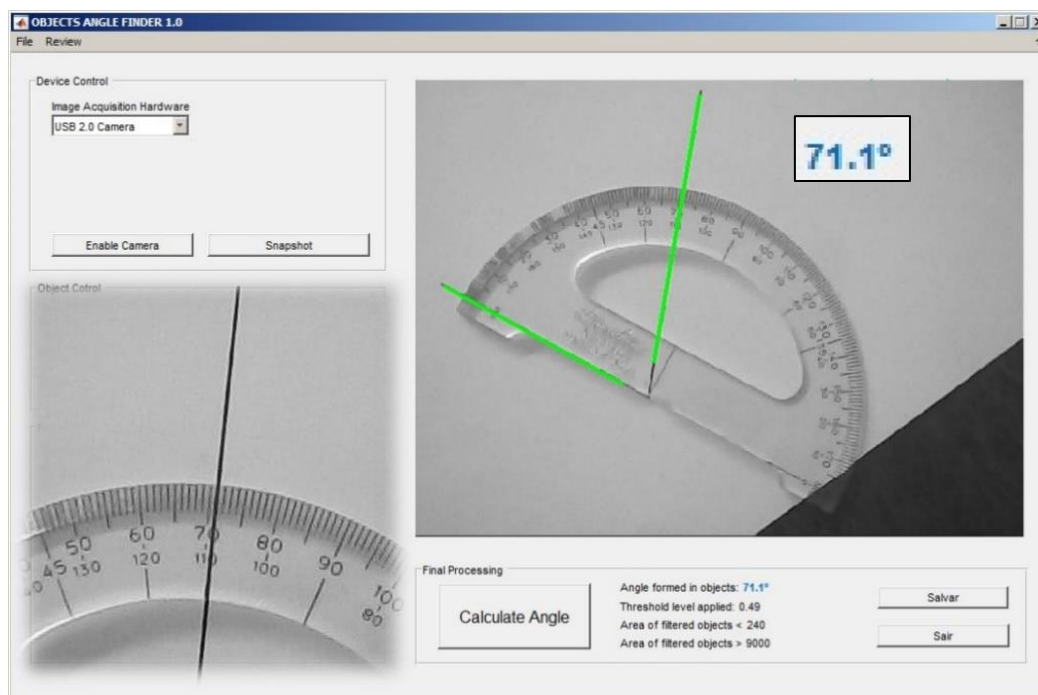
Figura B.3 – Borda detectada de um traço em ângulo.



Fonte: Elaborado pelo autor.

Na Figura B.4 as retas em verde são mostradas e são usadas para o cálculo do ângulo entre elas. As retas possuem o comprimento entre os pontos inicial e final das respectivas bordas traçadas anteriormente, é possível ainda encontrar a origem, no cruzamento das retas, em coordenadas de pixels.

Figura B.4 – Retas em ângulo obtidas por regressão linear.



Fonte: Elaborado pelo autor.

As principais *functions* para esta aplicação são apresentadas a seguir. Na Figura B.5, é apresentada a função que utiliza como input os dois conjuntos de bordas detectadas, traça as retas médias e calcula o ângulo entre elas.

