



**IDENTIFICAÇÃO DE FALHAS EM ESPAÇADORES DE LINHAS
DE TRANSMISSÃO UTILIZANDO VISÃO ESTÉREO
E REDES NEURAS ARTIFICIAIS**

ELDER OROSKI

**DISSERTAÇÃO DE MESTRADO EM ENGENHARIA ELÉTRICA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA**

**FACULDADE DE TECNOLOGIA
UNIVERSIDADE DE BRASÍLIA**

**UNIVERSIDADE DE BRASÍLIA
FACULDADE DE TECNOLOGIA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA**

**IDENTIFICAÇÃO DE FALHAS EM ESPAÇADORES DE LINHAS
DE TRANSMISSÃO UTILIZANDO VISÃO ESTÉREO
E REDES NEURAIIS ARTIFICIAIS**

ELDER OROSKI

**DISSERTAÇÃO DE MESTRADO ACADÊMICO SUBMETIDA AO DEPARTAMENTO
DE ENGENHARIA ELÉTRICA DA FACULDADE DE TECNOLOGIA DA UNIVERSI-
DADE DE BRASÍLIA COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A
OBTENÇÃO DO GRAU DE MESTRE EM ENGENHARIA ELÉTRICA.**

APROVADA POR:

**Prof. Adolfo Bauchspiess, ENE/UnB
(Orientador)**

**Prof. Cristiano Jacques Miosso Rodrigues, FGA/UnB
Examinador Externo**

**Prof. Geovany A. Borges, ENE/UnB
Examinador Interno**

BRASÍLIA, 21 DE OUTUBRO DE 2011.

FICHA CATALOGRÁFICA

OROSKI, ELDER

Identificação de Falhas em Espaçadores de Linhas de Transmissão Utilizando Visão Estéreo e Redes Neurais Artificiais [Distrito Federal] 2011.

xi, 124p., 210 x 297 mm (ENE/FT/UnB, Mestre, Engenharia Elétrica, 2011).

Dissertação de Mestrado – Universidade de Brasília, Faculdade de Tecnologia.

Departamento de Engenharia Elétrica

1. Inspeção de Linhas de Transmissão

3. Reconstituição de Imagem 3D

I. ENE/FT/UnB

2. Visão Estéreo

4. Redes Neurais

II. Título (série)

REFERÊNCIA BIBLIOGRÁFICA

OROSKI, E. (2011). Identificação de Falhas em Espaçadores de Linhas de Transmissão Utilizando Visão Estéreo e Redes Neurais Artificiais, Dissertação de Mestrado em Engenharia Elétrica, Publicação PPGENE.DM-456/2011, Departamento de Engenharia Elétrica, Universidade de Brasília, Brasília, DF, 124p.

CESSÃO DE DIREITOS

AUTOR: Elder Oroski

TÍTULO: Identificação de Falhas em Espaçadores de Linhas de Transmissão Utilizando Visão Estéreo e Redes Neurais Artificiais.

GRAU: Mestre ANO: 2011

É concedida à Universidade de Brasília permissão para reproduzir cópias desta dissertação de mestrado e para emprestar ou vender tais cópias somente para propósitos acadêmicos e científicos. O autor reserva outros direitos de publicação e nenhuma parte dessa dissertação de mestrado pode ser reproduzida sem autorização por escrito do autor.

Elder Oroski

Departamento de Eng. Elétrica (ENE) - FT

Universidade de Brasília (UnB)

Campus Darcy Ribeiro

CEP 70919-970 - Brasília - DF - Brasil

Dedico este trabalho aos meus pais que sempre me incentivaram, à minha esposa Beatriz que me apoiou nos momentos de dificuldade e ao conjunto de coincidências que tornaram o universo, a vida, as ciências, e este trabalho possíveis.

AGRADECIMENTOS

Agradeço à minha esposa Beatriz pelo suporte em todos os momentos, ao colega Hugo Caetano pelas importantes contribuições à este trabalho, ao professor Adolfo Bauchspiess pela orientação e incentivo nos momentos de dificuldade e a todos que direta ou indiretamente contribuíram para este trabalho.

RESUMO

IDENTIFICAÇÃO DE FALHAS EM ESPAÇADORES DE LINHAS DE TRANSMISSÃO UTILIZANDO VISÃO ESTÉREO E REDES NEURAIAS ARTIFICIAIS

Autor: Elder Oroski

Orientador: Prof. Adolfo Bauchspiess, ENE/UnB

Programa de Pós-graduação em Engenharia Elétrica

Brasília, 21 de outubro de 2011

As interrupções no fornecimento de energia são extremamente indesejáveis nas esferas econômicas e produtivas da sociedade. Para evitá-las são realizados, periodicamente, vários procedimentos de inspeção e manutenção, nas diferentes etapas do processo de transmissão de energia. O objetivo deste trabalho é automatizar o processo de inspeção do estado de conservação das garras espaçadoras dos condutores presentes nas linhas de transmissão. Atualmente, tais inspeções são feitas por meio de sobrevôo das linhas de transmissão por um helicóptero tripulado, a partir do qual é feito um diagnóstico visual do estado das garras espaçadoras. Para automatizar tal processo o LARA/UnB desenvolve um Veículo Aéreo Não Tripulado (VANT), com um par de câmeras embarcadas, com o objetivo de percorrer e inspecionar as linhas de transmissão. Uma vez adquirido um par de imagens da garra é realizado um processamento para que se possa obter a imagem tridimensional do contorno da garra separadora. Na sequência, descritores de tal contorno são repassados à uma Rede Neural, devidamente treinada, para que a garra separadora seja considerada em bom estado, "ok", ou requerendo manutenção, "nok". A aplicação destes métodos é indicada porque os processos de inspeções visuais, realizadas por helicópteros tripulados, além de apresentarem elevados custos econômicos, ficam submetidos à imprecisão da atuação humana. Neste cenário, a automação dos processos de inspeção pode aumentar os índices de disponibilidade energética, minimizando perdas econômicas e interrupções de fornecimento de energia a sistemas críticos. Para a classificação do objeto de inspeção foi escolhida a visão estéreo, pois esta permite calcular as coordenadas 3D para do objeto e a posterior reprojeção dos pontos do mesmo sob uma perspectiva padronizada. Já as redes neurais artificiais foram escolhidas, pois as mesmas são amplamente utilizadas para classificação de padrões. Os resultados das classificações realizadas se mostraram corretos em cerca de 85% dos casos (tendo sido desconsiderados os pares de imagens que falharam nas etapas detecção de bordas ou de correspondência, nestes casos novas imagens foram adquiridas).

ABSTRACT

IDENTIFICATION OF FAILURES IN TRANSMISSION LINES USING STEREO VISION AND ARTIFICIAL NEURAL NETWORKS

Author: Elder Oroski

Supervisor: Prof. Adolfo Bauchspiess, ENE/UnB

Programa de Pós-graduação em Engenharia Elétrica

The electrical energy disruptions are undesirable in the economy and productive areas. In order to avoid them, several procedures for inspection and maintenance are conducted periodically, in various stages of power transmission. This work's objective is to automate the process of examining the state of conservation of metal conductors separating claws present in transmission lines. Currently, such inspections are made by overflight of transmission lines by a manned helicopter, and a visual diagnosis is made regarding the state of the separating claws. To automate this process the LARA/UnB laboratory is developing an unmanned aerial vehicle (UAV) with an embedded pair of video cameras in order to travel and inspect the transmission lines. Once an pair of claw's images is acquired, image processing techniques are used to obtain a tridimensional image of the contour of the separating claw. Following, the boundary descriptors are passed to a properly trained neural network, so the separating claw is classified as being in proper conditions, "ok", or in need of maintenance, "nok". These methods are indicated because the processes of visual inspections performed by manned helicopters, in addition to having high economic costs, are subject to the imprecision of human performance. In this scenario, the automation of inspection processes may increase the rate of energy availability, minimizing economic losses and disruption of power supply to critical systems. For the object's classification was chosen stereo vision, because it allows to calculate the 3D coordinates of the object and the subsequent reprojection of points from a standard perspective. The artificial neural networks were chosen because they are widely used for pattern classification. The test results proved correct classification in about 85 % (disregarding the pairs of images that failed in the step of edge detection or correspondence, in these cases new images were acquired).

SUMÁRIO

1	INTRODUÇÃO	1
1.1	DEFINIÇÃO DO PROBLEMA	1
1.2	CONTEXTUALIZAÇÃO	4
1.3	OBJETIVOS	7
1.3.1	OBJETIVO GERAL	7
1.3.2	OBJETIVOS ESPECÍFICOS	7
1.4	ORGANIZAÇÃO DA DISSERTAÇÃO	7
2	IMAGENS DIGITAIS E VISÃO ESTÉREO	9
2.1	IMAGENS DIGITAIS	9
2.2	CÂMERAS	10
2.2.1	MODELAGEM MATEMÁTICA DE UMA CEO	12
2.2.2	MATRIZ DE PROJEÇÃO PERSPECTIVA	13
2.3	SEGMENTAÇÃO DE IMAGEM	16
2.4	VISÃO ESTÉREO	17
2.4.1	CALIBRAÇÃO DAS CÂMERAS	17
2.4.2	RETIFICAÇÃO	18
2.4.3	CORRESPONDÊNCIA	19
2.4.4	CORRESPONDÊNCIA POR FEIÇÕES	20
2.4.5	CORRESPONDÊNCIA BASEADA EM ÁREAS	20
2.4.6	TRIANGULAÇÃO E RECONSTRUÇÃO 3D	22
2.5	CADEIAS DIRECIONAIS	25
2.6	SÍNTESE EM SÉRIES DE FOURIER	28
3	REDES NEURAIS	31
3.1	REDES NEURAIS COMO CLASSIFICADORES	34
3.2	REDES NEURAIS MULTICAMADAS	35
3.3	APRENDIZADO EM REDES NEURAIS	35
3.4	ERRO QUADRÁTICO MÉDIO E A REGRA DELTA GENERALIZADA	37
3.5	BACKPROPAGATION	38
3.5.1	MÉTODOS DE SEGUNDA ORDEM	39
3.6	CONCLUSÃO	40
4	IMPLEMENTAÇÃO	41
4.1	INTRODUÇÃO	41
4.2	AQUISIÇÃO DE IMAGENS	43
4.3	DETEÇÃO DE BORDAS	46

4.4	CALIBRAÇÃO DAS CÂMERAS	48
4.5	RETIFICAÇÃO	49
4.6	CORRESPONDÊNCIA	50
4.7	MODELO 3D	51
4.8	REPROJEÇÃO 2D	52
4.9	DESCRITORES DE FOURIER	53
4.10	CLASSIFICAÇÃO POR REDES NEURAIS	54
5	RESULTADOS	57
5.1	RESULTADOS DA DETECÇÃO DE BORDA.....	57
5.2	RESULTADOS DA CALIBRAÇÃO E RETIFICAÇÃO DAS CÂMERAS	59
5.3	RESULTADOS DO <i>Matching</i>	60
5.4	RESULTADOS DA RECONSTRUÇÃO 3D	61
5.5	RESULTADOS DA REPROJEÇÃO 2D	63
5.6	RESULTADOS DAS CADEIAS DIRECIONAIS	64
5.7	RESULTADOS DOS DESCRITORES DE FOURIER.....	64
5.8	RESULTADOS DO TREINAMENTO DA REDE NEURAL	67
5.9	RESULTADOS DA REDE NEURAL	68
6	CONCLUSÕES E PERSPECTIVAS	71
6.1	CONTRIBUIÇÕES E PRINCIPAIS DIFICULDADES	71
6.2	TRABALHOS FUTUROS	72
	REFERÊNCIAS BIBLIOGRÁFICAS	73
	ANEXOS	77
A	EXEMPLO DE CADEIA DIRECIONAL CODIFICADA.....	79
B	CÓDIGO	81
B.1	FUNÇÃO PRINCIPAL	81
B.2	TRATAMENTO	83
B.3	SEGMENTAÇÃO DA IMAGEM	85
B.4	MATTING	87
B.5	PREPARATIVOS PARA OBTENÇÃO DA CADEIA DIRECIONAL	89
B.6	CADEIA DIRECIONAL.....	90
B.7	COORDENADAS TRIDIMENSIONAIS	94
B.8	CORRESPONDÊNCIA	96
B.9	RECONSTRUÇÃO 3D	98
B.10	REDES NEURAIS.....	101

LISTA DE FIGURAS

1.1	Exemplo de linha de transmissão (com destaque para os espaçadores de cabos de mesma fase).....	2
1.2	Exemplo de um espaçador de cabos de uma linha de 500 kV.....	2
1.3	Exemplo de um espaçador de cabos de uma mesma fase utilizados em linha de 500 kV.	3
1.4	VANT e câmeras a serem utilizados no processo de inspeção.	4
1.5	Ilustração da garra de um separador: (a) Bem ajustada (ok). (b) Solta (nok). ...	5
2.1	Efeito de quantização na aquisição de imagem por câmera digital.	10
2.2	Processo básico de formação de uma imagem em uma câmera <i>Pin-Hole</i>	11
2.3	Planos em uma câmera do tipo <i>Pin-Hole</i>	11
2.4	Sistemas de Coordenadas.	12
2.5	Planificação da imagem 2.4 no plano $h=0$	12
2.6	(a) Imagem Original. (b) Seleção da Região de Interesse. (c) Trimap [13].	17
2.7	Separação de planos obtida a partir da imagem exibida na figura 2.6 [13].	17
2.8	Exemplo de par estéreo: (a) Imagem Esquerda. (b) Imagem Direita.	20
2.9	Associação de feições da imagem esquerda com feições da imagem direita, relativa à figura 2.8.	20
2.10	Janelas de correlação.	21
2.11	Triangulação.	23
2.12	Espaço Imagem e Espaço Objeto.	24
2.13	Cadeia Direcional.....	25
2.14	Exemplo de uma imagem codificada em Cadeia de Freeman.....	26
2.15	Exemplo de projeção nos eixos x e y da imagem codificada da figura 2.14.....	27
3.1	Modelo de um neurônio biológico [23].	31
3.2	Neurônio Artificial [23].	32
3.3	(a) Função Degrau e (b) Função Linear.	33
3.4	Função Sigmóide.	33
3.5	Função Tangente Hiperbólica.....	34
3.6	Discriminador Linear [23].....	35
3.7	Ilustração de uma redes neurais multicamadas [24].	36
3.8	Ilustração de um mínimo local e global em uma superfície de erro [23].	37
4.1	Diagrama em blocos do reconhecimento de falhas nos espaçadores de mesma fase (com indicação das seções correspondentes).	42
4.2	Tela do software de aquisição de imagens SRI- Small Vision System.....	43
4.3	Cameras Videre modelo VTH-MDCS3-VARX.	44

4.4	Sistema de coordenadas referenciado à câmera esquerda (SCC).	44
4.5	Trecho de linha de transmissão com dois espaçadores, suspenso no LARA.	44
4.6	Ilustração da elevação das câmeras em relação à linha de transmissão.	45
4.7	Ilustração do azimute de orientação da câmera.	45
4.8	Par Estéreo. (a) Imagem obtida pela câmera da esquerda. (b) Imagem Obtida pela câmera da direita.	46
4.9	Par Estéreo. (a) Máscara utilizado no <i>Matting</i> da Imagem Esquerda. (b) Máscara utilizado no <i>Matting</i> da Imagem Esquerda.	47
4.10	Par Estéreo. (a) Contorno obtido a partir da imagem esquerda. (b) Contorno obtido a partir da imagem direita.	47
4.11	<i>Toolbox Calib</i> utilizada para calibração de cada uma das câmeras.	48
4.12	Imagens utilizadas para calibração das câmeras.	48
4.13	Posição das imagens utilizadas na Calibração de uma das câmeras.	49
4.14	Ilustração de um par estéreo retificado.	49
4.15	Pontos Relevantes C1 a C8: ‘Tokens’ para o processo de Correspondência entre pixels num par de imagens.	50
4.16	Imagem tridimensional reconstituída do contorno de uma garra metálica.	51
4.17	Vetor que descreve a orientação da câmera virtual \vec{C}	52
4.18	Plano ρ , paralelo a face frontal da garra metálica.	52
4.19	Plotagem do contorno de uma garra metálica a partir de Séries de Fourier Bidimensionas.	54
4.20	Interface de treinamento das Redes Neurais.	54
5.1	Ilustração de detecção de bordas: (a) Bem sucedida. (b) Mal sucedida.	57
5.2	Exemplo de detecção de borda para aquisição de imagem à 45°.	58
5.3	Visualização de correspondência de pixels, imagem típica OK.	60
5.4	Imagem Tridimensional do contorno de uma garra metálica.	62
5.5	Imagem reprojetaada a partir da imagem tridimensional de uma garra metálica.	63
5.6	Plotagens para diferentes números de harmônicos.	65
5.7	MSE ao longo do processo de treinamento da Rede Neural.	67
5.8	Gradiente do MSE ao longo do processo de treinamento da Rede Neural.	67
5.9	Imagens que compuseram o conjunto de testes (parte 1).	69
5.10	Imagens que compuseram o conjunto de testes (parte 2).	70

Símbolos

\mathbf{a}	Vetor da base do sistema de coordenadas da câmera;
a_0	Coefficiente da Série de Fourier relativo ao <i>offset</i> vertical (nível contínuo) da função $f(x)$;
a_n	Coefficiente da Série de Fourier relativo às componentes pares da função $f(x)$;
A_0	Coefficiente da Série de Fourier Bidimensional relativo ao <i>offset</i> vertical da função $f(x,y)$;
A_n	Coefficiente da Série de Fourier Bidimensional relativo às componentes pares horizontais da função $f(x,y)$;
b_0	Coefficiente da Série de Fourier relativo ao <i>offset</i> horizontal (nível contínuo) da função $f(x)$;
b_n	Coefficiente da Série de Fourier relativo às componentes ímpares da função $f(x)$;
B	Background;
B_0	Coefficiente da Série de Fourier Bidimensional relativo ao <i>offset</i> horizontal da função $f(x,y)$;
B_n	Coefficiente da Série de Fourier Bidimensional relativo às componentes ímpares horizontais da função $f(x,y)$;
C	Foco da Câmera;
C_n	Coefficiente da Série de Fourier Bidimensional relativo às componentes pares verticais da função $f(x,y)$;
d_h	Disparidade horizontal;
d_v	Disparidade vertical;
D_n	Coefficiente da Série de Fourier Bidimensional relativo às componentes ímpares verticais da função $f(x,y)$.
$E(w)$	Erro quadrático;
f	Distância focal da câmera;
$f(x)$	Função genérica representada em Séries de Fourier com referência horizontal;
$f(y)$	Função genérica representada em Séries de Fourier com referência vertical;
$f(x, y)$	Função genérica bidimensional representada em Séries de Fourier;
\mathcal{F}	Plano de Focal;

F	Foreground;
$g(v)$	Função de Ativação em um neurônio artificial;
\mathbf{h}	Vetor da base do sistema de coordenadas da câmera;
i_0	Coordenada vertical do ponto Principal M' ;
I	Intensidade em níveis de Cinza;
j_0	Coordenada horizontal do ponto Principal M' ;
\mathbf{m}	Vetor constituído por $[U, V, S]^T$;
M'	Ponto Principal;
o	Origem do sistema de coordenadas da câmera;
p	Ponto Objeto;
P	Matriz de projeção perspectiva;
\tilde{P}_{n1}	Matriz de Retificação relativa a câmera da esquerda;
\tilde{P}_{n2}	Matriz de Retificação relativa a câmera da direita;
\mathcal{R}	Plano de Retina;
\mathfrak{R}	Matriz de Rotação;
\mathbf{R}	Vetor que liga a origem dos sistemas de coordenadas externo e da câmera;
\mathbb{R}^n	Espaço vetorial n -dimensional;
\mathbf{S}	Vetor que liga a origem dos sistemas de coordenadas externo e o Ponto P ;
S	Produto interno entre \mathbf{T} e \mathbf{a} ;
t_m	Comprimento dos m primeiros segmentos de uma cadeia direcional codificada;
T	Comprimento total de uma cadeia direcional codificada;
\mathbf{T}	Vetor diferença entre os pontos P e O ;
\mathfrak{T}	Vetor de Translação;
T_{32}	Transformação Linear que leva um ponto tridimensional a sua projeção bidimensional;
T_{23}	Transformação Linear que leva um ponto bidimensional ao seu equivalente tridimensional;
$\mathbf{T}_{\mathbf{av}}$	Projeção do vetor \mathbf{T} no plano formado por \mathbf{a} e \mathbf{v} ;
u	Coordenada horizontal referenciada ao SCC;
U	Produto interno entre \mathbf{T} e \mathbf{v} multiplicado escalarmente por h ;
v	Coordenada vertical referenciada ao SCC;
\mathbf{v}	Vetor da base do sistema de coordenadas da câmera;

V	Produto interno entre \mathbf{T} e \mathbf{h} multiplicado escalarmente por h ;
\mathbf{w}	Vetor constituído pelas coordenadas de um ponto $[x, y, z]^T$;
\mathbf{W}	Vetor de pesos em um neurônio artificial;
W_i	i -ésimo peso neuronal em um neurônio artificial;
\mathbf{X}	Vetor de entrada de um neurônio artificial;
x_i	i -ésima entrada de um neurônio artificial;
y	Saída de um neurônio artificial.

Símbolos Gregos

α_i	Opacidade do i -ésimo pixel da imagem;
δu	Dimensão horizontal de um pixel;
δv	Dimensão vertical de um pixel;
Δt_i	Comprimento de um vetor que representa o contorno de um objeto codificado em cadeias direcionais;
Δx_i	Projeção de um segmento codificado em cadeia direcional no eixo horizontal da imagem;
Δy_i	Projeção de um segmento codificado em cadeia direcional no eixo vertical da imagem;
Φ	Ângulo de elevação das câmeras em relação ao espaçador de condutores de mesma fase em uma linha de transmissão;
Θ	Ângulo de azimute das câmeras em relação ao espaçador de condutores de mesma fase em uma linha de transmissão;
θ	Vetor de <i>bias</i> de um neurônio artificial;
η	Fator de Aprendizado;
∇	Gradiente.

Subscritos

min	Mínimo;
max	Máximo.

Sobrescritos

$-$	Valor médio;
d	desejado.

Siglas, Acrônimos e Abreviaturas

CEO	Camera Escura de Orifício (PHC em inglês);
CC	Correlação Cruzada;
CCD	Charge Coupled Device;
CMOS	Complementary Metal Oxide Semiconductor;
CNPQ	Conselho Nacional de Desenvolvimento Científico e Tecnológico;
CVC	Ciclo de Varredura de Correlação;
DFB	Descritores de Fourier Bidimensionais;
EQ	Erro Quadrático;
FPS	Frame Per Second;
<i>imD</i>	imagem Direita;
<i>imE</i>	imagem Esquerda;
LARA	Laboratório de Automação e Robótica;
MPP	Matriz de Projeção Perspectiva;
MSE	Mean Squared Error;
PE	Parâmetros Extínicos da Câmera;
PI	Parâmetros Intrínscos da Câmera;
PDI	Processamento Digital de Imagem;
PGEA	Programa de Pós Graduação em Engenharia de Sistemas Eletrônicos e de Automação;
RI	Região de Interesse;
RNA	Rede Neural Artificial;
SDA	Soma das Diferenças Absolutas;
SCI	Sistema de Coordenadas da Imagem;
SCC	Sistema de Coordenadas da Câmera;
SCE	Sistema de Coordenadas Externo;
UnB	Universidade de Brasília;
VANT	Veículo Aéreo Não Tripulado (UAV em inglês);
3D	Tridimensional.

1 INTRODUÇÃO

*Não há fatos eternos, como
não há verdades absolutas.*

Friedrich Nietzsche

Os processos de inspeção em linhas de transmissão vêm constituindo grande desafio para a área de automação nas últimas décadas. Quesitos como confiabilidade do processo e minimização dos tempos de execução e recursos alocados impulsionam pesquisas em todo o mundo. Atualmente tais processos são realizados por inspetores humanos e geralmente com baixo nível de automação. As inspeções normalmente são realizadas por helicópteros tripulados e a identificação das falhas é feita visualmente. Entre os custos deste processo encontram-se as alocações de um piloto e de um técnico para observação das linhas além do custo operacional do helicóptero. Considerando-se que as linhas de alta tensão possuem extensões elevadíssimas, chegando a aproximadamente 74 mil km no Brasil (somando-se as de linhas de 500kV e 750kV), o custo operacional de tais inspeções torna-se elevado às concessionárias.

Neste cenário, a proposta deste trabalho é automatizar o processo de inspeção a partir da utilização de um VANT (veículo aéreo não tripulado), de um par estéreo de câmeras, de algoritmos de processamento digital de imagens, visão computacional, entre outros. Espera-se desta forma minimizar os custos de inspeção e tornar o processo mais eficiente do ponto de vista econômico, de segurança e de confiabilidade.

1.1 DEFINIÇÃO DO PROBLEMA

Num mundo globalizado, a interrupção de alguns serviços essenciais pode gerar grandes prejuízos econômicos e sociais. O fornecimento de energia elétrica certamente se enquadra neste cenário. As interrupções destes serviços se dão principalmente por panes de geração ou por falhas nas linhas de transmissão. Para evitá-las são estabelecidos procedimentos de inspeção que verificam periodicamente o estado destas linhas.

A figura 1.1 ilustra uma linha de 500 kV com um de seus componentes (espaçadores de condutores de mesma fase) circulados. Um destes espaçadores, inserido em uma linha de transmissão, pode ser observados na figura 1.2.

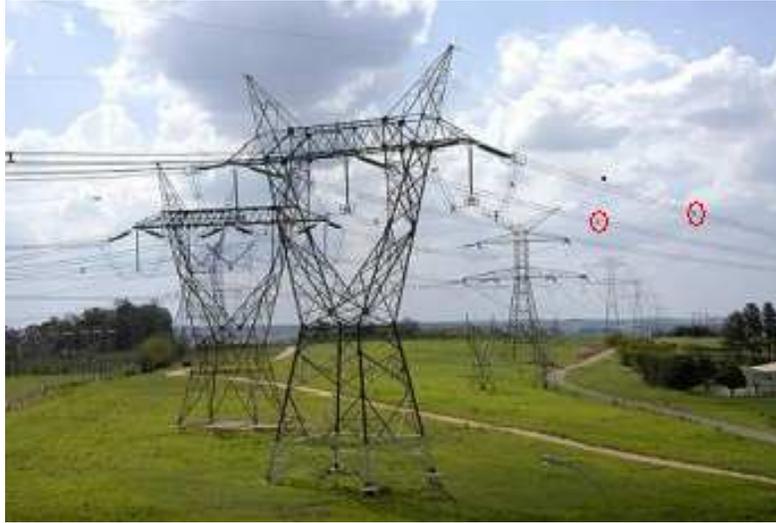


Figura 1.1: Exemplo de linha de transmissão (com destaque para os espaçadores de cabos de mesma fase).

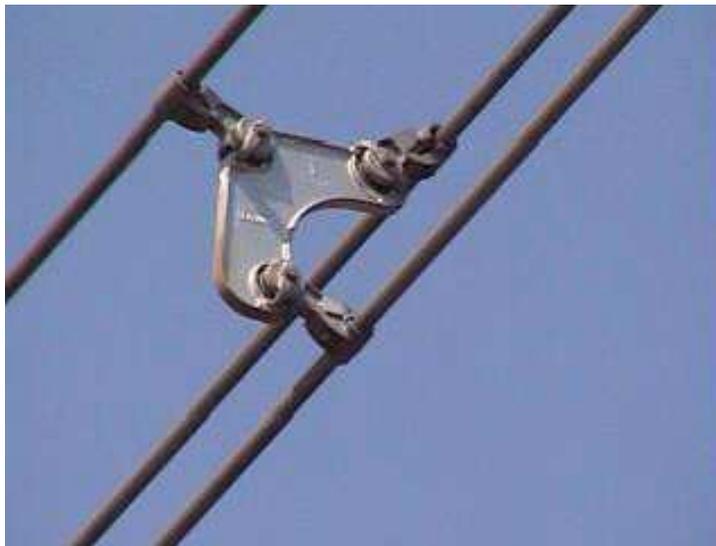


Figura 1.2: Exemplo de um espaçador de cabos de uma linha de 500 kV.

Nas torres de transmissão os dispositivos espaçadores servem para evitar que os cabos de uma mesma fase se toquem por ação do vento. Uma imagem mais detalhada de um destes dispositivos pode ser observada na figura 1.3. O processo de inspeção destes elementos consiste basicamente na verificação do estado das garras que compõem os elementos separadores, pois com o passar do tempo os parafusos que prendem estas garras sofrem de sajuste podendo ficar frouxas ou mesmo se soltar, levando assim a uma possível interrupção no fornecimento de energia elétrica em uma determinada região.



Figura 1.3: Exemplo de um espaçador de cabos de uma mesma fase utilizados em linha de 500 kV.

A inspeção periódica das garras dos espaçadores se justifica, pois o afrouxamento das garras pode levar ao rompimento de cabos na linha. Isto devido à colisão constante entre os cabos pela ação de cargas de vento perpendiculares aos mesmos.

Atualmente, o procedimento de inspeção nas linhas de transmissão é realizado por meio de sobrevôo de helicóptero com um técnico a bordo. Um levantamento visual do estado das garras do separador é realizado e as mesmas são submetidas à manutenção, caso se constate a necessidade. Este processo é extremamente custoso, do ponto de vista econômico, pois exige a alocação de um helicóptero, de um piloto e um técnico dedicado à verificação. E a tudo isso ainda soma-se o fato de que uma inspeção humana pode apresentar baixo grau de confiabilidade em algumas situações.

Este trabalho objetiva analisar um processo de inspeção dos espaçadores de uma mesma fase, em linhas de transmissão, usando métodos de visão estéreo e de redes neurais artificiais. Assim, o problema aqui tratado pode ser expresso pela seguinte pergunta:

A Identificação de falhas nas garras dos espaçadores de linhas de transmissão pode ser automatizada, de maneira eficiente, utilizando-se métodos da visão estéreo e de redes neurais artificiais?

A solução proposta a tal problema parte da premissa que um VANT, mostrado na figura 1.4, sobrevoaria as linhas de transmissão e faria a aquisição de imagens-estéreo das garras metálicas, que mantêm os espaçadores presos aos cabos da linha de transmissão. Com o processamento destas imagens as garras seriam classificadas em: "ok", para garras em bom estado, ou "nok", para garras que requerem manutenção. Tal classificação é realizada a partir de uma Rede Neural devidamente treinada para tanto. Sendo um técnico acionado para manutenção quando a classificação indicar problemas com o espaçador.

As razões para a utilização da visão estéreo e das redes neurais artificiais serão abordadas nas seções 2.5 e 3.1 .



Figura 1.4: VANT e câmeras a serem utilizados no processo de inspeção.

