Universidade de São Paulo
Instituto de Física de São Carlos
Departamento de Física e Informática

**AVALIAÇÃO FORMAL E APLICAÇÕES DE TÉCNICAS
DE APROXIMAÇÃO DE CONTORNOS EM IMAGENS
DIGITAIS ATRAVÉS DE SEGMENTOS DE RETA**

Sérgio Hiroshi Ishikawa

Dissertação apresentada ao Instituto de Física
de São Carlos, Universidade de São Paulo,
para obtenção do título de Mestre em Ciências
“Física Aplicada - opção Física Computacional”.

Orientador : Prof. Dr. Luciano da Fontoura Costa

São Carlos - SP

1996

OK

Ishikawa, Sérgio Hiroshi

Avaliação formal e aplicações de técnicas de aproximação de contornos em imagens digitais através de segmentos de reta / Sérgio Hiroshi Ishikawa. /// São Carlos, 1996.

151p.

Dissertação (Mestrado) - Instituto de Física de São Carlos, 1996.

Orientador: Prof. Dr. Luciano da Fontoura Costa.

1. Análise de imagens. 2. Visão Computacional. 3. Inspeção visual.

**UNIVERSIDADE
DE SÃO PAULO**

Instituto de Física de São Carlos

Av. Dr. Carlos Botelho, 1465
CEP 13560-250 - São Carlos - SP
Brasil

Fone (016) 272-6222

Fax (016) 272-2218

MEMBROS DA COMISSÃO JULGADORA DA DISSERTAÇÃO DE MESTRADO DE **SÉRGIO HIROSHI ISHIKAWA** APRESENTADA AO INSTITUTO DE FÍSICA DE SÃO CARLOS, DA UNIVERSIDADE DE SÃO PAULO, EM 15 DE AGOSTO DE 1996.

COMISSÃO JULGADORA:

Prof. Dr. Luciano da Fontoura Costa/IFSC-USP

Prof. Dr. Roberto de Alencar Lotufo/UNICAMP

Prof. Dr. Nelson Delfino D'Avila Mascarenhas/UFSCar

Dedico
aos meus pais, Milton e Namiko,
aos meus irmãos,
e à namorada Regina,
pela paciência, incentivo
e compreensão.

AGRADECIMENTOS

- Ao Dr. Luciano da Fontoura Costa pela paciência, incansável estímulo e encorajamento na orientação desse trabalho.
- Aos professores do Departamento de Física e Informática que contribuíram para a minha formação.
- Ao Dr. Keiichi Abe, professor da Faculdade de Informação da Universidade de Shizuoka-Japão, pelos primeiros passos nesta área de pesquisa.
- Ao Instituto de Pesquisas Meteorológicas - UNESP, por permitir a realização desse curso de mestrado e pelo apoio recebido.
- Aos pesquisadores, Dr. Shigetoshi Sugahara e Dr. Ernesto dos Santos Caetano Neto do Instituto de Pesquisas Meteorológicas, pelos inúmeros conselhos e sugestões.
- Aos professores, Dr. Rafael Linden (Universidade Federal do Rio de Janeiro-UFRJ), Dr. Norberto Cairasco (Faculdade de Medicina de Ribeirão Preto - USP), e Dra. Tânia de Fátima Salvini (Universidade Federal de São Carlos-UFSCar) pelas lâminas de tecidos analisados nessa monografia.
- Aos amigos e parentes pelo incentivo.
- Aos colegas do Grupo de Instrumentação e Informática, em especial aos do Grupo de Visão Cibernética e funcionários do IFSC-USP-São Carlos pela convivência sempre agradável.
- Às secretárias Wladerez Aparecida Gounella Caiado e Cláudia Tofaneli Figueiredo, por toda a assistência recebida.
- Às bibliotecárias da USP, UNESP e UNICAMP pelo auxílio na busca de referências bibliográficas.

SUMÁRIO

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS

RESUMO

ABSTRACT

1 - INTRODUÇÃO	1
2 - REVISÃO	4
2.1 Conceitos Básicos	4
2.1.1 Vizinhança-4 e vizinhança-8	5
2.1.2 caminho-4 e caminho-8	6
2.1.3 Conectividade-4 e conectividade-8	7
2.1.4 "Chain-Code"	7
2.1.5 Distância Euclidiana	8
2.1.6 Histograma polar de orientação	9
2.1.7 Quantização	10
2.1.8 Segmentos de reta digitais e sua geração	12
2.1.9 Representação de segmentos de reta e barras digitais	18
2.1.10 Afinamento	20
2.1.11 Algoritmo de perseguição de contornos	21
2.2 Aproximação por segmentos de reta	23
2.3 Representação simbólica da aproximação por segmentos de reta	25
2.4 Classificação dos algoritmos de aproximação por segmentos de reta	27
2.4.1 Algoritmos baseados em Perseguição de Contornos + Cálculo da Curvatura	28
2.4.2 Algoritmos da categoria "Merge"	28
2.4.3 Categoria "Split"	30
2.4.4 Categoria "Split/Merge"	30
2.5 "Vector Tracer"	31
2.5.1 Ângulos Acentuados	33
2.5.2 Ângulos Suaves	34
2.5.3 Algoritmo do "Vector Tracer"	35
2.5.4 Vantagens do "Vector Tracer"	36
2.5.5 Desvantagens do "Vector Tracer"	36
2.6 Algoritmos híbridos de aproximação por segmentos de reta	37
2.7 Transformada de Hough	38
2.7.1 Algoritmo da Transformada de Hough	43
2.7.2 Vantagens	44
2.7.3 Desvantagens	45
2.7.4 Procedimentos Pós-Transformada de Hough	47
2.7.4.1 Backmapping	47
2.7.4.2 Análise de conectividade	49
2.8 Distorções em imagens	52

2.9 Aplicações	54
3 - IMPLEMENTAÇÃO	56
3.1 Algoritmos de detecção de segmentos de retas	57
3.1.1 Módulo de perseguição de contornos + "Vector Tracer"	57
3.1.1.1 Algoritmo do "Vector Tracer 2"	60
3.1.1.2 Representação abreviada dos métodos para "Vector Tracer"	62
3.1.2 Módulo de transformada de Hough	63
3.1.2.1 Transformada de Hough	64
3.1.2.2 Resolução do espaço de Hough	65
3.1.2.3 "Backmapping"	67
3.1.2.4 Análise de conectividade	67
3.1.2.5 Representação abreviada dos métodos por transformada de Hough	68
3.2 Algoritmos para avaliação e comparação de desempenho	70
3.2.1 Gerador de segmentos de reta com parâmetros aleatórios	70
3.2.2 Algoritmo de reconstrução de contornos	72
3.2.3 Módulo para análise estatística dos resultados	73
3.3 Algoritmos para aplicações	76
3.3.1 Histograma gerado através da transformada de Hough	77
3.3.2 Histograma gerado através do "VectorTracer"	77
3.4 Informações adicionais	78
4 - RESULTADOS	79
4.1 Parâmetros considerados na avaliação de desempenho	79
4.1.1 Imagens sintéticas	79
4.1.2 Imagens reais	80
4.2 Avaliação de desempenho	82
4.2.1 Detecção de retas em imagens	82
4.2.2 Resultados da detecção de retas	83
4.3 Reconstrução de imagens e histograma de orientações	90
4.3.1 Resultados da reconstrução e determinação dos histogramas de orientações das imagens	92
5 - CONCLUSÕES E SUGESTÕES PARA DESENVOLVIMENTOS FUTUROS	96
5.1 Conclusões	96
5.2 Sugestões para desenvolvimentos futuros	100
ANEXO A - RESULTADOS DA AVALIAÇÃO COM RETAS	101
ANEXO B - RESULTADOS DE RECONSTRUÇÕES E HISTOGRAMAS POLARES DE ORIENTAÇÕES	103
ANEXO C - ESTRUTURAS DE DADOS	139
ANEXO D - SUGESTÕES DE PARÂMETROS PARA RECONSTRUÇÕES E PARA HISTOGRAMAS POLARES DE ORIENTAÇÕES	143
ANEXO E - RECURSOS EMPREGADOS	144
REFERÊNCIAS BIBLIOGRÁFICAS	145

LISTA DE FIGURAS

Figura 2.1 - Vizinhança-4 (a) e vizinhança-8 (b) .	6
Figura 2.2 - Caminho8 (a) e caminho 4 (b).	6
Figura 2.3 - Conectividade-4 entre "pixels" Q e R, conectividade-8 entre "pixels" Q e P.	7
Figura 2.4 - Freeman "Chain code".	8
Figura 2.5 - Curva original (a), figura quantizada com Interseção com grade (descrito na seção 2.1.7) (b), direções seguidas com caminho-8 (c) e "chain code" resultante (d).	8
Figura 2.6 - Distância Euclidiana.	9
Figura 2.7 - Imagem original (a) e histograma polar de orientações (b).	9
Figura 2.8 - Quantização Caixa quadrada (a) e quantização Interseção com grade (b).	11
Figura 2.9 - Representação da forma paramétrica da reta conhecida como normal.	12
Figura 2.10 - Reta ideal de inclinação θ , distância da origem ρ , posição inicial (x_i, y_i) e posição final (x_f, y_f) de comprimento L .	13
Figura 2.11 - Segmento de reta digital de largura de "intercept" w unitária gerada através de quantização Interseção com grade.	14
Figura 2.12 - Segmento da classe C_y gerado de forma errônea (a) e segmento da classe C_y gerado corretamente (b).	16
Figura 2.13 - Barra digital da classe C_x de largura $w=2$ com início (X_i, Y_i) e fim (X_f, Y_f) e comprimento L .	16
Figura 2.14 - Geração de uma barra da classe C_x com ponto inicial (x_i, y_i) , ponto final (x_f, y_f) e utilizando a largura $w=2$. O comprimento da barra é dado pela distância Euclidiana L .	18
Figura 2.15 - Dois segmentos de reta contínuo l_1 (a) e l_2 (b) dando origem a pontos inicial (x_i, y_i) e ponto final (x_f, y_f) idênticos.	19
Figura 2.16 - Uma curva (a) e sua correspondente afinada (b).	20
Figura 2.17 - Uma curva digital (a) e sua aproximação poligonal (b).	24
Figura 2.18 - Representação gráfica de um polígono aproximado através de segmentos de reta	27
Figura 2.19 - Condição inicial dos vetores a , b e c (a), cálculo do ângulo (b) e região de alta curvatura (c).	32
Figura 2.20 - Condição inicial com os vetores a , b e c alinhados (a), entrada dos vetores a e b numa região de curvatura elevada (b) e deslocamento dos vetores a , b e c (c).	33
Figura 2.21 - Condição inicial com os vetores a , b e c alinhados (a), seleção de um ponto de curvatura suave significativo (b) e deslocamento dos vetores a , b e c .	34
Figura 2.22 - Imagem (a), espaço paramétrico gerado para origem no canto inferior esquerdo (b) e espaço paramétrico para origem no centro (c).	39
Figura 2.23 - Pontos colineares na imagem (a), curvas senoidais geradas no espaço de Hough (b) e a reta detectada (c).	40
Figura 2.24 - Imagem (x, y) (a) e espaço de θ - ρ discretos com a representação das bandas (b).	42
Figura 2.25 - Segmento da classe C_x com largura de "intercept" $w=2$ e inclinação $\theta=7.1^\circ$ em (a) e $\rho=41.2^\circ$ em (b).	43
Figura 2.26 - Imagem original (a) e espaço paramétrico de Hough gerado (b).	44
Figura 2.27 - Ilustração do problema da sub amostragem (a) e da superamostragem (b).	45
Figura 2.28 - Retas replicadas obtidas devido a super amostragem (a) (tendo como imagem original a figura 2.27a), quebra ou perda na detecção de segmentos (b).	46
Figura 2.29 - Segmento de reta (a), pontos esparsos (b), vários segmentos distintos (c) e o segmento de reta resultante da análise do espaço de Hough (d).	46
Figura 2.30 - Imagem original (a) e espaço de Hough correspondente (b), objeto adicionado à imagem (a) em (c) e acumulador de Hough resultante (d).	46
Figura 2.31 - Espaço paramétrico resultante da transformada de Hough aplicada na imagem da figura 2.28(a) contendo 16.195 pontos maiores que 5 (a) e resultado após o "Backmapping" com 85 valores maiores que 5 (b).	48
Figura 2.32 - Histograma unidimensional (projecção da barra sobre o eixo x ou y dependendo da inclinação), resultado do histograma para $\text{gap}=0$ e $\text{gap}=1$.	50
Figura 2.33 - Distorções causadas pelo processo de afinamento.	52

Figura 2.34 - Distorções causadas pelo processo de aproximação poligonal .	53
Figura 3.1 - Diagrama esquemático do módulo usando o algoritmo "Vector Tracer".	57
Figura 3.2 - Segmento original (a) e (b) resultado aproximado por 9 segmentos, (c) imagem original e (d) resultado aproximado por 15 segmentos.	59
Figura 3.3 - Resultados (a) e (b) aproximados através de um segmento de reta das figuras 3.2.a e 3.2.b respectivamente.	61
Figura 3.4 - Quantidade de segmentos detectados em função do ângulo de inclinação dos segmentos para as duas versões do "Vector Tracer".	61
Figura 3.5 - Diagrama esquemático do módulo de detecção de segmentos de reta usando a transformada de Hough.	63
Figura 3.6 - Limites inferior e superior dos eixos x e y para uma imagem com a origem deslocada para o centro.	64
Figura 4.1 - Segmento de reta de inclinação $\theta = 1.9266$ e $\rho = 40.7765$ gerado com 119 "pixels" (a) e círculo de raio=30 "pixels" gerado pelo algoritmo de Bresenham (b).	80
Figura 4.2 - Células ganglionares (a) e (b), células do hipocampo (c) e (d), e células de músculo (e) e (f).	81
Figura 4.3 - Imagens da figura 4.2 onde foi introduzido ruído .	81
Figura 4.4 - Quantidade de segmentos de reta detectados (a) e fidelidade da reconstrução (b), em função da inclinação do segmento de reta gerado.	84
Figura 4.5 - Segmentos onde há quebra (l1) e onde não há quebra (l2) na perseguição de contornos .	84
Figura 4.6 - Quantidade de segmentos detectados (a) e fidelidade da reconstrução das transformadas de Hough de resolução simples e dupla (b) em função da inclinação da reta.	85
Figura 4.7 - Segmento original (a) e segmento detectado (b) com "pixels" p1 a p4 erroneamente detectados (p1 e p2 não existia e p3 e p4 foram perdidos).	89
Figura 4.8 - Exemplo de utilização de "gap" diferente de zero para imagens ruidosas.	91
Figura 4.9 - Imagem original ruidosa (a), afinada (d), a reconstrução da Hough1 (b), a reconstrução da Hough2 (e), o histograma da Hough1 (c) e o histograma da Hough2 (f).	94
Figura B.1 - Descrição da estrutura adotada para a apresentação dos resultados de reconstrução e histograma de acordo com o grupo de parâmetros utilizados.	103
Figura B.2 - Reconstrução e histograma de uma imagem contendo um círculo de raio=10 "pixels", usando o "Vector Tracer" e a transformada de Hough.	104
Figura B.3 - Reconstrução e histograma de uma imagem ruidosa contendo um círculo de raio=10 "pixels", usando o "Vector Tracer" e a transformada de Hough.	105
Figura B.4 - Reconstrução e histograma de uma imagem contendo um círculo de raio=30 "pixels", usando o "Vector Tracer" e a transformada de Hough.	106
Figura B.5 - Reconstrução e histograma de uma imagem ruidosa contendo um círculo de raio=30 "pixels", usando o "Vector Tracer" e a transformada de Hough.	107
Figura B.6 - Reconstrução e histograma de uma imagem contendo um círculo de raio=60 "pixels", usando o "Vector Tracer" e a transformada de Hough.	108
Figura B.7 - Reconstrução e histograma de uma imagem ruidosa contendo um círculo de raio=60 "pixels", usando o "Vector Tracer" e a transformada de Hough.	109
Figura B.8 - Reconstrução e histograma de uma imagem contendo uma célula ganglionar, usando o "Vector Tracer" e a transformada de Hough.	110
Figura B.9 - Reconstrução e histograma de uma imagem ruidosa contendo uma célula ganglionar, usando o "Vector Tracer" e a transformada de Hough.	111
Figura B.10 - Reconstrução e histograma de uma imagem contendo uma célula ganglionar (fig.4.14.a rotacionada), usando o "Vector Tracer" e a transformada de Hough.	112
Figura B.11 - Reconstrução e histograma de uma imagem ruidosa contendo uma célula ganglionar rotacionada, usando o "Vector Tracer" e a transformada de Hough.	113
Figura B.12 - Reconstrução e histograma de uma imagem contendo uma célula ganglionar, usando o "Vector Tracer" e a transformada de Hough.	114
Figura B.13 - Reconstrução e histograma de uma imagem ruidosa contendo uma célula ganglionar, usando o "Vector Tracer" e a transformada de Hough.	115
Figura B.14 - Reconstrução e histograma de uma imagem contendo uma célula ganglionar (fig. 4.18.a rotacionada), usando o "Vector Tracer" e a transformada de Hough.	116
Figura B.15 - Reconstrução e histograma de uma imagem ruidosa contendo uma célula ganglionar, usando o "Vector Tracer" e a transformada de Hough.	117

Figura B.16 - Reconstrução e histograma de uma imagem contendo células do hipocampo, usando o "Vector Tracer" e a transformada de Hough.	118
Figura B.17 - Reconstrução e histograma de uma imagem ruidosa contendo células do hipocampo, usando o "Vector Tracer" e a transformada de Hough.	119
Figura B.18 - Reconstrução e histograma de uma imagem contendo células do hipocampo (fig. 4.22.a rotacionada), usando o "Vector Tracer" e a transformada de Hough.	120
Figura B.19 - Reconstrução e histograma de uma imagem ruidosa contendo células do hipocampo, usando o "Vector Tracer" e a transformada de Hough.	121
Figura B.20 - Reconstrução e histograma de uma imagem contendo células do hipocampo, usando o "Vector Tracer" e a transformada de Hough.	122
Figura B.21 - Reconstrução e histograma de uma imagem ruidosa contendo células do hipocampo, usando o "Vector Tracer" e a transformada de Hough.	123
Figura B.22 - Reconstrução e histograma de uma imagem contendo células do hipocampo (fig. 4.26.a rotacionada), usando o "Vector Tracer" e a transformada de Hough.	124
Figura B.23 - Reconstrução e histograma de uma imagem ruidosa contendo células do hipocampo, usando o "Vector Tracer" e a transformada de Hough.	125
Figura B.24 - Reconstrução e histograma de uma imagem contendo células de músculo, usando o "Vector Tracer" e a transformada de Hough.	126
Figura B.25 - Reconstrução e histograma de uma imagem ruidosa contendo células de músculo, usando o "Vector Tracer" e a transformada de Hough.	127
Figura B.26 - Reconstrução e histograma de uma imagem contendo células de músculo (fig. 4.30.a rotacionada), usando o "Vector Tracer" e a transformada de Hough.	128
Figura B.27 - Reconstrução e histograma de uma imagem ruidosa contendo células de músculo, usando o "Vector Tracer" e a transformada de Hough.	129
Figura B.28 - Reconstrução e histograma de uma imagem contendo células de músculo, usando o "Vector Tracer" e a transformada de Hough.	130
Figura B.29 - Reconstrução e histograma de uma imagem ruidosa contendo células de músculo, usando o "Vector Tracer" e a transformada de Hough.	131
Figura B.30 - Reconstrução e histograma de uma imagem contendo células de músculo (fig. 4.34 rotacionada), usando o "Vector Tracer" e a transformada de Hough.	132
Figura B.31 - Reconstrução e histograma de uma imagem ruidosa contendo células de músculo, usando o "Vector Tracer" e a transformada de Hough.	133
Figura C.1 - Estrutura de listas ligadas.	140
Figura C.2 - Estrutura de pilha.	142

LISTA DE TABELAS

<i>Tabela 2.1 - Classificação do "pixel" de acordo com a quantidade de vizinhos iguais a "1".</i>	21
<i>Tabela 3.1 - Descrição do parâmetros e sua função e em que "VectorTracer" é usado.</i>	58
<i>Tabela 3.2 - Descrição dos parâmetros e os módulos onde são utilizados.</i>	63
<i>Tabela 3.3 - Matriz de correlação entre k variáveis aleatórias.</i>	75
<i>Tabela 4.1 - Tipos de células, o processo químico e fator de aumento usado no microscópio.</i>	80
<i>Tabela 4.2 - Descrição das características avaliadas para a medida do desempenho.</i>	82
<i>Tabela 4.3 - Resultados estatísticos obtidos para VT {2, 0.4, 0.98, 0.995} em (a), VT {3, 0.4, 0.98, 0.995} em (b) e para VT {5, 0.4, 0.98, 0.995} em (c).</i>	86
<i>Tabela 4.4 - Resultados estatísticos obtidos para HT {3,2,3,2,1.0} em (a), HT {3,2,3,2,1.0} em (b) e para HT {3,2,3,2,1.0} em (c).</i>	87
<i>Tabela 4.5 - Grupos de parâmetros para o "Vector Tracer" e transformada de Hough, para a reconstrução e geração de histogramas.</i>	90
<i>Tabela 5.1 - Principais características do "Vector Tracer" e transformada de Hough.</i>	99
<i>Tabela A.1 - Matrizes de correlação para VT1 (a) e VT2 (b) (VT {2, 0.4, 0.98, 0.995}).</i>	101
<i>Tabela A.2 - Matrizes de correlação para VT1 (a) e VT2 (b) (VT {3, 0.4, 0.98, 0.995}).</i>	101
<i>Tabela A.3 - Matrizes de correlação para VT1 (a) e VT2 (b) (VT {5, 0.4, 0.98, 0.995}).</i>	101
<i>Tabela A.4 - Matrizes de correlação para HT1 (a) e HT2 (b) (HT {3, 2, 3, 2, 1.0}).</i>	102
<i>Tabela A.5 - Matrizes de correlação para HT1 (a) e HT2 (b) (HT {3, 2, 3, 2, 1.5}).</i>	102
<i>Tabela A.6 - Matrizes de correlação para HT1 (a) e HT2 (b) (HT {3, 2, 3, 2, 2.0}).</i>	102
<i>Tabela B.1 - Estatísticas sobre as imagens processadas usando "Vector Tracer".</i>	134
<i>Tabela B.2 - Estatísticas sobre as imagens processadas usando Hough.</i>	134
<i>Tabela B.3 - Estatísticas sobre as imagens ruidosas processadas usando "Vector Tracer".</i>	134
<i>Tabela B.4 - Estatísticas sobre as imagens ruidosas processadas usando Hough.</i>	134
<i>Tabela B.5 - Resultados do processamento de imagens reais utilizando "Vector Tracer" e transformada de Hough.</i>	135
<i>Tabela D.1 - Parâmetros do "Vector Tracer" para reconstrução de imagens com pequenos detalhes (a) e detalhes maiores (b).</i>	143
<i>Tabela D.2 - Parâmetros da transformada de Hough para reconstrução de imagens com pequenos detalhes (a) e detalhes maiores (b).</i>	143

LISTA DE ABREVIACÕES

- HT : Transformada de Hough ("Hough transform").
- HT1 : Transformada de Hough usando acumulador com resolução simples.
- HT2 : Transformada de Hough usando acumulador com resolução dupla.
- VT : "Vector Tracer".
- VT1 : "Vector Tracer" original.
- VT2 : "Vector Tracer" modificado.
- HS : Acumulador de Hough de resolução simples (Hough Space).
- HS1 : Acumulador usado pela transformada de Hough.
- HS2 : Acumulador de Hough usado pelo "Backmapping".
- HSD : Acumulador de Hough de resolução dupla (Hough Space/Double).
- HSD1 : Acumulador usado pela transformada de Hough (resolução dupla).
- HSD2 : Acumulador usado pelo "Backmapping" (resolução dupla).
- BM : "Backmapping".
- AC : Análise de conectividade.
- OCR : Reconhecimento ótico de caracteres ("Optical Character Recognition").
- CCD : Dispositivo por acoplamento de carga ("Charged Coupled Device").

RESUMO

A importância de elementos de reta nos sistemas de visão dos primatas tem sido reconhecida desde as primeiras experiências eletrofisiológicas conduzidas a quase quatro décadas. Foram encontradas células sensíveis à orientação e ao comprimento de barras ao longo de toda a parte primária do sistema visual. Em muitas situações torna-se interessante capturar a essência de uma curva com o menor conjunto de pontos possível de forma a minimizarmos a redundância de informação. Tais evidências, aliadas ao fato de uma curva poder ser representada com certa precisão através de segmentos de reta, sugere uma melhor investigação sobre o uso de tais atributos na construção de sistemas de visão computacionais de inspiração biológica. Este trabalho desenvolve uma série de aspectos relacionados a representação de informações visuais, mais especificamente contornos, através de segmentos de reta, incluindo a descrição e a avaliação de desempenho de duas técnicas de aproximação por segmentos de reta (transformada de Hough e "Vector Tracer"), assim como a aplicação para a obtenção de histogramas polares de orientação. Um dos principais objetivos desta monografia é a avaliação formal, através da comparação de desempenho da transformada de Hough e do "Vector Tracer", como um meio de representação e análise de contornos.

ABSTRACTS

The importance of straight features in primate vision systems has been recognized since the first electrophysiological experiments on primate vision systems, conducted almost four decades ago. Cells sensitive to orientation and length of straight features, in variable degree of complexity, were found throughout most of the visual cortex. This suggests it is worth to investigate the use of straight features in the construction of biologically inspired visual systems. Such a concept is suitable because a curve can be represented with relative accuracy through piecewise linear segments. Under several circumstances, it is interesting to capture the essence of a curve with the minimum set of points, in such a way to reduce the redundancy of information. This work is aimed at a series of aspects related to the piecewise linear representation of visual information, including : the description and benchmarking of two techniques (Hough Transform and Vector Tracer), as well as their application for the extraction of orientation polar histograms. The main purpose of this work is to formally evaluate, through benchmarking, the performance of the Hough Transform and the Vector Tracer as a means of contour representation and analysis.

1. INTRODUÇÃO

A importância de elementos de reta nos sistemas de visão dos primatas tem sido reconhecida e estudada desde as primeiras experiências eletrofisiológicas (HUBEL e WIESEL (1962), HUBEL e WIESEL (1979) e BLASDEL e SALAMA (1986)). Foram encontradas células sensíveis à orientação e ao comprimento de barras ao longo de toda a parte primária do sistema visual. Tal evidência, aliada ao fato de uma curva poder ser representada com certa precisão através de segmentos de reta, sugere uma melhor investigação sobre o uso de tais atributos na construção de sistemas de visão de inspiração biológica.

Este trabalho desenvolve uma série de aspectos relacionados a representação de informações visuais através de segmentos de reta, incluindo a descrição de duas técnicas de aproximação por segmentos de reta, a avaliação de desempenho dessas técnicas e aplicações na obtenção de histogramas polares de orientação. Um dos objetivos principais desta monografia é a avaliação formal, através da comparação de desempenho da transformada de Hough (HOUGH, 1962) e do "Vector Tracer" (AGUI *et al.*, 1982), como um meio de representação e análise de contornos. A transformada de Hough utilizada para esta monografia é uma variante, baseada na proposta de DUDA e HART (1972). Esses algoritmos representam duas tendências importantes em análise de imagens, sendo portanto, escolhidos para esta monografia. No caso da transformada de Hough existem cerca de 160 referências de publicações entre 1995 e 1996 na base de dados INSPEC sobre a chave "Hough Transform".

A transformada de Hough tem um enfoque mais global, enquanto que o "Vector Tracer" é mais local e depende do acompanhamento dos contornos. A transformada de Hough é melhor para implementações paralelas, sendo que as extremidades dos segmentos detectados nem sempre pertencem à curva original. O "Vector Tracer" gera aproximações contínuas com pontos pertencentes ao contorno. É interessante comparar os desempenhos na detecção de segmentos de reta, pois as técnicas são de importância fundamental na extração de informações contidas na imagem, permitindo o uso do computador no reconhecimento de figuras tais como desenhos esquemáticos, arquitetônicos e mapas metropolitanos (FULFORD, 1981)(EJIRI *et al.* 1984)(JOSEPH, PRIDMORE, 1992).

Esta monografia tem como uma das principais motivações a avaliação formal em termos quantitativos e qualitativos, dos algoritmos para análise de imagens conforme recomendações de JAIN e BINFORD (1991) que apontam para a necessidade de desenvolvimento de metodologias experimentais, técnicas de avaliação comparativa, além de uma teoria baseada em suposições realísticas.