Figura B.5 – Código para computar ângulos em objetos binários.

```

1 function compute_Callback(hObject, eventdata, handles)
2 handles.ab1 = polyfit(handles.boundary1(:,2), handles.boundary1(:,1), 1);
3 handles.ab2 = polyfit(handles.boundary2(:,2), handles.boundary2(:,1), 1);
4 vect1 = [1 handles.ab1(1)]; % create a vector based on the line equation
5 vect2 = [1 handles.ab2(1)];
6 dp = dot(vect1, vect2);
7 % compute vector lengths
8 length1 = sqrt(sum(vect1.^2));
9 length2 = sqrt(sum(vect2.^2));
10 % obtain the larger angle of intersection in degrees
11 handles.angle=180-acos(dp/(length1*length2))*180/pi;
12 handles.figFinal=imshow(handles.frame);
13 hold on
14 handles.plot1=plot([handles.boundary1(1,2)
handles.boundary1(length(handles.boundary1),2)],...
15     [handles.ab1(1)*handles.boundary1(1,2)+ handles.ab1(2) ...
16     handles.ab1(1)*handles.boundary1(length(handles.boundary1),2)+handles.ab1(2)],
...
17     'g', 'linewidth', 3);
18 handles.plot2=plot([handles.boundary2(1,2)
handles.boundary2(length(handles.boundary2),2)],...
19     [handles.ab2(1)*handles.boundary2(1,2)+ handles.ab2(2) ...
20     handles.ab2(1)*handles.boundary2(length(handles.boundary2),2)+handles.ab2(2)],
...
21     'g', 'linewidth', 3);
22 hold off
23 set(handles.angleResult, 'String', [sprintf('%1.1f',handles.angle), '°'])
24 %
25 set(handles.text12, 'String', handles.sldThreshOut)
26 set(handles.text13, 'String', handles.sldFiltObjPOut)
27 set(handles.text14, 'String', handles.sldFiltObjGOut)
28 guidata(hObject,handles)
29 end

```

Fonte: Elaborado pelo autor.

A Figura B.6 mostra a função que traça o primeiro conjunto de pontos que formam a primeira borda. Os *inputs* são o ponto inicial da borda, seu tamanho e direção que a borda deverá ser detectada. Para a segunda borda a função é equivalente e desta forma não será apresentada.

Figura B.6 – Código para a detecção da borda de um objeto.

```

1 function displ_Callback(hObject, eventdata, handles)
2 %
3 %
4 handles.direction=get(handles.dir1, 'SelectedObject');
5 handles.direction=get(handles.direction, 'String');
6 handles.stateStartBelow=get(handles.chkStartBelow, 'Value');
7 if handles.stateStartBelow==0
8     coll=handles.sldBoundTip1Out;
9     row1=find(handles.frameObj(:,coll), 1);
10    handles.start1=[row1, coll];
11    handles.leng1Out=str2double(get(handles.leng1, 'String'));
12    handles.boundary1=bwtraceboundary(handles.frameObj, ...
13        handles.start1, 'N', 8, handles.leng1Out, handles.direction);
14    imshow(handles.frame); hold on;
15    plot(handles.boundary1(:,2), handles.boundary1(:,1), 'y', 'LineWidth',
16        2);
17    guidata(hObject,handles)
18 else
19     coll=handles.sldBoundTip1Out;
20     row1=find(handles.frameObj(:,coll), 1, 'last');
21     handles.start1=[row1, coll];
22     handles.leng1Out=str2double(get(handles.leng1, 'String'));
23     handles.boundary1=bwtraceboundary(handles.frameObj, ...
24         handles.start1, 'N', 8, handles.leng1Out, handles.direction);
25     imshow(handles.frame); hold on;
26     plot(handles.boundary1(:,2), handles.boundary1(:,1), 'y', 'LineWidth',
27         2);
28     guidata(hObject,handles)
29 end
30 end

```

Fonte: Elaborado pelo autor.

Ao se alterar o controle de direção de detecção da borda, de anti-horário para horário ou vice-versa, a borda é desenhada da mesma forma, com os mesmos inputs anteriores, Figura B.7. Outra característica desta detecção é a possibilidade de se utilizar como início da borda não o primeiro mas o último ponto detectado quando a linha vertical é posicionada sobre o objeto, recurso que também é disponibilizado na interface.

