



UNIVERSIDADE FEDERAL DE MINAS GERAIS

**PROGRAMA DE PÓS-GRADUAÇÃO EM
ENGENHARIA MECÂNICA**

**DESENVOLVIMENTO DE UM SISTEMA DE VISÃO DE BAIXO
CUSTO UTILIZANDO UM MANIPULADOR ROBÓTICO
INDUSTRIAL VISANDO A AUTOMATIZAÇÃO DE PROCESSO
DE SOLDAGEM**

FAGNER GUILHERME FERREIRA COELHO

Belo Horizonte, 30 de Maio de 2016.

FAGNER GUILHERME FERREIRA COELHO

**DESENVOLVIMENTO DE UM SISTEMA DE VISÃO DE BAIXO
CUSTO UTILIZANDO UM MANIPULADOR ROBÓTICO
INDUSTRIAL VISANDO A AUTOMATIZAÇÃO DE PROCESSO
DE SOLDAGEM**

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia Mecânica da Universidade Federal de Minas Gerais, como requisito parcial à obtenção do título de Mestre em Engenharia Mecânica.

Área de concentração: Processos de Fabricação

Orientador: Prof. Dr. Eduardo José Lima II

Departamento de Engenharia Mecânica, UFMG

Belo Horizonte

Escola de Engenharia da UFMG

2016



**Universidade Federal de Minas Gerais
Programa de Pós-Graduação em Engenharia Mecânica**

Av. Antônio Carlos, 6627 - Pampulha - 31.270-901 - Belo Horizonte – MG
Tel.: +55 31 3499-5145 - Fax.: +55 31 3443-3783
www.demec.ufmg.br - E-mail: cpgmec@demec.ufmg.br

**DESENVOLVIMENTO DE UM SISTEMA DE VISÃO DE
BAIXO CUSTO UTILIZANDO UM MANIPULADOR
ROBÓTICO INDUSTRIAL VISANDO A AUTOMATIZAÇÃO
DE PROCESSO DE SOLDAGEM**

FAGNER GUILHERME FERREIRA COELHO

Dissertação defendida e aprovada em 30 de Maio de 2016, pela Banca Examinadora designada pelo Colegiado do Programa de Pós-Graduação em Engenharia Mecânica da Universidade Federal de Minas Gerais, como parte dos requisitos necessários à obtenção do título de "**Mestre em Engenharia Mecânica**", na área de concentração de "**Processos de Fabricação**".

Prof. Dr. Eduardo José Lima II - UFMG - Orientador

Prof. Dr. Alexandre Queiróz Bracarense - UFMG – Examinador

Prof. Dr. Carlos Antônio Rennó - CEFET-MG – Examinador

AGRADECIMENTOS

Agradecimentos devem ser prestados a todos que contribuíram de forma direta e indireta, seja na confecção deste trabalho acadêmico, seja na formação de um cidadão de bem ao qual me tornei.

Com isso, inicio os agradecimentos a Deus nosso criador, aos meus pais: Geralda e Eustáquio, meus irmãos: Fernando, Flaviana e Flávio, minha esposa: Fabiana, e minha amada filha Letícia. A toda minha família, peço desculpas pelos momentos de ausência, e pelas faltas cometidas.

Aos colegas e amigos que adquiri na minha jornada na UFMG, são eles, do Departamento de Ciência da Computação: Bernardo Biesseck, Daniel Balbino pelas inúmeras dúvidas sanadas. Aos amigos do Departamento de Engenharia Mecânica: Marcelo Bomfim, Luciano Juliani, Ariel Rodríguez, Gabriel Mendes, pelas informações e ajudas, e Jonathas Henrique pelos encontros fora de hora. Todos foram de muita importância, me incentivando a continuar. Aos professores Eduardo José Lima II e Alexandre Queiróz Bracarense pelos ensinamentos, oportunidades, orientações, entusiasmos e pela amizade construída.

Aos professores Alexandre Queiroz Bracarense, e Eduardo José Lima II, mais uma vez, por terem aceitado e comprado à ideia apresentada, e por terem cedido todos os recursos necessários para a realização deste, que além de ser um projeto é um sonho que finalmente torna-se realidade.

Aos professores Alexandre Queiroz Bracarense do corpo docente do Departamento de Engenharia Mecânica da UFMG, e Carlos Antônio Rennó do corpo docente do Departamento de Eletrônica e Instrumentação Biomédica do CEFET/MG, por aceitarem o convite de participar de minha banca examinadora.

Meus sinceros agradecimentos.

"Bem aventurado o homem que acha sabedoria, e o homem que adquire conhecimento."

(Provérbios 3:13)

"Superação é ter a humildade de aprender com o passado, não se conformar com o presente e desafiar o futuro."

(Hugo Bethlem)

"Ninguém é tão grande que não possa aprender, nem tão pequeno que não possa ensinar."

(Píndaro)

SUMÁRIO

LISTA DE FIGURAS	7
LISTA DE TABELAS E QUADROS	11
LISTA DE ABREVIATURAS E TERMOS	12
RESUMO	15
1. INTRODUÇÃO	16
1.1 Relevância do tema	18
1.2 Objetivos.....	18
1.3 Motivação	19
1.4 Estrutura da dissertação	20
2. REVISÃO BIBLIOGRÁFICA	21
2.1 Sistemas de visão.....	21
2.1.1 Visão computacional	23
2.1.2 Sistema de visão humano e o sistema de visão artificial	27
2.2 Aquisição de imagens	29
2.3 <i>Pixels</i> e resolução	33
2.3.1 Digitalização de imagens.....	36
2.4 Processamento de imagens.....	39
2.4.1 Qualidade de imagens	41
2.4.2 Detecção de pontos, linhas e bordas em imagens	42
2.5 Filtros	46
2.6 Calibração de câmeras	49
2.6.1 Calibração intrínseca	52
2.6.2 Calibração extrínseca.....	53
2.7 Manipuladores robóticos	54
2.7.1 Graus de liberdade.....	55

2.7.2	Geração de trajetória	56
2.7.3	Soldagem robotizada.....	62
2.8	Trabalhos relacionados	63
3.	MATERIAIS E MÉTODOS.....	67
3.1	Método	67
3.2	Materiais.....	71
4.	RESULTADOS.....	76
4.1	Seleção da câmera.....	76
4.2	Calibração do sistema	83
4.3	Análise da aplicabilidade dos algoritmos de visão computacional	88
4.4	Funcionamento do programa	94
4.5	Realização dos ensaios.....	102
5.	ANÁLISE DOS RESULTADOS.....	111
6.	CONCLUSÕES	116
	ABSTRACT	118
7.	SUGESTÃO PARA TRABALHOS FUTUROS.....	119
8.	REFERÊNCIAS	120
8.1	Sítios visitados na internet	124

LISTA DE FIGURAS

FIGURA 2.1 - O olho humano.....	21
FIGURA 2.2 - Formação da imagem.....	22
FIGURA 2.3 - Relação das distâncias na formação da imagem no olho humano.....	22
FIGURA 2.4 - Sistemas de calibração.....	24
FIGURA 2.5 - Funcionamento de lentes convergentes.....	26
FIGURA 2.6 - Tamanhos da imagem.....	26
FIGURA 2.7 - Aquisição de imagem.....	31
FIGURA 2.8 - Conectividade dos pixels vizinhos ao pixel central.....	33
FIGURA 2.9 - Resolução de um <i>Megapixel</i>	34
FIGURA 2.10 - Resolução 640 x 480 pixels.....	34
FIGURA 2.11 – Avaliação na alteração da resolução de <i>pixels</i> em uma imagem.....	34
FIGURA 2.12 - Exemplo de conectividade.....	35
FIGURA 2.13 - Imagem 5x5 com 8 níveis de cinza.....	36
FIGURA 2.14 - Convenção utilizada em processamento de imagens.....	37
FIGURA 2.15 - Etapas fundamentais em processamento de imagens.....	39
FIGURA 2.16 - Detecção de pontos isolado a partir de um fundo constante.....	43
FIGURA 2.17 - Detecção de linhas na horizontal, inclinada a +/-45 graus e vertical.....	43
FIGURA 2.18 - Reta genérica representada no espaço (<i>x</i> , <i>y</i>), e no espaço (<i>a</i> , <i>b</i>).....	44
FIGURA 2.19 - Representação do centroides e <i>Bounding Box</i>	46
FIGURA 2.20 - Máscaras de convolução - SOBEL.....	48
FIGURA 2.21 - Transformação de coordenadas.....	49
FIGURA 2.22 - Configuração básica para a calibração flange-câmera.....	51
FIGURA 2.23 - Sistemas de transformação homogênea.....	52
FIGURA 2.24 - Sistema de referência em uma matriz mxn.....	52
FIGURA 2.25 - Representação das juntas do manipulador ASE.....	56
FIGURA 2.26 - Localização de um objeto em posição e orientação.....	57
FIGURA 2.27 - Acuidade e repetibilidade.....	59
FIGURA 2.28 - Diferentes modos de trajetória.....	60
FIGURA 3.1 - Calibração da junta 2 com acelerômetros.....	67
FIGURA 3.2 - Sequencia desenvolvida.....	68
FIGURA 3.3 - Representação esquemática do sistema proposto.....	69

FIGURA 3.4 - Tela principal do <i>software MatLab</i> [®]	74
FIGURA 3.5 - Tela principal do <i>software Mach3</i> [®]	74
FIGURA 3.6 - Ambiente de trabalho.	75
FIGURA 4.1 - Distorções em uma lente.	77
FIGURA 4.2 - <i>Checkerboard</i> (tabuleiro de damas).	78
FIGURA 4.3 - Definindo pontos de calibração na cena.	79
FIGURA 4.4 - Mosaico de imagens para câmera A.	81
FIGURA 4.5 - Vértices dos quadrados da câmera A	81
FIGURA 4.6 - Mosaico de imagens para a câmera B.	82
FIGURA 4.7 - Vértices dos quadrados para a câmera B.	83
FIGURA 4.8 - Definição do centro da câmera.....	84
FIGURA 4.9 - Conversão pixels para milímetros	84
FIGURA 4.10 - Definição da orientação da peça em relação as coordenadas da imagem.	86
FIGURA 4.11 - Calibrando o manipulador.....	87
FIGURA 4.12 - Calibrando o sistema.	88
FIGURA 4.13 - Diagrama de desenvolvimento do sistema proposto.....	88
FIGURA 4.14 - Testes para definição do limiar.	89
FIGURA 4.15 - Passo a passo para obtenção do contorno da peça.	90
FIGURA 4.16 - Uso da Transformada HOUGH para obtenção do contorno da peça.	90
FIGURA 4.17 - Tratando imagem obtida como YUY2 e limiar de 130.....	91
FIGURA 4.18 - Tratando imagem obtida como RGB e limiar de 130.	91
FIGURA 4.19 - Tratando imagem obtida como RGB e limiar de 140.	91
FIGURA 4.20 - Tratando imagem obtida como RGB e limiar de 150.	92
FIGURA 4.21 - Definindo a trajetória do manipulador.	93
FIGURA 4.22 - Processo de enchimento de chanfro.	94
FIGURA 4.23 - Processo de Obtenção e processamento da imagem.	95
FIGURA 4.24 - Programa gerenciador dos movimentos.	96
FIGURA 4.25- Imagem do ambiente a ser capturada.	96
FIGURA 4.26 - Imagens com e sem o bocal da tocha.	96
FIGURA 4.27 - Esqueletização do objeto.....	97
FIGURA 4.28 - Esqueletização do objeto.....	97
FIGURA 4.29 - Binarização do objeto encontrado.	98
FIGURA 4.30 - Algoritmo desenvolvido.....	99
FIGURA 4.31 - Obtenção do objeto na imagem e suas coordenadas em milímetros.	99

FIGURA 4.32 - Programa de geração de trajetória em <i>MatLab</i> [®]	101
FIGURA 4.33 - Código G gerado através do <i>MatLab</i> [®]	102
FIGURA 4.34 - Configuração inicial do sistema.	102
FIGURA 4.35 - Resultado da falta de um filtro adequado.....	103
FIGURA 4.36 - Sistema de visão e iluminação.	103
FIGURA 4.37 - Sistema de visão e iluminação atual.	104
FIGURA 4.38 - Mesa e sua área útil.....	104
FIGURA 4.39 - Extração da peça e definição dos pontos.	105
FIGURA 4.40 - Determinando o contorno da peça e suas coordenadas.....	105
FIGURA 4.41 - Avaliando o comportamento do algoritmo. Resultado ruim.....	106
FIGURA 4.42 - Avaliando o comportamento do algoritmo. Resultado bom.	106
FIGURA 4.43 - Avaliando o algoritmo sob alterações da iluminação do ambiente.....	106
FIGURA 4.44 - Testes realizados após substituição das lâmpadas do ambiente.....	107
FIGURA 4.45 - Testes realizados no algoritmo com reaproveitamento de peça.....	107
FIGURA 4.46 - Mesa com desgaste da pintura.	108
FIGURA 4.47 - Erro na definição do contorno.....	108
FIGURA 4.48 - Testes de soldagem sob uma nova configuração.	108
FIGURA 4.49 - Diferenciando as peças sob a nova configuração.....	109
FIGURA 4.50 - Definição dos pontos e ciclos de soldagem.	109
FIGURA 4.51 - Soldas realizadas sob a nova configuração.	110
FIGURA 4.52 - Resultados em escala	110
FIGURA 5.1 - Manipulador com cinco graus de liberdade.	111
FIGURA 5.2 - Identificando a posição da peça e gerando a trajetória do manipulador.	112
FIGURA 5.3 - Condições iniciais do sistema	112
FIGURA 5.4 - Robustez do sistema desenvolvido.	113
FIGURA 5.5 - Uso de sargento para manter a planicidade da mesa.....	114
FIGURA 5.6 - Correção ângulo de ataque da tocha.	115

LISTA DE TABELAS E QUADROS

QUADRO 2.1 - Visão humana versus visão artificial	27
QUADRO 2.2 - Propriedades do <i>Regionprops</i>	45
QUADRO 3.1 - Lista de materiais e equipamentos	71
QUADRO 4.1 - Resultado dos testes realizados com as <i>WebCam's</i>	80
TABELA 4.1 - Determinando o fator de conversão conforme imagens obtidas	85
TABELA 4.2 - Coordenada dos pontos	86
QUADRO 4.2 - Comparaçao dos resultados na aplicação dos filtros	93

LISTA DE ABREVIATURAS E TERMOS

A – Corrente em Ampères.

ABB – Asea Brown Bovery.

Alpha_c e *Alpha_c_error* - Coeficiente de defasamento da imagem e sua incerteza.

Blobs - Binary Large Object - Grande objeto binário

Bounding Box (BB) - Caixa que delimitadora de *pixels*.

Buffer Memory - Memória temporária para armazenamento de dados.

C.C. – Corrente Continua.

CCD - *Charged Coupled Device*. Captura imagem, armazena no *buffer memory*, converte em digital.

Checkerboard / Chessboard - Elemento utilizado para realizar calibração de câmeras.

Clustering - Aglomeração de *pixels* em uma determinada região da imagem.

CMOS - *Complementary Metal Oxide Silicon*. Captura imagem já em formato digital.

CNC – Controle Numérico de Comando de máquinas ferramentas.

DBCP – Distância Bico de Contato Peça.

DEMEC - Departamento de Engenharia Mecânica da Universidade Federal de Minas Gerais.

Distortion: (kc e kc_error) - Coeficiente de distorção e suas incertezas.

Drives - Dispositivos que permitem controle e interconexão entre elementos eletrônicos.

End effector - Dispositivo ou ferramenta montada na última junta de um manipulador robótico.

Focal Length (f ou fc e fc_error) - Distância focal em *pixels* e sua incerteza.

FCAW – *Flux-Cored Arc Weding* (Processo de soldagem a arco com arame tubular).

GMAW – *Gas Metal Arc Weding* (Processo de soldagem a arco com arame sólido).

GTAW - *Gas Tungsten Arc Welding* (Processo de soldagem a arco com eletrodo não consumível).

G-Code – Código usado para programação de máquinas ferramentas.

Hardware - Parte física do computador, ou que são utilizados para agregar a um sistema.

Hard Disk - Dispositivo não volátil para armazenamento de informações.

HSI - Modelo de processamento de imagens digitais, baseado na matriz, saturação e intensidade.

kHz – quilo Hertz.

LBW - *Laser Beam Welding and Cutting* (Processo de soldagem e corte a *laser*).

LRSS – Laboratório de Robótica Soldagem e Simulação.

MatLab[®] – *Matrix Laboratory* (*Software* para análise numérica).

Memory Stick - Dispositivo não volátil para armazenamento de informações.

mm – Comprimento em milímetros.

mm/s – Velocidade em milímetros por segundo.

ms – Tempo em milissegundos.

Overhauling - Termo utilizado ao processo de restauração que foi submetido o manipulador ASEA.

PAW - *Plasma Arc Welding and Cutting* (Processo de soldagem e corte a plasma).

Pick and Place - Tarefa em que um dispositivo automático, faz a manipulação de objetos.

Pinhole - Uma máquina fotográfica sem lente, utiliza uma caixa qualquer em que a luz não penetre.

Pixel - É o menor elemento que constitui uma imagem, quando esta é exibida em um dispositivo.

Pixel error (err) – Incerteza na medição da dimensão do *pixel*.

Principal Point (cc e cc_error) - Coordenadas, em *pixel*, do centro óptico da imagem, e sua incerteza.

PWM – *Pulse With Modulation* - (Modulação por Largura de Pulso).

Retrofitting - Termo utilizado para designar o processo de modernização de equipamentos.

Regionprops - Ferramentas do *MatLab[®]* que trata das características das regiões em uma imagem.

Region Growing - Promove o crescimento de regiões para detectar descontinuidades entre os *pixels*.

RGB - Processamento de imagem, que utiliza as três cores básicas, vermelho, verde e azul.

RPM - Rotações por minuto.

RSW - *Resistance Spot Welding* (Processo de soldagem por resistência elétrica por pontos).

SNR - Relação Sinal Ruído.

Snapshot - É uma expressão em inglês que significa foto instantânea.

Software - É um conjunto de instruções e componentes lógicos de um sistema processador de dados.

SVH - Sistema Visual Humano.

TCP – *Tool Center Point* (Ponto Central da Ferramenta).

Template Matching - Método utilizado para procurar descontinuidades em imagens.

TH – Transformada Hough.

Threshold - Processo de limiarização onde ocorre a conversão da imagem digital em binária.

Toolbox - Pacote de ferramentas adicionais do *MatLab®*.

UFMG - Universidade Federal de Minas Gerais.

USB - *Universal Serial Bus*, meio físico de interconexão e transferência de dados.

V – Tensão em Volts.

Webcam - Câmera de vídeo que permite capturar imagens e vídeos e transferi-los para o computador.

Wireless - Meio pelo qual utilizado para conectar equipamentos sem a necessidade de meio físico

YIQ - Processamento de imagens que se baseia na luminância, em fase, e quadratura (padrão *NTSC*).

RESUMO

O presente trabalho propõe-se agregar um sistema de visão de baixo custo ao processo de *Overhauling*, o qual foi submetido o manipulador robótico ASEA IRB6-S2 pertencente ao LRSS da UFMG. Desta forma, com o uso de elementos de visão que apresentem baixo investimento, associado a um sistema de controle desenvolvido e que funcione de forma confiável, e algoritmos que exigem a mínima interferência do operador, este processo visa apresentar uma solução bastante viável para o reaproveitamento de robôs que são inutilizados em chão de fábrica e podem ser doados a instituições acadêmicas para proporcionar estudos, pesquisas e desenvolvimento na área de robótica. Com isso, as técnicas de processamento de imagens digitais utilizadas neste trabalho e com uma *webcam* que compõe o sistema de visão computacional, que irá permitir que o manipulador robótico realize soldas após a identificação, localização e orientação de uma peça sobre uma mesa sem a necessidade de nenhum conhecimento prévio ou qualquer outro tipo de sensor que o auxilie. Com a aplicação de *softwares* para a análise matemática, algoritmos para processamento de imagem e de geração de trajetória, torna-se possível que o manipulador robótico realize a operação de soldagem sobre as coordenadas obtidas após o processamento da imagem capturada. O foco principal desta pesquisa, embasada no estudo do algoritmo proposto foi verificar não só o cumprimento da tarefa de posicionamento, bem como, a robustez frente às variações do entorno, demonstrando que o sistema apresente uma excelente relação custo-benefício.

Palavras Chaves: Robótica, Visão Computacional, Soldagem

1. INTRODUÇÃO

O mundo vem vivenciando um crescimento tecnológico em diversas áreas principalmente as que utilizam robôs. E estes se tornaram elementos fundamentais para a manufatura automatizada, com a sua repetibilidade, com destaque nas áreas que exigem elevado grau de precisão, confiabilidade e velocidade. Deste modo estudos relacionados ao desenvolvimento de sistemas de controle que utilizam estes dispositivos tem crescido nos últimos anos, e através do aumento dos recursos computacionais foi possível desenvolver robôs capazes de executar diversas tarefas que exigem algoritmos de controle mais complexos. Dos diversos manipuladores robóticos e seus sistemas utilizados em meio industrial, já é de conhecimento o elevado nível de atualização dos equipamentos, o que exige a renovação do parque industrial de modo a se manter competitivo no mercado fornecendo produtos de qualidade. Com isso, é inevitável o descarte de equipamentos que são definidos como "*ultrapassados*", mas que ainda apresentam condições de uso.

Máquinas ou equipamentos com tecnologia obsoleta podem ser atualizados ou reformados pela aplicação de técnicas modernas de automação e da utilização de componentes e dispositivos modernos. Este processo, conhecido como *retrofitting*, é muitas vezes uma solução para empresas que desejam aumentar a vida útil das máquinas, mantendo suas características periféricas com perfeito estado de conservação mecânica, segundo RIBEIRO *et al.* (2007). Mas no caso das grandes indústrias que utilizam da automação e de manipuladores, é comum a substituição dos modelos antigos por novos, a fim de tornar o sistema mais confiável, garantindo e até mesmo aumentando a produtividade, com redução de riscos operacionais. Devido ao baixo custo, os robôs descartados tornaram-se uma alternativa viável para universidades, laboratórios e centros de pesquisa, que atuam na área de robótica aplicada. Porém como estes equipamentos utilizam tecnologia obsoleta, alguns benefícios associados ao seu emprego para execução de tarefas podem não ser alcançados. Felizmente, a característica mecânica dos robôs industriais não vem sofrendo mudanças significativas ao longo do tempo. As principais diferenças entre os modelos antigos e os novos estão relacionados ao acionamento elétrico das juntas e ao sistema de controle, conforme LAGES *et al.* (2003), e LIMA II *et al.* (2004). Neste contexto, através de um processo de modernização tecnológica é possível agregar novas funcionalidades ao sistema robótico tornando a sua arquitetura de controle mais aberta e flexível como apresentado por BRUYNINCKX (2001), e LIPPIELLO *et al.* (2007).

O uso da visão computacional passou a ser um grande aliado para o desenvolvimento de pesquisas voltadas para a área de automação de processo, de modo que estes sistemas tenham a capacidade de tomar decisões através de informações extraídas do ambiente, garantindo uma maior autonomia. Segundo JAHNE e HAUBECKER (2000) a visão computacional é um assunto complexo, e comparando-o a um sistema biológico, é necessário dividi-lo para melhor compreendê-lo. Assim seu objetivo principal é prover ao computador as capacidades de percepção do sistema visual humano em relação ao ambiente SEBE *et al.* (2005).

Quando se faz uma comparação do olho humano e a interpretação feita pelo cérebro, respectivamente representados pela câmera e o computador, verifica-se que o trabalho realizado pelo cérebro nos parece muito simples e bastante elementar, porém uma análise mais detalhada mostra que ele a partir de uma grande quantidade de informações captada pelos olhos, é capaz de fazer o reconhecimento em apenas 150ms. Já os computadores atuais são na prática sistemas experimentais que nos ajudam a melhor compreender estas tarefas e consequentemente criar seu modelo e reproduzi-las, ALBUQUERQUE (2012).

Sistema de controle dotado de sensores de visão, como um todo, é definido como um sistema de processamento que monitora as “situações externas”, como presenças de obstáculos, mudanças nas dimensões e posições dos objetos no ambiente, permitindo que o manipulador possa reagir em situações inesperadas. Já as “situações internas” são relacionadas ao processo em si e se referem à velocidade de deslocamento, posicionamento, trajetória. Essa definição, por ser muito genérica, acaba por incluir todo o sistema de controle no sensor, apesar de outras definições tratarem este último apenas como o elemento de realimentação do sistema de controle.

Sistemas de visão são compostos basicamente por uma câmera e uma placa para digitalização das imagens, e de *softwares* que se encarregam de processar todos os algoritmos necessários para a realização das tarefas. Com a evolução destes sistemas, e a consequente redução nos preços das câmeras, a aplicação da visão computacional vem sendo utilizado em diversos segmentos. Podem-se destacar algumas aplicações industriais: controle metrológico, orientação de robôs, rastreabilidade de peças na linha de produção. Em processos de fabricação não seria diferente, uma vez que o operador, em sua grande maioria, está exposto diretamente ao processo, que exige o uso dos mais diversos equipamentos de proteção, ou que necessita de extrema atenção para manter a qualidade dos produtos, e sua própria integridade. Em uma linha de produção os manipuladores robóticos são capazes de desempenhar funções arriscadas ao manuseio humano,

em atividades repetitivas que podem levar o operador a fadiga, lentidão e subjetividade, e agregando a estes o sentido da visão, o processo tende a se tornar mais robusto. De acordo com FELICIANO *et al.* (2005), não há um sistema que resolva todos os desafios industriais, estes são específicos e configurados para cada aplicação. O desenvolvimento de sistemas de visão sólido em ambientes industriais com a necessária exatidão torna-se, portanto, um desafio.

O Laboratório de Robótica, Soldagem e Simulação da UFMG, que pertence ao DEMEC recebeu em 1998 uma doação pela Fachhochschule da Alemanha, um manipulador robótico, modelo ASEA IRB6-S2, fabricado pela ASEA (atual ABB), no ano de 1977, e que teve seu primeiro modelo lançado no mercado em 1974. Este foi o primeiro robô industrial totalmente elétrico produzido no mundo, já que até então, o acionamento era realizado por sistemas hidráulicos. Além disso, foi o primeiro manipulador a ser controlado por um microprocessor Intel® 8008. Este robô possui cinco graus de liberdade e capacidade de carga de 6 Kg, construído para aplicações de soldagem na indústria automobilística. Este equipamento passou por um processo de *overhauling*, como apresentado por BOMFIM (2013), onde apenas sua parte estrutural mecânica foi reaproveitada, e seu *hardware* (motores, sensores de posição, gabinete de controle e *softwares* geradores de trajetória), foram atualizados. Como comparativo, o processo de *retrofitting* visa atualizar os componentes eletromecânicos e eletrônicos do gabinete de controle, fazendo adaptações de forma a torná-lo intercambiável com alguns fabricantes de robôs. Este projeto propõe a integração de duas tecnologias, sistemas de controle para manipuladores robóticos e sistema de visão artificial, cujo objetivo é adotar um sistema de visão de baixo custo para controlar a trajetória de um robô remanufaturado que não utiliza linguagem proprietária desenvolvida pela ABB e sim pelo uso de algoritmos desenvolvidos para gerenciar sua cinemática, tornando-o capaz de realizar diferentes movimentações conforme a imagem capturada do produto, proporcionando autonomia e flexibilidade.

Para o desenvolvimento deste trabalho de pesquisa, foram utilizados algoritmos que basearam-se em diferentes conceitos teóricos encontrados na bibliografia no estado da arte, podendo citar um destes, o *Bounding Box*, BAREQUET (2001), que se mostrou mais adequado a suprir as necessidades da atividade proposta. Trata-se de um trabalho multidisciplinar e requer uma abordagem nas áreas de ótica (física e geométrica), eletrônica (*drives* de controle), e computação (algoritmos de visão, *drives* de comando e comunicação). A adequação de cada um dos tópicos acima citados para a aplicação em questão e sua interdependência representa uma interface única, cujos elementos serão apresentados nesta dissertação.

1.1 Relevância do tema

Nos últimos anos tem sido significativo o volume de investimento na área de automação dos processos pelas indústrias de produção em série. Esta técnica pode ser desde uma simples mecanização até a elevada automatização de dispositivos e células dedicadas. O uso de manipuladores robóticos empregados em meio industrial é elevado e tem como principal objetivo realizar atividades repetitivas, perigosas e que exigem elevada precisão, sendo estas realizadas com maior sucesso e rapidez, permitindo uma produção em grande escala e com maior qualidade. Aliado a esta tecnologia está o sistema de visão que vem para auxiliar o processo, de modo a exigir menos do operador que devido a grande jornada de trabalho, pode ser acometido por fadiga que comprometa a sua atividade, impactando na qualidade do produto final. Devido ao elevado custo adotado na indústria para automatizar seus processos, fica inviável transferir tal configuração para o meio acadêmico. Com isso, considera-se o uso de robôs industriais que se tornaram inservíveis para a indústria, mas não para o meio acadêmico. Com este equipamento remanufaturado dotado de linguagem não proprietária de seu fabricante, associado a um sistema de visão de baixo custo é possível desenvolver técnicas adequadas para manipulação e tratamento de imagens, viabilizando a tomada de decisões e o controle dos processos de fabricação e/ou outros equipamentos. As metodologias desenvolvidas podem ser então adaptadas para a utilização em controladores comerciais.

1.2 Objetivos

O objetivo principal do trabalho é avançar em relação à remanufatura realizada no manipulador ASEA, adaptando um sistema de visão computacional constituído por uma única câmera de baixo custo, fixada ao elemento final do manipulador robótico, de modo que a câmera faça parte da área de trabalho do mesmo. Com isso, objetos que estejam posicionados sobre uma mesa, podem ter sua dimensão e posição reconhecida pelo robô, para que então possa realizar uma atividade sobre a peça.

O trabalho conta com a reconstrução dos sistemas de controle e acionamento do manipulador robótico ao qual foi submetido por BOMFIM (2012), onde se desenvolveu e montou-se o *hardware* em um gabinete de controle e *software* desenvolvido com uma linguagem de programação não proprietária, de modo a deixar o sistema acessível a todos.

Os objetivos específicos para se alcançar o resultado esperado são:

1. Utilizar a imagem estática como fonte de dados, identificando posicionamento de peças sobre uma mesa e relacionando as coordenadas dos *pixels* com as coordenadas em milímetros do sistema mundo;
2. Minimizar a intervenção humana;
3. Realizar testes exaustivos, verificando posicionamento, dimensão, precisão, repetibilidade e robustez do algoritmo;
4. Obter as coordenadas, para geração da trajetória que é enviada diretamente para o robô através do *hardware* ligado do computador ao gabinete de controle, e possibilitar a atuação do manipulador;
5. Executar de cordões de solda, utilizando o processo GMAW/FCAW, de forma a validar o sistema;
6. Analisar e discutir os custos associados ao sistema de visão, e o quanto esse sistema pode ser simplificado a fim de baixar o custo, sem comprometer o desempenho e sua aplicabilidade na indústria.

A proposta fim deste trabalho é realizar a soldagem após a identificação da posição e orientação da mesma dentro de um espaço de trabalho limitado, bem definido e calibrado para as operações específicas. Sendo para estas condições o suficiente para a extração das informações necessárias, a análise de imagens bidimensionais, ou seja, sobre um plano (x, y), com o emprego de apenas uma única câmera. Limitando-se à verificação da posição e orientação do objeto no plano, não extraíndo informação de altura (z), costumeiramente utilizado em sistema de visão estéreo. Não se buscou a identificação e diferenciação de objetos na cena, estando o procedimento, limitado a analisar imagens contendo apenas um objeto que contenha dimensão retangular mínima de 100 x 100 milímetros.

1.3 Motivação

Apesar do sistema de visão ter limitações em suas aplicações, cada vez mais é observado à aplicação de sistemas de automação utilizando imagem em diferentes segmentos, não sendo diferente no ambiente industrial. Normalmente são sistemas dedicados, muitas vezes importados e com aplicações muito específicas.

Considerando o alto custo de elementos de sensoriamento voltados para a visão, conforme exposto anteriormente, e que se utiliza um robô de baixo custo devido ao processo de *overhauling* em que o mesmo foi submetido, este trabalho foi motivado a agregar um sistema que seja viável financeiramente. Para isso, é necessário o desenvolvimento de um sistema de visão de baixo custo sem comprometer a eficiência de seu funcionamento. Sendo assim o robô terá a capacidade de visualizar uma peça ou objeto de interesse independente do posicionamento e orientação, e utilizar esta informação para realizar uma tarefa, sem uso de outros elementos externos de posicionamento que possam ser utilizados para garantir a correta identificação e localização da peça na cena de interesse.

1.4 Estrutura da dissertação

Este texto de dissertação está estruturado de modo a transmitir informações e conhecimentos de forma que possibilite a compreensão do problema e sua resolução. O trabalho está dividido em 7 capítulos. O Capítulo 2 apresenta uma revisão da literatura, abordando o assunto de visão computacional, algoritmos de visão, manipuladores robóticos, e subitens necessários ao entendimento dos assuntos abordados. Neste mesmo capítulo apresenta-se a importância e passos implementados para o processamento de imagens digitais, além da definição de imagem e a representação de uma imagem digital e as técnicas de processamento de imagens. E por fim, foram citados alguns autores que vêm desenvolvendo pesquisas em áreas relacionadas com visão, robótica e soldagem, além de dar mais profundidade aos trabalhos que possuem uma relação estreita com o objetivo desta pesquisa. No Capítulo 3, são apresentadas a metodologia, o desenvolvimento do sistema, escolha dos componentes e a calibração do sistema. No Capítulo 4 são apresentados alguns resultados, sendo que o capítulo seguinte realiza uma análise que retoma os principais pontos abordados durante o trabalho e as conclusões obtidas a partir dos estudos realizados. No Capítulo 6 são apresentadas as conclusões em relação ao objetivo do trabalho. No Capítulo 7 são feitas sugestões e propostas para trabalhos futuros.

2. REVISÃO BIBLIOGRÁFICA

2.1 Sistemas de visão

Segundo SIEGWART *et al.* (2004), a visão é o sentido que os humanos mais utilizam, por meio dela recebe-se uma quantidade enorme de informação a respeito do ambiente que nos possibilita uma interação rica e inteligente dentro de um espaço dinâmico. A visão é responsável por cerca de 75% de nossa percepção e é resultado de operações ópticas, químicas e nervosas. Os olhos são órgãos responsáveis por captar informação luminosa e transformá-la em impulsos a serem decodificados pelo sistema nervoso.

Os elementos essenciais do olho humano, que estão apresentados na FIG. 2.1, possuem algumas funções fundamentais, sendo citadas aqui algumas delas: a refração, cujos elementos responsáveis são a córnea, o cristalino e os humores aquoso e vítreo; a auto focalização, que é de responsabilidade do cristalino e pelos músculos ciliares; o controle da luminosidade interna feito pela íris e pupila; a detecção, que está a cargo da retina (pelos cones e bastonetes) e a transmissão da informação ao cérebro, que é realizada pelo nervo e as vias ópticas, VENTURA (1995).

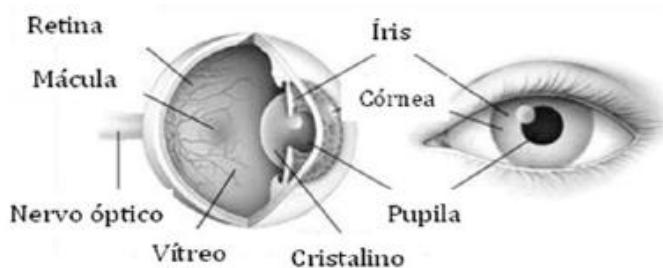


FIGURA 2.1 - O olho humano.