1.2 CONTEXTUALIZAÇÃO

Segundo Katrasnik, 2010, [1], existem 2 métodos básicos de inspeção de linhas de transmissão. Um deles se baseia no patrulhamento à pé da linha de transmissão e o outro na inspeção assistida por helicóptero. Na inspeção à pé, um grupo, normalmente de 2 pessoas, percorre a linha de transmissão verificando torre por torre. Já na inspeção por helicópteros, uma equipe, composta por um piloto e um técnico de inspeção, percorrem as linhas analisando visualmente o estado de seus componentes.

Num comparativo entre as metodologias apresentadas para inspeção de linhas de transmissão, inicialmente o método que faz uso dos helicópteros parece ser mais eficiente que a inspeção visual à pé. Mas apesar da inspeção assistida por helicóptero ser mais rápida em relação ao outro método, tal inspeção também é mais cara e menos acurada [1].

Na inspeção assistida por helicóptero, a aeronave precisa se deslocar em baixo nível

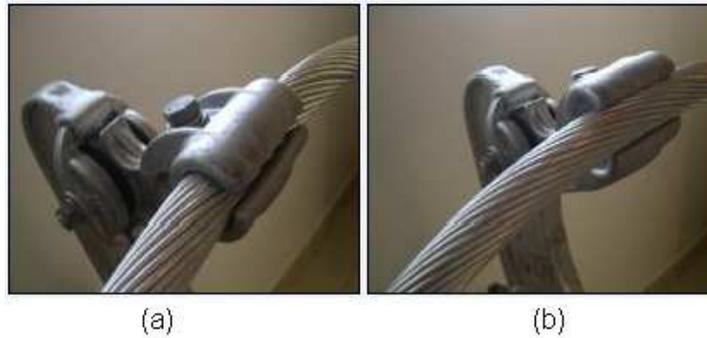


Figura 1.5: Ilustração da garra de um separador: (a) Bem ajustada (ok). (b) Solta (nok).

de vôo para que os componentes da linha possam ser visualizados em detalhes. Como o elemento avaliador das condições destes componentes é a visão humana, a exatidão/precisão do processo fica subordinada à percepção visual dos tripulantes que realizam tal avaliação.

Como as inspeções visuais são geralmente lentas, tediosas e subjetivas tem-se, às vezes, grandes defeitos não identificados e negligenciados [1].

No processo de inspeção baseado em helicópteros tripulados, além da exatidão/precisão tornar-se variável de acordo com as habilidades do tripulante, há ainda os fatores relacionados ao elevado custo econômico de inspecionar extensas linhas de transmissão. Estes custos são altos porque este método requer a alocação de um piloto, de um técnico que realizará a inspeção, fora o custo operacional do helicóptero em si.

Além do risco à tripulação do helicóptero, neste processo, tem-se o risco às habitações próximas às linhas de transmissão. Isto ocorre pois a realização de inspeção acontece suficientemente próxima da linha de alta tensão para que seus componentes possam ser inspecionados [2]. Neste cenário, além de custos econômicos elevados, precisão/exatidão variável, este processo, tal como é realizado, implica em sérios riscos à tripulação da aeronave e à população das regiões adjacentes.

Segundo Fu *et al*, 2006, [3], robôs autônomos de inspeção de linhas de alta tensão podem desempenhar várias tarefas em uma linha sem desenergizá-la e substituindo trabalhadores humanos em operações perigosas e altamente especializadas.

Para Whitworth et Al, 2001, [2], é necessário evitar áreas onde o baixo nível de vôo possa causar distúrbios aos moradores das proximidades durante as inspeções. O que implica usar o método de inspeção da linha à pé nestas regiões, algo extremamente custoso do ponto de vista econômico. Assim, uma abordagem possível é automatizar parcialmente o processo de inspeção, usando técnicas de vigilância por vídeo.

Segundo Caballero, 2005, [4], muitos dos problemas encontrados em inspeções, em áreas urbanas, podem ser solucionados usando veículos aéreos não tripulados, com plataformas de dimensões reduzidas que permitem a colocação de câmeras e outros sensores para a inspeção do objeto em questão. Desta forma não são gerados problemas com sobrevôo de áreas

habitadas, por aeronaves de grande porte. Assim, o custo operacional de inspeção seria reduzido. Além de automatizar o processo, o que aumentaria a confiabilidade da inspeção, uma vez que este não estaria sujeito a fatores humanos.

Para Djick, 2003, [5], a visão estéreo é muito utilizada para reconhecimento de objetos tridimensionais por ser capaz de prover suas coordenadas 3D. Para Ostendorp, 2000, [6], experimentos vêm demonstrando que imagens aéreas, digitais de alta resolução, etiquetadas com as coordenadas GPS, podem ser utilizadas com sucesso na identificação da maioria das falhas de linhas de transmissão. Neste cenário, foram exploradas câmeras de alta resolução, embarcadas em um VANT, para compor a visão estéreo do objeto a se inspecionar.

Uma outra possibilidade na inspeção de linhas de transmissão por VANT é expressa por Lamela, 1998, [7], que utiliza uma câmera e um laser para que as coordenadas do objeto inspecionado sejam obtidas por triangulação. Esta técnica é conhecida como imageamento ativo, e tem por vantagem não exigir a resolução do problema da correspondência (necessária nos métodos baseados em visão estéreo). As diferenças entre os métodos passivos e ativos de obtenção de imagem e a definição de correspondência serão abordadas com mais profundidade no capítulo 3.

Uma vez feita a aquisição das imagens pelo VANT as imagens precisam ser processadas, para que então os objetos representados possam ser classificadas em "ok" ou "nok". Os métodos de processamento de imagens podem ser encontrados em Gonzalez, 2002, [8], e acontecem seguindo as seguintes etapas: realce, segmentação e interpretação do conteúdo das imagens. As informações obtidas a partir das imagens podem ser utilizadas no planejamento de manutenção, permitindo ao engenheiro saber a localização e a natureza da pane [2].

O processo de classificação geralmente depende de estratégias de software para o reconhecimento de padrões. Em Haykin e Gonzalez & Woods, 2002, [8, 9], são encontradas, dentre outras, técnicas de reconhecimento de padrões baseadas em Redes Neurais Artificiais, que estão entre as mais utilizadas.

No laboratório LARA, da UnB, Oliveira & Guedes, 2007, [10], implementaram uma metodologia de avaliação de falhas nas garras de espaçadores em linhas de transmissão. Tal metodologia foi baseada no processamento de imagens e nas Redes Neurais Artificiais. O escopo do mesmo envolvia a aquisição de imagens a partir de uma câmera com posição fixa em relação ao objeto de inspeção. Dando continuidade a esse trabalho, esta dissertação abordará uma generalização de tal processo, em que se propõem a utilização de um par de câmeras, embarcadas em um VANT, para as inspeções. Com os recursos do par de imagens adquirido (Visão Estéreo) espera-se realizar a avaliação de falhas nos espaçadores sem a restrição de que as câmeras necessitem assumir uma posição fixa em relação ao objeto de inspeção.

1.3 OBJETIVOS

Os objetivos, geral e específicos, deste trabalho serão expostos nos itens a seguir.

1.3.1 Objetivo Geral

O objetivo deste trabalho consiste em desenvolver um método de identificação de falhas em linhas de transmissão, mais especificamente nas garras metálicas que prendem o dispositivo espaçador para separar os condutores de mesma fase. Para tanto serão utilizados métodos de visão estéreo, sobre as imagens das garras metálicas e de redes neurais, na classificação do estado de conservação destas garras.

1.3.2 Objetivos Específicos

Para se alcançar o objetivo geral são listados os seguintes objetivos específicos:

- Detectar as bordas das imagens do par estéreo;
- Obter a correspondência entre os *pixels* das imagens do par estéreo;
- Obter a imagem tridimensional do contorno da garra metálica;
- Classificar o estado das garras dos espaçadores a partir de uma Rede Neural devidamente treinada para tanto.

1.4 ORGANIZAÇÃO DA DISSERTAÇÃO

Esta dissertação está dividida conforme descrito a seguir:

No capítulo 2 trata-se da Visão Estéreo incluindo os seguintes tópicos: imagens digitais, detecção de bordas, calibração de câmeras, retificação de imagem e reconstrução de modelos tridimensionais de um objeto a partir de um par de imagens estéreo.

O capítulo 3 aborda os tópicos relativos às Redes Neurais Artificiais, como: definição de arquitetura, treinamento e aplicações das Redes Neurais Artificiais como sistemas classificadores.

O capítulo 4 descreve os algoritmos e as opções de implementação. Mais especificamente os descritivos dos processos de aquisição de imagem, dos tratamentos utilizados nos contornos da imagem para codificação em cadeias direcionais e obtenção dos Descritores de Fourier. Tem-se ainda a descrição da Rede Neural que permite a classificação das falhas nas linhas de alta tensão.

No capítulo 5 tem-se a apresentação dos resultados em cada etapa do processo de obtenção da imagem tridimensional das garras separadoras nas linhas de transmissão e de classificação de seu estado de conservação.

O capítulo 6 foi reservado às considerações finais, contribuições, dificuldades de implementação e a propostas de trabalhos futuros e às conclusões deste trabalho.

2 IMAGENS DIGITAIS E VISÃO ESTÉREO

Neste capítulo são apresentados tópicos relativos a imagens digitais, modelagem matemática de câmeras e princípios de Visão Estéreo. Estas técnicas são blocos construtivos da metodologia proposta para reconhecimento de falhas em linhas de transmissão.

2.1 IMAGENS DIGITAIS

As imagens digitais utilizam elementos fotossensíveis discretos (CCD) para captarem uma imagem. Tais sensores emitem sinais elétricos proporcionais à intensidade da onda eletromagnética (com um determinado comprimento de onda) que os atinge [11]. A unidade mínima da fotografia digital é o *pixel* ("picture element"). A cada pixel corresponde uma determinada área na imagem e cada pixel está associado a um valor da intensidade média de brilho/luz que é refletida na mesma área. Os pixels são ordenados de forma matricial ao longo da imagem, sendo suas coordenadas (u_p, v_p) dadas, em geral, relativamente ao canto superior esquerdo [11].

Um exemplo de uma matriz representando uma imagem digital é mostrada em (2.1).

$$\mathbf{I} = \begin{pmatrix} f(1, 1) & f(1, 2) & \dots & f(1, m) \\ f(2, 1) & f(2, 2) & \dots & f(2, m) \\ \vdots & \vdots & \vdots & \vdots \\ f(n, 1) & f(n, 2) & \dots & f(n, m) \end{pmatrix}. \quad (2.1)$$

Como os objetos focalizados por uma câmera normalmente são contínuos e as câmeras digitais trabalham com linhas/matrizes de elementos fotossensíveis discretos as imagens geradas pelas mesmas sofrem o efeito da **amostragem** que consiste na representação discreta da imagem de um objeto contínuo. Este efeito, bem como a quantização dos brilhos medidos, podem ser observados na figura 2.1.

As imagens digitais geralmente são classificadas em:

- Imagens monocromáticas: são funções $f(x, y)$, onde x e y são as coordenadas espaciais e o valor de f é proporcional ao brilho (ou níveis de cinza) no ponto (x, y) . Uma imagem digital é uma imagem $f(x, y)$, discretizada tanto em coordenadas espaciais quanto em brilho [8].
- Imagens coloridas: podem ser concebidas, por exemplo, como uma concatenação de três funções monocromáticas com as seguintes cores: vermelho: R (Red), verde: G

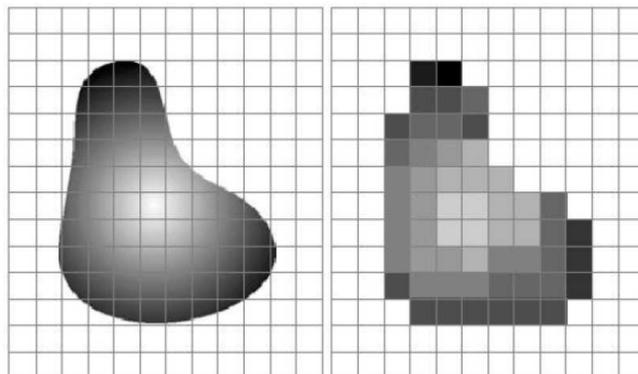


Figura 2.1: Efeito de quantização na aquisição de imagem por câmera digital.

(Green) e azul: B (Blue). Assim, os mesmos princípios válidos para as imagens monocromáticas podem ser aplicados às imagens coloridas, considerando que estas são compostas por três imagens monocromáticas.

Sejam u e v as coordenadas de um determinado pixel na imagem, e (i_0, j_0) as coordenadas do ponto principal M' (mostrado na figura 2.3). As coordenadas de uma imagem são dadas por

$$u = (j - j_0) \cdot \delta u \quad (2.2)$$

e

$$v = (i - i_0) \cdot \delta v, \quad (2.3)$$

onde: δu e δv são comprimentos das células do plano de retina nas direções u e v .

2.2 CÂMERAS

O modelo mais simples de captação de imagens por câmeras fotográficas é o *Pin-Hole Camera* (PHC), ou seja, a Câmera Escura de Orifício (CEO). Em tal modelo admite-se que a luz se propaga em linha reta e que todos os raios captados pela câmera convergem para um ponto C , denominado **foco**.

A câmera *pin-hole* é um modelo básico de câmera, sem lentes, com um pequeno orifício em uma das faces. Este tipo de câmera é interessante sob o ponto de vista matemático porque a modelagem de seu comportamento pode ser utilizada para a descrição matemática de câmeras mais complexas como uma câmera digital, bastante utilizada nas áreas de PDI (Processamento Digital de Imagem).

A CEO é composta por uma "caixa" com um furo em um dos lados. Idealmente a luz é impedida de penetrar nesta "caixa" a não ser pelo orifício situado em uma das faces da caixa. Pelo orifício central a luz é captada para dentro da câmera e a imagem é projetada na parede oposta ao orifício, sofrendo uma inversão. Este fenômeno pode ser observado na Figura 2.2.

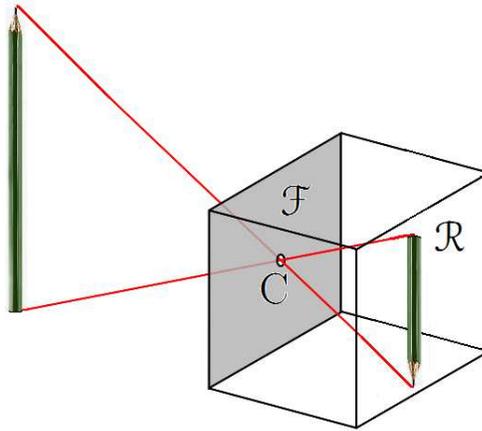


Figura 2.2: Processo básico de formação de uma imagem em uma câmera *Pin-Hole*.

Uma câmera escura de orifício pode ser descrita simplificada por três elementos matemáticos:

- Plano \mathcal{R} , denominado **plano de retina**;
- Plano \mathcal{F} , denominado **plano focal**;
- O ponto C , pertencente ao plano \mathcal{F} , designado **foco** da câmera;

Tais elementos matemáticos podem ser observados na figura 2.3:

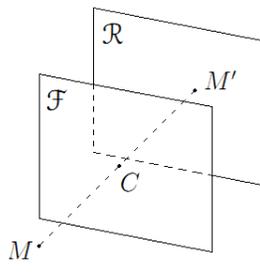


Figura 2.3: Planos em uma câmera do tipo *Pin-Hole*.

Dentre as funções desempenhadas por tais elementos cabe destacar que é no plano de retina que se tem a formação das imagens. Passando pelo ponto C e perpendicularmente aos planos R e F tem-se o **eixo óptico**. À interseção do eixo óptico e ao plano de retina denomina-se **ponto principal**.

2.2.1 Modelagem Matemática de uma CEO

Nesta seção objetiva-se relacionar as coordenadas do ponto a ser observado às posições do *pixel* onde se forma a imagem correspondente. Isto levando-se em consideração a posição e orientação da câmera. Para tanto, o processo de formação de uma imagem em uma câmera CEO pode ser descrito pela figura 2.4.

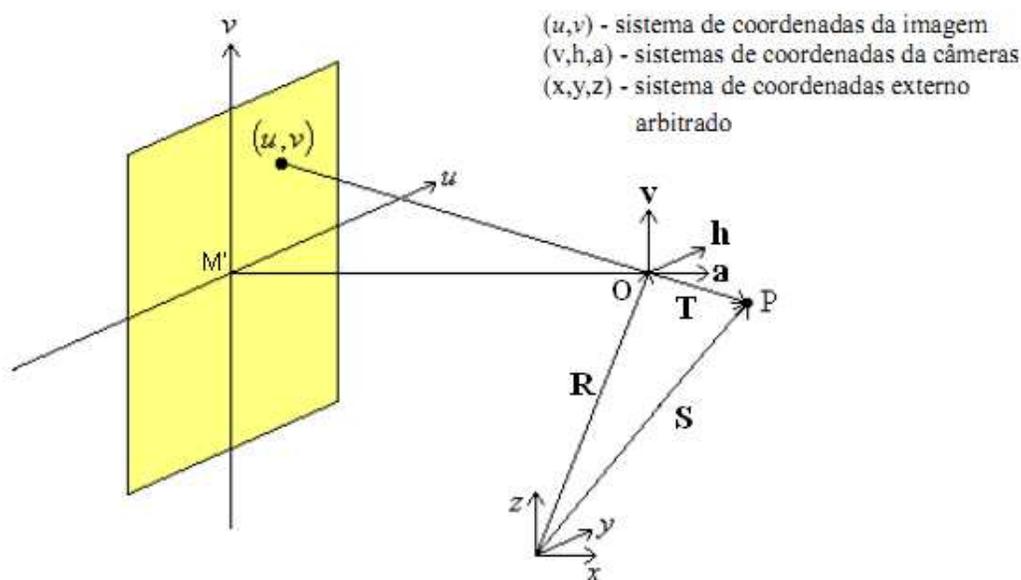


Figura 2.4: Sistemas de Coordenadas.

Somando-se vetorialmente os vetores que relacionam os sistemas de referência e o ponto a ser observado chega-se a

$$S = R + T \quad \therefore \quad T = S - R. \quad (2.4)$$

Seja um vetor T que representa a distância do objeto a ser observado em relação ao sistema de coordenadas da câmera $[v, h, a]$ e seja o vetor T_{av} sua projeção no plano $h = 0$. Assim pode-se estabelecer uma semelhança de triângulos no plano $h = 0$ (plano que contém a e b). Tais triângulos podem ser observados na figura 2.5:

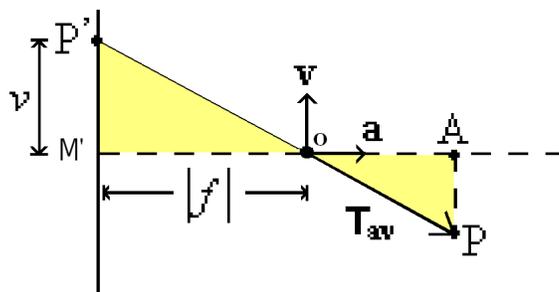


Figura 2.5: Planificação da imagem 2.4 no plano $h=0$.

Projetando-se o vetor \mathbf{T}_{av} nos eixos definidos pelos vetores \mathbf{a} e \mathbf{v} chega-se a

$$\frac{|\mathbf{T}_{av} \cdot \mathbf{v}|}{|\mathbf{T}_{av} \cdot \mathbf{a}|} = \frac{|\mathbf{v}|}{|f|} \quad (2.5)$$

onde $|f|$ a distância focal da câmera e considerando a mesma como positiva em relação ao eixo definido pelo vetor \mathbf{a} , tem-se a

$$\frac{\mathbf{T}_{av} \cdot \mathbf{v}}{\mathbf{T}_{av} \cdot \mathbf{a}} = \frac{v}{f}. \quad (2.6)$$

Considerando-se ainda que os vetores \mathbf{T} e \mathbf{T}_{av} possuem as mesmas projeções sobre o plano $\mathbf{h} = 0$ chega-se a

$$\frac{\mathbf{T} \cdot \mathbf{v}}{\mathbf{T} \cdot \mathbf{a}} = \frac{v}{f}. \quad (2.7)$$

Analogamente em relação ao eixo formado pelo vetor \mathbf{h} pode-se chegar à coordenada u a partir de

$$\frac{\mathbf{T} \cdot \mathbf{h}}{\mathbf{T} \cdot \mathbf{a}} = \frac{u}{f}. \quad (2.8)$$

Considerando a indexação dos pixels em uma imagem, definida pelas equações (2.2) e (2.3), obtém-se

$$\frac{\mathbf{T} \cdot \mathbf{v}}{\mathbf{T} \cdot \mathbf{a}} = \frac{(i - i_0) \cdot \delta v}{f} \quad (2.9)$$

e

$$\frac{\mathbf{T} \cdot \mathbf{h}}{\mathbf{T} \cdot \mathbf{a}} = \frac{(j - j_0) \cdot \delta u}{f} \quad (2.10)$$

onde δu e δv são as dimensões do pixel.

2.2.2 Matriz de Projeção Perspectiva

A Matriz de Projeção Perspectiva (MPP) é uma transformação linear responsável por converter um dado ponto tridimensional P , de coordenadas $[x, y, z]$, em uma projeção bidimensional deste ponto. Tal matriz é representada pela equação (2.20) e pode ser utilizada para representar o processo de aquisição de uma imagem por uma câmera do tipo *pinhole*.

Definindo-se $S = \mathbf{T} \cdot \mathbf{a}$, $U = f \cdot \mathbf{T} \cdot \mathbf{h}$ e $V = f \cdot \mathbf{T} \cdot \mathbf{v}$, obtém-se

$$\begin{cases} S = T_x \cdot a_x + T_y \cdot a_y + T_z \cdot a_z \\ V = f \cdot (T_x \cdot v_x + T_y \cdot v_y + T_z \cdot v_z) \\ U = f \cdot (T_x \cdot h_x + T_y \cdot h_y + T_z \cdot h_z) \end{cases} \quad (2.11)$$

o que na forma matricial pode ser escrito como

$$\begin{bmatrix} U \\ V \\ S \end{bmatrix} = \begin{bmatrix} f \cdot h_x & f \cdot h_y & f \cdot h_z \\ f \cdot v_x & f \cdot v_y & f \cdot v_z \\ a_x & a_y & a_z \end{bmatrix} \cdot \begin{bmatrix} T_x \\ T_y \\ T_z \end{bmatrix}. \quad (2.12)$$

Sendo que a relação entres os elementos U , V e S representa as coordenadas da imagem i e j , ou seja

$$j = \frac{U}{S} \quad (2.13)$$

e

$$i = \frac{V}{S}. \quad (2.14)$$

Substituindo-se a equação (2.4) em (2.12), obtém-se ainda

$$\begin{bmatrix} U \\ V \\ S \end{bmatrix} = \begin{bmatrix} f \cdot h_x & f \cdot h_y & f \cdot h_z \\ f \cdot v_x & f \cdot v_y & f \cdot v_z \\ a_x & a_y & a_z \end{bmatrix} \cdot \begin{bmatrix} S_x - R_x \\ S_y - R_y \\ S_z - R_z \end{bmatrix} \quad (2.15)$$

e

$$\begin{bmatrix} U \\ V \\ S \end{bmatrix} = \begin{bmatrix} f \cdot h_x & f \cdot h_y & f \cdot h_z & 0 \\ f \cdot v_x & f \cdot v_y & f \cdot v_z & 0 \\ a_x & a_y & a_z & 0 \end{bmatrix} \cdot \begin{bmatrix} S_x - R_x \\ S_y - R_y \\ S_z - R_z \\ 1 \end{bmatrix},$$

Assim obtém-se a equação

$$\begin{bmatrix} U \\ V \\ S \end{bmatrix} = \begin{bmatrix} f.h_x & f.h_y & f.h_z & -(R_x)' \\ f.v_x & f.v_y & f.v_z & -(R_y)' \\ a_x & a_y & a_z & -(R_z)' \end{bmatrix} \cdot \begin{bmatrix} S_x \\ S_y \\ S_z \\ 1 \end{bmatrix}. \quad (2.16)$$

Definindo ainda:

$$\begin{bmatrix} C_x' \\ C_y' \\ C_z' \end{bmatrix} = \begin{bmatrix} f.h_x & f.h_y & f.h_z \\ f.v_x & f.v_y & f.v_z \\ a_x & a_y & a_z \end{bmatrix} \cdot \begin{bmatrix} S_x \\ S_y \\ S_z \end{bmatrix}. \quad (2.17)$$

Sendo um ponto \mathbf{m} um vetor de coordenadas $(U, V, S)^T$ e \mathbf{S} com $(S_x, S_y, S_z, 1)^T$ pode-se reescrever a equação (2.16) como

$$\mathbf{m} = \mathbf{P} \cdot \mathbf{w}. \quad (2.18)$$

A equação (2.16), ou sua forma em (2.18), é conhecida como **Equação de Projeção Perspectiva**. Por facilidade em expressar cada elemento da matriz \mathbf{P} pode-se exprimi-la como

$$\mathbf{P} = \begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} & a_{1,4} \\ a_{2,1} & a_{2,2} & a_{2,3} & a_{2,4} \\ a_{3,1} & a_{3,2} & a_{3,3} & a_{3,4} \end{bmatrix}. \quad (2.19)$$

Fatorando-se a MPP em duas matrizes definidas a partir dos parâmetros de câmera pode-se chegar a

$$\mathbf{P} = \underbrace{\begin{pmatrix} \frac{f}{\delta u} & 0 & j_0 \\ 0 & \frac{f}{\delta v} & i_0 \\ 0 & 0 & 1 \end{pmatrix}}_{PI} \cdot \underbrace{\begin{pmatrix} \mathbf{h}^T & | & -\mathbf{C} \cdot \mathbf{h} \\ \mathbf{v}^T & | & -\mathbf{C} \cdot \mathbf{v} \\ \mathbf{a}^T & | & -\mathbf{C} \cdot \mathbf{a} \end{pmatrix}}_{PE}, \quad (2.20)$$

sendo PI os parâmetros intrínsecos da câmera e PE os parâmetros extrínsecos.

No segundo membro de (2.20) encontram-se os parâmetros intrínsecos: $\frac{f}{\delta u}$, $\frac{f}{\delta v}$, i_0 , j_0 e os parâmetros extrínsecos: \mathbf{h} , \mathbf{v} , \mathbf{a} . Os primeiros, como o próprio nome já sugere, representam as características internas da câmera, como distância focal, ponto principal e tamanho dos pixels. Já os últimos referem-se à posição e à orientação da câmera em relação a um sistema de coordenadas externo.

2.3 SEGMENTAÇÃO DE IMAGEM

A aplicação prática dos métodos de PDI requerem a utilização de imagens que retratem um ambiente real, e isto implica tratar imagens com vários planos e vários objetos. Assim, a separação dos planos e a seleção do objeto de estudo normalmente constitui uma das primeiras etapas do processamento de imagem. Para que tal intento possa ser atingido, o processo de segmentação de imagem subdivide uma imagem em suas regiões, ou objetos constituintes, sendo que a segmentação de imagens não-triviais é uma das mais difíceis tarefas em processamento de imagens [12].

A seleção de objetos presentes em uma imagem requer inicialmente uma separação de planos na mesma. Esta separação pode ser realizada por um processo de *Matching Digital*, que segundo [13] consiste em extrair o *foreground* de uma imagem a partir de estimativas de opacidade de cada pixel do objeto.

Os algoritmos de *Matching* tipicamente assumem que cada pixel I_i , em uma imagem de entrada, é uma combinação linear do *foreground* F_i e do *background* B_i [13]. Assim pode-se definir matematicamente a chamada **equação da composição** de uma imagem, definida como

$$F_i = \alpha_i \cdot B_i + (1 - \alpha_i) \cdot F_i, \quad (2.21)$$

em que α_i é a opacidade do i -ésimo pixel do *foreground*.

Uma imagem colorida pode ser representada por 3 canais (RGB). Assim, para cada pixel há 3 equações e 7 incógnitas, resultando em um problema complexo que requer uma intervenção do usuário para a obtenção de um bom *matte*.

A intervenção esperada pelo usuário é a segmentação grosseira da imagem em 3 níveis: *foreground* (mostrado em branco em 2.6 (c)), *background* (mostrado em preto em 2.6 (c)) e indefinida. Tal mapeamento é conhecido como *trimap*. Uma imagem de exemplo e sua segmentação podem ser visualizadas na Figura 2.6:

Uma vez conhecido o *trimap*, pode-se solucionar a equação (2.21) para F , B e α

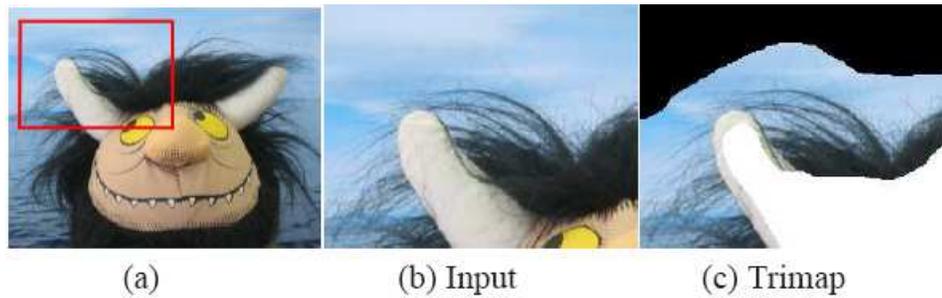


Figura 2.6: (a) Imagem Original. (b) Seleção da Região de Interesse. (c) Trimap [13].

simultaneamente, alternando a estimação de F e B com a de α . Na prática, isso significa que para a obtenção de bons resultados, a região desconhecida deve ser a menor possível, fazendo com que os métodos baseados em *trimaps* tenham dificuldade em lidar com significativa mistura de pixels ou com o objeto tenha muitos buracos [14].

A imagem obtida pela aplicação do processo de *Matching* à imagem da Figura 2.6 (b) pode ser visualizada na figura 2.7.



Figura 2.7: Separação de planos obtida a partir da imagem exibida na figura 2.6 [13].

2.4 VISÃO ESTÉREO

Em áreas de conhecimento como Fotogrametria e Visão Computacional, o objetivo central é a obtenção de informações de natureza qualitativa e quantitativa dos objetos [15]. A visão estéreo é um processo no qual a aquisição de imagens é feita por 2 (ou mais) imagens de um mesmo cenário. Assim, pode-se estimar a profundidade em que se encontra o objeto em relação à câmera. A análise de imagens estéreo surgiu como um importante método passivo para extração da estrutura tridimensional de uma determinada cena [16].

A diferença de posição do objeto nas imagens gera a chamada disparidade e a partir dela se estima a profundidade do objeto (coordenada z do objeto).

