Universidade de São Paulo

Instituto de Física de São Carlos

Departamento de Física e Informática

REVENDO O PROBLEMA DA DETECÇÃO DE RETAS ATRAVÉS DOS OLHOS DA ARANHA

Felipe Miney Gonçalves da Costa

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

EX.
01

USP/IFSC/SBI
8-2-001279

São Carlos – SP

1999

IFSC-USP BIBLIOTECA INFORMAÇÃO

da Costa, Felipe Miney Gonçalves

"Revendo o problema da detecção de retas através dos olhos da aranha"/
Felipe Miney Gonçalves da Costa – São Carlos, 1999

Dissertação (Mestrado) – Instituto de Física de São Carlos da Universidade de São Paulo, 1999
Área: Física Aplicada-opção Física Computacional
Prof. Dr. Luciano da Fontoura Costa

1. Detecção de retas. 2. Visão computacional. 3. Visão ativa

I. Título

27/07/9917:11

UNIVERSIDADE DE SÃO PAULO
Instituto de Física de São Carlos

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

E-mail: wladerez@if.sc.usp.br

Fone (016) 273-9333
Fax (016) 272-2218

MEMBROS DA COMISSÃO JULGADORA DA DISSERTAÇÃO DE MESTRADO DE FELIPE MINEY GONÇALVES DA COSTA APRESENTADA AO INSTITUTO DE FÍSICA DE SÃO CARLOS, DA UNIVERSIDADE DE SÃO PAULO, EM 06 DE JULHO DE 1999.

COMISSÃO JULGADORA:

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

Prof.Dr. Jan Frans Willem Slaets/IFSC-USP

Prof. Dr. Aparecido Nilceu Marana/UNESP-Bauru

05/07/9911:41

Aos meus pais, Helio e Neide,
e à namorada Cristina,
pelo amor e incentivo
que me sustentaram.

Agradecimentos

Ao orientador Dr. Luciano da Fontoura Costa, pelo incentivo, visão e por sempre saber qual o próximo passo.

Aos colegas do grupo de Visão Cibernética, pelo apoio, encorajamento e companheirismo.

Aos familiares, pelo carinho e incentivo.

À CAPES, pelo apoio financeiro que permitiu a realização deste trabalho.

Aos amigos Arthur, Ivan e Tiago, pelo encorajamento e inestimável companhia.

Aos professores do IFSC-USP que contribuíram para minha formação.

Aos técnicos do grupo de Visão Cibernética.

A Bruno Rosa, pelo apoio e amizade.

Às secretárias Cláudia Tofanelli e Wladerez Aparecida Gounella Caiado, pela assistência e apoio.

Aos Funcionários da Oficina Mecânica do IFSC-USP, pela ajuda na construção do protótipo.

Às bibliotecárias do IFSC-USP, pela ajuda na busca de referências bilbiográficas.

Aos amigos Alexandre, Cláudia, Hélio, Humberto, Márcio e Rodrigo.

A Igor, Renato, Antônia e Anita.

Índice

Lista de Figuras

Resumo

Abstract

1 – Introdução.....1

2 – Motivação: A Aranha Saltadora.....5

2.1 – Sistema Visual.....5

2.2 – Modelagem da Retina.....11

3 – Retas Digitais.....13

3.1 – Definição.....13

3.2 – Quantização.....15

3.3 – Caracterização.....22

3.4 – Generalização.....25

4 – Parametrização.....28

4.1 – Parâmetros.....28

4.2 – Definição Matemática.....29

4.3 – Mapeamento.....30

5 – Transformada de Hough e Janelas Lineares.....36

5.1 – Fundamentos da Transformada de Hough.....36

5.2 – Variações.....40

5.3 – Amostragem.....41

5.4 – Mapeamento com Janelas Lineares.....42

5.5 – Comparação.....43

6 – Detecção de Retas por Otimização.....45

6.1 – Introdução.....45

6.2 – "Nelder and Mead's Downhill Simplex".....45

6.3 – O Espaço de Parâmetros.....48

6.4 – Largura das Janelas Lineares.....49

6.5 – Máximos Locais.....52

6.6 – Simulated Annealing.....54
6.7 – Complexidade do Espaço de Parâmetros.....63
6.8 – Algoritmos Genéticos.....64
6.8.1 - Evolução e Genética na Biologia.....64
6.8.2 – O Algoritmo Genético de Holland.....65
6.8.3 - Elementos, Variações e Exemplo.....67
6.8.4 - Aplicação no Problema da Detecção de Retas.....73
7 – Implementação.....75
7.1 – Simulações.....75
7.2 – Protótipo.....82
7.2.1 – Interface.....87
7.2.2 – Controle.....90
8 – Resultados.....91
8.1 - Resultados Preliminares: Busca por Retas com e sem "Simulated Annealing".....91
8.2 – Algoritmo Genético para Determinação dos Parâmetros.....94
8.3 – Detecção de Retas.....99
8.4 – Protótipo.....106
9 – Conclusões e Pesquisa Futura.....108
10 – Bibliografia.....111

Lista de Figuras

Figura 1.1 – Imagem e resultado da detecção de bordas.....01
Figura 2.1 – Olhos da aranha saltadora.....05
Figura 2.2 – Campos visuais dos olhos da aranha saltadora.....06
Figura 2.3 – Músculos dos olhos principais da aranha saltadora.....07
Figura 2.4 - Receptores da camada 1 das retinas dos olhos principais.....08
Figura 2.5 - Movimentos da retina dos olhos principais.....09
Figura 2.6 – Modelos para a retina das aranhas saltadoras.....12
Figura 3.1 – Diagrama de Classificação de elementos retos.....14
Figura 3.2 – Vizinhanças 4 e 8 de um pixel.....16
Figura 3.3 - As direções do código de cadeia.....17
Figura 3.4 – "Square-Box Quantization".....18
Figura 3.5 – "Grid-Intersect Quantization".....19
Figura 3.6 – "Object Boundary Quantization".....19
Figura 3.7 – "Background Boundary Quantization".....20
Figura 3.8 – Retas digitais passando pela origem, com $0 < \theta < \pi/2$.....21
Figura 3.9 - Aplicação do critério de Melter para um conjunto de pontos que corresponde a uma reta digital.....25
Figura 3.10 - Aplicação do critério de Melter para um conjunto de pontos que não corresponde a uma reta digital.....25
Figura 3.11 – Convolução de retas com vetores digitais.....26
Figura 3.12 – Convolução de retas digitais com gaussianas.....27
Figura 3.13 – Convolução de máscaras circulares com reta digital.....27
Figura 4.1 - Parametrização *m-c*.....28
Figura 4.2 - Parametrização ρ–θ.....29
Figura 4.3 – Mapeamento de pontos em bandas.....31
Figura 4.4 – Mapeamento de reta digital.....32
Figura 4.5 – Regiões correspondentes a retas nos espaços *m-c* e ρ–θ.....33
Figura 4.6 – Um quadrante da figura anterior.....33
Figura 4.7 – Espaços de parâmetros para uma imagem com 5x5 pixels.....34

Figura 4.8 - Amostragem com grade regular. Exemplos de regiões não amostradas com pontos vermelhos, e de regiões amostradas mais de uma vez, com pontos amarelos.....35
Figura 5.1 – Matriz Acumuladora de Hough.....37
Figura 5.2 - Transformada de Hough de uma reta contínua.....38
Figura 5.3 - Transformada de Hough de uma reta descontínua.....38
Figura 5.4 - Transformada de Hough da região próxima a uma reta; Imagem sem ruído.....39
Figura 5.5 - Transformada de Hough da região próxima a uma reta; imagem ruidosa.....39
Figura 6.1 – Movimentos possíveis do Simplex.....46
Figura 6.2 - Fluxograma do método "Downhill Simplex".....47
Figura 6.3 – Patamares no espaço de parâmetros.....49
Figura 6.4 – Suavização do espaço de parâmetros. Varredura com janelas de largura 1(a), 2(b) e 3(c).....51
Figura 6.5 - Máximos locais no espaço de parâmetros.....52
Figua 6.6 - Efeito da largura das janelas lineares nos máximos locais.....53
Figura 6.7 - Encontrando o mínimo global.....55
Figura 6.8 - Probabilidade de um passo que aumente a energia, em função da variação de energia.....57
Figura 6.9 - Fluxograma de estratégia de "simulated annealing".....59
Figura 6.10 - Fluxograma do algoritmo Downhill Simplex com Simulated Annealing.....61
Figura 6.11 - Operador mutação.....66
Figura 6.12 - Operador crossover.....66
Figura 6.13 - Operador inversão.....67
Figura 6.14 - Fluxograma de algoritmo genético básico.....69
Figura 6.15 - Crossover em indivíduos diplóides.....71
Figura 7.1 - Aplicativo que gera o espaço de parâmetros através da varredura por janelas lineares.....75
Figura 7.2 - Região do espaço de parâmetros próxima ao pico.....76

Figura 7.3 - Máximo encontrado pelo algoritmo "downhill simplex".....77
Figura 7.4 - Aplicativo para avaliação dos métodos de otimização.....77
Figura 7.5 - Ajuste de ruídos na imagem das simulações.....78
Figura 7.6 - Menu para a visualização dos parâmetros
do método de otimização.....78
Figura 7.7 - Janela para visualização dos parâmetros.....79
Figura 7.8 - Janela para a edição dos parâmetros.....79
Figura 7.9 - Aplicativo para escolha de parâmetros via algoritmo genético.....80
Figura 7.10 – Esquema do protótipo.....82
Figura 7.11 – Diagrama de blocos do protótipo.....83
Figura 7.12 – Esquema do Driver para motor de passo.....84
Figura 7.13 – Protótipo.....85
Figura 7.14 – Protótipo com somente um fotodetector.....86
Figura 7.15 – Tela principal do aplicativo que controla o protótipo.....87
Figura 7.16 – Imagens capturada, filtrada e contornos detectados.....88
Figura 7.17 – Controle da rotação do motor de passo.....88
Figura 7.18 – Espaços de parâmetros gerados através do protótipo.....89
Figura 8.1 – Reta com ruído.....92
Figura 8.2 – Superfície de "fitness", para K=4000.....95
Figura 8.3 – Perfil de aranha utilizado em simulações.....96
Figura 8.4 – Imagem real e resultado da detecção de bordas.....98
Figura 8.5 – Resultados da detecção de uma reta, para K=4000.....101
Figura 8.6 – Resultados da detecção de uma reta, K=40000.....102
Figura 8.7 – Retas detectadas sobre os perfis de aranha, K=40000.....103
Figura 8.8 – Retas detectadas sobre os perfis de aranha, K=4000.....104
Figura 8.9 – Retas traçadas sobre contornos da figura 8.4, para K=4000.....105
Figura 8.10 – Retas traçadas sobre contornos da figura 8.4, para K=40000.....105
Figura 8.11 – Resultados obtidos com o protótipo.....107

Resumo

Visão é um processo que envolve uma grande quantidade de informações, as quais precisam ser otimizadas de alguma forma para propiciar um processamento eficiente. Grande parte das informações visuais estão contidas nos contornos de uma imagem e uma grande redução no volume dos dados pode ser conseguida com a análise dos contornos. Além dos contornos, a detecção de segmentos de reta é o próximo passo na compressão das informações visuais. A detecção de retas ocorre no sistema visual humano, e também no de outros seres vivos. Entre os invertebrados terrestres, o melhor sistema de visão é o das aranhas da família *Salticidae* e este apresenta características que facilitam a detecção de retas.

Este trabalho propõe um novo método de detecção de retas, baseado no sistema visual das aranhas saltadoras, que aborda este problema através de um enfoque inédito, por otimização. O método realiza a busca por retas através de janelas lineares. Para isso, a detecção de retas será feita em um espaço de parâmetros, com a utilização do algoritmo de maximização de funções "Downhill Simplex". O método desenvolvido leva em consideração a natureza discreta da imagem e do espaço de parâmetros utilizado, e este trabalho inclui um estudo detalhado destes espaços discretos. O método incorpora, para lidar adequadamente com as peculiaridades do problema, características como "Simulated Annealing" e largura adaptativa. O desempenho do método depende de um conjunto de parâmetros cujo comportamento é de difícil previsão, e a escolha de um conjunto foi realizada através de um algoritmo genético. O trabalho envolve também a construção de um protótipo para a realização de testes utilizando o método desenvolvido. Os resultados foram analisados quanto à precisão na detecção de retas, ao tempo de processamento e à movimentação das janelas lineares, relacionada aos esforços na busca por retas.

Abstract

Vision is a process that involves a large amount of information that need to be somehow optimized to allow efficient processing. Most of the visual information is contained in the contours of an image and a considerable reduction in the amount of data can be achieved by finding and processing these contours. The next step to further compress the visual data is to find straight segments, and represent the contours in terms of these entities. Straight-line segment detection is performed by the human visual system, as well as by other creatures. Among terrestrial invertebrates, the best visual system is that of the *Salticidae* family of spiders, also known as jumping spiders. This visual system presents some characteristics that facilitate the detection of straight-lines.

The present work proposes a new method for straight-line detection, based on the visual system of the jumping spiders, using linear windows. This method approach the straight-line detection problem through an optimization point of view yet unexplored in literature. The detection will be accomplished in a parameter space, using the "Downhill Simplex" maximization algorithm. The method considers the discrete nature of both the image and the parameter spaces, and this work includes a detailed analysis of these discrete spaces. The method also incorporates, to adequately deal with the specific characteristics of the problem, resources such as "Simulated Annealing" and adaptive width of the linear windows. The performance of the method depends on a set of parameters, which behavior is hard to predict, and the choice of an adequate set was made using a genetic algorithm. The work also involves the project and construction of a prototype, to evaluate the proposed method. Results were analyzed regarding their precision, processing time and the movements of the linear windows, related to the effort made to detect the straight lines.

1 – Introdução

A Visão computacional é um assunto que tem mantido cientistas ocupados em todo o mundo por décadas, e é um problema que ainda não foi resolvido de maneira geral e satisfatória. Os seres vivos, por outro lado, possuem sistemas de visão extremamente eficazes e versáteis, que garantiram a sobrevivência de suas espécies por milênios. Mesmo o mais avançado sistema de visão computacional existente hoje tem sua performance superada pelo de organismos menos evoluídos, como por exemplo o de alguns invertebrados.

Os processos que constituem a visão são extremamente complexos, e a quantidade de dados envolvidos é muito grande. Para lidar com essa grande quantidade de informações, o sistema de visão dos seres humanos utiliza mecanismos para a compressão da informação, eliminando redundâncias e enfatizando informações importantes para a sobrevivência. A informação que chega aos nossos olhos possui uma grande quantidade de redundâncias, e a maior parte dos dados significativos para o reconhecimento de objetos no campo visual está contida nas bordas destes objetos, como ilustra a Figura 1.1. Assim, os dados compondo um objeto quadrado com coloração uniforme podem ser reduzidos às suas bordas e à informação que determina a coloração no interior destas.

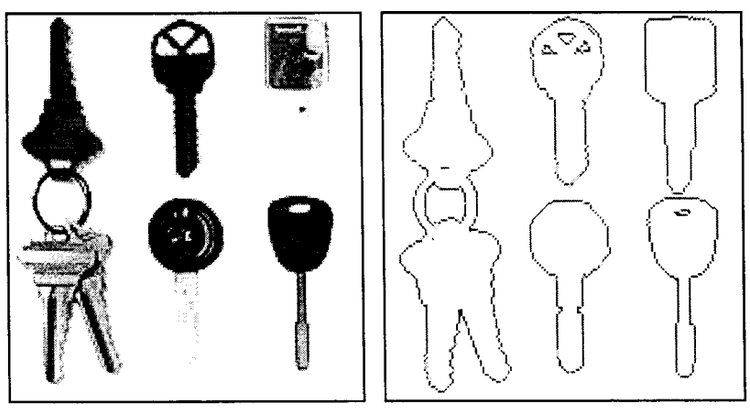


Figura 1.1 – Imagem e resultado da detecção de bordas.

Outra característica de imagens que pode ser comprimida é a presença de linhas retas, ou aproximadamente retas. Retas podem ser codificadas armazenando-se apenas seus pontos inicial e final, proporcionando uma grande redução na quantidade de informações necessárias para caracterizá-las [ZEKI 93]. No sistema visual humano existem estruturas neurais sensíveis a estímulos retos, de acordo com orientação, tamanho e posição destes [HUBEL 88]. Para que um estímulo visual reto possa ser reconhecido, diversas etapas de processamento ocorrem, desde a retina até o córtex visual.

Outra maneira existente na natureza de se reduzir a quantidade de informações para viabilizar o processamento visual é a visão ativa. Esta consiste em reduzir a área do campo visual com boa resolução, deixando a periferia do campo visual com uma menor densidade de fotoreceptores, e fazer com que o movimento dos olhos seja o responsável pelo foco nas regiões de interesse, colocando-as na parte central da retina. Nos seres humanos, a região central da retina é chamada de fóvea, e tem apenas meio milímetro de diâmetro. Nesta região a densidade de fotoreceptores é muito maior que no restante da retina.

Entre os invertebrados, uma família de artrópodes que possui um sistema visual muito eficiente e peculiar é a família *Salticidae*, ou das aranhas saltadoras. As aranhas dessa família não tecem teias para pegar suas presas, mas sim as perseguem e caçam. Por isso o seu sistema visual é o mais desenvolvido entre os invertebrados terrestres [FORSTER 85, LAND 85].

Os olhos principais das aranhas da família *Salticidae* têm como principal característica suas retinas, que têm formato estreito e alongado. Além disso, essas retinas são móveis, podendo realizar movimentos de rotação e translação. Essas características facilitam a detecção de retas, uma vez que o alinhamento da retina com uma reta provoca uma resposta acentuada.

Podemos então supor que, como os seres humanos, as aranhas saltadoras utilizam processamentos para reduzir a quantidade de informações visuais a serem interpretadas. Além disso: como o sinal de resposta para uma reta alinhada com a retina é forte, pouco ou nenhum processamento posterior é necessário para identificar essa característica da imagem analisada.

As soluções utilizadas por seres vivos na natureza são comprovadamente eficazes, pois permitiram a sobrevivência destes por milhares de anos. O estudo e utilização destas soluções em sistemas artificiais pode representar uma solução também eficiente neste tipo de sistemas, e as vantagens e desvantagens encontradas no ambiente artificial podem oferecer informações a respeito das capacidades e limitações do sistema biológico que as inspirou.

Partindo das características do sistema visual da aranha saltadora, mais especificamente utilizando uma janela de forma reta com capacidade de rotação e translação, este trabalho propõe um método para a detecção de retas em imagens. Este método utiliza algoritmos de maximização/minimização de funções para encontrar os pontos do espaço de parâmetros onde a resposta obtida por uma janela linear deslocando-se sobre a imagem seja maior, ou seja, para encontrar retas. Esta abordagem para o problema de detecção de retas, utilizando métodos de otimização para resolvê-lo tem grande potencial ainda inexplorado. O desenvolvimento do método leva em consideração problemas resultantes da natureza digital da imagem e do espaço de parâmetros, e algumas medidas, como a utilização de janelas lineares mais largas, serão adotadas para solucionar esses problemas.

O método desenvolvido também utiliza recursos para otimização eficiente na presença de máximos locais, que freqüentemente ocorrem quando se utiliza um mapeamento onde retas na imagem correspondem a pontos no espaço de parâmetros. Além disso, como o método de detecção de retas incorpora uma grande quantidade de parâmetros de controle, a escolha do melhor conjunto desses parâmetros é realizada através de uma seleção por algoritmo genético.

O espaço contendo as posições possíveis da janela linear é bidimensional, com o ângulo da reta normal à janela (θ) nas abcissas, e a distância da janela à origem (ρ) nas ordenadas. Sendo assim, o método de otimização a ser utilizado deve ser adequado para este caso.

O presente trabalho está organizado em 9 capítulos. O Capítulo 2 descreve em maiores detalhes o sistema visual e o comportamento visual das aranhas da família *Salticidae*, além de propor a modelagem das retinas em janelas lineares.

O Capítulo 3 fala sobre as retas digitais, suas características e definições, pois estas serão o objeto central utilizado no decorrer do trabalho. Também neste capítulo são tratadas as bandas digitais e é proposta uma generalização para os elementos retos.

A parametrização, por ser feita a partir de uma imagem digital, possui diversas peculiaridades que precisam ser consideradas, as quais serão mostradas no decorrer do Capítulo 4. A Transformada de Hough, a ser abordada no Capítulo 5, tem grande analogia com as janelas lineares, por utilizar o mesmo espaço de parâmetros e ser largamente utilizada para detecção de retas.

O Capítulo 6 é o mais extenso deste trabalho, e descreve passo a passo o método de detecção de retas proposto, motivando cada um dos recursos utilizados na seqüência em que foram incorporados ao método. Descrito o método, passamos à implementação deste, tratada no Capítulo 7, que mostra as características e interfaces dos programas criados para a realização do método de detecção de retas. Este capítulo descreve também o protótipo construído para a utilização do método. O Capítulo 8 é dedicado aos resultados obtidos com a utilização do método de detecção de retas desenvolvido, via simulações e através do protótipo.

Finalmente, o Capítulo 9 traz algumas conclusões e comentários a respeito do método de detecção de retas, avaliando os resultados obtidos.

2 – Motivação: A Aranha Saltadora

As aranhas da família *Salticidae* possuem um sistema de visão muito simples e específico se comparado ao dos primatas, mas ainda assim este sistema de visão é o mais sofisticado entre os invertebrados terrestres. Além disso, existem algumas peculiaridades nesse sistema que o tornam muito eficiente na detecção de retas, principalmente o formato alongado da retina dos olhos principais destes seres. Baseando-se neste sistema de visão, um método de detecção de retas será proposto no decorrer deste trabalho.

2.1 – Sistema Visual

O sistema visual das aranhas saltadoras é composto por quatro pares de olhos. Esses olhos são classificados em anteriores e posteriores, laterais e medianos. A figura 2.1 mostra a localização dos pares de olhos da aranha saltadora.

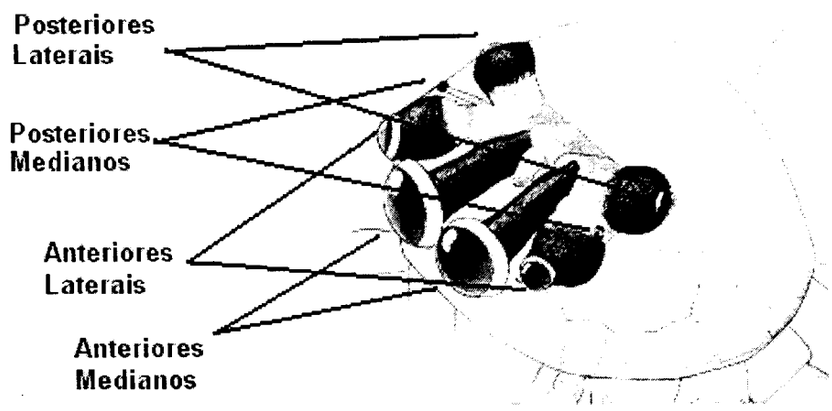


Figura 2.1 – Olhos da aranha saltadora (imagem obtida em http://spiders.arizona.edu/salticidae/++salticidae/anatomy/vision.html).

Os olhos posteriores medianos são atrofiados, tendo provavelmente perdido sua função em alguma etapa da evolução desses artrópodes. Os olhos laterais têm um grande ângulo de visão, fato evidenciado pelo formato esférico das retinas destes. São esses

pares de olhos os responsáveis pela detecção de movimento com ângulo de quase 360 graus. A Figura 2.2 mostra os olhos de uma aranha da família Salticidae e seus respectivos campos de visão.

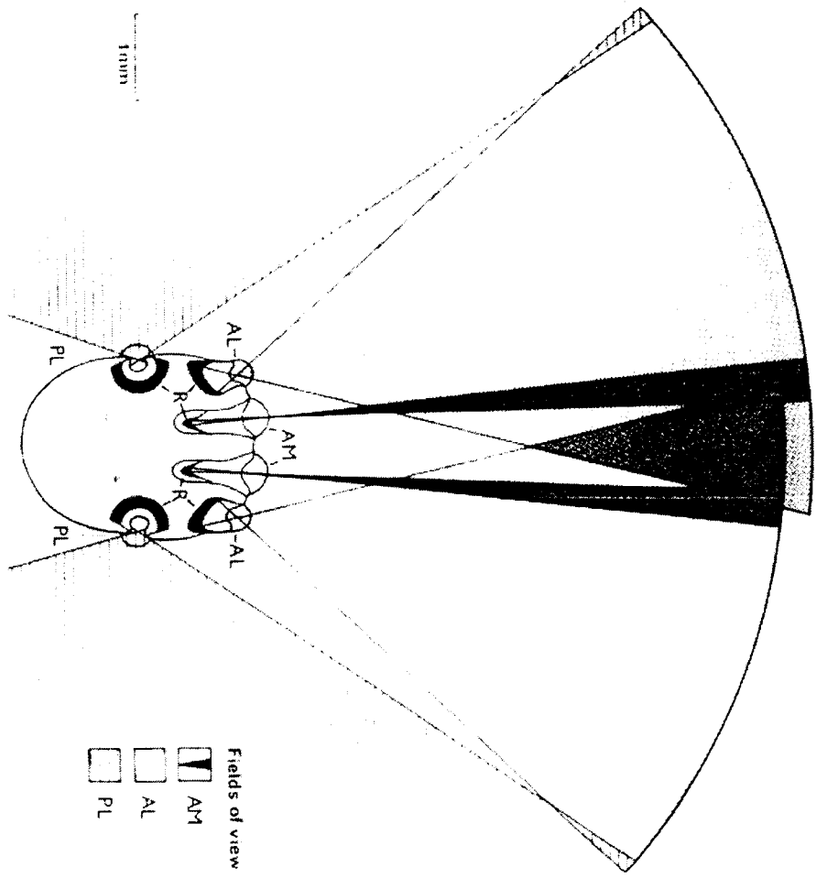


Figura 2.2 – Campos visuais dos olhos da aranha saltadora [FORSTER 82].

Podemos também notar que os campos visuais dos olhos anteriores laterais têm uma região de intersecção, indicando visão binocular nesta região do campo visual. Nesta região as aranhas saltadoras são capazes de percepção de profundidade, através da correspondência entre certas regiões da retina dos olhos anteriores medianos com distâncias ideais para perseguição de presas.

Os olhos anteriores medianos são também chamados de olhos principais. Estes olhos têm formato longo e tubular. Devido a isso, possuem uma boa resolução de

imagem, com maior distância focal e magnificação. Porém, esses olhos possuem um ângulo de visão estreito, como podemos ver na Figura 2.2. Para compensar essa limitação no campo de visão dos olhos principais, sua retina é móvel, capaz de movimentos de rotação e translação, acionada pelos músculos mostrados na Figura 2.3.

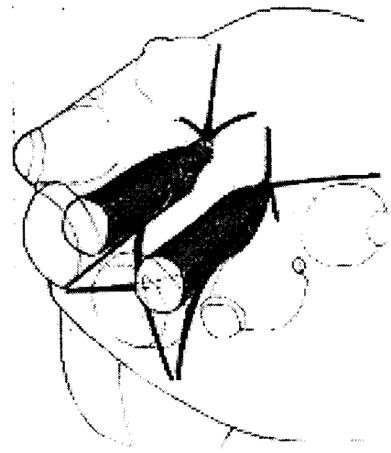


Figura 2.3 – Músculos dos olhos principais da aranha saltadora
(http://spiders.arizona.edu/salticidae/++salticidae/anatomy/vision.html)

Além disso, as retinas desses olhos têm formato alongado, com os fotoreceptores dispostos em linha, o que provoca uma resposta muito forte quando essas retinas se alinham com um estímulo reto no campo de visão da aranha. Esse par de olhos é responsável pela análise detalhada de cenas, para reconhecimento de presas e aranhas da mesma espécie, e a aranha saltadora faz isso através de uma série de movimentos complexos com as retinas dos olhos principais.

Nas espécies mais evoluídas da família *Salticidae*, o formato da retina dos olhos principais é levemente curvado, tendo o formato semelhante a um bumerangue. Land descobriu que os olhos principais da aranha se movem em conjunto, formando uma área em formato de "X" [LAND 85]. A Figura 2.4 mostra as retinas dos olhos principais de uma aranha saltadora.

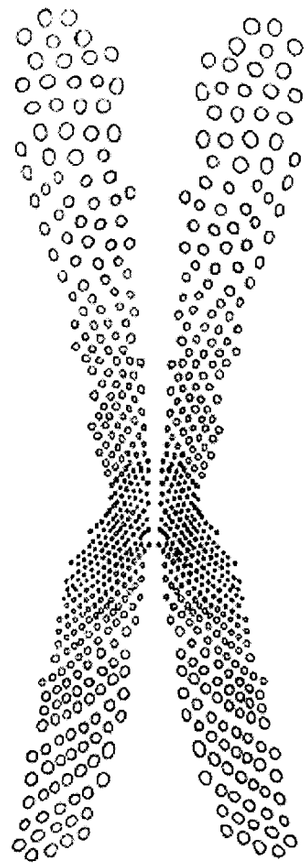


Figura 2.4 - Receptores da camada 1 das retinas dos olhos principais.

Outra característica da retina dos olhos anteriores medianos das aranhas saltadoras é que esta é formada por 4 camadas distintas de foto-receptores. A função destas camadas não foi completamente determinada, mas a hipótese mais provável é que estas respondam a diferentes comprimentos de onda e polarização da luz.

Land, em 1969 [LAND 85], identificou quatro tipos distintos de movimentos dos olhos principais da aranha saltadora. O primeiro deles é chamado de *Atividade Espontânea*, e consiste em movimentos laterais da retina, que ocorrem quando a aranha está alerta, independente da presença de alvos no campo de visão. Este movimento é representado na Figura 2.5 (i).

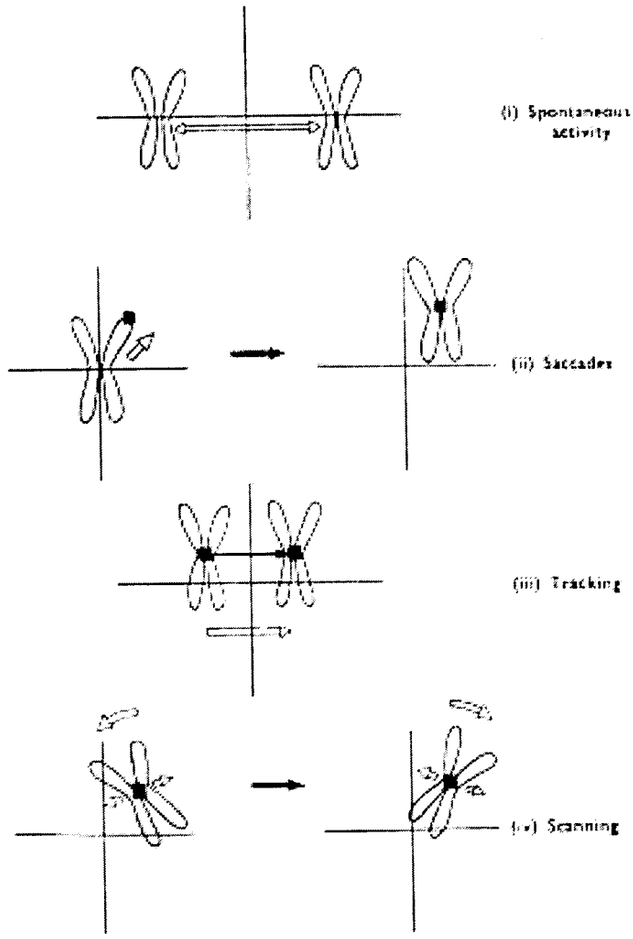


Figura 2.5 – Movimentos da retina dos olhos principais

O segundo tipo de movimento identificado foi o movimento de *Sacada*. Esse movimento é caracterizado por um deslocamento rápido da retina, que a centraliza em um alvo no campo de visão, como ilustra a Figura 2.5 (ii).

