Campus de São Carlos

"Aplicação da Transformada de
Hough em Inspeção Visual Automática"

Autor:

Ernany Palanaguá da Silva

Orientador:

Prof. Dr. Adilson Gonzaga

UNIVERSIDADE DE SÃO PAULO

**ESCOLA DE ENGENHARIA
DE SÃO CARLOS**

✓

UNIVERSIDADE DE SÃO PAULO
ESCOLA DE ENGENHARIA DE SÃO CARLOS
DEPARTAMENTO DE ENGENHARIA ELÉTRICA

**APLICAÇÃO DA TRANSFORMADA DE
HOUGH EM INSPEÇÃO VISUAL AUTOMÁTICA**

DEDALUS - Acervo - EESC

31100036390

AUTOR : Eng. ERNANY PARANAGUÁ DA SILVA
ORIENTADOR : Prof. Dr. ADILSON GONZAGA

Dissertação apresentada à Escola de Engenharia de São Carlos, da Universidade de São Paulo, como parte dos requisitos para obtenção do Título de Mestre em Engenharia Elétrica.

São Carlos
1996

Class.	Tese - EESC
Cut.	S543
Tombo	033/96

Eletricidade

it 0744157

Ficha catalográfica preparada pela Seção de Tratamento
da Informação do Serviço de Biblioteca - EESC-USP

S586a Silva, Ernany Paranaguá da
Aplicação da transformada de Hough em inspeção
visual automática / Ernany Paranaguá da Silva. -- São
Carlos, 1996.
199p.

Dissertação (Mestrado) -- Escola de Engenharia de
São Carlos-Universidade de São Paulo, 1995.
Orientador: Prof. Dr. Adilson Gonzaga

1. Transformada de Hough. 2. Inspeção visual
automática. I. Título

FOLHA DE APROVAÇÃO

Dissertação defendida e aprovada em 15-2-1996
pela Comissão Julgadora:

Prof. Doutor ADILSON GONZAGA - Orientador
(Escola de Engenharia de São Carlos - Universidade de São Paulo)

Prof. Doutor VALENTIN OBAC RODA
(Instituto de Física de São Carlos - Universidade de São Paulo)

Profª Doutora MARIA STELA VELUDO DE PAIVA
(Escola de Engenharia de São Carlos - Universidade de São Paulo)

Presidente da Comissão de Pós-Graduação
Prof. Dr. EDUARDO CLETO PIRES

Coordenador da Área de Engenharia Elétrica
Prof. Dr. GERALDO ROBERTO M. DA COSTA

"Sentimentos Fraternos"

Forte contra-senso que desorganiza a contribuição humana, no divino edifício do Cristianismo, é o impulso sectário que atormenta enormes fileiras de seus seguidores.

Mais reflexão, mais ouvidos aos ensinamentos de Jesus e essas batalhas injustificáveis estariam para sempre apagadas.

Ainda hoje, com as manifestações do plano espiritual na renovação do mundo, a cada momento surgem grupos e personalidades, solicitando fórmulas do Além para que se integrem no campo da fraternidade pura.

Que esperam, entretanto, os companheiros esclarecidos para serem efetivamente irmãos uns dos outros?

Muita gente se esquece de que a solidariedade legítima escasseia nos ambientes onde é reduzido o espírito de serviço e onde sobra a preocupação de criticar. Instituições notáveis são conduzidas à perturbação e ao extermínio, em vista da ausência do auxílio mútuo, no terreno da compreensão, do trabalho e da boa-vontade.

Falta de assistência? Não.

Toda obra honesta e generosa repercute nos planos mais altos, conquistando cooperadores abnegados.

Quando se verifique a invasão da desarmonia nos institutos do bem, que os agentes humanos acusem a si mesmos pela defecção nos compromissos assumidos ou pela indiferença ao ato de servir. E que ninguém peça ao Céu determinadas receitas de fraternidade, porque a fórmula sagrada e imutável permanece conosco no amai-vos uns aos outros."

Pelo Espírito Emmanuel

*Dedico este trabalho à minha Esposa Ângela,
pelo incentivo, compreensão e amor sempre recebidos.*

AGRADECIMENTOS

Ao Professor Adison Gonzaga pela orientação fornecida durante a elaboração deste trabalho.

Aos Colegas Celso, Evelina, Paulo Bentes, Prof^o Evandro, Prof^a Stela, Prof^o Jerson e tantos outros que durante esses anos de mestrado participaram do meu esforço.

Agradeço em especial aos meus pais que mesmo de longe sempre estiveram presente, apoiando e vibrando para que eu vencesse.

A todos os professores e funcionários do Departamento de Engenharia Elétrica da EESC/SC pela amizade.

SUMÁRIO

LISTA DE FIGURAS.....	i
LISTA DE TABELAS.....	v
RESUMO.....	vi
ABSTRACT.....	vii
1 INTRODUÇÃO.....	1
2 - INSPEÇÃO VISUAL AUTOMÁTICA.....	4
2.1 - Introdução.....	4
2.2 - Modelo Básico de IVA.....	5
2.2.1 - Aquisição de Imagens.....	5
2.2.2 - Técnicas Para Análise de Imagens.....	7
2.2.2.1 - Extração de Características.....	8
2.2.2.2 - Modelagem.....	9
2.2.2.3 - Detecção.....	9
2.3 - Processamento de Imagens.....	10
2.3.1 - Método de Segmentação.....	10
2.3.2 - Método de "Thresholding".....	11
2.3.3 - Subtração de Imagem.....	11
2.3.4 - Combinação de Características.....	12
2.3.5 - Técnica de Identificação por Cor.....	12
2.3.6 - Técnica da Transformada de Hough.....	13
2.4 - Aplicações de Inspeção Visual Automática.....	14
2.4.1 - Visão de Máquina na Indústria Eletrônica.....	14
2.4.2 - Avaliação da Qualidade de Superfícies Metálicas.....	19
2.4.3 - Aplicação em Manutenção de Linhas de Transmissão.....	20
2.4.4 - Controle de Qualidade em Tecelagem.....	21
2.4.5 - Controle de Qualidade na Indústria Alimentícia e na Agricultura.....	22

2.4.6 - Inspeção de cápsulas com uma Arquitetura Data Flow para IVA.....	23
2.4.7 - Aplicação da HT na Interpretação de Linhas e Segmentos.....	24
3 - A TRANSFORMADA DE HOUGH.....	25
3.1 - Introdução	25
3.2 - Fundamentos da Transformada de Hough.....	26
3.3 - Principais Modelos de Transformadas de Hough.....	30
3.3.1 - Transformada de Hough Generalizada.....	30
3.3.2 - Transformada de Hough Rápida.....	32
3.3.3 - Transformada de Hough Adaptativa.....	35
3.3.4 - Transformada de Hough Combinatorial.....	37
3.3.5 - Transformada de Hough Fuzzy.....	41
3.3.6 -Sistema de HT baseado em Redes Neurais.....	44
3.4- .Aplicações e uso da HT na detecção automatizada de traços.....	45
3.4.1 - Sensoriamento Remoto.....	45
3.4.2 - Sistema de Rastreamento com Radares.....	49
3.4.3 -Enfoque hierárquico para extração de linha baseado na HT	49
3.4.4 - Localização e reconhecimento de múltiplas imagens com HT.....	51
3.4.5 - Aplicações em Imagens Ruidosas.....	52
3.5 - Implementação em Multiprocessadores com o Uso da HT.....	53
4 - APLICAÇÃO DA HT PARA ANÁLISE DE PLACAS RETANGULARES DE MADEIRA.....	55
4.1 - Sistema de Inspeção.....	56
4.2 - Análise da Imagem.....	57
4.3 - Filtragem do Arranjo Acumulador.....	66
4.4 - Determinação de Características.....	71
4.4.1 - Cálculo dos lados, perímetro e área.....	71
4.4.2 - Cálculo do Centro de Massa.....	73

4.5 - Conclusão.....	78
5 - RESULTADOS E CONCLUSÕES.....	79
5.1 - Análise da HT.....	79
5.2 - Método dos Momentos.....	82
5.3 - Comparação do Método HT e Momentos com Enfoque Estatístico.....	83
5.3.1 - Teste t de Student Pareado e Não Pareado.....	84
5.3.2 - Modelo da Placa Inspeccionada.....	87
5.3.3 - Análise Estatística.....	88
5.3.3.1 - Imagens Rotacionadas.....	88
5.3.3.2 - Imagens Transladadas na Horizontal.....	91
5.3.3.3 - Imagens Transladadas na Vertical.....	94
5.3.3.4 - Análise das Coordenadas de Centro de Massa...	97
5.3.3.5 - Análise dos Cálculos do Prímetro da HT com o valor real.....	102
5.4 - Conclusões Finais.....	103
5.5 - Sugestões para Trabalhos Futuros.....	104
ANEXOS.....	105
REFERÊNCIAS BIBLIOGRÁFICAS.....	191

LISTA DE FIGURAS

Figura 2.1 - Componentes de um sistema de IVA.....	5
Figura 2.2 - (a) Iluminação de fundo; (b) Iluminação direcional; (c) Iluminação difusa; (d) Iluminação de faixa.....	6
Figura 2.3 - Modelo de subtração de imagem.....	12
Figura 2.4 - Localização do centro do círculo através da HT.....	14
Figura 2.5 - Tipos de defeitos.....	15
Figura 2.6 - Sistema de aquisição de imagem.....	16
Figura 2.7 - Típicos defeitos de fiação e conexão.....	16
Figura 2.8 - Algoritmo do detector de borda.....	17
Figura 2.9 - Detecção dos defeitos nos fios.....	18
Figura 2.10 - Exemplo típico de classes de defeitos.....	19
Figura 2.11 - Plataforma de inspeção	20
Figura 2.12 - Modelo de renda.....	21
Figura 3.1 - Imagem de bordas.....	25
Figura 3.2 - A transformada de Hough. (a) Espaço da Imagem (b) Espaço de Parâmetro.....	27
Figura 3.3 - (a) Representação da equação da reta em função de suas coordenadas polares. (b) Relação entre definição de reta em coordenadas polares.....	28
Figura 3.4 - Ilustração da HT. (a) Pontos no espaço da imagem. (b) Mapeamento dos pontos no espaço(ρ - θ). (c) Detecção das propriedades de colinearidade. Ponto A denota intersecção de curvas correspondentes aos pontos 1, 3 e 5 no plano x-y e similarmente ponto B corresponde aos pontos colineares 2, 3 e 4.(d) Propriedade reflectiva da HT.....	29
Figura 3.5 - Usando convolução para compensação de erros na GHT.....	31
Figura 3.6 - Geometria da Transformada de Hough Generalizada.....	32
Figura 3.7 - Exemplo de um processo de FHT.....	33
Figura 3.8 - Detecção de segmentos lineares usando a AHT. (a) Imagem	

(b) Espaço de parâmetros (limites $m = -1$ a 1 , $c = -256$ a 256).	
(c) Acumulador grosseiramente quantizado.....	36
Figura 3.9 - Detecção do centro do círculo usando a AHT. (a) Imagem.	
(b) Espaço de parâmetros (limites $a = 0$ a 128 , $b = 0$ a 128).	
(c) Acumulador grosseiramente quantizado.....	37
Figura 3.10 - (a) Linha com cinco pontos e seu respectivo arranjo acumulador. (b) O efeito de um ruído sobre o espaço da Transformada.....	39
Figura 3.11 - Três tipos de imagens utilizadas na comparação da CHT com a HT.....	40
Figura 3.12 - Pontos para detecção de linhas.....	41
Figura 3.13 - Quatro linhas encontradas com quatro pontos cada.....	42
Figura 3.14 - Na distribuição do arranjo, os parâmetros da linha central tem o maior valor de acumuladores que qualquer ponto no espaço de parâmetro.....	42
Figura 3.15 - Imagem do LANDSAT (512 x 512 pixels). Área aproximada de 15 por 15 Km.....	45
Figura 3.16 - Imagem após ser submetida ao detector de bordas.....	46
Figura 3.17 - As imagens foram geradas com os seguintes parâmetros:	
(a) lacuna $sg = 40$, “threshold” para seleção de máximos locais $T = 80$. (b) $sg = 40$, $T = 75$. (c) $sg = 40$ $T = 70$ e	
(d) $sg = 50$, $T = 75$	47
Figura 3.18 - As subdivisões de uma imagem.....	52
Figura 3.19 - Arquitetura da distribuição de memória de imagem (MI) e memória da HT (MHT).....	54
FIGURA 4.1 - Elementos de Inspeção.....	56
FIGURA 4.2 - Diagrama de Bloco do Sistema de Inspeção.....	57
FIGURA 4.3 - Placa de madeira usada na inspeção.....	58
FIGURA 4.4 - Histograma.....	58
FIGURA 4.5 - Três pontos em uma imagem.....	59
FIGURA 4.6- (a) Imagem Original; (b) Aplicando o detector de bordas....	60

FIGURA 4.7 - Representação geométrica de Hough.....	61
FIGURA 4.8 - Arranjo Acumulador (a) Diagrama de blocos (b) Construção da Tabela.....	62
FIGURA 4.9 - (a) Gráfico do arranjo acumulador ;(b) Projeção no plano imagem.....	63
FIGURA 4.10 - Intervalos de quantização.....	63
FIGURA 4.11 - Gráficos de Arranjos Acum. com valores diferentes de $\Delta\rho$ e $\Delta\theta$. (a) $\Delta\rho$ e $\Delta\theta = 20$, (b) $\Delta\rho$ e $\Delta\theta = 10$; (c) $\Delta\rho$ e $\Delta\theta = 5$, (d) $\Delta\rho$ e $\Delta\theta = 3$; (e) $\Delta\rho$ e $\Delta\theta = 2$ e (f) $\Delta\rho$ e $\Delta\theta = 1$	65
FIGURA 4.12 -(a) Placa de madeira analisada; (b) Imagem após o detector de bordas; (c) Gráfico referente ao arranjo acumulador; (d) Gráfico nº. de pontos por eixo- ρ ; (e) Gráfico nº. de pontos por eixo- θ	66
FIGURA 4.13 - (a) Arranjo acumulador após filtragem grosseira, (b)Projeção no plano imagem.....	67
FIGURA 4.14 - Inclinação entre as retas.....	69
FIGURA 4.15 - Projeção do arranjo acumulador do grupo III da Tabela4.1.....	70
FIGURA 4.16 - (a) Arranjo acumulador após filtragem fina (b) Figura projetada no plano imagem.....	71
FIGURA 4.17 - (a) Gráfico de nº de pontos por eixo - ρ do arranjo acumulador; (b) Gráfico de nº de pontos por eixo - θ ; (c)Aplicação da Transformada inversa da HT.....	72
FIGURA 4.18 - Projeção no plano imagem, tabela final do arranjo acumulador e cálculo dos segmentos " $\overline{AB}$ " e " $\overline{BC}$ ". (a) Figura 23 onde $ \theta_1 \text{ ou } \theta_2 < 45^\circ$ (ver anexo E), (b) Figura 33 $ \theta_1 \text{ ou } \theta_2 > 45^\circ$ (ver anexo E).....	74
FIGURA 4.19 - (a) Cálculo para imagens a 90° e a 0° , admitindo a 1ª condição ; (b) Cálculo para imagens quando $ \theta_1 $ ou $ \theta_2 $ for igual a 45°	75
FIGURA 4.20 - Fluxograma do cálculo do Centro de Massa.....	76

FIGURA 4.21 - Ampliação dos triângulos ABC da Fig.4.19(a) e (b) e cálculo dos ângulos das Resultantes, respectivamente.....	77
FIGURA 5.1 - Imagens aplicando o Detector de bordas e a HT (ver imagens nos Anexos D, A e F respectivamente).....	80
FIGURA 5.2 -Imagens dos valores de θ (a) Anexo F-32; (b) Anexo B-24.	81
FIGURA 5.3 - Gráfico de uma distribuição normal com intervalo de confiança de 95%.....	85
FIGURA 5.4 - Dimensões da Placa de Madeira Inspeccionada.....	87
FIGURA 5.5 - Gráfico das 37 imagens sem ruído da placa de madeira rotacionadas , usando o método da HT e dos Momentos, para o cálculo da área.....	90
FIGURA 5.6 - Gráfico das 37 imagens com ruído da placa de madeira rotacionadas , usando o método da HT e dos Momentos, para o cálculo da área.....	91
FIGURA 5.7 - Gráfico das 31 imagens sem ruído da placa de madeira transladada na horizontal , usando o método da HT e dos Momentos para o cálculo da área.....	93
FIGURA 5.8 - Gráfico das 31 imagens com ruído da placa de madeira transladada na horizontal , usando o método da HT e dos Momentos para o cálculo da área.....	94
FIGURA 5.9 - Gráfico das 31 imagens sem ruído da placa de madeira transladada na vertical , usando o método da HT e do Momentos para o cálculo da área.....	96
FIGURA 5.10 - Gráfico das 31 imagens com ruído da placa de madeira transladada na vertical , usando o método da HT e do Momentos.para o cálculo da área.....	97
FIGURA 5.11 - Gráfico das 31 imagens com e sem ruído da placa de madeira relacionanda com as coordenadas, usando o método da HT e do Momentospara o cálculo da área.....	101

LISTA DE TABELAS

TABELA 1 - Efeito da HT sobre uma linha reta no espaço m-c.....	26
TABELA 2 - Valores referentes ao exemplo do processo de FHT.....	34
TABELA 3 - Comparação do tempo de desempenho da HT com CHT.....	41
TABELA 4 - Efetividade dos plotes de traços comparados com o mapa Geográfico.....	49
TABELA 4.1 - Resultado da Filtragem Grosseira.....	68
TABELA 4.2 - Resultado da Filtragem Fina.....	71
TABELA 5.1 - Valores estatísticos relacionados com as imagens Rotacionadas.....	88
TABELA 5.2 - Valores estatísticos relacionados com as imagens Transladadas na horizontal.....	91
TABELA 5.3 - Valores estatísticos relacionados com as imagens Transladadas na vertical.....	94
TABELA 5.4 - Valores estatísticos relacionados com as coordenadas das imagens Rotacionadas sem ruído.....	98
TABELA 5.5 - Valores estatísticos relacionados com as coordenadas das imagens Rotacionadas com ruído.....	99
TABELA 5.6 - Valores estatísticos comparando o método da HT para o cálculo de perímetro das imagens inspecionadas com o valor real.....	103

RESUMO

Neste trabalho é proposto a técnica da Transformada de Hough para a Inspeção Visual Automática de placas retangulares, visando a determinação de características tais como perímetro, ortogonalidade, centro de massa, área, independente de rotação e translação do objeto dentro da imagem.

Os resultados obtidos por essa técnica são avaliados pela comparação com o método dos Momentos, o recurso mais utilizado para a determinação de área e centro de massa para imagens bi-dimensionais. A análise estatística dos dados mostra a robustez da técnica da Transformada Hough para imagens ruidosas.

Palavras-chave: Transformada de Hough; Inspeção Visual Automática.

ABSTRACT

In this work, is proposed the Hough transform method for the Automated Visual Inspection of rectangular boards. The aim is to extract features such as position, perimeter and area, despite of object translation and rotation.

The results obtained by this method are evaluated by comparing them with the method of Moments, the most used method to determine the area and center of mass for bi-dimensional images. Statistical analysis show the robustness of Hough transform for noisy images visual inspection.

Keywords: Hough transform; Automated Visual Inspection

CAPÍTULO 1 - INTRODUÇÃO

A operação de inspecionar envolve a comparação entre o produto real industrializado, ou parte dele, e suas características específicas padrões. A inspeção visual pode e tem sido realizada, em grande parte das empresas, de forma manual. Isto justifica, em alguns casos, a existência de departamentos inteiros dedicados à inspeção de produtos fabricados manualmente. Encontrar e classificar defeitos em peças fundidas, usinadas ou tratadas em indústrias mecânicas é uma área de ação que possui um grande número de pessoas trabalhando ainda manualmente em inspeção visual. Este é um local onde um sistema automático de inspeção pode aumentar sensivelmente o número de peças/dia inspecionadas em um sistema de produção. Como exemplo temos o citado em (KENT, 1986) onde um sistema de inspeção visual automática implantado em uma empresa de fabricação de carcaças de máquinas, reduziu o tempo de 20 horas para apenas 1 hora na inspeção de uma unidade. Quanto maior o nível de automação no processo de fabricação, maior a necessidade de alguma forma de monitoramento de qualidade automática e inspeção de objetos na produção. Dispositivos de monitoramento de qualidade automática são usados para determinar a aceitação ou rejeição de um objeto, e também para monitorar o posicionamento e o formato do objeto. Sem esses dispositivos, a efetividade de um sistema de fabricação automatizado será questionável.

Com o avanço da informática, mecânica e eletrônica nos últimos anos, cada vez mais novas técnicas dedicadas à inspeção tem sido desenvolvidas e a custos realmente compatíveis. Diversas características podem ser inspecionadas de forma automática utilizando-se recursos de digitalização de imagens e processamento da imagem digital. No entanto, várias restrições devem ser avaliadas na utilização de inspeção visual automática em indústrias (BATCHELOR, 1985). Estas restrições podem ser descritas como de caráter operacional, ambiental e intrínseca.

As restrições de caráter operacional, tanto técnicas quanto gerenciais, estão ligadas à necessidade de pessoal especializado para operação, manutenção e interpretação de todo o sistema. As de caráter ambiental estabelecem que equipamentos de inspeção visual não devem permanecer expostos a vibrações, sujeira, ataques químicos e outros efeitos próprios de ambientes industriais. Isto não quer dizer que esses equipamentos devam ficar em salas limpas, totalmente isolados do processo de produção. Algum cuidado é necessário para uma maior confiabilidade na inspeção.

As restrições intrínsecas estão relacionadas à capacidade e disponibilidade de equipamentos específicos para aplicações particulares. A precisão é o primeiro ponto considerado. Outros, tais como resolução, limite de operação, porcentagem de erro e repetições, são as características mais exigidas em um sistema de inspeção visual automática.

As vantagens, entretanto, superam as restrições quando o elemento custo final pode ser considerado. A informação de um produto pode ser armazenada e processada rapidamente. A velocidade da inspeção aumenta a velocidade da produção mesmo quando as peças sob inspeção são complexas. A integração entre a inspeção e o processo de produção permite um produto mais confiável.

Finalmente, os sistemas de inspeção visual automática permitem que os padrões de testes sejam consistentes e mais objetivamente aplicados do que quando inspecionados manualmente. Além disso, como as operações não possuem o contato humano, são menos perigosas tanto ao homem quanto ao produto a ser inspecionado.

O processo de inspeção visual automática tem sido motivo de pesquisas por vários autores. (FOSTER III et al. 1990); (CHIN, 1986); (ROSENFELD, 1985).

Como as aplicações de inspeção visual automática são inúmeras, dado a grande variedade de produtos industrializados, cada uma requer análise individual. O sistema proposto neste trabalho visa a determinação de características de objetos retangulares em uma esteira rolante. Estas

características permitirão a separação de peças defeituosas ou fora de especificação, aumentando assim a qualidade do produto final.

A teoria e as técnicas de "Reconhecimento de Padrões" podem ser empregadas, FU (1982), no sentido de automatizar, a níveis mais elevados, o sistema de inspeção visual. Uma técnica que demonstra grande robustez é a Transformada de Hough (HT) a qual, neste trabalho, é aplicada para a inspeção do controle de qualidade de pequenas placas de madeira.

O objetivo deste trabalho é utilizar os dados obtidos do arranjo acumulador da HT, para dar suporte a um sistema de inspeção visual automática. O estudo do arranjo acumulador permite a identificação de várias características de objetos retangulares, tais como a sua área, perímetro, ortogonalidade e centro de massa, durante um processo de inspeção.

No capítulo 2 são descritos algoritmos e técnicas para inspeção visual automática de produtos industriais. Discutem-se, também, alguns trabalhos relevantes publicados, mostrando suas aplicações, tais como técnicas de análise de imagem, estratégias de detecção e tópicos de implementações no hardware para aquisição da imagem. No capítulo 3 é apresentado um estudo sobre a Transformada de Hough, suas principais derivações e aplicações, vantagens e desvantagens. No capítulo 4 é analisada a técnica de HT para inspeção visual automática de placas retangulares de madeira, juntamente com o algoritmo utilizado. No capítulo 5 é feita uma comparação com resultados obtidos com a HT e utilizando Momentos, e também são apresentadas as conclusões finais.

CAPÍTULO 2 - INSPEÇÃO VISUAL AUTOMÁTICA

2.1 - Introdução

Em sistemas de produção, uma tentativa é geralmente feita para se conseguir segurança de 100% de qualidade em todos os setores. Uma das tarefas mais difíceis neste processo é a inspeção visual dos produtos. Avanços na tecnologia do computador, processamento de imagem, reconhecimento de padrões e inteligência artificial, tem resultado em equipamentos de análise de imagem industrial melhores e mais baratos.

A visão de máquina é de vital importância em muitas áreas da fabricação automatizada. Não apenas é necessária para uma grande quantidade de tarefas de montagem de robô, mas também está crescendo de importância no controle de qualidade industrial. Estima-se que 75% das aplicações potenciais da visão na indústria são na área de inspeção visual automática (IVA) (PUTTRÉ,1992). Embora muitas tarefas de inspeção pareçam simples ao olho humano, uma grande proporção do problema surge porque os produtos ou partes estão se movendo ao longo de uma esteira rolante, em taxas excedendo a 10 - 20 partes por segundo. Isso provoca um aumento de carga computacional no hardware, durante o processamento dos dados. Como exemplo, numa fábrica de biscoitos, mais de 1 milhão de biscoitos podem ser fabricados por dia e em torno de 1 milhão de "pixels" terão de ser processados a cada segundo. Assim, há uma tendência na construção de algoritmos compactos garantindo, principalmente, robustez e precisão.

A IVA tem sido largamente usada na Indústria Eletrônica. Aplicações incluem a inspeção de trilhas de placas de circuito impresso (HOU et al., 1992), juntas de solda (MAEDA et al., 1990), fabricação de cápsulas de diodo (PIRONEN et al., 1990), peças manufaturadas (WEN & LOZZI, 1993), ligações em CI's (TSUKAHARA et al., 1991), visão de robôs móveis (LI,

1994), e também em fábricas alimentícias como na classificação de tomates (SARFAR & WOLFE, 1985), inspeção de falhas na superfície de frutas (GRAF et al. 1981), na avaliação da qualidade das sementes (CHURCHILL et al. 1992) e diversos setores onde o processamento e inspeção de imagens se fazem necessários.

2.2 - Modelo Básico de IVA

2.2.1 - Aquisição de Imagens

O funcionamento de um sistema de IVA envolve examinar o mesmo tipo de produto repetidamente para se detectar falhas. A Fig.2.1 mostra os componentes básicos em um sistema de IVA.

FIGURA 2. 1 - Componentes de um sistema de IVA

Um sensor colocado sobre uma esteira adquire imagens dos objetos a serem reconhecidos ou inspecionados. Um tipo de sensor de imagem muito utilizado é uma câmera digital que recebe uma imagem via elementos sensores individuais. Cada elemento sensor registra o nível geométrico pelo montante de carga gerada. O tipo mais comum de dispositivo digital é o

Dispositivo por Carga Acoplada (CCD). A câmera digital tem a vantagem de não ficar sujeita à distorção geométrica, pois não há exploração do feixe eletrônico. A câmera digital é também menos susceptível ao ruído e à superexposição. As imagens adquiridas são enviadas a um digitalizador, o qual converte a imagem de níveis de cinza em uma matriz. Cada elemento dessa matriz corresponde à uma intensidade de luz num ponto particular da imagem. Esses elementos são chamados de “pixels”. Os “pixels” são armazenados na memória do processador, onde podem ser acessados pelos vários componentes do software. Um sistema típico é capaz de armazenar uma imagem de 512×512 “pixels”, onde cada “pixel” pode assumir um certo valor na escala de cinza.

