

MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA MECÂNICA

METODOLOGIA DE DETECÇÃO DE TRAJETÓRIA DE SOLDAGEM BASEADA
EM VISÃO COMPUTACIONAL

por

Omar Alejandro Bauernfeind

Dissertação para obtenção do Título de
Mestre em Engenharia

Porto Alegre, Julho de 2017

METODOLOGIA DE DETECÇÃO DE TRAJETÓRIA DE SOLDAGEM BASEADA
EM VISÃO COMPUTACIONAL

por

Omar Alejandro Bauernfeind
Engenheiro Eletrônico

Dissertação submetida ao Programa de Pós-Graduação em Engenharia Mecânica, da
Escola de Engenharia da Universidade Federal do Rio Grande do Sul, como parte dos
requisitos necessários para a obtenção do Título de

Mestre em Engenharia

Área de Concentração: Processos de Fabricação

Orientador: Prof. Dr. José Antônio E. Mazzaferro

Co-orientador: Prof. Dr. Flávio José Lorini

Aprovada por:

Prof. Dr. Rafael Antonio Comparsi Laranja,PROMEC/UFRGS

Prof. Dr. Arnaldo Ruben GonzalezPROMEC/UFRGS

Prof. Dr. Alcy Rodolfo Dos Santos CarraraDEMEC/UFRGS

Prof. Dr. Jakson Manfredini Vassoler
Coordenador do PROMEC

Porto Alegre, 28 de Julho de 2017.

AGRADECIMENTOS

Agradeço a meus pais por me ensinarem a importância do estudo e da instrução para a minha vida.

A minha mãe e irmã por seu constante apoio e motivação por todo o tempo do mestrado.

Ao Orientador (Prof. Dr. José Antônio E. Mazzaferro) e Co-Orientador (Prof. Dr. Flávio José Lorini) pela constante ajuda e por me orientar com o trabalho.

Agradeço a CAPES (Coordenação de Aperfeiçoamento de Pessoal de Nível Superior) pela concessão da bolsa durante todo o período de realização deste mestrado.

Agradeço a Universidade Federal do Rio Grande do Sul por me acolher como aluno de Pós-Graduação.

À comissão de avaliação pelo seu tempo, pelas críticas e sugestões.

Agradeço aos meus amigos pelo apoio e motivação.

RESUMO

Em geral, um problema importante em um sistema de soldagem robotizado, ou com manipuladores automáticos, é a necessidade de uma trajetória ser reprogramada quando a forma ou a posição das peças mudam. Como solução a esse problema é proposto um método que utiliza técnicas de visão computacional, para assim detectar a trajetória de soldagem em juntas de topo. O método deve ser dinâmico permitindo a identificação de juntas de topo com cantos não próximos, linhas não retas e tamanhos diferentes entre as peças, assim como apresentar robustez contra variáveis desfavoráveis do ambiente industrial como riscos nas peças ou no fundo e mudanças de luminosidade. A trajetória é detectada baseando-se na linha de bordas da imagem global e adicionalmente trabalha-se sobre duas características da linha de solda: distância quase constante entre linhas paralelas e pixels de linha de solda com menor intensidade de luminosidade que as bordas. Uma etapa é proposta para robustez do método sobre linha de bordas descontínuas e possíveis erros em ponto inicial e final de solda. O trabalho proposto é validado com distintas configurações de junta de topo, como com cantos de peças não coincidentes, linha de solda não reta e diferentes orientações de peças. Os pontos da trajetória de solda obtidos são comparados com uma linha de solda considerada ideal, resultando em uma média e desvio padrão geral inferiores à um valor de 0,5 mm. Uma validação experimental é realizada com uma solda executada por um robô industrial seguindo os pontos de solda detectados, com os resultados demonstrando que o método efetivamente pode definir uma trajetória de solda para aplicações industriais robotizadas.

Palavras-chave: Solda de arco robotizada; Detecção de Linha de Solda; Processamento de Imagens; Solda de junta de topo.

ABSTRACT

In general, one of the most important problems in a robotic welding system, or in automatic manipulations, is the requirement of a path trajectory to be reprogramed when the shape or the position of the welding pieces are changed. In order to detect the welding seam in butt joints, a method that makes use of computational vision techniques is proposed. The method needs to be dynamic against variation in the configuration of the pieces, so as curved or not straight seam lines, not coincident corners; beside of that, it must to present robustness against unfavorable variables of the industrial environment, so as scratches in the pieces or illumination changes. Two features of the welding line are taken into account to develop the method: almost constant distance between parallel seam lines and darker pixels in the center of the seam than in the borders. Moreover a robustness step is proposed over two weaknesses of the method: discontinuities in the edges of the welding line and possible errors in the location of the start and end welding points. The validation step of the method involves different configuration of butt joints, as pieces without corner being coincident, not straight welding line and different orientations. The points of the welding seam detected by the method, are compared against a set of welding points considered as an ideal set of points, getting as results a mean and standard deviation lower than ± 0.5 mm. An experimental test is carried out by an industrial robot that welds two pieces following the welding line points, with the results showing that the method can effectively define a welding trajectory for industrial robotics applications.

Keywords: Robotic arc welding; Welding seam detection; Image Processing; Butt Joint Welding.

ÍNDICE

1.	INTRODUÇÃO	1
1.1	Descrição do Problema.....	2
1.2	Objetivo do Trabalho.....	3
2.	REVISÃO BIBLIOGRÁFICA.....	4
2.1	Fundamentação teórica.....	4
2.1.1	Visão Computacional.....	4
2.1.1.1	Captura da imagem	5
2.1.1.2	Técnicas de iluminação	7
2.1.1.3	Processamento de imagem.....	9
2.1.1.4	Detecção de bordas	9
2.1.1.5	Transformada de Hough	13
2.1.1.6	Algumas relações básicas entre pixels.....	14
2.1.1.7	Ligação de borda preditiva (<i>Predictive Edge Linking</i> ou PEL).....	16
2.1.1.8	Calibração da Câmera	18
2.1.2	Sistemas de robôs de solda industriais.....	26
2.1.2.1	Breve introdução à robótica	26
2.1.2.2	Definição de Robô	27
2.1.2.3	Sistema de soldagem.....	28
2.1.2.4	Sistema robótico.....	28
2.1.2.5	Descrição do sistema robótico de soldagem	29
2.2	Estado da arte	30
3.	DESCRIÇÃO DO SISTEMA.....	38
3.1	Método 1: Detecção de linhas retas.....	38
3.1.1	Processo de reconhecimento da posição inicial e final da solda	42
3.2	Método 2: Detecção de linhas retas e curvas	48
3.3	Resumo do método 2 de detecção de linha de solda	49
3.4	Filtragem e detecção de bordas	52
3.4.1	Detecção de vizinhanças.....	53

3.4.2	Reta perpendicular das vizinhanças:.....	55
3.4.3	Avaliação de intensidades.....	56
3.4.4	Detecção do pixel do final da linha	57
3.4.5	Medição de comprimento de linhas de borda.....	58
3.4.6	Seleção de linha de borda que apresenta a linha de solda:	59
3.4.6.1	Detecção de pixels de linha paralela.....	60
3.4.6.2	Avaliação e adição de pixels.....	65
3.4.7	Critério para escolha do ponto de solda.....	67
3.5	Solução de linha de solda descontínua	69
3.5.1	Técnicas para correção de bordas	72
3.5.2	Análise e correção de conectividade.....	75
3.5.3	Adição da ROI $W_{ROI}(x, y)$ na matriz de linha de solda $B_{ls}(x, y)$	78
3.6	Correção dos pontos finais	79
3.6.1	Região de Interesse e linha de solda.....	81
3.6.2	Linha de Hough com os pontos de solda:.....	81
3.6.3	Detecção de cantos com técnica <i>K-cosine</i>	83
3.6.4	Detecção do ponto da linha de solda menos distante do canto.....	85
4.	AVALIAÇÃO EXPERIMENTAL	87
4.1	Descrição	87
4.2	Calibração.....	90
4.3	Resultados em simulação	92
4.3.1	Avaliação do Caso 1	93
4.3.2	Avaliação do Caso 2	94
4.3.3	Avaliação do Caso 3	95
4.3.4	Análise geral de resultados do método em diferentes configurações.....	97
4.4	Análise dos parâmetros de influência no método:.....	97
4.5	Testes experimentais	98
5.	CONCLUSÕES.....	102
5.1	Sugestões para trabalhos futuros.....	103
6.	PUBLICAÇÕES.....	104
	REFERENCIAS BIBLIOGRÁFICAS.....	105

LISTA DE FIGURAS

Figura 2.1	Sensibilidade espectral dos 3 tipos de cones.....	6
Figura 2.2	Círculos representando as cores primárias do sistema aditivo.....	7
Figura 2.3	Tipos de iluminação – (a) direta, (b) difusa , (c) traseira.....	8
Figura 2.4	Matriz de tamanho 3x3.....	11
Figura 2.5	Operadores de gradientes clássicos – (a) Robert, (b) Prewit, (c) Sobel.....	11
Figura 2.6	Representação de Hough.....	13
Figura 2.7	Vizinhança-4 em um pixel.....	14
Figura 2.8	Os quatro vizinhos diagonais de um pixel denotados por $N_D(p)$	15
Figura 2.9	Pixels com distância D_4	16
Figura 2.10	Pixels com distância D_8	16
Figura 2.11	Caminho em 4 direções com predição.....	18
Figura 2.12	Eixos de coordenadas que descrevem a geometria da câmera.....	20
Figura 2.13	Componentes de um sistema robótico.....	30
Figura 2.15	Detecção de pontos iniciais de solda proposto por Chen et al., 2006.....	31
Figura 2.16	Detecção de pontos iniciais de solda proposto por Zhu et al., 2005.....	32
Figura 2.17	Detecção de trajetória de soldagem proposto por Chen et al., 2007.....	34
Figura 2.18	Detecção de trajetória de soldagem proposto por Shi et al., 2009.....	35
Figura 2.19	Detecção de trajetória de soldagem proposto por Dinham e Fang, 2013.....	37
Figura 3.1	Diagrama de fluxo de método 1.....	39
Figura 3.2	Imagem de peças de trabalho.....	40
Figura 3.3	Borda de peças após aplicação de detecção de borda Canny.....	41
Figura 3.4	Linhas de Hough superpostas sobre as bordas da imagem.....	42
Figura 3.5	Imagem de 10x10 para verificar condições de ser ponto de interesse.....	43
Figura 3.6	Metodo1: escolha de ponto aleatório.....	45
Figura 3.7	Metodo1: escolha de segundo ponto.....	45
Figura 3.8	Metodo1: escolha de terceiro ponto.....	46
Figura 3.9	Metodo1: escolha do quarto ponto.....	46
Figura 3.10	Pontos de interseção sobrepostos sobre as peças – (a) esquerda, (b) direita.....	47

Figura 3.11	Detecção de ponto inicial e final da solda.....	48
Figura 3.12	Imagem de solda em tons de cinza $B_c(x,y)$	49
Figura 3.13	Fluxograma geral do método apresentado no trabalho.	51
Figura 3.14	‘Matriz de bordas iniciais’ $B_{bi}(x,y)$ obtida com detecção de bordas de Canny.	52
Figura 3.15	Procura de pixel de borda nas 8 vizinhanças.	53
Figura 3.16	Exemplo de média de coordenada de pixels vizinhos.....	54
Figura 3.17	Avaliação de pixels localização de pixels de linhas paralelas.	55
Figura 3.18	Matriz estimada de linha de solda $B_{els}(x,y)$	57
Figura 3.19	Dois casos onde um pixel é considerado final de linha – (a) um pixel de vizinhança, (b) dois pixels de vizinhança.	58
Figura 3.20	Matriz de linha de solda $B_{ls}(x,y)$	60
Figura 3.21	Exemplo de critério do sinal na linha perpendicular.....	62
Figura 3.22	Imagem ampliada de linha de solda em junta topo para apresentar a distancias relativamente constante.	64
Figura 3.23	Imagem ampliada de linha de borda que não abrange a linha de solda na totalidade.	65
Figura 3.24	Diferenças entre escolhas de pontos de solda (a) ponto central, (b) ponto de menor valor de intensidade.....	68
Figura 3.25	Linha de solda detectada em peças de trabalho.	69
Figura 3.26	Região de interesse (ROI) com linha de solda não completa.....	70
Figura 3.27	Fluxograma solução de descontinuidade na ‘Matriz de bordas inicial’ $B_{bi}(x,y)$	71
Figura 3.28	Janela $W_{ROI}(x,y)$ dentro da matriz $B_c(x,y)$	72
Figura 3.29	Imagem $W_{ROI}(x,y)$ após o processo de erosão com um elemento de estrutura como uma circunferência de raio 3.....	73
Figura 3.30	Filtro da mediana sobre ROI $W_{ROI}(x,y)$	74
Figura 3.31	Segmentação da linha de solda na imagem $W_{ROI}(x,y)$	74
Figura 3.32	Operação ‘remove’ na imagem $W_{ROI}(x,y)$	75
Figura 3.33	Método de união entre o pixel do final da linha e linhas de solda novas.....	76
Figura 3.34	Correção de caminho, os pixels entre a coordenada $(x_{fw} + 4, y_{fw} - 4)$ e o centro são transformados em pixels de borda.....	77
Figura 3.35	Pixels de solda não desejáveis nos pontos finais da linha de solda.	79

Figura 3.36	Fluxograma para correção de pontos finais de linha de solda.	80
Figura 3.37	Região de Interesse no extremo da linha de solda com pontos indesejáveis. ...	81
Figura 3.38	Etapas de detecção da linha na região de interesse.	82
Figura 3.39	Cantos coincidentes e cantos não coincidentes.	83
Figura 3.40	Linha de bordas no final de linha de solda.	84
Figura 3.41	Exemplo de método <i>K-cosine corner detection</i>	85
Figura 3.42	Exemplo de correção de final de linha.	86
Figura 4.1	Cálculo de erro para método de detecção de linha de solda.	88
Figura 4.2	Configurações de peças para avaliação do método.	89
Figura 4.3	Sistema de suporte para a câmera com suas medidas.	89
Figura 4.4	Ambiente de trabalho para detecção de linha de solda.	90
Figura 4.5	Padrão de pontos de calibração da câmera.	91
Figura 4.6	Padrão de pontos utilizados para processo de calibração.	92
Figura 4.7	Linha de pontos de solda obtida para peças em tipo de junta de topo, linha de solda reta e cantos coincidentes.	93
Figura 4.8	Erro de linha de solda detectada com relação a uma linha ideal, para linha (a) diagonal, (b) vertical, (c) horizontal.	94
Figura 4.9	Linha de pontos de solda obtida para peças em tipo de junta de topo, linha de solda reta e cantos não coincidentes.	94
Figura 4.10	Erro de linha de solda detectada com relação a uma linha ideal, para linha (a) diagonal, (b) vertical, (c) horizontal.	95
Figura 4.11	Linha de pontos de solda obtida para peças em tipo de junta de topo, linha de solda não reta e cantos coincidentes.	96
Figura 4.12	Erro de linha de solda detectada com relação a uma linha ideal – (a) diagonal, (b) vertical, (c) horizontal.	97
Figura 4.13	Exemplo de falha do algoritmo por sombra.	98
Figura 4.14	Conjunto de passos para teste experimental.	99
Figura 4.15	Sistema de soldagem.	100
Figura 4.16	Soldagem em peça de caso 3, (a) justapostas (b) soldadas, (c) linha de solda.	101

LISTA DE TABELAS

Tabela 3.1	Exemplo de representação de pontos de interesse.	44
Tabela 3.2	Descrição de linhas de borda por maior comprimento.	59
Tabela 3.3	Vetor com coordenadas de linha de solda e suas distâncias.	63

LISTA DE SIGLAS E ABREVIATURAS

BEM	<i>Binary Edge Map</i>
CCD	<i>Charge-Coupled Device</i>
CMOS	<i>Complementary Metal-Oxide-Semiconductor</i>
CLP	Controladores Lógicos Programáveis
FILO	<i>First Input Last Output</i>
IFR	<i>International Federation of Robotics</i>
LS&TC	Laboratório de Soldagem & Técnicas Conexas
PEL	<i>Predictive Edge Linking</i>
PROMECC	Programa de Pós-graduação em Engenharia Mecânica
RIA	<i>Robotic Industries Association</i>
ROI	<i>Region of Interest</i>
UFRGS	Universidade Federal do Rio Grande do Sul
RGB	<i>Red Green Blue</i>
CMY	Ciano, Magenta e Amarelo
CMYK	Ciano, Magenta, Amarelo e Preto
HSI	Matiz, Saturação e Intensidade

LISTA DE SÍMBOLOS

A	Valor de intensidade máxima
a	Parâmetro para limiar automático de detector de borda
b	Parâmetro para limiar automático de detector de borda
B_{4c}	Elemento para segmentação de 4 vizinhanças.
B_{bi}	Matriz binaria de bordas iniciais
B_c	Matriz de imagem em tons de cinza
B_{els}	Matriz binaria estimada de linha de solda
B_{ls}	Matriz binaria de linha de solda
C_x	Coordenada x da origem do plano da imagem, pixels
C_y	Coordenada y da origem do plano da imagem, pixels
$cont_{line}$	Contador para medição de comprimento de linhas de pixels de borda
d_{line}	Distancia entre linhas de bordas paralelas, pixels
d_{pixels}	Distancia entre pixels, pixels
d_x	Coordenada x de distancia de centro a centro de sensores da câmera, mm
d_y	Coordenada y de distancia de centro a centro de sensores da câmera, mm
f_c	Distância focal da câmera, mm
k_l	Coefficiente de distorção
K_{cos}	Variável ajustável da técnica K -cosine
n_{acc}	Acumulador para procura de pixel de borda
n_{line}	Indicador de número de linha em mapa de bits binário
N_{cx}	Número de elementos sensores na direção x
N_{cy}	Número de elementos sensores na direção y
N_{ls}	Número de pontos de linha de solda detectados pelo método
N_{lsi}	Número de pontos de linha de solda considerados ideais
O_{cc}	Origem da câmera, mm
O_i	Origem do plano da imagem, mm
P_d	Ponto P referenciado distorcido no plano da imagem, mm
P_{oz}	Coordenada z do ponto P referido ao plano da imagem, mm

r_{circ}	Radio de circunferência de estrutura para erosão, pixels
R	Matriz de rotação
r_1	Componente 1 da matriz de rotação
r_2	Componente 2 da matriz de rotação
r_3	Componente 3 da matriz de rotação
r_4	Componente 4 da matriz de rotação
r_5	Componente 5 da matriz de rotação
r_6	Componente 6 da matriz de rotação
r_7	Componente 7 da matriz de rotação
r_8	Componente 8 da matriz de rotação
r_9	Componente 9 da matriz de rotação
s_x	Fator de incerteza na escala da imagem
t_{low}	Limiar mínimo para detector de bordas Canny
t_{high}	Limiar máximo para detector de bordas Canny
x_{av}	Coordenada x de pixel de borda avaliado, pixels
x_{bnew}	Coordenada x de pixel de borda da media de dois anteriores, pixels
x_b	Coordenada x de pixel de borda, pixels
x_{cc}	Coordenada x do plano da câmera, mm
x_{ci}	Coordenada x do plano da imagem, mm
x_c	Coordenada x de pixel de canto, pixels
x_f	Coordenada x de pixel de descontinuidade de linha de solda, pixels
x_{fw}	Coordenada x de centro de ROI sobre descontinuidade de linha, pixels
x_{hl}	Coordenada x de pixel de solda sobre linha de Hough, pixels
x_{ini}	Coordenada x de pixel de inicio de linha de borda binaria, pixels
x_{lp}	Coordenada x de pixel de linha paralela detectada, pixels
x_{middle}	Coordenada x da metade entre duas linhas paralelas, pixels
x_w	Coordenada x referida ao sistema global, mm
x_{cd}	Coordenada x da imagem distorcida, pmm
X_f	Coordenada x na imagem do computador, pixels
x_{co}	Coordenada x não distorcida referida ao plano da imagem, mm
y_{av}	Coordenada y de pixel de borda avaliado, pixels
y_{bnew}	Coordenada y de pixel de borda apresentando a media, pixels

y_b	Coordenada y de pixel de borda, pixels
y_c	Coordenada y de pixel de canto, pixels
y_{cc}	Coordenada x da câmera, pixels
y_{ci}	Coordenada x do plano da imagem, pixels
y_f	Coordenada y de pixel de descontinuidades de linha de solda, pixels
y_{fw}	Coordenada y de centro de ROI sobre descontinuidades de linha, pixels
y_{hl}	Coordenada y de pixel de solda sobre linha de Hough, pixels
y_{ini}	Coordenada y de pixel de inicio de linha de borda binaria, pixels
y_{lp}	Coordenada y de pixel de linha paralela detectada, pixels
y_{middle}	Coordenada y da metade entre duas linhas paralelas, pixels
y_w	Coordenada y referida ao sistema global, mm
y_{cd}	Coordenada y da imagem distorcida, mm
Y_f	Coordenada y na imagem do computador, pixels
y_{co}	Coordenada y não distorcida referida ao plano da imagem, mm
z_{cc}	Coordenada z referida na câmera, mm
z_w	Coordenada z referida ao sistema global, mm
μ_{int}	Valor médio da probabilidade normalizada da densidade de intensidade
μ_{lines}	Valor médio das distancias entre pixels de linhas paralelas, pixels
μ_{line_new}	Valor médio das distancias entre pixels de linhas erosionadas, pixels
σ_{int}	Desvio padrão da probabilidade normalizada da densidade de intensidade
σ_{lines}	Desvio padrão das distancias entre pixels de linhas paralelas, pixels
σ_{line_new}	Desvio padrão das distancias entre pixels de linhas erosionadas, pixels
α_{PEL}	Ângulo para sentido de procura de linhas paralelas, baseado em passos do algoritmo PEL, radianos

1. INTRODUÇÃO

Para o projeto de um sistema de soldagem robotizado, deve-se considerar além do conhecimento básico do processo de soldagem, também a definição da trajetória a ser executada pelo braço manipulador. Na soldagem a união de dois materiais pode ser realizada através da fusão dos mesmos em íntimo contato, ou pela fusão de ambos com adição de outro material fundido, conduzido manualmente ou automatizado por um braço robótico.

Para a fusão dos materiais, estes devem possuir uma área em contato ou estarem muito próximos. Essa região de contato define uma linha, que indica a trajetória por onde deve ser conduzida a tocha para realizar a operação de soldagem.

Existem diferentes tipos de robôs para diversas tarefas, onde uma das principais áreas são os robôs que desenvolvem a soldagem. Na indústria há uma preocupação crescente em aumentar a lucratividade, observando os custos de produção, evitando desperdícios, principalmente em áreas como a soldagem. [Rodas et al., 2014]

Na atualidade a maioria dos robôs de soldagem são programados pelo método de ‘ensinar e reproduzir’ (*‘teach and playback’*), o que limita sua eficiência e flexibilidade [Shi et al., 2009]. Para que um soldador, ou um operador de robô de solda, possa determinar o parâmetro da linha de solda, a ferramenta mais importante é a sua visão. Com a mesma, o operador humano consegue posicionar a tocha em uma posição exata e determina o caminho a ser feito (linha de solda). Assim pode-se melhorar a autonomia do sistema de solda robótica, somando ao mesmo uma nova ciência, o processamento de imagens. [Chen e Chen, 2010]

A visão computacional tem uma área de desenvolvimento avançada em soldagem, focada em melhorar as características da solda em tempo real ou o seguimento de trajetória no momento em que o robô já está executando a mesma, [Zou et al., 2011], [Xu et al., 2004]. Porém, existem também métodos focados em detectar a trajetória de soldagem em sua totalidade antes de iniciar a solda, [Shi et al., 2009], [Chen e Chen, 2010], [Dinham e Fang, 2013]. Este último tipo de método de detecção de linha de solda será pesquisado neste trabalho.

Diante deste quadro, no Laboratório de Soldagem & Técnicas Conexas (LS&TC) da Universidade Federal do Rio Grande do Sul (UFRGS), pretende-se pesquisar na área de visão robótica incorporada a um robô de solda Yaskawa Motoman MA-1400. Assim, o desenvolvimento do presente trabalho, visa integrar o sistema de visão no robô para

determinar a trajetória de deposição de material sem intervenção humana por meio de programação dos pontos de solda.

1.1 Descrição do Problema

A soldagem é um dos processos fundamentais nas indústrias atuais, onde a formação profissional tem como requisito muitos anos de experiência e treinamento. Esta apresenta também características perigosas para o corpo humano, como a geração de calor, fumos perigosos e arcos elétricos. [Dinham e Fang, 2013]

As desvantagens citadas no parágrafo anterior e o incremento da demanda no mercado global competitivo aumentaram as oportunidades para a soldagem robotizada. Porém em muitos casos os robôs de solda são programados manualmente por métodos ‘ensinar e reproduzir’ (*‘teach and playback’*), que demanda um tempo considerável e caro na programação de trajetórias. Em casos de produção em massa este tempo é um problema importante. [Dinham e Fang, 2013]

Com o uso da visão computacional incorporada em um sistema de soldagem robótico, é possível obter uma flexibilidade e inteligência para ser aproveitada mais amplamente. A principal diferença entre um sistema robótico geral e um sistema robótico inteligente é que o último possui a habilidade de auto adaptação e auto determinação [Chen e Chen, 2010]. Em soldagem, a visão computacional pode ser utilizada para a identificação e localização da linha da solda, planejando a trajetória de soldagem automaticamente.

Assim como o sistema de visão computacional é uma opção para a solução da automação da trajetória de soldagem, o mesmo apresenta desafios únicos no campo da soldagem. Imperfeições, reflexões e riscos na superfície dos materiais podem dificultar consideravelmente ao sistema [Dinham e Fang, 2013]. Além disso o método tem que apresentar dinamismo frente a diferentes configurações de junta de topo, como a possibilidade de linhas de solda não retas, diferentes tamanhos de peças e cantos não coincidentes (quinas desfasadas).

O objetivo do presente trabalho é implementar um sistema para identificar trajetórias de soldagem em juntas de topo, utilizando sistemas de visão computacional. Para tanto, são propostos dois métodos: o primeiro para detecção de linhas retas de solda, enquanto o segundo método permite a detecção de pontos em linhas retas e curvas de solda. As trajetórias

identificadas devem ser executadas por um manipulador robótico associado a equipamentos de soldagem MIG/MAG.

1.2 Objetivo do Trabalho

O objetivo deste trabalho é desenvolver um sistema de visão incorporado num sistema de soldagem robótico com as seguintes características:

- utilizar uma câmera fixa que obtenha uma imagem de duas peças formando uma junta de topo a ser soldada no espaço de trabalho do robô;
- calibrar a câmera com um método de calibração eficiente;
- minimizar a intervenção humana;
- detectar a linha de solda em juntas topo;
- possuir flexibilidade para detectar a trajetória de solda com peças em diferentes posições e ângulos;
- possuir flexibilidade para detectar diferentes trajetórias como linha de solda não reta;
- possuir flexibilidade para detectar a trajetória em caso dos cantos das peças não coincidirem;
- garantir o funcionamento do sistema com variações no tom de cinza do fundo da imagem;
- avaliação do método comparando os pontos de solda obtidos com os pontos considerados ideais.
- resultados experimentais demonstrando que o algoritmo guia uma linha de solda aceitável.

2. REVISÃO BIBLIOGRÁFICA

Na presente revisão bibliográfica são descritas as principais técnicas utilizadas para desenvolver um sistema de visão computacional para reconhecimento de uma trajetória aplicada ao processo de soldagem implementado neste trabalho. As técnicas estão relacionadas com a calibração, o processamento e a análise de imagens, implementados pelo conjunto de métodos para permitir detectar as coordenadas da linha de solda. Os algoritmos de processamento têm como objetivo definir a trajetória da tocha, instalada como efetuator de um robô de soldagem.

2.1 Fundamentação teórica

Esta seção apresenta uma descrição geral dos aspectos considerados importantes para desenvolvimento deste trabalho. Para tanto, é dividida em dois tópicos: técnicas de visão computacional e descrição dos robôs de soldagem. Nas mesmas são descritas algumas técnicas de processamento de imagens para tarefas de soldagem de robôs, utilizadas neste trabalho.

2.1.1 Visão Computacional

A sensação de visão tem um papel importante na vida dos primatas: permite que eles detectem as propriedades espaciais do ambiente que são necessárias para as tarefas cruciais de sobrevivência [Ma et al., 2004]. Os primatas usam a visão para diferentes tarefas com uma aparentemente facilidade. Então, por que deve ser tão difícil para um computador ‘ver’?

Antes de tudo é necessário ter um conceito claro do que significa para um computador poder ver. Atualmente uma câmera digital pode entregar uma quantidade de ‘frames’ para um computador, de um modo similar ao que a retina faz com o cérebro. Porém, cada frame é uma coleção de valores positivos que medem a quantidade de luz incidente em cada local (ou pixel) em uma superfície. Somente essa informação não é suficiente para obter a identificação do ambiente. [Ma et al., 2004]

Os humanos podem detectar uma fruta sem muita dificuldade. Consegue-se ver a mesma em um ponto de vista diferente, e seguir detectando a fruta. A mesma situação existe se mudar a quantidade de luz ou se cortar a fruta. Deve-se entender que em cada caso diferente, os valores dos pixels mudam; o qual incrementa consideravelmente a dificuldade da visão computacional. [Ma et al., 2004]

Para uma máquina ser dotada de visão, essa deve ter um sensor foto sensitivo (uma câmera digital), assim como um mecanismo de processamento, que permita extrair informação de uma leitura de um arranjo de sensores. Diversas técnicas para visão computacional permitem recuperar a forma em três dimensões e as aparências de diferentes objetos. Essas técnicas abordam condições de captura e eliminação de ruídos para processar uma imagem e extrair características necessárias da mesma ao processo em análise. [Szeliski, 2010]

2.1.1.1 Captura da imagem

A luz incidente em um sensor de imagem é recolhida por uma área de sensores ativa, gerando sinal integrado pelo tempo de exposição (denominado tempo do obturador), que logo passa para um conjunto de amplificadores. Os dois principais tipos de sensores usados em câmeras digitais são o Dispositivo de Carga Acoplada (*'Charge-Coupled Device'* ou CCD) e o arranjo de transistores de óxido de metal em silício (*'Metal Oxide On Silicon'* ou CMOS) [Szeliski, 2010]. Analiticamente uma imagem digital é apresentada como um mapa de brilho $I(x,y)$ de duas dimensões, onde o tamanho do arranjo é discreto e inteiro e o termo pixel descreve cada elemento de brilho dentro do arranjo, tomando valores positivos, geralmente no intervalo de inteiros de [0,255]. [Ma et al., 2004]

As cores que os humanos e outros animais percebem de um objeto são determinadas pela natureza da luz refletida de um objeto. A luz visível é composta de uma banda de frequência no espectro eletromagnético relativamente estreito. [Gonzales e Woods, 2008]

A caracterização da luz é um tema central na ciência da cor. Se a luz é acromática (sem cor), o único atributo da mesma é a intensidade. Um exemplo de luz acromática são as imagens da televisão em branco e preto, assim o termo nível de cinza refere-se à medida escalar da intensidade da luz com o preto como mínimo valor, passando pelo cinza e finalmente chegando à cor branca. A luz cromática estende-se no espectro eletromagnético

desde 400 até 700 nm. Três valores básicos são usados para descrever a qualidade da luz acromática: radiação, luminância e brilho. Radiância é a quantidade de energia liberada de uma fonte de luz (geralmente medida em Watts). A luminância é a medida da quantidade de energia que um observador percebe de uma fonte de energia, medida em lumens (lm). O brilho é um descritor subjetivo praticamente impossível de medir que apresenta a noção acromática da intensidade, o mesmo também é um fator importante para descrever a sensação de cores. [Gonzales e Woods, 2008]

No olho humano, os sensores responsáveis pela visão em cores são denominados cones. Aproximadamente o 65% dos cones são sensíveis à luz vermelha, 33% à luz verde, e só 2% são sensíveis à luz azul. Na Figura 2.1 (adaptada de Gonzales e Woods, 2008) é ilustrada a resposta dos diferentes cones no olho humano.

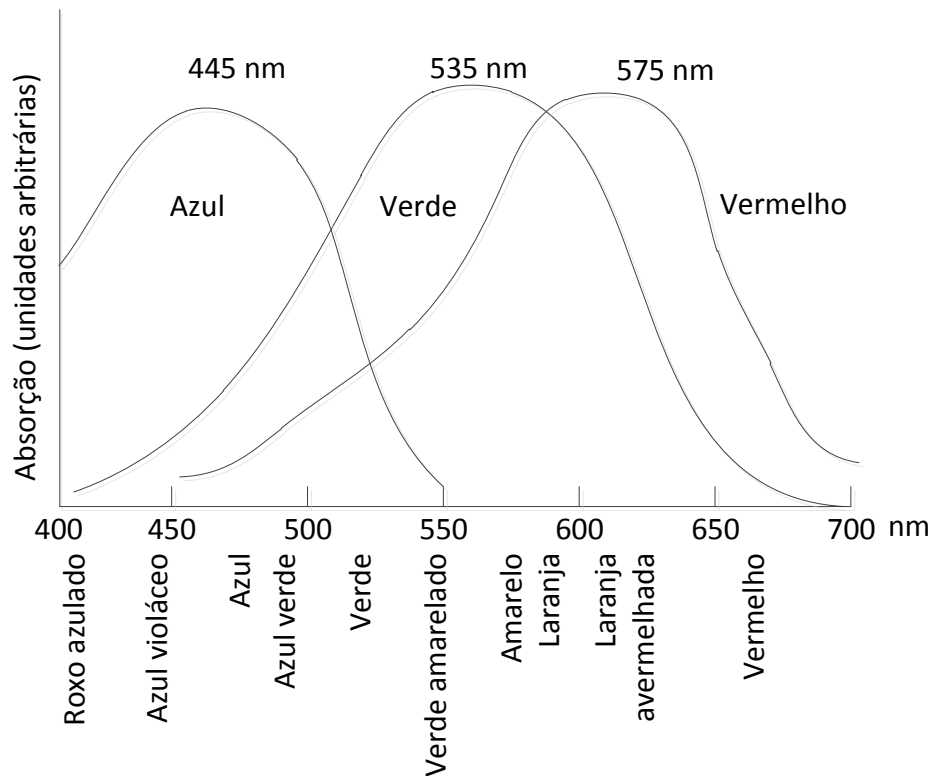


Figura 2.1 - Sensibilidade espectral dos 3 tipos de cones.

Devido a essas características de absorção do olho humano, as cores podem ser vistas como variáveis que dependem das combinações das cores primárias, vermelho (R), verde (G) e azul (B). Essas cores podem ser somadas para produzir as cores secundárias de luz,

magenta, ciano e amarelo, como apresentado na Figura 2.2 (adaptada de Gonzales e Woods, 2008), que é chamado de sistema aditivo.

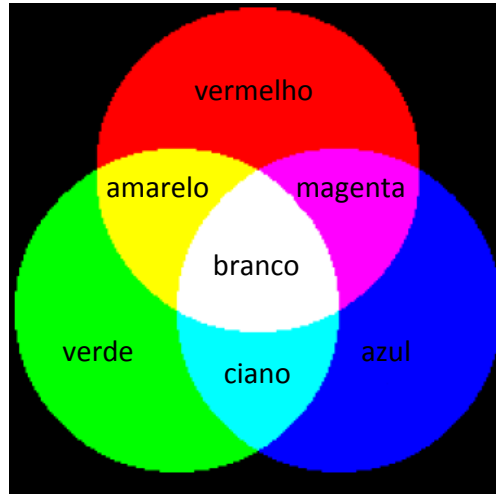


Figura 2.2 - Círculos representando as cores primárias do sistema aditivo.

Para especificar as cores em um padrão de uma imagem digital existe um sistema denominado modelo de cores. A maioria dos modelos de cores em uso são orientados em *hardware* (como para telas de cores ou impressoras) ou em aplicações onde o objetivo é a manipulação de cores. Em termos de *hardware*, os modelos mais comuns na prática são RGB (vermelho, verde e azul), para monitores em cores e câmaras de vídeo; CMY (ciano, magenta e amarelo) e CMYK (ciano, magenta, amarelo e preto) são modelos para imprimir em cores; e o modelo HSI (matiz, saturação e intensidade), corresponde aproximadamente à maneira com que os humanos descrevem e interpretam as cores. [Gonzales e Woods, 2008]

2.1.1.2 Técnicas de iluminação

A iluminação da cena é um fator chave na implementação de um algoritmo de detecção de objetos. Dependendo do tipo de iluminação aplicada sobre o objeto, a imagem apresenta diferentes características, como existência de sombras, reflexos, um bom ou ruim contraste entre fundo e objeto, entre outros.[Grassi, 2005] Dependendo da robustez do algoritmo, uma iluminação não desejada pode causar a completa falha do mesmo.