O terceiro movimento é o de *Rastreamento*, consistindo em movimentos da retina que fazem com que esta se mantenha centralizada e fixa em um alvo que se move pelo campo de visão. Esse tipo de movimento é mostrado na Figura 2.5 (iii) .

O último tipo de movimento, segundo Land, é o de *Varredura*. Este é o tipo mais complexo de movimento, e consiste em rotações e translações da retina em posições

próximas ao alvo. Esse movimento geralmente sucede um movimento de sacada, e serve para o reconhecimento de padrões. A Figura 2.5 (iv) ilustra esse movimento.

Segundo [FORSTER 82], a avaliação do alvo pelas aranhas saltadoras é baseada em três fatores: tamanho, distância e velocidade. Se o alvo tem mais que o dobro do tamanho da aranha, esta evita se aproximar, e foge se ele se aproxima.

Se o alvo tem tamanho adequado, e está a uma certa distância favorável, que varia de espécie para espécie, o comportamento da aranha se baseará na velocidade do alvo. Se o alvo estiver a uma velocidade superior a 4° por segundo, este será entendido como uma presa, e será perseguido. O comportamento de perseguição é controlado pelos olhos anteriores laterais.

Se o alvo se move a menos de 4° por segundo, ou está parado, a aranha deve decidir se este é uma presa ou se é outra aranha da mesma espécie. A análise do alvo é então realizada pelos olhos principais. Se o alvo for reconhecido como presa, este será atacado. Se for considerado uma outra aranha da mesma espécie, a reação dependerá do sexo e maturidade das aranhas envolvidas, além do fato de ser ou não época de acasalamento.

O reconhecimento de aranhas da mesma espécie envolve, além do reconhecimento visual, todo um ritual de comunicação visual. As aranhas da família *Salticidae* reconhecem outras da mesma espécie através de uma série de sinais com as patas e o abdômen, que identificam machos e fêmeas, e servem também para cortejar as fêmeas na época do acasalamento. A presença de cores fortes nas patas e abdômen dos machos evidencia a capacidade de distinção de cores nessas aranhas.

Experimentos a respeito do reconhecimento de outras aranhas da mesma espécie por parte de um macho da família *Salticidae* foram realizados por Crane, em 1949, e Drees, em 1952 [FORSTER 82]. Eles conseguiram, através de estudos com cartões onde estavam desenhados padrões similares a aranhas, determinar os fatores importantes que

incitam respostas das aranhas saltadoras. Concluíram que movimento, forma e tamanho são os principais fatores do reconhecimento, e que as cores reforçam o comportamento. Também foram realizados testes com associação de formas a estímulos aversivos, que comprovaram a capacidade das aranhas saltadoras de perceber e distinguir essas formas.

Em 1979, Forster realizou experimentos com potenciais presas, variando seus desenhos e velocidades [FORSTER 82]. As respostas obtidas das aranhas mostraram que formas preenchidas são perseguidas mais freqüentemente que as vazadas, e que alvos se movendo a maiores velocidades são perseguidos com mais freqüência independentemente da forma destes, evidenciando uma menor capacidade de reconhecimento de padrões nestas velocidades, em um comportamento provavelmente guiado pelos olhos anteriores laterais.

2.2 – Modelagem da retina

Nas aranhas saltadoras todo o processo de reconhecimento de padrões é realizado pelos olhos principais. A retina destes é alongada, e podemos adotar, em uma primeira análise, dois modelos para ela.

O primeiro modelo que pode ser usado é o de uma região fotodetectora alongada, que retorna como sinal a intensidade total do estímulo ao longo de sua extensão, como mostra a Figura 2.6 (a). Utilizando esse modelo, podemos varrer a imagem e reconstruí-la a partir dos resultados obtidos, analogamente à Transformada de Radon [JAIN 89]. Este modelo tem como principal vantagem a pouca capacidade de armazenamento de dados necessária para sua implementação e a aquisição direta (sem a necessidade de soma) do valor do estímulo para cada posição da janela linear. Entretanto, com ele não é possível se codificar a posição do estímulo ao longo da região fotodetectora, tornando mais trabalhoso o processo de se identificar as extremidades destes estímulos.

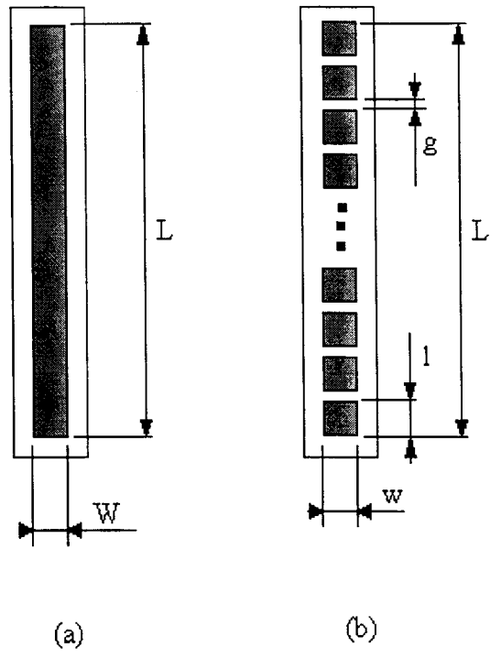


Figura 2.6 – Modelos para a retina das aranhas saltadoras

Outra forma de modelar a retina das aranhas saltadoras é como um arranjo linear de fotodetectores, mostrado na Figura 2.6 (b). Este modelo, além de ter maior analogia biológica (a retina é formada por um arranjo de células fotoreceptoras), permite também a codificação da posição de um estímulo ao longo da retina, fornecendo informação extra. Devido a tal característica, este será o modelo adotado no decorrer do trabalho.

Uma consideração importante a fazer é que, ao utilizarmos este modelo para a análise de imagens digitalizadas, precisamos assumir que o tamanho dos fotodetectores e o espaçamento entre eles é perfeitamente correspondente aos tamanhos e espaçamento dos "pixels" na resolução da quantização utilizada.

3 – Retas Digitais

Este capítulo tem por objetivo esclarecer alguns conceitos de retas digitais que serão utilizados durante o trabalho. A classificação dos elementos retos será feita de acordo com [COSTA 92], com a adição de algumas outras classes.

3.1 – Definição

Quando nos referimos a elementos retos, existem diversas considerações que precisam ser feitas quanto à natureza do espaço em questão. Elementos reais e ideais são diferentes, e espaços discretos implicam em diferenças significativas em relação aos contínuos.

Ao utilizarmos *Retas Ideais* e *Segmentos de Retas Ideais* (ISLS), estamos nos referindo aos elementos comumente usados para a representação de características de cenas em âmbito matemático. Estas entidades podem ser descritas com precisão por equações matemáticas, através de uma função do tipo $y = m.x+c$; são perfeitamente retas e possuem largura infinitesimal.

Entretanto, no mundo real, em cenas e imagens reais, o que encontramos são *Retas Reais* e *Segmentos de Retas Reais* (RSLS). Estas entidades diferem das retas e segmentos de reta ideais por não serem perfeitamente retas, mas é comum aproximarmos as entidades do mundo real por retas. O Diagrama da Figura 3.1 mostra as diversas classificações de elementos retos em imagens.

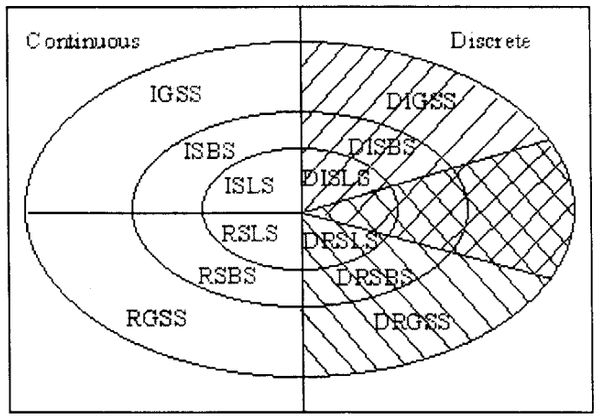


Figura 3.1 – Diagrama de Classificação de elementos retos (estendido de [COSTA 92]).

Quando passamos do mundo real ou matemático para uma cena digitalizada, uma transição de espaços contínuos para outro discreto, as representações de retas são agora chamadas de *Segmentos de Retas Digitais*. Estes são gerados pela quantização de um segmento de reta, que pode advir de uma cena real, ou diretamente a partir de uma reta ideal. As primeiras serão chamadas de *Segmentos de Retas Digitais Reais* (DRSLS, Digital Real Straight Line Segments), e as outras de *Segmentos de Retas Digitais Ideais* (DISLS, Digital Ideal Straight Line Segments).

Podemos também analisar os segmentos de reta como sendo casos particulares de Segmentos de Bandas Retas, com largura de banda unitária. Podemos ter bandas ideais, descritas matematicamente, ladeadas por retas ideais paralelas cuja distância entre si é a largura da banda. No mundo real, encontramos bandas reais, com bordas não perfeitamente retas ou paralelas, mas com características próximas às ideais, permitindo uma aproximação.

Assim como os segmentos de reta, as bandas retas podem ser originadas a partir de cenas reais ou ideais, e são chamadas, respectivamente, de *Segmentos de Bandas Retas Reais* (RSBS) *e Segmentos de Bandas Retas Ideais* (ISBS). No espaço discreto, temos os *Segmentos de Bandas Retas Digitais Reais* (RDSBS) e os *Segmentos de Bandas Retas Digitais Ideais* (IDSBS), também mostrados no diagrama da Figura 3.1.

O Círculo maior da figura 3.1 representa ainda outra generalização, chamada de Segmentos Retos Gerais. Estes serão tratados na Seção 3.4, e dizem respeito à convolução de retas com diversos tipos de entidades.

A intersecção entre o espaço de entidades reais e ideais, que ocorre na porção discreta do diagrama, é conseqüência da perda de informações devido à quantização das imagens, o que permite que a digitalização de uma reta real seja equivalente à digitalização de uma reta ideal.

3.2 – Quantização

Alguns conceitos básicos, necessários para a definição dos métodos de quantização, serão aqui apresentados e revisados.

Os primeiros conceitos importantes para os métodos de quantização são os conceitos de vizinhança de 4 e de vizinhança de 8.

Dado um pixel de coordenadas (x,y), a vizinhança de 4 desse pixel é definida como sendo os pixels de coordenadas (x-1,y), (x+1,y), (x,y-1) e (x,y+1). A distância desses pixels ao pixel (x,y) é de 1 pixel.

A vizinhança de 8 engloba, além dos pixels constituintes da vizinhança de 4, os pixels (x-1,y-1), (x-1,y+1), (x+1,y-1) e (x+1,y+1). A distância, em pixels, destes ao pixel (x,y) também é 1, mas a distância euclidiana entre estes é a diagonal, que mede $\sqrt{2}$.

A Figura 3.2 mostra as vizinhanças de 4 e 8 de um pixel.

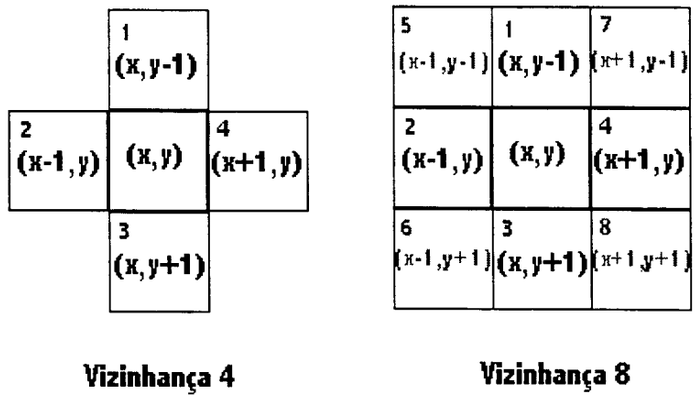


Figura 3.2 – Vizinhanças de 4 e 8 de um pixel.

Apresentados os conceitos de vizinhança de 4 e vizinhança de 8, as próximas definições importantes são as de caminho-4 e caminho-8. Um caminho de um pixel a outro é uma seqüência ordenada de pixels vizinhos. Se esses pixels têm vizinhança de 4, então o caminho formado é um caminho-4; se a vizinhança entre os pixels é uma vizinhança de 8, então o caminho é chamado de caminho-8 [GONZALEZ 87].

Ainda com relação à vizinhança dos pixels, podemos definir a conectividade das curvas. Se os pixels de uma curva formam um caminho-4, esta curva pode ser chamada de 4-conexa. Se o caminho formado for um caminho-8, a curva é dita 8-conexa.

Finalmente, um conceito importante e largamente utilizado para a caracterização de curvas digitais é o de "código de cadeia" (chain code). O código de cadeia de uma curva é uma seqüência de números inteiros de 0 a 7, indicando as direções a serem seguidas a partir de um pixel inicial para se obter a curva representada.

As direções são numeradas de 0 a 7, no sentido anti-horário, sendo 0 em direção ao pixel da direita, como mostra a Figura 3.3.

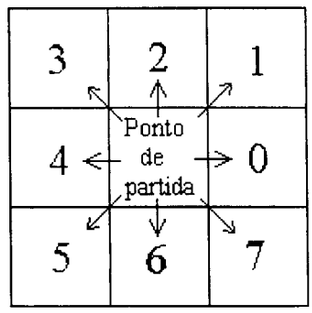


Figura 3.3 - As direções do código de cadeia.

Em se tratando dos esquemas de quantização, uma grande variedade destes pode ser encontrada na literatura. Apresentaremos aqui quatro métodos; dois para a quantização de linhas e dois para a quantização de bordas.

O primeiro método para quantização de linhas é o "Square-Box Quantization" (SBQ), ilustrado na Figura 3.4. Neste, são tomados o conjunto de "pixels" (x,y) para os quais pelo menos um ponto (X,Y) da linha contínua está na célula quadrada com centro em (x,y) e lado 1 [FREEMAN 74]. A Figura 3.4 mostra um exemplo da SBQ. Nesta, todas as curvas que passarem pela região colorida, chamada por Freeman de "campo receptivo", resultarão na mesma curva quantizada.

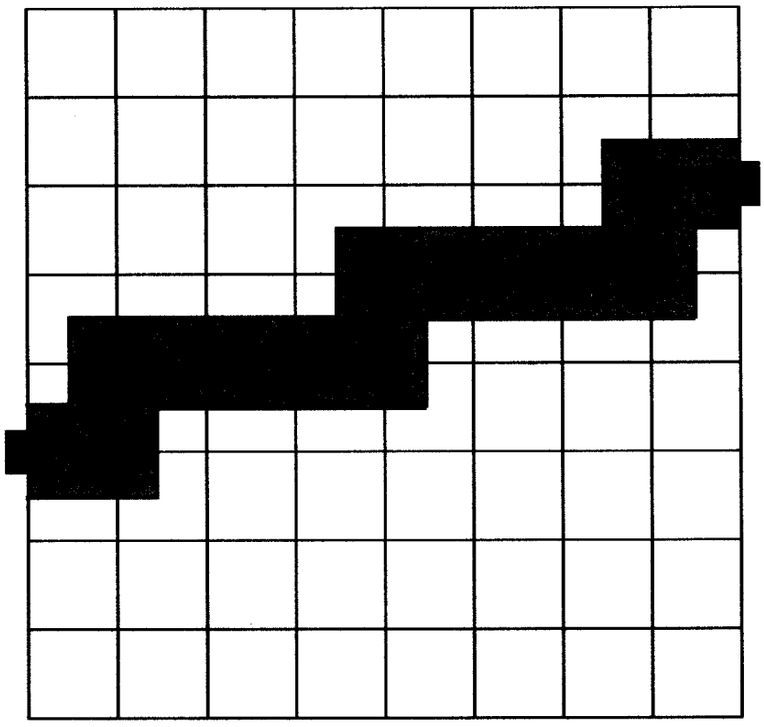


Figura 3.4 – "Square-Box Quantization"

O método conhecido como "Grid-Intersect Quantization" (GIQ) consiste em marcar, para cada ponto da imagem contínua (X,Y) onde X=x, o ponto para o qual y-0,5 $\leq$ Y < y+0,5. Um exemplo desse método de quantização é apresentado na Figura 3.5, com seu respectivo campo receptivo.

Os dois métodos para quantização de bordas que serão apresentados são o "Object Boundary Quantization" (OBQ) e o "Background Boundary Quantization" (BBQ). O primeiro deles, OBQ, marca os pontos dentro da região limitada pelo contorno contínuo sendo quantizado, e o segundo, BBQ, marca os pontos fora dessa região, adjacentes a ela. Exemplos dos dois métodos estão nas Figuras 3.6 e 3.7.

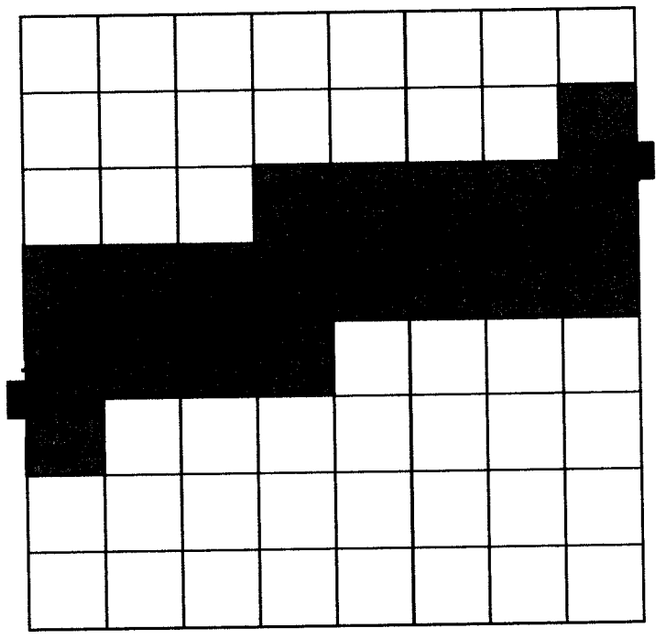


Figura 3.5 – "Grid-Intersect Quantization"

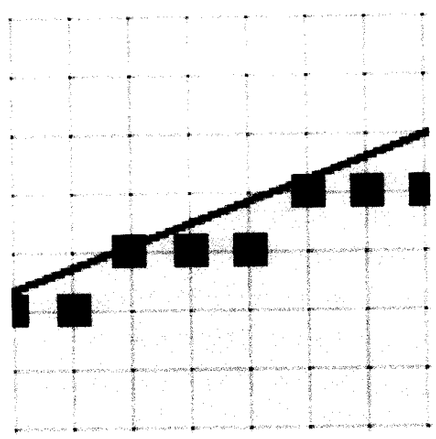


Figura 3.6 – "Object Boundary Quantization".

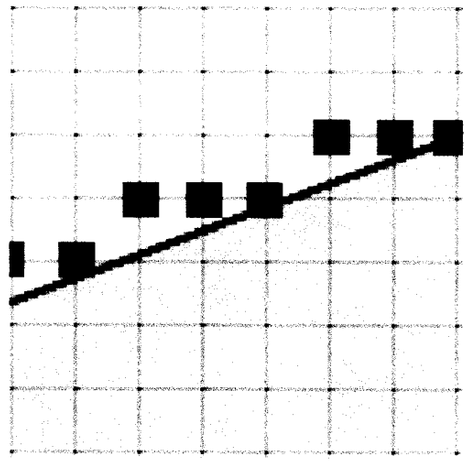


Figura 3.7 – "Background Boundary Quantization".

O resultado digital do método SBQ são curvas com caminho-4; enquanto o método GIQ resulta em curvas com caminho-8. As curvas com caminho-8 aproximam melhor entidades contínuas, pois evitam as redundâncias que geralmente aparecem nas curvas com caminho-4. Além disso, essas curvas possuem a vantagem de poderem ser representadas como funções matemáticas definidas em um domínio discreto. Assim, o método de quantização a ser utilizado no decorrer deste trabalho será o GIQ. É importante observar que, nessa descrição matemática, as retas devem ser separadas em dois grupos, dependendo de sua inclinação: o primeiro grupo englobaria as retas com inclinação entre –1 e 1, e as demais fariam parte do segundo grupo. A Figura 3.8 mostra todas as retas digitais passando pelo "pixel" central de uma imagem com 9 x 9 "pixels", para θ variando entre 0 e $\pi/2$.

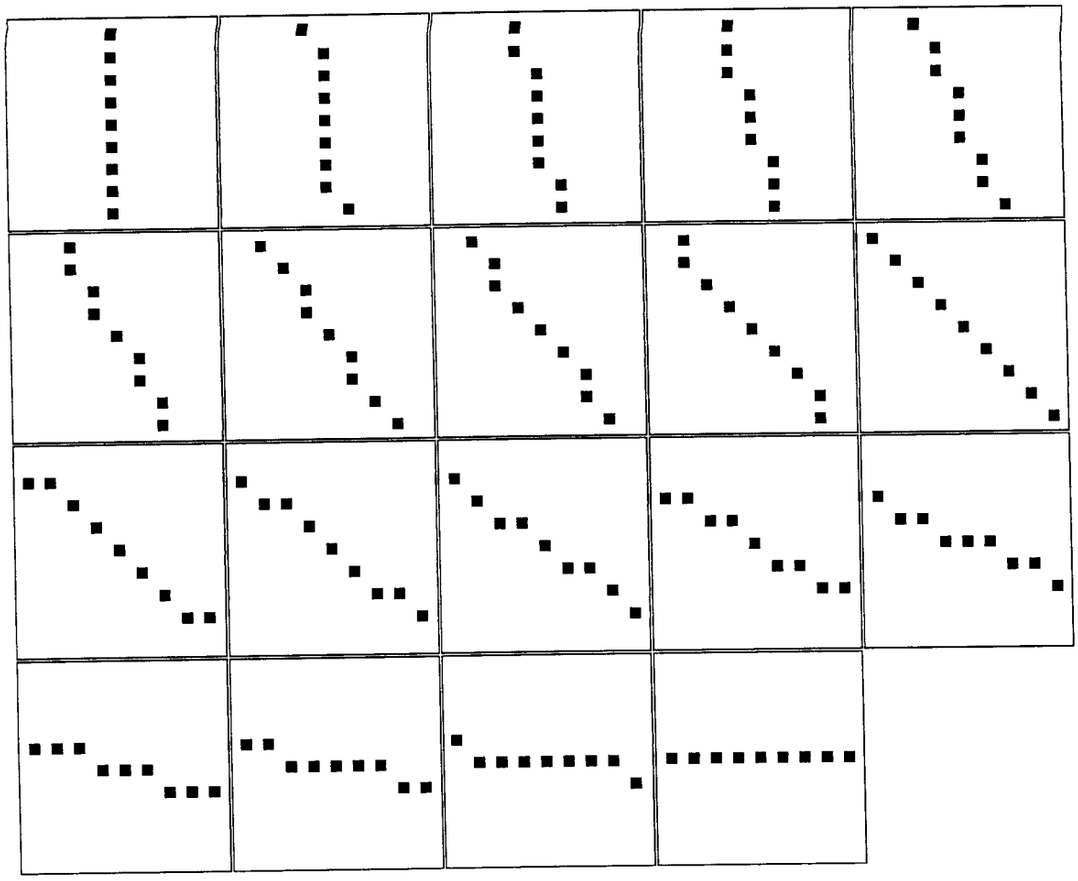


Figura 3.8 – Retas digitais passando pela origem, com $0 < \theta < \pi/2$.

3.3 – Caracterização

Para a caracterização de retas digitais é necessário, inicialmente, verificar quais conjuntos de pontos correspondem a retas digitais, e também determinar os parâmetros das retas reais correspondentes.

Uma das primeiras caracterizações de retas digitais foi apresentada por Freeman, em 1970 [FREEMAN 70], e era baseada em códigos de cadeia (chain codes). Segundo Freeman, a quantização GIQ de retas apresentaria três propriedades características:

(1) Estarão presentes no máximo duas direções, sendo estas consecutivas;
(2) Um dos valores do código sempre aparece isolado;
(3) Ocorrências sucessivas da direção isolada são o mais uniformemente espaçadas possível.

Porém, como a terceira condição é subjetiva, a caracterização de retas digitais apresentada por Freeman não é completamente exata.

Em 1974, Rosenfeld apresentou uma caracterização exata de retas digitais ideais que foi largamente utilizada por muito tempo, introduzindo o conceito de corda ("chord") [ROSENFELD 74]. Segundo Rosenfeld, uma curva digital entre os pontos A e B é um segmento de reta digital se esta apresentar a propriedade da corda. Verifica-se essa propriedade traçando uma reta ideal entre os pontos A e B, se para cada ponto da curva digital a distância entre este e a reta ideal é menor que 1, então a curva digital tem a propriedade da corda.

No mesmo ano, Brons apresentou um algoritmo recursivo capaz de gerar retas digitais, baseado nos três critérios de Freeman. Esse algoritmo utilizava o caráter periódico das retas digitais para gerá-las através da repetição de períodos, e não calculando cada "pixel" [BRONS 74].

Novamente aproveitando a regularidade das retas digitais; no ano seguinte, Arcelli e Massarotti apresentaram outra abordagem para a geração e caracterização das retas digitais [ARCELLI & MASSAROTTI 75]. O algoritmo apresentado baseava-se na convexidade da curva analisada, e tornava mais fácil o cálculo da propriedade da corda de Rosenfeld.

Em 1982, Kim e Rosenfeld definiram convexidade de uma região digital como sendo a possibilidade de, dados dois pontos nessa região, ligar esses pontos por uma reta digital que também pertence a região. Assim, uma curva digital é uma reta digital se e somente se ela é convexa [KIM & ROSENFELD 82].

Outro resultado interessante, também de 1982, foi o apresentado por Wu, que propôs uma versão mais estrita da terceira condição de Freeman e expandiu as duas primeiras condições para outros níveis hierárquicos das retas digitais ideais. O método descrito por Wu, para se obter os níveis hierárquicos superiores a partir do código de cadeia de uma curva digital, consiste em primeiramente remover a direção que aparece isolada, chamada de separador; e em seguida criar um novo código composto pelas quantidades do outro símbolo. Fazendo-se isso sucessivamente, o resultado será um único símbolo, ou o padrão definido pelas duas primeiras condições de Freeman deixará de existir [WU 82].

Por exemplo, se o código de cadeia da curva digital analisada for:

2 2 2 3 2 2 3 2 2 2 3 2 2 3 2 2 2 3 2 2 3 2 2 2 3 2 2 3

temos um símbolo que aparece sempre isolado (3), que será chamado de separador. As quantidades do símbolo (2) são 3 e 2, e o próximo nível hierárquico da representação, segundo Wu, seria:

3 2 3 2 3 2 3 2

Novamente, se seguirmos o mesmo procedimento, a seqüência resultante será:

1 1 1 1

Como restou apenas um símbolo, a seqüência inicial corresponde a uma reta digital.

Se a cadeia da curva analisada for:

4 4 4 5 4 4 5 4 4 5 4 4 5 4 4 4 5 4 4 5 4 4 5 4 4 5 4 4 4

Temos o separador (5), e os níveis superiores da seqüência ficam:

3 2 2 2 3 2 2 2 3

3 3

novamente caracterizando uma reta digital.

Finalmente, utilizando a cadeia:

1 2 2 1 2 2 1 2 2 2 1 2 2 2 1 2 2 1 2 2 1 2 2 2 1 2 2 2

Teremos no segundo nível:

2 2 3 3 2 2 3 3

Como não há um símbolo que aparece isolado, o padrão definido por Freeman foi quebrado, indicando que a seqüência não representa uma reta digital.

Mais recentemente, em 1993, Melter apresentou um novo método que possibilita verificar se um conjunto de pontos é uma reta digital, baseado no método dos mínimos quadrados [MELTER 93]. O método de Melter consiste em três passos:

(1) encontrar os parâmetros da melhor reta passando pelo conjunto de pontos dado, através do método dos mínimos quadrados;
(2) Com os parâmetros da reta encontrada, gerar a reta digital correspondente;
(3) Se a reta digital gerada corresponder ao conjunto inicial de pontos, então esse conjunto é uma reta digital.

As Figuras 3.9 e 3.10 mostram exemplos da aplicação do critério de Melter. Em (a), temos o conjunto de pontos que verificaremos se corresponde a uma reta digital; em (b) temos a melhor reta, obtida pelo método dos mínimos quadrados, que passa pelo conjunto analisado; e em (c) temos a quantização GIQ da reta encontrada.

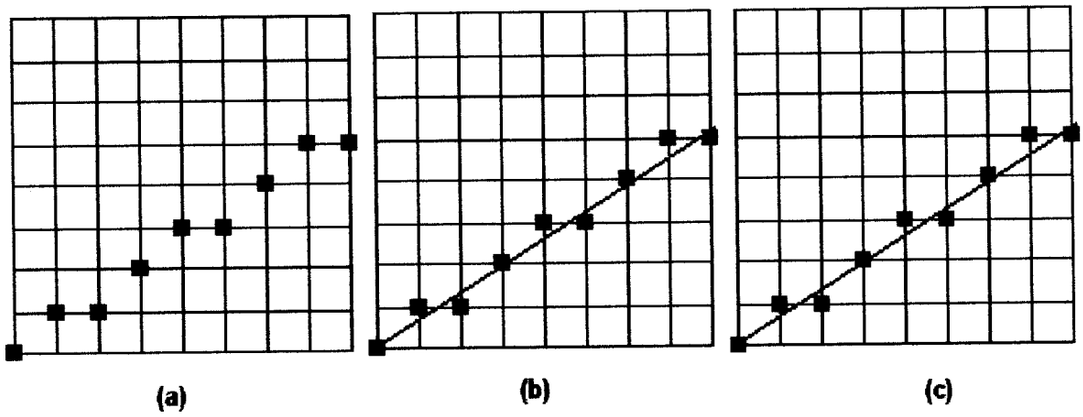


Figura 3.9 - Aplicação do critério de Melter para um conjunto de pontos que corresponde a uma reta digital.

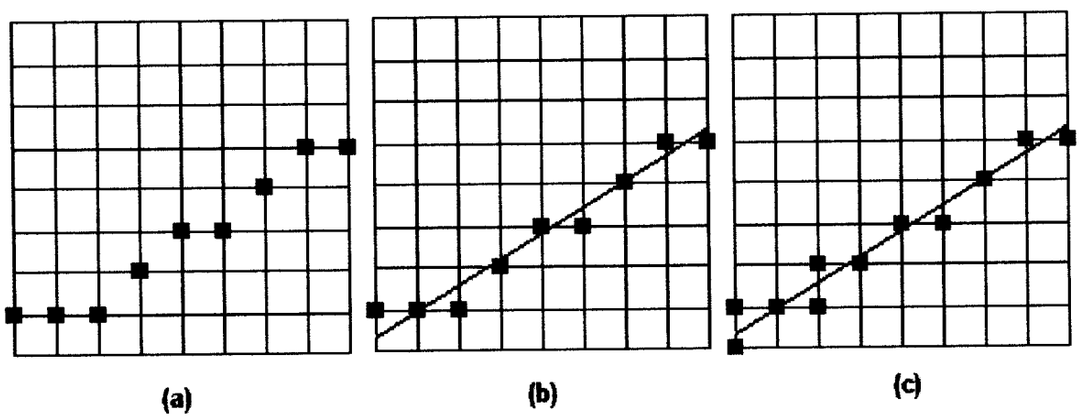


Figura 3.10 - Aplicação do critério de Melter para um conjunto de pontos que não corresponde a uma reta digital.

3.4 – Generalização

Nesta seção será apresentada uma generalização do conceito de retas digitais, introduzindo-se o conceito de segmentos retos digitais generalizados.

Os segmentos retos generalizados são uma extensão do conceito de retas digitais, com o objetivo de torná-las uma representação mais eficiente das retas reais. Como elementos retos reais, tanto no espaço contínuo como no discreto, não são na verdade

retos, e sim *aproximadamente* retos, podemos permitir uma certa tolerância nos elementos retos para que possamos ter uma melhor representação destes. Isso é feito através da convolução de retas digitais com outras entidades, como gaussianas, segmentos de reta e círculos.

Sob esse enfoque, as bandas digitais podem ser vistas como a convolução de retas digitais com segmentos de retas digitais, sendo estes verticais se a inclinação da reta é entre –1 e 1, e horizontais para inclinações fora dessa faixa; e tendo comprimento igual à largura de banda desejada. Assim, temos os resultados dessa operação gerando bandas digitais mostrados na Figura 3.11, para larguras de banda 1, 9, 14 e 22 pixels.