O processamento da imagem está em grande parte condicionado ao hardware do processador, que pode ser altamente complexo e caro, ou simples e barato. O custo está baseado na necessidade se de obter imagens de qualidade e no tempo de resposta do sistema.

No sistema de IVA, uma das mais críticas considerações do hardware é a iluminação. A iluminação está diretamente relacionada ao ambiente e à aplicação. Diferentes modos de iluminação podem ser usados. Exemplos típicos incluem a iluminação difusa, de fundo, direcional e de faixa, mostrados na Fig.2.2.

FIGURA 2. 2 (a) Iluminação de fundo; (b) Iluminação direcional;(c) Iluminação difusa; (d) Iluminação de faixa

Como cada caso é um caso dedicado para aquisição da imagem, BATCHELOR (1985) ilustra 63 métodos de iluminação e técnicas de visualização para posterior aquisição da imagem. Cada método é apresentado em termos do objetivo, do equipamento necessário e das aplicações típicas com comentários específicos em cada um. Isso permite uma idéia global dos problemas e das possíveis soluções existentes. BATCHELOR (1985) discute com grande clareza as considerações econômicas, técnicas e industriais sobre a IVA. Dependendo da aplicação, em certos casos ocorre a necessidade de mudar a iluminação do ambiente durante a inspeção.

O processador realiza o processamento de imagens para acentuar características relevantes e a detecção de defeitos. Após a análise, o processador separa as peças defeituosas das boas. Em determinados sistemas o processador também controla o sensor, o objeto, a esteira e a iluminação. Pode também ser capaz de focalizar ou ajustar o ganho do sensor, retardar o movimento do objeto variando a velocidade da esteira, ou variar a iluminação.

O sistema de IVA deve possuir funções não apenas para o controle de qualidade de objetos que estejam defeituosos, mas também para a medição de vários parâmetros - tamanho, área, perímetro, cor, textura de superfície e assim por diante. Através dessas informações, pode-se alimentar estágios anteriores ao processo a fim de corrigir perdas na qualidade. A estatística é também um dado importante que pode ser coletado diretamente dos computadores. Mesmo em setores onde robôs montam produtos complexos, o uso de IVA é útil não apenas para informar ao controlador onde componentes individuais estão mas, também, para detectar se existem componentes individuais que podem impedir a montagem.

2.2.2 - Técnicas para Análise de Imagens

Os algoritmos de IVA, embora diversos, utilizam um conhecimento à priori. Esse conhecimento é organizado em modelos que fornecem estratégias e padrões para o processo de inspeção. Esse processo é realizado

combinando a parte sob inspeção com um conjunto de modelos pré definidos. Um procedimento típico de tais sistemas é primeiro pré-compilar uma descrição de cada um dos defeitos conhecidos e, então, usar esses modelos de defeitos para detectar, numa imagem, cada estágio de um defeito e especificar sua localização.

Um algoritmo de inspeção deve apresentar certos requisitos para que possa superar os seguintes problemas:

- (1) A presença de ruído e interferência de objetos no fundo da imagem principal (Ex.: chocolate grudado na esteira numa linha de biscoitos);
- (2) Oclusão particular de um objeto por outro;
- (3) As consequências das falhas no objeto a ser procurado;
- (4) Os efeitos da iluminação não ideal (Ex.: não uniforme).

A construção de algoritmos eficientes, significa a escolha certa de características que levam ao reconhecimento.

Em geral, um sistema de IVA é composto de uma fase de treinamento e uma fase de detecção. Os três maiores componentes do sistema são extração de características, modelagem e detecção.

2.2.2.1 - Extração de Características

Características como linhas, curvas, buracos e cantos definem formas geométricas de uma parte sob inspeção. Outras características comuns ao problema de inspeção são dimensões, localizações e orientações. Esses dados são usados para gerar um modelo na fase de modelagem e são de novo extraídos das partes sob inspeção, na fase de detecção, para a localização das anomalias.

Dependendo da aplicação, o processo de extração de características transforma uma imagem em níveis de cinza em binária, usando um "thresholding" simples, ou simplesmente um sensor que produza imagens binárias. O uso de representação binária reduz o montante de dados que devem ser manuseados. Essa simplificação reduz a complexidade computacional em muitas aplicações de inspeção industrial, mas coloca

certas limitações na capacidade do sistema. Por exemplo, propriedades da superfície física tais como reflexão, orientação e texturas são perdidas na conversão para uma imagem binária. A desvantagem no processamento de imagens em níveis de cinza é a taxa lenta de extração.

2.2.2.2 - Modelagem

A modelagem é formada pelas características extraídas da imagem. Muitos modelos para inspeção são baseados em propriedades geométricas da silhueta do objeto a ser inspecionado.

Várias aplicações de inspeção industrial envolvem a inspeção de padrões planares (ex.: placas de circuitos, rótulos ou pacotes). Esses padrões podem ser representados por modelos contendo características bidimensionais. Em LIBERMAN¹ apud CHIN (1986), mostra-se um enfoque da modelagem de peças tridimensionais da indústria desenvolvendo computando-se as projeções bidimensionais de todas as possíveis posições estáveis da peça, em relação ao sensor.

2.2.2.3 - Detecção

O problema geral de detecção pode ser formulado por dois enfoques:

- (a) Detecção dos defeitos
- (b) Processo de verificação de todas as características

A escolha do processo de detecção é altamente dependente do tipo de modelo usado e da natureza do objeto a ser inspecionado. O enfoque de verificação é mais usado na inspeção industrial onde o treinamento do modelo pode ser facilmente feito "mostrando" ao sistema a parte do protótipo a ser inspecionado, enquanto que o enfoque de detecção requer a geração de um conjunto de padrões defeituosos.

¹ LIBERMAN, L. Model-driven vision for industrial automation, in "Advances in digital image processing-theory, applications, implementation" New York, P.Stucki,ed., 1979 apud CHIN, R.T. Algorithms and techniques for automated visual inspection, Madison,Wisconsin, Academic Press, 1986. p.594

Quando se projeta um modelo de uma peça, usando técnicas de detecção, levando em conta suas características e rotação, obtém-se uma diminuição do ruído e da distorção da imagem.

2.3 - Processamento de Imagens

Em muitas indústrias eletrônicas, especialmente na produção de placas de circuito impresso (PCB's), busca-se reduzir os defeitos de fabricação. Várias pesquisas em IVA foram mostradas por CHIN (1986). As técnicas e métodos aplicados a IVA têm mostrado que é possível alcançar a segurança desejada. Algumas delas serão discutidas a seguir.

2.3.1 - Método de Segmentação

As imagens segmentadas são formadas através das imagens generalizadas por associação de seus elementos em subconjuntos de objetos significantes na cena. Por exemplo, segmentando-se uma cena composta por poliedros planares, tem-se um subconjunto de "bordas" correspondentes às arestas dos poliedros, ou um subconjunto de regiões bidimensionais correspondentes às faces dos mesmos. O conhecimento sobre o domínio é importante para reduzir o montante de computação e para superar problemas de ruído ou dados inadequados.

Os métodos para segmentação podem ser divididos em algumas grandes áreas:

- a) métodos globais que atuam ao mesmo tempo em toda a imagem, geralmente por alteração do histograma;
- b) métodos que detectam regiões através da determinação da fronteira (borda) dessas regiões (RIGNEY et al.1992);
- c) métodos que detectam regiões através de pontos vizinhos com características semelhantes.

2.3.2 - Método de “Thresholding”

O “thresholding” é uma forma de binarização da imagem. Para aplicar o “thresholding” em uma imagem em níveis de cinza é preciso determinar o nível de limiar desejado. Abaixo desse valor todos os “pixels” da imagem serão transformados em nível “zero” e, acima, nível “um”, correspondendo a branco e preto, respectivamente.

A iluminação é um fator importante na determinação do valor de “thresholding”. Mesmo para a mesma imagem, variando a iluminação, deve-se variar o “thresholding”.

A vantagem do método é a diminuição de dados numa imagem e conseqüentemente, o montante de processamento realçando, também, a imagem do objeto com relação ao fundo. (WASZKA, 1978).

Em várias aplicações de IVA, utiliza-se o “thresholding”. NORTON-WAYNE (1991) e PIRONEN et al (1990) usam um “thresholding” adaptativo, ou seja, para cada parte da imagem é estabelecido um limiar podendo-se, assim, extrair de forma mais precisa todas as características da imagem.

2.3.3 - Subtração de Imagem

A subtração de imagem é o processo mais simples, em inspeção visual. Neste enfoque, a parte a ser inspecionada é explorada e sua imagem é comparada com a imagem ideal.

A Fig.2.3 mostra o processo de subtração como uma operação lógica, quando a imagem de teste e o modelo são ambas funções binárias. O resultado obtido é a diferença entre as duas imagens.

O problema de sensibilidade é a maior desvantagem da subtração de imagem, considerando o fato de que muitas, e talvez a maioria das imagens, não se correspondem ponto-a-ponto, devido à variação na iluminação, mesmo que não haja variação de posição.

FIGURA 2. 3 Diagrama esquemático de subtração de imagem

2.3.4 - Combinação de Características

É uma forma melhorada do método de subtração de imagem. Ao invés de se comparar parte da imagem sob inspeção com um modelo, as características extraídas da imagem e as definidas pelo modelo são comparadas. As características extraídas são em muitos casos padrões geométricos tais como cantos, linhas, curvas, etc. Esse processo de combinação em processamento de imagem é chamado de combinação de “template”.

O método tem melhor desempenho quando na sua aplicação se utiliza “templates” pequenos, os quais são invariantes às variações de formato e tamanho, reduzindo assim o custo computacional.

LECORDIER & MARTÍNI (1992) aplicaram esse método na inspeção de frascos em uma linha de produção, usando uma arquitetura Data Flow. Na General Motors, um sistema foi desenvolvido para inspecionar e alinhar CI's usando um método de combinação de “templates” múltiplos.

2.3.5 - Técnica de Identificação por Cores

Muitos sistemas de IVA usam valores de níveis de cinza para a extração de características. Baseado no formato e no tamanho do objeto,

esses sistemas podem classificar, separando objetos com ou sem defeitos. Contudo, em algumas aplicações podem ocorrer determinadas falhas, por exemplo, na identificação de resistores. A introdução da cor possibilita um melhor ajuste na identificação. Entretanto, comparado com o sistema de níveis de cinza, o sistema de visão colorida requer maior memória para o armazenamento das imagens e três vezes mais processamento, por utilizar as três cores primárias; consequentemente, é um sistema mais complexo de aquisição de imagem.

Estudos realizados por BUMBACA & SMITH (1987) mostraram a possibilidade de se criar sistemas que possam identificar um ou vários objetos e também sua cor, entre vários objetos desconhecidos. As cores dos objetos a serem identificados são restritas a uma das três cores básicas.

Em alguns exemplos, o uso da cor é necessária. Wayne Daley, pesquisador no Instituto de Tecnologia da Geórgia, usa um sistema de inspeção de cor para análise de alimentos. Outro trabalho também usando a cor é a inspeção da qualidade de aves, através de sinais de doença e outros problemas (PUTTRÉ, 1992).

2.3.6 - Técnica da Transformada de Hough

A HT é uma técnica muito utilizada na detecção de bordas retas, obtendo eficientes resultados (HUNT, 1990). Também tem demonstrado ser uma técnica robusta e resistente a ruídos. O seu método consiste na parametrização dos dados de imagem empregando coordenadas normais (ρ - θ) da linha.

Outra aplicação da HT é na detecção de objetos circulares. Uma vantagem particular dessa técnica é que, mesmo quando o limite de um objeto circular é parcialmente rompido por uma quebra, ou oclusão de uma parte, continua sendo possível estimar a localização do centro desse objeto, conforme Fig.2.4 (KIMME, 1975).

Existe um crescente número de aplicações em reconhecimento e inspeção de produtos manufaturados utilizando a HT (WEN & LOZZI, 1993), (KASTURI, 1990), (NAGATA & ZHA, 1991).

Estudos da HT serão mais detalhados no Capítulo 3.

FIGURA 2. 4 - Localização do centro do círculo através da HT.

2.4 - Aplicações de Inspeção Visual Automática

Os sistemas de IVA são úteis em aplicações onde a complexidade ou o volume de trabalho necessitam de uma supervisão contínua do homem. Sistemas de IVA num ambiente controlado tomam decisões mais precisas e são mais rápida do que a inspeção manual. Contudo, o homem tem a habilidade para determinar anormalidades quando os sistemas não são programados para verificar.

2.4.1 - Visão de Máquina na Indústria Eletrônica

A indústria eletrônica é uma das maiores forças diretoras na investigação da inspeção visual, devido a inúmeras tarefas e à crescente complexidade de seus produtos.

MAEDA et al. (1990) propuseram uma IVA para padrões de multicamada de pastilhas de LSI, usando um algoritmo de combinação de padrões em cascata. Os padrões de multicamada são construídos pela exposição após alinhamento do padrão da imagem detectada com o da imagem carregada. O alinhamento é executado, usando-se as bordas da imagem. Assim elimina-se a região onde os níveis de cinza são iguais nas duas imagens. Logo as partes que não são combinadas na região são consideradas como defeitos. O sistema foi construído e avaliado experimentalmente. Como resultado, foi verificado que a área toda do chip pode ser inspecionada e um defeito de 0,5 mm ou mais pode ser detectado num modo estável, conforme a Fig.2.5 .

FIGURA 2. 5 - Tipos de defeitos

A fim de automatizar o processo de inspeção visual nas fábricas de produção de diodos, OKAWA & MIZUNO (1991) sugeriram um sistema composto por um computador e uma câmera de TV industrial conectada a um microscópio. Baseando-se na teoria de decisão estatística e método de "thresholding", observou-se que os resultados das análises chegaram a 96,7%.

FIGURA 2.6 - Sistema de aquisição de imagem

A qualidade nas ligações em circuito integrado (CI) é de grande importância para a indústria eletrônica. Baseado nesse aspecto TSUKAHARA et al. (1991) desenvolveram um sistema de inspeção automática que captura uma imagem de CI com alto contraste, conforme pode ser visto na Figura 2.6.

FIGURA 2.7 - Típicos defeitos de fiação e conexão

Essa inspeção visual pode detectar defeitos nos fios e junções e pode avaliar a qualidade das ligações. Os defeitos detectados nos fios incluem os rompimentos, a proximidade e fiações incorreta em trilhas. Os defeitos nas junções, que têm o formato de uma bola, incluem deformação e colocação incorreta, como mostrado na Fig.2.7.

FIGURA 2. 8 - Algoritmo do detector de borda

A aquisição da imagem é através de uma câmera de área ampla, chamada de "Mega-câmera", que pode capturar uma imagem de 2048 x 2048 "pixels" usando um sensor de imagem linear CCD. Essa câmera pode capturar uma imagem em 2 s. A imagem com alto contraste é essencial para uma correta inspeção. O sistema usa 2 tipos de óptica para iluminação: uma vertical e outra lateral. O sistema de IVA utiliza a combinação de dois algoritmos. O primeiro, que mede as junções no formato de bola, é baseado na interpolação da imagem de resolução de 10 mm para 2,5 mm. Com isso, é possível medir o diâmetro da bola numa precisão de ± 5 mm. Aplicando um operador Laplaciano à imagem interpolada, é possível produzir uma imagem de segunda derivada. Pontos de cruzamento zero na imagem de segunda derivada indicam localizações de borda, conforme Fig.2.8. O segundo algoritmo para inspeção do fio é baseado num seguidor de borda, o qual

detecta a ausência de fios, fios quebrados e a distância entre eles, de acordo com a Fig.2.9.

FIGURA 2. 9 - Detecção dos defeitos nas fios

WEN & LOZZI (1993) utilizaram o método da HT em reconhecimento e inspeção de partes manufaturadas. O processo de reconhecimento empregou apenas os elementos de contorno dos objetos. A função principal da HT é determinar os mais prováveis parâmetros de identificação do objeto desconhecido, dentro de uma base de dados de modelos. Para isso, foram criados dois arranjos acumuladores, onde o primeiro acumulava a distribuição dos possíveis fatores de escalonamento, mostrando corretamente o valor e o segundo, mostrando possíveis ângulos de rotação.

LI (1994) propôs um sistema de visão para robôs móveis serem utilizados em escritórios. O robô é projetado para caminhar em corredores e salas, buscando e entregando objetos simples. O deslocamento é feito em faixas usando laser e duas câmeras. Para o processamento da imagem, OHTA et al.². apud LI (1994) usou programação dinâmica no espaço de Hough para solucionar o problema de correspondência na visão estéreo.

O sistema obteve bons resultados utilizando imagens reais. O método trinocular oferece menores taxas e melhores combinações para linha em

² OHTA et al. **Stereo by intra and intra-scanline search using dynamic programming**. 1985 apud LI, Z., Stereo correspondence based on line matching in Hough space using dynamic programming, IEEE Transactions on Systems, Man, and Cybernetics, 1994. p.145

várias orientações. Sua velocidade de execução numa SPARC - 1SUN é satisfatória.

A implementação da detecção de linha de Hough e programação dinâmica tem potencial para ser paralelizado e assim satisfazer o requerimento de tempo real das aplicações robóticas.

2.4.2 - Avaliação da Qualidade de Superfícies Metálicas

PIRONEN et al. (1990) desenvolveram um protótipo de IVA para superfícies metálicas de rolamento. O sistema é capaz de detectar e classificar defeitos de superfície em faixas de liga de cobre. A aquisição da imagem é baseada numa câmera "linescan" CCD e focos de luzes, com o uso de lentes Fresnel.

O reconhecimento de defeitos é baseado na análise das classes de bolhas defeituosas, conforme Fig.2.10. O Método de "thresholding" dinâmico permite o controle local do contraste, formato e tamanho de bolhas a serem detectados, resultando numa baixa taxa de alarme falso.

A identificação de defeito estatístico baseado nas características calculadas de cada bolha, na imagem segmentada, é segura apenas para grandes defeitos. O uso de técnicas de reconhecimento de padrão estrutural para defeitos constituídos de vários componentes tem dado promissoras melhorias na precisão da classificação.

Classificação das classes			
linha		risco claro horizontal
bolha	lasca s/ bolha próxima		risco preto vertical
lasca c/ bolha	erosão

FIGURA 2.10 - Exemplo típico de classes de defeitos

Uma avaliação sobre qualidade de superfícies laminadas e torneadas, utilizando métodos de inspeção visual, foi realizada por HOY & YU (1991). Dois métodos foram examinados e comparados. O primeiro foi utilizando as intensidades do histograma para caracterizar a aspereza e qualidade da imagem da superfície. As mudanças no histograma servem para quantificar o acabamento. O segundo utiliza a Transformada de Fourier Rápida 2D da imagem. A informação da magnitude e frequências obtidas da FFT são usadas como parâmetros de medida do acabamento da superfície.

Quanto ao desempenho dos métodos, o método do histograma requer bem menos tempo computacional do que o método FFT e pode portanto ser mais facilmente desenvolvido num processo de tempo real ou quase real. Em contraste, o método FFT 2D é computacionalmente muito dependente da geração de FFT.

2.4.3 - Aplicação em Manutenção de Linhas de Transmissão

Baseado no perigo da manutenção em linhas de transmissão de alta voltagem, GUNDIMEDA & VELUVALI (1991) propuseram um sistema de IVA utilizando um robô numa plataforma isolada móvel, de onde ele possa executar testes nos isoladores.

FIGURA 2.11 - Plataforma de inspeção

As imagens do ambiente são adquiridas por uma câmera CCD, onde os seus movimentos de rotação e translação são calculados usando a técnica de mínimos quadrados. A qualidade das imagens digitais depende grandemente da iluminação utilizada. A Fig.2.11 mostra o modelo de inspeção proposto.

No processamento da imagem as técnicas usadas são simples e eficientes. Inicialmente a câmera adquire uma imagem quadrada de 256 x 256 "pixels" de 256 níveis de cinza. São usadas 2 técnicas de uniformização para reduzir os ruídos e os efeitos espúrios: (a) ponderação de vizinhança e (b) filtragem mediana. Caso a qualidade da imagem não esteja boa, pode-se usar a equalização do histograma. Assim, após o detector de bordas e o "thresholding" a imagem resultante é comparada com uma base de dados disponíveis no sistema. Quando da publicação do trabalho a etapa final ainda não tinha sido implementada completamente. Entretanto, o sistema, nessa primeira fase, obteve bons resultados.

2.4.4 - Controle de Qualidade em Tecelagem

A inspeção visual torna-se muito desgastante quando a peça possui muitos detalhes a serem reconhecidos. NORTON-WAYNE (1991) estudou três formas de inspeção em cordões de renda como o da Fig.2.12

FIGURA 2. 12 - Modelo de Renda

O estudo foi realizado em laboratório, com amostras isoladas de rendas. As amostras eram colocadas sobre uma esteira escura e as imagens coletadas por uma câmera CCD "linescan". Os três métodos testados foram: (a) Aplicação de "thresholding" nas imagens de níveis de cinza, juntamente com uma forma de filtro binário não-linear. Esse método mostrou-se atraente, devido à sua simplicidade e economia computacional. (b) Comparação de padrões de níveis de cinza da imagem com um modelo. Isso é feito analisando os espectros de Fourier 2D. Entretanto, os resultados experimentais não obtiveram o sucesso esperado, pois os defeitos são essencialmente fenômenos locais, enquanto que as componentes espectrais de Fourier são funções globais. Assim, a transformação degrada o contraste do defeito. (c) Comparação da imagem binarizada da renda com um modelo padrão. Como os bordados de renda se repetem várias vezes, NORTON-WAYNE (1991) utiliza o modelo padrão deslocando-se sobre a imagem binarizada, realizando a subtração. Pelo resultado da diferença obtém-se os defeitos os quais podem ser detectados por um "thresholding". Esse método, entretanto, possui um elevado tempo computacional para uso "on-line".

2.4.5 - Controle de Qualidade na Indústria Alimentícia e na Agricultura

Segundo DAVIES & JOHNSTONE³ apud DAVIES; JOHNSTONE (1989) na inspeção de produtos alimentícios, tais como, biscoitos redondos movendo-se sobre uma esteira, uma técnica que se mostrou rápida na localização foi a HT. Nota-se que o enfoque baseado na HT é particularmente robusto, tendo mostrado ser excepcionalmente tolerante para produtos quebrados, sobrepostos ou deformados.

Uma análise estatística feita por CHURCHILL et al. (1992) mostrou que a consistência das medidas de visão de máquina foram maiores que as

³ DAVIES, E.R.; JOHNSTONE, A.I.C. Engineering tradeoffs in the design of a real-time system for the visual inspection of small products. Proc. 4th Conf. on VK Research in advanced manufacture, 1986 apud DAVIES, E.R.; JOHNSTONE, A.I.C. Methodology for optimising cost/speed tradeoffs in real-time inspection hardware, IEE Proceedings, v.136, Pt.E, n.1, 1989. p.63

medidas humanas e requer cerca de 1/3 do tempo. Esses dados foram retirados da inspeção de três espécies diferentes de sementes de grama. O sistema tem a finalidade de medir as propriedades físicas da semente da colheita (extensão, largura e grossura) e partículas contaminadas de um lote de sementes antes da sua separação, para determinar o tamanho da tela e bolsas de derivação. As imagens foram obtidas por um estéreo-microscópio ZIESS STEMI equipado com um retículo e enviadas para um microcomputador.

A detecção dos defeitos de superfície e textura é de grande importância para muitas aplicações de visão de máquina. REHKUGLER & THROOP (1987) investigaram o processamento de imagens em níveis de cinza para detecção de danos ocasionadas por pássaros, insetos e granizo em maçãs. Pesquisas de classificação de tomates de SARFAR & WOLFE (1985) usaram visão de máquina em nível de cinza para detectar cicatrizes, fissuras, buracos e partes deterioradas. Imagens coloridas (RGB) e infravermelhas foram processadas para detectar cicatrizes, cortes e deformações em pêssegos (MILLER & DELWICHE, 1991). RIGNEY et al. (1992) desenvolveram um sistema de controle de qualidade das hastes de aspargos. Medidas de diâmetro e extensão das hastes isoladas, embora requeressem classificação, são consideradas triviais para um sistema de IVA. Os algoritmos foram testados usando imagens em níveis de cinza para detecção de pontas difusas, pontas quebradas, cicatrizes e fissuras na haste. No tratamento dos algoritmos utilizou-se basicamente segmentação da imagem juntamente com um “thresholding” adaptativo e filtros Laplacianos.

2.4.6 - Inspeção de cápsulas com uma Arquitetura Data-Flow para IVA

Uma arquitetura Data Flow foi utilizada por LECORDIER & MARTÍNI (1992) para um sistema de IVA, cuja função era inspecionar 80.000 cápsulas farmacêuticas e a impressão nos frascos numa taxa de 5.000 p/h. A arquitetura é altamente paralela formada por 8 microprocessadores uPD7281,

os quais foram divididos com tarefas específicas no processamento da imagem. A imagem após sofrer uma redução de tamanho e aplicado um “thresholding”, que consiste em eliminar as possíveis diferenças na iluminação, é segmentada. Depois é feito uma comparação entre os dados da imagem com um modelo armazenado. As vantagens que o sistema oferece são a flexibilidade e a capacidade de executar tarefas complexas a custo menor e com tempos computacionais que, em muitos casos, são suficientes.

2.4.7 -Aplicação da HT na Interpretação de Linhas e Segmentos

Um sistema para a interpretação de desenhos de linhas foi desenvolvido por KASTURI et al. (1990), com a finalidade de localizar e separar segmentos de texto de vários tamanhos, estilos e orientação de fonte. O método utilizado foi a aplicação da HT nos centróides dos componentes conectados da imagem.

LI et al. (1989) propusera uma HT modificada que verifica a da continuidade do segmento. Para isso usou uma arquitetura sistólica. Várias simulações foram feitas no sentido de se obter um melhor desempenho quanto à velocidade de resposta.

CAPÍTULO 3 - A TRANSFORMADA DE HOUGH

3.1 - Introdução

A Transformada de Hough (HT), é hoje, uma técnica de grande utilidade em visão computacional. Sua aplicação até meados da década de 80, estava sendo muito lenta, devido à sua complexidade computacional de armazenamento e a dificuldade de um entendimento detalhado de suas propriedades. Contudo, nos últimos anos, muitas pesquisas têm sido realizadas, contribuindo, dessa forma para um melhor aproveitamento de seus recursos.

A utilização da HT requer que se conheça um pouco sobre a localização das bordas da imagem, a fim de que sejam considerados apenas os “pixels” de fronteira, como mostra a Fig.3.1. O método de Hough pode ser aplicado para detecção de curvas desde que as mesmas possam ser descritas de forma paramétrica (linha reta, círculos, elipses, etc).

FIGURA 3.1 - Imagem de bordas

Vários autores têm desenvolvido novos algoritmos, derivados da HT original (DUDA & HART (1972), BALLARD (1981), LI et al. (1985), ILLINGWORTH & KITTLER (1987), BEN-TZVI & SANDLER (1990), HAN et al. (1994). A implementação em arquiteturas paralelas tem surgido na tentativa de se obter maiores velocidades de processamento (BEN-TZVI et al. (1990), ROSENFELD et al. (1988)).