Três dos principais esquemas usados para iluminar um espaço de trabalhos de robô são apresentados na Figura 2.3 [adaptada de Grassi, 2005]. Uma técnica simples e utilizada neste trabalho é denominada iluminação direta, apresentada na Figura 2.3(a), onde as fontes de luz incidem diretamente acima do objeto. [Grassi, 2005]

A iluminação difusa, apresentada na Figura 2.3(b) é empregada para objetos lisos com superfícies regulares. Uma desvantagem da mesma é que gera sombra, razão pela qual preferencialmente deve ser evitada. O método apresentado na Figura 2.3(c), denominado iluminação traseira, apresenta as melhores características para o trabalho, já que praticamente gera uma imagem binária separando o objeto do fundo [Grassi, 2005]. Porém para soldagem, onde se utiliza uma bancada metálica, esse método não é aplicável.

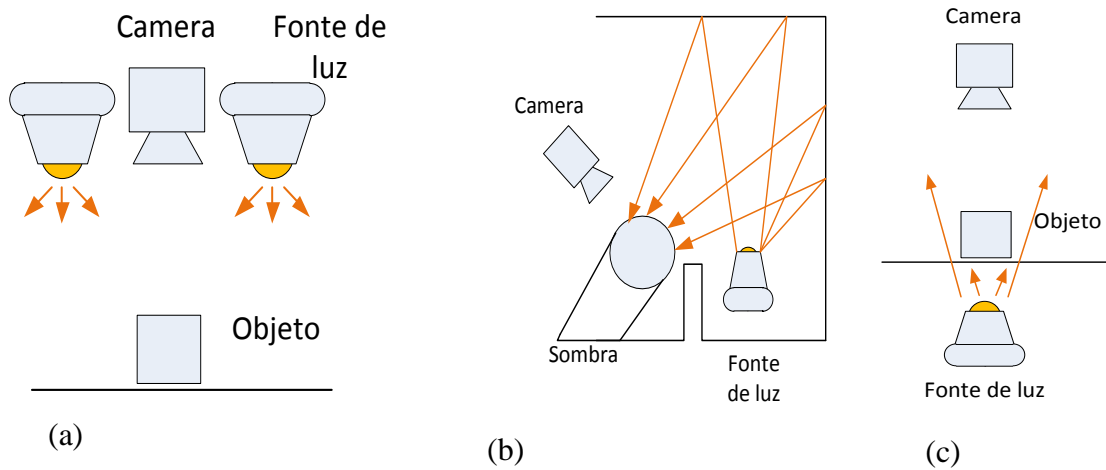


Figura 2.3 - Tipos de iluminação – (a) direta, (b) difusa , (c) traseira.

2.1.1.3 Processamento de imagem

O processamento de uma imagem corresponde a um tratamento digital de imagem em um computador. Não há uma definição clara quando se refere onde o processamento de imagem termina e onde iniciam outras técnicas como análise de imagens e visão computacional. Às vezes uma distinção é feita com a definição de processamento de imagem como um processo onde a entrada e saída são imagens. Por outro lado, têm-se campos como a visão computacional, onde o objetivo é o uso de computadores para emular a visão humana, incluindo aprendizagem e permitindo assim gerar ações baseadas em entradas visuais. A área de análise de imagem se situa entre o processamento de imagens e a visão computacional. [Gonzales e Woods, 2008]

2.1.1.4 Detecção de bordas

A detecção de bordas tem como finalidade detectar mudanças bruscas de intensidade entre vizinhanças de pixels dentro da imagem [Sha, 1997]. A maioria dos detectores de bordas consistem em três etapas: filtragem para redução de ruído, detecção de pontos de bordas e localização de bordas. A etapa de detecção de bordas extrai da imagem os pontos possíveis a serem candidatos a pontos de borda. A etapa de localização de bordas tem como função escolher os candidatos de borda que são os verdadeiros membros do conjunto de pixels que pertencem às bordas. [Gonzales e Woods, 2008]

A descrição básica da etapa de detecção de bordas, pode ser entendida analisando a derivada de uma função f_i unidimensional e discreta com variável independente x . Como em uma imagem o mínimo valor de Δx é de valor unitário, a derivada da função f_i pode ser escrita como a equação (2.1):

$$f' = \frac{df}{dx} = f(x) - f(x-1) \quad (2.1)$$

Para uma função discreta a derivada pode ser computada aplicando uma máscara de dois elementos em f_i , obtendo valores picos em regiões de variações de intensidade. As imagens tem duas dimensões, o vector gradiente de $f_i(x,y)$ é (f_{ix}, f_{iy}) , onde f_{ix} é a derivada na

direção de x e f_{iy} é a derivada na direção y . O valor do gradiente possui magnitude M_i e direção θ_i como definido na equação (2.2):

$$\theta_i = \arctan\left(\frac{f_{iy}}{f_{ix}}\right)$$

$$M_i = \sqrt{f_{ix}^2 + f_{iy}^2}$$
(2.2)

Um valor de magnitude normalizado $N_i(x,y)$ entre 0 e 100 do gradiente $M_i(x,y)$ (com (x,y) a localização do pixel avaliado) pode ser definido como a equação (2.3) [Sha, 1997]

$$N_i(x,y) = \frac{M_i(x,y)}{\max(M_i(x,y))} \cdot 100$$
(2.3)

As bordas são detectadas aplicando um limiar T_i no gradiente normalizada $N_i(x,y)$ como definido na equação (2.4)

$$E_i(x,y) = \begin{cases} 1 & \text{se } N_i(x,y) > T_i \\ 0 & \text{caso contrario} \end{cases}$$
(2.4)

Sendo $E_i(x,y)$ uma imagem binária. [Sha, 1997]

Detecção de bordas com operadores de gradientes

Para obter o gradiente de uma imagem f_i é requerido computar as derivadas parciais (f_{ix}, f_{iy}) em cada pixel localizado na mesma. Como uma imagem digital é discreta, uma aproximação da derivada parcial em uma vizinhança ao redor de um ponto pode ser obtida pela equação (2.5) [Gonzales e Woods, 2008]

$$f_{ix} = \frac{\partial f_i(x,y)}{\partial f_{ix}} = f(x+1,y) - f(x,y)$$

$$f_{iy} = \frac{\partial f_i(x,y)}{\partial f_{iy}} = f(x,y+1) - f(x,y)$$
(2.5)

Quando uma direção de bordas diagonal é de interesse, uma máscara em duas dimensões é necessária. O operador de gradientes cruzados de Roberts, apresentado na Figura 2.5(a), [Roberts, 1965] foi um dos primeiros à usar uma máscara em duas dimensões com uma preferência diagonal.

Os operadores de tamanho 2x2 são simples conceitualmente, mas apresentam problemas no momento de computar bordas direcionais em máscaras que são simétricas ao redor do ponto central. O mínimo tamanho de matriz para trabalhar nesses casos é de 3x3

trabalhando com a natureza dos dados em posições opostas no redor do ponto central. As mais simples aproximações digitais para derivadas parciais usando uma mascara de 3x3 são dados pela equação (2.6). [Gonzales e Woods, 2008]

$$\begin{aligned} f_{ix} &= \frac{\partial f_i(x, y)}{\partial f_i x} = (z_{i7} + z_{i8} + z_{i9}) - (z_{i1} + z_{i2} + z_{i3}) \\ f_{iy} &= \frac{\partial f_i(x, y)}{\partial f_i y} = (z_{i3} + z_{i6} + z_{i9}) - (z_{i1} + z_{i4} + z_{i7}) \end{aligned} \quad (2.6)$$

Onde os valores z_{i1} até z_{i9} referem-se aos pixels localizados ao redor do pixel avaliado (z_{i5}) como apresentado na Figura 2.4. Os valores da equação (2.6) podem ser implementados em uma imagem filtrando f_i com duas máscaras apresentadas na Figura 2.5(b). Estas máscaras são conhecidas como operadores de Prewitt. [Prewitt, 1970]

$$\begin{bmatrix} z_{i1} & z_{i4} & z_{i7} \\ z_{i2} & z_{i5} & z_{i8} \\ z_{i3} & z_{i6} & z_{i9} \end{bmatrix}$$

Figura 2.4 - Matriz de tamanho 3x3

Pode ser demonstrado que usando um valor de 2 na localização central permite que a imagem também seja filtrada. As máscaras para implementação são apresentadas na Figura 2.5(c) e são denominadas operadores de Sobel. [Sobel, 1970]

$$\begin{aligned} &\begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} & \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix} & \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \\ &(a) & (b) & (c) \end{aligned}$$

Figura 2.5 - Operadores de gradientes clássicos – (a) Robert, (b) Prewit, (c) Sobel.

Detecção de bordas Canny

O método de Canny [Canny, 1986] propõe um operador de bordas que consiste na soma de quatro exponenciais complexos, e pode ser aproximado com a primeira derivada de uma Gaussiana. A detecção de bordas é obtida pela convolução de uma imagem com ruído, com uma máscara, marcando as bordas como os pixels que tenham o máximo valor da saída

na convolução. Canny especifica três critérios para a caracterização da saída do operador: [Sha, 1997]

1. *boa detecção*. baixa probabilidade de falhas na detecção de eixos de borda, e baixa probabilidade de detectar falsos pontos de borda;
2. *boa localização*. os pontos detectados como bordas, devem estar o mais próximo do centro da verdadeira borda;
3. *só um pixel na borda*. se são detectados mais de um pixel na mesma borda, só um deles é considerado verdadeiro;

Canny demonstra que existe um conflito entre os critérios 1 e 2, assim ele procura uma função que consiga o melhor resultado entre os dois critérios. Aplica o terceiro critério para eliminar múltiplas respostas, e assim achar uma largura de um pixel em cada borda detectada.

No detector de Canny, os passos a serem seguidos no algoritmo são os seguintes: [Gonzales e Woods, 2008]

- aplica-se um filtro gaussiano na imagem;
- calcula-se o ângulo e a magnitude do gradiente em cada pixel. (Isso pode ser feito com qualquer uma das máscaras apresentadas na Figura 2.5);
- utiliza-se a técnica de supressão *Non-Maximum*. Esta técnica remove pixels que não são considerados partes da borda, e o resultado final produz uma borda de um pixel de espessura;
- aplica-se a técnica de histereses. A mesma usa dois limiares sobre a magnitude dos gradientes, primeiramente tem-se o limiar inferior (t_{low}), que indica que todos os pixels abaixo deste valor, não são considerados como bordas. O limiar superior (t_{high}), onde todos os pixels com um valor de gradiente superior a t_{high} são considerados bordas. Os pixels com valores de gradiente entre os dois limiares, apenas são considerados bordas se ficam contíguos a um pixel que tenha um valor de gradiente superior ao t_{high} ;

2.1.1.5 Transformada de Hough

A transformada de Hough, proposta por Hough, 1962, é uma ferramenta que tem como objetivo a extração de características de um conjunto ou sistema de pixels, de uma imagem binária, que se comportam como padrões conhecidos. Tais padrões podem ser círculos, retas e outras diferentes formas dentro da imagem. Neste trabalho, a técnica será utilizada para detectar retas na imagem. As possibilidades de utilização da técnica são muito variadas, mas um exemplo clássico é unir linhas retas cortadas. [Sha, 1997]

O enfoque com o qual se trabalha para obter a forma da reta é usando uma função equivalente à função convencional da reta comum, mas fazendo sua equivalência com funções senoidais. A transformada de Hough apresenta a parametrização de uma reta com a equação (2.7). [Sha, 1997]

$$\rho = x \cdot \cos(\theta) + y \cdot \sin(\theta) \quad (2.7)$$

Onde x e y são as coordenadas do pixel da imagem, θ é o ângulo entre o eixo x e uma linha reta que intercepta perpendicularmente uma reta que passa pelo pixel, ρ é a magnitude da distância perpendicular entre a linha a ser detectada e a origem (0,0). A Figura 2.6 exemplifica a utilização da equação (2.7). [Sha, 1997]

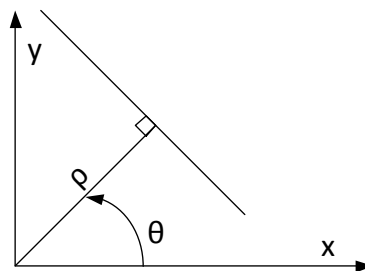


Figura 2.6 - Representação de Hough.

Supondo-se um conjunto de n pontos na imagem $[(x_1, y_1), \dots, (x_n, y_n)]$ e se procura detectar um conjunto de linhas retas que interceptam os mesmos. Com uso da equação (2.7) cada ponto é transformado ao plano (ρ, θ) . Toda curva correspondente com pontos colineares apresenta um ponto comum de interseção. Este ponto no plano (ρ, θ) , que pode ser (ρ_0, θ_0) , define os parâmetros da linha que passa a través dos pontos colineares. Por conseguinte, o

problema da detecção de pontos colineares pode ser convertido ao problema de detectar curvas concorrentes. [Duda e Hart, 1972]

Na prática o plano (ρ, θ) é quantizado em uma resolução de erro aceitável, sendo tratada como um arranjo de acumuladores em duas dimensões. Para cada ponto (x_i, y_i) no plano da imagem, a sua correspondente curva (ρ, θ) é inserida no arranjo incrementando em uma unidade cada componente ao longo da curva. Portanto cada membro do arranjo registra o número total de curvas passando no mesmo. Uma vez que todos os pontos foram tratados, o arranjo (ρ, θ) é registrado para detectar os pontos de maiores interseções, detectando assim os parâmetros das linhas que maiores pontos colineares apresentam. [Duda e Hart, 1972]

2.1.1.6 Algumas relações básicas entre pixels

Nesta seção serão consideradas algumas básicas, mas importantes relações entre pixels em uma imagem digital.

Vizinhanças de um pixel

Um pixel p nas coordenadas (x, y) tem 2 vizinhos horizontais e 2 vizinhos verticais, como apresentado nos pixels em tons de cinza na Figura 2.7, sendo suas coordenadas apresentadas na equação (2.8). [Fu et al. 1987].

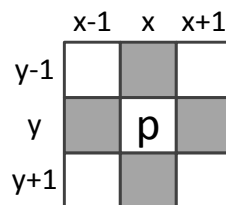


Figura 2.7 - Vizinhança-4 em um pixel.

$$\begin{aligned}
 &(x+1, y) \\
 &(x-1, y) \\
 &(x, y+1) \\
 &(x, y-1)
 \end{aligned}
 \tag{2.8}$$

O conjunto de coordenadas de pixels apresentado nas equações (2.8), é chamado de *vizinhança-4*, o qual é denotado por $N_4(p)$. Assim as coordenadas dos 4 vizinhos diagonais de

p , apresentados nos pixels em tons de cinza na Figura 2.8 da equação (2.9), tem as coordenadas que são denotadas por $N_D(p)$. [Fu et al. 1987].

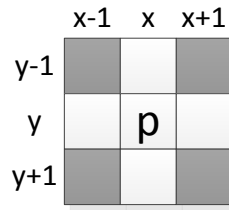


Figura 2.8 - Os quatro vizinhos diagonais de um pixel denotados por $N_D(p)$.

$$\begin{aligned}
 &(x+1, y+1) \\
 &(x+1, y-1) \\
 &(x-1, y+1) \\
 &(x-1, y-1)
 \end{aligned} \tag{2.9}$$

A combinação desses pontos, juntos com os pontos de vizinhança-4 definidos anteriormente, são denominados vizinhança-8 de p , denotado por $N_8(p)$. [Fu et al. 1987].

Medição de distâncias

Dados os pixels p , q e z , com coordenadas (x, y) , (s, t) e (u, v) respectivamente, é chamada D uma função de distância com as seguintes características: [Fu et al. 1987].

$$D(p, q) \geq 0$$

$$D(p, q) = D(q, p)$$

$$D(p, z) \leq D(p, q) + D(q, z)$$

A distância euclidiana entre p e q é definida por:

$$D(p, q) = \sqrt{(x-s)^2 + (y-t)^2} \tag{2.10}$$

Para medição de distância, os pixels que tem uma distância menor ou igual do que um valor r de (x, y) são pontos contidos em uma circunferência de raio r centrado em (x, y) . [Fu et al. 1987].

A distância D_4 entre p e q é definida como:

$$D_4(p, q) = |x-s| + |y-t| \tag{2.11}$$

Neste caso os pixels que tem uma distância D_4 menor ou igual a algum valor r de (x, y) formam um diamante centrado em (x, y) . Por exemplo, os pixels com uma distância D_4 que

seja menor ou igual à 2 com (x, y) , formam o contorno de distâncias com o ponto denotado por 0, mostrado na Figura 2.9:

$$\begin{array}{ccccc}
 & & & & 2 \\
 & & & & 2 & 1 & 2 \\
 & & & 2 & 1 & 0 & 1 & 2 \\
 & & & & 2 & 1 & 2 \\
 & & & & & & & & 2
 \end{array}$$

Figura 2.9 - Pixels com distância D_4 .

O valor de distância D_8 entre p e q é definida como:

$$D_8(p, q) = \max(|x-s|, |y-t|) \quad (2.12)$$

Neste caso os pixels com um valor de distância D_8 menor ou igual a algum valor r formam um quadrado centrado em (x, y) . Por exemplo, os pixels com uma distância D_8 que seja menor ou igual à 2, formam os contornos de distâncias iguais com o ponto denotado por 0, mostradas na Figura 2.10:

$$\begin{array}{ccccc}
 2 & 2 & 2 & 2 & 2 \\
 2 & 1 & 1 & 1 & 2 \\
 2 & 1 & 0 & 1 & 2 \\
 2 & 1 & 1 & 1 & 2 \\
 2 & 2 & 2 & 2 & 2
 \end{array}$$

Figura 2.10 - Pixels com distância D_8 .

2.1.1.7 Ligação de borda preditiva (*Predictive Edge Linking ou PEL*)

O algoritmo Ligação de Borda Preditiva (*Predictive Edge Linking ou PEL*) foi proposto por Akinlar e Chome, 2016. O mesmo recebe como entrada um Mapa de Imagem Binária (*Binary Edge Map ou BEM*) criada por um detector de bordas tradicional, e retorna um conjunto de segmentos como saída. Os autores descrevem o algoritmo em quatro etapas,

na primeira etapa são preenchidos saltos ou distâncias entre pixels de linhas. Na segunda etapa, os segmentos de bordas são criados. Na terceira etapa, os segmentos de borda que têm os seus finais de linha próximos uns dos outros são unidos para criar um segmento maior. Na quarta e última etapa, os segmentos de largura maior do que um pixel, são reduzidos a segmentos de largura de um pixel.

Neste trabalho será utilizada unicamente a etapa 2, denominada ‘Criar Segmentos’ do artigo que descreve o algoritmo PEL. [Akinlar e Chome, 2016]

Criar Segmentos: união de pixels contíguos para criação de correntes de pixels

Trata-se de um método heurístico que consiste na detecção de um pixel de borda arbitrário em uma imagem binária, e tendo em conta que o conjunto de pixels de borda forma linhas, os pixels unidos ao pixel arbitrário podem ser detectados em duas direções: uma na direção posterior e outra na direção reversa. Os conjuntos de pixels são combinados para criar só uma corrente ou conjunto, o qual permite que seja gerado um único segmento por linha de borda.

Para a criação de um segmento de borda, são realizados oito passos direcionais com predição. A predição é usada quando a direção atual muda. Considerando as oito direções possíveis, o motor de predição decide em que direção vai se mover após a direção ter mudado.

A Figura 2.11 [adaptada de Akinlar e Chome, 2016] ilustra o caminho em 4 direções, (a) Direita, (b) Abaixo, (c) Abaixo-Direita, (d) Abaixo-Esquerda. As outras 4 direções, Esquerda, Acima, Acima-Esquerda e Acima-Direita são simplesmente versões simétricas das direções anteriores.

Ilustrando como exemplo, iniciando na Figura 2.11(a), com um caminho na direção à direita. Isso é, no pixel com localização (x, y) , marcado com cor cinza escuro, um passo adiante do pixel $(x - 1, y)$, o pixel cor cinza claro. Por isso caracterizando como um movimento à ‘Direita’, assim imediatamente é testado o pixel localizado na direita $(x + 1, y)$, independentemente das predições dos movimentos anteriores. Em caso de não se encontrar um pixel de borda no pixel marcado com ‘1’, é feita uma revisão dos seguintes 6 vizinhos, tomando em conta a predição. Poderia ser testado em primeiro lugar o pixel localizado na posição ‘Direita-Abaixo’ $(x + 1, y + 1)$ ou poderia ser testado o da posição ‘Direita-Cima’ $(x + 1, y - 1)$. Para fazer essa predição é feita uma consulta no motor de predição, o qual, levando em conta os últimos oito movimentos, indica se a próxima direção em ser consultada seria o pixel de ‘direita-abaixo’ ou ‘direita-cima’. Na Figura 2.11(a) é detalhada a ordem em

que são testados os pixels vizinhos de (x, y) , dependendo da predição realizada. São verificados os pixels na ordem estabelecida, tão logo é detectado um pixel de borda é feito o deslocamento e a direção é mudada adequadamente.

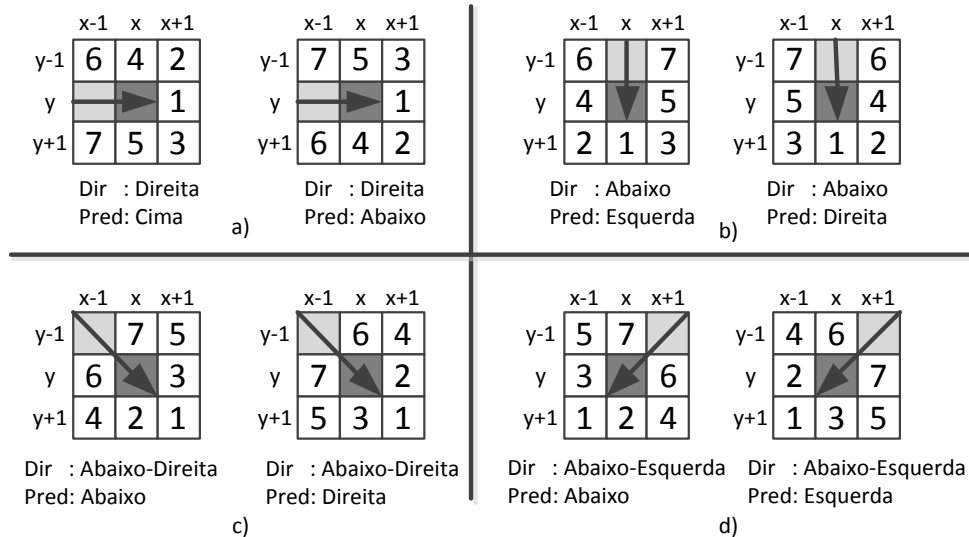


Figura 2.11 - Caminho em 4 direções com predição.

Para a criação de um segmento de borda, é realizada uma análise de oito direções preditiva. A predição é usada quando a direção atual muda, tendo em conta para a decisão do caminho a escolher, as últimas oito direções percorridas. Desta maneira o algoritmo poderia cursar uma linha de pixels e não ficar ‘atracado’ em caminhos de direções inesperadas que não tenham ‘saída’.

Finalmente, o algoritmo de segmentação de cada linha finaliza no momento em que não consiga detectar algum pixel de borda nas 7 direções testadas.

2.1.1.8 Calibração da Câmera

Para o robô poder seguir uma linha de solda de duas peças, ele deve acompanhar uma trajetória de pontos definidos no espaço referenciado a seu sistema de coordenadas. Uma imagem capturada e processada fornece pontos que são expressos em pixels, mas que deverá ser expresso em coordenadas do sistema cartesiano universal ou global [Chen et al. 2006]. O robô é configurado em função também de um sistema de coordenadas próprio (x_r, y_r, z_r) ,

sendo assim necessário estabelecer uma relação entre os pixels da imagem e os do sistema de coordenadas do robô. Essa correspondência é obtida por técnicas de calibração da câmera.

A calibração da câmera tem como objetivo obter duas classes de parâmetros da mesma, os parâmetros chamados de extrínsecos e os parâmetros denominados intrínsecos. Os parâmetros intrínsecos são os parâmetros internos da câmera, como a distância focal da câmera f_c , o coeficiente de distorção da lente k , o sistema de coordenadas da imagem para a origem do plano de imagem C_x, C_y , as distâncias de centro a centro entre pixels (dx, dy). Os parâmetros extrínsecos correspondem a posição no espaço 3D e a orientação da câmera em relação a um eixo de coordenadas globais (ver Figura 2.12). Neste trabalho o modelo de calibração de câmera aplicado é o método desenvolvido por Tsai, 1987.

No espaço de parâmetros extrínsecos, com 3 eixos angulares ortogonais e 3 posições de translação ortogonais, é possível descrever qualquer posição e rotação em relação a um eixo de coordenadas, com relação ao eixo de coordenadas globais (x_w, y_w, z_w em Figura 2.12). No método de calibração de câmera proposto por Tsai, a ideia principal é procurar por uma restrição ou equação que seja função só de um subconjunto de parâmetros de calibração, com isso é possível reduzir a dimensão do espaço de parâmetros desconhecidos. Essa restrição é denominada “princípio de restrição de alinhamento radial”. A mesma é função só da rotação relativa e da translação em x e em y , entre a câmera e os pontos de calibração. [Tsai, 1987]

Neste trabalho há só um parâmetro intrínseco a ser determinado (a distância focal f_c), por isso a calibração será feita utilizando um conjunto de pontos coplanares com distâncias conhecidas. Neste caso só um plano é utilizado, mas a quantidade de planos a serem determinados depende diretamente da quantidade de parâmetros intrínsecos a serem determinados pelo método. [Tsai, 1987]

Modelo da câmera

Nesta seção é descrito o modelo da câmera, que define os parâmetros de calibração e apresenta o princípio de restrição de alinhamento radial. Na Figura 2.12 são apresentados os eixos de coordenadas que descrevem o sistema de visão, incluindo o robô, a câmera e o sistema global. A câmera está centrada no ponto O_{cc} com os eixos x_{cc}, y_{cc}, z_{cc} . Em uma distância f_c , denominada distância focal, do centro da câmera, localiza-se o plano da imagem, com seus eixos x_{ci}, y_{ci} paralelos aos eixos da câmera x_{cc}, y_{cc} respectivamente. O sistema também apresenta as coordenadas globais representadas pelos eixos x_w, y_w, z_w e as coordenadas do robô com os eixos x_r, y_r, z_r .

O vector $P(x_w, y_w, z_w)$, descreve as coordenadas globais em 3D do ponto P . A localização $P(x_{cc}, y_{cc}, z_{cc})$ representa a coordenada em 3D do ponto P em relação ao sistema de coordenadas 3D da câmera. A câmera está centrada no ponto O_{cc} denominado centro ótico, e o eixo z_{cc} é denominado eixo ótico. O ponto $P_u(x_{co}, y_{co})$ é a coordenada de imagem de P se uma câmera perfeita estenopeica ('pinhole') é usada. $P_d(x_{cd}, y_{cd})$ é a coordenada do ponto na imagem real, a qual difere de $P_u(x_{co}, y_{co})$ devido às distorções das lentes. Porém, já que os valores $P(x_{cf}, y_{cf})$ são as coordenadas em pixels dentro da imagem, devem ser especificados parâmetros adicionais para relacionar as coordenadas da imagem no plano de imagem, com o sistema de coordenadas da imagem digital. As coordenadas do ponto P referidas ao robô são $P(x_r, y_r, z_r)$, que referidas com as coordenadas globais permitem ter a relação de posição entre a câmera e o robô. A seguir serão considerados 4 passos para transformar a coordenada do ponto expressa em coordenadas globais, em coordenadas expressas em pixels dentro da imagem digitalizada.

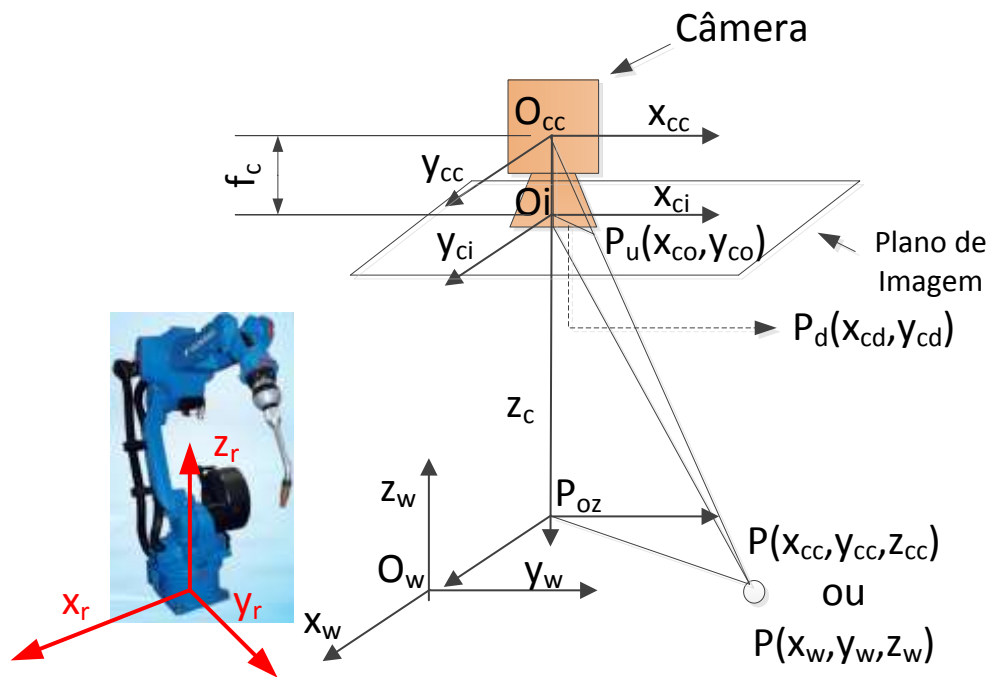


Figura 2.12 - Eixos de coordenadas que descrevem a geometria da câmera.

Passo 1

Primeiramente é necessário ter a localização de pontos conhecidos referidos a um sistema de coordenadas globais. Uma vez medidos esses pontos, as coordenadas de cada um dos mesmos, neste caso expressos como (x_w, y_w, z_w) , podem ser expressas em coordenadas 3D da câmera (x_{cc}, y_{cc}, z_{cc}) com a relação da equação (2.9).

$$\begin{bmatrix} x_{cc} \\ y_{cc} \\ z_{cc} \end{bmatrix} = R \begin{bmatrix} x_w \\ y_w \\ z_w \end{bmatrix} + T \quad (2.13)$$

Onde R é a matriz de rotação, que expressa a relação entre as rotações relativas entre a câmera e o sistema de coordenadas globais:

$$R = \begin{bmatrix} r_1 & r_2 & r_3 \\ r_4 & r_5 & r_6 \\ r_7 & r_8 & r_9 \end{bmatrix} \quad (2.14)$$

T é o vetor de translação, dos eixos x, y, z , entre a câmera e o sistema de coordenadas globais:

$$T = \begin{bmatrix} T_x \\ T_y \\ T_z \end{bmatrix} \quad (2.15)$$

Os parâmetros para ser calibrados são R e T .

Passo 2

Transformação do sistema de coordenadas 3D (x_{cc}, y_{cc}, z_{cc}) no ponto ideal no plano de imagem (x_{co}, y_{co}) , usando uma projeção perspectiva com uma geometria de câmera *estenopeica*.

$$\begin{aligned} x_{co} &= f_c \frac{x_{cc}}{z_{cc}} \\ y_{co} &= f_c \frac{y_{cc}}{z_{cc}} \end{aligned} \quad (2.16)$$

Onde o parâmetro para o ponto calibrado é a distância focal f_c .

Passo 3

Uma distorção de lentes radial é considerada. Para corrigir estas coordenadas, um valor escalar é somado nas coordenadas x_{co}, y_{co} como expresso na equação (2.17):

$$\begin{aligned}x_{cd} + D_x &= x_{co} \\y_{cd} + D_y &= y_{co}\end{aligned}\quad (2.17)$$

Onde (x_{cd}, y_{cd}) é a coordenada distorcida do ponto no plano de imagem e os valores a serem adicionados são:

$$\begin{aligned}D_x &= x_{cd}(k_1 r^2 + \dots) \\D_y &= y_{cd}(k_1 r^2 + \dots) \\r &= \sqrt{x_{cd}^2 + y_{cd}^2}\end{aligned}\quad (2.18)$$

O parâmetro a ser calibrado é o coeficiente de distorção k_1 . Nas câmeras, têm-se duas classes de distorções de lentes, radial e tangencial. Por experiência é demonstrado que para aplicações em visão computacional, só é considerada a distorção radial, e só vai ser preciso um termo [Tsai, 1987].

Passo 4

Transformação de coordenada de imagem real (x_{cd}, y_{cd}) em sistema de coordenadas imagem no computador (x_{cf}, y_{cf}) .

$$\begin{aligned}x_{cf} &= s_x d_x'^{-1} x_{cd} + C_x \\y_{cf} &= d_y'^{-1} y_{cd} + C_y\end{aligned}\quad (2.19)$$

Onde (C_x, C_y) são as coordenadas em linha e coluna do pixel do centro da imagem.

$$d_x' = d_x \frac{N_{cx}}{N_{fx}}\quad (2.20)$$

Os valores d_x são as distâncias de centro a centro entre os elementos sensores na direção x_{cc} , d_y é a distância centro a centro dos elementos sensores na direção y_{cc} . N_{cx} é o número de elementos sensores na direção x_{cc} , e N_{fx} é o número de pixels em uma linha que é amostrada pelo computador.

Um parâmetro a ser calculado é o fator de incerteza na escala da imagem s_x . O qual é incorporado como um fator de escala para contabilizar alguma incerteza devido a imperfeições da imagem. [Tsai, 1987]

Características da técnica descrita por Tsai, 1987

Segundo descrito pelo Tsai, 1987, a técnica de calibração de câmera proposta é baseada em 4 premissas.

Premissa 1 Como é considerada só a distorção radial, não importa qual é o valor da distorção, a direção do vetor $\overline{O_i P_d}$ da Figura 2.12, a qual se estende da origem O_i no plano da

imagem até o ponto de imagem $P_d(x_{cd}, y_{cd})$ não apresenta mudanças. A mesma está alinhada com o vetor $\overline{P_{oz}P}$, o qual se estende do eixo ótico até o ponto do objeto (x_{cc}, y_{cc}, z_{cc}) , (ver Figura 2.12).

Premissa 2 A variação na distância focal f_c não muda a direção do vetor $\overline{O_iP_d}$, já que f_c escala as coordenadas do plano de imagem x_{cd} e y_{cd} na mesma quantidade.

Premissa 3 A translação no eixo z_{cc} não muda a direção de $\overline{O_iP_d}$, já que uma alteração em z muda x_{co} e y_{co} na mesma escala.

Premissa 4 A restrição de que $\overline{O_iP_d}$ seja paralela com $\overline{P_{oz}P}$, para cada ponto, independente dos coeficientes de distorção radial k_l , a distância focal f_c , e o componente z_{cc} do vetor de translação 3D T , é suficiente para determinar a rotação 3D R , e os componentes x_{cc} e y_{cc} do vetor de translação em 3D, do sistema de coordenadas globais ao sistema de coordenada da câmera.