2.4.1 Calibração das Câmeras

Para que uma câmera possa ter suas imagens utilizadas analiticamente, faz-se necessário o levantamento de seus parâmetros. O processo responsável por mensurar tais parâmetros é

chamado **Calibração** e consiste na aquisição de um conjunto de imagens de um objeto de dimensões e formas conhecidas e a partir destas imagens pode-se estimar os parâmetros da câmera.

Os parâmetros de uma câmera são divididos em: **parâmetros intrínsecos** e **parâmetros extrínsecos**. Segundo [17], os parâmetros extrínsecos definem as operações de rotação (\mathfrak{R}) e translação (\mathfrak{T}) que relacionam o sistema de coordenadas da câmera (SCC) com o sistema de coordenadas externo (SCE). Já os parâmetros intrínsecos determinam o mapeamento de cada pixel como uma projeção dos raios de luz no sistema de coordenadas da câmera. Os parâmetros intrínsecos, para os modelos mais simples de câmera, são definidos como sendo a distância focal (f), a posição em pixels do centro da imagem (i_o, j_o) e os fatores de escala (S_u, S_v). Já os parâmetros extrínsecos podem ser definidos matematicamente como

$$\begin{pmatrix} x_c \\ y_c \\ z_c \end{pmatrix} = \mathfrak{R} \cdot \begin{pmatrix} x \\ y \\ z \end{pmatrix} + \mathfrak{T}, \quad (2.22)$$

sendo \mathfrak{R} uma matriz 3x3 que determina a orientação da câmera no espaço tridimensional e \mathfrak{T} a matriz de translação que define a posição da câmera em relação ao sistema de coordenadas globais.

Para que os parâmetros de uma câmera sejam levantados são utilizados padrões de calibração. Estes são normalmente tabuleiros quadriculados com dimensões conhecidas e são apresentados várias vezes à câmera para que, a partir de um sistema linear, possa-se chegar à medida dos parâmetros.

2.4.2 Retificação

A Visão Estéreo utiliza a disparidade (paralaxe) entre dois pontos correspondentes situados em imagens distintas como fonte de informação tridimensional. Tais pontos são denominados **homólogos**, assim como a linha que os une é chamada **epipolar** [18].

O processo de Retificação consiste na aplicação de transformações lineares nas imagens do par estéreo, visando que os mesmos pontos de uma imagem (pontos homólogos) se encontrem nas mesmas linhas no par estéreo de imagens. Assim, o processo de busca dos pontos homólogos (conhecido como correspondência ou *Matching*) tem seu espaço de busca reduzido a uma dimensão. Segundo Nogueira, 2007, [11], a retificação é definida como uma reamostragem do par de imagens estereoscópicas de forma que as linhas epipolares se tornem coincidentes com as linhas das imagens.

Seja um par estéreo constituído por uma imagem esquerda ImL , com coordenadas (i_L, j_L) , e por uma imagem direita ImR , com coordenadas (i_R, j_R) . Quando as imagens são adquiri-

das um mesmo ponto pode se encontrar em diferentes coordenadas (i,j) na imagem esquerda e na imagem direita. Já quando as imagens são submetidas ao processo de retificação, as coordenadas de linha passam a ser as mesmas em ambas as imagens ($i_L = i_R$), ou seja, o espaço de busca para que os pontos homólogos sejam encontrados se restringe apenas às coordenadas de coluna (coordenada j).

A reconstituição completa da cena filmada é portanto efetuada determinando-se, para cada pixel de uma imagem, aqueles que nas demais sejam representantes da mesma região da cena, seus pontos homólogos [19].

Para que uma imagem possa ser retificada, as matrizes de projeção perspectiva de cada uma das câmeras, denominadas P_L e P_R , precisam sofrer algumas transformações lineares, tornando-se P_{nL} , composta por elementos a_{mn} , e P_{nR} , composta por $b_{m,n}$. Tais transformações visam atender aos seguintes requisitos:

- Os centros ópticos c_{nL} e c_{nR} das câmeras expressas por P_{nL} e por P_{nR} devem coincidir com os centros c_L e c_R das câmeras P_L e P_R ;
- Pontos homólogos nas imagens imL e imR devem estar sempre na mesma linha, ou seja, as linhas epipolares devem ser paralelas a um dos eixos das imagens;
- Os planos projetados em $i = 0$ e em $j = 0$ nas duas imagens retificadas devem ser perpendiculares entre si, representando o princípio da ortogonalidade dos sistemas de referência;
- As resoluções horizontal e vertical das câmeras representadas por P_L e P_R , devem ser mantidas na nova configuração definida por P_{nL} e P_{nR} .

2.4.3 Correspondência

A correspondência, ou *Matching*, consiste em fazer a associação da projeção de um objeto em uma dada imagem com a projeção do mesmo objeto sobre outra imagem [15], ou seja, consiste em identificar quais *pixels* da imagem esquerda imL correspondem a quais *pixels* da imagem direita imR . Os pontos que não puderem ter sua disparidade determinada são chamados pontos de **oclusão**.

O cálculo da geometria epipolar significa não só obter alguma informação da posição relativa entre duas vistas sobre a mesma cena, como também eliminar erros de emparelhamento efetuado previamente [20].

As dificuldades do processo de *Matching* podem ser sintetizadas pelo fato de que os pontos correspondentes, em um par de imagens estéreo, podem: existir ou não, devido a problemas de oclusão, ou ainda serem ambíguos, devido a padrões repetitivos e a objetos transparentes [15].

O processo de correspondência pode ser realizado baseado em funções de **correlação** entre pequenas áreas de cada uma das imagens, por **feições**, que são pontos notáveis nas imagens, ou ainda pela combinação de ambos. Os dois próximos itens tratam de cada uma destas abordagens com mais detalhes.

2.4.4 Correspondência por Feições

O processo de correspondência entre pixels das imagens de um par estéreo pode ser realizado por feições, ou seja, por atributos dos objetos segmentados que sejam comuns às duas imagens. Tais feições também são conhecidas como *tokens*. A segmentação da imagem e o estabelecimento dos *tokens* são mostrados nas Figuras de exemplo 2.8 e 2.9:

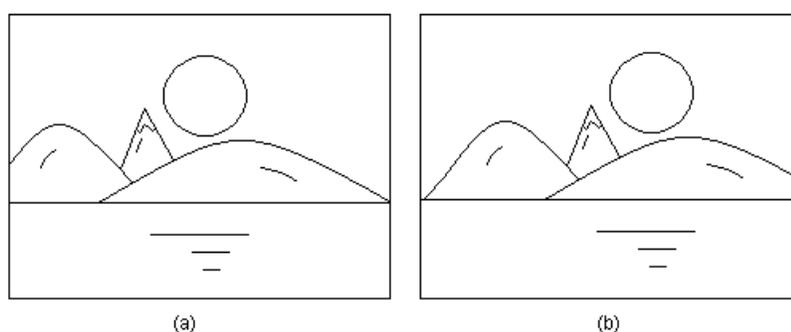


Figura 2.8: Exemplo de par estéreo: (a) Imagem Esquerda. (b) Imagem Direita.

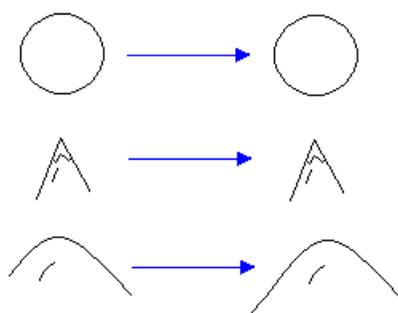


Figura 2.9: Associação de feições da imagem esquerda com feições da imagem direita, relativa à figura 2.8.

2.4.5 Correspondência Baseada em Áreas

O processo de correspondência baseado em áreas utiliza janelas de observação (pequenas áreas em cada uma das imagens do par estéreo) para comparação das imagens como um todo. Para se determinar o quão semelhantes são as janelas recorre-se a algumas métricas, dentre as quais as mais utilizadas são: A Soma das Diferenças Absolutas, **SDA**, mostrada na equação (2.23), e a Correlação Cruzada, **CC**, mostrada na equação (2.24). Para o processo de busca

da máxima correlação entre as áreas definem-se: uma **janela de referência**, localizada na imagem esquerda, imE , e **janelas de pesquisa**, localizada na imagem direita, imD . Define-se ainda um Ciclo de Varredura de Correlação, CVC, como sendo o intervalo no qual se procura, entre as janelas de pesquisa, a máxima correlação em relação a uma dada janela de referência.

Na Figura 2.10 podem ser visualizadas as janelas de referência e pesquisa no par de imagens estéreo. Pode-se observar que a janela de referência é fixa ao longo de um CVC, em sua posição na imagem imE , e a janela de pesquisa percorre toda a imagem imD . Ao final de um CVC a janela de pesquisa que obtiver a maior correlação, em relação à janela de referência, será associada a mesma. O processo se repete até que todas as janelas de referência tenham sido associadas a uma janela de pesquisa.

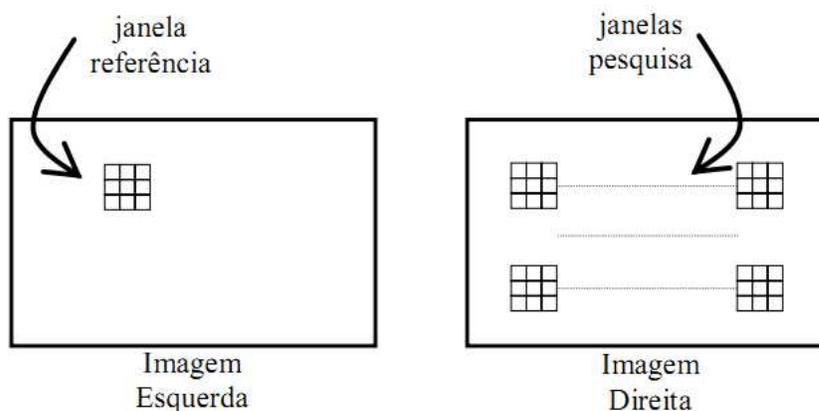


Figura 2.10: Janelas de correlação.

Cabe salientar que, num par de imagens estéreo retificado, o deslocamento das janelas de pesquisa pode acontecer apenas ao longo de uma linha horizontal, dentro de um CVC, uma vez que os pontos homólogos já se encontram nas mesmas linhas em ambas as imagens.

A correspondência constitui uma das etapas computacionalmente mais custosas. Assim, quando o método de correspondência se baseia em áreas, a função de correlação deve apresentar o menor custo computacional possível. Isto sem deixar de mensurar o quão parecidas são as imagens. A partir destas métricas pode-se estabelecer, ou não, uma correspondência entre os pixels que constituem as janelas em análise. Neste cenário busca-se otimizar a relação entre os custos computacionais e a precisão da medida de correlação.

Definindo-se $g_E(i, j)$ e $g_D(i, j)$ como os níveis de cinza dos pixels das janelas: esquerda (referência em imE) e direita (deslocamento em imD), respectivamente, e (a, b) como coordenadas de referência na janela pesquisa pode-se chegar às equações

$$SDA(a, b) = \sum_{i=0}^{f_y-1} \sum_{j=0}^{f_x-1} |g_E(i, j) - g_D(i + a, j + b)|, \quad (2.23)$$

e

$$CC(a, b) = \frac{\sum_{i=0}^{f_y-1} \sum_{j=0}^{f_x-1} [g_E(i, j) - \overline{g_E}] \cdot [g_D(i + a, j + b) - \overline{g_{D(a,b)}}]}{\sqrt{\left(\sum_{i=0}^{f_y-1} \sum_{j=0}^{f_x-1} [g_E(i, j) - \overline{g_E}]^2\right) \cdot \left(\sum_{i=0}^{f_y-1} \sum_{j=0}^{f_x-1} [g_D(i + a, j + b) - \overline{g_{D(a,b)}}]^2\right)}}, \quad (2.24)$$

sendo SDA a soma das diferenças absolutas e CC a correlação cruzada entre as janelas.

Uma vez definidas as correspondências entre os pixels das imagens imL e imR , pode-se calcular a diferença entre suas coordenadas, ou seja, a disparidade. As disparidades podem ser expressas como

$$d_h = x_E - x_D, \quad (2.25)$$

e

$$d_v = y_E - y_D, \quad (2.26)$$

onde: d_h é a disparidade horizontal (eixo x) e d_v é a disparidade vertical (eixo y).

A partir das disparidades pode-se inferir a profundidade do objeto focalizado pelas câmeras. Este processo baseia-se no fato de que a profundidade do objeto é inversamente proporcional à disparidade. Para imagens retificadas a disparidade vertical, d_v é nula, assim pode-se tratar unidimensionalmente a mesma. A profundidade é então dada por

$$z = \frac{b \cdot f}{d_h}, \quad (2.27)$$

onde: z é a profundidade de um ponto pertencente à cena focalizada pelas câmeras, b é a distância entre os centros ópticos das câmeras e f é a distância focal.

2.4.6 Triangulação e Reconstrução 3D

A Reconstrução 3D consiste basicamente no processo inverso ao da aquisição de imagem. Nesta, uma câmera gera uma imagem a partir de um determinado objeto. Naquela,

geram-se as coordenadas dos pontos que constituem o objeto a partir de um par de imagens do mesmo.

Para que as imagens tridimensionais de um dado objeto possam ser obtidas, as matrizes de projeção perspectiva e a associação de pixels entre as imagens bidimensionais, precisam ser conhecidas. Este processo, conhecido como **triangulação** permite chegar à profundidade do objeto em questão, pois o princípio básico da recuperação da profundidade utilizando imagens passivas é a triangulação [16].

Um triângulo para recuperação da profundidade é definido por 3 pontos: a fonte de luz, o ponto iluminado e seu ponto imagem. Assim, nas técnicas que utilizam a triangulação, o problema da correspondência é solucionado usando uma fonte de luz externa [16]. Desta forma, nas técnicas de imageamento ativo tem-se o problema da correspondência já resolvido, enquanto que as técnicas de imageamento passivo necessitam de mais de uma imagem do objeto para a determinação da profundidade do mesmo.

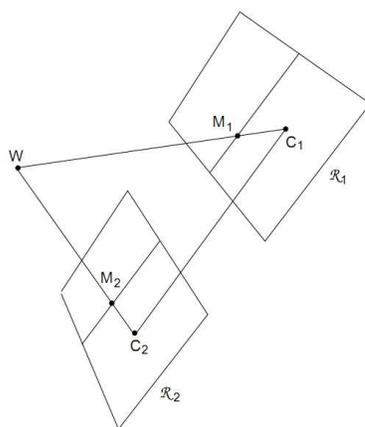


Figura 2.11: Triangulação.

A estereoscopia é um técnica passiva de triangulação que busca seu objetivo utilizando apenas a iluminação ambiente. Assim, a correspondência entre as imagens de um objeto precisa ser estabelecida para que se possa estabelecer a relação das imagens com o objeto físico no espaço [16].

Tanto o modelo tridimensional de um objeto pode ser levantado a partir de um par de imagens bidimensionais do mesmo, como a partir do modelo tridimensional pode-se escolher a planificação que seja mais adequada ao seu uso. Segundo [15] tais espaços podem ser designados por **espaço imagem** e **espaço objeto**, sendo o primeiro relativo às planificações da imagem do objeto e o último relativo ao objeto tridimensional em si. Tais espaços podem ser observados na figura 2.12.

A transformação T_{23} baseia-se essencialmente no estabelecimento da disparidade entre as imagens e no conhecimento das MPP das câmeras que geraram as mesmas.

Uma vez conhecida a disparidade pode-se reconstituir os contornos do objeto original a partir de duas novas matrizes de projeção perspectivas: \tilde{P}_{n1} e \hat{P}_{n2} . Estas precisam preservar

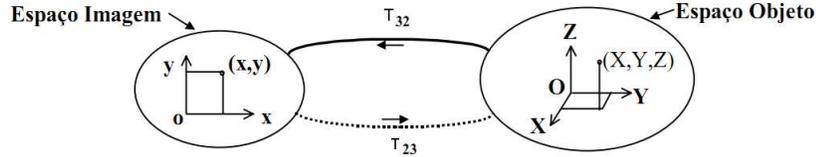


Figura 2.12: Espaço Imagem e Espaço Objeto.

os seus centros ópticos e necessitam que a linha que liga os centros óticos das câmeras (linha de base) esteja contida no plano focal. Assim as linhas epipolares serão paralelas [21].

As novas MPP devem possuir a mesma orientação, mas posições diferentes [21]. As posições (centros óticos) são as mesmas das câmeras anteriores, enquanto a orientação é modificada em ambas as câmeras devido à rotação, em relação ao centro ótico, de forma que os planos focais se tornem coplanares e que contenham a linha de base.

Seja W o vetor posição relativo ao ponto P e $(u', v')^T$ e sejam $(u'', v'')^T$ as coordenadas das imagens deste ponto em um par de imagens estéreo. Assim pode-se escrever W da seguinte forma [21]

$$W = A^{-1} \cdot y, \quad (2.28)$$

sendo:

$$A = \begin{bmatrix} (a_1 - u' \cdot a_3)^T \\ (a_2 - v'' \cdot a_3)^T \\ (b_1 - u' \cdot b_3)^T \\ (b_2 - v'' \cdot b_3)^T \end{bmatrix} \quad y = \begin{bmatrix} (-a_14 + u' \cdot a_34)^T \\ (-a_24 + v'' \cdot a_34)^T \\ (-b_14 + u'' \cdot b_34)^T \\ (-b_24 + v'' \cdot b_34)^T \end{bmatrix} \quad (2.29)$$

Aplicam-se as condições necessárias às matrizes de retificação (citadas na seção 2.4.2) e obtêm-se

$$P_{n1} = \left(\begin{array}{c|c} \mathbf{b}_1^T & b_{14}^T \\ \mathbf{b}_2^T & b_{24}^T \\ \mathbf{b}_3^T & b_{34}^T \end{array} \right) \quad P_{n2} = \left(\begin{array}{c|c} \mathbf{a}_1^T & a_{14}^T \\ \mathbf{a}_2^T & a_{24}^T \\ \mathbf{a}_3^T & a_{34}^T \end{array} \right), \quad (2.30)$$

onde \mathbf{a} e \mathbf{b} são vetores que compõem o sistema de orientação de cada uma das câmeras, SOC, (conforme Figura 2.4).

Com isso chega-se às matrizes que, em conjunto com a disparidade, podem gerar as coordenadas tridimensionais de um objeto.

2.5 CADEIAS DIRECIONAIS

Nas áreas de Processamento Digital de Imagem (PDI) muitas vezes se tem a situação em que apenas o contorno de um objeto já é suficiente para sua análise. Neste caso são utilizadas técnicas de codificação de contorno. Nestas abordagens o contorno do objeto de interesse é expresso por **Cadeias Direcionais Codificadas**. Assim, passa-se a utilizar tais cadeias como descritores da imagem em vez da imagem em si. Desta forma tem-se uma redução do "esforço" computacional pelo não processamento de todos os pixels da imagem mas apenas de sua cadeia de contornos direcionais codificada.

Na codificação do contorno de imagens, uma das técnicas mais utilizadas é a codificação em alfabeto S , no qual são utilizadas as seguintes letras $[0, 1, 2, 3, 4, 5, 6, 7]$, sendo que cada letra indica uma direção. Tal alfabeto pode ser observado na Figura 2.13.

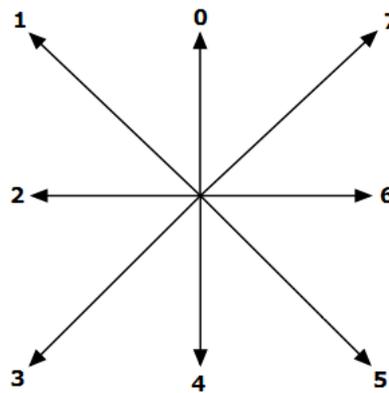


Figura 2.13: Cadeia Direcional.

Define-se uma cadeia direcional $c = a_1, a_2, a_3, \dots, a_k$ como um vetor de comprimento k em S^* (conjunto de todas as cadeias possíveis), cujos elementos correspondem a caracteres de S organizados de modo que seus respectivos segmentos orientados estejam conectados de forma a reproduzir os contornos de uma imagem [22].

Num exemplo didático pode-se codificar a imagem da Figura 2.14, no alfabeto S , encontrando-se a seguinte Cadeia Direcional como descritor de contorno:

$$[6, 6, 4, 6, 6, 6, 0, 6, 6, 4, 4, 4, 2, 2, 0, 2, 2, 2, 4, 2, 2, 0, 0, 0]$$

Nesta modelagem pode-se concluir, a partir das Equações 2.31, que vetores representados no alfabeto S , de ordem par, possuem módulo 1 enquanto os representados por letras de ordem ímpar possuem módulo igual a $\sqrt{2}$.

Abordando-se cada um dos vetores que são expressos pelo alfabeto S pode-se chegar à expressão genérica que calcula o comprimento de cada vetor

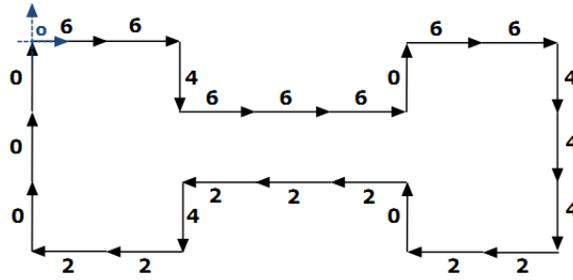


Figura 2.14: Exemplo de uma imagem codificada em Cadeia de Freeman.

$$\Delta t_i = \left[1 + \frac{\sqrt{2} - 1}{2} \cdot (1 - (-1)^{a_i})\right]. \quad (2.31)$$

Já abordando-se todo o contorno da imagem codificada pose-se chegar ao comprimento total da cadeia codificada, expresso por

$$T = \sum_{i=1}^k \left[1 + \frac{\sqrt{2} - 1}{2} \cdot (1 - (-1)^{a_i})\right]. \quad (2.32)$$

Pode-se ainda equacionar o comprimento dos m primeiros segmentos a partir da equação

$$t_m = \sum_{i=1}^m \left[1 + \frac{\sqrt{2} - 1}{2} \cdot (1 - (-1)^{a_i})\right]. \quad (2.33)$$

Tabela 2.1: Projeções para cada símbolo do alfabeto direcional.

Alfabeto	Projeção no eixo x	Projeção no eixo y
0	0	1
1	-1	1
2	-1	0
3	-1	-1
4	0	-1
5	1	-1
6	1	0
7	1	1

Conhecidos os segmentos que modelam o contorno de uma imagem, pode-se decompor cada um dos segmentos em suas componentes nos eixos x e y . Tais projeções podem ser observadas na Tabela 2.1 e as expressões que definem tais projeções estão em (2.34) e (2.35).

$$\Delta x_i = \text{sgn}(a_i - 4) \cdot \text{sgn}(a_i) \quad (2.34)$$

e

$$\Delta y_i = \text{sgn}(6 - a_i) \cdot \text{sgn}(2 - a_i). \quad (2.35)$$

O vetor que codifica a imagem da figura 2.14 têm suas projeções representadas na figura 2.15.

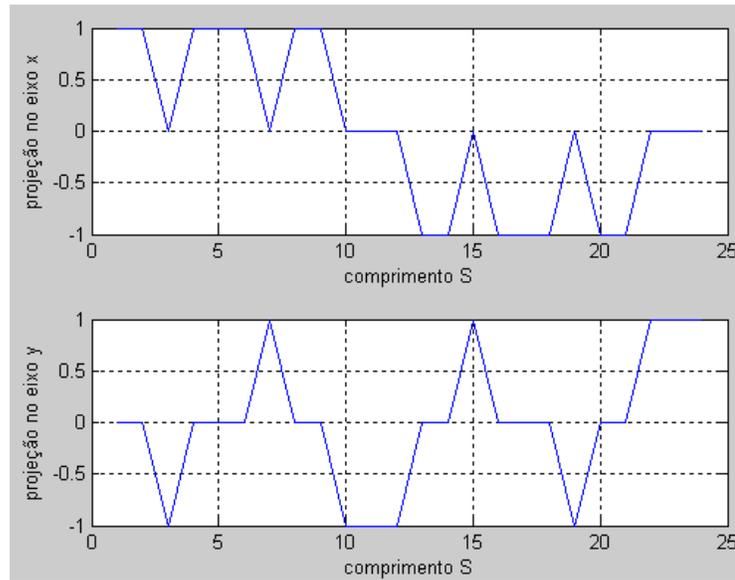


Figura 2.15: Exemplo de projeção nos eixos x e y da imagem codificada da figura 2.14.

De maneira mais genérica, ainda pode-se expressar o comprimento de cada um dos símbolos a partir da tabela 2.2:

Tabela 2.2: Comprimento dos segmentos.

Segmento	Comprimento
0	Δy
1	$\sqrt{\Delta x^2 + \Delta y^2}$
2	Δx
3	$\sqrt{\Delta x^2 + \Delta y^2}$
4	Δy
5	$\sqrt{\Delta x^2 + \Delta y^2}$
6	Δx
7	$\sqrt{\Delta x^2 + \Delta y^2}$

A representação das projeções, horizontal (eixo x) e vertical (eixo y), em séries de Fourier também é um método de compressão de dados, pois utilizando um número finito de coeficientes é possível reconstruir o contorno da imagem original, ou então utilizar tais coeficientes como vetores de características utilizadas para estabelecer padrões de imagens [22].

2.6 SÍNTESE EM SÉRIES DE FOURIER

As séries de Fourier decompõem uma função ou um sinal periódico em somatórios de funções harmônicas ímpares ($\sin x$), funções harmônicas pares ($\cos x$) e uma função constante

$$f(x) = a_0 + \sum_{n=1}^{\infty} [a_n \cdot \cos\left(\frac{2 \cdot \pi \cdot n}{T} \cdot x\right) + b_n \cdot \sin\left(\frac{2 \cdot \pi \cdot n}{T} \cdot x\right)], \quad (2.36)$$

sendo os coeficientes da série de Fourier unidimensional expressos pelas equações (2.37), (2.38) e (2.39)

$$a_0 = \frac{1}{T} \cdot \int_0^T f(\lambda) \cdot d\lambda, \quad (2.37)$$

$$a_n = \frac{2}{T} \cdot \int_0^T f(\lambda) \cdot \cos\left(\frac{2 \cdot \pi \cdot n}{T} \lambda\right) \cdot d\lambda, \quad (2.38)$$

e

$$b_n = \frac{2}{T} \cdot \int_0^T f(\lambda) \cdot \sin\left(\frac{2 \cdot \pi \cdot n}{T} \cdot \lambda\right) \cdot d\lambda. \quad (2.39)$$

Tal abordagem também pode ser aplicada a funções bidimensionais. Especificamente, quando se tratam contornos de imagens, as Séries de Fourier são aplicadas a cada uma das projeções (eixo x e y) da imagem.

$$f(x) = A_0 + \sum_{n=1}^{\infty} [A_n \cdot \cos\left(\frac{2 \cdot \pi \cdot n}{T} \cdot x\right) + B_n \cdot \sin\left(\frac{2 \cdot \pi \cdot n}{T} \cdot x\right)], \quad (2.40)$$

e

$$f(y) = C_0 + \sum_{n=1}^{\infty} [C_n \cdot \cos\left(\frac{2 \cdot \pi \cdot n}{T} \cdot y\right) + D_n \cdot \sin\left(\frac{2 \cdot \pi \cdot n}{T} \cdot y\right)], \quad (2.41)$$

onde os termos A_0 , equação (2.42), e C_0 , equação (2.43), representam o *offset* (deslocamento) de posição do contorno da imagem, enquanto A_n equação (2.44), equação B_n (2.45),

C_n equação (2.46) e D_n equação (2.47) são os coeficientes das harmônicas. Na determinação de tais coeficientes o método de integração utilizado foi o Trapezoidal pela sua simplicidade de implementação e porque os erros de integração inerentes a tal método se mostraram perfeitamente aceitáveis para a aplicação deste trabalho.

$$A_0 = \frac{1}{T} \cdot \int_0^T f(x) \cdot dx, \quad (2.42)$$

$$C_0 = \frac{1}{T} \cdot \int_0^T f(y) \cdot dy, \quad (2.43)$$

$$A_n = \frac{2}{T} \int_0^T f(\lambda) \cdot \cos\left(\frac{2 \cdot \pi \cdot n}{T} \cdot \lambda\right) \cdot d\lambda, \quad (2.44)$$

$$B_n = \frac{2}{T} \int_0^T f(\lambda) \cdot \sin\left(\frac{2 \cdot \pi \cdot n}{T} \cdot \lambda\right) \cdot d\lambda, \quad (2.45)$$

$$C_n = \frac{2}{T} \int_0^T f(\lambda) \cdot \cos\left(\frac{2 \cdot \pi \cdot n}{T} \cdot \lambda\right) \cdot d\lambda, \quad (2.46)$$

e

$$D_n = \frac{2}{T} \int_0^T f(\lambda) \cdot \sin\left(\frac{2 \cdot \pi \cdot n}{T} \cdot \lambda\right) \cdot d\lambda. \quad (2.47)$$

Uma vez estabelecido um vetor que codifique, em cadeias direcionais, o contorno da imagem alvo, a mesma pode ser decomposta em séries de Fourier. Para tanto, os vetores que modelam o contorno são decompostos em uma componente em relação ao eixo horizontal da imagem, e outra componente em relação ao eixo vertical. Uma vez feito tal procedimento pode-se gerar os coeficientes de Fourier para cada componente, ou seja, coeficientes A_0 , A_n , B_n para as componentes de direção horizontal (eixo x), e coeficientes C_0 , C_n , D_n para as componentes de direção vertical (eixo Y).

A decomposição em séries de Fourier é utilizada devido a capacidade de compressão de informação que se atinge com a decomposição. Assim, pode-se com apenas 2 escalares (A_0 e C_0) e 4 vetores de dimensão n (A_n , B_n , C_n , D_n) representar o conteúdo do contorno de um objeto da imagem.

Neste capítulo foram introduzidos os aspectos básicos sobre imagens digitais, modelamento de câmeras de Orifício Escuro, Visão Estéreo e modelagens de contornos de imagens por Séries de Fourier.

3 REDES NEURAIS

Os processos que regem a inteligência humana e os mecanismos que compõem a complexa ação do pensar sempre compuseram grande desafio à área das ciências biológicas. Uma vez descoberta a natureza elétrica destes processos, os desafios passaram a se distribuir também entre as áreas das ciências exatas. Levantamentos sobre possíveis modelos matemáticos que pudessem descrever os sinais elétricos e seus processos no cérebro humano culminaram no desenvolvimento de várias áreas, entre elas as Redes Neurais Artificiais (RNA). Os primeiros autores a tratarem estas redes foram: o médico Neurologista, Warren Mc Culloch, do MIT, e o Estatístico Walter Pitts, da universidade de Illinois. A inspiração principal na modelagem matemática de tais componentes foi o neurônio biológico.

A estrutura esquemática de um neurônio biológico é descrita pela figura 3.1.