FONTE: <http://www.laboratoriorigor.com.br/anatomia.html>. Disponível em Nov./2015.

O olho humano forma a imagem no plano da retina por projeção da luz vinda do mundo 3D através de uma lente (cristalino) que muda de dimensões para ajustar a distância focal. Já para a imagem formada em uma câmera digital, as dimensões e definições do orifício são os responsáveis por caracterizar a nitidez e o plano focal. A principal diferença entre o cristalino e uma lente óptica é que a primeira é flexível. A FIG. 2.2 representa ambos sistemas.

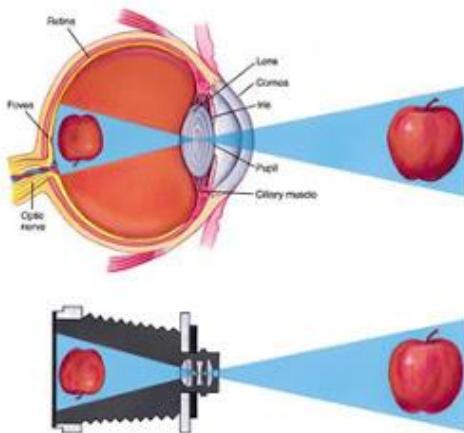


FIGURA 2.2 - Formação da imagem.

FONTE: <http://fisicaolohohumanond.blogspot.com.br/2012/03/formacao-da-imagem-no-olho-humano.html>. Disponível em Nov. /2015.

A distância entre o centro focal do cristalino e a retina varia de aproximadamente 14mm até cerca de 17mm, à medida que o poder de refração do cristalino aumenta do seu mínimo para o seu máximo. Quando o olho focaliza um objeto que está mais distante do que cerca de 3 metros, o cristalino apresenta o seu mais baixo poder de refração, em contrapartida quando o olho focaliza um objeto próximo, o cristalino está mais fortemente refrativo. Essa informação torna fácil o cálculo do tamanho da imagem de um objeto na retina. Na FIG. 2.3, o observador está olhando para um objeto de Y metros de altura e que está a X metros de distância. A percepção acontece pela relativa excitação dos receptores de luz, que transformam a energia radiante em impulsos elétricos que serão posteriormente decodificados pelo cérebro.

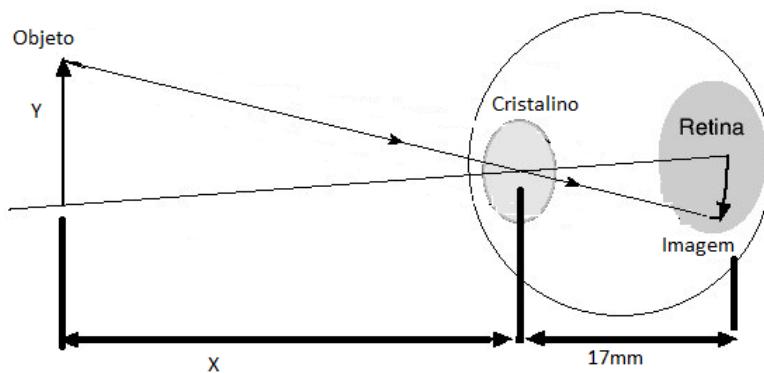


FIGURA 2.3 - Relação das distâncias na formação da imagem no olho humano.

FONTE: <http://www.ufpa.br/ensinofts/capitulo2.html>. Disponível em Out./2015.

Após a recepção da imagem, os impulsos elétricos devem ser interpretados pelo cérebro a fim de extrair informações como posições, cores e dimensões de objetos presentes na imagem.

2.1.1 Visão computacional

Grandes esforços vem sendo realizados para prover às máquinas a habilidade da visão humana, pois este sentido é uma das técnicas de detecção mais complexas, e vem sendo reproduzida para ser utilizada em sistemas robóticos e de automação. Enquanto sensores de pressão, proximidade, torque e força são importantes na melhoria do desempenho de sistemas de automação, a visão de máquina é admitidamente o sistema de sensoriamento de maior capacidade de medição e extração de informações, afirma FU (1987).

Visão artificial, visão de máquina, visão robótica ou visão computacional é o processo de extração, caracterização e interpretação de uma determinada informação a partir de imagens obtidas do mundo tridimensional. É a automatização e a integração de uma quantidade de processos e representações usadas para a percepção visual.

Segundo LENZ e TSAI (1989), quando um robô utiliza de uma câmera para estimar a posição e orientação de um objeto em relação à sua base, é necessário conhecer a localização relativa entre sua garra (elemento final) e a base do robô, entre a câmera e a garra e entre o objeto e a câmera. Diante destas informações é possível calcular a posição e orientação do objeto em relação à base do robô, que geralmente é também a referência do sistema de coordenadas do mundo. Conforme apresentado na FIG. 2.4, as tarefas que estimam essas três relações são conhecidas respectivamente como calibração do robô, calibração garra-câmera e calibração da câmera. Então a configuração sugerida pelos autores é que a câmera esteja instalada no *end effector*, ou seja, no elemento final do manipulador.

A calibração da garra do robô é obtida através dos cálculos das transformações dos sistemas de coordenadas de cada elo com a inclusão de suas cinemáticas inversa e direta. Com isso, é possível relacionar matematicamente as coordenadas relacionadas ao sistema coordenado da garra do manipulador robótico com as coordenadas de sua base. Por motivo de simplificação do problema e melhor desempenho, a literatura considera o sistema coordenado da base do robô como o de referência do mundo (origens coincidentes). Sempre que for mencionado sistema coordenado do mundo refere-se ao sistema de coordenadas de referência fixo, ou seja, ao qual estarão referenciadas todas as coordenadas dos pontos de interesse no resultado final, conforme VIANNA (2009).

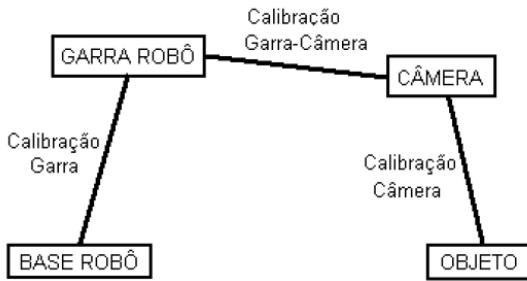


FIGURA 2.4 - Sistemas de calibração.

FONTE: VIANNA, 2009.

Diversas técnicas de reconhecimento de imagens tem sido apresentadas na literatura e geralmente são validadas através de protótipo de aplicações, pois em um ambiente industrial, raramente obtém-se as condições ideais de iluminação, contraste, posicionamento correto da peça e do ângulo de obtenção da imagem, além de outros fatores externos que dificultam a interpretação de uma cena, RUSS (2007).

Quando o sistema de visão é aplicado a um manipulador, em operações de *pick and place*, por exemplo, o objeto não precisa estar sempre na mesma localização (posição e orientação) para que possa executar sua tarefa adequadamente. Normalmente, em processos de soldagem, o objeto se encontra fixado sempre na mesma posição. Contudo, é interesse desse trabalho investigar a viabilidade econômica da utilização de um sistema de visão computacional para localização do objeto, assim como utilizado nas mencionadas operações de manipulação.

Uma imagem digital não é interpretada da mesma forma que um ser humano identifica uma cena, mas a partir de atributos que são extraídos da mesma e que estão relacionados entre si. Portanto o processo de tratamento de imagens, busca-se capturar algumas características e usá-los minimizar o espaço de pesquisa, conforme YHONG (1994).

Segundo TRUCCO e VERRI (1998), visão computacional, visão de máquina ou ainda visão robótica é um tema que relaciona diversas áreas de conhecimento, o que dificulta encontrar uma única definição. Seu conteúdo teórico passa por conceitos de inteligência artificial, robótica, processamento de sinais, reconhecimento de padrões, teoria de controle e neurociência. Mas LEE *et al.* (1986) definem visão computacional como um processo de extração, caracterização e interpretação da informação a partir de imagens do mundo tridimensional.

Utilizar-se de técnicas computacionais para estimar ou tornarem explícitas propriedades geométricas ou dinâmicas de imagens do ambiente que nos cerca, pode ser aplicada tanto na robótica fixa, quanto na móvel, em processos de medição, localização e identificação de objetos e determinação de suas formas geométricas, bem como no auxílio na navegação de robôs com o reconhecimento do ambiente, estimativa de velocidade, aceleração dentre outros.

A associação de inteligência e flexibilidade deste sistema, permite que a resposta se adeque de acordo com as variações do meio onde está implementado. Os três principais níveis, que auxiliam o uso da visão computacional mediante a área são:

➤ Baixo Nível:

Percepção - Aquisição da imagem.

Pré-Processamento - Redução de ruído e/ou realce de detalhes.

➤ Médio Nível:

Segmentação - Particionar a imagem em regiões de interesse.

Descrição - Determinar propriedades de objetos (tamanho, forma, etc.).

➤ Alto Nível:

Reconhecimento - Identificação de objetos a partir de um banco de dados.

As câmeras fotográficas podem ou não utilizar lentes, que são sistemas ópticos utilizados para focalizar e capturar determinada cena, sendo que os dispositivos que não possuem lentes, são aqueles com projeção central conhecidas como *pinhole*, que é um dispositivo constituído por um pequeno buraco permite que raios luminosos entrem, formando a imagem no material fotossensível, YOUNG (1989). As câmeras convencionais utilizam lentes esféricas biconvexas para capturar imagens, e estas fazem com que os raios luminosos que incidem paralelamente nelas sejam convergidos para um ponto. Para determinar a imagem de um objeto refletido precisa-se conhecer dois raios luminosos (r_1 e r_2) que incidem na lente. Como mostra a FIG. 2.5, o r_1 incide na lente passando pelo segundo foco da mesma (F2) e é refratado paralelamente ao seu eixo, já o r_2 é refratado passando pelo primeiro foco (F1), ALVARENGA *et al.* (2000).

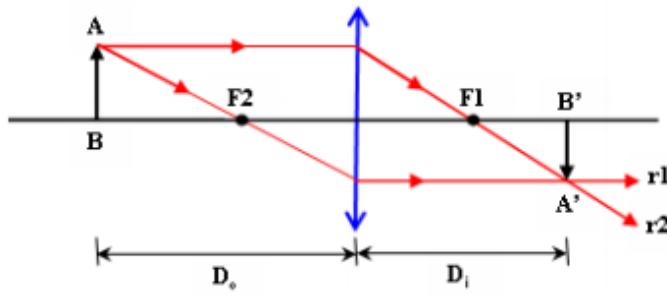


FIGURA 2.5 - Funcionamento de lentes convergentes.

FONTE: ALVARENGA, 2000.

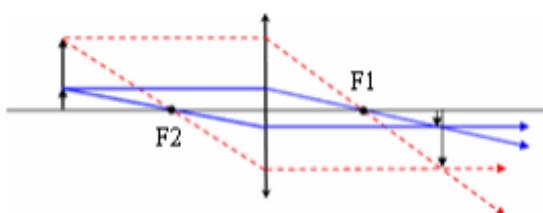
Com isso, para se determinar o tamanho da imagem de um objeto em relação à distância da lente, utiliza-se a EQUAÇÃO (2.1):

$$\frac{A'B'}{AB} = \frac{D_i}{D_o} \quad (2.1)$$

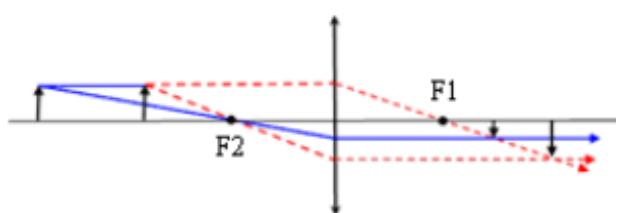
Onde $A'B'$ é o tamanho da imagem, AB é o tamanho do objeto, D_i e D_o são as distâncias entre a lente e a imagem, e entre a lente e o objeto, respectivamente. Conforme a EQUAÇÃO (2.2) D_i e D_o também podem ser relacionados com a distância focal da lente f , através da função de Gauss:

$$\frac{1}{f} = \frac{1}{D_i} + \frac{1}{D_o} \quad (2.2)$$

Na FIG. 2.6 (a) é exemplificada a variação do tamanho da imagem de um objeto. Através desta é possível ver que o tamanho da imagem do objeto pode variar de acordo com o tamanho do próprio objeto, ou variar devido à distância deste em relação à lente, como apresentado na FIG. 2.6(b). Neste último caso, quanto mais longe da lente estiver o objeto, menor será o tamanho da sua imagem.



(a) Variação do tamanho do objeto



(b) Variação da distância do objeto

FIGURA 2.6 - Tamanhos da imagem.

FONTE: ALVARENGA, 2000.

2.1.2 Sistema de visão humano e o sistema de visão artificial

Embora para o sistema de visão humano analisar e reconhecer objetos pareça ser trivial, esta pode ser uma tarefa complexa quando executada por um sistema de visão artificial. O grande desafio é fazer, com que um sistema artificial trabalhe em diferentes condições de luminosidade e contraste sem perder a capacidade de interpretar a cena, como ocorre com o sistema visual humano MARQUES et al. (1999). O QUADRO (2.1) lista as principais comparações entre os dois sistemas de visão.

QUADRO 2.1
Visão humana versus visão artificial

Característica	Sistema Visão Humano	Sistema Visão Artificial
Espectro de Luz	Limitado à faixa de luz visível (300nm a 700nm) do espectro de ondas.	Opera em praticamente todo o espectro eletromagnético.
Flexibilidade	Flexível, adapta-se a diferentes tarefas e condições de trabalho.	Inflexível. Bom desempenho apenas na tarefa para qual foi projetado.
Habilidade	Pode estabelecer estimativas relativamente precisas em assuntos subjetivos.	Pode efetuar medições exatas, baseadas em contagem de <i>pixels</i> , dependente da imagem digitalizada.
Cor	Possui capacidade de interpretação subjetiva de cores.	Mede objetivamente os valores de R, G e B para determinação de cor.
Sensibilidade	Capaz de se adaptar a diferentes condições de luminosidade, características físicas da superfície e distância do objeto. Limitado na distinção de diferentes níveis de cinza.	Sensível ao nível de iluminação, bem como à distinção em relação ao objeto e suas características físicas. Pode trabalhar com centenas de tons de cinza.
2D e 3D	Pode executar tarefas 3D e com múltiplos comprimentos de onda (dentro do espectro visível) facilmente.	Executa tarefas 2D com relativa facilidade, mas é lento e limitado em tarefas 3D.
Percepção	Sobre variações de brilho em escala logarítmica. A interpretação subjetiva de brilho depende da área em torno do objeto considerado.	Pode receber brilho em escala linear ou logarítmica.

FONTE: FORESTI, 2006.

O sistema de visão computacional é dividido em duas partes: *hardware* que é composto por equipamentos e circuitos eletrônicos destinados à aquisição, digitalização e armazenamento das imagens, e do *software* que possibilita a extração de informações desejadas. É possível que o processamento da imagem seja realizada no próprio *hardware* com uso de circuitos programáveis, o que proporciona respostas mais rápidas e aumenta a eficiência do processo. Este tipo de sistema não será abordado deste trabalho.

Para o uso de um sistema de visão artificial é necessário uma atenção especial com o ambiente ao qual será extraída a imagem, para que o custo computacional de processamento seja o menor possível no que diz respeito à quantidade de dados a serem computados, e de forma a ter uma maior velocidade de processamento. O uso de câmeras de alta definição, iluminação adequada, boa focalização e posicionamento do objeto são fatores relevantes para boa estruturação da cena. Como comparativo, em aplicações em ambientes móveis, onde não é possível planejar a cena que será capturada, uma vez que, este é dinâmico, não se pode prever o que será detectado na cena, com isso, será exigido um algoritmo mais complexo que utilize ferramentas computacionais mais poderosas.

O uso do sistema de visão robótica está baseada nas informações de ALVES (1988), que afirma que existem três gerações de robôs. A primeira considerada de baixo grau de desenvolvimento tecnológico, não possui sensores e sim paradas mecânicas. A segunda definida como de médio grau de tecnologia, utiliza de sensores internos e, desta maneira, são capazes de conhecer o seu estado em relação à sua base, porém ainda impossibilitados de reconhecer modificações imprevistas em seu ambiente de trabalho. A utilização de controle em malha fechada, permitiu o reconhecimento de modificações no ambiente de trabalho e com isso, o sistema reage de forma inteligente. Surgem a partir deste momento a terceira geração (atual). E estes manipuladores robóticos da terceira geração, possibilitam maior flexibilidade dos processos com consequente aumento de produtividade.

Uma subdivisão da terceira geração em relação ao nível de inteligência artificial se faz necessária. Os robôs de baixo nível são dotados de sensores que apenas identificam mudanças no ambiente de trabalho, porém não quantificam e nem qualificam estas mudanças. Já os robôs com alto nível de inteligência possuem sensores capazes de quantificar e qualificar as modificações no seu ambiente de trabalho, ou seja, os objetos manipulados agora podem ter posições e orientações diferentes, além disso, pode-se atuar com objetos de formas e dimensões diversas.

Sensores de imagem, associados a algoritmos de visão artificial, são responsáveis por tornar o robô mais flexível, e pode-se citar outras vantagens:

- Possibilidade de inspeção, aferição e alinhamento de peças;
- Manipulação de objetos sem posição pré-definida;
- Manipulação de objetos diferentes e/ou que possuam variação dimensional;
- Maior nível de inteligência e maior controle do processo produtivo.

2.2 Aquisição de imagens

TSAI (1987) informa que em visão artificial existem câmeras unidimensionais (câmeras de varredura de linhas) e bidimensionais (câmeras de área), sendo que a grande maioria das aplicações utiliza este segundo modelo. Dentro das classes de câmeras, existem inúmeras escolhas a serem feitas como por exemplo, se esta vai obter imagens monocromáticas ou coloridas, qual o ângulo e o campo de visão, resolução, entre outras características.

A resolução refere-se ao número de linhas e colunas na imagem, sendo determinada pela câmera, em que as de varredura de área sua resolução é expressa como linhas e colunas de *pixels*, como por exemplo 640x480, já no caso de varredura por linha sua resolução é expressa apenas por colunas de *pixels*, as linhas possuem sempre o valor 1.

Para tornar possível a análise computacional sobre imagens, é necessário que os dados estejam na forma digital, isto é, a luminosidade deve ser convertida em sinais elétricos distribuídos espacialmente. Para que o processo de aquisição de imagem seja bem compreendido é necessário tratar da imagem em si e suas formas de representação.

O fotodetector é capaz de transformar luz (fótons) que são sinais ópticos em sinais elétricos e pode ser construído utilizando materiais semicondutores. O dispositivo necessário para a aquisição da imagem é um elemento físico sensível à energia luminosa, ou seja, a câmera, é um transdutor que é responsável por transformar a energia luminosa em sinal elétrico. Este sinal de saída é proporcional à intensidade luminosa incidente no dispositivo fotossensível. A sensibilidade da câmera é uma unidade fotométrica usada para medir o nível de iluminação, e é definida como sendo a iluminação sobre uma superfície quando o fluxo luminoso de 1 lúmen

incide sobre uma área de 1 metro quadrado, é também conhecido por candela, e está diretamente relacionada com a quantidade de luz mínima necessária para garantir a qualidade de imagem. Porém, é aconselhável que o local a ser instalado a câmera tenha luminosidade aceitável para os olhos, assim o transdutor não irá apresentar nenhum problema.

O principal objetivo da iluminação é a otimização do contraste, que pode ser considerado como a qualidade do sinal. O contraste é uma medida da diferença entre duas regiões: o objeto e o fundo da imagem. O primeiro passo na determinação da iluminação é escolher como diferenciar estas duas regiões, afirma GONZALES (1993).

Uma câmera colorida tem menor sensibilidade do que as monocromáticas, porque precisam processar mais cores para gerar a imagem, portanto precisam de mais luz para gerar boas imagens. As câmeras com sensores de imagem funcionam de maneira similar a câmeras com filme. Em ambas, a luz é focalizada por uma lente (ou conjunto delas) e incide sobre o sensor de imagem sensível à luz CCD ou CMOS, que convertem a luz em elétrons através do efeito fotoelétrico, já a câmera com filme, existe uma película fotossensível, LITWILLER (2001).

Na fotografia digital a imagem formada pela objetiva dentro da câmera não atinge uma emulsão fotossensível (o filme em câmeras antigas) e sim o dispositivo eletrônico sensível à luz, que é responsável por interpretar os impulsos luminosos da imagem quanto à sua intensidade (luminância) e coloração (crominância). Estas informações são codificadas de forma digital e armazenadas numa memória temporária (*buffer memory*) e posteriormente enviadas para um dispositivo de armazenagem (*memory stick, hard disk, ou outro*).

O sensor fotossensível é composto por milhares de elementos sensíveis a luz, e a imagem formada sobre ele é dividida em vários elementos de imagem, chamados *pixels*. Cada *pixel* contém informações correspondentes a aquela área da imagem. O princípio de funcionamento pode ser explicado por analogia a um tabuleiro de damas, onde cada quadrado representa um *pixel*, e em cada ponto existe uma informação. Quando a imagem está colocada sobre a superfície do tabuleiro, cada quadrado é lido de forma sequencial por um circuito eletrônico que faz a conversão dos dados analógicos para digitais, A/D e transforma o valor de cada *pixel* em um valor digital por meio da quantidade de carga de cada *pixel* e converte essa medição para a forma binária, segundo GONZALES (2004).

A FIG. 2.7 mostra o esquema de um sistema de aquisição de imagem. Pode-se perceber que o sensor é uma matriz de elementos fotossensíveis. Assim a imagem de saída do sensor é uma discretização da cena do mundo real. Quanto maior for o número de elementos do sensor, melhor será a resolução da imagem obtida, já que cada elemento irá resultar em um *pixel*. Havendo uma maior quantidade de elementos significa que uma menor parte da cena do mundo será representada por um único valor de nível de cinza, fazendo com que não se perceba a transição entre um *pixel* e outro.

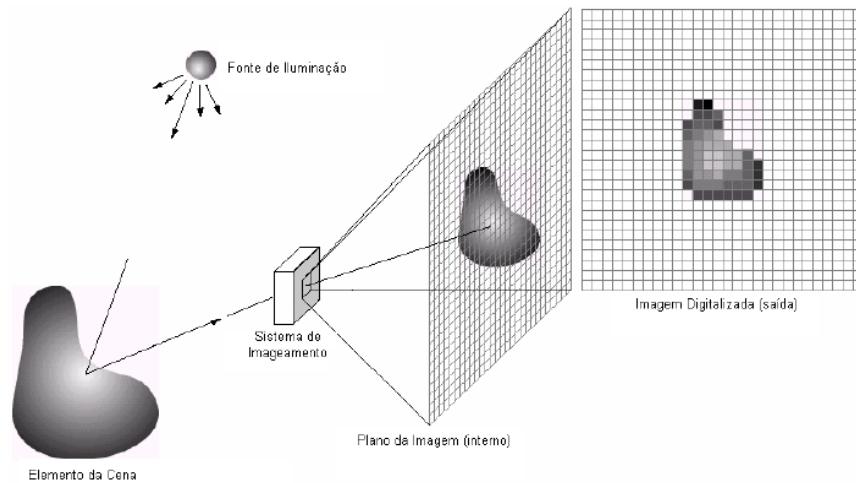


FIGURA 2.7 - Aquisição de imagem.

FONTE: GONZALES, 2004.

Os sensores de captação de imagem são susceptíveis a ruídos, sendo os mais comuns:

- os que ocorrem quando a capacidade de carga de um *pixel* é excedida;
- os associados à chegada aleatória de fótons;
- os que ocorrem na conversão de carga elétrica para tensão e na conversão A/D;
- os que ocorrem mesmo na ausência de luz. E os elétrons gerados termicamente são coletados no CCD.

Algumas são características das câmeras, e estas serão apresentadas a seguir, começa-se pelo ganho que é uma função normalmente implementada pelo *hardware*. A carga extraída de cada um dos condensadores do sensor é amplificada antes de ser quantizada. Isto tem o efeito de tornar as imagens mais claras, mas também mais ruidosas, visto que o ruído também é amplificado. Normalmente, o ganho pode ser colocado em modo automático, ou seja, a própria câmera regula a quantidade de ganho a ser aplicado através de um algoritmo implementado por

software que avalia a claridade de cada imagem precedente antes de regular o ganho da seguinte. Outra característica é o tempo de exposição, que é o tempo em que o diafragma, responsável por deixar entrar a luz para o sensor, se encontra aberto. Este tempo é limitado pela velocidade de transmissão das imagens adquiridas. Por exemplo, se a velocidade for de 15 imagens por segundo, o diafragma apenas pode estar aberto 1/15 segundo. Quanto mais tempo estiver aberto, mais clara fica a imagem obtida, mas ao contrário do ganho, este processo baixa a SNR da imagem. O tempo de abertura pode também por vezes ser posto em modo automático, no qual o *software* da câmera avalia, tal como para o ganho, o tempo que deve ser usado para obter a *frame* seguinte, baseado na quantidade de claridade que ficou submetida na captura anterior, de modo a não torná-las ruidosas, conforme JÜNGEL *et al.* (2003).

A claridade e o contraste são implementados apenas pelo *software* da câmera. Para o primeiro, cada *pixel* é adicionado um valor fixo de claridade o que tem o efeito de tornar a imagem mais clara (valor positivo) ou mais escura (valor negativo). Em casos extremos, este processo tem o efeito de torná-la ou completamente preta ou completamente branca. Este parâmetro da câmera não pode normalmente ser colocado em modo automático. Já para o contraste, cada *pixel*, subtrai metade do máximo valor possível, multiplica o resultado pelo valor de contraste escolhido e volta a somar metade do valor máximo possível. Esta operação tem o efeito de aproximar as cores claras do branco e as escuras do preto. Num caso extremo, a imagem ficaria branca e preta. Esta função serve para se poder distinguir melhor as cores. A saturação de uma cor refere-se ao quanto mais vibrante esta é. Uma cor pouco saturada aproxima-se do cinzento, enquanto se for muito saturada torna-se brilhante. Os algoritmos, implementados por *software*, variam de câmera para câmera. A coloração algumas vezes é erroneamente confundida com a saturação. Alterar este parâmetro na câmera tem o efeito de alterar completamente as cores de uma imagem e, como tal, não é usado por defeito pela câmera e também não pode ser posto em modo automático.

O balanço de brancos é uma das funções mais importantes da câmera. Costuma ser implementada por *software*. Diferentes fontes de iluminação dão tonalidades a uma imagem capturada que não seja tratada. Por exemplo, a luz do sol dá uma tonalidade azul às imagens, enquanto que a luz de uma lâmpada comum (tungstênio) dá-lhes uma tonalidade vermelha. Para compensar este problema, a câmera pode ajustar as tonalidades vermelha e azul (e por vezes até a verde, mas não é comum) da imagem de forma a que esta tenha um aspecto 'natural'.

2.3 Pixels e resolução

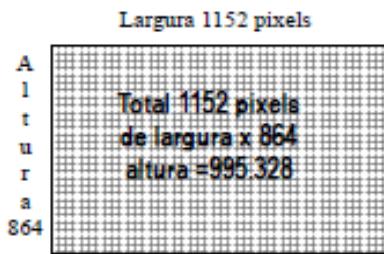
Um *pixel* é o elemento básico em uma imagem, é a menor unidade de uma imagem digital, e é formado basicamente pelo fotodiodo, ou seja, elemento fotossensível. A forma mais comum para o *pixel* é a retangular ou quadrada, por isso, frequentemente a organização de uma imagem está sob a forma de uma matriz de elementos em uma simetria quadrada, de modo também a facilitar a implementação eletrônica, seja na aquisição, seja na visualização de imagens. Este tipo de organização provoca o aparecimento de dois problemas importantes nas técnicas de processamento. Em primeiro lugar um *pixel* não apresenta as mesmas propriedades em todas as direções, isto é, ele é anisotrópico, esta propriedade faz com que ele tenha quatro vizinhos de borda e quatro vizinhos de diagonal, e obriga que seja definido o tipo de conectividade que será utilizada conforme apresenta a FIG. 2.8. Já o segundo problema é consequência direta do primeiro, ou seja, a distância entre um ponto e seus vizinhos não é a mesma para qualquer tipo de vizinho, será igual a 1 para vizinhos de borda e $\sqrt{2}$ para os de diagonal.



FIGURA 2.8 - Conectividade dos pixels vizinhos ao pixel central.

FONTE: O AUTOR, 2015.

Hoje se ouve muito falar em *megapixels*, isto porque como se pode imaginar, são necessárias quantidades enormes de *pixels* para compor uma imagem digital de alta resolução. Quanto mais elementos uma imagem possuir maior serão a sua qualidade e resolução. Como exemplo, uma câmera com a capacidade de gerar imagens de um *megapixel* deve possuir um milhão de sensores, ou seja, poderia ter as dimensões teóricas de 1.000 por 1.000 mil *pixels*. Sabe-se porém que as imagens fotográficas não são tradicionalmente quadradas então às dimensões do retângulo devem ser especificadas de forma que a resolução especifique a equação largura versus altura, ou seja, uma imagem com 1152 *pixels* de largura por 864 *pixels* de altura tem um total de 995.328 *pixels* ou um *megapixel* de resolução, conforme apresentado na FIG. 2.9.

FIGURA 2.9 - Resolução de um *Megapixel*.

FONTE: LENZ e TSAI, 1989.

Uma imagem digital é constituída por elementos discretos e a digitalização dos valores das coordenadas da imagem, pode ser dividida em dois: A amostragem é a resolução espacial, em que existe um número limitado de pontos em duas dimensões, e envolve a quantidade de linhas e colunas utilizadas, ou seja, depende do tamanho da matriz que está representando a imagem, conforme FIG. 2.10.

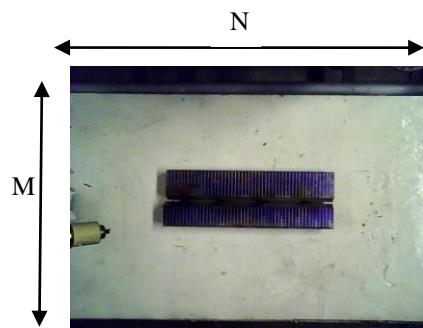
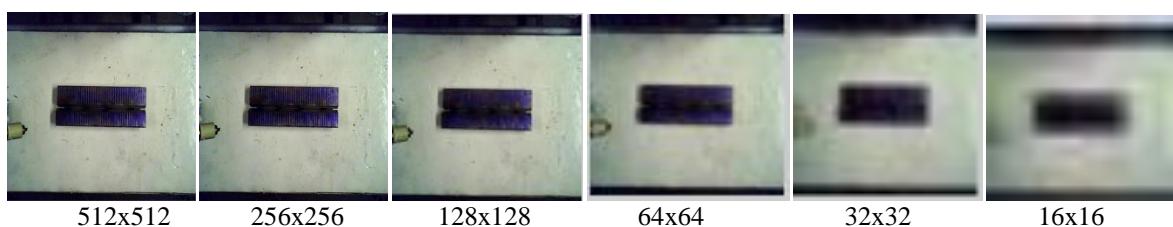


FIGURA 2.10 - Resolução 640 x 480 pixels.

FONTE: O AUTOR, 2015.

A segunda divisão da digitalização é a amostragem de amplitude, ou quantização, em que cada ponto é representado por um número que traduz o nível de cinza. Estes níveis podem ser representados por conjuntos cujos componentes estejam em Z^3 , dois elementos, são as coordenadas do *pixel* e o terceiro corresponde ao valor da intensidade. Quanto menor a resolução, maior é o tamanho *pixel* e menor é a qualidade da imagem, conforme FIG. 2.11:

FIGURA 2.11 – Avaliação na alteração da resolução de *pixels* em uma imagem.

FONTE: O AUTOR, 2015.

Para um determinado *pixel*, pode-se definir a adjacência (vizinhança qualquer) e conectividade (quando dois *pixel* satisfazem uma relação de aproximação e seus valores de cinza satisfazem a um dado critério de similaridade).

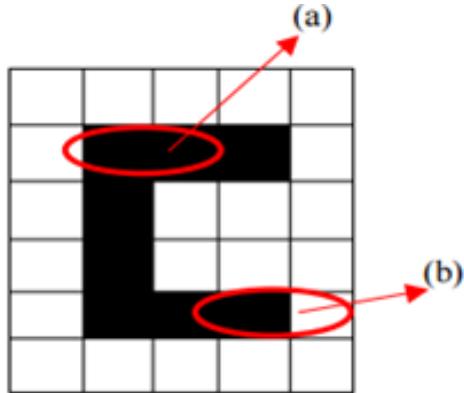


FIGURA 2.12 - Exemplo de conectividade.

FONTE: LETA et al., 2008.

No caso de conectividade apresentado acima, existem duas situações de *pixels* adjacentes. A diferença está na similaridade, em que na FIG. 2.12 no caso (a) os dois *pixels* são similares (mesmo nível de cinza), enquanto no caso (b) não há. Dessa forma, os *pixels* considerados em (a) estão conectados, e em (b) não estão conectados. Com isso a importância da conexão entre *pixels* define o que vem a ser bordas.

A imagem é uma grandeza bidimensional que representa o mundo tridimensional e, portanto, a informação de uma dimensão, a profundidade, é perdida. Porém já existem técnicas capazes de recuperar estes dados. Algumas destas técnicas utilizam estruturação de cenas, nas quais padrões são dispostos de forma a gerar uma referência para a imagem ou, num outro exemplo, projeções são feitas sobre os objetos possibilitando a identificação indireta da profundidade por meio de cálculos de relações geométricas. Há ainda o método que utiliza duas câmeras, denominado de visão estéreo para a captura da imagem, e segundo FU, GONZALES e LEE (1987), que informam que é possível determinar a profundidade da imagem se forem conhecidas as diferenças entre as coordenadas correspondentes da imagem (x_1 e x_2) formada em cada dispositivo, sua linha de base (distância entre os centros das duas lentes) e a distância focal, mas este processo não será abordado neste trabalho.

2.3.1 Digitalização de imagens

Uma imagem é definida como uma função discreta de intensidade luminosa $f(x, y)$, em que o valor de f em um ponto (x, y) qualquer, representa a intensidade desse ponto. Uma imagem bidimensional monocromática no plano x, y pode ser representada por uma função contínua $i(x, y)$ que especifica a intensidade de luz refletida. A imagem contínua é discretizada ao ser capturada por uma câmera que possui uma matriz de $m \times n$ elementos sensores de largura Δx e altura Δy . Isto resulta em uma imagem amostrada com resolução espacial de $m \times n$ pixels. A EQUAÇÃO (2.3) apresenta a intensidade luminosa média da superfície de um elemento sensor:

$$I_a(k, j) = \left[\int_0^{\Delta x} \int_0^{\Delta y} (i[((k - 1)\Delta x + x, (j - 1)\Delta y + y)] dx dy \right] / (\Delta x \cdot \Delta y) \quad (2.3)$$

Aonde a função não negativa $I_a(k, j)$ assume valores sobre uma faixa contínua de intensidade luminosa. A intensidade de cada pixel é quantizada com uma precisão de b bits, resultando em 2^b possíveis valores de intensidade (níveis de cinza). Desta forma, a imagem digital $I(k, j)$ é uma imagem analógica $i(x, y)$ amostrada com resolução espacial $m \times n$ e quantizada em intensidade com uma precisão de 2^b níveis de cinza.