Os cinco capítulos que integram essa monografia estão organizados como discutido a seguir.

O capítulo 1 introduz a motivação e os objetivos a serem alcançados com o desenvolvimento dessa monografia.

O capítulo 2 apresenta os conceitos básicos necessários para uma compreensão mais completa dos temas a serem apresentados no decorrer dessa monografia. Esse capítulo trata da definição das relações básicas entre os elementos de imagem ("pixels"), o processo de discretização de uma imagem, definição e geração de retas digitais, codificação de imagens, representação simbólica, histogramas polares, representação através de segmentos de reta, e uma revisão sobre vários algoritmos de aproximação de curvas digitais através de segmentos de reta. Esse capítulo apresenta ainda um tópico sobre distorções no processamento digital de imagens e a aplicação dos algoritmos em citometria.

O capítulo 3 apresenta os detalhes de implementação dos algoritmos apresentados no capítulo precedente e de outros algoritmos necessários para

a comparação dos métodos que foram implementados ou modificados pelo autor desta monografia.

O capítulo 4 apresenta a metodologia utilizada para a avaliação dos métodos, o uso dos algoritmos na reconstrução de imagens através de segmentos de reta, a obtenção de histogramas polares de orientação com atenção especial para citometria e resultados obtidos pela comparação dos métodos.

O capítulo 5 conclui este trabalho apresentando os principais pontos observados na comparação dos dois métodos de aproximação de curvas digitais através de segmentos de reta e demais conclusões. Apresenta ainda sugestões e orientação para projetos que poderão dar continuidade a esta monografia.

2. REVISÃO

O objetivo desta seção é fornecer uma visão geral sobre os conceitos e técnicas básicas considerados nessa monografia, assim como a caracterização dos principais algoritmos relacionados a detecção de contornos em imagens binárias.

2.1 Conceitos Básicos

Para tornar possível o processamento digital de imagens reais, primeiramente devemos introduzir a imagem no sistema computacional e para isso existem vários tipos de equipamentos : microdensitômetros, "scanners" (CCD), câmeras vidicon (GONZALEZ, WINTZ, 1987) e CCD. Desse modo, uma imagem, definida a partir de uma função contínua $F(x,y)$ (GONZALEZ, WINTZ, 1987), deve (FULFORD, 1981)(EJIRI *et al.* 1984)(KASTURI *et al.* 1990)(JOSEPH, PRIDMORE, 1992), ser discretizada tanto espacialmente como em amplitude. A discretização das coordenadas espaciais é definida como amostragem da imagem ("sampling") e a discretização da amplitude como quantização de níveis de cinza (GONZALEZ, WINTZ, 1987). Supondo que uma imagem $F(x,y)$ é aproximada através de amostras de espaçamento uniforme e na forma de uma matriz de $N \times M$, temos que:

$$F(x,y) \approx \begin{bmatrix} f(0,0).....f(0,M-1) \\ \dots \\ f(N-1,0).....f(N-1,M-1) \end{bmatrix} \quad (1)$$

Em (1), $F(x,y)$ representa a imagem original no espaço contínuo, e a matriz f representa o que chamamos de imagem digital, onde cada elemento da matriz corresponde a um elemento de imagem, que é chamado comumente de “pixel” (“picture element”)(GONZALEZ, WINTZ, 1987). Assim, cada “pixel” contém o valor correspondente a um nível de cinza quantizado. As quantidades N e M da expressão (1) que especificam o tamanho da imagem dependem da aplicação. As imagens a serem processadas também podem ter sido geradas através de técnicas de computação gráfica.

Neste trabalho empregamos esses dois métodos para a obtenção das imagens, isto é, imagens reais e sintéticas. As imagens consideradas no presente trabalho pertencem a um caso particular, pois possuem apenas dois níveis de cinza, ou seja, são imagens binárias, onde um nível corresponde ao branco e o outro ao preto. Convencionamos que o “pixel” preto é representado pelo valor “1” e o “pixel” branco pelo valor “0”. Apresenta-se a seguir alguns conceitos importantes sobre a relação entre “pixels” numa imagem digital.

2.1.1 Vizinhança-4 e vizinhança-8

Um “pixel” p de coordenadas (x,y) possui quatro “pixels” vizinhos : $(x+1,y)$, $(x,y-1)$, $(x-1,y)$ e $(x,y+1)$ (figura 2.1.a). Esse conjunto de “pixels” é chamado de vizinhança-4 de p , e cada um destes “pixels” está a uma distância de um “pixel” de p . No caso da vizinhança-4, essa distância de um “pixel” ao seu vizinho coincide com a distância Euclidiana.

Além dos vizinhos na vizinhança-4, há mais quatro “pixels” vizinhos na diagonal: $(x+1,y-1)$, $(x-1,y-1)$, $(x-1,y+1)$ e $(x+1,y+1)$. Quando somados aos “pixels” de vizinhança-4, esses “pixels” formam a vizinhança-8 de p (figura 2.1.b). Então o ponto p está a uma distância de 1 (um) “pixel” dos “pixels” situados na diagonal, mas a distância Euclidiana no caso dos “pixels” situados

na diagonal é de $\sqrt{2}$. Temos na figura 2.1 a representação gráfica destas relações de vizinhança (GONZALEZ, WINTZ, 1987).

Figura 2.1 - Vizinhança-4 (a) e vizinhança-8 (b) ¹.

2.1.2 Caminho-4 e caminho-8

Define-se o caminho ("path") de um "pixel" em (x_i, y_i) a outro em (x_n, y_n) como uma seqüência de "pixels" indexados (x_i, y_i) , (x_{i+1}, y_{i+1}) , ..., (x_n, y_n) , onde um "pixel" (x_k, y_k) é vizinho de outro "pixel" (x_{k+1}, y_{k+1}) para todos os valores de k , com $0 \leq k \leq n-1$. Se os "pixels" formam vizinhança-4, então o caminho é denominado caminho-4 (figura 2.2.b). Para vizinhança-8 o caminho é caminho-8 (figura 2.2.a). Exemplos da representação gráfica desta relação podem ser vistos na figura 2.2.

Figura 2.2 - Caminho-8 (a) e caminho-4 (b).

¹ Observa-se que embora em computação gráfica se adote o canto das coordenadas no canto superior esquerdo, o presente trabalho o considera no canto inferior esquerdo para finalidade de facilitar a interpretação matemática dos desenvolvimentos e resultados.

2.1.3 Conectividade-4 e conectividade-8

Seja S o conjunto formado por todos os “pixels” pretos contidos numa imagem. Um “pixel” $p \in S$ é dito conectado à outro “pixel” $q \in S$, se e somente se existe um caminho de p à q formado exclusivamente por “pixels” pertencentes a S (JAIN, KASTURI, SCHUNK, 1995). Caso esse conjunto S seja formado apenas por “pixels” com vizinhança-4 a conectividade é 4. Para o caso de vizinhança-8, a conectividade é 8. Na figura 2.3 acha-se representado um exemplo onde Q possui conexão-4 (“4-connected”) e conexão-8 (“8-connected”) em relação a R , enquanto Q possui apenas conexão-8 (“8-connected”) com relação a P .

Figura 2.3 - Conectividade-4 entre “pixels” Q e R , conectividade-8 entre “pixels” Q e P .

2.1.4 “Chain-Code”

É desejável que os contornos resultantes do processo de quantização sejam caracterizados pela propriedade de que cada “pixel” da seqüência que compõe esse contorno esteja conectado a pelo menos um dos oito “pixels” da sua vizinhança-8. Rotulando-se esses oito “pixels” vizinhos com números de 0 a 7, no sentido anti-horário, começando pelo eixo x positivo, pode-se representar esses contornos de forma simples através de seqüências de dígitos octais (FREEMAN, 1974). Este tipo de codificação é conhecido como “Freeman Chain-Code” e a sua representação mais típica se encontra na figura 2.4.

Figura 2.4 - Freeman "Chain code".

Abaixo, na figura 2.5 temos um exemplo de representação de uma curva utilizando-se "Freeman Chain-Code" :

Figura 2.5 - Curva original (a), figura quantizada com *Interseção com grade* (descrito na seção 2.1.7) (b), direções seguidas com "8-path" (c) e "chain code" resultante (d).

Geralmente essa codificação é obtida através da utilização de um algoritmo de perseguição de contornos como o descrito nesse capítulo. Essa codificação é bastante utilizada para representar os contornos extraídos de uma imagem, sendo que alguns algoritmos de aproximação poligonal o utilizam como formato de entrada para o contorno a ser aproximado. A seguir apresentamos a descrição das relações de distância utilizadas nessa monografia.

2.1.5 Distância Euclidiana

A medida de distância denominada Euclidiana, dada por $L = \sqrt{(X_1 - X_2)^2 + (Y_1 - Y_2)^2}$, é representada na figura 2.6.

Figura 2.6 - Distância Euclidiana.

Introduzimos a seguir um tipo especial de histograma usado.

2.1.6 Histograma polar de orientação

Um histograma é uma representação gráfica onde temos num dos eixos uma grandeza geralmente dividida em faixas e no outro uma medida de “população”. Nesta monografia, utilizamos o histograma polar de orientação representado na figura 2.7, cuja função é representar num gráfico polar, isto é com origem no centro, as direções predominantes dos objetos contidos na imagem. Isto é feito traçando-se segmentos ou barras partindo da origem e com comprimento que varia de acordo com o acumulado do comprimento dos segmentos de reta encontrados na imagem. Este valor acumulado é distribuído em faixas de acordo com o ângulo de inclinação do segmento de reta. A escolha da resolução angular é um processo empírico. As experiências realizadas com imagens indicam que um valor razoável está entre 5° e 10°.

Figura 2.7 - Imagem original (a) e histograma polar de orientações (b).

Esse tipo de histograma será apresentado como uma aplicação dos métodos utilizados, ao final desta monografia. Pode-se encontrar na literatura científica aplicações de histogramas de orientação em análise de solo (COSTA *et al.*, 1991), na extração de características faciais (BOWNS, MORGAN, 1993), na detecção de simetria (SUN, 1995) e em análise neuromorfometria (COSTA, 1995b). O algoritmo utilizado para gerar esse histograma será descrito no capítulo 3.

2.1.7 Quantização²

A quantização é um processo que leva formas contínuas aos seus correspondentes discretos (mapeamento não é biunívoco). Para que um esquema de quantização seja útil, este deve ser simples, padronizado e ser aplicável a qualquer tipo de contorno. Deve permitir ainda uma fácil codificação do contorno (por exemplo, usando "chain code"), e facilitar o processamento através de um computador digital (FREEMAN, 1974). A quantização neste caso consiste em se escolher as coordenadas da grade regular que mais se aproxima da curva. Esse processo, entretanto, não é tão óbvio como pode parecer. Embora existam várias possibilidades, esta monografia será restrita às duas formas de quantização relacionadas abaixo (FREEMAN, 1974) :

- quantização *Caixa quadrada* ("Square-Box quantization")
- quantização *Interseção com grade* ("Grid-Intersect quantization")

² Observa-se que o termo "quantização" tem sido usado de maneira incongruente na literatura sobre processamento digital de imagens. FREEMAN (1974) usa o termo para descrever a discretização enquanto outros autores usam o mesmo termo para a discretização em amplitude da imagem em níveis de cinza. Nesse caso, como se trata da atribuição de "pixels" em termos espaciais, o termo mais usual seria "sampling" (amostragem).

Figura 2.8 - Quantização *Caixa quadrada* (a) e quantização *Interseção com grade* (b).

Quantização Caixa Quadrada

Neste esquema, exemplificado na figura 2.8.a, utiliza-se o seguinte critério, definido em (FREEMAN, 1974).

“Sempre que uma curva passar por um quadrado definido por $x-0.5 < x \leq x+0.5$ e $y-0.5 < y \leq y+0.5$, a coordenada (x,y) é marcada”.

Pode-se verificar que este esquema gera imagens de curvas onde os “pixels” possuem vizinhança-4.

Quantização Interseção com Grade

Representando-se os pontos da curva contínua por (X,Y) , o critério caracterizando esse esquema de quantização é o seguinte (FREEMAN, 1974):

“ Sempre que $X=x$, isto é, quando a curva contínua cruza o eixo y , seleciona-se o ponto y correspondente do intervalo $y-0.5 < Y \leq y+0.5$ que seja mais próximo de Y ”.

“Sempre que $Y=y$, seleciona-se o ponto x correspondente dentro do intervalo $x-0.5 < X \leq x+0.5$ que seja mais próximo de X ”.

Este esquema, ilustrado na figura 2.8.b, de quantização produz imagens cujos “pixels” possuem vizinhança-8.

Na próxima seção são apresentadas as definições de retas, segmentos e barras no espaço contínuo e os correspondentes no espaço discreto, assim como os métodos empregados na detecção dos mesmos.

2.1.8 Segmentos de reta digitais e sua geração

Uma linha reta (comprimento $L \rightarrow \infty$) pode ser definida no espaço cartesiano contínuo (x,y) através de sua equação paramétrica na forma normal ou polar, conforme a expressão (2) e figura 2.9 :

$$\rho = x.\cos(\theta) + y.\sin(\theta) \quad (2)$$

onde:

θ : corresponde ao ângulo de inclinação da reta normal em relação ao eixo x , no sentido anti-horário e no intervalo $[0,\pi)$, implicando dessa forma a existência de valores negativos de ρ .

ρ : a distância do pé da normal em relação à origem.

Figura 2.9 - Representação da forma paramétrica da reta conhecida como normal.

Em COSTA (1992a) adota-se a seguinte definição de segmento de reta com inclinação da normal θ , distância ρ da origem e comprimento finito L :

“Um segmento de reta ideal é formado por um conjunto infinito de pontos (X,Y) no espaço Cartesiano contínuo que satisfaz a expressão (2) e se estende de uma extremidade inicial (X_i, Y_i) à uma extremidade final (X_f, Y_f) , tal que possua comprimento

$$L = \sqrt{(X_i - X_f)^2 + (Y_i - Y_f)^2} \text{ ”.}$$

A figura 2.10 apresenta um exemplo de segmento de reta ideal com inclinação θ e distância em relação à origem ρ e comprimento L .

Figura 2.10 - Reta ideal de inclinação θ , distância da origem ρ , posição inicial (x_i, y_i) e posição final (x_f, y_f) de comprimento L .

A escolha das coordenadas (X_i, Y_i) e (X_f, Y_f) irá depender da classe a qual pertencerem os segmentos de reta, se C_x ou C_y , conforme descrito a seguir.

Para uma melhor caracterização de segmentos de reta digitais, dividem-se os mesmos em duas classes distintas de acordo com o ângulo de inclinação (COSTA, 1995a). A classe $C_x = \{ \text{Segmentos de reta onde } (0.25\pi \leq \theta < 0.75\pi) \}$ composta de segmentos com projeção no eixo x maior que no eixo y e $C_y = \{ \text{Segmentos de reta onde } (0 \leq \theta < 0.25\pi) \text{ e } (0.75\pi \leq \theta < \pi) \}$ composta de segmentos com projeção no eixo y maior que no eixo x .

Então, caso o segmento pertença a classe C_x , o ponto inicial será o que possuir o menor valor de x . De outro modo, se pertencer a classe C_y o ponto inicial será o que possuir o menor valor de y . Esse critério de escolha será utilizado quando da geração de segmentos e barras digitais descrito nesta seção.

Apesar da caracterização de linha reta ser trivial no espaço geométrico contínuo, o mesmo não ocorre com o seu equivalente discreto (FREEMAN, 1974) (COSTA, 1992b). Já vimos como uma reta digital é definida como o resultado da quantização de uma reta dentro de um espaço geométrico discreto. De modo análogo, um segmento de reta digital é o resultado da quantização de um segmento de reta de extremo inicial (X_i, Y_i) e extremo final (X_f, Y_f) , dando origem a um número finito de pontos conforme pode ser visto na figura 2.11 a seguir.

Figura 2.11 - Segmento de reta digital de largura de "intercept" w unitária gerada através de quantização *Interseção com grade*.

Nessa figura, L (distância no plano contínuo) e L_d (distância no plano discreto) representam de modo equivalente a distância Euclidiana no plano Cartesiano. A diferença está no fato de que as extremidades de uma banda discretizada será vista como a linha vertical mais grossa e não mais a linha inclinada. Isto é válido para o caso de bandas que dão origem a segmentos da classe C_x . Para o caso de bandas que dão origem a segmentos da classe C_y , as extremidades serão representadas por linhas horizontais. A largura w mostrada na figura 2.11 é um valor real que indica a largura da interseção da banda com o eixo y e x no caso de segmentos das classes C_x e C_y respectivamente.

Embora possamos ter retas digitais (infinitas) no sentido teórico, na prática trabalha-se apenas com segmentos de reta digitais. O presente trabalho utiliza a quantização *Interseção com grade* para a geração de segmentos de reta digitais. As expressões em (3)/(4) e (5)/(6) são empregadas

respectivamente para a geração de segmentos de reta digitais das classes C_x e C_y descritas anteriormente:

$$\left\{ \begin{array}{l} y = \text{round}\left(\frac{\rho - x \cos(\theta)}{\sin(\theta)}\right) \\ \text{ou} \\ \frac{\rho - x \cos(\theta)}{\sin(\theta)} - \frac{1}{2} < y \leq \frac{\rho - x \cos(\theta)}{\sin(\theta)} + \frac{1}{2} \end{array} \right. \quad (3)$$

$$\left\{ \begin{array}{l} x = \text{round}\left(\frac{\rho - y \sin(\theta)}{\cos(\theta)}\right) \\ \text{ou} \\ \frac{\rho - y \sin(\theta)}{\cos(\theta)} - \frac{1}{2} < x \leq \frac{\rho - y \sin(\theta)}{\cos(\theta)} + \frac{1}{2} \end{array} \right. \quad (4)$$

As expressões (3) a (6) se originam do fato que segmentos de reta digitais são considerados como a quantização de bandas de largura w igual a um. Observa-se que existe uma correspondência biunívoca entre os segmentos de reta digitais e os conjuntos de pontos gerados por bandas com $w=1$ (COSTA, 1992a). Assim, para se gerar um segmento de reta da classe C_x , calcula-se os valores de y_i de acordo com a expressão (3) substituindo-se os valores de $x_i = x_{inicial}, \dots, x_{final}$ na mesma expressão. A geração de um segmento da classe C_y se dá de forma análoga, utilizando a expressão (5). Essa divisão em classes é feita para que no processo de geração de segmentos de reta digitais, não ocorra o aparecimento de falhas nos segmentos. Na figura 2.12.a exemplificamos os pontos gerados incorretamente para um segmento da classe C_y utilizando-se a expressão (3) à partir de um $x_{inicial}$ até um x_{final} , e em (b) o mesmo segmento dessa vez com os pontos gerados de forma correta de acordo com a sua classe, isto é utilizando-se a expressão (5) à partir de um $y_{inicial}$ até um y_{final} .

Figura 2.12 - Segmento da classe C_y gerado de forma errônea (a) e segmento da classe C_y gerado corretamente (b).

Partindo das definições anteriores de segmentos de reta e segmentos de reta digitais, deriva-se a definição de barra digital genérica como a quantização de uma banda de largura de “intercept”³ w , dividindo-as também nas classes C_x e C_y , conforme a seguinte descrição gráfica.

Figura 2.13 - Barra digital da classe C_x de largura $w=2$ com início (X_i, Y_i) e fim (X_f, Y_f) e comprimento L .

Deve-se lembrar que as extremidades das barras discretizadas obedecem o mesmo critério adotado para o caso dos segmentos discretizados abordado anteriormente.

³ O termo “intercept” sugere o ponto onde a banda cruza o eixo da ordenada ou da abscissa para bandas da classe C_x e C_y respectivamente.

Embora as expressões (3) a (6) possam ser utilizadas para gerar segmentos de largura unitária, quando se quer gerar barras de largura w maior que um, devemos utilizar as expressões (7) e (8) para a geração de barras digitais das classes C_x e C_y respectivamente:

$$\frac{\rho - x \cos(\theta)}{\sin(\theta)} - \frac{w}{2} < y \leq \frac{\rho - x \cos(\theta)}{\sin(\theta)} + \frac{w}{2} \quad (7)$$

$$\frac{\rho - y \sin(\theta)}{\cos(\theta)} - \frac{w}{2} < x \leq \frac{\rho - y \sin(\theta)}{\cos(\theta)} + \frac{w}{2} \quad (8)$$

Dessa forma, quando se tem barras da classe C_y , ou seja com inclinação entre $[0^\circ, 45^\circ]$ e $[135^\circ, 180^\circ]$, calcula-se x_{inf} e x_{sup} para cada y_i . Caso contrário, para as barras da classe C_x ou seja quando se tem inclinação entre $[45^\circ, 134^\circ]$ são calculados os valores de y_{inf} e y_{sup} para cada x_i . Então, para a classe C_x , calculam-se os valores de y_{inf} e y_{sup} através das seguintes expressões:

$$y_{inf} = \frac{\rho - x \cos(\theta)}{\sin(\theta)} - \frac{w}{2} \quad (9)$$

$$y_{sup} = \frac{\rho - x \cos(\theta)}{\sin(\theta)} + \frac{w}{2} \quad (10)$$

Para a classe C_y utilizam-se as expressões abaixo para o cálculo de x_{inf} e x_{sup} :

$$x_{inf} = \frac{\rho - y \sin(\theta)}{\cos(\theta)} - \frac{w}{2} \quad (11)$$

$$x_{sup} = \frac{\rho - y \sin(\theta)}{\cos(\theta)} + \frac{w}{2} \quad (12)$$

Dessa forma, geram-se as barras pertencentes à classe C_x , marcando-se para cada valor de x compreendido entre (x_i, y_i) e (x_f, y_f) , todos os pontos pertencentes à banda de largura w , dada pelos valores y_{inf} e y_{sup} obtidos

através das expressões (9) e (10). O processo de geração de uma barra da classe C_x encontra-se esquematizado na figura 2.14.

Figura 2.14 - Geração de uma barra da classe C_x com ponto inicial (x_i, y_i) , ponto final (x_f, y_f) e utilizando a largura $w=2$. O comprimento da barra é dado pela distância Euclidiana L .

O processo de geração de barras da classe C_y é semelhante ao utilizado para a geração de barras da classe C_x exceto que as expressões (11) e (12) são utilizadas nesse caso. A geração de barras utilizando-se a divisão das classes C_x e C_y é necessária para evitar o aparecimento de falhas nas barras geradas pelos mesmos motivos dos expostos para o caso dos segmentos de reta. Discutimos a seguir a representação simbólica de retas e barras digitais.

2.1.9 Representação de segmentos de reta e barras digitais

Um segmento de reta digital l pode ser representado por uma quádrupla descrevendo o ponto inicial (x_i, y_i) , o ponto final (x_f, y_f) , o ângulo de inclinação da reta normal θ (valores contínuos) e a distância em relação à origem ρ (valores contínuos) :

$$l = \{ (x_i, y_i), (x_f, y_f), \theta, \rho \} \quad (13)$$

Os parâmetros θ e ρ são necessários pois (x_i, y_i) e (x_f, y_f) não são suficientes para caracterizar um segmento de reta discreto. Isto ocorre porque

existe um conjunto contínuo e infinito de retas (com parâmetros θ e ρ aproximados) que, quando discretizadas darão origem ao segmento l com início (x_i, y_i) e final (x_f, y_f) . Esta situação é representada graficamente na figura abaixo.

Figura 2.15 - Dois segmentos de reta contínuo $l1$ (a) e $l2$ (b) dando origem a pontos inicial (x_i, y_i) e ponto final (x_f, y_f) idênticos.

Caso não se disponha do θ e do ρ do segmento de reta contínuo original que gerou o segmento, podemos obtê-los através do ajuste dos mínimos quadrados conforme apresentado por MELTER *et al.* (1993). Essa é uma representação compacta, pois foi omitido um outro parâmetro w que no caso de segmentos é assumido igual a um. Analogamente a representação de segmentos de reta digitais, pode-se representar barras digitais incluindo desta vez a largura w da banda original. Assim, a sua representação é a seguinte :

$$b = \{ (x_i, y_i), (x_f, y_f), \theta, \rho, w \} \quad (14)$$

Na próxima seção apresentamos o algoritmo de afinamento, que como o nome diz tem a função de reduzir a largura ou afinar contornos de figuras contidos numa imagem.

2.1.10 Afinamento

Esse algoritmo tem como objetivo afinar as regiões que compõem a imagem, idealmente à largura de um "pixel" (DAVIES, PLUMMER, 1981)(SMITH, 1987), de modo a reduzir a quantidade de dados da imagem. O afinamento é comumente usado nos estágios de pré-processamento para facilitar a análise e o reconhecimento de nível mais elevado para aplicações como OCR ("optical character recognition"), interpretação de diagramas, análise de impressões digitais, e detecção de características gráficas ("features") em visão computacional (O'GORMAN, 1990). De acordo com O'GORMAN (1990), as características que desejamos de um algoritmo de afinamento são: (a) que ele não gere regiões não contíguas a partir de regiões contíguas, (b) que o resultado sejam linhas que representem aproximadamente a sua linha central também chamada de esqueleto, (c) que o resultado possua no mínimo conectividade-8, (d) que a localização aproximada dos pontos extremos das regiões sejam mantidas, (e) e espera-se ainda que este algoritmo não gere ruído indesejável, o que na maioria das vezes não pode ser completamente evitado (SMITH, 1987). Na figura 2.16.b exemplificamos o afinamento da curva na figura 2.16.a.

Figura 2.16 - Uma curva (a) e sua correspondente afinada (b).

O algoritmo de afinamento utilizado para esta monografia foi idealizado por TAMURA (1975). É um algoritmo paralelo baseado em 4 sub-ciclos e que gera regiões perfeitamente 8-conectados, tendo sido comparado a outros algoritmos em (TAMURA, 1978). Para maiores detalhes sobre algoritmos de afinamento veja (ZHANG, SUEN, 1984), (O'GORMAN, 1990), (PERROTI,

LOTUFO, 1992) e (LAM *et al*, 1992). Geralmente o algoritmo de afinamento é empregado antes de se extrair os contornos dos objetos ou figuras contidos na imagem, o que é feito pelo algoritmo de perseguição de contornos descrito a seguir.

2.1.11 Algoritmo de perseguição de contornos

O algoritmo de perseguição de contornos tem como função identificar os pontos pertencentes à uma curva digital contida na imagem e armazenar essas coordenadas consecutivas em um vetor (TORIWAKI, 1991)(WANG *et al*, 1995). Nesse trabalho, o algoritmo de perseguição de contornos é usado para extrair curvas digitais da imagem e submetê-las ao algoritmo de aproximação por segmentos de reta chamado “Vector Tracer” (AGUI, 1982) a ser descrito nesta seção. Para essa finalidade a imagem deve ter passado por algum processo para a detecção de bordas e caso essas bordas não tenham a largura de um “pixel” a imagem deve passar pelo processo de afinamento. O algoritmo foi adaptado pelo autor desta monografia, tendo como base os algoritmos de LI *et al* (1982) e BELL e PAU (1990). Esse algoritmo trabalha com imagens de conectividade 4 e 8 (BELL, PAU, 1990)(TORIWAKI *et al*, 1991). O algoritmo de perseguição de contornos leva em consideração as seguintes características dos “pixels” encontrados nas imagens. Os “pixels” seguem à uma classificação de acordo com a quantidade de vizinhos iguais a “1”, conforme se vê na tabela 2.1.

Tabela 2.1 - Classificação do “pixel” de acordo com a quantidade de vizinhos iguais a “1”.

“Pixels” iguais a “1”	Classificação dos “pixels”
0	ponto isolado na imagem
1	extremidade de um segmento (inicio ou fim)
2	ponto pertencente ao contorno
≥ 3	ponto de ramificação ou cruzamento de contornos

Nessa implementação os casos onde se tem dois ou mais vizinhos iguais a “1” são tratados da mesma forma conforme o passo (2) do algoritmo de

perseguição de contornos. O algoritmo de perseguição de contornos proposto por LI *et al* (1982) é apresentado a seguir.