3.2 - Fundamentos da Transformada de Hough

A Transformada de Hough (HT), foi primeiramente proposta como um método para detecção de padrões complexos em imagens binárias, introduzida por Hough (1962) numa patente. Pela proposta de Hough a linha é mapeada usando os parâmetros Inclinação-Intersecção. Na Fig. 3.2a temos uma linha reta que conecta uma sequência de pontos expressos pela equação

$$y = m x + c \quad (1)$$

onde "m" representa a inclinação e "c" a intersecção. Considerando-se 3 pontos pertencentes à mesma reta, onde cada ponto do espaço imagem é transformado em uma equação de reta. Conforme mostra a Tabela 1.

TABELA 1 - Efeito da HT sobre uma linha reta no espaço m-c

	X	Y	$Y = m * X + C$	$C = -X m + Y$
A	3	4	$4 = m * 3 + C$	$C = -3 m + 4$
B	1,5	1	$1 = m * 1,5 + C$	$C = -1,5 m + 1$
C	0,5	-1	$-1 = m * 0,5 + C$	$C = -0,5 m + -1$

Essas retas então, são traçadas no espaço de parâmetros, definindo, assim, o ponto de intersecção, conforme Fig. 3.2b. Todavia, tanto a inclinação como a intersecção são ilimitadas e essa parametrização tem a desvantagem de ser sensível à escolha do eixo de coordenadas no plano da figura. O que complica a aplicação da técnica.

FIGURA 3.2 - A Transformada de Hough
a) Espaço da Imagem b) Espaço de Parâmetros

Uma solução para esse problema foi proposta por DUDA & HART (1972), onde utilizaram coordenadas polares para definir a reta, trabalhando com os parâmetros Ângulo-Raio ao invés de Inclinação-Intersecção. Com isso, os estudos tomaram outro ritmo, um ritmo acelerado, proporcionando novos melhoramentos e perspectivas de grandes avanços.

O método de DUDA & HART (1972), envolve um mapeamento de retas do "espaço de imagem", para conjuntos de pontos num "espaço de parâmetros Ângulo-Raio". No plano da imagem, uma reta parametrizada pode ser representada por dois parâmetros: ρ sendo a distância da normal até a linha da origem da imagem, e θ sendo o ângulo dessa normal como eixo horizontal conforme pode ser visto na Fig. 3.3.

A equação da reta correspondente a essa geometria é:

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

FIGURA 3.3 - (a) Representação da equação da reta em função de suas coordenadas polares. (b) Relação entre definição de reta em coordenadas polares.

Os novos parâmetros utilizados para representar o espaço são definidos agora por ρ e θ . Logo, o problema de detectar pontos colineares pode ser convertido no problema de se encontrar curvas concorrentes.

Segundo DUDA & HART,(1972) as propriedades para se transformar ponto em curva são:

Propriedade 1 - Um ponto no plano da figura corresponde a uma curva senoidal no plano de parâmetros.

Propriedade 2 - Um ponto no plano de parâmetros corresponde a uma reta no plano da figura.

Propriedade 3 - Pontos pertencentes a mesma linha reta no plano da figura correspondem a curvas através de um ponto no plano de parâmetros.

Propriedade 4 - Pontos pertencentes a mesma curva no plano de parâmetros correspondem a linhas através do mesmo ponto no plano da figura.

Na Fig.3.4(d) tem-se a propriedade de reflexão, onde os pontos “A”, “B” e “C” se repetem quando a senóide inicia um novo ciclo, entretanto com sinal negativo.

FIGURA 3.4 - Ilustração da HT. (a) Pontos no espaço da imagem. (b) Mapeamento dos pontos no espaço $(\rho-\theta)$. (c) Detecção das propriedades de colinearidade. Ponto A denota intersecção de curvas correspondentes aos pontos 1, 3 e 5 no plano $x-y$ e similarmente ponto B corresponde aos pontos colineares 2, 3 e 4. (d) Propriedade reflexão da HT.

De posse das propriedades, é feito o mapeamento das curvas no arranjo acumulador. (A construção do arranjo acumulador bidimensional é definida dentro de um erro aceitável em ρ e θ formando uma grade). Esta grade pode ser limitada no intervalo $0 \leq \theta < \pi$ e $-R \leq \rho < R$, onde R é o tamanho da célula, desde que pontos fora desse retângulo correspondam a linhas no plano da imagem que não cruzem a célula. Com a limitação de θ , toda linha no plano $x-y$ corresponde a um único ponto no plano $(\rho-\theta)$. Com a equação 2, a cada ponto (x,y) do espaço imagem, a curva correspondente é

colocada no arranjo incrementando-se de uma unidade o contador em cada célula ao longo da curva.

A análise do arranjo, após os cálculos da equação 2, revelam células que possuem números elevados de contagens. Esses valores de contagem representam, para cada célula, dentro de um erro de quantização, a quantidade de pontos coincidentes de uma reta. Portanto, o método da HT requer a identificação desses máximos locais. Muitas vezes ocorrem vários picos altos próximos uns dos outros no arranjo. Com isso, existe a necessidade de utilizar um "threshold" para filtrar a quantidade de erros e aproximações e assim poder identificar realmente os parâmetros da imagem.

Os valores de "threshold" estão diretamente relacionados às características de cada imagem.

3.3 - Principais modelos de Transformada de Hough

3.3.1 - Transformada de Hough Generalizada

A Transformada de Hough Generalizada (GHT) é uma variação da HT original. BALLARD (1981) utilizou um mapeamento entre o espaço da imagem e o espaço da HT. A imagem inicialmente é tratada por operadores de borda. Os operadores mais comuns são de dois tipos: o operador de Gradiente (exemplo: PREWITT⁴ apud BALLARD (1981)) e o operador de HUECKEL (1973), os quais transformam os níveis de cinza da imagem em uma rampa e um degrau, respectivamente. Com isso, cada ponto na imagem não mais contém informações do nível de cinza mas, ao contrário, cada ponto contém uma magnitude e direção representando a orientação da mudança do nível de cinza local.

No algoritmo, o mapeamento é definido a partir da orientação das bordas em relação a um ponto de referência. O ponto de referência é um

⁴PREWITT, J.M.S. *Object enhancement and extraction*. New York, Academic Press, 1970 apud BALLARD, D.H. *Generalizing the Hough transform to detect arbitrary shapes*, Pattern Recognition, 1981. p.114

ponto arbitrário, sendo considerado como a origem do sistema de coordenadas locais para o formato. O grande segredo para a identificação de formatos arbitrários da GHT é o uso da informação direcional. Tal informação, além de permitir uma identificação mais rápida do formato, também melhora sua precisão. Por exemplo, na detecção de círculos, sem a informação direcional, qualquer grupo significativo de pontos de borda teria grande probabilidade de ser detectado.

No Arranjo Acumulador, um problema que surge é na detecção dos pontos máximos. Muitas vezes, pontos na vizinhança do ponto ideal são incrementados, ao invés do próprio ponto. Um meio de diminuir esse problema é utilizar um modelo de erro formal na etapa de incrementação como pode ser visto na Fig.3.5. Esse modelo pode especificar um conjunto de pontos próximos ao invés de um único ponto. SHAPIRO (1978a)(1978b)(1975) fez extensivo trabalho neste assunto.

Trabalhos realizados anteriormente utilizaram imagens de bordas binárias para detectar curvas analíticas em DUDA & HART (1972) e não analíticas em MERLIN & FARBER (1975). Uma outra proposta feita por BALLARD (1981) foi a detecção de curvas analíticas em imagens em níveis de cinza; já O'GORMAN & CLOWES (1976) estudaram especialmente linhas retas; SHAPIRO (1978a) (1978b) elaborou várias propriedades para detecção de círculos e parábolas em imagens ruidosas.

FIGURA 3.5 - Usando convolução para compensação de erros na GHT.

Em imagens arbitrárias, para cada formato de imagem a GHT cria uma tabela, a tabela "R" onde serão colocadas as distâncias de pontos da superfície ao ponto de referência, conforme mostra a Fig. 3.6. Dado o limite do formato, a tabela "R" pode ser facilmente construída examinando os

pontos limites do formato e requer um número de operações proporcional ao número de pontos do limite. Segue sua forma:

- escolhe-se um ponto de referência y para o formato;
- para cada ponto limite x_n da borda computar a direção do gradiente ($\phi(x)$) e o valor de $r = y - x_n$ (distância da borda da imagem ao ponto de referência);
- com os valores de r indexados aos valores de ϕ constrói-se a tabela "R".

FIGURA 3.6 - Geometria da Transformada de Hough Generalizada

Uma das propriedades importantes na GHT é o fato de se poder fazer mudanças de escala, rotação, reversões da base da figura, e translação do ponto de referência de r . Todas essas propriedades podem ser computados pelas modificações diretas para a tabela "R".

3.3.2 - Transformada de Hough Rápida

Na transformada convencional HT vista em DUDA & HART (1972), cada ponto do traço no espaço da imagem gera "votos" para um conjunto de pontos no espaço de parâmetro. Já LI et al.(1986) propuseram um novo enfoque à HT, usando os princípios de formulação do hiperplano para o conjunto de pontos do espaço de parâmetros. Esses hiperplanos são divididos gradativamente por um algoritmo em hipercubos de baixa e alta resolução. Em seguida é realizada uma contagem nos hiperplanos e um valor

de "threshold", determinado inicialmente, limita a criação de subdivisões de hipercubos. A decisão se um hipercubo recebe um voto de um hiperplano dependerá do hiperplano cruzar o hipercubo. Esse algoritmo foi chamado de Transformada de Hough Rápida (FHT). Esse enfoque hierárquico leva a uma redução significativa da complexidade computacional e o requerimento de armazenamento simultâneo.

Devido ao problema da formulação do hiperplano e à representação "K-tree" do espaço de parâmetros, o teste de intersecção pode ser feito incrementalmente, sem multiplicação. Isso é importante para uma futura implementação de hardware VLSI e para extensões do método para manejar superfícies de ordem maior no espaço de parâmetro.

FIGURA 3.7 - Exemplo de um processo de FHT

Através da Fig. 3.7 é possível ter uma visão geral do processo da FHT. Essa figura é um hiperplano bidimensional formado por oito retas.

Inicialmente o hiperplano é dividido em quatro hipercubos com endereços $(0,0)$, $(0,1)$, $(1,0)$, $(1,1)$. Escolhendo um "thresholding" $T=7$ observa-se pelos endereços de "K-Tree", mostrados na Tabela 2, que o hipercubo $(1,1)$ obteve um ponto formado por 8 cruzamentos. Em seguida o hipercubo $(1,1)$ é dividido em 4 hipercubos (de tamanho $1/4 \times 1/4$). Nota-se que o hipercubo $(11,11)$ depois de um "thresholding" encontrou o maior cruzamento.

Uma nova divisão é feita de tamanho $1/8 \times 1/8$ e encontrando o valor 8. Tal processo de divisão pode ser repetido até a desejada resolução ser alcançada.

TABELA 2 - Valores referentes ao exemplo do processo de FHT.

Voto	Endereço K-Tree			
tamanho	(0,0)	(0,1)	(1,0)	(1,1)
$1/2 \times 1/2$	3	3	4	8
tamanho	(10,10)	(10,11)	(11,10)	(11,11)
$1/4 \times 1/4$	3	5	6	8
tamanho	(110,110)	(110,111)	(111,110)	(111,111)
$1/8 \times 1/8$	8	4	4	3

O armazenamento requerido pelo exemplo dado através da Fig.3.7 são 4 registradores de cruzamentos em cada nível de divisão. Uma implementação direta do método de Hough (no qual o espaço do parâmetro é uniformemente dividido e cada hipercubo é testado contra cada hiperplano), poderia ter requerido 64 registradores de "voto" para a escolhida dimensão e quantização. O algoritmo da FHT executa o teste para 13 hipercubos ($1 + 4 + 4 + 4$) de diferentes resoluções, enquanto que um enfoque direto executa o teste para 64 hipercubos.

Alguns problemas surgem em imagens complexas. O valor de "threshold" pode ser difícil para se determinar e, se esse for colocado muito baixo, a eficiência do método pode deteriorar-se, pois o algoritmo irá explorar muitos quadrantes. Outro problema é relacionado ao método de teste de intersecção incremental. Isso requer traço da imagem ao seu centro. Se existem muitos traços na imagem, isso pode representar uma grande sobrecarga.

O algoritmo FHT é altamente paralelo. O processamento para os hipercubos são independentes uns dos outros. Além disso o teste de

intersecção para um hiperplano não depende de outros hiperplanos. Quando a implementação paralela é considerada, surge o problema de sincronização.

A sincronização envolve manter a ordem parcial da escolha política de controle para a FHT. A política de classificação de contagem de "voto" é mais difícil de manter.

3.3.3 - Transformada de Hough Adaptativa

ILLINGWORTH & KITTLER (1987) desenvolveram a Transformada de Hough Adaptativa (AHT). A ideia principal é uma estratégia iterativa de procura "Grosseira para Fina" para detectar linhas e círculos em espaços de parâmetros 2D e 3D.

Na transformada padrão quando o arranjo acumulador torna-se muito grande, começam a surgir aspectos indesejáveis. Com isso requer-se um espaço de armazenamento grande. Se K parâmetros tem de ser determinados e cada faixa de parâmetro é dividida em " L " intervalos, o armazenamento requerido é L^k elementos. A AHT utiliza, inicialmente, um arranjo acumulador pequeno e de tamanho fixo. Na primeira etapa, cada parâmetro é resolvido apenas muito grosseiramente. Tem-se assim uma ideia básica da imagem.

A adequabilidade da AHT para detectar formatos de linha e círculo, depende da identificação correta das regiões de alta densidade que produzem no espaço de parâmetros, mesmo quando o espaço é grosseiramente quantizado.

No algoritmo adaptativo, uma das primeiras decisões que deve ser tomada é definir o tamanho do arranjo acumulador, geralmente pequeno. ILLINGWORTH & KITTLER (1987) escolheram um arranjo 9x9, o qual atendeu às expectativas. Os dados acumulados são inteligentemente usados para redefinir a faixa do parâmetro, de modo que áreas interessadas possam ser investigadas em maiores detalhes.

Um importante aspecto do algoritmo é sua habilidade para redefinir parâmetros, independente de cada iteração. Os critérios usados para identificar o melhor pico é o valor resultante da divisão do número de contagens pelo número de células acumuladoras. Tomando como referência a posição do melhor pico, calcula-se em torno deste quantas iterações forem necessárias para identificar a imagem.

Na Fig.3.8a foram necessárias sete iterações para determinar os parâmetros da linha. A Fig.3.8b mostra as retas que são geradas de parâmetros da AHT. Na Fig.3.8c tem-se arranjo acumulador. Os três pontos hachurados representam o possível ponto de intersecção da reta.

FIGURA 3.8 - Detecção de segmentos lineares usando a AHT. (a) Imagem. (b) Espaço de Parâmetros (limites $m = -1$ a 1 , $c = -256$ a 256). (c) Acumulador grosseiramente quantizado.

Na Fig. 3.9a foram necessárias 4 iterações para determinar o centro do círculo. No arranjo acumulador, na Fig. 3.9c, observa-se apenas um ponto hachurado, isso facilita a interpretação. A célula sombreada é o pico selecionado pela limitação do arranjo.

O enfoque dos métodos de LI et al.(1986) é similar à AHT, à medida em que o espaço de parâmetros é investigado em várias resoluções,mas com um

maior detalhamento nas regiões onde ocorre uma alta densidade de pontos. Entretanto, esses métodos diferem da AHT quanto ao grau de liberdade permitida na redefinição dos limites do parâmetro, isto é, o grau de flexibilidade na colocação e escolha do formato da janela que define a faixa de parâmetros em estudo.

FIGURA 3.9 - Detecção do centro do círculo usando a AHT. (a) Imagem. (b) Espaço de parâmetros (limites $a = 0$ a 128 , $b = 0$ a 128). (c) Acumulador grosseiramente quantizado.

Quando a imagem se torna muito complexa, os efeitos de quantização torna sua interpretação mais difícil. Nesse caso é necessário adotar estratégias com um certo grau de complexidade, envolvendo combinações de deslocamento, expandindo e reduzindo a faixa de parâmetros do atual interesse. Muitas vezes o método tem sido desviado para resoluções grosseiras, onde padrões de votos estendidos de 2 ou mais objetos podem sobrepor-se e produzir um pico falso, cuja posição não é relacionada aos parâmetros de qualquer traço real na imagem.

3.3.4 - Transformada de Hough Combinatorial

A Transformada de Hough Combinatorial (CHT) de BEN-TZVI & SANDLER (1990) é apresentada como um algoritmo que tem um desempenho melhor quando comparada com a HT padrão. Essa, calcula os

parâmetros associados com todas as possíveis combinações de segmentos de linha de 2 pontos, ao invés de calcular todos os possíveis valores de ρ_j , para a faixa toda de valores θ_j . As informações utilizadas pela CHT são as disponíveis na distribuição extraída dos pontos da imagem; com isso não fica dependente somente das informações retiradas do espaço da transformada.

Uma das dificuldades da HT é a implementação em tempo real, devido ao montante de computação envolvida em calcular ρ_j para a faixa inteira de θ_j em cada ponto da imagem.

Cada dois pontos na imagem indica uma possível colinearidade. Os parâmetros $(\rho-\theta)$ desse segmento de linha podem ser derivados da equação paramétrica da HT. Três pontos colineares, significa que suas respectivas senóides se interceptam em um ponto comum no plano de parâmetros, isto é, as 3 combinações de intersecção de dois pontos tem os mesmos valores de $(\rho-\theta)$. Para determinar se qualquer parte de um conjunto de "m" pontos são colineares, toda as possíveis combinações de segmentos de linhas de 2 pontos entre eles necessitam ser calculados. Cada um desses cálculos, cada par de pontos, contribui com um ponto para o arranjo acumulador no plano $(\rho-\theta)$ da intersecção, como pode ser visto na Fig. 3.10.

Quanto maior o número de pontos na imagem, maior será o número de

$$C_m^2 = \frac{m!}{2!(m-2)!} \quad (3)$$

ciclos computacionais necessários, ou seja, maior o valor de C_m^2 . Quando a quantidade C_m^2 é muito elevada utiliza-se o recurso de segmentação da imagem. Com isso cada segmento é trabalhado separadamente, mas todos os segmentos irão atualizar o mesmo arranjo acumulador.

A segmentação da imagem é um recurso usado por vários autores. Em WALLACE (1985) a segmentação da imagem considera que segmentos da mesma linha contribuem separadamente para a mesma entrada no espaço da transformada. Entretanto, ainda é feita uma procura através de todos os valores possíveis do parâmetro dependente, ao invés dos mais prováveis.

Segmentar a imagem neste caso é equivalente a uma redução na resolução, que pode não ser aceitável.

FIGURA 3.10 - (a) Linha com cinco pontos e seu respectivo arranjo acumulador. (b) O efeito de um ruído sobre o espaço da transformada

O aumento do número de segmentos nos quais a imagem é divulgada (reduzindo o tamanho de cada segmento) melhora a performance do tempo da CHT, isso porque existem menos pontos de borda num segmento. Contudo, a resolução em θ dita o tamanho do arranjo acumulador requerido.

Na avaliação dos resultados da implementação da HT e CHT para as imagens da Fig. 3.11, onde todas são imagens 128 x 128 "pixels", foi utilizado apenas um único processador. Aplicando a CHT, as imagens foram divididas em 64 segmentos (8 x 8) de 21 x 21 "pixels" cada, com sobreposição de 5 "pixels" entre cada 2 segmentos adjacentes (verticalmente e horizontalmente). A HT é aplicada com uma resolução de 128 em θ . Pelos dados da tabela 3, observa-se que a CHT mostrou-se superior sobre a HT.

FIGURA 3.11 - Três tipos de imagens utilizadas na comparação da CHT com a HT

TABELA 3 - Comparação do tempo de desempenho da HT segundo Hough(1962) com CHT.

Imagem	Nº de Pontos de borda na imagem	Nº de ciclos de computação na HT(mN)	Nº de ciclos de comput. na CHT (nº de combinações de pares)
Fig.3.11 a	188	24064	1423
Fig.3.11 b	584	74752	6019
Fig.3.11 c	378	48384	5669

Na CHT, a distribuição dos pontos nas bordas da imagem influenciam o tempo de computação. O desempenho melhor ocorre quando os pontos estão distribuídos uniformemente por toda a imagem, de modo que a segmentação é totalmente explorada. O caso pior é quando todos os pontos da borda residem em 1 segmento.

A principal desvantagem do algoritmo está na dependência de sua performance de tempo sobre a distribuição dos pontos de borda na imagem.

3.3.5 - Transformada de Hough Fuzzy

O método desenvolvido por HAN et al.(1994) preserva a idéia básica de HT, mas permite detecção de formas aproximadas utilizando a teoria Fuzzy, um meio de lidar com conceitos aproximados. Assim, pontos ruidosos em imagens complicadas são eliminados na inspeção. Tal método foi chamado de Transformada de Hough Fuzzy (Fuzzy HT). Essa técnica descobre formatos ajustando os pontos de dados, o que evita os formatos falsos detectados quando se usa a HT convencional.

FIGURA 3.12 - Pontos para detecção de linhas.

Na Fig.3.12 aplicando a HT padrão encontra-se quatro retas com quatro pontos cada. O resultado pode ser visto na Fig.3.13.

A Fuzzy HT sugere uma análise na região através de uma convolução $W(R)$, onde R seria um valor coerente para as necessidades da imagem. Tal valor é testado experimentalmente.

$$W(x,y) = W(\sqrt{x^2 + y^2}) \quad (4)$$

FIGURA 3.13 - Quatro linhas encontradas com quatro pontos cada.

Considere-se $W(R)$ como um valor sendo não negativo, decrescendo gradativamente de $W(0)$ a $W(R) = 0$ para algum raio " R ". Com isso, a função da Transformada é afetada não apenas por pontos da imagem, diretamente nas linhas, mas também, por aqueles localizados dentro da distância " R " das linhas. Assim, ao invés de transformar cada ponto da imagem (x_i, y_i) para o espaço de parâmetros, transforma-se todos os pontos dentro da distância " R " na imagem (x_i, y_i) .

FIGURA 3.14 - Na distribuição do arranjo, os parâmetros da linha central terão o maior valor de acumuladores que qualquer ponto no espaço de parâmetro.

Entretanto, isso aumenta o tempo de computação de HT, pois o número de pontos dentro de um círculo é proporcional ao quadrado de seu raio

mostrado na Fig.3.14. Para contornar esse problema ao invés de se considerar todos os pontos dentro da região do círculo, pode-se ter o mesmo efeito transformando-se o resultado em apenas um ponto, como a HT padrão. Quando se obtiver todos os pontos resultantes aplica-se uma convolução unidimensional no espaço de parâmetros, ao longo do eixo de ρ nos casos de detecção de linha e de círculo.

A diferença entre a HT e a Fuzzy HT é a máscara $g(d)$. Tal máscara é a seguinte função exponencial:

onde k , σ e R são constantes escolhidas aleatoriamente.

$$g(d) = k e^{\frac{-d^2}{\sigma^2}} \begin{cases} \text{se } d < R \\ 0 & \text{outros} \end{cases} \quad (5)$$

- k = somente valores positivos
- σ = valores de modo que g é numericamente não muito íngreme.

A máscara foi usada nas experiências para convoluir o acumulador antes de detectar a posição máxima e os pontos de imagem dentro da distância “ R ”, de uma linha ou círculo. Os resultados foram satisfatórios tanto para círculo quanto para retas.

Quando a imagem for nítida, os segmentos de borda forem densos e com pouco ruído, a HT convencional irá dar bons resultados. Se os dados forem esparsos e ruidosos, a Fuzzy HT com apropriada escolha de σ , funciona muito bem. Entretanto, o problema surge para determinar os valores de σ e R , pois dependendo dos valores escolhidos, surgirão resultados diferentes.

3.3.6 - Sistema de HT baseado em Redes Neurais

Um sistema de inspeção automatizado sugerido por HOU, T.H. et al. (1993) baseou-se em Redes Neurais (RN) para aplicações em Tecnologia de Montagem em Superfície.

Os resultados experimentais mostraram que a HT pode efetivamente reduzir o montante de dados de processamento enquanto preserva as posições das bordas e a informação de orientação. A RN de retropropagação, com os dados do arranjo acumulador, é capaz de classificar o status de qualidade com alto desempenho. Essa rede é formada por 3 camadas desenvolvidas por RUMELHART et al.⁵ apud HOU, T.H. (1993). As RN's são conhecidas pela sua capacidade de aprendizado e adaptação a ruídos em padrões de entrada. Além disso, não requerem uma distribuição gaussiana no padrão de entrada.

Pelo fato da HT exigir um intenso processamento computacional, DEMPSEY & MCVEY (1992) sugeriram um sistema baseado em Redes Neurais para auxiliar no desempenho quanto à velocidade dos resultados. Um circuito analógico foi proposto para o mapeamento do espaço de parâmetros e uma rede de otimização Hopfield para a detecção dos picos. Com a aplicação de amplificadores operacionais tem-se conseguido menores tempos.

O grande problema na implementação em hardware tem sido fazer as conexões internas e externas. E também, a quantidade de CI's e componentes eletrônicos envolvidos.

Algumas possíveis áreas de futuras pesquisas incluem a otimização do projeto de amplificadores operacionais (neurônios) para implementação com VLSI, investigação da implementação óptica e investigação da transformada modificada usando a arquitetura neural.

⁵RUMELHART et al. *Learning internal representations by error propagation*. Cambridge, McClelland, 1986 apud HOU, T.H. et al. A neural network-based automated inspection system with an application to surface mount devices, New York, International Journal of Production Research, v.31, n.5, 1993. p.1171-1187

3.4 - Aplicações e uso da HT na Detecção Automatizada de Traços

3.4.1 - Sensoriamento Remoto

O sensoriamento remoto é uma técnica utilizada nas pesquisas por cientistas para coletar imagens oriundas de satélites. Existem vários tipos de sensores que rastreiam todo o planeta, enviando diversas informações sobre o mapeamento da superfície. Entretanto, tais informações, imagens, estão sujeitas a muitos ruídos e distorções que dificultam a sua interpretação. WANG & HOWARTH (1990) utilizaram a técnica de HT para a análise dessas imagens. Esse trabalho faz uma comparação entre os resultados de uma interpretação automática com a visual.

Na interpretação visual, o mapeamento dos traços das imagens são analisados por cientistas se que utilizam de seus conhecimentos e suas experiências para traduzir o que representa a imagem.

Na interpretação automática utilizando a HT, obteve-se bons resultados em relação à interpretação visual. A imagem usada foi a do LANDSAT5, mostrada na Fig. 3.15, a qual mostra parte da província de Grenville.

FIGURA 3.15 - Imagem do Landsat (512 x 512 "pixels"). Área aproximada de 15 x 15 km.

O procedimento do algoritmo envolve o uso de detectores de bordas juntamente com um seguidor de bordas produzindo uma imagem binarizada conforme pode ser visto na Fig. 3.16.

FIGURA 3.16 - Imagem do LANDSAT após ser submetida ao detector de bordas.

Em seguida aplica-se a HT pelo método descrito por DUDA & HART (1972). Com a descoberta dos máximos locais, aplica-se a HT inversa e analisa-se o perfil das linhas.