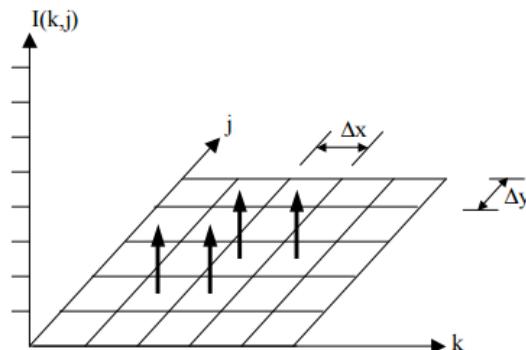


FIGURA 2.13 - Imagem 5x5 com 8 níveis de cinza.

FONTE: VIANNA, 2009.

Sabe-se que a luz é uma forma de energia, e por isso, $f(x, y)$ pode ser infinita e positiva, conforme a EQUAÇÃO (2.4):

$$0 < f(x, y) < \infty \quad (2.4)$$

A natureza de $f(x, y)$ pode ser caracterizada por dois componentes: a iluminação $i(x, y)$ e a refletância $r(x, y)$, onde se pode escrever o produto de $i(x, y)$ por $r(x, y)$ e resulta em $f(x, y)$, como representado na EQUAÇÃO (2.5).

$$f(x, y) = i(x, y) \cdot r(x, y) \quad (2.5)$$

Sendo que:

$$0 < i(x, y) < \infty \quad (2.6)$$

$$0 < r(x, y) < 1 \quad (2.7)$$

Isso significa que a refletância é limitada entre a absorção total, $r(x, y) = 0$; e a refletância total, $r(x, y) = 1$. Com isso, uma imagem digital pode ser representada por uma função discreta de intensidade de luz, $f(x, y)$, onde x e y representam as coordenadas espaciais e f o brilho proporcional em cada ponto (x, y) . Em imagens monocromáticas, f está relacionado com os níveis de cinza em cada ponto. Na câmera, a imagem é varrida da esquerda para a direita, de cima para baixo, em uma série de linhas horizontais que são transmitidas para o receptor, conforme a FIG. 2.14. Na câmera de vídeo, o sistema de lentes focaliza a imagem sobre uma camada fotocondutora. Nessa camada, cargas elétricas proporcionais à luminosidade da imagem em cada ponto são desenvolvidas e armazenadas capacitivamente, e um feixe de elétrons é usado para converter essa carga em corrente elétrica. Esse feixe de elétrons é focado em um pequeno ponto circular e é defletido continuamente horizontal e verticalmente até varrer toda a imagem.

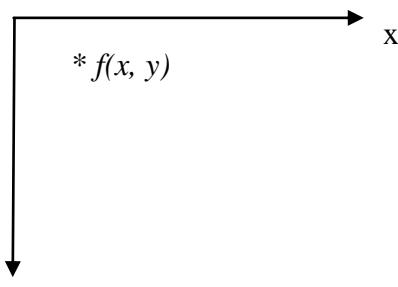


FIGURA 2.14 - Convenção utilizada em processamento de imagens.

FONTE: O AUTOR, 2015

Já para uma imagem colorida, que não será abordado neste trabalho, ocorre à inclusão além da informação de luminosidade de cada *pixel*, a informação da cor do mesmo. Para representar estas informações são necessários três valores. Desta forma, modelos de representação de cor se baseiam em subespaços de cores tridimensionais. Em se tratando de imagens coloridas, os modelos mais utilizados no processamento de imagens são o *RGB*, *YIQ* e *HSI*, GONZALES e WOODS (2000).

O processo de segmentação, que se baseia nas descontinuidades que são mudanças abruptas na luminância ou crominância da imagem pode ser dividido em diversas técnicas para tratar seus níveis de cinza, tais como: detecção de pontos isolados, detecção de linhas e detecção de bordas; e similaridade, também conhecido como semelhança nos níveis de cinza: limiares (*Thresholding*), crescimento de regiões (*Region Growing*), aglomeração (*Clustering*).

Para imagens em escala de cinza onde os objetos em cena contrastam bem com o fundo da imagem, a limiarização é o método mais simples de segmentação, que de modo geral, isola regiões de uma imagem que possuam propriedades comuns entre si. Uma imagem pode ser segmentada analisando-se propriedades como luminosidade, cor, textura e movimento. A escolha apropriada de um limiar de nível de cinza T no meio do vale que separa os dois picos e permite classificar e rotular os *pixels* em duas categorias: de fundo (0) ou de objetos (1), resultando em uma imagem binária $I_B(k, j)$ segmentada, e pode ser representada pela EQUAÇÃO (2.8):

$$I_B(k, j) = \begin{cases} 1 & \text{se } f(x, y) > T \\ 0 & \text{se } f(x, y) \leq T \end{cases} \quad (2.8)$$

Os valores serão zero para intensidades menores ou iguais a T e um para os outros *pixels*. A limiarização pode ser solucionada a partir de duas técnicas: global ou local. As técnicas globais consistem em determinar um único limiar para toda a imagem. Porém, esta técnica traz problemas causados pela variação de iluminação. O que pode melhorar o uso do *threshold* é quando o mesmo é aplicado de forma local, ou seja, quando a função $T(x, y)$ varia o valor de T de acordo com as coordenadas (x, y) apresentadas na EQUAÇÃO (2.9):

$$g(x, y) = \begin{cases} 0 & \text{se } f(x, y) < T(x, y) \\ 1 & \text{se } f(x, y) \geq T(x, y) \end{cases} \quad (2.9)$$

2.4 Processamento de imagens

Processar uma imagem consiste em transformá-la sucessivamente com o objetivo de extrair mais facilmente a informação nela presente. É importante fazer uma comparação entre o processamento de imagem e a computação gráfica. Esta última parte de uma informação para obter uma imagem, já o processamento de imagens trata de obter informações de uma imagem obtida da cena. Deste ponto de vista estes dois segmentos são exatamente métodos opostos, mas isto não quer dizer que as técnicas envolvidas em cada caso não possam ser as mesmas ou pelo menos complementares, afirma GONZALES (1993). É evidente que neste sentido, processar uma imagem como é feito pelo SVH é extremamente complexo conforme ROSENFELD (1993). Realizar estas mesmas tarefas com a ajuda de máquinas, exige por antecedência uma compreensão “filosófica” do mundo ou dos conhecimentos humanos. Esta característica faz com que o processamento de imagens seja, atualmente, uma disciplina com extrema dependência do sistema no qual ele está associado, não existindo no entanto uma solução única e abrangente para todos os problemas. Daí a não existência, até o momento, de sistemas de análise de imagens complexos e que funcionem para todos os casos. O interesse por métodos de processamento de imagens digitais decorre de duas grandes áreas de aplicação: melhoramento de imagens para percepção e interpretação humana e processamento de dados para identificação automática por máquinas, já afirmavam GONZALEZ e WOODS (2000). Para o processamento de imagens são necessários alguns passos fundamentais, que estão apresentados na FIG. 2.15.

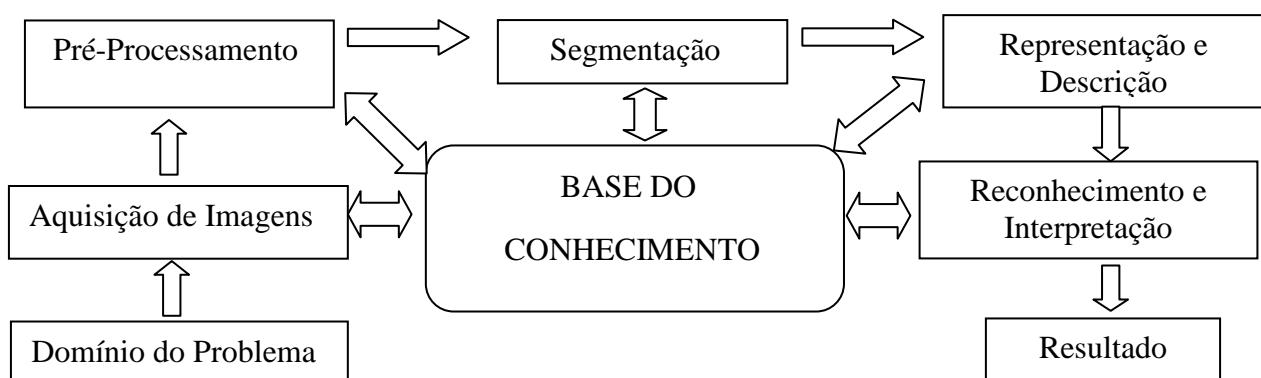


FIGURA 2.15 - Etapas fundamentais em processamento de imagens.

FONTE: O AUTOR, 2015.

De acordo com a aplicação, resultados diferentes podem ser obtidos e nem sempre se fazem necessárias à realização de todas as etapas apresentadas. De acordo com LETA *et al.* (2008), o

primeiro passo é a aquisição da imagem, no qual o elemento transdutor capta a imagem transformando a informação de intensidade luminosa em sinal elétrico. Dependendo do tipo de transdutor utilizado o sinal de saída pode ser digital ou analógico. No segundo passo é feito o pré-processamento, de modo a realizar uma melhoria na qualidade da imagem, com correções de distorções geométricas e remoção de ruídos, e isolamento de regiões, a fim de possibilitar um melhor desempenho das etapas posteriores, com algoritmos menos robustos e com garantia de resultados mais apurados. Exemplos de pré-processamento são a implementação de filtros para eliminação de ruídos, que geram distorção ou perda de informação, e algoritmos de realce de contraste. No próximo passo, realiza-se a segmentação, onde ocorre a separação da imagem em partes ou objetos constituintes agrupando áreas com características similares, considerando alguns de seus atributos, como por exemplo, o nível de cinza dos *pixels*, cor ou até mesmo a textura, visando caracterizar a representatividade dos objetos na cena, como indica BINS *et al.* (1996). A segmentação é considerada, dentre todas as etapas de processamento de imagens, a etapa mais crítica do tratamento da informação, é um processo empírico e adaptativo, procurando sempre se adequar às características particulares de cada tipo de imagem e aos objetivo que se pretende alcançar, e este passo determina o eventual sucesso ou fracasso da análise. O algoritmo de segmentação proposto por BINS *et al.* (1996) é baseado na tradicional técnica de crescimento de regiões, que consiste em um processo de iteratividade no qual regiões são agrupadas a partir de *pixels* individuais. Tal implementação propõe um agrupamento de regiões baseado no conceito de região vizinha mais similar. Neste método são definidos dois limiares, o primeiro é o de similaridade, abaixo do qual duas regiões são consideradas similares e o segundo é o da área, valor de área mínima para que uma região seja individualizada.

O diferencial entre as técnicas de classificação convencionais de processamento de imagem e a técnica de segmentação é que essa última tem como premissa o agrupamento de *pixels* com características similares em termos de tons e texturas, formando regiões homogêneas, sendo estas então submetidas ao processo de classificação VENTURIERI e SANTOS (1998).

A descrição consiste em extrair características dos objetos que resultem em informação quantitativa ou informação básica para distinção entre classes de objetos. E para reconhecer objetos rotulados em imagens segmentadas torna-se necessário descrevê-lo por meio de atributos que possam ser obtidos a partir da sua imagem. Estas características são chamadas de descritores de linha e de área.

O último passo é o reconhecimento e interpretação. No reconhecimento são atribuídos rótulos aos objetos de acordo com a informação de seus descritores. Um exemplo de parâmetros pertinentes a um objeto é o seu perímetro e área. Em seguida atribui-se a cada região contigua de *pixels*, na área pré-classificada de objeto, um indicador que os identifica, e assim aplica-se um algoritmo que execute medidas específicas como a área (A), perímetro (P), posição na imagem (x_c , y_c), quantidade de furos, etc. Os rótulos são atribuídos individualmente a cada objeto em função de cada descritor distinto. A interpretação irá inferir significado a um conjunto de objetos reconhecidos, ou seja, trata-se também de uma rotulação. O bloco base de conhecimento se comunica com todas as etapas do processamento, pois detêm todo o conhecimento prévio sobre o domínio do problema e a natureza esperada do resultado. Neste bloco é possível reduzir o custo computacional, pois consiste em definir qual informação tem que ser subtraída da imagem e de que forma esta informação estará disponível como resultado.

2.4.1 Qualidade de imagens

O processamento de imagens é caracterizado por soluções específicas, de modo que técnicas que funcionam bem em uma área podem se mostrar totalmente inadequadas em uma outra área, afirmaram GONZALES e WOODS (2000). A avaliação da qualidade da imagem é a medição de um atributo (propriedades ou características), e é utilizada em diversas aplicações na área de processamento de imagens, e pode ser avaliada por duas metodologias, a subjetiva e a objetiva.

No método subjetivo, a avaliação da qualidade é realizada, propriamente, por um ou mais observadores. O ser humano é especialista na avaliação da qualidade de imagens, conseguindo, por exemplo, perceber pequenas variações de contraste e foco, e assim, avaliar se uma imagem é boa ou ruim, e apresenta resultados confiáveis, porém sua aplicação prática é limitada. Por exemplo, uma das principais aplicações envolvendo o uso de métricas de qualidade de imagem está diretamente relacionada a sistemas automáticos de restauração de imagens. Em tais situações, faz-se necessário obter algum dado objetivo que indique a qualidade da imagem, já diziam GONZALEZ e WOODS (2000).

Métodos objetivos medem a qualidade da imagem automaticamente, produzindo um dado numérico que quantifica a qualidade de uma ou mais características específicas da imagem. E pode ser considerada satisfatória quando produz os mesmos resultados que um método subjetivo, dadas às mesmas circunstâncias relacionadas à imagem.

Uma característica básica de uma imagem que deve ser considerada é o borramento, que afeta diretamente sua qualidade, e está associado ao método objetivo. Há várias causas que proporcionam esta disfunção, tais como, a compressão de uma imagem, a falta de foco ou mesmo em um processo para redução do ruído. Por sua vez, o ruído pode ser causado durante a obtenção da imagem, dependendo do modo como a luz percorre na sua formação, e tem quatro fatores principais: a largura de banda limitada do sistema de imagem, a existência de movimento durante a aquisição, imagem fora de foco, e turbulência se ocorrer em meio participativo como a água. CHOI, JUNG e CHEON (2009) apresentam um método de avaliação cega, baseado na quantidade de borramento e ruído contidos em uma imagem. Neste método, se assume que estes são os principais fatores do decaimento da qualidade de uma imagem, e há duas etapas de análise. Primeiro, ocorre a detecção das bordas da imagem, depois, a análise do borramento nessas bordas é feita a partir da diferença de intensidade entre o *pixel* atual e a média da intensidade de seus *pixels* vizinhos, e será considerada uma imagem de baixa qualidade, se essa diferença for muito baixa.

Já o ruído, também precisa da detecção das bordas, porém primeiro é aplicado um filtro para retirá-lo por completo imagem, para que não afete a detecção das bordas, e este último não resulta em elementos totalmente conectados, e que para isso aconteça deve-se aplicar a ligação das bordas ou de fronteiras, após a detecção.

2.4.2 Detecção de pontos, linhas e bordas em imagens

A detecção de linhas pode ser implementada como um procedimento do tipo *template matching*, que nada mais é do que a maneira mais comum de procurar por descontinuidades em imagens através da sua varredura por uma máscara que respondam a pontos, linhas e bordas em diferentes orientações, seguida de um limiar para selecionar os *pixels* que pertençam a linha. É um processo complexo, pois é necessário achar *pixels* que são semelhantes e testá-los para verificar se são parte de uma mesma linha ou região.

Segundo GONZALEZ e WOODS (2000), pode-se utilizar o gradiente de uma imagem $f(x, y)$ para detecção de bordas, sendo que o vetor gradiente aponta na direção de mudança mais rápida de f na posição (x, y) . Na detecção das bordas a magnitude do vetor é uma quantidade importante, pois equivale a maior taxa de aumento de intensidade $f(x, y)$ por unidade de distância na direção deste vetor.

Existem vários métodos bem conhecidos que utilizam os gradientes para detecção de bordas. Um exemplo é o método de SOBEL (1990), também conhecido como máscara de SOBEL, que é a aplicação de um filtro sobre uma região da imagem com o objetivo de obter o gradiente da intensidade em cada ponto, dando a direção e a quantidade da maior variação de claro para escuro. Outra técnica utilizada para segmentação e detecção de linhas em uma imagem é a “Transformada de Hough”, que é realizada em nível de processamento global, sendo avaliados todos os *pixels* com base em um novo espaço de parâmetros, afirma DUDA *et al.* (1972).

A detecção de pontos isolados pode ser obtida através do uso da máscara apresentada na FIG. 2.16, onde ocorre a medição das diferenças ponderadas entre o ponto central e seus vizinhos, sendo que o nível de cinza de um ponto isolado deverá ser diferente do nível de cinza de seus vizinhos.

$$\begin{matrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{matrix}$$

FIGURA 2.16 - Detecção de pontos isolados a partir de um fundo constante.
FONTE: O AUTOR, 2015.

Já na FIG. 2.17 estão apresentadas as máscaras utilizadas para detectar linhas verticais, horizontais ou inclinadas a 45°, adotando-se o mesmo princípio utilizado para os pontos isolados por diferenças ponderadas.

$$\begin{array}{ccc} -1 & -1 & -1 & -1 & -1 & 2 & 2 & -1 & -1 & -1 & 2 & -1 \\ 2 & 2 & 2 & -1 & 2 & -1 & -1 & 2 & -1 & -1 & 2 & -1 \\ -1 & -1 & -1 & 2 & -1 & -1 & -1 & -1 & 2 & -1 & 2 & -1 \end{array}$$

Horizontal +45° -45° Vertical

FIGURA 2.17 - Detecção de linhas na horizontal, inclinada a +/-45 graus e vertical.
FONTE: O AUTOR, 2015.

No caso de detecção de bordas, que é o mais comum para identificar descontinuidades significantes em imagens, que representa o limite entre duas regiões com propriedades distintas de níveis de cinza. Para a detecção e realce de bordas, aplicam-se habitualmente filtros espaciais

lineares de dois tipos: (a) baseados no gradiente da função de luminosidade da imagem, e (b) baseados no Laplaciano de $I(x, y)$. As bordas são consideradas as características mais relevantes em um grande número de algoritmos de visão computacional apresentados por AVCIBAS *et al.* (2002) e LI (2002), pois podem ser utilizadas para descrever a estrutura de uma cena. O conjunto de *pixels* raramente caracteriza completamente uma fronteira devido ao ruído, as quebras na fronteira por causa de iluminação não uniforme e a outros efeitos que introduzem descontinuidades de intensidades. Para tanto os algoritmos de detecção de bordas são tipicamente seguidos de procedimentos de ligação e de outros procedimentos de detecção de fronteiras, todos os pontos que forem similares serão ligados, formando uma fronteira de *pixels* que compartilham alguma propriedade comum.

A Transformada Hough é uma técnica de processamento, desenvolvida por HOUGH (1962), para ligação de pontos e detectar curvas que determinem uma forma dentre relações globais entre os *pixels*. A linha é a forma geométrica mais simples de ser detectada e será usada para explicar os conceitos da TH, uma linha contida em uma imagem pode ser equacionada de várias maneiras. A mais adequada é a equação paramétrica da reta apresentada na EQUAÇÃO (2.10):

$$\rho = x \cdot \cos\theta + y \cdot \sin\theta \quad (2.10)$$

Que nos permite obter equações para retas em todas as direções, o que não é possível se usarmos a equação geral da reta, no espaço de parâmetros (a, b) :

$$y = a \cdot x + b \quad (2.11)$$

Na reta genérica apresentada na FIG. 2.18 está associada a dois parâmetros (a, b) , ao criar-se um plano com eixos referentes aos parâmetros da reta, cria-se um espaço de parâmetros.

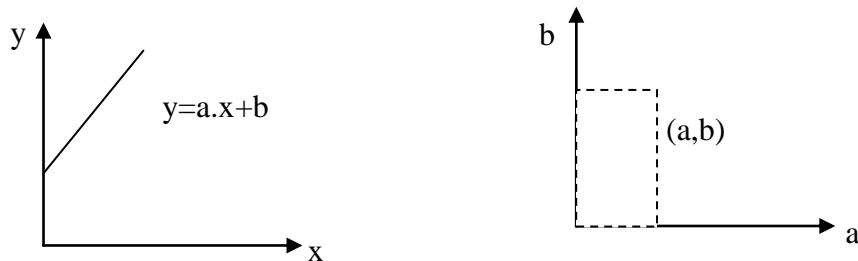


FIGURA 2.18 - Reta genérica representada no espaço (x, y) , e no espaço (a, b) .

FONTE: O AUTOR, 2015.

Os parâmetros são ilimitados, logo à medida que a reta cresce em direção paralela ao eixo y o coeficiente angular a tende a infinito, impossibilitando uma descrição matemática dessas retas, sendo melhor parametrizá-las usando (θ, ρ) . Sendo possível representar qualquer reta no espaço cartesiano e os parâmetros são limitados aos seguintes intervalos:

$$-90^\circ \leq \theta < 90^\circ \text{ e } 0 \leq |\rho| \leq \sqrt{(w^2 + h^2)} \quad (2.12)$$

Sendo w e h a largura e a altura da imagem em *pixels*. Para gerar o espaço paramétrico, faz-se uma busca exaustiva por cada *pixel* branco da imagem, e são calculados dois vetores de parâmetros ρ e θ , para cada θ é calculado um ρ . A depender da resolução adotada para θ , estes cálculos demandam um grande custo computacional. Para diminuir o número de *pixels* processados, é promovida a extração do contorno da imagem binarizada, que irá mapear as possíveis retas da imagem em um espaço paramétrico (ρ, θ) , que nada mais é do que uma matriz 2D de acumuladores que indicam quantos votos cada par de parâmetros (ρ, θ) obteve no processo. Este método apesar de bastante eficiente para a detecção de retas em imagens, é sensível a ruídos e problemas de iluminação, o que deixa o sucesso ou fracasso do resultado final muito dependente da segmentação aplicada à imagem original.

Os pacotes de computação numérica, e de processamento de imagens disponíveis no *MatLab*[®] foram utilizados para o desenvolvimento do algoritmo que permite a aquisição de imagens instantâneas (*snapshots*), ou uma sequencia de imagens. Dentro do *Toolbox* de processamento de imagens do *MatLab*[®], existe uma função chamada *regionprops*, que apresenta as características sobre as regiões de uma imagem, no QUADRO (2.2) estão estas propriedades.

QUADRO 2.2
Propriedades do *Regionprops*

<i>Area</i>	<i>EquivDiameter</i>	<i>MajorAxisLength</i>
<i>Bounding Box</i>	<i>EulerNumber</i>	<i>MinorAxisLength</i>
<i>Centroid</i>	<i>Extent</i>	<i>Orientation</i>
<i>ConvexArea</i>	<i>Extrema</i>	<i>PixelIdxList</i>
<i>ConvexHull</i>	<i>FilledArea</i>	<i>PixelList</i>
<i>ConvexImage</i>	<i>FilledImage</i>	<i>Solidity</i>
<i>Eccentricity</i>	<i>Image</i>	<i>SubarrayIdx</i>

FONTE: http://www.mathworks.com/help/images/ref/regionprops.html?s_tid=srchtitle. Disponível em 20/07/2015.

De todas as propriedades apresentadas na tabela anterior, faz-se um estudo apenas de uma, o *Bounding Box* que é basicamente uma caixa que delimita pixels de modo que o processamento exija pouca capacidade computacional, é uma ferramenta utilizada para medir regiões de uma imagem, sendo que para a mensuração do valor do *pixel* só é válido para uma imagem em tons de cinza.

O *Bounding Box* que pode ser traduzido como "caixa delimitadora" é representado como $[x_0 \ y_0 \ w \ h]$, onde (x_0, y_0) são as coordenadas do ponto superior esquerdo, e w, h são a largura e altura, com isso, esta função retorna o menor retângulo que contém a região. Esta propriedade permite restringir a atenção a uma determinada região que esteja delimitada, de modo que esta região corresponda a uma sub-imagem. Na FIG. 2.19 tem-se a representação do que foi explicado acima, o quadro tracejado remete ao *Bounding Box*, acrescenta-se como informação (x_c, y_c) , a centroide do objeto detectado, conforme HOFF (2014).

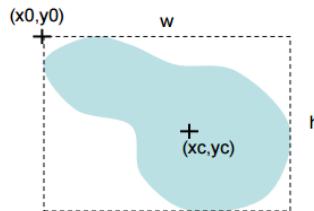


FIGURA 2.19 - Representação do centroide e *Bounding Box*.

FONTE: HOFF, 2014.

Gera-se uma lista de todas as áreas detectadas, posteriormente faz-se a concatenação das matrizes que representam estas áreas. Então estes grandes *blobs* que são uma coleção de dados binários armazenados como uma única entidade, tendo como resultado a centróide, o baricentro, e o *bounding box* (*BB*). Finalmente é desenhada uma caixa em torno do maior *blob*.

2.5 Filtros

Imagens geralmente quando são adquiridas estarão contaminadas por ruídos de origem variada, sombras e variação de iluminação podem fazer com que regiões de borda apareçam borradass, limiares utilizados, regiões pequenas podem aparecer no fundo da imagem ou buracos pequenos podem ser criados na superfície dos objetos em cena. Uma das técnicas de processamento mais utilizadas para melhorar a qualidade de uma imagem digital é a aplicação de algoritmos que permitam eliminar regiões indesejáveis, por causa de métodos de aquisição, ou das condições em

que a imagem foi capturada, que introduzem algum tipo de ruído na imagem, para tanto, operadores iterativos derivados da morfologia matemática podem ser utilizados para filtrar imagens usando técnicas locais, de modo a minimizar este tipo de problema, suavizando contornos e eliminando parte do ruído.

A filtragem é uma técnica de processamento que modifica e/ou melhora a informação (contraste) contida na imagem, realçando contornos ou removendo características (ruídos), e é uma operação de vizinhança que realiza a transformações *pixel a pixel*, que não depende apenas do nível de cinza de um determinado *pixel*, mas também do valor dos níveis de cinza dos seus vizinhos na imagem. Com isso, o *pixel* da imagem resultado é obtido processando os *pixels* na vizinhança deste correspondente na imagem origem. As classificações mais usuais dos filtros baseiam-se na sua linearidade e na resposta em frequência, é também normal uma terceira classificação que distingue filtros especiais dos demais filtros, onde é aplicada a representação da transformada de Fourier na imagem, e este não será abordado neste trabalho.

Segundo MARQUES (1999), os *templates*, máscaras, operadores de vizinhança, janelas, ou filtros, é como são chamados os operadores locais mais empregados em processamento de imagens, para aguçamento das mesmas. E basicamente trata-se de uma matriz bidimensional, em que cada elemento é multiplicado, convoluído, pelo *pixel* correspondente, ou seja, calcula-se a intensidade de um determinado *pixel* em função da intensidade de seus vizinhos. O cálculo é baseado em ponderação, isto é, utilizam-se pesos diferentes para *pixels* vizinhos diferentes, e o resultado desta operação é armazenado como o novo valor do nível de cinza na nova imagem. Os operadores locais combinam a intensidade de certo número de *pixels* (janela), para computar o valor da nova intensidade na imagem de saída.

Uma forma fundamental de caracterizar filtros é a forma como eles amplificam ou atenuam certas bandas de frequência. Em geral, existem muitos tipos diferentes de respostas em frequência, mas para imagens, a grande categoria de filtros passa-baixa, e passa-alta, são suficientes, tendo em conta que de uma forma ou de outra, dependendo dos parâmetros que são introduzidos, um filtro passa-banda pode operar como filtro passa-baixa ou passa-alta. Sabe-se que esses tipos de filtros são bastante utilizados para suavização de imagens, tentando atenuar ao máximo o efeito do ruído e preservar arestas em imagens digitais. Esses filtros conseguem amenizar de forma geral os efeitos do ruído, mas introduzem borramento nas imagens.

Um filtro do tipo passa-baixa elimina os valores relacionados às altas frequências, com isso, o efeito deste filtro é o de suavização da imagem, uma vez que as altas frequências que correspondem às transições abruptas são atenuadas. A suavização tende pelo mesmo motivo, diminuir o ruído em imagens e apresenta efeito de borramento.

Os filtros passa-alta respondem a mudanças abruptas da intensidade de luz numa imagem, por isso são utilizados no realce de detalhes como contornos. O objetivo principal das técnicas de realce é o de processar de modo que o resultado seja mais apropriado para uma aplicação específica do que a imagem original. Como os componentes de baixas frequências são responsáveis pelas características que variam lentamente em uma imagem, tais como o contraste total e a intensidade média. O efeito visual deste tipo de filtro é de tornar as transições entre diferentes regiões mais nítidas.

O filtro de SOBEL é um filtro não linear utilizado para realçar bordas e apresenta uma grande imunidade a ruído, sua primeira derivada detecta o declive máximo da curva de níveis de cinzento. O tamanho da janela (3×3) não pode ser alterado. Este processo consiste na convolução de uma janela para detectar as orlas horizontais e verticais respectivamente, utilizando as máscaras h_1 e h_2 com a imagem $f(x, y)$:

$$h_1 = \begin{matrix} 1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{matrix} \quad h_2 = \begin{matrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{matrix}$$

FIGURA 2.20 - Máscaras de convolução - SOBEL.

FONTE: O AUTOR, 2015.

O detector desenvolvido por CANNY (1986) utiliza um algoritmo multi-estágios para detectar uma ampla margem de bordas na imagem, este é um filtro de convolução f muito mais resistente ao ruído do que os outros filtros. Este método difere dos outros filtros detectores de bordas, pois usa dois limiares diferentes, também chamados de *thresholds*, sendo um para contornos fortes e outro para contornos fracos. Este método baseia-se em procurar por pontos em que o módulo do gradiente de intensidade é máximo na direção do vetor gradiente, considerando-os como elementos de borda.

2.6 Calibração de câmeras

Entre diversos problemas relacionados com aquisição de imagens, evidenciasse as imperfeições do conjunto ótico, utilizado para obtenção das mesmas, como distorção nas lentes e deformações no sensor ótico. As imperfeições fazem com que as imagens adquiridas pelas câmeras necessitem de correções, caso contrário estas podem interferir nos resultados.

De acordo com FU (1987), uma transformação de perspectiva projeta pontos da cena tridimensional em um plano. Estas transformações oferecem uma aproximação à maneira na qual uma imagem é formada ao se olhar para o mundo tridimensional. Considere uma câmera ao capturar uma imagem de um objeto situado em frente da mesma, a origem do sistema de coordenadas fixo na câmera é localizada no plano da imagem e o seu eixo é alinhado com o da lente da câmera. A distância entre a lente e o plano da imagem é denominada distância focal efetiva, e dado um ponto de um objeto visualizado pela câmera, este possui coordenadas do ponto em referência as coordenadas da câmera. Da mesma forma, a imagem possui coordenadas do ponto expressas em referencial da câmera.

A calibração de câmeras utilizadas em sistemas de visão é o processo de determinação dos elementos que governam a relação entre a imagem bidimensional que a câmera reconhece e a informação tridimensional da imagem real do objeto, conforme BASU (1995). Com isso, a calibração do atuador em relação à câmera se faz necessária, uma vez que, a visão auxilia na tarefa de um manipulador, por exemplo. Para tanto o robô deve conhecer a posição da peça no sistema coordenado de referência do mundo, e através desta imagem é possível obter a posição da peça em relação ao sistema coordenado da câmera e a calibração ocorre para transformar essas coordenadas para o sistema coordenado de interesse. A transformação de coordenadas do sistema utiliza a estrutura apresentada na FIG. 2.21.

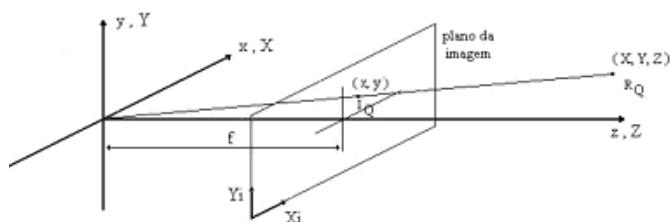


FIGURA 2.21 - Transformação de coordenadas.

FONTE: TSAI, 1987.

Apresentado na FIG. 2.21, o modelo do processo de formação de imagem, em que se assume como centro de projeção a origem de um referencial euclidiano espaço \mathbb{R}^3 , no qual convergem os feixes de luz emitidos por pontos no espaço \mathbb{R}^2 . O sistema de coordenadas (x, y, z) , da câmera é colocado de tal forma que o plano da imagem coincida com o plano (x, y) e o eixo óptico fique ao longo do eixo z . Então, o centro do plano da imagem estará na origem, e o centro da lente estará nas coordenadas $(0, 0, f)$. Onde f representa a distância focal entre o plano da imagem e o centro da lente. Assume-se aqui que o sistema de coordenadas da câmera está alinhado com o sistema de coordenadas do mundo (X, Y, Z) . Considerando $Z > 1$, é o mesmo que dizer que todos os pontos de interesse ficam na frente da lente da câmera, e a EQUAÇÃO (2.13), representa esta relação.

$$R_Q = R_{T_I} = I_Q \quad (2.13)$$

A metodologia da transformada segue o seguinte padrão, conforme a EQUAÇÃO (2.14):

$$\begin{bmatrix} R_{QX} \\ R_{QY} \\ R_{QZ} \\ 1 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix} \begin{bmatrix} I_{QX} \\ I_{QY} \\ ? \\ k \end{bmatrix} \quad (2.14)$$

Simplificando, temos a EQUAÇÃO (2.15):

$$\begin{bmatrix} R_{QX} \\ R_{QY} \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} I_{QX} \\ I_{QY} \end{bmatrix} \quad (2.15)$$

Resolvendo:

$$[R_Q I_{Q'}] = [T] [I_Q I_{Q'}] \quad (2.16)$$

$$[R_Q I_{Q'}] [I_Q I^{-1}_{Q'}] = [T] [I_Q I_{Q'}] [I_Q I_{Q'}]^{-1} \quad (2.17)$$

$$[T] = [R_Q I_{Q'}] [I_Q I_{Q'}]^{-1} \quad (2.18)$$

No método proposto por LENZ e TSAI (1989), e apresentado na FIG. 2.22, o robô executa vários movimentos com a câmera fixada à sua garra e, ao final de cada movimento, é capturada uma imagem do bloco de calibração que é composto por uma matriz de pontos em sua superfície.

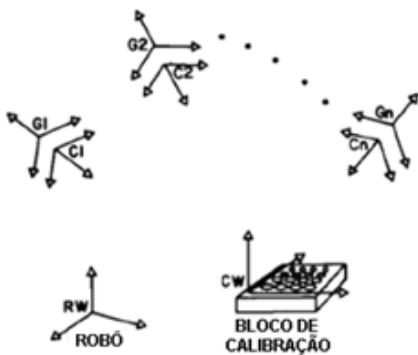


FIGURA 2.22 - Configuração básica para a calibração flange-câmera.

FONTE: LENZ e TSAI, 1989.