Algoritmo de Perseguição de contornos

- (1) Prepare uma pilha vazia; Busque na imagem a partir do canto superior esquerdo, um “pixel” que ainda não esteja marcado. Quando a busca chegar ao canto inferior direito da imagem, o processo está terminado; grave a sua coordenada e passe o foco de atenção para este “pixel”; passe para o modo de busca de “pixels”;
- (2) Grave a posição do “pixel”; Marque este “pixel”; Verifique na sua vizinhança-8 quantos “pixels” ainda não estão marcados;
 - Se for 0 este “pixel” é o final da curva, então vá para o passo 3.
 - Se for 1, então passe a atenção para esse “pixel” e vá para o passo (2);
 - Se for ≥ 2 , então é um cruzamento ou uma bifurcação. Passe o foco de atenção para um destes dois e armazene o(s) outro(s) na pilha.
- (3) Armazene os segmentos extraídos em uma estrutura de dados;
- (4) Retire da pilha uma coordenada e mova o foco de atenção para ela e vá para o passo (2). Se a pilha estiver vazia, o processo de perseguição de um contorno contíguo foi concluído. Vá para (1).

No passo (2), a busca por um “pixel” vizinho igual a “1” segue o critério que mantém a perseguição na direção do deslocamento anterior (BELL, PAU, 1990). A essa direção anterior é sempre atribuído o valor zero no passo (1). Caso o “pixel” dessa direção não possua o valor um, então é checado o vizinho a esquerda e depois o vizinho a direita ⁴, alternadamente até que um “pixel” com valor “1” seja encontrado. O passo (3) corresponde ao algoritmo de aproximação poligonal proposto por AGUI *et al* (1982). Nesse estágio verifica-se a distância do último ponto do contorno ao primeiro, se os mesmos forem vizinhos, o algoritmo fecha o contorno. Após a aproximação poligonal realizada

⁴ Em termos das oito direções do código de Freeman.

pelo “Vector Tracer” os pontos resultantes são armazenados em uma estrutura de lista ligada dinâmica. Em algumas aplicações pode ser interessante ou necessário que o algoritmo de perseguição de contornos retorne o “Freeman Chain-Code”. O passo 3 será descrito em detalhes no capítulo 3, na seção que trata da implementação do algoritmo “Vector Tracer”.

Uma das conseqüências do uso do algoritmo de aproximação poligonal é a redução ou compressão dos dados da imagem, conforme pode ser visto no exemplo acima. De acordo com SMITH (1987) alguns algoritmos obtêm taxas de compressão na razão de 15:1 para imagens “razoáveis”. Possuem ainda a propriedade de suavizar pequenas distorções no contorno provocadas pelo processo de discretização, afinamento ou devido ao ruído na imagem (SMITH, 1987). Essa suavização ocorre devido as propriedades dos algoritmos de aproximação que acabam eliminando algumas dessas distorções. O algoritmo de aproximação poligonal é usado para extrair as características das figuras originais, possuindo assim muitas aplicações em reconhecimento de padrões (FULFORD, 1981)(KUROZUMI, DAVIS, 1982)(EJIRI *et al*, 1990)(AGUI, 1988). A seguir descrevemos os algoritmos de aproximação por segmentos de reta.

2.2 Aproximação por segmentos de reta ⁵

Uma curva pode ser representada com certa precisão através de segmentos de reta. Em muitas situações torna-se importante capturar a essência da curva com a menor quantidade de pontos possível, de forma a minimizarmos a redundância de informação. ATTNEAVE (1954) mostrou que os vértices (regiões de alta curvatura) de um contorno contém grande parte das informações sobre a forma do mesmo. Essa demonstração gerou uma série de trabalhos a respeito de técnicas de detecção desses vértices (ROSENFELD, JOHNSTON, 1973)(ROSENFELD, WESKA, 1975)(TEH, CHIN, 1989)(ABE *et*

⁵ A “Aproximação Poligonal” consiste de um caso especial onde se faz a aproximação por segmentos de reta de uma curva fechada. O termo é utilizado com o intuito de abreviar “Aproximação por segmentos de reta” ou “Piecewise Linear Approximation”.

al., 1991)(ANSARI, DELP, 1991). Esses vértices, também chamados de pontos dominantes, correspondem aproximadamente às interseções dos segmentos adjacentes e estão localizadas geralmente nas regiões de alta curvatura. Em geral há duas abordagens para a detecção dos pontos dominantes. Na primeira detecta-se os vértices ou pontos dominantes através do esquema do cálculo do ângulo para a detecção do vértice (ROSENFELD, JOHNSTON, 1973)(ROSENFELD, WESKA, 1975). TEH e CHIN (1989) contém uma boa revisão dessas técnicas. Na segunda abordagem obtém-se os pontos dominantes (vértices) através da aproximação do contorno por meio de segmentos de reta, sujeito entretanto a certas restrições quanto ao ajuste ótimo (MONTANARI, 1970). A aproximação poligonal é uma técnica que utiliza a segunda abordagem para representarmos curvas digitais, ou seja, através de segmentos de reta, conforme ilustrado na figura 2.17.

Figura 2.17 - Uma curva digital (a) e sua aproximação poligonal (b).

Existem inúmeros trabalhos que tratam da revisão de algoritmos de aproximação poligonal, entre eles podemos citar : KUROSUMI e DAVIS (1982), DUNHAM (1986) e PIKAZ (1995). A aproximação poligonal é também chamada de “piecewise linear approximation” ou aproximação por segmentos lineares, ou ainda aproximação por segmentos de reta. Observa-se que a aproximação poligonal no espaço discreto feita com o uso de segmentos de reta de comprimento de dois “pixels” consegue representar com exatidão qualquer curva (COSTA, 1992a). Para a utilização dos algoritmos de aproximação por segmentos de reta que trabalham apenas com as coordenadas dos pontos, primeiramente devemos perseguir os contornos na imagem, isto é, devemos extrair os contornos que compõem a imagem e armazená-las em um vetor.

Este processo de perseguição de contornos depende do tipo de imagem disponível. No presente trabalho, estamos trabalhando com imagens binárias. Caso a imagem tenha sido discretizada e os contornos contidos nessa imagem não tenham a largura de um “pixel” após a obtenção do mapa de bits da imagem (“BITMAP”), utilizamos o algoritmo de afinamento. Esse processo de perseguição de contornos tem como característica não só a redução da quantidade de dados da imagem, mas possibilita também a utilização do algoritmo de aproximação por segmentos de reta chamado “Vector Tracer” utilizado nessa avaliação. Essa redução da quantidade de dados da imagem ocorre porque não precisamos armazenar a imagem original inteira, mas apenas as coordenadas dos “pixels” que compõe o objeto, definido pelas suas bordas. Essas coordenadas podem ser representadas pela posição (x,y) na imagem ou por códigos tipo Freeman “chain-code”. Alguns algoritmos de aproximação por segmento de reta utilizam como entrada esse “chain-code”, outras como é o caso do “Vector Tracer” utilizam a coordenada dos “pixels”. Discute-se a seguir a representação de uma curva aproximada por segmentos de reta e a classificação dos algoritmos.

2.3 Representação simbólica da aproximação por segmentos de reta

Dado uma curva C formada por um conjunto de pontos ordenados que podem ser conectados por uma seqüência de segmentos de reta :

$$C = \{ p_i = (x_i, y_i) \mid i = 1, n \} \quad (15)$$

Pode-se representar essa curva por aproximação por segmentos de reta (“piecewise linear approximation”) da seguinte maneira :

$$C' = \{ (p'_i, p_{i+1}'), (p'_{i+2}, p_{i+3}'), \dots (p'_{n-1}, p'_n) \mid i = 1, m \mid E \leq \varepsilon \} \quad (16)$$

onde $m < n$ e C' não precisa ser obrigatoriamente um subconjunto de C ,

E é o erro da aproximação obtida de acordo com a norma do erro adotada e ε é o erro preestabelecido.

Os pares de pontos (p'_i, p_{i+1}') nem sempre coincidem com os pontos do próximo segmento. Essa curva formada pela seqüência de pontos em C' não pode se distanciar muito de C . O erro E pode ser determinado por um dos seguintes critérios : soma dos erros absolutos (norma l_1) erro médio quadrático (norma l_2) ou o erro uniforme (norma l_∞).

RAY e RAY (1993), por exemplo, utilizam a norma l_1 , PIKAZ e DINSTEIN (1995), utilizam a norma l_2 . Outros autores utilizam a norma uniforme l_∞ (RAMER, 1972)(DUDA, HART, 1973)(WILLIAMS, 1978)(KUROZUMI, DAVIS, 1982)(DUNHAM, 1986).

O critério de AOYAMA e KAWAGOE (1991) usado para medir essa distância o classifica em dois tipos : deslocamento e deflexão. O critério do deslocamento leva em conta a distância Euclidiana dos pontos da curva em relação à reta aproximada enquanto que o critério de deflexão leva em conta o ângulo de desvio da curva em relação ao segmento aproximado (AOYAMA, KAWAGOE, 1991). Podemos visualizar na figura 2.18 abaixo um exemplo de uma curva aproximada por segmentos de reta.

Figura 2.18 - Representação gráfica de um polígono aproximado através de segmentos de reta

Assim neste exemplo $C' = \{l_1, l_2, l_3, l_4\}$ e $l_1 = \{(p_1, p_2)\}$, $l_2 = \{(p_2, p_3)\}$, $l_3 = \{(p_4, p_5)\}$ e $l_4 = \{(p_5, p_6)\}$, considerando-se C como uma curva fechada e $p_i = (x_i, y_i)$.

Apresenta-se a seguir uma descrição mais detalhada dos algoritmos de aproximação poligonal inclusive com a devida classificação dos mesmos.

2.4 Classificação dos algoritmos de aproximação por segmentos de reta

Os algoritmos de aproximação poligonal podem ser divididos basicamente em três classes principais, de acordo com as técnicas básicas empregadas (CESAR JR, COSTA, 1995) : (1) perseguição de contorno + cálculo da curvatura, (2) abordagens globais tipo transformada de Hough e (3) implementações híbridas. Descreve-se a seguir de forma sucinta os algoritmos pertencentes às classes (1) e (3). A transformada de Hough será descrita separadamente na seção 2.7.

2.4.1 Algoritmos baseados em Perseguição de Contornos + Cálculo da Curvatura

A abordagem baseada em perseguição de contornos e cálculo da curvatura e /ou grafos é subdividida segundo PIKAZ (1995) em “merge”, “split” e “split/merge”. Essas categorias são descritas a seguir incluindo alguns exemplos de algoritmos.

2.4.2 Algoritmos da categoria “Merge”

Os algoritmos dessa categoria funcionam baseados na varredura (“scan”) linear da curva digital, iniciando numa das extremidades da curva. Encontra-se o segmento aproximado mais longo possível, adicionando-se ao ponto inicial, um ponto após o outro e enquanto um certo critério estiver sendo satisfeito (WU, WANG, 1993). O critério pode ser a distância do segmento aproximado em relação a curva original ou a diferença angular entre o segmento aproximado e a curva original. O último ponto do segmento encontrado torna-se o início do próximo segmento. Esse procedimento é repetido até o fim da curva. Descrevemos a seguir vários algoritmos de aproximação por segmentos de reta.

MONTANARI (1970) utiliza uma técnica de programação não linear para encontrar o polígono de perímetro mínimo ou MPP (“minimum-perimeter polygon”). Nesse método, movimenta-se cada ponto do contorno dentro de um certo domínio de forma a obter a colinearidade com outros pontos do contorno. Esse algoritmo reduz iterativamente o número de vértices e converge para um mínimo local. SKLANSKY *et al.* (1972) utiliza um polígono de mínimo perímetro para representar a aproximação poligonal de uma curva digital. Baseia-se na idéia de se esticar um elástico de forma que o mesmo se mantenha dentro de células que representam os pontos que constituem o contorno.

WILLIAMS (1978) utiliza um método chamado de “cone-intersection” para encontrar o mais longo segmento aproximado. A máxima distância Euclidiana entre o segmento e os pontos a qual ele aproxima está dentro de um limiar preestabelecido. Para cada ponto que forma o contorno traça-se um círculo

com raio d . Deve-se encontrar o maior segmento de forma que todos os sub-segmentos com o mesmo ponto inicial estejam dentro da interseção do cone formado pelos círculos.

KUROZUMI e DAVIS (1982) utilizam um método chamado de aproximação “minimax”. O critério usado é que o polígono encontrado tenha o mínimo de lados e que a máxima distância Euclidiana entre os pontos e o segmento aproximado esteja dentro de uma tolerância preestabelecida. Os vértices encontrados por esse algoritmo nem sempre pertencem à curva original.

AGUI *et al.* (1982) propõem uma técnica chamada de “Vector Tracer”. Neste método encontra-se a aproximação poligonal de uma curva através do cálculo da curvatura entre pontos consecutivos. Considera-se a existência de três vetores para efetuar o cálculo do cosseno em regiões de curvatura elevada e em regiões de curvatura suave. Este método será descrito separadamente na seção 2.5.

DUNHAM (1986) usa cone-intersection especificando-se o erro uniforme (l_{∞}) e encontra um polígono com número mínimo de segmentos usando programação dinâmica. RAY e RAY (1993) propõe um algoritmo que minimiza quantidade de segmentos com o mínimo erro possível maximizando a função objetivo composta pelo comprimento do segmento de reta e pela soma dos erro absoluto entre o segmento e a curva digital.

PIKAZ (1995) propõe um algoritmo baseado na eliminação iterativa dos vértices condicionado à um erro mínimo preestabelecido. O número de vértices de saída pode ser controlado e a condição de parada é baseada no número de pontos eliminados ou no valor do erro corrente. Os resultados desse algoritmo independem do ponto inicial.

As principais desvantagens dessa classe de algoritmos é que não leva em consideração aspectos globais da curva e alguns vértices podem ser perdidos ou deslocados (AOYAMA, KAWAGOE, 1991). Entre os algoritmos mais avançados que usam MERGE estão o de MONTANARI (1970) que utiliza programação não linear (otimização) e DUNHAM (1986) que utiliza programação dinâmica. Uma característica desses algoritmos é que precisam de perseguição de contornos.

2.4.3 Algoritmos da categoria “Split”

Esta classe de algoritmos é baseada no particionamento recursivo das curvas nos pontos situados a uma maior distância perpendicular de uma corda que liga dois extremos de um trecho da curva. Leva em conta o deslocamento (AOYAMA, KAWAGOE, 1991).

RAMER (1972) e DUDA e HART (1973) propõe um algoritmo que, dado dois pontos situados em extremidades distintas da curva, traça-se uma reta ligando essas extremidades. Calcula-se então a distância perpendicular de cada ponto da curva à reta traçada encontrando-se a máxima distância. Se essa distância máxima for menor que um limiar preestabelecido o processo é terminado. Senão, divide-se a curva em duas partes no ponto de maior distância e repete-se o processo para as duas partes até que a máxima distância seja menor que o limiar.

Com o intuito de minimizar a grande desvantagem desses algoritmos, que é a dependência do ponto inicial, foi proposto um novo algoritmo que combina a detecção de pontos dominantes (TEH, CHIN, 1989) e a aproximação poligonal denominada “curvature-based polygonal approximation” (WU, WANG, 1993). Nesse método, um algoritmo de detecção de pontos dominantes é aplicado de forma a extrair os vértices em potencial. Cada segmento de curva entre dois vértices em potencial são então particionados através de aproximação poligonal.

2.4.4 Algoritmos da categoria “Split/Merge”

Baseiam-se em iterações de “split” e “merge” levando a um número de vértices local mínimo e a um melhoramento da localização dos vértices (PIKAZ, 1995). PAVLIDIS e HOROWITZ (1974) aproximam os pontos do contorno interpolando segmentos de reta. Os vértices do polígono aproximado nem sempre estão sobre a curva original. A aproximação resultante também não é necessariamente contínua, mas pode ser facilmente modificada, conforme afirmam os autores. A localização e o número de vértices é sensível a escala e a orientação.

Uma técnica denominada “curvature guided polygonal approximation” utilizando uma combinação de detecção de pontos de curvatura máxima positiva e curvatura mínima negativa como pontos iniciais para um algoritmo de “split-merge” foi introduzida por ANSARI e DELP (1991). Devido à introdução de falsas concavidades e convexidades locais provenientes da representação discreta dos contornos e de erros de quantização, os autores utilizam um filtro Gaussiano para suavizar o contorno. O método é sensível à escala porém menos sensível à orientação. Os autores apontam que se o contorno do objeto for circular, o algoritmo se degenera para o tipo “split-merge”, porque há a possibilidade de não se detectar regiões de curvatura suave.

2.5 "Vector Tracer"

O “Vector Tracer” é uma técnica de Aproximação Poligonal que foi concebida por AGUI *et al.* (1982). O seu princípio de funcionamento é baseado no cálculo dos ângulos de uma curva digital e é semelhante ao sugerido por (ROSENFELD, JOHNSTON, 1973) e utilizado posteriormente por (ROSENFELD, WESKA, 1975).

A técnica do “Vector Tracer” funciona associada a três vetores a , b e c de tamanhos idênticos, como mostra a figura 2.19.a. Dada uma curva digital definida pelos pontos $p_0, \dots, p_{n-1}$, onde $p_i = (x_i, y_i)$ e v , o “tamanho” do vetor definido pela diferença do índice dos pontos consecutivos, isto é, não em termos de distância Euclidiana, mas, em termos de índice do vetor, temos os vetores a , b e c definidos pelas expressões (17), (18) e (19) a seguir :

$$a_j = (x_j - x_{j+v}, y_j - y_{j+v}) \quad (17)$$

$$b_j = (x_j - x_{j-v}, y_j - y_{j-v}) \quad \text{onde } j = 2v, \dots, n-v-1 \quad (18)$$

$$c_k = (x_k - x_{k+v}, y_k - y_{k+v}) \quad \text{onde } k = 0, \dots, n-3v-1 \quad (19)$$

e calcula-se o coseno em p_j e p_k (coseno do ângulo formado pelos vetores a_i e b_j (20) e o coseno do ângulo formado pelos vetores b_j e c_k (21), respectivamente) como:

$$\cos(a,b)_j = \langle a, b \rangle / |a_j| \cdot |b_j| = (ax_j \cdot bx_j + ay_j \cdot by_j) / |a_j| \cdot |b_j| \quad (20)$$

$$\cos(b,c)_k = \langle b, c \rangle / |b_j| \cdot |c_k| = (bx_j \cdot cx_k + by_j \cdot cy_k) / |b_j| \cdot |c_k| \quad (21)$$

onde : $\langle a, b \rangle$ é o produto escalar dos vetores $a=(ax,ay)$ e $b=(bx,by)$.

Movimenta-se estes vetores a partir de uma das extremidades, ao longo da curva digital que se quer aproximar por segmentos de reta. Realiza-se a aproximação por segmentos de reta, calculando-se o menor ângulo formado pelos segmentos a e b ($\hat{\text{ângulo}}(a,b)$) e b e c ($\hat{\text{ângulo}}(b,c)$), conforme a figura 2.19.(a) e (b).

Figura 2.19 - Condição inicial dos vetores a , b e c (a), cálculo do ângulo (b) e região de alta curvatura (c).

A aproximação por segmentos de reta funciona de maneira diferente quando se tem uma curva digital com ângulo acentuado e quando se tem uma curva digital "suave". Um ângulo é considerado acentuado caso o coseno do menor ângulo formado pelos vetores a e b (ver figuras 2.19 (a) e (b)) for inferior à um determinado valor de limiar ("threshold") $vth1$. Caso esse ângulo seja maior que $vth1$, verifica-se se o ângulo formado pelos vetores b e c é menor que o limiar $vth2$. Caso esse ângulo seja menor, o mesmo é considerado suave.

2.5.1 Ângulos Acentuados

Primeiramente escolhe-se um comprimento razoável para os vetores a , b e c , e supõe-se que estão posicionados relativamente como mostra a figura 2.20.a. Fixando o vetor c , movimenta-se simultaneamente os vetores a e b , checando-se o valor do ângulo (a,b) conforme mostrado nas figuras 2.20.b (note-se que o vetor b está desconectado de c) . Quando se tem ângulos acentuados como o do ponto Q , se traçarmos um gráfico em que na abcissa temos a quantidade de “pixels” avançados e no eixo das ordenadas o ângulo entre os vetores a e b , obtemos o gráfico da figura 2.19.c, onde o valor mínimo estará na posição correspondente ao ponto Q .

Figura 2.20 - Condição inicial com os vetores a , b e c alinhados (a), entrada dos vetores a e b numa região de curvatura elevada (b) e deslocamento dos vetores a , b e c (c).

Dessa forma, verificamos o valor do ângulo (a, b) e armazenamos em uma matriz os valores do coseno, a partir do ponto onde ele se torna menor que um valor de limiar ν_{th1} preestabelecido, até se tornar novamente maior que ν_{th1} . Ao determinarmos o ponto onde se tem o menor valor, podemos detectar o ponto onde se tem o ângulo acentuado. Ao detectar-se o ponto onde se tem o ângulo acentuado, move-se o início do vetor c para este ponto juntamente com os vetores a e b conforme pode ser visto na figura 2.20.c.

2.5.2 Ângulos Suaves

Se aplicarmos a mesma técnica de aproximação por segmentos de reta para detecção de ângulos acentuados em uma curva suave, como os valores dos ângulos são usualmente grandes, não podemos realizar a aproximação. Assim, neste tipo de curvas, a aproximação por segmentos de reta é realizada calculando-se o ângulo entre os vetores b e c . Fixamos o vetor c e movimentamos os vetores a e b , calculando o *ângulo* (b, c) conforme as figuras 2.21.a e 2.21.b. E procuramos por um ponto Q onde o valor do *ângulo* (b, c) seja menor que um valor $vh2$ preestabelecido. Assim, movemos o início do vetor c para o ponto Q juntamente com os vetores a e b (figura 2.21.c) e prosseguimos a detecção.

Figura 2.21 - Condição inicial com os vetores a , b e c alinhados (a), seleção de um ponto de curvatura suave significativo (b) e deslocamento dos vetores a , b e c .

A seguir encontra-se descrito o algoritmo de aproximação poligonal denominado “Vector Tracer” na forma apresentada em (AGUI *et al*, 1988).

2.5.3 Algoritmo do “Vector Tracer”

```

posiciona vetores  $a$ ,  $b$  e  $c$ ;
faça até o fim dos pontos vetorizados
    calcula angulo_agudo e angulo_suave;
    se ( angulo_agudo <  $vth1$  )
        armazena o angulo_agudo num buffer;
        avança vetores  $a$  e  $b$ ;
        sinaliza que está numa região de alta curvatura;
    senão
        se ( passou por uma região de alta curvatura )
            /* Passa por aqui ao sair da região de alta curvatura */
            seleciona o ponto de maior curvatura;
            avança os vetores  $a$ ,  $b$  e  $c$ ;
        senão
            se ( angulo_suave <  $vth2$  )
                seleciona o ponto de curva suave;
                avança os vetores  $a$ ,  $b$  e  $c$ ;
            senão avança os vetores  $a$  e  $b$ ;

```

Como os trechos com ângulos acentuados e os de ângulos suaves costumam aparecer numa mesma figura, a aproximação por segmentos de reta para ambos os casos é realizada ao mesmo tempo (ou de forma integrada).

A "Aproximação por segmentos de reta" tem início com o estado de alinhamento do vetor c numa das extremidades da curva digital e termina quando o vetor a atinge a outra extremidade da curva digital. A precisão do resultado final da aproximação depende dos valores adotados para os limiares $vth1$ e $vth2$ e para o comprimento dos vetores. Relaciona-se abaixo as vantagens e desvantagens desse algoritmo.

2.5.4 Vantagens do “Vector Tracer”

Entre as vantagens apresentadas por esse algoritmo temos a sua simplicidade, a velocidade de processamento e a contiguidade da aproximação obtida. Devido a essa característica de contiguidade podemos classificá-lo como um algoritmo de aproximação poligonal.

2.5.5 Desvantagens do “Vector Tracer”

Entre as desvantagens do método temos a necessidade de ajuste de parâmetros $vlen$ (tamanho mínimo do vetor), $vth1$ (limiar de ângulo acentuado) e $vth2$ (limiar de ângulo suave). Outra desvantagem dessa técnica é que depende da perseguição de contornos e não é particularmente adequado para implementações paralelas. Temos ainda uma outra desvantagem que em alguns casos há o aparecimento de distorções nas aproximações obtidas. Na seção 2.7 descreve-se alguns tipos de distorções que ocorrem no processamento digital de imagens.

2.6 Algoritmos híbridos de aproximação por segmentos de reta

Os algoritmos de implementações híbridas utilizam a combinação de várias técnicas. FORESTI *et al.* (1994) utilizaram uma combinação de “Labeling”⁶ e transformada de Hough. Nessa técnica, a imagem de entrada passa por um processo de “labeling” (rotulação), onde a lista dos rótulos dos componentes de imagem são aplicados à uma versão modificada de transformada de Hough. Essa técnica permite detectar colinearidade, paralelismo e convergência de bordas na imagem. Uma técnica que utiliza uma pré-segmentação da imagem em faixas horizontais de varredura, seguido da vetorização de segmentos de reta e por fim a utilização da transformada de Hough foi apresentada por Costa (1995a). Essa abordagem possui como vantagem a alta velocidade e a redução do espaço requerido para o acumulador de Hough. Outra técnica que combina a perseguição de contorno e a aplicação da transformada rápida de Fourier (FFT), realizando a aproximação por segmentos de reta utilizando técnicas de processamento de sinais foi proposta por CESAR e COSTA (1995). Essa técnica permite uma determinação rápida e precisa da curvatura através do cálculo da primeira e segunda derivadas através de FFT e IFFT. A utilização de filtro Gaussiano permite a operação do método em multiescala. Descreve-se a seguir uma das técnicas de aproximação por segmentos de reta utilizado nesta monografia.

⁶ Um algoritmo de “labeling” encontra todos componentes conectados numa imagem e atribui um único rótulo (“label”) para todos os “pixels” que pertencem ao mesmo componente (JAIN *et al.* 1995).

2.7 Transformada de Hough

A Transformada de Hough para detecção de retas é baseada no mapeamento de elementos característicos da imagem (geralmente seus contornos) para um espaço paramétrico, de forma que características especiais (por exemplo picos) neste último indiquem possíveis instâncias de segmentos de reta na imagem (COSTA, 1994). A transformada de Hough foi originalmente desenvolvida por P.V.C. Hough em 1962, (HOUGH, 1962), tendo sido inicialmente empregada em visão computacional para detecção de curvas paramétricas por HOUGH (1962) e DUDA e HART (1972) e generalizada mais recentemente para a detecção de formas ("shapes") não paramétricas (BALLARD, 1981). Ao longo do tempo outras técnicas de transformada de Hough tem surgido, tais como a transformada de Hough randômica (RHT) (XU, OJA, 1993) e transformada de Hough rápida (FHT) (GUIL et al, 1995). Na RHT os pontos que compõe as bordas na imagem são escolhidos de maneira aleatória, de forma que os pontos processados é um subconjunto dos pontos da imagem. Na FHT o espaço paramétrico é particionado em quadrantes de maneira que possa ser examinado em várias escalas. Esses quadrantes são processados subdividido-se até que a quantidade de votos seja menor que um limiar ou um determinado tamanho seja atingido. O método original de Hough utilizava a parametrização "slope-intercept" para mapear um ponto na imagem usando a equação:

$$y = mx + b \quad (22)$$

Assim, o espaço paramétrico é formado pelos parâmetros "slope" e "intercept" (lembrando que "slope" é a tangente do ângulo de inclinação da reta e o "intercept" é o ponto em que a mesma corta o eixo y). Este algoritmo apresenta como problema que tanto o "slope" como o "intercept" tendem ao infinito quando uma reta se aproximava da vertical. Este problema foi solucionado por (ROSENFELD, 1971) utilizando a troca dos parâmetros x e y .

DUDA e HART (1972) apresentaram uma variação da Transformada de Hough que utiliza a parametrização normal, que também evita o problema da representação comentada acima. Nessa abordagem transforma-se pontos (x_i, y_i) da imagem em curvas senoidais no plano paramétrico definidas pela expressão (2) e figura 2.9, apresentadas na seção referente a definição e geração de retas. Na implementação adotada, deslocamos a origem para o centro da imagem, obtendo dessa forma, uma utilização mais efetiva do espaço paramétrico θ - ρ e também uma redução de 50% no seu tamanho, (COSTA, 1992a) conforme pode ser observado na figura a seguir.

Figura 2.22 - Imagem (a), espaço paramétrico gerado para origem no canto inferior esquerdo (b) e espaço paramétrico para origem no centro (c).

COSTA (1992a) Verificou experimentalmente, dentro de certas condições, que as parametrizações "slope-intercept" e normal levam a desempenhos semelhantes respectivamente à precisão da detecção dos parâmetros.