Em muitas aplicações a interpretação visual fornece resultados seguros, contudo, em certos casos, o método automatizado pode identificar uma borda onde um intérprete humano não pode. Uma razão para isso é que olhos humanos são mais sensíveis às mudanças na parte escura da escala de cinza do que na parte mais clara (GONZALES & WINTZ, 1987). Assim bordas com o mesmo contraste nas duas pontas da escala de cinza podem ser tratadas diferentemente pelo intérprete. Portanto, com a HT, pode-se identificar mais traços do que são selecionados através da inspeção visual.

Deve ser apontado, contudo, que o desenho dos traços pela HT, também indicam muitas linhas não exibidas no mapa geológico. Essas linhas podem ser traços e algumas podem ser erros. As extensões totais dos traços detectados pela HT para cada imagem são mostrados na Tabela 4. WANG & HOWARTH (1990) utilizaram quatro valores de "threshold" (T) e um limite de

“pixels” para preenchimento de cada lacuna, WANG⁶ apud WANG; HOWARTH (1990).

A Fig.3.17 mostra quatro exemplos de traços de imagem produzidos aplicando-se HT inversa e análise de perfil de linha reta nos arranjos de acumuladores da Fig.3.16. Os resultados da comparação entre os dados da HT e o mapa geológico indicam que a identificação de traços automatizada é capaz de detectar melhor as falhas geológicas do que a interpretação visual de geólogos.

FIGURA 3.17 - Transformada de Hough da imagem do LANDSAT: imagens foram geradas com os seguintes parâmetros : (a) lacuna $sg=40$, "threshold" para seleção de máximos locais $T=80$. (b) $sg=40$, $T=75$. (c) $sg=40$ $T=70$ e (d) $sg=50$, $T=75$.

⁶WANG, J.F. A new automated linear-feature network detection and analysis (LINDA) system and its applications. Waterloo, Canada, 1988 apud WANG; HOWARTH. Use of the hough transform in automated lineament detection, IEEE Transaction on Geoscience and Remote Sensing, v.28, n.4, 1990. p.566

Uma comparação quantitativa foi executada usando o índice de efetividade "E".

$$E = \frac{L_{TM}}{L_{MAP}} * 100\% \quad (6)$$

Sugerido por STRONG⁷ apud WANG; HOWARTH (1990) onde:

- L_{TM} = extensão total dos traços (mm) comuns do traço da imagem e do mapa geológico.
- L_{MAP} = extensão total das falhas (mm) no mapa geológico.

TABELA 4 - Efetividade dos traços de imagem comparados com o mapa Geográfico

Imagem	Nº "Pixels"	Threshold máx	Comp. total (mm)	L_{TM} (mm)	L_{MAP} (mm)	Efetividade E (%)
Mapa geológico				1009	1009	100,00
Interp. visual				506	1009	50,15
Seguidor de borda				548	1009	54,31
HT-1	40	80	560	542	1009	53,72
HT-2	40	75	885	660	1009	65,41
HT-3	40	70	1405	695	1009	68,8
HT-4	50	75	1034	698	1009	69,18

Analisando-se os dados das experiências, a interpretação visual identificou aproximadamente 50% das falhas mostradas no mapa geológico. Os seguidores de borda resultaram num índice de efetividade logo acima de 54%. Os melhores resultados foram obtidos com os mapas de traços

⁷ STRONG, W.L. Remote sensing of seismic lines and other disturbance features associated with oil and gas development: A comparison of Landsat Thematic Mapper and multispectral scanner imagery. 1986 apud WANG; HOWARTH. Use of the Hough transform in automated lineament detection. IEEE Transaction on Geoscience and Remote Sensing, v.28, n.4, 1990. p.566

automatizados, onde índices de efetividade variam de 53,7 a 69,2% dependendo dos valores de "thresholds" utilizados.

3.4.2 - Sistema de Rastreamento com Radares

Na aplicação da HT em aquisição de dados por radar, CARLSON et al.(1994a), mostrou algumas vantagens comparadas com técnicas tradicionais. Essas vantagens incluem detecção melhorada, flexibilidade na implantação e aquisição pelo rastreamento. O conceito é similar aos algoritmos de detecção antes de rastrear, que usam informações preliminares de explorações prévias para auxiliar nas declarações de alvo.

O algoritmo utilizado é o HT padrão (DUDA & HART, 1972). Dois "thresholds" são aplicados como filtro, sendo que o segundo "threshold" serve como uma sintonia fina.

Um possível problema para esse método é quando a área abrangida pelo radar é muito grande. Tem-se assim um aumento de pontos falsos devido ao excesso de senóides no plano de parâmetros da HT e muitas vezes o segundo "threshold" não é suficiente para limitar. Uma alternativa é uma integração binarizada (CARLSON et al. 1994c). Outra alternativa é o uso de vários "thresholds" e espaço de parâmetros onde alvos largos são removidos para espaço de parâmetros menores e enviados para o próprio espaço de Hough.

3.4.3 - Enfoque hierárquico para extração de linha baseado na HT

Os métodos hierárquicos para a extração de linhas baseados na HT, tem sido classificados quanto ao número de categorias. Esses métodos diferem principalmente no enfoque usado para segmentar a curva. Hierarquias bidimensionais incluem métodos de topo para baixo e de baixo para cima. O método de topo para baixo descrito por SCHNEIER (1982) usa

versões de resolução reduzida para guiar a localização de estruturas lineares. Esse método é usado especialmente para a detecção de intensidades de cristas numa imagem ao longo das linhas. Bordas "quadtree" (SCHNEIER, 1981) é outro método de topo para baixo que fornece uma aproximação linear peça por peça para um mapa de borda. Em cada nó o limite é representado pela direção, intersecção, magnitude e erro direcional da melhor borda que passa através da região. A imagem é dividida sucessivamente independente do nó, até se obter uma aproximação satisfatória. Entretanto, tais divisões dependem, principalmente, do tamanho do termo do erro direcional.

O algoritmo de PRINCEN, I. et al.(1990) foi baseado numa estrutura de pirâmide com cada camada da pirâmide dividindo a imagem completa num número de sub-imagens. Inicialmente é feita uma descrição local dos contornos formados por segmentos de linha, com isso aplica-se a HT.

Em cada sub-imagem um segmento de linha é representado como um conjunto de segmentos de linha de nível baixo, com parâmetros particulares θ e ρ . Isso significa que no menor nível, um grupo consiste de um conjunto com um membro e cada sub-imagem irá conter uma linha de grupos que representam os segmentos de linha passando através dela. O método procede numa maneira hierárquica de baixo para cima, agrupando segmentos de linha dentro das vizinhanças locais, dando origem ao formato do objeto em estudo.

O mecanismo de agrupamento utilizado é baseado na relação entre tamanho dos segmentos e a resolução da representação. Isso leva, naturalmente, aos critérios de que restrições no agrupamento colinear de linhas longas devem ser mais rigorosas do que para linhas mais curtas. O uso da HT para estabelecer esses agrupamentos, ao invés de comparação sequencial par a par, é um meio eficiente e natural para incorporar os critérios de resolução.

WEISS & BOLDT (1986) descreveram um método de agrupamento de baixo para cima que não é baseado numa estrutura de pirâmide fixa. Esse método consiste de 2 estágios. Uma etapa de ligação é usada para estabelecer possíveis grupos com base na proximidade, contraste e

colinearidade. Cada segmento de linha pode ser ligado a um número de outros segmentos de linha que caem na vizinhança definida pelo raio de ligação. No segundo estágio, possíveis grupos de linha determinados da ligação são testados para dar continuidade, ajustando-se uma linha ao conjunto dos pontos finais das linhas, contribuindo e calculando uma medida de continuidade. Um raio de procura distinta é usado para determinar os segmentos de linhas consideradas para fusão. Se a medida de continuidade está abaixo de um valor de "threshold" pré determinado, então o grupo é aceito. O processo é sequencial e segmentos de linha são fundidos no primeiro grupo encontrado. Os segmentos não fundidos também são propagados para um número limitado de níveis.

O fato da estrutura hierárquica, em sua implementação, não ser regular, resulta num algoritmo mais livremente determinado. Uma estrutura regular, todavia, permite um algoritmo mais eficiente e que deve ser mais fácil de implementar em uma arquitetura paralela. Esta fornece uma divisão do trabalho em cada um dos níveis num número de processos independentes que podem ser implementados com apenas um pequeno montante de memória local e computações bem simples.

3.4.4 - Localização e reconhecimento de múltiplas imagens com HT

O reconhecimento e a localização de um objeto conhecido são tarefas fundamentais que traduzem em complexidade e adaptabilidade de vários métodos. NAGATA & ZHA (1991) apresentaram um enfoque para reconhecer e localizar um objeto parcialmente visível em imagens múltiplas. Os dados de entrada são várias imagens em nível de cinza, sob diferentes condições de iluminação. O método estabelece a correspondência imagem-objeto, através de um exame da consistência entre os traços extraídos da superfície das imagens e os traços descritos do objeto modelo.

A descrição de cena é baseada nos traços de superfície, que são vistos como regiões isoladas, conforme Fig.3.18. Assim, tem-se a flexibilidade de

incluir tanto superfícies planares, como esféricas, cilíndricas e cônicas. Para essa descrição, NAGATA & ZHA (1991) usaram um método estéreo fotométrico modificado, criando um mapa de traços. Através dos traços resultantes são identificadas e descritas usando-se uma HT modificada baseada nos mapas esféricos gaussianos dos traços. A reconstrução da imagem se dá por estratégias de planejamento heurístico baseados no conceito de conjuntos fuzzy.

FIGURA 3.18 - As subdivisões de uma imagem.

3.4.5 - Aplicações em Imagens Ruidosas

A performance e sensibilidade da distribuição de imagens ruidosas foram largamente estudadas por HUNT et al. (1990) para a detecção de linha pela HT. Pelos testes nas imagens, o processador Gaussiano e HT são encontrados como sendo muito menos sensíveis a certos tipos de ruídos do que o processador Laplaciano. Quando em um ambiente a distribuição do ruído é desconhecida, a HT e o processador Gaussiano podem ser relativamente robustos. Entretanto, se a distribuição do ruído é conhecida e não é gaussiano, a performance pode ser melhor obtida usando um processador apropriado da teoria de detecção do sinal. Os dois processadores da teoria de detecção de sinal são o detector ótimo para ruído aditivo e gaussiano e o detector ótimo para ruído aditivo e Laplaciano.

3.5 - Implementação em Multiprocessadores com o Uso da HT

A HT, devido à sua complexidade computacional não é facilmente implementável em tempo real. Um enfoque para se conseguir implementação em tempo real é pelo processamento paralelo. Várias configurações foram sugeridas para isso. Estas diferem na forma pela qual a imagem e os espaços de transformada são distribuídos entre os elementos processadores (EP), que são refletidos na maneira que a memória é organizada.

As possíveis configurações são:

- (1) - Quando todos os EP's tem acesso aos pontos de toda a imagem, com cada um computando a HT para um diferente segmento do espaço da transformada, ou seja, memória de imagem global e memória distribuída.
- (2) - Cada EP tem acesso a um diferente segmento de pontos da imagem e computa a HT para o espaço inteiro da transformada, ou seja, memória de imagem distribuída e memória de transformada global.
- (3) - Cada EP tem acesso a um segmento de pontos da imagem e computa a HT para um diferente segmento do espaço da transformada, ou seja, memória de imagem distribuída e memória de transformada distribuída (ROSENFELD, 1988).

No trabalho realizado por OLSON et al.⁸ apud BEN-TZVI et al. (1990) todos os EP's dão acesso simultaneamente à mesma entrada numa memória de imagem. Entretanto, esse tipo de acesso à memória da imagem torna-se como um sistema uniprocessador.

A terceira configuração implementada por ROSENFELD (1988) mostra um processador paralelo SIMD conectado por malha não tão eficiente, pois envolve muito "overhead" no deslocamento global e uma soma de dados de todas as partes da malha.

BEN-TZVI et al. (1990) propôs um sistema utilizando a 3ª configuração num processador paralelo MIMD, usando uma arquitetura de memória

⁸OLSON et al. *Low level image analysis on an MIND architecture*. London, 1987. apud BEN-TZVI et al. *Synchronous multiprocessor implementation of the Hough transform*, London, Computer Vision Graphics and Image Processing, v52, 1990. p.438

segmentada "Cross-bar", para permitir aos EP's terem acesso a diferentes segmentos de memória de imagem.

Na implementação paralela da transformada, cada EP calcula a HT para todos os pontos de borda (x_i, y_i) em diferentes faixas de valores de θ . Em termos de configuração de memória isso significa distribuir a memória da HT em uma dimensão, isto é, em apenas θ , como mostrado na Fig.3.19. A memória da imagem é simultaneamente distribuída onde cada EP processa blocos de memória de imagem para diferentes valores y .

FIGURA 3.19 - Arquitetura da distribuição de memória de imagem (MI) e memória da HT (MHT).

Todos os segmentos de memória de imagem são processados sincronamente, de modo que, em qualquer tempo, todos os EP's tem acesso aos "pixels" com a mesma posição relativa em seus segmentos. Cada EP calcula a HT para a sua faixa de valores θ e para coordenada (x, y) de "pixels" não zero em seu segmento, como também de outros. Para evitar problemas no acesso da memória da imagem, cada EP acessa seu próprio segmento de memória e transmite os dados lidos para outros EP's.

A implementação de uma arquitetura paralela da HT mostrou-se eficiente comparada com um sistema dedicado de HT. Várias pesquisas tem sido realizadas com o intuito de alcançar novos progressos no processamento paralelo de imagens com a HT (ROSENFELD et al., 1988).

CAPÍTULO 4 - APLICAÇÃO DA HT PARA ANÁLISE DE PLACAS RETANGULARES DE MADEIRA

Em linhas de produção industrial, investe-se hoje uma grande soma de recursos para se conseguir a “qualidade total”. Uma das tarefas mais difíceis nesse processo é a inspeção visual dos produtos. Com avanços na tecnologia da informática, processamento de imagens, reconhecimento de padrões e inteligência artificial, consegue-se equipamentos de análise de imagens industriais melhores e mais baratos.

A HT tornou-se uma técnica muito importante para a análise de imagens. Também é usada na visão de máquinas e em reconhecimento de objetos. Com a avaliação e filtragem do arranjo acumulador para a inspeção de objetos pode-se rapidamente determinar características simples de inspeção visual tais como, valores dos lados, perímetro, área e localização do centro de massa de objetos 2D. Isso permite a aplicação do processo possibilitando a implementação de IVA na indústria, a custos bastante reduzidos.

Analisando as várias derivações da HT, o modelo que demonstrou melhor desempenho para aplicação em IVA foi a HT segundo DUDA & HART (1972). A escolha foi feita devido à facilidade de implementação e velocidade de processamento, frente às características a serem obtidas neste trabalho.

Apesar das outras implementações da HT fornecerem resultados mais precisos e otimizados, a literatura referenciada no capítulo 3 demonstra que a velocidade de processamento nesses casos, torna inviável a aplicação neste trabalho.

O objeto inspecionado aqui, é bastante simples. Consiste de uma placa de madeira retangular com área e dimensões definidas, e o algoritmo

implementado para a inspeção foi desenvolvido em Linguagem "C", utilizando o compilador da Borland versão 3.1 para DOS.

4.1 - Sistema de Inspeção

O Sistema de Inspeção mostrado na Fig.4.1 é composto pelos seguintes elementos:

- ✧ um computador 486 DX2 - 66mhz;
- ✧ uma placa "frame-grabber";
- ✧ uma câmera CCD modelo KP - 110 HITACHI;
- ✧ quatro spots de luz de 50 watts;

FIGURA 4.1 - Elementos de Inspeção

O algoritmo desenvolvido para o processamento da imagem pode ser resumido nas seguintes etapas do diagrama de blocos, conforme Fig.4.2.

Inicialmente o bloco "Processamento de Imagem" irá capturar a imagem recebida pela câmera e descompactá-la enviando-a ao bloco seguinte. Na segunda etapa, no bloco "Processamento da HT", serão

terceira etapa, o bloco “Resultados”, com os dados obtidos dos cálculos da HT serão feitas as filtrações do arranjo acumulador obtendo os valores como área, perímetro e centro de massa independentes de rotação ou translação do objeto. Cada etapa será descrita com maiores detalhes no decorrer deste capítulo.

FIGURA 4.2 - Diagrama de Blocos do Sistema de Inspeção

4.2 - Análise da Imagem

Para a iluminação do objeto foram utilizados quatro spots de 50 watts a uma distância de 60cm do objeto. Tal iluminação permite que não ocorra sombras nos quatro lados do objeto. Uma iluminação mal planejada prejudica a extração de dados das imagens.

FIGURA 4.3 - Placa de madeira usada na inspeção

A imagem é capturada por uma câmera CCD, (Dispositivo Acoplado por Carga) modelo KP - 110 HITACHI. Essa imagem é enviada a uma placa "Frame-Grabber" a qual irá projetá-la em uma tela de vídeo com 256 níveis de cinza.

O algoritmo processa essa imagem e seleciona uma região ativa que contém a imagem de 128 X 128 "pixels" e a grava na forma de arquivo. São gravados também vários arquivos, rotacionando e transladando a mesma placa de madeira. Esses arquivos servirão de base para medir a performance do sistema.

FIGURA 4.4 - Histograma

Como a imagem da Fig.4.3 é em escala de cinza, é necessário que se utilize um detector de bordas para realçar os contornos da mesma, preparando-a para a HT.

Pela análise do histograma da placa de madeira, observa-se que a maior quantidade de pontos da placa pertence a uma região cuja largura de banda está em torno de 30, conforme a Fig.4.4. Dessa forma, pode-se manipular o contraste da imagem e consequentemente detectar os limites da mesma.

Aplicando-se um detector derivativo simples sobre a imagem, considerando um “threshold” de 30, obtém-se uma imagem de bordas para que possa ser usada a HT. O detector de borda considerado realiza pouquíssima computação, aumentando a velocidade e obtendo resultados satisfatórios para a aplicação em questão.

FIGURA 4.5 - Três pontos em uma imagem

$$Dif_1 = |A(i, j) - A(i, j + 1)| \quad \text{Eq.4.1}$$

$$Dif_2 = |A(i, j) - A(i + 1, j)| \quad \text{Eq.4.2}$$

Observando-se a Fig.4.5 tem-se que:

$$A_{ij} = \begin{cases} 1 \text{ (fundo)} & \text{se } \begin{cases} Dif_1 > 30 \\ Dif_2 < 30 \\ \text{ou} \\ Dif_1 < 30 \\ Dif_2 > 30 \end{cases} \\ 0 \text{ (borda)} & \text{se } \begin{cases} Dif_1 \geq 30 \\ Dif_2 \geq 30 \\ \text{ou} \\ Dif_1 \leq 30 \\ Dif_2 \leq 30 \end{cases} \end{cases}$$

Assim, aplicando esse detector de borda sobre a Fig.4.6(a) obtém-se apenas o contorno do retângulo, conforme mostra a Fig.4.6(b). Existem vários tipos de detectores de bordas (DAVIS(1975), HUECKEL(1973), LOW(1991), SCHALKOFF(1989)), mas esta técnica mostrou-se eficiente no tratamento da imagem, visto que nas aplicações em IVA, a velocidade de processamento é fundamental.

FIGURA 4.6- (a) Imagem Original; (b) Aplicando o detector de bordas

Depois de aplicar o detector de borda, faz-se uma varredura sobre a imagem e aloca-se dois ponteiros, "valores_vet_x" para as coordenadas de "x" e "valores_vet_y" para as coordenadas de "y". Assim, durante a execução do programa é possível utilizar os valores dos ponteiros e evitar desperdício de alocação de memórias com números desnecessários, já que as matrizes alocam 580.000 células para o arranjo acumulador, as quais nem todas são preenchidas com pontos referentes à imagem.

A HT é um mapeamento do plano da imagem no espaço de parâmetros. Nessa parametrização uma linha reta é representada de acordo com a Fig.4.7, conforme visto no Capítulo 3,

$$\rho = x \cos \theta + y \sin \theta \quad \text{Eq. 4.3}$$

FIGURA 4.7 - Representação geométrica de Hough

onde ρ é a distância do segmento de linha da origem do sistema de coordenadas da imagem, e θ é o ângulo da normal com o eixo "x" positivo, e "P" é um ponto qualquer na reta "r".

No Anexo-O consta o programa em Linguagem C para o cálculo do arranjo acumulador. Inicia-se utilizando os valores dos ponteiros de "x" e "y" que representam cada ponto da Fig.4.6(b). A cada valor de θ , o qual varia em um determinado intervalo Δ , calcula-se para todos os pontos da imagem os valores de ρ respectivos, de acordo com a Eq. 4.3. Na tabela do arranjo acumulador a cada posição (ρ, θ) calculados, incrementa-se de 1, conforme Fig.4.8. Dessa forma durante toda a variação de θ as células do arranjo

acumulador serão incrementadas produzindo assim o espaço de parâmetros da imagem analisada.

FIGURA 4.8 - Arranjo acumulador (a) Diagrama de blocos; (b) Construção da Tabela

FIGURA 4.9 - (a) Gráfico do arranjo acumulador ;(b) Projeção no plano imagem.

A Fig.4.9(a) mostra o gráfico representativo do arranjo acumulador onde foi definido o intervalo de teta entre $-90^\circ \leq \theta \leq 90^\circ$. Como os picos 1 e 2 estão exatamente sobre o ângulo -90° , pela propriedade de reflexão da HT, vista no Capítulo 3, ocorre a formação dos picos 5 e 6 em 90° . Consequentemente o arranjo acumulador apresenta a formação de seis aglomerações de picos.

Na Fig.4.9(b) tem-se o resultado da projeção do arranjo acumulador da Fig.4.9(a) no plano imagem. Observa-se que os picos 5 e 6 são responsáveis pela determinação das retas 5 e 6 as quais produziram o retângulo $A'B'C'D'$, com as mesmas proporções que a imagem real, o retângulo ABCD.

O intervalo de teta utilizado no arranjo acumulador estabelecido neste trabalho foi de $-90^\circ \leq \theta \leq 89^\circ$. Caso isso não seja feito, em imagens cujos pontos representativos coincidam nessa posição, e possuindo a imagem quatro retas, o arranjo acumulador irá detectar seis retas.

A quantização do arranjo acumulador é feita de modo que todos os valores de ρ sejam calculados e fiquem simétricos ao eixo. Com isso, quando se obtém cada resultado de ρ verifica-se no arranjo qual a posição que deverá ser incrementada, seguindo os intervalos de quantização, conforme Fig.4.10. Assim obedece-se a propriedade de reflexão da HT.

FIGURA 4.10 - Intervalos de quantização

Para obter-se uma melhor definição no arranjo acumulador, em relação aos pontos elevados, que representam as retas, foram reduzidos os valores de $\Delta\rho$ e $\Delta\theta$. O valor utilizado neste trabalho foi de "1", tanto para $\Delta\rho$ como para $\Delta\theta$.

No início do programa definiu-se como global uma “ Matriz Huga” de 300 linhas por 180 colunas e aumentou-se o tamanho reservado para a pilha, para se evitar estouro de memória pelo compilador. O motivo de ser 180 colunas é devido ao valor de $\Delta\theta$ ser igual a "1". Já o valor de 300 linhas foi um ponto máximo encontrado durante os cálculos.

Uma comparação com diferentes valores de $\Delta\rho$ e $\Delta\theta$, são mostrados em 6 exemplos (Fig.4.11). Nota-se que as mudanças no gráfico do arranjo acumulador são bem visíveis e que quanto menores forem os valores de $\Delta\rho$ e $\Delta\theta$, melhor será o reconhecimento dos pontos em estudo. Nas Fig.4.11 (a), (b) e (c) as linhas dos gráficos estão distorcidas e com pouca qualidade na resolução. A partir de (d), (e) e (f), observa-se uma melhor qualidade na formação dos quatro picos principais: dois grupos de picos elevados que representam os lados maiores e dois grupos de picos baixos que correspondem aos lados menores.

FIGURA 4.11 - Gráfico de Arranjos Acumuladores com valores diferentes de $\Delta\rho$ e $\Delta\theta$. (a) $\Delta\rho$ e $\Delta\theta = 20$, (b) $\Delta\rho$ e $\Delta\theta = 10$, (c) $\Delta\rho$ e $\Delta\theta = 5$, (d) $\Delta\rho$ e $\Delta\theta = 3$, (e) $\Delta\rho$ e $\Delta\theta = 2$ e (f) $\Delta\rho$ e $\Delta\theta = 1$.

4.3 - Filtragem do Arranjo Acumulador

A projeção do arranjo acumulador em um gráfico tridimensional, conforme Fig.4.12(c), mostra uma visão mais clara das grandezas analisadas. O eixo "x" representa as coordenadas de θ $[-90^\circ, 89^\circ]$, e no eixo "y" os valores de ρ . O eixo "z" indica a quantidade de pontos em cada célula da matriz (θ, ρ) .

FIGURA 4.12 -(a) Placa de madeira analisada; (b) Imagem após o detector de bordas; (c) Gráfico referente ao arranjo acumulador; (d) Gráfico nº. de pontos por eixo- ρ ; (e) Gráfico nº. de pontos por eixo- θ

Como a imagem estudada é de uma placa de madeira formada por quatro retas ortogonais, após o detector de bordas (Fig.4.12(b)), pelo arranjo

acumulador observa-se a formação de quatro concentrações de picos. Esses picos, ou seja, esses conjuntos de células, representam as possíveis posições das retas no plano da imagem, pois possuem grande quantidade de pontos. Ordena-se, então, os valores das células em ordem decrescente e faz-se uma filtragem grosseira, ou seja, aplica-se a função "threshold" que irá zerar os valores abaixo do limite determinado, deixando uma certa porcentagem de pontos no arranjo, para realizar os cálculos. Essa porcentagem é estipulada inicialmente e depende do objeto analisado. Para a placa de madeira, foi estabelecido a busca dos 50 maiores valores do arranjo acumulador.

FIGURA 4.13 - (a) Arranjo acumulador após filtragem grosseira, (b) Projeção no plano imagem

Na Fig.4.13(a) tem-se o resultado de uma filtragem grosseira, referente à imagem da Fig.4.12(b). Observa-se a formação de quatro conjuntos de células. A finalidade desse tipo de filtragem é limitar a região onde se encontram os picos principais. Pela propriedade da HT, conforme Capítulo 3, cada ponto no espaço de parâmetros corresponde a uma reta no plano imagem. Assim, na Fig.4.13 (b) tem-se a representação no plano imagem das 50 retas mais prováveis obtidas através da filtragem grosseira.

Os grupos de retas 1, 2, 3 e 4 correspondem respectivamente aos conjuntos de células indicadas nos picos 1, 2, 3 e 4, conforme Fig.4.13(a). Na Tabela 4.1 são mostrados os 50 pontos restantes após a filtragem grosseira.