O padrão deve ter um formato definido por uma estrutura que permita uma nítida diferenciação em relação a outros objetos semelhantes que possam estar presentes no mesmo espaço de visão da câmera. O seu formato também deve permitir uma fácil manipulação e movimentação no espaço de visão da câmera. Isso permitirá que o campo visual esteja corretamente preenchido pela captura de várias vistas válidas do padrão, o que representará um maior número de pontos de referência corretamente identificados e espalhados no campo visual da câmera. A FIG. 2.22 apresentou uma configuração básica dos sistemas coordenados para a calibração flange-câmera, enquanto que a FIG. 2.23 apresenta as matrizes de transformação homogênea entre os diversos sistemas coordenados, considerando:

- CW - sistema coordenado do bloco de calibração. As coordenadas dos pontos de sua superfície são conhecidas;
- RW - sistema coordenado fixo da base do robô (através dos sensores das juntas sabe-se o quanto cada junta se movimentou, com isso é possível saber a posição da garra do robô);
- Ci - sistema coordenado da câmera com o eixo z coincidindo com o eixo óptico e os eixos x e y paralelos aos eixos X e Y da imagem, portanto as coordenadas encontradas através da imagem e referenciadas ao sistema coordenado da imagem podem ser consideradas as mesmas das coordenadas no sistema da câmera;
- Gi - sistema coordenado da garra do robô.

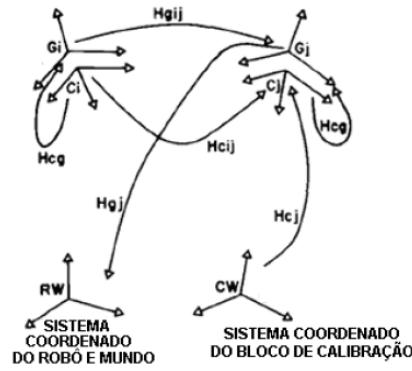


FIGURA 2.23 - Sistemas de transformação homogênea.

FONTE: LENZ e TSAI, 1989.

2.6.1 Calibração intrínseca

Considere que uma câmera possui um matriz de $m \times n$ elementos sensores, cada *pixel* com largura Δx e altura Δy , e que a localização do referencial fixo na câmera tem a sua origem no centro (c_x , c_y) da matriz de sensores e rotacionado 180° em torno do eixo z em relação ao sistema de coordenadas em *pixel* (p_x , p_y), conforme a FIG. 2.24.

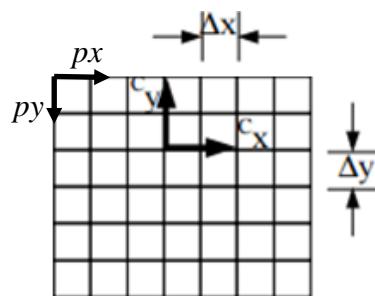


FIGURA 2.24 - Sistema de referência em uma matriz mxn.

FONTE: LETA et al., 2008.

Esta calibração determina as características geométricas internas da câmera tais como distância focal (f), e é guardada no vetor f_c (2×1), e este valor pode ser obtido a partir de: $x.dx/fc \cong f.s$ ou $y.dy/fc \cong f$, onde x e y são as distâncias entre centros dos elementos sensores vizinhos segundo as direções X e Y, e s é um fator de incerteza horizontal devido a erros de sincronização; coeficiente de distorção radial e/ou tangencial - representa os coeficientes de distorção da imagem (radial e tangencial) e são guardadas no vetor kc (5×1), sendo que $kc(1)$, $kc(3)$ e $kc(5)$ são os coeficientes da distorção radial de 2^a, 4^a e 6^a ordem, respectivamente, e $kc(2)$ e $kc(4)$ são os coeficientes da distorção tangencial de 1^a e 2^a ordem, respectivamente; fator de escala,

posição do centro ótico (ccx, ccy) que são guardadas no vetor cc (2×1), LENZ e TSAI (1989). Segundo WANG (1992), além dos parâmetros já citados devem ser considerados: dimensão do *pixel* nas direções x e y do sistema coordenado da imagem (dx, dy), fator de escala nas duas direções (sx, sy), a orientação do plano da imagem $\{pi\}$ descrita com relação ao plano da câmera $\{c\}$ - representa o ângulo entre os eixos x e y da imagem na memória *frame* e é guardado na variável *alpha_c*. Ocorre a transformação das coordenadas 3D no sistema câmera em 2D no sistema da memória *frame*. Técnicas de calibração de câmeras podem, em geral, ser classificadas de acordo com o critério de solução utilizado por BASU (1995):

- a. por equações lineares, tem-se a rapidez, facilidade de implementação, porém apresenta distorções que não são corrigidas;
- b. por otimização não linear considerasse as distorções na câmera e a ótima precisão, em contra partida, necessitasse de computação intensiva para cálculo, além de necessitarem de boa estimativa inicial para não levar à falsos resultados.

LENZ, TSAI e ZHANG propuseram técnicas de calibração que utilizam composições dos métodos citados anteriormente e fez-se o uso de algumas hipóteses simplificadoras, que dependem da aplicação do sistema visão robótico. O método a ser utilizado depende da aplicação, pois fatores como precisão, eficiência e versatilidade devem ser considerados.

2.6.2 Calibração extrínseca

A calibração extrínseca é a determinação de características geométricas externas como posição e orientação tridimensional do sistema da câmera $\{pi\}$ em relação a um sistema de coordenadas fixo tomado como referência $\{r\}$ LENZ e TSAI (1989), para isto são utilizados pontos conhecidos nos sistemas coordenados da imagem e de referência, utilizados para realizar a transformação entre as coordenadas 3D no sistema mundo e do sistema da câmera, para cada posição e/ou orientação da câmera/plano de calibração. Assim, existem seis parâmetros extrínsecos: a matriz de rotação (3×3) e as três componentes do vetor de translação (3×1). Simplificações podem ser utilizadas com o objetivo de facilitar a solução do problema. Em ambientes onde o manipulador trabalha com peças sobre um plano fixo, pode-se fazer constante a coordenada z e mais dois ângulos que definem a orientação da garra, assim o elemento final do robô estará sempre perpendicular ao plano de trabalho, sendo necessária a determinação somente de duas variáveis de posição (x e y) além do ângulo de orientação segundo o eixo z da câmera.

2.7 Manipuladores robóticos

A palavra robô, vem sendo cada vez mais utilizada em nosso dia-a-dia, a mesma teve origem de uma peça teatral tcheca do início dos anos 20, de autoria de Karel Capek, intitulada “Os Robôs Universais de Rossum”, que deu originem ao termo robô por GROOVER (1989). A palavra tcheca *roboata* significa servidão ou trabalho forçado e, quando traduzida para o inglês, transformou-se em *robot*, recebendo a forma robô no português.

A definição do robô industrial dada pela Associação das Indústrias de Robótica é a seguinte: “Um manipulador reprogramável, multifuncional, projetado para mover materiais, peças, ferramentas ou dispositivos especiais em movimentos variáveis programados para a realização de uma variedade de tarefas”.

Autonomia, versatilidade e reprogramação, são características que diferenciam os robôs de uma máquina automática projetada para uma determinada função não facilmente modificável. Segundo STADLER (1995), um sistema robótico é composto essencialmente dos seguintes componentes:

- Atuadores - produzem os movimentos. Fonte de energia são sistemas elétricos, pneumáticos ou hidráulicos;
- Comunicação - transmitir informações ou receber instruções de um operador remoto;
- Computador para controle - integra as atividades dos vários sistemas;
- Órgão terminal - conhecido também como punho, é o dispositivo de manipulação de objetos fixado ao final do último elo do manipulador;
- Manipulador - mecanismo que consiste em uma série de segmentos de braços fixados entre si através de juntas que proporcionam o movimento;
- Fonte de energia - sistema que fornece energia para o funcionamento dos diversos componentes;
- Sensores - transdutores que têm como entrada fenômenos físicos e como saída impulsos elétricos.

O robô é uma poderosa ferramenta de trabalho que auxilia os operadores na realização de atividades que comprometam a saúde e o bem estar do mesmo, BASTOS (1999), assim desenvolver novas tecnologias que trabalhem em paralelo com estes equipamentos podem solucionar outras questões

relacionadas a atividades que os mesmos não podem executar devido a falta de autonomia ou dados operacionais.

Então um manipulador é um braço mecânico que move a ferramenta para a posição desejada no espaço. Quando um robô é selecionado para uma determinada aplicação, alguns critérios devem ser avaliados:

- Graus de liberdade para manipular a ferramenta;
- Tamanho e a forma do espaço de trabalho;
- Carga máxima;
- Repetibilidade e precisão da tarefa;
- Velocidade máxima do robô.

2.7.1 Graus de liberdade

No âmbito de robótica, graus de liberdade referencia a quantidade mínima de informações necessárias para determinar as direções cartesianas que um elemento pode mover-se no espaço. Em geral, quanto mais graus de liberdade um manipulador possuir, menor será a carga útil que o mesmo suportará no elemento final, menor a velocidade de deslocamento do TCP, maiores serão os erros de posicionamento e mais elevado será o seu custo. Para descrever a posição de um objeto no espaço 3D, há necessidade de seis parâmetros, CRAIG (1989):

- ☒ três parâmetros definem a posição no eixo cartesiano: X, Y e Z;
- ☒ três parâmetros definem a orientação do objeto (rolamento (*yaw* - ϕ) – giro em torno do eixo X, arfagem (*pitch* - θ) – giro em torno do eixo Y e guinada (*roll* - ψ) – giro em torno do eixo Z).

Matricialmente tem-se a EQUAÇÃO (2.19):

$$R(\phi, \theta, \psi) = R_Z(\psi)R_Y(\theta)R_X(\phi) = \\ \begin{pmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi & \cos\phi \end{pmatrix} \quad (2.19)$$

Um robô de seis graus de liberdade consegue movimentar um objeto ou ferramenta em qualquer posição e orientação no espaço. Se este dispositivo tiver cinco juntas, um dos movimentos da ferramenta será dependente dos outros, e na FIG. 2.25 apresenta-se o manipulador utilizado neste trabalho.

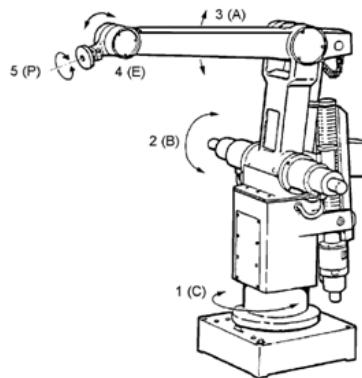


FIGURA 2.25 - Representação das juntas do manipulador ASEA.

FONTE: Product Manual ASEA, 1984.

Os graus de liberdade do corpo do robô definem a posição do centro da flange do punho. Assim, quando se move os eixos A, B e C do robô, cada qual num determinado ângulo, a posição do centro da flange fica perfeitamente definida. A partir daí, os graus de liberdade do punho (E e P) definem a orientação da flange, ou seja, o ângulo que o TCP forma com o antebraço do robô.

2.7.2 Geração de trajetória

Sistemas robóticos são produzidos para executar tarefas sob determinadas especificações, ou tolerâncias, sendo normalmente classificados pelo volume de trabalho, número de graus de liberdade, custo e precisão. O desempenho de um manipulador robótico pode ser avaliado considerando-se o volume de trabalho que é o espaço tridimensional no qual a ferramenta de trabalho, que é fixada ao *end effector* do robô, pode alcançar com o objetivo de realizar uma tarefa qualquer.

A precisão dos movimentos é subdividida em três: acuidade, repetibilidade e resolução espacial. Segundo PREISING (1995), a repetibilidade que está associada à cinemática inversa é obtida quando o comando do movimento, para um determinado alvo, é realizado no espaço das juntas de modo que o manipulador robótico apresenta sua capacidade de posicionar repetidamente seu

elemento final em um determinado ponto. Estes movimentos podem sofrer influências de folgas mecânicas, da flexibilidade e das limitações do sistema de controle. Em relação à precisão, esta indica o grau de aproximação do órgão terminal com relação a um determinado ponto dentro do volume de trabalho, sendo o oposto do erro cometido no posicionamento. E por fim, a acuidade é obtida quando o comando do movimento é realizado no espaço cartesiano.

Os robôs possuem diversos atuadores que são responsáveis pelo movimento de cada junta. O movimento combinado de todas as juntas cria a trajetória de movimento da extremidade do manipulador. O processo usado para coordenar os movimentos dos atuadores, para chegar a um curso desejado de movimento, é implementado dentro do computador do robô e é chamado de controle de trajetória.

O acionamento das juntas através da cinemática direta ou inversa, que permite que seja controlado a partir da especificação do ângulo de rotação ou da translação. Mas se forem enviadas para o controlador dos motores as coordenadas cartesianas de destino, a trajetória que será efetuada ao se movimentar do ponto inicial ao ponto especificado será desconhecida, ao contrário do que acontece com as informações de rotação e/ou translação de cada junta.

Para representar a orientação do objeto apresentado na FIG. 2.26, são necessários três vetores colunas que constituem uma matriz 3×3 , onde a orientação é definida a partir dos vetores unitários dos três eixos principais de $\{B\}$ em relação a $\{A\}$, obtendo-se a EQUAÇÃO (2.20).

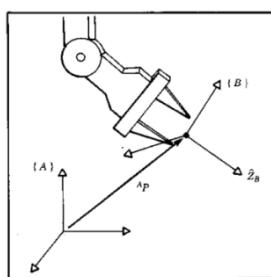


FIGURA 2.26 - Localização de um objeto em posição e orientação.

FONTE: CRAIG, 1989.

$${}^A_B R = [A_{X_B} \quad A_{Y_B} \quad A_{Z_B}] \quad (2.20)$$

Para descrever um sistema de coordenadas completo é necessário relacionarmos o sistema de coordenadas {B} em relação a um referencial inicial {A}, uma posição de (P_{BORG}) em relação a {A}, o que pode ser descrito pela EQUAÇÃO (2.21).

$$\{B\} = \{{}_B^A R, {}^A P_{BORG}\} \quad (2.21)$$

De posse de ${}_B^A R$ e ${}^A P_{BORG}$, é possível transformar a descrição de um ponto em {B} para {A}, como pode ser visto pela EQUAÇÃO (2.22).

$$A_P = {}_B^A R {}^B P + {}^A P_{BORG} \quad (2.22)$$

Essa equação pode ser substituída pela EQUAÇÃO (2.23), com isso o objeto pode ter sua orientação e posição representados por uma matriz reduzida.

$${}_B^A T = \begin{bmatrix} {}_B^A R & {}^A P_{BORG} \\ 0000 & 1 \end{bmatrix} \quad (2.23)$$

Deflexões nos elos e não-linearidades em transdutores de posição, reduzem a acuidade do robô, e para a realização deste cálculo é necessário considerar erros de posicionamento. Sendo Q o vetor da posição desejada e Q_i o vetor posição para o i -ésimo posicionamento, define-se o erro de posicionamento conforme VUOLO (1992).

$$E = Q_i - Q \quad (2.24)$$

A média para várias repetições é dada pela EQUAÇÃO (2.25):

$$\bar{Q} = \frac{1}{n} \sum_{i=1}^n Q_i \quad (2.25)$$

A acuidade é a precisão na qual o manipulador pode ir para um determinado ponto, e está relacionada com a cinemática, e é calculada conforme a EQUAÇÃO (2.26).

$$P = \bar{Q} - Q \quad (2.26)$$

Já a repetibilidade permite identificar o quanto “fora” do ponto desejado em uma série de tentativas o órgão terminal atingiu, ou seja, é a precisão com a qual um robô pode retornar a um ponto que ele esteve antes, e está esta relacionada com a cinemática inversa do robô. Atrito e folgas entre o transdutor de posição e uma junta afetam a repetibilidade. Desta forma tem-se que a repetibilidade é definida como sendo o desvio padrão das medidas de posicionamento, conforme PREISING (1995).

$$\sigma^2 = \frac{1}{n-1} \sum_{i=1}^n Q_i^2 - \frac{n}{n-1} \bar{Q}^2 \quad (2.27)$$

Sendo σ o desvio padrão das medidas ou a repetibilidade. Assim sendo, a acuidade de posição e a repetibilidade de orientação serão descritas conforme as equações que se seguem.

$$p = \sqrt{\sigma_x^2 + \sigma_y^2 + \sigma_z^2} \quad (2.28)$$

$$o = \sqrt{\sigma_\phi^2 + \sigma_\theta^2 + \sigma_\psi^2} \quad (2.29)$$

Sendo que ϕ, θ, ψ , os ângulos que representam as rotações em relação aos eixos x, y, z respectivamente. A resolução espacial não pode ser menor que o menor movimento detectado pelos sensores de posição ou menor que comando que o controlador é capaz de acionar. Atritos, deflexões elásticas, folgas, e outras imperfeições mecânicas contribuem para redução da resolução. A FIG. 2.27 apresenta uma ilustração dos conceitos de acuidade e repetibilidade

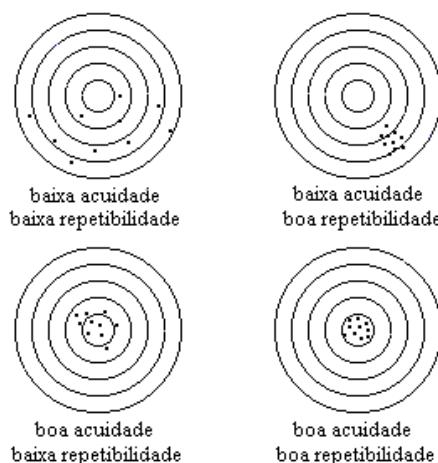


FIGURA 2.27 - Acuidade e repetibilidade.

FONTE: ADADE FILHO, 1992.

O controle de trajetória ponto a ponto envolve o posicionamento da extremidade do robô, em situações, sem ditar o curso exato a ser seguido entre estes pontos. Este método de controle é satisfatório para aplicações que exigem o posicionamento exato da extremidade do manipulador nos pontos onde serão executadas as operações, portanto o braço do robô está em repouso e não em movimento durante estas operações, o que ocorre, por exemplo, no processo de soldagem por resistência elétrica por pontos, em que trajetória pode ser realizada de diferentes modos, que estão apresentadas abaixo:

- Se dois motores do braço do robô começam a operar simultaneamente. O primeiro motor que move o TCP ao longo do eixo X, completará seu percurso antes que o segundo motor que move a ferramenta ao longo do eixo Y. Esta trajetória é mostrada na FIG. 2.28 (a);
- Se os dois motores começam a operar simultaneamente. O motor que move o TCP ao longo do eixo Y completará seu movimento antes do motor X. Esta trajetória é mostrada na FIG. 2.28 (b);
- E se os dois motores começam e terminam o movimento simultaneamente. Esta trajetória será o resultado de coordenação entre os motores do robô, visto que este começam e terminam os movimentos simultaneamente. Este modo resulta em movimentos mais suaves, sem pulos súbitos e em uma trajetória cujos segmentos curtos se assemelham a linhas diretas. Este método de coordenação entre os motores é conhecido como interpolação de juntas e é usado pela maioria dos robôs. Esta trajetória é mostrada na FIG. 2.28 (c).

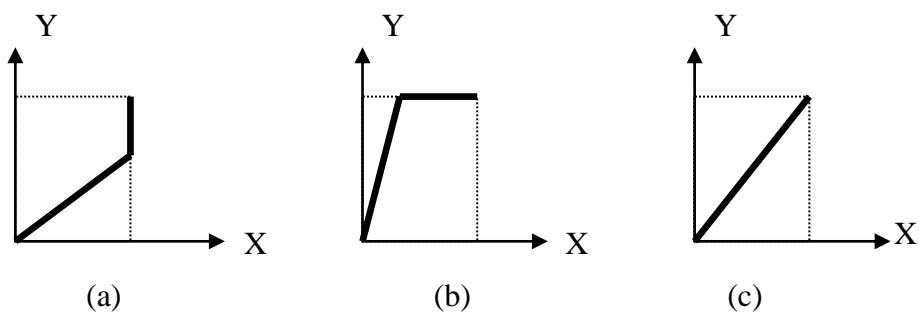


FIGURA 2.28 - Diferentes modos de trajetória.

FONTE: O AUTOR, 2015.

O movimento interpolado de juntas necessita que sejam realizadas várias computações preliminares antes dos comandos serem emitidos aos motores. O computador conhece as coordenadas das juntas do ponto atual e do ponto no qual o TCP será movido. Então, ele subtrai

um valor do outro, chegando a um conjunto de números que designa qual o movimento a ser executado por cada uma das juntas para chegar ao ponto desejado. Os movimentos dos motores do robô são coordenados por envio simultâneo dos comando de movimento de cada motor (intervalos curtos e periódicos). A quantidade de movimento comandada para cada intervalo permanece proporcional à relação da quantidade total de movimento requerida para cada junta, como determinado pela diferença das coordenadas destas.

No controle de trajetória contínua assegura-se movimentos da extremidade do robô por todos os pontos de uma trajetória desejada, ao invés de posicionamento exato somente no ponto inicial e final como em controle de ponto a ponto. Portanto, controle de trajetória contínua pode ser usado enquanto o braço do robô está em movimento, como soldagem a arco, pintura por spray e assim por diante.

O controle de trajetória contínua pode ser realizado através de dois métodos completamente diferentes. No primeiro, o braço é movido manualmente (por um operador humano) ao longo da trajetória desejada, e as coordenadas das juntas são registradas na memória do computador. O controlador simplesmente repete o movimento registrado sempre que é operado. Os controladores dos robôs que usam este método requerem uma memória consequentemente grande, mas não tem que executar cálculos complexas durante o movimento. O segundo método envolve comandos ao controlador não somente para o ponto inicial e final, mas a forma exata da trajetória a ser executada pela extremidade do robô entre estes pontos (por exemplo, uma linha reta). O controlador executa os cálculos exigidos para mover o braço ao longo da trajetória desejada à uma velocidade desejada e enviam comandos de movimento aos motores do robô de acordo com os resultados destes cálculos. O controle calcula as coordenadas das juntas desejadas as quais irão ser transmitidas como comandos aos motores do robô para mover o braço ao longo da trajetória desejada. Esta computação pode ser bastante complexa e aumenta com o número total de graus de liberdade do robô. Três observações adicionais que interessam ao controle de trajetória contínua:

- O movimento alcançado não é necessariamente uma linha reta exata. Quanto menor a distância entre dois pontos consecutivos – o número máximo de segmentos nos quais a trajetória é dividida – mais o movimento se assemelhará a uma linha direta;

- A velocidade desejada determina o número máximo de pontos nos quais as trajetórias podem ser divididas. Quanto maior a velocidade, menor é a distância percorrida entre dois pontos consecutivos;
- Além da velocidade, o controle de trajetória deveria também levar em conta o princípio da aceleração e desaceleração do braço no começo e fim de cada movimento para evitar arranques causados pela partida e parada descontrolada.

2.7.3 Soldagem robotizada

O processo de soldagem quando realizado de forma automática é dotado de componentes e dispositivos mecanizados/eletrônicos que permitam a movimentação do TCP, e também de sensores para monitorar parâmetros de soldagem e a partir deles, controlar todo o processo. Quando o processo se torna robotizado, diz-se que o mesmo é uma forma específica de soldagem automática a qual é definida pela *American Welding Society – AWS* (Associação Americana de Soldagem) como: “soldagem com equipamento (robô, manipulador, etc.) que executa operações de soldagem, após programação, sem ajuste ou controle por parte do operador de solda”. Os processos de soldagem mais comumente que podem ser robotizados, são:

- ✓ GMAW;
- ✓ GTAW;
- ✓ LBW;
- ✓ PAW;
- ✓ RSW.

O processo de soldagem a arco elétrico é um processo contínuo de soldagem usado para fazer juntas soldadas longas onde uma conexão sólida entre as partes metálicas é exigida. As correntes variam entre 100A e 300A, e voltagens de 10V a 30V, e o arco elétrico formado entre a tocha de soldagem e as partes metálicas produz uma temperatura elevada o suficiente para fundir as peças, além de produzir radiação ultravioleta, tornando este processo perigoso para o trabalhador humano quando este estiver submetido durante muitas horas consecutivas. O robô é portanto adequado para esse tipo de operação uma vez que ele pode seguir uma trajetória pré-definida de soldagem. Um dos problemas que o robô pode apresentar é no caso de áreas de difícil acesso, que apresentará dificuldades de alcançar. Existem também problemas decorrentes de variações dimensionais e nas formas das peças a serem soldadas. Em um processo robotizado, o

manipulador normalmente não tem como compensar estas situações, sendo este problemas difíceis para os sensores. Com isso, a aplicação de sensores de visão tendem a visualizar a área de trabalho do robô, e fornece informações sobre este ambiente, principalmente sobre o item a ser trabalhado, de modo a permitir que o controlador utilize esses dados para correções e tomada de decisões.

O robô certamente tende a ajudar, porque seu uso permite que as pessoas sejam poupadadas de situações perigosas ou insalubres. Isto não só melhora as relações operárias mas também pode aumentar a produtividade pela eliminação de paradas para repouso e recuperação que são requeridas frequentemente. Na luz destes fatos deveria ser reconhecido que os robôs e as pessoas são muito diferentes, e que devem ser feito treinamentos com os melhores soldadores para que possam programar um sistema robótico, e obter o máximo aproveito do mesmo, e não o contrário.

2.8 Trabalhos relacionados

BRACARENSE (1986), apresentou um dos primeiros sistemas automáticos para guiar a tocha de soldagem no processo TIG, e quanto a variação dos parâmetros, bem como do arco elétrico são influenciados pela alteração da geometria do eletrodo não consumível.

DUTRA e ROSA (1995), apresentaram um sistema em que o sensor é o próprio arco voltaico que fornece informações da posição da tocha por causa da variação da corrente que circula por ele quando se altera a altura do bico de contato da tocha em relação à mancha catódica, em uma junta chanfrada ou de filete. Assim, com apresentado por BRACARENSE (1986), este sistema apresenta como vantagem o fato de não haver custo com sensor, não há *offset* (distância entre a tocha e o sensor), indiferente a radiação luminosa provocada pelo arco, além dos respingos e fumos, e dispensa o uso de um sistema óptico.

CORKE et al. (1996) apresentaram um manipulador robótico dotado de um sistema de visão em um sistema em malha aberta, "Ver depois Mover" - *Look then Move*, onde a extração da informação visual e o controle do movimento são tarefas totalmente separadas, o controle visual obtém informações da posição alvo do manipulador e gera sua trajetória sem preocupar-se com o entorno do robô.

RODRIGUES (1997) analisou a robustez de um procedimento para calibração de robôs através da visão computacional. Sistema dotado de uma câmera CCD e utilizando um robô do tipo SCARA, avaliou e concluiu que apesar de todas as fontes de incertezas, tais como, erros de discretização de imagem, cálculos cinemáticos e alinhamento da câmera com o elo do manipulador, o sistema apresentou-se robusto a estas incertezas e problemas relacionados com alinhamento do eixo óptico não é considerado como crítico.

KOIKE et al. (1999) apresentaram um sistema desenvolvido para monitorar a poça de fusão no processo de soldagem GMAW, onde foi abordado técnicas de uso de visão computacional e algoritmos de processamentos de imagens que extraem a largura do e da poça, e o alinhamento do arame de modo a obter melhor qualidade de soldas. A obtenção das imagens ocorre durante o processo de soldagem, onde existe excesso de luminosidade devido o arco voltaico, que foi filtrado através de um filtro óptico passa-banda com centro no comprimento de onda do laser. O algoritmo utilizado baseou-se no pré-processamento para selecionar a região de interesse, reduzindo a quantidade de *pixels* a serem analisados, com isso, tem-se um custo computacional menor.

RUDEK et al. (2001) desenvolveram um sistema de visão que auxilia a manufatura em fábricas, na seleção de parafusos. Além de apresentar uma metodologia para desenvolvimento de um banco de dados para arquivar, recuperar e manipular, com base no comportamento dos objetos nas imagens obtidas. Com isso, fez-se o reconhecimento e classificação de imagens semelhantes através de sua variação de cores. Se um parafuso for identificado corretamente em uma célula de manufatura, esta poderá obter todas as informações relativas deste objeto, e permite que as operações que utilizam estes parafusos sejam realizadas de forma eficiente. Entretanto, se a peça ainda não existir no banco, ela é classificada com outros atributos.

MOLINA et al. (2002) apresentaram um sistema de visão utilizado também em processos de soldagem, baseado em segmentação de imagens por textura e usando o segundo momento de inércia da imagem segmentada, para encontrar a linha de solda, possibilitando a correta detecção do cordão. Neste trabalho é feito uma comparação entre a Transformada Hough e o Momento de Inércia, onde o primeiro apesar de bastante eficiente para detecção de retas em imagens, é sensível a ruídos e a problemas de iluminação, o que deixa o resultado dependente da segmentação aplicada, já o segundo tem um menor custo computacional e leva em consideração características intrínsecas da

imagem para determinar objeto alongado que representa o cordão de solda. A segmentação baseia-se em informações de textura obtidas a partir do segundo momento central.

ALFARO et al. (2003), desenvolveram um sistema de visão que permitiu automatizar processos de soldagem utilizando robôs. Este é semelhante ao primeiro apresentado nesta seção, tendo como diferencial o modo de processar a imagem e obter resultados em tempo real adotando lógica Fuzzy. Que permitiu uma escolha sobre os dados processados convencionalmente, e controlando as variáveis geométricas obtidas de imagens como variáveis do processo de soldagem.

LIMA II (2005) desenvolveu o trabalho "Soldagem Robotizada com Eletrodo Revestido", que visava o desenvolvimento de um processo de soldagem com eletrodo revestido utilizando uma manipulador robótico que permitisse ao programador um controle total sob a variação de parâmetros e movimentos. Para isso era necessário que o robô fosse dotado de arquitetura de aberta para programação que permitisse o desenvolvimento de algoritmos para geração de trajetórias. Por fim, desenvolveu-se a modelagem cinemática do IRB6 e geração de trajetórias em *MatLab®*.

GRASSI (2005) adotou um sistema de visão agregado a um manipulador robótico com arquitetura fechada para a tarefa de manipulação. Utilizou uma câmera CCD para captura da imagem, e o programa foi desenvolvido em linguagem *Visual Basic*, que após a determinação da posição, orientação da peça envia ao controlador do manipulador em linguagem nativa do mesmo as coordenadas da peça a ser manipulada.

CHAUMETTE e HUTCHINSON (2006) baseado no que foi desenvolvido anteriormente por SHIRAI et al. (1973) e por CORKE (1996), eles trabalharam na melhora da precisão do sistema robótico, introduzindo dentro do sistema de controle o sinal de realimentação visual em forma de malha fechada, sendo que, a informação visual é introduzida a cada ciclo do controle a fim de corrigir a posição do manipulador até atingir a posição desejada.

COELHO et al. (2013) descrevem a aplicação de um sistema de visão desenvolvido com equipamentos não dedicados e agregados em um manipulador robótico remanufaturado que utiliza CNC. Uma webcam foi montada no elemento final do manipulador que identificava o tamanho da chapa e definia a dimensão do cordão de solda a ser efetuado.

PINHO (2013) propôs uma nova arquitetura em código aberto para acionamento e controle do robô ASEA IRB6, de modo a controlar os motores e ser capaz de fazer a leitura de dados dos sensores. Esta arquitetura foi baseada em microcontroladores PIC 18F2550, ligados em uma rede com protocolo Modbus utilizado para se comunicar com o computador através da porta serial. A possibilidade de controlar os motores de forma individual ou conjunta, aliando o *MatLab*® com o PIC, além de definição da velocidade, foi o diferencial deste trabalho.

VILARINHO et al. (2013), adotaram o uso de um sistema de visão baseado em controle de soldagem através de um seguidor de junta. Este sistema de visão compreende o uso de uma câmera CCD e um sistema laser. Durante o processo de soldagem é possível a visualização e variação dos parâmetros. Este trabalho difere do que foi desenvolvido por DUTRA e ROSA, pois neste foi adicionado um sistema de monitoramento do sinal via *Wireless*.

Este capítulo apresentou conceitos que serão necessários para o desenvolvimento do sistema de visão que será utilizados no Robô ASEA IRB6, no contexto em que o projeto está sendo desenvolvido. No Capítulo 3 serão apresentadas a aplicação destes conceitos neste projeto.

3. MATERIAIS E MÉTODOS

3.1 Método

Este trabalho aborda o desenvolvimento de um sistema de visão que apresente um baixo custo agregado a *hardware* e *software* independentes, aplicados em um manipulador robótico fixo industrial remanufaturado. O desenvolvimento deste sistema, bem como de qualquer outro, requer a determinação de equipamentos e avaliação das condições de operação. Nesta pesquisa, adotou-se o mesmo método que foi desenvolvido por CORKE (1996), sendo que sua aplicação está limitada a tarefas altamente específicas e seu desempenho depende da correta calibração tanto do sistema mecânico quanto o visual.

Para o auxílio na calibração do sistema mecânico do manipulador remanufaturado, foram utilizados acelerômetros conforme apresentado por LOPES (2015), de modo a garantir que a quantidade de pulsos enviados aos *drives* dos motores seja a necessária para a realização de movimentos e deslocamento em trajetória definida. Na FIG. 3.1 tem-se a montagem para calibração da junta 2.

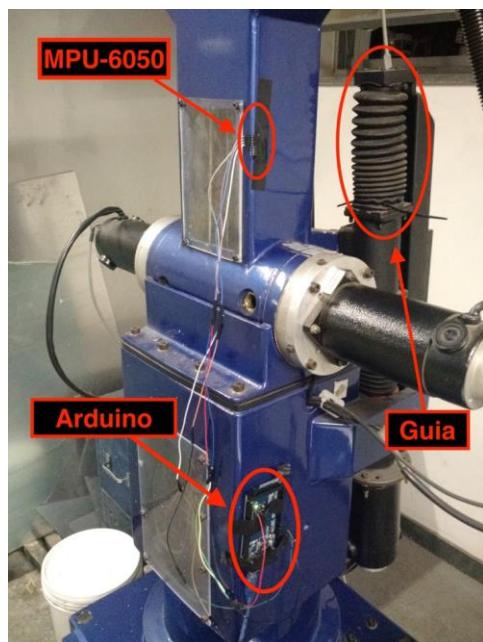


FIGURA 3.1 - Calibração da junta 2 com acelerômetros.

FONTE: LOPES, 2015.

O projeto do sistema de visão computacional de baixo custo tem como estrutura a FIG. 3.2 que apresenta uma sequência desenvolvida, e que é explicada a seguir.

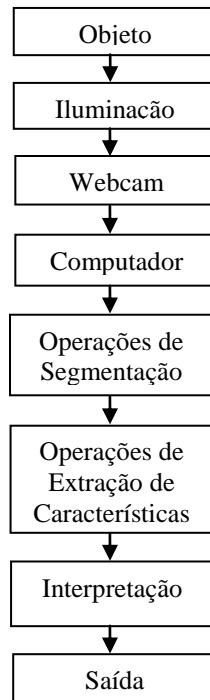


FIGURA 3.2 - Sequencia desenvolvida.

FONTE: O AUTOR, 2015.

O item denominado "Objeto" está limitado a uma peça com dimensão mínima de 100 mm x 100 mm x 10 mm, e os atributos a serem extraídos da imagem dizem respeito à forma do objeto que possuem superfícies planas, que na aplicação das técnicas de análise de imagens mantém uma relação homográfica. No quesito "Iluminação", tratou-se do uso da luz natural do ambiente sem ter que realizar alterações sobre o mesmo, com isso, conseguiu-se reduzir o custo de alguns elementos que são utilizados em um sistema de visão, tais como: a escolha da *WebCam* como elemento sensor, tornando-se capaz de substituir ou até mesmo eliminar os tópicos de "Lentes e Óptica" + "Câmera" + "Placa de Aquisição". Nas etapas de "Operações de Segmentação" e "Operações de Extração de Características", estas foram pensadas para que fossem utilizados algoritmos simples, mas funcionais para a proposta do sistema. E as duas últimas etapas de "Interpretação" e "Saída", o proposto para o sistema deve ser capaz de após determinar as características da peça na imagem, converter em informações métricas para que o manipulador possa se deslocar sobre a peça no ambiente 3D.