Figura B.7 – Código para alteração na direção de detecção da borda.

```

1 function dir1_SelectionChangedFcn(hObject, eventdata, handles)
2 % object in dir1
3 %
4 %
5 if (hObject == handles.clock1)
6     handles.direction=get(handles.clock1, 'String');
7     handles.leng1Out=str2double(get(handles.leng1, 'String'));
8     handles.boundary1=bwtraceboundary(handles.frameObj, ...
9     handles.start1, 'N', 8, handles.leng1Out, handles.direction);
10    imshow(handles.frame); hold on;
11    plot(handles.boundary1(:,2), handles.boundary1(:,1), 'y', 'LineWidth',
12);
13    guidata(hObject,handles)
14 else
15     handles.direction=get(handles.counter1, 'String');
16     handles.leng1Out=str2double(get(handles.leng1, 'String'));
17     handles.boundary1=bwtraceboundary(handles.frameObj, ...
18     handles.start1, 'N', 8, handles.leng1Out, handles.direction);
19     imshow(handles.frame); hold on;
20     plot(handles.boundary1(:,2), handles.boundary1(:,1), 'y', 'LineWidth',
21);
22     guidata(hObject,handles)
23 end

```

Fonte: Elaborado pelo autor.

A função usada para traçar a linha vertical de detecção dos pontos iniciais das bordas é mostrada na Figura B.8. Esta função, referente à primeira borda, também é equivalente à função de detecção da segunda borda, por este motivo não é apresentada neste apêndice.

Figura B.8 – Código para detecção do ponto inicial da borda.

```

1 function sldBoundTip1_Callback(hObject, eventdata, handles)
2 handles.sldBoundTip1Out=round(get(handles.sldBoundTip1, 'Value'));
3 imshow(handles.frame)
4 hold on
5 plot([handles.sldBoundTip1Out handles.sldBoundTip1Out], [1 handles.camRes(1)],
6 'g', 'linewidth', 1.5)
7 guidata(hObject,handles)
8 end

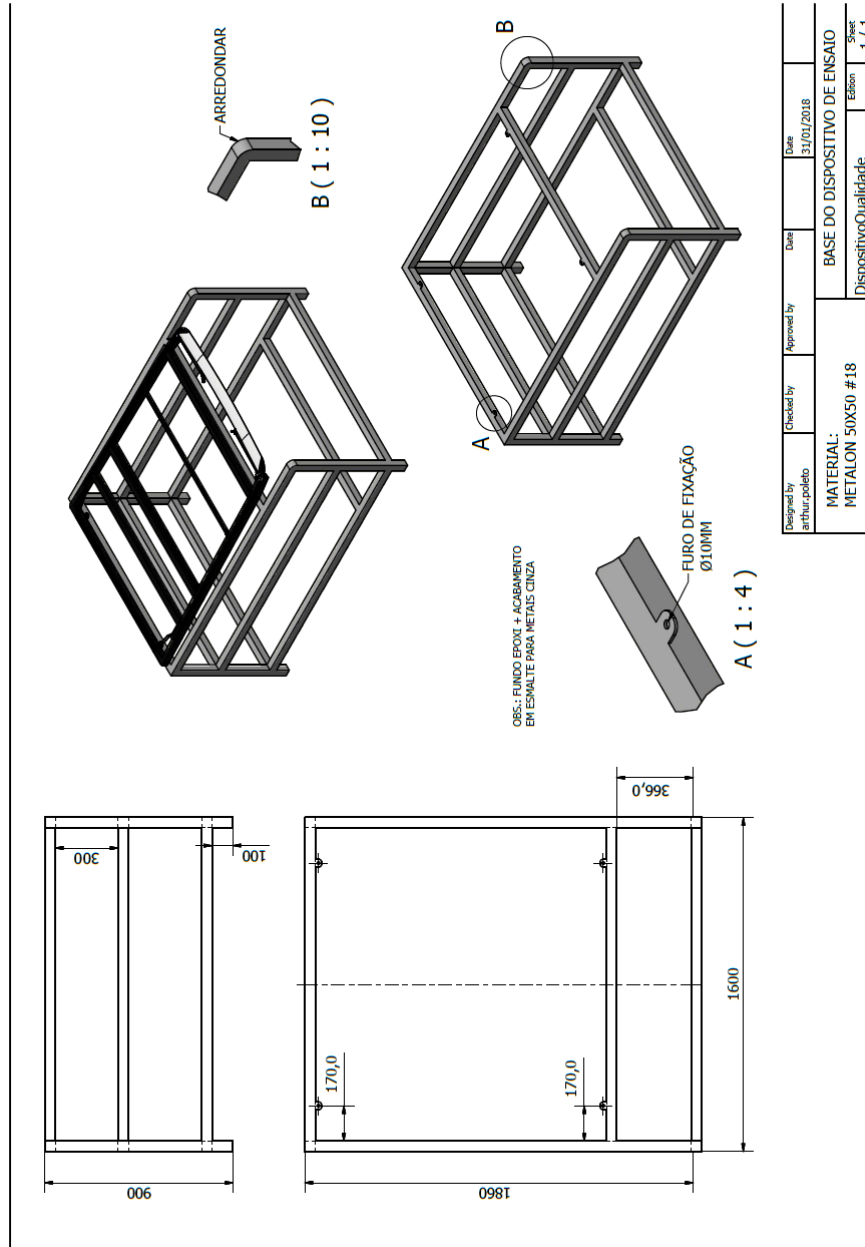
```