Temos então as seguintes propriedades da modalidade de Transformada de Hough introduzida por DUDA e HART (1972), observando-se que isso é restrito para espaços contínuos:

1. Um ponto na imagem corresponde a uma curva senoidal no espaço paramétrico;
2. Um ponto no espaço paramétrico corresponde a um segmento de reta na imagem;
3. Pontos pertencentes ao mesmo segmento de reta na imagem correspondem a curvas que passam por um ponto em comum no espaço paramétrico;
4. Pontos pertencentes a mesma curva no espaço paramétrico correspondem a segmentos de reta através de um mesmo ponto na imagem.

Dessa forma, curvas no plano θ - ρ correspondentes a pontos colineares no plano x - y ou da imagem, possuem um ponto comum de interseção. Assim, este ponto no espaço θ - ρ define as coordenadas da reta passando através dos pontos colineares. Em 2.23.b temos a representação gráfica do espaço paramétrico na qual foram mapeados os pontos em 2.23.a, e em 2.23.c a reta detectada (correspondente às interseções em (b)) em coordenadas contínuas.

Figura 2.23 - Pontos colineares na imagem (a), curvas senoidais geradas no espaço de Hough (b) e a reta detectada (c).

A seguir descrevemos com maiores detalhes o mapeamento do espaço paramétrico de volta para a imagem através da explicação das expressões

utilizadas. Rearranjando-se as expressões (7) e (8) da definição de barras digitais (seção 2.1.7), obtém-se a expressão (23) a seguir :

$$x \cos(\theta) + y \sin(\theta) - \frac{w}{2} \leq \rho < x \cos(\theta) + y \sin(\theta) + \frac{w}{2} \quad (23)$$

COSTA (1995a) estendeu o trabalho de MCILROY (1984) com parametrização “slope-intercept” para a parametrização normal, de forma que a expressão anterior (23) leve a geração de curvas senoidais no espaço paramétrico de centro conforme a expressão (2) e largura que varia adaptativamente ao longo da variação de θ conforme a expressão (24) e restrita as condições dadas pelas expressões (25) e (26). Entretanto, para o espaço discreto, esta correspondência não é direta, pois quando uma reta é discretizada ela deixa de ser uma reta, e passa a ser composta por um conjunto de pontos discretos. Num segmento de reta discreto esses pontos estão contidos numa banda discretizada de largura w igual a um. Essas bandas são adotadas como modelo de retas a serem encontradas nas imagens. Assim, um ponto A pertencente a essa banda reta na imagem é mapeado em uma banda sinuosa no espaço paramétrico de Hough, conforme ilustrado na figura 2.24.

Um outro ponto B , contido nessa banda reta e representada na mesma figura, mapeia uma outra banda sinuosa no espaço paramétrico de Hough. Assim, concluímos que a região de interseção (RI) dessas bandas senoidais no espaço paramétrico de Hough define bandas retas que passam por A e B na imagem. A inclusão de novos pontos nessa banda definida pelas extremidades A e B vai restringindo a região RI . Idealmente, deveríamos amostrar cada RI para cada possível segmento digital.

Figura 2.24 - Imagem (x,y) (a) e espaço de θ - ρ discretos com a representação das bandas (b).

Esse mapeamento dos pontos na imagem para o espaço paramétrico de Hough é feito através das seguintes expressões:

$$x \cos(\theta) + y \sin(\theta) - \frac{m(\theta)}{2} \leq \rho < x \cos(\theta) + y \sin(\theta) + \frac{m(\theta)}{2} \quad (24)$$

onde:

$$m(\theta) = |w \cdot \cos(\theta)| \text{ para } (0 \leq \theta < 0.25\pi) \text{ ou } (0.75\pi < \theta < \pi) \quad (25)$$

$$m(\theta) = w \cdot \sin(\theta) \text{ para } (0.25\pi \leq \theta < 0.75\pi) \quad (26)$$

Note-se que a largura da banda $\rho_{min} \leq \rho < \rho_{max}$ ($\delta\rho$) varia adaptativamente com a inclinação θ . A razão dessa largura adaptativa é que qualquer segmento cuja largura de “intercept” (isto é, a largura no cruzamento com o eixo y para segmentos da classe C_x , e a largura no cruzamento com o eixo x para segmentos da classe C_y) se mantenha constante, a largura da banda dada por (24) deve variar. Isto pode ser melhor entendido através do gráfico a seguir, que representa dois casos de segmentos da classe C_x com inclinação distinta, mas que possuem a mesma largura de “intercept” w .

Figura 2.25 - Segmento da classe C_x com largura de "intercept" $w=2$ e inclinação $\theta=7.1^\circ$ em (a) e $\theta=41.2^\circ$ em (b)

Para o caso dos segmentos da classe C_y o funcionamento é análogo ao aplicado aos segmentos da classe C_x . Desta forma a largura de "intercept" w se mantém invariante no domínio da imagem, permitindo uma representação mais coerente dos segmentos digitais. Apresenta-se a seguir o algoritmo da transformada de Hough baseada na proposta de DUDA e HART (1972) e COSTA (1995a).

2.7.1 Algoritmo da Transformada de Hough

Para cada "pixel" preto da imagem faça

Para ($\theta=0$ até 180 , com passo de 1) faça

calcule $\rho = x.\cos(\theta) + y.\sin(\theta)$

calcule $\rho_{sup} = \rho + m(\theta)/2$

calcule $\rho_{inf} = \rho - m(\theta)/2$

incremente as posições ($\rho_{inf} \leq \rho < \rho_{sup}$) do espaço ($\theta - \rho$)

onde $m(\theta) = |w.\cos(\theta)|$ se $(0 \leq \theta < 0.25\pi)$ ou $(0.75\pi < \theta < \pi)$

$m(\theta) = w.\sin(\theta)$ se $(0.25\pi \leq \theta < 0.75\pi)$

Na figura 2.26.a encontramos a representação gráfica de uma imagem a qual foi aplicada a transformada de Hough e em (b) o espaço paramétrico de Hough resultante.

Figura 2.26 - Imagem original (a) e espaço paramétrico de Hough gerado (b).

Nessa figura (2.26.b) podemos notar a existência de faixas mais escuras (mais votos) que mostram as regiões onde ocorreram uma maior concentração de bandas senoidais mapeadas pelos pontos na imagem. Esses picos no acumulador de Hough significam que na imagem há probabilidade de existirem retas com esses parâmetros θ e ρ . O mapeamento do espaço paramétrico de volta para a imagem é feito para cada pico da matriz acumuladora que ultrapassar um determinado valor de limiar ("threshold").

2.7.2 Vantagens

Entre as vantagens da TH estão a relativa facilidade em paralelizar este algoritmo e uma certa imunidade ao ruído de fundo. Outras vantagens incluem a possibilidade de se detectar múltiplas retas digitais simultaneamente, certa tolerância à distorção e oclusão na imagem e também o de realizar a segmentação e o ajuste de curvas ao mesmo tempo. Há ainda a vantagem de não necessitar de perseguição de contornos. DUDA e HART (1972), GONZALES e WINTZ (1987), ILLINGWORTH e KITTLER (1988) e COSTA e SANDLER (1993) tratam do algoritmo com maiores detalhes.

2.7.3 Desvantagens

Algumas das limitações da Transformada de Hough são que o resultado é sensível à quantização dos parâmetros θ e ρ , isto é, deve-se ter cuidado na escolha desses parâmetros para que não ocorra nem a super-amostragem e

nem a sub-amostragem. O espaço de Hough utilizando a super-amostragem encontra-se representado na figura 2.27.a, e em (b) o correspondente espaço com sub-amostragem. Na super-amostragem há a replicação de picos no espaço de Hough dando origem a mais de um segmento (ver figura 2.27a), enquanto que na sub-amostragem pode haver a perda ou a quebra dos segmentos obtidos (ver figura 2.27b).

Como a TH não leva em conta o aspecto da contigüidade dos pontos colineares temos os seguintes problemas: a TH não acha as extremidades dos segmentos, pontos colineares esparsos podem dar origem a picos no espaço de Hough e segmentos distintos podem ser detectados como um só. Na figura 2.29 podemos ver a representação desses três casos que levam à detecção de um único segmento (d).

Figura 2.27 - Ilustração do problema da sub-amostragem (a) e da super-amostragem (b)

Figura 2.28 - Retas replicadas obtidas devido a super-amostragem (a) (tendo como imagem original a figura 2.27a), quebra ou perda na detecção de segmentos (b).

Figura 2.29 - Segmento de reta (a), pontos esparsos (b), vários segmentos distintos (c) e o segmento de reta resultante da análise do espaço de Hough (d).

Além desses problemas, há a interferência entre pontos pertencentes a objetos distintos como os representados nas figuras 2.30 (b) e (d) que geram uma distribuição de pontos indesejáveis ("ruído indesejável") no espaço de Hough (ver também as figuras 2.26a e 2.26b). A figura 2.30.d mostra o aparecimento de interferência causada pela adição de um segundo objeto (figura 2.30.c) na imagem da figura 2.30.a.

Figura 2.30 - Imagem original (a) e espaço de Hough correspondente (b), objeto adicionado à imagem (a) em (c) e acumulador de Hough resultante (d)

Dessa forma torna-se necessário um pós-processamento para validar as indicações da transformada de Hough, bem como encontrar as extremidades dos segmentos. COSTA (1992b) propõe técnicas como a análise de conectividade através da construção do histograma unidimensional e o preenchimento do "gap" para solucionar esse problema, apresentando ainda algumas variantes de Transformada de Hough e suas aplicações e um mecanismo para redução e agrupamento dos segmentos detectados de acordo

com a sua similaridade. Algumas dessas soluções, implementadas para esta monografia serão descritas a seguir.

2.7.4 Procedimentos Pós-Transformada de Hough

Descreve-se nesta seção algoritmos utilizados para contornar algumas das deficiências da Transformada de Hough comentadas na seção anterior.

2.7.4.1 Backmapping

GERIG e KLEIN (1986) introduziram um algoritmo denominado “Backmapping” cujo objetivo é enfatizar as partes mais relevantes na matriz acumuladora, minimizando o ruído de fundo que é característico na TH (GERIG, KLEIN, 1986)(GERIG, 1987). Utiliza-se um novo acumulador de Hough com as mesmas dimensões do anterior e executa-se o mesmo procedimento que a TH (GERIG, KLEIN, 1986). Isto é, para cada θ no intervalo $[0, \pi)$ determina-se ρ_{inf} e ρ_{sup} . Porém em vez de atribuir votos nesse outro acumulador de Hough, armazenamos os parâmetros que identificam a posição no primeiro acumulador de Hough que possui o maior voto. Depois de localizada a posição de maior voto obtido pelo mapeamento para o espaço paramétrico de Hough, de um dado “pixel” na imagem, essa mesma posição é agora incrementada no segundo acumulador de Hough. Isso faz com que mantenhamos apenas os pontos “mais prováveis” de se encontrar segmentos realmente significativos. Na figura 2.31 observamos o resultado da utilização do algoritmo de “Backmapping” onde pode-se constatar a diminuição dos pontos significativos da matriz acumuladora. Esse processo têm como característica principal a redução significativa do tempo de análise do espaço de Hough e permite ainda uma diminuição do valor de “threshold”.

Figura 2.31 - Espaço paramétrico resultante da transformada de Hough aplicada na imagem da figura 2.28(a) contendo 16.195 pontos maiores que 5 (a) e resultado após o "Backmapping" com 85 valores maiores que 5 (b).

Ainda com relação a figura 2.30b, ao se considerar apenas o maior valor dentro de cada banda senoidal gerada, obtivemos a contagem de 85 pontos no acumulador com valor acima de 5. Ao se considerar os dois maiores valores e os três maiores valores contidos em cada banda senoidal gerada, a contagem obtida foi de 170 e 254 respectivamente em relação aos valores do acumulador. Apresenta-se a seguir o algoritmo de "Backmapping" proposto por GERIG e KLEIN (1986).

Algoritmo do Backmapping

Para cada "pixel" preto da imagem faça

Para ($\theta=0$ até 180) faça

$$\rho = x.\cos(\theta) + y.\sin(\theta)$$

$$\rho_{sup} = \rho + m(\theta)/2$$

$$\rho_{inf} = \rho - m(\theta)/2$$

Armazene os pontos mapeados para o espaço de Hough

(para cada teta os valores do intervalo ($\rho_{inf} \leq \rho < \rho_{sup}$))

Selecione os n maiores valores e incremente no segundo espaço de Hough.

onde $m(\theta) = |w.\cos(\theta)|$ se ($0 \leq \theta < 0.25\pi$) ou ($0.75\pi < \theta < \pi$)

$m(\theta) = w.\sin(\theta)$ se ($0.25\pi \leq \theta < 0.75\pi$)

2.7.4.2 Análise de conectividade

Através da análise de conectividade pode-se confirmar a existência dos segmentos de reta, assim como encontrar as suas extremidades. Gerando-se as barras digitais correspondentes às posições do acumulador de Hough que possuem valor acima de um limiar ($hth2$), isso é com inclinação θ e distância ρ , obtém-se o histograma unidimensional, que é a projeção de uma barra sobre o eixo x ou y dependendo da sua classe conforme exemplo dado na figura 2.31. Utiliza-se então um algoritmo que verifica a presença de uma barra com as características do histograma unidimensional gerado, confirmando também as extremidades e verificando-se o comprimento mínimo ($hth1$) dos sub-segmentos encontrados. Esse algoritmo verifica também a existência de descontinuidades nos segmentos, isto é, pequenas lacunas ou "gaps" que aparecem nos segmentos devido a problemas de ruído na imagem ou mesmo resultantes do processo de extração de segmentos. A representação gráfica do histograma unidimensional e do "gap" se encontra ilustrada na figura 2.31.

Figura 2.32 - Histograma unidimensional (projeção da barra sobre o eixo x ou y dependendo da inclinação), resultado do histograma para $\text{gap}=0$ e $\text{gap}=1$.

Os “gaps” que possuem comprimento maior que o permitido serão considerados como a separação de segmentos independentes, caso contrário serão preenchidos e dão origem a segmentos considerados como um de maior comprimento. Os segmentos de comprimento menor que um valor preestabelecido são desprezados, enquanto que os validados têm as coordenadas das suas extremidades armazenadas numa lista ligada. Deve-se observar que tanto o comprimento dos segmentos como o do “gap” podem ser considerados como uma distância Euclidiana, assim os valores tornam-se adaptativos e o valor obtido é compensado de acordo com a variação da inclinação da reta. Nessa monografia, o valor adaptativo é adotado apenas para o cálculo do comprimento dos segmentos. A utilização do cálculo do “gap” adaptativo é comentada no capítulo 6. As expressões (27) e (28) a seguir, são utilizadas para o cálculo do comprimento dos segmentos, considerando-se p_i e p_{i+k} como as extremidades do segmento de reta, para segmentos das classes C_x e C_y respectivamente.

$$l = \left| \frac{p_i - p_{i+k}}{\sin(\theta)} \right| \quad (27)$$

$$l = \left| \frac{p_i - p_{i+k}}{\cos(\theta)} \right| \quad (28)$$

Essas expressões são empregadas no algoritmo de Análise de Conectividade descrito a seguir.

Algoritmo da Análise de Conectividade

(1) Armazene cada elemento do segundo acumulador de Hough maior que o limiar *hth2* na fila de prioridade descendente⁷

Para cada elemento retirado dessa fila de prioridade, faça :

- (2) gere a barra digital de largura *w* com inclinação θ e distância ρ , na forma de um histograma unidimensional;
- (3) confirme a existência dos segmentos e das suas extremidades preenchendo o "gap" adaptativo de acordo com o valor preestabelecido;
- (4) se o segmento for maior que o limiar *hth1* armazene as coordenadas das extremidades numa lista; apague da imagem os pontos correspondentes ao segmento;

A transformada de Hough gera uma aproximação por segmentos de reta descontínua. Isto, entretanto não configura uma desvantagem, pois dependendo da aplicação isto pode não ser problemático.

Para uma revisão completa sobre os vários tipos de transformada de Hough recomenda-se a leitura dos trabalhos de ILLINGWORTH e KITTLER (1988), COSTA (1992a) e LEAVERS (1993).

⁷ Uma fila de prioridade descendente é uma estrutura de dados onde a recuperação, seguida da remoção do elemento é feita a partir do maior elemento (TENEMBAUM, 1990).

2.8 Distorções em imagens

Descrevemos nessa seção alguns tipos de distorções provocadas pelos algoritmos utilizados em processamento digital de imagens. O processo de afinamento (SMITH, 1987) e o de vetorização (HORI e OKAZAKI, 1990)(AOYAMA, KAWAGOE, 1991) geralmente introduzem distorções na informação obtida. Existem vários tipos de distorções que ocorrem na vetorização de imagens ("vector-to-raster conversion") e que afetam a qualidade dos resultados obtidos pelos algoritmos em visão computacional (HORI, OKAZAKI, 1990)(AOYAMA, KAWAGOE, 1991), são eles:

1. Distorção da forma do objeto decorrente do processo de afinamento ou de outro método de extração do esqueleto das regiões da imagem e do processo de vetorização das bordas. Esse tipo de erro ocorre principalmente na vizinhança dos pontos de cruzamento dos objetos (ver figuras 2.33.(a) - (d)).
2. Distorção geométrica provocada devido ao erro na escolha dos vértices e pontos de inflexão. Algumas figuras possuem vários vetores paralelos que exprimem a sua forma. Essa propriedade geométrica nem sempre é preservada pelos algoritmos (ver figuras 2.34.(a) - (b)).
3. Erro na extração dos vértices nos pontos de inflexão ou nos locais de máxima curvatura, e na seleção inadequada de vértices na aproximação por segmentos de reta (ver figuras 2.34.(c) - (h)).

Figura 2.33 - Distorções causadas pelo processo de afinamento.

Figura 2.34 - Distorções causadas pelo processo de aproximação poligonal.

Devido a essas distorções introduzidas e/ou encontradas nos contornos presentes nas imagens digitais, diversos autores têm proposto formas de minimizar ou resolver este problema (WEISS, BOLDT, 1986)(HORI, OKAZAKI, 1990)(AOYAMA, KAWAGOE, 1991).

2.9 Aplicações

A figura 2.35 apresenta alguns dos produtos que podem ser obtidos a partir dos segmentos aproximados produzidos pelos algoritmos de aproximação por segmentos de reta implementados no presente trabalho. Temos o histograma polar de orientação (COSTA, 1991), a reconstrução de imagens em termos de segmentos de reta, e também as estatísticas dos segmentos obtidos, tais como o maior segmento, o menor segmento, a média dos comprimentos dos segmentos obtidos. Com algum processamento adicional, podemos obter resultados como descritores de textura (MARANA, COSTA e LOTUFO, 1995), a identificação de cantos (ENGELBRECHT, WAHL, 1988)), e polígonos (SANDLER *et al*, 1993) formados pelos segmentos extraídos.

Figura 2.35 - Descrição dos produtos obtidos pela análise dos segmentos aproximados.

Dentre esses produtos mostrados na figura, a reconstrução em termos de segmentos de reta é implementada com o objetivo de comparar dos algoritmos de aproximação por segmentos de reta avaliados, quanto a fidelidade das aproximações obtidas, em relação à imagem original. O histograma polar de orientações, descrito na seção 2.1.6, é implementado na forma de uma aplicação em citometria e é descrito a seguir.

Aplicação do histograma polar em neuromorfometria

Os histogramas polares de orientação são importantes para a caracterização de uma imagem em termos da distribuição de orientação predominantes. Essa importância pode ser melhor compreendida através de um exemplo como descrito por COSTA (1991) onde se emprega o histograma polar de orientações para a análise das orientações preferenciais das partículas do agregado. Encontra-se descrito que a resistência mecânica é menor no sentido dessas orientações preferenciais e é um subsídio importante a ser utilizado no projeto de fundações em construção civil. Essa análise morfométrica, feita de maneira automática, com o uso do computador visa dar maior confiabilidade, repetibilidade e rapidez a esse processo (COSTA, 1991). A partir dos segmentos aproximados obtidos pelos algoritmos, gera-se o histograma polar de orientações para as imagens. Os algoritmos utilizados na obtenção desses histogramas serão descritos na seção 3.3, e as imagens obtidas de tecidos de rato, serão apresentadas na seção 4.3, juntamente com os resultados. Apresentamos a seguir detalhes da implementação dos diversos algoritmos utilizados nessa monografia.

3. IMPLEMENTAÇÃO

Nesta seção descreve-se a implementação dos algoritmos da transformada de Hough e do “Vector Tracer” como utilizados neste trabalho. Descrevemos também os algoritmos utilizados na comparação desses métodos e os algoritmos implementados para a aplicação apresentada no capítulo 5. O sistema desenvolvido para a avaliação dos métodos é composto de 4 módulos básicos, divididos em duas classes de algoritmos, que são enumerados abaixo:

Algoritmos de detecção de segmentos de retas

- 1) Módulo de Transformada de Hough;
- 2) Módulo de segmentação dos contornos + “Vector Tracer”;

Algoritmos usados na comparação de desempenho

- 3) Gerador de segmentos de reta com parâmetros aleatórios;
- 4) Reconstrução de imagens a partir de segmentos de reta.
- 5) Módulo para análise estatística dos resultados.
- 6) Histograma polar de orientação.

A descrição de cada um desses módulos é feita nas seções a seguir.

3.1 Algoritmos de detecção de segmentos de retas

Descreve-se nessa seção os algoritmos de detecção de segmentos de reta empregados.

3.1.1 Módulo de perseguição de contornos + “Vector Tracer”

Este módulo é composto pelo algoritmo de perseguição de contornos apresentado na seção anterior e que funciona de forma integrada ao algoritmo de aproximação por segmentos de reta denominado “Vector Tracer” (AGUI *et al.*, 1982). Na figura 3.1 é apresentado o diagrama esquemático do processo de aproximação de contornos por segmentos de reta utilizando o “Vector Tracer”.

Figura 3.1 - Diagrama esquemático do módulo usando o algoritmo “Vector Tracer”.

Nessa figura adotou-se retângulos para representar processos e elipses para as estruturas de dados em geral. Utilizamos também setas tracejadas para representar o fluxo de dados e setas cheias para o fluxo de programa. Nessa figura, *borda* significa a imagem de entrada após o processo de afinamento, caso seja necessário. *Lista* é a representação dos segmentos aproximados na forma de uma lista encadeada. O algoritmo “Vector Tracer” necessita do ajuste de três parâmetros : *vlen* (comprimento mínimo dos

segmentos aproximados), *vth1* (limiar de ângulo acentuado), *vth2* (limiar de ângulo suave). A versão modificada do “Vector Tracer” que introduzimos nesta monografia será denominada “Vector Tracer 2” que necessita de um quarto parâmetro *vth3* (limiar de ângulo suave) que será descrito mais adiante nesta seção. Esses parâmetros estão relacionados na tabela a seguir.

Tabela 3.1 - Descrição dos parâmetros e sua função e em que “VectorTracer” é usado.

Parâmetro	Descrição da função dos parâmetros	Vector Tracer1	Vector Tracer2
<i>vlen</i>	comprimento dos vetores de controle <i>a</i> , <i>b</i> e <i>c</i>	x	x
<i>vth1</i>	limiar de ângulo acentuado entre vetores <i>a</i> e <i>b</i>	x	x
<i>vth2</i>	limiar de ângulo suave entre os vetores <i>b</i> e <i>c</i>	x	x
<i>vth3</i>	limiar de ângulo suave no reprocessamento		x

Apresentamos a seguir as descrições dos algoritmos e a motivação para a modificação do algoritmo “Vector Tracer”.

Descreve-se abaixo, o trecho correspondente ao passo (3) do algoritmo de perseguição de contornos de LI *et al* (1982) e modificado segundo o critério de escolha do próximo “pixel” de BELL e PAU (1990). Nesse trecho de programa do algoritmo de perseguição de contornos é feita a escolha dos parâmetros para a aproximação poligonal por meio do “Vector Tracer”.

Se *pnum* (quantidade de pontos) é maior que 1

se *pnum* > 3.*vlen*

VectorTracer (*vth1*, *vth2*, *vlen*);

senão

VectorTracer (*vth1*, *vth2*, 1);

converte as coordenadas para origem no centro da imagem;

armazena os segmentos obtidos na lista ligada;

incrementa o contador de segmentos;

Quando a quantidade de pontos perseguidos na imagem (*pnum*) for maior que “1” (um), realiza-se a aproximação, caso contrário, o trecho perseguido é considerado como um ponto isolado e é descartado. Se a

quantidade de pontos for maior que três vezes o tamanho mínimo escolhido para os vetores auxiliares de aproximação (vetores a , b e c), o “Vector Tracer” é chamado para aproximar com comprimento de vetor igual a v_{len} , caso contrário o mesmo é chamado com comprimento de vetor igual a “1” (um).

Após experimentos com o “Vector Tracer” observou-se que restavam vários segmentos de reta em contornos quase retos, o que ocorre devido ao processo de quantização. Em determinados ângulos de inclinação ocorria a detecção de uma quantidade maior de segmentos, conforme pode ser observado na figura abaixo.

Figura 3.2 - Segmento original (a) e (b) resultado aproximado por 9 segmentos, (c) imagem original e (d) resultado aproximado por 15 segmentos.

Desta forma optou-se em reprocessar estes contornos de forma a obter uma aproximação com uma quantidade menor de segmentos de reta.

Descreve-se a seguir, o trecho correspondente ao passo 3 do algoritmo de perseguição de contornos onde é realizado a escolha dos parâmetros para a aproximação poligonal por meio do “Vector Tracer”. Quando a quantidade de pontos perseguidos for maior que “1” (um), realiza-se a aproximação, caso contrário, o trecho perseguido é considerado como um ponto isolado e é descartado. Se a quantidade de pontos for maior que três vezes o tamanho mínimo escolhido para os vetores auxiliares de aproximação (vetores a , b e c), o “Vector Tracer” é chamado para aproximar com comprimento de vetor igual a v_{len} , caso contrário o mesmo é chamado com comprimento de vetor igual a “1” (um).

```

Se  $pnum > 1$  (quantidade de pontos)
    Se  $pnum > vlen * 3$ 
        VectorTracer (  $vth1, vth2, vlen$  );
    senão
        VectorTracer (  $vth1, vth2, 1$  );
    Se  $pnum2 > 3$  (pontos resultantes da aproximação)
        copia os pontos em  $pnt2$  para  $pnt$  e faz  $pnum = pnum2$ ;
        VectorTracer2 (  $vth3, 1$  );
    converte as coordenadas para origem no centro da imagem;
    armazena os segmentos obtidos em uma lista ligada;
    incrementa o contador de segmentos;

```

Nesse trecho de programa pnt é a estrutura que contém as coordenadas perseguidas na imagem e $pnum$ indica a quantidade desses pontos. Os vértices resultantes da aproximação por segmentos de reta são armazenados em $pnt2$ e $pnum2$ indica a quantidade desses vértices. Apresentamos a seguir o algoritmo “Vector Tracer 2”, que foi introduzido pelo autor desta monografia, com o intuito de remover segmentos colineares da aproximação obtida.

3.1.1.1 Algoritmo do “Vector Tracer 2”

```

posiciona vetores  $a$  e  $b$ ;
faça até o fim dos pontos perseguidos
    calcula ângulo_suave entre os vetores  $a$  e  $b$ ;
    se ( ângulo_suave <  $vth3$  )
        marca e armazena como um vértice;
    senão
        avança os vetores;

```

A seguir apresentamos na figura 3.3 os resultados da aproximação dos segmentos de reta mostrados nas figura 3.2.a e (c). Pode-se observar que os

segmentos foram aproximados por uma quantidade menor de segmentos quando utilizamos o “Vector Tracer 2”. No caso específico da figura 3.2.a e (c), os segmentos foram aproximados por apenas um segmento.

Figura 3.3 - Resultados (a) e (b) aproximados através de um segmento de reta das figuras 3.2.a e 3.2.b respectivamente.

Apresentamos a seguir, a título de ilustração, o diagrama correspondente a quantidade de segmentos detectados pelas duas técnicas “Vector Tracer” em função do ângulo de inclinação dos segmentos gerados através dos algoritmos descritos na seção 3.2.1, para um conjunto de 1000 segmentos gerados com comprimento mínimo de 20 “pixels”.

Figura 3.4 - Quantidade de segmentos detectados em função do ângulo de inclinação dos segmentos para as duas versões do “Vector Tracer”.

Apresentamos a seguir a representação abreviada para caracterização de resultados dos algoritmos “Vector Tracer”.