TABELA 4.1 - Resultado da Filtragem Grosseira

Matriz		ρ	θ	
1 - Valor da Matriz	20 - Valor RO =	-80 - Valor TETA =	-67	Grupo I
2 - Valor da Matriz	29 - Valor RO =	-79 - Valor TETA =	-66	
3 - Valor da Matriz	21 - Valor RO =	-78 - Valor TETA =	-65	
4 - Valor da Matriz	25 - Valor RO =	-76 - Valor TETA =	-64	
5 - Valor da Matriz	24 - Valor RO =	-75 - Valor TETA =	-63	
6 - Valor da Matriz	20 - Valor RO =	-74 - Valor TETA =	-62	
7 - Valor da Matriz	22 - Valor RO =	-72 - Valor TETA =	-61	
8 - Valor da Matriz	21 - Valor RO =	-71 - Valor TETA =	-60	
9 - Valor da Matriz	24 - Valor RO =	21 - Valor TETA =	-66	Grupo II
10 - Valor da Matriz	21 - Valor RO =	22 - Valor TETA =	-63	
11 - Valor da Matriz	23 - Valor RO =	23 - Valor TETA =	-62	
12 - Valor da Matriz	21 - Valor RO =	44 - Valor TETA =	14	Grupo III
13 - Valor da Matriz	21 - Valor RO =	44 - Valor TETA =	15	
14 - Valor da Matriz	26 - Valor RO =	45 - Valor TETA =	15	
15 - Valor da Matriz	20 - Valor RO =	46 - Valor TETA =	15	
16 - Valor da Matriz	34 - Valor RO =	46 - Valor TETA =	16	
17 - Valor da Matriz	24 - Valor RO =	47 - Valor TETA =	16	
18 - Valor da Matriz	37 - Valor RO =	47 - Valor TETA =	17	
19 - Valor da Matriz	21 - Valor RO =	48 - Valor TETA =	15	
20 - Valor da Matriz	21 - Valor RO =	48 - Valor TETA =	16	
21 - Valor da Matriz	48 - Valor RO =	48 - Valor TETA =	17	
22 - Valor da Matriz	31 - Valor RO =	48 - Valor TETA =	18	
23 - Valor da Matriz	30 - Valor RO =	49 - Valor TETA =	16	
24 - Valor da Matriz	33 - Valor RO =	49 - Valor TETA =	17	
25 - Valor da Matriz	27 - Valor RO =	49 - Valor TETA =	18	
26 - Valor da Matriz	26 - Valor RO =	49 - Valor TETA =	19	
27 - Valor da Matriz	26 - Valor RO =	50 - Valor TETA =	19	
28 - Valor da Matriz	22 - Valor RO =	50 - Valor TETA =	20	
29 - Valor da Matriz	21 - Valor RO =	50 - Valor TETA =	21	
30 - Valor da Matriz	20 - Valor RO =	51 - Valor TETA =	20	
31 - Valor da Matriz	35 - Valor RO =	51 - Valor TETA =	21	Grupo IV
32 - Valor da Matriz	45 - Valor RO =	51 - Valor TETA =	22	
33 - Valor da Matriz	24 - Valor RO =	52 - Valor TETA =	21	
34 - Valor da Matriz	34 - Valor RO =	52 - Valor TETA =	22	
35 - Valor da Matriz	61 - Valor RO =	52 - Valor TETA =	23	
36 - Valor da Matriz	65 - Valor RO =	53 - Valor TETA =	22	
37 - Valor da Matriz	60 - Valor RO =	53 - Valor TETA =	23	
38 - Valor da Matriz	68 - Valor RO =	53 - Valor TETA =	24	
39 - Valor da Matriz	32 - Valor RO =	54 - Valor TETA =	23	
40 - Valor da Matriz	40 - Valor RO =	54 - Valor TETA =	24	
41 - Valor da Matriz	23 - Valor RO =	80 - Valor TETA =	17	Grupo IV
42 - Valor da Matriz	26 - Valor RO =	80 - Valor TETA =	18	
43 - Valor da Matriz	23 - Valor RO =	81 - Valor TETA =	16	
44 - Valor da Matriz	29 - Valor RO =	81 - Valor TETA =	17	
45 - Valor da Matriz	44 - Valor RO =	81 - Valor TETA =	18	
46 - Valor da Matriz	52 - Valor RO =	81 - Valor TETA =	19	
47 - Valor da Matriz	30 - Valor RO =	82 - Valor TETA =	19	
48 - Valor da Matriz	24 - Valor RO =	83 - Valor TETA =	20	
49 - Valor da Matriz	31 - Valor RO =	84 - Valor TETA =	21	
50 - Valor da Matriz	58 - Valor RO =	85 - Valor TETA =	24	

A análise da Tabela 4.1 revela quatro grupos distintos de ρ 's e θ 's. A classificação das retas em cada um dos quatro grupos é feita considerando a distância relativa entre elas ou a variação angular. Dessa forma, varrendo-se o arranjo acumulador,

$$\text{se } \begin{cases} (\rho_n - \rho_{n+1}) \leq \Delta \\ \text{ou} \\ (\theta_n - \theta_{n+1}) \leq \Delta \end{cases}$$

onde n = posição do elemento na matriz, a reta pertence ao grupo analisado, caso contrário a reta pertence ao próximo grupo.

Assim, quando a diferença entre dois ρ 's ou dois θ 's for maior que um determinado intervalo (estabelecido aqui $\Delta=15$), considera-se a reta pertencente a outro grupo. Pode-se observar que cada grupo refere-se a um determinado lado da imagem. Dessa forma obtém-se retas distintas em grupos de células distintas. Como exemplo, tem-se na Tabela 4.1, a diferença do valor de ρ_8 para ρ_9 de 90, o que confirma a mudança de grupo.

Para a localização exata das quatro células é necessário fazer uma filtragem fina em cada grupo. Essa filtragem leva em consideração a inclinação entre cada reta, ou seja, seus valores de ângulos de θ relacionados com os valores de ρ e o valor máximo de cada ponto da célula na matriz, conforme Fig.4.14.

FIGURA 4.14 - Inclinação entre as retas

Devido a distribuição dos pontos no arranjo acumulador, conforme Fig.4.15, há a necessidade de se garantir a localização do ponto mais

significante, ou seja, a célula mais elevada. Dessa forma são realizadas duas varreduras no arranjo, uma no sentido vertical, fixando ρ e outra no sentido horizontal, fixando θ . Na Fig.4.15 é mostrado um trecho do arranjo acumulador referente ao Grupo III da Tabela 4.1.

A função que executa a filtragem fina chama-se "Filtro_Modificado()", mostrado no Anexo O. Essa função analisa cada grupo da Tabela 4.1, e verifica a célula que contém a maior quantidade de pontos. Descoberta essa célula, certifica-se se os valores de ρ e θ obedecem ao valor do intervalo adotado. Na análise da projeção do Grupo III da Tabela 4.1, conforme Fig.4.15, observa-se que os intervalos são obedecidos e a coordenada $\theta = 24$ e $\rho = 53$ corresponde ao maior pico.

FIGURA 4.15 - Projeção do arranjo acumulador do grupo III da Tabela 4.1

Na Tabela 4.2 tem-se as quatro coordenadas finais da placa de madeira (Fig.4.12(a)), como resultado dessa filtragem fina.

TABELA 4.2 - Resultado da Filtragem Fina

Retas	θ	ρ	Valor da Matriz
1	-66	-79	29
2	-66	21	24
3	24	53	68
4	24	85	58

Na Fig.4.16 é mostrado graficamente o resultado dessa filtragem tanto no plano de parâmetros como no plano imagem. A projeção das retas no plano imagem foram exatamente sobre as bordas da imagem analisada.

FIGURA 4.16 - (a) Arranjo acumulador após filtragem fina (b) Figura projetada no plano imagem

4.4 - Determinação de Características

4.4.1 - Cálculo dos lados, perímetro e área

Com os valores de ρ , θ , e o valor da célula obtidos do arranjo acumulador após a filtragem, calcula-se a área, o perímetro e o centro de massa do objeto em relação à imagem.

FIGURA 4.17 - (a) Gráfico de nº de pontos por eixo - ρ do arranjo acumulador; (b) Gráfico de nº de pontos por eixo - θ ; (c) Aplicação da Transformada inversa da HT.

Pela Fig.4.17(a), quando projetam-se os dados do arranjo já filtrado num gráfico de nº de pontos por eixo - ρ obtém-se os quatro valores de ρ 's. Como as placas são de mesmo formato e dimensão, variando a posição da mesma na tela, a distância entre $\overline{\rho_1\rho_2}$ e $\overline{\rho_3\rho_4}$ permanecem constantes, mesmo que a imagem seja rotacionada ou transladada.

Logo, obtém-se os seguintes segmentos (Fig. 4.17 (c)):

$$\overline{AB} = \overline{DC} = |\rho_1 + \rho_2|$$

$$\overline{AD} = \overline{BC} = |\rho_3 - \rho_4|$$

Assim, tem-se:

$$\begin{cases} \text{Área} = \overline{AB} * \overline{BC} \\ \text{Perímetro} = \overline{AB} + \overline{BC} + \overline{CD} + \overline{AD} \end{cases}$$

Um dado que pode ser obtido da Fig.4.17(b) é a determinação das bordas da imagem que são ortogonais e paralelas, duas a duas. No plano imagem, o segmento de reta $\overline{BC}$, contido na reta 2, conforme Fig.4.17(c), é paralelo ao segmento de reta $\overline{AD}$, contido na reta 1. Isso pode ser observado pelo mesmo valor de θ (θ_1). O mesmo acontece também com os segmentos $\overline{AB}$ e $\overline{DC}$. Caso a diferença entre θ_1 e θ_2 seja de 90° as retas são as ortogonais.

Após a identificação dos pontos da matriz referentes à imagem, projeta-se o retângulo na tela do computador, e sobre ele é aplicado a transformada inversa da HT, ou seja, são projetadas as retas referentes à filtragem fina do arranjo acumulador, de acordo com a Fig.4.17(c).

4.4.2 - Cálculo do Centro de Massa

O Cálculo do centro de massa está baseado nos valores finais da filtragem do arranjo acumulador. Quando se projeta esses valores no plano imagem, têm-se uma melhor visualização desses cálculos. Na Fig.4.18 são mostradas duas imagens em diferentes posições e suas respectivas tabelas finais.

Para os cálculos foram determinadas quatro condições:

1. Condição : Se $|\theta_1|$ ou $|\theta_2| = 90^\circ \Rightarrow$ As coordenadas de centro de massa são:

$$Coord_x = \rho_4 - \frac{(\rho_4 - \rho_3)}{2} \quad e \quad Coord_y = \rho_1 - \frac{(\rho_1 - \rho_2)}{2}$$

conforme mostra a Fig.4.19(a).

2. Condição : Se $|\theta_1|$ ou $|\theta_2| = 45^\circ$. O valor da hipotenusa será dada por,

$$\Rightarrow R = \rho_4 - \frac{(\rho_4 - \rho_3)}{2}, \text{ conforme Fig.4.19(b).}$$

3. Condição : Se $|\theta_1|$ ou $|\theta_2| > 45^\circ \Rightarrow$ calcula-se as coordenadas de acordo com a Fig.4.18(b)

4. Condição: Se $|\theta_1|$ ou $|\theta_2| < 45^\circ \Rightarrow$ calcula-se as coordenadas de acordo com a Fig.4.18(a).

Observa-se que os valores de θ 's variam sempre dentro de um intervalo em relação ao eixo x positivo, onde θ_1 e θ_2 pertencem ao intervalo $[-90^\circ$ a $-1^\circ]$ e os valores de θ_3 e θ_4 ao intervalo $[0^\circ$ a $89^\circ]$. A Fig.4.18 possui dois casos típicos de cálculo de centro de massa, referentes às condições 3 e 4 definidas.

FIGURA 4.18 - Projeção no plano imagem, tabela final do arranjo acumulador e cálculo dos segmentos " $\overline{AB}$ " e " $\overline{BC}$ ". (a) Figura 23 onde $|\theta_1 \text{ ou } 2| < 45^\circ$ (ver anexo E), (b) Figura 33 $|\theta_1 \text{ ou } 2| > 45^\circ$ (ver anexo E),

FIGURA 4.19 - (a) Cálculo para imagens a 90° e a 0°, admitindo a 1ª condição; (b) Cálculo para imagens quando θ_1 ou θ_2 for igual a 45°.

No fluxograma, conforme Fig.4.20, é descrito o procedimento para o cálculo do centro de massa :

FIGURA 4.20 - Fluxograma do cálculo do Centro de Massa

Inicialmente, testa-se as quatro condições citadas anteriormente, sendo que para cada condição são realizadas rotinas diferentes.

1ª Condição: calcula-se diretamente as coordenadas de centro de massa “Coord_x” e “Coord_y”, conforme Fig.4.19 (a).

2ª Condição: calcula-se a resultante e em seguida as coordenadas. “Coord_x” e “Coord_y”, conforme Fig.4.19 (b).

3ª Condição e 4ª Condição: os segmentos “AB” e “BC” são calculados de acordo com o valor de $|\theta_1|$ e $|\theta_2|$. Se for maior que 45° , conforme a

Fig.4.18 (a), se for menor, segundo a Fig.4.18 (b). Em seguida, encontram-se os valores da hipotenusa e o seu arco tangente. O valor deste arco também é testado com 45° para o cálculo do ângulo da resultante, e se for maior procede-se de acordo com a Fig.4.21 (a), caso contrário, conforme a Fig.4.21 (b). Finalmente com o valor do ângulo da Resultante ($\hat{\text{Ângulo_R}}$), encontram-se as Coord_x e Coord_y , as quais serão as coordenadas do centro de massa do objeto em relação à imagem.

$$\text{Coord_x} = R \cos (\hat{\text{Ângulo_R}})$$

$$\text{Coord_y} = R \sin (\hat{\text{Ângulo_R}})$$

FIGURA 4.21 - Ampliação dos triângulos ABC da Fig.4.18(a) e (b) e cálculo dos ângulos das Resultantes, respectivamente.

4.5 - Conclusão

Neste capítulo foi apresentado a técnica da HT utilizada para inspecionar placas de madeiras, determinando características juntamente com o fluxograma do programa.

Através do arranjo acumulador determina-se os valores dos lados, do perímetro, da área, e as coordenadas do centro de massa do objeto em relação à imagem. Esses valores são obtidos por intermédio de uma filtragem grosseira-fina e de critérios de localização.

Depois de executar todas os parâmetros de extração da imagem, aplicação da HT e filtragem do arranjo, aplica-se a HT inversa. Observa-se que, as retas são traçadas exatamente sobre os limites da imagem. Isso mostra que aplicando a HT, e avaliando-se o arranjo acumulador pode-se obter bons resultados quando o objeto a ser inspecionado é previamente conhecido e apresenta peculiaridades, tais como ortogonalidade de seus lados.

CAPÍTULO 5 - RESULTADOS E CONCLUSÕES

No ambiente industrial é exigido que uma tarefa, dependendo da sua aplicação, seja executada de maneira mais rápida e precisa possível. Às vezes, a velocidade do sistema deve ser um pouco sacrificada a fim de se conseguir uma melhor precisão.

A finalidade desta dissertação é o emprego da técnica da HT na inspeção de placas de madeira. Essa técnica mostrou ser uma ferramenta muito robusta e bastante imune a ruídos. Com isso, aplicando um detector de bordas na imagem mesmo com a presença de ruídos ou falhas, é possível traçar as retas dessa imagem. Para abordar os erros de medida e a confiabilidade no método proposto, foram utilizadas 198 imagens da placa de madeira, sendo algumas rotacionadas e outras transladadas na vertical e na horizontal. Os resultados de tais imagens foram comparados com a aplicação de "Momentos" (HU,1961; 1962) para definição das mesmas variáveis como, área e o centro de massa do objeto na imagem.

5.1 - Análise da HT

A HT é uma técnica na qual não se necessita de todos os pontos da imagem para a sua localização. Pela Fig.5.1, observa-se que mesmo obtendo-se uma imagem faltando diversos pontos, localizam-se as retas que formam a mesma.

Para se obter uma melhor exatidão nos resultados extraídos do arranjo acumulador, é necessário que os intervalos de $\Delta\theta$ e $\Delta\rho$ sejam pequenos. Quanto menor a variação, melhor a filtragem final dos picos elevados na matriz. Assim, foram testados vários valores sendo que a melhor resolução, atingiu-se quando se usou valores de $\Delta\theta$ e $\Delta\rho$ iguais a 1.

Imagem	Após detector de Borda	Aplicação da HT
D17
F7
F11

FIGURA 5.1 -Imagens aplicando o detector de bordas e a HT (ver imagens nos Anexos D e F)

Em algumas filtragens do arranjo acumulador os valores de θ encontrados não são ortogonais entre si. Entretanto, são exatamente os valores mais significativos localizados durante as filtragens. Consequentemente, os resultados derivados como área, perímetro e centro de massa são afetados.

Na Fig.5.2, nas imagens F-32 e B-24 (ver anexos F e B), observa-se alguns desses erros.

FIGURA 5.2 - Imagens dos valores de θ (a) Anexo F-32; (b) Anexo B-1.

Na Fig.5.2 (a), os valores θ_3 e θ_4 da tabela do arranjo acumulador estão defasados em 3° graus. Observa-se tal desvio quando se projeta esses valores no gráfico. O mesmo ocorre na Fig.5.2 (b), numa situação mais crítica, onde todos os valores de θ rejeitam a hipótese de ortogonalidade. Entretanto, mesmo com tais desvios, consegue-se obter resultados aproximados dos valores reais, para área, perímetro e centro de massa.

5.2 - Método dos Momentos

O método dos Momentos é um equacionamento matemático que se baseia no estudo da mecânica dos corpos, inicialmente apresentada por Hu (1961,1962). As propriedades de invariância para Rotação, Escala e Translação podem ser derivadas usando as funções de Momentos. A transformada de Momentos de uma função imagem $f(x,y)$, é dada por:

$$m_{pq} = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} x^p y^q f(x,y) dx dy \quad (5.1)$$

sendo $p, q = 0, 1, 2, \dots, \infty$

Assim, a função imagem pode, através da representação inicial, ter uma infinidade de conjuntos de coeficientes, m_{pq} , os quais são derivados da projeção da imagem em um conjunto 2D de funções básicas polinomiais. Aplicações práticas da transformada requerem um número limitado de momentos. No caso de uma discretização espacial de uma imagem $M \times N$, denominada $f(i,j)$, a Eq. 5.1 pode ser formulada usando dois somatórios:

$$m_{pq} = \sum_{i=0}^M \sum_{j=0}^N i^p j^q f(i,j) \quad (5.2)$$

Assumindo para uma região R de uma imagem binarizada com uma região de intensidade igual a 1, os momentos $(p, q)^{th}$, são dados por

$$m_{pq} = \sum \sum_{(i,j) \in R} i^p j^q \quad (5.3)$$

Assim, todos os pontos (i,j) usados na computação tanto da borda quanto do interior, são pontos de R . Portanto, m_{00} é simplesmente o número total de pontos numa região, e as coordenadas do centro de massa são dadas por $\hat{i}$ e $\hat{j}$. Logo o Momento central pode ser expresso por :

$$\mu_{pq} = \sum_{i,j \in R} (i - \hat{i})^p (j - \hat{j})^q \quad (5.4)$$

onde

$$\hat{i} = (m_{10} / m_{00})$$

$$\hat{j} = (m_{01} / m_{00})$$

m_{10} = corresp. ao somatório das coordenadas em x da região binarizada R.

m_{01} = corresp. ao somatório das coordenadas em y da região binarizada R.

5.3 - Comparação do Método HT e Momentos com Enfoque Estatístico

O comportamento das imagens inspecionadas pelo método da HT mostrou-se constante. Entretanto, para que estes resultados tenham credibilidade quanto ao seu desempenho, optou-se por uma comparação estatística com outro método. Devido a inúmeros métodos de reconhecimento de padrões (CHIN,1986; NORTON-WAYNE,1991; PUTRÉ,1992), a escolha do método de Momentos foi estabelecida por ser uma técnica simples e possuir resultados imediatos analisando os critérios de velocidade de execução (BOIKO et.al.,1989), facilidade de implementação e principalmente pelas suas características de invariâncias quanto à rotação e translação (FLUSSER,1993; LIN et. al.,1994).

Para uma análise estatística de áreas e centros de massa entre os dois métodos, foram coletadas imagens da placa de madeira pela câmera e, classificadas em três tipos de amostras considerando-se cada tipo a existência ou não de ruído significativo.

⇒ rotacionadas $\begin{cases} 37 & \text{imagens com ruído} \\ 37 & \text{imagens sem ruído} \end{cases}$

⇒ transladadas em X $\begin{cases} 31 & \text{imagens com ruído} \\ 31 & \text{imagens sem ruído} \end{cases}$

⇒ transladadas em Y $\begin{cases} 31 & \text{imagens com ruído} \\ 31 & \text{imagens sem ruído} \end{cases}$

5.3.1 - Teste t de Student Pareado e Não Pareado.

Devido à necessidade de comparação entre os resultados obtidos pelo método da HT e pelo método dos Momentos, aplica-se o Teste t de Student Pareado (PARADINE & RIVETT, 1974; ACHCAR, 1995), para verificar se existe ou não diferença entre as amostras, de área e centro de massa dos respectivos métodos. Assim, para os conjuntos amostrais serão testadas as hipóteses H_0 contra as hipóteses H_1 :

$$\begin{cases} H_0: \delta_0 = 0 & \text{logo têm-se} & \mu_{HT} = \mu_{MM} \\ H_1: \delta_0 \neq 0 & \text{logo têm-se} & \mu_{HT} \neq \mu_{MM} \end{cases}$$

onde μ_{HT} = média populacional do Método da HT

e μ_{MM} = média populacional do Método de Momentos

Caso a hipótese H_0 seja não rejeitada, admite-se não haver diferença entre os métodos HT e Momentos. Caso contrário, admite-se haver diferença entre esses métodos.

Independente da hipótese H_0 do Teste t de Student Pareado ser ou não rejeitada, aplica-se em seguida o Teste t de Student (não pareado), onde novamente as mesmas hipóteses H_0 e H_1 serão testadas. Compara-se desta vez, as amostras HT e/ou Momentos (MM) com os valores reais de área e centro de massa da imagem inspecionada.

Neste trabalho foi adotado um grau de confiabilidade de 95%, ou seja, para cada hipótese, tem-se uma confiança de que 95% dos seus valores estejam dentro de um certo intervalo definido, conforme Fig.5.3. Onde $(1 - \alpha)$ delimita a região onde os diversos valores amostrais de área e centro de massa são aceitos como iguais à média populacional μ . A partir de $|\alpha/2|$ os valores não mais são aceitos como iguais a μ , indicando os 5% restantes de probabilidade:

FIGURA 5.3 - Gráfico de uma distribuição normal com intervalo de confiança de 95%

Aplicou-se esses testes separadamente às amostras de imagens rotacionadas, transladadas na vertical e transladadas na horizontal, para os métodos da HT e Momentos.

Segue abaixo um roteiro simplificado dessa aplicação:

Suponha que x_{1j} indique o valor de uma das amostras do método da HT e x_{2j} o mesmo para o método dos Momentos. Assim tem-se (x_{1j}, x_{2j}) como os pares amostrais de onde se tira o vetor de diferença (D_j)

$$D_j = x_{1j} - x_{2j} \quad \text{onde } j = 1, 2, 3, \dots, n \quad (5.5)$$

onde n = quantidade de elementos das amostras.

Assume-se então que, as diferenças D_j representam n observações vindas de uma distribuição normal com média $\bar{D}$, variância σ_d^2 e desvio padrão σ , onde:

$$\overline{D} = \frac{\sum D_j}{n} \quad ; \quad \sigma_d^2 = \frac{\sum_{j=1}^n (D_j - \overline{D})^2}{n-1} \quad ; \quad \sigma = \sqrt{\sigma_d^2} \quad (5.6)$$

sendo $n - 1$, os graus de liberdade da distribuição t Student.

Considerando o exposto acima o valor do Teste t de Student Pareado (t_1^*) é dado por:

$$t_1^* = \frac{\overline{D}}{\sigma / n} \quad (5.7)$$

O Teste t de Student não pareado (t_2^*) é definido como:

$$t_2^* = \frac{\overline{x} - \mu_{\text{Real}}}{\sqrt{\frac{\sum (x_j - \overline{x})^2}{n-1}}} \quad (5.9)$$

onde $\overline{x}$ = média dos valores amostrais da HT ou MM

x_j = valores de uma das amostras do método HT ou MM

Pela Tabela do Teste t de Student (Anexo J), obtém-se o valor do t_{tab} correspondente a $(n - 1)$ graus de liberdade e intervalo de 95% de confiança.

Assim, testa-se os valores de t_1^* e t_2^* :

● Se $|t_1^*| \leq t_{\text{tab}} \Rightarrow$ conclui-se que $\mu_{\text{HT}} = \mu_{\text{MM}}$, não rejeitando H_0 . Dessa forma calcula-se t_2^* apenas com uma das amostras, logo :

$\Rightarrow$ Se $|t_2^*| \leq t_{\text{tab}}$, não rejeita H_0 , considera $\mu_{\text{HT}} = \mu_{\text{MM}} = \mu_{\text{Real}}$, assim, não existe diferença entre os métodos com o valor real.

$\Rightarrow$ Se $|t_2^*| > t_{\text{tab}}$, rejeita H_0 , logo $(\mu_{\text{HT}} = \mu_{\text{MM}}) \neq \mu_{\text{Real}}$. Observa-se que neste caso, nenhum dos métodos fornece resultados que possa ser considerado igual ao valor real.

● Se $|t_1^*| > t_{lab}$, conclui-se que $\mu_{HT} \neq \mu_{MM}$, rejeitando H_0 , onde os valores encontrados pelos métodos não pertencem ao intervalo de 95% de confiança. Dessa forma calcula-se t_2^* tanto para o método da HT quanto para o método dos Momentos. Verificando-se qual dos dois métodos não rejeita a hipótese H_0 e obviamente qual se aproxima do valor real.

5.3.2 - Modelo da Placa Inspecionada

A placa de madeira inspecionada neste trabalho, possui as seguintes dimensões reais:

FIGURA 5.4 - Dimensões da Placa de Madeira Inspecionada

logo, sua área é $133,92\text{cm}^2$. Para transformar os valores da área em “pixels”, digitaliza-se uma imagem da placa de madeira e em seguida a binariza. Assim, o somatório dos pontos do interior da placa corresponderá a área em “pixels”:

$$\text{Área} = 3467 \text{ “pixels”}$$

As imagens utilizadas nos cálculos possuem todas o tamanho de 128×128 “pixels”, correspondendo a uma imagem real com as dimensões de 26×26 cm. Dessa forma calcula-se os valores dos lados e do perímetro:

lado1 = 34,46 “pixels”

lado2 = 92,55 “pixels”

Perímetro = 254,02 “pixels”

Para o cálculo das áreas pela HT aplica-se o seguinte critério :

$$\text{Área} = (\text{lado1} * \text{lado2}) + \text{perímetro}$$

A soma do perímetro a área refere-se aos pontos pertencentes a borda que não foram incluídos. Os valores das áreas encontradas pelo método da HT, são as mesmas tanto para imagens com ou sem ruídos. Observa-se que o ruído não interferiu no resultado final.

5.3.3 - Análise Estatística

5.3.3.1 - Imagens Rotacionadas

TABELA 5.1 - Valores estatísticos relacionados com as imagens Rotacionadas

Tabela Estatística - Rotação					
		HT s/ ruído	MM s/ ruído	HT c/ ruído	MM c/ ruído
Média		3468.67	3466.67	3468.67	4592.02
T.t. Pareado	t_1^*	1.659		-34.103	
T.t.	t_2^*	0.563		0.563	5.595
t tabela.	t_{lab}	2.028			

De acordo com os dados estatísticos da Tabela 5.1, o Teste t de Student Pareado aplicado aos métodos HT e Momentos s/ ruído, forneceu os seguintes valores :

$$P \{ |t_1^*| \leq t_{lab} \}$$

$$H_0 : \delta_0 = 0 \quad \text{logo} \quad \mu_{HT} = \mu_{MM} \quad \therefore \text{hipótese não rejeitada}$$

logo se deduz que a diferença não é significativa entre as médias, as quais podem ser consideradas iguais. Dessa forma aplica-se o Teste t de Student não Pareado apenas para uma amostra, já que a hipótese H_0 não foi rejeitada, portanto :

$$P \{ |t_2^*| \leq t_{\text{tab}} \}$$

$$H_0 : \delta_0 = 0 \quad \text{logo} \quad \mu_{\text{HT}} = \mu_{\text{MM}} = \mu_{\text{Real}}$$

$\therefore$ nova hipótese também não rejeitada

Visto que o valor do Teste t de Student (Anexo J) está dentro do intervalo de confiança considerado, tanto o método da HT quanto o dos Momentos, pode ser aplicado a imagens rotacionadas sem ruído, pois estatisticamente seus valores foram considerados constantes e iguais aos valores reais.