Calibração da câmera usando um conjunto de pontos coplanares

A seguir descreve-se a calibração da câmera com uma quantidade de pontos que compartilham um mesmo plano, por isso a coordenada z_w de cada ponto pode ter o valor de $z_w=0$. [Tsai, 1987]

Etapa 1 Computar a orientação 3D, e posição (x_{cc}, y_{cc}) :

a) computar as coordenadas da imagem distorcida (x_{cd}, y_{cd}) :

- detectar os pontos em linha e coluna de cada pixel que representa um ponto de calibração. (x_{cfi}, y_{cfi}) .
- obter os valores de $Ncx, Nfx, d'x, dy$ usando informação do fabricante da câmera.
- usar (C_x, C_y) como o pixel do centro da memória.
- obter (x_{cdi}, y_{cdi}) com as seguintes fórmulas matemáticas:

$$\begin{aligned} x_{cdi} &= s_x^{-1} d'_x (x_{cfi} - C_x) \\ y_{cdi} &= d_y (y_{cfi} - C_y) \end{aligned} \quad (2.21)$$

Para $i = 1, \dots, N$, onde N corresponde com o número de pontos de calibração.

b) obter as cinco incógnitas $T_y^{-1}r_1, T_y^{-1}r_2, T_x^{-1}T_x, T_y^{-1}r_4, T_y^{-1}r_5$:

Para cada ponto i com $(x_{wi}, y_{wi}, z_{wi}), (x_{cdi}, y_{cdi})$, trabalha-se com as seguintes incógnitas:

$$\begin{bmatrix} y_{cdi}x_{wi} & y_{cdi}y_{wi} & y_{cdi} & -x_{cdi}x_{wi} & -x_{cdi}y_{wi} \end{bmatrix} \begin{bmatrix} T_y^{-1}r_1 \\ T_y^{-1}r_2 \\ T_y^{-1}T_x \\ T_y^{-1}r_4 \\ T_y^{-1}r_5 \end{bmatrix} = x_{cdi} \quad (2.22)$$

Com N (número de pontos do objeto) muito maior que cinco, um sistema sobre determinado de equações lineares pode ser estabelecido e resolvido para as cinco incógnitas procuradas.

c) obter $(r_1, \dots, r_9, T_x, T_y)$ de $(T_y^{-1}r_1, T_y^{-1}r_2, T_y^{-1}T_x, T_y^{-1}r_4, T_y^{-1}r_5)$:

• obter $|T_y|$ de $T_y^{-1}r_1, T_y^{-1}r_2, T_y^{-1}T_x, T_y^{-1}r_4, T_y^{-1}r_5$:

Com C uma submatriz de 2x2 da matriz de rotação R , definida como:

$$C = \begin{bmatrix} r_1' & r_2' \\ r_4' & r_5' \end{bmatrix} = \begin{bmatrix} r_1/T_y & r_2/T_y \\ r_4/T_y & r_5/T_y \end{bmatrix} \quad (2.23)$$

Se não há uma coluna ou linha com valores em zero, computar o valor de T_y^2 como:

$$T_y^2 = \frac{S_r - \left(S_r^2 - 4(r_1'r_5' - r_4'r_2')^2 \right)^{1/2}}{2(r_1'r_5' - r_4'r_2')} \quad (2.24)$$

Com $S_r = r_1'^2 + r_2'^2 + r_4'^2 + r_5'^2$; em caso contrário (tendo uma linha ou coluna completamente com valores em zero), o valor de T_y^2 é:

$$T_y^2 = (r_i'^2 + r_j'^2)^{-1} \quad (2.25)$$

Onde r_i', r_j' são os elementos da linha ou coluna de C que não estão zerados.

• determinação do sinal de T_y :

Escolher um ponto i com coordenada de imagem no computador (x_{cfi}, y_{cfi}) que esteja localizado distante do centro (C_x, C_y) ; a coordenada global do objeto é (x_{wi}, y_{wi}, z_{wi}) . Escolher o sinal de T_y como +1 e computar as seguintes fórmulas:

$$\begin{aligned}
r_1 &= (T_y^{-1} r_1) T_y \\
r_2 &= (T_y^{-1} r_2) T_y \\
r_4 &= (T_y^{-1} r_3) T_y \\
T_x &= (T_y^{-1} T_x) T_y \\
x &= r_1 x_w + r_2 y_w + T_x \\
y &= r_4 x_w + r_5 y_w + T_y
\end{aligned} \tag{2.26}$$

Sendo que x e x_{cc} tem o mesmo sinal e y e y_{cc} tem o mesmo sinal, então o sinal de T_y é +1, em caso contrário $sign(T_y) = -1$.

- calcular a matriz de rotação em 3D, ou r_1, \dots, r_9 :

Obter r_1, r_2, r_4, r_5 e T_x com a equação (2.26) para criar a matriz R segundo estabelecido na equação (2.27).

$$R = \begin{bmatrix} r_1 & r_2 & (1 - r_1^2 - r_2^2)^2 \\ r_4 & r_5 & s(1 - r_4^2 - r_5^2)^2 \\ r_7 & r_8 & r_9 \end{bmatrix} \tag{2.27}$$

Com $s = sign(r_1 r_4 + r_2 r_5)$, e os valores de r_7, r_8, r_9 são obtidos com a regra da mão direita sendo a matriz de rotação uma matriz ortonormal.

Etapa 2: Obtenção da distância focal f_c , o coeficiente de distorção k_l e a posição z_{cc}

a) obter o valor aproximado da distância focal f_c com a seguinte equação:

$$\begin{bmatrix} y_{ci} & -d_y Y_i \end{bmatrix} \begin{bmatrix} f_c \\ T_z \end{bmatrix} = w_i d_y Y_i \tag{2.28}$$

Com $y_i = r_4 x_{wi} + r_5 y_{wi} + T_y$ e $w_i = r_7 x_{wi} + r_8 y_{wi}$. Com $f_c > 0$ o valor de R é o valor da equação (2.27); em caso de $f_c < 0$, o valor de R é:

$$R = \begin{bmatrix} r_1 & r_2 & -(1 - r_1^2 - r_2^2)^2 \\ r_4 & r_5 & -s(1 - r_4^2 - r_5^2)^2 \\ -r_7 & -r_8 & r_9 \end{bmatrix} \tag{2.29}$$

b) calcular um valor aproximado de f_c e T_z :

Para cada ponto de calibração, usar a equação linear (2.28) para f_c e T_z . Ainda é um valor aproximado porque é considerada distorção k_l com valor zerado.

c) calcular o valor exato de f_c, T_z, k_l :

Com o valor inicial de f_c e T_z obtido e $k_l = 0$, usar a seguinte equação com um esquema de otimização padrão para obter os valores corretos:

$$d'_y Y + d_y Y k_1 r^2 = f \frac{r_4 x_w + r_5 y_w + r_3 z_w + T_y}{r_7 x_w + r_8 y_w + r_9 z_w + T_z} \quad (2.30)$$

Segundo Tsai, 1987, o valor correto geralmente é obtido com uma ou duas iterações.

2.1.2 Sistemas de robôs de solda industriais

Nesta subseção é apresentada uma descrição básica de um robô, incluindo o significado da palavra, uma descrição de seus componentes e suas configurações típicas. A seguir é feita uma básica descrição de um sistema de soldagem, e finalmente, com esses conhecimentos incorporados, vai ser descrito o que é entendido como um robô dedicado na área de soldagem.

2.1.2.1 Breve introdução à robótica

O termo robô foi originalmente utilizado em 1921 pelo dramaturgo checo Karen Capek, na peça teatral “Os Robôs Universais de Russum (R.U.R.)” como referência a um autômato que acaba rebelando-se contra o ser humano. Robô deriva da palavra "robota" de origem eslava, que significa "trabalho forçado".

Na década de 40, o escritor Isaac Asimov tornou popular o conceito de robô como uma máquina de aparência humana não possuidora de sentimentos, onde seu comportamento seria definido a partir da programação feita por seres humanos, de forma a cumprir determinadas regras éticas de conduta. O termo robótica foi criado por Asimov para designar a ciência que se dedica ao estudo dos robôs e que se fundamenta pela observação de três leis básicas [Siciliano, 1995]:

1ª. Um robô não pode fazer mal a um ser humano e nem consentir, permanecendo inoperante, que um ser humano se exponha a situação de perigo;

2ª. Um robô deve obedecer sempre às ordens de seres humanos, exceto em circunstâncias em que estas ordens entrem em conflito com a 1ª lei;

3ª. Um robô deve proteger a sua própria existência, exceto em circunstâncias que entrem em conflito com a 1ª e 2ª leis.

2.1.2.2 Definição de Robô

Segundo a *Robotic Industries Association* (RIA), robô industrial é definido como um "manipulador multifuncional reprogramável projetado para movimentar materiais, partes, ferramentas ou peças especiais, através de diversos movimentos programados, para o desempenho de uma variedade de tarefas" (Rivin, 1988).

Um robô industrial é formado pela integração dos seguintes componentes.

- a) manipulador mecânico: Descreve o aspecto mecânico e estrutural do robô. Consiste da combinação de elementos estruturais rígidos (corpos ou elos) conectados entre si através de articulações (juntas), sendo o primeiro corpo denominado base e o último denominado extremidade terminal, onde será vinculado o componente efetuator (garra ou ferramenta).
- b) atuadores: Componentes que convertem energia (pode ser elétrica, hidráulica ou pneumática) em energia mecânica. Geralmente são usados sistemas de transmissão para enviar a potência gerada nos atuadores até os elos, e assim gerar o movimento.
- c) sensores: Fornecem parâmetros sobre o comportamento do manipulador, geralmente em termos de posição e velocidade dos elos em função do tempo, e do modo de interação entre o robô e o ambiente operativo (força, torque, sistema de visão) à unidade de controle. As juntas utilizadas para vincular os elos de um robô são normalmente acopladas a sensores.
- d) unidade de controle: Responsável pelo controle e monitoramento dos parâmetros requeridos para realizar as tarefas solicitadas ao robô. Tem um sistema de gerenciamento (computador industrial, CLP) que recebe as informações, processa os valores recebidos pelos sensores, e origina o movimento requerido no robô.
- e) unidade de potência: responsável pelo fornecimento de potência necessária à movimentação dos atuadores. A bomba hidráulica, o compressor e a fonte elétrica são as unidades de potência associadas aos atuadores hidráulicos, pneumáticos e eletromagnéticos, respectivamente.
- f) efetuator: É o elemento de ligação entre o robô e o meio que o cerca. Exemplos podem ser tipo garra ou ferramenta; como no caso específico, efetuator é a tocha de soldagem, com tarefa de efetuar a soldagem de duas ou mais peças, com juntas de topo.

2.1.2.3 Sistema de soldagem

Na definição de solda pelo *Welding Handbook*, 1987, cita: “Solda: uma coalescência localizada de metais ou não-metais, produzida ou pelo aquecimento dos materiais até a temperatura de soldagem, com ou sem a aplicação de pressão, ou pela aplicação de pressão somente, com ou sem o uso de metal de adição”. [Machado, 2007]

Tecnologia MIG/MAG

O processo MIG/MAG (*Metal Inert Gas/Metal Active Gas*), também conhecido como processo de soldagem por arco elétrico com gás de proteção (GMAW) usa o calor do arco elétrico para fundir um arame-eletrodo continuamente alimentado e os componentes metálicos a serem soldados. A fusão é realizada sob proteção gaseosa para evitar de contaminações por parte dos gases atmosféricos. [Machado, 2007]

O processo pode ser semi-automático, mecanizado ou automático, atingindo características excelentes para robotização. Condições para atingir isso são que quando adequadamente ajustado, o processo permite soldar em todas as posições, além disso, o mesmo produz pouquíssima escória, não sendo necessária remover as mesmas. [Machado, 2007]

A estabilidade deste processo de soldagem depende de parâmetros de soldagem como corrente, voltagem, velocidade de soldagem, distância bico de contato com a peça, gás de proteção, etc.

2.1.2.4 Sistema robótico

Os robôs são compostos de elos conectados com juntas para formar uma cadeia cinemática. As juntas em geral são rotacionais (de revolução) ou lineares (prismáticas). A junta de revolução permite uma rotação relativa entre dois elos. A prismática permite um movimento linear relativo entre dois elos. As juntas de revolução são denotadas pela letra R e as prismáticas pela letra P. [Spong et al. 2005]

Ao falar de robôs manipuladores, os mesmos podem ser classificados com diferentes critérios, como serem a fonte de potência, a sua geometria a sua área de aplicação ou seu

método de controle. Como fonte de energia os robôs podem ser acionados elétrica, hidráulica ou pneumaticamente. Como área de aplicação, os robôs podem ser de montagem ou de não montagem (robôs de solda, pintura com spray, manipuladores de materiais). No método de controle, os robôs podem ser classificados em servo (controle em laço aberto, com movimentos limitados a paradas mecânicas predeterminadas) e não servo (controle a laço fechado para determinar seu movimento). [Spong et al. 2005]

Descrevendo a classificação de robôs com relação à sua geometria, a maioria dos mesmos atualmente possui seis ou menos graus de liberdade. Os manipuladores são usualmente classificados cinematicamente na base das suas primeiras três juntas do braço, com o punho descrito separadamente. As maiorias desses manipuladores se enquadram entre uma de cinco tipos de geometrias: articulada (RRR), esférica (RRP), SCARA (RRP), cilíndrico (RPP), ou cartesiano (PPP). Uma sexta classe de manipuladores são chamados de robôs paralelos onde seus elos são arranjados em uma cadeia cinemática fechada. [Spong et al. 2005]

Os principais setores de robôs utilizados são aqueles que desenvolvem a soldagem. Uma preocupação em aumentar a lucratividade, dentro das normas, a observarem os custos de produção, evitando desperdícios, principalmente em áreas como a soldagem, é intensa. [Rodas et al. 2014]

2.1.2.5 Descrição do sistema robótico de soldagem

A definição de um robô industrial para soldagem ao arco elétrico protegida por gás define o mesmo como um autômato com movimentos de mais de três eixos que são programáveis pelo usuário e podem ser controlados com sensores. O mesmo é equipado com uma tocha de soldagem e pode desenvolver trabalhos de soldagem. [ISF Aachen, 2002]

Para aplicações de soldagem, é necessário começar com uma trajetória e ter a certeza de que a mesma esteja em tempo real como função dos resultados observados dos parâmetros de solda. [Pires et al. 2003].

Um manipulador robótico é só um componente em um denominado sistema robótico, ilustrado na Figura 2.13, o qual consiste do braço, fonte externa de potência, ferramenta do efetuador, sensores externos e internos, interface de computador o computador controlador. O

software programado deve ser considerado uma parte integral do sistema, já que a maneira em como o robô é programado e controlado pode ter um maior impacto em seu rendimento. [Spong et al. 2005]

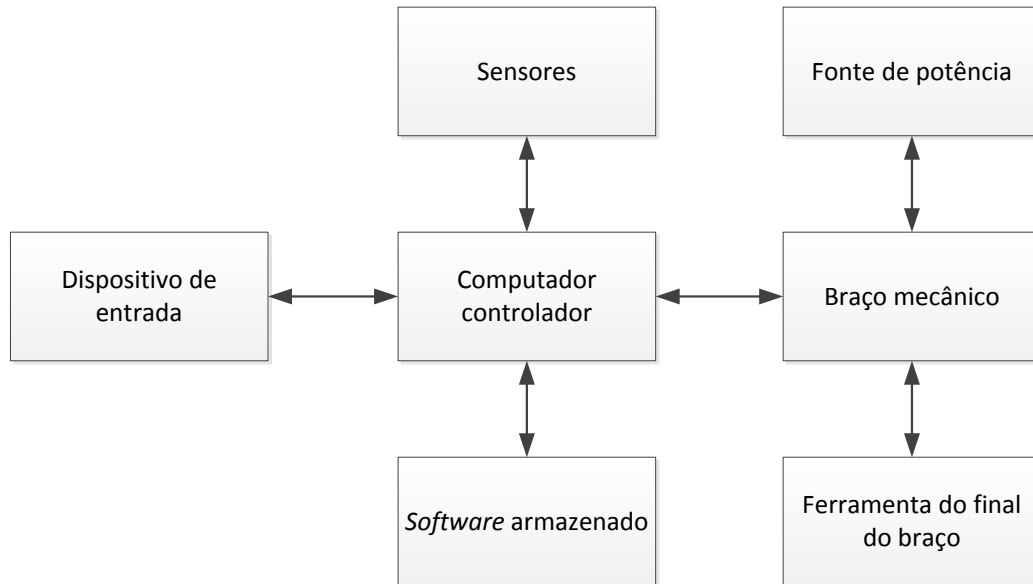


Figura 2.13 - Componentes de um sistema robótico.

No sistema de soldagem robótica usado neste trabalho, o computador controlador é de modelo DX-100, o braço mecânico é um robô Yaskawa Motoman MA-1400, a ferramenta do final do braço é uma tocha de MIG/MAG. No esquema da Figura 2.13 pode ser adicionada uma fonte de potência para o sistema de soldagem, neste trabalho a mesma é da marca Fronius.

2.2 Estado da arte

Diferentes estudos podem ser encontrados na literatura relacionando um sistema de visão aplicado à soldagem robotizada. Dentre os métodos para a detecção de pontos iniciais de solda, Chen et al., 2006 descreve um algoritmo de detecção da coordenada do ponto inicial sobre uma peça de alumínio para soldagem de junta de topo (apresentada na Figura 2.14(a)). O método proposto sugere uma técnica conhecida como correspondência de modelos ('*template matching*'), baseia-se na pesquisa de um padrão conhecido, apresentado na Figura

2.14(b), em toda a imagem. A imagem capturada é processada para diminuir a relação sinal-ruído, e facilitar a comparação com o modelo (*'template'*).

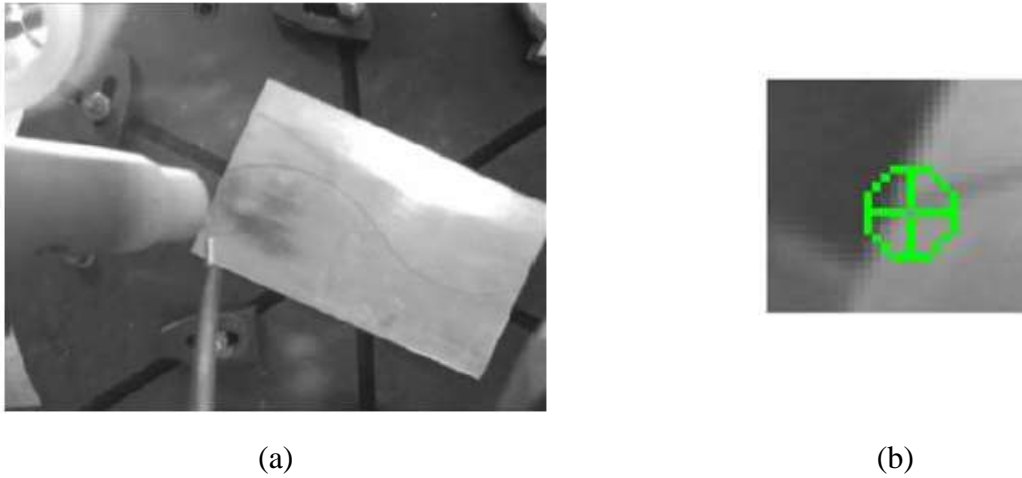


Figura 2.14 - Detecção de pontos iniciais de solda proposto por Chen et al., 2006.

Zhu et al, 2005 propõe localizar a linha de solda e posteriormente o ponto inicial, em uma junta de topo, com uma linha de solda reta, apresentada na Figura 2.15(a). Inicialmente é desenvolvida uma técnica de pré-processamento de imagem, usando um filtro de mediana, em conjunto com a técnica de erosão na imagem em tons de cinza, apresentado na Figura 2.15(b), obtendo uma linha de solda com maior largura para facilitar a detecção da mesma. Aplica-se a técnica de correspondência global (*'global matching'*) em toda a imagem com o padrão ilustrado na Figura 2.15(c); complementarmente uma técnica sugerida pelo autor é implementada para rejeitar padrões falsos detectados que não sigam a linha reta da linha de solda. Finalmente o ponto inicial da solda é determinado como o ponto que tem o menor valor da coordenada das colunas e seguindo diversas técnicas para detectar o ponto inicial, a Figura 2.15(d) é obtida.

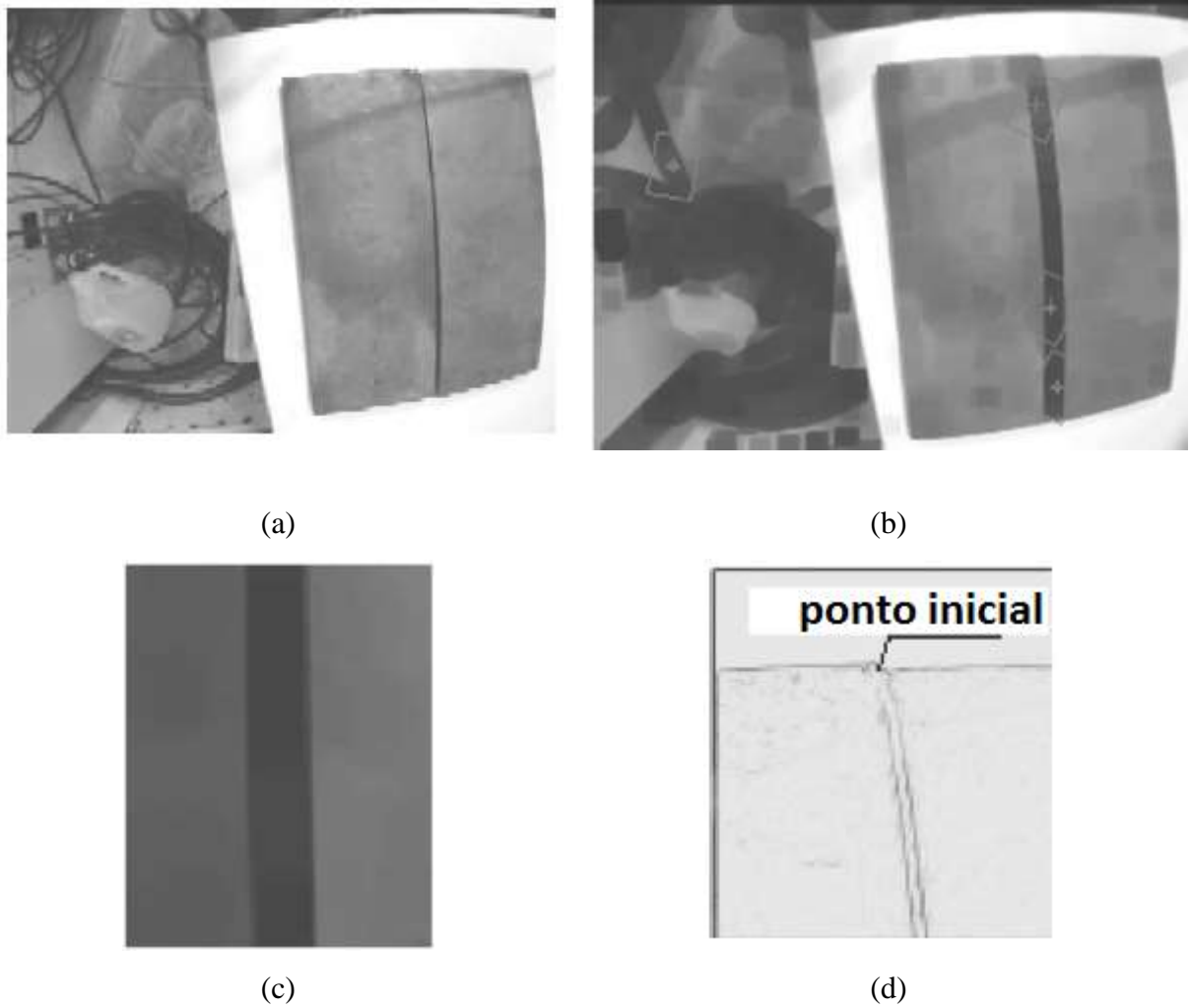


Figura 2.15 - Detecção de pontos iniciais de solda proposto por Zhu et al., 2005.

Um método que muda o conceito de determinação dos pontos iniciais da solda com padrões de modelos ou *'template'* é o método apresentado por Wei et al, 2012. Primeiramente, é obtida uma imagem do final da linha de solda, seguindo-se a detecção de bordas pelo método denominado erosão de bordas. O segundo passo reside em processar a imagem com um procedimento denominado erosão direcional, que consiste em executar a erosão na imagem binária de bordas, com elementos de estrutura linear e direcional; o resultado é a eliminação de todas as bordas que não seguem um padrão linear. Com isto, só ficariam as bordas com um padrão de linha, o último processo é aplicar a transformada de Hough [Hough, 1962], para localizar as linhas que apresentam as bordas das peças e a linha

de solda. Como resultado é obtido o valor de interseção das linhas que representam os valores de coordenadas do pixel de interesse na imagem, isto é, o ponto inicial de solda.

Kong et al., 2007, propõe um método para o reconhecimento da posição inicial de solda, baseada em tecnologia de detecção de cantos. O trabalho também trata de junta de topo, e uma linha de solda aleatória, mas com um fundo mais escuro que as peças a serem soldadas. Inicialmente como pré-processamento, utiliza-se o processo de homogeneização da imagem com um filtro de mediana, um processo de realce do contraste da imagem (com um operador Laplaciano), e finalmente uma segmentação de fundo e peça com um método de limiar automático. Com a imagem da peça em formato binário, é feito um processo de melhoria da mesma, e o resultado é processado para que sejam detectados os cantos com o método de detecção de cantos proposto por Harris (para maior informação do método de Harris, o leitor pode se referir à Harris e Stephens, 1988).

Chen et al., 2007 apresenta um algoritmo para reconhecer a linha de solda para ambientes complexos de soldagem. Trabalha com duas imagens, a imagem do ambiente com a peça ilustrada na Figura 2.16(a) e sem a peça ilustrada na Figura 2.16(b); separando a região da peça, da região do ambiente. Uma filtragem da imagem é feita por uma técnica proposta que tem como parâmetro o desvio padrão do ruído. Também aplica um método de segmentação por limiar (*Threshold Segmentation*) automático obtendo a ilustrada na Figura 2.16(c). A imagem é processada com uma série de técnicas como detecção de bordas, desbaste (*thining*), remoção de ruído, obtendo uma linha de solda limpa como resultado na ilustrada na Figura 2.16(d).

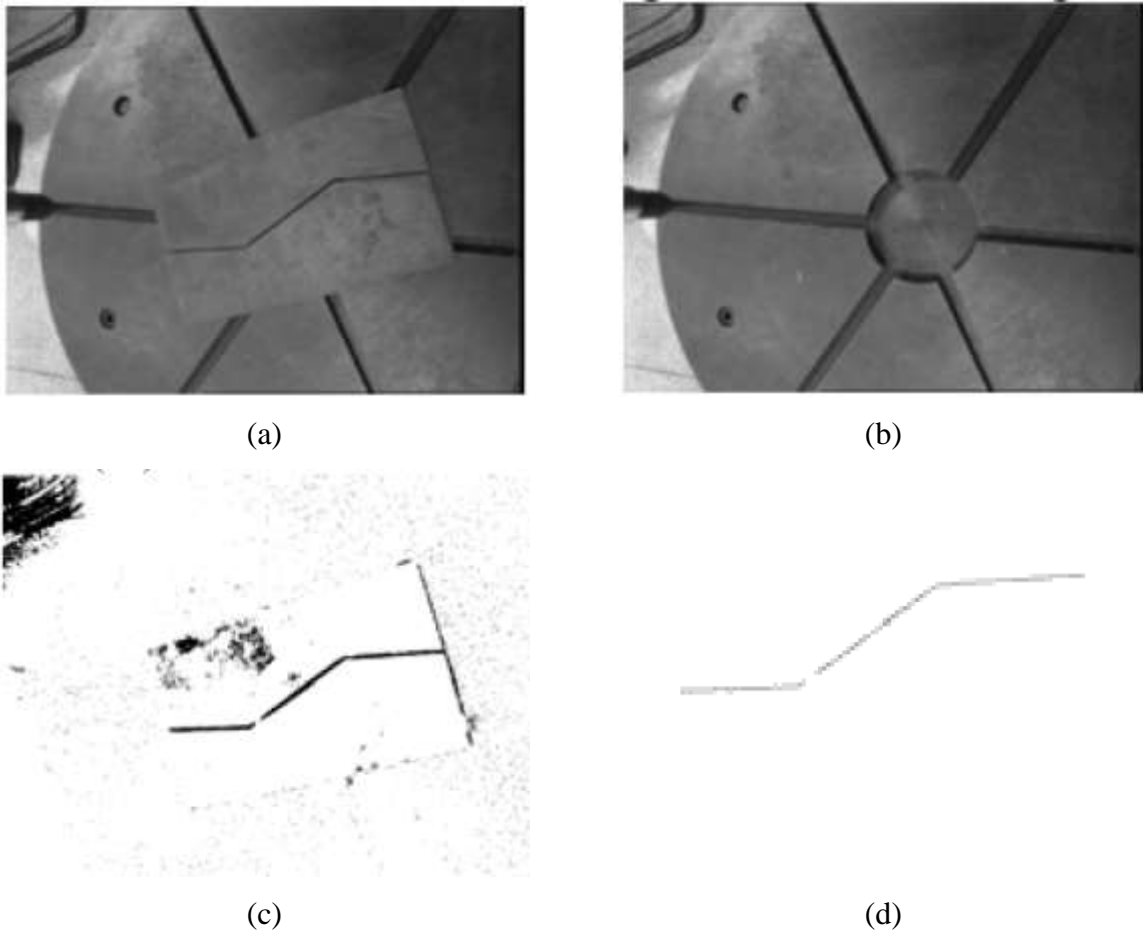
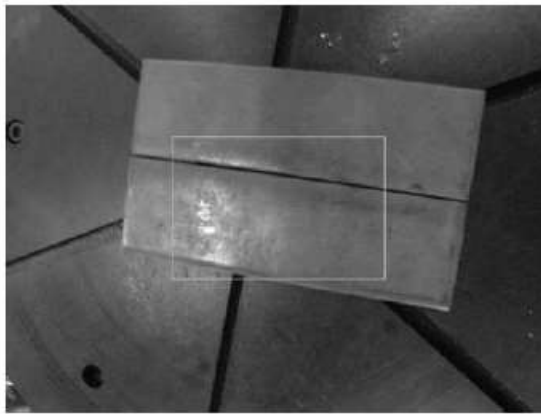


Figura 2.16 - Detecção de trajetória de soldagem proposto por Chen et al., 2007.

Chen e Chen, 2010 trabalha com uma configuração estéreo de câmeras instaladas no braço robótico, utilizando-se o método próprio para determinar a linha de solda e as bordas das peças, com um ajuste paramétrico de polinômio. Assim é obtida uma equação que determine a linha de solda e as bordas das peças. A interseção desses dois polinômios, traz como resultado a estimação do ponto inicial de soldagem. Em alguns casos o ponto estimado tem um erro elevado. Para corrigir o problema, como proposta de solução é feita uma janela ao redor do ponto e aplicada uma técnica de detecção de cantos, que tem como resultado o ponto correto de início de solda. Uma vez identificado em duas câmeras, devidamente posicionados, obtém-se as coordenadas no espaço do ponto de início de solda.

Um problema no momento de detectar a linha de solda é separar as imagens das peças de solda daquelas do fundo do plano da imagem. Para solucionar este problema, um método denominado região de interesse predefinida (*'Region of interest'* ou ROI) é proposto em Shi et al., 2009. Propõe-se neste método detectar a linha de solda e os pontos iniciais e finais de

peças em juntas de topo, através de uma imagem obtida por uma câmera, onde em uma janela no centro da imagem da Figura 2.17(a) é aplicada a técnica de Canny de detecção de bordas obtendo a ilustração da Figura 2.17(b). O algoritmo desenvolvido procura dentro da janela linhas paralelas, trabalhando com a condição que a média de intensidade dos pixels entre as duas linhas paralelas, tenha uma intensidade menor que os pixels que estão sobre as linhas. Com isso consegue-se detectar as linhas de solda dentro da janela obtendo a Figura 2.17(c). A janela é deslocada tendo como referência o ângulo da linha de solda em cada ponto final, para procurar novos pontos de linha de solda com o algoritmo descrito, até que os pontos finais de solda não estejam nas bordas da janela, definindo o final das linhas de solda. Na janela que detém o final da linha de solda, é aplicada a técnica de cantos de Harris para detectar corretamente o ponto inicial e final de solda obtendo assim a ilustração da Figura 2.17(d).



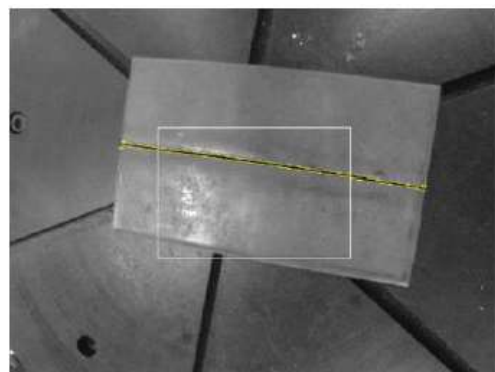
(a)



(b)



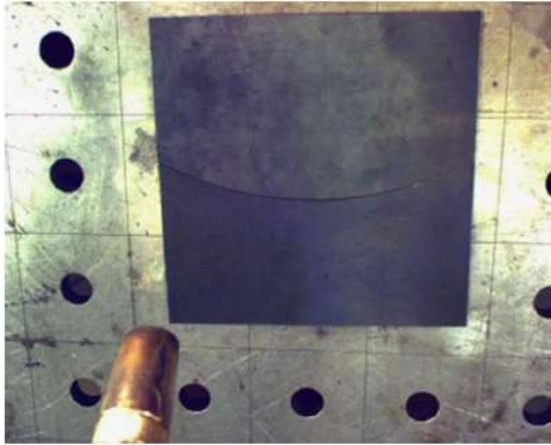
(c)



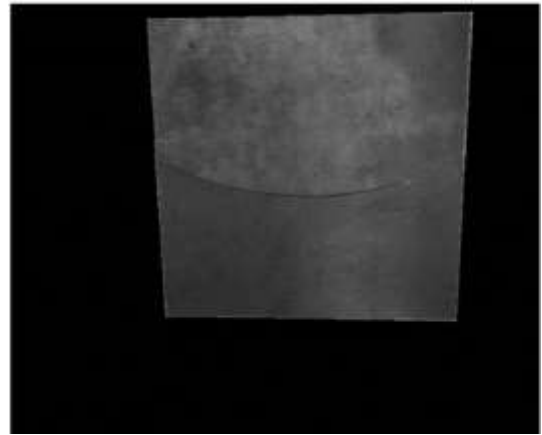
(d)

Figura 2.17 - Detecção de trajetória de soldagem proposto por Shi et al., 2009.

Outro método de detecção de linha de solda, que não consiste em separar as peças do fundo com a técnica de ROI, é o método desenvolvido pelo Dinham e Fang, 2013, onde trabalha-se com imagem ilustrada na Figura 2.18(a), aplica-se uma filtragem da mediana para deixar mais suave e diminuir os ruídos da imagem. Para detectar as bordas da imagem, é usado o método de limiar adaptativo de Otsu, 1979, seguido do método de Sobel. Por meio da transformada de Hough, são detectadas as linhas da borda das peças. Para comprovar que as linhas sejam das bordas das peças, são analisados os dois lados de cada linha de Hough, com uma janela deslizante, obtendo os valores da média de intensidade de cada lado; finalmente é feita uma subtração entre os dois valores, se o valor estiver acima de um limiar (valor limite estabelecido), ele é considerado borda da peça de solda, zerando todos os valores que não são correspondidos com as peças de solda, obtém-se apenas os pixels de interesse apresentado na Figura 2.18(b). Com a nova imagem, é preciso detectar as bordas novamente, para isso aplica-se o método de Sobel obtendo a ilustração da Figura 2.18(c). As linhas das bordas são detectadas com algoritmos de seguimento de contornos, procurando a linha com maior comprimento e eliminando as mesmas. Com isso, permanece definida a linha de solda e alguns pixels espúrios apresentados na Figura 2.18(c). A linha de solda é considerada como o *cluster* de pixels de maior comprimento em torno do eixo x ou em torno do eixo y obtendo os pixels correspondentes com a mesma ilustrados na Figura 2.18(d).



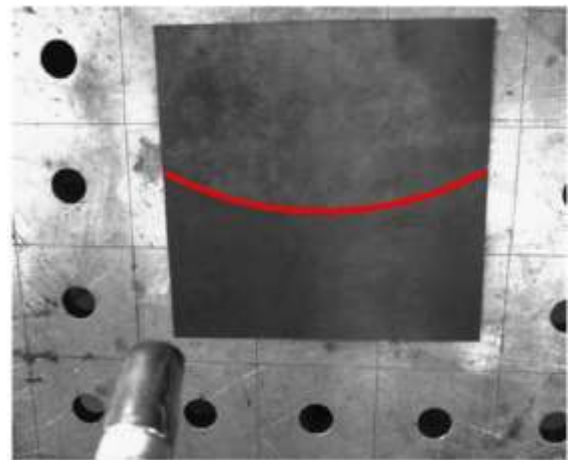
(a)



(b)



(c)



(d)

Figura 2.18 - Detecção de trajetória de soldagem proposto por Dinham e Fang, 2013.

3. DESCRIÇÃO DO SISTEMA

Lembrando que o objetivo do presente trabalho é implementar um sistema para identificar trajetórias de soldagem em juntas de topo, utilizando sistemas de visão computacional. Para tanto, são propostos dois métodos: o primeiro para detecção de linhas retas de solda, enquanto o segundo método permite a detecção de pontos em linhas retas e curvas de solda. As trajetórias identificadas devem ser executadas por um manipulador robótico associado a equipamento de soldagem MIG/MAG.