3.1.1.2 Representação abreviada dos métodos para “Vector Tracer”

De forma análoga a representação dos métodos adotada para a transformada de Hough, essa representação tem o intuito de identificar cada resultado obtido pelos processos através de ênuplas. Cada ênupla contém o valor atribuído ao parâmetro correspondente. Adotamos a seguinte representação para a aproximação obtida utilizando o “Vector Tracer”.

$$VT1 \{ vlen=2, vth1=0.4, vth2=0.99 \}$$

$$VT2 \{ vlen=3, vth1=0.4, vth2=0.99, vth3=0.995 \}$$

onde : VT1 é o “Vector Tracer” original e VT2 é o “Vector Tracer” modificado para esta monografia, e $vlen$ é o comprimento mínimo do vetor de aproximação, $vth1$ é o limiar de ângulo agudo e $vth2$ é o limiar de ângulo suave. Ao longo da monografia os termos VT1 e Vector1 serão usados para designar o “Vector Tracer” original, enquanto os termos VT2 e Vector2 poderão ser usados para designar o “Vector Tracer” modificado.

3.1.2 Módulo de transformada de Hough

Esse módulo é composto por três sub-módulos : a transformada de Hough propriamente dita, o “Backmapping” e a Análise de conectividade. A estrutura deste módulo é apresentada na figura 3.5.

Figura 3.5 - Diagrama esquemático do módulo de detecção de segmentos de reta usando a transformada de Hough.

Os algoritmos que constituem esse módulo utilizam vários parâmetros que devem ser ajustados para uma melhor utilização. Esses parâmetros estão relacionados na tabela a seguir.

Tabela 3.2 - Descrição dos parâmetros e os módulos onde são utilizados.

Parâmetro	Descrição da função do parâmetro	Hough	Back.	Análise
<i>hth1</i>	comprimento mínimo dos segmentos			x
<i>hth2</i>	limiar do acumulador de Hough		x	x
<i>ene</i>	enésimo maior pico do acumulador		x	
<i>gap</i>	“gap” máximo que será preenchido			x
<i>w</i>	largura da banda reta na imagem	x	x	x

A descrição desses parâmetros será feita na seção correspondente ao módulo em que são utilizados.

3.1.2.1 Transformada de Hough

A transformada de Hough utilizada neste trabalho é baseada na proposta de DUDA e HART (1972) e Costa (1995a) e utiliza a parametrização normal dada pela expressão (2) e representada na figura 2.9.

Nesta implementação a origem foi deslocada para o centro da imagem para um aproveitamento mais efetivo do espaço paramétrico de Hough, conforme justificativa apresentada na seção 2.2.4. Dessa forma os limites da imagem ficam como na representação dada na figura 3.6.

Figura 3.6 - Limites inferior e superior dos eixos x e y para uma imagem com a origem deslocada para o centro.

Para o cálculo desses limites utilizam-se as expressões (29) a (32) abaixo, considerando-se que $XSize$ é o tamanho do eixo x na imagem e $YSize$ é o tamanho do eixo y na imagem.

$$XP2 = \text{int} (Xsize/2) \quad (29)$$

$$XN2 = -(XSize-XP2-1) \quad (30)$$

$$YP2 = \text{int} (YSize/2) \quad (31)$$

$$YN2 = -(YSize-YP2-1)$$

(32)

onde : $\text{int} ()$ retorna a parte inteira da expressão.

Além disso, nessa implementação da transformada de Hough, utilizaram-se tabelas para os senos ($tsin$), os cossenos ($tcos$) e largura da banda (bwi) para o mapeamento da banda senoidal no espaço paramétrico de Hough, conforme descritas em maiores detalhes na seção 3.52.

3.1.2.2 Resolução do espaço de Hough

Nessa implementação adotou-se uma quantização uniforme para a representação do espaço paramétrico ou acumulador de Hough (θ - ρ). Esse espaço é construído na forma de uma matriz bidimensional. A transformada de Hough foi implementada de forma a permitir duas resoluções diferentes para o espaço paramétrico. Uma delas (TH1) utiliza resolução simples para o espaço paramétrico (HS1), com espaçamento de 1 em 1 tanto para o parâmetro θ como para o parâmetro ρ . A outra transformada de Hough (TH2) utiliza resolução dupla para o espaço paramétrico (HSD1), ou seja com espaçamento de meio em meio para os parâmetros θ e ρ . Essas duas resoluções foram escolhidas com o intuito de se verificar a influência desses parâmetros no tempo de processamento e nas aproximações obtidas. Abaixo descrevemos o cálculo do tamanho do espaço de Hough para a resolução simples e dupla.

Resolução simples

Para o espaço de Hough com resolução simples (HS1), o parâmetro θ é amostrado no intervalo $[0, \pi)$ com passo de um em um grau e o parâmetro ρ é amostrado com passo de um em um no intervalo $[-ROMAX, +ROMAX]$ que depende das dimensões da imagem. O cálculo das dimensões do acumulador de Hough com resolução simples é dado pelas expressões a seguir (33) e (34) a seguir.

$$ROMAX = round (XP2 * sin(\pi/4) + YP2 * cos(\pi/4)) \quad (33)$$

onde : *round* é uma função que retorna o inteiro mais próximo.

$$ROMAX2 = ROMAX*2+1 \quad (34)$$

Assim as dimensões do espaço paramétrico θ - ρ denominado HS1 (espaço de Hough com resolução simples) ficam $[0,179]$ para o eixo θ e $[0,ROMAX2-1]$ para o eixo ρ .

Resolução Dupla

No caso do espaço paramétrico com resolução dupla (HSD1), o parâmetro θ é amostrado no intervalo $[0,\pi)$ com passo de meio em meio grau e o parâmetro ρ é amostrado de meio em meio no intervalo $[-ROMAXD,+ROMAXD]$ que depende das dimensões da imagem. O cálculo das dimensões do espaço paramétrico de Hough é dado pelas expressões (35) e (36) a seguir :

$$ROMAXD = round ((XP2*\sin(\pi/4) + YP2*\cos(\pi/4)) *2) \quad (35)$$

$$ROMAXD2 = ROMAXD*2+1 \quad (36)$$

onde : *round* é uma função que retorna o inteiro mais próximo.

Assim as dimensões do espaço paramétrico θ - ρ denominado HSD1 (espaço de Hough com resolução dupla) ficam $[0,359]$ para o eixo θ e $[0,ROMAXD2-1]$ para o eixo ρ . Descreve-se abaixo os parâmetros a serem ajustados para a utilização das transformadas de Hough TH1 e TH2, e também a forma adotada para a otimização do cálculo dos senos e cosenos na transformada de Hough.

Parâmetros da transformada de Hough

A transformada de Hough utiliza como parâmetro a largura da banda senoidal gerada (w) no espaço paramétrico de Hough. Essa largura é obtida

através da tabela *bwi* para o caso da resolução simples e *bwid* para resolução dupla.

3.1.2.3 “Backmapping”

Na presente monografia adotou-se o algoritmo de “Backmapping”, idealizado por (GERIG, KLEIN, 1986)(GERIG, 1987), e descrito na seção 2.2.5. Esse processamento é efetuado pelas funções Backmap1 utilizando resolução simples para o espaço paramétrico de Hough (HS1) após a execução de TH1 e tendo como saída HS2 (espaço de Hough com resolução simples) com as mesmas dimensões de HS1. O Backmap2 utilizando resolução dupla para o espaço paramétrico de Hough (HSD1) após a execução de TH2 utiliza como saída o HSD2 (espaço de Hough com resolução dupla) com as mesma dimensões de HSD1.

Parâmetros para “Backmapping”

O algoritmo de “Backmapping” também utiliza o parâmetro *w* (largura da banda senoidal gerada no espaço de Hough), assim como na transformada de Hough e utiliza ainda o parâmetro *hth2* (valor de limiar do acumulador de Hough) acima do qual os picos nesse espaço serão considerados como significativos. A largura da banda é obtida através de da tabela *bwi* para o caso da resolução simples e *bwid* para resolução dupla.

3.1.2.4 Análise de conectividade

A análise de conectividade é um procedimento executado de forma a confirmar a existência dos segmentos de reta na imagem, assim como encontrar as suas extremidades. Nesse processo, busca-se segmentos de comprimento maior que um valor preestabelecido e os segmentos detectados, são removidos da imagem para que o mesmo não seja replicado posteriormente. Esta abordagem permite ainda que pequenas lacunas

chamadas “gaps” que surgem nas bandas retas sejam preenchidas (COSTA, 1992a). O objetivo desse procedimento é evitar que pequenas interrupções ou quebras dos segmentos resultantes dos processamentos anteriores, ou mesmo do processo de extração do segmento, produzam segmentos quebrados quando detectados no processo de Análise de Conectividade.

Para cada coordenada θ - ρ do acumulador de Hough onde se tem um valor acima do limiar $hth2$, gera-se o histograma unidimensional e aplica-se o preenchimento das lacunas ou “gaps”. Encontra-se a seguir os segmentos de reta de comprimento Euclidiano maior que $hth1$, armazenando-se na lista ligada e removendo-se da imagem os pontos confirmados, para evitar a replicação dos segmentos. Para essa implementação foi estabelecido um critério para a ordem na geração do histograma unidimensional. Os picos no acumulador de Hough juntamente com os parâmetros θ e ρ são armazenados numa estrutura dinâmica em ordem decrescente em relação apenas ao valor do pico no acumulador. A estrutura utilizada será descrita na seção correspondente a descrição das estruturas de dados empregadas.

Parâmetros para Análise de Conectividade

Nesse módulo, os parâmetros utilizados são o limiar de comprimento mínimo de um segmento a ser detectado ($hth1$) e o limiar mínimo do acumulador de Hough a ser considerado para a análise de conectividade ($hth2$) o tamanho do *gap*, e a largura da banda reta a imagem (w). Descreve-se a seguir a representação adotada para os resultados da transformada de Hough.

3.1.2.5 Representação abreviada dos métodos por transformada de Hough

Essa representação tem o intuito de identificar cada resultado de aproximação obtido pelos processos através de ênuplas. Cada ênupla contém o valor atribuído ao parâmetro correspondente para a obtenção do resultado que se quer identificar. Adotamos a seguinte representação para as aproximações obtidas pela transformada de Hough :

Precisão adotada { *hth1*, *hth2*, *w*, *gap*, *bm* }

Então, para descrever os segmentos de reta obtidos pela transformada de Hough com resolução simples (HT1) utilizamos :

HT1 {*hth1*=3, *hth2*=5, *w*=1.0, *gap*=2, *bm*=1}

onde : utilizou-se o espaço de Hough com resolução simples, *hth1*=3, *hth2*=5, *w*=1.0, *gap*=2 e com utilização de "Backmapping".

E para os segmentos de reta obtidos pela transformada de Hough com resolução dupla (HT2) representamos como :

HT2 {*hth1*=3, *hth2*=5, *w*=1.0, *gap*=2, *bm*=0}

onde : utilizou-se resolução dupla para o espaço de Hough, *hth1*=3, *hth2*=5, *w*=1.0, *gap*=2 e sem a utilização de Backmapping.

Os termos HT1 e Hough1 poderão ser usados para referenciar de forma abreviada o algoritmo da transformada de Hough usando acumulador de resolução simples, enquanto que os termos HT2 e Hough2 serão usados para referenciar a transformada de Hough que utiliza o acumulador de resolução dupla.

3.2 Algoritmos para avaliação e comparação de desempenho

Descreve-se nessa seção os algoritmos utilizados para comparação de desempenho. Essa classe engloba os algoritmos empregados na geração dos números pseudo-aleatórios, dos segmentos de reta, assim como para gerar as estatísticas desejadas.

3.2.1 Gerador de segmentos de reta com parâmetros aleatórios

O gerador de segmentos de reta com parâmetros aleatórios funciona baseado em dois módulos : o gerador de números aleatórios e o gerador de segmentos de reta. Descreve-se a seguir esses módulos.

Gerador de números pseudo-aleatórios

O gerador de números aleatórios portátil baseado na função *ran3* (PRESS *et al.*, 1992), foi traduzido para a linguagem C. Trata-se de um gerador de números cuja saída são números no intervalo [0.0,1.0] e distribuição com desvio uniforme. O gerador original foi separado em duas funções sendo uma para a inicialização e a outra apenas para a obtenção dos números pseudo-aleatórios.

A inicialização e a obtenção dos números aleatórios é feita da seguinte forma:

```
sran3 ( inum );  
num = (double) ran3 () * MAX;
```

onde : *inum* (número de inicialização), deve ter valor negativo e MAX pertence ao intervalo dado por [0, MAX] a ser gerado. Nesse exemplo o número aleatório é obtido na variável *num*.

Gerador de segmentos de reta

Este gerador produz segmentos de reta de acordo com a inclinação θ (da reta normal em relação ao eixo x no sentido anti-horário e no intervalo $[0, \pi)$) e distância ρ (distância algébrica em relação à origem) fornecidos pelo gerador de números pseudo-aleatórios descrito na seção anterior. Os segmentos de reta gerados seguem a quantização *Interseção com grade*. Os parâmetros para a geração dos segmentos de reta estão definidos no seguinte intervalo:

$$\theta \in [0, \pi] \text{ e}$$

$$\rho \in [-\rho, \rho] \text{ onde } \rho = -XN2^8.$$

Por exemplo: para uma imagem quadrada de tamanho 128x128, $XSize=128$ e $YSize=128$, então $XN2=-63$ $\rho \in [-63, 63]$

O tamanho dos segmentos gerados pode ser definido em termos da quantidade de “pixels” que formam o segmento, ou termos da distância Euclidiana de suas extremidades. Nesta monografia o tamanho para os segmentos gerados foi definido em termos de “pixels”. Isto é cada segmento gerado deve possuir pelo menos n “pixels” de comprimento, caso contrário o mesmo é descartado. O objetivo é não permitir que segmentos de reta muito pequenos que caem nos cantos da imagem sejam considerados para as estatísticas geradas, ocasionando uma interpretação errônea das mesmas. O algoritmo utilizado verifica se os segmentos gerados (que sempre caem dentro da imagem), possuem a quantidade mínima de “pixels” exigida, caso não possuam são descartados e o algoritmo gera um novo segmento.

⁸ Esta variável, assim como o seu cálculo foi apresentado na seção 3.1.1.1.

3.2.2 Algoritmo de reconstrução de contornos

O algoritmo de reconstrução de contornos a partir dos segmentos aproximados obtidos é utilizado para que os contornos obtidos sejam comparados aos contornos originais. O algoritmo traça os segmentos em uma imagem auxiliar tendo como base as extremidades dos segmentos obtidos pelas duas técnicas de aproximação por segmentos de reta descritas nesta monografia. O traçado feito corresponde ao esqueleto do segmento e não a uma banda de largura w igual a "1" (um). Essa imagem obtida será comparada com a imagem original para se obter os dados estatísticos. O desenho dos segmentos de reta na imagem auxiliar é realizado de forma diferente para os dados obtidos pelo "Vector Tracer" e pela transformada de Hough. Abaixo descreve-se os dois casos.

Reconstrução dos contornos obtidos pela transformada de Hough

Para a reconstrução dos contornos obtidos pela transformada de Hough, gera-se os segmentos extraídos da estrutura de lista encadeada através do procedimento descrito na seção 2.1.8, utilizando as extremidades do segmento, além da inclinação θ e da distância ρ .

Reconstrução dos contornos obtidos pelo "Vector Tracer"

Para a reconstrução dos contornos obtidos pelo "Vector Tracer" gera-se os segmentos extraídos da estrutura de lista encadeada através do algoritmo de BRESENHAM para geração de segmentos (BRESENHAM, 1965), cujo código é baseado na implementação de VAN AKEN e KILLEBREW (1988).

3.2.3 Módulo para análise estatística dos resultados

A partir dos resultados obtidos pelos algoritmos empregados, extrai-se dados estatísticos para a caracterização dos mesmos, quanto a precisão do resultado e ao tempo de processamento. Os parâmetros levados em conta nesta monografia foram : (1) a quantidade de segmentos detectados (em unidades), (2) o tempo de processamento (em milisegundos) (3) a quantidade de “pixels” na imagem original, (4) a quantidade de “pixels” detectados corretamente (em porcentagem) e (5) a quantidade de “pixels” detectados em locais onde não existiam na imagem original (em porcentagem).

Quantidade de segmentos de reta detectadas (*sd*)

Esse parâmetro é dado em unidades e significa com quantos segmentos consegue-se representar a imagem de entrada. Quanto menor o seu número, melhor o algoritmo, desde que a fidelidade seja considerada. Essa fidelidade é expressa pelo parâmetro *pcd* descrito a seguir.

Tempo de processamento (*tp*)

O tempo de processamento considerado para os algoritmos exclui a leitura ou geração da imagem, o cálculo do histograma polar de orientação e a exibição das imagens.

Quantidade de “pixels” contidos na imagem original (*ptot*)

A quantidade total de “pixels” pretos (“1”) na imagem original. Corresponde ao número de pontos gerados para um determinado segmento de reta ou círculo, através de técnicas de computação gráfica.

Quantidade de “pixels” corretamente detectadas (*pcd*)

A quantidade de “pixels” corretamente detectadas em termos de porcentagem, os “pixels” que existem na imagem original e que foram detectadas na mesma posição na imagem obtida. Esse dado é expresso em porcentagem.

Quantidade de “pixels” erroneamente detectadas (*ped*)

A quantidade de “pixels” erroneamente detectada é expressa em termos de porcentagem e representa os “pixels” que foram encontrados na imagem obtida e que não existiam naquela posição na imagem original. Esse dado é expresso em porcentagem.

Dos parâmetros citados anteriormente foram extraídas :

- as médias
- o máximo
- o mínimo
- o desvio padrão
- correlação

A média (BARLOW, 1989) foi calculada pela expressão (37) a seguir :

$$\bar{x} = \frac{1}{N} \sum_{i=1}^N x_i \quad (37)$$

O desvio padrão (BARLOW, 1989) pela expressão (38) a seguir:

$$\sigma = \sqrt{\frac{1}{N-1} \sum (x_i - \bar{x})^2} \quad (38)$$

O coeficiente de correlação expressa a relação que uma variável tem com outra (BARLOW, 1989), e é um número entre -1 e +1. Se esse coeficiente for zero as duas variáveis são não correlacionadas. Se tender a -1 então existe uma forte correlação negativa. Isto é, se uma variável cresce, a outra decresce, e vice-versa. E se o coeficiente tender a 1 existe uma forte relação entre as duas variáveis, isto é se uma cresce, a outra também cresce. A correlação é definida a seguir.

Dado N observações de k variáveis aleatórias

$$V_1 = \{v_{11}, v_{12}, \dots, v_{1N}\}$$

$$V_2 = \{v_{21}, v_{22}, \dots, v_{2N}\}$$

...

$$V_k = \{v_{k1}, v_{k2}, \dots, v_{kN}\}$$

O coeficiente de correlação é calculado usando-se as expressões a seguir.

$$\text{cov}(V_1, V_2) = \frac{1}{N} \sum (v_{1i} - \overline{V_1})(v_{2i} - \overline{V_2}) \tag{39}$$

$$\text{correlação}(V_1, V_2) = \frac{\text{cov}(V_1, V_2)}{\sigma_{V_1} \cdot \sigma_{V_2}} \tag{40}$$

Quando se tem diversas variáveis geralmente monta-se a matriz de correlação que possui a seguinte forma:

Tabela 3.3 - Matriz de correlação entre k variáveis aleatórias.

	Variável 1	variável 2	...	Variável K
Variável 1	1	correlação(V1,V2)	...	correlação(V1,Vk)
Variável 2	correlação(V2,V1)	1	...	correlação(V2,Vk)
...	...	...	...	...
Variável k	correlação(Vk,V1)	correlação(Vk,V2)	...	1

Nessa tabela, a matriz diagonal é sempre “1”, e essa matriz é simétrica. O valor zero em alguma posição da matriz significa que a correlação é indeterminada, e que essas duas variáveis não devem ser correlacionadas.

Descreve-se na próxima seção o algoritmo utilizado para a geração do histograma polar de orientações.

3.3 Algoritmos para aplicações

As aproximações por segmentos de reta obtidos pelas técnicas “Vector Tracer” e pela transformada de Hough são utilizadas para gerar os histogramas de orientação com o intuito de aplicá-los na caracterização e análise de imagens. Gera-se o histograma de orientações no intervalo $[0^\circ, 180^\circ]$ (subdividido em subintervalos de largura igual à um), utilizando-se um vetor para acumular a soma dos comprimentos dos segmentos de reta detectados. Para cada segmento de reta detectado, de comprimento maior que um mínimo especificado, adiciona-se o seu comprimento na posição correspondente do vetor. O histograma assim obtido possui uma variação muito grande de ângulo para ângulo, o que dificulta a sua visualização, sendo portanto necessária uma suavização. Suaviza-se o histograma dividindo-se o intervalo em faixas de tamanho apropriado, denominadas de resolução angular (ra) do histograma (restrito à múltiplos de 2), e calcula-se a média aritmética dos valores dentro dessas faixas e atribuindo-o a faixa toda. A primeira e a última faixas possuem a metade da largura das outras faixas ($ra/2$) pois as inclinações próximas de 0° e de 180° se complementam quando se representa graficamente o histograma polar de orientações (ver figura 2.7). O cálculo do comprimento dos segmentos para a geração desses histogramas de orientação é feito de maneira um pouco diferente para as duas técnicas. Descreve-se a seguir o processamento requerido para os segmentos extraídos pela transformada de Hough e pelo “Vector Tracer”.

3.3.1 Histograma gerado através da transformada de Hough

O cálculo do comprimento dos segmentos aproximados obtidos pela transformada de Hough é feito considerando-se o ângulo de inclinação e o tamanho (dado pela distância Euclidiana) obtidos no momento da análise de conectividade. Isto é, no instante em que os segmentos e as respectivas extremidades são confirmadas na imagem, o seu comprimento é calculado e armazenado na matriz de histograma. Como descrito na seção 2.2.3.1, o comprimento dos segmento é obtido empregando-se as expressões (22) e (23). Apresenta-se a seguir o algoritmo do histograma polar.

Algoritmo do histograma polar

- (1) Calcule a média do histograma dividido em faixas (ra).
- (2) Normalize a matriz de histograma para representar graficamente.
- (3) Mostrar graficamente a matriz.

3.3.2 Histograma gerado através do “VectorTracer”

O comprimento dos segmentos aproximados obtidos pelo algoritmo “VectorTracer” é a distância Euclidiana das extremidades dada pela expressão apresentada na seção 2.1.5. O ângulo de inclinação do segmento de reta é dado pela expressão (40) a seguir.

$$\theta = \arctg\left(\frac{x_1 - x_0}{y_1 - y_0}\right) \quad (41)$$

onde (x_0, y_0) e (x_1, y_1) são as extremidades do segmento.

O histograma é calculado a partir dos dados armazenados na lista de segmentos, descrita a seguir, juntamente com as outras estruturas utilizadas

para essa monografia. Apresenta-se a seguir o algoritmo para o cálculo do histograma polar para os segmentos extraídos pelo “Vector Tracer”.

Algoritmo do histograma polar

- (1) Gere o histograma calculando-se o comprimento dos segmentos através da expressão (41) e armazene na matriz de histograma.
- (2) Calcule a média do histograma dividido em faixas (ra).
- (3) Normalize a matriz de histograma para representar graficamente.
- (4) Mostrar graficamente a matriz.

3.4 Informações adicionais

Os programas foram implementados utilizando a linguagem C (KERNIGHAN, RITCHIE, 1986). Os programas podem ser utilizados em ambiente MS-Windows e no ambiente Unix. O sistema utilizado para o desenvolvimento e para a geração dos dados foi um PC-PENTIUM-133 utilizando sistema operacional Windows95 e Slackware-Linux3.0. No ambiente Windows95 foi utilizado o compilador Borland C/C++ (EZZEL, 1992), (WEISKAMP, FLAMIG, 1992) (PETZOLD, 1993), (QUE, 1994), (WALNUM, 1994), (YAO, 1994), (FAISON, 1995). No ambiente Unix utilizou-se o compilador GNU g++ e a biblioteca gráfica do XWindows (X11R6). Os programas também foram testados em estações de trabalho DEC ALPHA 3000/800 com sistema operacional OSF/1-AXP, e IBM/590 com sistema operacional AIX-4.2.

4. RESULTADOS

Descreve-se nesta seção os dados considerados, a metodologia empregada e os resultados obtidos para essa avaliação de desempenho.

4.1 Parâmetros considerados na avaliação de desempenho

Os dados considerados se dividem em duas categorias : imagens sintéticas geradas por técnicas de computação gráfica e imagens reais obtidas com o uso de uma câmera CCD acoplada a um microscópio ótico por transmissão.

4.1.1 Imagens sintéticas

As imagens geradas por técnicas de computação gráfica são constituídas por segmentos de retas gerados utilizando-se o gerador de retas descrito na seção 3.2.1. Foram também produzidos círculos com raio entre 10 e 60 utilizando o algoritmo de Bresenham para círculos apresentada por FOLEY *et al.* (1992). Abaixo, na figura 4.1, representamos um segmento e um círculo gerados pelos métodos adotados.

Figura 4.1 - Segmento de reta de inclinação $\theta = 1.9266$ e $\rho = 40.7765$ gerado com 119 “pixels” (a) e círculo de raio=30 “pixels” gerado pelo algoritmo de Bresenham (b).

A próxima seção descreve as imagens reais utilizadas na avaliação desenvolvida nessa monografia.

4.1.2 Imagens reais

As imagens reais utilizadas para essa avaliação foram obtidas com uma câmera CCD (SONY SSC-C374 colorida com resolução de 500x700) acoplada à um microscópio ótico por transmissão (CARL ZEISS modelo AXIOLAB). As imagens obtidas são provenientes de tecidos de rato respectivos a três regiões, conforme descrito na tabela 4.1 a seguir.

Tabela 4.1 - Tipos de células, o processo químico e fator de aumento usado no microscópio.

TECIDO	PROCESSO QUÍMICO	AUMENTO
Células ganglionares da retina	Peroxidase	400
Hipocampo	NISSL	400
Músculo	N-atpase	100

As imagens obtidas tiveram as bordas dos neurônios enfatizadas manualmente, com o auxílio de um software de processamento de imagens (PHOTOPAINT). Após a detecção de bordas as imagens foram convertidas para binária e delas foram extraídas regiões de tamanho 128x128 que eram mais interessantes. As imagens obtidas são mostradas na figura 4.2 a seguir.

Figura 4.2 - Células ganglionares (a) e (b), células do hipocampo (c) e (d), e células de músculo (e) e (f).

Para se avaliar o comportamento dos algoritmos frente a imagens ruidosas, introduziu-se ruído uniforme (tipo canal binário simétrico) nas imagens anteriores. Deve-se observar que essa é uma avaliação preliminar dos algoritmos em imagens ruidosas, de cunho ilustrativo e informal. Essas figuras com ruído são mostradas na figura 4.3 a seguir.

Figura 4.3 - Imagens da figura 4.2 onde foi introduzido ruído.

Introduzimos na imagem original, um ruído composto de “pixels” pretos ou brancos, sendo que o “pixel” introduzido é preto quando naquela coordenada da imagem existir um “pixel” branco e vice-versa. As coordenadas dos “pixels” a serem introduzidos são escolhidas de forma aleatória

(uniformemente) utilizando-se o gerador de números randômicos descrito na seção 3.2. Nas imagens apresentadas o “ruído” cobre 2% da área total da imagem.

4.2 Avaliação de desempenho

A avaliação de desempenho foi dividida em duas partes : na primeira analisamos os resultados dos algoritmos em imagens contendo segmentos de reta gerados de maneira randômica, enquanto na segunda parte apresentamos os resultados dos algoritmos na reconstrução e geração de histogramas polares de orientação em imagens contendo círculos de diversos raios e as imagens reais de neurônios apresentadas na seção 4.1.2. Tendo como finalidade a medição do desempenho desses algoritmos, consideramos as seguintes características, mostradas na tabela abaixo:

Tabela 4.2 - Descrição das características avaliadas para a medida do desempenho.

Parâmetro	Descrição
<i>sd</i>	quantidade de segmentos detectados
<i>tp</i>	tempo de processamento
<i>ptot</i>	quantidade de “pixels” pretos existentes na imagem original
<i>pcd</i>	quantidade de “pixels” detectados corretamente
<i>ped</i>	quantidade de “pixels” erroneamente detectados

A seguir descreve-se os resultados da comparação dos algoritmos na detecção de retas.