Para as imagens rotacionadas com ruído, conforme Tabela 5.1, aplicando o Teste t de Student Pareado, observa-se:

$$H_0 : \delta_0 = 0$$

$$P \{ |t_1^*| > t_{\text{tab}} \} \quad \therefore \quad \text{hipótese rejeitada, o valor de } -34.103 \text{ não pertence ao intervalo de confiança}$$

Assim, aplica-se o Teste t de Student (não pareado) para as duas amostras e observa-se que :

a) Para a HT:

$$P \{ |t_2^*| \leq t_{\text{tab}} \}$$

$$H_0 : \delta_0 = 0 \quad \text{logo,} \quad \mu_{\text{HT}} = \mu_{\text{Real}} \quad \therefore \quad \text{hipótese não rejeitada}$$

b) Para Momentos

$$P \{ |t_2^*| > t_{\text{tab}} \}$$

$$H_0 : \delta_0 = 0 \quad \text{então} \quad \mu_{\text{MM}} \neq \mu_{\text{Real}}$$

- hipótese rejeitada, pois 5.108 está fora do intervalo de confiança

Os valores obtidos para as imagens rotacionadas com ruído utilizando o método de Momentos, estatisticamente ficaram fora dos limites do intervalo de confiança. Já o método da HT, mostrou-se mais estável, visto ter uma média dentro do intervalo de 95% de confiança.

FIGURA 5.5 - Gráfico das 37 imagens sem ruído da placa de madeira rotacionadas, usando o método da HT e dos Momentos, para o cálculo da área.

Na Fig.5.5, observa-se que tanto os valores de área pelo método da HT como por Momentos, seus valores estão próximos da área real.

FIGURA 5.6 - Gráfico das 37 imagens com ruído da placa de madeira rotacionadas, usando o método da HT e dos Momentos, para o cálculo da área.

De acordo com o gráfico da Fig.5.6, o método dos Momentos com ruído teve uma variação muito grande em relação ao método da HT, confirmado pelo método estatístico do Teste t de Student.

5.3.3.2 - Imagens Transladadas na Horizontal

TABELA 5.2 - Valores estatísticos relacionados com as imagens Transladadas na horizontal

Tabela Estatística - Transladada na Horizontal					
		HT s/ ruído	MM s/ ruído	HT c/ ruído	MM c/ ruído
Média		3469.58	3465.54	3469.58	5395.25
T.t.Pareado	t_1^*	1.826		-11.615	
T.t.	t_2^*	0.854		0.854	3.088
t tabela.	t_{tab}	2.042			

Analisando a Tabela 5.2, observa-se que as imagens transladadas na horizontal, tiveram comportamento semelhante às imagens rotacionadas. As hipóteses para o Teste t de Student Pareado e não pareado foram:

a)- Imagens sem ruído

1a - Teste t de Student Pareado

$$P \{ |t_1^*| \leq t_{lab} \}$$

$$H_0 : \delta_o = 0 \quad \text{então} \quad \mu_{HT} = \mu_{MM} \quad \therefore \quad \text{hipótese não rejeitada}$$

2a - Teste t de Student (não pareado)

$$P \{ |t_2^*| \leq t_{lab} \}$$

$$H_0 : \delta_o = 0 \quad \text{então} \quad \mu_{HT} = \mu_{MM} = \mu_{Real} \quad \therefore \quad \text{hipótese não rejeitada}$$

$\Rightarrow$ Logo, como os valores estão dentro do intervalo de confiança, qualquer um dos métodos pode ser considerado.

b) Imagens com ruído

1b - Teste t de Student Pareado

$$P \{ |t_1^*| > t_{lab} \}$$

$$H_0 : \delta_o = 0 \quad \text{então} \quad \mu_{HT} \neq \mu_{MM}$$

- hipótese rejeitada, o valor de -11.615 não pertence ao intervalo de confiança

2b - Teste t de Student (não pareado)

$\Rightarrow$ Para a HT:

$$P \{ |t_2^*| \leq t_{lab} \}$$

$$H_0 : \delta_o = 0 \quad \text{então} \quad \mu_{HT} = \mu_{Real}$$

hipótese não rejeitada

⇒ Para Momentos

$$P \{ |t_2^*| > t_{\text{tab}} \}$$

$$H_0 : \delta_0 = 0 \quad \text{então} \quad \mu_{\text{MM}} \neq \mu_{\text{Real}}$$

- hipótese rejeitada, pois 3.088 está fora do intervalo de confiança

O resultado mostra que os valores de áreas calculados pelo método de Momentos foram rejeitados, sendo considerado apenas os da HT.

FIGURA 5.7 - Gráfico das 31 imagens sem ruído da placa de madeira transladada na horizontal, usando o método da HT e dos Momentos, para o cálculo da área.

Novamente como ocorre com as imagens rotacionadas sem ruído, as imagens transladadas na horizontal mantiveram uma média oscilando em torno do valor da área real. Assim tanto a HT quanto Momentos, foram considerados estar dentro do intervalo de confiança adotado.

FIGURA 5.8 - Gráfico das 31 imagens com ruído da placa de madeira transladada na horizontal , usando o método da HT e dos Momentos, para o cálculo da área.

Nas imagens transladadas na horizontal ocorreu um ruído crescente aplicando o método dos Momentos, o qual pode ser observado no gráfico da Fig.5.8. Esse ruído foi aumentando gradativamente durante o deslocamento da placa de madeira na imagem. O método da HT mostrou-se novamente mais robusto a ruídos.

5.3.3.3 - Imagens Transladadas na Vertical

TABELA 5.3 - Valores estatísticos relacionados com as imagens Transladadas na vertical

Tabela Estatística - Transladada na Vertical					
		HT s/ ruído	MM s/ ruído	HT c/ ruído	MM c/ ruído
Média		3465.16	3459.48	3465.16	5395.25
T.t.Pareado	t_1^*	1.919		-49.758	
T.t.	t_2^*	-0.254		0.254	8.902
t tabela	t_{lab}	2.042			

Observando-se a Tabela 5.3, verifica-se que as imagens transladadas na vertical, tem comportamento semelhante aos dados da Tabela 5.2 e 5.1 das imagens rotacionadas e transladadas na horizontal, respectivamente. Os testes de hipóteses para o Teste t Pareado e Teste t não pareado foram:

a)- Imagens sem ruído

1a - Teste t de Student Pareado

$$P \{ |t_1^*| \leq t_{tab} \}$$

$$H_0 : \delta_0 = 0 \quad \text{então} \quad \mu_{HT} = \mu_{MM} \quad \therefore \quad \text{hipótese não rejeitada}$$

2a - Teste t de Student (não pareado)

$$P \{ |t_1^*| \leq t_{tab} \}$$

$$H_0 : \delta_0 = 0 \quad \text{então} \quad \mu_{HT} = \mu_{MM} = \mu_{Real} \quad \therefore \quad \text{hipótese não rejeitada}$$

logo como os valores estão dentro do intervalo de confiança, qualquer um dos métodos pode ser considerado.

b) Imagens com ruído

1b - Teste t de Student Pareado

$$P \{ |t_1^*| > t_{tab} \}$$

$$H_0 : \delta_0 = 0 \quad \text{então} \quad H_1 : \mu_{HT} \neq \mu_{MM}$$

- hipótese rejeitada , não pertence ao intervalo de confiança. Como existe diferença entre o método HT e Momentos, aplica-se o Teste t de Student (não pareado) para os dois métodos.

2b - Teste t de Student (não pareado)

⇒Para a HT:

$$P \{ |t_2^*| \leq t_{tab} \}$$

$$H_0 : \delta_0 = 0 \quad \text{então} \quad H_0 : \mu_{HT} = \mu_{Real} \quad \therefore \quad \text{hipótese não rejeitada}$$

⇒ Para Momento

$$P \{ |t_2^*| > t_{\text{tab}} \}$$

$$H_0 : \delta_0 = 0 \quad \text{então} \quad : \quad \mu_{\text{MM}} \neq \mu_{\text{Real}}$$

- hipótese rejeitada pois resultados estão fora do intervalo de confiança.

O resultado mostra que os valores de áreas calculados pelo método dos Momentos foram rejeitados, sendo considerado apenas os da HT.

FIGURA 5.9 - Gráfico das 31 imagens sem ruído da placa de madeira transladada na vertical, usando o método da HT e dos Momentos, para o cálculo da área.

Pelo gráfico da Fig.5.9, os valores das áreas variaram mais em relação às imagens transladadas na horizontal, contudo permaneceram dentro do intervalo de confiança, confirmado assim as hipóteses do Teste t de Student Pareado e não pareado.

FIGURA 5.10 - Gráfico das 31 imagens com ruído da placa de madeira transladada na vertical , usando o método da HT e dos Momentos, para o cálculo da área.

Novamente, conforme o gráfico da Fig.5.10, as imagens na presença de ruídos quando transladadas na vertical, apresentaram o mesmo problema de distorção no resultado da área. Logo, a HT mostrou ser mais constante na presença de ruídos.

5.3.3.4 - Análise das Coordenadas de Centro de Massa

Em relação às coordenadas de centro de massa do objeto, tanto para imagens rotacionadas ou transladadas na horizontal e na vertical, são adotados os mesmos critérios de avaliação feito para as áreas.

Analisando apenas as coordenadas de centro de massa das imagens rotacionadas tem-se:

$$\text{Coordenada Real } \begin{cases} x = 67 \text{ pixels} \\ y = 63 \text{ pixels} \end{cases}$$

1) - Imagens sem ruído

TABELA 5.4 - Valores estatísticos relacionados com as coordenadas das imagens Rotacionadas sem ruído

Tabela Estatística - Coordenadas Rotacionadas					
		HT	MM	HT	MM
Coordenadas		x		y	
Média		67.291	67.218	62.472	62.567
T.t.Pareado	t_1^*	1.065		-1.106	
T.t.	t_2^*	0.851		-0.861	
t tabela.	t_{lab}	2.028			

- Teste t de Student Pareado :

a) para coordenada x da HT com MM

$$P \{ |t_1^*| \leq t_{lab} \}$$

$$H_0 : \delta_0 = 0 \quad \text{então:} \quad \mu_{X-HT} = \mu_{X-MM} \quad \therefore \quad \text{hipótese não rejeitada}$$

b) para coordenada y da HT com MM

$$P \{ |t_1^*| \leq t_{lab} \}$$

$$H_0 : \delta_0 = 0 \quad \text{então:} \quad \mu_{Y-HT} = \mu_{Y-MM} \quad \therefore \quad \text{hipótese não rejeitada}$$

Como a hipótese H_0 foi não rejeitada, pode-se considerar apenas uma das médias para o cálculo do Teste t. de Student (não pareado)

- Teste t de Student (não pareado) :

a) para coordenada x

$$P \{ |t_2^*| \leq t_{lab} \}$$

$$H_0 : \delta_0 = 0 \quad \text{então:} \quad \mu_{X-HT} = \mu_{X-MM} = \mu_{\text{Real} - X}$$

$$\therefore \quad \text{hipótese não rejeitada}$$

b) para coordenada y

$$P \{ |t_2^*| \leq t_{\text{tab}} \}$$

$$H_0 : \delta_0 = 0 \quad \text{então :} \quad \mu_{Y-HT} = \mu_{Y-MM} = \mu_{\text{Real}} - Y$$

$\therefore$ hipótese não rejeitada

logo como os valores estão dentro do intervalo de confiança, qualquer um dos métodos pode ser considerado para o cálculo das coordenadas de centro de massa.

2) - Imagens com ruído

TABELA 5.5 - Valores estatísticos relacionados com as coordenadas das imagens Rotacionadas com ruído

Tabela Estatística - Coordenadas Rotacionadas c/ ruído					
		HT	MM	HT	MM
		x		y	
Média		67.291	73.162	62.472	59.162
T.t.Pareado	t_1^*	-23.955		7.670	
T.t.	t_2^*	0.851	4.326	-0.861	-4.288
t tabela.	t_{tab}	2.028			

a)- Teste t de Student Pareado para coordenada x da HT com MM

$$P \{ |t_1^*| > t_{\text{tab}} \}$$

$$H_0 : \delta_0 = 0 \quad \text{então:} \quad \mu_{X-HT} \neq \mu_{X-MM} \quad \therefore \quad \text{hipótese rejeitada}$$

b) - Teste t de Student Pareado para coordenada y da HT com MM

$$P \{ |t_1^*| > t_{\text{tab}} \}$$

$$H_0 : \delta_0 = 0 \quad \text{então:} \quad \mu_{Y-HT} \neq \mu_{Y-MM} \quad \therefore \quad \text{hipótese rejeitada}$$

Devido ao fato das médias das coordenadas x e y serem diferentes conclui-se que estão fora do intervalo de confiança. Assim, calcula-se o Teste t de Student (não pareado) para verificar quais dos dois métodos se aproximam do valor real. Dessa forma tem-se :

a) Teste para coordenada x da HT em relação ao valor real

$$P \{ |t_2^*| \leq t_{\text{tab}} \}$$

$H_0 : \delta_0 = 0$ então: $\mu_{X-HT} = \mu_{X-Real} \therefore$ hipótese não rejeitada

b) Teste para coordenada x do MM em relação ao valor real

$$P \{ |t_2^*| > t_{\text{tab}} \}$$

$H_0 : \delta_0 = 0$ então: $\mu_{X-MM} \neq \mu_{X-Real} \therefore$ hipótese rejeitada

c) Teste para coordenada y da HT em relação ao valor real

$$P \{ |t_2^*| \leq t_{\text{tab}} \}$$

$H_0 : \delta_0 = 0$ então $\mu_{Y-HT} = \mu_{Y-Real} \therefore$ hipótese não rejeitada

d) Teste para coordenada y do MM em relação ao valor real

$$P \{ |t_2^*| > t_{\text{tab}} \}$$

$H_0 : \delta_0 = 0$ então: $\mu_{Y-MM} \neq \mu_{Y-Real} \therefore$ hipótese rejeitada

Observa-se que os valores obtidos confirmam os resultados estatísticos realizados com as áreas das imagens rotacionadas. As coordenadas x e y pelo Método da HT mantiveram-se constantes na presença de ruído, o que não ocorreu com o Método dos Momentos.

Os gráficos referentes às coordenadas das imagens transladadas na horizontal e vertical estão nos Anexos M e N.

FIGURA 5.11 - Gráfico das 31 imagens com e sem ruído da placa de madeira, usando o método da HT e dos Momentos, para o cálculo das coordenadas do centro de massa.

As imagens analisadas neste trabalho, mostram que, para se aplicar Momentos é necessário um ambiente sem interferências externas, sem ruídos, o que não acontece na realidade. Em um ambiente industrial, as peças inspecionadas estão sujeitas a ruídos e falhas podem prejudicar o resultado final, sendo necessário assim, a implementação de filtros, os quais, para a operação em tempo real introduz um excesso de computação.

5.3.3.5 - Análise do Cálculo de Perímetro pela HT com o Valor Real

A HT possui duas características a mais em relação ao método de Momentos, o cálculo dos lados e do perímetro. No Anexo L estão os valores dos lados e perímetros das imagens inspecionadas.

FIGURA 5.10 - Gráfico das imagens rotacionadas, transladadas na vertical e horizontal da placa de madeira, comparando o método da HT com o valor real, para o cálculo dos perímetros.

Aplicando o Teste t de Student (Pareado e não pareado) para as imagens rotacionadas, transladadas na vertical e na horizontal com relação ao valor real tem-se:

resultados, as imagens podem estar bastante ruidosas, conforme as imagens utilizadas neste trabalho. Para se usar Momentos, há a necessidade de uma iluminação adequada, pois qualquer falha na aquisição da imagem, prejudica o seu desempenho.

A HT mostrou-se eficaz para a análise de uma placa simples previamente conhecida, formada por quatro retas perpendiculares. Entretanto para imagens mais complexas utilizar a HT pode inviabilizar um projeto em tempo real. Os trabalhos realizados por outros autores utilizando a HT (LI et al.,1986; ILLINGWORT & KITTLER, 1987; BEN-TZVI & SANDLER,1990; HOU et al.,1993; HAN et al.,1994) tem demonstrado a utilização de novas técnicas para melhorar o desempenho da HT, com isso surge o problema do tempo computacional gasto para realizar pequenas tarefas.

5.5 - Sugestões para Trabalhos Futuros

- Adaptar o sistema para inspecionar outros objetos em uma esteira rolante;
- Analisar outras formas de objetos;
- Implementação da HT em hardware, fornecendo ao sistema os dados do arranjo acumulador.

ANEXOS

ANEXOS A- Imagens transladadas na horizontal sem ruído sendo aplicando Momentos, Detector de bordas e HT,as quais são utilizadas nas conclusões. Total de 31 imagens analisadas.

Figuras	Imagem	Aplicação do Momentos	Detector de Borda	Aplicando a HT
1
2
3
4
5
6

Figuras	Imagem	Aplicação do Momentos	Detector de Borda	Aplicando a HT
7
8
9
10
11
12

Figuras	Imagem	Aplicação do Momentos	Detector de Borda	Aplicando a HT
13
14
15
16
17
18

Figuras	Imagem	Aplicação do Momentos	Detector de Borda	Aplicando a HT
19
20
21
22
23
24

Figuras	Imagem	Aplicação do Momentos	Detector de Borda	Aplicando a HT
25
26
27
28
29
30

Figuras	Imagem	Aplicação do Momentos	Detector de Borda	Aplicando a HT
31

ANEXOS B- Imagens transladadas na horizontal e com ruído, sendo aplicado Momentos, Detector de bordas e HT. Total de 31 imagens analisadas

Figuras	Imagem	Aplicação do Momentos	Detector de Borda	Aplicando a HT
1
2
3
4
5
6

Figuras	Imagem	Aplicação do Momentos	Detector de Borda	Aplicando a HT
7
8
9
10
11
12

Figuras	Imagem	Aplicação do Momentos	Detector de Borda	Aplicando a HT
13
14
15
16
17
18

Figuras	Imagem	Aplicação do Momentos	Detector de Borda	Aplicando a HT
19
20
21
22
23
24

Figuras	Imagem	Aplicação do Momentos	Detector de Borda	Aplicando a HT
25
26
27
28
29
30

Figuras	Imagem	Aplicação do Momentos	Detector de Borda	Aplicando a HT
31

ANEXOS C- Imagens transladadas na vertical sem ruído, sendo aplicando Momentos, Detector de bordas e HT. Total de 31 imagens analisadas

Figuras	Imagem	Aplicação do Momentos	Detector de Borda	Aplicando a HT
1
2
3
4
5
6

Figuras	Imagem	Aplicação do Momentos	Detector de Borda	Aplicando a HT
7
8
9
10
11
12

Figuras	Imagem	Aplicação do Momentos	Detector de Borda	Aplicando a HT
13
14
15
16
17
18

Figuras	Imagem	Aplicação do Momentos	Detector de Borda	Aplicando a HT
19
20
21
22
23
24

Figuras	Imagem	Aplicação do Momentos	Detector de Borda	Aplicando a HT
25
26
27
28
29
30

Figuras	Imagem	Aplicação do Momentos	Detector de Borda	Aplicando a HT
31

ANEXOS D- Imagens transladadas na vertical e com ruído, sendo aplicado Momentos, Detector de bordas e HT. Total de 31 imagens analisadas

Figuras	Imagem	Aplicação do Momentos	Detector de Borda	Aplicando a HT
1
2
3
4
5
6

Figuras	Imagem	Aplicação do Momentos	Detector de Borda	Aplicando a HT
7
8
9
10
11
12

Figuras	Imagem	Aplicação do Momentos	Detector de Borda	Aplicando a HT
13
14
15
16
17
18

Figuras	Imagem	Aplicação do Momentos	Detector de Borda	Aplicando a HT
19
20
21
22
23
24

Figuras	Imagem	Aplicação do Momentos	Detector de Borda	Aplicando a HT
25
26
27
28
29
30

Figuras	Imagem	Aplicação do Momentos	Detector de Borda	Aplicando a HT
31

ANEXOS E- Imagens rotacionadas e sem ruído, sendo aplicado Momentos, Detector de bordas e HT. Total de 37 imagens analisadas

Figuras	Imagem	Aplicação do Momentos	Detector de Borda	Aplicando a HT
1
2
3
4
5
6

Figuras	Imagem	Aplicação do Momentos	Detector de Borda	Aplicando a HT
7
8
9
10
11
12

Figuras	Imagem	Aplicação do Momentos	Detector de Borda	Aplicando a HT
13
14
15
16
17
18

Figuras	Imagem	Aplicação do Momentos	Detector de Borda	Aplicando a HT
19
20
21
22
23
24

Figuras	Imagem	Aplicação do Momentos	Detector de Borda	Aplicando a HT
25
26
27
28
29
30

Figuras	Imagem	Aplicação do Momentos	Detector de Borda	Aplicando a HT
31
32
33
34
35
36

Figuras	Imagem	Aplicação do Momentos	Detector de Borda	Aplicando a HT
37

ANEXOS F- Imagens rotacionadas e com ruído, sendo aplicado Momentos, Detector de bordas e HT. Total de 31 imagens analisadas

Figuras	Imagem	Aplicação do Momentos	Detector de Borda	Aplicando a HT
1
2
3
4
5
6

Figuras	Imagem	Aplicação do Momentos	Detector de Borda	Aplicando a HT
7
8
9
10
11
12

Figuras	Imagem	Aplicação do Momentos	Detector de Borda	Aplicando a HT
13
14
15
16
17
18

Figuras	Imagem	Aplicação do Momentos	Detector de Borda	Aplicando a HT
19
20
21
22
23
24

Figuras	Imagem	Aplicação do Momentos	Detector de Borda	Aplicando a HT
25
26
27
28
29
30

Figuras	Imagem	Aplicação do Momentos	Detector de Borda	Aplicando a HT
31
32
33
34
35
36

Figuras	Imagem	Aplicação do Momentos	Detector de Borda	Aplicando a HT
37

Anexo G - Tabela das imagens transladadas na horizontal

Tabela de Imagens Transladadas na Horizontal

IMAGEM REAL Área = 3467		HT c/ e s/ ruído			MM s/ ruído			MM c/ ruído		
X	Y	Área	X	Y	Área	X	Y	Área	X	Y
32.08	63	3472	30.1	63.5	3455	30	62	3785	18	68
35.52	63	3468	32.08	63.72	3478	32	63	4000	21	68
37.72	63	3465	35.52	62.05	3457	35	62	3953	24	69
39.65	63	3468	37.72	62.47	3455	37	62	4116	27	69
41.84	63	3472	39.65	62.55	3455	39	62	4182	30	69
44.36	63	3467	41.84	62.61	3450	42	62	4275	32	69
46.93	63	3465	44.36	62.3	3475	43	62	4433	33	68
49.12	63	3471	46.93	62.03	3482	46	62	4481	36	65
51.6	63	3471	49.12	63.62	3449	48	62	4603	38	69
53.84	63	3470	51.6	62.25	3472	51	62	4800	41	68
55.72	63	3472	53.84	63.06	3463	52	62	4854	42	68
58.6	63	3472	55.72	62.39	3466	55	62	5124	45	69
59.49	63	3470	58.6	63.21	3477	57	62	5140	47	68
62.37	63	3471	59.49	63.49	3476	59	62	5471	49	68
64.57	63	3472	62.37	62.53	3470	62	62	5462	50	67
67.07	63	3468	64.57	62.16	3473	63	62	5466	51	67
68.69	63	3465	67.07	62.83	3467	66	62	5623	53	66
71.79	63	3472	68.69	62.37	3468	68	62	5790	55	67
73.71	63	3471	71.79	62.75	3466	70	62	5787	57	67
74.72	63	3467	73.71	62.31	3458	72	62	6004	57	68
78.94	63	3472	74.72	63.17	3454	74	62	5932	59	68
80.09	63	3471	78.94	63.6	3471	77	62	6127	61	68
83.07	63	3465	80.09	62.72	3495	79	62	6171	63	67
85.37	63	3476	83.07	62.8	3442	81	62	6174	65	68
86.35	63	3468	85.37	63.11	3473	83	62	6297	67	70
88.96	63	3463	86.35	62.77	3450	85	63	6378	69	68
91.3	63	3471	88.96	62.56	3476	87	62	6469	67	70
93.91	63	3467	91.3	63.03	3458	89	62	6535	69	69
96.11	63	3471	93.91	63.81	3463	92	62	6610	71	68
99.85	63	3474	96.11	63.29	3474	95	62	6631	73	67
101	63	3470	99.85	63.21	3464	97	62	6580	77	66

Anexo H - Tabela das imagens Rotacionadas

Tabela de Imagens na Rotação

IMAGEM REAL Área = 3467		HT c/ e s/ ruído			MM s/ ruído			MM c/ ruído		
X	Y	Área	X	Y	Área	X	Y	Área	X	Y
67	63	3468	67.34	62.2	3469	67	63	4564	73	56
67	63	3470	66.15	62.34	3467	67	63	4793	74	59
67	63	3470	66.89	62.37	3469	67	63	4796	74	60
67	63	3471	66.95	62.32	3469	67	63	4862	75	61
67	63	3470	67.6	62.45	3470	67	63	4864	75	61
67	63	3472	67.35	62.66	3477	67	63	4641	74	60
67	63	3472	66.96	62.45	3476	67	63	4640	73	59
67	63	3470	67.2	62.4	3478	67	63	4642	74	60
67	63	3471	67.2	62.47	3478	68	63	4642	74	60
67	63	3468	67.23	62.38	3470	67	63	4479	73	61
67	63	3471	67.46	62.45	3463	67	62	4456	70	60
67	63	3468	67.45	62.38	3471	68	62	4301	69	61
67	63	3470	67.03	62.68	3470	67	62	4479	70	69
67	63	3462	67.35	62.5	3471	67	62	4301	70	58
67	63	3467	67.3	62.45	3470	67	62	4300	73	60
67	63	3469	67.45	62.58	3465	67	62	4293	75	60
67	63	3471	66.88	62.44	3453	67	62	4278	73	60
67	63	3473	67.35	62.45	3451	68	62	4441	73	61
67	63	3465	67.44	62.38	3470	67	62	4237	73	63
67	63	3466	68.1	62.44	3463	68	62	4253	73	63
67	63	3470	67.06	62.65	3456	67	62	4421	74	60
67	63	3463	67.35	62.25	3466	67	62	4434	74	60
67	63	3462	66.85	62.55	3456	67	62	4599	73	59
67	63	3468	67.4	62.48	3453	67	62	4609	74	60
67	63	3470	66.84	62.56	3460	67	62	4618	75	58
67	63	3472	67.45	62.45	3463	67	62	4453	74	57
67	63	3466	67.69	62.35	3462	67	63	4621	73	57
67	63	3470	67.5	62.47	3472	67	63	4800	74	58
67	63	3471	67.56	62.58	3471	67	63	4809	74	58
67	63	3468	67.37	62.45	3472	67	63	4800	74	57
67	63	3472	67.45	62.55	3474	68	63	4803	73	56
67	63	3470	67.55	62.35	3467	68	63	4793	73	56
67	63	3464	67.8	62.45	3463	68	63	4788	74	57
67	63	3463	67.36	62.58	3457	68	63	4779	73	57
67	63	3470	66.88	62.68	3479	67	63	4782	72	56
67	63	3468	67.39	62.78	3461	67	63	4785	72	55
67	63	3470	67.6	62.5	3465	67	63	4749	73	56

Anexo I - Tabela das imagens transladadas na vertical

Tabela de Imagens Transladadas na Vertical

IMAGEM REA Área = 3467		HT c/ e s/ ruído			MM s/ ruído			MM c/ ruído		
X	Y	Área	X	Y	Área	X	Y	Área	X	Y
82	20.089686	3460	81.45	21.1	3441	81	22	4757	60	31
82	22.959641	3465	81.45	25.05	3443	81	24	4746	61	33
82	25.829596	3468	81.23	26.32	3452	82	27	4937	62	38
82	28.699552	3473	81.23	30.15	3480	80	31	4978	62	41
82	31.569507	3459	81.25	32.27	3486	81	33	4987	62	45
82	34.439462	3450	81.48	34.27	3488	80	37	5157	64	49
82	37.309417	3455	81.87	36.75	3464	82	37	5286	64	52
82	40.179372	3460	82.45	41.2	3442	81	41	5237	63	54
82	43.049327	3473	81.56	43.15	3465	82	43	5122	62	57
82	45.919283	3463	81.4	46.19	3458	81	46	4946	61	60
82	48.789238	3453	81.25	48.62	3435	82	49	5077	63	62
82	51.659193	3456	80.56	50.89	3467	81	50	5065	64	65
82	54.529148	3466	80.67	53.74	3439	81	54	5247	66	69
82	57.399103	3459	80.45	57.77	3445	81	57	5422	68	74
82	60.269058	3458	80.66	59.85	3440	81	60	5414	67	76
82	63.139013	3449	80.3	64.45	3451	81	64	5189	65	80
82	66.008969	3471	80.47	65.48	3470	81	66	5172	65	83
82	68.878924	3470	81.44	69.56	3487	81	69	5167	64	87
82	71.748879	3472	81.33	72.47	3486	81	72	5489	63	85
82	74.618834	3472	81.47	74.58	3455	81	74	5342	63	87
82	77.488789	3470	81.47	76.98	3461	81	77	5367	63	90
82	80.358744	3471	81.34	80.25	3459	81	80	5364	64	93
82	83.2287	3468	81.45	83.24	3478	82	83	5314	64	96
82	86.098655	3471	81.34	86.41	3455	82	86	5181	63	97
82	88.96861	3467	81.78	88.54	3455	82	88	5278	65	100
82	91.838565	3471	80.45	92.57	3481	82	92	5351	65	103
82	94.70852	3472	80.56	95.41	3445	82	95	5262	64	108
82	97.578475	3470	80.45	97.42	3445	82	96	5102	63	109
82	100.44843	3467	80.55	97.99	3446	82	97	4944	62	115
82	101.99821	3470	80.44	101.89	3469	82	100	4978	62	118
82	104.86816	3471	80.76	102.89	3456	82	102	5034	62	122

Anexo J - Tabela dos Valores do Teste t de Student correspondentes aos níveis de probabilidade, GOMES(1976).