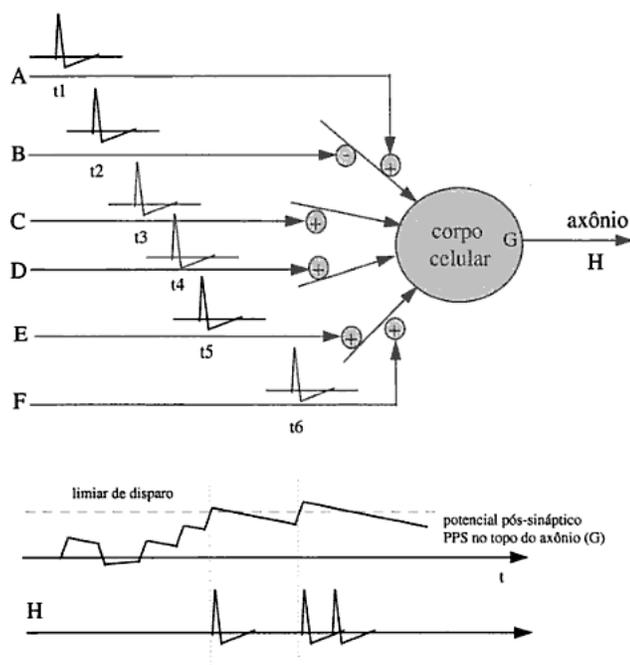


Figura 3.1: Modelo de um neurônio biológico [23].

A célula básica que compõem a estrutura que permite ao ser humano pensar, o neurônio biológico, teve várias modelagens matemáticas ao longo do tempo. Uma das primeiras abordagens, desenvolvida por McCulloch e Pitts, na década 40, entendia o neurônio como um sistema composto por entradas e saídas binárias. Um conjunto de vias de entrada recebiam sinais elétricos binários, e se o somatório destes sinais fosse capaz de romper um determinado limiar (conhecido como Função de Ativação) um sinal binário de seria disparado. A analogia de tal modelagem matemática com a Álgebra Booleana é clara, e os desenvolvimentos de ambas as áreas são temporalmente muito próximos. Apesar da simplicidade deste primeiro modelo, a partir dele pôde-se desenvolver toda uma nova área de pesquisa, as Redes

Neurais Artificiais.

Um diagrama de blocos do neurônio Artificial de McCulloch Pitts é descrito na figura 3.2 e a equação que o descreve é definida como

$$y = g\left(\sum_{i=1}^n W_i \cdot x_i\right), \quad (3.1)$$

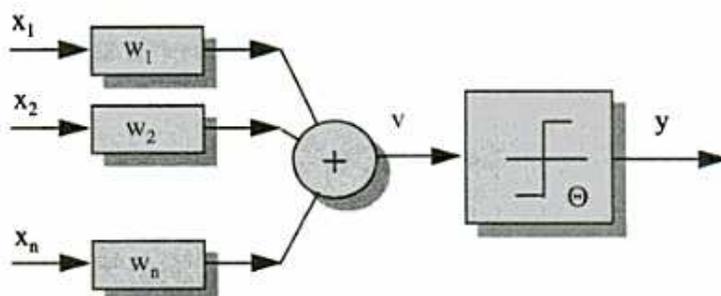


Figura 3.2: Neurônio Artificial [23].

sendo:

- Vetor de entrada: X ;
- Pesos de ponderação: W ;
- Somatório: responsável por acumular o somatório das entradas x_n ponderadas pelos pesos w_n ;
- Função de Ativação: função $sgn(x)$;

Tal estrutura mais tarde ficou conhecida como Neurônio de McCulloch e pode ser modelada como um discriminador linear cujas entradas são binárias[23].

As principais funções de ativação são expressas abaixo:

- Função Degrau: foi inicialmente utilizada pela sua fácil implementação e por ser capaz de expressar saturação do eixo das ordenadas. Esta função é dada por (3.2) e seu gráfico é ilustrado na figura 3.3 (a).

$$g(v) = \begin{cases} 1 & \text{se } v \geq 0, \\ 0 & \text{se } v < 0. \end{cases} \quad (3.2)$$

- Função Linear: apesar de não apresentar as regiões de saturação, esta função possui toda sua extensão regida pela proporcionalidade de 1º grau. Ela pode ser observada em (3.3) e seu gráfico pode ser observado na figura 3.3 (b).

$$g(v) = av + b. \quad (3.3)$$

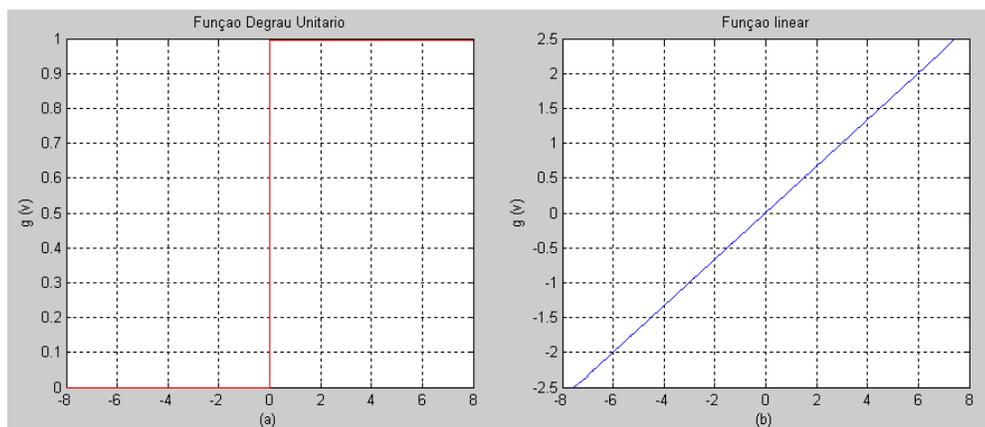


Figura 3.3: (a) Função Degrau e (b) Função Linear.

- Função Sigmóide: surge quase como uma fusão das funções Linear e Degrau, pois possui as características de linearidade, dentro de uma determinada faixa, e saturação nas faixas externas à linear. Cabe salientar que esta função apenas excursiona nos valores positivos do eixo das ordenadas. Esta função é dada por

$$g(v) = \frac{1}{1 + e^{-v}}, \quad (3.4)$$

e seu gráfico é ilustrado na figura 3.4.

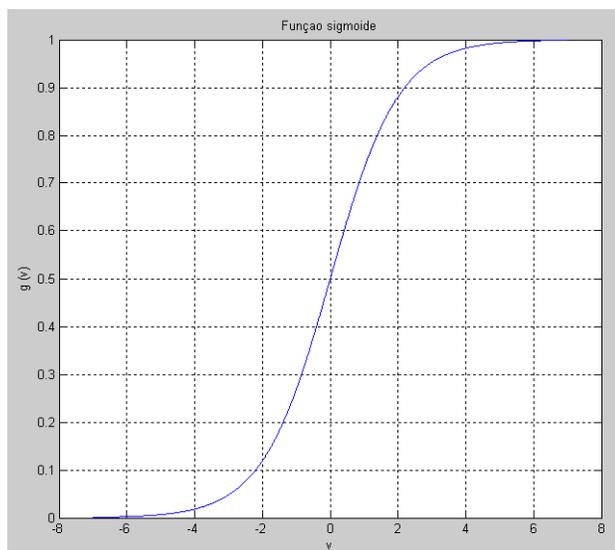


Figura 3.4: Função Sigmóide.

- Função Tangente Hiperbólica: uma das funções de ativação mais utilizadas, pois preserva todas as características da função Sigmóide com uma única diferença: A imagem desta função excursiona no intervalo $[-1, 1]$, atingindo assim, a região negativa do eixo das ordenadas. O ganho da região linear pode ainda ser ajustado pelo parâmetro a . Esta função pode é dada por

$$g(a.v) = \tanh(a.v) = \frac{1 - e^{-a.v}}{1 + e^{-a.v}}, \quad (3.5)$$

e seu gráfico é mostrado na figura 3.5.

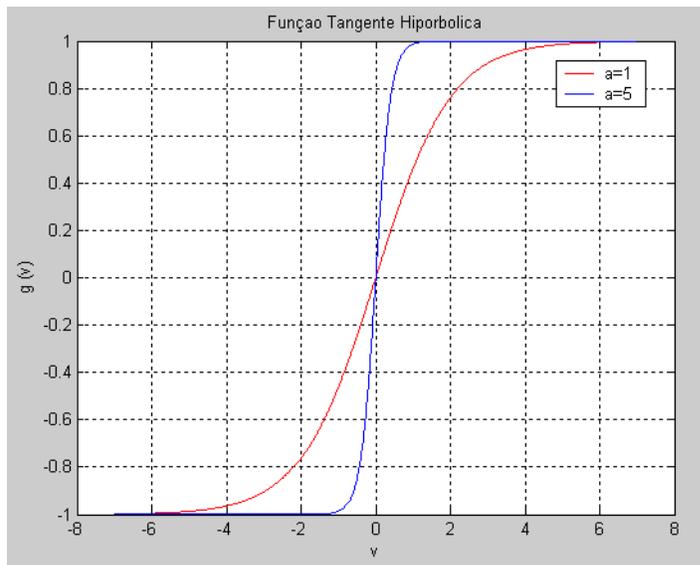


Figura 3.5: Função Tangente Hiperbólica.

3.1 REDES NEURAIS COMO CLASSIFICADORES

Classificar elementos em diferentes grupos é um processo relativamente simples para um ser humano, mas no âmbito da Inteligência Artificial tal processo pode requerer grandes esforços computacionais. Para a atividade de classificação de elementos genéricos em, por exemplo, 2 conjuntos, uma Rede Neural teria que ser antecipadamente treinada, para que os pesos sinápticos \mathbf{W} sejam ajustados para tal operação.

Um Neurônio Artificial tem a estrutura adequada para atuar como discriminador linear e um discriminador linear pode ser utilizado como classificador de padrões [23]. Assim, a saída de um neurônio classificador é levada a 0 quando apresenta-se, ao neurônio, uma entrada pertencente a um dado conjunto. A mesma saída é levada a 1 quando apresenta-se uma entrada pertencente a outro conjunto.

Sejam dois conjuntos A e B com seus elementos misturados, formando um conjunto C , ($C = A \cup B$) e apresentados a um neurônio classificador. Considera-se que os elementos

de C são linearmente separáveis nos conjuntos A e B . O processo de separação linear pode ser observado ilustrativamente na Figura 3.6. Isto só será possível se os conjuntos (coleções) formarem aglomerados no espaço R^n , tal que seja possível passar um hiperplano que separe estes dois aglomerados [23].

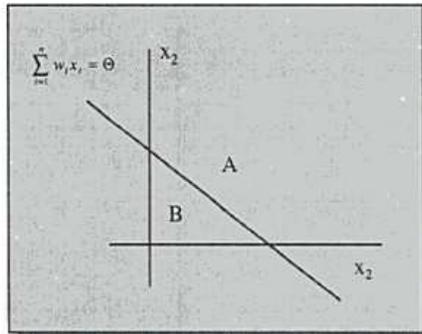


Figura 3.6: Discriminador Linear [23].

3.2 REDES NEURAI MULTICAMADAS

Dando prosseguimento às ideias de McCulloch, Frank Rosenblatt, da Universidade de Cornell, "fundiu" vários Neurônios Artificiais para compor o que viria a ser uma verdadeira Rede Neural. A estrutura idealizada por Rosenblatt compunha-se basicamente de:

- Camada de entrada: onde os padrões (ou seja, as entradas) são apresentadas à Rede;
- Camadas intermediárias ou ocultas: onde acontece a maior parte do processamento, onde os pesos de ponderação "adaptam" as entradas para a saída desejada.
- Camada de saída: onde o resultado do processamento é apresentado;

3.3 APRENDIZADO EM REDES NEURAI

Similarmente ao processo de aprendizado humano constatou-se que os neurônios artificiais só poderiam servir a algum propósito mais específico se um processo anterior à sua utilização fosse realizado: o treinamento. Uma Rede Neural aprende sobre seu ambiente através de um processo iterativo de ajuste de pesos sinápticos e dos limiares das funções de ativação, também chamadas de *bias* [9].

A interação entre a rede e o ambiente se dá pela apresentação de um conjunto de treinamento à rede. Assim, assume-se um conjunto ψ , com n elementos, no qual cada elemento é composto por uma entrada x_i e uma saída y_i . (para a rede neural, entende-se que a saída y_i é a resposta desejada para a entrada x_i). No processo de aprendizagem cada um dos elementos

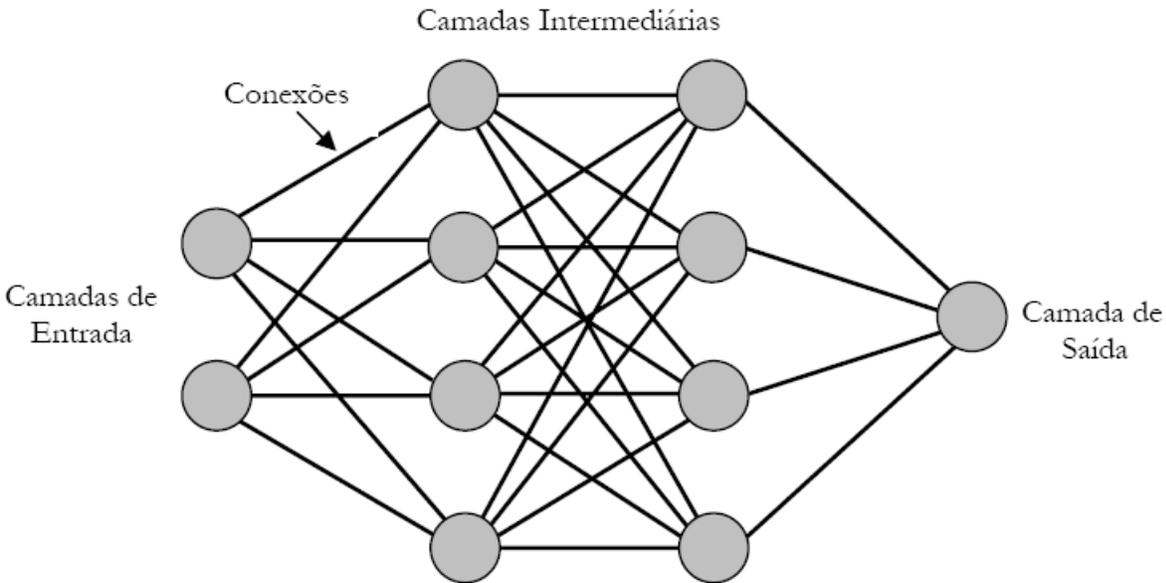


Figura 3.7: Ilustração de uma rede neurais multicamadas [24].

do conjunto ψ é apresentado à rede neural para que os seus pesos sinápticos e valores de *bias* possam ser ajustados de forma a aproximar a resposta da rede à saída exibida pelo conjunto de treinamento. Neste cenário, define-se como **ciclo de aprendizagem** a apresentação de todos os exemplos do conjunto de treinamento à rede. Idealmente, a rede se torna mais instruída sobre seu ambiente a cada iteração do processo de aprendizado [9]. Assim, vários ciclos são aplicados até que algum critério de parada do algoritmo de aprendizado seja atingido. Normalmente os pesos sinápticos \mathbf{W}_n e os limiares θ são ajustados até que se obtenha erro mínimo na saída da rede.

No processo de treinamento de uma rede busca-se minimizar o erro quadrático, descrito pela equação (3.6). O inconveniente é que a busca pela minimização do erro num curto intervalo de tempo pode levar os vetores \mathbf{W} e θ a variações bruscas e isto pode gerar a chamada **convergência prematura**, situação na qual se atinge um mínimo local e não se consegue sair deste. Um mínimo local de uma superfície, plotada a partir de uma função de erro quadrático, é mostrada na figura 3.8.

Para evitar a convergência prematura no processo de treinamento, controla-se a taxa de aprendizagem η , expressa na equação (3.11), e por meio desta controla-se a "velocidade" de atualização dos Pesos Sinápticos \mathbf{W} . Assim, com menor η tem-se uma menor variação nos pesos sinápticos e vice versa. Como um η muito baixo leva a uma resposta muito lenta no treinamento e um η muito elevado pode levar a convergência prematura deve-se buscar uma taxa de aprendizagem que otimize os quesitos de tempo e convergência.

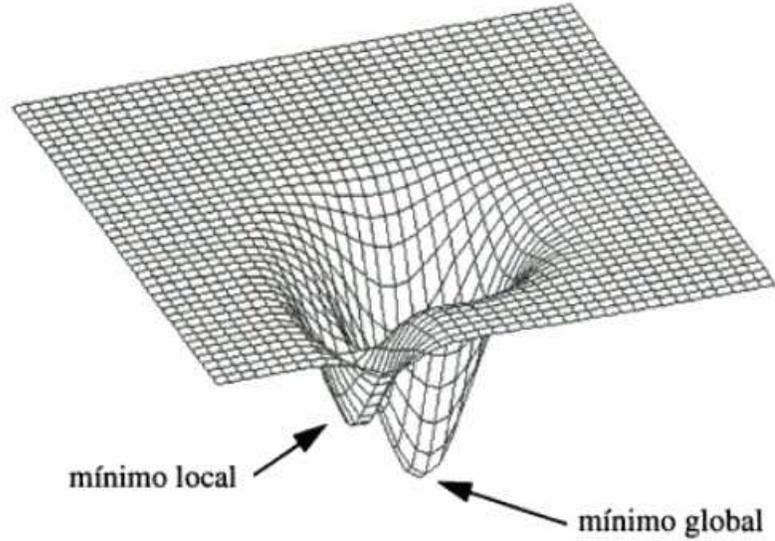


Figura 3.8: Ilustração de um mínimo local e global em uma superfície de erro [23].

3.4 ERRO QUADRÁTICO MÉDIO E A REGRA DELTA GENERALIZADA

No treinamento de uma rede neural buscam-se os pesos sinápticos \mathbf{W} e as *bias* θ que possibilitam que o erro quadrático, expresso pela equação (3.6), seja mínimo. Para tanto $\nabla E(w)$ é igualado a zero para que o valor de mínimo possa ser encontrado.

Neste cenário, define-se y_d como a saída desejada para uma dada entrada x , ambas pertencentes a um conjunto de treinamento ψ . Para se quantificar o erro quadrático (EQ) sobre uma determinada rede neural tem-se

$$E(w) = \sum_{i=1}^L [g(w^t \cdot x_i^d) - y_i^d]^2, \quad (3.6)$$

A partir do EQ definido em (3.6) pode-se equacionar o gradiente de tal função para que se possa buscar o vetor \mathbf{W} que minimize o erro quadrático.

$$\begin{aligned} \frac{\partial E(w)}{\partial w} &= 2 \cdot \sum_{i=1}^L [g(w^t \cdot x_i^d) - y_i^d] \cdot \frac{dg(w^t \cdot x_i^d)}{dw} \\ &= 2 \cdot \sum_{i=1}^L (g(w^t \cdot x_i^d) - y_i^d) \cdot \frac{dg(v)}{dv} \cdot \frac{d(w^t \cdot x_i^d)}{dw} \\ &= 2 \sum_{i=1}^L (y_i - y_i^d) \cdot \frac{dg(v)}{dv} \cdot x_i^d; \end{aligned}$$

$$\frac{\partial E(w)}{\partial w} = -2 \sum_{i=1}^L \delta_i \cdot x_i^d, \quad (3.7)$$

$$\text{onde } \delta_i = (y_i^d - y_i) \cdot \frac{dg(v)}{dv}. \quad (3.8)$$

Neste cenário a atualização da rede neural é feita segundo

$$w(k+1) = w(k) + \Delta w(k). \quad (3.9)$$

Com $\Delta w(k)$ seguindo a direção oposta a do gradiente tem-se

$$\Delta w(k) = -\eta \frac{dE(w)}{dw} = -\frac{\eta}{2} \frac{dE(w)}{dw}. \quad (3.10)$$

Assim chega-se a chamada *Regra Delta Generalizada*, dada por

$$\Delta w(k+1) = w(k) + \eta \cdot \sum_{i=1}^L \delta_i \cdot x_i^d. \quad (3.11)$$

Equações como a (3.11) podem ser utilizadas em várias metodologias de aprendizados de redes neurais.

3.5 BACKPROPAGATION

À medida que as Redes se tornam mais complexas os processos de treinamento se tornam igualmente mais complicados. O algoritmo de treinamento mais célebre no treinamento de redes multicamadas é o "**Backpropagation**" (Algoritmo de Retropropagação). Os passos a serem executados em tal algoritmo são exibidos abaixo:

Para redes com várias camadas, estabelecendo-se a seguinte notação:

- m representa o índice de camada;
- $j_m = 1, 2, 3, \dots, J_m$ é o índice do neurônio da m -ésima camada;
- $j_{(m-1)} = 1, 2, \dots, J_{m-1}$ é o índice da componente do vetor de entrada do neurônio.

Dada uma rede de M camadas com J_0, J_1, \dots, J_M elementos em cada camada. Assim, o Algoritmo de Backpropagation segundo [23] é expresso nos passos que seguem:

1. Escolhem-se os parâmetros iniciais w_0 e η .
2. Estabelece-se algum critério de parada para o algoritmo de treinamento.
3. Calculam-se os erros de saída, da camada de saída e das camadas subsequentes:

com $l = 1, 2, \dots, L$

$$\delta_{y(jM),l} = (y_{jM,l}^d - y_{jM,l}) \text{ e } \delta_{jM,l} = \delta_{y(jM),l} \cdot \frac{dg(v_{jM})}{dv_{jM}}$$

Variando-se $m = (M - 1) \dots, 1, 0$:

$$\delta_{jm,l} = \left[\sum_{j(m+1)=1}^{J(m+1)} \delta_{j(m+1),l} \cdot w_{m+1,j(m+1),jm} \right] \cdot \frac{dg(h_{jm})}{dh_{jm}}$$

4. Atualizam-se os parâmetros pela equação seguinte:

$$\Delta w_{m,,jm,,j(m-1)} = 2 \cdot \eta \sum_{l=1}^L \delta_{jm,l} \cdot u_{j(m-1),l}^d$$

Notando-se que: $u_{j(0),l}^d = x_{j(0),l}^d$

5. Recalcula-se: $w(1) = w(0) + \Delta(w)$ e repete-se o ciclo a partir do item 3 até que se alcance o critério de parada.

3.5.1 Métodos de Segunda Ordem

O algoritmo *backpropagation* é uma implementação baseada no método do gradiente, em que os vetores de parâmetros W e θ são ajustados na direção oposta do vetor gradiente. Tal método é considerado um método de primeira ordem pois faz uso da derivada primeira (gradiente) da função de custo para ajuste dos pesos da rede. Segundo Iyoda, 2000, [25], os métodos de primeira ordem são conhecidos pela baixa eficiência no tratamento de problemas de larga escala, pois apresentam taxas de convergência muito pobres, especialmente em regiões próximas a mínimos locais.

Nos métodos de segunda ordem, além do vetor gradiente da função de erro, faz-se também uso da matriz Hessiana (matriz de derivadas de segunda ordem). Com isso tem-se um alto custo computacional no processo de treinamento e consequentemente elevado tempo de execução. Para Jones *et al.*, 2005, [26], um dos algoritmos de segunda ordem mais rápidos no treinamento de RNAs é o algoritmo proposto por Levenberg Marquardt, que aproxima localmente a superfície de erro por uma função quadrática, e com isso simplifica o cálculo da matriz Hessiana utilizando apenas a matriz Jacobiana (matriz de derivadas de primeira ordem).

O método treinamento Levenberg Marquardt (*LM*) é um dos mais utilizados pelo seu custo benefício, uma vez que possui as vantagens dos métodos de segunda ordem e um custo computacional reduzido em relação aos mesmos.

3.6 CONCLUSÃO

Neste capítulo foram introduzidos os aspectos básicos sobre as Redes Neurais Artificiais, como sua conceituação, seus principais componentes e os principais métodos de treinamento de uma RNA.

4 IMPLEMENTAÇÃO

Neste capítulo são apresentados os algoritmos implementados bem como explicações mais detalhadas de pontos críticos ao desenvolvimento deste trabalho.

4.1 INTRODUÇÃO

Para emular a aquisição das imagens das linhas de transmissão (Figura 4.5), feitas pelo helicóptero (figura 1.4), as câmeras (visualizadas na figura 4.3) foram colocadas em diferentes posições em relação à garra metálica (figura 4.8). Foram adquiridos 7 conjuntos de imagens, cada um deles com 10 pares de imagens da garra em perfeito estado (compondo o grande conjunto das imagens *ok*). Também foram adquiridos outros 7 conjuntos de imagens das garras em um estado em que houvesse necessidade de alguma manutenção (compondo o grande conjunto das imagens *nok*), cada um deles com 20 pares de imagens.

O maior número de imagens no conjunto *nok* em relação ao *ok* se deve à existência de um maior número de possibilidades de garras metálicas no estado "com necessidade de manutenção" do que no "perfeito estado". Exemplos representativos das principais situações possíveis precisavam ser levantados para o treinamento da Rede Neural, por isso a diferença de cardinalidade entre os conjuntos.

Uma vez adquiridas as imagens, as mesmas foram processadas no software MATLAB, versão R2009a, em computador portátil (*notebook*), com processador Core I3, do fabricante Intel, com 2,7 GHz de Clock e 4 GB de memória RAM.

Um diagrama geral do pré-processamento dos pares de imagens estéreo, da formação da imagem tridimensional, da sua planificação, codificação e classificação por Redes Neurais é expresso na figura 4.1. Faz-se notar neste diagrama de blocos que todas as etapas anteriores à correspondência são duplicadas até que se tenha o contorno tridimensional da garra metálica. Após isso o processamento segue um fluxo único até a classificação das imagens.

As próximas seções irão detalhar as etapas do diagrama de blocos 4.1.

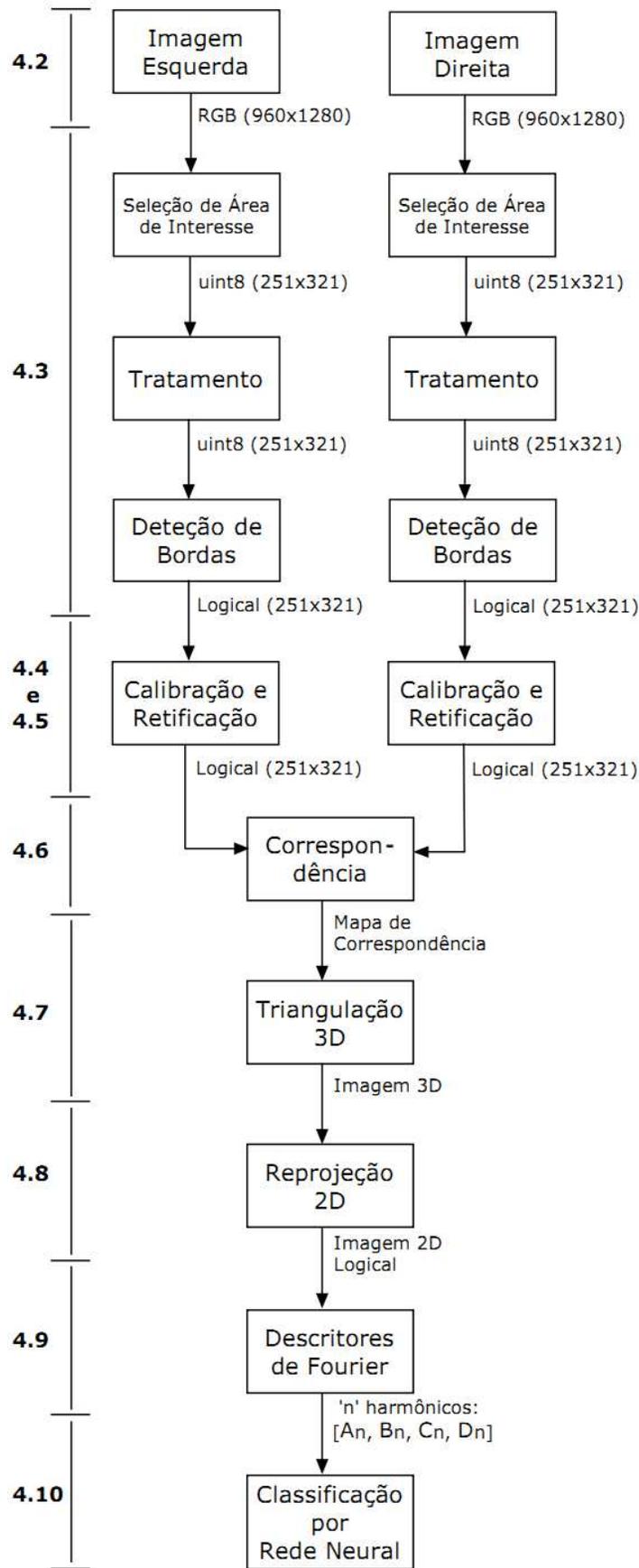


Figura 4.1: Diagrama em blocos do reconhecimento de falhas nos espaçadores de mesma fase (com indicação das seções correspondentes).

Indicam-se o tamanho e o formato das imagens em cada etapa do processo, representado pelo diagrama de blocos da figura 4.1, bem como relaciona-se a seção que trata de cada etapa a partir da numeração à esquerda da mesma.

4.2 AQUISIÇÃO DE IMAGENS

A aquisição das imagens foi realizada por um par de câmeras do fabricante Videre, modelo VTH-MDCS3-VARX, em formato RGB do tipo *Bitmap*, com 7.5 FPS, 1280x960 pixels. A interface utilizada para o recebimento de dados das câmeras foi do tipo *firewire* (IEEE 1394). O software utilizado para visualização e controle das câmeras foi o SRI *Small Vision System* fornecido pelo fabricante da câmera. O software e um par de imagens adquiridas a aproximadamente 1,5m podem ser observadas na figura 4.2:



Figura 4.2: Tela do software de aquisição de imagens SRI- Small Vision System.

Assim como na fotografia convencional, o tremer de um fotógrafo degrada a imagem. Num processo de inspeção, baseada em aquisição de imagens aéreas, o trepidar do helicóptero irá degradar as imagens obtidas. Para compensar tal efeito, Jones, 2000, [19], sugere que os movimentos angulares do helicóptero (guinada, rolagem e arfagem) sejam compensados colocando-se as câmeras em uma plataforma giro-estabilizada, travando a linha de visada num referencial inercial.

As distorções inseridas pelas lentes das câmeras (distorção radial e longitudinal) não foram corrigidas neste trabalho. Isto por que o foco do mesmo encontra-se em se obter o formato do contorno das garras metálicas (desprezando-se pequenas distorções nos eixos axiais ou radiais das lentes).

Para a simulação de uma situação de inspeção de linhas de transmissão, as duas câmeras foram acopladas por uma barra de alumínio de 240 mm que as une na horizontal, conforme figura 4.3.



Figura 4.3: Camaras Videre modelo VTH-MDCS3-VARX.