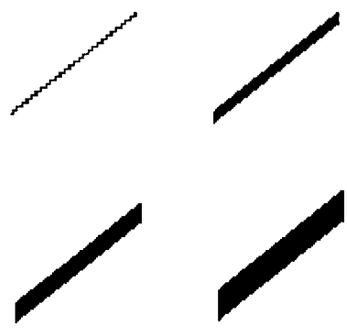


Figura 3.11 – Convolução de reta com vetores digitais, formando Bandas Digitais.

O resultado da convolução de retas digitais com gaussianas resulta em um "borramento" da reta digital, fazendo com que ela cubra uma região maior do espaço, e também aumentando a chance de correspondência com uma reta real. A Figura 3.12 mostra essa operação, com gaussianas cujas variâncias aumentam gradativamente (0,18,30,50).

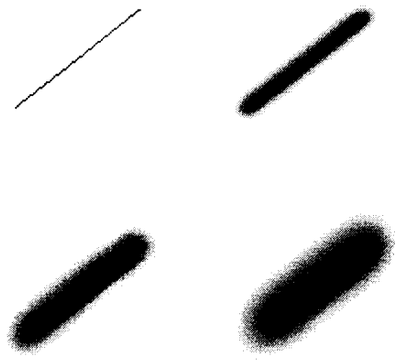


Figura 3.12 – Convolução de retas digitais com gaussianas

Essa operação de convolução pode também ser feita utilizando-se uma máscara circular, sendo o resultado mostrado na Figura 3.13, para círculos de diâmetros 1, 6, 13 e 21.

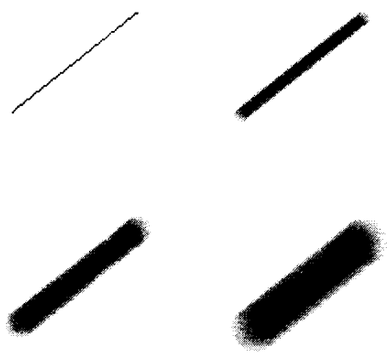


Figura 3.13 – Convolução de máscaras circulares com reta digital.

4 – Parametrização

A parametrização do espaço discreto x-y contendo as retas digitais pode ser realizada de diferentes maneiras, e alguns cuidados devem ser tomados. Esta seção mostra alguns tipos de parametrização e suas características, assim como as equações que devem ser utilizadas para essas parametrizações, levando em consideração a natureza discreta dos espaços analisados.

4.1 – Parâmetros

Para a parametrização de retas digitais, vários pares de parâmetros podem ser utilizados. Entre eles, podemos destacar dois:

(1) a inclinação (m) e a intersecção com o eixo das ordenadas (c)

o mapeamento do espaço x-y para o espaço *m-c* é feito através da equação

$y = m.x+c$

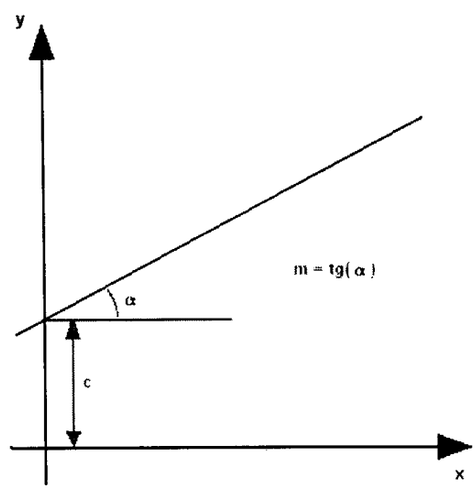


Figura 4.1 - Parametrização *m-c*.

(2) o ângulo do vetor normal (θ) e a distância à origem (ρ)

neste caso, o mapeamento do espaço x-y para o $\rho - \theta$ é feito com a equação

$y = (\rho\text{-}x.cos(\theta))/sen(\theta)$

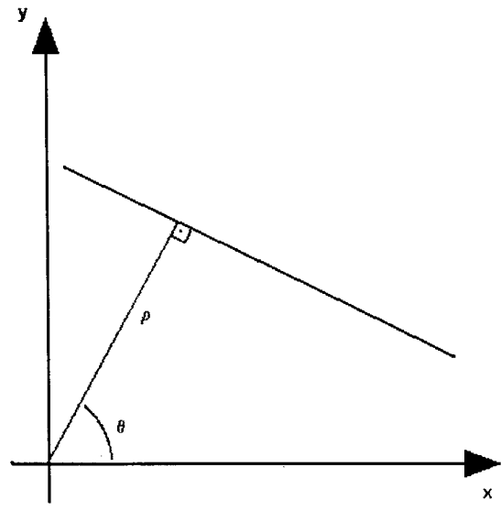


Figura 4.2 - Parametrização ρ–θ.

4.2 – Definição Matemática

Se considerarmos a natureza discreta de uma imagem, as equações utilizadas em sua parametrização se modificam. No espaço *m-c*, a correspondência não é mais de um ponto para uma reta, mas sim de um ponto para uma banda reta, e matematicamente a parametrização é feita da seguinte maneira:

$$m.x + c - \frac{\Delta}{2} \leq y < m.x + c + \frac{\Delta}{2}$$

Onde Δ representa a largura de banda. Analogamente, para o espaço ρ–θ, considerando a imagem como discreta, temos a parametrização:

$$\frac{\rho - x.\cos(\theta)}{\sin(\theta)} - \frac{\Delta}{2} \leq y < \frac{\rho - x.\cos(\theta)}{\sin(\theta)} + \frac{\Delta}{2}$$

Manipulando as expressões acima, chegamos àquelas que caracterizam o mapeamento de um para muitos:

$$y - mx - \frac{\Delta}{2} < c \leq y - mx + \frac{\Delta}{2}$$

$$y.sen(\theta) + x.\cos(\theta) + \frac{\Delta sen(\theta)}{2} < \rho \leq y.sen(\theta) + x.\cos(\theta) - \frac{\Delta.sen(\theta)}{2}$$

4.3 – Mapeamento

Um aspecto importante do processo de mapeamento entre a imagem e o espaço de parâmetros é que este processo mapeia um espaço discreto (x,y) em um espaço contínuo (m,c) ou (ρ,θ). Devido a esse fator, cada ponto no espaço x-y corresponde não a uma reta no espaço *m-c*, ou a uma senóide no espaço $\rho-\theta$, mas sim a bandas retas/senoidais nesses espaços de parâmetros.

Como conseqüência, ao mapearmos uma reta digital, cada ponto corresponderá a uma banda, reta no espaço *m-c* e senoidal no espaço $\rho-\theta$. A intersecção dessas bandas formará uma região com parâmetros *m-c* e $\rho-\theta$ que geram retas que, quando quantizadas, correspondem à reta mapeada.

A Figura 4.3 é composta por três partes. A primeira é uma imagem digital com 2 pontos, a segunda representa o espaço de parâmetros *m-c* com as bandas correspondentes a esses pontos, e a terceira é o espaço de parâmetros $\rho-\theta$. As bandas têm cor correspondente ao ponto que as gerou, e a região amarela representa a intersecção das bandas, região nos espaços de parâmetros que corresponde a retas passando pelos dois pontos no espaço x-y.

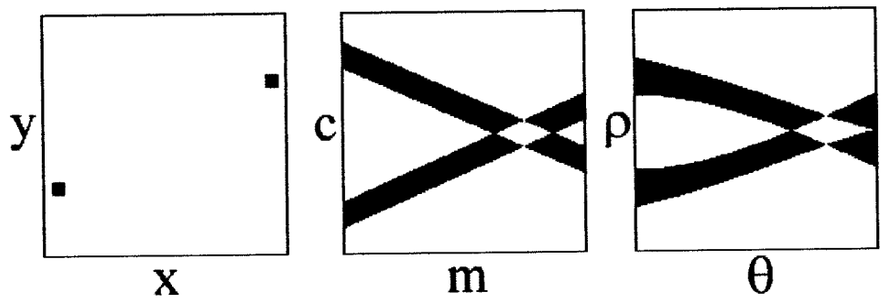


Figura 4.3 – Mapeamento de pontos em bandas.

As bandas em azul significam que qualquer reta no espaço *m-c*, assim como qualquer senóide no espaço ρ–θ, que esteja contida dentro da banda irá corresponder ao "pixel" em azul da figura à esquerda.

A seqüência de imagens da Figura 4.4 mostra a correspondência em bandas de todos os pontos de uma reta digital, e as regiões resultantes desse mapeamento.

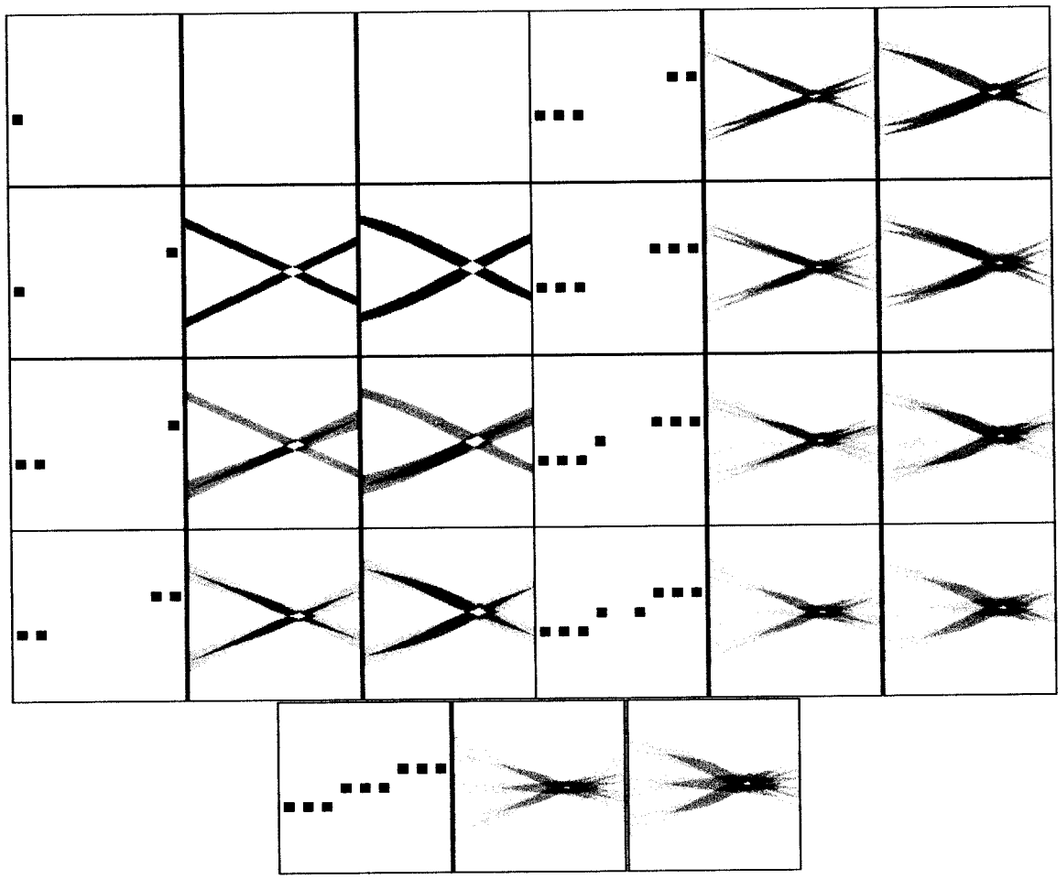


Figura 4.4 – Mapeamento de reta digital.

Repetindo o processo de mapeamento para todas as retas digitais existentes em uma imagem de um dado tamanho, podemos gerar um gráfico representando as regiões correspondentes a cada reta digital possível na imagem. A Figura 4.5 mostra essas regiões para as retas possíveis com inclinação entre –1 e 1, em uma imagem de 9x9 "pixels".

Para melhor visualizar as regiões do espaço de parâmetros correspondentes a retas, e devido à simetria presente na figura anterior, foi gerado apenas um quadrante do espaço mostrado anteriormente, com maior resolução, conforme mostra a Figura 4.6.

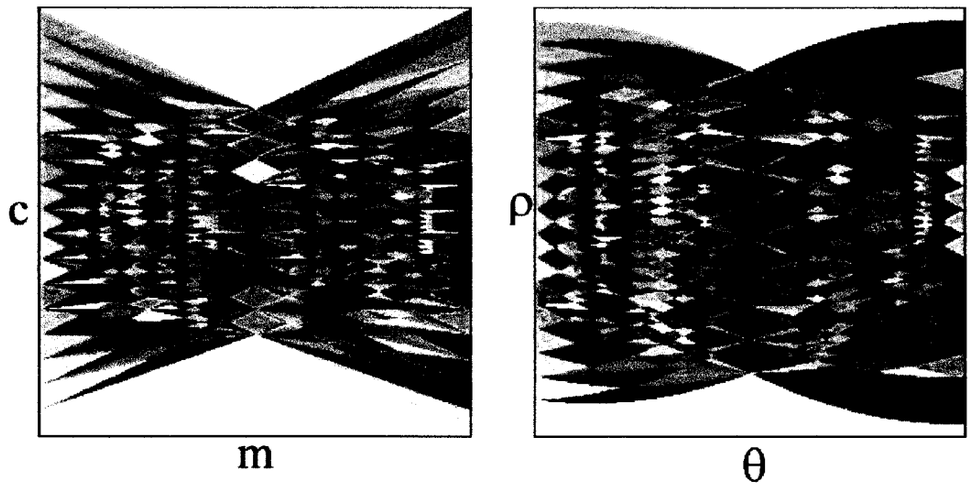


Figura 4.5 – Regiões correspondentes a retas nos espaços *m-c* e *ρ–θ*.

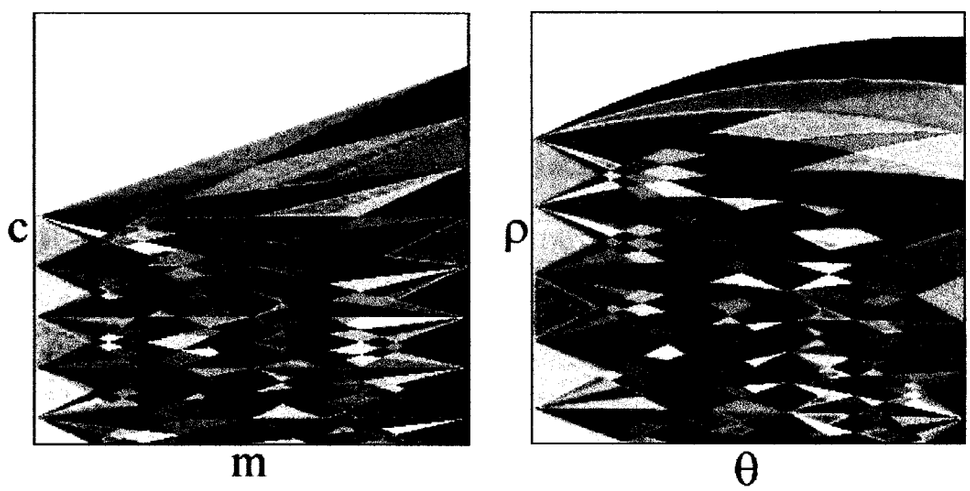


Figura 4.6 – Um quadrante da figura anterior.

Diminuindo-se a resolução da imagem, o número de retas digitais possíveis também diminui, ocasionando um aumento nas regiões mostradas nas Figuras 4.5 e 4.6. Essas regiões, para uma imagem de 5x5 pixels, são mostradas na Figura 4.7, na qual se verifica que o número de retas digitais possíveis diminuiu muito.

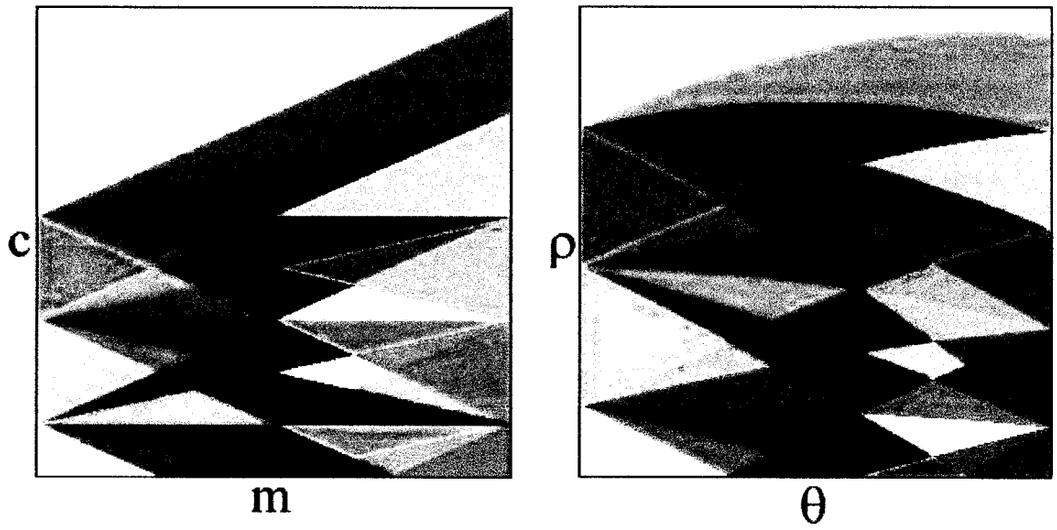


Figura 4.7 – Espaços de parâmetros para uma imagem com 5x5 "pixels".

Como podemos observar, o tamanho das regiões correspondentes a cada reta digital não é constante, ocorrendo uma grande variação no tamanho e forma destas. Essa característica torna mais difícil o mapeamento adequado da imagem para os espaços de parâmetros, pois ao traçarmos sobre estes espaços de parâmetros uma grade regular, com as intersecções representando os pontos onde a amostragem será feita, algumas das regiões serão amostradas mais de uma vez, enquanto outras não terão nenhuma intersecção sobre elas, indicando que não foram amostradas nenhuma vez.

A Figura 4.8 mostra ocorrências desses casos, sobrepondo uma grade regular sobre os espaços mostrados na Figura 4.7. Algumas regiões que não foram amostradas foram marcadas com um ponto vermelho, e exemplos de regiões amostradas mais de uma vez foram marcadas com um ponto amarelo.

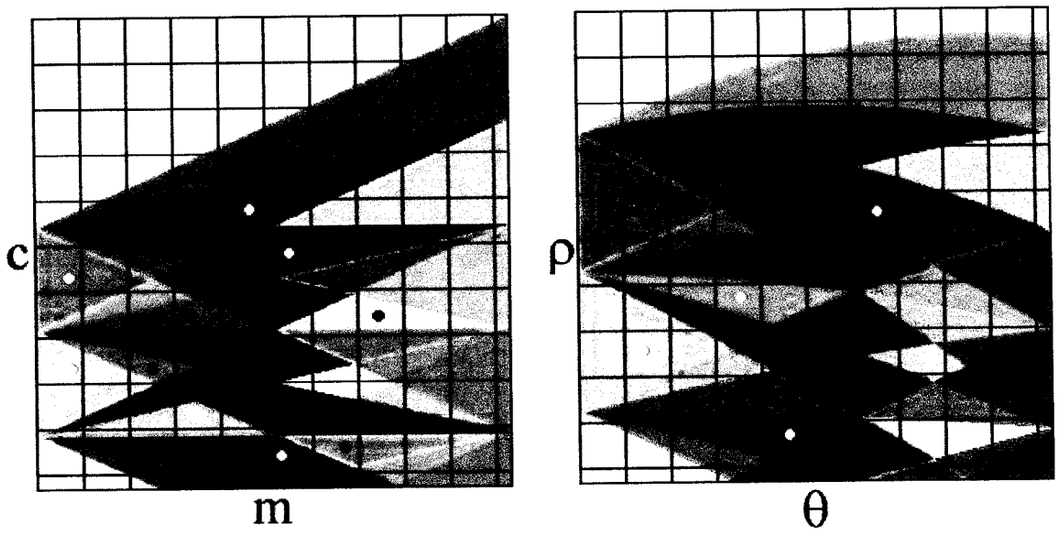


Figura 4.8 - Amostragem com grade regular. Exemplos de regiões não amostradas com pontos vermelhos, e de regiões amostradas mais de uma vez, com pontos amarelos.

5 – Transformada de Hough e Janelas Lineares

A Transformada de Hough [HOUGH 62] é, atualmente, um dos métodos mais amplamente utilizados para a detecção de retas em imagens. O método desenvolvido neste trabalho, baseado no sistema de visão das aranhas saltadoras, possui analogia com a Transformada de Hough, que será aqui apresentada.

5.1 – Fundamentos da Transformada de Hough

A Transformada de Hough, originalmente, realizava um mapeamento da imagem para o espaço de parâmetros *m-c* e, posteriormente, foi proposto seu mapeamento no espaço ρ–θ [DUDA 72], como descrito na seção 4, através das correspondências:

$$y = m.x+c;$$
$$\rho = x\cos(\theta) + y\,sen(\theta).$$

Essa transformada geralmente é aplicada em uma imagem contendo as bordas detectadas da cena a ser analisada. Para cada ponto da borda (x,y) da imagem, uma senóide é traçada no espaço de parâmetros. Porém, além da natureza discreta da imagem a ser transformada, o espaço de parâmetros agora também é discreto, e composto por uma matriz chamada de *Matriz Acumuladora*.

Traçar uma curva na matriz acumuladora equivale a incrementar de uma unidade o valor das células do acumulador pelas quais a curva passa. Como discutido na seção anterior, devido à natureza digital dos domínios analisados, o mapeamento depende também da largura de banda Δ, e cada ponto corresponde a uma banda senoidal de largura Δ [COSTA 95].

A Figura 5.1 mostra o mapeamento no acumulador de Hough, para uma reta digital.

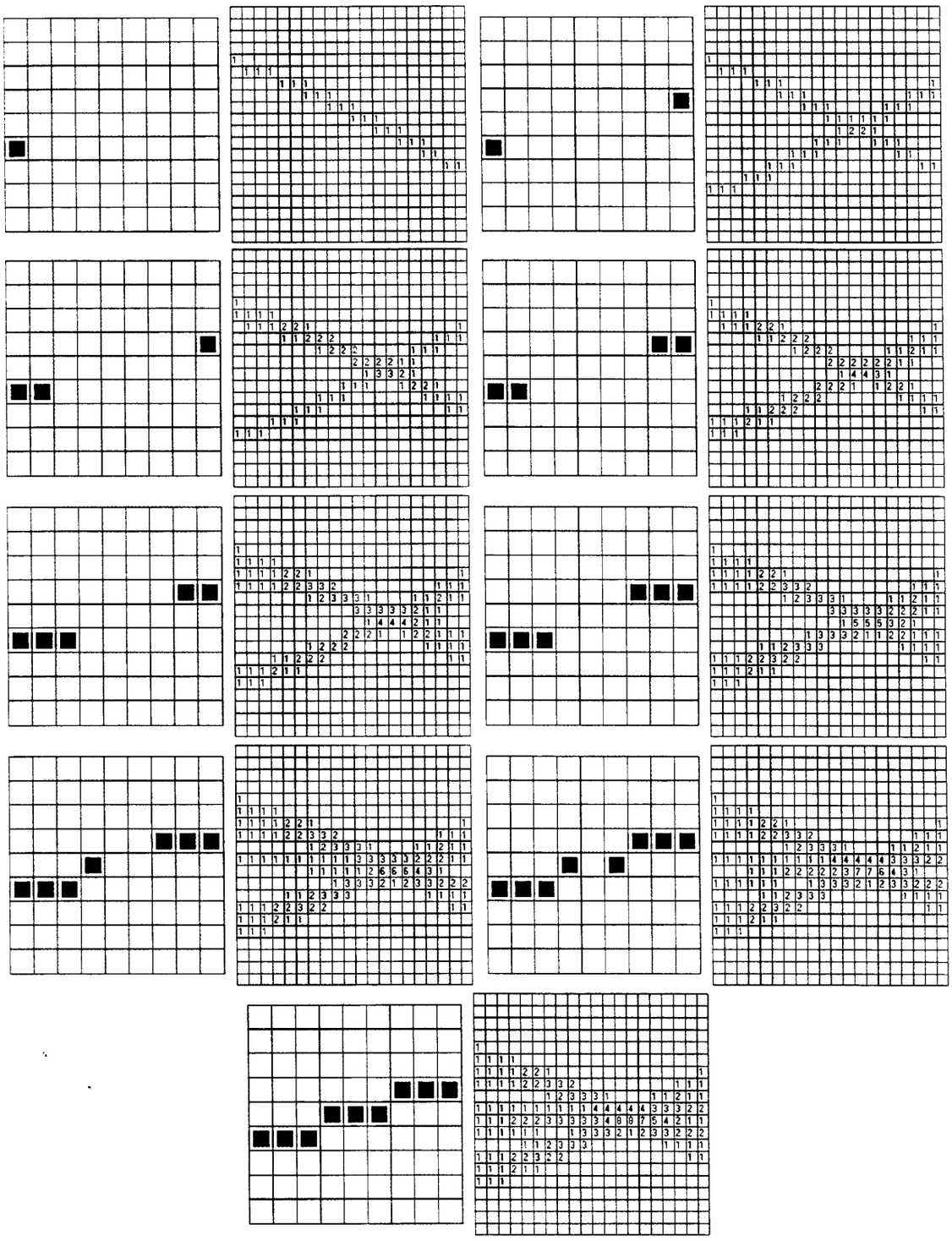


Figura 5.1 – Espaço imagem e respectivas Matrizes Acumuladoras de Hough

A Transformada de Hough evidencia algumas características da imagem, principalmente as retas, que são mapeadas em picos no espaço de parâmetros. Como para cada par $\rho-\theta$ existe apenas uma pequena região no acumulador de Hough, essa transformada não oferece informação sobre a conectividade das retas. A Figura 5.2 mostra a transformada de Hough de uma reta contínua, enquanto a Figura 5.3 mostra a transformada de uma reta descontínua.

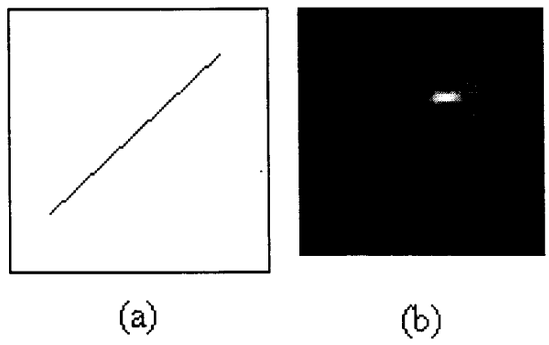


Figura 5.2 - Transformada de Hough de uma reta contínua.

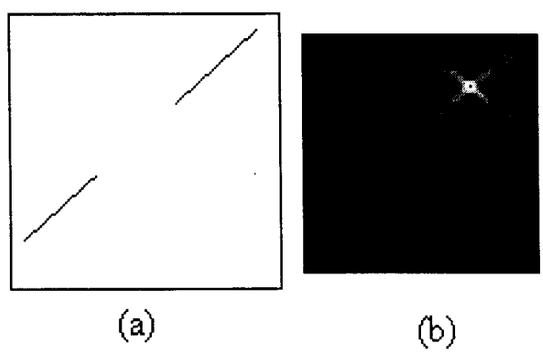
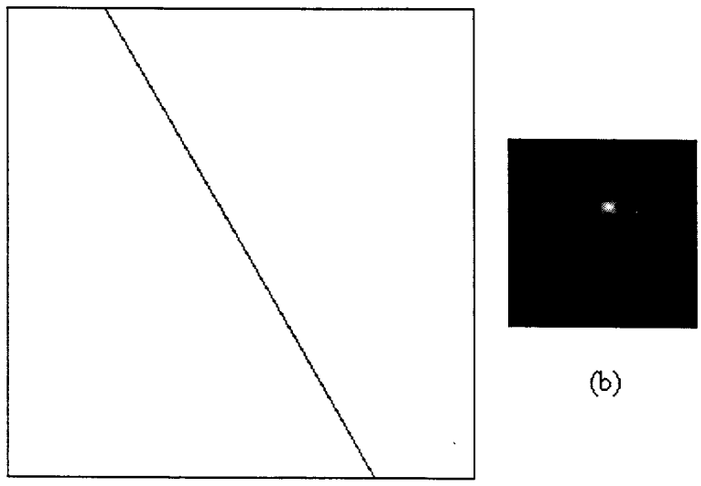


Figura 5.3 - Transformada de Hough de uma reta descontínua.

Apesar de não fornecer informações sobre a continuidade das retas, a Transformada de Hough tem em seu favor o fato de que é tolerante a ruídos na imagem, como mostram as Figuras 5.4 e 5.5. Na Figura 5.4, temos a transformada de Hough de

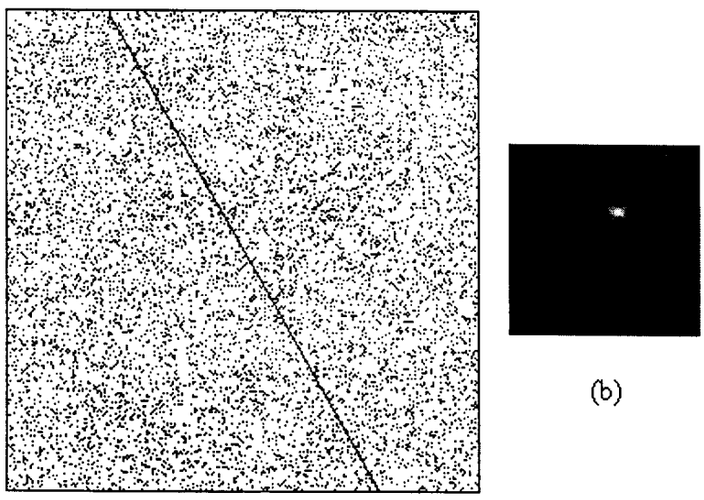
uma imagem sem ruído; e na Figura 5.5 a imagem é muito ruidosa, mas mesmo assim podemos identificar o pico resultante da reta nesta imagem.



(b)

(a)

Figura 5.4 - Transformada de Hough da região próxima a uma reta; imagem sem ruído.



(b)

(a)

Figura 5.5 - Transformada de Hough da região próxima a uma reta; imagem ruidosa.

O esforço computacional envolvido na execução da transformada de Hough é considerável. Porém, como esse mapeamento envolve um grande número de operações

independentes, a Transformada de Hough pode ser facilmente implementada em arquiteturas de processamento paralelo.

5.2 – Variações

Ao longo do tempo, diversas variações da Transformada de Hough foram propostas, cada uma visando solucionar alguns problemas ou tornar o método mais eficiente para algumas aplicações. Algumas dessas técnicas consistem em modificar a maneira como o acumulador de Hough é calculado, seja alterando o valor dos votos de acordo com a relevância dos pontos [STEPHENS 90], analisando a imagem por partes [SANDLER 90] ou aplicando formas de pré-processamento que diminuiriam o esforço computacional do cálculo do acumulador. A Transformada de Radon foi também utilizada para se calcular a Transformada de Hough [LEAVERS 88].

Outras propostas interessantes, utilizando análise com diversas escalas foram a Transformada de Hough Generalizada, para o reconhecimento de padrões genéricos [MERLIN 75]; a Transformada de Hough Adaptativa, que analisa a imagem através de uma janela de resolução crescente [ILLINGWORTH 87] e a Transformada de Hough Hierárquica [PRINCEN 90], que consiste na criação de uma estrutura piramidal através da aplicação da Transformada de Hough com diversas resoluções.

A Transformada de Hough Probabilística ou "Monte Carlo", proposta por Kiryati em 1991[KIRYATI 91] utiliza um subconjunto de pontos da imagem para o cálculo da Transformada de Hough, e diversos trabalhos propõem soluções para a determinação do número de elementos a serem selecionados nessas transformadas [YLÄ-JÄÄSKI 94, SHAKED 95], e para compensar problemas de amostragem [HORIKI 98].