Graus de Liberdade	10%	5%	2%	1%	0,1%
1	6.31	12.71	31.82	63.66	636.62
2	2.92	4.30	6.97	9.92	31.60
3	2.35	3.18	4.54	5.84	12.94
4	2.13	2.78	3.75	4.60	8.61
5	2.02	2.57	3.37	4.03	6.86
6	1.94	2.45	3.14	3.71	5.96
7	1.90	2.36	3.10	3.50	5.41
8	1.86	2.31	2.90	3.36	5.04
9	1.83	2.26	2.82	3.25	4.78
10	1.81	2.23	2.76	3.17	4.59
11	1.80	2.20	2.72	3.11	4.44
12	1.78	2.18	2.68	3.06	4.32
13	1.77	2.16	2.65	3.01	4.22
14	1.76	2.14	2.62	2.98	4.14
15	1.75	2.13	2.60	2.95	4.07
16	1.75	2.12	2.58	2.92	4.02
17	1.74	2.11	2.57	2.90	3.97
20	1.73	2.09	2.53	2.84	3.85
21	1.72	2.08	2.52	2.83	3.82
22	1.72	2.07	2.51	2.82	3.79
23	1.71	2.07	2.50	2.81	3.77
24	1.71	2.06	2.49	2.80	3.75
25	1.71	2.06	2.49	2.79	3.73
26	1.71	2.06	2.48	2.78	3.71
27	1.70	2.05	2.47	2.77	3.69
28	1.70	2.05	2.47	2.76	3.67
29	1.70	2.04	2.46	2.76	3.66
30	1.70	2.04	2.46	2.75	3.65
40	1.68	2.02	2.42	2.70	3.55
60	1.67	2.00	2.39	2.66	3.46
120	1.65	1.98	2.36	2.62	3.37
∞	1.65	1.96	2.33	2.58	3.29

Anexo M - Gráfico referente às coordenadas das imagens transladadas na vertical. Total de 31 imagens analisadas.

Anexo N - Gráfico referente às coordenadas das imagens transladadas na horizontal. Total de 31 imagens analisadas..

Anexo O- Listagem do Programa Principal da Ht em Linguagem C

```
//=====//
// PROGRAMA PRINCIPAL DA TRANSFORMADA DE HOUGH //
// //
// TESE DE MESTRADO - APLICACAO DA TRANSFORMADA DE HOUGH EM INSPECAO //
// VISUAL AUTOMATICA //
// //
// AUTOR: Eng. ERNANY PARANAGUÁ DA SILVA //
// //
//=====//
//
// TAREFAS: - ABRE OS ARQUIVO *.DAT QUE CONTENHA AS IMAGENS;
//
// - CRIA DOIS PONTEIROS COM OS DADOS DA IMAGEM;
//
// - DESENHA NA TELA A IMAGEM INSPECIONADA;
//
// - PROCESSA O ARRANJO ACUMULADOR;
//
// - ORDENA OS VALORES DA MATRIZ E ALOCA EM UM PONTEIRO;
//
// - REALIZA FILTRAGEM GROSSEIRA;
//
// - REALIZA FILTRAGEM FINA;
//
// - CRIA UM PONTEIRO COM OS VALORES DOS RESULTADOS
//
// - CALCULA : * LADOS DO RETANGULO
//
// * PERIMETRO
//
// * AREA
//
// * CENTRO DE MASSA
//
// - DESENHA NA TELA O RESULTADO DAS FILTRAGENS E COLOCA O
// RESULTADO DOS CALCULOS
//
//
//=====//
// OUTROS MODULOS:
//
//=====//
//
// - BIBLIOTECA GRAFICA: * GPRINT1.I
//
//
// MainWindow(); // ABERTURA DE TELA
// changetextstyle(); // ESTILO DE LETRA
// DrawBorder(); // DESENHO DE BORDAS
// Pause(); // PAUSA NO SISTEMA
// Parada(); // FINALIZA A FUNCAO
// StatusLine(); // VERIFICA O ESTADO DA LINHA
// SayGoodbye(); // FINALIZA O PROGRAMA
// Beep(); // AVISO SONORO
// RESTORE_VIDEO(); // RESTAURA O VIDEO
// SAVE_VIDEO(); // SALVA UMA AREA ESPECIFICA DO VIDEO
// graph_init(); // INICIALIZACAO DOS PARAMETROS GRAFICOS
// erasestr(); // APAGA STRING NA TELA
```



```

//      gprintf();      // ESCRIBE NA TELA
//      gprintxy();     // ESCRIBE NA TELA EM POSICAO ESPECIFICADA
//      le_tela();       // LE DADOS NA TELA
//      le_string();     // VERIFICA TAMANHO DOS CARACTERES DE ESCRITA
//      gprintc();      // RESTAURA A COR DE FUNDO DA TELA APOS ESCRITA
//
//
// - DETECTOR.C
//
//      TAREFAS: - CALCULA O DETECTOR DE BORDAS PARA A HT
//
//                  - CALCULA A AREA E O CENTRO DE MASSA POR MOMENTO
//
// - CAPTIMAG.C
//
//      TAREFA: - CAPTURA IMAGEM DA FRAME-GRABBER
//
//=====//

```

INICIO DO PROGRAMA

```

//=====//
// Definicoes e Inclusoes
//=====//

```

```

#include <time.h>
#include <dos.h>
#include <stdio.h>
#include <conio.h>
#include <math.h>
#include <stdlib.h>
#include <graphics.h>
#include <stdarg.h>
#include <gprint1.i>
#define pi 3.14159265358979
#define maximo(valor1,valor2) valor1>valor2 ? valor1 : valor2

```

```

//=====//
//      Variaveis Globais
//=====//

```

```

extern unsigned _stklen = 64000U; // AUMENTAR O TAMANHO DA PILHA

```

```

int cont=-1,cont2=-1,valor1,valor2;
int huge matriz[300][180]; // MATRIZ GLOBAL PARA O ARRANJO ACUMULADOR

```

```

char nome_arq_x[15];
char nome_arq_y[15];
char nome_ind[15];
float divisao_ro,divisao_teta,delta_teta,delta_ro,periodo;
int n,pontos;

```

```

void calculo(int *vetor_x,int *vetor_y,int indice_maior,int quant_vetor);
void mostra_matriz(void);
void mat_vet(void);
void threshold(float limite);
float valor_maior(void);

```

```

void ordena (int **vetor);
void intervalo (int **vetor);
void desenha_retas(int *vetor_x,int *vetor_y,int quant_vetor,int *pont_matriz,int orto);//,int
imag_hor_ver);
void Filtro_Modificado(void);
void save(void);

//=====//
//PROGRAMA PRINCIPAL
//
// NOME: MAIN()
//
// FUNCOES: FORNECE AS TELAS DE OPCOES PARA EXECUTAR O PROGRAMA
//
//=====//

void main(void)
{
    FILE *op;
    register int i, j;
    char nome[13];
    int assin[5], result,cor=0;
    int cont_ponto=0,pf,bs;
    char c,s,p,ta,pa,pe,de,pee;
    int valor_tela;
    int *valores_vet_x;
    int *valores_vet_y;
    int *pont_matriz;
    int a,b,RO,TETA;
    int jj=0,tamanho,maior_numero=0,temporario,num=1,orto;

    //----- entrada_leitura_vetores() -----//

    graph_init();
    MaxColors = getmaxcolor() + 1; /* Read maximum number of colors*/
    MaxX = getmaxx();
    MaxY = getmaxy();

    /* Le o tamanho da tela */
    MainWindow("==== Programa da Transformada de Hough =====");

    SAVE_VIDEO(40, 40, 450, 102);
    setcolor(15);
    setfillstyle(SOLID_FILL, 1);
    bar(41, 41, 449, 101);
    rectangle(40,40,450,102);
    result = 0;

    outtextxy( 70, 50, "OPEN:");
    le_string(70,60,nome);

    RESTORE_VIDEO(40,40);

    /* abertura de arquivo */
    if((op=fopen(nome,"rb")) !=NULL) {
        /* leitura do cabecalho */
        for(i=0;i<5;i++)fscanf(op,"%d",&assin[i]);
        /* certificar-se se e uma figura gravada pelo programa */
        if(assin[0] != 67) result++;
        if(assin[1] != 69) result++;
        if(assin[2] != 76) result++;
    }

```

```

if(assin[3] != 83) result++;
if(assin[4] != 79) result++;
if(result == 0) {
    fscanf(op,"%d",&pf);
    fscanf(op,"%d",&bs);

    for(i=0;i<pf;i++)
        for(j=0;j<bs;j++) {
            fscanf(op,"%d",&cor);
            if(cor==0){
                putpixel(j+20,i+20,0);
                cont_ponto++;
            }
            else
                putpixel(j+20,i+20,15);
        }
    }
else{
    outtextxy(150, 66,"Arquivo");
    outtextxy(150, 72,"Irreconhecivel");
    Parada();
    SayGoodbye();
    closegraph();
    exit(1);
}

}
else {
    outtextxy(150, 66, "Arquivo nao");
    outtextxy(150, 72, "encontrado");
    Parada();
    SayGoodbye();
    closegraph();
    exit(1);
}

fclose(op);

if((valores_vet_x=(int *) malloc(cont_ponto*sizeof(int)))==NULL){
    fprintf(stderr, "Nao posso alocar memoria para valores_vet_x\n");
    Pause();
    exit(1);
}
if((valores_vet_y=(int *) malloc(cont_ponto*sizeof(int)))==NULL){
    fprintf(stderr, "Nao posso alocar memoria para valores_vet_y\n");
    Pause();
    exit(1);
}
a=0;
for(i=0;i<pf;i++)
    for(j=0;j<bs;j++) {
        valor_tela=getpixel(j+20,i+20);
        if(valor_tela==0){
            // escreve os pontos da imagem para o vetor
            *(valores_vet_x+a)=j;
            *(valores_vet_y+a)=i;
            temporario = maximo(*(valores_vet_x+a),*(valores_vet_y+a));
            maior_numero = maximo(maior_numero,temporario);
            a++;
        }
    }
}

```



```

    }
}

Beep();
Parada();

//-----//

calculo(valores_vet_x,valores_vet_y,maior_numero,cont_ponto);
MainWindow("==== Programa da Transformada de Hough =====");
pos_x=10;
pos_y=10;
settextstyle(0, HORIZ_DIR, 1);
settextjustify( LEFT_TEXT, TOP_TEXT );

gprintf(&pos_x,&pos_y,"Imagem proxima de 90 ou 0 ?...(s/n)");
do {
    ta=getch();
} while ((ta!='N') && (ta!='n') && (ta!='S') && (ta!='s'));
pos_x=340;
pos_y=10;
gprintf(&pos_x,&pos_y,"%c",ta);
if(ta=='s' || ta=='S')
    periodo=divisao_teta-10;
else if(ta=='n' || ta=='N')
    periodo=divisao_teta;
pos_x=10;
pos_y=20;

gprintf(&pos_x,&pos_y,"Deseja salvar o ARRANJO_ACUMULADOR ?...(s/n)");
do {
    s=getch();
}while ((s!='N') && (s!='n') && (s!='S') && (s!='s'));
gprintf(380,pos_y-10,"%c",s);
if(s=='s' || s=='S')
    save();
gprintf(&pos_x,&pos_y,"Deseja mostrar o arranjo acumulador ?...(s/n)");
do {
    p=getch();
} while ((p!='N') && (p!='n') && (p!='S') && (p!='s'));
gprintf(380,pos_y-10,"%c",p);
if((p=='s' || p=='S')){
    mostra_matriz();
    MainWindow("==== Programa da Transformada de Hough =====");
    settextstyle(0, HORIZ_DIR, 1);
    settextjustify( LEFT_TEXT, TOP_TEXT );
    pos_x=10;
    pos_y=10;
}
mat_vet();

Again:
do {
    jj+=1;
    MainWindow("==== Programa da Transformada de Hough =====");
    settextstyle(0, HORIZ_DIR, 1);

```

```

settextjustify( LEFT_TEXT, TOP_TEXT );
pos_x=10;
pos_y=10;

gprintf(&pos_x,&pos_y,"Deseja Aplicar Threshold ?...(s/n)");
do {
    c=getch();
} while ((c!='N') && (c!='n') && (c!='S') && (c!='s'));
gprintxy(300,pos_y-10,"%c",c);
if(c=='s' || c=='S'){
    gprintf(&pos_x,&pos_y,"Aplicar Valor = ");
    tamanho=le_tela(150,pos_y-10);
    threshold(tamanho);
    clearviewport();
    settextstyle(0, HORIZ_DIR, 1);
    settextjustify( LEFT_TEXT, TOP_TEXT );
    pos_x=10;
    pos_y=10;
    gprintf(&pos_x,&pos_y,"Deseja mostrar as coordenadas do arranjo acumulador ?...(s/n)");
    do {
        pa=getch();
    } while ((pa!='N') && (pa!='n') && (pa!='S') && (pa!='s'));
    pos_x=500;
    pos_y=10;
    gprintf(&pos_x,&pos_y,"%c",pa);

    pos_x=10;
    pos_y=30;
    num=1;

    if(pa=='s' || pa=='S'){
        for(a=0;a<=divisao_ro;a++)
            for(b=0;b<periodo;b++) //Mudado pois intervalo de (-90 a 89) opcao
                if(*(matriz+a)+b != 0){
                    RO= delta_ro*(a-floor(divisao_ro/2));
                    TETA=(b-(divisao_teta/2))*delta_teta;
                    gprintf(&pos_x,&pos_y,"%2d - Valor da Matriz  %3d - Valor RO = %4d -
Valor TETA = %4d", num++, *(matriz+a)+b,RO,TETA);
                }
    }
}

```

```

Beep();
Parada();
}

//=====
//  Aplicar limites  aos valores ortogonais
//=====
if((pont_matriz=(int *) malloc(divisao_teta*sizeof(int)))==NULL){
    fprintf(stderr, "Nao posso alocar memo para pont_matriz\n");
    Pause();
    exit(1);
}
MainWindow("===== Programa da Transformada de Hough =====");
settextstyle(0, HORIZ_DIR, 1);
settextjustify( LEFT_TEXT, TOP_TEXT );
pos_x=10;
pos_y=10;
gprintf(&pos_x,&pos_y,"Deseja salvar o ARRANJO_ACUMULADOR ?...(s/n)");
do {
    s=getch();
}

```

```

} while ((s!='N') && (s!='n') && (s!='S') && (s!='s'));
gprintxy(380,pos_y-10,"%c",s);
if(s=='s' || s=='S')
    save();
pos_y+=10;
gprintxy(250,pos_y,"FILTRAGEM");
pos_y+=20;
gprintf(&pos_x,&pos_y,"FILTRAGEM GROSSEIRA OU FINA -tecle  ""o"" ");
do {
    de=getch();
} while ((de!='o') && (de!='O'));
gprintxy(390,pos_y-10,"%c",de);
//=====
//    Aplicar filtro no arranjo acumulador
//=====

if(de=='o' || de=='O'){
    do{
        gprintf(&pos_x,&pos_y,"Deseja aplicar Filtro_Modificado no arranjo acumulador
?...(s/n)");
        do {
            pe=getch();
        } while ((pe!='N') && (pe!='n') && (pe!='S') && (pe!='s'));
        gprintxy(450,pos_y-10,"%c",pe);
        if(pe=='s' || pe=='S')
            Filtro_Modificado();
        gprintf(&pos_x,&pos_y,"Deseja aplicar novamente o filtro ?...(s/n)");
        do {
            pee=getch();
        } while ((pee!='N') && (pee!='n') && (pee!='S') && (pee!='s'));
        gprintxy(380,pos_y-10,"%c",pee);
    } while ((pee!='N') && (pee!='n') );//&& (pee!='S') && (pee!='s'));
    //=====
    //    Mostrar arranjo acumulador
    //=====

    gprintf(&pos_x,&pos_y,"Deseja mostrar o arranjo acumulador ?...(s/n)");
    do {
        p=getch();
    } while ((p!='N') && (p!='n') && (p!='S') && (p!='s'));
    gprintxy(380,pos_y-10,"%c",p);
    if(p=='s' || p=='S'){
        mostra_matriz();
        MainWindow("==== Programa da Transformada de Hough =====");
        settextstyle(0, HORIZ_DIR, 1);
        settextjustify( LEFT_TEXT, TOP_TEXT );
        pos_x=10;
        pos_y=10;
    }

    settextstyle(0, HORIZ_DIR, 1);
    settextjustify( LEFT_TEXT, TOP_TEXT );

    gprintf(&pos_x,&pos_y,"Deseja salvar o ARRANJO_ACUMULADOR ?...(s/n)");
    do {
        s=getch();
    } while ((s!='N') && (s!='n') && (s!='S') && (s!='s'));
    gprintxy(380,pos_y-10,"%c",s);
    if(s=='s' || s=='S')

```



```

        save();

        orto=0;
        desenha_retas(valores_vet_x,valores_vet_y,cont_ponto,pont_matriz,orto);
    }
}
}while(c=='s' || c=='S');
if (j == 1){
    goto Again;
}

free(pont_matriz);
free(valores_vet_x);
free(valores_vet_y);
SayGoodbye();
closegraph();

}

//=====================================================//
// NOME: CALCULO()
//
// FUNCAO: CALCULO DO ARRANJO ACUMULADOR
//
//=====================================================//

void calculo(int *vetor_x,int *vetor_y,int indice_maior,int quant_vetor)
{
    register int a,j;
    int temp, r1, t1, D1;
    float r2, r3,r;
    clock_t start, end;//marca o tempo de execucao
    MainWindow("===== Programa da Transformada de Hough =====");
    settextrstyle(0, HORIZ_DIR, 1);
    settextrjustify( LEFT_TEXT, TOP_TEXT );
    outtextxy(20, 20, "Entre com a variacao de teta (menor valor de angulo ""1"": ");
    delta_teta=le_tela(480,20);
    outtextxy(20, 40, "Entre com a variacao de ro(menor valor ""2"": ");
    delta_ro=le_tela(400 ,40);

    start = clock();//inicio do tempo de execucao

    D1 = ceil(1.2 *indice_maior) * 2;
    divisao_ro = D1/delta_ro; //verificar ceil(D1/delta_ro)
    divisao_teta = (180/delta_teta);

    for(a=0; a<=divisao_ro; a++)
        for(j=0; j<divisao_teta; j++) //Modificado Mudado pois intervalo de (-90 a 89)
            (*(matriz+a+j))=0;

    r2 = -90;
    for(j=0;j<quant_vetor;j++) { //quantidade de pontos na imagem
        for(a=0; a<divisao_teta;a++) { //Mudado pois intervalo de (-90 a 89)
            r = (*(vetor_x+j)* cos(r2*pi/180)) + (*(vetor_y+j)*sin(r2*pi/180));
            r3 = r/delta_ro; //Definido como float
            temp= r/delta_ro;
            // Definido como int
            // Caso mude para valor float o intervalo passa para

```

```

// r3>0.5 ....intervalo de (0 == 1,9) ; (2 == 3,9).....
// r3<=0.5....intervalo de (-6 == -4,1) ; (-4 == -2,1) ..
if(r3>0) {
    if(r3>(0.5+temp)) r = ceil(r3); //Para angulo de -90 a 90 == " 0.5 "
    else r = floor(r3);
    //intervalo de (-1 == 0,9) ; (1 == 2,9) ; (3 == 5,9).....para ro= 2...
    //intervalo de (-0.51 == 0,5) ; (0.51 == 1.5) ; (1.51 == 2.5) ; (2.51 == 3.5).....para
    ro= 1..
}
if(r3<=0) {
    if(r3>(-0.5+temp)) r = ceil(r3);
    //O valor e' negativo de " -0.5 " ,caso contrario o intervalo altera...
    else r = floor(r3);
    //intervalo de ..... (-5 == -3,1) ; (-3 == -1,1) ; (-1 == 0,9) .....para ro=2
    //intervalo de (-5.5 == -4.51) ; (-4.5 == -3.51) ; (-3.5 == -2.51) ;
    (-2.5 == 1.51).....para ro= 1..
}
r1 = (divisao_ro/2) + r;
// Este ponto nao importa colocar a funcao " ceil() " or
// a funcao " floor " pois de qualquer forma o resultado
// sera uma funcao " int ".
t1 = (divisao_teta/2) + (r2/delta_teta);
*(matriz+r1)+t1 += 1;
r2 += delta_teta;
}
r2=-90;
}
end = clock(); //final do tempo de execucao
gprintxy( 12,202,"O tempo do calculo do arranjo foi: %4.6f segundos", (end - start) / CLK_TCK);
Beep();
Parada();
}

//=====//
// NOME : SAVE()
//
// FUNCAO: SALVAR O ARRANJO ACUMULADOR
//
//=====//

void save(void)
{
    int a=0,j;
    char arquivo[12];
    FILE *arquivo_acumulador;

    gprintf(&pos_x,&pos_y,"Digite o nome do arquivo : ");
    le_string(270,pos_y-10,arquivo);
    if((arquivo_acumulador = fopen(arquivo, "wb"))==NULL) {
        outtextxy(pos_x+20,pos_y+20,"Nao posso criar arquivo ""ACUMULADOR""\n");
        Pause();
        exit(1);
    }
    for(a=0;a<=divisao_ro;a++) {
        for(j=0;j<periodo;j++) //Mudado poi intervalo (-90 a 89)
            fprintf(arquivo_acumulador, " %4d ", (*(matriz+a)+j));
        fprintf(arquivo_acumulador, "\n");
    }
    fclose(arquivo_acumulador);
}

```

```

}

//=====================================================//
// NOME :  MOSTRA_MATRIZ()
//
// FUNCAO :  MOSTRAR A MATRIZ DO ARRANJO ACUMULADOR NA TELA
//
//=====================================================//
void mostra_matriz()
{
    int a, k, j, cont,mult, r1;
    int lim_delta_teta1, lim_delta_teta2,coluna,Xi,decrementa;
    float r2,r;

    MainWindow("TABELA DO ARRANJO ACUMULADOR");
    settextstyle(2, HORIZ_DIR, 3);
    settextjustify( LEFT_TEXT, TOP_TEXT );

    mult = (divisao_teta/17);// " mult " = a quantidade de telas mostradas
                                // A divisao e' feita por 10 pois na tela cabem
                                // 10 colunas....

    lim_delta_teta1=0;
    lim_delta_teta2 =0;
    coluna=0;
    pos_x=0;
    pos_y=10;
    for(j=0;j<=mult;j++) { //quantidades de telas
        a = 0;
        r2 = -90 + (delta_teta*17*j); //variacao de teta de 10 em 10
        do {
            pos_x+=20;
            gprintf(&pos_x,&pos_y,"%4.0f",r2);
            pos_y=10;
            r2 += delta_teta;
            a += 1;
            lim_delta_teta1++;
            if(lim_delta_teta1>=periodo) //Mudado pois intervalo de (-90 a 89)break;
                break;                                // Limite maximo do eixo x,
                                                        // o eixo de teta
        } while(a<17);
        k = 0;
        r1 = 0;
        cont =10;
        pos_y=20;
        do {
            r2 = -90 + (delta_teta*17*j);
            r = r1 - floor(divisao_ro/2);
            // Verificar se mudando os valores de delta_ro ou delta_teta modifica
            // a tabela
            pos_x=0;
            gprintf(&pos_x,&pos_y,"%5.0f",r*delta_ro); //Valores da tabela do eixo Y
            Xi=0;
            decrementa=0;
            do {
                decrementa++;
                Xi++;
                pos_x+=20;
                pos_y-=8;
            } while(Xi<=mult);
            r1 = r2;
            pos_y+=8;
        } while(cont>0);
    }
}