4.2.1 Detecção de retas em imagens

As imagens sintéticas geradas para a presente avaliação são constituídas por uma seqüência de 1000 imagens contendo apenas um segmentos de reta, gerados com parâmetros aleatórios, dentro de uma imagem de 128x128 “pixels”, conforme descrito na seção 3.2.1. Nessa avaliação foram extraídos dados estatísticos como a média, o máximo, o

mínimo e o desvio padrão para as variáveis mencionadas na seção 3.2.3. Essa análise estatística inclui ainda a geração das matrizes de correlação das variáveis descritas, visando uma melhor caracterização do desempenho dos algoritmos na detecção das retas. Descrevemos a seguir os principais resultados obtidos para essa avaliação.

4.2.2 Resultados da detecção de retas

Entre os vários gráficos gerados para visualizar a descrição quantitativa do desempenho há um que mostra a distribuição da quantidade de segmentos detectados de acordo com a inclinação da reta gerada. Obtivemos também um outro gráfico que mostra a quantidade de pontos detectados em função do ângulo de inclinação da reta gerada. Esses gráficos estão representados nas figuras 4.4.a e 4.4.b, respectivamente. Na figura 4.4.b, o termo “fidelidade” está relacionado à quantidade de “pixels” (*pcd*) corretamente detectados. Isto é, quanto maior o *pcd*, mais fiel é o algoritmo em reconstruir a imagem original. De acordo com o gráfico da figura 4.4.b, o “Vector Tracer” (incluindo as duas versões) consegue detectar de forma bastante fiel os segmentos gerados sem uma quebra excessiva dos mesmos. Os dados experimentais mostram que quanto menor o parâmetro *vlen* (abreviado como *v* na figura), maior a quebra dos segmentos detectados pelo “Vector Tracer”. Os gráficos das figuras 4.4.a e 4.4.b mostram que os dois algoritmos produzem resultados semelhantes para os parâmetros utilizados, tanto em relação à quantidade de segmentos detectados como em relação à fidelidade à imagem original. Vale lembrar que isto é válido para o caso específico das imagens e parâmetros utilizados.

(a)

(b)

Figura 4.4 - Quantidade de segmentos de reta detectados (a) e fidelidade da reconstrução (b), em função da inclinação do segmento de reta gerado (gráfico normalizado para as 1000 imagens geradas).

Os picos no gráfico da figura 4.4.a, no intervalo de 90° a 135° , não é propriamente um problema do “Vector Tracer”, mas sim decorrente do processo de perseguição de contornos. Isso ocorre em imagens como a da figura 4.5 a seguir, onde se tem no segmento superior (II), o ponto p_1 , que é o primeiro “pixel” encontrado pelo algoritmo de perseguição de contornos, provocando a quebra do segmento obtido. Porém, essa deficiência pode ser contornada através de um pós-processamento conforme sugerido na seção 5.2.

Figura 4.5 - Segmentos onde há quebra ($l1$) e onde não há quebra ($l2$) na perseguição de contornos.

No caso da transformada de Hough, observou-se o aparecimento de regiões onde há uma quantidade maior de quebra dos segmentos, conforme pode ser visto na figura 4.6.a. Esse fenômeno ocorre de forma mais acentuada no caso da transformada de Hough de resolução simples, o que indica que para esse caso, essa quebra se deve ao fato da resolução ser insuficiente. Já o aparecimento de pequenos picos para o caso da transformada de Hough de resolução dupla, conforme a figura abaixo, nas regiões próximas à múltiplos de 45° , leva a crer que o fato é provocado também pelo processo de quantização espacial das retas geradas. Os gráficos da figura 4.6.a e 4.6.b mostram que a transformada de Hough com acumulador de resolução dupla obteve um desempenho melhor na detecção de retas, pois produz uma quantidade menor de segmentos quebrados aliados à uma fidelidade maior em relação à imagem original.

(a) (fig. 4.6)

Figura 4.6 - Quantidade de segmentos detectados (a) e fidelidade da reconstrução das transformadas de Hough de resolução simples e dupla (b) em função da inclinação da reta (gráfico normalizado para as 1000 imagens geradas).

Discutimos a seguir as tabelas contendo características quantitativas extraídas pela análise estatística, como a média, valor máximo, o mínimo e o desvio padrão, para avaliar o desempenho dos algoritmos. Essas tabelas utilizam a notação abreviada introduzida nas seções 3.1.1.2 e 3.1.2.5, respectivamente, para identificar os resultados dos algoritmos “Vector Tracer” e transformada de Hough. Através da análise dessas tabelas, foram obtidas várias informações importantes sobre o desempenho dos algoritmos na detecção de retas em imagens. As tabelas 4.3.a a (c) mostram os resultados obtidos pelas técnicas “Vector Tracer 1” e “Vector Tracer 2” na detecção de retas. De acordo com essas tabelas conclui-se, por exemplo, que o aumento do comprimento dos vetores auxiliares (vetores a , b e c) usados para o cálculo da curvatura do contorno pelo Vector1 e pelo Vector2, leva a uma diminuição na quebra dos segmentos detectados. No caso do Vector 1, usando-se $vlen$ igual à 2, a média para sd foi igual à 16, para $vlen$ igual à 3, sd foi igual à 9.46 e para $vlen$ igual à 5, sd foi igual à 1.14. No caso do Vector 2, usando-se $vlen$ igual à 2, a média para sd foi igual à 1.57, para $vlen$ igual à 3, sd foi igual à 1.33 e para $vlen$ igual à 5, sd foi igual à 1.14. Um outro ponto observado nessas tabelas é que, pelo menos para os casos considerados nesta monografia, o aumento de $vlen$ leva a um aumento na fidelidade do algoritmo de “Vector Tracer”. Em relação ao tempo de processamento das imagens contendo

segmentos de reta não houve uma variação muito acentuada para o “Vector Tracer”.

Tabela 4.3 - Resultados estatísticos obtidos para VT {2, 0.4, 0.98, 0.995} em (a), VT {3, 0.4, 0.98, 0.995} em (b) e para VT {5, 0.4, 0.98, 0.995} em (c).

	Vector Tracer 1				Vector Tracer 2			
	média	máximo	mínimo	desvio	média	máximo	mínimo	desvio
<i>sd</i>	16	33	1	8.23	1.57	4	1	0.69
<i>tp</i>	7.42	9.8	6.8	0.32	7.46	9.2	6.6	0.29
<i>pcd</i>	64.67	100	50.78	9.36	78.68	100	50.78	10.29
<i>ped</i>	35.33	49.22	0	9.36	21.32	49.22	0	10.29

(a)

	Vector Tracer 1				Vector Tracer 2			
	média	máximo	mínimo	desvio	média	máximo	mínimo	desvio
<i>sd</i>	9.46	22	1	6.81	1.33	3	1	0.57
<i>tp</i>	7.38	8	6.8	0.3	7.4	8	6.8	0.29
<i>pcd</i>	70.55	100	50.78	12.13	80.15	100	52	9.85
<i>ped</i>	29.45	49.22	0	12.13	19.85	48	0	9.85

(b)

	Vector Tracer 1				Vector Tracer 2			
	média	máximo	mínimo	desvio	média	máximo	mínimo	desvio
<i>sd</i>	1.14	2	1	0.35	1.14	2	1	0.35
<i>tp</i>	7.7	8.8	6.8	0.32	7.7	8.2	6.8	0.32
<i>pcd</i>	80.77	100	52	9.53	80.77	100	52	9.53
<i>ped</i>	19.23	48	0	9.53	19.23	48	0	9.53

(c)

Nessa comparação também ficou clara a superioridade do algoritmo Vector2, pois o Vector1 apenas igualou o desempenho com *vlen*, nesse caso igual à 5. A utilização de valores de *vlen* menores que 5 é justificada principalmente em imagens diferentes de segmentos de reta, e onde há objetos contendo pequenos detalhes.

Os resultados estatísticos para o processamento das imagens através da Hough1 e Hough2 são apresentadas nas tabelas 4.4.a a (c). Pela análise dessas tabelas obtivemos subsídios para várias conclusões à respeito do desempenho dos dois algoritmos. No caso da Hough1, por exemplo, o aumento da largura da banda reta na análise de conectividade, dada por *w*, de 1.0 para 2.0, levou a uma diminuição da quebra dos segmentos de 7.11 para 1.55, e no caso da Hough2 esse valor caiu de 3.96 para 1.01. Em

contrapartida, esse aumento da largura w reduziu a fidelidade do algoritmo Hough1 de 92.46 para 69.9 e do Hough2 de 92.83 para 84.91. Um outro ponto a se observar é que esse aumento em w levou a um pequeno aumento no tempo de processamento. O valor mínimo de pcd igual à zero na tabela 4.4.b e (c) para a Hough1 deve-se ao fato da detecção de retas paralelas.

Tabela 4.4 - Resultados estatísticos obtidos para HT {3,2,3,2,1.0} em (a), HT {3,2,3,2,1.5} em (b) e para HT {3,2,3,2,2.0} em (c).

	Transformada de Hough 1				Transformada de Hough2			
	média	máximo	mínimo	desvio	média	máximo	mínimo	desvio
<i>sd</i>	7.11	40	1	5.24	3.96	15	1	3.05
<i>tp</i>	101.65	130	40	21.17	206.89	260	90	40.85
<i>pcd</i>	92.46	100	65.62	6.64	92.83	100	70.59	6.32
<i>ped</i>	25.11	104.69	0	20.54	14.59	56.41	0	10.44

(a)

	Transformada de Hough 1				Transformada de Hough 2			
	média	máximo	mínimo	desvio	média	máximo	mínimo	desvio
<i>sd</i>	3.25	25	1	3.06	1.22	6	1	0.63
<i>tp</i>	118.32	160	50	25.54	227.15	300	100	45.81
<i>pcd</i>	76.65	100	0	9.92	85.13	100	66.34	7.01
<i>ped</i>	30.88	101.56	0	16.49	15.08	36.36	0	7.29

(b)

	Transformada de Hough 1				Transformada de Hough 2			
	média	máximo	mínimo	desvio	média	máximo	mínimo	desvio
<i>sd</i>	1.55	11	1	1.21	1.01	3	1	0.13
<i>tp</i>	135.81	160	50	29.38	248.24	300	100	50.81
<i>pcd</i>	69.9	100	0	13.29	84.91	100	66.41	7.15
<i>ped</i>	31.35	100	0	14.54	14.91	33.59	0	7.09

(c)

Pela comparação entre Hough1 e Hough2, verificou-se que o tempo de processamento da Hough2 é praticamente o dobro da Hough1, devido ao dobro de resolução usado no caso da Hough2. Outro dado interessante é que a Hough2 quebra menos os segmentos quando comparado à Hough1, sendo também a fidelidade maior. Além dessas medidas, foram também geradas as matrizes de correlação respectivas aos parâmetros de desempenho considerados, sendo apresentados nas tabelas B.8 a B.13 para o “Vector Tracer” e nas tabelas B.14 a B.19 para a transformada de Hough.

Pela análise das matrizes de correlação do “Vector Tracer”, verifica-se uma dependência positiva entre pcd e ped , como poderia ser esperado. Além disso, quando se observa a correlação dessas duas variáveis em relação às outras, verifica-se que as correlações com pcd e ped são complementares. Isso é devido ao fato de que na perseguição de “pixels” em imagens com retas, todos os “pixels” são perseguidos. Um outro fato que justifica esse resultado é que na reconstrução, pcd é dado por ped subtraído de $ptot$, ou seja ped é complemento de pcd em relação à $ptot$. Outra dependência encontrada nessas matrizes de correlação do “Vector Tracer” foi a correlação entre tp e $ptot$. Isto pode ser comprovado observando-se a variação do tempo de processamento do “Vector Tracer” em relação aos vários tipos de imagens, conforme apresentada nas tabelas ao final desta seção.

As matrizes de correlação para as transformadas de Hough mostram que existe uma correlação linear positiva entre o tempo de processamento e a quantidade de “pixels” pretos na imagem ($ptot$), comprovada pelas tabelas de resultados apresentada ao final desta seção. A relação complementar existente entre pcd e ped , ao contrário do que acontece no “Vector Tracer” não se verifica no caso da transformada de Hough, como no caso do “Vector Tracer”. Isso ocorre porque no caso da Hough1 (grupos 1 e 2) e na Hough2 (no grupo 1) ocorre a detecção de pontos espúrios provocada pela utilização de um valor de “gap” maior que zero. Essa situação encontra-se representada na figura abaixo, onde $p1$ e $p2$ foram detectados e $p3$ e $p4$ foram perdidos.

Figura 4.7 - Segmento original (a) e segmento detectado (b) com “pixels” $p1$ a $p4$ erroneamente detectados ($p1$ e $p2$ não existiam e $p3$ e $p4$ foram perdidos).

4.3 Reconstrução de imagens e histograma de orientações

Para a reconstrução dos círculos e das imagens reais apresentadas na seção 4.1.1 e 4.1.2, utilizamos as configurações dos parâmetros conforme a tabela 4.5 para o “Vector Tracer” e para a transformada de Hough. Deve-se lembrar que o Vector1 não utiliza o parâmetro $vth3$. A utilização de diferentes parâmetros tem como objetivo avaliar a sensibilidade da resposta dos algoritmos em relação à escolha dos parâmetros. Os parâmetros foram escolhidos de forma a se obter resultados semelhantes em termos da quantidade de segmentos detectados e o histograma polar, para o “Vector Tracer” e a transformada de Hough e estão divididos em três grupos conforme a tabela abaixo.

Tabela 4.5 - Grupos de parâmetros para o “Vector Tracer” e transformada de Hough, para a reconstrução e geração de histogramas.

Técnica	Vector1/Vector2				Hough1/Hough2				
Parâmetro	vlen	vth1	vth2	vth3	hth1	hth2	n	gap	w
Grupo 1	3	0.4	0.95	0.99	3	2	3	2	1.0
Grupo 2	3	0.4	0.9	0.99	3	2	3	2	1.5
Grupo 3	3	0.4	0.85	0.99	3	2	3	2	2.0

Os algoritmos ajustados com os parâmetros do segundo grupo fazem, em geral, a reconstrução dos contornos com uma quantidade menor de segmentos se comparada aos parâmetros do primeiro grupo, e uma quantidade de segmentos maior que os do terceiro grupo. Como descrito na seção 3.2.2 a reconstrução dos segmentos é feita em termos do esqueleto (BRESENHAM, 1965) e não da banda de largura w igual à 1.

Em alguns casos pode ser preferível a reconstrução dos contornos em termos de bandas. Nesses casos perde-se em termos de resolução da largura, mas não em relação a resolução da inclinação da retas reconstruídas.

Ainda em relação aos parâmetros, nos casos da reconstrução em imagens ruidosas, atribui-se o valor zero para o “gap” como um meio de evitar a detecção de retas indesejáveis, pelas transformadas de Hough1

(HT1 {3, 2, 3, 2, 1.0 }) e Hough2 (HT2 {3, 2, 3, 2, 1.0}), mostradas respectivamente nas figuras 4.8.c e (f) a seguir.

Figura 4.8 - Exemplo de utilização de “gap” diferente de zero para imagens ruidosas.

Nessa figura, o imagem original (a) é afinada (d) e os resultados da reconstrução da imagem usando o “Vector Tracer 1” (VT1 {3, 0.4, 0.99 }) e o “Vector Tracer 2” (VT2 {3, 0.4, 0.99, 0.995 }) pode ser verificado nas figuras 4.8.b e (e) respectivamente. No caso da comparação de desempenho em imagens ruidosas, a imagem utilizada para se comparar os “pixels” que foram perdidos e o “pixels” que foram acrescentados às imagens, é a imagem original sem ruído introduzido e afinada.

Os resultados dessa avaliação na reconstrução de imagens e na geração de histogramas são apresentados a seguir.

4.3.1 Resultados da reconstrução e determinação dos histogramas de orientações das imagens

As imagens processadas para a obtenção das reconstruções em termos de segmentos de reta e do histograma polar de orientações incluem imagens de círculos de raio igual a 10, 30 e 60, e imagens reais obtidas com o uso de uma câmera CCD. Foram também avaliadas versões das imagens reais rotacionadas de 90° no sentido horário e imagens nas quais introduziu-se ruído uniforme conforme descrito anteriormente na seção 4.1.2. Os resultados do processamento das imagens são mostrados através de figuras contendo as imagens processadas, a reconstrução em termos de segmentos de reta e o histograma polar de orientações correspondente. Para uma correta visualização dos resultados apresentados, veja a descrição gráfica da estrutura das figuras, apresentada na figura B.1 (Anexo B). As figuras seguem a seguinte estrutura : são formados de 3 grupos, sendo o primeiro de (a) a (j), o segundo de (k) a (t) e o terceiro de (u) a (D). Cada grupo de imagens é processada conforme os valores apresentados na tabela 4.5.

As figuras B.2 a B.31 (Anexo B) mostram os resultados obtidos para a reconstrução das imagens e a geração do histograma polar, utilizando os algoritmos “Vector Tracer” e transformada de Hough. Deve-se lembrar que nas imagens dos círculos sem ruído apresentadas nas figuras B.2, B.4 e B.6, a imagem afinada mostrada é a própria imagem original. No Anexo B, após as figuras, apresenta-se as tabelas contendo os dados estatísticos dos resultados da aplicação dos algoritmos em imagens reais. Essas tabelas serão referenciadas ao longo da discussão dos resultados obtidos na reconstrução e no histograma polar de orientações.

As figuras contendo os resultados das reconstruções e os histogramas polares de orientações são apresentadas no Anexo B, juntamente com a avaliação estatística nas imagens reais.

Pela observação dos resultados das reconstruções, verificou-se que o “Vector Tracer” produz uma reconstrução contínua enquanto que a transformada de Hough produz reconstrução tipicamente descontínua. É Por conta dessa descontinuidade, na comparação visual, tem-se a impressão de

que o “Vector Tracer” é “melhor”, porém quando se analisa os dados estatísticos, verifica-se que a fidelidade da transformada de Hough é maior. A única exceção encontrada foi para o caso das reconstruções de círculos de raio igual à 60, onde a transformada de Hough obteve resultado tanto visual como quantitativo superior. Isso é devido a que o “Vector Tracer” gera uma aproximação mais grosseira no caso específico de círculos grandes, ou seja com ângulos bastante suaves, com os parâmetros utilizados. Para minimizar esse efeito basta aumentar os valores de $vth2$ e $vth3$. Em geral, nas imagens processadas, a transformada de Hough produz uma aproximação com uma quantidade de segmentos maior do que o “Vector Tracer”, acompanhada de uma melhor fidelidade.

Baseado nos resultados dos histogramas de círculos, verificou-se que o “Vector Tracer” produz um histograma com uma melhor distribuição de orientações. No caso da transformada de Hough, encontramos resultados relativamente incompatíveis com o histograma de um objeto circular nas figuras B.2 (o), (t), (y) e (D) e figura B.4.(y). Este tipo de distorção é causada em parte devido à remoção dos segmentos que são detectados em cada passo, somados ao fato de um círculo ser um caso particularmente crítico de contorno discretizado.

É interessante observar que a introdução de ruído não afeta de forma acentuada nem a quantidade de segmentos detectadas e nem o histograma polar obtido. O que ocorre em geral, com a quebra dos segmentos que o ruído provoca, é uma redistribuição dos picos nas inclinações próximas da que havia na imagem original, o que contribui para um aumento dos picos do histograma que não eram significativos. A única técnica que mostrou um comportamento diferente das demais foi a Hough2 na geração de histogramas polares em imagens ruidosas utilizando o “gap” igual à zero (figuras B.11.j, B.13.j, B.15.j, B.17.j, B.21.j, B.29.j, B.31.j). Esse efeito ocorre porque o parâmetros do grupo 1, utilizados para a obtenção do histograma polar com a Hough2 em imagens ruidosas, é inadequado. A Hough2 é mais sensível aos detalhes (maior resolução) quando o parâmetro w é igual à 1. Uma das formas de minimizar esse problema é atribuindo valores maiores que 1 para o “gap”. O resultados dessa mudanças podem ser visualizados na figura abaixo.

Figura 4.9 - Imagem original ruidosa (a), afinada (d), a reconstrução da Hough1 (b), a reconstrução da Hough2 (e), o histograma da Hough1 (c) e o histograma da Hough2 (f).

Nessa solução podemos verificar o aparecimento de retas espúrias, conforme comentado anteriormente nessa seção. Uma solução melhor é a utilização de valores maiores para a largura da banda reta na imagem, ou seja o parâmetro w ajustado para valores acima de 1, como os parâmetros dos grupos 2 e 3, como pode ser observado nas figuras apresentadas mais adiante nessa seção.

No geral, tanto o “Vector Tracer” como a transformada de Hough produziram resultados muito semelhantes e mostraram boa insensibilidade geral à variação dos parâmetros considerados, o que favorece a robustez ao ajuste dos parâmetros. Um item importante na diferenciação entre os algoritmos “Vector Tracer” e transformada de Hough é o tempo de processamento, pois enquanto no Vector1 e no Vector2 o tempo de processamento não varia de forma acentuada em função da quantidade de “pixels” na imagem, na transformada de Hough (Hough1 e Hough2) nota-se um aumento de razão aproximadamente linear nesse índice. No “Vector Tracer” o tempo de processamento considerando-se todas as imagens processadas ficou na faixa de 10 milissegundos, tanto nas imagens originais como nas imagens ruidosas. No caso da Hough1, o tempo de processamento nas imagens originais ficou na casa de 550 milissegundos e em imagens ruidosas o tempo subiu para a casa dos 938 milissegundos. Já o tempo da Hough2 ficou

na casa dos 1.15 segundos e 1.98 segundos, respectivamente para imagens originais e ruidosas. O fato do tempo de processamento variar tanto no caso da Hough1 e Hough2 e permanecer quase que inalterado se deve a que no “Vector Tracer”, os “pixels” que se encontram “sozinhos” na imagem são descartados na fase de perseguição de contornos. Isto não ocorre no caso da transformada de Hough, pois como ela mapeia globalmente todos os “pixels” pretos existentes na imagem, o tempo de processamento será proporcional ao aumento do número de “pixels” pretos provocado pela introdução de ruído. Esses valores de tempo de processamento podem ser encontrados nas tabelas B.1 a B.5 , apresentadas no Anexo B, após as figuras.

Apresenta-se a seguir as conclusões e as sugestões para desenvolvimentos futuros

5. CONCLUSÕES E SUGESTÕES PARA DESENVOLVIMENTOS FUTUROS

Esta monografia teve como objetivo principal a comparação de duas técnicas de aproximação de contornos em imagens digitais, mais precisamente o “Vector Tracer” e a transformada de Hough, através de segmentos de reta. Apresentamos ainda uma aplicação envolvendo as duas técnicas, que permitiu a comparação do desempenho das técnicas consideradas. As duas seções que se seguem apresentam respectivamente as conclusões e sugestões para futuros desenvolvimentos que poderão dar continuidade aos desenvolvimentos abordados nesta monografia.

5.1 Conclusões

Uma revisão dos conceitos básicos relativos ao processamento digital de imagens foi abordada no Capítulo 2, onde se apresentou uma revisão dos principais conceitos utilizados no decorrer do desenvolvimento da monografia, tais como a definição de segmentos de reta e a sua geração, além da descrição dos principais métodos de aproximação de contornos através de segmentos de reta, incluindo os dois métodos de detecção de retas avaliados nesta monografia. Foram também discutidos conceitos relativos a discretização de imagens, incluindo esquemas de quantização e relações básicas entre os “pixels” na imagem (vizinhança, conectividade e caminho). Foram ainda discutidos tópicos como a definição de distâncias, codificação de contornos e algoritmos para processamento de imagens binárias.

No Capítulo 3 foram descritos os detalhes de implementação dos algoritmos utilizados nesta monografia. Tal descrição inclui o algoritmo “Vector Tracer” e a sua versão modificada, a transformada de Hough usando o acumulador com resolução simples e a transformada de Hough usando acumulador com resolução dupla. Nesse capítulo também abordamos questões relativas à análise estatística dos resultados obtidos, assim como os algoritmos usados na reconstrução das imagens originais através dos segmentos de reta detectados e na geração do histograma polar de orientações.

O Capítulo 4 traz os resultados das análises estatísticas do desempenho dos algoritmos, comentados e ilustrados através de exemplos gráficos. Inclui-se aí exemplos da reconstrução das imagens em termos de segmentos de reta e correspondentes histogramas polares de orientações. Discute-se ainda os efeitos da escolha dos parâmetros na obtenção de bons resultados, juntamente com as principais características dos algoritmos comparados.

Nas aplicações em reconstrução de contornos e na geração do histograma polar de orientações foram obtidos bons resultados com todos os algoritmos utilizados. Verificou-se que os algoritmos são pouco sensíveis, dentro de um certo intervalo, ao ajuste dos parâmetros, fato esse que reforça a robustez dos algoritmos avaliados. No caso da geração do histograma com a Hough2 em imagens ruidosas, verificou-se que os parâmetros do grupo 1 são inapropriados, sendo os do grupo 2 e 3 os mais indicados. A utilização do valor de “gap” igual à um minimiza esse problema, porém acarreta a detecção de retas espúrias pelas transformadas de Hough (Hough1 e Hough2).

O tempo de processamento é um item de diferenciação entre os algoritmos, pois enquanto no Vector1 e no Vector2 o tempo de processamento não varia de forma acentuada de acordo com a quantidade de “pixels” na imagem, na transformada de Hough (Hough1 e Hough2) nota-se um aumento de forma proporcional nesse índice.

Os resultados obtidos pelos algoritmos são relativamente semelhantes. Uma das principais diferenças é que o “Vector Tracer” produz aproximações contínuas com os vértices do polígono pertencendo a curva, enquanto a transformada de Hough gera aproximações cujas extremidades dos segmentos caracteriza-se por aproximação tipicamente descontínua. Caso haja

necessidade da obtenção de aproximações contínuas com a transformada de Hough, deve-se utilizar um algoritmo para reconexão dos segmentos, conforme comentado na seção seguinte. Tendo como subsídio os resultados obtidos, o algoritmo "Vector Tracer" é recomendado em aplicações onde a continuidade e o tempo de processamento é crítico, enquanto a transformada de Hough se presta melhor nos casos onde a fidelidade em relação à imagem original é mais crítica.

Apesar dos resultados obtidos favorecerem a técnica "Vector Tracer", deve-se lembrar que este algoritmo necessita do afinamento das imagens, o que não é uma exigência da transformada de Hough. Outra desvantagem dessa técnica quando comparada à transformada de Hough é que a mesma não gera projeções. Além disso, possuindo uma estrutura de controle e um fluxo de dados mais simples, a transformada de Hough se presta para implementações eficientes em arquiteturas sistólicas dedicadas.

Como recomendado por JAIN e BINFORD (1991) e adotado por COSTA (1992a), essa monografia se preocupou em abordar aspectos de avaliação comparativa em termos quantitativos de algoritmos para análise de imagens.

Apresenta-se a seguir uma tabela comparativa das principais características dos algoritmos "Vector Tracer" e transformada de Hough, verificada nesta avaliação de desempenho.

Tabela 5.1 - Principais características do "Vector Tracer" e transformada de Hough.

Características	Vector Tracer	Transformada de Hough
Tempo de processamento (<i>tp</i>)	Bastante reduzido se comparado à transformada de Hough.	Alto em relação ao "Vector Tracer", em alguns casos é em torno de 15 (Hough1) e 30 vezes (Hough2), logo o tempo de processamento para a Hough1 é aproximadamente a metade da Hough2.
Fidelidade	Boa.	Boa, superando a do "Vector Tracer".
Aspecto visual da reconstrução	Bastante satisfatória.	Bastante satisfatória, porém com o aparecimento de pequenas falhas na detecção dos segmentos devido à natureza mais global da técnica e devido à remoção dos segmentos extraídos.
Continuidade da aproximação	Contínua.	Predominantemente descontínua.
Histograma polar de orientações	Bons, possuindo uma distribuição dos picos mais uniforme que a transformada de Hough.	Bons. Quando se utiliza a Hough2 deve-se usar uma largura w maior que 1 e/ou "gap" maior que zero.
Deteção de retas	Bom desempenho na detecção de retas, sendo que o Vector2 produz em geral aproximações com a mesma fidelidade do Vector1, porém com uma menor quantidade de segmentos.	Possui um bom desempenho na detecção de retas. A Hough2 produz em geral aproximações com maior quantidade de segmentos e maior fidelidade que a Hough1.

A seguir apresenta-se as principais sugestões de trabalhos que poderão enriquecer os tópicos já desenvolvidos nesta monografia.