Fonte: Elaborado pelo autor.

Esta aplicação se mostrou bem confiável na medição *non contact* de ângulos nos objetos presentes na cena. A simplicidade também é uma característica deste sistema devido ao fato da necessidade de se utilizar apenas uma câmera digital USB conectada ao notebook. Neste caso utilizou-se uma câmera USB com resolução de 640x480 pixels.

APÊNDICE C - DESENHO DO DISPOSITIVO METÁLICO

Figura C.1 – Desenho do dispositivo desenvolvido.



Fonte: Elaborado pelo autor.

APÊNDICE D - INVARIANTES DE MOMENTO

O momento bidimensional de ordem $(p + q)$ de uma função que tem uma distribuição de densidade $f(x, y)$ é definido em termos da integral de Riemann como:

$$m_{p,q} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x^p y^q f(x, y) dx dy \quad (1)$$

Onde $p, q = 0, 1, 2, \dots$

A integral é calculada em cima da área correspondente ao objeto se considerarmos $f(x, y)$ uma função limitada baseada em pixels e que possui valores diferentes de zero somente em uma área finita do plano xy .

Caso considerarmos ainda uma função $f(x, y)$ onde a intensidade da área finita atinja somente valores *um*, ou seja, uma imagem binária, tem-se o momento geral como segue:

$$f(x, y) = \begin{cases} 1 & \text{Objeto} \\ 0 & \text{Background} \end{cases} \quad (2)$$

$$m_{p,q} = \sum_x \sum_y x^p y^q \quad (3)$$

Desta forma o termo $f(x, y)$ pode ser desconsiderado nas equações de momento apresentadas a seguir. Os momentos são classificados em ordens e as ordens dos momentos dependem dos índices p e q do momento $m_{p,q}$. Serão mostrados neste apêndice apenas os momentos até a ordem 2,

O momento de ordem zero descreve a área de um objeto, que corresponde à soma de todos os pixels do objeto em uma imagem binária.

$$m_{0,0} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) dx dy \quad (4)$$

Os momentos de primeira ordem guardam as informações do centroide de um objeto, que pode ser encontrado dividindo estes momentos pela área.

$$\begin{aligned} m_{1,0} &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} xf(x,y)dxdy \\ m_{0,1} &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} yf(x,y)dxdy \end{aligned} \quad (5)$$

Coordenadas do centroide de um objeto:

$$\begin{aligned} x_c &= \frac{m_{1,0}}{m_{0,0}} \\ y_c &= \frac{m_{0,1}}{m_{0,0}} \end{aligned} \quad (6)$$