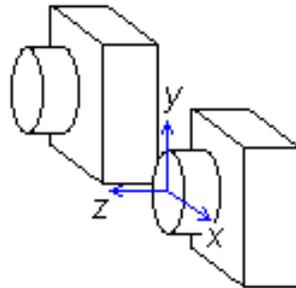


Figura 4.4: Sistema de coordenadas referenciado à câmera esquerda (SCC).

Para facilitar a aquisição de imagens das garras metálicas, em laboratório, foi utilizado um trecho de linha de transmissão. A estrutura utilizada é mostrada na figura 4.5.

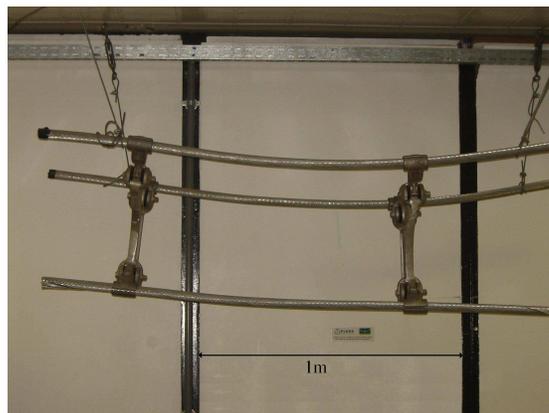


Figura 4.5: Trecho de linha de transmissão com dois espaçadores, suspenso no LARA.

O sistema de coordenadas externo (SCE), representado na figura 4.6, foi fixado em relação a um ponto fixo no laboratório. Sendo o plano xz paralelo ao solo e o eixo y paralelo às colunas verticais do laboratório.

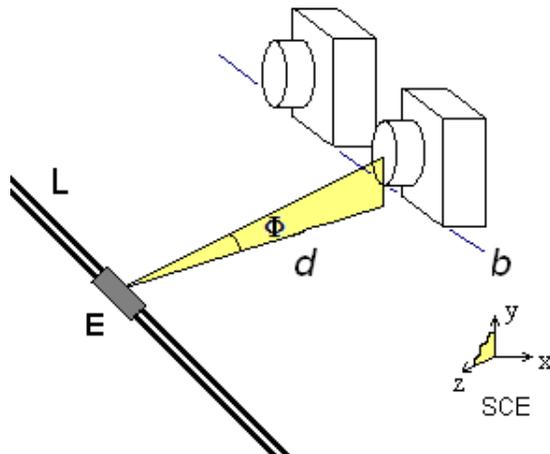


Figura 4.6: Ilustração da elevação das câmeras em relação à linha de transmissão.

A distância d das câmeras, em relação ao espaçador E , é definida como a projeção horizontal do segmento de reta que liga a origem do sistema de coordenadas das câmeras (SCC) ao ponto central do cabo envolto pelo espaçador. Tal ponto é definido como o centro geométrico do polígono definido pelos tokens C_3 , C_4 , C_5 e C_6 (tais pontos podem ser observados na figura 4.15).

A elevação das câmeras é definida como o ângulo entre o segmento de reta d , paralelo ao eixo z (SCE), e o segmento de reta que liga o ponto central do espaçador E ao centro óptico da câmera esquerda (origem do SCC). Tal ângulo pode ser observado na figura 4.6.

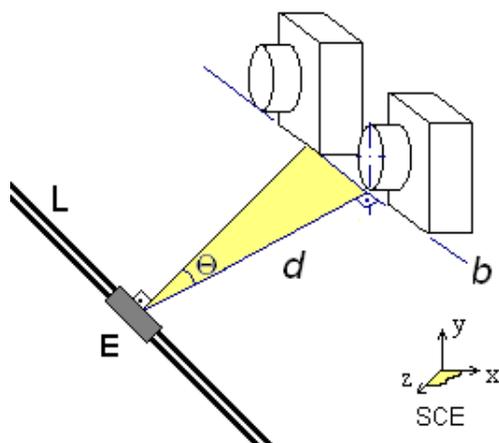


Figura 4.7: Ilustração do azimute de orientação da câmera.

Conforme pode ser observado na figura 4.7, o azimute das câmeras (em relação à garra do espaçador) é expresso como o ângulo entre o segmento de reta (d) que cruza o eixo y do SCC (perpendicular à linha de base das câmeras, b) e um segmento de reta que passa pela centro do espaçador (perpendicular à linha de transmissão).

As imagens adquiridas para este trabalho foram obtidas de uma distância, d , variando entre 1,0 m e 1,6 m entre as câmeras e a linha de transmissão (L). O ângulo de azimute (Θ) das

câmeras oscilou entre 0° e 45° . Já a elevação (Φ) das câmeras não foi variada, permanecendo constante em 0° .

A partir das imagens que englobam todo um trecho de condutores e os próprios espaçadores metálicos foram selecionadas áreas que continham somente a Região de Interesse (RI). Em virtude da seleção da RI da imagem, ou seja, do "recorte" de um pedaço da imagem, a resolução da mesma foi reduzida para 251×321 pixels. Tal imagem é ilustrada na figura 4.8.

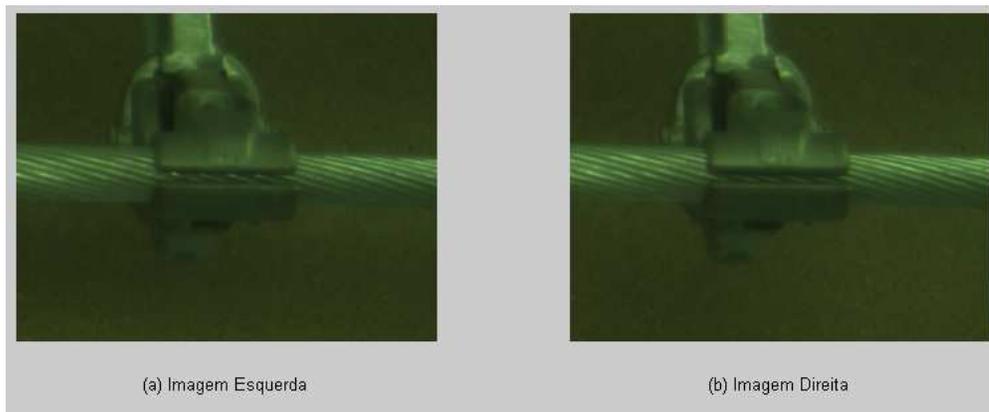


Figura 4.8: Par Estéreo. (a) Imagem obtida pela câmera da esquerda. (b) Imagem Obtida pela câmera da direita.

Para que tal imagem pudesse ser obtida foi utilizada a função *imcrop()* do software Matlab. Uma região do condutor entre a garra superior e inferior foi selecionada pelo usuário e, a partir das dimensões desta marcação, foram definidas as fronteiras da Região de Interesse na imagem completa.

4.3 DETEÇÃO DE BORDAS

Para que a informação em uma imagem (ou em um conjunto das mesmas) possa ser extraída, muitas vezes se faz necessário não só separar uma região da imagem como RI, mas também separar a Região de Interesse em vários planos. O usual é a utilização de 2 planos básicos: o *foreground* (plano principal) e o *background* (plano de fundo).

Para a separação de planos na imagem foi utilizado o algoritmo de *alpha matting* [27, 28], que faz uso de uma abordagem analítica *Closed Form Solution*. Para que tal algoritmo possa separar adequadamente os planos da imagem, foram utilizadas duas máscaras sobrepostas às imagens das RIs: uma de cor de branca, que define as regiões pertencentes ao *foreground* e uma de cor negra que define as regiões pertencentes ao *background*. A imagem da RI bem como as máscaras típicas podem ser observadas na figura 4.9.

Como a posição das câmeras em relação à linha de transmissão é variável em cada caso (depende da posição do helicóptero), a posição das máscaras de *foreground* e *background*

também o é em relação à imagem obtida. Para o ajuste das posições das máscaras foram utilizadas as coordenadas obtidas a partir da função *imcrop()* do software Matlab, que já foi aplicada na seleção da RI.

Após a seleção do plano principal algumas funções de tratamento são utilizadas para que a imagem obtida no processo de *alpha matting* possa se adequar às próximas etapas (detecção de bordas e obtenção dos descritores de Fourier da imagem).

Neste processo as imagens são binarizadas a partir de um limiar variável calculado pela função *graythresh()* e partes do contorno da imagem que se encontrem desconectados (em dimensões inferiores a 50 pixels) são fechados pela função *bwmorph()*, ambas funções do Matlab.

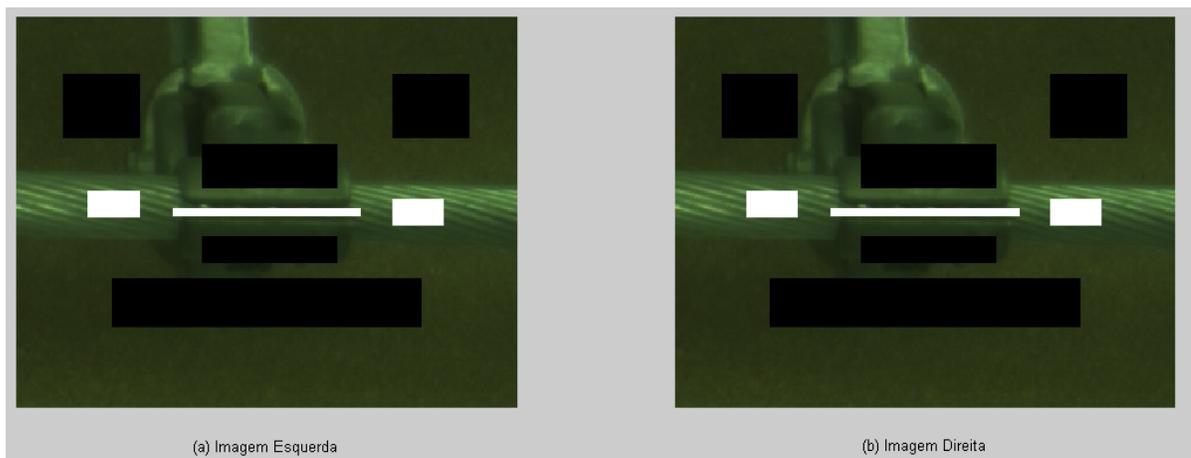


Figura 4.9: Par Estéreo. (a) Máscara utilizado no *Matting* da Imagem Esquerda. (b) Máscara utilizado no *Matting* da Imagem Esquerda.

Para finalizar o processo de Detecção de Bordas foi utilizada a função *bwperim()*, cujo código é apresentado no anexo B3, para que o contorno (perímetro) do condutor preso pela garra metálica possa ser obtido. O resultado obtido pode ser observado na figura 4.10.

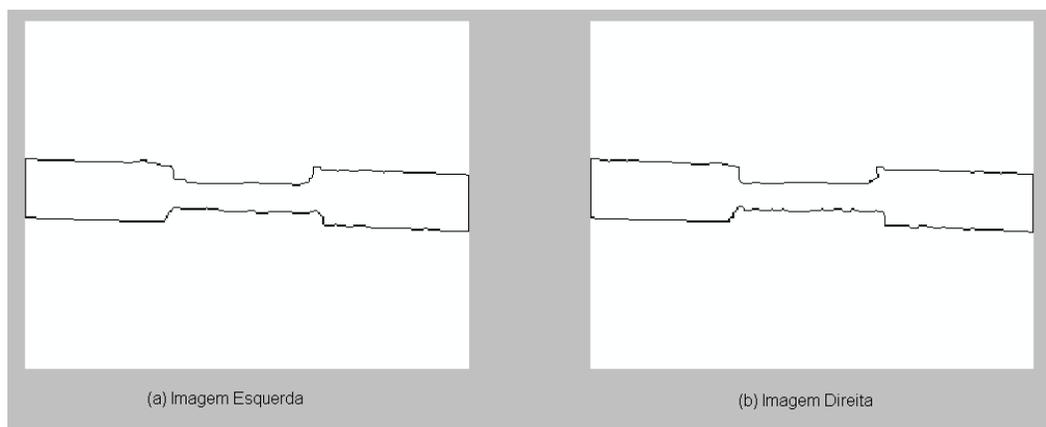


Figura 4.10: Par Estéreo. (a) Contorno obtido a partir da imagem esquerda. (b) Contorno obtido a partir da imagem direita.

4.4 CALIBRAÇÃO DAS CÂMERAS

Na área de Visão Computacional uma das etapas mais basilares é a calibração das câmeras. Neste trabalho, para o processo de Calibração das Câmeras individualmente, foi utilizada a *toolbox Calib*, desenvolvida pela Caltech para o software Matlab. A figura 4.11 mostra o menu principal da *toolbox*.



Figura 4.11: *Toolbox Calib* utilizada para calibração de cada uma das câmeras.

Uma vez calibradas as câmeras individualmente, partiu-se para a calibração do conjunto estéreo. Para tanto foi utilizada o conjunto de funções *stereo_gui* também da *tollbox Calib*. O conjunto de imagens de calibração utilizado é mostrado na figura 4.12.

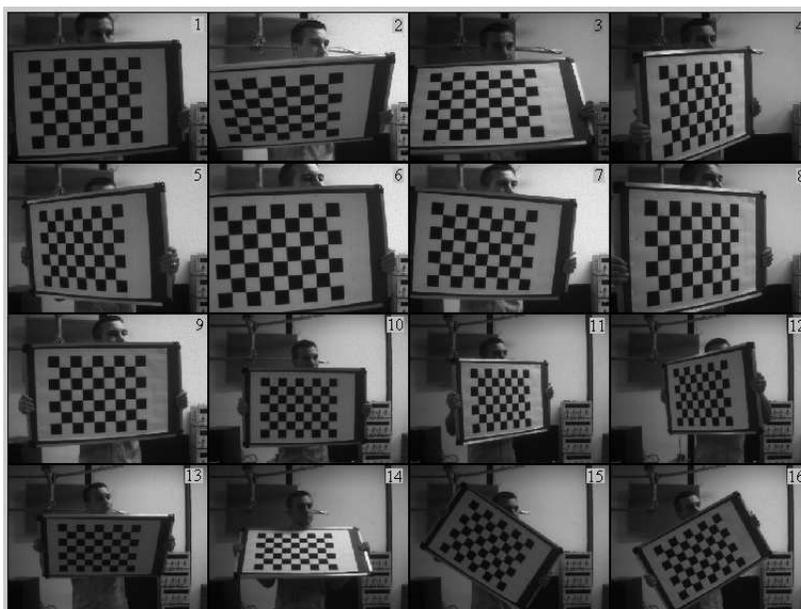


Figura 4.12: Imagens utilizadas para calibração das câmeras.

A posição das imagens de calibração em relação a uma das câmeras pode ser observada na figura (4.13):

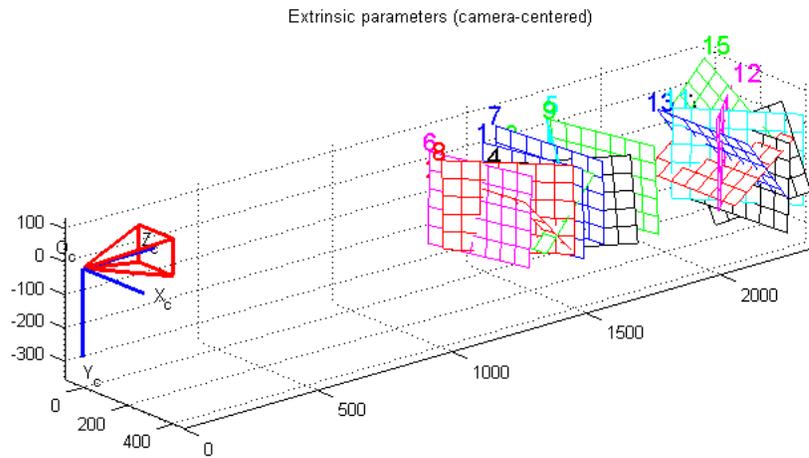


Figura 4.13: Posição das imagens utilizadas na Calibração de uma das câmeras.

4.5 RETIFICAÇÃO

No contexto da Visão estéreo uma das etapas mais custosas computacionalmente é a etapa do estabelecimento da correspondência de pixels de uma imagem do par estéreo na outra. Para reduzir tal custo, neste trabalho optou-se por realizar a correspondência apenas nas bordas do objeto alvo, dentro da Região de Interesse. Mas mesmo com tal simplificação, ainda tem-se uma busca bidimensional para a correspondência. Para que este processo possa ser simplificado ainda mais, ou seja, transformado numa busca unidimensional, faz-se uso do processo de Retificação.

Neste trabalho foi utilizada a *toolbox Calib*, desenvolvido pela Caltech, para a Retificação das imagens. Pode-se observar, na figura 4.14 que os pontos homólogos encontram-se alinhados nas dimensões y das imagens, ou seja, os pontos equivalentes do objeto amostrado (como P_1' e P_1'' em vermelho e P_2' e P_2'' em azul), encontram-se nas mesmas linhas de cada imagem. Pode-se observar ainda que existem distorções (partes em branco) nas bordas das imagens. Isto se deve ao processo de Retificação que "distorce" a imagem para que os pontos homólogos possam se alinhar nas dimensões de linha das imagens.

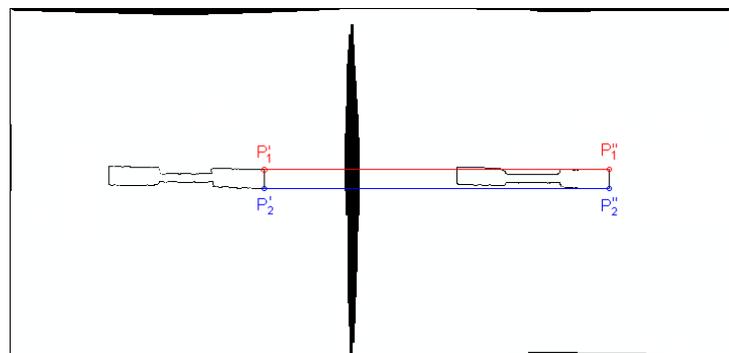


Figura 4.14: Ilustração de um par estéreo retificado.

4.6 CORRESPONDÊNCIA

O processo de correspondência entre pixels de imagens de um par estéreo constitui uma das etapas mais custosas em termos computacionais. Assim, visando reduzir o "esforço" computacional e o tempo de processamento algumas considerações foram feitas no levantamento do diagrama de correspondência:

- A busca pela correspondência entre pixels das imagens do par estéreo só foi realizada nas bordas das garras metálicas.
- Foram utilizados pontos notáveis nas imagens (*tokens*) para auxiliar o processo de correspondência.
- Optou-se por modelar as correspondências entre os pixels por Diagramas de Correspondência Esparsos.

Como as imagens já tinham sido submetidas à Retificação os índices de linhas de pixels correspondentes eram iguais em ambas as imagens, faltando apenas a busca pelos índices de coluna. Desta forma foi empreendida uma busca unidimensional (apenas os índices de colunas) para o estabelecimento de correspondência entre pixels equivalentes nos pares de imagens.

A correspondência entre pixels de imagens do par estéreo apenas foi realizada nas regiões de bordas das garras metálicas porque tais regiões possuem informações suficientes para a classificação do estado da garra. Além disso, foram utilizados pontos notáveis nas bordas (*tokens*) para facilitar o processo de correspondência. Tais pontos foram nomeados como Pontos Relevantes e podem ser observados na figura 4.15. Desta forma, a correspondência foi realizada trecho a trecho, ou seja, os pixels que se encontram na região limitada por C_1 (em vermelho na figura 4.15), apenas se associam com pontos da mesma região na outra imagem do par estéreo, sendo este princípio válido para todas as regiões.

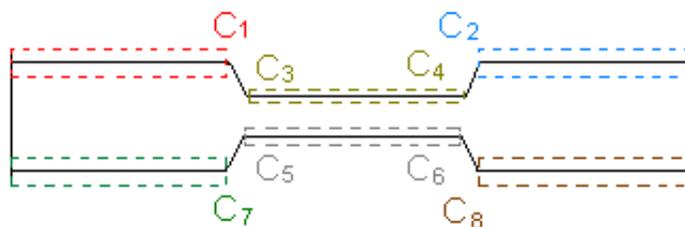


Figura 4.15: Pontos Relevantes C_1 a C_8 : ‘Tokens’ para o processo de Correspondência entre pixels num par de imagens.

A partir dos Pontos Relevantes identificados, janelas de correlação foram definidas para que regiões de uma imagem possam ser comparadas com regiões de outra. O método de

estimativa da correlação utilizado foi a Soma das Diferenças Absolutas (SDA), expresso pela equação (2.23). As janelas de pixels com menor SDA, sem inconsistências, tiveram seus pixels associados e armazenados na matriz D . Sendo nesta a dimensão i comum às duas imagens (pelo processo de retificação) e o pixel da imagem esquerda, indexado por (i_n, jL_n) , associado ao pixel da imagem direita, indexado por (i_n, jR_n) , para as n correspondências possíveis.

$$D = \begin{bmatrix} i_1 & jL_1 & jR_1 \\ i_2 & jL_2 & jR_2 \\ i_3 & jL_3 & jR_3 \\ \vdots & \vdots & \vdots \\ i_n & jL_n & jR_n \end{bmatrix}. \quad (4.1)$$

4.7 MODELO 3D

Neste trabalho as coordenadas 3D foram obtidas a partir da função *rectify()*, desenvolvida por [21], nesta são utilizadas as matrizes de calibração \hat{P}_{n1} e \hat{P}_{n2} , equação (2.30) e a matriz de disparidade D , equação (4.1) das imagens para a determinação da imagem tridimensional do contorno da garra metálica.

Um exemplo de obtenção das coordenadas tridimensionais de uma garra metálica pode ser observado na figura 4.16.

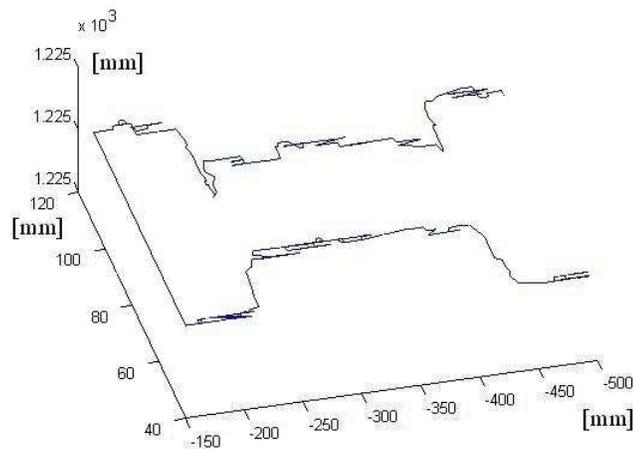


Figura 4.16: Imagem tridimensional reconstituída do contorno de uma garra metálica.

4.8 REPROJEÇÃO 2D

Seja um vetor \vec{C}_0 , representando a orientação inicial de uma câmera virtual, com uma matriz de projeção perspectiva P_v . Dado um plano ρ , paralelo e coincidente ao plano frontal da garra metálica, com um vetor \vec{n} , normal ao plano ρ . Tais elementos geométricos foram utilizados para se estimar a translação e as rotações necessárias para que a câmera virtual atingisse a orientação paralela a \vec{n} e obtivesse a mesma distância em relação ao plano ρ . A partir destes elementos geométricos as imagens tridimensionais puderam ser reprojetaadas para um plano paralelo ao plano frontal da garra metálica (ρ), com uma distância constante da mesma, independentemente da orientação e distância do par de câmeras reais. A câmera virtual e seu vetor de orientação \vec{C} podem ser vistos na figura 4.17 e o plano ρ pode ser observado na figura 4.18.

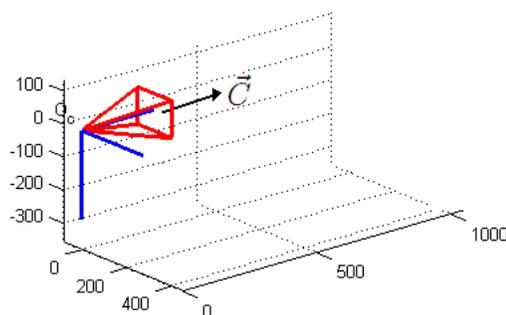


Figura 4.17: Vetor que descreve a orientação da câmera virtual \vec{C} .

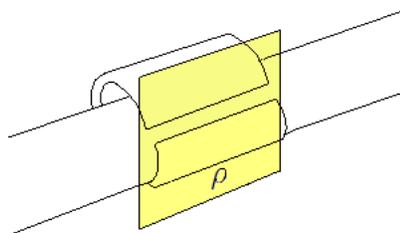


Figura 4.18: Plano ρ , paralelo a face frontal da garra metálica.

As imagens tridimensionais puderam ser projetadas através da Matriz de Projeção Perspectiva P_v de uma câmera virtual situada hipoteticamente a 1000 mm da imagem 3D, com orientação perpendicular ao plano ρ . Tal procedimento foi realizado com objetivo de padronizar a distância e a orientação da câmera de observação em relação à garra metálica.

A escolha dos parâmetros da câmera virtual se deu da seguinte forma: Os parâmetros intrínsecos foram adotados arbitrariamente, já os parâmetros extrínsecos foram escolhidos de forma que a orientação da câmera virtual fosse paralela ao vetor normal ao plano ρ a uma distância padronizada da imagem 3D.

A determinação dos coeficientes do plano ρ foi realizada a partir dos *tokens* (Pontos

Principais) da imagem tridimensional do contorno da garra metálica. Como os tokens podem não estar perfeitamente contidos no mesmo plano foi utilizado o Método dos Mínimos Quadrados (MMQ) na determinação do plano que melhor interpolasse os Pontos Relevantes (mostrados na figura 4.15 na sua versão bidimensional).

Uma vez determinados os coeficientes do plano ρ , pôde-se chegar ao seu vetor normal e com isso pode-se orientar a câmera virtual numa perspectiva desejada para a planificação da imagem tridimensional (reprojeção).

4.9 DESCRITORES DE FOURIER

Mesmo com todos os avanços em termos de desempenho de *Hardware* e *software*, o processamento de imagens ainda constitui um grande desafio sob o ponto de vista do custo computacional. Processar imagens coloridas (RGB) de alta resolução pode aumentar o tempo de processamento inviabilizando uma série de aplicações que exijam curtos tempos de resposta. Neste cenário, a compactação de imagens com a mínima perda de informação possível se torna essencial. Neste trabalho, a técnica utilizada foi a modelagem dos contornos das imagens a partir de Séries de Fourier Bidimensionais (SFB). Após a reprojeção da imagem tridimensional do contorno da garra para um plano paralelo a sua face frontal, tal projeção foi utilizada para a geração dos Descritores de Fourier. Tal procedimento se deve ao fato de que ao invés de se trabalhar com uma imagem de 251x321 pixels trabalha-se apenas com os vetores dos Descritores de Fourier. Para que a imagem do contorno da garra tivesse uma descrição que visualmente fosse fidedigna foram utilizados 25 coeficientes de Fourier. Como um Descritor de Fourier Bidimensional possui 2 coeficientes que representam os *offsets* horizontal e vertical (A_0 e B_0), 2 coeficientes que representam as contribuições das harmônicas, de ordem par e ímpar, na horizontal (A_n e B_n) e 2 coeficientes que representam as contribuições das harmônicas, de ordem par e ímpar, na vertical (C_n e D_n). Assim tem-se: $2 \times (2 \text{ coeficientes de Offset}) + 25 \times (4 \text{ coeficientes de Harmônicos})$. Totalizando 102 coeficientes. Como os 2 coeficientes de *Offset*, A_0 e B_0 , simplesmente deslocam a imagem horizontal e verticalmente tais coeficientes foram ignorados. Desta forma pode-se ter uma maior imunidade à translação da imagem e passa-se a descrever uma imagem apenas por um vetor composto por 100 coeficientes de Fourier ao invés de uma matriz de 1280x960. A plotagem do contorno de uma imagem ilustrativa a partir de 25 coeficientes de Fourier pode ser observada na figura 4.19.

Tais coeficientes são repassados a uma rede Neural, devidamente treinada, que se encarrega da classificação da garra metálica, considerando-a como: "ok" ou "nok" (com necessidade de manutenção).

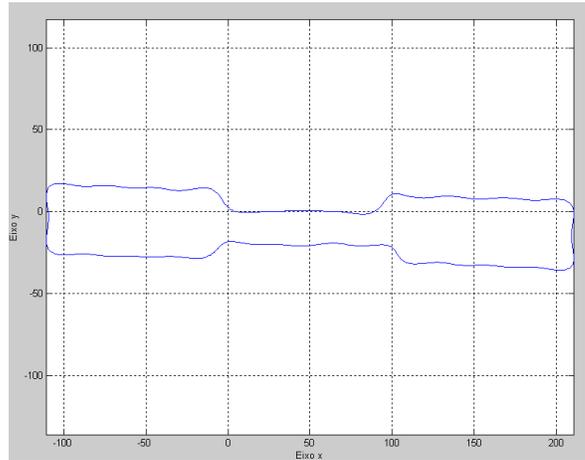


Figura 4.19: Plotagem do contorno de uma garra metálica a partir de Séries de Fourier Bidimensionas.

4.10 CLASSIFICAÇÃO POR REDES NEURAIS

As Redes Neurais utilizadas foram implementadas a partir da função *newff()* da *toolbox nn_tools* do Matlab, uma das interfaces desta plataforma pode ser observada na figura 4.20.

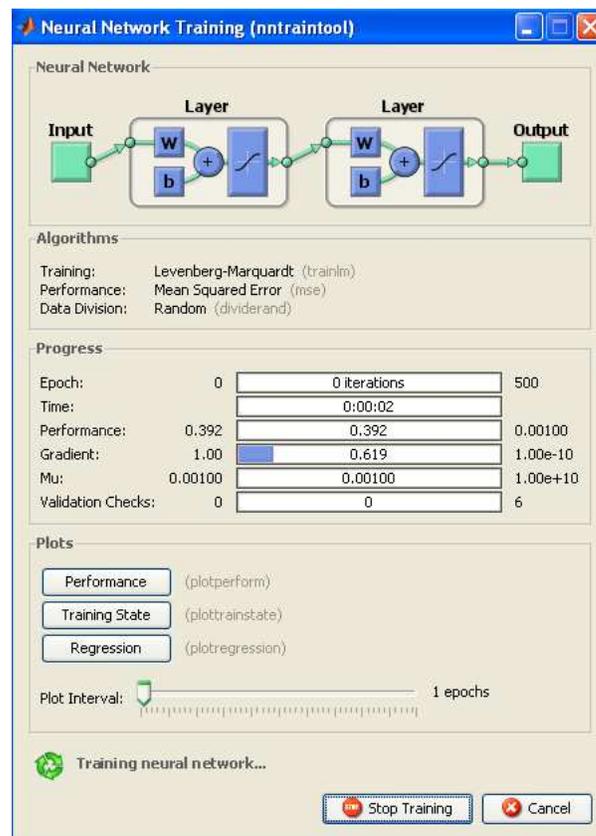


Figura 4.20: Interface de treinamento das Redes Neurais.