5.2 Sugestões para desenvolvimentos futuros

Considerando-se os resultados apresentados nesta monografia, e com o intuito de ampliar a pesquisa sobre esse tema, apresentamos a seguir algumas sugestões que poderão servir de base para novos trabalhos nessa área.

Como os algoritmos foram avaliados sem contudo se utilizar todas as faixas de parâmetros possíveis, recomenda-se um estudo mais detalhado do comportamento dos algoritmos ao longo de toda a faixa de parâmetros. Por exemplo, seria interessante observar os efeitos da variação do “gap” na transformada de Hough de maneira a se definir possíveis controles adaptativos desse parâmetro.

Outro ponto interessante seria a avaliação dos algoritmos com outros tipos de imagens, incluindo imagens contendo mais de um segmento de reta. Além disso, como observado na seção 4.1.2, há a necessidade da avaliação do comportamento dos algoritmos em imagens ruidosas e o estudo do efeito do ruído na detecção de segmentos de reta, a ser realizado de maneira mais formal e abrangente.

Assim como proposto por ANSARI e DELPI (1991) e WU e WANG (1993) pode ser interessante fazer experiências tendo o Vector Tracer como passo preliminar de um outro algoritmo de aproximação poligonal, por exemplo RAMER (1972) ou DUDA e HART (1973). Isto é justificado pois os algoritmos da classe “split” são sensíveis aos pontos iniciais e a escolha dos pontos iniciais poderia ser feita pelo “Vector Tracer”.

Como as extremidades dos segmentos detectados pela transformada de Hough podem ou não estar sobre a curva original, sendo portanto uma aproximação descontínua, caso haja a necessidade da obtenção de aproximações contínuas, sugere-se a utilização de um algoritmo como o descrito por WEISS e BOLDT (1986).

ANEXO A

RESULTADOS DA AVALIAÇÃO COM RETAS

Resultados do processamento das imagens contendo retas pelo "Vector Tracer", expressos pelas matrizes de correlação.

Tabela A.1 - Matrizes de correlação para VT1 (a) e VT2 (b) (VT {2, 0.4, 0.98, 0.995}).

VT1	<i>sd</i>	<i>tp</i>	<i>ptot</i>	<i>pcd</i>	<i>ped</i>
<i>sd</i>	1	-0.15	0.43	0.16	-0.16
<i>tp</i>	-0.15	1	0.71	-0.21	0.21
<i>ptot</i>	0.43	0.71	1	-0.12	0.12
<i>pcd</i>	0.16	-0.21	-0.12	1	-1
<i>ped</i>	-0.16	0.21	0.12	-1	1

(a)

VT2	<i>sd</i>	<i>tp</i>	<i>ptot</i>	<i>pcd</i>	<i>ped</i>
<i>sd</i>	1	0.09	0.12	-0.29	0.29
<i>tp</i>	0.09	1	0.78	-0.04	0.04
<i>ptot</i>	0.12	0.78	1	-0.1	0.1
<i>pcd</i>	-0.29	-0.04	-0.1	1	-1
<i>ped</i>	0.29	0.04	0.1	-1	1

(b)

Tabela A.2 - Matrizes de correlação para VT1 (a) e VT2 (b) (VT {3, 0.4, 0.98, 0.995}).

VT1	<i>sd</i>	<i>tp</i>	<i>ptot</i>	<i>pcd</i>	<i>ped</i>
<i>sd</i>	1	-0.42	0.46	-0.31	0.31
<i>tp</i>	-0.42	1	0.57	0.03	-0.03
<i>ptot</i>	0.46	0.57	1	-0.25	0.25
<i>pcd</i>	-0.31	0.03	-0.25	1	-1
<i>ped</i>	0.31	-0.03	0.25	-1	1

(a)

VT2	<i>sd</i>	<i>tp</i>	<i>ptot</i>	<i>pcd</i>	<i>ped</i>
<i>sd</i>	1	-0.01	0.19	-0.22	0.22
<i>tp</i>	-0.01	1	0.62	0	0
<i>ptot</i>	0.19	0.62	1	-0.1	0.1
<i>pcd</i>	-0.22	0	-0.1	1	-1
<i>ped</i>	0.22	0	0.1	-1	1

(b)

Tabela A.3 - Matrizes de correlação para VT1 (a) e VT2 (b) (VT {5, 0.4, 0.98, 0.995}).

VT1	<i>sd</i>	<i>tp</i>	<i>ptot</i>	<i>pcd</i>	<i>ped</i>
<i>sd</i>	1	0.1	0.12	-0.19	0.19
<i>tp</i>	0.1	1	0.95	-0.08	0.08
<i>ptot</i>	0.12	0.95	1	-0.11	0.11
<i>pcd</i>	-0.19	-0.08	-0.11	1	-1
<i>ped</i>	0.19	0.08	0.11	-1	1

(a)

VT2	<i>sd</i>	<i>tp</i>	<i>ptot</i>	<i>pcd</i>	<i>ped</i>
<i>sd</i>	1	0.11	0.12	-0.19	0.19
<i>tp</i>	0.11	1	0.96	-0.1	0.1
<i>ptot</i>	0.12	0.96	1	-0.11	0.11
<i>pcd</i>	-0.19	-0.1	-0.11	1	-1
<i>ped</i>	0.19	0.1	0.11	-1	1

(b)

Resultados do processamento das imagens contendo retas pela transformada de Hough, expressos pelas matrizes de correlação.

Tabela A.4 - Matrizes de correlação para HT1 (a) e HT2 (b) (HT {3, 2, 3, 2, 1.0}).

HT1	<i>sd</i>	<i>tp</i>	<i>ptot</i>	<i>pcd</i>	<i>ped</i>
<i>sd</i>	1	0.31	0.3	0.19	-0.07
<i>tp</i>	0.31	1	0.97	0.02	0.09
<i>ptot</i>	0.3	0.97	1	0.03	0.05
<i>pcd</i>	0.19	0.02	0.03	1	0.1
<i>ped</i>	-0.07	0.09	0.05	0.1	1

(a)

HT2	<i>sd</i>	<i>tp</i>	<i>ptot</i>	<i>pcd</i>	<i>ped</i>
<i>sd</i>	1	0.18	0.18	0.37	0.19
<i>tp</i>	0.18	1	0.99	0.04	0.06
<i>ptot</i>	0.18	0.99	1	0.04	0.04
<i>pcd</i>	0.37	0.04	0.04	1	-0.16
<i>ped</i>	0.19	0.06	0.04	-0.16	1

(b)

Tabela A.5 - Matrizes de correlação para HT1 (a) e HT2 (b) (HT {3, 2, 3, 2, 1.5}).

HT1	<i>sd</i>	<i>tp</i>	<i>ptot</i>	<i>pcd</i>	<i>ped</i>
<i>sd</i>	1	0.12	0.14	-0.26	0.38
<i>tp</i>	0.12	1	0.98	0.07	0.04
<i>ptot</i>	0.14	0.98	1	0.07	0.04
<i>pcd</i>	-0.26	0.07	0.07	1	-0.56
<i>ped</i>	0.38	0.04	0.04	-0.56	1

(a)

HT2	<i>sd</i>	<i>tp</i>	<i>ptot</i>	<i>pcd</i>	<i>ped</i>
<i>sd</i>	1	0.05	0.05	-0.17	0.29
<i>tp</i>	0.05	1	0.99	-0.04	0.07
<i>ptot</i>	0.05	0.99	1	-0.04	0.07
<i>pcd</i>	-0.17	-0.04	-0.04	1	-0.96
<i>ped</i>	0.29	0.07	0.07	-0.96	1

(b)

Tabela A.6 - Matrizes de correlação para HT1 (a) e HT2 (b) (HT {3, 2, 3, 2, 2.0}).

HT1	<i>sd</i>	<i>tp</i>	<i>ptot</i>	<i>pcd</i>	<i>ped</i>
<i>sd</i>	1	0.08	0.1	-0.3	0.4
<i>tp</i>	0.08	1	0.99	0.06	-0.01
<i>ptot</i>	0.1	0.99	1	0.05	0.01
<i>pcd</i>	-0.3	0.06	0.05	1	-0.94
<i>ped</i>	0.4	-0.01	0.01	-0.94	1

(a)

HT2	<i>sd</i>	<i>tp</i>	<i>ptot</i>	<i>pcd</i>	<i>ped</i>
<i>sd</i>	1	-0.01	0	-0.05	0.04
<i>tp</i>	-0.01	1	0.99	-0.08	0.1
<i>ptot</i>	0	0.99	1	-0.07	0.1
<i>pcd</i>	-0.05	-0.08	-0.07	1	-1
<i>ped</i>	0.04	0.1	0.1	-1	1

(b)

ANEXO B

RESULTADOS DE RECONSTRUÇÃO E HISTOGRAMA POLAR DE ORIENTAÇÕES

Apresenta-se abaixo o gabarito da representação gráfica adotada para os resultados das reconstruções, em termos de segmentos de reta e histogramas polares de orientações das imagens reais processadas pelos algoritmos

Grupo1

Imagem original (a)	Vector1 { 3, 0.4, 0.95 } (b)	Hough1 {3,2,3,2,1.0} (c)	Histograma VT1 (d)	Histograma Hough1 (e)
Imagem afinada (f)	Vector2 {3, 0.4, 0.95, 0.99} (g)	Hough2 {3,2,3,2,1.0} (h)	Histograma VT2 (i)	Histograma Hough2 (j)

Grupo2

Imagem original (k)	Vector1 { 3, 0.4, 0.9 } (l)	Hough1 {3,2,3,2,1.5} (m)	Histograma VT1 (n)	Histograma Hough1 (o)
Imagem afinada (p)	VT2 {3, 0.4, 0.9, 0.99} (q)	HT2 {3,2,3,2,1.5} (r)	Histograma VT2 (s)	Histograma Hough2 (t)

Grupo3

Imagem original (u)	VT1 { 3, 0.4, 0.85 } (v)	HT1 {3,2,3,2,2.0} (w)	Histograma VT1 (x)	Histograma Hough1 (y)
Imagem afinada (z)	VT2 {3, 0.4, 0.85, 0.99} (A)	HT2 {3,2,3,2,2.0} (B)	Histograma VT2 (C)	Histograma Hough2 (D)

Figura B.1 - Descrição da estrutura adotada para a apresentação dos resultados de reconstrução e histograma de acordo com o grupo de parâmetros utilizados.

A seguir apresenta-se os resultados dos algoritmos na reconstrução das imagens reais e do histogramas polares de orientações gerados.

Figura B.2 - Reconstrução e histograma de uma imagem contendo um círculo de raio=10 "pixels", usando o "Vector Tracer" e a transformada de Hough.

Figura B.3 - Reconstrução e histograma de uma imagem ruidosa contendo um círculo de raio=10 "pixels", usando o "Vector Tracer" e a transformada de Hough.

Figura B.4 - Reconstrução e histograma de uma imagem contendo um círculo de raio=30 "pixels", usando o "Vector Tracer" e a transformada de Hough.

Figura B.5 - Reconstrução e histograma de uma imagem ruidosa contendo um círculo de raio=30 "pixels", usando o "Vector Tracer" e a transformada de Hough.

Figura B.6 - Reconstrução e histograma de uma imagem contendo um círculo de raio=60 "pixels", usando o "Vector Tracer" e a transformada de Hough.

Figura B.7 - Reconstrução e histograma de uma imagem ruidosa contendo um círculo de raio=60 "pixels", usando o "Vector Tracer" e a transformada de Hough.

Figura B.8 - Reconstrução e histograma de uma imagem contendo uma célula ganglionar, usando o "Vector Tracer" e a transformada de Hough.

Figura B.9 - Reconstrução e histograma de uma imagem ruidosa contendo uma célula ganglionar, usando o "Vector Tracer" e a transformada de Hough.

Figura B.10 - Reconstrução e histograma de uma imagem contendo uma célula ganglionar (fig. 4.14.a rotacionada), usando o "Vector Tracer" e a transformada de Hough.

Figura B.11 - Reconstrução e histograma de uma imagem ruidosa contendo uma célula ganglionar rotacionada, usando o "Vector Tracer" e a transformada de Hough.

Figura B.12 - Reconstrução e histograma de uma imagem contendo uma célula ganglionar, usando o "Vector Tracer" e a transformada de Hough.

Figura B.13 - Reconstrução e histograma de uma imagem ruidosa contendo uma célula ganglionar, usando o "Vector Tracer" e a transformada de Hough.

Figura B.14 - Reconstrução e histograma de uma imagem contendo uma célula ganglionar (fig. 4.18.a rotacionada), usando o "Vector Tracer" e a transformada de Hough.

Figura B.15 - Reconstrução e histograma de uma imagem ruidosa contendo uma célula ganglionar, usando o "Vector Tracer" e a transformada de Hough.

Figura B.16 - Reconstrução e histograma de uma imagem contendo células do hipocampo, usando o "Vector Tracer" e a transformada de Hough.

Figura B.17 - Reconstrução e histograma de uma imagem ruidosa contendo células do hipocampo, usando o "Vector Tracer" e a transformada de Hough.

Figura B.18 - Reconstrução e histograma de uma imagem contendo células do hipocampo (fig. 4.22.a rotacionada), usando o "Vector Tracer" e a transformada de Hough.

Figura B.19 - Reconstrução e histograma de uma imagem ruidosa contendo células do hipocampo, usando o "Vector Tracer" e a transformada de Hough.

Figura B.20 - Reconstrução e histograma de uma imagem contendo células do hipocampo, usando o "Vector Tracer" e a transformada de Hough.

Figura B.21 - Reconstrução e histograma de uma imagem ruidosa contendo células do hipocampo, usando o "Vector Tracer" e a transformada de Hough.

Figura B.22 - Reconstrução e histograma de uma imagem contendo células do hipocampo (fig. 4.26.a rotacionada), usando o "Vector Tracer" e a transformada de Hough.

Figura B.23 - Reconstrução e histograma de uma imagem ruidosa contendo células do hipocampo, usando o "Vector Tracer" e a transformada de Hough.

Figura B.24 - Reconstrução e histograma de uma imagem contendo células de músculo, usando o "Vector Tracer" e a transformada de Hough.

Figura B.25 - Reconstrução e histograma de uma imagem ruidosa contendo células de músculo, usando o "Vector Tracer" e a transformada de Hough.

Figura B.26 - Reconstrução e histograma de uma imagem contendo células de músculo (fig. 4.30.a rotacionada), usando o "Vector Tracer" e a transformada de Hough.

Figura B.27 - Reconstrução e histograma de uma imagem ruidosa contendo células de músculo, usando o "Vector Tracer" e a transformada de Hough.

Figura B.28 - Reconstrução e histograma de uma imagem contendo células de músculo, usando o "Vector Tracer" e a transformada de Hough.

Figura B.29 - Reconstrução e histograma de uma imagem ruidosa contendo células de músculo, usando o "Vector Tracer" e a transformada de Hough.

Figura B.30 - Reconstrução e histograma de uma imagem contendo células de músculo (fig. 4.34 rotacionada), usando o "Vector Tracer" e a transformada de Hough.

Figura B.31 - Reconstrução e histograma de uma imagem ruidosa contendo células de músculo, usando o "Vector Tracer" e a transformada de Hough.

A seguir apresenta-se as tabelas com os resultados estatísticos extraídos do processamento das imagens reais de neurônios apresentada na seção 4.1.2.

Tabela B.1 - Estatísticas sobre as imagens processadas usando "Vector Tracer".

	Vector Tracer 1				Vector Tracer 2			
	<i>sd</i>	<i>tp</i>	<i>pcd</i>	<i>ped</i>	<i>sd</i>	<i>tp</i>	<i>pcd</i>	<i>ped</i>
média	60.22	0.0100	40.14	55.00	54.64	0.0101	39.64	55.50
máximo	132.00	0.0120	57.25	69.78	127.00	0.0120	53.16	69.15
mínimo	31.00	0.0080	25.96	41.26	28.00	0.0080	26.38	45.35
desvio	30.50	0.0010	7.92	7.58	29.09	0.0012	7.29	7.00

Tabela B.2 - Estatísticas sobre as imagens processadas usando Hough.

	Transformada de Hough 1				Transformada de Hough 2			
	<i>sd</i>	<i>tp</i>	<i>pcd</i>	<i>ped</i>	<i>sd</i>	<i>tp</i>	<i>pcd</i>	<i>ped</i>
média	67.08	0.5511	67.79	32.22	70.75	1.1539	76.49	23.82
máximo	146.00	0.9500	91.76	50.72	156.00	1.9400	92.34	34.05
mínimo	29.00	0.3100	47.88	12.76	33.00	0.6300	65.52	11.49
desvio	31.95	0.1970	14.63	12.24	35.17	0.4181	8.82	6.09

Tabela B.3 - Estatísticas sobre as imagens ruidosas processadas usando "Vector Tracer".

	Vector Tracer 1				Vector Tracer 2			
	<i>sd</i>	<i>tp</i>	<i>pcd</i>	<i>ped</i>	<i>sd</i>	<i>tp</i>	<i>pcd</i>	<i>ped</i>
média	76.64	0.0106	43.48	50.67	71.44	0.0108	42.75	51.40
máximo	138.00	0.0130	58.36	68.30	131.00	0.0130	54.28	68.30
mínimo	32.00	0.0090	30.21	36.06	31.00	0.0090	30.00	40.14
desvio	31.10	0.0013	6.11	7.45	29.37	0.0014	5.49	7.12

Tabela B.4 - Estatísticas sobre as imagens ruidosas processadas usando Hough.

	Transformada de Hough 1				Transformada de Hough 2			
	<i>sd</i>	<i>tp</i>	<i>pcd</i>	<i>ped</i>	<i>sd</i>	<i>tp</i>	<i>pcd</i>	<i>ped</i>
média	77.03	0.9383	63.28	22.96	77.58	1.9794	71.41	15.98
máximo	140.00	1.3400	84.39	43.19	152.00	2.7700	84.76	29.60
mínimo	36.00	0.6800	44.23	1.12	39.00	1.3900	60.49	1.13
desvio	31.02	0.1908	13.05	14.96	32.42	0.4348	7.08	9.40

Tabela B.5 - Resultados do processamento de imagens reais utilizando "Vector Tracer" e transformada de Hough

Figura	Grupo	Vector1				Vector2				Hough1				Hough2			
		<i>sd</i>	<i>tp</i>	<i>pcd</i>	<i>pcd+ped</i>	<i>sd</i>	<i>tp</i>	<i>pcd</i>	<i>pcd+ped</i>	<i>sd</i>	<i>tp</i>	<i>pcd</i>	<i>pcd+ped</i>	<i>sd</i>	<i>tp</i>	<i>pcd</i>	<i>pcd+ped</i>
B.2	1	9	0.007	46.43	96.43	9	0.007	46.43	96.43	12	0.25	92.86	100	8	0.59	85.71	85.71
B.3	1r	10	0.008	46.43	91.07	10	0.007	46.43	91.07	8	0.7	76.79	76.79	7	1.720	75	75
B.2	2	8	0.007	39.29	100	8	0.007	39.29	100	10	0.31	67.86	96.43	10	0.96	75	89.29
B.3	2r	9	0.008	41.07	92.86	9	0.008	41.07	92.86	6	0.71	57.14	75	10	1.980	73.21	83.93
B.2	3	8	0.007	44.64	100	8	0.007	44.64	100	8	0.33	53.57	100	8	0.56	78.57	89.29
B.3	3r	9	0.008	41.07	92.86	9	0.007	41.07	92.86	10	0.77	41.07	80.36	6	1.610	67.86	78.57
B.4	1	23	0.007	40.48	100	23	0.008	40.48	100	27	0.31	93.45	104.76	27	0.68	92.86	98.81
B.5	1r	25	0.008	45.24	95.83	25	0.008	45.24	95.83	28	0.73	86.31	86.9	27	1.610	82.74	83.93
B.4	2	19	0.008	26.19	100	19	0.008	26.19	100	23	0.36	60.71	102.38	26	0.76	74.4	101.19
B.5	2r	21	0.008	32.74	95.83	21	0.009	32.74	95.83	19	0.75	63.69	90.48	25	1.660	68.45	86.9
B.4	3	15	0.008	19.05	97.62	15	0.008	19.05	97.62	32	0.34	42.86	105.95	23	0.78	82.14	110.12
B.5	3r	18	0.008	27.98	94.64	18	0.008	27.98	94.64	27	0.82	36.9	88.69	23	1.800	80.36	92.86
B.6	1	31	0.009	42.35	98.82	20	0.009	37.65	98.82	46	0.4	91.76	100	52	0.92	96.47	112.94
B.7	1r	35	0.01	41.76	92.35	29	0.01	39.12	92.35	52	1.060	84.12	84.71	52	2.460	84.71	85.59
B.6	2	16	0.01	9.41	98.82	16	0.01	9.41	98.82	44	0.5	76.47	107.06	40	0.92	74.12	111.76
B.7	2r	23	0.01	21.18	91.18	23	0.01	21.18	91.18	38	1.180	65.88	87.06	49	2.310	69.41	88.24
B.6	3	16	0.01	10.59	98.82	16	0.01	10.59	98.82	48	0.48	48.24	112.94	32	1.030	81.18	110.59
B.7	3r	22	0.01	21.47	91.18	22	0.01	21.47	91.18	52	1.100	38.53	89.71	33	2.530	74.12	90.88
B.8	1	44	0.008	57.25	98.51	37	0.008	53.16	98.51	45	0.32	86.62	100.37	52	0.65	88.48	101.49
B.9	1r	55	0.009	58.36	94.42	49	0.009	54.28	94.42	51	0.69	79.18	82.16	49	1.440	78.07	80.3
B.8	2	34	0.009	41.64	98.51	31	0.009	41.26	98.51	32	0.34	67.66	93.31	39	0.68	73.98	100.74
B.9	2r	48	0.009	48.33	94.42	44	0.009	47.58	94.42	43	0.71	62.83	85.13	46	1.450	72.49	87.73
B.8	3	32	0.009	36.8	98.51	29	0.009	36.43	98.51	29	0.37	52.04	97.4	33	0.76	69.52	98.14
B.9	3r	47	0.009	43.87	94.42	43	0.009	43.12	94.42	37	0.8	44.24	86.62	40	1.600	63.57	86.62

Figura	Grupo	Vector1				Vector2				Hough1				Hough2			
		<i>sd</i>	<i>tp</i>	<i>pcd</i>	<i>pcd+ped</i>	<i>sd</i>	<i>tp</i>	<i>pcd</i>	<i>pcd+ped</i>	<i>sd</i>	<i>tp</i>	<i>pcd</i>	<i>pcd+ped</i>	<i>sd</i>	<i>tp</i>	<i>pcd</i>	<i>pcd+ped</i>
B.10	1	43	0.009	51.67	97.77	35	0.008	47.58	97.77	50	0.31	86.25	101.12	50	0.63	86.25	99.26
B.11	1r	41	0.009	50.56	95.17	35	0.009	47.96	95.17	59	0.71	84.39	86.62	54	1.460	84.76	87.36
B.10	2	33	0.009	43.87	97.77	31	0.009	43.87	97.77	37	0.34	59.85	93.68	39	0.7	72.12	95.17
B.11	2r	34	0.009	43.49	94.8	33	0.009	43.49	94.8	44	0.75	58.74	84.39	45	1.520	66.54	84.01
B.10	3	31	0.009	42.75	97.77	29	0.009	42.75	97.77	30	0.38	50.93	95.91	34	0.77	71.75	96.65
B.11	3r	32	0.009	42.75	94.8	31	0.009	42.75	94.8	36	0.81	48.33	88.85	39	1.640	68.03	89.59
B.12	1	42	0.009	46.36	98.08	42	0.009	46.36	98.08	54	0.31	88.89	107.66	52	0.66	92.34	109.2
B.13	1r	45	0.009	49.43	96.93	41	0.009	47.13	96.93	53	0.68	81.99	84.29	48	1.390	81.61	83.52
B.12	2	38	0.009	43.3	97.7	38	0.009	43.3	97.7	37	0.32	62.45	96.55	39	0.66	74.33	98.47
B.13	2r	40	0.009	42.15	96.55	39	0.009	42.15	96.55	45	0.7	60.15	85.44	41	1.420	70.5	89.66
B.12	3	36	0.009	44.06	97.32	36	0.009	44.06	97.32	33	0.35	56.7	100	34	0.73	70.88	99.62
B.13	3r	38	0.01	41.38	95.4	37	0.01	41.38	95.4	38	0.76	52.49	89.66	39	1.580	67.82	91.19
B.14	1	49	0.009	52.06	96.63	46	0.009	50.94	96.63	47	0.31	91.76	110.11	50	0.7	89.51	107.87
B.15	1r	53	0.009	53.56	91.76	50	0.009	51.31	91.76	50	0.69	79.78	80.9	49	1.500	78.65	79.78
B.14	2	43	0.009	46.07	96.63	42	0.009	45.69	96.63	35	0.33	67.04	98.88	37	0.71	70.41	98.88
B.15	2r	50	0.009	46.82	91.76	49	0.009	45.32	91.76	40	0.72	62.92	85.77	41	1.510	66.67	82.77
B.14	3	39	0.009	41.2	96.25	39	0.009	41.2	96.25	34	0.38	61.42	100.75	34	0.76	69.29	99.25
B.15	3r	45	0.009	41.2	90.64	45	0.009	41.2	90.64	36	0.78	49.44	83.15	40	1.610	63.3	89.89
B.16	1	81	0.01	39.46	90.23	73	0.01	38.51	90.23	111	0.63	86.21	103.45	122	1.520	87.55	103.64
B.17	1r	99	0.011	43.49	88.89	93	0.011	43.3	88.89	110	1.040	81.42	83.91	115	2.480	81.42	84.29
B.16	2	77	0.011	34.29	89.85	71	0.011	33.14	89.85	101	0.68	60.73	98.47	102	1.420	72.99	97.51
B.17	2r	92	0.011	37.55	87.93	86	0.011	36.59	87.93	104	1.070	58.43	87.55	105	2.300	71.07	87.74
B.16	3	71	0.011	31.8	90.04	67	0.011	31.61	90.04	84	0.73	52.3	97.89	80	1.550	66.86	96.36
B.17	3r	88	0.011	37.74	88.31	83	0.012	37.55	88.31	95	1.120	51.15	86.97	85	2.390	66.67	87.74

Figura	Grupo	Vector1				Vector2				Hough1				Hough2			
		sd	tp	pcd	pcd+ped	sd	tp	pcd	pcd+ped	sd	tp	pcd	pcd+ped	sd	tp	pcd	pcd+ped
B.18	1	79	0.01	39.23	90.77	69	0.01	38.85	90.77	111	0.69	88.65	105.38	123	1.470	90.19	107.88
B.19	1r	105	0.011	45.77	87.88	96	0.011	45.58	87.88	104	1.090	75.77	78.85	113	2.470	76.73	80.19
B.18	2	74	0.01	34.42	90.96	65	0.011	35.19	90.96	97	0.67	65.19	103.27	101	1.470	73.27	100.96
B.19	2r	100	0.011	41.73	87.5	92	0.011	41.73	87.5	101	1.050	54.62	81.35	99	2.350	62.12	82.88
B.18	3	68	0.01	32.12	89.81	63	0.011	31.54	89.81	74	0.75	47.88	96.92	86	1.530	69.23	95.77
B.19	3r	96	0.011	40.96	86.35	89	0.012	41.15	86.35	88	1.130	44.23	83.65	94	2.420	62.5	83.85
B.20	1	127	0.011	48.13	93.97	117	0.012	47.13	93.97	146	0.83	84.91	104.74	156	1.670	87.79	106.47
B.21	1r	138	0.012	48.13	93.39	131	0.013	47.84	93.39	140	1.180	75	77.3	152	2.440	79.17	81.32
B.20	2	113	0.012	40.66	93.82	106	0.012	41.38	93.82	105	0.84	62.36	99.71	116	1.920	70.11	102.01
B.21	2r	126	0.013	40.52	92.96	123	0.013	40.52	92.96	126	1.200	58.05	84.2	132	2.740	63.94	85.06
B.20	3	111	0.012	38.65	92.96	105	0.012	39.22	92.96	109	0.93	51.72	102.44	98	1.920	65.52	99.57
B.21	3r	122	0.013	38.07	92.53	120	0.013	37.93	92.53	112	1.330	46.98	88.07	113	2.770	61.49	87.5
B.22	1	132	0.011	47.18	92.62	127	0.012	47.18	92.62	136	0.83	83.36	98.7	155	1.650	86.54	102.46
B.23	1r	137	0.012	48.05	89	129	0.012	47.9	89	136	1.200	76.27	78	142	2.420	78.29	80.17
B.22	2	118	0.012	41.1	93.63	110	0.012	41.68	93.63	101	0.86	61.22	100.14	117	1.790	71.92	101.16
B.23	2r	128	0.013	42.98	89.58	117	0.013	43.42	89.58	126	1.190	58.47	83.79	123	2.590	64.69	83.36
B.22	3	116	0.012	40.38	93.05	107	0.012	40.81	93.05	97	0.95	52.68	99.86	101	1.940	65.85	98.41
B.23	3r	125	0.012	41.68	89	114	0.013	41.68	89	109	1.340	48.91	87.99	110	2.770	60.49	85.09
B.24	1	51	0.01	46.52	94.28	38	0.009	45.27	94.28	65	0.45	88.81	106.47	63	0.91	87.31	107.71
B.25	1r	77	0.01	47.26	98.51	65	0.011	47.26	98.51	87	0.85	81.09	88.06	88	1.740	82.09	89.8
B.24	2	37	0.01	33.58	95.52	28	0.01	32.59	95.52	45	0.49	63.68	96.27	45	0.99	75.62	99
B.25	2r	64	0.011	39.05	99.25	56	0.011	36.82	99.25	69	0.89	62.69	90.8	69	1.830	75.12	96.27
B.24	3	34	0.01	31.59	95.02	32	0.01	32.84	95.02	42	0.53	55.47	98.26	45	1.080	71.64	99.75
B.25	3r	61	0.011	38.56	98.76	54	0.011	37.81	98.76	53	0.98	56.97	94.28	60	2.000	68.16	97.76