3.1 Método 1: Detecção de linhas retas

A partir das imagens de peças em juntas de topo para executar o processo de soldagem entre as mesmas procura-se identificar linhas retas, que identifiquem trajetórias de soldagem. Uma imagem das peças de trabalho formando uma junta de topo, tomados como estudo no trabalho é apresentado na imagem da Figura 3.2 com dimensões de 2048x1536 pixels.

Com o objetivo de proporcionar de ter uma descrição geral do primeiro método, o diagrama de fluxo da Figura 3.1 é apresentado. O código para desenvolver o método é escrito completamente em Matlab® [MathWorks, 2015], começando o mesmo com uma filtragem da imagem para redução de ruídos, continuando com a detecção de bordas das peças, para assim identificar as mesmas. Como as peças apresentam bordas retas, com a etapa de detecção de linhas retas é possível detectar as mesmas. Com as interseções das linhas retas são obtidos os cantos das mesmas (aqui chamados de pontos de interesse), assim é proposto o método de detecção de pontos de interesse. Finalmente as peças são separadas tendo em conta seus pontos de interesse, e os pontos em comum entre cada peça definem os pontos inicial e final de solda.

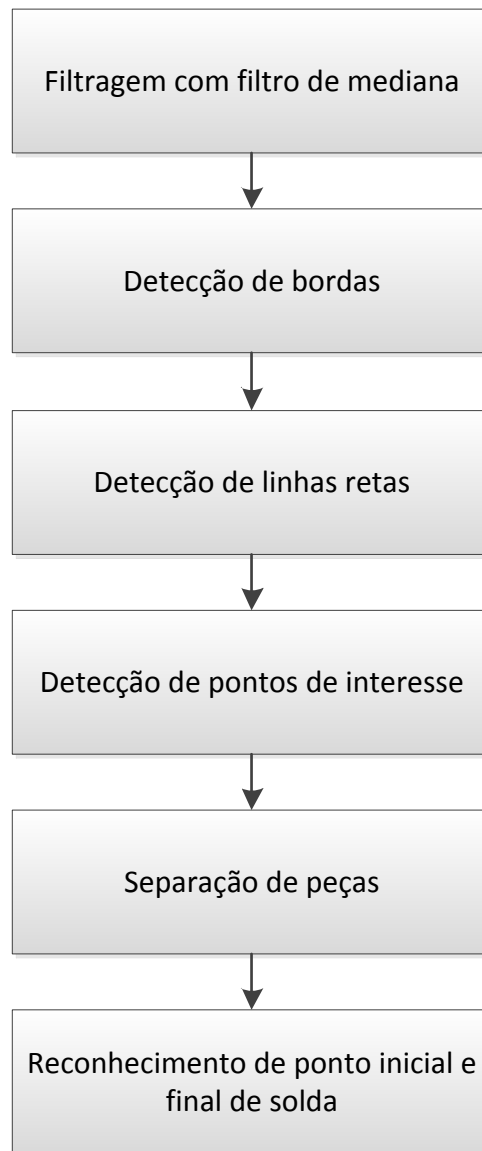


Figura 3.1 - Diagrama de fluxo de método 1.

No momento de detecção de bordas da imagem da solda, as imperfeições superficiais como os arranhões nas peças ou no fundo, não são desejadas. Para reduzir estas detecções, um filtro de mediana é implementado sobre a imagem. A vantagem de um filtro de mediana é que as linhas do fundo são reduzidas, mas as linhas fortes das linhas de solda são preservadas [Dinham e Fang, 2013]. Para as imagens abordadas neste trabalho, um filtro de mediana de 3x3 apresentou em princípio os melhores resultados ao obter a imagem de bordas comparado com outros tamanhos de filtro.

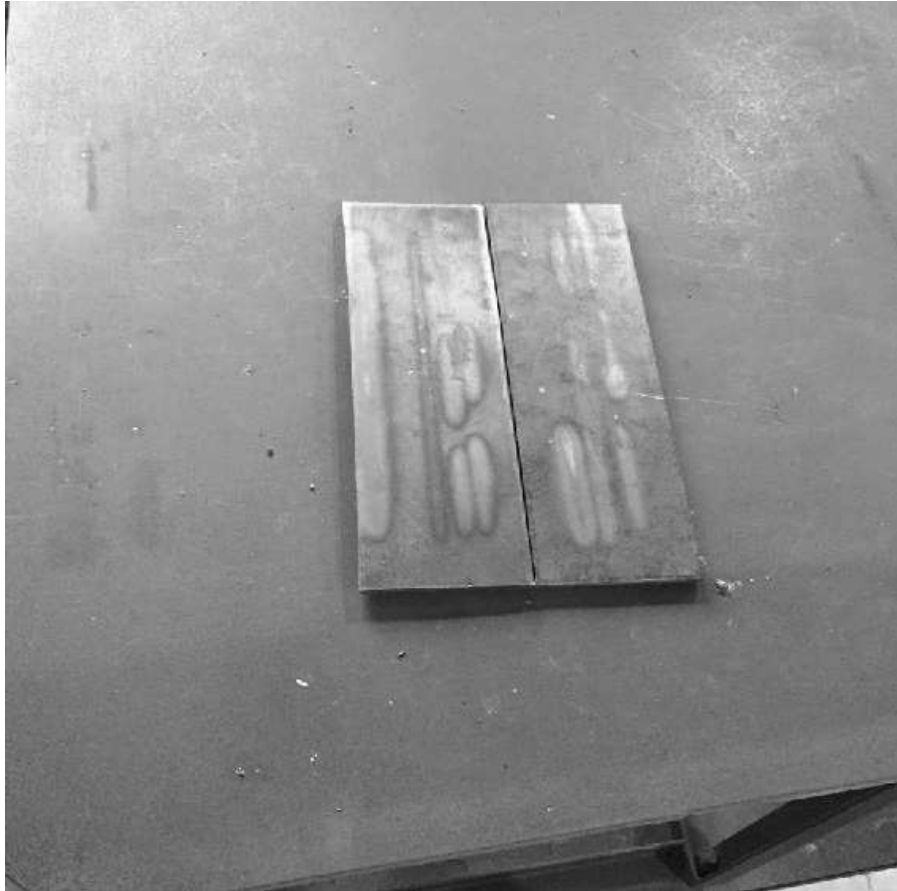


Figura 3.2 - Imagem de peças de trabalho.

Existem diversos métodos de análise para identificação das linhas que apresentam as bordas das figuras das imagens. O método de detecção de bordas proposto por Canny, 1986 é escolhido por apresentar uma linha de borda de largura de 1 pixel, o que permite assim medir com maior facilidade o comprimento das linhas, sendo o mesmo usado em trabalhos como Shi et al, 2009 ou Chen e Chen, 2010. Para o algoritmo ser robusto às mudanças de luminosidade, é aplicada uma técnica para o valor do limiar mínimo t_{low} e limiar máximo t_{high} do detector de bordas, em função da média e do desvio padrão das intensidades dos pixels na imagem [Hossain et al. 2016] como apresentado na equação (3.1).

$$\begin{aligned} t_{low} &= \max (0, (\mu_{int} - \sigma_{int}) / a) \\ t_{high} &= \min (A, (\mu_{int} + \sigma_{int}) / b) \end{aligned} \quad (3.1)$$

Onde μ_{int} é o valor médio da probabilidade normalizada da função densidade de probabilidade.

σ_{int} é o desvio padrão da probabilidade da função densidade de probabilidade.

a é um parâmetro entre 5 e 7, geralmente 6.

b é um parâmetro entre 2 e 4, geralmente 3.

A é o valor de intensidade máxima.

A Figura 3.3 ilustra o resultado da aplicação da detecção de bordas sobre a imagem da Figura 3.2.

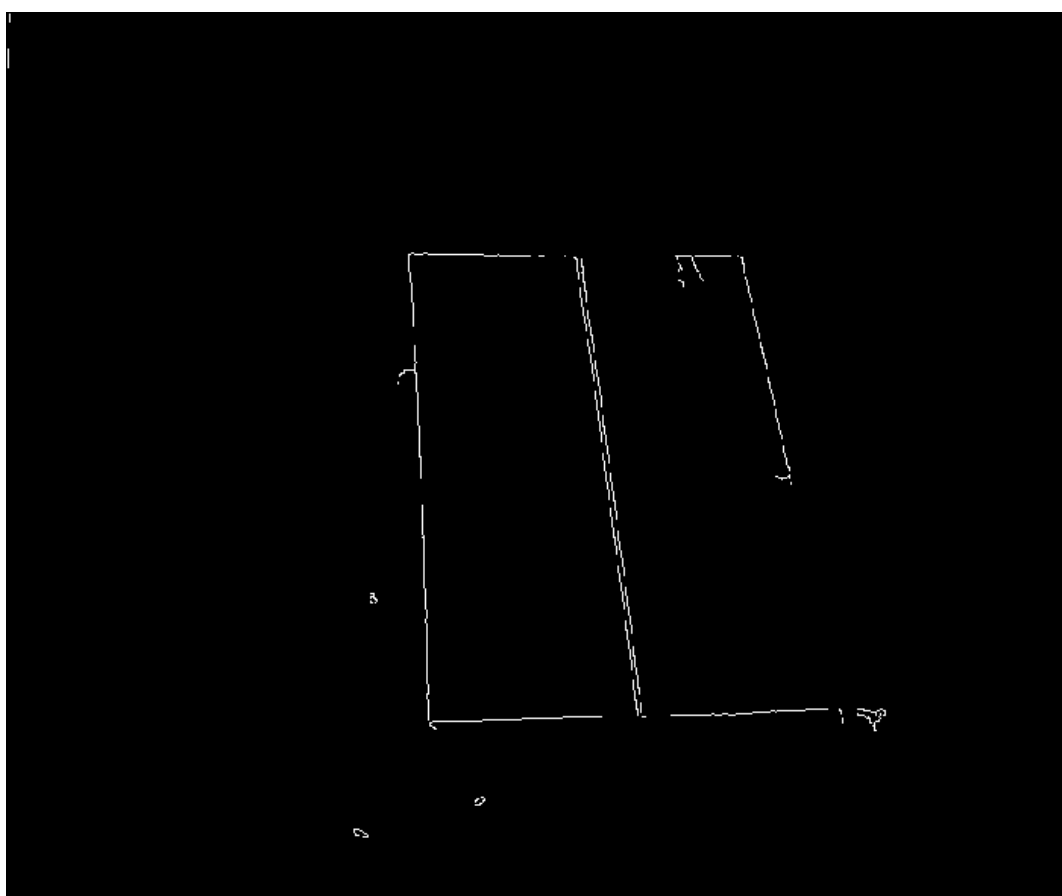


Figura 3.3 - Borda de peças após aplicação de detecção de borda Canny.

Uma vez obtidas as bordas da imagem, apresentado na Figura 3.3 como as peças apresentam padrões retos, pode-se obter as equações das linhas retas que ligam a maioria dos pixels de bordas. A solução é obtida pelo método da transformada de Hough, que como resultado gera 5 linhas retas que apresentam as bordas das peças, como apresentado na Figura 3.4.

● Pontos de interesse

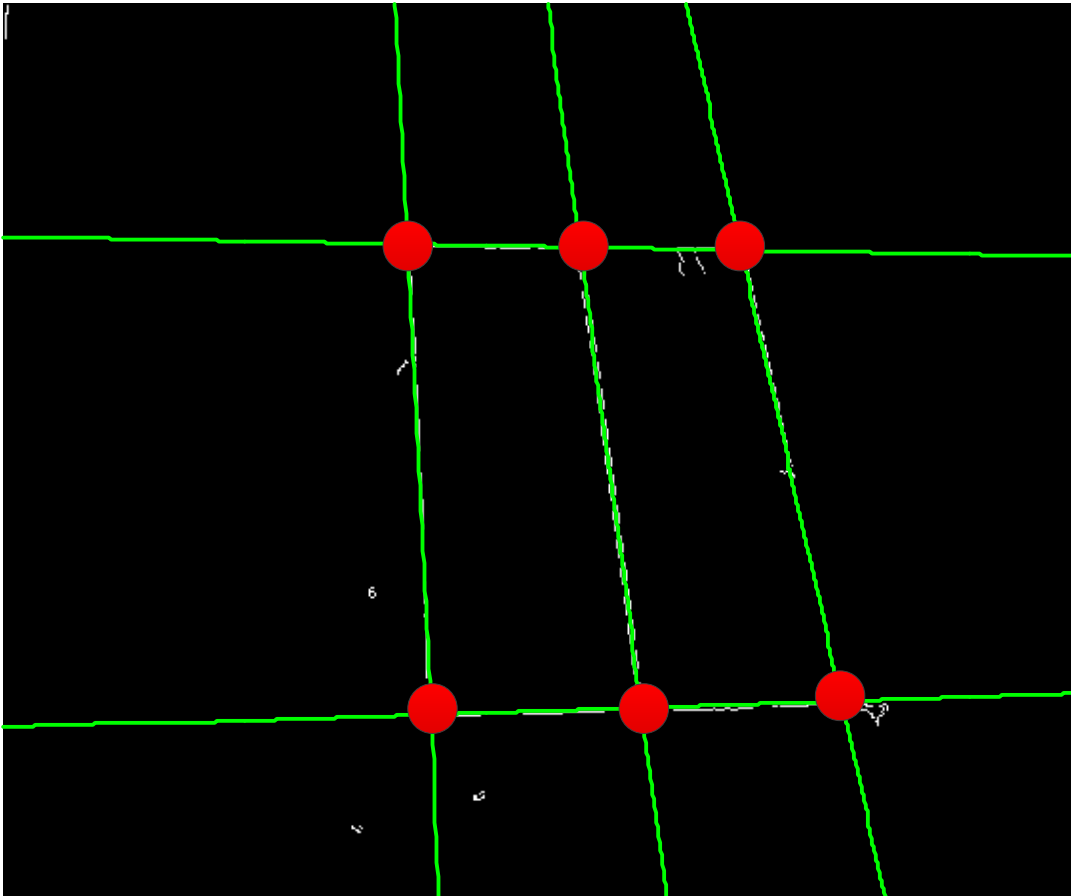


Figura 3.4 - Linhas de Hough superpostas sobre as bordas da imagem.

Analisando a Figura 3.4, pode-se perceber que com a interseção das linhas são obtidos os pontos para estimar a coordenada dos cantos de cada uma das chapas, destacando-se 6 pontos que são denominados pontos de interesse, onde dois destes correspondem aos pontos de início e final da solda.

3.1.1 Processo de reconhecimento da posição inicial e final da solda

Como mencionado no item anterior, a interseção das linhas de Hough permite detectar pontos de interesse, mas constata-se que as mesmas linhas podem ter uniões fora da imagem, ou em alguns casos, uniões falsas dentro da mesma imagem. Para limitar as

intersecções falsas, são feitas duas considerações: 1 – a intersecção será válida só dentro do plano da imagem, 2 – se dentro de uma janela de 10x10 com o centro da janela como o ponto de intersecção, não se detectarem pixels de borda, a intersecção será invalidada. Na Figura 3.5 é apresentado um caso de um ponto de interesse validado, localizado dentro da imagem e apresentando pontos de borda dentro da janela 10x10.

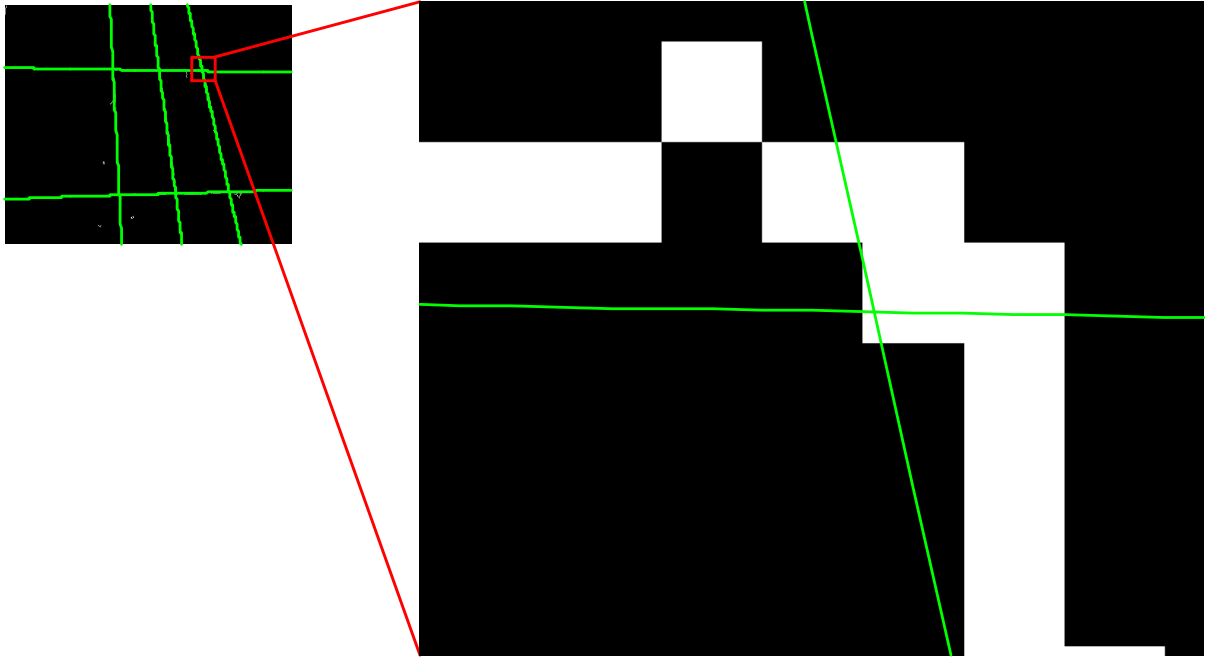


Figura 3.5 - Imagem de 10x10 para verificar condições de ser ponto de interesse.

Com os limites anteriores impostos sobre a figura, são detectados 6 pontos de interesse, marcados com círculos vermelhos na Figura 3.4, que fazem referência a cada canto das peças. Na sequência detecta-se se cada ponto pertence ou não à trajetória de soldagem.

O método aqui proposto se baseia no seguinte princípio: 4 pontos de interesse pertencem a cada canto de cada peça, o método reconhece à que peça corresponde cada um dos 6 pontos de interesse. Dois pontos de interesse vão pertencer a duas peças, os quais são correspondidos com os cantos compartilhados.

A seguir são descritas as etapas de análise para diferenciar pontos de interesse em cada peça. Primeiramente é feita uma tabela com as coordenadas x e y de cada ponto de interesse, e nas colunas contíguas, enumeradas cada uma das duas linhas retas que cortam no mesmo ponto. As retas são enumeradas arbitrariamente, com valores inteiros que vão do 1 até o 6. Por exemplo, na Tabela 3.1 são listados dados de dois pontos, um ponto de interesse que tenha a coordenada x_1 e y_1 e as linhas retas L_1 e L_2 , o mesmo compartilha a reta L_1 com outro ponto de coordenadas x_2 e y_2 , que também tem a linha 3.

Tabela 3.1 - Exemplo de representação de pontos de interesse.

	Coordenada x	Coordenada y	Primeira reta	Segunda reta
Ponto 1	x_1	y_1	L_1	L_2
Ponto 2	x_2	y_2	L_1	L_3

Para detectar que ponto de interesse pertence com cada peça, as premissas usadas são que cada peça pode ter como máximo dois pontos de interesse por linha e que ao escolher um ponto de interesse, uma vez escolhido um ponto, o próximo ponto de interesse da peça tem que estar localizado em uma das duas linhas retas que interceptam o ponto atual. A seguir vai ser descrito o método para a separação dos pontos de interesse:

1. escolha aleatória de um ponto de interesse. Figura 3.6
2. escolha do segundo ponto de interesse: o próximo ponto de interesse tem que ser localizado sobre uma das duas linhas retas que interceptam no ponto de interesse atual. Uma das linhas retas terá na mesma dois pontos de interesse e outra terá três pontos de interesse. O próximo ponto de interesse a ser escolhido é o mesmo da linha que possui dois pontos de interesse, como apresentado na Figura 3.7.
3. escolha do terceiro ponto de interesse: Neste caso só uma linha pode ser escolhida, mas a mesma tem dois pontos de interesses disponíveis. O ponto escolhido é o mesmo que possui a menor distância euclidiana com o ponto de interesse escolhido no passo 2, conseguindo a ilustração da Figura 3.8.
4. escolha do quarto ponto de interesse: Uma vez localizados no terceiro ponto, só fica disponível uma linha reta que intersecta o mesmo e que não tem seus dois pontos já usados. O ponto pertencente com esta ultima linha reta é escolhido, conseguindo a Figura 3.9.

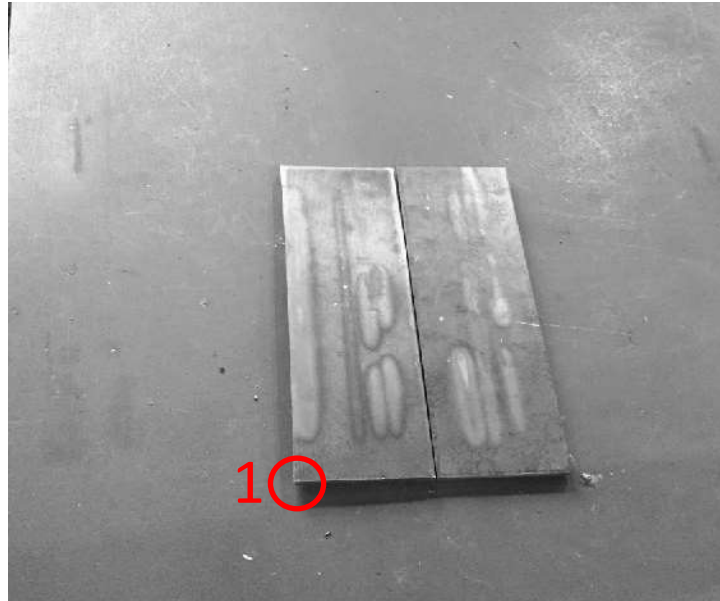


Figura 3.6 - Metodo1: escolha de ponto aleatório.

- Pontos de interesse escolhidos
- Pontos de interesse rejeitados

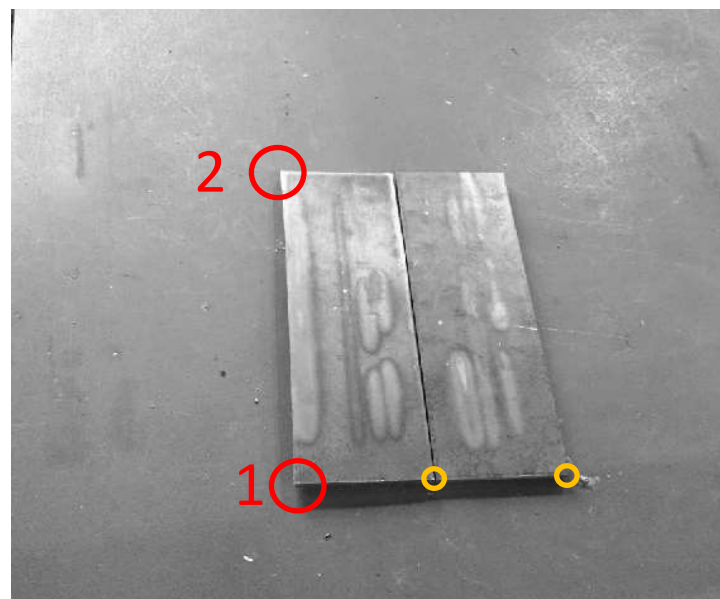


Figura 3.7 - Metodo1: escolha de segundo ponto.

- Pontos de interesse escolhidos
- Pontos de interesse rejeitados

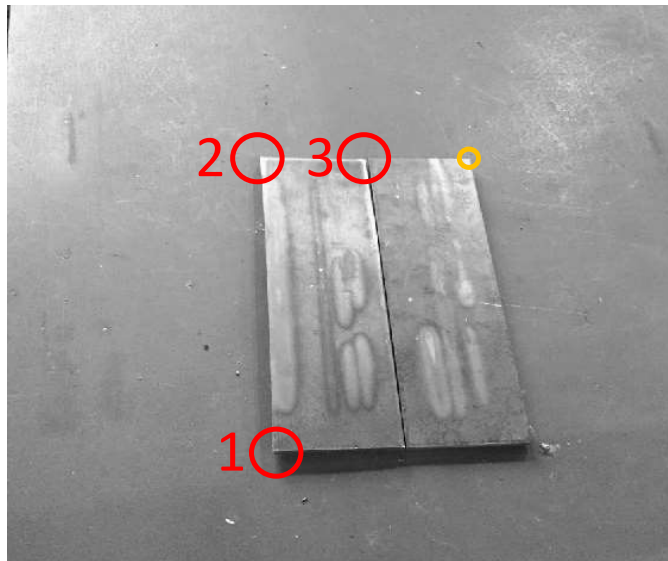


Figura 3.8 - Metodo1: escolha de terceiro ponto.

- Pontos de interesse escolhidos
- Pontos de interesse rejeitados

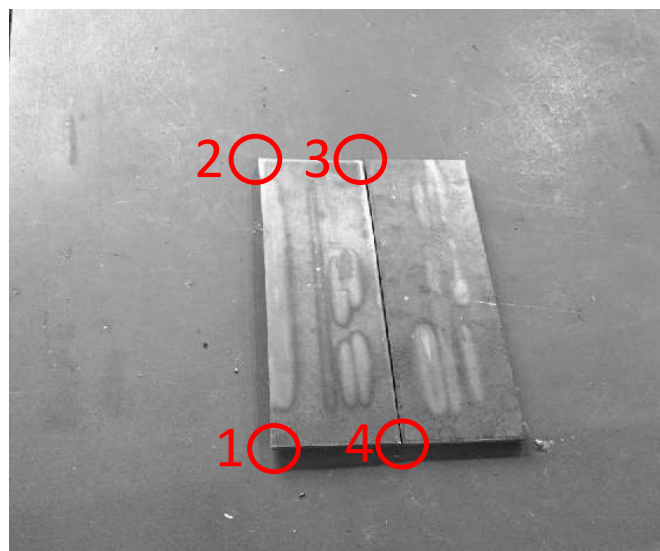


Figura 3.9 - Metodo1: escolha do quarto ponto.

Escolha de pontos para a segunda peça: Uma vez escolhidos os pontos de interesse que pertencem a uma peça, um ponto aleatório é escolhido dos pontos de interesse ainda disponíveis. O procedimento é realizado da mesma maneira desde o passo dois até o quatro.

Uma vez obtidos os pontos pertencentes a cada peça, é possível alcançar os resultados mostrados na Figura 3.10, onde os pontos que apresentam os cantos da peça esquerda estão dentro do círculo vermelho e os pontos que apresentam a peça direita estão dentro do círculo azul.

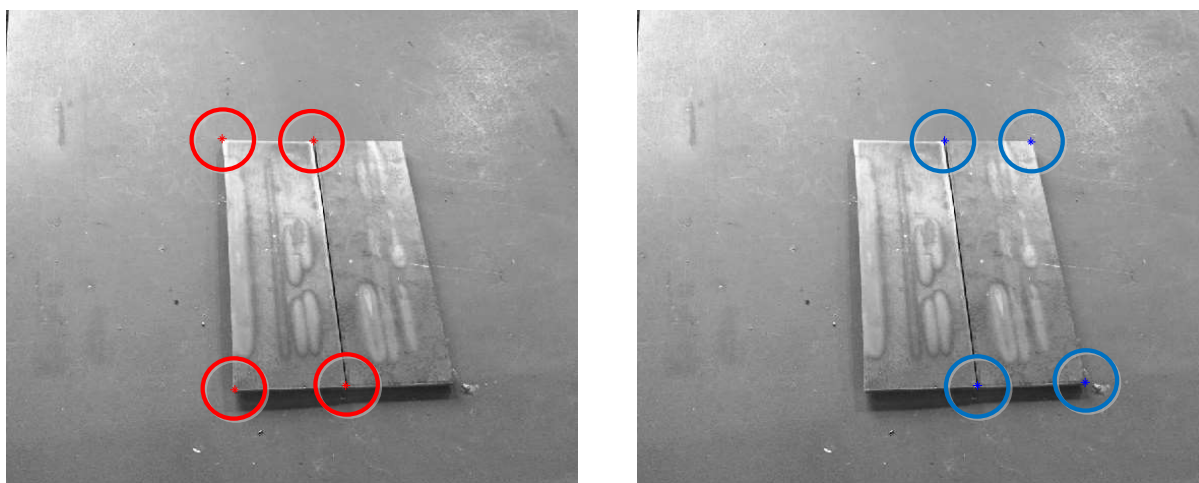


Figura 3.10 - Pontos de interseção sobrepostos sobre as peças – (a) esquerda, (b) direita.

Uma vez escolhidos os pontos de cada peça, os mesmos são armazenados em um vetor que representa cada peça (neste caso haverá dois vetores). Na sequência da análise são comparados esses vetores, e os pontos em comum de cada um deles são finalmente definidos como sendo os pontos de início e final de solda. A imagem obtida com os pontos de início e final da solda é apresentada na Figura 3.11.

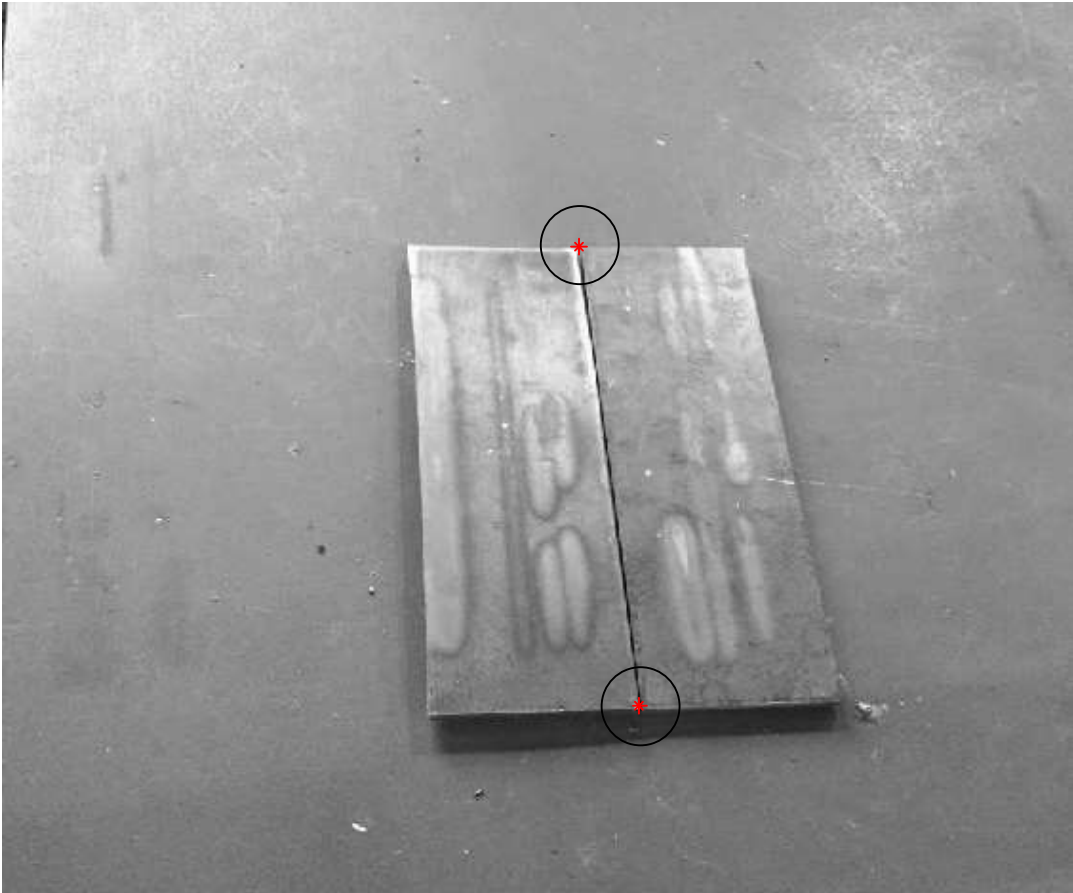


Figura 3.11 - Detecção de ponto inicial e final da solda.

3.2 Método 2: Detecção de linhas retas e curvas

Nas seções anteriores, a detecção de pontos iniciais e finais foi atingida unicamente em peças que apresentam as seguintes características:

- configuração tipo topo;
- peças do mesmo tamanho;
- as peças devem compartilhar dois cantos;
- linha de solda em linha reta;

Em muitos casos na prática estas condições não são atingidas, exigindo assim um algoritmo com maior dinâmica na detecção de linha de solda. Mais especificamente em peças com as seguintes características:

- configuração tipo topo;

- peças de diferentes tamanhos;
- as peças não precisam compartilhar os cantos;
- linha de solda pode ser curva;

A imagem com a qual será trabalhado nesta seção é chamada imagem de solda em tons de cinza, a mesma é uma imagem em tons de cinza de duas peças de soldagem em junta tipo topo, apresentada como $B_c(x,y)$ na Figura 3.12.

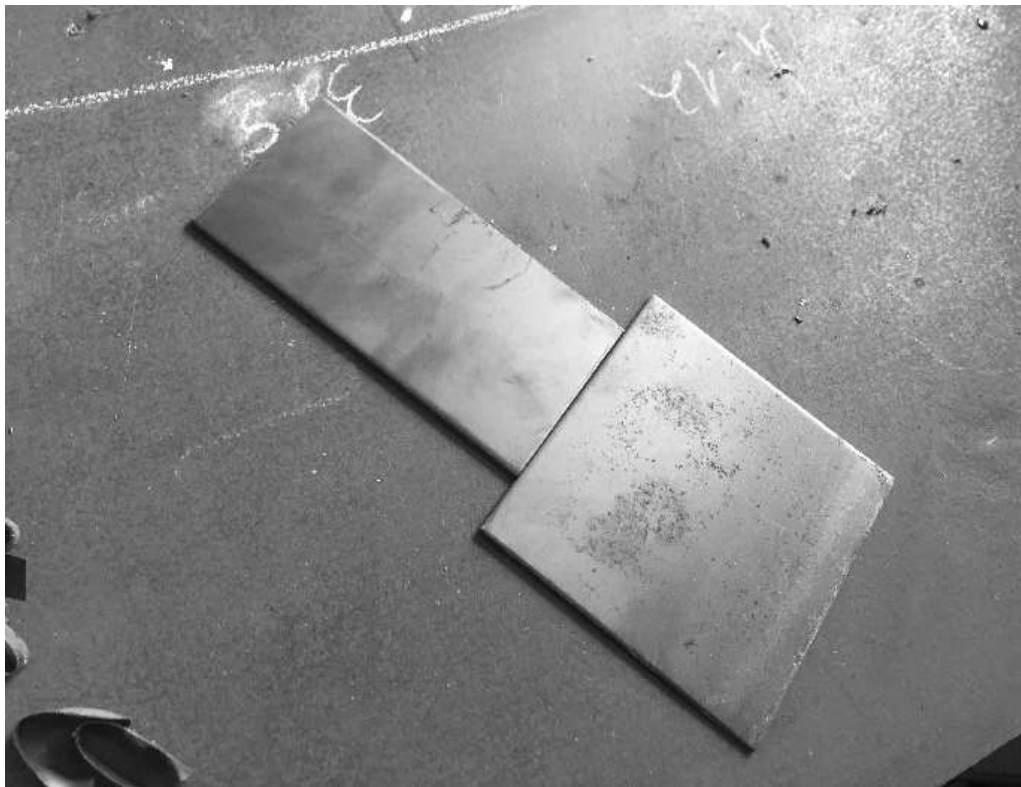


Figura 3.12 - Imagem de solda em tons de cinza $B_c(x,y)$.

3.3 Resumo do método 2 de detecção de linha de solda

Um resumo do método de detecção de linha de solda proposto neste trabalho é apresentado no fluxograma da Figura 3.13, onde são descritas as etapas do pré-processamento da imagem ('Filtragem com filtro da mediana', 'Detecção de bordas'), detecção de uma linha que apresenta a estimação da linha de solda ('Varredura global de pixels com linhas paralelas e pixel central com menor intensidade', 'Medição e filtragem de linha de borda com maior

comprimento’), obtenção de uma matriz de linha de solda (‘Detecção e medição de distâncias entre pixels de linhas paralelas’, ‘Adição de novos pixels baseados em intervalos de distâncias’), finalizando com etapas de robustez frente à descontinuidade de linha de solda e correção de ponto inicial e final da solda. O código total foi desenvolvido com o *software* Matlab®.