Optou-se por uma arquitetura de 2 camadas, utilizando o método *Back-propagation*, com a função de treinamento *trainblm()*. A função de performance escolhida foi o *MSE* e a taxa

de aprendizado adotada foi de 0,001, com limite de 5000 épocas.

A entrada para a Rede Neural foi um vetor contendo os descritores de Fourier dos contornos das imagens e sua saída a classificação da garra metálica: "ok"(nível lógico 1) ou "nok"(nível lógico 0).

Para o conjunto de treinamento foram utilizadas 70 imagens de garras metálicas com estado "ok" e 140 imagens com "nok", ambas em 7 poses. Para os conjuntos de validação e teste foram utilizados conjuntos de 20 pares de imagens, sendo 11 "ok" e 9 "nok".

5 RESULTADOS

Para a análise da viabilidade do processo tratado nesta dissertação foram selecionados 70 pares de imagens representando garras metálicas "ok" e 140 pares referindo-se a garras "nok". No conjunto de amostras foram variadas as distâncias, da câmera em relação ao espaçador, de 1,0 m à 1,60 (tal distância é mostrada na figura 4.6). Também foi variado o ângulo de azimute, das câmeras em relação ao espaçador, de 0° à 45° (tal ângulo pode ser observado na figura 4.7). O ângulo de elevação das câmeras (mostrado na figura 4.6) permaneceu constante em 0° durante os experimentos.

Para melhor exposição dos dados resultantes cada etapa do processo foi isolada e seus resultados são expostos nos itens seguintes.

5.1 RESULTADOS DA DETEÇÃO DE BORDA

No conjunto de imagens de análise considerou-se como sucesso na detecção de bordas as imagens obtidas que conservaram as formas do objeto original. O critério adotado para tal análise foi a comparação visual da imagem original e do contorno obtido. Pode-se observar imagens consideradas como uma detecção de bordas bem sucedida na figura 5.1 (a) e imagens consideradas inadequadas na figura 5.1 (b).

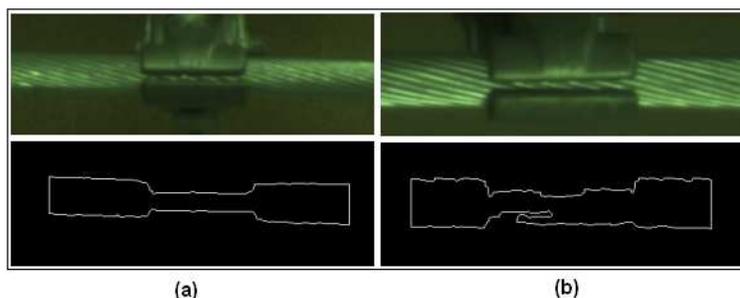


Figura 5.1: Ilustração de detecção de bordas: (a) Bem sucedida. (b) Mal sucedida.

Para que as diferentes situações de posicionamento (da câmera em relação à garra) fossem exploradas, foram estabelecidos, para cada distância e ângulo arbitrados, conjuntos de 10 pares de imagens sem necessidade de manutenção, chamados conjuntos "ok". Para os conjuntos de imagens que representam as garras com necessidade de manutenção, denominados conjuntos "nok", foram selecionados conjuntos de 20 pares de imagens para cada configuração. Tal fato se deve à existência de um maior número de possibilidades de configurações de garras desajustadas do que em bom estado. Desta forma tem-se 7 conjuntos de 10 pares de imagens "ok" e 7 conjuntos de 20 pares de imagens "nok". Para cada nova fase do processo foram repostas imagens para que as perdas em uma etapa não resultassem em

redução no número de elementos do conjunto de imagens utilizado pela etapa posterior.

Os resultados obtidos a partir dos conjuntos de imagens que representam a garra metálica "ok" estão dispostas na tabela 5.1.

Tabela 5.1: Índice de Sucesso obtido na Detecção de Borda (ISDB).

Conjunto de Imagens	Distância (m)	Azimute	ISDB(%)
ok1	1,0	0°	90
ok2	1,2	0°	85
ok3	1,4	0°	75
ok4	1,6	0°	75
ok5	1,2	15°	95
ok6	1,2	30°	80
ok7	1,2	45°	75

A coluna "Azimute" das tabelas 5.1 e 5.2 pode ser visualizada na figura 4.7 como o ângulo θ .

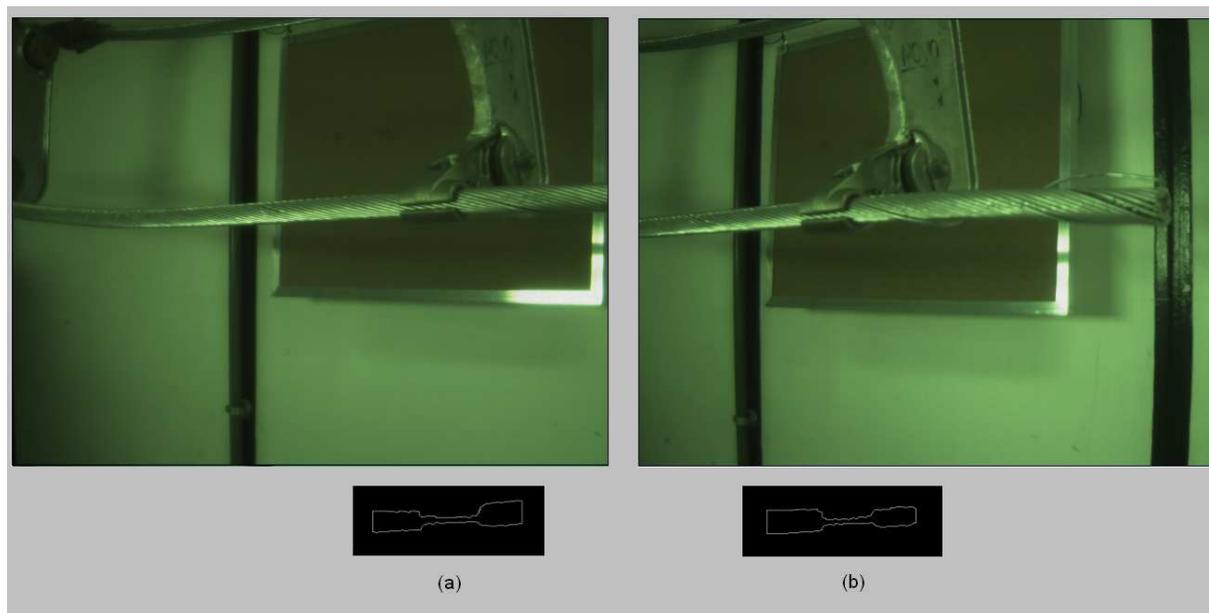


Figura 5.2: Exemplo de detecção de borda para aquisição de imagem à 45°.

Para os conjuntos de imagens que representam o separador metálico "nok" o mesmo procedimento foi realizado e os resultados encontram-se na tabela 5.2.

Os insucessos de algumas detecções de bordas se devem principalmente a fenômenos como a influência da iluminação oscilatória do ambiente (lâmpadas de corrente alternada) e

Tabela 5.2: Índice de Sucesso obtido na Detecção de Borda (ISDB).

Conjunto de Imagens	Distância (m)	Azimute	ISDB(%)
nok1	1,0	0°	85
nok2	1,2	0°	85
nok3	1,4	0°	85
nok4	1,6	0°	85
nok5	1,2	15°	95
nok6	1,2	30°	80
nok7	1,2	45°	75

principalmente a seleção manual das máscaras variáveis (utilizadas no processo de *matching*).

5.2 RESULTADOS DA CALIBRAÇÃO E RETIFICAÇÃO DAS CÂMERAS

A calibração das câmeras foi realizada a partir da *toolbox calib / Stereo_gui* do software Matlab. Tal ferramenta foi desenvolvida pela Caltech e é *freeware*.

A partir da apresentação do painel quadriculado, apresentado na figura 4.12, pôde-se estimar os parâmetros intrínsecos (distância focal f_c e ponto principal M') e extrínsecos (rotação \mathfrak{R} e translação \mathfrak{T}) para as câmeras, assim como suas variâncias e erros. Tais parâmetros são apresentados na tabela 5.3.

Tabela 5.3: Calibração das câmeras.

Parâmetro	Valor
f_{left}	[2283.3 2288.2] \pm [16.9 16.4]
M'_{left}	[607.3 418.1] \pm [30.2 22.8]
f_{right}	[2309.3 2315.4] \pm [15.8 15.8]
M'_{right}	[625.5 415.3] \pm [29.1 23.1]
\mathfrak{R}	[-0.009 -0.016 -0.007]
\mathfrak{T}	[-240.856 -2.273 0.768]

5.3 RESULTADOS DO MATCHING

Seguindo o processo de classificação das garras metálicas dos espaçadores, as imagens que foram submetidas a detecção de bordas são levadas à função de Correspondência, que tem por objetivo estabelecer as correspondências entre os pixels das imagens. Para cada par estéreo foi gerada uma matriz (4.1) que associa os índices de coluna de um pixel da imagem esquerda com o índice de coluna de um pixel da imagem direita. Para que tal matriz possa ser visualizada o seguinte artifício foi realizado: apenas os pontos que possuíam correspondência estabelecida em ambas as imagens foram plotados na imagem esquerda, que foi escolhida arbitrariamente. Após tal processo pôde-se chegar à imagem expressa pela figura 5.3.

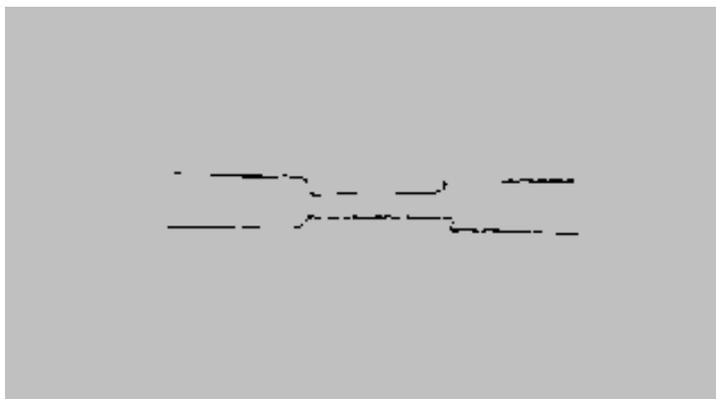


Figura 5.3: Visualização de correspondência de pixels, imagem típica OK.

O critério adotado neste trabalho para a aceitação de uma correspondência não está ligado diretamente ao número de pixels associados, mas sim à presença de pixels associados nas regiões definidas por cada um dos Pontos Relevantes (Figura 4.15). Assim, cada um dos pares de imagem estéreo teve seu processo de Correspondência considerado como correto se em cada um dos Pontos Relevantes forem encontrados *pixels* suficientes para não alterar o formato da imagem original. Da mesma forma, se na citada região houver ausência de *pixels* associados em pelo menos um dos Pontos Relevantes a Correspondência é considerada como insuficiente e as imagens são descartadas. Se tal critério for levado ao limite apenas são necessários 8 pixels associados, desde que cada um destes se encontre em cada um dos 8 Pontos Relevantes (já sendo estes pontos suficientes para se recompor a imagem original). Na prática foram encontrados na média 650 pixels associados, em cada par de imagens estéreo.

Os resultados obtidos na Correspondência das imagens que representam as garras sem necessidade de manutenção são apresentadas nas tabelas 5.4 e 5.5.

Tabela 5.4: Índice de Sucesso obtido na Correspondência.

Conjunto de Imagens	Distância (m)	Azimute	Acertos(%)
ok1	1,0	0°	90
ok2	1,2	0°	80
ok3	1,4	0°	60
ok4	1,6	0°	60
ok5	1,2	15°	80
ok6	1,2	30°	80
ok7	1,2	45°	50

Tabela 5.5: Índice de Sucesso obtido na Correspondência.

Conjunto de Imagens	Distância (m)	Azimute	Acertos(%)
nok1	1,0	0°	80
nok2	1,2	0°	80
nok3	1,4	0°	60
nok4	1,6	0°	50
nok5	1,2	15°	70
nok6	1,2	30°	80
nok7	1,2	45°	40

5.4 RESULTADOS DA RECONSTRUÇÃO 3D

A partir da matriz A (definida em (4.1)), das matrizes de calibração das câmeras (expressos na tabela 5.3 e da equação (2.28) pode-se chegar às coordenadas tridimensionais do contorno da garra metálica. Os pontos obtidos foram ligados e o resultado pode ser observado na figura (5.4).

Nesta etapa os bons resultados dependem basicamente dos resultados na etapa anterior (Correspondência). De forma que os resultados da obtenção das coordenadas tridimensionais do objeto refletem os resultados obtidos na Correspondência. Assim, pode-se afirmar que uma correspondência aceitável gera uma imagem tridimensional aceitável e vice versa. Tais resultados podem ser observados nas tabelas (5.6) e (5.7):

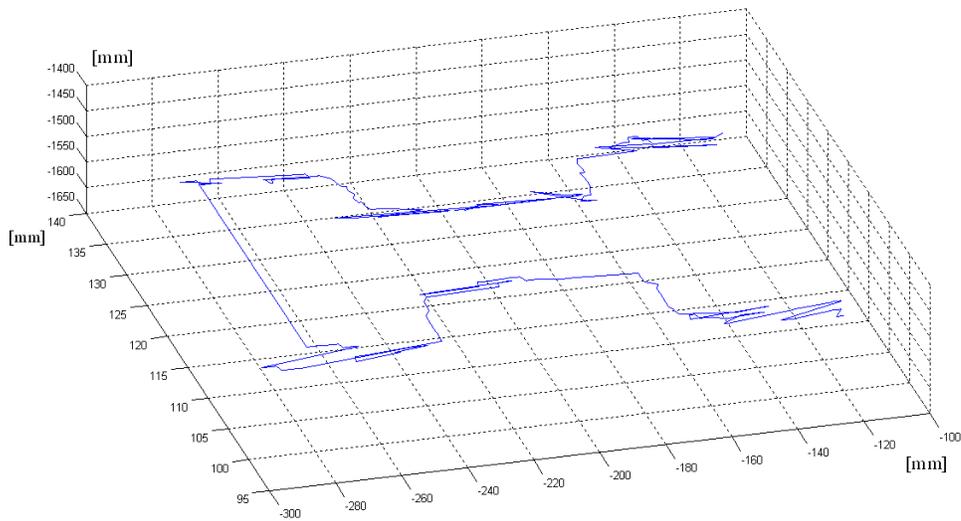


Figura 5.4: Imagem Tridimensional do contorno de uma garra metálica.

Tabela 5.6: Sucesso na obtenção das coordenadas tridimensionais.

Conjunto de Imagens	Distância (m)	Azimute	Acertos(%)
ok1	1,0	0°	90
ok2	1,2	0°	80
ok3	1,4	0°	60
ok4	1,6	0°	60
ok5	1,2	15°	80
ok6	1,2	30°	80
ok7	1,2	45°	50

Tabela 5.7: Sucesso na obtenção das coordenadas tridimensionais.

Conjunto de Imagens	Distância (m)	Azimute	Acertos(%)
nok1	1,0	0°	80
nok2	1,2	0°	80
nok3	1,4	0°	60
nok4	1,6	0°	50
nok5	1,2	15°	70
nok6	1,2	30°	80
nok7	1,2	45°	40

5.5 RESULTADOS DA REPROJEÇÃO 2D

Após a obtenção das coordenadas tridimensionais do contorno da garra metálica estas precisam ser novamente reprojctadas sob uma perspectiva padronizada. Assim, todas as imagens, independentemente do ângulo e distância de aquisição, são levadas às redes neurais como se vistas a partir de uma mesma orientação e distância das câmeras.

Devido à utilização de *tokens* (mostrados na Figura 4.15) como elementos de auxílio na etapa de Correspondência, foram inseridos erros de $+/- 3 \text{ pixels}$ na mesma. Estes erros se refletem sob a forma de descontinuidades nas coordenadas relativas à profundidade (eixo z) da imagem 3D. Tais descontinuidades inserem erros na coordenada de profundidade dos Pontos Relevantes 3D, e estes são utilizados para determinar os coeficientes do plano ρ e de seu vetor normal \vec{n} . Como este vetor tem a função de orientar a câmera virtual para a reprojeção da imagem 3D, sem a sua determinação precisa não se torna viável a reprojeção de imagem por este método.

Neste cenário pode-se concluir que esta metodologia se mostrou incapaz de atingir a perspectiva ideal da imagem tridimensional devido a erros na obtenção das coordenadas dos Pontos Relevantes.

Com o insucesso do processo de reprojeção, pelo método descrito, as imagens foram giradas manualmente até que a perspectiva ideal da imagem 3D fosse alcançada. Tais imagens reprojctadas podem ser observadas na figura 5.5.

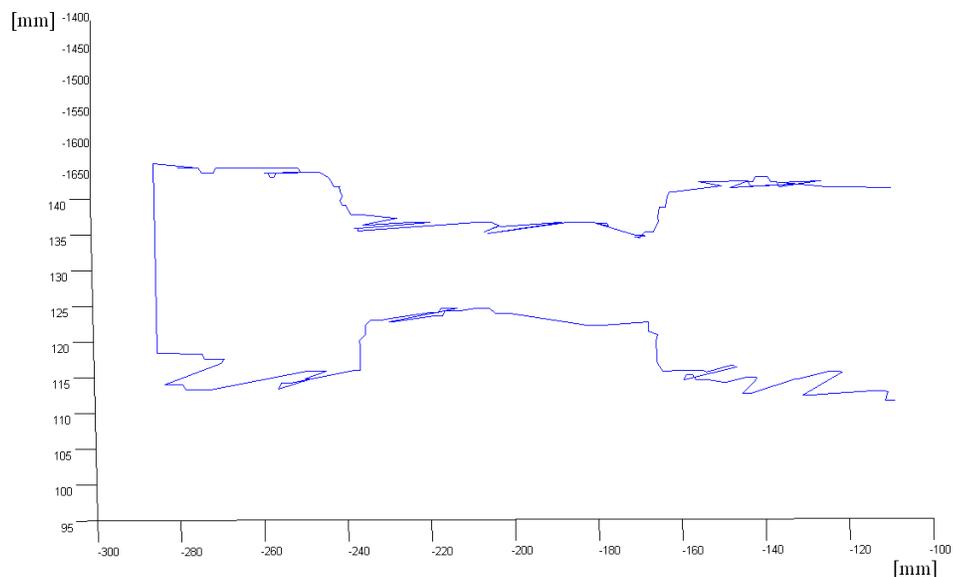


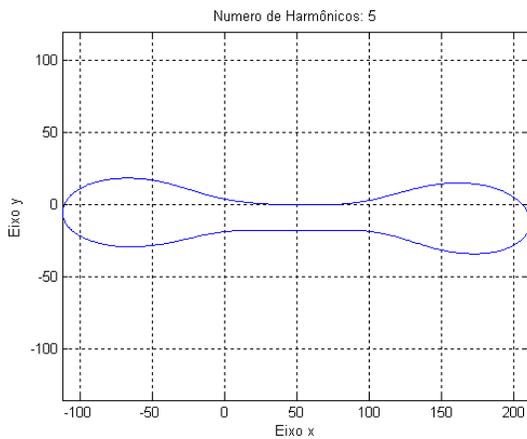
Figura 5.5: Imagem reprojctada a partir da imagem tridimensional de uma garra metálica.

5.6 RESULTADOS DAS CADEIAS DIRECIONAIS

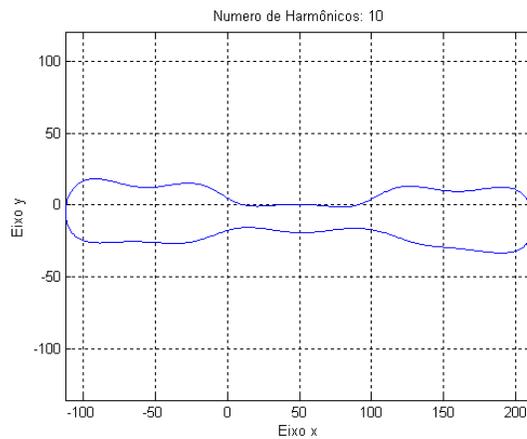
Após o processo de reprojeção da imagem tridimensional, tem-se disponível o contorno da garra metálica numa perspectiva favorável à sua análise. A imagem do contorno é então codificada em cadeias direcionais (conforme a figura 2.14). Um exemplo de um vetor que descreve o contorno da imagem pode ser observado no apêndice A.

5.7 RESULTADOS DOS DESCRITORES DE FOURIER

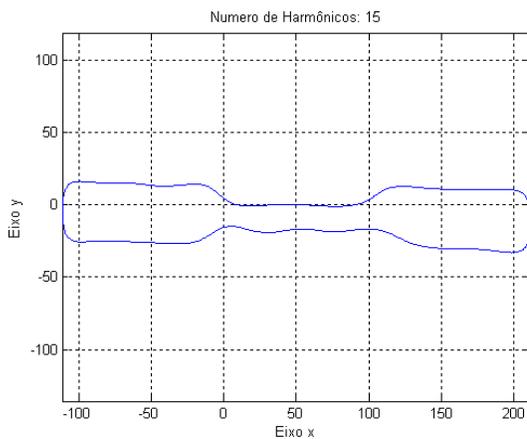
Para expressar de maneira compacta as informações de contorno e reduzir o volume de informações tratadas pela Rede Neural foram obtidos os Descritores de Fourier do contorno do objeto. No tocante ao número de harmônicos necessários para tal intento, foram analisadas plotagens para 10, 15, 20 e 25 harmônicos. Os critérios adotados neste experimento visavam minimizar o número de coeficientes sem alteração significativa no formato do contorno do objeto. Podem ser observadas as imagens plotadas a partir de diferentes números de harmônicos relativas ao par de imagens exibido na Figura (4.8).



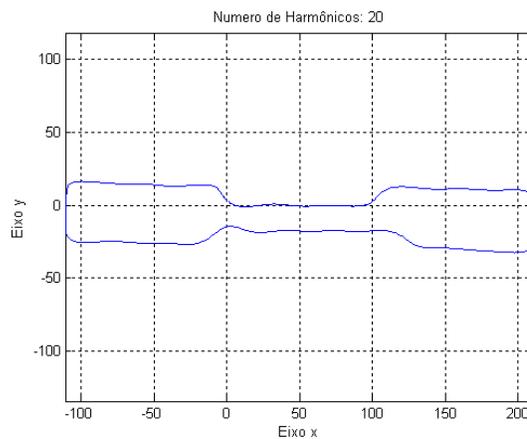
(a) 5 harmônicos



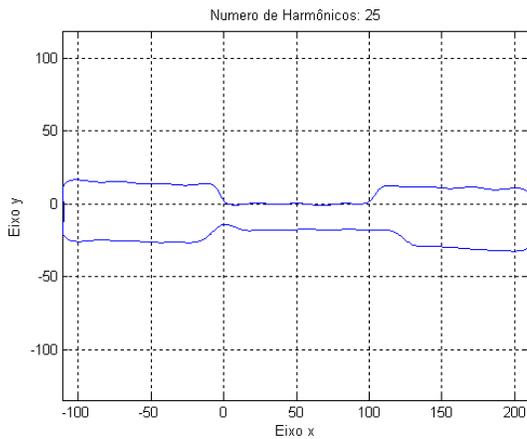
(b) 10 harmônicos



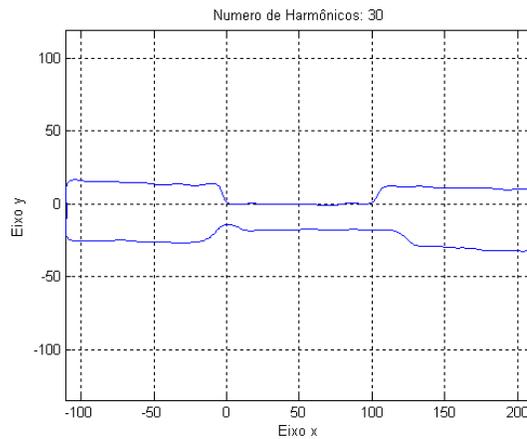
(c) 15 harmônicos



(d) 20 harmônicos



(e) 25 harmônicos



(f) 30 harmônicos

Figura 5.6: Plotagens para diferentes números de harmônicos.

Segundo uma análise visual o ponto ótimo foi encontrado em 25 harmônicos, figura 5.6 (e). Pode-se observar os coeficientes da mesma figura expressos na Tabela (5.8).

Tabela 5.8: Descritores de Fourier.

n	A_n	B_n	C_n	D_n
1	-46,594	-385,577	-98,787	-8,834
2	-19,097	35,303	20,273	5,534
3	-36,128	43,692	69,029	0,706
4	5,0645	-0,248	-14,513	1,113
5	-6,854	-9,600	-20,434	-7,855
6	-5,125	11,448	1,596	8,782
7	-11,391	-1,275	2,620	7,569
8	-0,299	6,229	-8,789	-4,607
9	-1,579	6,059	-2,164	-0,391
10	2,611	3,218	3,370	-1,706
11	0,184	1,951	3,884	0,405
12	-7,881	4,447	1,120	-4,193
11	-0,590	1,902	1,561	-0,829
12	-3,718	0,704	0,036	0,117
13	-4,183	0,939	-1,666	0,792
14	-6,519	-0,100	-0,685	3,075
15	-1,517	-1,732	-1,695	0,596
16	1,9857	-0,818	0,400	-0,602
17	-0,471	4,204	-1,530	0,245
18	-1,041	0,348	0,899	0,938
19	-6,012	0,192	0,782	-2,858
20	-1,604	-1,722	-1,109	-1,584
21	-0,763	-3,409	1,633	-0,257
22	-0,482	1,157	-0,649	0,890
23	0,017	-1,022	-0,024	0,040
24	-0,482	1,158	-0,649	0,890
25	0,017	-1,022	-0,024	0,040

Os coeficientes A_0 e C_0 foram desprezados na composição do vetor de entrada das Redes Neurais, pois estes representam apenas os deslocamentos da "imagem" vertical e horizontalmente, respectivamente. Assim a não utilização dos mesmos torna a imagem plotada invariante aos movimentos de translação, o que contribui para robustez da etapa final de classificação da garra metálica.

Cabe salientar ainda o papel de filtro exercido pela aplicação dos descritores de Fourier. Os componentes de alta frequência, representados na imagem como ondulações e pequenas distorções, não aparecem nas plotagens com número reduzido de harmônicos. Assim,

a utilização de um número limitado de harmônicos, neste caso 25, faz com que as séries de Fourier operem como um Filtro Passa Baixa e sirvam também para eliminar pequenas imperfeições oriundas da etapa de detecção de borda.

5.8 RESULTADOS DO TREINAMENTO DA REDE NEURAL

Um bom critério de parada para o treinamento de uma Rede Neural é o ponto no qual a rede apresenta uma boa capacidade de generalização e o menor erro quadrático médio possível. A evolução do treinamento da Rede Neural pode ser observada na figura 5.7.

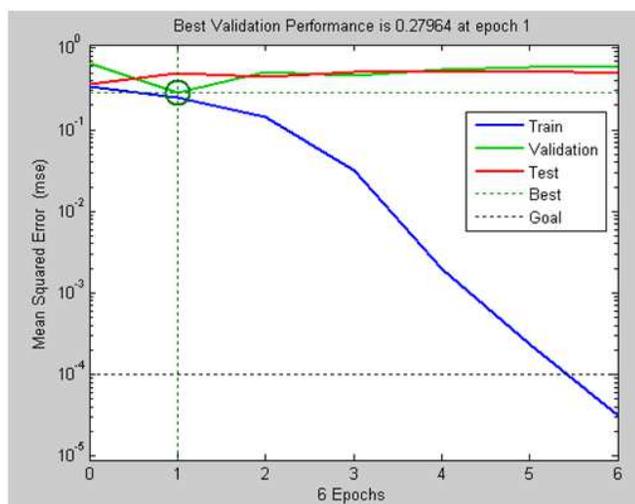


Figura 5.7: MSE ao longo do processo de treinamento da Rede Neural.

Outro critério a ser observado é o gradiente do MSE, pois quando este tende a zero tem-se o ponto de mínimo na superfície de erro (Figura 3.8). A evolução do gradiente do erro médio quadrático pode ser observado na figura 5.8.

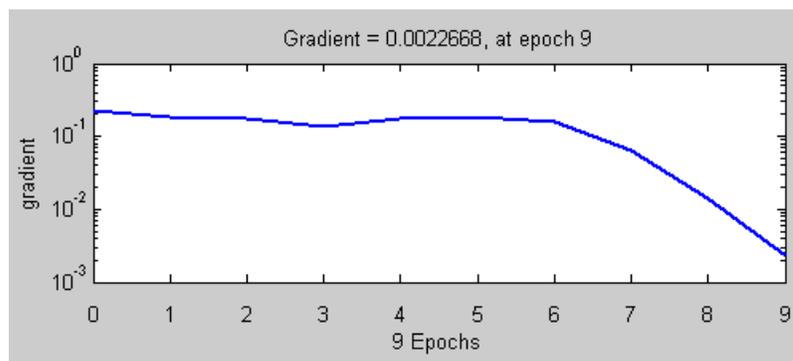


Figura 5.8: Gradiente do MSE ao longo do processo de treinamento da Rede Neural.

5.9 RESULTADOS DA REDE NEURAL

Como mencionado anteriormente, a Rede Neural treinada foi submetida a um conjunto de 20 entradas de teste, sendo 11 delas representativas de garras metálicas em "bom estado"(nível lógico "1") e 9 delas representando garras "com necessidade de manutenção"(nível lógico "0"). Os resultados desta aplicação podem ser visualizados na tabela 5.9.

Tabela 5.9: Resultado da classificação pela Rede Neural.

Par Estéreo	Resultado Desejado	Resultado Obtido
1	1	0,9877
2	0	-0,1086
3	1	0,8853
4	0	0,0034
5	1	-0,2314
6	0	0,2314
7	0	0,5432
8	1	0,7654
9	0	0,1439
10	1	0,9912
11	0	0,3450
12	0	0,2965
13	1	0,9654
14	1	0,8935
15	1	0,7652
16	0	0,1245
17	0	0,5459
18	1	0,8254
19	1	0,8845
20	1	0,9237

O limiar de classificação utilizado para a classificação foi de 0,5. Ou seja, valores menores que 0,5 foram classificados como "nok" e valores maiores de 0,5 foram classificados como "ok".

A partir dos resultados pôde-se inferir que apenas 3 das garras foram classificadas erroneamente (marcadas em vermelho na tabela 5.9). Destes tem-se 2 falsos positivos (situações de defeito classificadas como "ok") e um falso negativo (situação de garra em perfeitas condições classificada como "nok").