No que se trata da especificação funcional, deve-se determinar os requisitos mínimos, bem como as limitações do sistema proposto. De modo a:

- ❖ Descrever de forma geral, o modo de operação e seu benefício;
- ❖ Descrever os objetos, mediante seu tamanho, orientação, cor, textura, etc.;
- ❖ Exigência de desempenho, baseado na velocidade, acurácia, confiabilidade;
- ❖ Necessidades mecânicas, quanto ao limite de tamanho físico e montagens;
- ❖ Iluminação do ambiente, sujeira, poeira;
- ❖ Interface do equipamento, dispositivos acoplados;
- ❖ Interface com o operador.

O posicionamento da câmera na cena requer o conhecimento de diversos aspectos, como onde os objetos de interesse estão dispostos, quais objetos podem confundir o sistema de visão e onde eles estão localizados na cena. Deste posicionamento, consegue-se determinar uma área de atuação do sistema. Inicialmente apresenta-se a configuração desenvolvida pelo sistema proposto, e a FIG. 3.3 representa esquematicamente esta configuração.

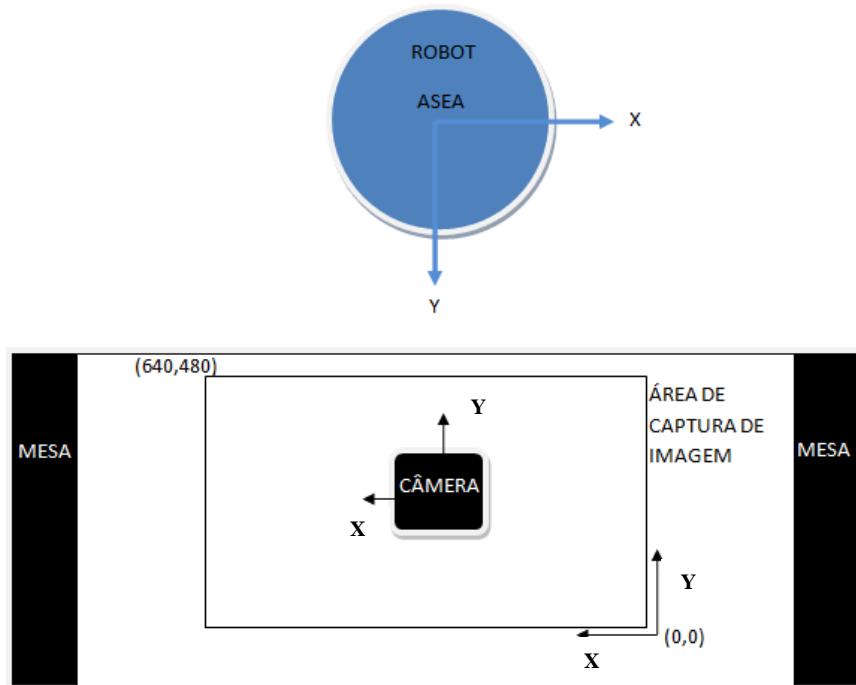


FIGURA 3.3 - Representação esquemática do sistema proposto.

FONTE: O AUTOR, 2015.

Em relação ao tipo de ajuste de foco, foram avaliadas as câmeras que possuem foco fixo, ou seja, que não permitem ajuste de acordo com a distância do objeto, bem como a varifocal, aquela que permite ajuste manual da distância de seu foco.

Quanto à lente, o dispositivo escolhido possui uma com íris fixa, que não permite a abertura ou fechamento, ou seja, a quantidade de luz que entra não é alterada. Este tipo deve ser utilizado em locais onde a luminosidade é boa e constante.

A variação do parâmetro iluminação pode gerar coordenadas incorretas, e vale ressaltar que os sistemas de visão artificiais funcionam muito bem apenas para as situações que foram programadas. Se o resultado do processo de segmentação não for o esperado, deve-se então alterar a iluminação ou o contraste entre o objeto e o fundo da imagem.

A seguir é apresentada uma prévia do que se obtém seguindo a metodologia proposta. A informação a ser trabalhada pelo sistema de visão, a imagem, é capturada por um elemento transdutor, uma *webcam*, e então é digitalizada e armazenada para que em seguida possa ser processada por um algoritmo. Este, por sua vez, é responsável por todo desenvolvimento do tratamento e interpretação da imagem e extração das informações desejadas. Uma interface entre o computador e o robô torna-se necessária para transformar os dados resultantes do processamento da imagem obtida em um formato que o controlador do manipulador robótico comprehenda. O objetivo do módulo de visão é a conversão das imagens em YUV (este é o formato em que são disponibilizadas as imagens da *webcam*, que são versões escaladas dos sinais $U = 0,492$ (B-Y) e $V = 0,877$ (R-Y)) em escala de cinza, e posterior processamento para a obtenção da informação sobre a localização do objeto. Após receber os dados na linguagem adequada, o controlador envia comandos para os atuadores do robô, e então a tarefa é executada. Por fim este sistema obtém uma nova imagem que analisa computacionalmente se o objetivo da tarefa foi realizado com sucesso, fazendo assim um controle qualitativo do processo. A cena em que o sistema está inserido apresenta as seguintes características:

- Ambiente físico e real com iluminação constante;
- O formato do objeto restringe-se ao retangular e quadrado;
- A posição de captura da imagem não varia de acordo com a espessura da peça.

Essas características determinam o ambiente como sendo parcialmente observável, estático e contínuo. Esses fatores resultam em um ambiente que necessita de algumas simplificações para ser interpretado de forma conveniente. A análise está relacionada a parte do tratamento onde existe uma descrição da informação presente na imagem, parametrizando com a intenção de descrever situações dentro de uma imagem, e melhoria da qualidade da imagem tem como objetivo aumentar o contraste ou colocar em evidência alguma região de interesse em particular.

3.2 Materiais

No QUADRO (3.1) apresentam-se os elementos necessários desde a etapa de obtenção e processamento de imagens, até a realização da soldagem.

QUADRO 3.1

Lista de materiais e equipamentos.

Utilizado para	Identificação do Elemento Utilizado		
5	Manipulador Robótico ASEA-IRB6		ASEA®
1	<i>WebCam Microsoft® LifeCam VX-800</i> Sensor: CMOS VGA; Resolução: (640x480 pixel); Foco fixo.		Microsoft®
1	<i>WebCam Nipponic® NIP-VC76P</i> Sensor: CMOS VGA; Resolução: (800x600 pixel); Foco manual.		Nipponic®
6	CPU: processador Pentium Dual Core E 2140 – 1.6GHz, placa mãe GIGA BYTE GA-915GZM-S2, memória 1.0GB DDR2, HD 160 GB SATA, portas seriais, porta paralela e portas USB.		DELL®
7	<i>Software Mach3®</i>		ArtSoft®
1, 2, 3, 7	<i>Software MatLab®</i>		The MathWorks Inc.®

7	<i>Encoder HUBDISK-2 -</i> 2048 ciclos por revolução (2048 CPR)		USDigital®
7	Módulo de transmissão do <i>encoder</i> óptico – EM1-2-2048-I – com 02 (dois) canais de quadratura e 2048 CPR de resolução.		USDigital®
7	<i>Driver G320 Servo Drive.</i> Tensão de operação de 18 a 80V e frequência de operação do sinal <i>PWM</i> de 20kHz.		GeckoDrive®
8	Motor C.C. 70V, 5A. Velocidade máxima de 3.300 rpm.		Keling®
7, 9	<i>Breakout board</i> modelo C1G.		CNC4®
10	Fonte de alimentação de 1.750W/70V.		-
4, 11	Chapas de aço carbono 1045 de dimensões 150X250.		-
4, 11	Consumível: Arame Sólido SFA/AWS A5.28 ER120S-G com diâmetro de 1.0mm. Taxa de Deposição: 5.4kg/h. Tensão de operação: 18 - 28V. Corrente elétrica de operação: 80-280A. Especificação Esab®: ARISTOROD 79.		Esab®
4	Máquina para soldagem <i>FCAW</i> modelo VI 320.		White Martins®
4	Conjunto tocha, suporte da tocha e sensor anti-colisão modelo Tbi KS-1.		TBI®

4, 11	Gás de proteção (100% dióxido de carbono).		White Martins®
Legenda (Utilizado para): 1 - Obter Imagens; 2 - Processar Imagens; 3 - Gerar Trajetórias; 4 - Realizar Soldagem; 5 - Manipular; 6 - Armazenar, gravar, processar, gerenciar; 7 - Converter; 8 - Movimentar; 9 - Interfacear; 10 - Alimentar; 11 – Proteger			

Fonte: O AUTOR, 2015.

Para os *softwares* apresentados no Quadro 3.1 e que foram utilizados no desenvolvimento deste trabalho, pode-se falar que o *MatLab®* 7, que está apresentado na FIG. 3.4, foi selecionado por ser difundido no uso de simulação matemática, com capacidade de realizar operações matriciais, processar sinais, além de manipular outras funções especializadas, e esta versão está licenciada para uso no laboratório. Ele trabalha com uma linguagem de programação de alto nível, em um ambiente interativo, para o desenvolvimento de algoritmos, análise e visualização de dados e computação numérica. Além do ambiente interativo, outra facilidade do *MatLab®* é a possibilidade de execução de arquivos texto, contendo uma sequência de instruções definidas pelo usuário e que podem ser criados e editados dentro ou fora do seu ambiente CHAPMAN (2011).

Em se tratando de cinemática, a manipulação de matrizes de transformação homogênea 4x4 é logo pensada, de modo que o movimento de um elemento que necessite de uma interpolação torna-se facilmente descrita utilizando-se o *software* manipulador de matrizes. Para programar a cinemática do manipulador ASEA, é então necessário gerar um código fonte das cinemáticas inversas e da inversa de atuadores, e tem como problema encontrado, é a utilização do *MatLab®* para modular as entradas *step/direction* dos *drives* com os dados obtidos destas operações. Uma solução encontrada foi a utilização do *software* Mach 3®, que permite desenvolver programas com até 500 linhas de Código *G*, sem a necessidade de adquirir sua licença, conforme afirmou BOMFIM (2013).

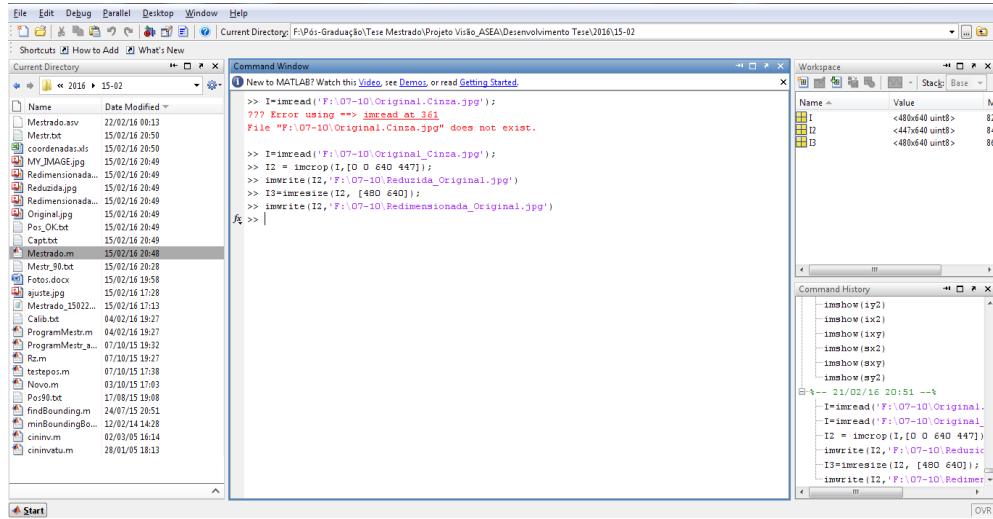


FIGURA 3.4 - Tela principal do *software MatLab*®.

FONTE: O AUTOR, 2015.

A ArtSoft® fabricante do *Mach3*® relata que o *software* é um controlador que viabiliza o controle de equipamento com até 6 eixos, tanto prismáticos quanto rotacionais, interpretando o *G-Code* para máquinas de controle numérico, e outro ponto de estímulo é a grande rede de comunicação existente entre os usuários do sistema pelo o mundo. Os diversos fóruns e grupos de pesquisas proporcionam uma excelente base de suporte, o que torna mais eficiente e rápido o desenvolvimento de novas aplicações e funcionalidades. A FIG. 3.5 apresenta a tela principal do *software*, e vale ressaltar os dois pontos destacados na imagem, que são o quadro maior localizado a esquerda encontram-se os comandos gerados em Código G, e os valores contidos nos eixos X, Y, Z, A, B, e C representam o deslocamento de cada um, ou seja, os pulsos enviados dos *drives* aos motores.



FIGURA 3.5 - Tela principal do *software Mach3*®.

FONTE: O AUTOR, 2015.

Outra qualidade importante do *software* é permitir a inclusão de sistemas de monitoramento externo, na FIG. 3.6 é apresentado o ambiente de trabalho sendo destacado o acréscimo de um botão de emergência para que o movimento do manipulador possa ser interrompido caso seja detectado alguma colisão ou situação indevida.



FIGURA 3.6 - Ambiente de trabalho.

FONTE: O AUTOR, 2015.

Manipuladores robóticos BROERING (2005), podem realizar soldagem por longos períodos, bem como proporcionar maior produtividade, repetibilidade e melhor condição de trabalho ao soldador, uma vez que este passa a não mais atuar diretamente, ou constantemente, em um ambiente altamente insalubre. Sendo o robô um equipamento flexível, é factível a rápida alteração do *software* de controle para execução de tarefas distintas. Logo, além de ter maior produtividade comparada ao processo manual, atua em diferentes tarefas, seja no processo de fabricação de peças ou em manutenção, fato que não ocorre em automações dedicadas.

4. RESULTADOS

4.1 Seleção da câmera

A seleção da câmera é um processo que se torna necessário quando seu sistema de coordenadas apresenta diferenças em relação às coordenadas do mundo real. O procedimento de calibração foi validado utilizando diversas imagens para avaliar sua sensibilidade. Neste contexto, valores nominais de uma câmera comercial foram usados e os erros e incertezas dos sistemas calculados. Os resultados encontrados são satisfatórios para aplicações mais robustas, que admitam erros na ordem de três milímetros, compensados através de um fator de correção no algoritmo. O procedimento de calibração necessita de um conjunto de pontos na área da imagem para que se consiga implementar a calibração por meio da avaliação desta matriz de transformação entre o sistema de visão e o sistema de referência do mundo.

O modelo geométrico que se espera obter com a calibração deve ser uma derivação do modelo *pinhole* geralmente adotado para as câmeras, explica FAUGERAS (1993). Neste modelo, o processo de formação da imagem na câmera é aproximado por uma projeção em perspectiva, onde as coordenadas de um ponto do ambiente e a sua projeção no plano de imagem são relacionadas pela EQUAÇÃO (4.1), e representado de matematicamente na EQUAÇÃO (4.2).

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \quad (4.1)$$

$$p = M_{int} \cdot P \quad (4.2)$$

onde p representado por x' , y' e z' são as coordenadas do ponto no plano da imagem, e P denotado por X , Y , Z são as coordenadas do ponto no ambiente, e M_{int} é matriz resultante que possui informações de calibração dos parâmetros intrínsecos da câmera. Por fim, f representa as distâncias focais.

Quando precisão é um requisito, um modelo mais fiel deve ser utilizado. Normalmente, o modelo *pinhole* é estendido para considerar a distorção causada pelas lentes, também conhecido como aberração esférica. As principais correções utilizadas são para as componentes radiais e a

tangencial da distorção, através da adição de alguns coeficientes ao equacionamento exposto acima.

Uma lente de baixa distorção permite medições altamente precisas. Uma lente com uma distância focal curta captura uma imagem ampla de um alvo a uma distância curta, mas a distorção da imagem aumenta. A distorção da imagem aumenta mesmo quando se utiliza uma lente olho-de-peixe, que tem um campo de visão de quase 180°.



FIGURA 4.1 - Distorções em uma lente.

FONTE: KEYENCE, 2015.

Diversos são os métodos utilizados para obter os conjuntos de parâmetros da câmera e estes podem ser encontrados na literatura ZHANG (1996), HEIKKILÄ e SILVÉN (1997), FAUGERAS (1993). Neste trabalho foi utilizado o método descrito em HEIKKILÄ e SILVÉN (1997), aplicando a metodologia em *MatLab*® através de BOUGUET (2013), por considerar a distorção causada pelas lentes e por dispensar a utilização de arranjos ou procedimentos experimentais complexos. Através do uso deste método, é possível realizar a calibração a partir da observação pela câmera, de um gabarito de calibração plano a partir de diversos (pelo menos dois) pontos de observação diferentes.

A calibração de sistemas de visão para aplicações em robótica tem como objetivo a determinação da relação entre o sistema de coordenadas da câmera e o sistema de coordenadas de referência do manipulador robótico. Desta forma é possível a utilização do sistema de visão no auxílio de diversas atividades para as quais o manipulador possa estar programado.

A calibração da câmera independente do ambiente em que a mesma está inserida foi realizada como um elemento avaliativo das características intrínsecas da mesma que auxiliaram na escolha do melhor dispositivo óptico. Para tal foi realizado o seguinte procedimento, inicialmente foi utilizado o padrão conhecido como *checkerboard*, ou também chamado de *chessboard*, que está apresentado na FIG. 4.2.

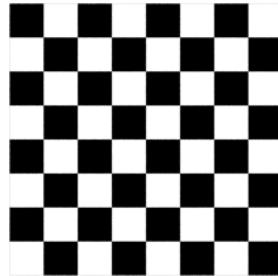


FIGURA 4.2 - *Checkerboard* (tabuleiro de damas).

FONTE: O AUTOR, 2015.

Este padrão possui sessenta e quatro objetos, representados pelos quadrados, distribuídos em toda a área que será focalizada pela câmera. O padrão é composto por quadrados pretos e brancos, facilitando a detecção dos vértices e quinas e a interseção de linhas da imagem com a ajuda do alto contraste e sua geometria simples. Quanto maior o número de imagens capturadas e de pontos utilizados no procedimento, menor será o erro encontrado e mais robusto será o sistema de visão. Estes pontos de calibração em diferentes posições na imagem englobando toda sua área é possível calcular uma matriz de transformação homogênea consistente que irá representar a correspondência entre os sistemas coordenados de referência do mundo com o sistema de coordenadas da imagem, pois o erro no cálculo da matriz é minimizado por um fator de N , que representa o número de pontos de calibração, como demonstrado por LENZ e TSAI (1989).

A única informação que deve ser conhecida com precisão é o tamanho das arestas dos quadrados que compõem o gabarito e o número de quadrados nas direções horizontal e vertical. Neste método, o conjunto de parâmetros é obtido através da minimização iterativa da distância entre as coordenadas dos pontos do gabarito nas imagens utilizadas na calibração e as coordenadas dos pontos reprojetados nestas mesmas imagens, segundo um conjunto de valores iniciais para os parâmetros, geralmente obtido pela solução analítica de um sistema linear derivado da EQUAÇÃO (4.2), solução que somente é possível se os coeficientes de distorção não forem

considerados. Uma implementação deste método está disponível na forma de um *Toolbox* para o *MatLab®* na página da *Web* em http://www.vision.caltech.edu/bouguetj/calib_doc/, que permite ler imagens, extrair pontos, calibrar câmeras, controlar precisões, corrigir imagens distorcidas, alterar o modelo intrínseco da câmera escolhendo quais os parâmetros a otimizar. Todas estas e outras ferramentas que compõem o método de calibração, baseiam-se em HEIKKILÄ e SILVÉN (1997). Um método indicado como extremamente preciso HERBST (2003), utiliza um procedimento de quatro passos, que são os seguintes: no passo 1, é realizada uma aproximação linear para determinar uma solução analítica dos parâmetros da câmera (sem considerar a distorção da lente); no passo 2, estima-se a distorção radial e tangencial da lente e dos restantes parâmetros a partir de uma otimização não linear utilizando o método dos Mínimos Quadrados; no passo 3, é realizada uma correção e posterior extração dos pontos de calibração na imagem; e por fim no último e quarto passo, ocorre à correção das coordenadas que possam ter sido obtidas através de imagens distorcidas. Na FIG. 4.3 apresenta três pontos identificados na cor vermelha, que foram inseridos no ambiente para auxiliarem no processo de calibração.

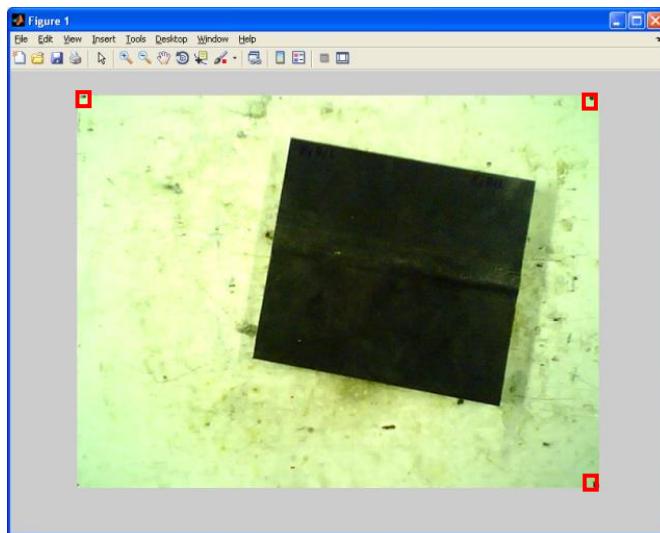
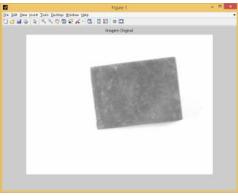
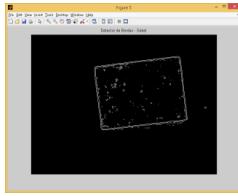
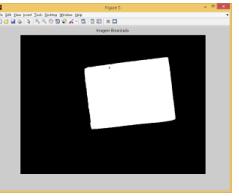
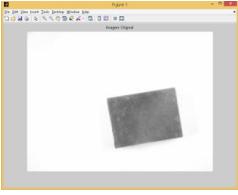
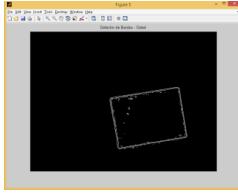
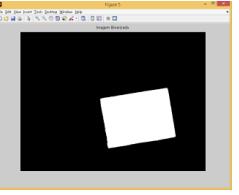
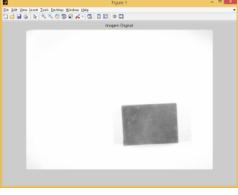
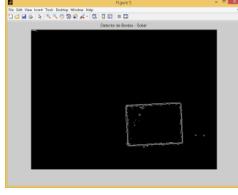
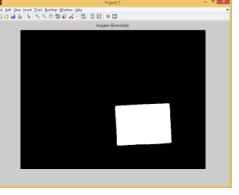
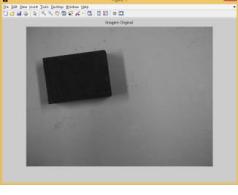
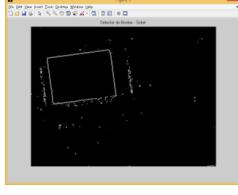
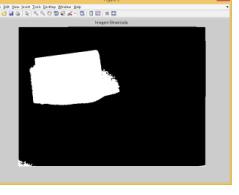
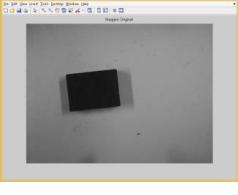
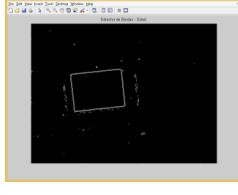
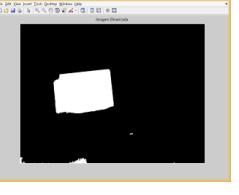
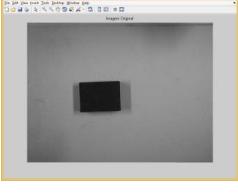
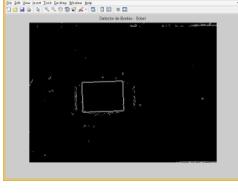
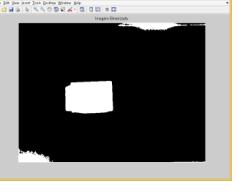


FIGURA 4.3 - Definindo pontos de calibração na cena.

FONTE: O AUTOR, 2015.

Para a escolha da câmera foi realizado o mesmo procedimento para ambas, de modo a optar por aquela que apresentar melhores resultados. Esta etapa foi realizada em duas fases, sendo nesta primeira a obtenção de novas imagens de modo aleatório, sendo que foram retiradas 03 fotos com ambas as câmeras de custo baixo em alturas diferentes, e estão apresentadas no QUADRO (4.1) como imagem original, o detector de bordas “SOBEL” e binarização.

QUADRO 4.1
Resultado dos testes realizados com as *WebCam's*.

Fotos obtidas com a câmera com ajuste de foco automático:		
1 - Imagem Original	2 - Detector de bordas SOBEL	3 - Binarização
		
		
		
Fotos obtidas com a câmera com ajuste de foco manual		
1 - Imagem Original	2 - Detector de bordas SOBEL	3 - Binarização
		
		
		

FONTE: O AUTOR, 2015.

No QUADRO (4.1) apresentado anteriormente é possível observar que as imagens em escala de cinza da câmera com ajuste de foco automático, mesmo que em um primeiro momento apresentassem uma menor qualidade, foi possível obter resultados melhores, ou seja, são menos sensíveis a ruídos, com melhor detecção do objeto de interesse, conforme apresentado nas imagens binarizadas à direita, para ambos os equipamentos.

A segunda fase consistiu em utilizar o *checkerboard*, e obter sete imagens diferentes. Para o processo de calibração a FIG. 4.4 apresenta o mosaico obtido com a primeira câmera, enquanto que na FIG. 4.5, tem-se os vértices dos quadrados encontrados nas imagens durante o processo de extração de quinas (*Extraction of Grid Corners*).

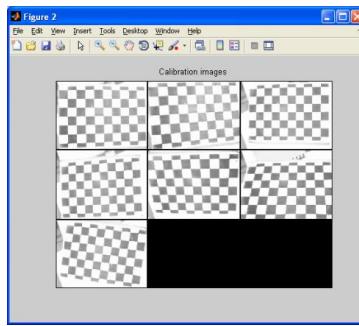


FIGURA 4.4 - Mosaico de imagens para câmera A.

FONTE: O AUTOR, 2015.

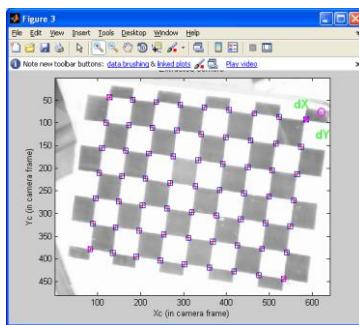


FIGURA 4.5 - Vértices dos quadrados da câmera A .

FONTE: O AUTOR, 2015.

Como resultado, o programa mostra os vértices estimados dos quadrados da grelha de calibração, na ausência da distorção conforme apresentado na FIG. 4.5. No caso em que a imagem tem elevada distorção, a estimativa destes vértices poderá ser fraca, ou seja, estarem afastados dos vértices reais dos quadrados na imagem. Assim, é possível introduzir um valor inicial para o coeficiente de distorção radial da lente, de forma a “ajudar” o *software* a encontrar os vértices dos quadrados na imagem.

Após uma primeira calibração da câmera, é possível realizar uma série de procedimentos que permitem analisar e estimar com maior precisão os parâmetros da câmera, por exemplo: a utilização da ferramenta de análise dos erros que permite observar o gráfico da correlação cruzada do erro de re-projeção; a visualização 3D gráfica dos parâmetros extrínsecos da câmera (várias posições e/ou orientações do plano de calibração relativamente à câmera ou vice-versa); a utilização, como parâmetros iniciais, da distorção da lente estimada na fase de calibração e da re-projeção dos pontos de calibração para uma nova e mais precisa extração desses pontos; adicionar e suprimir imagem de calibração; realizar novas calibrações escolhendo os parâmetros a aperfeiçoar ou optando por não os estimar; visualização gráfica do efeito da distorção ao longo da área da imagem, mostrada por mesmos níveis de distorção, permitindo comparar a importância da componente radial versus a componente tangencial da distorção, de forma a escolher o modelo de distorção apropriado.

Os resultados relevantes da calibração da câmera A (NIP-VC76P - Nipponic®), após a otimização apresentou os seguintes valores:

Focal Length: $fc = [932.61605 \ 928.58993] \pm [24.08841 \ 24.12205]$

Principal point: $cc = [447.31942 \ 305.88530] \pm [15.23891 \ 12.28776]$

Distortion: $kc = [0.05018 \ -0.57353 \ -0.00033 \ -0.00030 \ 0.00000] \pm [0.05064 \ 0.19566 \ 0.00325 \ 0.00586 \ 0.00000]$

Pixel error: $err = [0.85689 \ 0.95417]$

Para a segunda câmera foram obtidas outras sete imagens, nas mesmas condições anteriores, e a FIG.4.6 apresenta o mosaico destas imagens, enquanto que a FIG.4.7 os vértices dos quadrados obtidos.

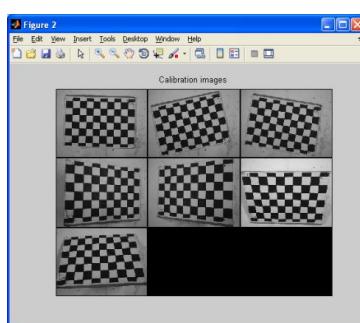


FIGURA 4.6 - Mosaico de imagens para a câmera B.

FONTE: O AUTOR, 2015.

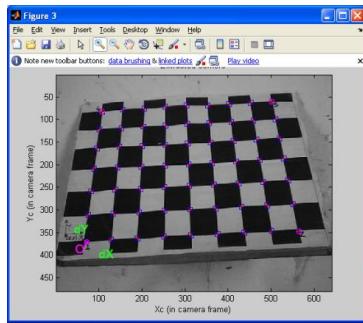


FIGURA 4.7 - Vértices dos quadrados para a câmera B.

FONTE: O AUTOR, 2015.

Também existem os resultados relevantes para a calibração da câmera B (*LifeCam VX-800 - Microsoft®*), após a otimização apresentou os seguintes valores:

$$\text{Focal Length: } fc = [687.45286 \ 686.88997] \pm [9.91374 \ 9.82313]$$

$$\text{Principal point: } cc = [319.34986 \ 235.00081] \pm [7.92026 \ 6.45537]$$

$$\begin{aligned} \text{Distortion: } kc = & [0.06663 \ -0.21094 \ 0.00270 \ -0.00198 \ 0.00000] \pm [0.05733 \ 0.31279 \\ & 0.00355 \ 0.00443 \ 0.00000] \end{aligned}$$

$$\text{Pixel error: } err = [0.59806 \ 0.60385]$$

Após a avaliação dos resultados obtidos, a *WebCam* escolhida para desenvolvimento do trabalho foi a B (*LifeCam VX-800 - Microsoft®*), pois apresentaram menores valores de distância focal, ponto principal e o mais importante o erro de *pixel*. E este equipamento apresenta um melhor custo benefício em relação ao outro modelo avaliado. Com isso, a escolha desta tem como considerações a vantagem em relação ao valor para aquisição, mas baseado na avaliação de critérios de desempenho.

4.2 Calibração do sistema

Para o funcionamento correto do sistema de visão de baixo custo neste projeto, foi necessário definir a posição e orientação da câmera, em função do referencial do objeto observado, a partir de uma imagem obtida sob os parâmetros intrínsecos da câmera e pelo modelo do objeto. Para esta etapa definiu-se o centro da câmera na imagem, conforme apresenta a FIG. 4.8, em que o centro desta coincide com o centro do sistema cartesiano estabelecido na imagem. Com este ponto definido, o *end effector* foi deslocado até ele de modo a determinar a posição em

coordenadas cartesianas. Este processo foi realizado em diversas posições sobre a área útil da mesa.



FIGURA 4.8 - Definição do centro da câmera.

FONTE: O AUTOR, 2015.

O método utilizado para identificar os vértices em objetos que contenham quinas em linha reta se baseia em realizar uma varredura e identificar alterações bruscas de direção, e a intercessão destas retas definem um ponto que é a quina do objeto em análise. De acordo da FIG. 4.9, os quatro pontos coordenados do objeto, foram determinados com auxílio de uma folha quadriculada, com 30 mm x 30 mm, em que a posição zero do eixo coordenado da imagem foi definida como o ponto inicial e referenciado a uma coordenada em *pixels*.

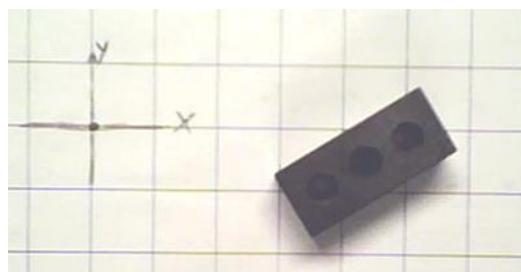


FIGURA 4.9 - Conversão pixels para milímetros .

FONTE: O AUTOR, 2015.

Após este procedimento, é possível realizar a calibração do sistema, com uso de uma peça padrão de referência. A TAB. 4.1 apresenta as dimensões desta peça, além de conter os valores obtidos por meio do algoritmo desenvolvido para detecção de bordas a partir das imagens capturadas pela câmera em diferentes alturas. Para a análise correta, deve-se realizar a conversão de unidades de *pixels* para milímetros, utilizando o fator que relaciona o comprimento da peça

em *pixel* com sua espessura em milímetros. Não se preocupou com a precisão com que a peça de referência foi construída, mas sim com o quanto mais próximo o sistema é capaz de identificar as características geométricas da mesma.

Tabela 4.1

Determinando o fator de conversão conforme imagens obtidas.

Comprimento x Largura x altura (mm)	Altura da câmera (mm)	Comprimento x Largura (pixel)	Calibração (pixel / mm)
50 x 22 x 22	235	157	7.14
	250	146	6.64
	275	132	6.00
	300	122	5.54
	350	105	4.77
	400	90	4.09
	450	79	3.59
	550	65	2.95

FONTE: O AUTOR, 2015.

A partir da espessura, calcula-se a proporcionalidade sobre a mesma região na imagem, este fator de escala é aplicado nas demais dimensões, com isso, o algoritmo recebe o valor em milímetros da peça. Estabelecida a relação entre a dimensão em milímetros da peça com seu comprimento em *pixels*, e a calibração do sistema, considera-se que a melhor posição da câmera é aquela que apresenta maior valor do fator de conversão em que a quantidade de *pixels* representa 1,0 mm, além de apresentar menor distorção na imagem. Para definir a posição da peça na imagem foram consideradas as distâncias das quinas até as respectivas bordas da imagem, e para determinar a orientação, e aplicando a EQUAÇÃO (4.3) é possível determinar o coeficiente angular.

$$\operatorname{tg}\alpha = m = \frac{y_2 - y_1}{x_2 - x_1} \quad (4.3)$$

Onde:

tg: função tangente

α: valor do ângulo do segmento de reta

m: coeficiente angular

x₁, *x₂*, *y₁* e *y₂*: coordenadas dos pontos do segmento de reta

Para garantir a repetibilidade do processo, é importante que o mesmo seja realizado sob as mesmas condições. Para isso, a peça é fotografada em diferentes orientações, mas com a mesma distância até a câmera, com uma resolução de 640 x 480 *pixels*. A FIG. 4.10 apresenta todas as imagens obtidas da peça sobre diversas posições e orientações, bem como a indicação de cada quina, que foram representadas por P1, P2, P3 e P4.