Os momentos de segunda ordem são utilizados para o cálculo da orientação, ou inclinação de um objeto, seus eixos inerciais principais, sua excentricidade e outras características, através de seus momentos centrais.

$$\begin{aligned} m_{2,0} &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x^2 f(x,y)dxdy \\ m_{0,2} &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} y^2 f(x,y)dxdy \\ m_{1,1} &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} xyf(x,y)dxdy \end{aligned} \quad (7)$$

A definição de momento central é mostrada com a redução nas coordenadas espaciais com as coordenadas do centroide. Com a translação do objeto nas coordenadas espaciais, os momentos centrais não variam.

$$\mu_{p,q} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} (x - x_c)^p \cdot (y - y_c)^q \cdot f(x,y) \cdot dxdy \quad (8)$$

De onde obtém-se diretamente, $\mu_{0,0} = m_{0,0}$ e $\mu_{1,0} = \mu_{0,1} = 0$, que são os momentos centrais de ordem zero e de primeira ordem.

O momento central de qualquer ordem pode ser diretamente dos momentos espaciais, com $(p + q) \geq 1$.

$$\mu_{p,q} = \frac{m_{p,q}}{m_{0,0}} - \left(\left(\frac{m_{1,0}}{m_{0,0}} \right)^p \left(\frac{m_{0,1}}{m_{0,0}} \right)^q \right) \quad (9)$$

Da Equação (9), os momentos centrais de primeira ordem são definidos.

$$\begin{aligned} \mu_{1,0} &= \frac{m_{1,0}}{m_{0,0}} - \left(\frac{m_{1,0}}{m_{0,0}} \right) = 0 \\ \mu_{0,1} &= \frac{m_{0,1}}{m_{0,0}} - \left(\frac{m_{0,1}}{m_{0,0}} \right) = 0 \end{aligned} \quad (10)$$

Da mesma forma, os momentos centrais de segunda ordem são calculados.

$$\begin{aligned} \mu_{2,0} &= \frac{m_{2,0}}{m_{0,0}} - \left(\frac{m_{1,0}}{m_{0,0}} \right)^2 = \frac{m_{2,0}}{m_{0,0}} - x_c^2 \\ \mu_{0,2} &= \frac{m_{0,2}}{m_{0,0}} - \left(\frac{m_{0,1}}{m_{0,0}} \right)^2 = \frac{m_{0,2}}{m_{0,0}} - y_c^2 \\ \mu_{1,1} &= \frac{m_{1,1}}{m_{0,0}} - \left(\left(\frac{m_{1,0}}{m_{0,0}} \right) \left(\frac{m_{0,1}}{m_{0,0}} \right) \right) = \frac{m_{1,1}}{m_{0,0}} - x_c y_c \end{aligned} \quad (11)$$

O tensor inercial J é composto a partir dos três momentos centrais de segunda ordem, com ele são calculados os autovalores que correspondem aos eixos principais do objeto.

$$J = \begin{bmatrix} \mu_{2,0} & \mu_{1,1} \\ -\mu_{1,1} & \mu_{0,2} \end{bmatrix} \quad (12)$$

Dado que $\det(\lambda I - J) = 0$, tem-se que os autovalores, ou seja, os eixos principais são:

$$\lambda_{1,2} = \sqrt{\frac{1}{2}(\mu_{2,0} + \mu_{0,2}) \pm \sqrt{4\mu_{1,1}^2 - (\mu_{2,0} - \mu_{0,2})^2}} \quad (13)$$

O autovetor do tensor inercial com o menor autovalor é correspondente à orientação do objeto. Onde θ é o ângulo entre o eixo maior do objeto e o eixo horizontal x .

$$\theta = \frac{1}{2} \arctan\left(\frac{2\mu_{1,1}}{\mu_{2,0} - \mu_{0,2}}\right) \quad (14)$$