As imagens que compõem o conjunto de teste estão dispostas nas figuras 5.9 e 5.10.

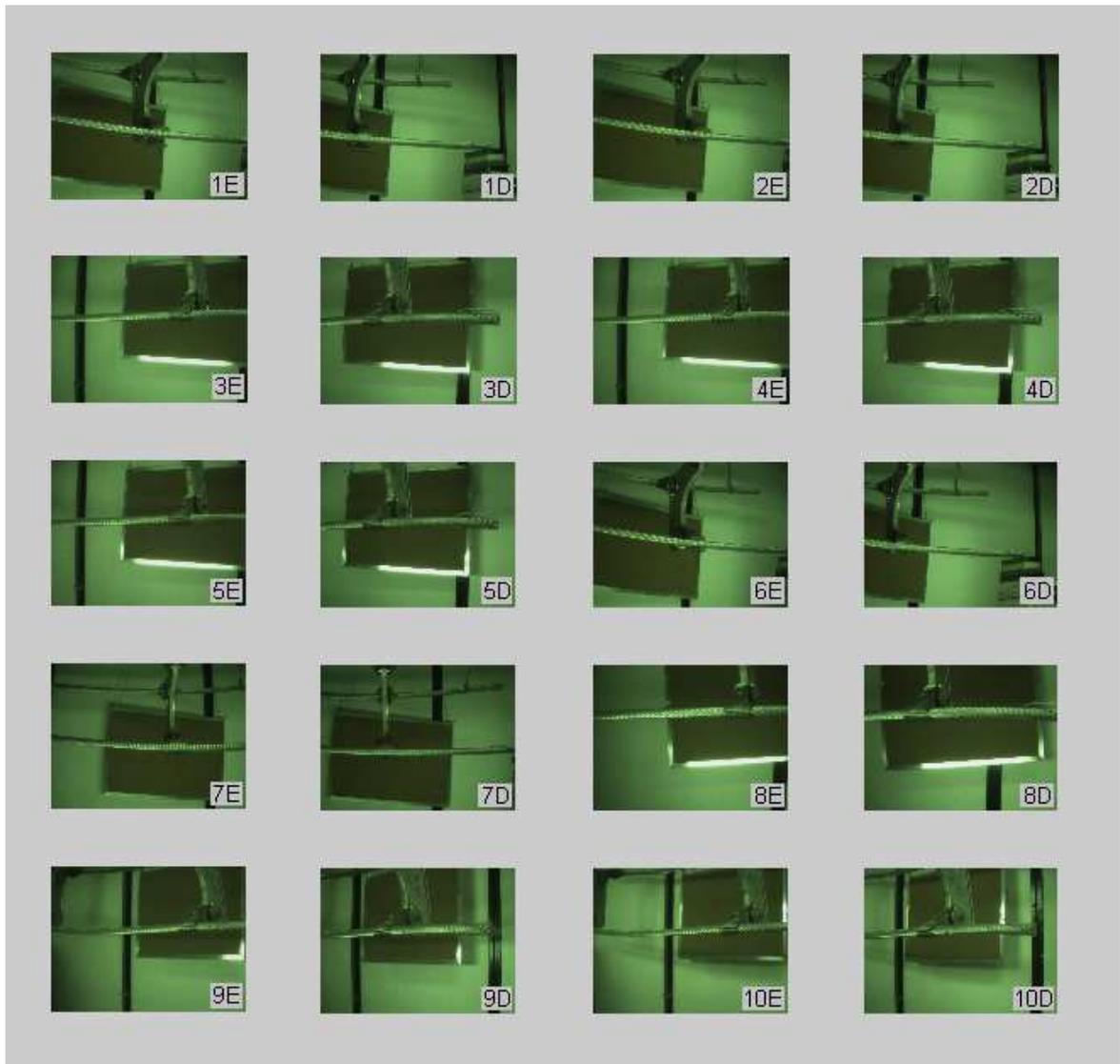


Figura 5.9: Imagens que compuseram o conjunto de testes (parte 1).

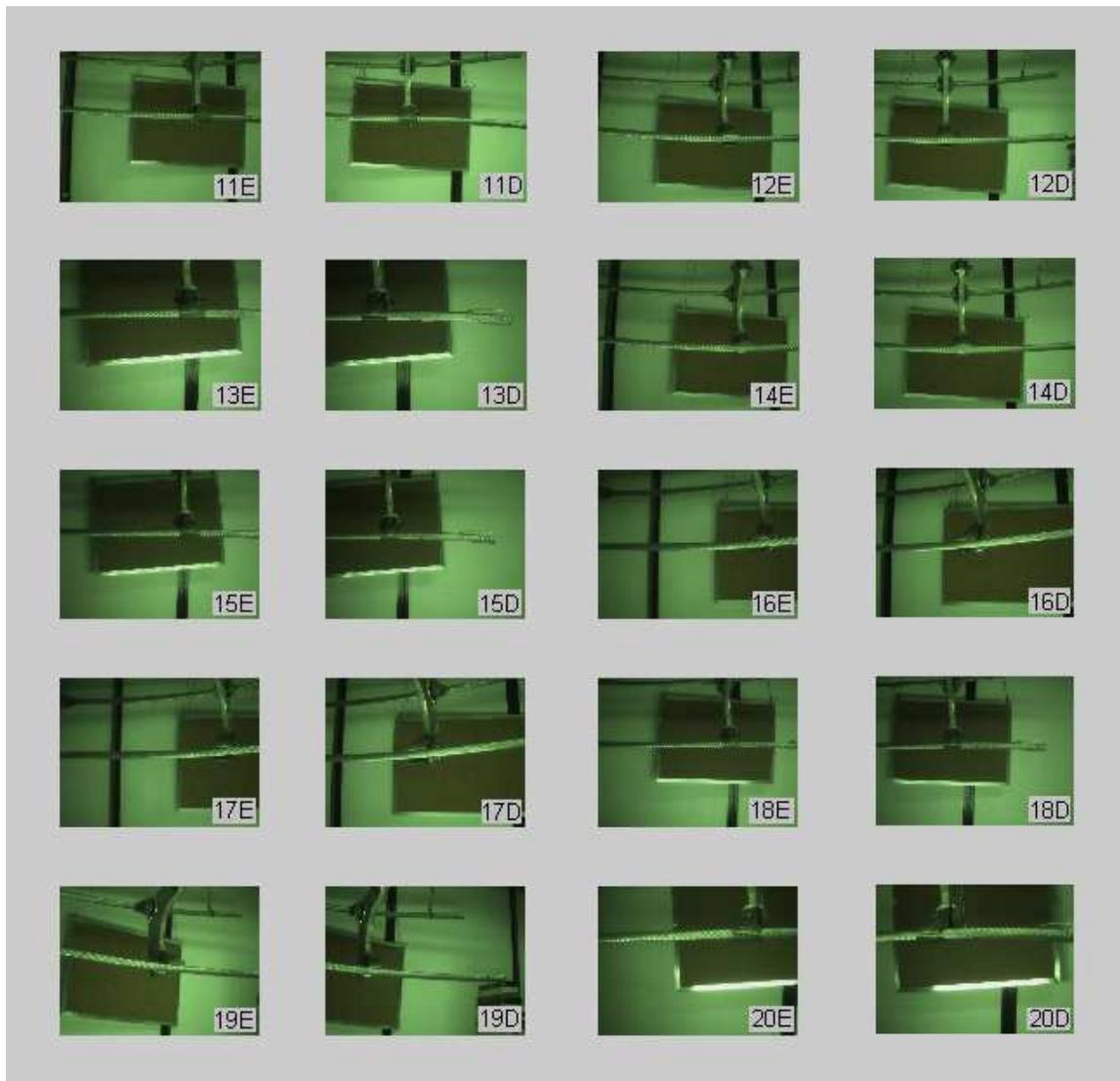


Figura 5.10: Imagens que compuseram o conjunto de testes (parte 2).

6 CONCLUSÕES E PERSPECTIVAS

A abordagem adotada neste trabalho é, do ponto de vista econômico, mais interessante do que os métodos tradicionais, uma vez que o custo de vôo de um VANT é muito inferior ao de um helicóptero tripulado. Além disso, são dispensadas as figuras do piloto e do técnico que observaria o estado dos separadores metálicos das linhas de transmissão. Sob o prisma da confiabilidade no diagnóstico o processo passa a ser mais uniforme pois o mesmo algoritmo de identificação de falhas é executado em todos os separadores que compõem a linha de transmissão.

Sob o aspecto do objetivo deste trabalho pode-se concluir que o mesmo foi atingido com restrições, pois o aproveitamento das imagens adquiridas foi não foi total (houve perdas). E sob o prisma do tempo computacional gasto desde o processamento das imagens até que os DFB do contorno do objeto pudessem ser levados à rede neural, sugere-se que o processo não seja online, mas realizado em 2 etapas separadas:

1. Aquisição de imagens (sem processamento das mesmas);
2. Análise, em solo, das imagens adquiridas.

Assim, o VANT passaria a percorrer as linhas de transmissão apenas para a aquisição de um conjunto de imagens de cada espaçador. Tais imagens seriam processadas em solo a posteriori, e em caso de necessidade de manutenção uma equipe seria deslocada para o local no qual se encontraram as garras classificadas como defeituosas.

6.1 CONTRIBUIÇÕES E PRINCIPAIS DIFICULDADES

Entre as principais contribuições deste trabalho estão:

1. Automação do processo de inspeção dos espaçadores nas linhas de transmissão, o que evitaria tempo gasto por técnicos inspetores na observação visual de cada garra metálica de cada trecho de uma linha de transmissão;
2. Minimização dos custos econômicos de inspeção, pois o custo operacional de um VANT é muito inferior ao de um helicóptero tripulado.
3. Eliminação do risco aos tripulantes do helicóptero, uma vez que o processo realizado por VANT não necessita de tripulação.

4. Uniformização dos procedimentos de inspeção, uma vez que as imagens das garras metálicas seriam processadas pelos mesmos algoritmos e não pela visão humana que pode ser subjetiva.

Na formulação de um processo automatizado de inspeção dos espaçadores, anteriormente citados, têm-se como principais dificuldades os seguintes tópicos:

- Dificuldade em se obter num sistema de posicionamento do helimodelo com a precisão/exatidão necessárias para que as imagens obtidas do objeto de inspeção sejam adequadas ao processo de classificação;
- Dificuldade em giroestabilizar as câmeras, para que as imagens não sejam distorcidas durante as manobras realizadas pelo aeromodelo;
- Uma vez realizada a aquisição de imagens, tem-se ainda a dificuldade em se realizar o processo de correspondência de pixels entre as imagens do par estéreo;
- Dificuldade em selecionar a região de interesse em ambas as imagens sem que as imagens tenham sua posição relativa alterada.

6.2 TRABALHOS FUTUROS

Após a realização deste trabalho pode-se observar que o tratamento dos seguintes tópicos poderia melhorar a qualidade do processo de identificação tratado nesta dissertação:

- Melhoramento do algoritmo de posicionamento das máscaras de *background* e *foreground* sobre a imagem para a segmentação da mesma;
- Correção das distorções radiais e tangenciais (principalmente a primeira) inseridas na imagem pelas lentes das câmeras;
- No tocante à etapa de correspondência dos *pixels* sugere-se a utilização não só das imagens das bordas do objeto mas de uma pequena região em volta das mesmas para redução das oclusões por efeito de quantificação das imagens.
- Melhoria do método de planificação das imagens 3D, uma vez que neste trabalho este procedimento teve que ser feito manualmente. Sugere-se uma busca pela melhoria na exatidão/precisão dos Pontos Principais, que teoricamente estão contidos no plano ρ que é utilizado para a definição da orientação da câmera virtual, e conseqüentemente planifica a imagem 3D numa perspectiva favorável ao processo de classificação.

REFERÊNCIAS BIBLIOGRÁFICAS

- [1] J. KATRASNIK, F. PERNUS, and B. LIKAR, “A survey of mobile robots for distribution power line inspection,” *IEEE Transactions on Power Delivery*, vol. 25, pp. 485–487, 2010.
- [2] C. WHITWORTH, “Aereo video inspection of overhead power lines,” *Power Engineering Journal*, February 2001.
- [3] S. Y. FU, Y. C. ZHANG, L. CHENG, Z. LIANG, and Z. G. HOU, “Motion based image deblur using recurrent neural network for power transmission line inspection robot,” in *International Joint Conference on Neural Network*, 2006, pp. 3854–3855.
- [4] F. CABALLERO, L. MERINO, J. FERRUZ, and A. OLLERO, “A visual odometer without 3d reconstruction for aerial vehicles. applications for building inspection,” in *IEEE International Conference on Robotics and Automation*, 2005, pp. 4673–4675.
- [5] H. V. DIJICK and V. HEIJDEN, “Object recognition with stereo vision and geometric hashing,” *Pattern Recognition Letters*, vol. 24, pp. 137–139, 2003.
- [6] M. OSTENDORP, “Innovative airborne inventory and inspection technology for electric power lines conditions assessments and defect reporting,” vol. 15, pp. 123–125, 2000.
- [7] H. LAMELA, H. GARCIA, and M. FERRERAS, “Optoelectronic instrumentation system in an autonomous unmanned aerial vehicle,” *IEEE*, vol. 7, pp. 1342–1343, 1998.
- [8] R. GONZALES and R. WOODS, *Digital Image Processing*. Prentice-Hall, 2002.
- [9] S. HAYKIN, *Redes Neurais, Princípios e Prática*. Prentice-Hall, 1999.
- [10] F. OLIVEIRA and A. GUEDES, “Automação dos processos de correspondência e orientação relativa em visão estéreo,” Ph.D. dissertation, Universidade de Brasília, Julho 2007.
- [11] F. M. A. NOGUEIRA, “Reconstrução tridimensional por visão estéreo e detecção de oclusões,” Ph.D. dissertation, Universidade Estadual de Campinas. Faculdade de Engenharia Elétrica e Computação, Dezembro 2007.
- [12] R. GONZALES, *Digital Image Processing Using MATLAB*. Prentice-Hall, 2004.
- [13] A. LEVIN and Y. LISCHINSKI, D. and WEISS, “Spectral matting,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, pp. 1699–1712, 2008.

- [14] J. Wang and M. Cohen, “an interactive optimization approach for unified image segmentation and matting,” in *Proceedings of the Tenth IEEE International Conference on Computer Vision - Volume 2*, 2005, pp. 936–943.
- [15] M. GALO, “Automação dos processos de correspondência e orientação relativa em visão estéreo,” Ph.D. dissertation, Universidade Estadual de Campinas. Faculdade de Engenharia Elétrica e Computação, Dezembro 2003.
- [16] U. R. DHOND and J. K. AGGARWAL, “Structure from stereo,” *IEEE transactions on systems, man and cybernetics*, vol. 19, pp. 1489–1492, 1989.
- [17] J. BATISTA, H. ARAÚJO, and A. ALMEIDA, “Interactive multistep explicit camera calibration,” *IEEE Trans. On Robotics and Automation*, vol. 15, pp. 897 – 917, 1999.
- [18] M. GALO and C. L. TOZZI, “Inclusão de injunções epipolares na solução do problema de correspondência,” in *SIBGRAPI '97. - X Brazilian Symposium of Computer Graphic and Image Processing*, 1997.
- [19] D. JONES, “Image signal process,” *Proc.-Vis. Image Signal Process.*, vol. 147, pp. 157–158, 2000.
- [20] O. FAUGERAS and Q. LUONG, *The Goemetry of multiple Images*. MIT Press, 2001.
- [21] A. FUSIELLO, E. TRUCO, and A. VERRI, “Rectification with unconstrained stereo geometry,” in *British Machine Vision Conference*, 1998, pp. 1167–1186.
- [22] R. Dougherty and R. Giardina, *Mathematical Methods for Artificial Intellingence and Autonomous Systems*. Prentice-Hall International Editions, 1988.
- [23] Z. KOVÁCS, *Redes Neurais Artificiais*. Editora da Física, 2006.
- [24] S. HAYKIN, *Redes Neurais, Princípios e Prática*. Bookman, 2001.
- [25] E. M. IYODA, “Inteligência computacional no projeto automático de redes neurais híbridas e redes neurofuzzy heterogêneas,” Ph.D. dissertation, Universidade Federal de Campinas, Janeiro 2000.
- [26] D. JONES, J. Watson, and K. J. Brown, “Comparison of hot rolled steel mechanical property prediction models using linear multiple regression, non-linear multiple regression and non-linear artificial neural networks,” *Ironmaking and Steelmaking*, vol. 32, pp. 435–442, 2005.
- [27] A. LEVIN, D. LISCHINSKI, and Y. WEISS, “A closed form solution to natural image matting,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, pp. 228–242, 2006.

- [28] A. LEVIN and Y. WEISS, “Learning to combine bottom-up and top-down segmentation,” *Int. J. Comput. Vision*, vol. 81, pp. 105–118, 2009.

APÊNDICES

...0000000000000000000000000070000000000070000000000...
...007000000000000700000000000700000000007000000...
...0000070000000000007000000000070000000000700...
...00012112112121000000000000000111010000000000...
...00000006676676776700000001221220000070000012...
...2222222122111111223222000001222222222222222...
...22112110000000013444443444434454444444444534...
...34343434343007000070000700007000070000000000...
...00000000000100000000000000000000010000000000...
...00000000001000013444344434443444344434443400...
...00700000700000700000700000700000700000700000...
...70000070000070000001121210070000700070000700...
...00701333333332333222220000007000700007000...
...0700070000700007000700007000070007000000000...
...000000006767670000000000000000000000010010001...
...00100000000001210107677712121001010001055670...
...0000000066666666666666666666666666666666666...
...666666666666666666666666666666666666666666...
...666666666666666666666666666666666666666666...
...666666666666666666666666666666666666666666...
...666666666666666666666666666666666666666666...
...666666666666666666666666666666666666666666...
...666666666666666666666666666666666666666666...
...4444444444444444444444444444444444444444444...
...4444444444444444444444444444444444444444444...
...4444444444444444444444444444444444444444444...
...4444444444444444444444444444444444444444444...
...4444444444444444444444444444444444444444444...
...4444444444444444444444444444444444444444444...
...4444444444444444444444444444444444444444444...
...4444444444444444444444444444444444444444444...
...4323234444444444444444444444444444444444444...
...4444444444444444444444444444444444444444444...
...4444444444444444444444444444444444444444444]

B.1 FUNÇÃO PRINCIPAL

```
% principal.m Script principal
% Aloca o caminho das imagens e o nome das imagens, as máscaras de
% foreground e background e inicia a execução dos demais scripts.

clear all, clc
close all

% Localização dos diretórios que contém as imagens
fileFolder = fullfile('C:\Master\ok10_a15');

% Criação da lista com o nome dos arquivos
dirOutputL = dir(fullfile(fileFolder,'imL*.bmp'));
dirOutputR = dir(fullfile(fileFolder,'imR*.bmp'));

% Vetor contendo os nomes dos arquivos
fileNamesL = {dirOutputL.name}';
fileNamesR = {dirOutputR.name}';

% Loop principal para aquisição das cadeias codificadas e descritores de
for k = 1:(length(fileNamesL)+length(fileNamesR))/2
    clc;
    disp('Iteração: ')
    disp(k);
    disp('No total de: '),disp((length(fileNamesL)+length(fileNamesR))/2),
    disp('-----');
    % Obtenção da estrutura com as imagens

    datL = processFile(fileNamesL{k});
    datR = processFile(fileNamesR{k});

    %coordenadas da Área de Interesse na imagem.
    imageL(k).coord_AI=datL.coord_AI;
    imageR(k).coord_AI=datR.coord_AI;

    %imagem em si.
    imageL(k).imagem=datL.imagem;
    imageR(k).imagem=datR.imagem;

    % imagens com detecção de bordas.
    imageL(k).bordas=datL.perimetro;
    imageR(k).bordas=datR.perimetro;

    %máscara utilizada
    cadeiaL(k).mask=datL.mask;
    cadeiaR(k).mask=datR.mask;

    % Cadeia codificada.
    cadeiaL(k).fcc = datL.cadeia.fcc;
    cadeiaR(k).fcc = datR.cadeia.fcc;

    % Cadeia normalizada para rotação.
```

```
cadeiaL(k).diff = datL.cadeia.diff;
cadeiaR(k).diff = datR.cadeia.diff;

% Cadeia normalizada quanto ao ponto de partida.
cadeiaL(k).mm = datL.cadeia.mm;
cadeiaR(k).mm = datR.cadeia.mm;

% Cadeia normalizada para rotação e ponto de partida.
cadeiaL(k).diffmm = datL.cadeia.diffmm;
cadeiaR(k).diffmm = datR.cadeia.diffmm;

% Ponto de partida.
cadeiaL(k).x0y0 = datL.cadeia.x0y0;
cadeiaR(k).x0y0 = datR.cadeia.x0y0;

end

save cadeiaL;
save cadeiaR;
```

B.2 TRATAMENTO

```
function data = processFile(imageFile)
%batchProcessFile Realiza a leitura da imagem, a obtenção da região de
% interesse (ROI) localizada no centro da imagem, a aplicação das
% restrições (máscaras) e a execução do algoritmo de processamento das
% imagens.
%
% DATA = processFile(IMAGEFILE, IMASKF, IMASKB)
% Recebe o nome da imagem (IMAGEFILE), em vez dos dados, as máscaras de
% foreground e background (IMASKF, IMASKB) e retorna na estrutura DATA a
% imagem inicial com o resultado da segmentação sobreposto e a cadeia
% codificada do contorno do cabo segmentado pela execução do script ccf.

% Caminho do arquivo de imagem (cabo OK e cabo Defeito)

imageFilePath= fullfile('C:\Master\ok10_a15',imageFile);

% Leitura do arquivo
A = imread(imageFilePath);

%% Dimensões das imagens
imageW=321;
imageH=251;
%%
imshow(A);
%Obtenção d
%[xmin,ymin,width,height]
rect=getrect;
rect=round(rect);

%Coordenadas do Quadrado;
xmin=rect(1);
xmax=rect(1)+rect(3);
ymin=rect(2);
ymax=rect(2)+rect(4);

%coordenadas do centro da figura;
xc=(xmin+xmax)/2;
yc=(ymin+ymax)/2;

%Area de Interesse
AI=[xc-160, yc-125,imageW-1,imageH-1];
%Coordenadas da Área de Interesse na imagem plena.
coord_AI=[AI(1), AI(1)+AI(3)+1, AI(2), AI(2)+AI(4)+1];

% Pega a região de interesse (ROI) no centro da imagem
Icrop = imcrop(A,AI);

% Camadas dividindo
iR = Icrop(:, :, 1);
iG = Icrop(:, :, 2);
iB = Icrop(:, :, 3);

%% Foreground
```

```

Imaskf=zeros(imageH,imageW);
%retângulo do cabo
imin = round((imageH-rect(4))/2+2);
imax = round((imageH+rect(4))/2-2);
jmin = round((imageW-rect(3))/2.2);
jmax = round((imageW+rect(3))/1.9);
for i=max(1,imin):min(imageH,imax)
    for j=max(1,jmin):min(imageW,jmax)
        Imaskf(i,j)=1;
    end
end
%Retângulos do cabo à esquerda
imin = round((imageH-2*rect(4))/2-5);
imax = round((imageH+2*rect(4))/2-5);
jmin = round((imageW-rect(3))/2-8*rect(4));
jmax = round((imageW-rect(3))/2-4*rect(4));
for i=max(1,imin):min(imageH,imax)
    for j=max(1,jmin):min(imageW,jmax)
        Imaskf(i,j)=1;
    end
end
%Retângulos do cabo à direita
imin = round((imageH-2*rect(4))/2);
imax = round((imageH+2*rect(4))/2);
jmin = round((imageW+rect(3))/2+4*rect(4));
jmax = round((imageW+rect(3))/2+8*rect(4));
for i=max(1,imin):min(imageH,imax)
    for j=max(1,jmin):min(imageW,jmax)
        Imaskf(i,j)=1;
    end
end
%% Background
Imaskb=zeros(imageH,imageW);
%Retângulo de baixo
imin = round((imageH+rect(4)+rect(3))/2-10);
imax = round((imageH+rect(4))/2+rect(3)-30);
jmin = round((imageW-2*rect(3))/2);
jmax = round((imageW+2*rect(3))/2);
for i=max(1,imin):min(imageH,imax)
    for j=max(1,jmin):min(imageW,jmax)
        Imaskb(i,j)=1;
    end
end
%Retângulo de cima da garra
imin = round((imageH-12*rect(4))/2+5);
imax = round((imageH-4*rect(4))/2);
jmin = round((imageW-rect(3)+rect(4))/2+5);
jmax = round((imageW+rect(3)-rect(4))/2);
for i=max(1,imin):min(imageH,imax)
    for j=max(1,jmin):min(imageW,jmax)
        Imaskb(i,j)=1;
    end
end
%Retângulo logo abaixo da garra
imin = round((imageH+4*rect(4))/2);%era 2
imax = round((imageH+8*rect(4))/2);%era 2
jmin = round((imageW-rect(3)+rect(4))/2+5);
jmax = round((imageW+rect(3)-rect(4))/2);
for i=max(1,imin):min(imageH,imax)

```

```

        for j=max(1,jmin):min(imageW,jmax)
            Imaskb(i,j)=1;
        end
    end
end

%quadrados à esquerda
imin = round((imageH-22*rect(4))/2);
imax = round((imageH-12*rect(4))/2);
jmin = round((imageW-rect(3))/2-10*rect(4));
jmax = round((imageW-rect(3))/2-4*rect(4));
for i=max(1,imin):min(imageH,imax)
    for j=max(1,jmin):min(imageW,jmax)
        Imaskb(i,j)=1;
    end
end

%quadrados à direita
imin = round((imageH-22*rect(4))/2);
imax = round((imageH-12*rect(4))/2);
jmin = round((imageW+rect(3))/2+4*rect(4));
jmax = round((imageW+rect(3))/2+10*rect(4));
for i=max(1,imin):min(imageH,imax)
    for j=max(1,jmin):min(imageW,jmax)
        Imaskb(i,j)=1;
    end
end

%%
Imaskf=logical(Imaskf);
Imaskb=logical(Imaskb);
%-----
% Geração da imagem para processamento (imagem inicial com as máscaras).
iR(Imaskf) = 255;
iG(Imaskf) = 255;
iB(Imaskf) = 255;

iR(Imaskb) = 0;
iG(Imaskb) = 0;
iB(Imaskb) = 0;

% Concatenação das camadas RGB para início do processamento.
imagem = cat(3, iR, iG, iB);

% Script para processamento de imagens.
executa;

% Resultados.
data.coord_AI=round(coord_AI);%Coordenadas da Região de Interesse.
data.res = result; % Imagem com cabo segmentado sobreposto.
data.cadeia = ccf(im_final); % Estrutura com as cadeias.
data.perimetro=perimetro;
data.imagem=Icrop;
data.mask=imagem;

```

B.3 SEGMENTAÇÃO DA IMAGEM

% e uma série de operações morfológicas para obtenção de uma imagem

```

% binária contendo o cabo segmentado.

% Parâmetros para algoritmo de obtenção do "alpha matting"
win_size = 2;
levels_num = 4;
active_levels_num = 3;
thr_alpha = 0.3;
epsilon = 0.1^5;
%sig=0.1^5;

% Execução do algoritmo para obtenção da imagem alpha (Levin et al.)
runMatting2

% Binarização do alpha matting gerado.
level = graythresh(alpha);
bw = im2bw(alpha, level);
%figure, imshow(bw); title('bw');

% Conecta pixels desconectados 5x (conectividade 4).
bw = bwmorph(bw, 'bridge', 10);%era 5
%figure, imshow(bw); title('Bridge');

% Remove de componentes desconectados < 70 pixels (conectividade 8).
BWopen = bwareaopen(bw,70);

% Realiza operação morfológica de fechamento ("Closing").
BWclose = bwmorph(BWopen,'close');
%figure, imshow(BWclose); title('Removed');

% Realiza operação de dilatação a partir dos elementos estruturais
% gerados.
%im_top = imtophat(bw, strel('disk', 10));
%im_dilate = imdilate(bw, strel('diamond', 3));
%im_erode = imerode(im_dilate, strel('diamond', 2));
se90 = strel('line', 3, 90);
se0 = strel('line', 3, 0);
im_dilate = imdilate(BWclose, [se90 se0]);
%figure, imshow(im_dilate); title('Dilate');

% Preenche buracos na imagem binária (conjuntos de pixels do background
% que não podem ser alcançados a partir do preenchimento do background
% limitado pelas bordas da imagem.
im_fill = imfill(im_dilate, 'holes');
%im_fill = bwmorph(im_dilate, 'fill');
%figure, imshow(im_fill); title('filled');

% Suavização das bordas do cabo pela realização da operação de erosão e
% dilatação usando o elemento estrutural criado pela função strel().
seL1 = strel('line',5,0);
seL2 = strel('line',5,90);
im_final = imerode(im_fill,[seL1 seL2]);
im_final = imdilate(im_final,[seL1 seL2]);
%figure, imshow(im_final); title('Final');

% Obtenção do contorno do cabo segmentado sobreposto na imagem inicial
% com o uso da função bwperim().
perimetro = bwperim(im_final);

result = im_final; %

```

B.4 MATTING

```
%runMatting2.m
% Script principal criado por Levin et al. para a geração do alpha matting
% a partir de uma abordagem analítica (Closed Form Solution) do problema,
% contendo algumas modificações.
%
%
% Análise dos parâmetros de entrada
if (~exist('thr_alpha','var'))
    thr_alpha=[];
end
if (~exist('epsilon','var'))
    epsilon=[];
end
if (~exist('win_size','var'))
    win_size=[];
end

if (~exist('levels_num','var'))
    levels_num=1;
end
if (~exist('active_levels_num','var'))
    active_levels_num=1;
end

% Leitura da imagem a ser processada
I=double(Icrop)/255;
%I=double(imread(img_name))/255;

% Leitura da imagem com a máscara de foreground e background (restrições)
mI=double(imagem)/255;
%MI=double(imread(scribs_img_name))/255;
%MI=double(scribs_img_name)/255;

consts_map=sum(abs(I-mI),3)>0.001;
if (size(I,3)==3)
    consts_vals=rgb2gray(mI).*consts_map;
end
if (size(I,3)==1)
    consts_vals=mI.*consts_map;
end

% Execução do algoritmo
alpha=solveAlphaC2F(I,consts_map,consts_vals,levels_num, ...
    active_levels_num,thr_alpha,epsilon,win_size);

%figure, imshow(alpha);
drawnow;
%[F,B]=solveFB(I,alpha);
%figure, imshow([F.*repmat(alpha,[1,1,3]),B.*repmat(1-alpha,[1,1,3])])

function alpha=solveAlpha(I,consts_map,consts_vals,varargin)

    [h,w,c]=size(I);
    img_size=w*h;

    A=getLaplacian1(I,consts_map,varargin{:});
```

```

D=spdiags(consts_map(:),0,img_size,img_size);
lambda=100;
x=(A+lambda*D)\(lambda*consts_map(:).*consts_vals(:));

alpha=max(min(reshape(x,h,w),1),0);

function alpha=solveAlphaC2F(I,consts_map,consts_vals,...
    ...levels_num,active_levels_num,varargin);

if (length(varargin)>0)
    if (~isempty(varargin{1}))
        thr_alpha=varargin{1};
    end
end

if (~exist('thr_alpha','var'))
    thr_alpha=0.02;
end
erode_mask_w=1;
active_levels_num=max(active_levels_num,1);
if (levels_num>1)
    sI=downSmpIm(I,2);
    s_consts_map=round(downSmpIm(double(consts_map),2));
    s_consts_vals=round(downSmpIm(double(consts_vals),2));
    s_alpha=solveAlphaC2F(sI,s_consts_map,s_consts_vals,levels_num-1,...
        min(levels_num-1,active_levels_num),varargin{:});
    alpha=upSampleAlphaUsingImg(s_alpha,sI,I,varargin{2:end});
    talpha=alpha.*(1-consts_map)+consts_vals;
    consts_map=min(consts_map+imerode((alpha>=(1-thr_alpha)),...
        ...ones(erode_mask_w*2+1))+imerode((alpha<=(thr_alpha)),...
        ...ones(erode_mask_w*2+1)),1);
    consts_vals=round(talpha).*consts_map;
    %figure, imshow([consts_map,alpha])
end
if (active_levels_num>=levels_num)
    alpha=solveAlpha(I,consts_map,consts_vals,varargin{2:end});
end

function [A,A1]=getLaplacian1(I,consts,epsilon,win_size)

if (~exist('epsilon','var'))
    epsilon=0.0000001;
end
if (isempty(epsilon))
    epsilon=0.0000001;
end
if (~exist('win_size','var'))
    win_size=1;
end
if (isempty(win_size))
    win_size=1;
end

neb_size=(win_size*2+1)^2;
[h,w,c]=size(I);
n=h; m=w;

```

```

img_size=w*h;
consts=imerode(consts,ones(win_size*2+1));

indsM=reshape([1:img_size],h,w);

tlen=sum(sum(1-consts(win_size+1:end-win_size,...
...win_size+1:end-win_size)))*(neb_size^2);

row_inds=zeros(tlen,1);
col_inds=zeros(tlen,1);
vals=zeros(tlen,1);
len=0;
for j=1+win_size:w-win_size
    for i=win_size+1:h-win_size
        if (consts(i,j))
            continue
        end
        win_inds=indsM(i-win_size:i+win_size,j-win_size:j+win_size);
        win_inds=win_inds(:);
        winI=I(i-win_size:i+win_size,j-win_size:j+win_size,:);
        winI=reshape(winI,neb_size,c);
        win_mu=mean(winI,1)';
        win_var=inv(winI'*winI/neb_size-win_mu*win_mu'+epsilon/neb_size*eye(c));

        winI=winI-repmat(win_mu',neb_size,1);
        tvals=(1+winI*win_var*winI')/neb_size;

        row_inds(1+len:neb_size^2+len)=reshape(repmat(win_inds,1,neb_size),...
...neb_size^2,1);
        col_inds(1+len:neb_size^2+len)=reshape(repmat(win_inds',neb_size,1),...
...neb_size^2,1);
        vals(1+len:neb_size^2+len)=tvals(:);
        len=len+neb_size^2;
    end
end

vals=vals(1:len);
row_inds=row_inds(1:len);
col_inds=col_inds(1:len);
A=sparse(row_inds,col_inds,vals,img_size,img_size);

sumA=sum(A,2);
A=spdiags(sumA(:,0),img_size,img_size)-A;

return

```

B.5 PREPARATIVOS PARA OBTENÇÃO DA CADEIA DIRECIONAL

```

function c = ccf(im_final)
%ccf.m Gera a cadeia codificada
%
% C = ccf(IM_FINAL)
% Recebe a imagem binária IM_FINAL com o cabo segmentado, obtém seu
% contorno e retorna a estrutura C contendo as cadeias.