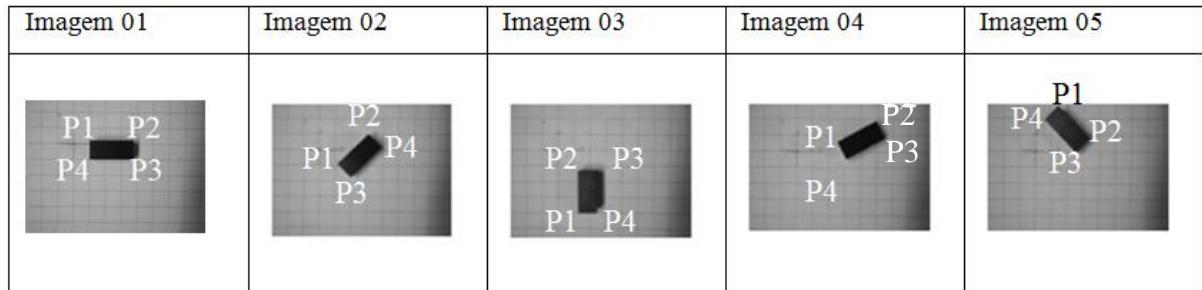


FIGURA 4.10 - Definição da orientação da peça em relação as coordenadas da imagem.

FONTE: O AUTOR, 2015.

As coordenadas dos pontos estão representadas na TAB. 4.2, bem como os valores dos coeficientes angulares calculados por meio da EQUAÇÃO (4.3) para dois segmentos de reta (P1-P2 e P2-P3, por exemplo), e os outros dois segmentos são obtidos por paralelismo.

Tabela 4.2
Coordenadas dos pontos.

Imagen	Segmento	Coordenadas em <i>pixels</i> (X, Y)	Coeficiente angular em relação ao eixo X
1	P1 → P2	(223,153); (378,153)	0.00°
	P2 → P3	(378,153); (378,222)	90.00°
2	P1 → P2	(231,218); (341,111)	43.20°
	P2 → P3	(341,111); (387,158)	136.00° (46.00 + 90.00)
3	P1 → P2	(236,393); (236,239)	90.00°
	P2 → P3	(236,239); (301,239)	0.00°
4	P1 → P2	(310,133); (447,60)	28.10°
	P2 → P3	(447,60); (478,120)	152.70° (62.70 + 90.00)
5	P1 → P2	(247,7); (355,120)	46.30°
	P2 → P3	(355,120); (306,166)	43.20°

FONTE: O AUTOR, 2015.

Agora de posse de um papel milimetrado, com espaçamentos de 5 mm, fez-se a verificação e validação do deslocamento preciso do TCP do manipulador mediante marcações na folha, conforme apresentado na FIG. 4.11, e como já apresentado na FIG. 3.2, o sistema coordenado do manipulador, a ponta da ferramenta desloca-se na direção Y no sentido positivo e na direção X no sentido negativo.

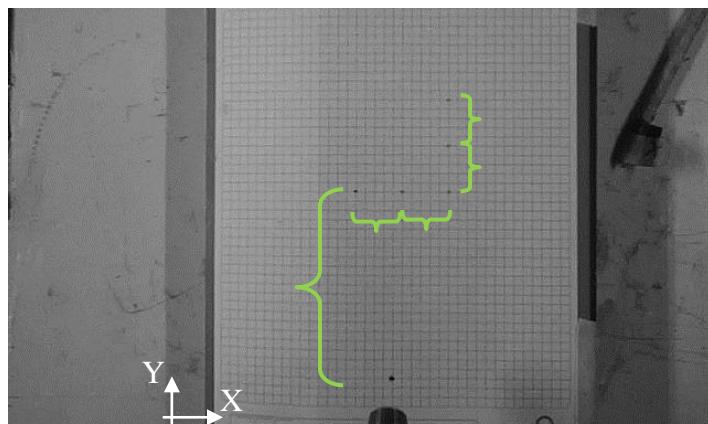


FIGURA 4.11 - Calibrando o manipulador.

FONTE: O AUTOR, 2015.

Esta etapa apresentou resultados satisfatórios, e neste caso, a calibração é realizada para a obtenção de uma lei matemática para compensação dos erros em um manipulador robótico como efeito de correção dos desvios de sua posição e orientação, ou também, é um procedimento para determinar valores reais das dimensões geométricas e características mecânicas de um robô. A necessidade da utilização de métodos de calibração é devido à incerteza na localização absoluta do manipulador em relação à sua localização original assumida para solucionar o problema de planejamento da trajetória.

A necessidade de calibrar existe pois, o robô está operando em um sistema de controle de malha aberta, sem sensores que determinem erros de posicionamento da ferramenta, com isso, as fontes de erros como: conhecimento impreciso de parâmetros cinemáticos, erros nos servos-controladores das juntas, elasticidade em elos e juntas, folga nas engrenagens, dentre outros, comprometem a acurácia do manipulador robótico. Aliado a calibração, utilizou-se uma maior discretização dos pontos para a geração da trajetória.

Após o processo de calibração, adotou-se um ponto que deve ser tocado pelo manipulador sempre que o sistema for iniciado para garantir que o processo de calibração continue validado, conforme destaque na FIG. 4.12.



FIGURA 4.12 - Calibrando o sistema.

FONTE: O AUTOR, 2015.

4.3 Análise da aplicabilidade dos algoritmos de visão computacional

Vale ressaltar mais uma vez que as formas geométricas das peças consideradas neste trabalho são todas conhecidas e medidas manualmente com equipamentos metrológicos devidamente aferidos. A partir de uma dada imagem, o pré-processamento para extrair a informação de contorno das peças e suas coordenadas está disposto em forma de diagrama na FIG. 4.13. Os *pixels* identificados delimitam o contorno do objeto, e cada posição destes, é processada e transformada em coordenadas cartesianas que serão enviadas ao manipulador robótico para que efetue a trajetória encontrada e realize a tarefa determinada.



FIGURA 4.13 - Diagrama de desenvolvimento do sistema proposto.

FONTE: O AUTOR, 2015.

Na aquisição da imagem a câmera está sempre na mesma posição, de forma a visualizar a mesa e sobre sua área útil identificar a localização do objeto de estudo. O primeiro passo no pré-processamento é binarizar a imagem, de tal modo que selecione na imagem a região de interesse e identifique o objeto que será segmentado do restante da imagem. Reduzindo também a quantidade de *pixels* a serem analisados, diminuindo desta forma o tempo de processamento.

Após a binarização, a imagem pode apresentar ruídos, o que pode comprometer a etapa posterior de extração de bordas, portanto filtros devem ser aplicados de modo a atenuar os níveis de ruídos, e este processo de minimização do ruído elimina regiões ou segmentos isolados de *pixels* assim como preenche pequenos espaços vazios dentro do contorno do objeto. Após esse pré-processamento utiliza-se o algoritmo *Bounding Box* para extração de bordas, o qual retorna os pontos sobre o contorno que são principais na distinção da forma do objeto. Ele pode ser qualquer retângulo ao longo de um eixo arbitrário, desde que o elemento esteja totalmente dentro da fronteira. Para aplicações práticas, os limites de uma *BB* são ajustados geralmente alinhado com as direções do sistema de coordenadas. Antes de aplicar o método proposto para uma imagem, é necessário pré-processá-la para tornar-se uma imagem binária. A imagem é então digitalizada horizontalmente para localizar quatro valores (topo, inferior, esquerda e direita), definindo-se assim uma caixa. Deixe "0" denotar um *pixel* de plano de fundo e "1" é um *pixel* do primeiro plano. Se durante o processo de varredura da imagem os *pixels* forem alterados de $0 \rightarrow 1$, significa que um limite a partir do fundo para o primeiro plano é atingido durante a digitalização. Se esta é a primeira vez para chegar ao elemento da imagem, o topo e a esquerda são as posições inscritas para o objeto. A posição topo é mantida para o objeto enquanto que a posição esquerda será ajustada selecionando o menor valor nas linhas de varredura subsequentes. Já para o caso em que os *pixels* são alterados de $1 \rightarrow 0$, significa que um limite do primeiro plano ao fundo é atingido durante a digitalização. O comprimento dos *pixels* de primeiro plano conectado desta linha de exploração pode então ser obtido. Se o comprimento é menor do que um limiar, a mesma será considerada como ruído ou oscilação e serão descartados. Caso contrário, a posição "correta" será definida e irá ser ajustada selecionando o maior valor nas linhas de exploração subsequentes. As figuras abaixo apresentam a falha e o sucesso no primeiro processo de segmentação utilizado:



a) Nível de Cinza Peça 1 b) Binarização Peça 1 c) Nível de Cinza Peça 2 d) Binarização Peça 2

FIGURA 4.14 - Testes para definição do limiar.

FONTE: O AUTOR, 2015.

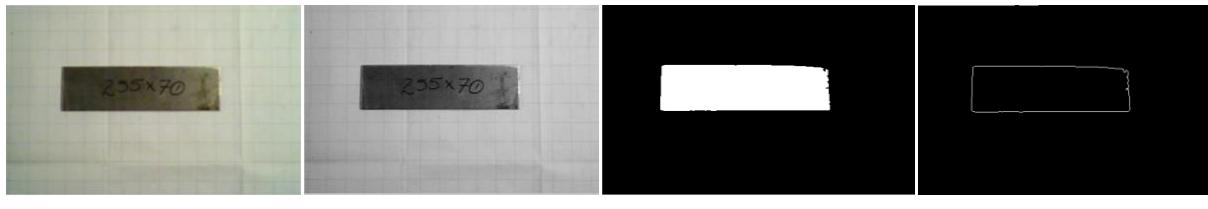


FIGURA 4.15 - Passo a passo para obtenção do contorno da peça.
FONTE: O AUTOR, 2015.

A FIG.4.14 e a FIG. 4.15 apresentaram resultados interessantes, mas para que o sistema se apresentasse menos sensível a pequenas variações do ambiente, fez-se uma alteração do algoritmo, e com isso uma nova definição do limiar foi necessária. A Transformada Hough foi testada de vários modos, alterando todos os seus parâmetros, mas não apresentou resultado satisfatório para a etapa que define valores dos *pixels* das quinas do objeto na imagem. A FIG. 4.16 que apresenta o melhor contorno obtido é possível perceber que os segmentos de reta não foram capazes de se intercederem. Então o uso desta ferramenta foi desconsiderado neste trabalho.

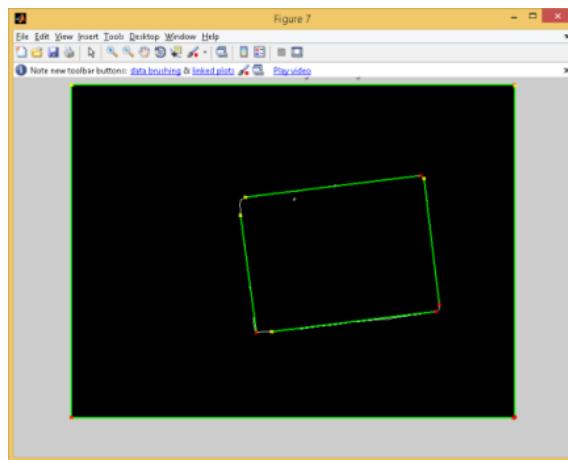


FIGURA 4.16 - Uso da Transformada HOUGH para obtenção do contorno da peça.
FONTE: O AUTOR, 2015.

Foi verificado que o valor do limiar de 150 quando a mesa estava nova, ou seja, sem manchas era satisfatório, já com o tempo e o desgaste da mesma o valor deve ser alterado para 140 no espaço de cores YUY2. A FIG. 4.17 apresenta uma sequencia de imagens obtidas com o valor do limiar a 130.

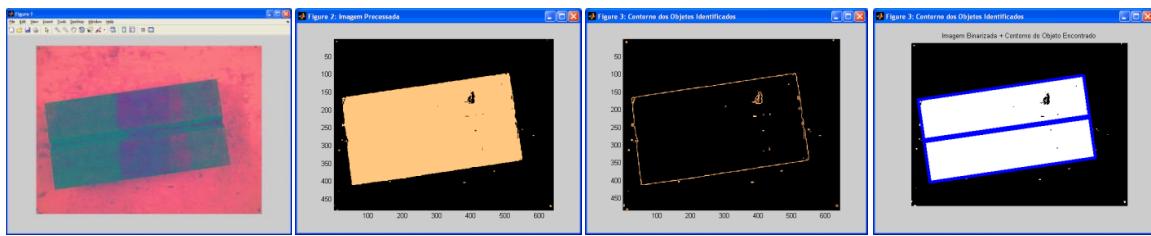


FIGURA 4.17 - Tratando imagem obtida como YUY2 e limiar de 130.

FONTE: O AUTOR, 2015.

Na FIG. 4.18 tem-se uma imagem obtida sobre as mesmas condições, alterando apenas o espaço de cores que neste caso foi o RGB com o limiar de 130.

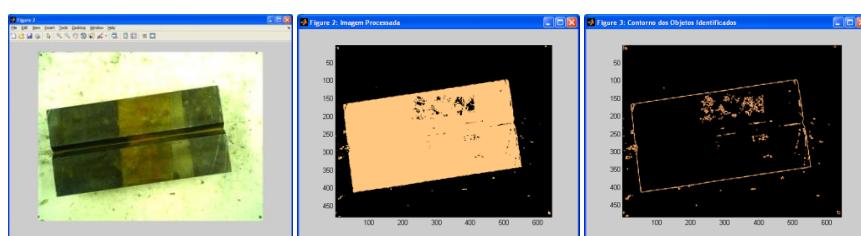


FIGURA 4.18 - Tratando imagem obtida como RGB e limiar de 130.

FONTE: O AUTOR, 2015.

Na FIG. 4.19 tem-se a imagem também obtida sobre as mesmas condições, alterando apenas o espaço de cores que neste caso foi o RGB com o limiar de 140.

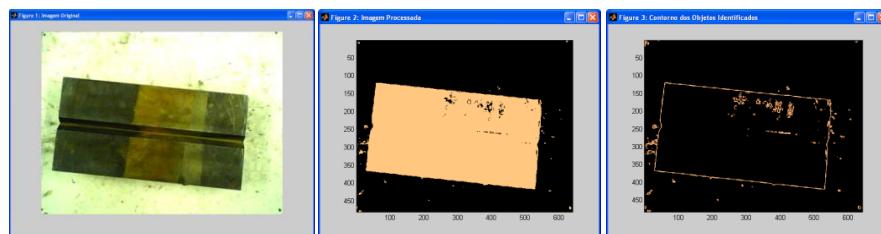


FIGURA 4.19 - Tratando imagem obtida como RGB e limiar de 140.

FONTE: O AUTOR, 2015.

Na FIG. 4.20 apresenta a imagem obtida com as mesmas condições anteriores, alterando o espaço de cores que neste caso foi o RGB com o limiar de 150.

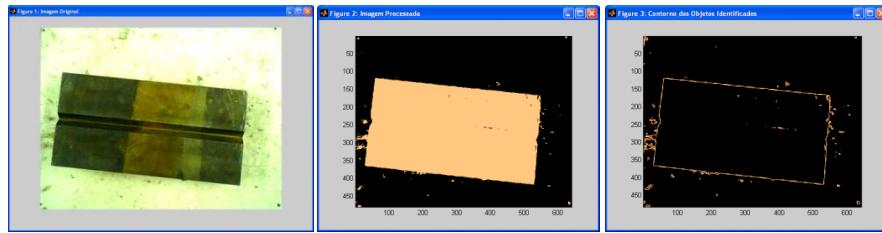


FIGURA 4.20 - Tratando imagem obtida como RGB e limiar de 150.

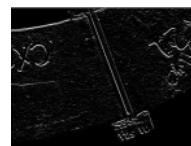
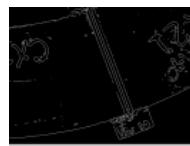
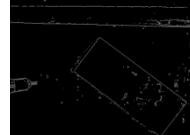
FONTE: O AUTOR, 2015.

No pré-processamento, os pontos equidistantes descrevem o contorno do objeto, e considerando-se agora a região de bordas como entradas, o processo de esqueletização consiste em reduzir a espessura de cada região de borda até a espessura mínima (um *pixel*) antes que o segmento se quebre. Após implementações e testes em várias imagens, variando parâmetros de entradas, de todos os filtros, fez-se uma avaliação sobre todos os filtros aplicados e podem-se apresentar quais apresentaram melhores resultados, e esta avaliação considera apenas os filtros Passa-Baixa e Passa-Alta.

Comparando o desempenho de todos os filtros do tipo Passa-Baixa, que suavizam a imagem original e eliminam ruídos, pode-se afirmar, segundo o que foi constatado experimentalmente, que o filtro que apresenta melhores resultados foi o Gaussiano, pois se consegue suavizar a imagem e ao mesmo tempo manter a imagem próxima da original. Em contrapartida, o filtro Passa-Baixa é o que apresenta piores resultados, pois torna a imagem muito desfocada.

Dos filtros do tipo Passa-Alta que detectam contornos dos objetos da imagem, o que se mostrou mais eficaz foi o filtro de CANNY, mas é de notar que o filtro de SOBEL consegue em certos tipos de imagem uma definição maior, perdendo para o de CANNY quando a imagem filtrada tem zonas com alta luminância. O filtro de SOBEL apresenta uma boa detecção de contornos, comparativamente com os outros filtros, os contornos estão bem mais salientados. Continua-se a verificar que o ruído originário pela reflexão da luz continua a comprometer a qualidade da imagem filtrada. Já o Filtro de CANNY consegue realçar os relevos com eficácia e apesar da presença de ruído proveniente da reflexão da luz é possível detectar onde está o contorno. Este filtro não elimina o ruído, mas consegue separar eficazmente os *pixels* que representam contorno dos que são indesejáveis. Mesmo com esta implicação, o filtro de SOBEL foi o escolhido. Diversos foram os testes realizados para definir qual o tipo de filtro a ser utilizado, e no QUADRO (4.2) estão representados apenas dos filtros de SOBEL e CANNY que apresentaram melhores resultados.

QUADRO 4.2.
Comparação dos resultados na aplicação dos filtros.

Imagen Original	CANNY	SOBEL
		
		

FONTE: O AUTOR, 2015.

Para as imagens apresentadas no QUADRO (4.2) observa-se que para a correta identificação das bordas, são necessários valores de limiares para histerese de gradiente muito baixos, o que leva a um grande número de bordas identificadas, devido a imperfeições na peça e a ruídos.

Com a informação da posição dos *pixels* correspondentes ao ponto inicial e final do trajeto a ser percorrido conforme FIG. 4.21, extrai-se um erro (em *pixels*) com relação ao centro da imagem e a esse erro é associado um ganho proporcional que é implementado ao valor da coordenada em milímetros para o manipulador.

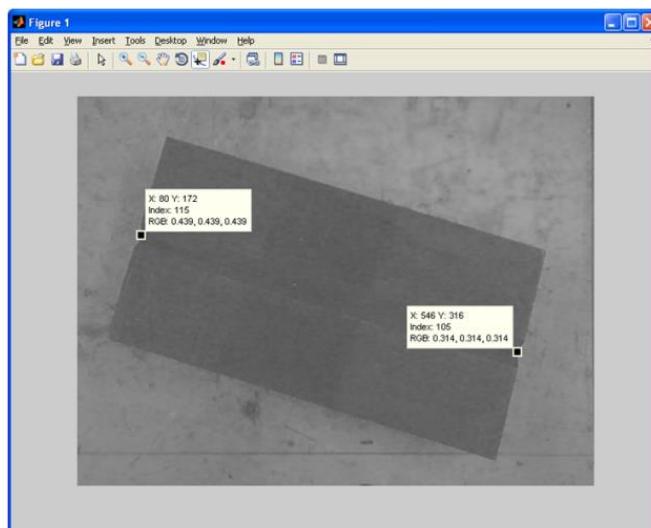


FIGURA 4.21 - Definindo a trajetória do manipulador.

FONTE: O AUTOR, 2015.

Após esta etapa, pensou-se como seria realizada a tarefa do manipulador. Conforme a FIG. 4.22 a identificação do vale na imagem determinaria em um processo de soldagem do tipo de enchimento, onde são realizados sucessivos cordões de solda. O que já é um desafio grande, uma vez que, o manipulador identificando esta região, deveria realizar seu trajeto perfeitamente em linha reta.



FIGURA 4.22 - Processo de enchimento de chanfro.

FONTE: O AUTOR, 2015.

4.4 Funcionamento do programa

Para a obtenção das imagens, processamento e geração das trajetórias, tornou-se necessário a integração entre o *MatLab*[®] e o *Mach3*[®], de modo que o robô programe a posição inicial e final da ferramenta em um movimento do tipo linear, juntamente com os valores de velocidade e altura da ferramenta. A FIG. 4.23 apresenta o programa em *MatLab*[®] para o experimento em questão.

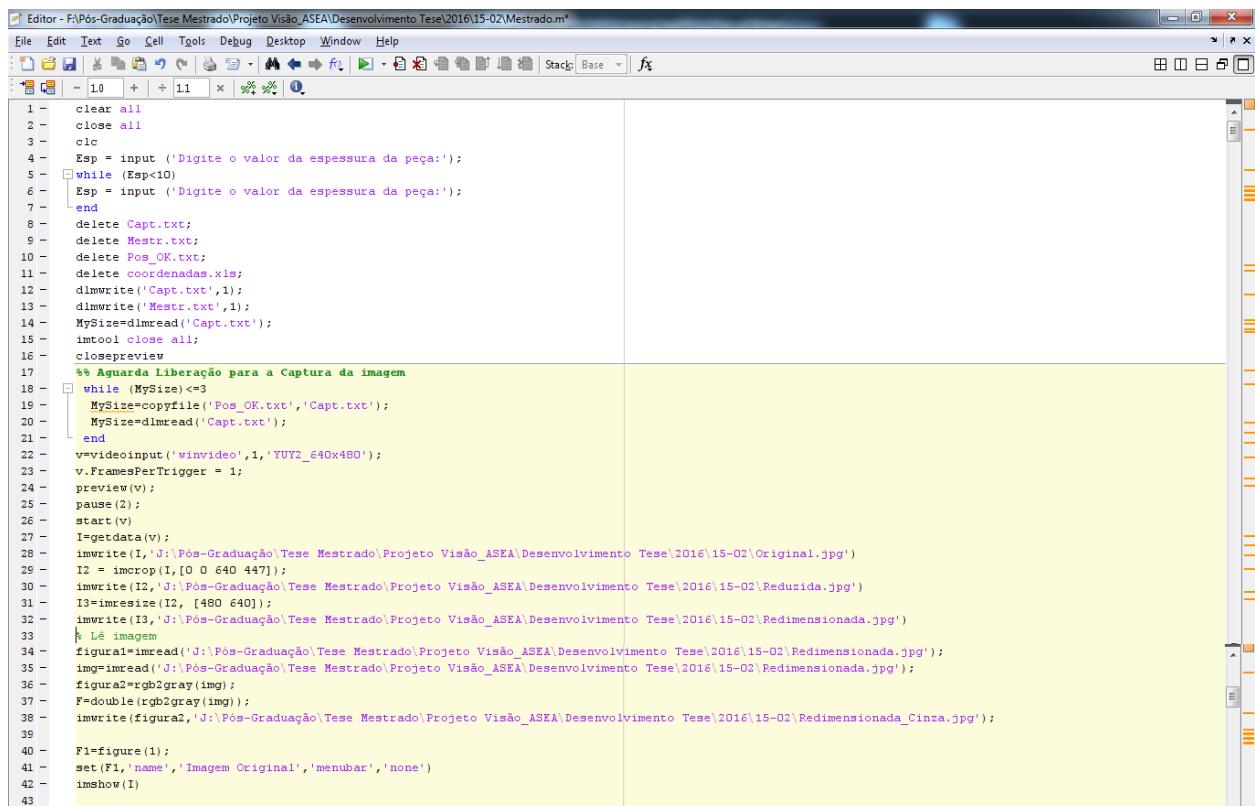
Conforme pode ser visto na FIG. 4.23, da linha 1 até a linha 16 é feita uma limpeza dos cálculos e arquivos anteriores para evitar possíveis *bugs* no algoritmo, nesta etapa é também definida a espessura da chapa que está sendo utilizada, para que seja mantido constante a DBCP.

Neste momento é iniciado o *Visual Basic Script* do *Mach3*[®], conforme apresentado na FIG. 4.24, sendo que nas 8 primeira linhas, é iniciado o movimento do manipulador, tocando no ponto de calibração, apresentado na FIG. 4.12, e em seguida parando na posição de captura de imagem. Já da linha 8 até a linha 14 o programa gerenciador dos movimentos informa ao *MatLab*[®] que o manipulador já está na posição de captura de imagem.

Voltando na FIG. 4.23, da linha 18 até a linha 21, após a liberação do processo da captura de imagem, o programa inicia a rotina chamando a *WebCam*, que está conectada ao computador via

cabo *USB*, e a mesma já apresenta a imagem a ser capturada na tela do computador, é aguardado um tempo para a estabilização da iluminação, e posteriormente é obtida a imagem, conforme apresentado na FIG. 4.25.

Já para o processamento de imagem a primeira operação é a redução desta para eliminar o bocal da tocha conforme FIG. 4.26. Após o recorte do bocal na imagem a mesma é redimensionada e convertida para tom de cinza, de acordo com as linhas 28 até a linha 38. Para passar a imagem para uma matriz de duas dimensões de tamanho dinâmico, o comando: *img = imread('caminho/nome.extensão');* torna-se necessário uma vez que, “*img*” é a matriz que contém informações da imagem, e “*imread*” é a função que passa a imagem devidamente tratada para o programa, recebendo como parâmetro o diretório onde se encontra a imagem desejada.



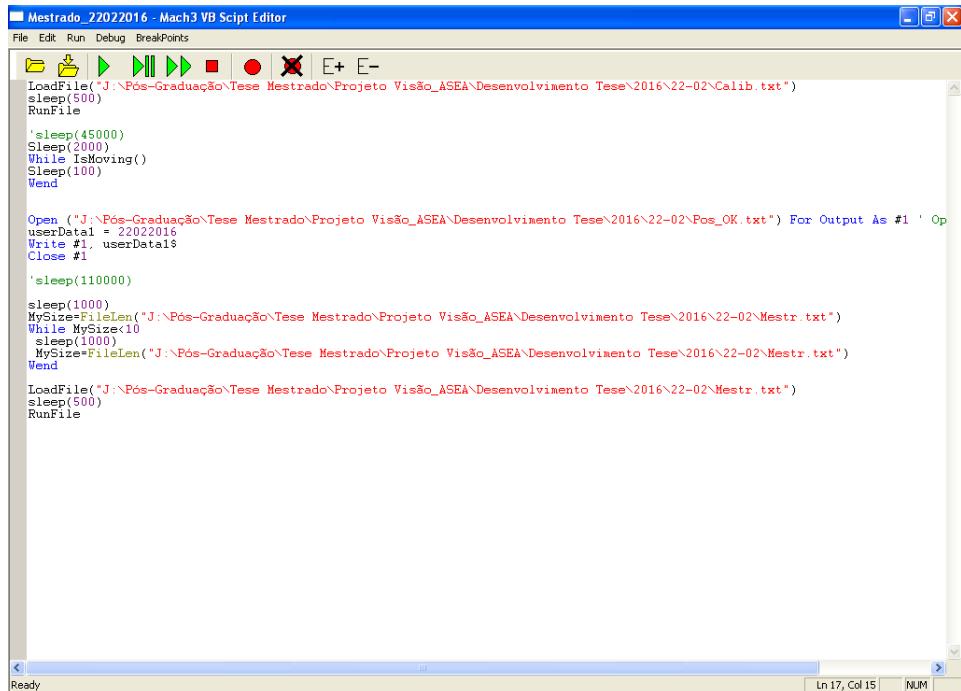
```

Editor - F:\Pós-Graduação\Tese Mestrado\Projeto Visão_ASEA\Desenvolvimento Tese\2016\15-02\Mestrado.m
File Edit Text Go Cell Tools Debug Desktop Window Help
Stack: Base fx
- 10 + 11 x
1 - clear all
2 - close all
3 - cic
4 - Esp = input ('Digite o valor da espessura da peça:');
5 - while (Esp<10)
6 -   Esp = input ('Digite o valor da espessura da peça:');
7 - end
8 - delete Capt.txt;
9 - delete Mestr.txt;
10 - delete Pos_OK.txt;
11 - delete coordenadas.xls;
12 - dlmwrite ('Capt.txt',1);
13 - dlmwrite ('Mestr.txt',1);
14 - MySize=dlmread ('Capt.txt');
15 - imtool close all;
16 - closepreview
17 - % Aguarda Liberação para a Captura da imagem
18 - while (MySize)<3
19 -   MySize=copyfile ('Pos_OK.txt','Capt.txt');
20 -   MySize=imread ('Capt.txt');
21 - end
22 - v=videoinput ('winvideo',1,'YUY2_640x480');
23 - v.FramesPerTrigger = 1;
24 - preview(v);
25 - pause(2);
26 - start(v);
27 - I=getdata(v);
28 - imwrite(I,'J:\Pós-Graduação\Tese Mestrado\Projeto Visão_ASEA\Desenvolvimento Tese\2016\15-02\Original.jpg')
29 - I2 = imcrop(I,[0 0 640 447]);
30 - imwrite(I2,'J:\Pós-Graduação\Tese Mestrado\Projeto Visão_ASEA\Desenvolvimento Tese\2016\15-02\Reduzida.jpg')
31 - I3=imresize(I2, [480 640]);
32 - imwrite(I3,'J:\Pós-Graduação\Tese Mestrado\Projeto Visão_ASEA\Desenvolvimento Tese\2016\15-02\Redimensionada.jpg')
33 - % Lê imagem
34 - figura1=imread ('J:\Pós-Graduação\Tese Mestrado\Projeto Visão_ASEA\Desenvolvimento Tese\2016\15-02\Redimensionada.jpg');
35 - img=imread ('J:\Pós-Graduação\Tese Mestrado\Projeto Visão_ASEA\Desenvolvimento Tese\2016\15-02\Redimensionada.jpg');
36 - figura2=rgb2gray(img);
37 - F=double(rgb2gray(img));
38 - imwrite (figura2,'J:\Pós-Graduação\Tese Mestrado\Projeto Visão_ASEA\Desenvolvimento Tese\2016\15-02\Redimensionada_Cinza.jpg');
39 -
40 - F1=figure(1);
41 - set(F1,'name','Imagen Original','menubar','none')
42 - imshow(I)
43

```

FIGURA 4.23 - Processo de Obtenção e processamento da imagem.

FONTE: O AUTOR, 2015.



```

Mestrado_22022016 - Mach3 VB Script Editor
File Edit Run Debug BreakPoints
LoadFile("J:\Pós-Graduação\Tese Mestrado\Projeto Visão_ASEA\Desenvolvimento Tese\2016\22-02\Calib.txt")
sleep(500)
RunFile

'sleep(45000)
Sleep(2000)
While IsMoving()
Sleep(100)
Wend

Open ("J:\Pós-Graduação\Tese Mestrado\Projeto Visão_ASEA\Desenvolvimento Tese\2016\22-02\Pos_OK.txt") For Output As #1 ' Output
userData1 = 2202016
Write #1, userData1$
Close #1

'sleep(110000)

sleep(1000)
MySize=FileLen("J:\Pós-Graduação\Tese Mestrado\Projeto Visão_ASEA\Desenvolvimento Tese\2016\22-02\Mestr.txt")
While MySize<10
sleep(1000)
MySize=FileLen("J:\Pós-Graduação\Tese Mestrado\Projeto Visão_ASEA\Desenvolvimento Tese\2016\22-02\Mestr.txt")
Wend

LoadFile("J:\Pós-Graduação\Tese Mestrado\Projeto Visão_ASEA\Desenvolvimento Tese\2016\22-02\Mestr.txt")
sleep(500)
RunFile

```

FIGURA 4.24 - Programa gerenciador dos movimentos.

FONTE: O AUTOR, 2015.



FIGURA 4.25- Imagem do ambiente a ser capturada.

FONTE: O AUTOR, 2016.

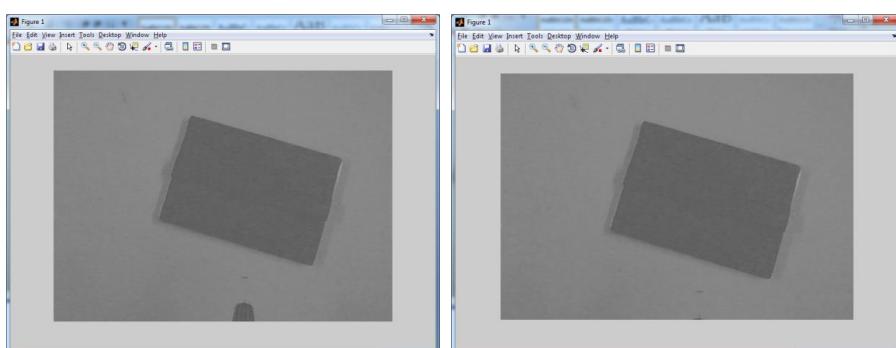


FIGURA 4.26 - Imagens com e sem o bocal da tocha.

FONTE: O AUTOR, 2016.

Para modificar as cores da imagem colorida para escala de cinza, foi utilizado o seguinte comando: `figura2=rgb2gray(img);` sendo que "figura2" é a matriz que vai guardar informações da imagem com cores em escala de cinza, e “`rgb2gray(img);`” recebe uma matriz contendo informações originais que são modificados e resulta em uma imagem em escala de cinza.

Para a imagem apresentada na FIG. 4.26, foi necessário realizar o redimensionamento da mesma devido ao aparecimento do bocal da tocha. Tal procedimento consiste no seguinte: `imcrop` - corta a imagem, solicitando que seja definido uma sub-imagem retangular da figura atual, `I2 = imcrop(I,[0 0 640 447]);` sendo 640x447 a região a qual a nova imagem foi retirada. E para que a nova imagem permaneça com 640x480 é necessário redimensioná-la, utilizando `imresize` – “`I3=imresize(I2, [480 640]);`”;

Na FIG. 4.27 é apresentada uma operação de remoção de valores de RG da imagem RGB inicial, e em seguida atribuindo a cor cobre ao objeto encontrado com esta operação.

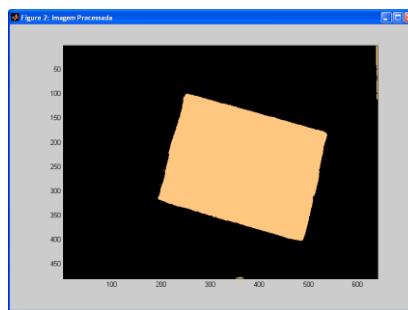


FIGURA 4.27 - Esqueletização do objeto.

FONTE: O AUTOR, 2016.

Já na FIG. 4.28 é apresentada a esqueletização do objeto encontrado, mediante aplicação de convoluções e do filtro de SOBEL.

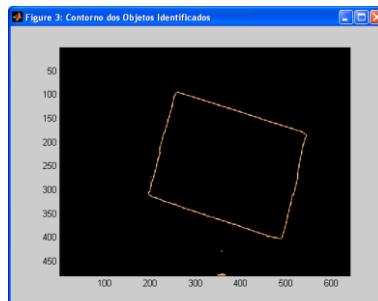


FIGURA 4.28 - Esqueletização do objeto.

FONTE: O AUTOR, 2016.

Com a realização da binarização do objeto encontrado na imagem, a FIG. 4.29 apresenta o resultado desta operação.