Foi também proposto um método que combina a Transformada de Hough Probabilística com a Transformada de Hough Generalizada, para reconhecimento de padrões genéricos com maior rapidez, chamado de "Randomized Generalized Hough Transform" [FUNG 96]; além de um método que define o acumulador de Hough com

células Fuzzy [CHATZIS 96], e abordagens probabilísticas para esse tipo de construção do acumulador de Hough [CHATZIS 97].

5.3 – Amostragem

Uma questão importante na execução da transformada de Hough é a resolução do espaço de Hough, ou acumulador. A escolha de quantizações inadequadas pode resultar em problemas de subamostragem ou superamostragem.

De um modo geral, quando a resolução do espaço de parâmetros é muito grande, ocorre a super-amostragem, que resulta na replicação dos picos no espaço de Hough, levando à detecção de múltiplas retas onde na verdade há apenas uma.

Por outro lado, se a resolução adotada para o espaço de parâmetros for muito baixa, ocorre a subamostragem, resultando na impossibilidade de diferenciação dos votos dos pontos que constituem retas próximas, pois estes são computados na mesma célula do acumulador.

Devido às irregularidades das regiões ocupadas por retas, como discutido na Seção 4.3, dificilmente um esquema usual de quantização conseguirá amostrar eficientemente todo o espaço de parâmetros. Deve-se, no entanto, ter certos cuidados na escolha da resolução da matriz acumuladora de Hough, para que os problemas de amostragem sejam minimizados. Neste trabalho, a resolução da matriz acumuladora será duas vezes maior que a resolução da imagem correspondente.

5.4 – Mapeamento por Janelas Lineares

O mapeamento por janelas lineares é um processo que possui grande analogia com o sistema de visão das aranhas saltadoras e com a transformada de Hough [COSTA 94]. Este método consiste em mapear a imagem em um espaço $\rho-\theta$, utilizando uma janela linear móvel, e para cada posição $\rho-\theta$ da janela, define-se um valor para o espaço de parâmetros baseado no número de elementos de borda alinhados com essa janela.

Utilizando-se o mapeamento por janelas lineares, o mapeamento não é mais feito de pontos em senóides, mas sim de retas em pontos. Isso permite a construção do espaço de parâmetros $\rho-\theta$ ponto a ponto, fato que será de grande valor. A movimentação da janela linear pode ser feita com maior ou menor resolução, correspondendo à resolução do espaço de parâmetros adequado para o método.

Neste procedimento, assim como nos descritos anteriormente, a natureza discreta da imagem e do espaço de parâmetros deve ser considerada. Portanto, para se calcular o valor de um ponto no espaço de parâmetros, deve-se passar pela imagem uma janela linear de largura tal que os pontos dentro desta janela sejam mapeados em senóides que passam pelo ponto analisado $\rho-\theta$ do espaço de parâmetros.

5.5 – Comparação

Os resultados obtidos através da transformada de Hough de uma imagem são idênticos aos resultados da varredura da mesma imagem com a utilização de janelas lineares. A Figura 5.7 mostra o espaço de parâmetros ρ-θ para o resultado das duas operações, estando o resultado da transformada de Hough em (a) e o da varredura com janelas lineares em (b).

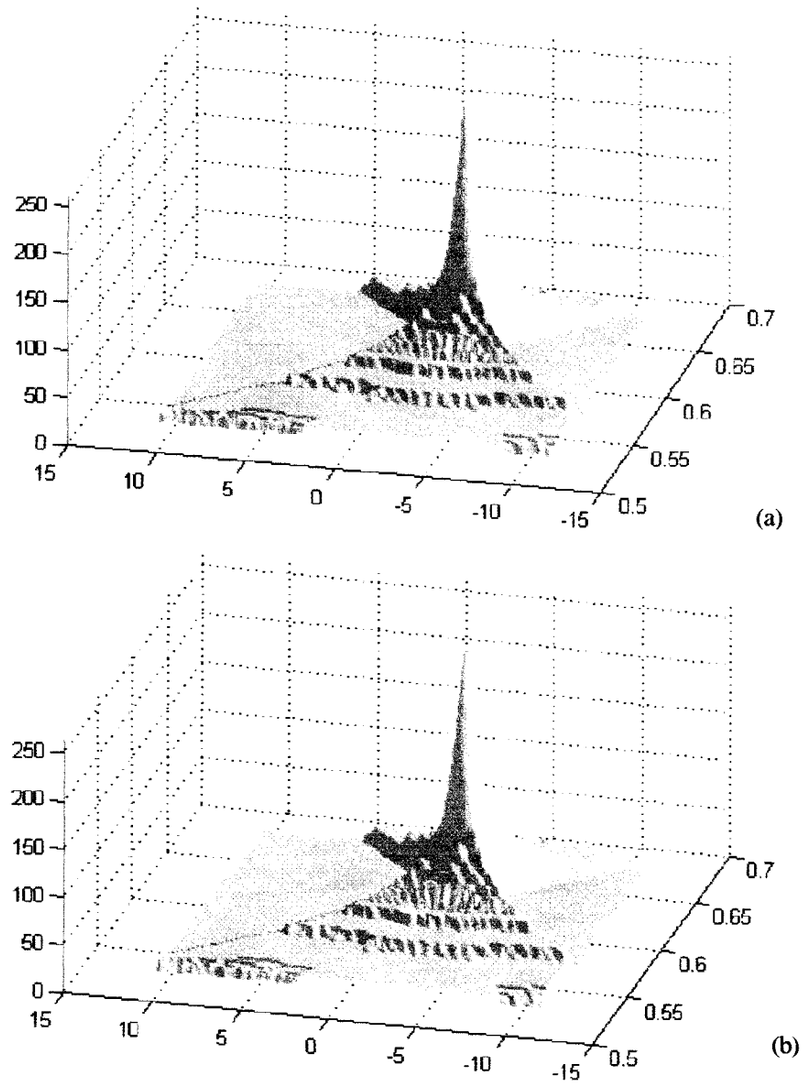


Figura 5.7 – Espaço de parâmetros para a transformada de Hough (a) e para a varredura com janelas lineares (b).

Podemos notar que as Figuras 5.7 (a) e 5.7 (b) são iguais, comprovandoa equivalência entre as transformações. Com relação ao tempo de processamento, a transformada de Hough pode ser calculada mais rapidamente que a varredura por janelas lineares, na maioria dos casos. As características que deixam a transformada de Hough mais rápida são: o fato desta só processar os pontos da imagem que representam elementos de borda, enquanto a varredura com janelas lineares precisa calcular o valor para cada ponto do espaço de parâmetros; e mesmo quando a imagem analisada tem muitos elementos de borda, a transformada de Hough faz uma varredura na imagem, enquanto a janela linear faz uma varredura no espaço de parâmetros, que geralmente tem maior resolução.

Apesar do cálculo do espaço de parâmetros utilizando as janelas lineares ser mais lento, essa abordagem permite o cálculo do valor de um ponto neste espaço sem que seja necessário o cálculo de todo o espaço, bastando para isso posicionar a janela linear na posição desejada e verificar a quantidade de pontos de borda encontrados. Essa característica nos permite criar um método que busca os pontos de máximo calculando apenas os pontos necessários para a execução de um algoritmo de otimização.

6 – Detecção de Retas por Otimização

6.1 – Introdução

A detecção de retas através da Transformada de Hough é geralmente feita através do cálculo completo da matriz acumuladora, seguido de uma operação de "threshold" para isolar os pontos mais votados. Uma abordagem por otimização constituiria em utilizar um algoritmo de busca de máximos de funções para encontrar os picos da matriz acumuladora, que correspondem a retas na imagem. Porém, a abordagem por otimização tem sido completamente ignorada na literatura especializada, sendo aqui discutida pela primeira vez.

Utilizando o mapeamento por janelas lineares, podemos calcular o valor de um ponto no espaço de parâmetros $\rho-\theta$ sem ter que construir todo este espaço, pois o valor que procuramos pode ser obtido com o posicionamento da janela linear na posição desejada. Assim, podemos utilizar um algoritmo de otimização, e somente os pontos necessários para esse algoritmo serão calculados no espaço de parâmetros, o que diminuirá o custo computacional da busca por retas, tornando-a mais rápida.

6.2 – "Nelder and Mead's Downhill Simplex"

Neste trabalho, o método de otimização escolhido foi o apresentado por Nelder e Mead, em 1965 [Nelder 65]. Esse método é conhecido como Downhill Simplex, e consiste em "mover" um simplex sobre a superfície de busca segundo algumas regras, que fazem essa estrutura chegar até o extremo da função analisada.

No caso da maximização de uma função bidimensional, o simplex consiste em uma estrutura com três vértices. Uma configuração inicial é escolhida, e para esta calculamos os valores da função nos três vértices. Os vértices são então classificados em melhor, pior e médio; e escolhe-se a nova posição do pior vértice entre algumas possibilidades, mostradas na Figura 6.1; e de acordo com algumas regras.

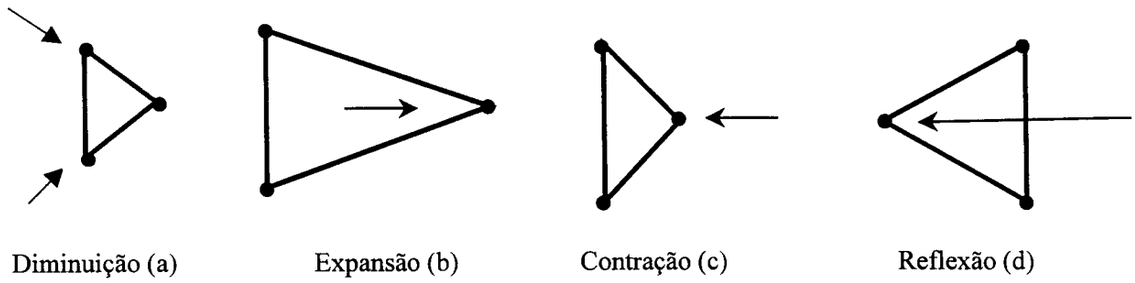


Figura 6.1 – Movimentos possíveis do Simplex

A primeira possibilidade a ser testada é a reflexão (d). Se o valor do vértice refletido for melhor que todos os outros, faz-se uma reflexão com expansão (b), e escolhe-se o melhor resultado entre (d) e (d)+(b). Se a reflexão não tiver um bom resultado, faz-se uma contração (c), e uma reflexão com contração (d)+(c), novamente escolhendo-se o melhor entre os dois resultados. Se nenhum desses resultados for melhor que a configuração inicial, a operação realizada é a diminuição (a), onde o pior vértice e o intermediário são movidos em direção ao melhor. A Figura 6.2 mostra um fluxograma que ilustra o funcionamento do método proposto por Nelder e Mead.

O método de Nelder e Mead converge rapidamente para um ponto de máximo próximo, mas é muito suscetível à presença de máximos locais, convergindo para estes. Além disso, patamares na superfície da função fazem com que o simplex estacione nestes, prejudicando também a busca. Assim, faz-se necessário um estudo da superfície onde este algoritmo está sendo lançado, para lidar adequadamente com a presença de características como máximos locais e patamares.

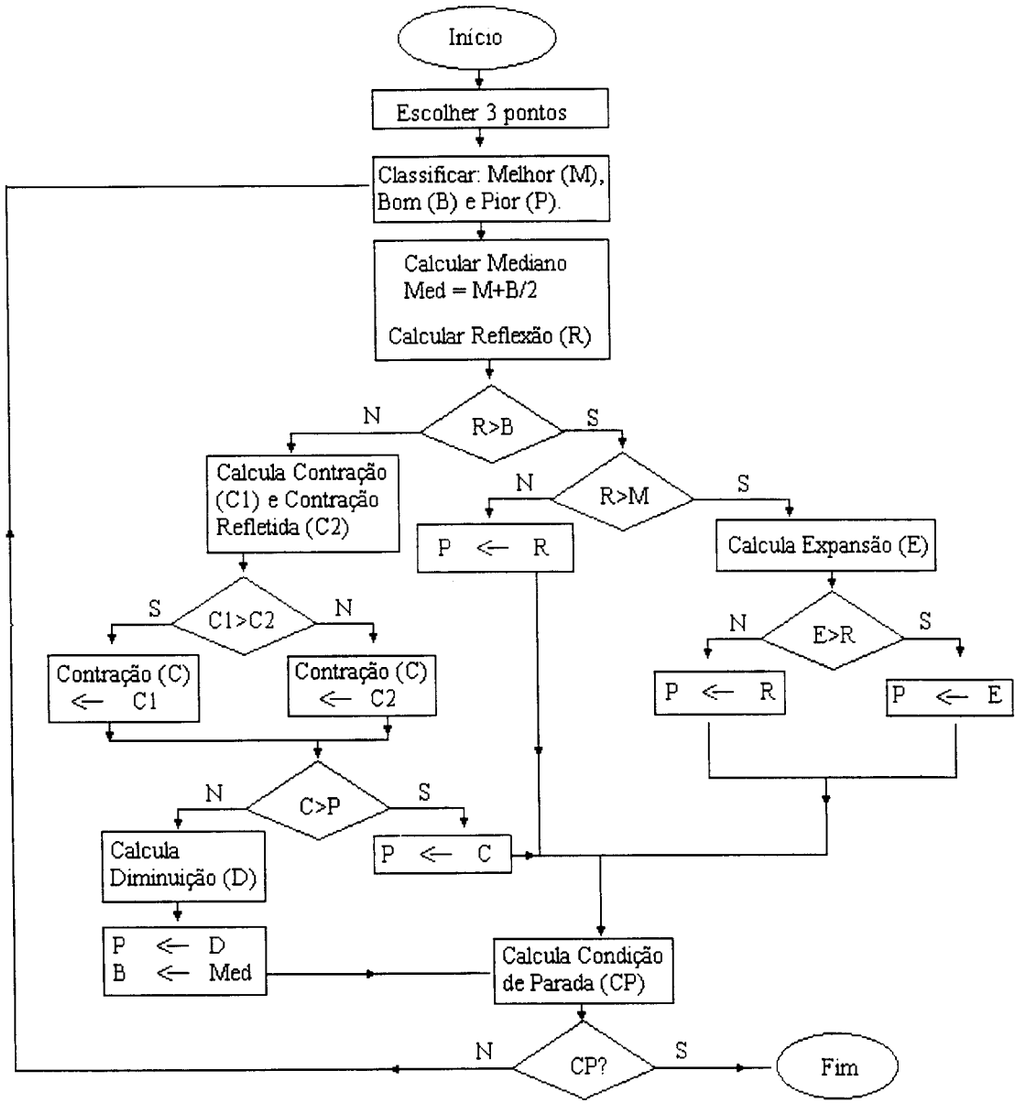


Figura 6.2 - Fluxograma do método "Downhill Simplex"

A configuração mais provável para os pontos iniciais do simplex é aquela onde os pontos estão ao lado de um pico, isto é, o pico procurado não se encontra entre os vértices do simplex. Desse modo, o passo mais provável é a reflexão com expansão, que levará o pior ponto do simplex a uma posição mais próxima ao pico.

A reflexão simples ocorre para impedir que o vértice movido passe pelo ponto de máximo e ainda se afaste mais. Como as reflexões (simples, com contração e com expansão) ocorrem por definição alternadamente em vértices diferentes, essas operações acabarão levando os vértices a uma configuração em que envolvem o pico desejado.

Tendo o "simplex" envolvido o pico com seus vértices, os movimentos que ocorrem são contrações e reduções, até que os vértices estejam próximos o suficiente do pico para que a condição de parada seja atendida. O critério de parada utilizado pelo método de Nelder e Mead estabelece uma tolerância, e calcula um valor f, através da fórmula:

$$f=20*abs(b-w)/(abs(b)+abs(w))$$

onde b é o valor do melhor vértice e w é o valor do pior vértice. Se f for menor que a tolerância estabelecida, o método termina.

6.3 – O espaço de parâmetros

A superfície gerada pela convolução de uma janela linear com uma imagem é muito semelhante à transformada de Hough dessa imagem, e cada célula desse espaço tem um valor inteiro correspondente à soma dos pixels contendo elementos de borda que passam pela banda digital correspondente.

Devido à natureza discreta desses espaços, existem neles patamares que dificultam a busca por máximos efetuada pelo algoritmo de Nelder e Mead. Um exemplo desses patamares é mostrado na Figura 6.3.

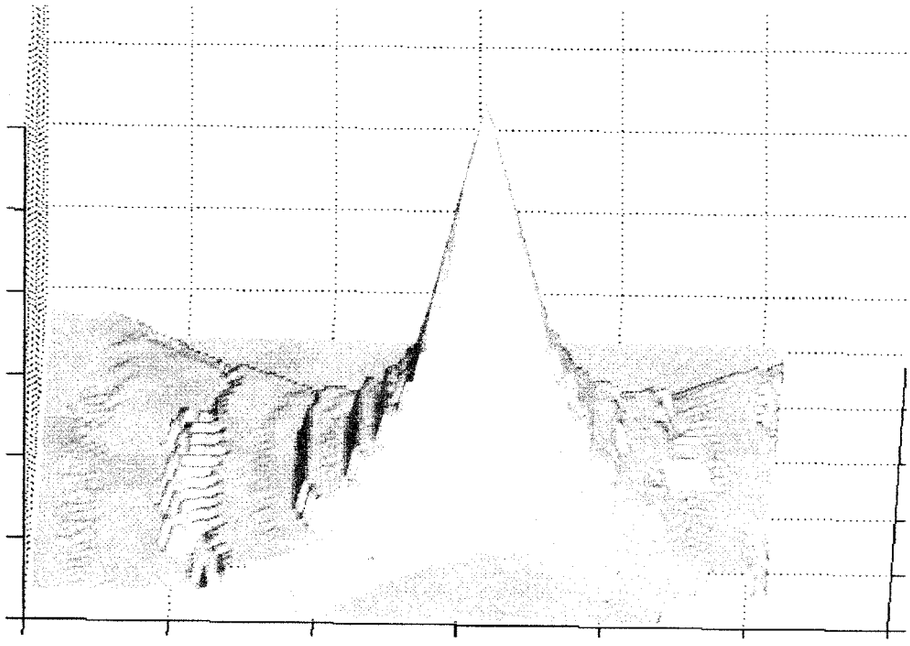


Figura 6.3 – Patamares no espaço de parâmetros.

O critério de parada utilizado pelo algoritmo de Nelder e Mead é calculado utilizando a diferença entre os valores dos vértices do "simplex". Desse modo, quando a diferença entre os três valores for menor que uma certa tolerância, o algoritmo termina. Se há a presença de patamares no espaço de parâmetros em questão, quanto maiores estes forem, maior é a chance dos vértices do simplex ficarem nivelados antes de convergirem para o máximo.

6.4 – Largura das Janelas Lineares

A largura da janela linear utilizada para a análise da imagem tem grande influência no espaço de parâmetros resultante. Apesar da perda de precisão nas informações quanto à localização das retas, ao aumentarmos a largura das janelas lineares, ocorre uma suavização da superfície resultante no espaço de parâmetros.

Essa suavização, mostrada na Figura 6.4, melhora o desempenho do algoritmo utilizado na busca de pontos de máximo, pois reduz o tamanho dos patamares que o prejudicavam. A parte superior (a) da Figura 6.4 mostra um pico resultante da parametrização através de uma janela linear de largura 1; a parte intermediária (b) mostra o mesmo pico, mapeado com uma janela linear de largura 2; e a parte inferior (c) é o resultado do mapeamento deste pico por uma janela de largura 3.

Como podemos observar nas Figuras 6.3 e 6.4, os patamares no espaço de parâmetros ocorrem ao redor dos picos. Realizando a busca com uma janela linear de largura maior, consegue-se a redução desses patamares mostrada na Figura 6.4 (b) e (c), e o ponto de máximo encontrado possui menor precisão, mas é um bom local para se iniciar a busca com uma janela linear mais estreita. Assim, a busca por pontos de máximo agora consegue contornar a presença de patamares, iniciando-se a busca com uma janela linear de maior largura, e diminuindo-se esta até se obter uma localização satisfatória do pico procurado.

Além disso, podemos também verificar na Figura 6.4 que o pico gerado pela parametrização com as janelas mais largas (b, c) abrangem uma área maior do espaço de parâmetros. Isso aumenta a área em que os pontos iniciais do simplex podem ser escolhidos de forma eficaz.

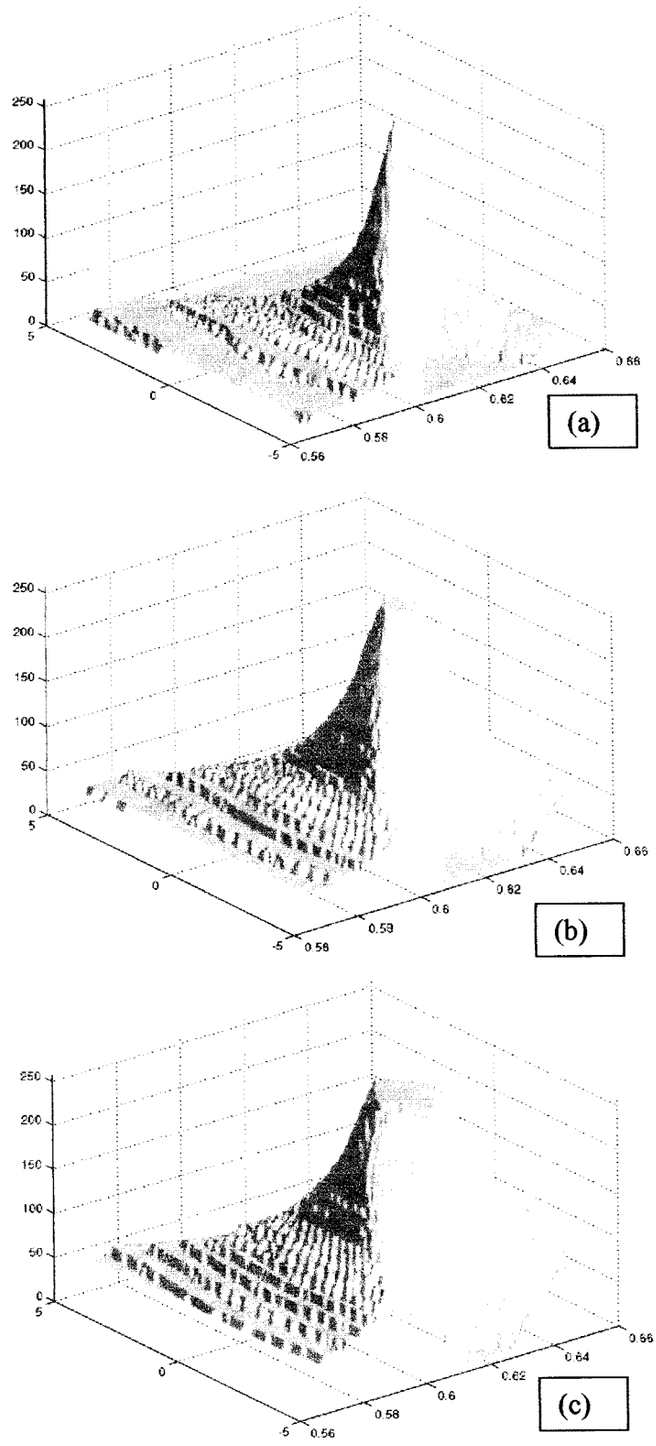


Figura 6.4 – Suavização do espaço de parâmetros. Varredura com janelas de largura 1(a), 2(b) e 3(c).

6.5 – Máximos locais

A presença de máximos locais é outro fator que pode levar a resultados incorretos na busca por retas utilizando janelas lineares. Como dito anteriormente, o algoritmo de Nelder e Mead não contorna esses máximos locais, convergindo para eles se estiverem mais próximos que o máximo procurado.

A ocorrência de máximos locais pode ser devida a interferências entre contornos, que ocorre na matriz acumuladora de Hough, ou à existência de retas muito próximas, de modo que seus picos se interceptem. A Figura 6.5 mostra a ocorrência de máximos locais como resultado da parametrização de retas próximas.

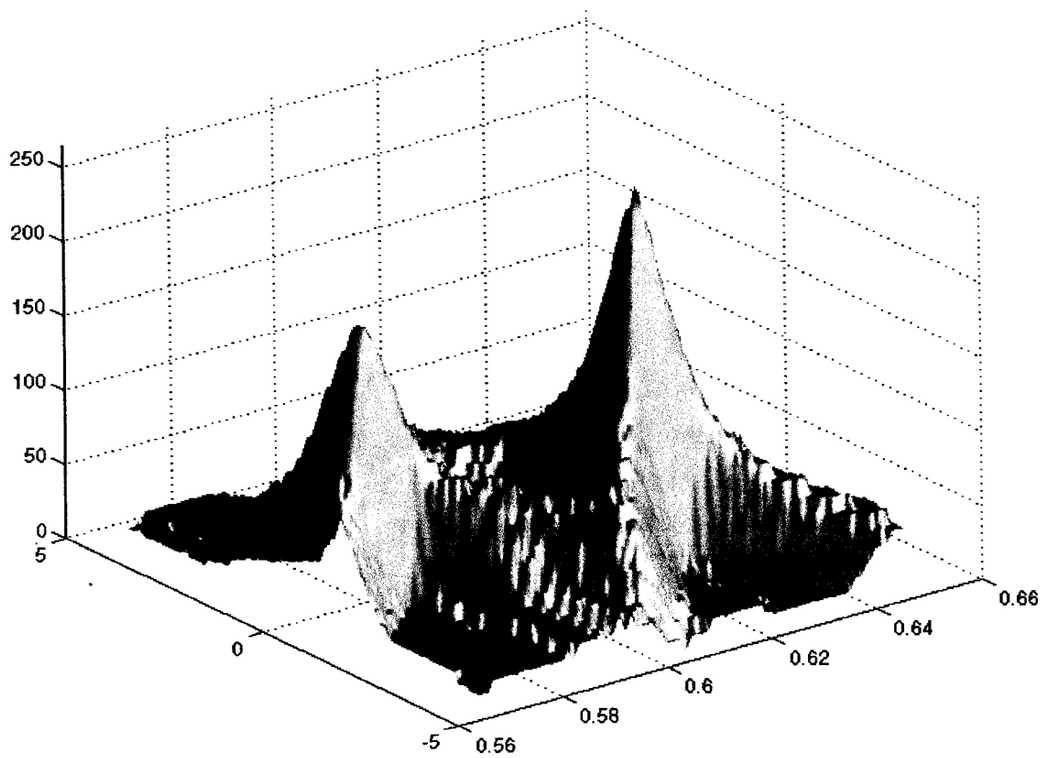
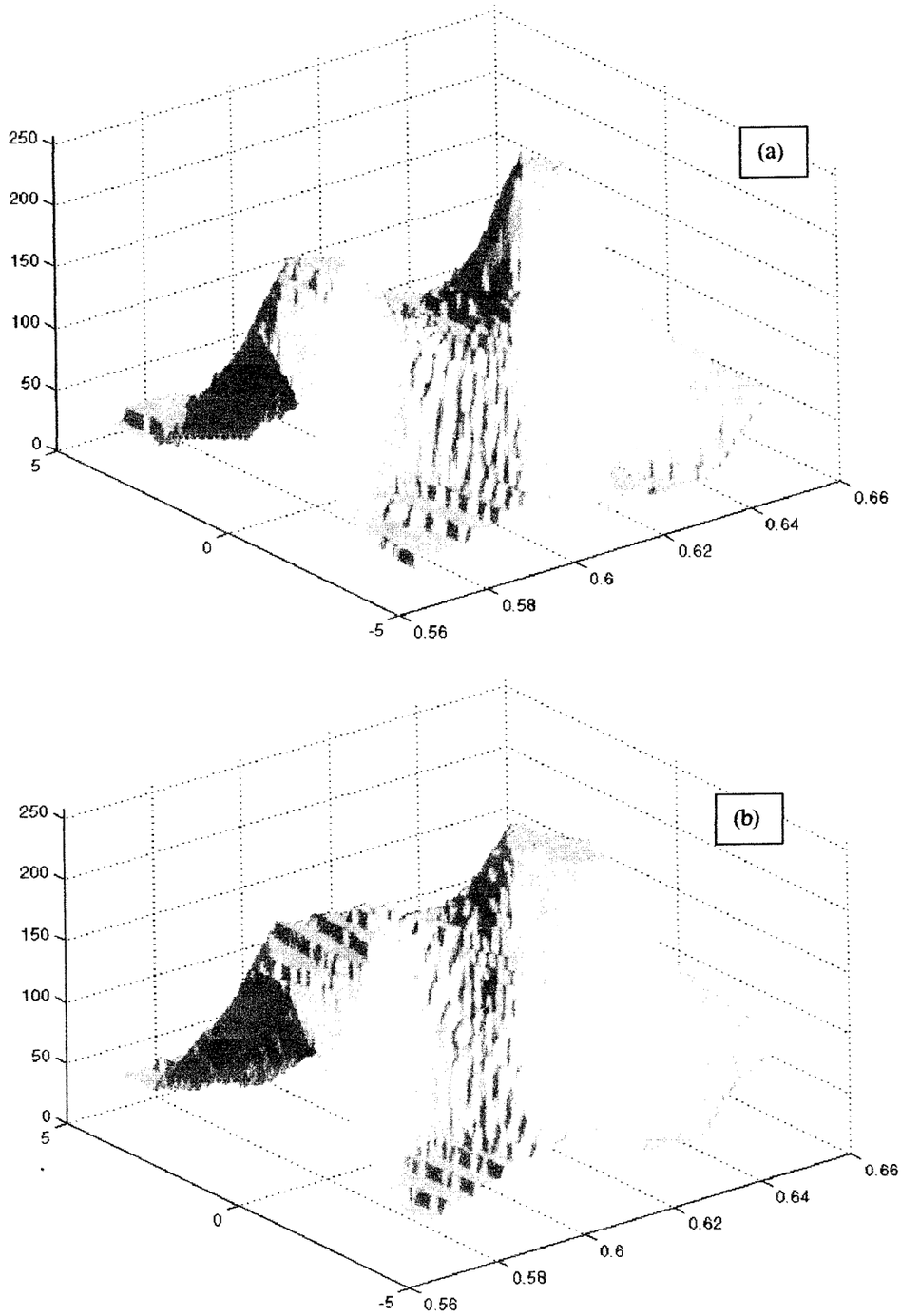


Figura 6.5 - Máximos locais no espaço de parâmetros.



Figua 6.6 - Efeito da largura das janelas lineares nos máximos locais.

A utilização de janelas lineares de maior largura reduz parcialmente o problema dos máximos locais, pois a suavização da superfície do espaço de parâmetros pode resultar no atenuamento, ou até mesmo no desaparecimento de um máximo local. Porém, para se contornar eficientemente o problema dos máximos locais, é necessária a utilização de um algoritmo que consiga encontrar o pico procurado sem parar nos máximos locais.

O efeito da varredura com janelas lineares de diferentes larguras, para a situação mostrada na Figura 6.5 está mostrado na Figura 6.6, para janelas lineares de larguras 2 (a) e 3 (b).

6.6 – Simulated Annealing

O método de "Simulated Annealing" foi inicialmente proposto por Kirkpatrick, em 1983 [KIRKPATRICK 83], na área de minimização combinatorial, onde a função de custo é discreta. Esse método é capaz de sair de mínimos locais, e apesar de não poder garantir que o mínimo global será encontrado, ele apresenta bons resultados em encontrar soluções ótimas ou quase ótimas mesmo quando muitas (centenas) variáveis estão envolvidas. As primeiras e mais utilizadas aplicações de "Simulated Annealing" até hoje são em arquitetura de computadores e na disposição de elementos e conexões em circuitos impressos [VECCHI 83].

Uma característica do método de "Simulated Annealing" é sua capacidade de primeiramente seguir o comportamento "geral" de uma função, ignorando os mínimos locais até uma região de energia mais baixa, e então realizar uma análise mais refinada, convergindo para o mínimo desta região [CORANA 87].

O método de "Simulated Annealing" funciona de maneira análoga ao resfriamento de materiais. O nome "Simulated Annealing" pode ser traduzido como "Recozimento Simulado", onde a palavra recozimento se refere ao resfriamento lento de materiais

visando aliviar tensões internas neste, permitindo que as partículas se acomodem lentamente em um estado de menor energia interna, formando, por exemplo, estruturas cristalinas ao invés de amorfas.