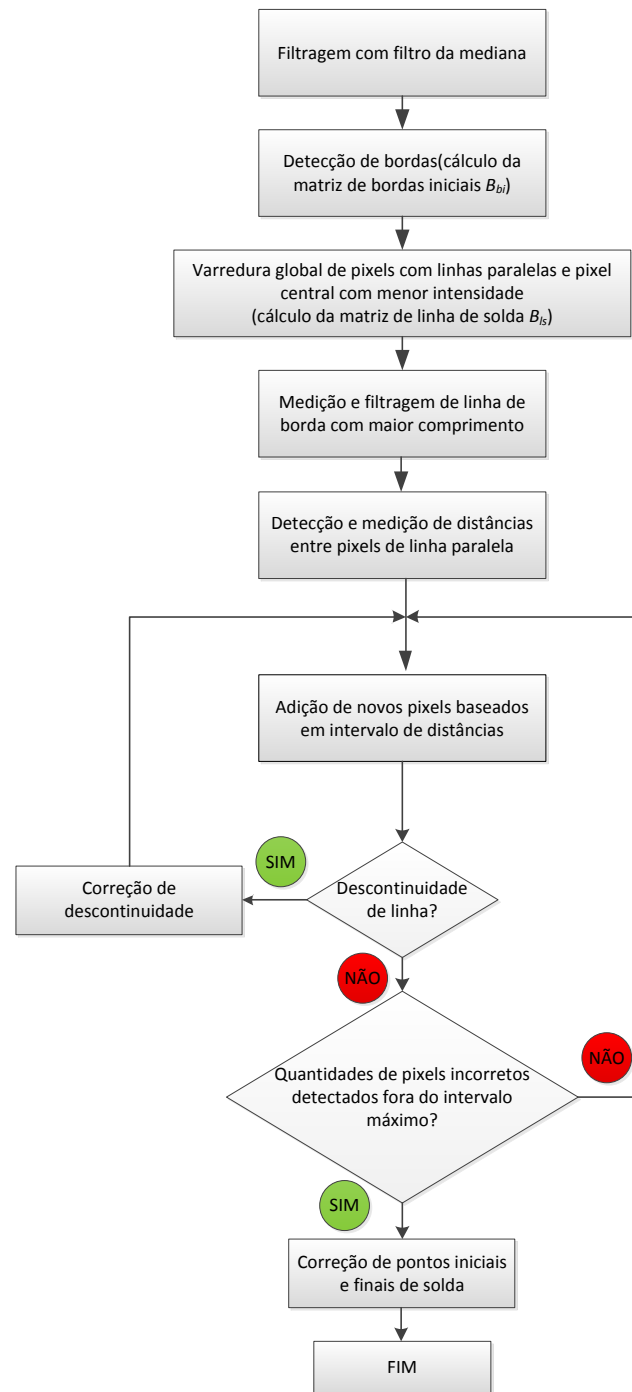


Figura 3.13 - Fluxograma geral do método apresentado no trabalho.

3.4 Filtragem e detecção de bordas

A aplicação inicial do algoritmo consiste em filtrar a imagem e detectar as bordas da peça. Com esse objetivo é aplicado um filtro da mediana de 3x3 em conjunto com a técnica de detecção de bordas de Canny, utilizando os limiares adaptativos como descrito na equação (3.1). A imagem obtida (Figura 3.14) é chamada de matriz de bordas iniciais e apresentada com o símbolo $B_{bi}(x,y)$.

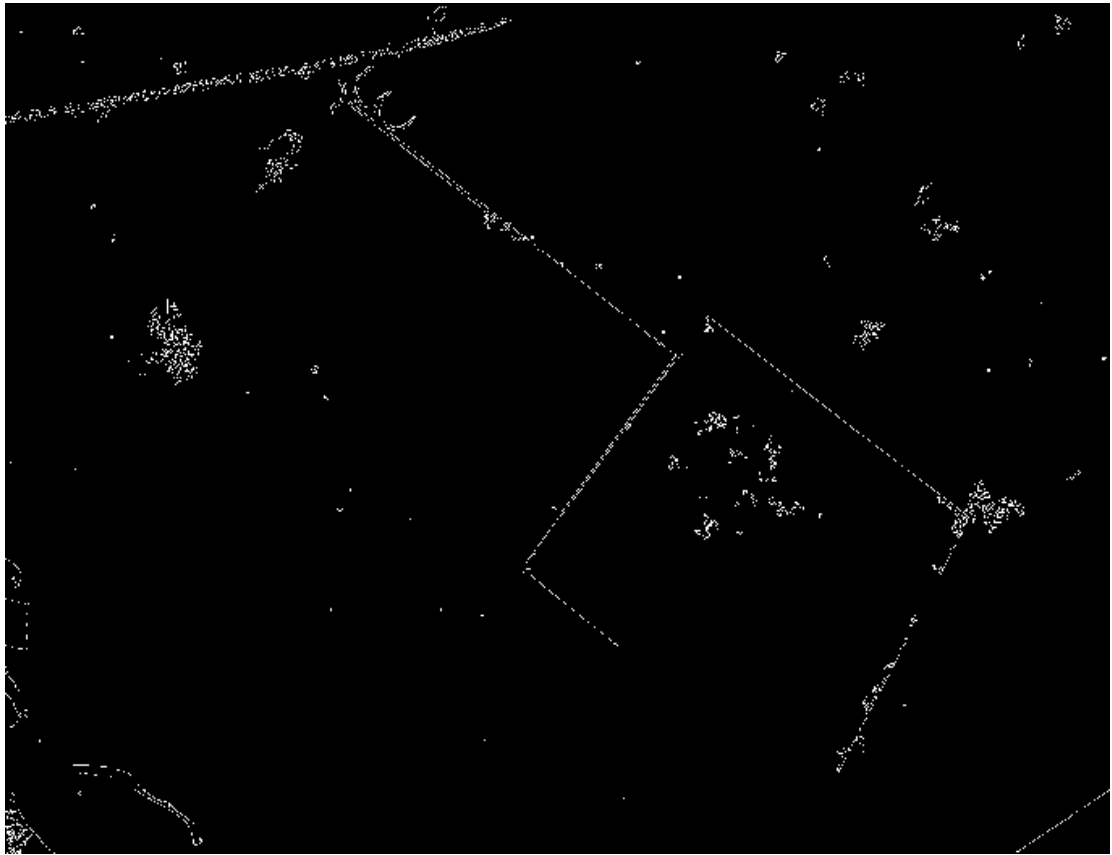


Figura 3.14 - ‘Matriz de bordas iniciais’ $B_{bi}(x,y)$ obtida com detecção de bordas de Canny.

Dentro dos pixels mostrados na Figura 3.14 verifica-se que alguns pertencem à linha de solda e outros não. Os pixels que pertencem ao conjunto de pixels da linha de solda vão ser denominados ‘pixels de linha de solda’, e os demais pixels, vão ser chamados de ‘pixels falsos’.

Primeiramente são descritos uma série de passos com o objetivo de obter uma estimativa inicial dos pixels denominados ‘pixels de linha de solda’. O princípio básico dos métodos se baseia no artigo de Shi et al. 2009, que avalia dois conjuntos de pixels, o primeiro localizado em uma linha e o segundo em outra linha paralela. Para que os dois conjuntos sejam considerados pixels de linha de solda, a intensidade de cada um deles deve apresentar um valor maior que a intensidade dos pixels localizados entre as linhas.

3.4.1 Detecção de vizinhanças

Neste passo é identificado cada pixel de borda detectado na matriz de bordas iniciais $B_{bi}(x,y)$, uma vez detectado um pixel de bordas, são procurados nas 8 vizinhanças do mesmo, outros pixels de bordas; como apresentado na Figura 3.15, onde o pixel de borda avaliado é o pixel vermelho ‘p’ e os 8 vizinhos estão representados por cores cinza. A condição para que o pixel seja considerado um possível pixel de borda é de que o pixel avaliado tenha como mínimos dois pixels de borda como vizinhança, caso contrário o mesmo é considerado pixel falso.

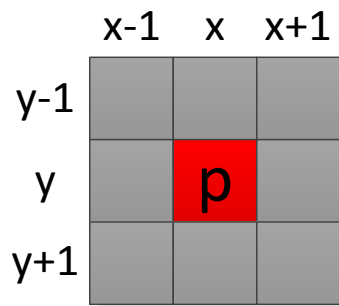


Figura 3.15 - Procura de pixel de borda nas 8 vizinhanças.

O conceito de pixel de borda é qualquer valor na matriz que tenha uma intensidade maior ou igual ao ‘1’ (apresentado com cor branco na BEM), e os valores que não são considerados bordas são os pixels com um valor de intensidade ‘0’ (apresentado com cor preto na BEM).

Em caso afirmativo da condição, é necessário que a quantidade de vizinhos seja dois, para assim conseguir uma linha reta que apresente o valor da reta tangencial da linha na localização do pixel avaliado. Na hipótese da quantidade de vizinhos serem dois, o algoritmo

para a etapa de ‘Reta perpendicular das vizinhanças’, caso apresente mais de 2, vizinhos o procedimento é o seguinte:

1. são medidas as distâncias euclidianas d_{pixels} entre cada par de coordenadas de pixels de borda (x_{b1}, y_{b1}) e (x_{b2}, y_{b2}) respectivamente. Como apresentado na equação (3.2).

$$d_{pixels} = \sqrt{(x_{b2} - x_{b1})^2 + (y_{b2} - y_{b1})^2} \quad (3.2)$$

2. obtenção do conjunto de pixels com a menor distância.
3. esses dois pixels são representados por uma nova coordenada de pixel (x_{bnew}, y_{bnew}) com cada coordenada como o valor médio das coordenadas do par de pixels, equação (3.3).

$$\begin{pmatrix} x_{bnew} \\ y_{bnew} \end{pmatrix} = \begin{pmatrix} (x_{b2} + x_{b1})/2 \\ (y_{b2} + y_{b1})/2 \end{pmatrix} \quad (3.3)$$

4. avaliação do número de pixels vizinhos, caso seja maior do que 2, retornar ao item 1. Caso seja igual a 2, o algoritmo passa à etapa ‘Reta perpendicular das vizinhanças’.

É possível entender que as coordenadas que apresentam os valores médios das coordenadas dos pixels vizinhos não são necessariamente valores inteiros. Um exemplo é apresentado na Figura 3.16, onde o pixel avaliado tem a coordenada central (1,1). Os pixels com menor distância d_{pixels} são o pixel da coordenada (0,0) e o pixel da coordenada (1,0); assim o pixel resultante entre esses 2, com uso da equação (3.3), tem a coordenada (0.5,0).

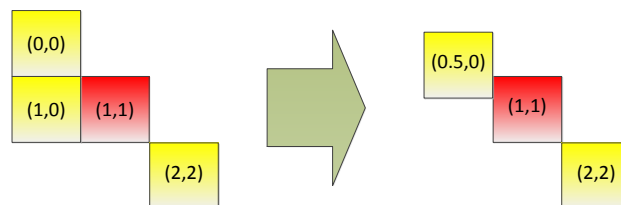


Figura 3.16 - Exemplo de média de coordenada de pixels vizinhos.

3.4.2 Reta perpendicular das vizinhanças:

Uma vez que as coordenadas dos dois pixels apresentam as vizinhanças, obtém-se o ângulo entre os dois vizinhos do pixel avaliado. Denominando (x_{bnew1}, y_{bnew1}) e (x_{bnew2}, y_{bnew2}) as coordenadas em duas dimensões do vizinho 1 e do vizinho 2 respectivamente, o ângulo entre eles é obtido pela equação (3.4):

$$\alpha = \text{atan2} (y_{bnew2} - y_{bnew1}, x_{bnew2} - x_{bnew1}) \quad (3.4)$$

Onde ‘*atan2*’ apresenta a função inversa da tangente, mas com dois argumentos. Para cada número real e com valores de x e y diferentes de zero, $\text{atan2}(y, x)$ é o ângulo em radianos entre o eixo positivo x de um plano e um ponto em coordenadas (x, y) . O ângulo é positivo para ângulos em sentido anti-horário e negativo para ângulos em sentido horário. O objetivo de usar dois argumentos em lugar de um, com, por exemplo, computar $\text{atan}(y/x)$, é manter a informação dos sinais das entradas, com objetivo de retornar o quadrante apropriado do ângulo computado, que não seria possível com a função *arctangent*.

Somando-se 90 graus ao ângulo α , pode-se obter o ângulo de uma linha perpendicular, a qual permitiria avaliar a existência de pixels de borda de possíveis linhas paralelas. Tem-se assim duas direções para serem testadas, em sentido positivo e negativo. Assim como apresentado na Figura 3.17.

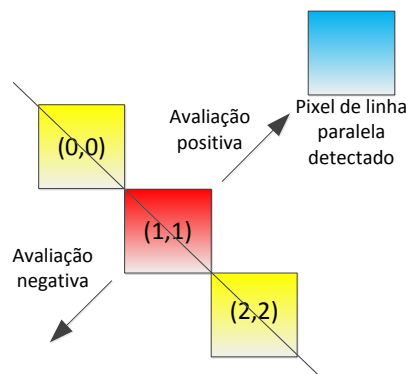


Figura 3.17 - Avaliação de pixels localização de pixels de linhas paralelas.

A Figura 3.17 ilustra como os pixels de borda das linhas paralelas são procurados numa linha reta nos dois sentidos, separadamente. Para o teste de pixels em cada uma das direções é usado um acumulador, que em cada pixel testado, se a avaliação indica que não é borda, o acumulador aumenta em uma unidade. Caso o acumulador atinja um valor limite, o

algoritmo entende que não foi detectada borda, assim o acumulador é reiniciado em ‘0’ e a avaliação é feita no outro sentido com o mesmo procedimento.

Caso o algoritmo detecte um pixel de borda localizado em algum sentido de avaliação, o valor do acumulador é usado como valor de distância medida entre os pixels. Essa distância é representada com o símbolo d_{line} e tem como unidade a quantidade de pixels entre linhas.

3.4.3 Avaliação de intensidades

Quando é detectado um pixel de linha paralela, o procedimento é fazer uma nova avaliação. Sendo a coordenada do pixel avaliado (x_{av}, y_{av}) , e a coordenada do pixel de linha paralela detectado (x_{lp}, y_{lp}) , a coordenada de um pixel localizado no ponto médio entre esses dois pixels (x_{middle}, y_{middle}) teria as coordenadas apresentadas na equação (3.5):

$$\begin{pmatrix} x_{middle} \\ y_{middle} \end{pmatrix} = \begin{pmatrix} (x_{av} + x_{lp})/2 \\ (y_{av} + y_{lp})/2 \end{pmatrix} \quad (3.5)$$

Trabalhando sobre a imagem de solda em tons de cinza $B_c(x, y)$, é avaliado se o valor de intensidade da coordenada do pixel do meio $B_c(x_{middle}, y_{middle})$ é menor que o valor de intensidade na coordenada do pixel avaliado $B_c(x_{av}, y_{av})$, assim como menor que o valor de intensidade na coordenada do pixel da linha paralela detectado $B_c(x_{lp}, y_{lp})$.

Uma nova matriz é criada, estimada da matriz de linha de solda, descrita como matriz ‘Matriz estimada de linha de solda’ apresentada como $B_{els}(x, y)$. Em caso de se confirmar a condição proposta anteriormente, a coordenada do pixel da linha paralela e a coordenada do pixel avaliados dentro da matriz estimada de linha de solda, são consideradas como pixels de bordas, como apresentado na equação (3.6).

$$\begin{aligned} & [B_{els}(x_{av}, y_{av}) = 1] \wedge [B_{els}(x_{lp}, y_{lp}) = 1] \\ & \text{se e somente se} \\ & [B_c(x_{av}, y_{av}) > B_c(x_{middle}, y_{middle})] \wedge [B_c(x_{lp}, y_{lp}) > B_c(x_{middle}, y_{middle})] \end{aligned} \quad (3.6)$$

A ‘Matriz estimada de linha de solda’ $B_{els}(x, y)$ obtida é apresentada na Figura 3.18:

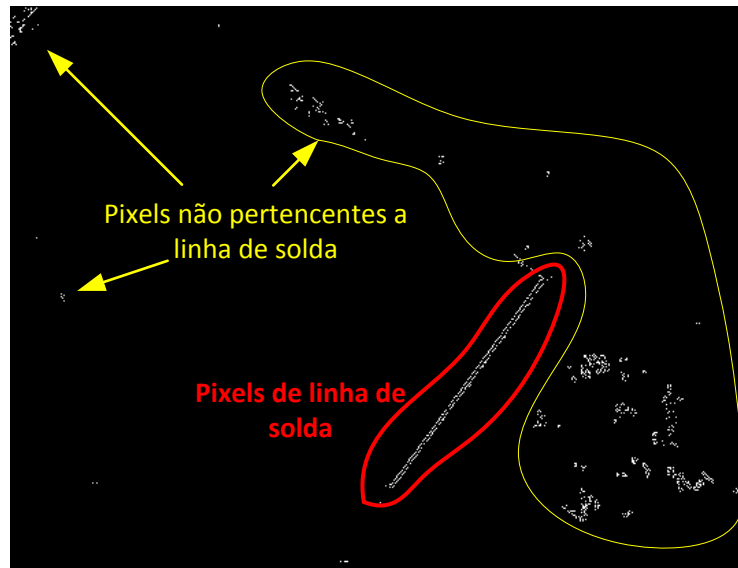


Figura 3.18 - Matriz estimada de linha de solda $B_{els}(x, y)$.

Idealmente na Figura 3.18 é esperado que os pixels presentes sejam só pixels do conjunto de pixels da linha de solda (pixels de borda dentro da linha vermelha); porém como é ilustrado e indicado com as setas amarelas na imagem tem-se pixels que não pertencem a esse conjunto.

Da Figura 3.18, percebe-se que o conjunto de pixels que pertencem à verdadeira linha de solda está contido na linha com maior comprimento. Por esse motivo o passo seguinte é procurar a linha mais longa de pixels e fazer a eliminação de todos os demais pixels. Os passos da aplicação para obter o objetivo do algoritmo são descritos a seguir, tomando como exemplo a imagem da Figura 3.18.

3.4.4 Detecção do pixel do final da linha

Inicialmente é feita uma varredura em toda a imagem capturada, testando todos os pixels. Sendo detectado um pixel com valor 1 (cor branco do pixel), são avaliados os 8 vizinhos denominados vizinhança-8. A avaliação considerada é detectar o pixel do final da linha com as seguintes condições:

1- o pixel é aceito como final de linha se possui só um vizinho. Na Figura 3.19(a) o pixel vermelho é considerado final da linha, e os pixels amarelos são os pixels de borda nas proximidades.

2- caso o pixel possua dois vizinhos, o pixel é considerado final de linha, somente se as coordenadas dos vizinhos apresentam uma distância euclidiana entre eles de um pixel em vizinhança-4; o valor da distância é calculado entre os pixels vizinhos com a equação (3.2). Na Figura 3.19(b) o pixel do final da linha é o pixel vermelho e os pixels amarelos são os pixels de bordas nas proximidades.



Figura 3.19 - Dois casos onde um pixel é considerado final de linha – (a) um pixel de vizinhança, (b) dois pixels de vizinhança.

3.4.5 Medição de comprimento de linhas de borda

Uma vez detectado o pixel do final da linha, é feito um seguimento dos pixels vizinhos com o objetivo de percorrer e medir cada linha do conjunto de pixels na totalidade de seu comprimento. Para isso é utilizado o algoritmo Ligação de borda preditiva (*Predictive Edge Linking* ou *PEL*) na seção ‘Criar Segmentos: união de pixels contíguos para criação de correntes de pixels’ conforme descrito por Akinlar e Chome, 2016. O objetivo procurado é percorrer uma linha de pixels de borda ao longo de todo seu comprimento, usando um algoritmo de predições com a tentativa de escolher as corretas decisões. Essas predições são baseadas em direções de deslocamentos anteriores entre pixels.

Cada linha é identificada com um número, n_{line} com valor inicial ‘1’ e incrementando em uma unidade por cada linha nova. Em conjunto é adicionado um contador $cont_{line}$ que indica a quantidade de pixels contidos em cada linha. O contador $cont_{line}$ é iniciado em ‘0’ em cada linha nova escolhida, assim para cada novo pixel detectado com o algoritmo PEL, o

valor de $cont_{line}$ incrementa em uma unidade, gerando-se uma Tabela 3.2, que apresenta um vetor representando N linhas detectadas.

Tabela 3.2 - Descrição de linhas de borda por maior comprimento.

n_{line}	$cont_{line}$
1	Nº pixels linha 1
2	Nº pixels linha 2
N	Nº pixels linha N

3.4.6 Seleção de linha de borda que apresenta a linha de solda:

Avaliando informações na Tabela 3.2 seleciona-se a linha com o maior número de pixels de borda, e todas as demais linhas são eliminadas. Certificando-se de que os pixels da linha escolhida pertencem à linha de solda, é criada uma matriz nova denominada ‘Matriz de linha de solda’ $B_{ls}(x,y)$, que contem unicamente os pixels de borda escolhidos, como ilustrado na Figura 3.20.

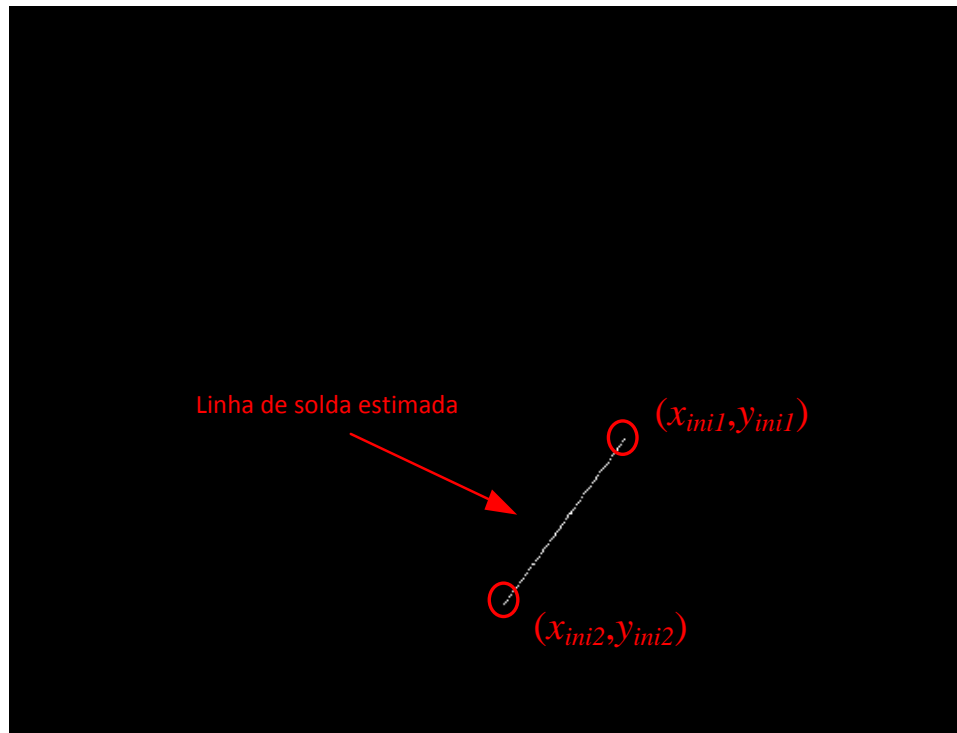


Figura 3.20 - Matriz de linha de solda $B_{ls}(x, y)$.

Uma vez detectado o conjunto de pixels que pertencem ao conjunto de pixels de linha de solda, percebe-se que existem ainda pixels de borda na matriz de bordas iniciais $B_{bi}(x, y)$ que devem de ser adicionados na matriz $B_{ls}(x, y)$. Os conjuntos de pixels que ainda não foram detectados são:

- a) pixels da linha paralela à linha detectada na Figura 3.20;
- b) em muitos casos a linha da Figura 3.20 não apresenta a totalidade da linha de solda, por isso têm que ser adicionados os outros pixels nos extremos da linha até serem adicionados os pixels de todo o conjunto de linha de solda;
- c) pixels paralelos aos pixels adicionados no item b);

3.4.6.1 Detecção de pixels de linha paralela

Para detectar e adicionar o conjunto de pixels das linhas paralelas da linha da ‘Matriz de linha de solda’ $B_{ls}(x, y)$ apresentada na Figura 3.18, trabalha-se sobre a ‘Matriz estimada de linha de solda’ $B_{els}(x, y)$, apresentada na Figura 3.20, avaliando as coordenadas de cada

pixel de borda da linha da ‘Matriz de linha de solda’ $B_{ls}(x, y)$. São procuradas as coordenadas dos dois pixels do final da linha em $B_{ls}(x, y)$, com o algoritmo de ‘detecção do pixel do final da linha’. Uma vez detectadas as coordenadas dos mesmos, são indicados seus valores como (x_{ini1}, y_{ini1}) , e (x_{ini2}, y_{ini2}) respetivamente. Escolhida a coordenada do final de linha (x_{ini1}, y_{ini1}) , e aplicando-se ‘Criar Segmentos: união de pixels contíguos para criação de correntes de pixels’ do algoritmo PEL são percorridos todos os pixels de borda até chegar ao outro extremo do final de linha (x_{ini2}, y_{ini2}) .

Para cada pixel de borda percorrido é buscada uma linha reta interpolada entre seus vizinhos com a técnica de ‘detecção de vizinhanças’ descrita anteriormente. Assim somando ou subtraindo 90 graus ao ângulo da reta é obtido o valor perpendicular da mesma. Nessas direções são buscados os pixels de borda da linha paralela e medida a distância em pixels com a técnica ‘reta perpendicular das vizinhanças’.

Critério para escolha do sinal da reta perpendicular

Para que ao longo de toda a linha percorrida a direção perpendicular avaliada se mantenha positiva ou negativa é preciso seguir um critério no momento de escolher somar ou subtrair 90 graus no ângulo da linha tangencial. Tendo em conta que com o algoritmo de PEL é possível ter informação da direção de movimentação do passo quando um novo pixel de borda é avaliado, o ângulo escolhido para a linha paralela estaria condicionado a esta última direção. Assim a distância positiva ou negativa é obtida diretamente somando ou subtraindo 90 graus na equação (3.7).

$$\alpha_{PEL} = \begin{cases} \operatorname{atan2}(y_{b2} - y_{b1}, x_{b2} - x_{b1}) & \text{if } PEL = \begin{cases} \text{Down,} \\ \text{Down_Left,} \\ \text{Down_Right,} \\ \text{Right} \end{cases} \\ \operatorname{atan2}(y_{b2} - y_{b1}, x_{b2} - x_{b1}) + \pi & \text{if } PEL = \begin{cases} \text{Up,} \\ \text{Up_Left,} \\ \text{Up_Right,} \\ \text{Left} \end{cases} \end{cases} \quad (3.7)$$

As 8 direções de PEL apresentam as possíveis direções do passo entre o pixel borda anterior e o pixel borda a ser avaliado (a seta verde na Figura 3.21). Nem sempre a direção percorrida pelo PEL é igual ao ângulo da reta tangencial, mas esta é uma estimativa útil para escolha do sinal do ângulo. Na Figura 3.21 são apresentados 8 exemplos do ângulo α_{PEL} obtido com o critério da equação (3.7), na qual a linha vermelha que apresenta a distância

positiva aponta sempre para o meio e a linha azul que apresenta a distância com sinal negativo, aponta para fora.

Deve-se notar que sem considerar este critério o conjunto de pixels que tem a seta verde com direções diferentes de 180 graus, por exemplo, abaixo e acima, teriam a sua linha perpendicular positiva apontando no mesmo sentido. Isso seria totalmente indesejável no caso de haver uma linha de solda com as duas direções, como o caso de uma linha de solda curva, o qual daria como resultado distâncias com sinais errados, impossibilitando ao algoritmo detectar a linha de solda em todo seu comprimento.

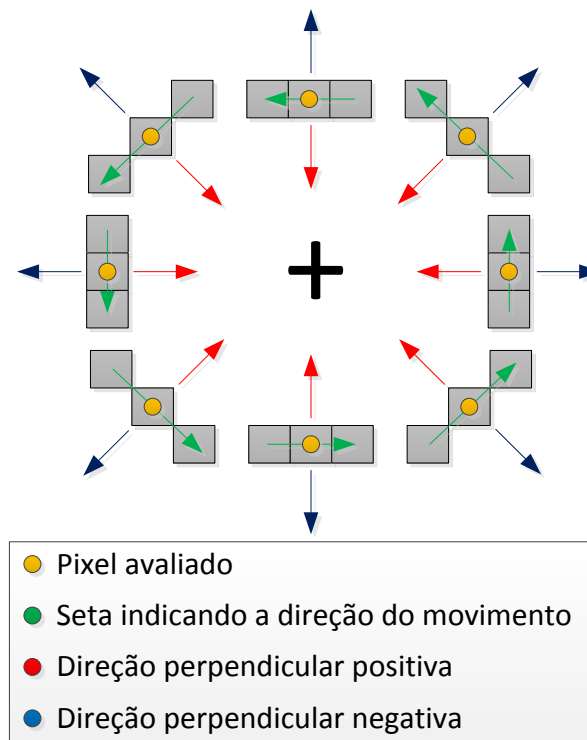


Figura 3.21 - Exemplo de critério do sinal na linha perpendicular.

Detecção de pixels

Uma vez que um critério sobre o sinal da linha perpendicular seja definido, os pixels são procurados nas duas direções. Cada pixel de linha paralela detectado é salvo em um vetor:

- conjunto de coordenadas de pixels paralelos detectados (x_{lp}, y_{lp}) ;
- conjunto de coordenadas de pixels avaliados (x_{av}, y_{av}) ;
- o conjunto de distâncias entre o conjunto de pixels da linha paralela detectados e os pixels avaliados d_{line} ;

- o sentido de cada uma das distâncias. Positivo se o pixel paralelo detectado foi com avaliação em sentido positivo, e negativo em caso contrário;

O vetor é apresentado na Tabela 3.3, para N conjuntos de pixels detectados. A expressão ‘*sign*’ indica o sentido da distância entre os pixels.

Tabela 3.3 - Vetor com coordenadas de linha de solda e suas distâncias.

Pixel paralelo (x)	Pixel paralelo (y)	Pixel avaliado (x)	Pixel avaliado (y)	Comprimento de linhas (pixels)
$x_{lp}(1)$	$y_{lp}(1)$	$x_{av}(1)$	$y_{av}(1)$	$(sign) d_{lines}(1)$
$x_{lp}(2)$	$y_{lp}(2)$	$x_{av}(2)$	$y_{av}(2)$	$(sign) d_{lines}(2)$
$x_{lp}(N)$	$y_{lp}(N)$	$x_{av}(N)$	$y_{av}(N)$	$(sign) d_{lines}(N)$

Apenas um sentido de distância entre pixels deve ser escolhido, já que a linha paralela é localizada unicamente em um lado da linha da Figura 3.20. O método para discriminar o sentido consiste em avaliar o vetor obtido anteriormente, usar um acumulador que indique a quantidade de distâncias positivas e a quantidade de distâncias negativas. Finalmente é eliminado o conjunto de distâncias com o acumulador indicando a menor quantidade, ficando com distâncias de um só sentido.

Na prática o conjunto de distâncias entre as linhas de borda d_{lines} da solda apresenta um modelo completamente condizente com uma distribuição Gaussiana, por seguir um valor de distância relativamente constante, como é possível notar na imagem ampliada de uma junta de topo na Figura 3.22.



Figura 3.22 - Imagem ampliada de linha de solda em junta topo para apresentar as distâncias relativamente constantes.

O conjunto de valores de distâncias permite definir uma medida da média e do desvio padrão, obtidas pelas equações (3.8).

$$\begin{aligned}\mu_{lines} &= 1/N \sum_{n=1}^N d_{line}(n) \\ \sigma_{lines} &= \sqrt{1/N \sum_{n=1}^N (d_{line}(n) - \mu_{lines})^2}\end{aligned}\tag{3.8}$$

Por último, as coordenadas dos pixels paralelos detectados no sentido de distâncias dominantes, são adicionados como pixels de borda na ‘Matriz de linhas de solda’ $B_{ls}(x, y)$. Todas as coordenadas dos pixels na matriz de estimação de linha de solda $B_{els}(x, y)$, assim como da ‘Matriz de bordas inicial’ $B_{bi}(x, y)$ que coincidem com os pixels de borda da matriz de linha de solda $B_{ls}(x, y)$, passam a ter um valor de 2, para que os mesmos possam ser identificados, como indica a equação (3.9).

$$(B_{els}(x, y) = 2 \wedge B_{bi}(x, y) = 2) \Leftrightarrow B_{ls}(x, y) = 1\tag{3.9}$$

3.4.6.2 Avaliação e adição de pixels

Os extremos da linha da ‘Matriz de linha de solda’ $B_{ls}(x, y)$, apresentada na Figura 3.18, podem não apresentar o final da linha de solda por vários motivos: Na criação da ‘Matriz estimada de linha de solda’ $B_{els}(x, y)$, apresentada na Figura 3.20, podem existir pixels de linha de solda da ‘Matriz de tons de cinza’ $B_c(x, y)$ que não cumpram com a condição de avaliação de intensidades, tendo como resultado uma linha que não abranja por completo a linha de solda. Um exemplo é exibido na imagem ampliada da Figura 3.23, onde a linha de solda da $B_{els}(x, y)$ é apresentada como uma linha vermelha. Com o objetivo de solucionar o problema descrito, propõe-se esta etapa:

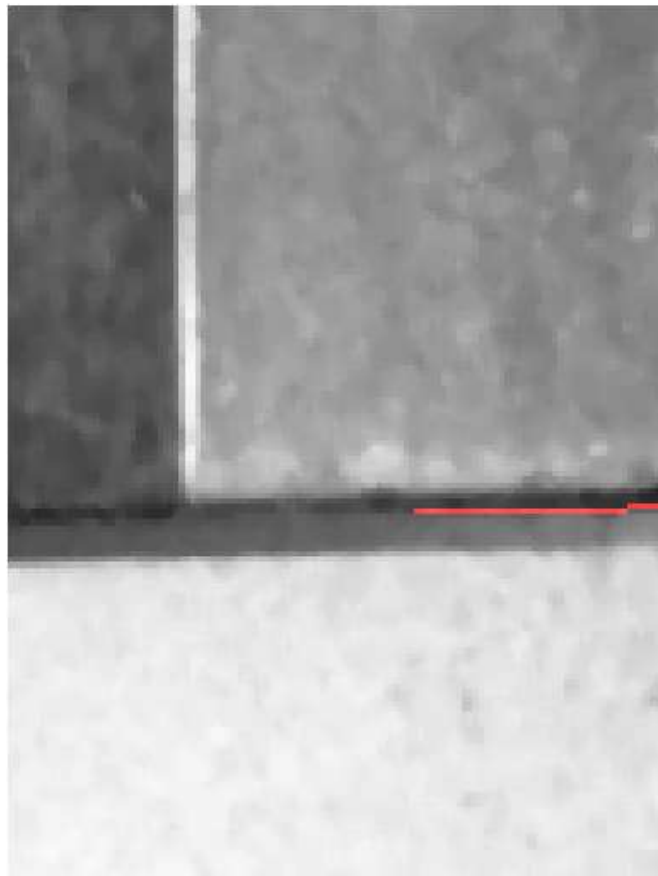


Figura 3.23 - Imagem ampliada de linha de borda que não abrange a linha de solda na totalidade.

As matrizes a serem usadas são a ‘Matriz de linha de solda’ $B_{ls}(x, y)$ (Figura 3.20) e a matriz de bordas iniciais $B_{bi}(x, y)$ (Figura 3.14). Primeiramente é escolhida a coordenada de um extremo de linha, (x_{ini1}, y_{ini1}) já detectado anteriormente, para processar a imagem com os

passos a serem descritos em seguida, finalmente é selecionado o outro extremo para também implementar os mesmos passos.

Uma vez detectada a coordenada do pixel de final de linha (x_{ini1}, y_{ini1}) , deve-se lembrar que esse valor de pixel na matriz de bordas inicial teria um valor de $B_{bi}(x_{ini1}, y_{ini1})=2$. Assim é possível procurar um pixel na vizinhança da coordenada $B_{bi}(x_{ini1}, y_{ini1})$ que ainda não foi adicionado na matriz $B_{bi}(x, y)$, com um valor de borda de $B_{bi}(x, y) = 1$. Caso se encontre um pixel com esse valor, o mesmo deve ser avaliado para uma possível incorporação nesta última matriz.

Os procedimentos de avaliação sobre o novo pixel de borda detectado, são desenvolvidos unicamente sobre a matriz de bordas iniciais $B_{bi}(x, y)$, e descritos nos seguintes passos:

1. uma reta é criada interpolando os vizinhos do pixel avaliado fazendo uso da técnica de ‘Detecção e média de vizinhanças’. Considera-se um valor de pixel borda na vizinhança será igual a ‘2’, mas nessa técnica o algoritmo entende que esse valor também pertence a um valor de pixel de borda.
2. caso não seja um pixel de final de linha, é somando 90 graus ao ângulo da reta para obter o valor perpendicular da mesma. Nessa direção são procurados os pixels de borda da linha paralela e medida a distância d_{line} com a técnica ‘reta perpendicular das vizinhanças’.
3. caso seja detectado um pixel na linha paralela, é avaliado se o valor da distância entre os dois pixels ingressa no intervalo $(\mu_{line} \pm 1,5\sigma_{line})$.