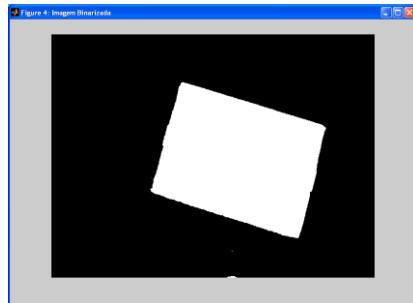
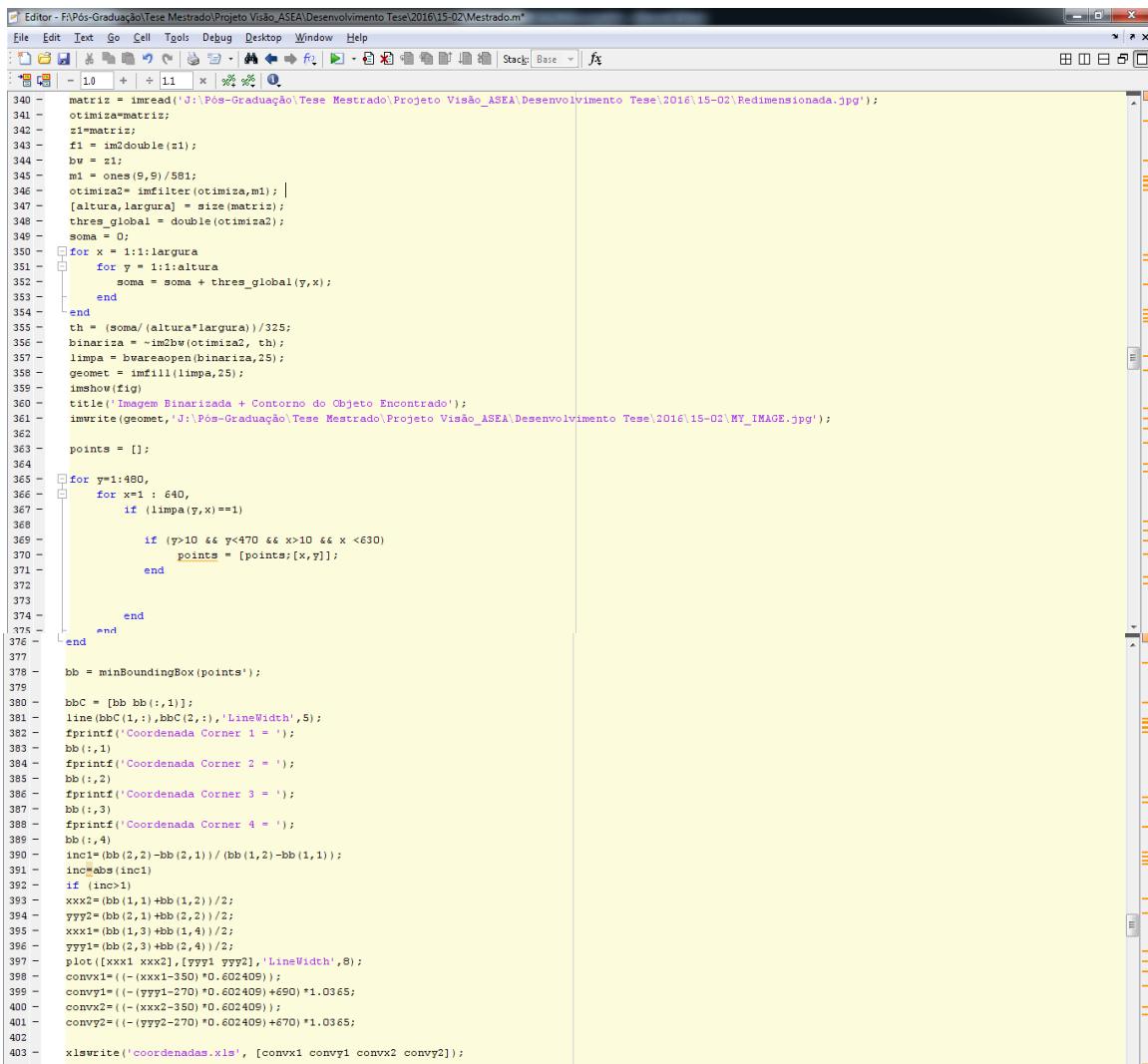


FIGURA 4.29 - Binarização do objeto encontrado.

FONTE: O AUTOR, 2016.

Como resultados da segmentação tem-se as centroides das regiões encontradas na imagem, acompanhados da quantidade de *pixels* presentes em cada região, além das coordenadas do menor retângulo envolvente que inclui a região *Bounding Box*. Com o uso de filtros de SOBEL, Gaussianos, convoluções, e limiares, determinam se a quantidade de objetos obtidos na imagem extraída da cena, tornando a imagem totalmente binarizada. Após este processo aplica-se o algoritmo *Bounding Box* para detecção dos pontos coordenados extremos do objeto, conforme apresentado na FIG 4.30. É possível observar ainda, que para o valor da variável "inc" sendo maior do que 1, este será tratado de uma determinada maneira, o que implica na orientação da peça sobre a mesa.

Após esta etapa é gerado o contorno no objeto encontrado, apresentado na FIG. 4.31, e a cada quina é atribuído um valor em *pixel* devido à aplicação do *Bounding Box* e estes valores são convertidos para milímetros e então são armazenados em uma planilha, que serão lidos posteriormente para gerar a trajetória do manipulador e serão convertidos em código G que será executado no *Mach3®* para o deslocamento do TCP ao longo de sua trajetória para realizar a tarefa programada.



```

Editor - F:\Pós-Graduação\Projeto Mestrado\Projeto Visão_ASEA\Desenvolvimento Tese\2016\15-02\Mestrado.m
File Edit Text Go Cell Tools Debug Desktop Window Help
File Edit Text Go Cell Tools Debug Desktop Window Help
340 - matriz = imread('J:\Pós-Graduação\Projeto Mestrado\Projeto Visão_ASEA\Desenvolvimento Tese\2016\15-02\Redimensionada.jpg');
341 - optimiza=matriz;
342 - z1=matriz;
343 - f1 = im2double(z1);
344 - bw = z1;
345 - m1 = ones(9,9)/551;
346 - optimiza2= imfilter(optimiza,m1); |
347 - [altura,largura] = size(matriz);
348 - thres_global = double(optimiza2);
349 - soma = 0;
350 - for x = 1:1:largura
351 -     for y = 1:1:altura
352 -         soma = soma + thres_global(y,x);
353 -     end
354 - end
355 - th = (soma/(altura*largura))/325;
356 - binariza = ~im2bw(optimiza, th);
357 - limpaa = bwareaopen(binariza,25);
358 - geomet = imfill(limpaa,25);
359 - imshow(fig)
360 - title('Imagen Binarizada + Contorno do Objeto Encontrado');
361 - imwrite(geomet,'J:\Pós-Graduação\Projeto Mestrado\Projeto Visão_ASEA\Desenvolvimento Tese\2016\15-02\MY_IMAGE.jpg');
362 -
363 - points = [];
364 -
365 - for y=1:480,
366 -     for x=1 : 640,
367 -         if (limpaa(y,x)==1)
368 -
369 -             if (y>10 && y<470 && x>10 && x <630)
370 -                 points = [points;x,y];
371 -             end
372 -
373 -         end
374 -     end
375 - end
376 -
377 - bb = minBoundingBox(points');
378 -
379 - bbC = [bb bb(:,1)];
380 - line(bbC(1,:),bbC(2,:),'LineWidth',5);
381 - fprintf('Coordenada Corner 1 = ');
382 - bb(:,1)
383 - fprintf('Coordenada Corner 2 = ');
384 - bb(:,2)
385 - fprintf('Coordenada Corner 3 = ');
386 - bb(:,3)
387 - fprintf('Coordenada Corner 4 = ');
388 - bb(:,4)
389 - inc1=(bb(2,2)-bb(2,1))/(bb(1,2)-bb(1,1));
390 - inc=abs(inc1)
391 - if (inc>1)
392 -     xxz1=(bb(1,1)+bb(1,2))/2;
393 -     yyz1=(bb(2,1)+bb(2,2))/2;
394 -     xxz2=(bb(1,3)+bb(1,4))/2;
395 -     yyz2=(bb(2,3)+bb(2,4))/2;
396 -     plot([xxz1 xxz2],[yyz1 yyz2],'LineWidth',8);
397 -     convx1=((-(xxx1-350)*0.602409);
398 -     convy1=((-(yyy1-270)*0.602409)+690)*1.0365;
399 -     convx2=((-(xxx2-350)*0.602409);
400 -     convy2=((-(yyy2-270)*0.602409)+670)*1.0365;
401 -     convy2=((-(yyy2-270)*0.602409)+670)*1.0365;
402 -
403 - xlswrite('coordenadas.xls', [convx1 convy1 convx2 convy2]);

```

FIGURA 4.30 - Algoritmo desenvolvido.

FONTE: O AUTOR, 2015.

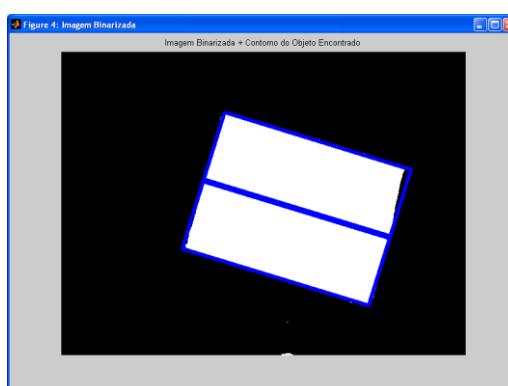


FIGURA 4.31 - Obtenção do objeto na imagem e suas coordenadas em milímetros.

FONTE: O AUTOR, 2016.

Na FIG. 4.32, tem-se a geração da trajetória em *MatLab*[®], sendo que a matriz identidade apresentada na linha 404 torna-se necessária para informar a distância da flange com a ferramenta montada, ou seja, as dimensões da tocha e DBCP. Posteriormente há a configuração da velocidade, na linha 408, a ser desenvolvida pelo TCP do robô. Esta velocidade está limitada a 55 mm/s, valores maiores que isso não pode ser atingida devido a limitações do *driver* que envia pulsos aos motores. Já o passo apresentado na linha 409, é necessário para a discretização do sinal, sendo que quanto menor, maior será a discretização, e menor será o erro na trajetória do robô, tornando o movimento mais contínuo e suave. E a variável “deltat”, da linha 410, representa o intervalo de tempo necessário para que o TCP movimente entre dois pontos.

São lidos os valores, nas linhas 411 a linha 414, das coordenadas da planilha que foi gerada, e estes valores ajudaram nos cálculos dos passos independentes dos eixos X e Y. Posteriormente é criado um arquivo temporário com o nome de "Mestrado1.txt", linha 421, será nele que serão armazenados o código G que será lido pelo *Mach3*[®] e que proporcionará o movimento do manipulador. Mas para isso, são necessárias as funções da cinemática inversa do manipulador e cinemática inversa de atuadores que informam o número de voltas que o motor terá que movimentar para ferramenta chegar até o destino programado, conforme linhas 426 até a linha 461. E então é gerado o código G, como apresentado na FIG. 4.33, onde as variáveis X, Y, Z, A e B representam as juntas 1, 2, 3, 4 e 5, respectivamente, e o movimento do *end effector* é interpolado, ou seja, a tocha move junta a junta.

```

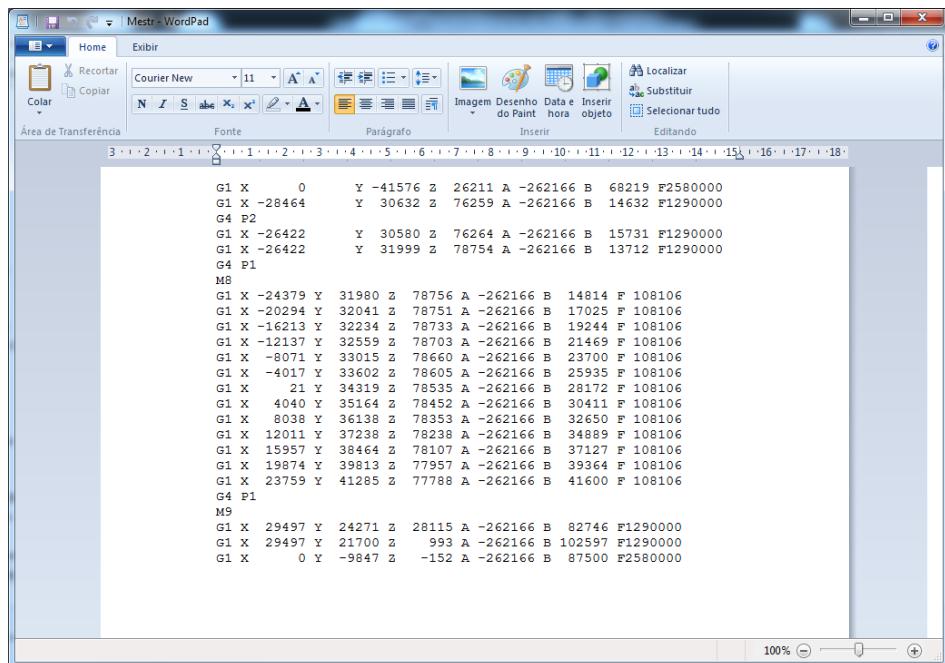
Editor - F:\Pos-Graduação\Tese Mestrado\Projeto Visão_ASEA\Desenvolvimento Tese\2016\15-02\Mestrado.m
File Edit Text Go Cell Tools Debug Desktop Window Help
Stack: Base | fx
404 - TST=[ 1 0 0 0;
405 - 0 1 0 0;
406 - 0 0 1 370;
407 - 0 0 0 1];
408 - vel=10;
409 - passo=10;
410 - deltat=abs(passo/vel);
411 - X1 = xlsread('coordenadas.xls',1,'A1');
412 - Y1 = xlsread('coordenadas.xls',1,'B1');
413 - X2 = xlsread('coordenadas.xls',1,'C1');
414 - Y2 = xlsread('coordenadas.xls',1,'D1');
415 - npassos = sqrt((X2-X1)^2+(Y2-Y1)^2);
416 - if npassos>30
417 - npassos= sqrt(sqrt((X2-X1)^2+(Y2-Y1)^2));
418 - end
419 - PassoX=(X2-X1)/npassos;
420 - PassoY=(Y2-Y1)/npassos;
421 - fid = fopen('Mestrado1.txt','w');
422 - Z1=445+Esp;
423 - Z2=Z1-8;
424 - %Posição encontrada
425 - TOT_ini=[1 0 0 X1;0 -1 0 Y1;0 0 -1 Z1;0 0 0 1];
426 - teta=cinvinv(TOT_ini,TST);
427 - A=cininvatu(teta)*8196/(2*pi);
428 - fprintf(fid,'G1 X %6.0f',A(1));
429 - fprintf(fid,' Y %6.0f',A(2));
430 - fprintf(fid,' Z %6.0f',A(3));
431 - fprintf(fid,' A %6.0f',A(4));
432 - fprintf(fid,' B %6.0f F1290000\n',A(5));
433 - Aant=A;
434 - fprintf(fid,'G4 P2\n');
435 - TOT_ini=[1 0 0 X1+(PassoX/2);0 -1 0 Y1+(PassoY/2);0 0 -1 Z2;0 0 0 1];
436 - teta=cinvinv(TOT_ini,TST);
437 - A=cininvatu(teta)*8196/(2*pi);
438 - fprintf(fid,'G1 X %6.0f',A(1));
439 - fprintf(fid,' Y %6.0f',A(2));
440 - fprintf(fid,' Z %6.0f',A(3));
441 - fprintf(fid,' A %6.0f',A(4));
442 - fprintf(fid,' B %6.0f F1290000\n',A(5));
443 - Aant=A;
444 - fprintf(fid,'G4 P1\n');
445 - fprintf(fid,'M8\n');
446 - for p=2:(npassos+1),
447 - x(p)=X1+PassoX*(p-1);
448 - y(p)=Y1+PassoY*(p-1);
449 - TOT=[1 0 0 x(p);0 -1 0 y(p);0 0 -1 Z2;0 0 0 1];
450 - teta=cinvinv(TOT,TST);
451 - A=cininvatu(teta)*8196/(2*pi);
452 - fprintf(fid,'G1 X %6.0f',A(1));
453 - fprintf(fid,' Y %6.0f',A(2));
454 - fprintf(fid,' Z %6.0f',A(3));
455 - fprintf(fid,' A %6.0f',A(4));
456 - fprintf(fid,' B %6.0f',A(5));
457 - K=60;
458 - feedrate=sqrt((A-Aant)*(A-Aant))/deltat*K;
459 - Aant=A;
460 - fprintf(fid,' F %6.0f\n',108106);
461 - end
462 -
463 - fprintf(fid,'G4 P1\n');
464 - fprintf(fid,'M9\n');
465 - fclose(fid);
466 - pause(5);
467 - movefile('Mestrado1.txt','Mestr.txt')
468

```

FIGURA 4.32 - Programa de geração de trajetória em *MatLab®*

FONTE: O AUTOR, 2015.

Na linha 466 até 468, o arquivo é fechado, é aguardado 5 segundos, e então o arquivo temporário é renomeado "Mestr.txt". E então conforme a FIG. 4.33, o *Visual Basic Script* do *Mach3®*, que estava aguardando, recebe um "OK" do *MatLab®* para retomar o movimento do manipulador já para as coordenadas finais para execução da tarefa, com destaque para as funções M8 e M9 do código G, que servem para habilitar e desabilitar o relé que fará o gatilho acionar e realizar a alimentação do arame para o processo de soldagem.



The screenshot shows a Microsoft WordPad window titled "Mestr - WordPad". The menu bar includes "Home" and "Exibir". The ribbon tabs include "Fonte", "Parágrafo", "Inserir", and "Editando". The status bar at the bottom shows "100%" and zoom controls. The main content area contains a large block of G-code:

```

G1 X 0 Y -41576 Z 26211 A -262166 B 68219 F2580000
G1 X -28464 Y 30632 Z 76259 A -262166 B 14632 F1290000
G4 P2
G1 X -26422 Y 30580 Z 76264 A -262166 B 15731 F1290000
G1 X -26422 Y 31999 Z 78754 A -262166 B 13712 F1290000
G4 P1
M8
G1 X -24379 Y 31980 Z 78756 A -262166 B 14814 F 108106
G1 X -20294 Y 32041 Z 78751 A -262166 B 17025 F 108106
G1 X -16213 Y 32234 Z 78733 A -262166 B 19244 F 108106
G1 X -12137 Y 32559 Z 78703 A -262166 B 21469 F 108106
G1 X -8071 Y 33015 Z 78660 A -262166 B 23700 F 108106
G1 X -4017 Y 33602 Z 78605 A -262166 B 25935 F 108106
G1 X 21 Y 34319 Z 78535 A -262166 B 28172 F 108106
G1 X 4040 Y 35164 Z 78452 A -262166 B 30411 F 108106
G1 X 8038 Y 36138 Z 78353 A -262166 B 32650 F 108106
G1 X 12011 Y 37238 Z 78238 A -262166 B 34889 F 108106
G1 X 15957 Y 38464 Z 78107 A -262166 B 37127 F 108106
G1 X 19874 Y 39813 Z 77957 A -262166 B 39364 F 108106
G1 X 23759 Y 41285 Z 77788 A -262166 B 41600 F 108106
G4 P1
M9
G1 X 29497 Y 24271 Z 28115 A -262166 B 82746 F1290000
G1 X 29497 Y 21700 Z 993 A -262166 B 102597 F1290000
G1 X 0 Y -9847 Z -152 A -262166 B 87500 F2580000

```

FIGURA 4.33 - Código G gerado através do *MatLab*®

FONTE: O AUTOR, 2015.

4.5 Realização dos ensaios

Inicialmente de modo bem rústico a câmera esteve afixada ao elemento final do manipulador conforme apresentado na FIG. 4.34. Mas conforme os experimentos foram sendo realizados, percebeu-se que não havia precisão no posicionamento da câmera que se deslocava e até mesmo vibrava devido à instabilidade do acrílico ao qual a mesma esta montada.



FIGURA 4.34 - Configuração inicial do sistema.

FONTE: O AUTOR, 2013.

A não utilização de filtros adequados proporcionaram imagens completamente inviáveis para o proposto. A FIG. 4.35 apresenta um exemplo do que foi exposto. Além disso, percebeu-se a necessidade de pintar a mesa para melhorar o processo de obtenção e processamento da imagem.

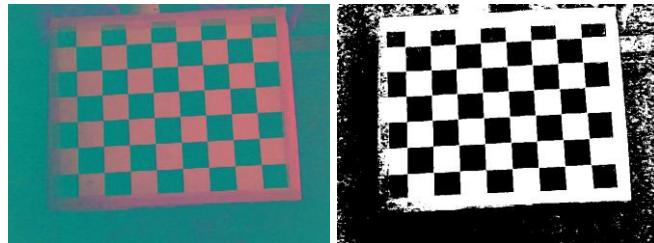


FIGURA 4.35 - Resultado da falta de um filtro adequado.

FONTE: O AUTOR, 2013.

Pensando em melhoria do processo de obtenção de imagem de modo a eliminar sombras e correção da iluminação, testou-se uma nova configuração para montagem da câmera, sendo esta fixa à mesa, conforme FIG. 4.36. Agora, a câmera está devidamente fixada em estrutura rígida e foram adicionados dois conjuntos de iluminação a 45° .

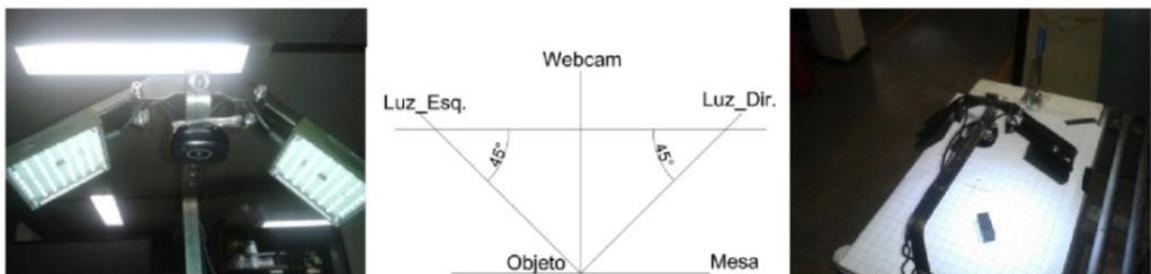


FIGURA 4.36 - Sistema de visão e iluminação.

FONTE: O AUTOR, 2013.

Mas o sistema proposto na FIG. 4.36 apresentou grande limitação quanto à área útil a ser avaliada na mesa por estar muito próxima da mesma, com isso, poucas seriam as posições em que a peça poderia estar para ser capturada pela câmera, o que implica diretamente no dimensional da peça. Para correção deste problema, todo o sistema de visão foi montado em uma chapa rígida no TCP do manipulador, como apresentado na FIG. 4.37.



FIGURA 4.37 - Sistema de visão e iluminação atual.

FONTE: O AUTOR, 2013.

Ao contrário do que foi apresentado na FIG. 4.29, agora a mesa foi pintada em duas cores, sendo a cor branca que indica a área útil de captura de imagem, onde é possível mover a peça, e a outra parte não útil foi pintada na cor preta, e vale ressaltar que para esta situação a iluminação externa não se fez necessária, pois a parte mais clara foi pintada com tinta branca fosca para evitar brilho excessivo e produção de ruído na imagem, e a peça considerada neste trabalho, que foi obtida através do processo de usinagem, sendo necessário sua oxidação para que ocorra um contraste em relação à mesa. A FIG. 4.38 apresenta tal condição.

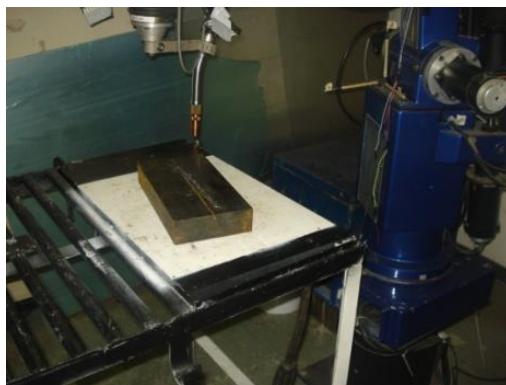


FIGURA 4.38 - Mesa e sua área útil.

FONTE: O AUTOR, 2014.

Com os pontos calibrados e definição dos valores de conversão de *pixels* para milímetros, os testes foram iniciados e obtidos os seguintes dados apresentados na FIG. 4.39. Estes pontos são primordiais para que o manipulador possa realizar sua tarefa.

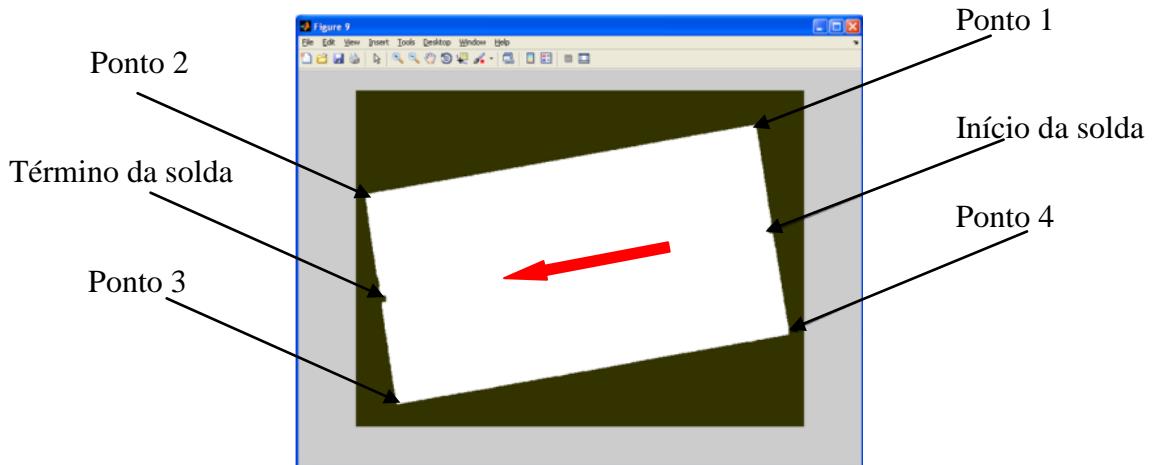


FIGURA 4.39 - Extração da peça e definição dos pontos.

FONTE: O AUTOR, 2015.

Na FIG. 4.40 estão representadas as quinas obtidas, para a mesma imagem é apresentado o valor da coordenada do ponto 1 em *pixels* que é (563.2298; 100.8796). Este valor posteriormente é convertido em milímetros e repassado ao manipulador.

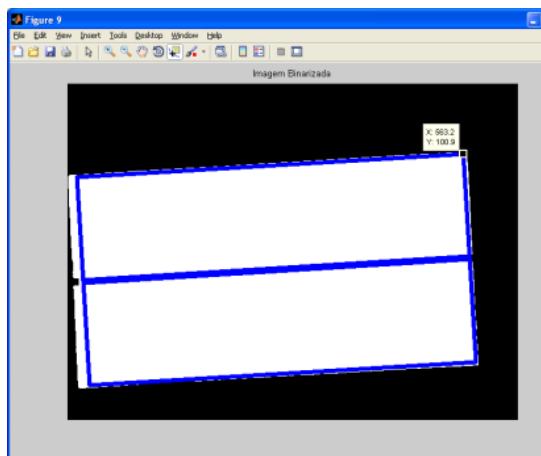


FIGURA 4.40 - Determinando o contorno da peça e suas coordenadas.

FONTE: O AUTOR, 2015.

Em um novo teste foi realizada a substituição da peça por uma outra com dimensões de 115 mm x 100 mm, mas sobre as mesmas condições para a captura da imagem, ou seja, mantida a iluminação, e a posição do manipulador. E como resultado, a FIG. 4.41 apresenta condições não satisfatórias. Isso auxiliou na determinação das dimensões mínimas que as peças devem apresentar para serem utilizadas neste sistema.

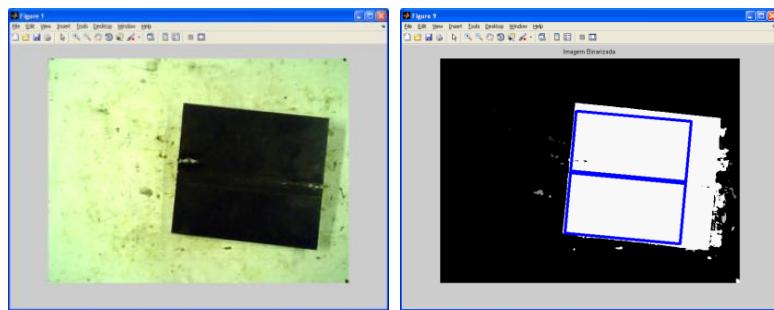


FIGURA 4.41 - Avaliando o comportamento do algoritmo. Resultado ruim.

FONTE: O AUTOR, 2015.

Ao contrário da peça da ilustração anterior, é possível verificar na FIG. 4.42, que utilizando uma peça com dimensões de 150 mm x 120 mm, e apesar do elevado ruído que contem a imagem binarizada o contorno do objeto ocorreu de forma satisfatória, se avaliarmos a quina superior da direita da peça junto com a do contorno em azul notasse que este último representa as dimensões do objeto da imagem.

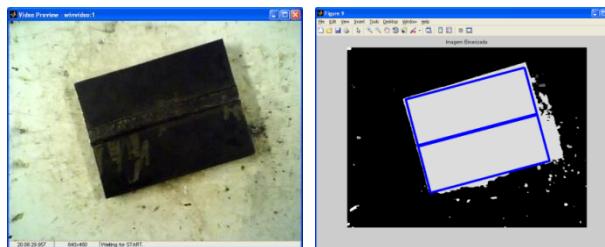


FIGURA 4.42 - Avaliando o comportamento do algoritmo. Resultado bom.

FONTE: O AUTOR, 2015.

Em um determinado momento dos experimentos uma lâmpada do laboratório que fica superior ao manipulador se queimou, e testes foram realizados para avaliar o quanto prejudicial isso se tornaria. Os resultados estão apresentados na FIG. 4.43.

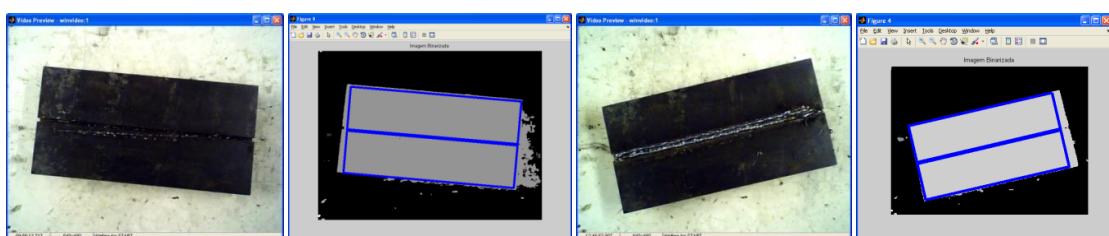


FIGURA 4.43 - Avaliando o algoritmo sob alterações da iluminação do ambiente.

FONTE: O AUTOR, 2015.

Após a resolução deste empecilho, tendo como solução à substituição de ambas as lâmpadas, por um novo modelo com potência e fluxo luminoso superior as que estavam instaladas anteriormente, foram feitos novos testes para avaliar novos resultados, a FIG. 4.44 representa as fotos obtidas com a última configuração válida após substituição das lâmpadas.

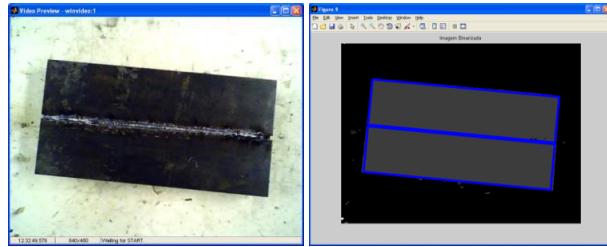


FIGURA 4.44 - Testes realizados após substituição das lâmpadas do ambiente.

FONTE: O AUTOR, 2015.

Vários testes de soldagem foram realizados sobre diversas peças, e sobre uma peça em especial, foi considerado a possibilidade de remoção dos cordões depositados para realização de novos testes. Daí fez-se a captura e tratamento da imagem sobre esta peça reaproveitada, e os resultados estão apresentados na FIG. 4.45.

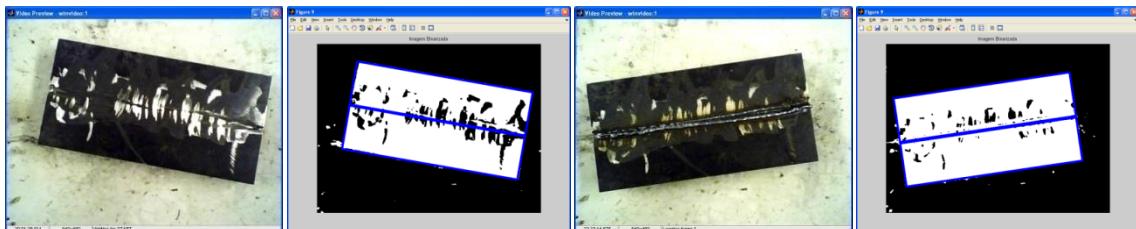


FIGURA 4.45 - Testes realizados no algoritmo com reaproveitamento de peça.

FONTE: O AUTOR, 2015.

Um dos pontos positivos do sistema implementado é que mesmo após vários ciclos de soldagem, e com a consequente desgaste da mesa, a mesma não apresenta mais sua condição inicial, ou seja, perfeitamente branca, conforme foi apresentado na FIG. 4.38, mas mesmo assim o algoritmo foi capaz de identificar a peça e determinar sua posição e o manipulador realizar sua tarefa de forma consistente e adequada. A FIG. 4.46 apresenta a mesa já desgastada.



FIGURA 4.46 - Mesa com desgaste da pintura.

FONTE: O AUTOR, 2016.

Já uma citação desfavorável ao algoritmo está apresentada na FIG. 4.47, em que a mancha amarelada da mesa prejudicou a binarização e consequentemente causou erro no *Bounding Box*, não resultando no contorno esperado. Verificou-se posteriormente que tal erro pode ser ocasionado também devido alguma alteração de posição ou na rotação da câmera, ou na altura de captura a imagem.

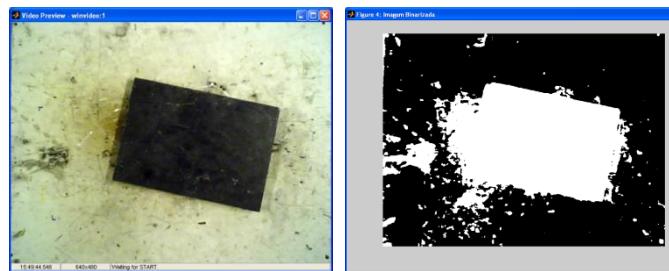


FIGURA 4.47 - Erro na definição do contorno.

FONTE: O AUTOR, 2016.

Uma nova condição de uso do algoritmo foi considerada, sendo que para neste novo caso o manipulador deve realizar a soldagem na peça que está montada conforme configuração da FIG. 4.48. A chapa maior possui um rebaixo e neste é montada uma segunda peça que deve ser unida através da solda de filete contínua em ambos os lados.



FIGURA 4.48 - Testes de soldagem sob uma nova configuração.

FONTE: O AUTOR, 2016.