Analogamente, um algoritmo que permite apenas iterações que levem em direção ao mínimo procurado pode ficar preso em um mínimo local; mas se esta busca for realizada permitindo alguns movimentos que se afastem de mínimos locais, feitos em quantidades controladas, então o mínimo global pode ser encontrado.

A Figura 6.7 mostra uma situação em que um passo não ótimo é necessário para se encontrar o mínimo global. Iniciando a busca no ponto A, um algoritmo que só aceita configurações de menor energia convergirá para o mínimo local marcado pelo ponto B. Por outro lado, se configurações de maior energia forem aceitas no início da busca, existe uma chance de se chegar ao ponto C, de onde o algoritmo pode alcançar o mínimo global D; ou mesmo de se pular diretamente para próximo do ponto D. Se as configurações de maior energia forem aceitas menos freqüentemente com o tempo, o algoritmo não sairá do vale contendo o ponto D.

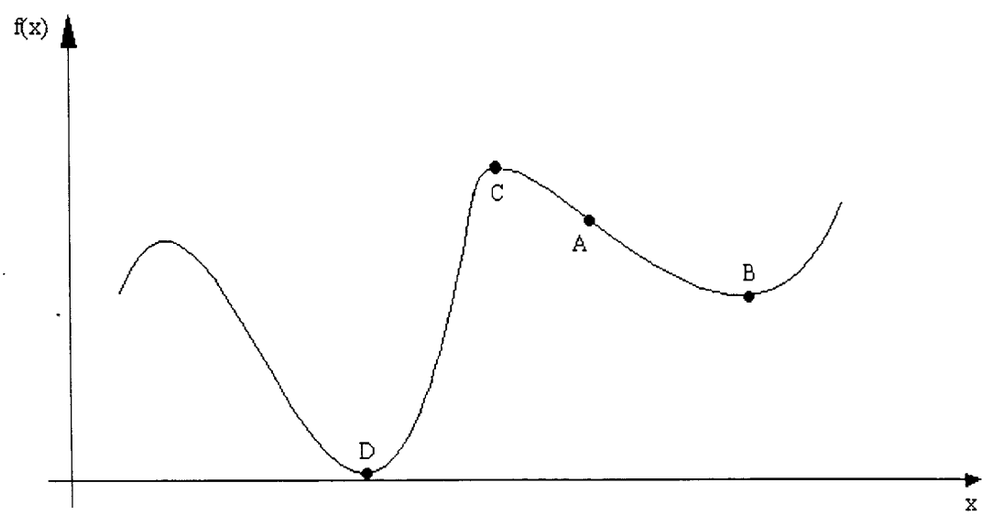


Figura 6.7 - Encontrando o mínimo global.

Devido à sua analogia com o resfriamento de materiais, o parâmetro que controla os passos "não ótimos" que o algoritmo permitirá é chamado de Temperatura. O algoritmo inicia a busca a uma temperatura mais elevada, permitindo maiores flutuações da energia em uma direção não ótima.

Com a redução gradual da temperatura, esses movimentos que aumentam a energia do sistema passam a ser mais freqüentemente rejeitados. A probabilidade de aceitar um movimento que aumenta a energia uma quantidade Δf é da forma:

$$P(\Delta f) = exp(-\Delta f/T)$$

Assim, quanto menor a temperatura (T), menores os aumentos de energia que serão aceitos. Esta fórmula foi proposta por Metropolis et al., em 1953 [METROPOLIS 53], para problemas de mecânica estatística. A Figura 6.8 mostra a probabilidade (área cinzenta) de se dar um passo para uma configuração de maior energia, para duas temperaturas diferentes.

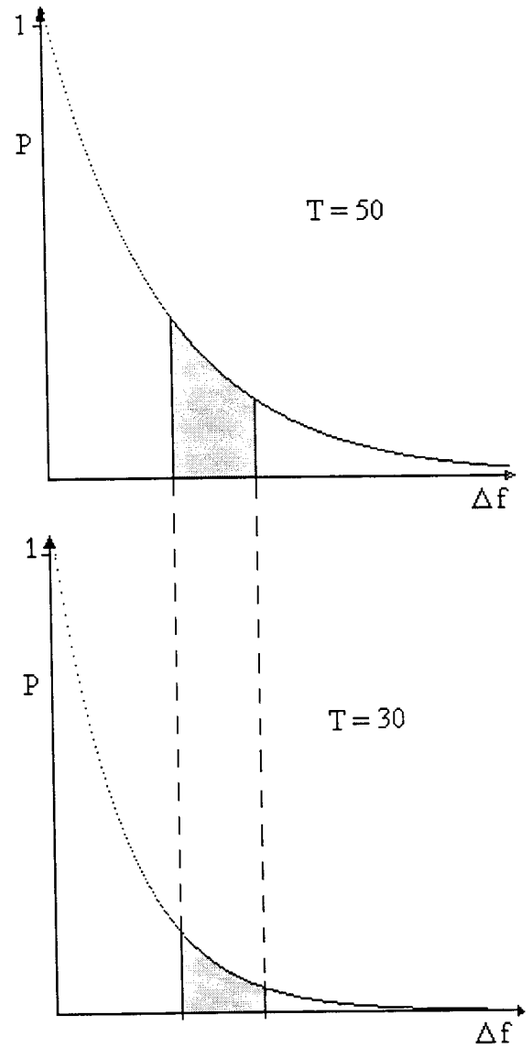


Figura 6.8 - Probabilidade de um passo que aumente a energia, em função da variação de energia.

O funcionamento adequado de um método de "Simulated Annealing" depende do que chamamos "Estratégia de Annealing". A Estratégia de "annealing" deve ser adequada para cada problema específico. A definição da estratégia de "annealing" consiste em designar valores para diversos parâmetros, que controlarão o processo de minimização.

Os parâmetros de uma estratégia convencional de "annealing" são:

- Temperatura Inicial: determina a probabilidade inicial de aceitação de configurações que aumentem a energia do sistema.
- Máximo de Iterações por Temperatura: nas estratégias de "annealing", a temperatura é abaixada após um certo número de passos aceitos em uma dada temperatura, ou quando o número máximo de iterações é atingido.
- Fator de Resfriamento: a temperatura pode ser abaixada linear ou exponencialmente, rápida ou lentamente, de acordo com o que define este parâmetro.

A Figura 6.9 mostra um fluxograma de uma estratégia de "annealing", utilizando os parâmetros citados acima.

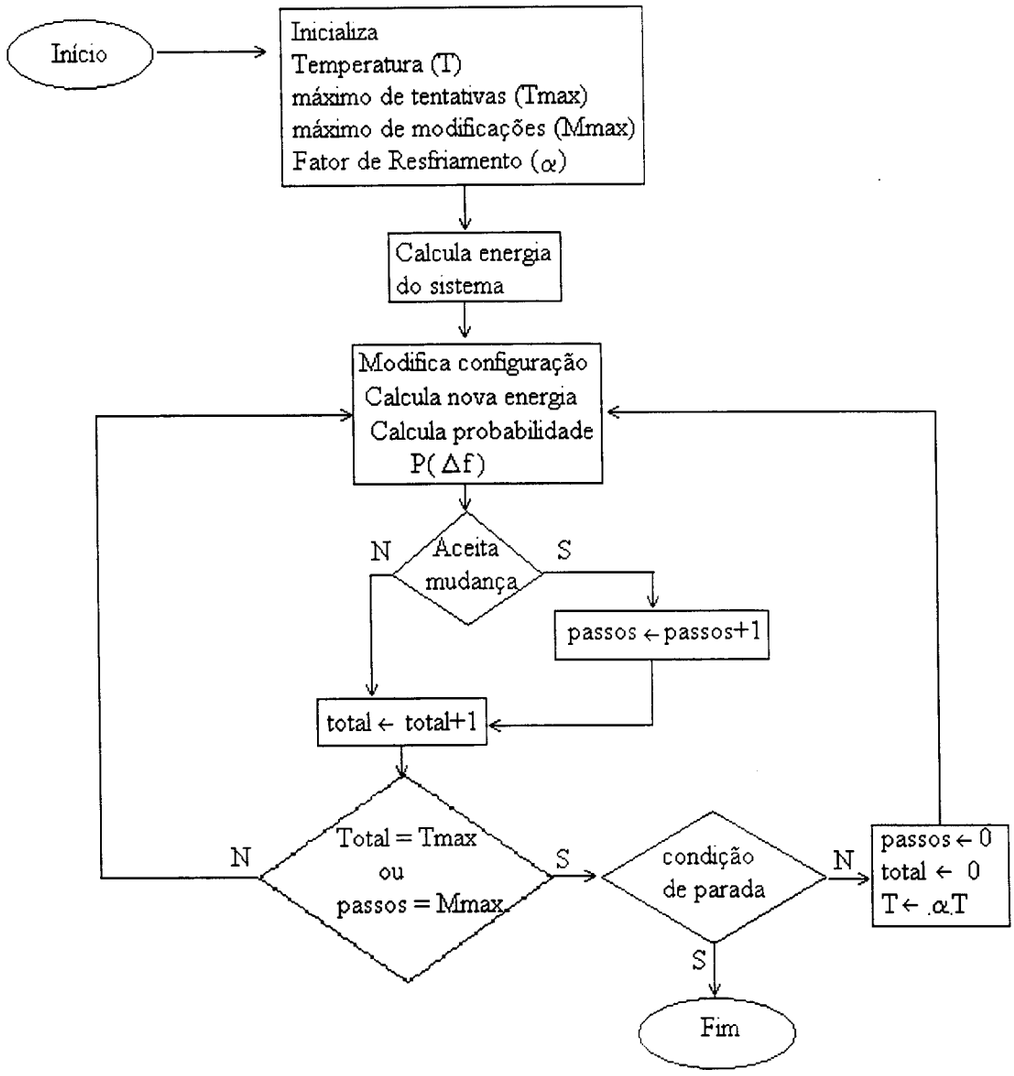


Figura 6.9 - Fluxograma de estratégia de "simulated annealing".

Diversas variações foram propostas para o método de "Simulated Annealing". Em 1987, Corana et al. propuseram uma variação de "Simulated Annealing" para ser utilizada em funções contínuas [CORANA 87]. Este método incorpora um vetor de tamanho pré-definido e variável com a temperatura para efetuar os "passos" no domínio da função. Vanderbilt, em 1984, também propôs uma maneira de implementar estratégias de "annealing" em domínios contínuos [VANDERBILT 84].

Uma estratégia de "annealing" pode ser incorporada ao algoritmo "simplex" de Nelder e Mead, permitindo uma busca por mínimos eficiente na presença de mínimos locais. Esse método foi apresentado por Press et al. em 1992 [PRESS 92], e consiste em incorporar flutuações aos vértices do "simplex", flutuações cujas magnitudes são controladas por um parâmetro "Temperatura".

O Algoritmo de "Downhill Simplex" com a incorporação de "Simulated Annealing" funciona de maneira muito semelhante ao método original proposto por Nelder e Mead. Após a escolha dos pontos iniciais para a realização da busca, são calculados os valores da função nestes pontos e valores das flutuações a serem adicionadas a eles. Estas flutuações são da forma:

$f'(x) = f(x) + K.T.log(z)$, onde z é um número aleatório entre 0 e 1.

Como resultado das flutuações, os valores dos vértices são "embaralhados", e a próxima fase do algoritmo, a de classificação desses vértices e escolha do pior para ser movimentado, é feita baseando-se nos valores que sofreram alterações proporcionais ao parâmetro temperatura. Desse modo, uma iteração para uma configuração de maior energia pode ocorrer quando o algoritmo escolhe um vértice de menor energia real, que parece ser o de maior energia devido às flutuações adicionadas.

Da mesma forma, ao se calcular os prováveis vértices para substituição do pior vértice (reflexão, expansão, etc.), estes também tem uma flutuação adicionada ao seu valor original. A Figura 6.10 mostra um fluxograma do algoritmo resultante da incorporação de "Simulated Annealing" ao método "downhill simplex". Os valores seguidos de um " * " são os valores com flutuações adicionadas.

Além da estrutura mostrada no fluxograma da Figura 6.10, o algoritmo precisa também de rotinas para o controle da estratégia de "annealing", ou seja, para abaixar a temperatura periodicamente, verificando o número de iterações realizadas para cada temperatura.

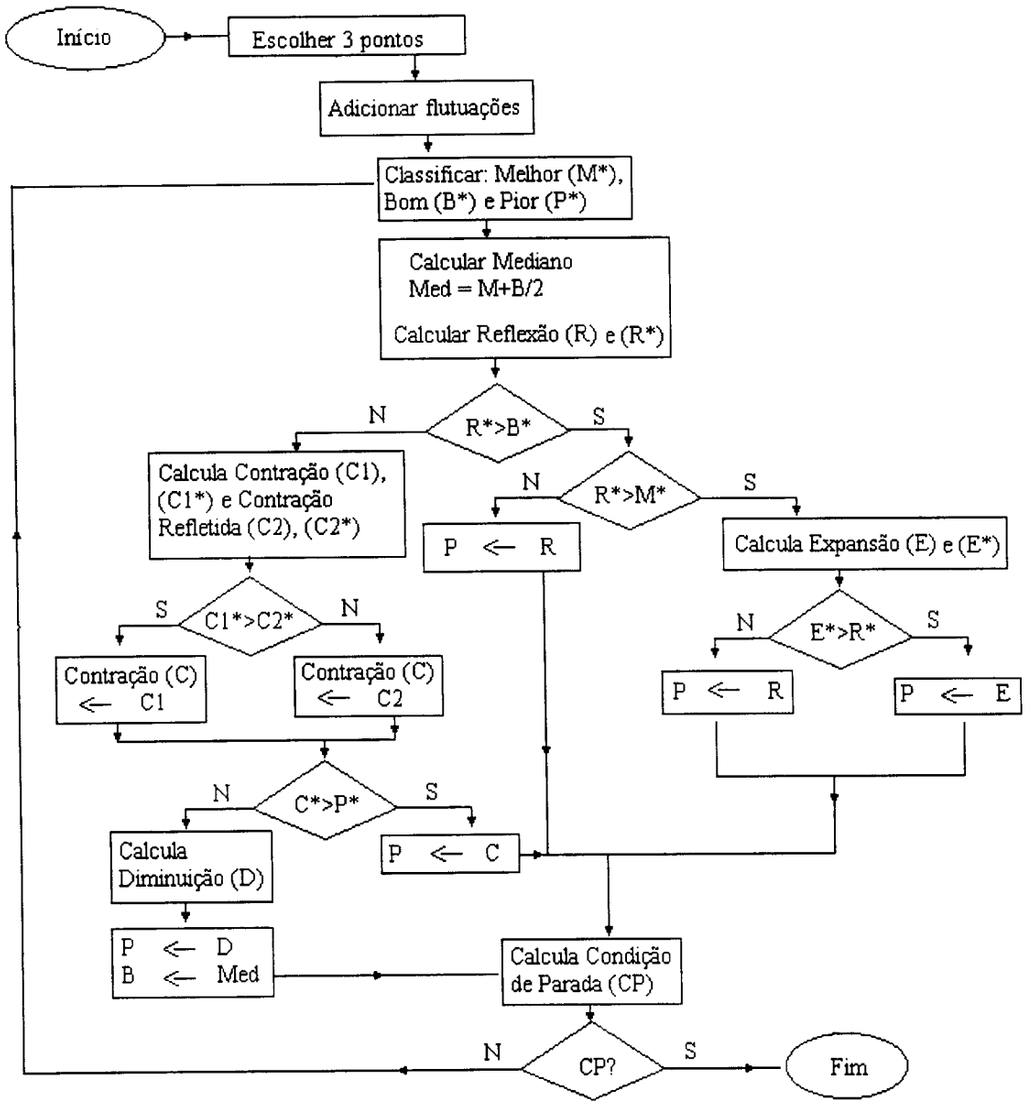


Figura 6.10 - Fluxograma do algoritmo "Downhill Simplex" com "Simulated Annealing".

Além da utilização de "Simulated Annealing", ao algoritmo "downhill simplex" foi incorporada a largura adaptativa da janela linear, para diminuir os patamares na superfície de parâmetros.

O algoritmo resultante faz uma busca por máximos utilizando inicialmente "simulated annealing" e uma janela linear mais larga, e reinicia a busca com uma janela mais estreita no ponto onde a busca inicial terminou; e assim sucessivamente até que seja realizada uma busca com janela de largura mínima.

Como o ponto de início da segunda busca já está mais próximo ao máximo procurado, a temperatura inicial utilizada para a estratégia de "annealing" nesta busca é menor, reduzindo a quantidade de movimentos não ótimos.

6.7 – Complexidade do espaço de parâmetros

O desempenho de um algoritmo com "Simulated Annealing" depende fortemente da utilização de um conjunto de parâmetros adequado. Assim, diversos testes, com parâmetros diferentes, precisam ser feitos para determinar aquele que resulta em uma estratégia que seja mais eficiente para o problema em questão.

No algoritmo desenvolvido para detecção de retas, diversos parâmetros estão envolvidos, sendo estes não só de "annealing", mas também referentes à largura das janelas lineares e ao critério de parada do algoritmo "downhill simplex".

Os parâmetros envolvidos no algoritmo de detecção de retas são:

- Temperatura inicial;
- Número de iterações por temperatura;
- Fator de resfriamento;
- Largura inicial das janelas lineares;
- Critério de parada do simplex.

A busca por uma configuração ótima neste espaço multidimensional de parâmetros é complicada de se realizar, e buscas preliminares mostraram que este espaço é fortemente não-linear, dificultando-a ainda mais.

Como existem muitas possibilidades para conjuntos de parâmetros, e o procedimento para avaliação de cada uma dessas possibilidades é trabalhoso, surge a necessidade de utilizar um método para amostrar de maneira eficaz o espaço formado pelos parâmetros do algoritmo. A escolha desses parâmetros será então realizada via algoritmo genético.

6.8 – Algoritmos Genéticos

Algoritmos genéticos foram introduzidos por John Holland, de 1960 a 1975, na universidade de Michigan [HOLLAND 75]. Seu objetivo original era estudar o fenômeno da adaptação como ocorre na natureza, e extrair seus mecanismos para reproduzi-la em sistemas computacionais.

O nome "algoritmo genético" vem dessa tentativa de reproduzir computacionalmente os mecanismos da evolução natural, e muitos dos termos utilizados na implementação de um algoritmo genético também advêm da biologia.

6.8.1 - Evolução e Genética na Biologia

As menores partes funcionais constituindo os seres vivos são as células, e cada uma destas, para um dado ser vivo, contém um mesmo conjunto de *cromossomos*. *Cromossomos* são cadeias de DNA, onde as informações necessárias para a definição e formação do ser vivo estão contidas [RAVEN 96].

Cada cadeia de DNA pode ser dividida em *genes*. Um *gene* é uma porção do DNA responsável pela criação de uma proteína, e estas proteínas serão responsáveis pelas características desse ser vivo. Cada configuração possível de um gene resulta em uma característica diferente, e essas possibilidades são chamadas de genes *alelos*.

Em alguns seres, os cromossomos são arranjados em pares. Esses seres são chamados *diplóides*. Por outro lado, seres cujos cromossomos não estão dispostos em pares são chamados *haplóides*. Durante a reprodução de organismos diplóides, os cromossomos de um par podem trocar material genético, no processo conhecido como *crossover* ou recombinação. Esses cromossomos modificados se separam para formar gametas haplóides.

O conjunto do material genético de um organismo é chamado *genoma*. O conjunto das características físicas de um organismo, decorrentes de sua constituição genética, é chamado de *fenótipo*.

O conceito de *fitness* representa a probabilidade de um organismo sobreviver no meio em que vive, e se reproduzir, gerando descendentes.

6.8.2 - O Algoritmo Genético de Holland.

O Algoritmo genético de Holland consiste em, a partir de uma população de "indivíduos" representados por seqüências de bits (cromossomos), obter novas gerações desta, mais adaptadas ao "ambiente" representado por condições aplicadas a essas seqüências. Essa transição entre gerações utiliza um tipo de "seleção natural", e operações com os bits inspiradas geneticamente em mutação, crossover e inversão.

A seleção proposta por Holland baseia-se em um valor de "fitness" designado a cada indivíduo. "Fitness" representa o quanto cada indivíduo está adaptado ao ambiente, e a forma de calculá-lo varia de acordo com o que se busca na população.

O método de Holland dá a cada indivíduo uma chance de ser escolhido para "reprodução" proporcional ao seu fitness. Assim, indivíduos mais aptos têm maior probabilidade de gerar filhos.

Os operadores genéticos propostos por Holland são a mutação, o crossover e a inversão. A mutação consiste em escolher aleatoriamente uma posição da seqüência de bits e modificá-la, como mostra a Figura 6.11.

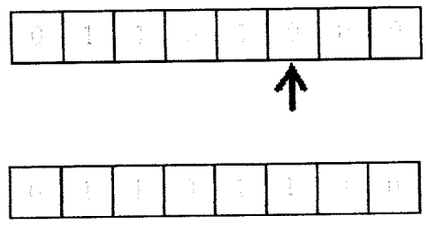


Figura 6.11 - Operador mutação.

O Crossover consiste em escolher um ponto dos cromossomos "pais", e trocar os bits localizados após este ponto entre esses cromossomos, como ilustra a Figura 6.12. Este operador tem grande analogia biológica, pois reproduz os resultados da recombinação genética em seres haplóides (com apenas um cromossomo de cada tipo), e foi uma das principais inovações do método proposto por Holland com relação às técnicas existentes na época.

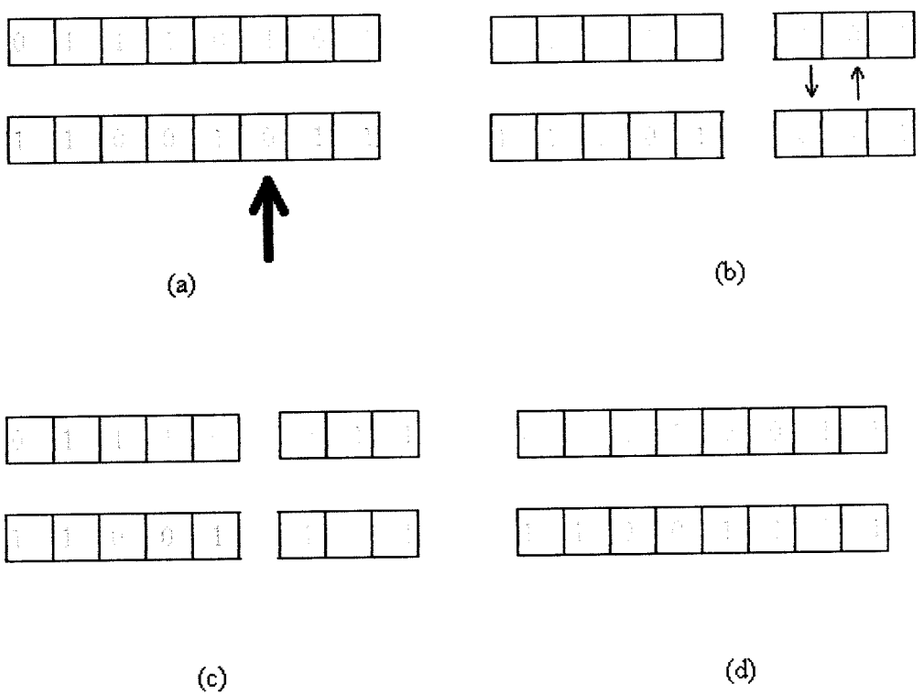


Figura 6.12 - Operador crossover.

Finalmente, o operador inversão funciona escolhendo dois pontos de um cromossomo, e invertendo a posição dos bits localizados entre os pontos escolhidos, como mostra a Figura 6.13.

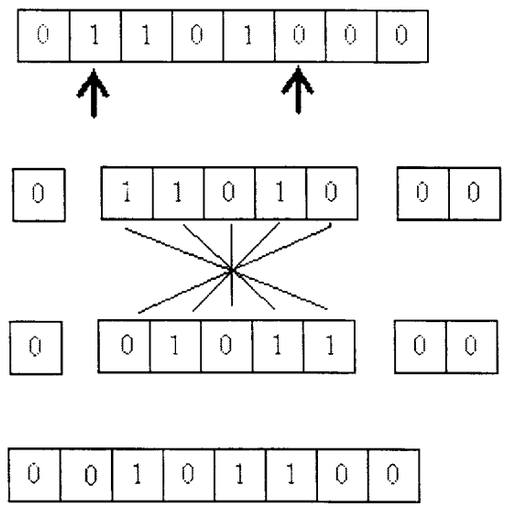


Figura 6.13 - Operador inversão.

O método proposto por Holland tornou-se a base para muitos outros avanços da chamada "computação evolutiva", que englobava todos os esforços para solucionar problemas ou otimizar parâmetros através de uma "evolução" de soluções candidatas.

6.8.3 - Elementos, Variações e Exemplo

Na área dos algoritmos genéticos, muitas propostas diferentes surgiram após o algoritmo proposto por Holland, mas todos têm alguns elementos básicos em comum.

O primeiro elemento característico é a presença de uma população de soluções codificadas de um modo tal que possam "evoluir", seja em seqüências de bits, em arranjos de símbolos que representam componentes, ou em seqüências de números reais.

Um parâmetro de avaliação de cada indivíduo (fitness), aliado a um método de seleção baseado nesse parâmetro, constitui outro elemento sempre presente nos algoritmos genéticos.

A recombinação, ou "crossover", também é encontrada em todos os algoritmos genéticos, pois é esta operação que garante uma diversificação orientada dos indivíduos ou soluções que formam sucessivas populações. Aliada ao "crossover", a operação de mutação é o último elemento que está sempre presente nos algoritmos genéticos. Esta operação permite que novas configurações sejam geradas para evitar que a população pare de evoluir.

Os algoritmos genéticos operam através da criação de novas gerações a partir das anteriores, e o fazem por um certo número de gerações, geralmente maior que 50. As operações de seleção, "crossover" e mutação ocorrem na geração de cada indivíduo de todas as gerações. A Figura 6.14 mostra o fluxograma básico de um algoritmo genético.

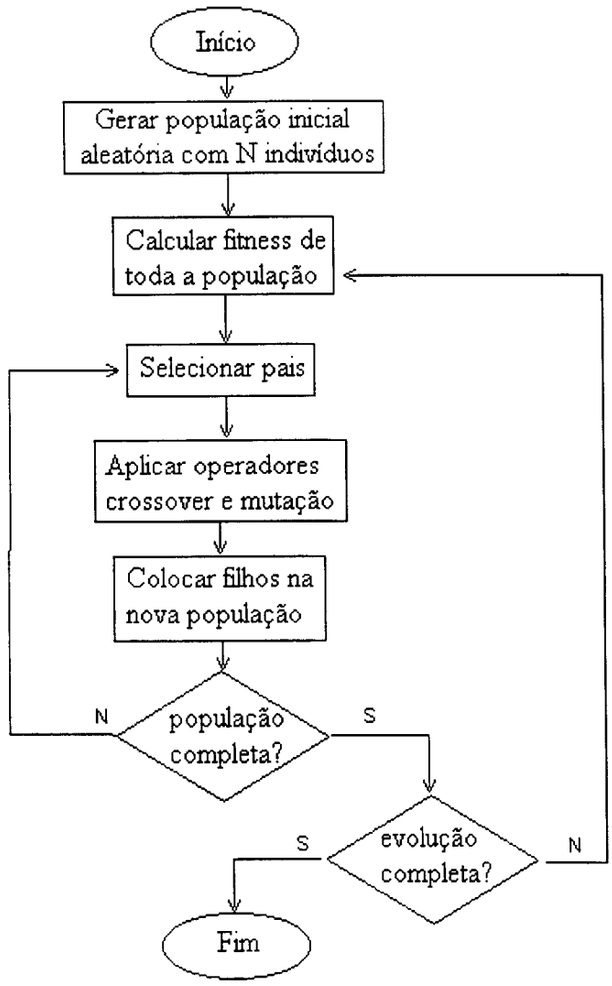


Figura 6.14 - Fluxograma de algoritmo genético básico.

A seleção de indivíduos para reprodução é o primeiro fator importante para o desempenho do algoritmo genético. Existem diversas maneiras de se amostrar a população de pais para criar uma nova geração, propostas por diversos pesquisadores.

O primeiro método de seleção, utilizado por Holland em seu algoritmo genético original, chama-se "roulette wheel" ou Seleção Proporcional ao Fitness [GOLDBERG 89]. Neste método, somam-se todos os "fitness" da população; sorteia-se um número aleatório variando de 0 até o valor somado, e percorre-se a população somando-se os

fitness dos indivíduos até atingir o valor sorteado. Isso é equivalente a utilizar uma roleta, com cada um dos indivíduos ocupando uma fatia proporcional a seu "fitness".

Como este método pode eventualmente escolher o pior indivíduo mais de uma vez (com probabilidade muito pequena, mas ainda assim existente), em 1987 James Baker propôs o método "Amostragem Estocástica Universal". Este consiste em, ao invés de rodar a "roleta" N vezes para uma população de N indivíduos, rodá-la apenas uma vez, mas com N ponteiros igualmente espaçados ao redor dela.

Foram propostos métodos de seleção que fossem menos dependentes das variâncias dos "fitness" dos indivíduos da população, como o "Sigma Scaling", onde a probabilidade de um indivíduo ser escolhido é função de seu "fitness", da média da população e do desvio padrão desta; e o "Seleção de Boltzmann", que diminui gradualmente a probabilidade de se escolher um indivíduo com "fitness" baixo no decorrer das gerações, de maneira muito similar ao "simulated annealing", discutido na seção 6.6.

Além desses métodos, outros mecanismos foram adicionados ao processo de seleção, o mais importante deles sendo o "elitismo". O elitismo consiste em manter alguns dos melhores indivíduos de uma população na próxima geração, impedindo que eles sejam perdidos por "crossovers" ou mutações.

A operação de "crossover" não teve muitas modificações durante a evolução dos algoritmos genéticos. Basicamente, o algoritmo utiliza um valor para a probabilidade de "crossover", e realiza esta operação de acordo com essa probabilidade.

O operador de "crossover" proposto por Holland pode ser chamado de "crossover simples". Um outro tipo de "crossover" que pode ser utilizado em um algoritmo genético é o "crossover duplo". Neste, dois pontos são escolhidos nos cromossomos, e troca-se entre eles a porção da cadeia contida entre esses pontos. Da mesma maneira, obteve-se o "crossover múltiplo", utilizando diversos pontos dos cromossomos.

Uma modificação maior do operador de "crossover", com o intuito de desfazer a influência das seqüências que têm grande possibilidade de serem movidas juntas, é o "crossover uniforme". Este tipo de "crossover" troca cada bit dos cromossomos entre eles, de acordo com uma probabilidade, geralmente entre 50 e 80% [MITCHELL 98].

Algoritmos genéticos podem também ser implementados utilizando indivíduos diplóides. Nesse caso, o "crossover" ocorre no par de cromossomos de um indivíduo, antes desse par se dividir, para que cada cromossomo se junte a um vindo de um par de cromossomos de outro indivíduo. A Figura 6.15 ilustra o "crossover" com indivíduos diplóides. Os pais (1) e (2) geram os filhos (3) e (4), com material genético diferente.