Em caso afirmativo são adicionados ao vetor apresentado na Tabela 3.3, os seguintes valores:

- conjunto de coordenadas de pixels paralelos detectados.
- conjunto de coordenadas de pixels avaliados.
- valor de distância entre os pixels.

Com o objetivo de adaptar os valores da equação Gaussiana que apresenta a distância entre linhas paralelas d_{line} , são atualizados os valores da média μ_{line} e desvio padrão σ_{line} . A atualização é feita considerando todos os valores de distância d_{line} da Tabela 3.3 já atualizada, e implementando a equação (3.8).

Finalmente as coordenadas de pixel paralelo e do pixel avaliado são adicionadas na matriz de linha de solda com o valor de $B_{ls}(x, y) = 1$. Assim também os valores da coordenada do pixel na matriz de bordas iniciais $B_{bi}(x, y)$ e na matriz estimada de linha de solda $B_{els}(x, y)$ são igualados a '2' como indicado na equação (3.9).

Em caso negativo indicaria que o pixel avaliado e o pixel paralelo detectado já não pertencem a linha de solda. Neste caso são implementados os seguintes procedimentos:

- não é adicionado valor no vetor da Tabela 3.3, e não são atualizados os valores de media ' μ_{line} ' e desvio padrão ' σ_{line} ' da equação (3.8).
- na 'Matriz de linha de solda' $B_{ls}(x, y)$ os valores dos pixels das coordenadas dos pixels avaliados e do pixel da linha paralela são considerados como pixels de linha de solda com o valor $B_{ls}(x, y) = 1$. As mesmas coordenadas na 'Matriz de bordas iniciais' $B_{bi}(x, y)$ e na 'Matriz de estimação de linha de solda' $B_{els}(x, y)$ são modificadas para um valor de 2, como indicado na equação (3.9).
- define-se um acumulador, denominado 'acumulador de erro em linha de solda', que incrementa seu valor em uma unidade por cada pixel de linha paralela fora do intervalo estabelecido. No momento do acumulador atingir um limite, o algoritmo finaliza a análise do extremo da linha em que é avaliado, reiniciando em zero para serem feitos todos os procedimentos já descritos mas agora no outro extremo (x_{ini2}, y_{ini2}) da linha da 'Matriz de linha de solda' $B_{ls}(x, y)$ da Figura 3.20. Uma vez que o acumulador atinja o máximo valor nas duas linhas, o algoritmo de detecção de solda finaliza.

3.4.7 Critério para escolha do ponto de solda

Fazendo uso das coordenadas salvas na Tabela 3.3, uma possibilidade para obtenção das coordenadas dos pixels de solda pode ser através do valor médio entre os dois pixels de linhas paralelas, porém podem ocorrer casos em que, por reflexos ou riscos indesejáveis, as linhas de borda que representam as linhas de solda flutuam de tamanho e o ponto médio difere do ponto de solda verdadeiro. Um exemplo pode ser observado na Figura 3.24(a), onde os pontos apresentam um leve afastamento da linha de solda, e por uma variação das linhas paralelas, o erro ainda se amplia.

A escolha para o ponto de solda consiste em testar todos os pontos localizados entre os dois pixels de bordas de linhas paralelas da Tabela 3.3, para escolher o pixel na menor intensidade da imagem em tons de cinza. Caso seja detectado mais de um pixel com mesmo valor da menor intensidade, o pixel de solda será o valor de coordenada da média entre as coordenadas dos pixels de menor intensidade. O exemplo da robustez deste método pode ser notado na Figura 3.24(b), onde os pontos de solda foram localizados no centro da linha de solda independentemente de uma variação nas linhas de borda, reduzindo consideravelmente o erro.

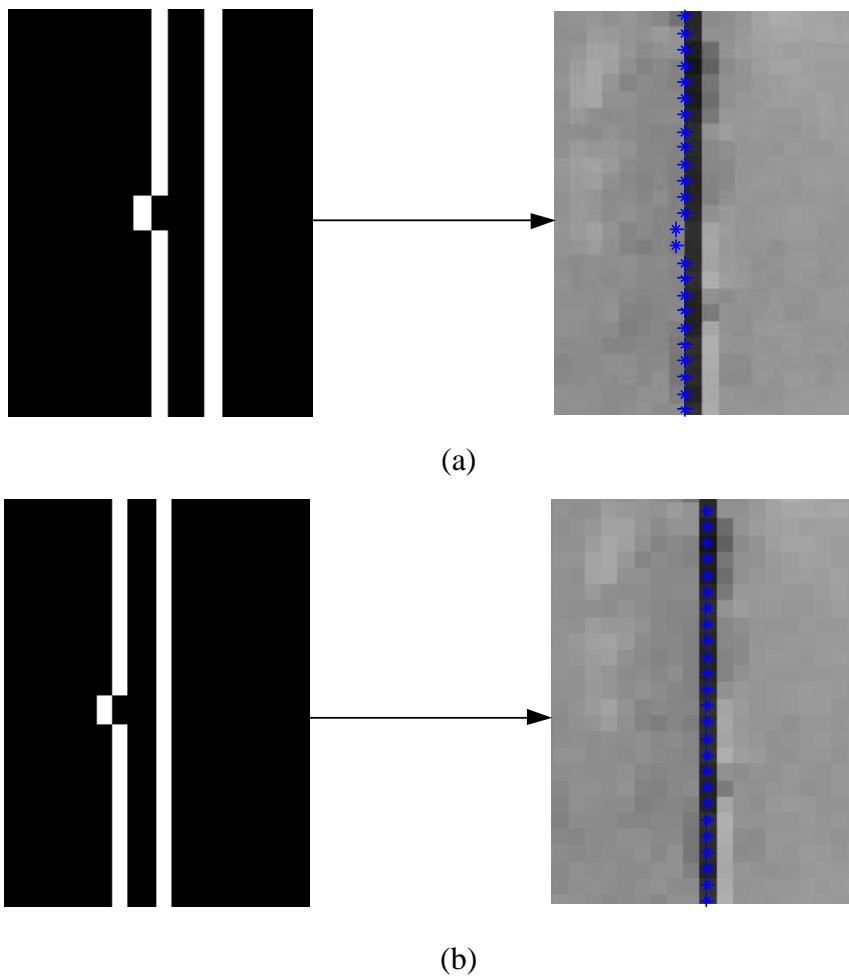


Figura 3.24 - Diferenças entre escolhas de pontos de solda (a) ponto central, (b) ponto de menor valor de intensidade.

Com este novo conjunto de coordenadas obtidas, e sobrepondo-o à imagem original das peças de solda, obtém-se a linha indicada na Figura 3.25.

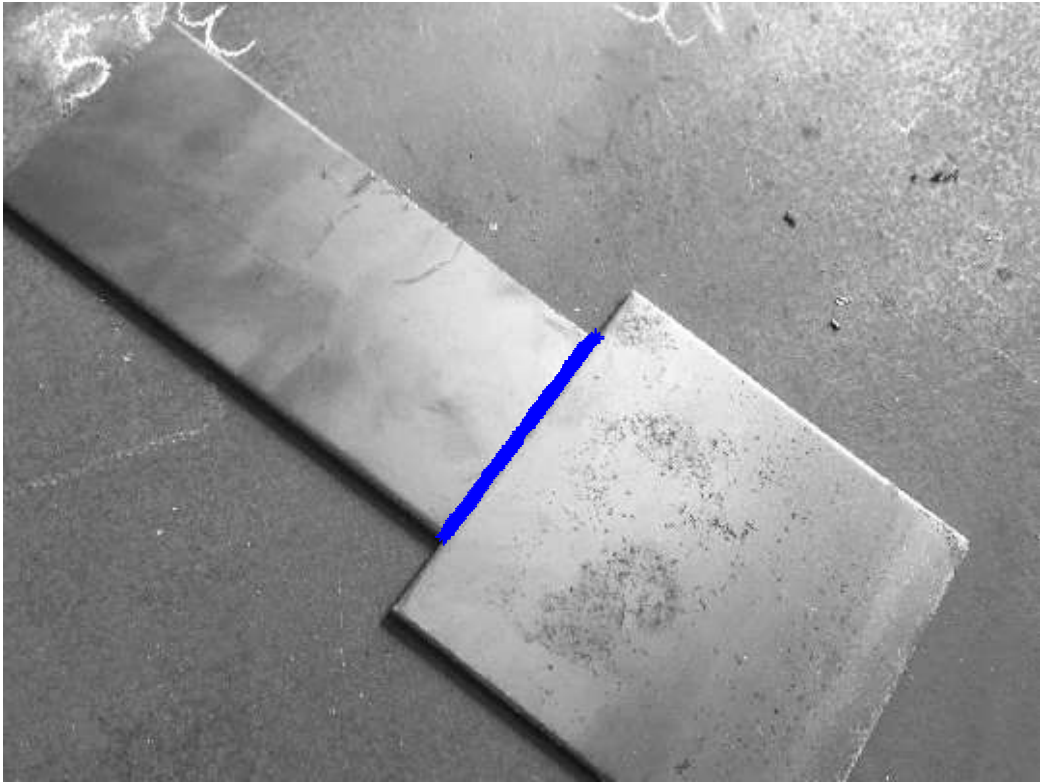


Figura 3.25 - Linha de solda detectada em peças de trabalho.

3.5 Solução de linha de solda descontínua

Em alguns casos, por problemas de iluminação ou ângulo da câmera, a matriz de bordas inicial $B_{bi}(x,y)$ não apresenta continuidade no conjunto de pixels de borda que compõem as linhas de solda. Um exemplo de uma Região de Interesse (ROI) na matriz de bordas inicial com uma linha de solda não completa é apresentada na Figura 3.26.

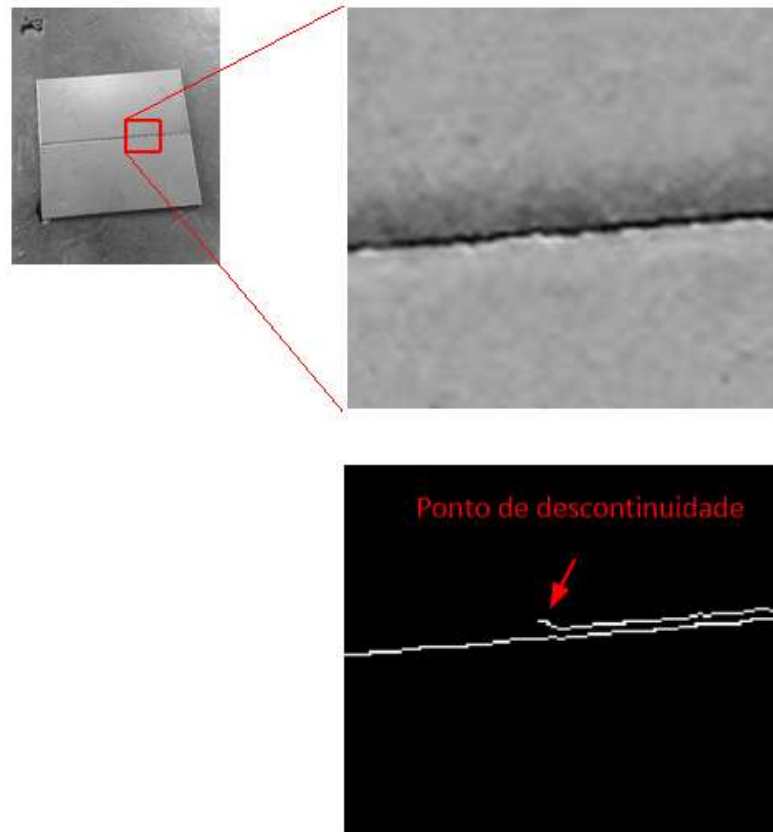


Figura 3.26 - Região de interesse (ROI) com linha de solda não completa.

Nesse caso a verificação do algoritmo se conclui ao atingir um pixel de final de linha, o que seria uma maneira incorreta de finalizar, já que o esperado é que o algoritmo finalize ao atingir o limite do acumulador de erro em linha de solda.

São propostos uma série de procedimentos para conseguir a correção da linha de soldagem na ‘Matriz de bordas inicial’ $B_{bi}(x, y)$. Como anteriormente descrito, o mapa de bordas inicial é fundamental para a avaliação da linha de solda, não só na técnica apresentada no trabalho atual, como em outros trabalhos [Shi et al., 2009, Dinham e Fang, 2013]. Caso exista uma descontinuidade da linha de pixels de borda que apresentam a linha de solda nesta matriz, a técnica descrita anteriormente daria como resultado uma linha de solda que não abrange a totalidade da verdadeira linha.

Os procedimentos para solucionar um problema na descontinuidade são:

- aplicação de técnicas para correção de bordas, encarregadas de trabalhar e corrigir localmente os conjuntos de pixels inclusos em uma ROI com o pixel detectado como final da linha no centro da mesma;
- garantir que o pixel do final da linha tenha conexão com os novos pixels de bordas detectados, para isso se aplica a técnica de ‘Análise e correção de conectividade’;
- adição do conjunto de pixels novos na ‘Matriz de bordas inicial’ $B_{bi}(x,y)$;
- aplicação de uma seção do algoritmo PEL para percorrer os novos pixels e avaliar se os mesmos pertencem à linha de solda;

O fluxo de procedimentos é apresentado na Figura 3.27.

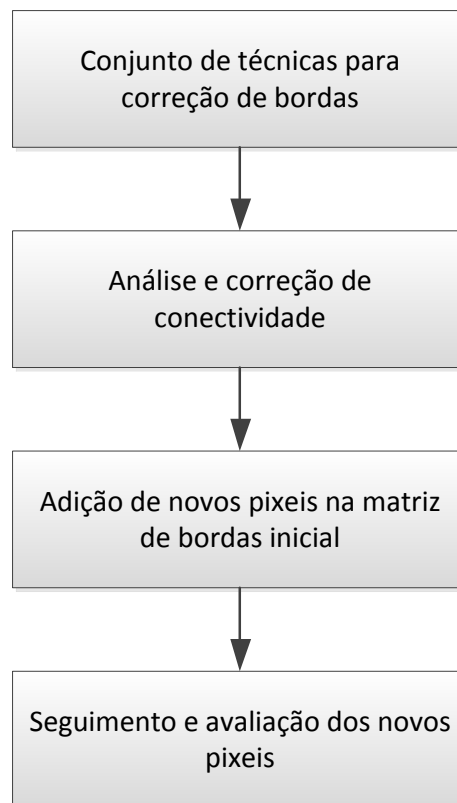


Figura 3.27 - Fluxograma solução de descontinuidade na ‘Matriz de bordas inicial’ $B_{bi}(x,y)$.

3.5.1 Técnicas para correção de bordas

Neste conjunto de técnicas se trabalha com a matriz de imagem cinza originalmente obtida $B_c(x, y)$, e uma Região de Interesse (ROI) denominada $W_{ROI}(x, y)$ de tamanho $K \times L$, onde $K < M$ e $L < N$, com $M \times N$ o tamanho da imagem total. O conjunto de pixels $W_{ROI}(x, y)$ possui no seu centro a coordenada do pixel de descontinuidade (x_f, y_f) da matriz de bordas inicial $B_{bi}(x, y)$. A relação entre $B_c(x, y)$ e $W_{ROI}(x, y)$ respeita a equação (3.10):

$$w(i, j) = f(i + x_f - K/2, j + y_f + j - N/2) \quad (3.10)$$

Onde $i = 0, 1, 2, \dots, K-1$ e $j = 0, 1, 2, \dots, N-1$. A janela $W_{ROI}(x, y)$ envolve todos os pixels de $B_c(x, y)$ dentro da área de superposição, como apresentado na Figura 3.28.

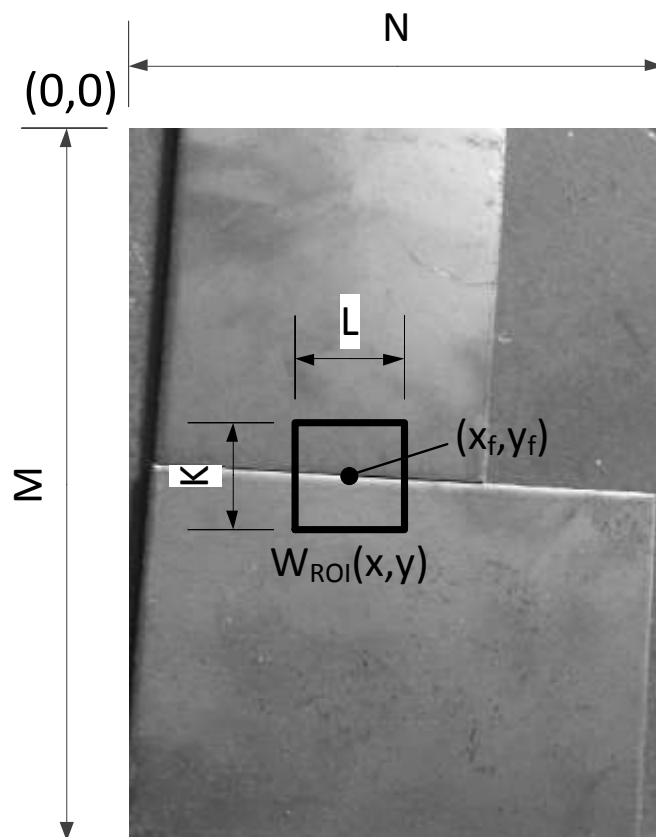


Figura 3.28 - Janela $W_{ROI}(x,y)$ dentro da matriz $B_c(x, y)$.

O problema de descontinuidade da linha dentro da `Matriz de bordas iniciais` $B_{bi}(x, y)$ ocorre por causa de uma linha de solda com regiões onde o detector de bordas usado não detecta bordas. Assim, tendo em conta que na matriz $B_c(x,y)$ a intensidade da região de linha de solda é menor que nas bordas, é proposta uma técnica de erosão em imagens cinza, com objetivo de um incremento da largura de linha de solda. Seja B_{circ} o elemento de estrutura representado com uma circunferência de raio r_{circ} , a erosão com a imagem $W_{ROI}(x,y)$ é definida pela equação (3.11):

$$\begin{aligned} (W_{ROI}(x, y) \ominus B_{circ})(x, y) &= \min \{W_{ROI}(x+i, y+j)\} \\ \text{Com } i^2 + j^2 &\leq r_{circ}^2 \end{aligned} \quad (3.11)$$

Considerando um raio r_{circ} de valor $r_{circ}=3$, a imagem $W_{ROI}(x, y)$ após uma erosão de imagem cinza com o elemento B_{circ} , tem como resultado a Figura 3.29.

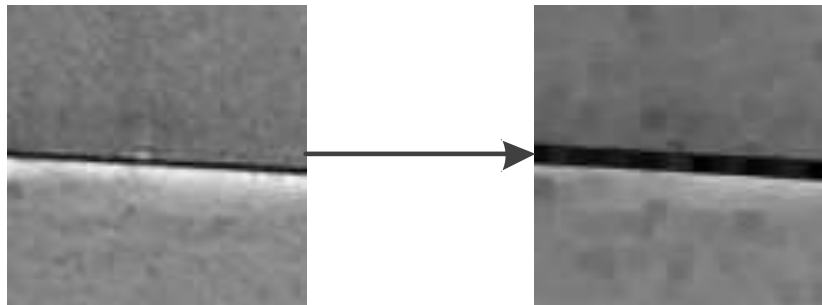


Figura 3.29 - Imagem $W_{ROI}(x, y)$ após o processo de erosão com um elemento de estrutura como uma circunferência de raio 3.

Para obter uma maior homogeneidade na linha de solda, e assim uma linha mais suave no momento de obtenção das bordas, é usado um filtro da mediana com a ROI $W_{ROI}(x, y)$ na imagem da Figura 3.29, cujo resultado é apresentado na Figura 3.30.

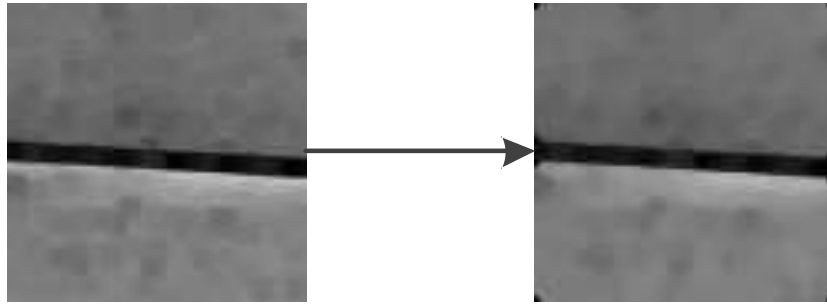


Figura 3.30 - Filtro da mediana sobre ROI $W_{ROI}(x, y)$.

Como ilustrado na Figura 3.30 percebe-se que o nível de intensidades permite fazer uma boa diferenciação entre as peças e a linha de solda, já que a iluminação é aproximadamente homogênea. Por este motivo é razoável fazer uma segmentação binária da imagem usando como limiar o método de minimização de inter-variância de classes proposto por Otsu [Otsu, 1979].

Uma vez obtido o valor de limiar ou *threshold*, a segmentação é realizada conforme apresentado na equação (3.12).

$$W_{ROI}(x, y) = \begin{cases} 1 \rightarrow W_{ROI}(x, y) \geq threshold \\ 0 \rightarrow W_{ROI}(x, y) < threshold \end{cases} \quad (3.12)$$

O resultado é apresentado na Figura 3.31:

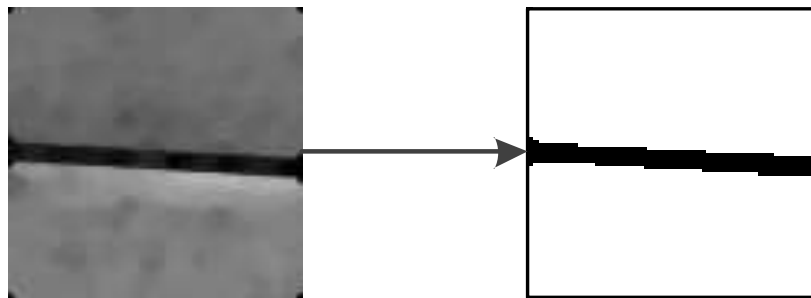


Figura 3.31 - Segmentação da linha de solda na imagem $W_{ROI}(x, y)$.

Para obter as bordas da linha de soldagem é aplicada a operação morfológica ‘remove’ em $W_{ROI}(x, y)$. Nesta operação é feita a erosão da imagem binária $W_{ROI}(x, y)$ com um elemento de estrutura ‘4-connected neighbour’ (o elemento é denominado B_{4C}) e a imagem obtida é subtraída dela mesma sem a erosão, como indica na equação (3.13):

$$W_{ROI}(x, y) = W_{ROI}(x, y) - (W_{ROI}(x, y) (-) B_{4C}) \quad (3.13)$$

O resultado é apresentado na Figura 3.32.

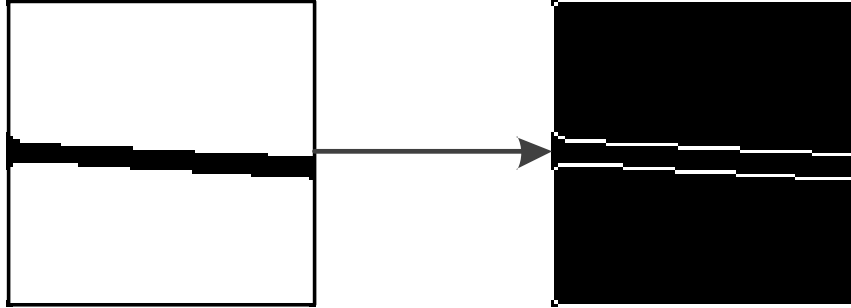


Figura 3.32 - Operação 'remove' na imagem $W_{ROI}(x, y)$.

3.5.2 Análise e correção de conectividade

Uma vez obtida a imagem $W_{ROI}(x, y)$ como mostrado na Figura 3.32, deve-se certificar que existe uma conectividade com o centro da imagem $W_{ROI}(x, y)$, com o pixel de coordenada (x_f, y_f) da imagem em tons cinza $B_c(x, y)$. Devido ao fato de ter sido usado um processo morfológico de erosão nos passos anteriores, a linha de solda aumenta sua largura se afastando do ponto do final de linha que anteriormente pertencia a linha de solda na 'Matriz de bordas inicial' $B_{bi}(x, y)$. Por isso é necessária a união da nova linha de solda, com as coordenadas do centro da figura $W_{ROI}(x, y)$, também chamado final de linha.

Detecção do pixel de borda mais próximo

Como solução é proposto um método baseado na seção denominada 'Preencher lacunas' (ou *Fill Gaps*) do método PEL, Akinlar e Chome, 2016. A Figura 3.33 [adaptada de Akinlar e Chome, 2016] ilustra o método de união entre o pixel do final da linha e as linhas de solda novas. A ideia é detectar o caminho reto e fazer a união entre o pixel do final da linha e um pixel de borda n pixels distantes em uma determinada direção. Tem-se 8 possíveis cenários de teste (para cada uma das possíveis direções), na Figura 3.33(a) é referenciado o exemplo de testar uma direção a direita, mas por simetria é facilmente trasladável acima, esquerda e abaixo; assim na Figura 3.33(b) fazendo exemplo das direções diagonais, a direção

é abaixo-direita, mas também por simetria é também aplicável para acima-direita, acima-esquerda e abaixo-esquerda.

Na ‘Matriz de bordas inicial’ $B_{bi}(x, y)$, a localização (x_f, y_f) apresenta o pixel final de linha, e a mesma tem só um pixel de borda como vizinhança. Com as coordenadas relativas entre esses dois pixels, é possível obter a direção dos mesmos e usar esse dado para desenvolver o método. Uma vez obtida a direção dos pixels, passa-se a trabalhar na janela $W_{ROI}(x, y)$, onde procura-se um caminho para a união do pixel central com as bordas da solda.

Como exemplo, na Figura 3.33(a) é apresentada uma seção da ROI $W_{ROI}(x, y)$, onde a coordenada do pixel (x_{fw}, y_{fw}) é o centro da janela $W_{ROI}(x, y)$ (lembrar que o centro de $W_{ROI}(x, y)$ apresenta o pixel de descontinuidade (x_f, y_f) de B_{bi}). A representação em cor cinza do pixel na coordenada (x_{fw}, y_{fw}) e o pixel em cor cinza claro na coordenada $(x_{fw} - 1, y_{fw})$ permitem obter o dado da direção, já que esses pixels de bordas existem na ‘Matriz de bordas inicial’ $B_{bi}(x, y)$, mas não necessariamente na Região de Interesse $W_{ROI}(x, y)$. Desse modo é possível notar que o pixel do meio está se movimentando na direção direita porque seu vizinho, distinguido com cinza claro, é localizado à esquerda, na coordenada relativa $(x_{fw} - 1, y_{fw})$.

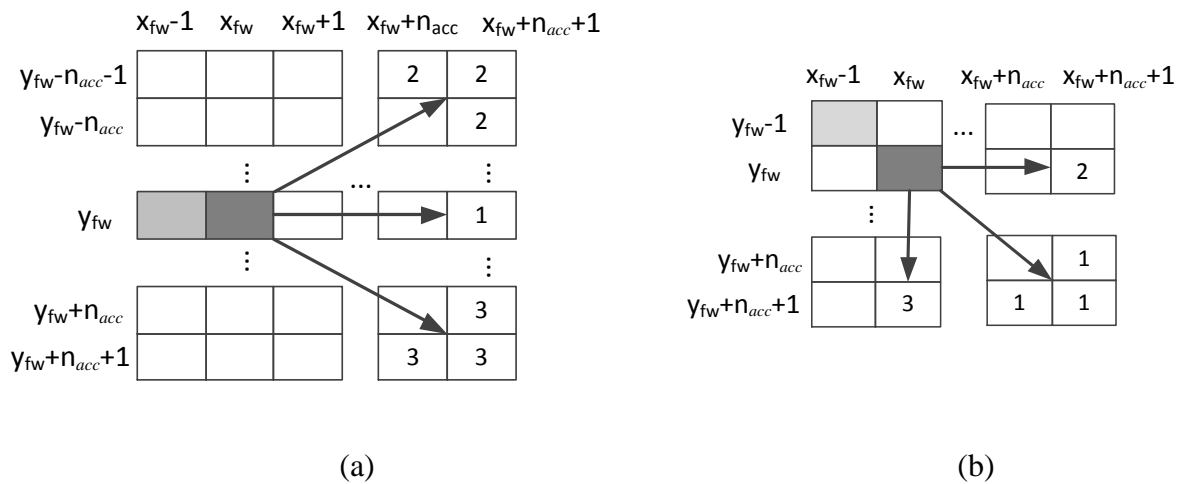


Figura 3.33 - Método de união entre o pixel do final da linha e linhas de solda novas.

Sendo n_{acc} um acumulador que inicia em $n_{acc} = 1$, para cada valor do mesmo é avaliado em primeiro lugar o pixel localizado em $(x_{fw} + n_{acc} + 1, y)$, marcado com ‘1’ na Figura 3.33(a), em caso de não se detectar pixel de borda, são avaliadas em segundo lugar as localizações de $(x_{fw} + n_{acc}, y_{fw} - n_{acc} - 1)$, $(x_{fw} + n_{acc} + 1, y_{fw} - n_{acc} - 1)$ e

$(x_{fw} + n_{acc} + 1, y_{fw} - n_{acc})$, marcadas com '2' na Figura 3.33(a), assim em caso de não detectar um pixel de borda, finalmente são avaliadas em terceiro lugar as coordenadas de $(x_{fw} + n_{acc}, y_{fw} + n_{acc} + 1)$, $(x_{fw} + n_{acc} + 1, y_{fw} + n_{acc} + 1)$ e $(x_{fw} + n_{acc} + 1, y_{fw} + n_{acc})$, marcadas com '3' na Figura 3.33(a). Caso não seja detectado pixel de borda nestas três avaliações, o valor de n_{acc} é incrementado em uma unidade e os três testes são avaliados novamente em suas respectivas ordens. O algoritmo finaliza quando é detectado um valor de borda na direção avaliada.

Correção de caminho

Após o algoritmo detectar um caminho entre um pixel de borda e o centro da imagem $W_{ROI}(x,y)$, os pixels localizados no menor caminho entre essas duas coordenadas são transformados em pixels de borda. Considerando a coordenada do pixel de borda detectado e a coordenada do centro da ROI W_{ROI} , o menor caminho é uma linha reta entre estas. Assim os pixels localizados nesta linha reta são transformados em pixels de borda, um exemplo onde o pixel de borda detectado é $(x_{fw} + 4, y_{fw} - 4)$ é apresentado na imagem da Figura 3.34.

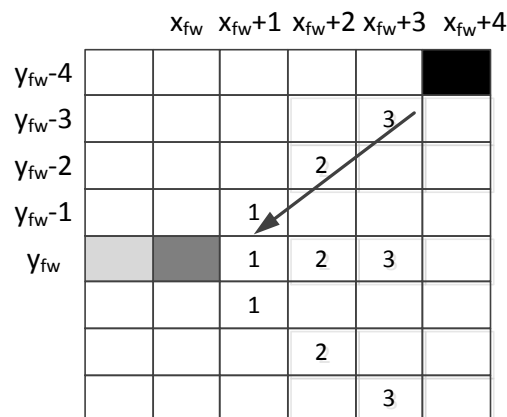


Figura 3.34 - Correção de caminho, os pixels entre a coordenada $(x_{fw} + 4, y_{fw} - 4)$ e o centro são transformados em pixels de borda.

3.5.3 Adição da ROI $W_{ROI}(x, y)$ na matriz de linha de solda $B_{ls}(x, y)$

O objetivo é centrar a nova janela de bordas corrigidas $W_{ROI}(x, y)$ na localização (x_f, y_f) dentro da matriz de bordas iniciais $B_{bi}(x, y)$. Como descrito anteriormente, a localização dos pixels que já pertencem à borda de verdadeiros positivos, é representada com um valor de ‘ $B_{bi}(x, y) = 2$ ’ na matriz de bordas iniciais, $B_{bi}(x, y)$. Com isso tem-se a possibilidade de igualar todos os valores das coordenadas dos pixels de $B_{bi}(x, y)$ com $W_{ROI}(x, y)$, nas coordenadas de sobreposição entre eles, mantendo os valores de $B_{bi}(x, y)$ que já foram considerados como pixels pertencentes à linha de solda, como apresentado na equação (3.14):

$$B_{bi}(x, y) = \begin{cases} W_{ROI}(x, y) \rightarrow (x_f - K/2 < x < x_f + K/2) \\ W_{ROI}(x, y) \rightarrow (y_f - L/2 < y < y_f + L/2) \\ B_{bi}(x, y) \rightarrow B_{bi}(x, y) = 2 \\ B_{bi}(x, y) \rightarrow else \end{cases} \quad (3.14)$$

Uma vez corrigida a seção da linha de solda com as coordenadas dentro da sobreposição da janela $W_{ROI}(x, y)$ com a matriz de bordas iniciais $B_{bi}(x, y)$, são avaliadas as condições para que os pixels adicionados sejam considerados ‘pixels de linha de solda’, como apresentado na seção ‘Avaliação e adição dos pixels nos extremos da linha de solda estimada’.

Para avaliar os novos pixels considera-se que a avaliação da media μ_{line} e do desvio padrão σ_{line} da equação (3.8) deve ser modificada, já que um processo de erosão foi aplicado anteriormente na linha de solda. O processo expande a largura da linha de solda em um valor $2 r_{circ}$ (lembrar que r_{circ} era o raio da estrutura circular no processo de erosão) em cada lado. Como são duas as linhas, a expansão da largura da linha de solda é de $4r_{circ}$.

O conjunto de distâncias medidas destes novos pixels com suas linhas paralelas, pode se aproximar ao conjunto de distâncias já medido anteriormente d_{line} salvas na matriz da Tabela 3.3, mas somado com uma constante $4 r_{circ}$. A relação da nova média μ_{line_new} e novo desvio padrão σ_{line_new} com os valores de média μ_{line} e desvio padrão σ_{line} da equação (3.8), são apresentados na equação (3.15).

$$\begin{aligned}\mu_{line_new} &= \mu_{line} + 4r_{circ} \\ \sigma_{line_new} &= \sigma_{line}\end{aligned}\quad (3.15)$$

Para avaliar o novo valor de distância medida d_{new} , deve-se verificar se o mesmo se enquadra no intervalo $(\mu_{line_new} \pm 1,5\sigma_{line_new})$. Caso o valor esteja no intervalo esperado, o passo seguinte é atualizar os valores de média μ_{line} e desvio padrão σ_{line} , o que não pode ser feito diretamente, já que o valor da distância foi aumentado em $4 r_{circ}$. Considerando que o valor de distância novo medido é representado pela variável d_{line_new} , o valor de distância d_{line} para atualizar a média μ_{line} e o valor do desvio padrão σ_{line} , é apresentado na equação (3.16):

$$d_{line} = d_{line_new} - 4r_{circ} \quad (3.16)$$

Por último são realizadas as etapas já descritas na seção ‘Avaliação e adição dos pixels nos extremos da linha de solda estimada’, com o armazenamento das coordenadas dos novos pixels de linha de solda detectados na matriz da Tabela 3.3, assim como o valor de distância da equação (3.16).

3.6 Correção dos pontos finais

No momento de avaliar os pontos de solda apresentados nos extremos da linha de solda, podem surgir alguns casos com condições não desejáveis. Nos cantos, com frequência são detectados pixels de solda fora da linha de solda, mas que seriam aceitos dentro do intervalo de distâncias considerado como aceito, Figura 3.35.

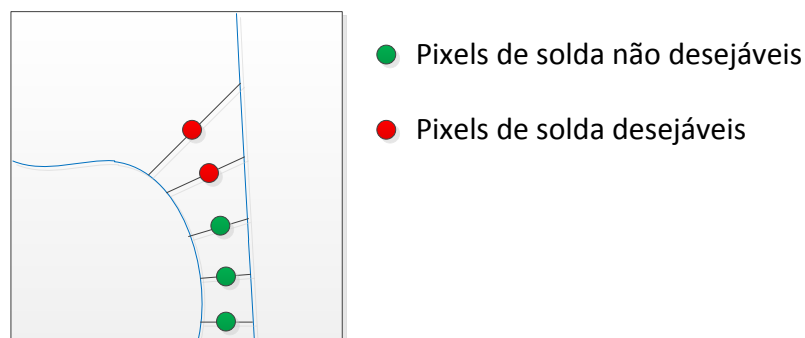


Figura 3.35 - Pixels de solda não desejáveis nos pontos finais da linha de solda.