```

```

% A partir da imagem recebida, obtém as coordenadas de cada ponto dos
% contornos dos objetos na imagem por meio da função bwboundaries() da
% toolbox e em seguida o maior dos contornos obtidos, o cabo.
%B = boundaries(im_final); % Coordenadas das fronteiras
[B,L] = bwboundaries(im_final,'noholes');
%B Possui as fronteiras de cada objeto constante na imagem.
%L possui a própria imagem.
d = cellfun('length', B);
[max_d, k] = max(d);
b = B{k}; % Maior fronteira

% Gera a partir das coordenadas de cada ponto da fronteira do cabo, obtidas
% pela função bwboundaries(), a imagem binária do contorno.
[M N] = size(im_final);
bim = bound2im(b, M, N, min(b(:, 1)), min(b(:, 2)));
%figure, imshow(bim); title('bim');

% Obtenção das cadeias codificadas da imagem binária b com o contorno do
% cabo segmentado.
c = fchcode(b);

```

B.6 CADEIA DIRECIONAL

```

function c = fchcode(b,conn,dir)
%FCHCODE computa o codigo da cadeia direcional a uma borda
%C = FCHCODE(B) computa um codigo de cadeia de Freeman 8-conectada de
% um par de coordenadas bidimensionais contidas em B, um array np-por-2.
%C é uma estrutura com os seguintes campos:

% c.fcc = codigo de cadeia de Freeman(1-por-np)
% c.diff = Primeira diferença do codigo c.fcc (1-por-np)
% c.mm = Inteiro de menor magnitude de c.fcc (1-por-np)
% c.diffmm = Primeira diferença do código c.mm (1-por-np)
% c.x0y0 = coordenadas onde o codigo começa (1-por-2)

%C = FCHCODE(B, CONN) produz as mesmas saidas acima, mas
% com a conectividade do codigo especificada em CONN. CONN pode
% ser 8 para um codigo de cadeia 8-conectada, ou pode ser 4 para
% um codigo de cadeia 4-conectada. Especificar CONN=4 é somente válido
% se a sequencia de entrada ,B, contem transições dos valores restritos
% 0,2,4,6

%C = FCHCODE(B,CONN,DIR) produz as mesmas saida acima, mas, vai alem,
% especifica a direção do codigo desejada.Valores para DIR podem ser:

% 'same' É o padrão.A mesma ordem da sequencia de pontos de B.
% 'reverse' O codigo de saida é computado na direção oposta da
% direção dos pontos em B.O ponto de inicio para cada DIR é o mesmo

% Assume-se que os elementos de B correspondem a um denso pixel,
% totalmente conectado, numa borda fechada. B nao pode conter pares de
% coordenadas duplicadas, exceto a primeira e ultima posições, o que é uma
% representação comum em programas que traçam bordas.

%REPRESENTAÇÃO DO CODIGO DE CADEIA DE FREEMAN
%A tabela abaixo mostra os codigos de cadeia de Freeman e seus
%correspondentes em deltax e deltay. Uma cadeia-8 é convertida para
%uma cadeia-4 se:(1)SE CONN=4; e (2)SOMENTE TRANSIÇÕES 0,2,4,6 OCORREREM

```

%NO CODIGO-8. Perceba que dividindo 0,2,4,6 por 2 tem-se o codigo-4.

%	DELTA X	DELTA Y	8-CODE	CORRESP	4-CODE
%	0	1	0		0
%	-1	1	1		
%	-1	0	2		1
%	-1	-1	3		
%	0	-1	4		2
%	1	-1	5		
%	1	0	6		3
%	1	1	7		

% A formula $z = 4 * (\text{deltax} + 2) + (\text{deltay} + 2)$ nos fornece a seguinte
% sequencia correspondente ás linhas da tabela respectivamente:
% $z = 11, 7, 6, 5, 9, 13, 14, 15$. Estes valores podem ser utilizados como indices
% na tabela, aumentando o rendimento do codigo implementado. Esta nao é uma
% formula unica, mas é baseada nos menores inteiros potencias de 2 (2 e 4).

%PRELIMINARES

```
if nargin == 1
    dir = 'same';
    conn = 8;
elseif nargin == 2
    dir = 'same';
elseif nargin == 3
    %nao se faz nada
else
    error('Numero de entradas incorreto')
end
[np, nc] = size(b);
if np < nc
    error('B deve ser do tamanho np-por-2')
end
```

% Se as coordenadas do primeiro e ultimo ponto sao as mesmas, entao
% elimina-se o ultimo ponto.

```
if isequal(b(1,:), b(np,:))
    np = np - 1;
    b = b(1:np,:);
end
```

%Constroi-se a tabela de codigos utilizando os indices da formula z:
C(11)=0;C(7)=1;C(6)=2;C(5)=3;C(9)=4;
C(13)=5;C(14)=6;C(15)=7;

%FINAL DAS PRELIMINARES

```
% COMEÇO DO PROCESSO
x0 = b(1,1);
y0 = b(1,2);
c.x0y0 = [x0,y0];
```

%Tenha certeza de que as coordenadas estao organizadas sequencialmente:
%Obtenha deltax e deltay entre pontos sucessivos em b. A ultima linha de a
%é a primeira linha b.
a = circshift(b, [-1 ,0]);
%DEL = a-b é uma matriz nr por 2 na qual as linhas contem o deltax e deltay
%entre pontos sucessivos em b. Os 2 componentes na k-ésima linha da matriz
%DEL são deltax e deltay entre os pontos (xk, yk) e (xk+1, yk+1). A ultima
%linha de DEL contem o deltax e deltay entre (xnr, ynr) e (x1, y1), isto é
%entre o ultimo e primeiro pontos de b.

```

DEL = a-b;

%Se o valor de abs de um dos componentes (ou ambos) do par (deltax,deltay)
%é maior que 1, então pela definição , a curva é quebrada ( ou os pontos
%estão fora de ordem), e o programa termina.
if any(abs(DEL(:,1))>1) || any(abs(DEL(:,2))>1);
    error('A curva de entrada está quebrada ou os pontos fora de ordem')
end
%Cria-se um vetor simples indexado usando a formula descrita abaixo
z = 4*(DEL(:,1) + 2) + (DEL(:,2) + 2);

%Use a o indice para mapear para uma tabela.A sequencia é a cadeia
%8-codificada de Freeman, organizada em um array 1 por np.
fcc = C(z);

%Confira se a direção da sequencia do codigo necessita ser revertida.
if strcmp(dir,'reverse')
    fcc = coderev(fcc);
end

% Se a conectividade é especificada, cheque que todos os componentes de fcc
% sao 0,2,4 ou 6.
if conn == 4
    val = find(fcc == 1 || fcc == 3 || fcc == 5 || fcc == 7);
    if isempty(val)
        fcc = fcc./2;
    else
        warning('O codigo 4-conectado nao pode ser satisfatorio')
    end
end

%plotagem auxiliar (retirar após debug);
% centro=[425,100];
% plot_chain(fcc,centro,'fcc');pause;

%Cadeia codificada de Freeman para a estrutura de saida
c.fcc = fcc;

% Obtenha a primeira diferença de fcc
c.diff = codediff(fcc,conn);
% centro=[950,150];
% plot_chain(c.diff,centro,'diff');pause;

% Obter o codigo do inteiro de menor magnitude
c.mm = minmag(fcc);
% centro=[425,100];
% plot_chain(c.mm,centro,'minmag');pause;

% Obter a primeira diferença de c.mm
c.diffmm = codediff(c.mm,conn);
% centro=[950,150];
% plot_chain(c.diffmm,centro,'diffmm');pause;

%-----%

function cr = coderev(fcc)
%Percorrer a sequencia da cadeia codificada de Freeman 8-conectada fcc na
%direção oposta, mudando os valores de cada segmento de codigo. O ponto de
%partida nao é mudado. fcc é uma array 1 por np.
% Mova o array da esquerda para a direita. Isto redefine o ponto de partida
% como o ultimo ponto e reverte a ordem de "viagem" no codigo.

```

```

cr = fliplr(fcc);

%Depois, obtenha os novos valores do código percorrendo-o na direção
%oposta.(0 vira 4, 1 vira 5,...,5 vira 1, 6 vira 2, e 7 vira 3).
ind1 = find(0 <= cr & cr <= 3);
ind2 = find(4 <= cr & cr <= 7);
cr(ind1) = cr(ind1) + 4;
cr(ind2) = cr(ind2) - 4;

%-----%

function z = minmag(c)
%MINMAG encontra o inteiro de menor magnitude de uma cadeia codificada.
%Z = MINMAG(C) encontra o inteiro de menor magnitude numa dada cadeia
%codificada de Freeman 4 ou 8-conectada, C. Assume-se que o código é um
%array 1 por np.
% O inteiro de menor magnitude deve começar com min(c), mas pode haver mais
% de um valor. Encontre todos.
I = find(c == min(c));
% e desloque cada um para a esquerda tal que comece com min(c)
J = 0;
A = zeros(length(I),length(c));
for k = I;
    J = J + 1;
    A(J,:) = circshift(c,[0 -(k-1)]);
end
% A matriz A contém todos os possíveis candidatos para o inteiro de menor
% magnitude. Começando com a segunda coluna, sucessivamente encontre os
% mínimos de cada coluna de A. O número de candidatos diminuem com os
% movimentos para a direita em A. Isto é refletido nos elementos de J.
% Quando length(J) = 1, um candidato sobra. Este é o inteiro de menor
% magnitude.
[M,N] = size(A);
J = (1:M)';
for k = 2:N
    D(1:M,1) = Inf;
    D(J,1) = A(J,k);
    amin = min(A(J,k));
    J = find(D(:,1) == amin);
    if length(J) == 1
        z = A(J,:);
        return
    end
end
end

%-----%

function d = codediff(fcc,conn)
%CODEDIFF computa a primeira diferença da cadeia codificada.
% D = CODEDIFF(FCC) computa a primeira diferença do código, FCC. O código
% FCC é tratado como uma sequência circular, então o último elemento de D é
% a diferença entre o último e o primeiro elementos de FCC. O código de
% entrada é um vetor 1 por np.
%A primeira diferença é encontrada contando-se o número de mudanças de
%direção ( no sentido anti-horário) que separa dois elementos adjacentes do
%código.
sr = circshift(fcc,[0, -1]);%desloca a entrada para a esquerda em uma posição
delta = sr - fcc;
d = delta;
I = find(delta < 0);

```

```

type = conn;
switch type
    case 4
        d(I) = d(I) + 4;
    case 8
        d(I) = d(I) + 8;
end

function plotsSF(coef,cadeia,imagem,num_harm)
%% Plota as formas decompostas em Séries de Fourier (SF).
%%parâmetros: Recebe os coeficientes de Fourier, sua respectivas cadeias
%%direcionais, qual das imagens a ser plotada e o nº de harmônicas.

c=coef;

if num_harm>size(c{imagem}.A,2)
    disp('Não existem Harmônicas suficientes');return;
end

% Comprimento da cadeia codificada
K = length(cadeia(imagem).mm);

T=0;
t(1)=0;
for i=2:K
    %Comprimento de cada arco;
    deltat(i-1)= 1+((sqrt(2)-1)/2)*(1-(-1)^(cadeia(imagem).mm(i-1)));
    %Comprimento acumulado do arco;
    t(i)=t(i-1)+deltat(i-1);
    % Comprimento total do arco
    T=T+(1+((sqrt(2)-1)/2)*(1-(-1)^(cadeia(imagem).mm(i))));
end

%Série de Fourier implementada
%%
for p=1:K
    x(p)=c{imagem}.A0;
    y(p)=c{imagem}.C0;
    for n=1:num_harm
        x(p)= x(p)+ c{imagem}.A(n)*cos(2*n*pi*t(p)/T) + ...
            ...c{imagem}.B(n)*sin(2*n*pi*t(p)/T);

        y(p)= y(p)+ c{imagem}.C(n)*cos(2*n*pi*t(p)/T) + ...
            ...c{imagem}.D(n)*sin(2*n*pi*t(p)/T);
    end
end
end
plot(x,y,'b-');grid;
xlabel('Eixo x'), ylabel('Eixo y'),title(num_harm), axis('equal');

```

B.7 COORDENADAS TRIDIMENSIONAIS

```

%-----
function xyz_cena=reconsti(ij_imag1,P1,ij_imag2,P2,k_linhas,k_colunas)

```

```

xyz_cena=reconsti(ij_imagl,P1,ij_imag2,P2,k_linhas,k_colunas)
%
%Efetua o calculo das tres coordenadas dos pontos de uma cena
%arbitraria, a partir das posicoes dos pixels correspondentes
%em duas imagens distintas da mesma. Cada linha em ij_imagl e
%em ij_imagr deve corresponder 'as coordenadas i e j do pixel
%de um dos pontos a serem reconstituídos. P1 e P2 sao as matrizes
%de transformacao perspectiva das cameras que geraram cada uma
%das imagens. A matriz xyz_cena retornada pela funcao contem
%em cada linha as coordenadas x,y,z de um unico ponto.
%k_linhas e k_colunas: valores somados às posições de cada pixel das imagens retificadas
%
%          calculadas, com o objetivo de evitar índices nulos ou negativos
%
%          na representação matricial (estes valores serão subtraídos
%
%          dos índices i,j durante o processo de reconstituição da cena a
%
%          partir das imagens stereo).

ij_imagl=[ij_imagl(:,1)-k_linhas ij_imagl(:,2)-k_colunas];
ij_imagr=[ij_imag2(:,1)-k_linhas ij_imag2(:,2)-k_colunas];

k=size(ij_imagl);

if((k(1,2)~=2) | (k(1,1)~=size(ij_imagr,1)))
    disp('Entradas invalidas');
else
    k=k(1,1);
    xyz_cena=zeros(k,3);
    for j=1:k
        A(1,1:3)=P1(1,1:3)-ij_imagl(j,2)*P1(3,1:3);
        A(2,1:3)=P1(2,1:3)-ij_imagl(j,1)*P1(3,1:3);
        A(3,1:3)=P2(1,1:3)-ij_imagr(j,2)*P2(3,1:3);
        A(4,1:3)=P2(2,1:3)-ij_imagr(j,1)*P2(3,1:3);

        y(1,1)=-P1(1,4)+ij_imagl(j,2)*P1(3,4);
        y(2,1)=-P1(2,4)+ij_imagl(j,1)*P1(3,4);
        y(3,1)=-P2(1,4)+ij_imagr(j,2)*P2(3,4);
        y(4,1)=-P2(2,4)+ij_imagr(j,1)*P2(3,4);

        xyz_cena(j,1:3)=((inv((A')*A))*A'*y)';
    end
end

%Determinina os limites não nulos da Imagem:
function [min_i,max_i,min_j,max_j]=limit(image_in)
%%
[i,j]=find(image_in==1);

[value, max_i]=max(i);
[value, min_i]=min(i);

[value, max_j]=max(j);
[value, min_j]=min(j);

function image_out=border_destructor(image_in)
%% Destrói as bordas oriundas da rectificação.

image_in=logical(image_in);
%tamanho da borda a preencher (em pixels):

```

```

borda=50;

%borda superior;
for i=1:borda
    for j=1:size(image_in,2)
        image_in(i,j)=0;
    end
end

%borda inferior;
for i=(size(image_in,1)-borda):size(image_in,1)
    for j=1:size(image_in,2)
        image_in(i,j)=0;
    end
end

%borda direita;
for i=1:size(image_in,1)
    for j=1:borda
        image_in(i,j)=0;
    end
end

%borda esquerda;
for i=1:size(image_in,1)
    for j=(size(image_in,2)-borda):size(image_in,2)
        image_in(i,j)=0;
    end
end

image_out=image_in;

```

B.8 CORRESPONDÊNCIA

```

function [Y,XL,XR]= corresp3(imL,imR)
%Correspondência entre pixels

imL=logical(imread('imageL_rectified1.bmp'));
imR=logical(imread('imageR_rectified1.bmp'));

imL=border_destructor(imL);
%Elimina as bordas brancas (0) oriundas da rectificação.
imR=border_destructor(imR);
%Elimina as bordas brancas (0) oriundas da rectificação.

%bordas na ordem correta
bordasL_cell = bwboundaries(imL,'noholes');
bordasR_cell = bwboundaries(imR,'noholes');
bordasL = bordasL_cell{celula(bordasL_cell)};
bordasR = bordasR_cell{celula(bordasR_cell)};

% %janelas
% ny=6;
% nx=10;

[iL,jL]=find(imL==1);%indices i e j para a imagem left.
[iR,jR]=find(imR==1);%indices i e j para a imagem right.

```

```

il_min=min(il);%Ranges de linha na imagem Esquerda
il_max=max(il);

ir_min=min(ir);%Ranges de linha na imagem direita
ir_max=max(ir);

jl_min=min(jl);%Ranges de coluna na imagem Esquerda
jl_max=max(jl);

jr_min=min(jr);%Ranges de coluna na imagem direita
jr_max=max(jr);

%correções
jl_min=jl_min+round((jl_max-jl_min)/10);
jl_max=jl_max-round((jl_max-jl_min)/10);

jr_min=jr_min+round((jr_max-jr_min)/10);
jr_max=jr_max-round((jr_max-jr_min)/10);

% desvios
dsup = abs(il_max - ir_max);
dinf = abs(il_min - ir_min);
desvio = max(dsup,dinf);
ddir = abs(jl_max - jr_max);
desq = abs(jl_min - jr_min);
dispari = (ddir+desq)/2;
minsoma = dispari - (jr_max - jr_min)/10;
maxsoma = dispari + (jr_max - jr_min)/10;

% %janela
% dx = ceil((jl_max-jl_min)/nx);
% dy = ceil((il_max-il_min)/ny);
dx = 15;
dy = 6;

index = 1;

for n=1:size(bordasL,1)
    subL = imL(bordasL(n,1)-ceil(dy/2)+1:bordasL(n,1)+floor(dy/2),...
        ...bordasL(n,2)-ceil(dx/2)+1:bordasL(n,2)+floor(dx/2));
    indice1=1;

    for icorresp=(bordasL(n,1)-desvio):(bordasL(n,1)+desvio)
        indice2=1;

        for jcorresp=jr_min:jr_max

            if ((imR(icorresp,jcorresp)~=0))

                if ((imR(icorresp,jcorresp)~=0)&&(bordasL(n,2)-maxsoma)..
                    ...<jcorresp)&&(jcorresp<bordasL(n,2)-minsoma))

                    subR = imR((icorresp-ceil(dy/2)+1):(icorresp+floor(dy/2)),...
                        ... (jcorresp-ceil(dx/2)+1):(jcorresp+floor(dx/2)));
                    correlacao = ~xor(subL,subR);
                    correlacao = correlacao + 3*subL.*subR;
                    correl(indice1,indice2) = sum(correlacao(:));
                    if(bordasL(n,2)<jcorresp)
                        correl(indice1,indice2) = 0;
                    end
                end
            end
        end
    end
end

```

```

        end
    end
    indice2=indice2+1;
end
indice1=indice1+1;
end
if(exist('correl','var'))
    maxcorrel = max(correl(:));
    [a,b] = find(correl==maxcorrel);
    if (length(b)<2)
        for c=1:length(a)
            Y(index) = bordasL(n,1);
            XL(index) = bordasL(n,2);
            XR(index) = jr_min+b(c)-1;
            index=index+1;
        end
    end
    clear correl;
end
end
end

```

B.9 RECONSTRUÇÃO 3D

```

function f3D

%Leitura das imagens:
%imL=imread('imageL_rectified1.bmp');
%imR=imread('imageR_rectified1.bmp');

clear all, clear, clc
close all

% Localização dos diretórios que contém as imagens

fileFolder = fullfile('C:\Master\image_nok_1.5m');

% Criação da lista com o nome dos arquivos
dirOutputL = dir(fullfile(fileFolder,'imageL_rectified*.bmp'));
dirOutputR = dir(fullfile(fileFolder,'imageR_rectified*.bmp'));

% Vetor contendo os nomes dos arquivos
fileNamesL = {dirOutputL.name}';
fileNamesR = {dirOutputR.name}';

%% Matriz de Calibração
load Calib_Results_stereo_rectified.mat

PL = [fc_left_new(1) 0 cc_left_new(1) 0;...
      ... 0 -fc_left_new(2) cc_left_new(2) 0; 0 0 1 0];

PR = [fc_right_new(1) 0 cc_right_new(1);...
      ... 0 -fc_right_new(2) cc_right_new(2); 0 0 1]*[R_new' -T_new];

```

```

for k =1:(length(fileNamesL)+length(fileNamesR))/2
    clc;
    disp('Iteração: ')
    disp(k);disp('No total de: '),disp((length(fileNamesL)+length(fileNamesR))/2),
    disp('-----');

    %Definição do endereço das imagens
    Adress_imL= fullfile(fileFolder,fileNamesL{k});
    Adress_imR= fullfile(fileFolder,fileNamesR{k});

    %Leitura do par de imagens
    imL=imread(Adress_imL);
    imR=imread(Adress_imR);

    %Corta bordas:
    imL=border_destructor(imL);
    %Elimina as bordas brancas (0) oriundas da retificação.
    imR=border_destructor(imR);
    %Elimina as bordas brancas (0) oriundas da retificação.

    % Correlação:
    [Y,XL,XR]= corresp3(imL,imR);

    ij_imagL=[Y',XL'];
    ij_imagR=[Y',XR'];

    % Reconstituição 3D:
    xyz_cena=reconsti(ij_imagL,PL,ij_imagR,PR,0,0);

    % Varredura para encontrar os y iguais na figura 3D.
    % Objetivo: Ligar os pontos tridimensionais
    % (emulando a figura 3D completa)
    ind=1;
    ind2=1;
    for i=ind2:length(xyz_cena(:,1))
        for j=1:length(xyz_cena(:,1))
            if xyz_cena(i,2)==xyz_cena(j,2)
                py(ind,1)=xyz_cena(i,2);
                py(ind+1,1)=xyz_cena(j,2);

                px(ind,1)=xyz_cena(i,1);
                px(ind+1,1)=xyz_cena(j,1);

                pz(ind,1)=-1574;
                pz(ind+1,1)=-1574;

                ind=ind+2;
            end
        end
        ind2=ind2+1;
    end

    for i=1:length(xyz_cena(:,1))
        px(i,1)=xyz_cena(i,1);
        if rem(i,2)==0
            py(i,1)=2;
        elseif rem(i,2)~=0
            py(i,1)=-23;
        end
        pz(i,1)=-1574;
    end

```

```

    end
    xyz(k).coord=xyz_cena;
    %plot3(xyz_cena(:,1),xyz_cena(:,2),xyz_cena(:,3)); pause;
    save xyz;
end

function result=celula(c)
valor=max(cellfun('length', c));
for i=1:size(c,1)
    if(length(c{i})==valor)
        result=i;
    end
end
end

function xyz_cena=reconsti(ij_imag1,P1,ij_imag2,P2,k_linhas,k_colunas)

%xyz_cena=reconsti(ij_imag1,P1,ij_imag2,P2,k_linhas,k_colunas)
%
%Efetua o cálculo das três coordenadas dos pontos de uma cena
%arbitrária, a partir das posições dos "pixels" correspondentes
%em duas imagens distintas da mesma. Cada linha em ij_imag1 e
%em ij_imag2 deve corresponder às coordenadas i e j do pixel
%de um dos pontos a serem reconstituídos. P1 e P2 são as matrizes
%de transformação perspectiva das câmeras que geraram cada uma
%das imagens. A matriz xyz_cena retornada pela função contém
%em cada linha as coordenadas x,y,z de um unico ponto.

ij_imag1=[ij_imag1(:,1)-k_linhas ij_imag1(:,2)-k_colunas];
ij_imag2=[ij_imag2(:,1)-k_linhas ij_imag2(:,2)-k_colunas];

k=size(ij_imag1);

if((k(1,2)~=2) | (k(1,1)~=size(ij_imag2,1)))
    disp('Entradas invalidas');
else
    k=k(1,1);
    xyz_cena=zeros(k,3);
    for j=1:k
        A(1,1:3)=P1(1,1:3)-ij_imag1(j,2)*P1(3,1:3);
        A(2,1:3)=P1(2,1:3)-ij_imag1(j,1)*P1(3,1:3);
        A(3,1:3)=P2(1,1:3)-ij_imag2(j,2)*P2(3,1:3);
        A(4,1:3)=P2(2,1:3)-ij_imag2(j,1)*P2(3,1:3);

        y(1,1:1)=-P1(1,4)+ij_imag1(j,2)*P1(3,4);
        y(2,1:1)=-P1(2,4)+ij_imag1(j,1)*P1(3,4);
        y(3,1:1)=-P2(1,4)+ij_imag2(j,2)*P2(3,4);
        y(4,1:1)=-P2(2,4)+ij_imag2(j,1)*P2(3,4);

        xyz_cena(j,1:3)=(inv((A')*A))*A'*y)';
    end
end
end

```

```

%-----
function [min_i,max_i,min_j,max_j]=limit(image_in)
%%
[i,j]=find(image_in==1);

[value, max_i]=max(i);
[value, min_i]=min(i);

[value, max_j]=max(j);
[value, min_j]=min(j);

```

B.10 REDES NEURAIAS

```

%-----
function net=rede_neural

%-----

clear all;

%%
[P,T]=nn_format;%(coef,cadeia);%vetorização dos Coeficientes de Fourier:
%Ordem: [A, B, C, D]';
%OBS: Os níveis cc [A0, B0] estão sendo desconsiderados (invariância a
%translação)

%-----
%      NEWFF(P,T,S,TF,BTF,BLF,PF,IPF,OPF,DDF) takes,
%      P - RxQ1 matrix of Q1 representative R-element input vectors.
%      T - SNxQ2 matrix of Q2 representative SN-element target vectors.
%      Si - Sizes of N-1 hidden layers, S1 to S(N-1), default = [].
%           (Output layer size SN is determined from T.)
%      TFi - Transfer function of ith layer. Default is 'tansig' for
%            hidden layers, and 'purelin' for output layer.
%      BTF - Backprop network training function, default = 'trainlm'.
%      BLF - Backprop weight/bias learning function, default = 'learngdm'.
%      PF - Performance function, default = 'mse'.
%      IPF - Row cell array of input processing functions.
%            Default is {'fixunknowns','remconstantrows','mapminmax'}.
%      OPF - Row cell array of output processing functions.
%            Default is {'remconstantrows','mapminmax'}.
%      DDF - Data division function, default = 'dividerand';
%-----
net = newff(P,T,30,({'tansig','tansig'}),'trainlm','learngdm','mse');

%      net.layers{1}.transferFcn='purelin';
%      net.layers{2}.transferFcn='purelin';%'tansig';
%      net.performFcn='mse';
%      net.trainFcn='trainbfg';%melhor treinamento que existe (ônus: tempo Computacional)
%      %net.trainFcn='trainlm';

net.trainParam.epochs = 500;
net.trainParam.goal= 1e-3;
net.trainParam.lr=0.01;

net = train(net,P,T);

```

```
save net;
```