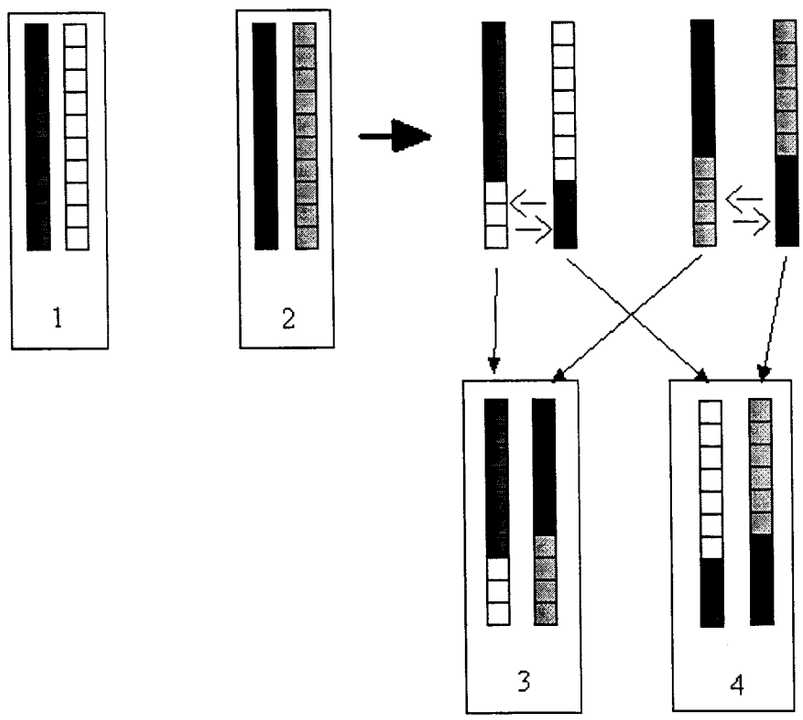


Figura 6.15 – "Crossover" em indivíduos diplóides.

Nos primeiros algoritmos de computação evolutiva, antes da apresentação do algoritmo genético de Holland, o único fator que introduzia mudanças nos indivíduos de

uma população era a mutação. Com a presença do operador "crossover", a probabilidade de mutação durante a evolução de uma população foi diminuída, pois o "crossover" introduz mudanças na população, e tem menor possibilidade de desfazer uma seqüência bem sucedida [MITCHELL 98].

Para exemplificar o funcionamento de um algoritmo genético, vamos utilizar uma população pequena, com N= 4 indivíduos, com tamanho l=10 bits cada um. A probabilidade de "crossover" será de p_c=0.7 e a probabilidade de mutação será p_m = 0.001.

A função de "fitness" utilizada será o número de zeros de cada indivíduo, e a seleção será feita com amostragem proporcional ao "fitness". A população inicial é:

Indivíduo	Cromossomo	Fitness
1	0010001110	6
2	1101100101	4
3	0011010101	5
4	0100111101	4

O resultado da seleção escolheu para pais primeiramente os indivíduos 4 e 1. Não haverá "crossover" nem mutação. Em seguida foram escolhidos os indivíduos 3 e 1. Neste caso, haverá "crossover" no sétimo bit, e haverá mutação no oitavo bit do indivíduo 8. A população da segunda geração é:

Indivíduo	Cromossomo	Fitness
5	0100111101	4
6	0010001110	6
7	0011010110	5
8	0010001001	7

Esse procedimento será repetido diversas vezes, até que o número de gerações planejado tenha sido atingido, ou um indivíduo com o "fitness" desejado seja conseguido.

6.8.4 - Aplicação no Problema da Detecção de Retas.

O primeiro passo para se utilizar o algoritmo genético na determinação do melhor conjunto de parâmetros para o algoritmo de detecção de retas por otimização é definir a codificação desses parâmetros em cromossomos.

Os cromossomos serão constituídos por cadeias de bits de comprimento 48, da seguinte forma:

- Temperatura Inicial - bits 1 a 8, com valores de 44 a 300.
- Fator de Abaixamento da Temperatura - bits 9 a 16, com valores de 0.5 a 0.9.
- Máximo de iterações por temperatura - bits 17 a 24, com valores de 15 a 190.
- Tolerância do algoritmo downhill simplex - bits 25 a 32, com valores de 0.01 a 4.
- Largura inicial da janela linear - bits 33 a 40, com valores de 1 a 5.
- Abaixamento da temperatura inicial com a diminuição da largura da janela linear - bits 41 a 48, com valores de 0.5 a 0.9.

Utilizando esse modelo para os cromossomos da população, uma população aleatória de 50 indivíduos é criada, e a partir desta evoluirão as melhores soluções.

O segundo passo para a aplicação do algoritmo genético é a definição de uma função de "fitness" adequada ao problema que está sendo endereçado. A função de fitness definirá o que significa "resolver bem" o problema, e medirá o quanto um "indivíduo" se adequa à solução do problema em questão. No caso da busca de parâmetros para o método de detecção de retas, a função de "fitness" utilizada será o desempenho do método com os parâmetros aplicados, calculado através do tempo e precisão médios desse método no decorrer de diversas execuções. A precisão do método

será calculada através da diferença das posições em ρ e θ entre as retas existentes na imagem e as encontradas pelo método.

A probabilidade de "crossover" para o algoritmo genético foi escolhida 0.7, por este ser um valor utilizado freqüentemente e com sucesso em diversos casos. Será utilizado elitismo, sendo o melhor indivíduo mantido de uma geração para outra. Como o elitismo evita a perda de algumas boas seqüências de bits, a probabilidade de mutação pode ser um pouco maior que o convencional (0.01), para garantir a diversidade da população sem, no entanto, prejudicar as configurações boas, e será definida como 0.02.

7 – Implementação

Durante o desenvolvimento do método de busca de retas, vários programas foram criados, incorporando as características necessárias à execução do método. Os programas foram implementados com a utilização da plataforma Borland Delphi.

7.1 - Simulações

O primeiro programa implementado realiza a varredura de uma imagem com uma janela linear com a largura determinada, e mostra o espaço de parâmetros resultante. A Figura 7.1 mostra a janela principal deste aplicativo.

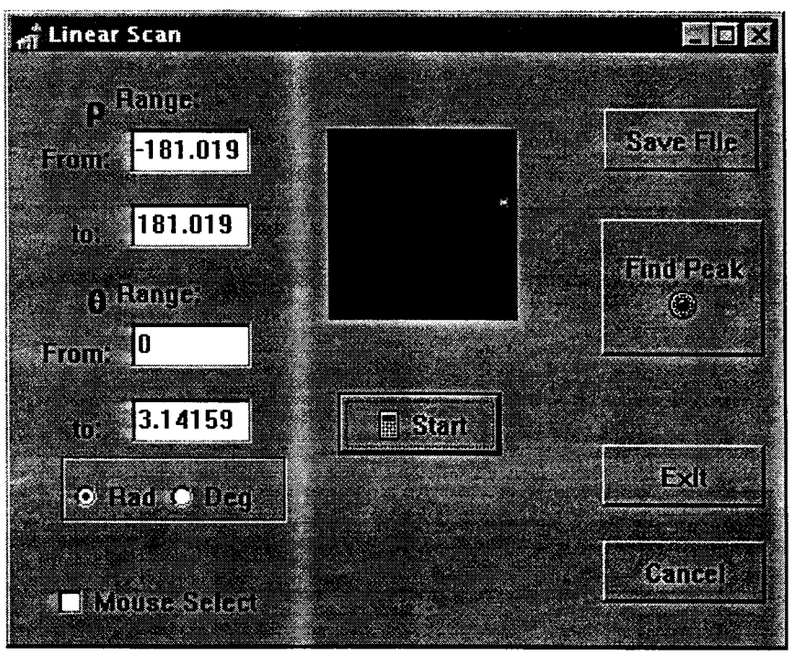


Figura 7.1 - Aplicativo que gera o espaço de parâmetros através da varredura por janelas lineares.

Este aplicativo permite também o "zoom" em uma região específica do espaço de parâmetros, que pode ser definida numericamente, indicando-se as faixas desejadas em ρ e θ; ou pode ser marcada com o mouse, bastando para isso selecionar a opção "mouse

select". A Figura 7.2 é o mesmo espaço de parâmetros mostrado na Figura 7.1, mas com um zoom na região onde existe um pico.

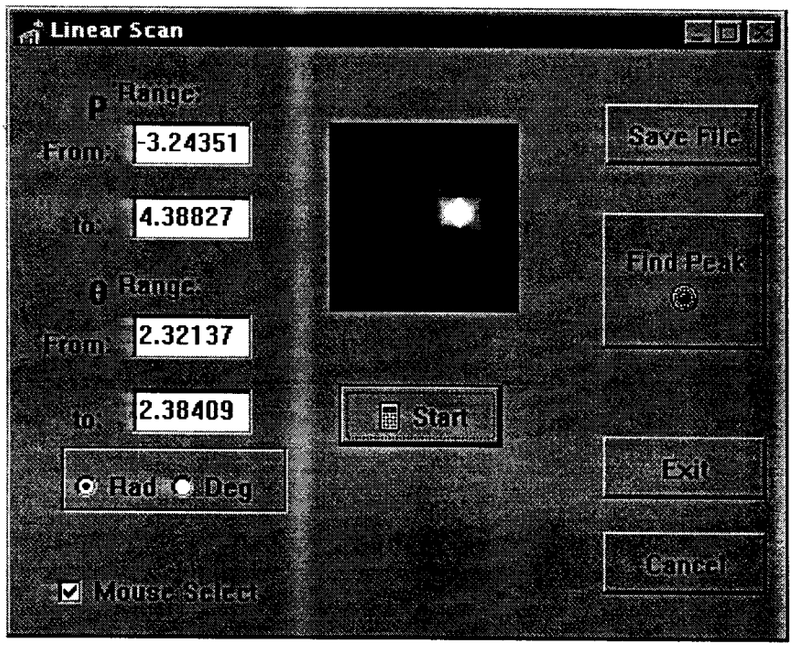


Figura 7.2 - Região do espaço de parâmetros próxima ao pico.

Os limites em ρ e θ do espaço de parâmetros da figura estão mostrados na janela do aplicativo.

O botão "Find Peak" lança um algoritmo de Downhill simplex no centro da área do espaço de parâmetros mostrada na janela do aplicativo, e coloca um ponto vermelho no máximo encontrado. O botão "Save File" grava a imagem do espaço de parâmetros em um arquivo no formato bitmap. A Figura 7.3 mostra o máximo encontrado pelo algoritmo de downhill simplex no espaço de parâmetros acima.

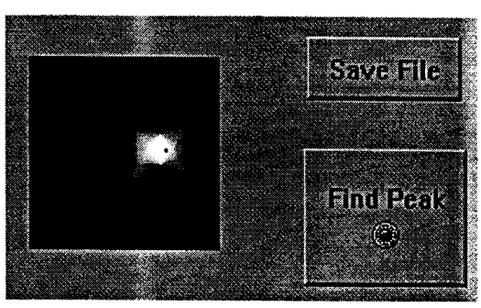


Figura 7.3 - Máximo encontrado pelo algoritmo "downhill simplex".

Este programa nos permite visualizar o espaço de parâmetros para diversas imagens, e também verificar o desempenho do algoritmo proposto por Nelder e Mead em situações simples.

O próximo aplicativo implementado tem como objetivo avaliar o desempenho de diversos algoritmos de maximização, e incorpora uma estratégia de annealing ao algoritmo de Downhill simplex. A janela principal desse aplicativo está na Figura 7.4.

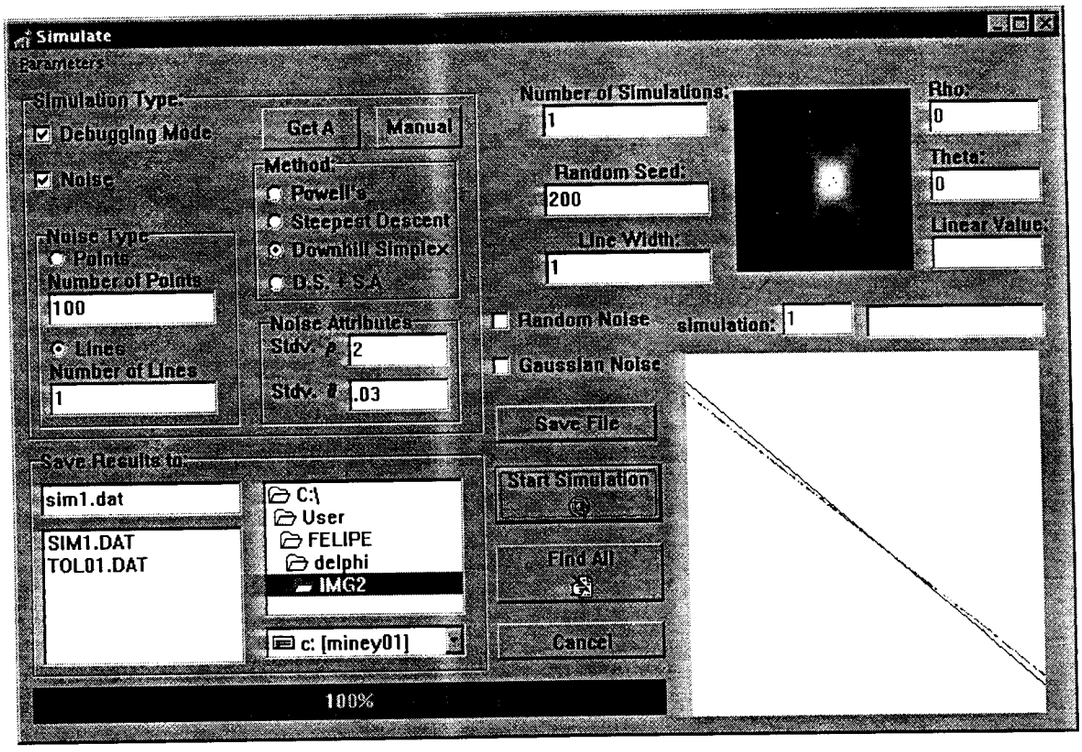


Figura 7.4 - Aplicativo para avaliação dos métodos de otimização.

Este aplicativo tem diversos recursos. As imagens utilizadas para a simulação podem ser de diversos tipos, sendo constituídas sempre de uma reta, que é o objetivo da busca; e ruídos, que podem ser pontos uniformemente espalhados ou retas com 25% dos pontos apagados, e parâmetros próximos aos da reta original. O grupo "noise type" controla a adição desses ruídos à imagem. Os parâmetros do ruído na forma de retas podem ser modificados no grupo "noise attributes", onde os desvios padrão em ρ e θ são

especificados. As médias destas distribuições são os parâmetros da reta original. A Figura 7.5 mostra esses recursos.

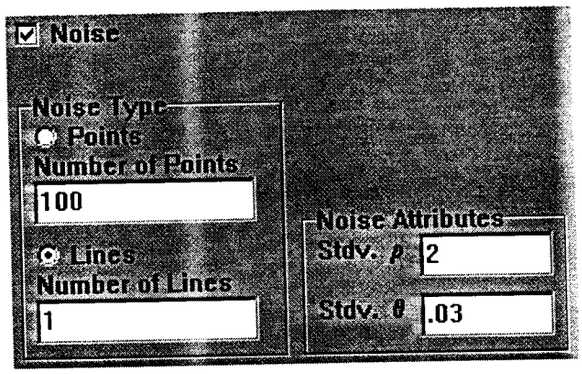


Figura 7.5 - Ajuste de ruídos na imagem das simulações.

O botão que ativa o modo de verificação de erros ("Debugging mode") faz com que o programa construa o espaço de parâmetros completo próximo as retas, e mostre alguns dados durante a execução. Quando este modo está desativado, o espaço de parâmetros não é mostrado, e todos os resultados são gravados em arquivo.

Quanto aos parâmetros de "annealing" e da largura adaptativa, estes podem ser vistos ou modificados através do menu "parameters", mostrado na Figura 7.6. As janelas para visualização e modificação de parâmetros são mostradas nas Figuras 7.7 e 7.8.

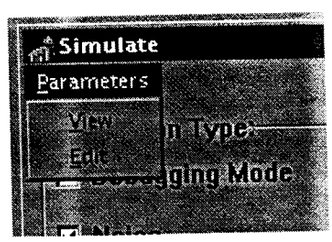


Figura 7.6 - Menu para a visualização dos parâmetros do método de otimização.

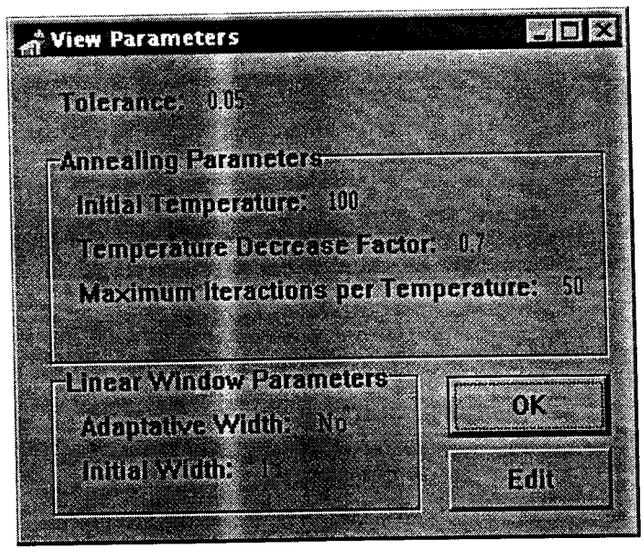


Figura 7.7 - Janela para visualização dos parâmetros.

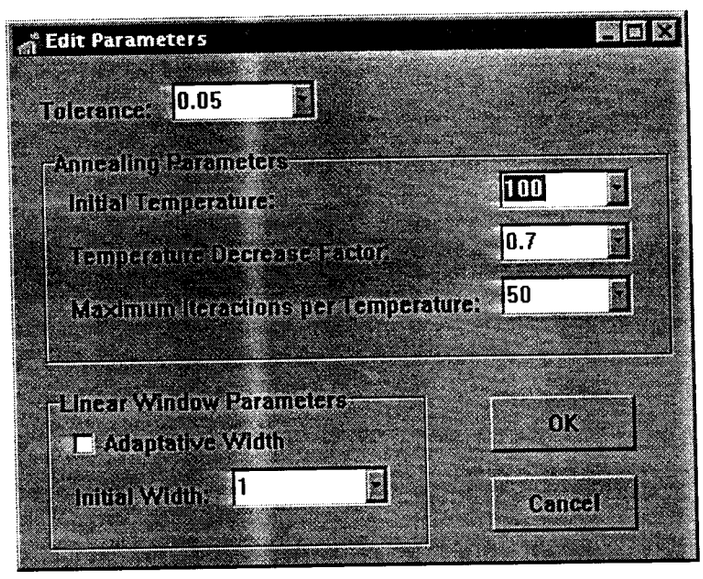


Figura 7.8 - Janela para a edição dos parâmetros.

Para a realização de simulações em imagens pré-definidas, pode-se utilizar o botão "Get A", que carrega uma imagem de um arquivo à escolha do usuário. Pode-se também desenhar as retas através do botão "manual".

Para encontrar o melhor conjunto de parâmetros para o método de otimização, foi utilizado um algoritmo genético. A janela principal do aplicativo para a utilização do algoritmo genético está na Figura 7.9.

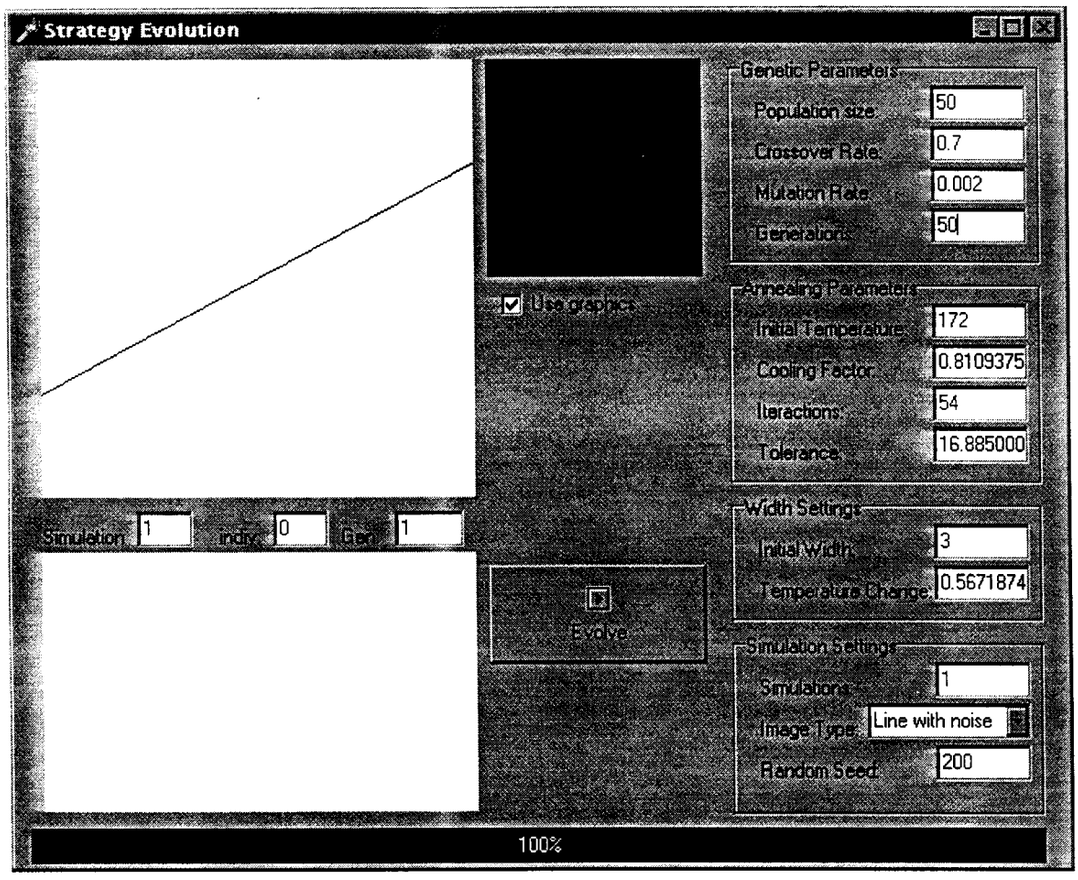


Figura 7.9 - Aplicativo para escolha de parâmetros via algoritmo genético.

Os grupos "Annealing Parameters" e "Width Settings" são preenchidos automaticamente, com os dados do indivíduo (conjunto de parâmetros) que está sendo testado. Os parâmetros do algoritmo genético (Tamanho da população, taxa de crossover, taxa de mutação e número de gerações) podem ser modificados para a realização de diferentes testes.

Os testes podem ser feitos para 3 tipos de imagens: retas com ruído, perfis de aranha e imagens reais (sendo esta o resultado da detecção de bordas de uma imagem capturada por uma câmera). A escolha entre estes tipos é feita no menu "Image type".

Na imagem localizada à esquerda do botão "Evolve" é desenhado um gráfico contendo os "fitness" máximo e médio para cada geração, e os dados do melhor indivíduo, ao fim da evolução, são gravados em arquivo.

7.2 – Protótipo

A concepção e construção de um protótipo para o sistema de detecção de retas proposto também é parte do trabalho desenvolvido. Esta seção descreve a construção e o funcionamento desse protótipo, além de alguns resultados obtidos com ele.

O protótipo consiste em uma estrutura de alumínio na qual está preso um motor de passo acoplado a uma engrenagem. Nessa estrutura está fixo também o anel externo de um mancal de rolamento. No anel interno desse mancal está presa outra engrenagem, no centro da qual está colocada uma câmera CCD, como mostra a Figura 7.10.

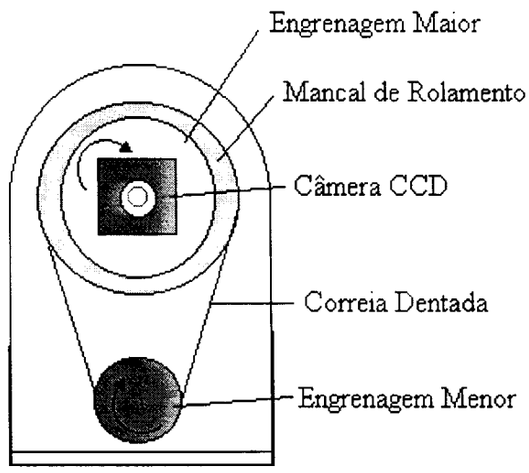


Figura 7.10 – Esquema do protótipo.

O motor de passo utilizado no protótipo tem a resolução de 200 passos por volta, ou seja, de 1,8° por passo. O fator de redução entre as duas engrenagens é de 1 para 7,2 e, portanto, para cada passo do motor, a câmera é rotacionada de 0,25°.

Como este protótipo utiliza como sensor uma câmera CCD convencional, diferentes valores de ρ são obtidos fazendo-se leituras de diferentes colunas do CCD, e janelas lineares de larguras maiores são facilmente simuladas utilizando-se mais colunas do CCD ao mesmo tempo. A variação do parâmetro θ é feita com a rotação da câmera, de modo

que as colunas do CCD estão sempre posicionadas perpendicularmente ao valor de θ desejado.

A aquisição de imagens da câmera é feita através de uma placa Video Blaster RT300, da Creative Labs. O controle do motor de passo é feito através de um driver, sendo este controlado pela porta paralela do micro computador. A Figura 7.11 mostra um diagrama de blocos que ilustra o funcionamento do conjunto.

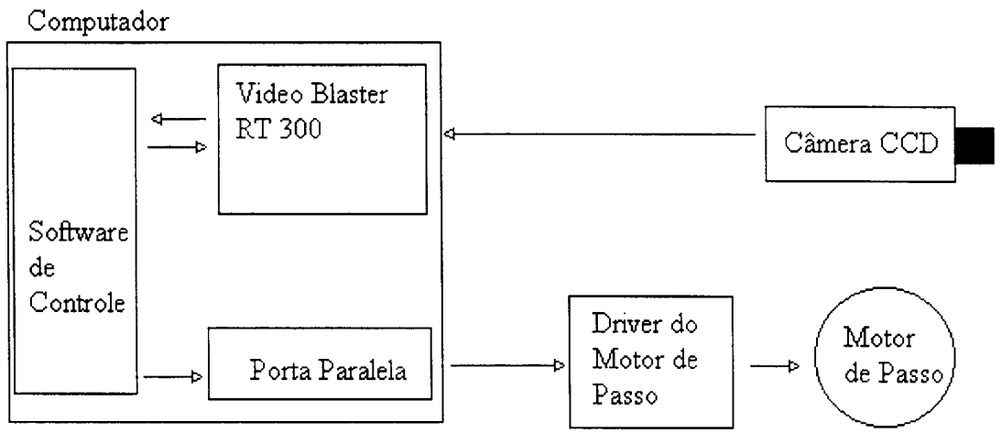


Figura 7.11 – Diagrama de blocos do protótipo.

O controle da aquisição de imagens e da rotação do motor de passo foi realizado via software, utilizando-se para isso um programa desenvolvido em Delphi 3.0. O componente utilizado para a aquisição de imagens foi um shareware, TSCap para Delphi 3.0.

O driver utilizado para o motor de passo está esquematizado na Figura 7.12, e opera através da polarização sucessiva das bobinas do motor. O sinal mandado pela porta paralela do micro determina o sentido da rotação, e manda os pulsos que constituem o "clock" que acionará o motor.

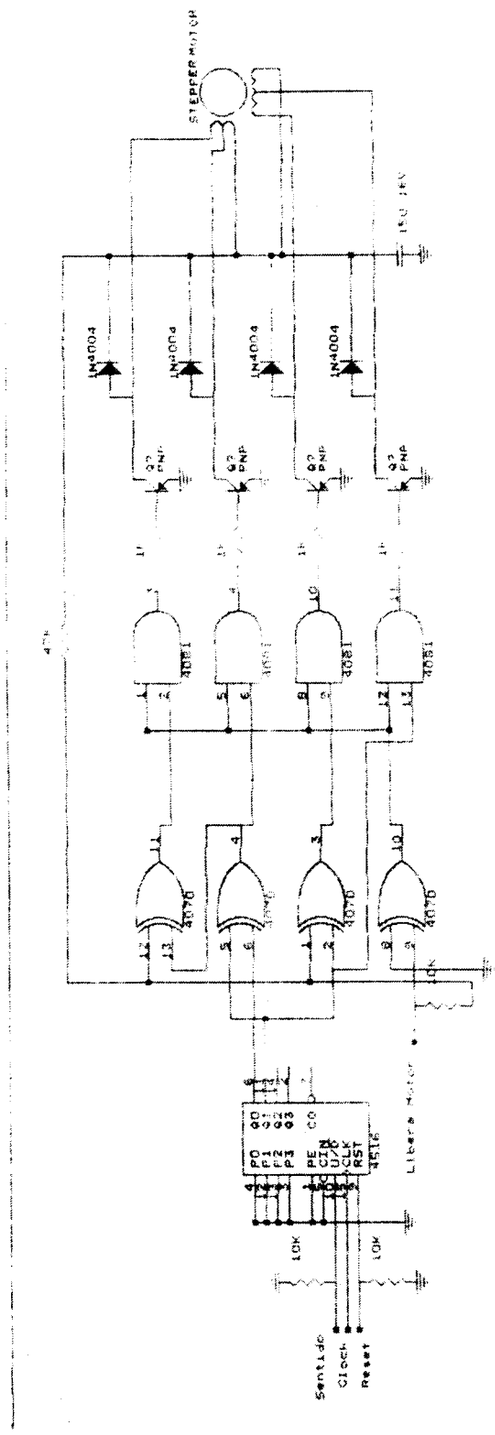


Figura 7.12 – Esquema do Driver para motor de passo.

O protótipo tem capacidade para acomodar engrenagens de diferentes tamanhos acopladas ao motor de passo, variando assim o ângulo percorrido em cada passo. A estrutura do protótipo foi inteiramente concebida para este trabalho, e sua construção foi realizada com o auxílio da oficina mecânica do IFSC. A Figura 7.13 mostra o protótipo construído, com uma câmera acoplada.

Figura 7.13 – Protótipo.

Este protótipo utiliza um sensor convencional, uma câmera CCD para simular as janelas lineares, e por ser grande e relativamente pesado, devido ao tamanho das engrenagens, tem seu desempenho limitado, pois o motor de passo não consegue movimentar rapidamente todo o conjunto. Outros protótipos foram propostos, mas sua construção não foi efetuada por envolver a engenharia de mecanismos complexos e precisos.

Um dos protótipos propostos foi a de um dispositivo utilizando apenas um fotodetector, e com mecanismos ópticos e mecânicos para o direcionamento da luz que chega até o fotodetector. Este dispositivo está esquematizado na Figura 7.14.

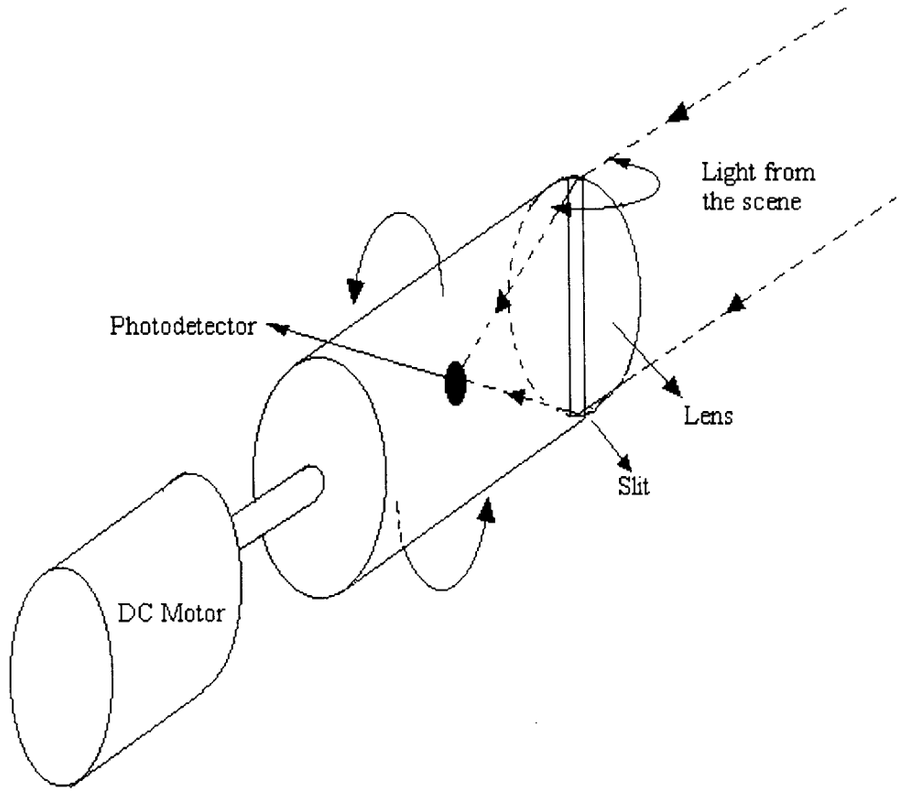


Figura 7.14 – Protótipo com somente um fotodetector.