Uma série de passos para resolver o problema é proposta nesta seção, descritos pelo fluxograma apresentado na Figura 3.36. Uma região de interesse (ROI) na zona do final de linha de solda é gerada para um trabalho com maiores detalhes e menor quantidade de processamento. Dentro dessa imagem, os pixels de solda que realmente pertencem à linha de solda vão ter um comportamento que segue uma linha reta. Por isso com a finalidade de obter a equação da linha reta, a ‘transformada de Hough para detecção de linhas retas’ é implementada, representada pelo bloco ‘Linha de pontos de solda’ no fluxograma.

O final da linha de solda vem acompanhado por um ou dois cantos de peças. Tendo em conta isto, uma técnica de detecção de cantos em imagem de borda denominada ‘*Detector de cantos K-cosine*’ [Sun, 2008]. É detectado só um canto, o mesmo que apresenta o maior valor do cosseno do ângulo. Desse modo o ponto que representa o final de solda é considerado como o ponto de menor distância entre a linha que apresenta os pontos de solda e o canto detectado.

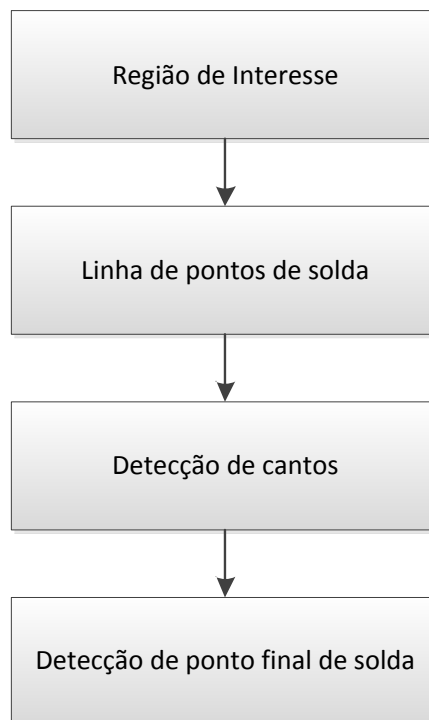


Figura 3.36 - Fluxograma para correção de pontos finais de linha de solda.

3.6.1 Região de Interesse e linha de solda

Em todo conjunto de pixels que pertencem à linha de solda, são escolhidos os dois localizados nos extremos. Uma Região de Interesse (ROI) é criada por cada um desses pixels, usando sua localização como o centro da janela. Uma janela num extremo da linha de solda, com o problema descrito anteriormente é apresentado na Figura 3.37. Uma análise dessa região é apresentada na sequência da Figura 3.37.

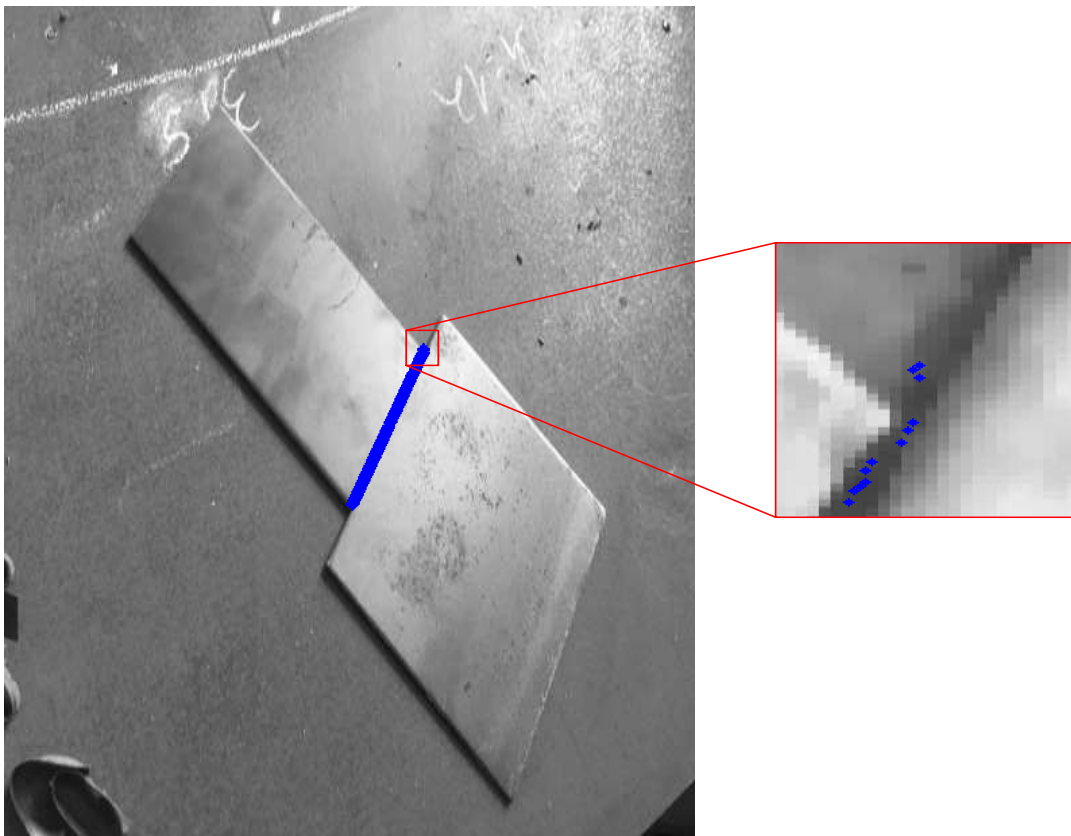


Figura 3.37 - Região de Interesse no extremo da linha de solda com pontos indesejáveis.

3.6.2 Linha de Hough com os pontos de solda:

Na região de interesse, é criada uma janela unicamente com os pontos de interesse. É possível notar que nesta região o comportamento da linha de solda é similar a uma linha reta. A ‘transformada de Hough para detecção de linhas retas’ é uma opção atrativa para a detecção dos parâmetros da linha reta, já que tem a particularidade de detectar a linha que passa pela

maior quantidade de pontos, com o qual se teria um certo nível de discriminação com os pontos de linha de solda que não seguem um padrão de linha reta nessa região.

As imagens da Figura 3.38 ilustram os procedimentos de detecção de uma linha reta que apresente a tendência de pixels da linha de solda na Região de Interesse (ROI).

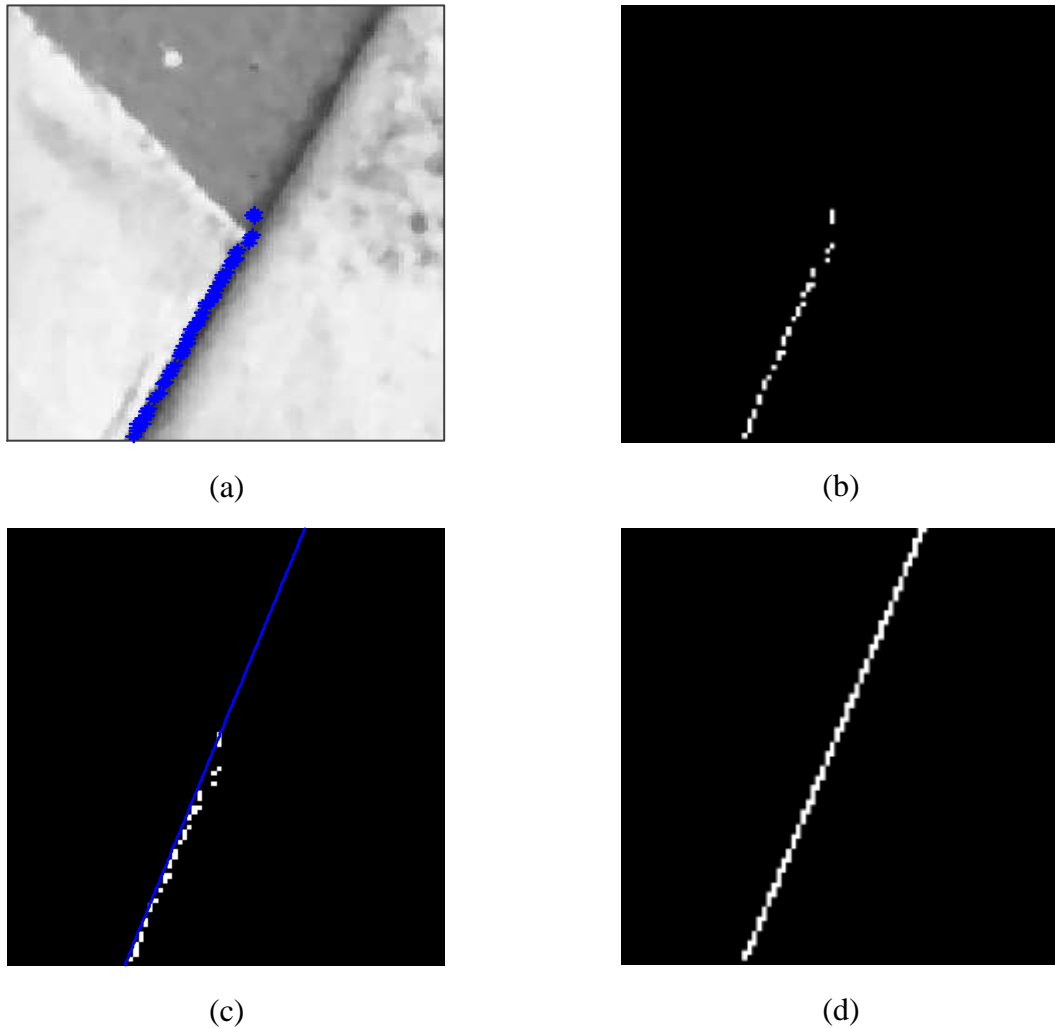


Figura 3.38 - Etapas de detecção da linha na região de interesse.

Na Figura 3.38(a) a imagem com centro no último ponto de solda é apresentada, em seguida os pontos de solda são transformados em pontos brancos e os demais pixels em pontos pretos, como apresentado na Figura 3.38(b). A linha de Hough é detectada pelo método da transformada de Hough aplicada à linhas retas e a mesma é apresentada superposta com os pontos de linha de Hough na Figura 3.38(c). Finalmente é criada uma imagem binária

representando a linha de Hough dos pontos de solda na Figura 3.38(d).

3.6.3 Detecção de cantos com técnica *K-cosine*

No momento de avaliar as características da imagem na região do final da linha de solda em junta de topo, tem-se dois casos particulares. Um caso consiste em que os cantos estejam muito próximos, denominado caso de cantos coincidentes como apresentado na Figura 3.39(a); neste caso se uma solda é desenvolvida, o ponto inicial pode ser considerado como o ponto central entre os cantos. Um segundo caso, com maior dificuldade são os denominados cantos não coincidentes apresentado na Figura 3.39(b), onde os cantos tem uma distância maior comparados com o caso anterior; neste caso ao ser desenvolvida uma solda, o ponto inicial não pode ser considerado como o ponto médio entre ambos cantos. Como o último caso é o caso mais geral, o método desenvolvido será aplicado nele. Uma adequada detecção da coordenada do canto permite obter uma correta coordenada do início ou do final da solda.

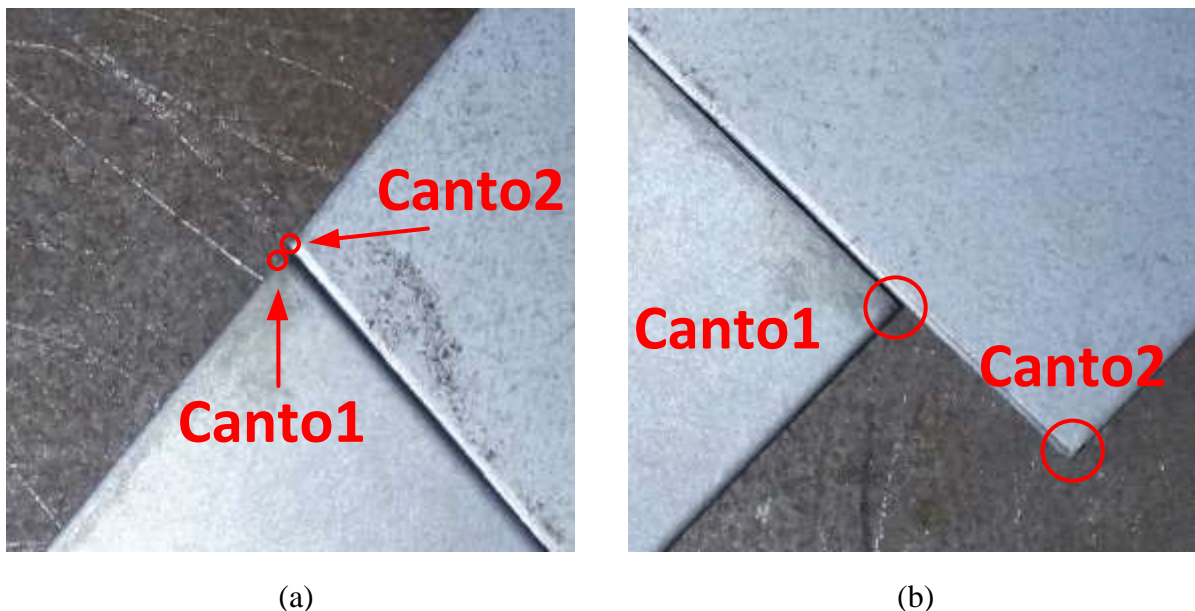


Figura 3.39 - Cantos coincidentes e cantos não coincidentes.

A detecção de cantos pode ser atingida usando só a imagem binária de bordas da zona em interesse. Se a técnica de detecção de bordas de Canny é usada na Região de Interesse (ROI), geralmente as imagens binárias podem apresentar muitas bordas que não pertencem às peças, como apresentado na Figura 3.40(a), onde as bordas que não pertencem às peças estão dentro das linhas azuis e as bordas que pertencem estão dentro das linhas vermelhas. Para solução do problema as duas linhas com maior comprimento são filtradas, eliminando todos os demais pixels de borda detectados, como apresentado na Figura 3.40(b).

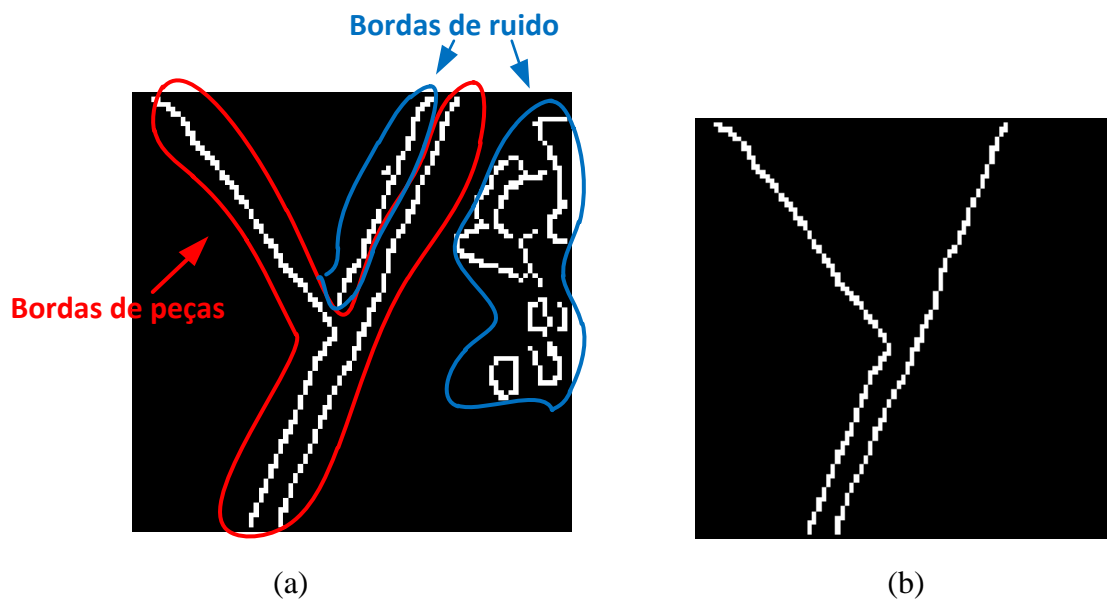


Figura 3.40 - Linha de bordas no final de linha de solda.

Uma vez que um pré-processamento da imagem de borda for atingido, inicia-se o processo da detecção de borda pelo método ‘*K-cosine corner detection*’, apresentado por Sun, 2008. O método se baseia nos ângulos entre um pixel (ponto P_i na Figura 3.41) central e seus vizinhos afastados K_{cos} pixels (pontos P_{i+k} e P_{i-k} na Figura 3.41), sendo K_{cos} um valor inteiro real positivo, como apresentado na Figura 3.41.[Sun, 2008] O valor de K_{cos} deve ser relativamente alto para não cair em cantos falsos, que podem ser pequenas variações em bordas, aqui o valor usado é de $K_{cos} = 15$.

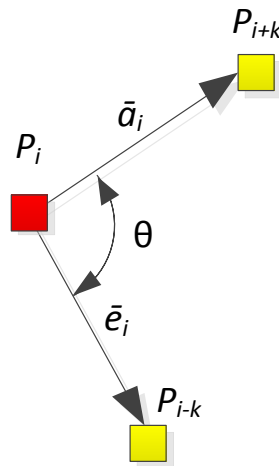


Figura 3.41 - Exemplo de método *K-cosine corner detection*.

Onde a_i e e_i são os vetores entre as coordenadas da imagem de P_{i+k} e P_i , e P_i e P_{i-k} respectivamente e θ apresenta o ângulo entre os dois vetores a_i e e_i . A relação matemática para a obtenção do cosseno do ângulo entre os pontos é apresentada na equação (3.17). O número de pixels entre os pontos usados para calcular os valores dos vetores a_i e e_i é representado pelo valor de K_{cos} .

$$C_i = \cos(\theta_i) = \frac{a_i \bullet e_i}{(\|a_i\| \bullet \|e_i\|)} \quad (3.17)$$

$$\theta_i = \cos^{-1}(C_i)$$

A equação (3.17) é aplicada ao longo das duas linhas da Figura 3.40(b), e a coordenada com menor ângulo é escolhida finalmente como o ponto do canto. É possível notar que só um canto é procurado, já que o algoritmo procura detectar linhas de solda de peças com cantos não coincidentes.

3.6.4 Detecção do ponto da linha de solda menos distante do canto

Uma vez obtidos os dados da coordenada do ponto do canto e as coordenadas da linha reta que apresenta os pontos de solda (Figura 3.38(d)), a coordenada do ponto de fim da linha de solda é detectada como o ponto sobre a linha de solda que apresente a menor distância com a coordenada do canto, como apresentado na equação (3.18).

$$\arg \min_{x_{hl}(i), y_{hl}(i)} \left(\sqrt{(x_c - x_{hl}(i))^2 + (y_c - y_{hl}(i))^2} \right) \text{ com } i = 1, 2, \dots, K_{cos} \quad (3.18)$$

Onde x_c, y_c representam as coordenadas do canto e $x_{hl}(i), y_{hl}(i)$ apresentam as coordenadas de cada pixel de um vector de K_{cos} componentes que representa a linha de Hough da Figura 3.38(d).

Finalmente todos os pontos de linha de solda localizados dentro da região de interesse são eliminados, tendo como resultado só o ponto que representa o início ou final da solda, como no exemplo apresentado na Figura 3.42.

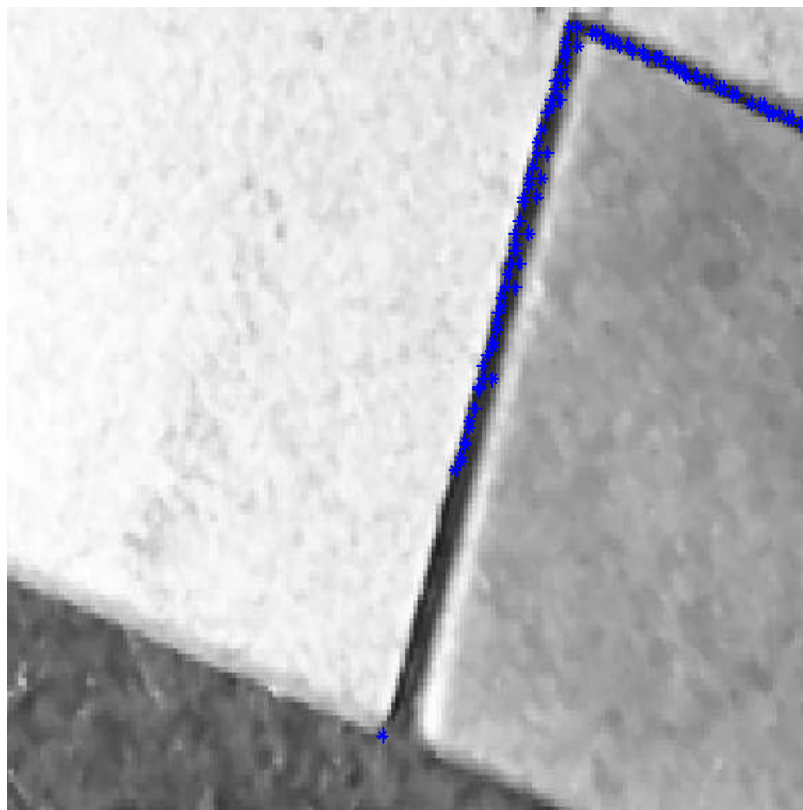


Figura 3.42 - Exemplo de correção de final de linha.

4. AVALIAÇÃO EXPERIMENTAL

4.1 Descrição

Para a avaliação do sistema de visão desenvolvido, foram executados testes experimentais na área de trabalho do Laboratório de Soldagem & Técnicas Conexas (LS&TC) da Universidade Federal do Rio Grande do Sul, com o objetivo de comprovar a funcionalidade do método e entender a magnitude do erro produzido. O processo é implementado utilizando uma câmera montada em um suporte com fontes de iluminação, um robô de solda MIG-MAG, onde os movimentos do robô são controlados com um controlador DX-100 e os parâmetros do processo de soldagem Mig/Mag são controlados com uma fonte Fronius.

Com o objetivo de medir a precisão do método, os pontos de solda obtidos pelo método 2, de detecção de linhas de solda retas e curvas são comparados com pontos considerados ideais seguindo o fluxograma apresentado na Figura 4.1. Para obtenção dos pontos considerados ideais, linhas de pontos dentro da imagem foram escolhidas cuidadosamente de forma manual. A imagem é ampliada e os pontos de solda considerados ideais são escolhidos como os pontos localizados nos pixels mais escuros entre as duas bordas da linha de solda. Assim os pixels que correspondem à estas características são escolhidos manualmente formando o conjunto de pixels ideais.

As coordenadas nos dois conjuntos de pixels são transformadas em coordenadas globais, por meio de um sistema de calibração, de maneira a permitir determinar melhor a magnitude do erro (diferenças em coordenadas globais).

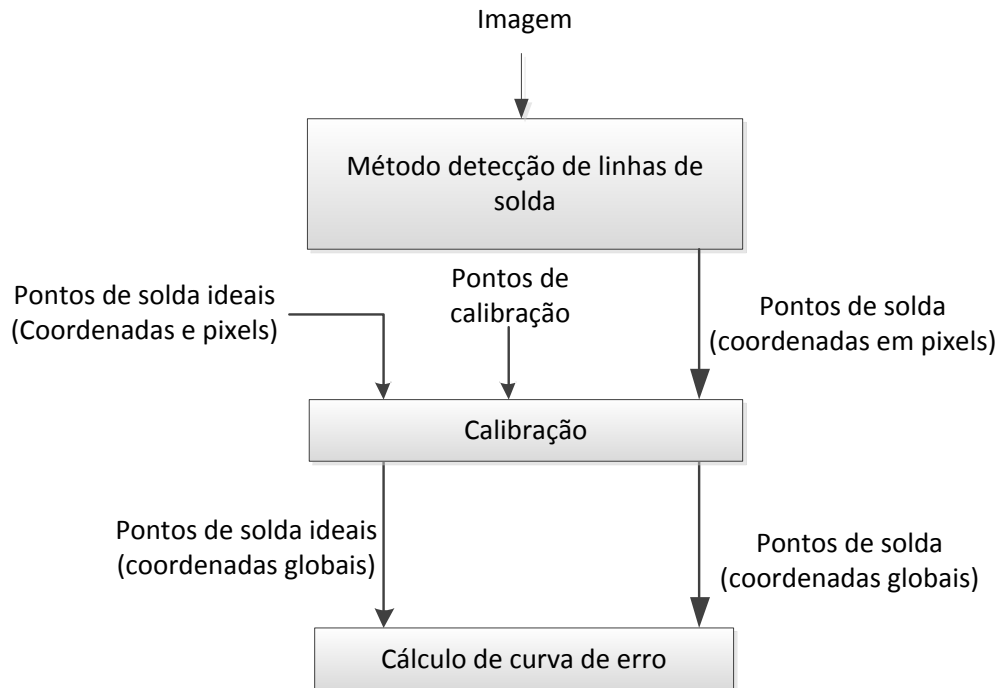


Figura 4.1 - Cálculo de erro para método de detecção de linha de solda.

O cálculo da curva de erro $Erro(i)$, com magnitude em milímetros, é conseguido com o valor da mínima magnitude de distância euclidiana entre cada ponto considerado ideal (x_{lsi}, y_{lsi}) e os pontos de solda obtidos pelo método (x_{ls}, y_{ls}) , seguindo a equação (4.1).

$$Erro(i) = \min \left[\sqrt{(x_{ls}(j) - x_{lsi}(i))^2 + (y_{ls}(j) - y_{lsi}(i))^2} \right] \quad (4.1)$$

onde $i = 1, \dots, N_{lsi}$; $j = 1, \dots, N_{ls}$

Os valores N_{lsi} e N_{ls} são a quantidade de pontos de solda ideais e quantidade de pontos de solda obtidos pelo método, respectivamente. Finalmente as curvas de erro são desenvolvidas com o software Matlab®. Diferentes configurações nas peças de trabalho foram adotadas para a avaliação do sistema, considerando três casos que são apresentados na Figura 4.2:

Caso 1.– junta de topo, linha de solda reta e cantos coincidentes, apresentado na Figura 4.2(a);

Caso 2.– junta de topo, linha de solda reta e cantos não coincidentes, como apresentado na Figura 4.2(b);

Caso 3. – junta de topo, linha de solda não reta e cantos coincidentes, apresentado na Figura 4.2(c);

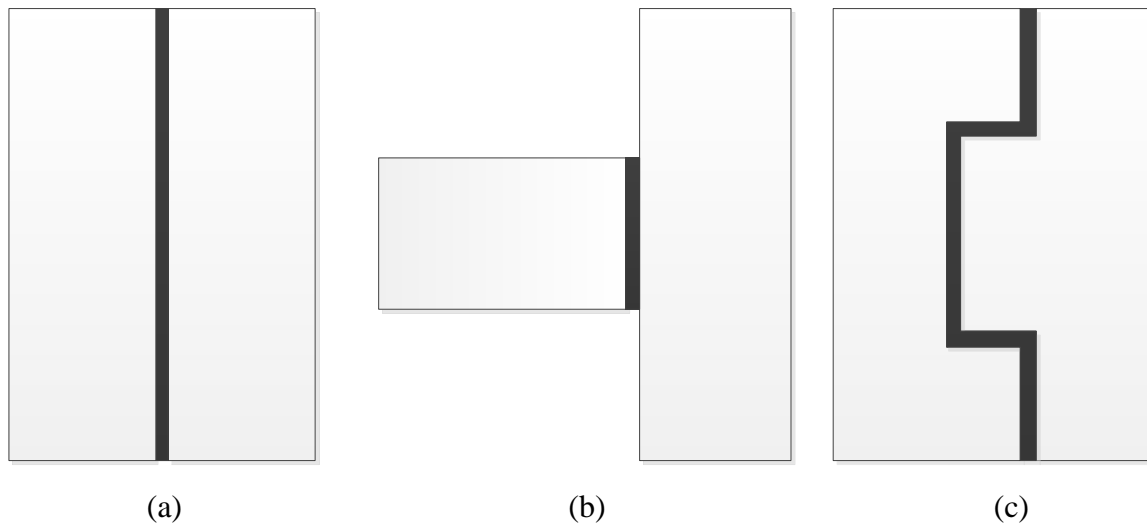


Figura 4.2 - Configurações de peças para avaliação do método.

A câmera é montada sobre um suporte que permite que a mesma esteja a 90 graus em relação à bancada de soldagem. Uma lâmpada posicionada próxima à câmera proporciona uma iluminação direta da peça. O sistema do suporte com as dimensões é apresentado na Figura 4.3

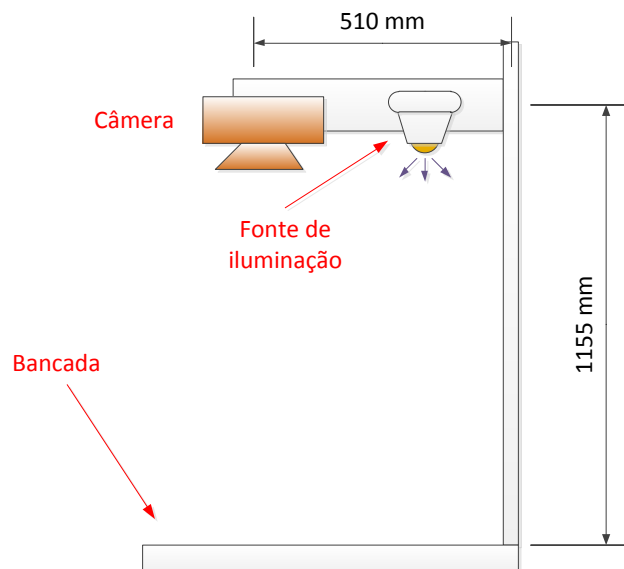


Figura 4.3 - Sistema de suporte para a câmera com suas medidas.

O ambiente de ensaio é apresentado na Figura 4.4. A operação de solda sobre duas peças é guiada com pontos obtidos pelo algoritmo. O caso escolhido para esses resultados é o caso 3, já que é o que maior complexidade apresenta.

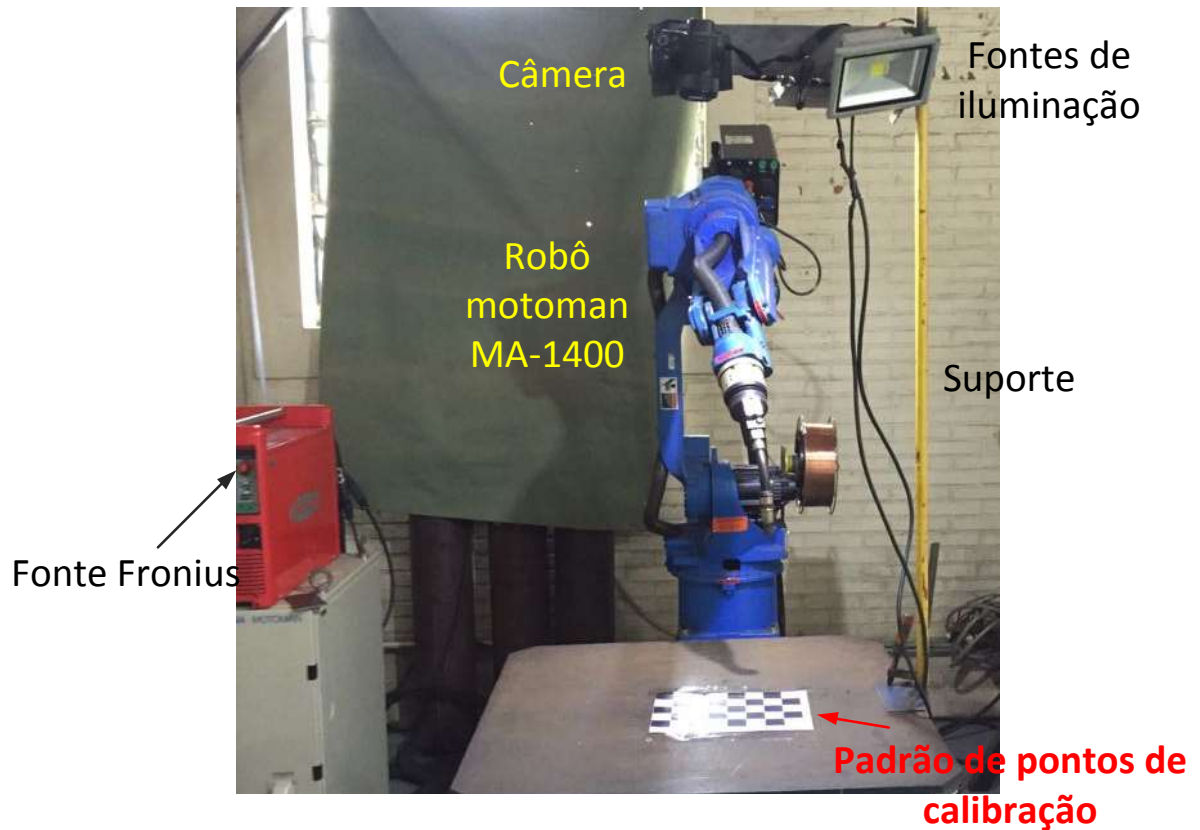


Figura 4.4 - Ambiente de trabalho para detecção de linha de solda.

4.2 Calibração

A calibração é o processo que antecede o processo de detecção da linha de solda. Como mencionado, a mesma tem a finalidade de transformar as coordenadas dos pixels dentro da imagem em coordenadas de posição no espaço de trabalho.

A ferramenta utilizada para conseguir este objetivo é um padrão de pontos coplanares, com distâncias conhecidas (ver Figura 4.4 acima). Desta maneira, com o conhecimento das coordenadas em pixels dos pontos e das coordenadas globais de cada um deles, podem-se obter os valores dos parâmetros característicos da câmera.

O padrão de pontos coplanares é uma matriz de blocos pretos e brancos, de 6 linhas x 9 colunas. Todos os blocos são quadrados e apresentam uma medida de 33,42 mm de lado. Os

parâmetros são considerados com valores de distância medidas nos cantos dos blocos, desta forma obtém-se as coordenadas globais de cada um dos pontos (x_w, y_w, z_w) . O ponto de coordenadas globais $(0, 0, 0)$ situa-se no canto superior esquerdo da figura. Na Figura 4.5 é apresentada uma parte do sistema de blocos:

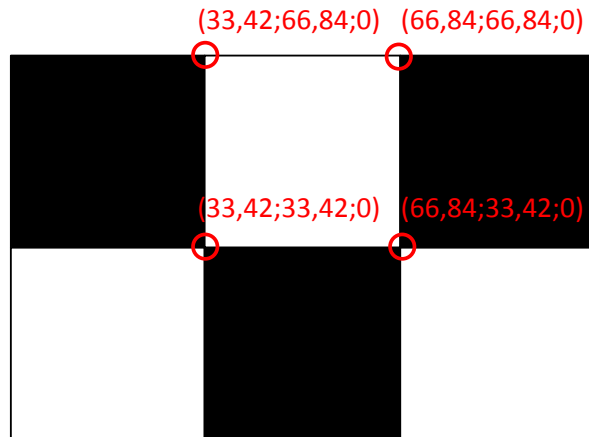


Figura 4.5 - Padrão de pontos de calibração da câmera.

Na Figura 4.5 observa-se que todos os pontos compartilham a mesma coordenada z_w , com valor de zero. No momento em que se obtém a imagem da área de trabalho, as coordenadas em pixels no tabuleiro são marcadas manualmente; o ponto $(0,0,0)$ é posicionado no canto superior esquerdo do mesmo e cada um dos outros pontos usados para fazer a calibração é marcado com azul.

Na área de trabalho os eixos x_w e y_w são alinhados paralelamente com os eixos x_r e y_r da coordenada do robô, com o objetivo de facilitar a transformação entre dois sistemas de referência. A Figura 4.6 mostra a imagem do tabuleiro sobre a bancada de soldagem, com os pontos indicados.