Para este caso, são consideradas as mesmas condições anteriormente apresentadas, sendo necessário apenas adequação ao algoritmo uma vez que existem duas trajetórias a serem realizadas. Para que se consiga êxito no proposto, é necessário fazer uma diferenciação entre as peças uma vez que elas possuem a mesma cor, com isso é necessário fazer uma marcação na cor branca na peça que é montada no rebaixo para que os processamentos possam ocorrer da mesma forma anteriormente apresentada, sem a necessidade de mudanças complexas no algoritmo, e a FIG. 4.49 apresenta os resultados obtidos nestas operações. Enquanto que a FIG. 4.50 demonstra os pontos e o sentido de soldagem considerado mediante esta nova condição. Vale ressaltar que a marcação ou pintura não é o suficiente para contaminar a solda, pois a "perna" do cordão de solda não possui altura suficiente para atingir a região marcada.

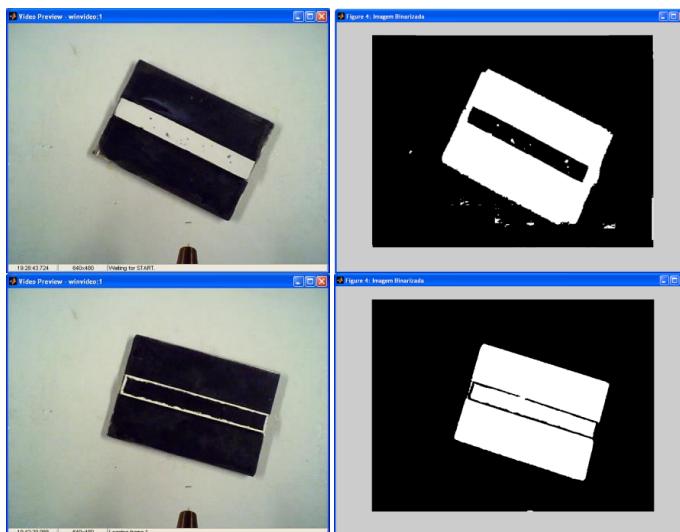


FIGURA 4.49 - Diferenciando as peças sob a nova configuração.

FONTE: O AUTOR, 2016.

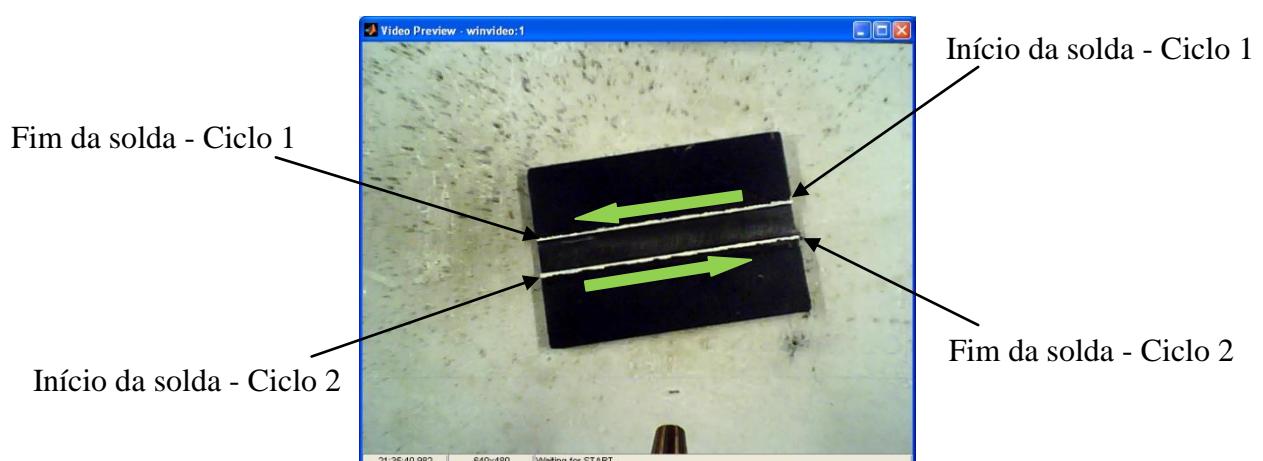


FIGURA 4.50 - Definição dos pontos e ciclos de soldagem.

FONTE: O AUTOR, 2016.

Na FIG. 4.51 e na FIG. 4.52 estão os resultados obtidos do processo de soldagem sob esta nova configuração. Vale lembrar que a avaliação não é da qualidade da solda, por isso não serão feitos testes macro e micrográficos da mesma, mas sim obter coordenadas e realizar a trajetória do robô.



FIGURA 4.51 - Soldas realizadas com a nova configuração.

FONTE: O AUTOR, 2016.



FIGURA 4.52 - Resultados em escala.

FONTE: O AUTOR, 2016.

5. ANÁLISE DOS RESULTADOS

Todo o sistema foi desenvolvido considerando um elemento retangular, mas a função *regionprops*, não está limitada a apenas esta forma geométrica, permite identificar diversas formas, em contrapartida, o manipulador robótico utilizado apresenta uma limitação de movimentos por não possuir o sexto eixo que permitiria movimentos rotacionais do eixos quatro e cinco simultaneamente, conforme apresentado na FIG. 5.1.



FIGURA 5.1 - Manipulador com cinco graus de liberdade.
FONTE: O AUTOR, 2015.

A FIG. 5.1 apresenta o manipulador robótico utilizado no desenvolvimento deste trabalho, e um detalhe a cerca de suas últimas juntas, que são conhecidas como *pitch* e *roll*, sendo que um manipulador de seis eixos possui a terceira junta chamada de *yaw*, que são assim definidas:

- ***Yaw*** produz um movimento da direita para a esquerda, assim como aquele usado para dizer não.
- ***Pitch*** produz um movimento de sobe e desce da mão, assim como aquele usado para dizer até logo.
- ***Roll*** produz um balanço lateral da mão, assim como aquele usado para dizer mais ou menos.

Como pontos positivos de todo o sistema, pode-se destacar a robustez: em que faz a separação do objeto da imagem mesmo se a cena 3D estiver com ruídos; a repetitividade na obtenção dos dados de posicionamento da peça sobre a mesa; a não perda da calibração do manipulador. A sequência de imagens da FIG. 5.2 apresenta a identificação da localização da peça sobre a mesa. É interessante ressaltar que mesmo com a imagem não estando em tão boa qualidade obteve o contorno de modo satisfatório.

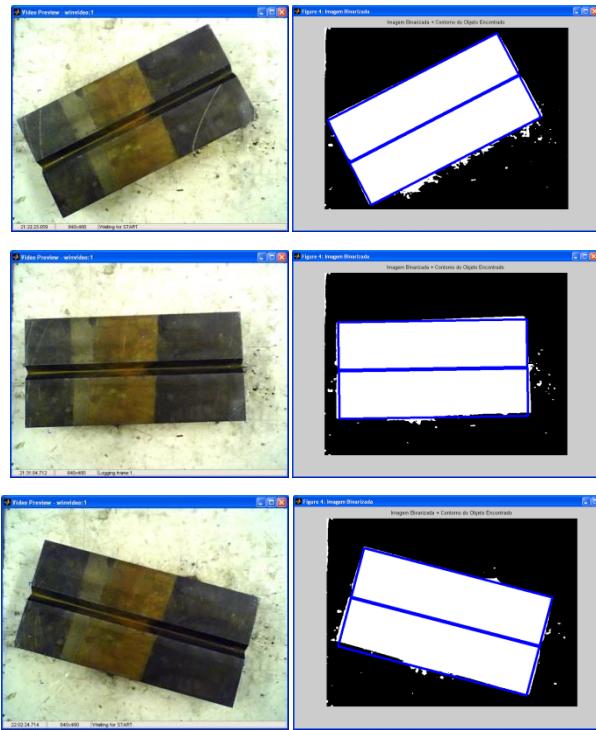


FIGURA 5.2 - Identificando a posição da peça e gerando a trajetória do manipulador.

FONTE: O AUTOR, 2015.

Uma primeira consideração foi levantada, conforme as literaturas obtidas, em que a cena e as condições de iluminação deveriam ser as melhores possíveis, então inicialmente a condição da mesa era conforme apresentada na FIG. 5.3, e na mesma figura está a identificação do contorno.



(a) com iluminação; (b) sem iluminação; (c) detecção de bordas.

FIGURA 5.3 - Condições iniciais do sistema

FONTE: O AUTOR, 2015

As literaturas falam das condições perfeitas de iluminação em que um sistema de visão deve estar submetido, mas com o que foi proposto abriu-se mão da grande flexibilidade que traz uma elevação no custo. A medida que foram feitos testes de soldagem e ocorreram movimentação de peças sobre a mesa, a pintura da mesma foi desgastando, neste ponto é que foi possível avaliar a robustez do algoritmo que não apresentou nenhuma necessidade de alteração mediante esta nova condição da cena 3D. A FIG. 5.4 apresenta as condições atuais da mesa.

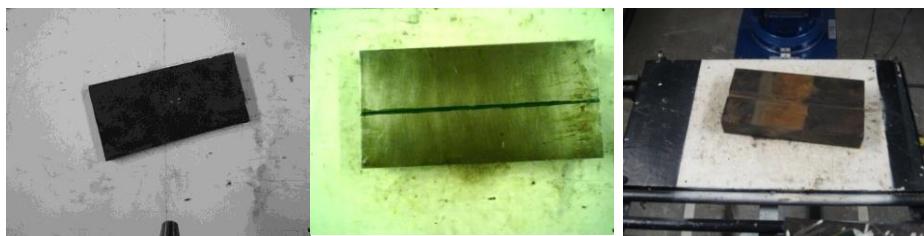


FIGURA 5.4 - Robustez do sistema desenvolvido.

FONTE: O AUTOR, 2015.

A escolha pela câmera de menor valor, foi comprovada como assertiva, uma vez que mediante o algoritmo, e às grandes variações ocorridas na cena, não houve necessidade de implementações. E considerando que o robô é um manipulador que passou por um processo de remanufatura, este também teve um desempenho satisfatório para o que foi almejado neste trabalho.

Para a seleção do sistema de soldagem, parte-se do princípio que o emprego da soldagem automática e suas variantes necessariamente não implicam que o equipamento adotado deve ser altamente sofisticado e caro, e este processo não deve ser sempre considerado para a obtenção de uma produção em massa, com isso, uma avaliação de todos os fatores relativos que leva em conta os efeitos, lucro, qualidade, deve ser feita. Neste trabalho, que visa o uso de um sistema totalmente de baixo custo, e como o desenvolvimento da trajetória de soldagem foi realizada por um manipulador robótico remanufaturado, considerou-se o sistema MAG mais básico.

Os ensaios realizados apresentaram uma robustez satisfatória do sistema desenvolvido, valendo ressaltar a necessidade de haver uma diferença entre as cores dos objetos na cena, sendo que ao menos o contorno da peça deve apresentar tonalidade mais escura que a da mesa, conforme foi comprovado na FIG. 4.45 anteriormente apresentada, em que foram removidos com auxílio de uma lixadeira elétrica os cordões de solda depositados, apresentando partes brilhantes da peça, e mesmo assim foi possível detectar o contorno do objeto e realizar a trajetória. Levando este processo para o ambiente industrial, em que se é a todo momento avaliado o tempo ciclo de operação de uma determinada atividade, este sistema deve ser melhorado pois o mesmo apresenta um elevado custo de tempo no processamento da imagem, gastando em média 02'49" para obter a imagem e realizar a soldagem utilizando os equipamentos especificados na TAB. 3.1.

Diversos foram os empecilhos encontrados durante o desenvolvimento deste projeto, dentre estes se pode citar: dificuldade e morosidade em importar itens para o sistema de controle do

manipulador robótico, sendo que devido um mau funcionamento do motor 05 foi necessário substituição do mesmo e de seu *encoder* para continuidade das atividades. Falhas apresentadas no conjunto motor/*encoder* do eixo 1, obrigaram a desmontagem completa do robô de sua base, e uma vez que foi realizada tal tarefa, foi então trocado o conjunto pelo novo importado, para que tal problema não mais seja apresentado.

Problemas recorrentes de travamento da haste que interliga o motor a junta 05, sendo necessário realizar uma melhor lubrificação e substituição do rolamentos que não foram encontrados com facilidade.

A cada uma das intervenções apresentadas acima, necessitou de um processo de calibração de todos os eixos manualmente, e posterior ajustes conforme já configurado no algoritmo.

Percebeu-se que o manipulador ao deslocar em uma linha reta ao longo da mesa o mesmo estava aproximando o TCP da mesa, com isso, foi-se necessário realizar o nivelamento e o travamento desta ao solo.

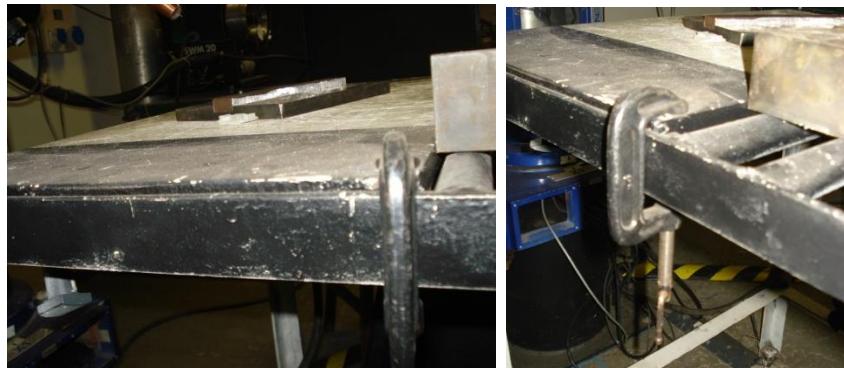


FIGURA 5.5 - Uso de sargento para manter a planicidade da mesa.

FONTE: O AUTOR, 2015.

A cada orientação da peça é necessário obter uma inclinação correta da ferramenta, no caso a tocha de soldagem para que a mesma possa obter o ângulo de ataque de forma adequada e realizar a soldagem de forma correta. Para esta etapa, grande foi a dificuldade para a correção deste parâmetro, uma vez que a falta do sexto eixo do manipulador resultou em mais uma variável a ser considerada no momento da geração da trajetória para os dois sentidos de soldagem, conforme apresentado na FIG. 5.6. Com a solução desta situação comprovou-se que os recursos utilizados são os mínimos necessários para o sistema de baixo custo proposto.



FIGURA 5.6 - Correção ângulo de ataque da tocha.

FONTE: O AUTOR, 2015.

6. CONCLUSÕES

Com o desenvolvimento desse trabalho, as seguintes conclusões podem ser obtidas:

Aplicabilidade dos algoritmos clássicos

A primeira técnica de segmentação abordada neste trabalho visa a detecção de pontos e linhas onde é capaz de detectar contornos de objetos, porém ela é sensível a ruídos e imperfeições na imagem. Já a técnica de detecção de bordas é capaz de detectar os contornos de forma mais eficiente. A utilização da derivada, vista no cálculo do gradiente para a detecção de bordas, pode causar ruídos. Por este motivo são utilizados os operadores de SOBEL, que possuem resultados semelhantes, porém sem causar ruídos. A técnica que utiliza o operador laplaciano permite detectar onde se encontram os *pixels* porém é a que causa mais ruídos das três técnicas abordadas, por utilizar uma segunda derivada.

Robustez e limitações

Méritos devem ser apontados sobre a robustez do sistema ao se considerar a detecção do contorno do objeto sob condições não necessariamente definidas como ideais pelas literaturas. Como limitações apontamos a condição da peça não poder ser totalmente brilhante, o que resulta em brilho excessivo e a consequente não definição de seu contorno. As dimensões da peça também são uma limitação do sistema que implica que não se tornou capaz de apresentar resultados satisfatórios, quando obtidas imagens sobre a mesma posição e orientação da câmera.

Gama de aplicações

Os resultados obtidos neste experimentos podem ser estendidos a aplicações não só para geração de trajetórias lineares, mas também curvilíneas, seja para a aplicação de soldagem elétrica, pintura, traçagem, aplicação de selantes, processos de corte, fresagem, entre outros. Já para a operação de soldagem a ponto, furação, parafusamento, manipulação *pick and place*, fazendo as devidas substituições da ferramenta no TCP do manipulador, a atividade se desenvolveria de forma ideal, uma vez que a precisão de posicionamento obtida com o sistema foi elevada.

Custo baixo

O maior fator que impede a adoção em massa de robôs é seu alto custo. O tempo que leva para se recuperar o investimento em um robô depende dos custos de compra, instalação, manutenção, e a

produtividade obtida com o mesmo. E como um dos maiores benefícios do sistema desenvolvido neste trabalho é em relação ao seu baixo custo de sua implementação, vale ressaltar que o uso de um manipulador remanufaturado que não utiliza mais linguagem de programação proprietária passou por um processo que o torna totalmente intercambiável entre outros robôs que possuam esta mesma configuração, ou seja, 5 graus de liberdade. A inclusão do sistema de visão de baixo custo através de uma *WebCam*, se justifica, uma vez que, não seria economicamente viável utilizar um sistema usual industrialmente, que possui custos elevados em um manipulador que pode ser considerado pela indústria como ultrapassado. Foi possível comprovar que o manipulador robótico utilizado neste trabalho apresenta excelente resultados para o que foi proposto, e que para o âmbito acadêmico, atende as necessidades.

Precisão em um nível adequado para processo de soldagem

Para o processo de soldagem, conforme informado anteriormente, se está preocupado apenas com a qualidade visual da solda, como foi tratado e representado na sequencia de imagens da FIG. 4.51, o nível de precisão do sistema é adequado para tal operação, pois proporcionou um cordão totalmente retilíneo. Com o uso do robô de cinco eixos o ângulo de ataque pode ser corrigido deixando seu último eixo perpendicular a mesa e girando-o em relação ao ângulo necessário obtido pela inclinação da reta, o que proporcionou uma boa localização do cordão, e consequentemente uma penetração satisfatória.

Robô de 5 eixos adequado para as soldas planas, mesmo que com outras geometrias

Manipuladores com menos de 6 graus de liberdade não podem atingir posições e orientações arbitrárias no espaço tridimensional. O acréscimo de graus de liberdade pode reduzir significativamente o desempenho de um robô: em geral, quanto mais graus de liberdade existir num braço de robô, menor será a capacidade de carga útil; menores velocidades poderão ser atingidas; maiores serão os erros de posicionamento; e mais alto será o seu custo. No caso do ASEA possuir cinco eixos , sendo três eixos no corpo e dois no punho, avaliou-se a trajetória que o mesmo irá realizar. Nesse caso, utilizado para soldas planas, independente da geometria a ser utilizada o mesmo apresenta resultados satisfatórios.

ABSTRACT

This work proposes to add a low vision system cost by overhauling process which was submitted to the robotic manipulator ASEA IRB6-S2 belonging to the LRSS of the UFMG. Thus, with the use of vision elements that have low investment, associated with a control system developed and that works reliably, and algorithms that require minimal operator intervention, this process aims to present a very viable solution for the reuse robots that are destroyed in factory floor and can be donated to academic institutions to provide study, research and development in robotics. With this, the digital image processing techniques used in this work and a webcam that makes up the computer vision system, which will allow the robotic manipulator perform welds after the identification, location and orientation of a piece on a table without no prior knowledge or any other type of sensor that assists. With the application software for mathematical analysis, algorithms for image processing and path generation, it becomes possible that the robotic manipulator to perform the welding operation on the coordinates obtained after processing of the captured image. The main focus of this research, based on the study of the proposed algorithm is to verify not only the fulfillment of the task of positioning, as well as the robustness against the environment variations, demonstrating that the system presents excellent value for money.

Key words: *Robotics, Computer Vision, Welding*

7. SUGESTÃO PARA TRABALHOS FUTUROS

- Substituição do *Mach3*[®] pelo Arduino, de modo que não seja necessário gerar o código de máquina, e o controle dos motores ocorre diretamente, em testes realizados, este demonstrou total intercambiabilidade.
- Desenvolvimento de todo este sistema utilizando *softwares* livres, como Linux[®], distribuição Ubuntu 10.04 para o controle dos *drives* e do Scilab[®] para geração da trajetória, e aquisição e processamento de imagens, assim como é feito com o MatLab[®].
- Utilizando um sistema laser para obter os contornos da peça de modo que seja independente das cores do objeto/mesa, e sem interferência de iluminação externa. E que seja capaz de extrair características de interesse na imagem, como aglomerados, descontinuidades retas, curvas, quinas que são importantes para caracterizar o chanfro.
- Utilização do sistema atual para atividades de manipulação de peças, pintura, seleção e exclusão de peças que não estejam de acordo com o processo.
- Desenvolver um sistema de visão estereoscópica de modo que se possa determinar a distância da peça, bem como sua espessura, e qual a altura que o manipulador deve alcançar para realizar tarefa sobre a mesma.

8. REFERÊNCIAS

- 1** Adade Filho, A., Fundamentos de Robótica: Cinemática, Dinâmica e Controle de Manipuladores Robóticos. São José dos Campos: Instituto Técnico de Aeronáutica, 353 p. 1992.
- 2** Albuquerque, M.P.; Caner, E.S.; Mello, A. G.; Albuquerque, M.P., Análise de Imagens e Visão Computacional, CBPF, 2012.
- 3** Alfaro, S.; Bauchspieß, A.; Drews, P., Intelligent Systems for Welding Process Automation. In: Proceedings of Insert Conference Abbreviation, 2003.
- 4** Alvarenga, B.; Máximo, A. Curso de Física. Volume 2 São Paulo, Scipione, 414p, 2000.
- 5** Alves, J. B. M., Controle de Robô, Cartgraf, Campinas, 1988, São Paulos. pp 1-23.
- 6** ARTSOFT. Mach 3 CNC Controller: Software Installation and Configuration. ArtSoft: USA, 2007.
- 7** ASEA, Product Manual, CK 09-1510E. Janeiro, 1984.
- 8** Avcibas, I.; Sankur, B.; Sayood, K. Statistical Evaluation of Image Quality Measures. Journal of Electronic Imaging, 11:206-223, 2002.
- 9** Barequet, Gill; Har-Peled, Sariel, "Efficiently approximating the minimum-volume bounding box of a point set in three dimensions", Journal of Algorithms 38 (1): 91–109, 2001.
- 10** Bastos, T. F., Aplicação de Robôs nas Indústrias. Espírito Santo, UFES. 1999. Disponível em: <<http://www2.ele.ufes.br/~tfbastos/RobMov/robosindustriais.pdf>> . Disponível em 08/03/2011.
- 11** Basu, A., Active Calibration of Cameras: Theory and Implementation. IEEE Transactions on Systems, Man and Cybernetics, United States of America, v. 25, n. 2, p. 256-265, Fevereiro, 1995.
- 12** Bins, L., Fonseca, L. and Erthal, G., 1996, Satellite imagery segmentation: a region growing approach. In VIII Brazilian Symposium on Remote Sensing, pp. 677–680.
- 13** Bomfim, M.H.S.; Bracarense, A.Q.; Lima II, E.J.; Gontijo, R.A. Overhauling of an ASEA Robô IRB6 with open architecture. 11th IEEE International Conference on Trust, Security and Privacy in Computing and Communications (Trust Com-2012). Liverpool, UK, pp. 482-489, 2012.
- 14** Bomfim, M.H.S., Projeto de um gabinete de controle com interface USB para robôs de até seis graus de liberdade. Trabalho de Graduação em Engenharia Mecânica. UFMG. Belo Horizonte, 2009, 46 p., 2013.
- 15** Bracarense, A. Q., "Estudo da sensibilidade do arco de soldagem TIG à variações geométricas do ânodo", Mestrado em Engenharia Metalúrgica e de Minas, Universidade Federal de Minas Gerais, 1986, 187 p.
- 16** Broering, C. E., Desenvolvimento de Sistemas para a Automação da Soldagem e do Corte Térmico. Fev. 2005. 96p. Dissertação de Mestrado. Dept. de Engenharia Mecânica UFSC, Florianópolis, 2005.

- 17** Bruyninckx, H., Open Robot Control Software: The Orocó Project, IEEE International Conference on Robotics and Automation, Piscataway, NJ, 2001, pp. 2523-2528.
- 18** Canny, J. F. A computational Approach to Edge Detection. In: IEEE Transactions on Pattern Analysis and Machine Intelligence, 8(6). p. 679-689, 1986.
- 19** Chapman, S. J. Programação em MatLab para Engenheiros. 2^a edição. São Paulo:Editora CENGAGE, 2011, pp. 38-251.
- 20** Choi, M. G.; Jung, J. H.; Jeon, J. W. No-Reference Image Quality Assessment Using Blur and Noise. World Academy of Science, Engineering and Technology, n. 12, v. 28. p 163-167, Janeiro, 2009.
- 21** Coelho, F. G. F.; Bomfim, M. H.S.; Bracarense, A. Q.; Lima II, E. J., Soldagem Robotizada Utilizando Sistema de Visão Computacional em um Manipulador com Arquitetura Aberta. Brazil Automation, São Paulo, 2013.
- 22** Corke, P.; Hutchinson, S.; Hager, G., A Tutorial on Visual Servo Control. IEEE Transactions on Robotics and Automation, U.S.A. v. 12, n. 5, p. 651-670, Outubro, 1996.
- 23** Craig, J. J. Introduction to Robotics Mechanics and Control. 2^a edição. São Paulo: Editora Addison-Wesley Publishing Company, 1989, pp. 1-144.
- 24** Duda, R.; Hart, P., Use of the Hough Transformation to Detect Lines and Curves in Pictures. Comm. of ACM 15, 1. 11-15; 1972.
- 25** Dutra, J. C.; Rosa, V. M., O Arco Elétrico como Sensor de Procura de Junta. XXI ENTS. Caxias do Sul. 1995 p807-820.
- 26** Faugeras, Olivier. Three dimensional computer vision. Massachusetts: MIT Press, 695 p. 1993.
- 27** Feliciano, F. F.; Souza, I. L.; Leta, F. R., Visão Computacional Aplicada à Metrologia Dimensional Automatizada: Considerações Sobre sua Exatidão. Engevista, V.7, n. 2, p.38-50, Dezembro, 2005.
- 28** Foresti, R. L, Sistema de Visão Robótica para Reconhecimento de Contornos de Componentes na Aplicação de Processos Industriais. Dezembro, 2006.
- 29** Fu, King Sun; Gonzalez, Rafael; Lee, George. Robotics: Control, Sensing, Vision and Intelligence. Singapore: McGraw Hill, 580 p. 1987.
- 30** Gonzales, R. C. Digital Image Processing, Addison Wesley, New York, 1993.
- 31** Gonzalez, R. C.; Woods, R. E., Processamento de imagens digitais. 1. ed. São Paulo: Edgard Blucher, 2000.
- 32** Gonzalez, R. C. ; Woods, R. E.; Eddins, S., Digital Image Processing Using MatLab. New Jersey: Prentice Hall, 609 p. 2004.
- 33** Grassi, M.V., Desenvolvimento e Aplicação de Um Sistema de Visão para Robô Industrial de Manipulação. UFRGS. Porto Alegre, 2005.

- 34** Groover, M.; Weiss, M.; Nagel, R.; Odrey, N., Industrial Robotics: Technology, Programming, and Applications. 2. ed. Singapore: McGraw Hill, 546 p. 1989.
- 35** Heikkilä, J., Silvén, O., A Four-step Camera Calibration Procedure with Implicit Image Correction. IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'97), San Juan, Puerto Rico, pp. 1106-1112, 1997.
- 36** Herbst, E., Camera Calibration by Corners Detection, 2003. <http://www.tjhsst.edu/~eherbst/techlab/>. Disponível em 01/05/2015.
- 37** Hoff, W., Department of Electrical Engineering and Computer Science. Colorado School of Mines Image and Multidimensional Signal Processing. 2014.
- 38** Hough, P.V.C., "Method and Means for Recognizing Complex Patterns," U.S. Patent No. 3069654, 1962.
- 39** Jahne, B.; Haubecker, H., Computer Vision and Applications. Academic Press, 2000.
- 40** Jüngel, Matthias; Hoffmann, Jan; Lötzsch, Martin - "A Real-Time Auto-Adjusting Vision System for Robotic Soccer" - 7th International Workshop on RoboCup 2003.
- 41** Keyence - 2015, Measurement Application Guide - Thickness/Width Measurement. AS_48408_AG_600850_GB_WW_1074-3.pdf.
- 42** Koike, C. C.; Barros, M.H.L.L.; Carvalho, G.C.; Alfaro, S.C.A., Monitoração da Poça de Fusão no Processo de Soldagem GMAW. XV Congresso Brasileiro de Engenharia Mecânica. Águas de Lindoia/SP, Novembro, 1999.
- 43** Lages, W. F.; Henriques, R. V. B.; Bracarense, A. Q., Arquitetura Aberta para Retrofitting de Robôs, Manet Notes Workshop, Bragança Paulista, 2003, SP.
- 44** Lee, G.; Gonzalez, R. C.; Fu, K. S.. Tutorial on robotics. United States of America: IEEE Computer Society Press, 1986. 731 p.
- 45** Lenz, R.; Tsai, R.Y., A New Technique for Fully Autonomous and Efficient 3D Robotics hand/Eye Calibration. IEEE Transactions on Robotics and Automation, U.S.A. v. 5, n. 3, p. 345-358. Junho, 1989.
- 46** Leta, F. R.; Conci, A.; Azevedo, E. , Computação Gráfica: Teoria e Prática – Vol 2. Ed. Elsevier. Rio de Janeiro, 2008.
- 47** Li, X. Blind Image Quality Assessment. In Proceedings of the IEEE International Conference on Image Processing, pp. I:449-452, 2002.
- 48** Lima II, E.J.; Torres, G. C. F. ; Felizardo, I. ; Ramalho Filho, F. A. ; Bracarense, A. Q. . Robotização e Otimização do Processo de Soldagem Orbital de Tubos. In: VIII Seminário de Automação de Processos da Associação Brasileira de Metalurgia e Materiais, 2004, Belo Horizonte - MG, 2004.

- 49** Lima II, E. J.; Castro, C. A. C.; Bracarense, A. Q.; Campos, M. F. M. Determination of the Relation Between Welding Parameters, Weld Pool Width and Weld Bead Geometry Using High Speed Camera. In COBEF, 2005.
- 50** Litwiller, D., CCD vs. CMOS: Facts and Fiction, Photonics Spectra, January, 2001, 154-158.
- 51** Lopes, G.C., Autocalibração de um Braço Robótico Utilizando Acelerômetros. DEMEC, UFMG. Belo Horizonte, Dezembro, 2015.
- 52** Marques, F. O.; Vieira N. H., Processamento Digital de Imagens. Brasport. Rio de Janeiro, 1999.
- 53** Molina L.; Carvalho E.; Moura M.; Freire E.; Montalvão J., Um Método de Visão Robótica para Identificação de Cordões de Solda em Tanques de Armazenamento Visando Inspeção Automatizada. Centro de Ciências Exatas e Tecnologia. UFSe. Maio, 2002.
- 54** Pinho, T. B. R.; Rede de Instrumentação e Atuação para Manipulador IRB6. Engenharia de Controle e Automação, UFMG, 2013.
- 55** Preising, B.; Hsia, T., Robot Performance Measurement and Calibration Using a 3D Computer Vision System, v. 13, n. 4, p. 327-337. Agosto, 1995.
- 56** Ribeiro, A. S.; Lima II, E. J.; Bomfim, M. H. S.; Almeida, A. G. S, Metodologia para Implementação de Retrofitting de Controladores de Equipamentos de Automação de Processos. 8º Congresso Ibero-Americanano de Engenharia Mecânica. Cusco. PE, 2007.
- 57** Rodrigues, D. L., Procedimento Automático para Calibração de Sistemas de Visão Robótica para Operações Pick-and-Place. ITA, 1997.
- 58** Rosenfeld, A., Image Analysis and Computer Vision. Computer Vision Graphics and Image Processing – Vol 59, p. 367-404.
- 59** Rudek, M.; Coelho, L. S.; Canciglieri J. O., Visão Computacional Aplicada a Sistemas Produtivos: Fundamentos e Estudo de Caso. In: XXI Encontro Nacional De Engenharia De Produção, 2001.
- 60** Russ, John. C. - The Image Processing Handbook the Fifth edition. CRC Ed. Boca Raton, FL, 2007
- 61** Sebe, N.; Cohen, I.; Garg, A.; Huang, T. S. Machine Learning in Computer Vision. Springer, 2005.
- 62** Shirai, Y.; Inoue, H., Guiding a Robot by Visual Feedback in Assembling Tasks. IEEE Pattern Recognition, Beijing. v. 5, n. 3, p. 99-108. Abril, 1973.
- 63** Siegwart, R. ; Nourbakhsh, I., Introduction to Autonomous Mobile Robots. MIT Press. Cambridge, 2004. Capítulo 4.
- 64** Sobel, I. An Isotropic 3x3 Gradient Operator. Machine Vision for Three-Dimensional Scenes, Academic Pres, 1990.
- 65** Stadler, W., Analytical Robotics and Mechatronics. United States of America. McGraw-Hill, 1995. 576p.

- 66** Trucco, E.; Verri, A., Introductory Technique for 3D Computer Vision. U.S.A.: Prentice Hall, 343 p. 1998.
- 67** Tsai, R. Y., A Versatile Camera Calibration Technique for High-Accuracy 3D Machine Vision Metrology Using Off-The-Shelf TV Cameras and Lenses. IEEE Journal of Robotics and Automation, RA-3(5):323-344, August, 1987.
- 68** Ventura, L.; Neto, J. C. C., Ametropias Oculares. Revista Brasileira de Ensino de Física, 1995, pp. 305-316.
- 69** Venturieri, A.; Santos, J. R. Técnicas de Classificação de Imagens para Análise de Cobertura Vegetal. In: Assad, E. D.; Sano, E. E. ed. Sistemas de informação geográfica: Aplicações na agricultura. 2 ed. Brasília. EmbrapaCPAC, 1998, Cap. 2, p. 13-29.
- 70** Vianna, M. E., Calibração de Sistemas de Visão Computacional para Aplicação em Automação e Robótica. Belo Horizonte, 2009.
- 71** Vilarinho, L. O.; Machado, M. V.; Mota, C. P.; Finiz Neto, R. M., Embedded Monitoring and Control Systems for Pipe Welding. 2013. Rio Pipeline Conference and Exposition, Brazilian Petroleum, Gas and Biofuels Institute. September , 2013.
- 72** Vuolo, J. H., Fundamentos da Teoria dos Erros. São Paulo. Edgard Blücher, 1992, 249p.
- 73** Wang, C., Extrinsic Calibration of a Vision Sensor Mounted on a Robot. IEEE Transactions on Robotics and Automation, Boston, v. 8, n. 2, p. 161-175. Abril 1992.
- 74** Yhong, G.; Hongjiang, Z., Image Database System With Fast Image Indexing Capability Based on Color Histograms, Proceedings of 1994 IEEE region 10's 9th Annual International Conference, 1994.
- 75** Young, M. The Pinhole Camera: Imaging Without Lens or Mirrors. The Physics Teacher, December, 1989.
- 76** Zhang, Y. J., A Survey on Evaluation Methods for Image Segmentation. Pattern Recognition, v.31 p1335-1346, 1996.

8.1 Sítios visitados na internet

- 77** Camera Calibration Toolbox for MatLab. http://www.vision.caltech.edu/bouguetj/calib_doc. Disponível em 10/12/2013.
- 78** (MachSupport, 2012). Disponível em: <<http://www.machsueport.com/>>. Disponível em 10/10/2014.
- 79** (MatLab, 2012). http://www.mathworks.com/help/images/ref/regionprops.html?s_tid=gn_loc_drop. Disponível em 10/09/2014.