No protótipo da Figura 7.14, a luz é refratada por uma lente móvel, que faria o papel da movimentação em ρ, e o conjunto inteiro seria capaz de rotação, para a movimentação em θ. A luz refratada passaria por uma fenda, e seria concentrada por uma lente sobre o fotodetector, cujo sinal resultante seria proporcional a intensidade luminosa total passando pela fenda. Este protótipo utiliza o modelo com apenas uma região fotodetectora, e não o de vários detectores justapostos.

Alternativamente, ao invés de concentrar a luz passando pela fenda, esta poderia incidir sobre um arranjo de fotodetectores, com largura 2 ou 3, para facilitar a detecção de bordas e a codificação da posição dos estímulos ao longo da linha de fotodetectores.

7.2.1 – Interface

A interface para a aquisição de imagens e controle do motor de passo, assim como a detecção de retas, foram implementadas em Delphi 3.0. O Aplicativo desenvolvido possui apenas uma janela principal, mostrada na Figura 7.15, onde todas as operações disponíveis são apresentadas de maneira simples.

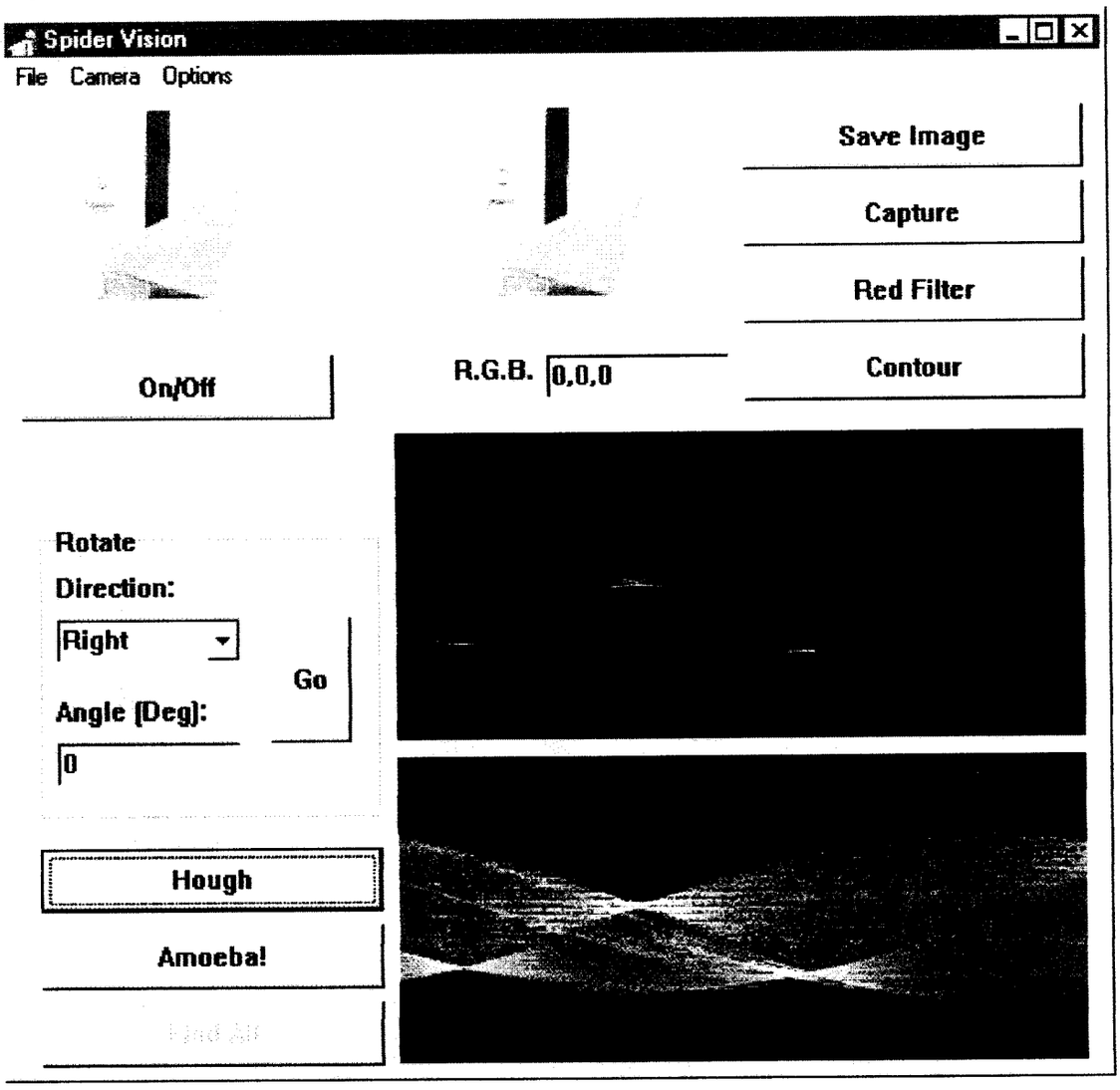


Figura 7.15 – Tela principal do aplicativo que controla o protótipo.

A imagem mostrada à direita, na Figura 7.16, foi capturada pela câmera. A imagem central é o resultado de uma filtragem da primeira, e a imagem da esquerda foi obtida detectando-se as bordas da imagem central. Desse modo, para cada posição da câmera, a respectiva imagem é capturada, filtrada e suas bordas são detectadas.

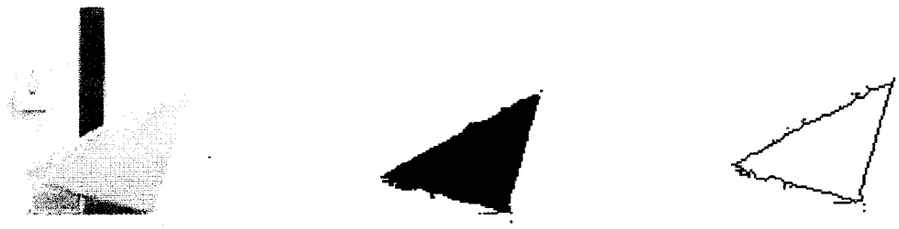


Figura 7.16 – Imagens capturada, filtrada e contornos detectados.

O aplicativo permite o controle da rotação do motor de passo, bastando para isso que se digite, nos locais mostrados na Figura 7.17, o ângulo que se deseja rotacionar, e a direção desta rotação.

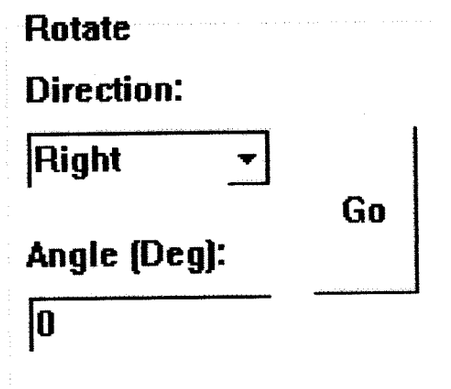


Figura 7.17 – Controle da rotação do motor de passo.

As imagens mostradas na Figura 7.18 são os resultados da parametrização por janelas lineares da imagem da Figura 7.16. A parte superior da figura mostra o resultado da parametrização em si, e a parte inferior mostra este resultado em uma escala logarítmica, para facilitar a visualização de algumas características. Na Figura 7.18,

podemos facilmente visualizar os picos, representados por pontos mais claros, nas superfícies de parâmetros.

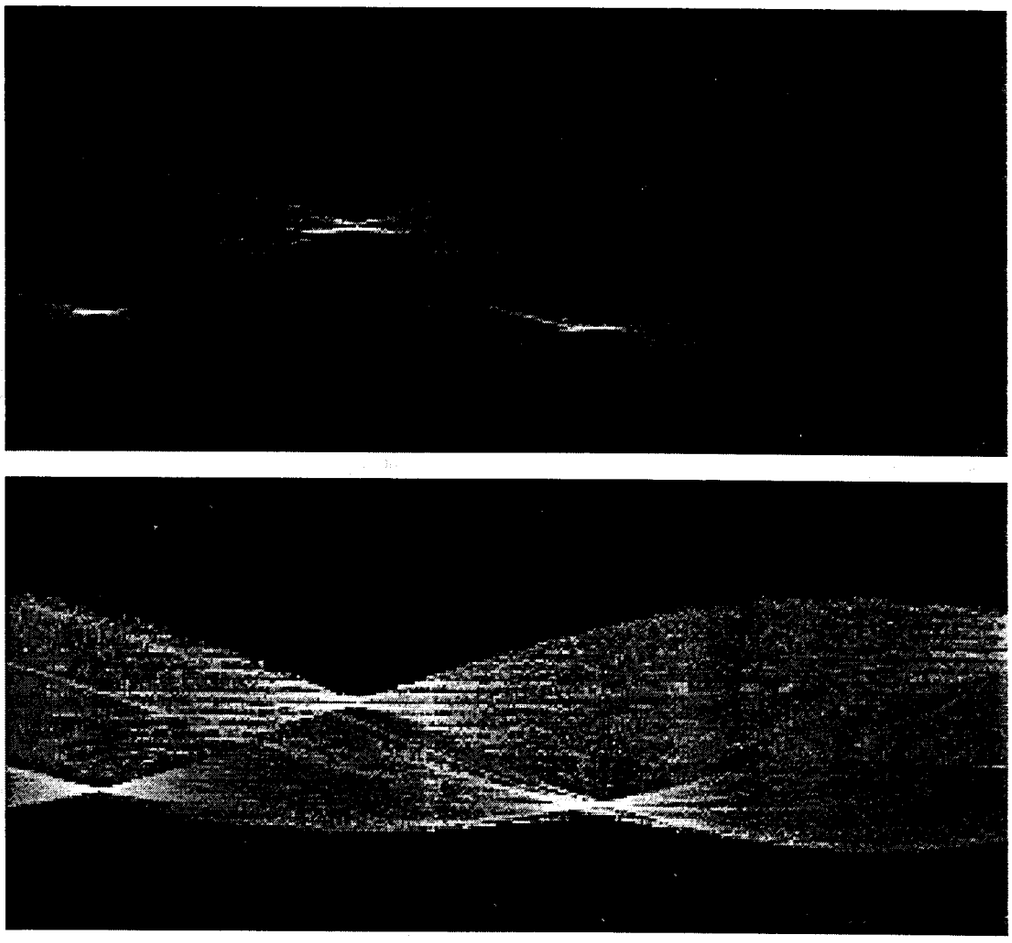


Figura 7.18 – Espaços de parâmetros gerados através do protótipo.

7.2.2 – Controle

A movimentação do motor de passo, feita através da porta paralela do computador, é controlada da seguinte maneira:

- O bit menos significativo do byte que é mandado para a porta paralela faz o papel do clock do motor de passo, ou seja, para que um passo seja dado, é necessário que esse valor vá de 0 para 1 e novamente para 0.
- O segundo bit desse byte determina o sentido da rotação, sendo que 0 significa rotação para a esquerda, e 1, rotação para a direita.

Os valores que são enviados para a porta, segundo as descrições acima, são:

1) Movimento para a direita:

00000010 – 2

Delay

00000011 – 3

Delay

00000010 – 2

2) Movimento para a esquerda:

00000000 – 0

Delay

00000001 – 1

Delay

00000000 – 0

O "delay" representado acima é o tempo que o pulso precisa ficar com valor 1 para que o motor se movimente.

8 – Resultados

Neste capítulo são apresentados os resultados obtidos para a escolha da configuração dos parâmetros do método de busca de retas por janelas lineares, juntamente com os resultados obtidos na detecção de retas com a utilização do algoritmo considerando o melhor conjunto de parâmetros encontrado.

8.1 - Resultados Preliminares: Busca por Retas com e sem "Simulated Annealing"

Os primeiros resultados obtidos no trabalho são relacionados à comparação do método "downhill simplex" simples com sua versão que incorpora "simulated annealing". Para o "simulated annealing" foram utilizados alguns parâmetros, escolhidos entre poucas possibilidades, não garantindo assim um conjunto ótimo de parâmetros. As possibilidades de parâmetros para "annealing" utilizadas foram:

- Temperatura Inicial: 200, 150 ou 100 (To).
- Fator de Resfriamento: 0.9, 0.7 e 0.5 (β).
- Máximo de Iterações: 50.

As imagens utilizadas para esses testes correspondem a uma reta com o pior tipo de ruído para o algoritmo de busca: pontos de ruído alinhados, com parâmetros próximos ao da reta procurada, como mostra a Figura 8.1. O tamanho dessas imagens é de 256x256 “pixels”.

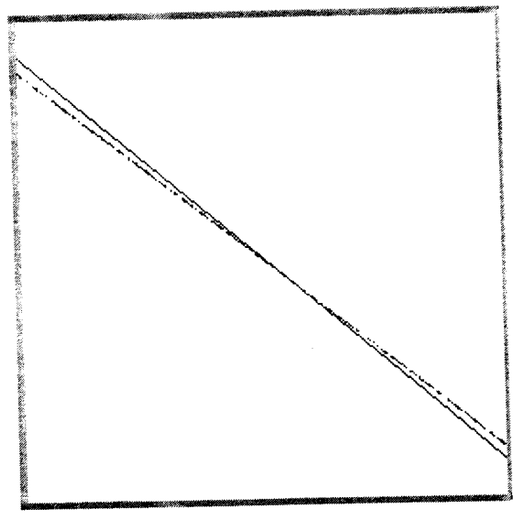


Figura 8.1 - Reta com ruído.

O experimento foi executado diversas vezes para cada configuração de parâmetros, e para cada caso os parâmetros da reta encontrada foram comparados aos da reta original e aos do ruído. O algoritmo encontrou a "reta" formada pelos pontos de ruído alinhados algumas vezes. A Tabela 1 mostra o número de vezes, em 1000 experimentos, que cada algoritmo encontrou a reta certa.

Técnica	*Corretas*	*Incorretas*
Sem annealing	803	197
β=0.5; T_0=200	831	169
β=0.7; T_0=200	823	177
β=0.9; T_0=200	858	142
β=0.9; T_0=50	825	175

Tabela 1 – Número de retas encontradas corretamente.

Além disso, foram também colocados em uma tabela os resultados desses algoritmos com relação ao tempo de processamento e à precisão dos parâmetros encontrados. Na Tabela 2, o erro em ρ está representado por ε_ρ, e o erro em θ por ε_θ. Os tempos mostrados correspondem ao desempenho dos algoritmos em computadores PC

Pentium 200 MHz MMX. Vários casos foram avaliados, com apenas uma reta, e com o ruído linear presente.

	ε_θ *(graus)*	ε_ρ *(pixels)*	*Tempo (ms)*
1 reta	-0°00'04''±0°02'45'' (-0°07'35''; 0°11'22'')	-0.040±0.075 (-0.17; 0.26)	1803.24±561.79 (549; 4888)
1 reta, +SA T_0=50 β = 0.9	-0°07'04''±3°45'41'' (-0°10'12''; 12°54'31'')	-0.014±0.087 (-0.43; 0.33)	7682.19±3301.44 (604; 14391)
2 retas	0°15'13''±0°34'37'' (-0°15'05''; 3°13'59'')	0.38±0.97 (-0.27; 5.81)	2907.55±1179.69 (549; 8183)
2 retas +SA T_0=200 β = 0.5	0°14'53''±0°35'57'' (-0°19'02''; 3°20'12'')	0.36±0.91 (-1.32; 6.34)	3011.04±951.48 (714; 9722)
2 retas +SA T_0=200 β = 0.7	0°15'48''±0°38'09'' (-0°25'51''; 3°33'37'')	0.38±0.95 (-1.19; 6.77)	5535.43±1627.90 (989; 14006)
2 retas +SA T_0=200 β = 0.9	0°13'38''±0°37'15'' (-0°36'57''; 4°31'28'')	0.29±0.81 (-2.14; 5.00)	12131.38±4618.3 (1044; 21366)
2 retas +SA T_0=50 β = 0.9	0°13'14''±0°33'33'' (-0°11'33''; 4°20'16'')	0.32±0.84 (-0.57; 5.98)	5891.88±2718.30 (714; 14446)

Tabela 2 – Erros e tempos de processamento para algumas estratégias, na forma "média ± desvio padrão (mínimo; máximo).

Os valores da Tabela 2 estão no formato "média ± desvio padrão (mínimo, máximo)". Podemos perceber que nos casos em que há apenas a reta a ser encontrada o erro médio para os valores de ρ e θ foi bem menor que nos casos em que a reta ruidosa está presente. Isso ocorre porque nas ocasiões em que há duas retas, o algoritmo algumas vezes convergiu para a reta errada.

Com relação aos erros em ρ e θ, os algoritmos testados com "simulated annealing" e temperatura inicial To = 200 resultaram em uma precisão maior, mas o

tempo necessário de processamento também foi aumentado. Com a utilização de $\beta = 0.9$ e modificando-se a temperatura inicial para 50, ocorre uma grande diminuição no tempo necessário, sem haver muita perda em precisão.

O desempenho do algoritmo com fator de resfriamento $\beta = 0.7$ foi pior do que o mesmo com $\beta = 0.5$, o que indica que o espaço formado pelos parâmetros de annealing é não-linear. Nesses primeiros resultados também não foi modificado o número máximo de iterações por temperatura.

8.2 - Algoritmo Genético para Determinação dos Parâmetros

Como dito na Seção 6.8, um fator decisivo nas escolhas resultantes da aplicação destes é a função de "fitness" utilizada. No caso da avaliação de um algoritmo para detecção de retas, a definição do que é um método "melhor" implica em um compromisso entre a precisão do método e a velocidade de execução deste.

O cálculo da função de "fitness" foi adaptado de acordo com as necessidades deste projeto, e um modelo básico para este cálculo é da forma:

$$fitness = \frac{\ln(1 + K.prec)}{\sqrt{time}}$$

Segundo o modelo acima, a precisão interfere no "fitness" de tal forma que uma precisão muito baixa diminui muito o fitness, mas depois de um certo ponto, os aumentos em precisão não são muito significativos. O tempo de execução diminui o fitness, favorecendo métodos mais rápidos. A Figura 8.2 mostra a superfície de "fitness" para tempos entre 2000 e 8000 milissegundos, e precisões entre 0 e 5. Essas faixas de variação foram as freqüentemente encontradas nos experimentos.

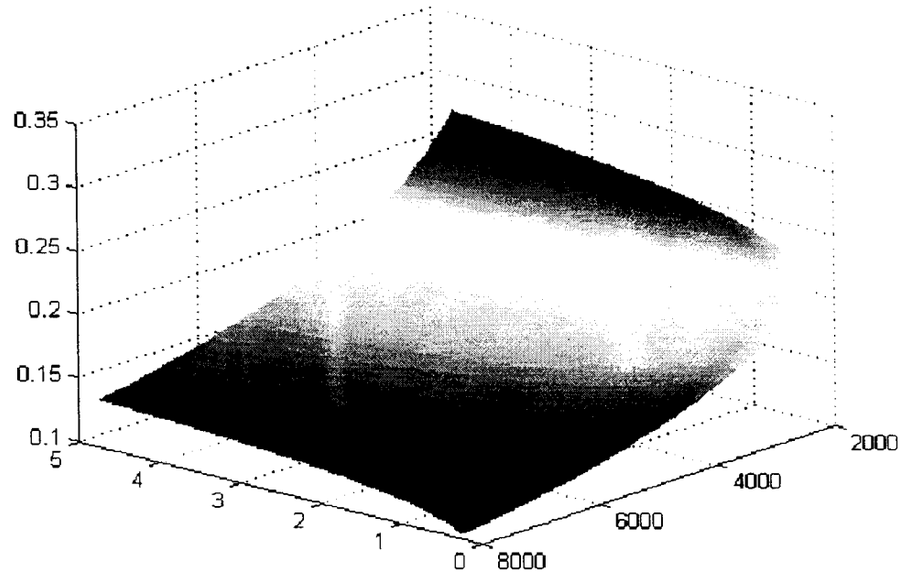


Figura 8.2 – Superfície de "fitness", para K=4000.

Alterando-se o valor da constante K, o "fitness" pode ser ajustado para favorecer métodos mais rápidos ou mais precisos. Essa característica é muito importante, pois direciona a escolha de um conjunto de parâmetros de acordo com as necessidades do problema endereçado.

Os primeiros testes foram realizados com um valor baixo para K (K=4000), favorecendo métodos mais rápidos. Foram realizadas 5 execuções, cada uma com populações de 50 indivíduos sendo evoluídos por 80 gerações. A imagem utilizada para essas simulações foi a de um perfil de aranha, mostrado na Figura 8.3. O espaço de parâmetros foi particionado em 64 partes, e o algoritmo foi lançado uma vez em cada uma das partes. Os resultados obtidos foram como mostra a Tabela 3.

Evolução	Melhor "fitness"	Temp. Inicial	Fator de Resfria/o	Máx. de Iterações	Toler.	Largura Inicial	Resfria/o c/ largura
1	0.9966	154	0.54	50	2.33	1	0.54
2	1.403	334	0.52	18	2.96	1	0.60
3	1.095	156	0.54	35	2.32	1	0.74
4	0.8747	90	0.62	50	3.17	1	0.68
5	1.755	213	0.52	29	0.87	1	0.57

Tabela 3 – Resultados para detecção de um perfil de aranha, K=4000

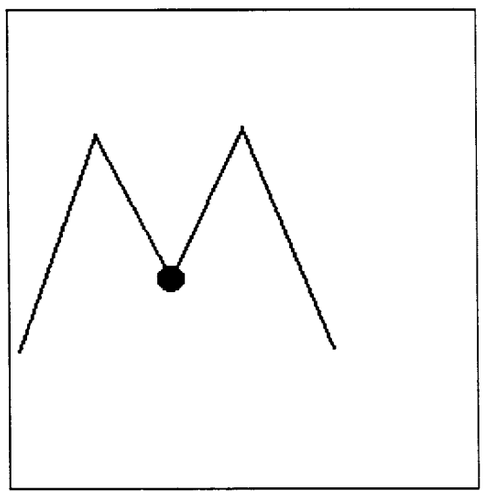


Figura 8.3 – Perfil de aranha utilizado em simulações.

Como pôde-se observar, as tolerâncias para o critério de parada do algoritmo foram razoavelmente altas, significando que a diferença entre os valores dos vértices do "simplex" não precisa ser muito pequena para que o algoritmo termine. Ocorreu também que todos os conjuntos de parâmetros utilizaram uma largura inicial 1 para a janela linear. Além disso, notamos que o fator de diminuição da temperatura foi bem pequeno, indicando um "resfriamento" rápido. As temperaturas iniciais não foram muito altas, com exceção do resultado obtido em (2).

Novamente utilizando populações de 50 indivíduos sendo evoluídos por 80 gerações, designamos um valor maior para a constante K (K = 40000), com a intenção de

se obter um método mais preciso, ainda que mais lento. Para esse valor de K, os resultados obtidos estão na Tabela 4.

Evolução	Melhor "fitness"	Temp. Inicial	Fator de Resfria/o	Máx. de Iterações	Toler.	Largura Inicial	Resfria/o c/ largura
1	1.274	152	0.69	15	0.12	2	0.57
2	1.442	204	0.64	17	0.07	2	0.63
3	1.190	232	0.59	22	0.09	3	0.52
4	1.465	227	0.57	20	0.06	2	0.59
5	0.941	181	0.78	24	0.10	3	0.67

Tabela 4 – Resultados para detecção de um perfil de aranha, K=40000.

Os resultados obtidos com essas execuções do método mantiveram o número de iterações por temperatura relativamente baixo, e as temperaturas dos melhores indivíduos foram abaixadas consideravelmente com a diminuição da largura das janelas. Por outro lado, a largura inicial das janelas lineares nos métodos foi 2 ou 3, as tolerâncias foram bem menores e as temperaturas iniciais foram um pouco mais altas, indicando que o algoritmo nessas condições realmente fornece resultados mais precisos.

Os resultados para as estratégias testadas com imagens contendo apenas uma reta e ruído, como mostrado na Figura 8.1 estão nas Tabelas 5 e 6, para K=4000 e 40000, respectivamente.

Evolução	Melhor "fitness"	Temp. Inicial	Fator de Resfria/o	Máx. de Iterações	Toler.	Largura Inicial	Resfria/o c/ largura
1	2.351	122	0.56	22	1.23	2	0.59
2	2.505	134	0.61	28	1.72	1	0.74
3	1.983	117	0.58	20	1.12	1	0.68
4	2.775	103	0.55	25	0.98	1	0.84
5	1.802	136	0.59	31	1.47	1	0.62

Tabela 5 – Resultados para a detecção de uma reta, K=4000.

Os parâmetros dos melhores indivíduos são semelhantes aos encontrados utilizando o mesmo critério de "fitness" para as imagens contendo perfis de aranha, sendo a principal diferença as tolerâncias menores encontradas para o caso de uma só reta, e as temperaturas iniciais mais baixas para a estratégia de "annealing".

Evolução	Melhor "fitness"	Temp. Inicial	Fator de Resfria/o	Máx. de Iterações	Toler.	Largura Inicial	Resfria/o c/ largura
1	2.784	230	0.63	32	0.09	2	0.54
2	3.221	284	0.65	27	0.06	2	0.53
3	3.402	272	0.63	36	0.11	2	0.54
4	2.983	197	0.70	28	0.08	3	0.52
5	3.112	315	0.67	32	0.05	2	0.52

Tabela 6 – Resultados para a detecção de uma reta, K=40000.

Como este caso apresenta uma ocorrência proposital de um máximo local, ao favorecermos a precisão, a estratégia de "annealing" adotada pelos melhores indivíduos utiliza temperaturas iniciais mais altas, e redução de temperatura mais lenta, para contornar este máximo local. Além destes testes, foram realizados testes com uma imagem resultante da detecção de bordas de uma imagem real, mostrada na Figura 8.4.

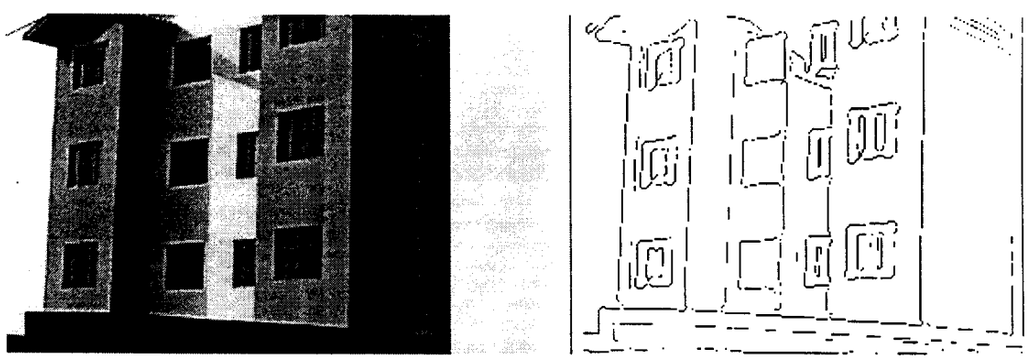


Figura 8.4 – Imagem real e resultado da detecção de bordas.

A Tabela 7 apresenta os resultados para os melhores conjuntos de parâmetros, utilizando a imagem da Figura 8.4 e K=4000. Para K=40000, os resultados estão na Tabela 8.

Evolução	Melhor "fitness"	Temp. Inicial	Fator de Resfria/o	Máx. de Iterações	Toler.	Largura Inicial	Resfria/o c/ largura
1	1.439	142	0.56	18	1.24	1	0.74
2	1.520	119	0.58	22	0.93	1	0.66
3	0.983	136	0.53	23	1.72	1	0.53
4	1.266	132	0.56	20	1.33	1	0.85
5	1.412	135	0.59	20	1.27	1	0.64

Tabela 7 – Resultados para a detecção de retas em imagem real, K=4000.

Evolução	Melhor "fitness"	Temp. Inicial	Fator de Resfria/o	Máx. de Iterações	Toler.	Largura Inicial	Resfria/o c/ largura
1	1.508	240	0.63	23	0.12	2	0.52
2	1.840	276	0.57	21	0.08	2	0.54
3	1.429	311	0.62	31	0.11	1	0.71
4	1.677	259	0.65	28	0.13	2	0.56
5	1.549	217	0.59	26	0.11	1	0.63

Tabela 8 – Resultados para a detecção de retas em imagem real, K=40000.

8.3 – Detecção de retas

Obtidos os conjuntos de parâmetros acima, foram analisados os valores para as retas encontradas pelo algoritmo utilizando o melhor conjunto de parâmetros de cada caso, para verificar a posição das retas encontradas comparadas às retas da imagem original.

Devido às altas tolerâncias da condição de parada encontradas para o algoritmo em sua versão mais veloz (K=4000), as retas encontradas por este estão próximas às originais, mas com uma certa imprecisão em ρ e θ.

A Tabela 9 mostra, para testes com apenas uma reta e ruído, alguns valores obtidos para as retas originais e as retas encontradas. Os valores de θ estão em radianos. A coluna à direita mostra os valores dos tempos utilizados na execução de cada caso.

ρ original	ρ encontrado	Erro em ρ	θ original (rad)	θ encontrado (rad)	Erro em θ (rad)	Tempo (ms)
2.45	2.27	0.18	0.5381	0.5423	0.0042	2055
2.91	3.32	0.41	0.2047	0.2001	0.0046	2538
-3.75	-3.15	0.60	1.1830	0.9833	0.1997	2229
-1.81	-1.58	0.23	1.7355	1.7742	0.0387	2842
0.53	0.48	0.05	0.8329	0.7904	0.0425	3070

Tabela 9 – Erros e tempos para a detecção de uma reta, K=4000.

Para se ter uma noção gráfica dos resultados, a Figura 8.5 mostra algumas imagens com retas originais e detectadas. As retas originais são mostradas tracejadas.

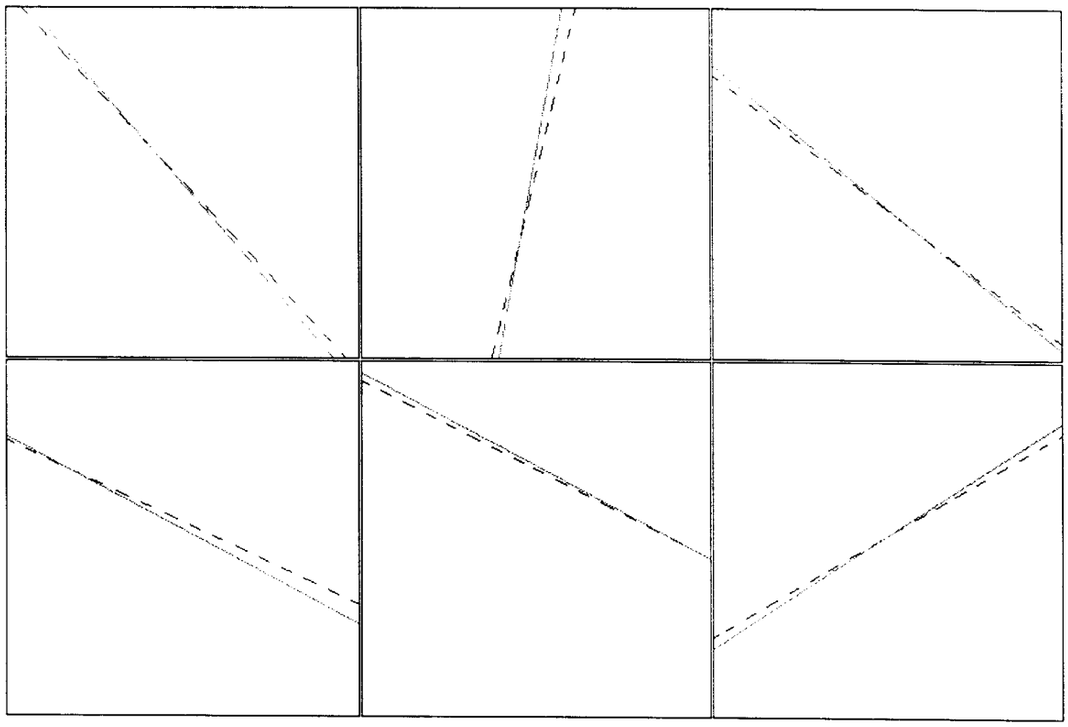


Figura 8.5 – Resultados da detecção de uma reta, para K=4000.