Figura	Grupo	Vector1				Vector2				Hough1				Hough2			
		<i>sd</i>	<i>tp</i>	<i>pcd</i>	<i>pcd+ped</i>	<i>sd</i>	<i>tp</i>	<i>pcd</i>	<i>pcd+ped</i>	<i>sd</i>	<i>tp</i>	<i>pcd</i>	<i>pcd+ped</i>	<i>sd</i>	<i>tp</i>	<i>pcd</i>	<i>pcd+ped</i>
B.26	1	49	0.009	49.38	95.29	38	0.01	45.91	95.29	58	0.45	84.62	106.95	64	0.95	89.33	110.17
B.27	1r	79	0.01	55.83	100	71	0.011	51.61	100	92	0.84	82.63	91.32	85	1.760	80.89	89.58
B.26	2	37	0.01	31.27	96.03	30	0.01	31.76	96.03	44	0.48	66	100.25	41	0.99	71.71	96.28
B.27	2r	67	0.011	45.91	100.74	65	0.011	44.91	100.74	73	0.88	65.26	94.29	64	1.850	71.22	93.8
B.26	3	31	0.01	28.29	96.03	29	0.01	29.53	96.03	42	0.52	51.61	95.78	38	1.100	67.25	98.01
B.27	3r	63	0.011	42.43	100	61	0.011	41.19	100	56	0.96	55.58	94.04	57	1.960	67.74	96.53
B.28	1	58	0.01	44.26	96.6	48	0.01	44.26	96.6	72	0.52	86.6	101.91	78	1.100	87.23	98.72
B.29	1r	80	0.011	46.6	99.15	73	0.011	47.23	99.15	84	0.89	83.4	88.3	81	1.870	80.21	85.53
B.28	2	45	0.011	28.3	96.81	40	0.011	28.51	96.81	61	0.58	62.13	97.02	63	1.170	72.55	95.11
B.29	2r	70	0.012	32.13	99.57	65	0.012	32.77	99.57	73	0.94	61.06	89.15	71	1.920	73.62	94.04
B.28	3	41	0.01	25.96	95.74	37	0.01	26.38	95.53	53	0.62	48.72	95.11	56	1.310	66.81	95.53
B.29	3r	68	0.011	30.21	98.51	64	0.011	30	98.3	62	1.020	48.09	91.28	60	2.120	66.6	95.32
B.30	1	60	0.01	51.91	97.02	49	0.01	50	97.02	75	0.55	85.11	97.87	86	1.150	87.23	102.55
B.31	1r	91	0.011	47.87	97.02	82	0.011	47.45	97.02	96	0.88	76.6	82.55	95	1.910	78.3	84.47
B.30	2	47	0.01	30.85	97.23	42	0.01	29.79	97.23	65	0.58	65.32	99.36	65	1.220	71.06	94.68
B.31	2r	78	0.011	36.81	97.02	74	0.011	35.96	97.02	78	0.92	61.06	86.6	83	1.960	70.85	90.21
B.30	3	47	0.01	28.72	96.38	41	0.011	27.23	96.38	58	0.62	53.4	98.3	53	1.310	69.15	96.81
B.31	3r	77	0.011	34.04	96.6	73	0.011	33.19	96.6	67	0.99	50	89.57	66	2.080	65.32	91.06

onde : Figura corresponde à imagem da figura relacionada e o grupo representa o grupo de parâmetros usados. A letra *r* identifica uma imagem ruidosa. O tempo de processamento (*tp*) é em relação à um computador PENTIUM 133mhz.

ANEXO C

ESTRUTURAS DE DADOS

Para a utilização dos algoritmos, houve a necessidade da implementação de vários tipos de estruturas de dados especiais, tais como pilhas, listas ligadas e vetores, que serão descritas a seguir.

Listas

Para armazenar os segmentos de reta das aproximações, obtidas através do “Vector Tracer” e da transformada de Hough, foi implementada uma estrutura de listas dinâmicas encadeadas (TENENBAUM *et al*, 1990). Essa lista encadeada possui a seguinte estrutura :

```
struct pixnode
{
    int    y;
    int    x;
    double c
    struct pixnode *next;
};

typedef struct pixnode *PIXNODEPTR;
```

Nessa estrutura, “y” contém a coordenada em y e “x” contém a coordenada em x da extremidade de um segmento. O campo “c” armazena códigos como o ângulo de inclinação da reta (θ) e distância (ρ) para segmentos extraídos pela transformada de Hough. Para os segmentos

extraídos pelo “Vector Tracer” esse campo não é utilizado. A representação gráfica dessa estrutura de listas ligadas encontra-se na figura A.1.

Figura C.1 - Estrutura de listas ligadas.

Vetores

Dentre os vetores ou “arrays” especiais utilizados na forma de uma tabela nessa implementação destacamos os seguintes :

- Tabela com os valores da banda senoidal adaptativa no intervalo $[0, \pi)$.
- Tabela com os valores dos senos e cossenos no intervalo $[0, \pi)$.

A descrição desses vetores se encontra abaixo.

Tabela da largura da banda senoidal adaptativa

São duas as tabelas geradas para esta finalidade. Uma delas possui a largura da banda adaptativa para o espaço de Hough com resolução simples e a outra para o espaço de Hough com resolução dupla. Essas tabelas armazenam os valores de $m(\theta)/2$ (expressão (19) na seção 2.2.3) usado para o mapeamento do espaço paramétrico no intervalo $[0, \pi)$. A diferença das duas tabelas é que uma possui passo de um em um grau (bwi) e a outra possui passo de meio em meio grau ($bwid$). A tabela bwi é utilizada para TH1 e Backmap1 e a tabela $bwid$ para TH2 e Backmap2.

Tabela dos cosenos e dos senos.

A tabela de senos e cosenos utilizados para o mapeamento da banda senoidal no espaço paramétrico de Hough de acordo com a expressão (2), visa a redução do tempo de processamento para o transformada de Hough e também para o algoritmo de “Backmapping”. Para o espaço de Hough com resolução simples (HS1 e HS2), cria-se dois vetores *tcos*[0,179] e *tsin*[0,179] e armazena-se os valores dos senos e cosenos para cada θ utilizando passo de um em um grau. Para o espaço de Hough com resolução dupla (HSD1 e HSD2) cria-se dois vetores *tcosd*[0,359] e *tsind*[0,359] e armazena-se os valores de seno e cosseno para cada θ usando passo de meio em meio grau.

Pilhas

As estruturas de pilhas são utilizadas no algoritmo de perseguição de contornos descrito na seção 2.1.11, e no algoritmo de “Backmapping” para armazenamento dos picos do acumulador de Hough classificados em ordem decrescente. Essa estrutura de pilha foi implementada utilizando-se alocação dinâmica e possui a seguinte estrutura:

```
struct snode
{
    int    y;
    int    x;
    int    c;
    struct snode *next;
}

typedef struct pnode *NODEPTR;
```

Na perseguição de contornos, os campos “y” e “x” armazenam as coordenadas cartesianas do “pixel” e o campo “c” não é utilizado. No algoritmo

de "Backmapping" os campos "y" e "x" contém os parâmetros θ e ρ do acumulador de Hough respectivamente, e "c" contém o valor do pico no acumulador.

A representação gráfica desta estrutura se encontra na figura A.2.

Figura C.2 - Estrutura de pilha.

ANEXO D

SUGESTÃO DE PARÂMETROS PARA RECONSTRUÇÃO E ANÁLISE DE IMAGENS

A seguir sugerimos alguns parâmetros, podendo ser necessário um ajuste fino conforme a aplicação. Em imagens ruidosas, o valor do “gap” (TH), deve ser ajustado para 0, de forma a reduzir a detecção de retas falsas. Para a reconstrução de imagens contendo objetos de escala reduzida, recomenda-se o ajuste conforme as tabelas D.1.a e D.1.b para o “Vector Tracer” e tabelas D.2.a e D.2.b para a transformada de Hough respectivamente.

Tabela D.1 - Parâmetros do “Vector Tracer” para reconstrução de imagens com pequenos detalhes (a) e detalhes maiores (b).

Técnica/parâmetro	vlen	vth1	vth2	vth3
Vector1	1 a 3	0.4 a 0.5	≥ 0.9	—
Vector2	1 a 3	0.4 a 0.5	≥ 0.9	≥ 0.99

(a)

Técnica/parâmetro	vlen	vth1	vth2	vth3
Vector1	3 a 5	0.4 a 0.5	≥ 0.99	—
Vector2	3 a 5	0.4 a 0.5	≥ 0.99	≥ 0.995

(b)

Tabela D.2 - Parâmetros da transformada de Hough para reconstrução de imagens com pequenos detalhes (a) e detalhes maiores (b).

Técnica/parâmetro	hth1	hth2	n	gap	w
Hough1	≥ 3	2 a 4	3 a 5	0 a 2	1.0 a 1.5
Hough2	≥ 3	2 a 4	3 a 5	0 a 2	1.0 a 1.5

(a)

Técnica/parâmetro	hth1	hth2	n	gap	w
Hough1	≥ 3	2 a 4	1 a 3	1 a 2	1.0 a 2.0
Hough2	≥ 3	2 a 4	1 a 3	1 a 2	1.0 a 2.0

(b)

ANEXO E

RECURSOS EMPREGADOS

Para a elaboração deste trabalho nas várias fases e para a avaliação do desempenho foram utilizados os seguintes recursos:

Hardware

- . Microcomputador PC/AT PENTIUM 133 Mhz / 16Mbytes de RAM
- . Estação de trabalho DEC ALPHA-AXP 3000/800 (OSF/1-2.0)
- . Estação de trabalho IBM/590 (AIX-4.2)
- . Microscópio CARL ZEISS modelo AXIOLAB acoplada à uma câmera CCD.
- . Câmera colorida CCD SONY modelo SSC-C374 com resolução de 500x700.

Software

- MS-WINDOWS 95
- Slackware Linux v.3.00 / Xwindows X11R6
- Borland C/C++ 4.5 e GNU gcc/g++ v.2.70 para a geração dos programas
- MS Word 6.0 para a edição de textos
- MS Excell 5.0 para a geração de gráficos
- Corel-Draw para a confecção de desenhos ilustrativos
- Photo Styler 2.0 SE e Corel-PhotoPaint para tratamento de imagens

REFERÊNCIAS BIBLIOGRÁFICAS

- [1] ABE, K., ARCELLI, C., HELD, A., **Split and Merge for Hierarchical Contour Sketch via Dominant Point Detection**, In : ICDAR'1991, Sept. 1991, St. Malo, France.
- [2] AGUI, T., IITSUKA, H., NAKAJIMA, M., **Image Processing for Urban Map Utilizing Pyramid Hierarchical Data**, Systems, Computers and Controls, vol.13, n.5, pp.56-63, 1982.
- [3] AGUI, T., NAKAJIMA, M., NAGAO, T., **Applied Image Processing with Turbo Pascal**, Tokyo: Kogakusha, 1988 , 341p (em japonês).
- [4] ANSARI, N., DELP, E. J., **On Detecting Dominant Points**, Pattern Recognition, vol.24, n.5, pp.441-451, 1991.
- [5] AOYAMA, H., KAWAGOE, M., **A Piecewise Linear Approximation Method Preserving Visual Feature Points of Original Figures**, CVGIP: Graphical Models and Image Processing, vol.53, n.5, Sept. 1991.
- [6] ATTNEAVE, F., **Some informational aspects of visual perception**, Psychological Review., vol.61, n.3, pp.183-193, 1954.
- [7] BALLARD, D. H., **Generalizing the Hough transform to detect arbitrary shapes**, Pattern Recognition, vol.13, n.2, pp.111-122, 1981.
- [8] BARLOW, R. J., **Statistics : A Guide to Use of Statistical Methods in the Physical Sciences**, John Wiley & Sons Ltd., England, 1989, 204p.
- [9] BELL, B., PAU, L.F., **Contour Tracking and Corner Detection in a Logic Programming Environment**, IEEE PAMI, vol.12, n.9, pp.913-917, Sept. 1990.
- [10] BLASDEL, G. G., SALAMA, G., **Voltage-sensitive dyes reveal a modular organization in monkeys striate cortex**, Nature, vol.321, pp.579-585, Jun. 1986.
- [11] BOWNS, L., MORGAN, M. J., **Facial features and axis of symmetry extracted using natural orientation information**, Biological Cybernetics, n.70, pp.137-144, 1993.
- [12] BRESENHAM, J. E., **Algorithm for computer control of a digital plotter**, IBM Systems Journal, vol.4, n.1, pp.25-30, 1965.

- [13] BRESENHAM, J. E., **A Linear Algorithm for Incremental Digital Display of Circular Arcs**, Communications of the ACM, vol.20, n.2, pp.100-106, 1977.
- [14] CESAR JUNIOR, R. M., COSTA, L. da F., **Piecewise Linear Segmentation of Digital Contours in $O(N \cdot \log(N))$ Through a Technique Based on Effective Digital Curvature Estimation**, Real Time Imaging, vol.1, pp.409-417, 1995.
- [15] COSTA, L., da F., LENG, X., SANDLER, M.B., SMART, P., **Semiautomated analysis of clay samples**, Rev. Sci. Instrum., vol.62, n.9, pp.2163-2166, Sept. 1991.
- [16] COSTA, L. da F., **Effective Detection of Line Segments with Hough Transform**, Ph.D. thesis, King's College, University of London, London, UK, May 1992a.
- [17] COSTA, L. da F., **Towards a Versatile Framework for Intermediate-level Computer Vision**, In: MVA'92 IAPR WORKSHOP ON MACHINE VISION APPLICATIONS, Dec. 7-9, 1992b, Tokyo-Japan, pp.261-264.
- [18] COSTA, L. da F., SANDLER, M.B., **Effective detection of digital bar segments with Hough transform**, Computer Graphics, Vision, and Image Processing, vol.55, n.1, pp.180-191, May 1993.
- [19] COSTA, L. da F., RODA, V. O., KÖBERLE, R., **A Biologically-Inspired System for Visual Pattern Recognition**, In: ISIE'94 IEEE INT. SYMP. ON IND. ELECTRONICS, May 25-27th, 1994, Santiago-Chile, pp.25-30.
- [20] COSTA, L. da F., **Towards real-time detection of discrete straight features with a hybrid technique based on preliminary curve segmentation and zoomed-adaptive parametric mapping**, Real-Time Imaging, n.1, pp.203-213, 1995a.
- [21] COSTA, L. da F., **Computer vision based morphometric characterization of neural cells**, Rev. Sci. Instrum., vol.66, n.7, pp.3770-3773, Jul. 1995b.
- [22] DAVIES, E.R., PLUMMER, A.P.N., **Thinning Algorithms: A critique and a new methodology**, Pattern Recognition, vol.14, pp.53-63, 1981.
- [23] DUDA, R. O., HART, P. E., **Use of the Hough transformation to detect line and curves in pictures**, Communications of the ACM, vol.15, n.1, pp.11-15, Jan. 1972.
- [24] DUDA, R. O., HART, P. E., **Pattern classification and scene analysis**, John Wiley & Sons Ltd., New York, 1973, pp.328-339.

- [25] DUNHAM, J. G., **Optimum uniform piecewise linear approximation of planar curves**, IEEE Trans. PAMI, vol.8, n.1, pp.67-75, 1986.
- [26] EJIRI, M., *et al*, **Automatic recognition of engineering drawing and maps in image analysis application**, Kasturi, R., Trivedi (ed), Marcel Decker, pp.73-126, 1990.
- [27] ENGELBRECHT, J. R., WAHL, F. M., **Polyhedral Object Recognition Using Hough-Space Features**, Pattern Recognition, vol.21, n.2, pp.155-167, 1988.
- [28] EZZEL, B., **Windows 3.1 Graphics Programming**, Ziff-Davis Press, Emeryville, CA, 1992, 480p.
- [29] FAISON, T., **Borland C++ 4.5 Object-Oriented Programming**, 4.ed., Indianapolis, IN, USA, Sams Publishing, 1995, 904p.
- [30] FREEMAN H., **Computer processing of line-drawing images**, Computing Surveys, vol.6, n.1, pp.57-97, Mar. 1974.
- [31] FOLEY, J. D., VAN DAM, A., FEINER, S. K., HUGHES, J. F., **Computer Graphics : Principles and Practice**, Addison-Wesley Publishing Co., 1992, 1174p.
- [32] FORESTI, G., MURINO, V., REGAZZONI, C.S., VERNAZZA, G., **Grouping of Rectilinear Segments by the Labeled Hough Transform**, CVGIP : Image Understanding, vol.58, n.3, pp.22-42, Nov. 1994.
- [33] FULFORD, N. C., **The Fastrak automatic digitizing system**, Pattern Recognition, vol. 14, n.1-6, pp. 65-74, 1981.
- [34] GERIG, G., KLEIN, F., **Fast Contour Identification through Efficient Hough Transform and Simplified Interpretation Strategy**, In : Proc. 8th INT. CONF. PATTERN RECOGNITION, vol.1, pp.498-500, Paris, France, Oct. 27-31, 1986.
- [35] GERIG, G., **Linking Image-Space and Accumulator-Space: A New Approach for Object-Recognition**, In : 1st IEEE INT. CONF. COMPUTER VISION, London, pp.112-117, 1987.
- [36] GONZALES, R. C., WINTZ, P., **Digital Image Processing**. 2.ed. Addison-Wesley Publishing Co., 1987, 503p.
- [37] GUIL, N., VILLALBA, J., ZAPATA, E. L., **A Fast Hough Transform for Segment Detection**, IEEE Trans. Image Processing, vol.14, n.11, pp.1541-1548, 1995.

- [38] HORI, O., OKAZAKI, A., **High quality vectorization based on a generic object model**, Pre-Proc, IAPR WORKSHOP ON STRUCTURAL AND SYNTACTIC PATTERN RECOGNITION, New Jersey, Jun, 1990, pp.137-153.
- [39] HOUGH, P. V. C., **Method and means for recognizing complex patterns**, U.S. Patent 3.069.654, Dec. 1962.
- [40] HUBEL, D. H., WIESEL, T. N., **Receptive fields, binocular interaction, and functional architecture in the cat's visual cortex**, J. Physiolog. (London), n160, pp.106-154, 1962.
- [41] HUBEL, D. H., WIESEL, T. N., **Brain Mechanics of Vision**, Scientific American, pp.130-144, Sept.1979.
- [42] ILLINGWORTH, J., KITTLER, J., **A survey of the Hough transform**, Computer Vision, Graphics, and Image Processing, vol.44, p.87-116, 1988.
- [43] JAIN, R.C., BINFORD, T.O., **Dialogue : ignorance, myopia and naiveté in computer vision systems**, CVGIP : Image Understanding, vol.53, pp.112-117, 1991.
- [44] JAIN, R.C, KASTURI, R., SCHUNCK, B. G., **Machine Vision** , USA, McGraw-Hill, 1995, 549p.
- [45] JOSEPH, S. H., PRIDMORE, T. P., **Knowledge-directed interpretation of mechanical engineering drawings**, IEEE Trans. PAMI, vol.14, n.9, pp.928-940, Sep. 1992.
- [46] KASTURI, R., SIVA, S., O'GORMAN, L., **Techniques for line drawing interpretation: An overview**, In: MVA'90 IAPR WORKSHOP ON MACHINE VISION APPLICATIONS, nov.28-30, 1990, Tokyo, pp.151-160.
- [47] KERNIGHAN, B. W., RITCHIE, D. M., **C: A linguagem de programação**, Trad: Pedro Sérgio Nicoletti, Rio de Janeiro, RJ, Editora Campus, 1986, 208p.
- [48] KUROZUMI, Y., DAVIS, W., **Polygonal Approximation by the Minimax Method**, Computer Graphics and Image Processing, vol.19, pp.248-264, 1982.
- [49] LAM, L., LEE, S., SUEN, C. Y., **Thinning Methodologies - A Comprehensive Survey**, IEEE Trans. PAMI, vol.14, n.9, pp.869-885, Sept. 1993.
- [50] LEAVERS, V. F., **SURVEY: Which Hough Transform?**, CVGIP : Image Understanding, v.58, n.2, pp.250-264, Sept. 1993.

- [51] LI, Z., YOKOI, S., TORIWAKI, J., FUKUMURA, T., **Border Following and Reconstruction of Binary Pictures Using Grid Point Representation**, Trans. of the Institute of Electronics and Communication Engineer of Japan, vol.J65-D, n.10, pp.1203-1210, Oct. 1982. (em japonês)
- [52] MARANA, A. N., COSTA, L. da F., LOTUFO, R. de A., **Análise de Textura Utilizando Transformada de Hough e Morfologia Matemática**, In : VIII SIBGRAPI, out. 1995, São Carlos, SP, pp.135-142.
- [53] McILROY, M. D., **A note on discrete representation of lines**, AT&T Technical Journal, vol. 64, n. 2, pp.481-490, Feb. 1984.
- [54] MELTER, R. A., STOJMENOVIC, I., ŽUNIĆ, J., **A new characterization of digital lines by least square fits**, Pattern Recognition Letters, vol.14, n.2, pp.83-88, 1993.
- [55] MONTANARI, U., **A Note on Minimal Length Polygonal Approximation to a Digitized Contour**, Communications of the ACM, vol.13, n.1, pp.41-47, Jan. 1970.
- [56] O'GORMAN, L., **K x K Thinning**, Computer Vision, Graphics, and image Processing, vol.51, pp.195-215, 1990.
- [57] PAVLIDS, T., HOROWITZ, S. L., **Segmenting of Plane Curves**, IEEE Trans. on Computers, vol.C-23, n.8, pp.860-870, 1974.
- [58] PERROTTI, F. A., LOTUFO, R. de A., **Um Novo algoritmo Paralelo de Afinamento**, In: V SIBIGRAPI, pp.285-293, nov. 1992.
- [59] PETZOLD, C., **Programando para Windows 3.1**, Trad. J. R. D. Pereira dos Santos, São Paulo: Makron Books, 1993, 1034
- [60] PIKAZ, A., DINSTEIN, I., **An algorithm for polygonal approximation based on iterative point elimination**, Pattern Recognition Letters, vol.16, n.6, pp.557-563, 1995.
- [61] PRESS, W. H., TEUKOLSKY, S. A., VETTERLING, W. T., FLANNERY, B. P., **NUMERICAL RECIPES in FORTRAN**, 2.ed., New York, USA, Cambridge University Press, 1992, 961p.
- [62] RAMER, U., **An Iterative Procedure for the Polygonal Approximation of Plane Curves**, Computer Graphics and Image Processing, n.1, pp.244-256, 1972.
- [63] RAY, B. K., RAY, K. S., **Determination of Optimal Polygon from digital Curve Using L_1 Norm**, Pattern Recognition, vol.26, n.4, pp.505-509, 1993.

- [64] ROSENFELD, A., **Picture Processing by Computer**, Academic Press, 1971.
- [65] ROSENFELD, A., JOHNSTON, E., **Angle detection on digital curves**, IEEE Trans. on Computers, vol.22, n.9, 875-878, Sept. 1973.
- [66] ROSENFELD, A., WESKA, J. S., **An improved method of angle detection on digital curves**, IEEE Trans. on Computers, vol.24, n.9, pp.940-941, Sept.1975.
- [67] SANDLER, M. B., COSTA, L. da F., FRASER, D. A., LIN, W., **A Complete Computer Vision System for Polyhedron Recognition**, In : I ISBAI - SIMPÓSIO BRASILEIRO DE AUTOMAÇÃO INTELIGENTE, UNESP-SBA, Rio Claro, São Paulo, pp.209-222, 8-10 set, 1993.
- [68] SKLANSKY J., CHAZIN, R. L., HANSEN, B. J., **Minimum-Perimeter Polygons of Digitized Silhouettes**, IEEE Trans. on Computers, vol.C-21, n.3, pp.260-268, March 1972.
- [69] SMITH, R. W., **Computer processing of line images: A survey**, Pattern Recognition, vol. 20, n.1, pp.7-15, 1987.
- [70] SUN, C., **Symmetry detection using gradient information**, Pattern Recognition Letters, vol.16, pp.987-996, 1995.
- [71] TAMURA, H., **Further considerations on line thinning schemes**, Paper of IECEJ Technical Group on Pattern Recognition and Learning, PRL 75-66, Dec. 1975 (em japonês).
- [72] TAMURA, H., **A Comparison of Line Thinning Algorithms from Digital Geometry Viewpoint**, In : Proc. of 4th INT. JOINT. CONF. PATTERN RECOGNITION, Nov. 1978, Kyoto-Japan, pp.715-719.
- [73] TEH, C. H., CHIN, R. T., **On the detection of dominant points on digital curves**, IEEE Trans. PAMI, vol.11, n.8, pp.859-872, 1989.
- [74] TENENBAUM, A. M., LANGSAM, Y., AUGENSTEIN, M. J., **Data structures using C**, Englewood Cliffs, N.J., U.S.A.: Prentice-Hall, Inc., 1990, 622p.
- [75] TORIWAKI, J., **Digital Image Processing for Image Understanding**, 2.ed. Tokyo, Japan: Shokodo, 1991, 168p. (em japonês)
- [76] VAN AKEN, J. R., KILLEBREW JR., C., R., **Better Bit-Mapped Lines**, Byte, pp.249-253, March 1988.
- [77] WALNUM, C., **Borland C++ 4.x Tips, Tricks and Traps**, Trad: Leonardo N. Lopes dos Santos, Rio de Janeiro, RJ, Brasil, Axcel Books, 1994, 568p.

- [78] WANG, M.J., WU, W., HUANG, L, WANG, D., **Corner detection using bending value**, Pattern Recognition Letters, vol.16, n.6, pp.575-583, 1995.
- [79] WEISS, R., BOLDT, M., **Geometric grouping applied to straight lines**, IEEE, pp.489-495, 1986.
- [80] WEISKAMP, K., FLAMIG, B., **The Complete C++ Primer**, 2.ed., San Diego, CA, U.S.A.: Academic Press, Inc., 1992, 540p.
- [81] WILLIAMS, C. M., **An Efficient Algorithm for Piecewise Linear Approximation of Planar Curves**, Computer Graphics and Image Processing, n.8, pp.286-293, 1978.
- [82] WU, W., WANG, M., **Detecting the Dominant Points by the Curvature-Based Polygonal Approximation**, CVGIP: Graphical Models and Image processing, vol.55, n.2, pp.79-88, March 1993.
- [83] XU, L., OJA, E., **Randomized Hough Transform (RHT) : Basic Mechanisms, Algorithms, and Computational Complexities**, CVGIP : Image Understanding, vol.57, n.2, pp.131-154, Mar. 1993.
- [84] YAO, P., **Borland C++ 4.0 : Programming for Windows**, Trad: João Nobrega Tortello, São Paulo, SP, Brasil, Makron Books, 1995, 983p.
- [85] ZHANG, T. Y., SUEN, C. Y., **A Fast Parallel Algorithm for Thinning Digital Patterns**, Communications of the ACM, vol.27, n.3, pp.236-239, Mar. 1984.