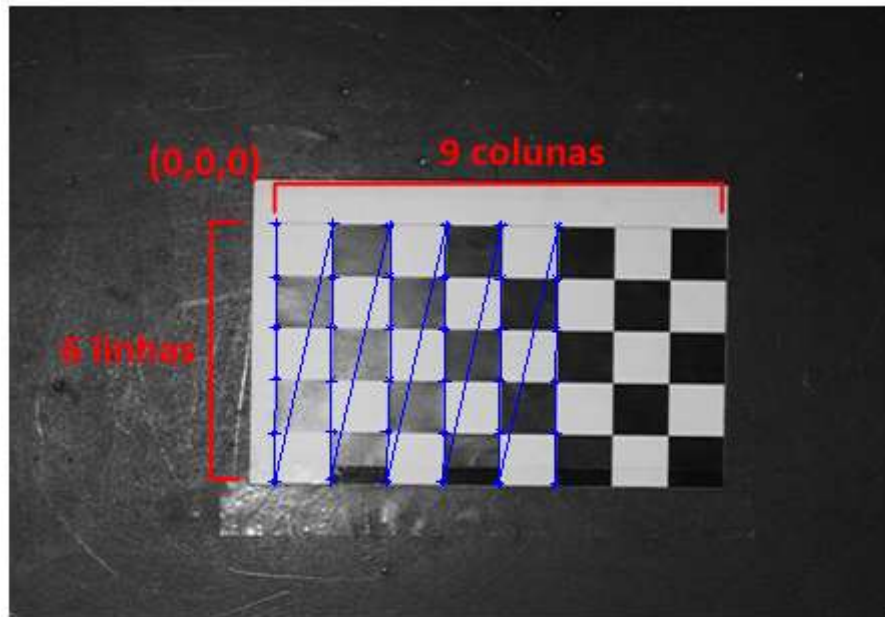


Figura 4.6 - Padrão de pontos utilizados para processo de calibração.

4.3 Resultados em simulação

Como mencionado na descrição da avaliação experimental, nos resultados vão ser avaliados 3 tipos de configurações de peças diferentes. Para cada configuração são consideradas 3 orientações diferentes a fim de avaliar a robustez frente às orientações. Uma curva de erro é gerada com a diferença obtida entre os pontos de solda detectados pelo algoritmo e uma linha de pontos desenhada manualmente, considerada como ideal. As curvas de erro, apresentadas nas ilustrações da Figura 4.8, Figura 4.10 e da Figura 4.12, são criadas usando a equação (4.1) no *software* Matlab®.

Finalmente a média e o desvio padrão são obtidos considerando as 3 avaliações de cada configuração. Com isso entende-se como o método funciona para cada uma das diferentes configurações.

4.3.1 Avaliação do Caso 1

As 3 imagens de diferentes orientações da configuração em ‘junta de topo, linha de solda reta e cantos coincidentes’, com os pontos de solda superpostos, são apresentados na Figura 4.7.

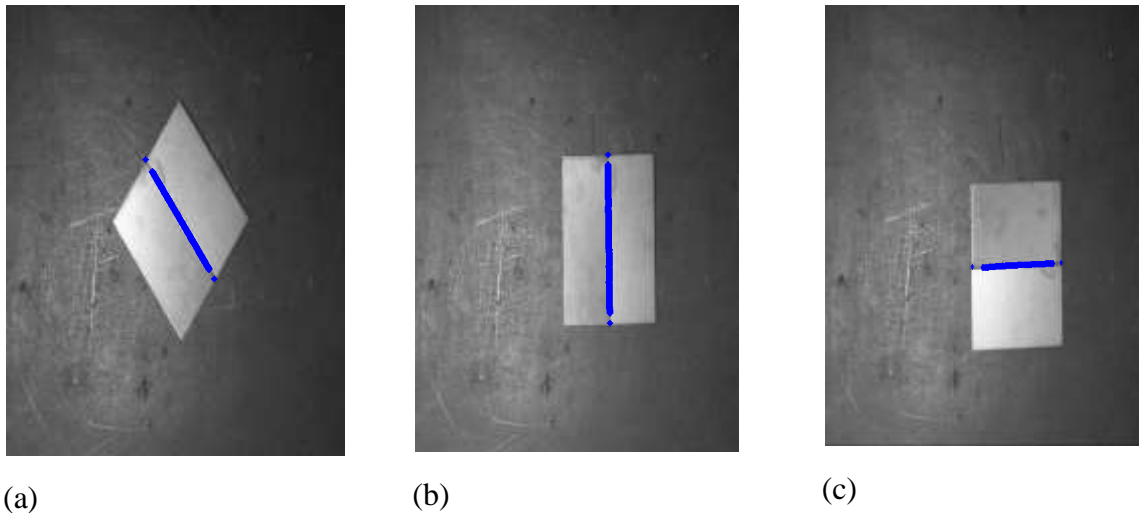


Figura 4.7 - Linha de pontos de solda obtida para peças em tipo de junta de topo, linha de solda reta e cantos coincidentes.

Pelas imagens da Figura 4.7 que as peças estão posicionadas sobre uma bancada com riscos e uma iluminação de uma maior intensidade no canto inferior esquerdo, o qual gera também sombras indesejáveis. Igualmente as peças não possuem uma textura totalmente homogênea. Todos esses fatos dificultam uma perfeita detecção de bordas na imagem utilizando o algoritmo.

O algoritmo apresentado aqui identificou a totalidade da linha de solda em 3 ângulos diferentes. As discontinuidades nos finais das mesmas são devidas ao fato da correção do ponto inicial e final de solda, como apresentado na seção de ‘Correção de pontos finais’.

As curvas de erro entre os pontos de linha de solda detectados pelo algoritmo e os pontos da linha de solda ideal, de cada ensaio, são apresentadas na Figura 4.8.

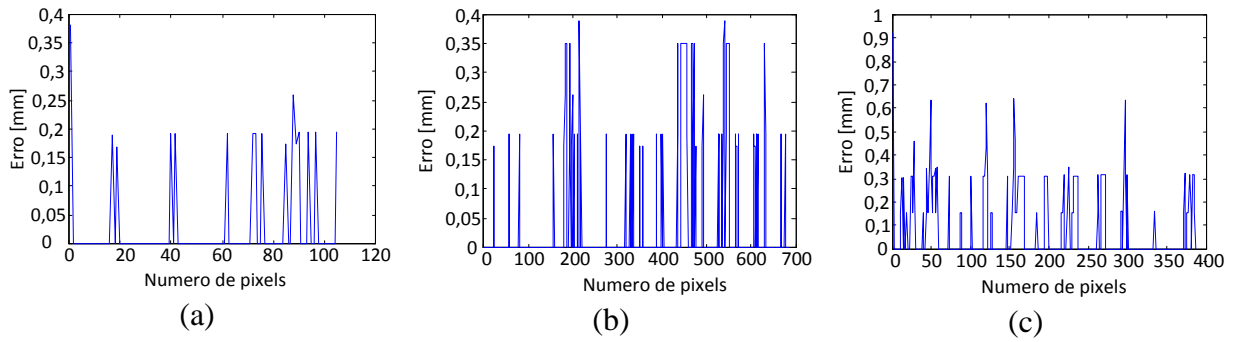


Figura 4.8 - Erro de linha de solda detectada com relação a uma linha ideal, para linha (a) diagonal, (b) vertical, (c) horizontal.

Conforme os dados obtidos, ilustrados na Figura 4.8, média do erro foi 0,05 mm e do desvio padrão 0,113 mm, com um máximo valor de 0,91 mm.

4.3.2 Avaliação do Caso 2

As três imagens de diferentes orientações da configuração em ‘tipo de junta de topo, linha de solda reta e cantos coincidentes’, com os pontos de solda superpostos, são apresentados na Figura 4.9.

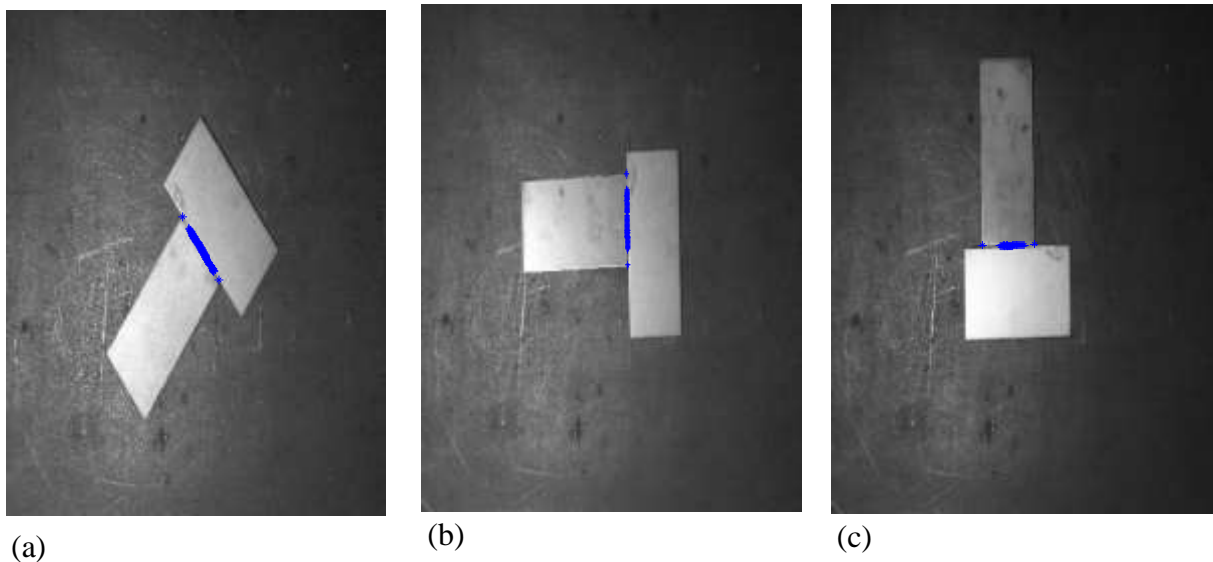


Figura 4.9 - Linha de pontos de solda obtida para peças em tipo de junta de topo, linha de solda reta e cantos não coincidentes.

Nota-se na Figura 4.9 que as peças estão posicionadas sobre uma bancada com riscos e uma iluminação de uma maior intensidade no canto inferior esquerdo, o qual gera também sombras indesejáveis. Igualmente as peças não possuem uma textura totalmente homogênea. Todos esses fatos dificultam uma perfeita detecção de bordas na imagem, utilizando o algoritmo.

O algoritmo aplicado identificou a totalidade da linha de solda em 3 ângulos diferentes. As discontinuidades nos finais das mesmas são devido ao fato da correção do ponto inicial e final de solda, como apresentado anteriormente.

As curvas de erro entre os pontos de linha de solda detectados pelo algoritmo e os pontos da linha de solda ideal, de cada ensaio, são apresentadas na Figura 4.10.

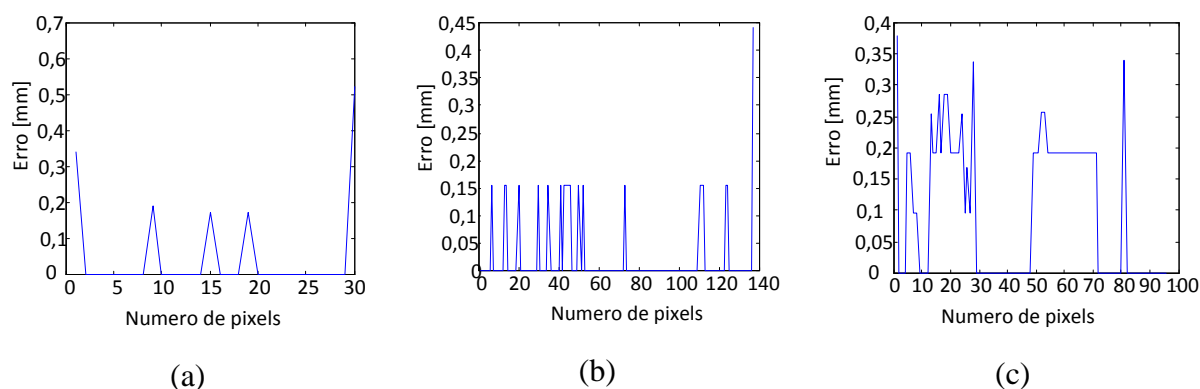


Figura 4.10 - Erro de linha de solda detectada com relação a uma linha ideal, para linha (a) diagonal, (b) vertical, (c) horizontal.

Pelos dados obtidos, ilustrados na Figura 4.10, média do erro foi 0,0533 mm. e do desvio padrão 0,096 mm com um máximo valor de 0,52 mm.

4.3.3 Avaliação do Caso 3

As três imagens de diferentes orientações da configuração em ‘tipo de junta de topo, linha de solda não reta e cantos coincidentes’, com os pontos de solda superpostos, são apresentados na Figura 4.11.

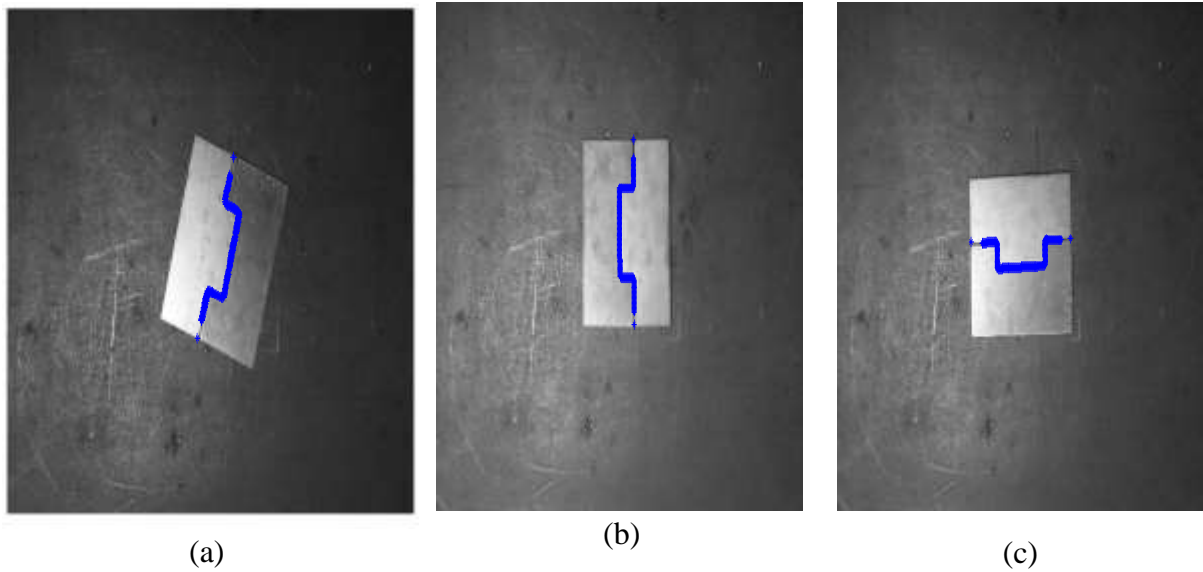


Figura 4.11 - Linha de pontos de solda obtida para peças em tipo de junta de topo, linha de solda não reta e cantos coincidentes.

Nota-se na Figura 4.11 que as peças estão posicionadas sobre uma mesa com arranhões e uma iluminação de uma maior intensidade no canto inferior esquerdo, o qual gera também sombras indesejáveis. Igualmente as peças não possuem uma textura totalmente homogênea. Todos esses fatos dificultam uma perfeita detecção de bordas na imagem, utilizando o algoritmo.

O algoritmo apresentado aqui detecta a totalidade da linha de solda em 3 ângulos diferentes. As discontinuidades nos finais das mesmas são devido ao fato da correção do ponto inicial e final de solda, como apresentado anteriormente.

As curvas de erro entre os pontos de linha de solda detectados pelo algoritmo e os pontos da linha de solda ideal, de cada ensaio, são apresentadas na Figura 4.12.

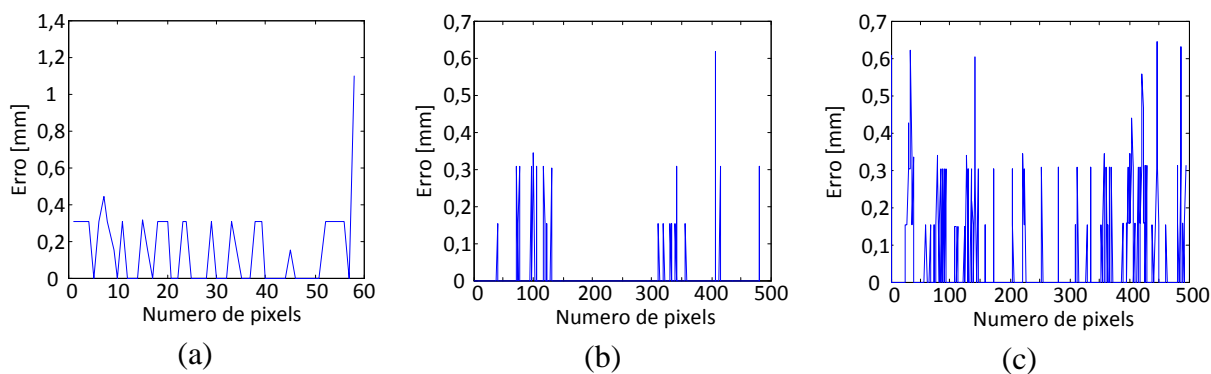


Figura 4.12 - Erro de linha de solda detectada com relação a uma linha ideal – (a) diagonal, (b) vertical, (c) horizontal.

O valor da média para o erro foi 0,0413 mm. e do desvio padrão foi 0,109 mm com um máximo valor de 1,1 mm.

4.3.4 Análise geral de resultados do método em diferentes configurações

Constata-se que o erro em milímetros nas 3 curvas teve um valor relativamente baixo, com uma média menor que 0,2 mm e um desvio padrão menor do que 0,5 mm. Como desvantagem pode se citar que em alguns casos tem-se picos de erro, chegando até a uma magnitude de 3 mm, porém são pontos isolados que com uma linha de interpolação entre todos os pontos reduziria notavelmente os picos.

Com o citado anteriormente conclui-se que o algoritmo segue uma trajetória aceitável para conseguir uma linha de solda cujos desvios poderiam ser admitidos em peças com junta de topo, em diferentes arranjos entre elas.

4.4 Análise dos parâmetros de influência no método:

A etapa do algoritmo que mais problema apresenta é a detecção de bordas, que consiste na base do método para trabalhar como entrada para as subsequentes tarefas.

Muitos procedimentos propostos no algoritmo tiveram como motivo a robustez frente a uma detecção de bordas com defeitos. Como por exemplo, a ‘proposta como solução de bordas inicial para uma linha de solda não completa’ ou a escolha do valor mínimo como pixel de linha de solda em lugar do valor localizado no meio entre as linhas de borda de solda.

Fatores que podem fazer com que o método apresente uma linha de solda inacabada ou com defeitos são: sombras por uma iluminação não perfeita, uma inclinação grande da câmera que atinja um limite onde não seja bem perceptível a linha de solda, reflexos nas bordas das peças por ter uma iluminação pontual.

O fator mencionado anteriormente que maior problema tem ocasionado nos diversos ensaios neste trabalho foram as sombras, motivo pelo qual a iluminação teve que ser direta e não difusa. Uma sombra na continuação da linha de solda pode resultar em que o algoritmo entenda a mesma como parte da linha de solda, já que o algoritmo se baseia na procura inicial de uma linha escura. Um exemplo deste fenômeno é apresentado na Figura 4.13, onde uma parte da sombra foi reconhecida como linha de solda pelo algoritmo proposto.

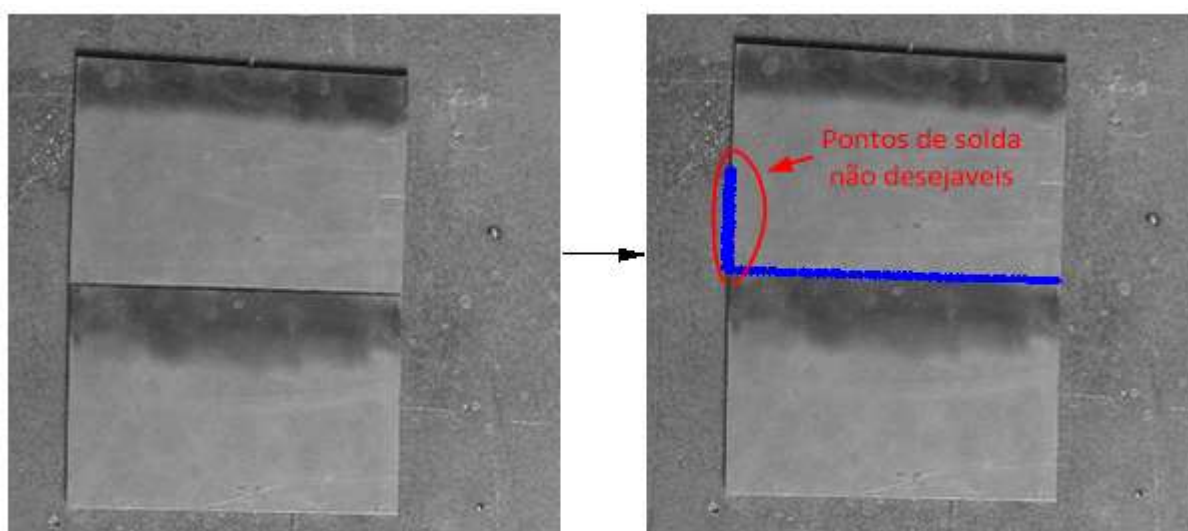


Figura 4.13 - Exemplo de falha do algoritmo por sombra.

4.5 Testes experimentais

Para verificar os resultados de processamento de imagens em uma aplicação real, experimentos foram realizados utilizando um robô Motoman MA-1400 localizado nas instalações do LS&TC. Como efetuador no braço manipulador se utilizou uma tocha de um sistema de soldagem Mig-Mag, controlada por uma fonte Fronius.

Na validação experimental uma série de passos apresentada na Figura 4.14 foi necessária para atingir o objetivo. Os pontos de solda obtidos como resultado do método apresentado, seguido da calibração permite gerar os pontos de solda calibrados e referidos ao sistema de coordenadas globais.

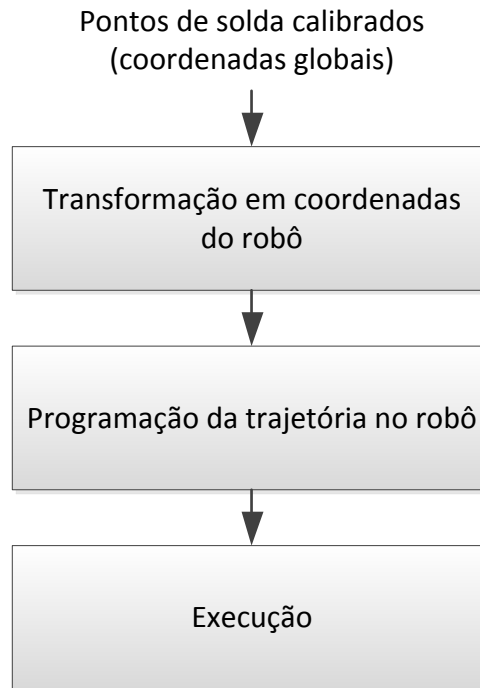


Figura 4.14 - Conjunto de passos para teste experimental.

O passo da ‘Transformação em coordenadas do robô’ é requerido como etapa prévia da programação da trajetória dentro do robô, para expressar cada ponto nos eixos de coordenada do mesmo. Para facilitar esta etapa os eixos de duas dimensões em coordenadas globais (x_w, y_w) foram alinhados com os eixos das coordenadas referidas ao robô x_r, y_r . Assim a tocha do robô foi posicionado na posição (0, 0, 0) das coordenadas globais e os valores x_r, y_r obtidos nessa posição são os valores a serem somados para a transformação de coordenadas. A coordenada z_r é mantida constante e representa a distância bico de contato peça, sendo definida pelo usuário.

Para programar a trajetória no robô, precisam ser inseridos manualmente os pontos de trajetória em coordenadas robô usando o denominado caixa de aprendizagem (‘*teach pendant*’), que basicamente é a interface humano-máquina para programar a central de processamento do robô (neste caso o modelo da central de processamento é o DX-100). A programação do caixa de aprendizagem foi desenvolvida seguindo o manual DX100 da

Yaskawa. Outro método possível é o controle do robô por meio de Ethernet, mas nesse momento a Universidade não possui o *software* encarregado para levar a cabo a tarefa (nos modelos de robô Yaskawa Motoman, o *software* encarregado da comunicação Ethernet é pago). O método dessa etapa foi detectar padrões de linhas retas entre os pontos obtidos pelo método e inserir o ponto inicial e final dentro da trajetória do robô.

Finalmente a solda levada a cabo pelo sistema robotizado de soldagem é executada. O mesmo é apresentado na Figura 4. 15.



Figura 4. 15 - Sistema de soldagem.

Os parâmetros de soldagem assim como a espessura das peças são apresentados na Tabela 4. 1.

Tabela 4. 1 – Parâmetros de soldagem.

Parâmetros	Valor
Corrente média	92 A
Tensão média	19,8 V
Polaridade	CCEP
Distância bico de contato peça	18 mm
Velocidade de soldagem	5 mm/s
Espessura da peça	3 mm
Tipo de gás	C10 (90% Argônio, 10% CO ₂)
Vazão do gás	15 l/min
Tipo de eletrodo	ER70S-6 diâmetro 0,8 mm

Para o teste foram utilizados os pontos do caso 3, apresentados na Figura 4.11(a). A tocha posicionada sobre o ponto do início da solda é apresentada na Figura 4.16(a). Um processo de soldagem guiado pelos pontos obtidos do algoritmo foi realizado, produzindo o cordão de solda ilustrado nas imagens da Figura 4.16(b) e Figura 4.16(c).



(a)



(b)



(c)

Figura 4.16 - Soldagem em peça de caso 3, (a) justapostas (b) soldadas, (c) linha de solda.

5. CONCLUSÕES

Considerando-se a aplicação das técnicas de processamento de imagem abordadas nesse trabalho, com objetivo de detectar uma linha de solda a ser seguida como trajetória por um robô industrial que a venha executar, infere-se que se alcançaram os objetivos básicos propostos.

Analisando os resultados dos testes experimentais efetuados, ilustrados com um estudo de caso, na união de duas peças em junta topo por soldagem de topo, comprova-se com sucesso um novo algoritmo detector de linha de solda, que se baseia em algoritmos já existentes, mas que também faz uso da propriedade da linha de solda manter uma largura relativamente constante. Além disso, foram adicionadas técnicas para conseguir um algoritmo robusto frente a erros na etapa de detecção de bordas das peças a serem soldadas, assim como correções em pontos iniciais e finais.

A aplicação do método apresenta resultados aceitáveis (uma média de erro inferior a 0,2 mm e um desvio padrão inferior a 0,5 mm), em condições de ambiente controlável, que foram com uma iluminação direta e uma câmera perpendicular com o ambiente de trabalho. Os testes avaliados foram sobre peças em junta de topo, com cantos coincidentes, canto não coincidentes, linhas retas e linhas não retas. O erro foi comprovado comparando os pontos de solda obtidos com pontos de solda escolhidos manualmente e considerados como a melhor opção.

Com a configuração em tipo de junta de topo, linha não reta e cantos coincidentes foi realizada uma solda com o robô Motoman MA-1400 do laboratório LS&TC da Universidade Federal do Rio Grande do Sul, comprovando com resultados experimentais que o sistema consegue seguir satisfatoriamente uma trajetória de soldagem.

A parte mais crítica do algoritmo é a detecção de bordas realizada inicialmente, já que a mesma depende de muitos fatores do ambiente, como a iluminação, material das peças a serem soldadas, existem ou não riscos na mesa ou nas peças, etc. Ainda não existe uma etapa que seja 'perfeita' para atingir uma detecção de bordas que considere ou leve em conta as diversas condições de ambiente. Caso se consiga aperfeiçoar esta etapa, o algoritmo dependeria em menor magnitude de um ambiente controlado (em iluminação e ângulo da câmera) para funcionar corretamente.

5.1 Sugestões para trabalhos futuros

Este trabalho abre linhas de pesquisa em diferentes áreas a serem melhoradas. Primeiramente, como já mencionado, uma etapa crítica neste tipo de análise, com visão computacional é a detecção de bordas das peças a serem soldadas. Uma tentativa de melhora nesta etapa pode ser sugerida em várias maneiras, como uma possível etapa de segmentação antes da detecção de bordas.

Outro problema importante para a detecção de bordas são as sombras e os reflexos ocasionados por uma iluminação não completamente direta sobre as peças. Existem diversos métodos para a eliminação de sombras nas imagens, mas neste algoritmo tem que se ter em conta que a linha de solda também é uma sombra e não deve ser eliminada. Um estudo mais aprimorado em um algoritmo de eliminação de sombras e preservação da linha de solda é uma possibilidade interessante.

Alternativamente uma proposta para um futuro trabalho é um estudo bem detalhado do comportamento da detecção de linha de solda em diferentes ângulos da câmera e sob diferentes tipos de iluminação.

O trabalho de detecção de linhas de solda pode se estender para diferentes tipos de juntas, como junta filete ou sobreposta. Não se encontram na atualidade muitos trabalhos que abordem processamento de imagens detectem a linha de solda nesse tipo de juntas, constituindo-se uma área interessante como sugestão para novas pesquisas.

6. PUBLICAÇÕES

Uma publicação do método desenvolvido neste trabalho, especificamente do método 1 descrito mais na frente, foi publicado no Congresso Nacional de Soldagem, 2016, Belo Horizonte. O nome da publicação é ‘Processamento de Imagens Aplicado à Automação e Robotização de Operações de Soldagem ao Arco Elétrico’, sendo os autores Bauernfeind, Omar Alejandro; Mazzaferro, José Antônio E.; Lorini, Flávio José; Machado, Ivan Guerra.

REFERÊNCIAS BIBLIOGRÁFICAS

Akinlar, C.; Chome, E. **PEL: A Predictive Edge Linking algorithm**. Journal of Visual Communication and Image Representation, p. 159–171, 2016.

American Welding Society. **Welding Handbook**. 8^a Ed., AWS, Miami, v. 1, p. 66-87, 1987.

Bauernfeind, O. A.; Mazzaferro, J. A. E.; Lorini, F. J.; Machado, I. G. **Processamento de Imagens Aplicado à Automação e Robotização de Operações de Soldagem ao Arco Elétrico**. Congresso Nacional de Soldagem, 2016, Belo Horizonte. Anais do XLII CONSOLDA, 2016.

Canny, J. **A Computational Approach to Edge Detection**. IEEE Transactions on Pattern Analysis and Machine Intelligence. PAMI-8, n. 6, 1986.

Chen, X.; Chen S. Tao.; Yucheng, L. L. **Practical method to locate the initial weld position using visual technology**. The International Journal of Advanced Manufacturing Technology. v. 30, p. 663–668, 2006.

Chen, X. Z.; Chen S. B.; Lin T. **Recognition of Macroscopic Seam for Complex Robotic Welding Environment**. Robotic Welding, Intelligence and Automation, p. 171-178, Springer Berlin Heidelberg, 2007.

Chen, X. Z.; Chen, S.B. **The autonomous detection and guiding of start welding position for arc welding robot**. Industrial Robot: An International Journal. v. 37, p. 70–78, 2010.

Dinham, M.; Fang, G.; Zou, J.J. **Experiments on Automatic Seam Detection for a MIG Welding Robot**. AICI 2011: International Conference on Artificial Intelligence and Computational Intelligence, Part II, LNAI 7009, p. 390-397. Springer-Verlag, Berlin Heidelberg, 2011.

Dinham, M.; Fang, G. **Autonomous weld seam identification and localization using eye-in-hand stereo vision for robotic arc welding**. Robotics and Computer-Integrated Manufacturing, p. 288–301, 2013.

Duda, R. O.; Hart, P. E. **Use of the Hough transformation to detect lines and curves in pictures**. Graphics and Image Processing, v. 15, p. 11-15, 1972.

Fu, K. S.; Gonzales, R. C.; Lee, C. S. G. **Robotics: Control, Sensing, Vision and Intelligence**. McGraw-Hill, 1987.

Gonzales, R. C.; Woods, R. E. **Digital Image Processing**. 3. ed. Pearson Education International, 2007.

Grassi, M. V. **Desenvolvimento e aplicação de um sistema de visão para robô industrial de manipulação**. Dissertação de Mestrado, Universidade Federal do Rio Grande do Sul, Programa de Pós-Graduação em Engenharia Mecânica, 2005.

Harris C.; Stephens M. **A combined corner and edge detector**. Proceedings of the Fourth Alvey Vision Conference. Manchester: the University of Sheffield Printing Unit, p.147-151, 1988.

Hough, P. V. C. **Method and means for recognizing complex patterns**. U. S. Patent 3, 069 654, 1962.

Hossain, F.; Asaduzzaman, M.; Yousuf, M. A. **Dynamic Thresholding based Adaptive Canny Edge Detection**. International Journal of Computer Applications, v. 135, p. 37-41, 2016.

ISF Aachen. **A Guide to Welding Technology by the ISF Aachen, Germany. ISF Aachen Welding Technology Part I**. 2002.

Kong, M.; Shi, F.H.; Chen, S.B.; Lin, T. **Recognition of the Initial Position of Weld Based on the Corner Detection for Welding Robot in Global Environment**. *Robotic Welding, Intelligence and Automation, LNCIS 362*, p. 249–255. Springer-Verlag Berlin Heidelberg, 2007.

Ma, Y.; Soatto, S.; Kosecka, J.; Sastry, S. S. **An Invitation to 3-D Vision, From Images to Geometric Models**. Springer Science+Business Media, LLC. New York, 2004.

Machado, I. G. **Soldagem & Técnicas Conexas: Processos**. Porto Alegre, 2007.

MathWorks. **Matlab**. Disponível em www.mathworks.com Acesso em 08 Julho, 2017.

Otsu, N. **A threshold selection based on gray-level histograms**. *IEEE Transactions on Systems, Man, And Cybernetics*. 1979.

Pires, J. N.; Loureiro, A.; Godinho, T.; Ferreira, P.; Fernando, B.; Morgado, J. **Welding Robots**. *IEEE Robotics & Automation Magazine*, 2003

Prewitt, J. M. S. **Object enhancement and extraction**. *Picture Processing and Psychopictorics*, Lipkin, B. S., and Rosenfeld, A. Academic Press, New York, 1970.

Rodas, A.; Alves, D. M.; Araújo, G. R.; Ataíde de Oliveira Junior, J.; Palmeira, T. P. H.; Gonçalves, L. B. **Soldagem robótica aplicada a indústria automobilística: A evolução e processos adaptativos para o desenvolvimento do setor**. *Ciências exatas e tecnológicas*, Maceió, 2014.

Roberts, L. G. **Machine perception of three-dimensional solids**. *Optical and Electro-Optical Information Processing*, Tippet, J. T. (ed.), MIT Press, Cambridge, Mass., 1965.

Rivlin, E. I. **Mechanical Design of Robots**. McGraw-Hill, USA, 1988.

Shah, M. **Fundamentals of Computer Vision**. Computer Science Department, University of Central Florida. Orlando, FL 32816, 1997.

Shi, F.; Lin, T.; Chen, S. **Efficient weld seam detection for robotic welding based on local image processing**. *Industrial Robot: An International Journal*, v. 36, p.277 – 283, 2009.

Siciliano, B.; Sciavicco, L.; Villani, L.; Oriolo, G. **Robotics: Modelling, Planning and Control**. 1. ed. London: Springer, 2011.

Sobel, I. E. **Camera models and machine perception**. Ph. D. dissertation Stanford University, Palo Alto, Calif., 1970.

Spong, M. W.; Hutchinson, S.; Vidyasagar, M. **Robot Dynamics and Control**. 2. ed. Wiley, 2005.

Sun, T. **K-Cosine Corner Detection**. *Journal of Computers*, v. 3, No. 7. p. 16-22, 2008.

Szeliski, R. **Computer Vision: Algorithms and Applications**. Springer, 2010.

Tsai, Y. R. **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*, 1987.

Wei, S.; Wang, J.; Lin, T.; Chen, S. **Application of image morphology in detecting and extracting the initial welding position**. *Journal of Shanghai Jiaotong University (Science)*. v. 17, p. 323–326, 2012.

Xu, D.; Wang, L.; Tan, M. **Image Processing and Visual Control Method for Arc Welding Robot**. *Proceedings of the 2004 IEEE. International Conference on Robotics and Biomimetics*, p. 22-26, Shenyang, China, 2004.

Yaskawa. **DX100 Operator's Manual for Spot and Arc Welding Using Motor Gun.** Manual No. RE-CSO-A044. Part Number: 163543-1CD, Revision: 2.

Zhu, Z. Y.; Lin, T.; Piao, Y. J.; Chen, S. B. **Recognition of the initial position of weld based on the image pattern match technology for welding robot.** International Journal of Advanced Manufacturing Technology, v. 26, p. 784–788, 2005.

Zou, Y.; Li, Y.; Jiang, L.; Xue, L. **Weld pool image processing algorithm for seam tracking of welding robot.** 6th IEEE Conference on Industrial Electronics and Applications, 2011.