Ainda para imagens contendo apenas uma reta, o conjunto de parâmetros obtido para K = 40000 foi utilizado, e alguns dos resultados estão na Tabela 10.

ρ original	ρ encontrado	Erro em ρ	θ original (rad)	θ encontrado (rad)	Erro em θ (rad)	Tempo (ms)
4.23	4.18	0.05	0.2944	0.2938	0.0004	8255
1.63	1.62	0.01	1.6385	1.6382	0.0003	7910
1.12	1.09	0.02	1.2590	1.2585	0.0005	9258
-5.92	-5.95	0.03	2.3077	2.3071	0.0006	8815
2.88	2.86	0.02	0.6491	0.6487	0.0004	8704

Tabela 10 – Erros e tempos para a detecção de uma reta, K=40000.

Como podemos notar, os erros para essa estratégia de detecção de retas foram bem menores, mas o tempo necessário para a execução aumentou consideravelmente. A Figura 8.6 mostra algumas das retas encontradas, traçadas sobre as retas originais.

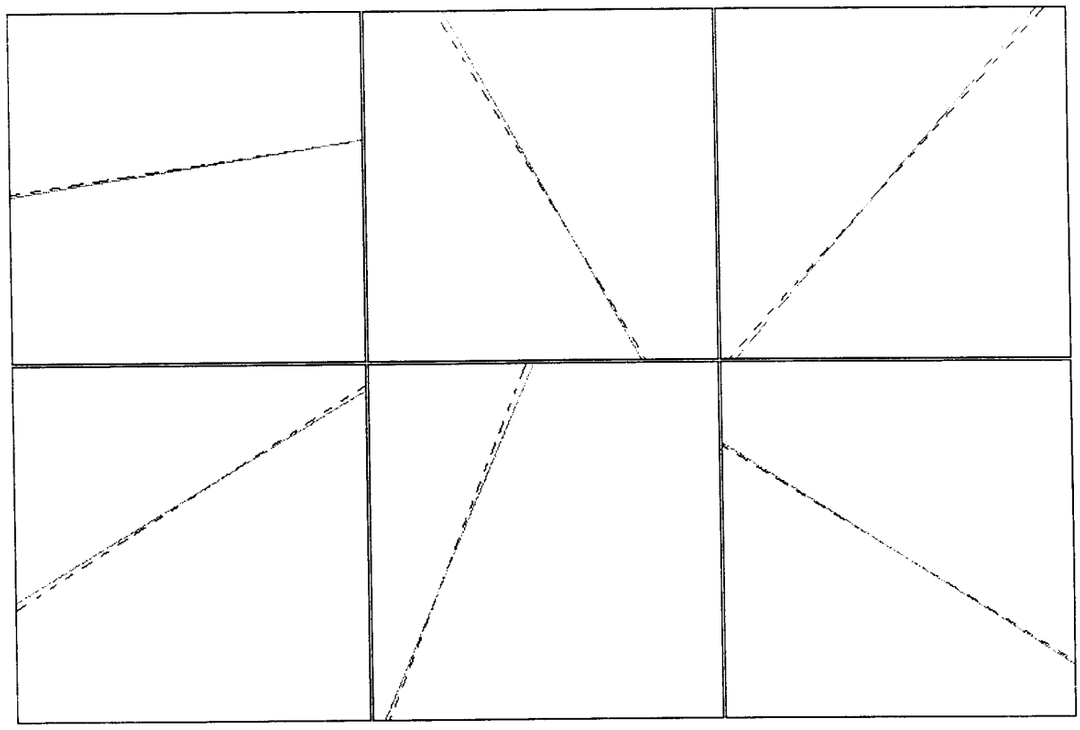


Figura 8.6 – Resultados da detecção de uma reta, K=40000.

Para imagens contendo o perfil de aranha mostrado na Figura 8.3, os resultados obtidos serão colocados na Tabela 12, mostrando os erros médios em ρ e θ para os dois valores de K (K = 4000 e K = 40000), para 50 execuções. Os valores de ρ e θ para os segmentos de reta da Figura 8.3 estão na Tabela 11.

Reta	ρ	θ
1	96	2.7925
2	28	2.7053
3	24	0.4014
4	-40	0.5236

Tabela 11 – Valores dos parâmetros para as retas da figura 8.3.

Reta	$\varepsilon\rho$ (K=4000)	$\varepsilon\theta$ (K=4000)	$\varepsilon\rho$ (K=40000)	$\varepsilon\theta$ (K=40000)
1	1.35	0.0308	0.06	0.0013
2	1.02	0.0113	0.02	0.0010
3	0.94	0.0092	0.04	0.0006
4	0.78	0.0125	0.02	0.0007

Tabela 12 – Erros médios em ρ e θ para as retas encontradas.

A Figura 8.7 apresenta várias imagens, com as retas detectadas traçadas em vermelho sobre o perfil de aranha original. Foram utilizados os melhores indivíduos para K=40000. Para K=4000, a precisão foi bem menor, e os resultados estão na Figura 8.8.

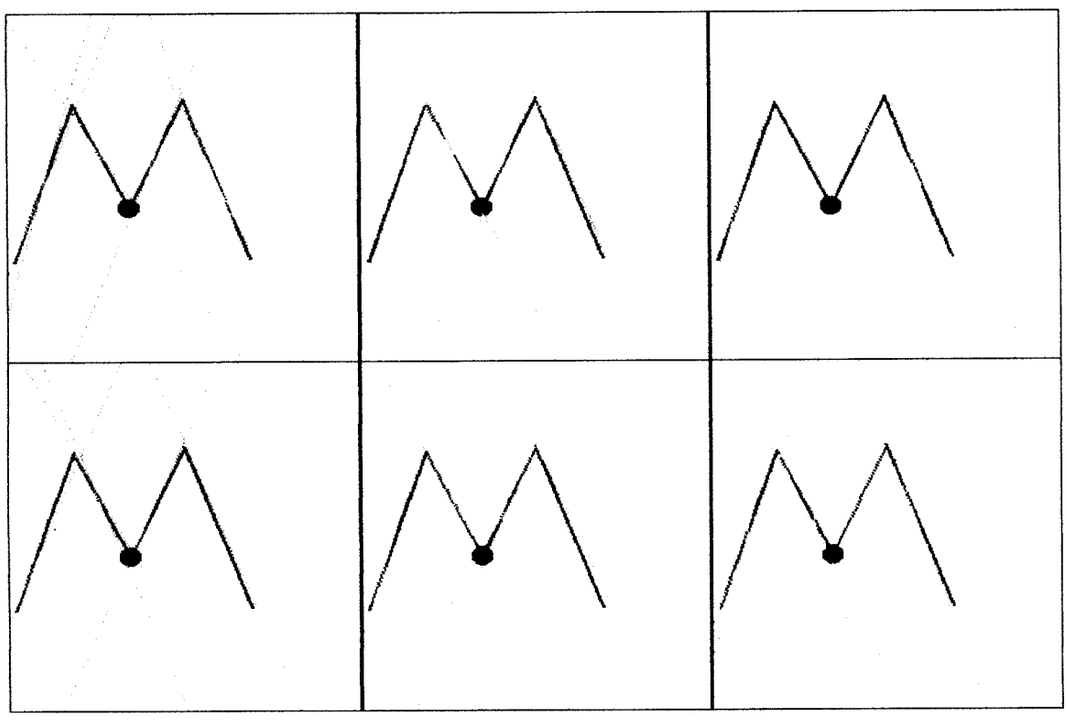


Figura 8.7 – Retas detectadas sobre os perfis de aranha, K=40000.

BIBLIOTECA

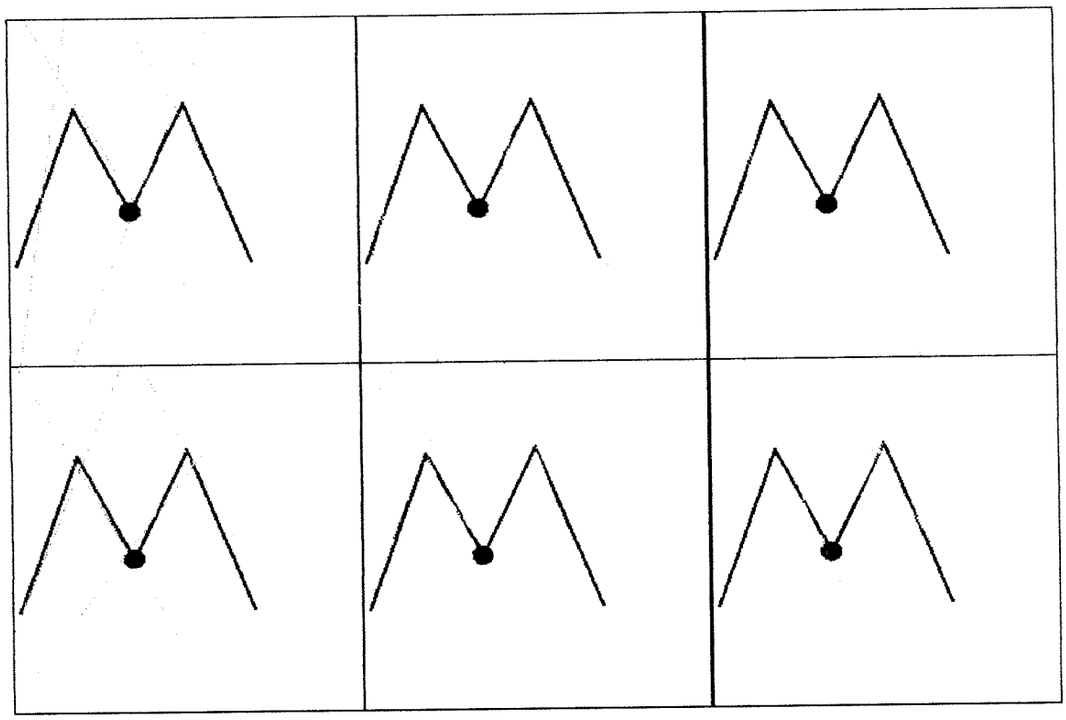


Figura 8.8 – Retas detectadas sobre os perfis de aranha, K=4000.

Quanto aos tempos de execução, com a utilização dos conjuntos de parâmetros obtidos para K=40000, o tempo necessário para o processamento foi aproximadamente três vezes maior que o utilizado pelos algoritmos com parâmetros escolhidos para K=4000, como mostra a Tabela 13.

	Tempo (ms) K=4000	Tempo (ms) K=40000
1	29537	73048
2	30203	87901
3	19428	69153
4	31037	90519
5	28950	80827

Tabela 13 – Tempos de execução da detecção de retas em perfis de aranha.

Para a detecção de retas na imagem da figura 8.4, com um algoritmo resultando da seleção com K=4000, os resultados estão na Figura 8.9.

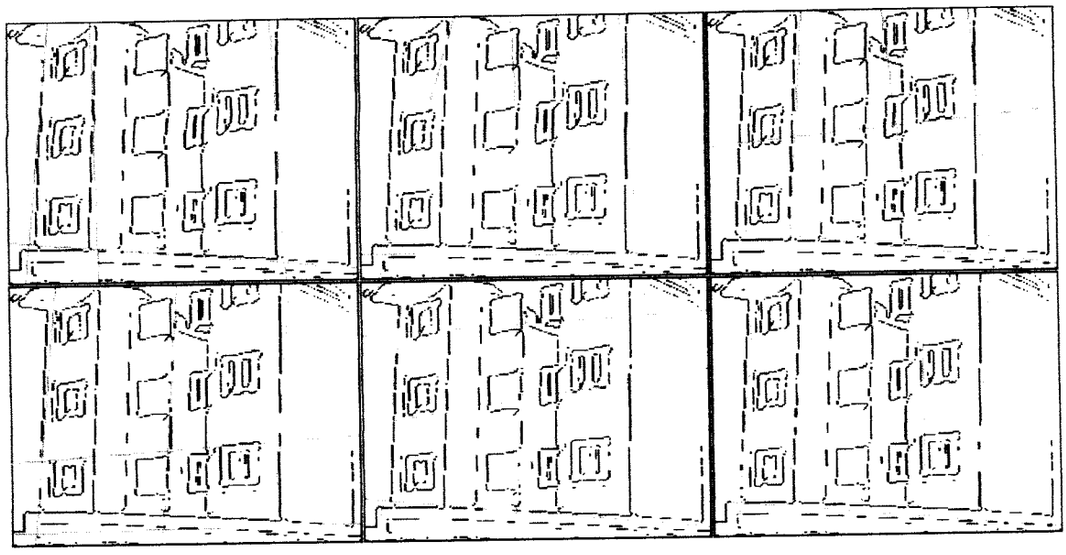


Figura 8.9 – Retas traçadas sobre contornos da Figura 8.4, para K=4000.

Podemos notar que muitas retas não foram encontradas, e algumas retas foram encontradas em posições erradas, provavelmente considerando pontos de diversas retas como sendo parte da mesma. A Figura 8.10 mostra alguns resultados obtidos para esta mesma imagem, para K=40000.

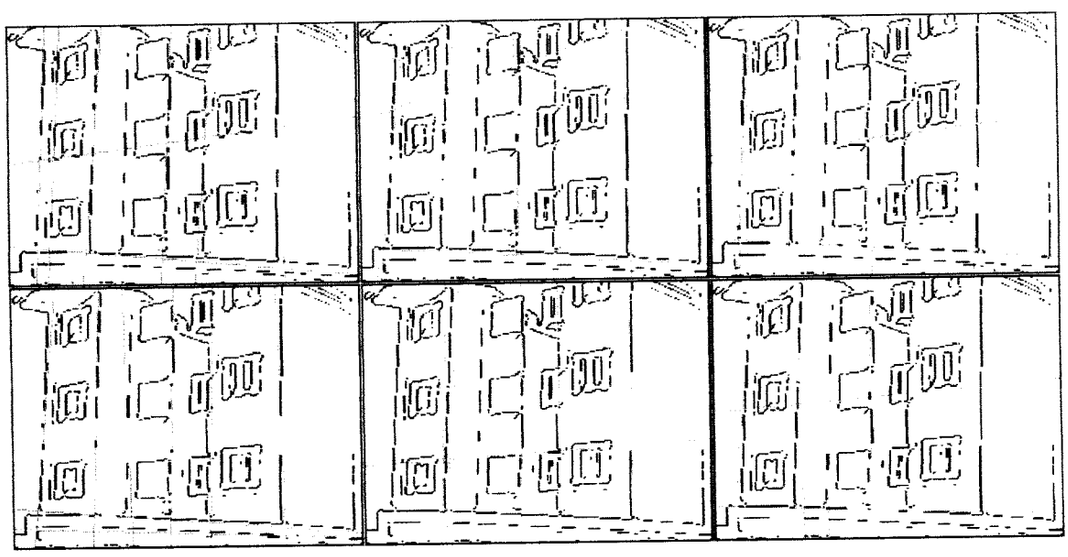


Figura 8.10 – Retas traçadas sobre contornos da Figura 8.4, para K=40000.

Nos resultados da Figura 8.10, algumas retas ainda deixaram de ser detectadas, mas as que foram detectadas estão mais próximas a retas que estão presentes nos contornos da figura. Algumas retas foram detectadas mais de uma vez, o que resultou em retas muito próximas.

Além da medida dos erros e tempos de processamento, outra medida interessante utilizada foi a de "energia" gasta na busca por retas. Essa energia é calculada através das distâncias percorridas pela janela linear em ρ e θ. A Tabela 14 mostra a movimentação da janela linear em ρ e θ, assim como os tempos de processamento para algumas configurações de "simulated annealing" e para a otimização através do método da descida de gradiente.

Método	Movimento em ρ	Movimento em θ	Tempo de process.
Simplex + annealing melhor para k=4000	6.9257	0.5361	2538
Simplex + annealing melhor para k=40000	14.3288	1.0459	8255
Descida de gradiente	17.3381	1.2502	11306

Tabela 14 - Movimento da janela linear para o "downhill simplex" e descida de gradiente

8.4 – Protótipo

Alguns resultados foram também obtidos através do protótipo. A principal diferença entre a varredura com uma janela linear simulada por computador e a varredura com a utilização do protótipo é que os passos em ρ e θ, no caso do protótipo, têm uma resolução fixa, enquanto a janela linear pode ter uma maior ou menor resolução de posicionamento, de acordo com a necessidade. Por isso, o espaço de parâmetros gerado pelo protótipo possui menor resolução.

O conjunto de parâmetros utilizado pelo método de busca do protótipo foi o que obteve melhores resultados para a detecção de retas no perfil de aranha, para K=40000.

As imagens analisadas com o protótipo não tiveram suas bordas detectadas por um filtro sofisticado, e sim pela presença da cor vermelha. Portanto, as imagens utilizadas foram simples, e com a presença de elementos vermelhos. A Figura 8.11 mostra alguns resultados obtidos com o protótipo.

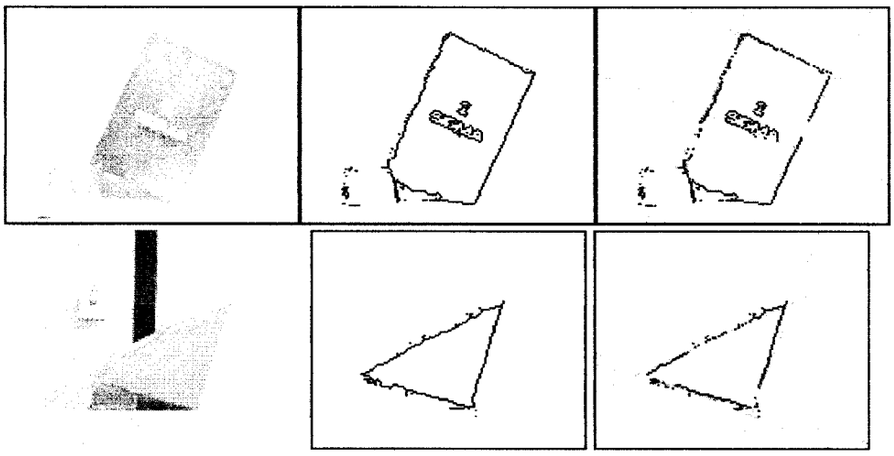


Figura 8.11 – Resultados obtidos com o protótipo.

9 – Conclusões e pesquisa futura

Os sistemas biológicos utilizam mecanismos que precisam ser eficazes em um ambiente de evolução, e seleção natural, contínuos. Nada mais lógico, então, do que utilizar esses mecanismos como inspiração para resolver problemas que confrontamos. Este trabalho tem como inspiração o sistema de visão das aranhas saltadoras, o melhor dentre os invertebrados terrestres.

Baseado neste sistema de visão, uma abordagem por otimização foi proposta para a detecção de retas, uma abordagem com grande potencial, e até então inexplorada. Este tipo de abordagem utiliza métodos de otimização para encontrar os máximos em um espaço de parâmetros, onde retas são mapeadas em picos. Esta característica do mapeamento de retas em picos está presente na transformada de Hough, com a qual foi traçado um paralelo.

A utilização de um método baseado em janelas lineares requer uma melhor compreensão das retas digitais, e de uma maneira mais geral, dos elementos retos digitais, incluindo bandas digitais, entre outros. Este trabalho revê os conceitos de retas digitais, e expande estes conceitos para melhor acomodar as imperfeições inerentes a retas no mundo real.

A parametrização da imagem em um espaço ρ-θ é um processo que, apesar de matematicamente simples, envolve diversas peculiaridades por ocorrer entre um espaço discreto e outro contínuo. A cada reta digital na imagem, corresponde não um ponto, mas uma região do espaço de parâmetros. Obtendo-se os limites de cada uma dessas regiões, podemos ter uma visão mais clara das tesselações formadas por estas regiões no espaço de parâmetros, e assim avaliar melhor problemas de sub e superamostragem. Isto também nos permite a compreensão do problema de busca de retas como um caso de otimização, onde mínimos locais podem aparecer.

Escolhido um método de otimização, como o "Downhill Simplex" de Nelder e Mead, resta avaliar se seu desempenho é adequado para o problema em questão, e adaptá-lo às necessidades da aplicação. Neste contexto, surgem os recursos de "simulated annealing", para lidar com máximos locais; e de largura adaptativa da janela linear, para a suavização da superfície do espaço de parâmetros e contorno dos problemas criados pelos patamares criados pela natureza digital deste espaço.

Para que estes recursos fossem eficazes, tornou-se necessário escolher um conjunto adequado de parâmetros que regeriam o funcionamento do método de detecção de retas. Por se tratar de um espaço multidimensional complexo, foi utilizado um algoritmo genético, uma abordagem também de inspiração biológica, para encontrar um bom conjunto destes parâmetros.

Além das medidas dos erros e tempos de processamento,foi também utilizada a "energia" gasta na busca por retas. Esta é calculada através das distâncias percorridas pela janela linear em ρ e θ, e corresponde aproximadamente a uma medida de gasto de energia por parte da aranha saltadora para a movimentação de seus olhos. Tendo comparado a energia utilizada pelo método de "Downhill Simplex" com a utilizada por outros métodos de otimização, como o "Steepest Descent" e o método de Powell, esta é bem menor que a dos demais, além de ser mais rápida. Isso indica que talvez o método de varredura utilizado pelas aranhas saltadoras seja semelhante ao método de otimização proposto por Nelder e Mead, que inclusive apresenta maior simplicidade algorítmica.

A utilização de janelas lineares de diversas larguras, que facilita a busca por retas, pode também ter analogia com a morfologia da retina das aranhas saltadoras, que é constituída por quatro camadas distintas, com larguras decrescentes, sendo a mais larga a primeira camada. Como no caso das janelas com largura adaptativa, talvez as aranhas saltadoras realizem uma análise multi-escala das imagens em seu campo visual.

Como a técnica desenvolvida para detecção de retas é complexa, uma maior quantidade de resultados seria necessária para se avaliar completamente as limitações e

vantagens desta. Não obstante, o método é eficaz para a detecção de retas em alguns casos testados, e pode ser utilizado em situações práticas.

Uma adição de grande importância a ser pesquisada para este método é um meio de se determinar os pontos iniciais e finais dos segmentos de reta detectados, o que poderia ser feito através de uma análise utilizando uma janela linear com largura suficiente para se verificar essas características. A implementação de outros protótipos, menores e mais velozes, também poderá viabilizar a utilização deste método de detecção de retas em plataformas móveis e dedicadas.

10 - Bibliografia

[ARCELLI & MASSAROTTI 75] ARCELLI, C. and MASSAROTTI, A. "Regular Arcs in Digital Contours" IN: **Computer Graphics and Image Processing.** Vol. 4, 1975. (pp. 339 - 360)

[BRONS 74] BRONS, R. "Linguistic Methods for the Description of a Straight Line on a Grid" IN: **Computer Vision, Graphics and Image Processing.** Vol. 3, 1974. (pp. 48 - 62)

[CHATZIS 96] CHATZIS, V. and PITAS, I., "Fuzzy Cell Hough Transform" IN: **Proceedings of EUSIPCO-96**. Vol. 3, Sep 1996. (pp. 1892 - 1898)

[CHATZIS 97] CHATZIS, V. and PITAS, I., "Randomized Fuzzy Cell Hough Transform" IN: **Proceedings of FUZZ IEEE ' 97**, 1997. (pp. 1185 - 1189)

[CORANA 87] CORANA, A., MARCHESI, M. and RIDELLA, S. "Minimizing Multimodal Functions of Continuous Variables with the "Simulated Annealing" Algorithm" IN: **ACM Transactions on Mathematical Software**, Vol. 13, n. 3, Sep 1987. (pp. 262 - 280)

[COSTA 92] COSTA, L.F., **Effective Detection of Line Segments with Hough Transform.** Ph.D. Thesis, King's College, University of London, 1992.

[COSTA 94] COSTA L.F., "On jumping Spiders and Hough Transform" IN: **Systems and Control 94 - IASTED.** Lugano, Switzerland, Jun 1994. (pp. 138 - 141)

[COSTA 95] COSTA, L.F., " Towards Real - Time Detection of Discrete Straight Features with a Hybrid Technique Based on Preliminary Curve Segmentation and Zoomed - Adaptive Parametric Mapping" IN: **Real - Time Imaging.** N.1, 1995. (pp. 203 - 213)

[DUDA 72] DUDA, R.O. and HART, P.E. "Use of the Hough Transformation to Detect Lines and Curves in Pictures" IN: **Communications of the ACM.** Vol. 15, n. 1, jan 1972. (pp. 11 - 15)

[FORSTER 82] FORSTER, L., "Visual Communication in Jumping Spiders" IN: **Spider Communication.** Princeton University Press, New Jersey, 1982.

[FORSTER 85] FORSTER, L., "Target Discrimination in Jumping Spiders (Araneae: Salticidae)" IN: **Neurobiology of Arachnids.** Springer - Verlag, Heidelberg, 1985.

[FREEMAN 70] FREEMAN, H., "Boundary Encoding and Processing" IN: **Picture Processing and Psychopictorics.** Academic Press, New York/London, 1970. (pp. 241 - 266)

[FREEMAN 74] FREEMAN, H., "Computer Processing of Line-Drawing Images" IN: **Computing Surveys**, vol. 6, n. 1, march 1974. (pp. 57 - 97)

[FUNG 96] FUNG, P.F., LEE, W.S., and KING, I., "Randomized Generalized Hough Transform for 2-D Grayscale Object Detection" IN: **Proceedings of ICPR'96.** 1996. (pp. 511 - 515)

[GOLDBERG 89] GOLDBERG, D.E. **Genetic Algorithms in Search, Optimization and Machine Learning** , Addison-Wesley Publishing Company, 1989.

[GONZALEZ 87] GONZALEZ, R.C. and WINTZ, P., **Digital Image Processing.** Addison - Wesley Publishing Co., 1987.

[HOLLAND 75] HOLLAND, J.H. **Adaptation in Natural and Artificial Systems**, University of Michigan Press, 1975.

[HORIKI 98] HORIKI, S., WATANABE, T., KIMURA, A. and TAYAMA, N., "Improvement of Combinatorial Hough Transform to Compensate Sampling Errors of Digital Lines" IN: **Systems and Computers in Japan.** Vol. 29, n. 2, 1998. (pp. 57 - 67)

[HOUGH 62] HOUGH, P.V.C., **Method and Means for Recognizing Complex Patterns.** U.S.Patent 3.069.654, Dec 1962.

[HUBEL 88] HUBEL, D.H., **Eye, Brain and Vision.** Scientific American Library, New York, 1988.

[ILLINGWORTH 87] ILLINGWORTH, J. and KITTLER, J. "The Adaptive Hough Transform" IN: **IEEE Transactions on Pattern Analysis and Machine Intelligence.** Vol. 9, n. 5, sep 1997. (pp. 690 - 698)

[JAIN 89] JAIN, A. K., **Fundamentals of Digital Image Processing.** Prentice-Hall, Englewood Cliffs, 1989.

[KIM & ROSENFELD 82] KIM, C.E. and ROSENFELD, A. "Digital Straight Lines and Convexity of Digital Regions" IN: **IEEE Transactions on Pattern Analysis and Machine Intelligence.** Vol. 4, n. 2, mar 1992. (pp. 149 - 153)

[KIRKPATRICK 83] KIRKPATRICK, J., GELATT, C.D. and VECCHI, M.P. "Optimization by Simulated Annealing" IN: **Science**, Vol. 220, n.4598, May 1983. (pp. 671 - 680)

[KIRYATI 91] KIRYATI, N., ELDAR, Y. and BRUCKSTEIN, A. M. "A Probabilistic Hough Transform" IN: **Pattern Recognition.** Vol. 24, n.4, 1991. (pp. 303 - 316)

[LAND 85] LAND, M.F., "Morphology and Optics of Spider Eyes" IN: **Neurobiology of Arachnids.** Springer - Verlag, Heidelberg, 1985.

[LEAVERS 88] LEAVERS, V.F. **Shape Parametrisation and Object Recognition in Machine Vision.** Ph.D Thesis, King's College, London, UK, jun 1988.

[MARR 82] MARR, D. **Vision**, W. H. Freeman and Company, New York, 1982.

[MELTER 93] MELTER, R.A., STOJMENOVIC, I. and ZUNIC, J. "A New Characterization of Digital Lines by Least Square Fits" IN: **Pattern Recognition Letters.** Vol. 14, n.2, 1993. (pp. 83 - 88)

[MERLIN 75] MERLIN, P.M. and FARBER, D.J., "A Parallel Mechanism for Detecting Curves in Pictures" IN: **IEEE Transactions on the Computers.** Vol. 24, jan 1975. (pp. 96 - 98)

[METROPOLIS 53] METROPOLIS, N., ROSENBLUTH, A., ROSENBLUTH, M., TELLER, A. and TELLER, E. "Equation of State Calculations by Fast Computing Machines" IN: **Journal of Chemical Physics**, Vol. 21, 1953. (pp. 1087 - 1090)

[MITCHELL 98] MITCHELL, M. **An Introduction to Genetic Algorithms**, The MIT Press, Massachusetts, 1998.

[NELDER 65] NELDER, J.A. and MEAD, R. "A simplex Method for Function Minimization" IN: **The Computer Journal**, 1965. (pp. 308 - 313)

[PRESS 92] PRESS, W.H., TEUKOLSKI, S.A., VETTERLING, W.T. and FLANNERY, B.P. **Numerical Recipes in C**, Cambridge University Press, 1992.

[PRINCEN 90] PRINCEN, J., ILLINGWORTH, J. and KITTLER, J. "A Hierarchical Approach to Line Extraction Based on the Hough Transform" IN: **Computer Vision, Graphics and Image Processing.** Vol 52, 1990. (pp. 57 - 77)

[RAVEN 96] RAVEN, P. H. and JOHNSON, G. B. **Biology**, WCB/McGraw-Hill, 1996.

[ROSENFELD 74] ROSENFELD, A. "Digital Straight Line Segments" IN: **IEEE Transactions on Computers.** Vol. 23, n. 12, Dec. 1974. (pp. 1264 - 1269)

[SHAKED 95] SHAKED, D., YARON, O. and KIRYATI, N. "Deriving Stopping Rules for the Probabilistic Hough Transform by Sequential Analysis" IN: **Computer Vision and Image Understanding.** Vol. 63, n. 3, may 1996. (pp. 512 - 526)

[SANDLER 90] SANDLER, M.B. and BEN - TZVI, D. "A Combinatorial Hough Transform" IN: **Pattern Recognition Letters.** Vol. 11, 1990. (pp. 167 - 174)

[STEPHENS 90] STEPHENS, R.S. "A Probabilistic Approach to Hough Transform" IN: **BMVC.** Oxford, UK, sep 1990. (pp. 55 - 60)

[VANDERBILT 84] VANDERBILT, D. and LOUIE, S. G. "A Monte Carlo Simulated Annealing Approach to Optimization over Continuous Variables" IN: **Journal of Computational Physics**, Vol. 56, 1984. (pp. 259 - 271)

[VECCHI 83] VECCHI, M.P. and KIRKPATRICK, S. "Global Wiring by Simulated Annealing" IN: **IEEE Transactions on Computer-Aided Design**, Vol. 2, n. 4, Oct 1983. (pp. 215 - 222)

[WU 82] WU, L.D. " On the Chain Code of a Line" IN: **IEEE Transactions on Pattern Analysis and Machine Intelligence.** Vol. 4, n. 3, may, 1982. (pp. 347 - 353)

[YLÄ-JÄÄSKI 94] YLÄ-JÄÄSKI, A. and KIRYATI, N. "Adaptive Termination of Voting in the Probabilistic Circular Hough Transform" IN: **IEEE Transactions on Pattern Analysis and Machine Intelligence.** Vol. 16, n. 9, sep 1994. (pp. 911 - 915)

[ZEKI 93] ZEKI, S., **A Vision of the Brain.** Blackwell Scientific Publications, London, 1993.