Tabela D.1 – Orientação θ do objeto

$\mu_{2,0} - \mu_{0,2}$	$\mu_{1,1}$	θ	<i>Intervalo</i>
<i>Zero</i>	<i>Zero</i>	0°	–
<i>Zero</i>	<i>Positivo</i>	$+45^\circ$	–
<i>Zero</i>	<i>Negativo</i>	-45°	–
<i>Positivo</i>	<i>Zero</i>	0°	–
<i>Negativo</i>	<i>Zero</i>	-90°	–
<i>Positivo</i>	<i>Positivo</i>	$\frac{1}{2} \arctan\left(\frac{2\mu_{1,1}}{\mu_{2,0} - \mu_{0,2}}\right)$	$0^\circ < \theta < +45^\circ$
<i>Positivo</i>	<i>Negativo</i>	$\frac{1}{2} \arctan\left(\frac{2\mu_{1,1}}{\mu_{2,0} - \mu_{0,2}}\right)$	$-45 < \theta < 0^\circ$
<i>Negativo</i>	<i>Positivo</i>	$\frac{1}{2} \arctan\left(\frac{2\mu_{1,1}}{\mu_{2,0} - \mu_{0,2}}\right) + 90^\circ$	$+45^\circ < \theta < +90^\circ$
<i>Negativo</i>	<i>Negativo</i>	$\frac{1}{2} \arctan\left(\frac{2\mu_{1,1}}{\mu_{2,0} - \mu_{0,2}}\right) - 90^\circ$	$-90^\circ < \theta < -45^\circ$

Fonte: KILIAN, 2001.

Pode-se obter a excentricidade o objeto diretamente dos momentos centrais de segunda ordem por:

$$\varepsilon = \frac{(\mu_{2,0} - \mu_{0,2})^2 - 4\mu_{1,1}^2}{(\mu_{2,0} + \mu_{0,2})^2} \quad (15)$$

No entanto, considerando os semieixos maior e menor, respectivamente a e b , a excentricidade é calculada da forma que segue:

$$\varepsilon = \frac{\sqrt{a^2 - b^2}}{a} \quad (16)$$

Até neste momento, os descritores baseados em momento são capazes de descrever bem as formas dos objetos, mantendo-se invariantes quanto à orientação e translação do objeto na matriz espacial, porém mantêm uma relação de dependência quanto ao tamanho do objeto. Isto pode ser entendido como a formação de imagens de um objeto de diferentes distâncias em relação à câmera. Quanto mais próximo o objeto estiver da câmera, maior será o tamanho deste na imagem.

Neste trabalho não foi realizada nenhuma mudança em relação ao tamanho do objeto, no entanto será apresentada a formulação do momento central normalizado, que permite a identificação do objeto com diferentes fatores de escala.

$$v_{p,q} = \frac{\mu_{p,q}}{m_{0,0} \binom{p+q+2}{2}} \quad (17)$$

Note que para realizarmos a comparação do objeto com diferentes escalas, também é necessário utilizarmos um fator de escala para o momento. Normalmente este fator de escala utilizado é a área do objeto, dividimos o momento central por potencias de $A = m_{0,0}$ para encontrarmos o momento central normalizado $v_{p,q}$.

A partir do momento (A.3) serão mostrados abaixo os momentos em forma de somatório das ordens que são utilizadas neste trabalho. Novamente, para a área do objeto tem-se:

$$m_{0,0} = A = \sum_x \sum_y x^0 y^0 \quad (18)$$

Momento de primeira ordem:

$$m_{1,0} = \sum_x \sum_y x^1 y^0 \quad m_{0,1} = \sum_x \sum_y x^0 y^1 \quad (19)$$

Momento de segunda ordem:

$$m_{2,0} = \sum_x \sum_y x^2 y^0 \quad m_{0,2} = \sum_x \sum_y x^0 y^2 \quad m_{1,1} = \sum_x \sum_y x^1 y^1 \quad (20)$$