```



```

        gprintf(&pos_x,&pos_y,"%4d", *(matriz+r1)+coluna));
        //Matriz do arranjo acumulador

        lim_delta_teta2++;

        coluna++;
        if(lim_delta_teta2>=periodo) //Limite maximo do eixo x,
            break; //Mudado pois intervalo de (-90 a 89)
            // o eixo de teta
    } while(Xi<17);
    pos_y+=10;
    lim_delta_teta2-= decrementa;
    coluna-=decrementa;
    r1 += 1;
    cont+=10;
    if(cont>=445) { // 47 = 50 - 3 linhas de mensagem
        // Quando coloca no modo " textmodo(64) " para video VGA a tela passa
        // para 50 linhas.
        // Se tirar este modo o video passa para 25 linhas
        Parada();
        clearviewport();
        pos_x=0;
        pos_y=10;
        cont=10;
        settextstyle(2, HORIZ_DIR, 3);
        settextjustify( LEFT_TEXT, TOP_TEXT );
    }
    k += 1;
}while(k<=divisao_ro); //Variacao maxima dos valores da tabela do eixo Y
lim_delta_teta2+=17; //tela maxima para teta
coluna+=17; //teta maxima para teta
Parada();
clearviewport();
pos_x=0;
pos_y=10;
cont=10;
settextstyle(2, HORIZ_DIR, 3);
settextjustify( LEFT_TEXT, TOP_TEXT );

}
}

//=====//
// NOME : MAT_VET()
//
// FUNCAO : ALOCA PONTEIRO PARA O RESULTADO DO ARRANJO ACUMULADOR
//
//=====//

void mat_vet()
{
    /*=====
        FORAM USADAS DUAS VARIAVEIS GLOBAIS:
            cont,cont2
        =====*/
    int a,j,*vetor;

    /*=====
        CRIANDO UM VETOR DE ELEMENTOS NAO NULOS
        =====*/

```

```

/*=====
ESTA PARTE DA FUNCAO APENAS VERIFICA A QUANTIDADE DE ELEMENTOS NAO
NULOS E ALOCA UM PONTEIRO PARA ESTES ELEMENTOS
=====*/

for(a=0;a<=divisao_ro;a++)
    for(j=0;j<periodo;j++) //Mudado pois intervalo de (-90 a 89)
        if (matriz[a][j] != 0) cont += 1;
vetor =(int*) malloc((cont+1)*sizeof(int));
if (!vetor){
    printf ("\nFalha na solicitacao de memoria\n");
    printf ("\nTecla algo para continuar");
    getch();
    exit(1);
}

/*-----*/

cont = -1;
for(a=0;a<=divisao_ro;a++)
    for(j=0;j<periodo;j++) { //Mudado pois intervalo de (-90 a 89)
        if (matriz [a][j] != 0){
            cont += 1;
            vetor [cont] = matriz [a][j];
        }
    }
intervalo (&vetor);
free(vetor);
}

//=====
// NOME : VALOR_MAIOR()
//
// FUNCAO : ENCONTRAR O MAIOR VALOR DO ARRANJO ACUMULADOR
//
//=====

float valor_maior(void)
{
    int a,j;
    float res,abc=0;

    for(a=0;a<=divisao_ro;a++)
        for(j=0;j<periodo;j++) { //Mudado pois intervalo de (-90 a 89)
            res = maximo(*(matriz+a)+j),*(matriz+a)+j+1));
            abc = maximo(abc,res);
        }
    gprintf(&pos_x,&pos_y,"Maior numero da matriz e %8.2f", abc);
    return(abc);
}

//=====
// NOME : ORDENA()
//

```

```
// FUNCAO : ORDENA O PONTEIRO QUE CONTEM OS VALORES DO ARRANJO ACUMULADOR
//
//=====//
```

```
void ordena (int **vetor)
{
```

```
    int a_maior, ind1=0, ind2=0, maior=0, *vet_ord;
```

```
    if((vet_ord=(int *) malloc((cont2+1)*sizeof(int)))==NULL){
        fprintf(stderr, "Nao posso alocar memoria para vet_ord - ordena()");
        Pause();
        exit(1);
    }
```

```
    for (ind2=0; ind2<=cont2; ind2++){
        maior=0;
        for (ind1=0; ind1<=cont2; ind1++){
            if ((*vetor)[ind1] >= maior){
                maior = (*vetor)[ind1];
                a_maior = ind1;
            }
            vet_ord[ind2] = maior;
            (*vetor)[a_maior] = 0;
            pos_x=10;
            gprintf(&pos_x, &pos_y, "Elemento [%i] do intervalo : %i", ind2, vet_ord[ind2]);
        }
        Parada();
        free(vet_ord);
    }
```

```
//=====//
// NOME : INTERVALO()
//
// FUNCAO : VERIFICA O INTERVALO DE INTERESSE PARA A FILTRAGEM GROSSEIRA
//
//=====//
```

```
void intervalo (int **vetor)
{
```

```
    int ind, *vet_int;
    float inter, limite, mxm;
```

```
    mxm = valor_maior();
    gprintf(&pos_x, &pos_y, "Intervalo de interesse ?(em %) ");
    inter=le_tela(280, pos_y-10);
    limite = (float)(mxm-((inter/100)*mxm));
    gprintf(&pos_x, &pos_y, "limite : %4.2f", limite);
```

```
/*=====
ESTA PARTE DA FUNCAO APENAS VERIFICA A QUANTIDADE DE ELEMENTOS DO
INTERVALO DE INTERESSE E ALOCA UM PONTEIRO PARA ESTES ELEMENTOS
=====*/
```

```
    for (ind=0; ind<=cont; ind++){
        if ((*vetor)[ind] >= limite) cont2 += 1;
        vet_int=(int*) malloc((cont2+1)*sizeof(int));
```



```

    if (!vet_int){
        outtextxy(pos_x,pos_y,"Falha na solicitacao de memoria1 em intervalo()");
        Pause();
        exit(1);
    }

    /*-----*/
    cont2 = -1;
    for (ind=0;ind<=cont;ind++){
        if ((*vetor)[ind] >= limite){
            cont2 += 1;
            vet_int[cont2] = (*vetor)[ind];
        };
    }
    ordena (&vet_int);
    free(vet_int);
}

//=====//
// NOME : THRESHOLD()
//
// FUNCAO : REALIZA A FILTRAGEM GROSSEIRA
//
//=====//

void threshold(float limite)
{
    int i,j;
    for(i=0;i<=divisao_ro;i++){
        for(j=0;j<periodo;j++) //Mudado pois intervalo de (-90 a 89)
        {
            if ((*matriz+i)+j) < limite)
                ((*matriz+i)+j) = 0;
        }
    }
}

//=====//
// NOME : Filtro_Modificado()
//
// FUNCAO : FILTRAGEM FINA DO ARRANJO ACUMULADOR
//
//=====//

void Filtro_Modificado(void)
{
    int a,b,RO1,RO2=0,TETA1,TETA2=0,MAT2=0,variacao,Temp1,Temp2;
    int MAT1, indice_a=0,indice_b=0,Delta_R,Delta_T;

    gprintf(&pos_x,&pos_y,"Digite o valor da variacao: ");
    variacao=le_tela(250,pos_y-10);
    for(a=0;a<=divisao_ro;a++){
        for(b=0;b<periodo;b++) //Mudado pois intervalo de (-90 a 89)

            if ((*matriz+a)+b) != 0){
                RO1= delta_ro*(a-floor(divisao_ro/2));
                TETA1=(b-(divisao_teta/2))*delta_teta;
                MAT1=(*matriz+a)+b;
                Temp1=RO1 - RO2;
            }
    }
}

```

```

Delta_R=abs(Temp1);
Temp2=TETA1 - TETA2;
Delta_T=abs(Temp2);
if((Delta_R<=variacao)&&(Delta_R>0)){
    if(Delta_T==0){
        if(MAT1 > MAT2){
            RO2=RO1;
            MAT2 = MAT1;
            (*(matriz+indice_a)+indice_b)=0;
            indice_a=a;
            indice_b=b;
        }
        else if(MAT1<MAT2){
            (*(matriz+a)+b) = 0;
        }
        else if(MAT1==MAT2){
            if(TETA1 < TETA2){ //Mudado para <
                TETA2=TETA1;
                (*(matriz+indice_a)+indice_b)=0;
                indice_a=a;
                indice_b=b;
            }
            else{
                (*(matriz+a)+b) = 0;
            }
        }
    }
    else if((Delta_T>=delta_teta)&&(Delta_T<=variacao)){ //2*delta_teta){
        if(MAT1 > MAT2){
            MAT2 = MAT1;
            RO2=RO1;
            TETA2=TETA1;
            (*(matriz+indice_a)+indice_b)=0;
            indice_a=a;
            indice_b=b;
        }
        else if(MAT1 < MAT2){
            (*(matriz+a)+b) = 0;
        }
        else if(MAT1==MAT2){
            if(TETA1 < TETA2){ //Mudado para <
                TETA2=TETA1;
                (*(matriz+indice_a)+indice_b)=0;
                indice_a=a;
                indice_b=b;
            }
            else{
                (*(matriz+a)+b) = 0;
            }
        }
    }
    else {
        RO2=RO1;
        TETA2=TETA1;
        MAT2=MAT1;
        indice_a=a;
        indice_b=b;
    }
}
}

```

```

else if(Delta_R==0){
    if((Delta_T>=delta_teta)&&(Delta_T<=variacao)){ //2*delta_teta){
        if(MAT1 > MAT2){
            TETA2=TETA1;
            MAT2 = MAT1;
            *(*(matriz+indice_a)+indice_b)=0;
            indice_a=a;
            indice_b=b;
        }
        else if(MAT1< MAT2){
            MAT2 = MAT2;
            *(*(matriz+a)+b) = 0;
        }
        else if(MAT1==MAT2){
            if(TETA1 < TETA2){ //Mudado para <
                TETA2=TETA1;
                *(*(matriz+indice_a)+indice_b)=0;
                indice_a=a;
                indice_b=b;
            }
            else{
                *(*(matriz+a)+b) = 0;
            }
        }
    }
    else {
        RO2=RO1;
        TETA2=TETA1;
        MAT2=MAT1;
        indice_a=a;
        indice_b=b;
    }
}

else{
    RO2=RO1;
    TETA2=TETA1;
    MAT2=MAT1;
    indice_a=a;
    indice_b=b;
}
}

//-----//
//----Repete a primeira parte com a inversao dos for's-----//
//-----//

for(b=0;b<periodo;b++) //Mudado pois intervalo de (-90 a 89)
for(a=0;a<=divisao_ro;a++)

    if(*(*(matriz+a)+b) != 0){
        RO1= delta_ro*(a-floor(divisao_ro/2));
        TETA1=(b-(divisao_teta/2))*delta_teta;
        MAT1=*(*(matriz+a)+b);
        Temp1=RO1 - RO2;
        Delta_R=abs(Temp1);
        Temp2=TETA1 - TETA2;
        Delta_T=abs(Temp2);
        if((Delta_R<=variacao)&&(Delta_R>0)){

```



```

if(Delta_T==0){
    if(MAT1 > MAT2){
        RO2=RO1;
        MAT2 = MAT1;
        *(*(matriz+indice_a)+indice_b)=0;
        indice_a=a;
        indice_b=b;
    }
    else if(MAT1<MAT2)
        *(*(matriz+a)+b) = 0;
    else if(MAT1==MAT2){
        if(TETA1 < TETA2){ //Mudado para <
            TETA2=TETA1;
            *(*(matriz+indice_a)+indice_b)=0;
            indice_a=a;
            indice_b=b;
        }
        else
            *(*(matriz+a)+b) = 0;
    }
}
else if((Delta_T>=delta_teta)&&(Delta_T<=variacao)){ //2*delta_teta){
    if(MAT1 > MAT2){
        MAT2 = MAT1;
        RO2=RO1;
        TETA2=TETA1;
        *(*(matriz+indice_a)+indice_b)=0;
        indice_a=a;
        indice_b=b;
    }
    else if(MAT1 < MAT2)
        *(*(matriz+a)+b) = 0;
    else if(MAT1==MAT2){
        if(TETA1 < TETA2){ //Mudado para <
            TETA2=TETA1;
            *(*(matriz+indice_a)+indice_b)=0;
            indice_a=a;
            indice_b=b;
        }
        else
            *(*(matriz+a)+b) = 0;
    }
}
else {
    RO2=RO1;
    TETA2=TETA1;
    MAT2=MAT1;
    indice_a=a;
    indice_b=b;
}
}
else if(Delta_R==0){
    if((Delta_T>=delta_teta)&&(Delta_T<=variacao)){
        if(MAT1 > MAT2){
            TETA2=TETA1;
            MAT2 = MAT1;
            *(*(matriz+indice_a)+indice_b)=0;
            indice_a=a;
            indice_b=b;
        }
    }
}

```

```

else if(MAT1< MAT2){
    MAT2 = MAT2;
    *(*(matriz+a)+b) = 0;
}
else if(MAT1==MAT2){
    if(TETA1 < TETA2){ //Mudado para <
        TETA2=TETA1;
        *(*(matriz+indice_a)+indice_b)=0;
        indice_a=a;
        indice_b=b;
    }
    else
        *(*(matriz+a)+b) = 0;
}
}
else {
    RO2=RO1;
    TETA2=TETA1;
    MAT2=MAT1;
    indice_a=a;
    indice_b=b;
}
}
else{
    RO2=RO1;
    TETA2=TETA1;
    MAT2=MAT1;
    indice_a=a;
    indice_b=b;
}
}

Beep();
getch();
}

//=====================================================//
// NOME : DESENHA_RETAS()
//
// FUNCOES : - COLOCA NA TELA A IMAGEM INSPECIONADA;
//           - TRACA AS RETAS DA FILTRAGEM GROSSEIRA E FINA
//           - CALCULA = * OS LADOS DO OBJETO INSPECIONADO,
//                 * O PERIMETRO
//                 * A AREA
//                 * O CENTRO DE MASSA
//=====================================================//

void desenha_retas(int *vetor_x,int *vetor_y,int quant_vetor,int *pont_matriz,int orto)
{
    int a,b,RO,TETA,midx,midy,lim_orto,ortogonal,ind_orto;
    int xx,yy,result_x,result_y,positivo_RO1,VALOR_TESTE_MATRIZ;
    int quant_pontos=0,valor_pontos,lado1,valor_ro[3],cont_ro=0,media_centro[3];
    int angulo_teta[3],cont=0,cont_ang=0,dados[3],RO1=0,TETA1=0,RO2,TETA2,cont_media=0;
    float Perimetro,Area,seno ,cosseno;
    float dist_x,dist_y,Coord_x,Coord_y,Result_hipotenusa,Valor_arco,arco_tang,temp;
    char pe;

    setviewport( 0, 0, MaxX, MaxY, 1 );      /* ABERTURA DA TELA TODA */

```

```

clearviewport();

midx = getmaxx()/2;
midy = getmaxy()/2;

setcolor(14);

// MOLDURA DO VGA (640 x 480)

line(0,0,2*midx,0); //valores ( Xo,Yo,X1,Y1 )
line(0,0,0,2*midy);
line(0,2*midy,2*midx,2*midy);
line(2*midx,0,2*midx,2*midy);

pos_x=10;
pos_y=10;
gprintf(&pos_x,&pos_y," FUNCAO DESENHO ");

//----- Colocando a imagem na tela-----//

for (a=0;a<quant_vetor;a++)
    putpixel(midx+(vetor_x+a), midy+(vetor_y+a),15);

//-----//
setcolor(15);
setlinestyle(2,16,3);

if(orto==0){
    for(a=0;a<=divisao_ro;a++)
        for(b=0;b<periodo;b++) //Mudado pois intervalo de (-90 a 89)
            if(*(matriz+a)+b) != 0){
                quant_pontos++;
                RO= delta_ro*(a-floor(divisao_ro/2));
                TETA=(b-(divisao_teta/2))*delta_teta;
                for(xx=-319;xx<=319;xx++){
                    cosseno=(cos(TETA*pi/180));
                    seno=sin(TETA*pi/180);

                    if(TETA==0 ) {
                        for(yy=-239;yy<=239;yy++){
                            result_x=midx+RO;
                            result_y=midy+yy;
                            if(result_x>=0 && result_x<=639)
                                if(result_y>=0 && result_y<=479)
                                    putpixel(result_x,result_y,2);
                        }
                        xx=320;
                    }
                    else if(TETA==90){ // || TETA==90 {
                        cosseno=0;
                        yy=(RO-(xx*cosseno))/seno;
                        result_x=midx+xx;
                        result_y=midy+yy;
                        if(result_x>=0 && result_x<=639)
                            if(result_y>=0 && result_y<=479)
                                putpixel(result_x,result_y,3);
                    }
                    else{
                        yy=(RO-(xx*cosseno))/seno;

```



```

        result_x=midx+xx;
        result_y=midy+yy;
        if(result_x>=0 && result_x<=639)
            if(result_y>=0 && result_y<=479)
                putpixel(result_x,result_y,14);
    }
}

}

//=====
else if(orto==1){

    gprintf(&pos_x,&pos_y,"Digite o valor do limite ortogonal desejado= ");
    lim_orto=le_tela(380,pos_y-10);

    for(b=0;b<periodo;b++){ //Mudado pois intervalo de (-90 a 89)
        TETA=(b-(divisao_teta/2))*delta_teta;
        if(TETA>=0)
            ortogonal=TETA-90;
        if(TETA<0)
            ortogonal=TETA+90;
        ind_orto=ortogonal/delta_teta + divisao_teta/2;
        if(pont_matriz[ind_orto]>=lim_orto && pont_matriz[b]>=lim_orto) {
            for(a=0;a<=divisao_ro;a++){
                RO= delta_ro*(a-floor(divisao_ro/2));
                if(*(matriz+a)+b) != 0 ){
                    for(xx=-319;xx<=319;xx++){
                        cosseno=(cos(TETA*pi/180));
                        seno=sin(TETA*pi/180);

                        if(TETA==0 ) {
                            for(yy=-239;yy<=239;yy++){
                                result_x=midx+RO;
                                result_y=midy+yy;
                                if(result_x>=0 && result_x<=639)
                                    if(result_y>=0 && result_y<=479)
                                        putpixel(result_x,result_y,2);
                            }
                        }
                        xx=320;
                    }
                }
            }
            else if(TETA==90){ // || TETA==90) {
                cosseno=0;
                yy=(RO-(xx*cosseno))/seno;
                result_x=midx+xx;
                result_y=midy+yy;
                if(result_x>=0 && result_x<=639)
                    if(result_y>=0 && result_y<=479)
                        putpixel(result_x,result_y,3);
            }
            else{
                yy=(RO-(xx*cosseno))/seno;
                result_x=midx+xx;
                result_y=midy+yy;
                if(result_x>=0 && result_x<=639)
                    if(result_y>=0 && result_y<=479)
                        putpixel(result_x,result_y,14);
            }
        }
    }
}

```



```

    dados[cont]=lado1;
    media_centro[cont_media++]=lado1/2+RO1;
    valor_ro[cont_ro++]=RO1;
    valor_ro[cont_ro++]=RO2;
    angulo_teta[cont_ang++]=TETA1;
    angulo_teta[cont_ang++]=TETA2;
    cont++;
    break;
case 3:
    if((RO1<0)&&(RO2>0)){
        positivo_RO1=abs(RO1);
        lado1=positivo_RO1+RO2;
    }
    else
        lado1=abs(abs(RO1)-abs(RO2));
    dados[cont]=lado1;
    media_centro[cont_media++]=lado1/2+RO1;
    valor_ro[cont_ro++]=RO1;
    valor_ro[cont_ro++]=RO2;
    angulo_teta[cont_ang++]=TETA1;
    angulo_teta[cont_ang++]=TETA2;
    cont++;
    break;
case 4:
    if((RO1<0)&&(RO2>0)){
        positivo_RO1=abs(RO1);
        lado1=positivo_RO1+RO2;
    }
    else
        lado1=abs(abs(RO1)-abs(RO2));
    dados[cont]=lado1;
    media_centro[cont_media++]=(lado1/2)+RO1;
    valor_ro[cont_ro++]=RO1;
    valor_ro[cont_ro++]=RO2;
    angulo_teta[cont_ang++]=TETA1;
    angulo_teta[cont_ang++]=TETA2;
    cont++;
    break;
case 5:
    if((RO1<0)&&(RO2>0)){
        positivo_RO1=abs(RO1);
        lado1=positivo_RO1+RO2;
    }
    else
        lado1=abs(abs(RO1)-abs(RO2));
    dados[cont]=lado1;
    media_centro[cont_media++]=lado1/2+RO1;
    valor_ro[cont_ro++]=RO1;
    valor_ro[cont_ro++]=RO2;
    angulo_teta[cont_ang++]=TETA1;
    angulo_teta[cont_ang++]=TETA2;
    cont++;
    break;
case 6:
    if((RO1<0)&&(RO2>0)){
        positivo_RO1=abs(RO1);
        lado1=positivo_RO1+RO2;
    }
    else
        lado1=abs(abs(RO1)-abs(RO2));

```



```

        dados[cont]=lado1;
        media_centro[cont_media++]=lado1/2+RO1;
        valor_ro[cont_ro++]=RO1;
        valor_ro[cont_ro++]=RO2;
        angulo_teta[cont_ang++]=TETA1;
        angulo_teta[cont_ang++]=TETA2;
        cont++;
        break;
    }
    RO1=RO2;
    TETA1=TETA2;
}
gprintf(&pos_x,&pos_y,"Vetor dados[0]= %.2d",dados[0]);
gprintf(&pos_x,&pos_y,"Vetor dados[1]= %.2d",dados[1]);

//=====
//
//Cálculo do centro de massa do objeto
//
//=====

if(abs(valor_ro[0])>abs(valor_ro[1]))
    dist_y=abs(valor_ro[0]) - dados[0]/2;
else
    dist_y=abs(valor_ro[1]) - dados[0]/2;
if(abs(valor_ro[2])>abs(valor_ro[3]))
    dist_x=abs(valor_ro[2]) - dados[1]/2;
else
    dist_x=abs(valor_ro[3]) - dados[1]/2;

if(abs(angulo_teta[0])<50){ //inversao dos valores : "dist_x" e "dist_y"
    temp = dist_x;
    dist_x = dist_y;
    dist_y = temp;
}
gprintf(&pos_x,&pos_y,"Teste dist_x = %.2f",dist_x);
gprintf(&pos_x,&pos_y,"Teste dist_y = %.2f",dist_y);

//=====

if(abs(angulo_teta[0])>=85){ //qdo o valor e' praticamente 90 graus
    Coord_x=dist_x;
    Coord_y=dist_y;
    gprintf(&pos_x,&pos_y,"Teste Coord_x = %.2f",Coord_x);
    gprintf(&pos_x,&pos_y,"Teste Coord_y = %.2f",Coord_y);
}
else{
    Result_hipotenusa=sqrt(pow(dist_x,2) + pow(dist_y,2));
    gprintf(&pos_x,&pos_y,"Teste Result_hipotenusa = %.2f",Result_hipotenusa);

    arco_tang=atan2(dist_y,dist_x)*180/pi; //passando de rad p/ graus
    gprintf(&pos_x,&pos_y,"Teste arco_tang = %.2f",arco_tang);

    if(arco_tang > abs(angulo_teta[0]))
        Valor_arco=arco_tang - abs(angulo_teta[0]);
    else
        Valor_arco=arco_tang + abs(angulo_teta[2]);
    gprintf(&pos_x,&pos_y,"Teste Valor_arco = %.2f",Valor_arco);

    Coord_x=Result_hipotenusa*cos(Valor_arco*pi/180);

```

```

        Coord_y=Result_hipotenusa*sin(Valor_arco*pi/180);
    }
    gprintf(&pos_x,&pos_y,"Coord_x = %.2f",Coord_x); //Coordenada do objeto em x
    gprintf(&pos_x,&pos_y,"Coord_y = %.2f",Coord_y); //Coordenada do objeto em y

// =====

    for(a=0;a<=3;a++){
        if(angulo_teta[a]!=0 || valor_ro[a]!=0 || media_centro[a]!=0){
            gprintf(&pos_x,&pos_y,"Angulo_teta[%d]= %.2d Valor_ro[%d]=
%.2d",a,angulo_teta[a],a,valor_ro[a]);
            // gprintf(&pos_x,&pos_y,"Valor_ro[%d]= %.2d",a,valor_ro[a]);
        }
        if(media_centro[a]!=0)
            gprintf(&pos_x,&pos_y,"Media_centro[%d]= %.2d",a,media_centro[a]);
    }
    Area=(float)(dados[0]*dados[1]);
    Perimetro=(float)(2*dados[0])+(2*dados[1]);
    gprintf(&pos_x,&pos_y,"Valor da Area= %.2f",Area);
    gprintf(&pos_x,&pos_y,"Valor do Perimetro= %.2f",Perimetro);
}

gprintf(&pos_x,&pos_y,"Deseja verificar o centro de massa?...(s/n)");
do {
    pe=getch();
} while ((pe!='N') && (pe!='n') && (pe!='S') && (pe!='s'));
gprintfxy(380,pos_y-10,"%c",pe);
if(pe=='s' || pe=='S'){
    //clearviewport();
    TETA=-70;
    RO=-42;
    for(a=0;a<1;a++){
        for(xx=-319;xx<=319;xx++){
            cosseno=(cos(TETA*pi/180));
            seno=sin(TETA*pi/180);

            if(TETA==0 ) {
                for(yy=-239;yy<=239;yy++){
                    result_x=midx+RO;
                    result_y=midy+yy;
                    if(result_x>=0 && result_x<=639)
                        if(result_y>=0 && result_y<=479)
                            putpixel(result_x,result_y,15);
                }
                xx=320;
            }
            else if(TETA==90){
                cosseno=0;
                yy=(RO-(xx*cosseeno))/seno;
                result_x=midx+xx;
                result_y=midy+yy;
                if(result_x>=0 && result_x<=639)
                    if(result_y>=0 && result_y<=479)
                        putpixel(result_x,result_y,15);
            }
            else{
                yy=(RO-(xx*cosseeno))/seno;
                result_x=midx+xx;

```

```

        result_y=midy+yy;
        if(result_x>=0 && result_x<=639)
            if(result_y>=0 && result_y<=479)
                putpixel(result_x,result_y,15);
    }
}
TETA=12;
RO=39;
for(a=0;a<1;a++){
    for(xx=-319;xx<=319;xx++){
        cosseno=(cos(TETA*pi/180));
        seno=sin(TETA*pi/180);

        if(TETA==0) {
            for(yy=-239;yy<=239;yy++){
                result_x=midx+RO;
                result_y=midy+yy;
                if(result_x>=0 && result_x<=639)
                    if(result_y>=0 && result_y<=479)
                        putpixel(result_x,result_y,2);
            }
            xx=320;
        }
        else if(TETA==90){
            cosseno=0;
            yy=(RO-(xx*cosseeno))/seno;
            result_x=midx+xx;
            result_y=midy+yy;
            if(result_x>=0 && result_x<=639)
                if(result_y>=0 && result_y<=479)
                    putpixel(result_x,result_y,3);
        }
        else{
            yy=(RO-(xx*cosseeno))/seno;
            result_x=midx+xx;
            result_y=midy+yy;
            if(result_x>=0 && result_x<=639)
                if(result_y>=0 && result_y<=479)
                    putpixel(result_x,result_y,14);
        }
    }
}
}
Parada();
}
//=====//

```


Anexo P - Listagem do Programa "Detector.C".

```
//=====//
//PROGRAMA COMPLEMENTO DO PRINCIPAL
//
//NOME    : DETECTOR.C
//
//FUNÇÃO  : CÁLCULO DE BORDA E MOMENTOS
//
//=====//

#include <stdio.h>
#include <conio.h>
#include <graphics.h>
#include <dos.h>
#include <math.h>
#include <stdlib.h>
#include <stdarg.h>
#include <gprint1.i>

int huge MATRIZ_IMAGEM[127][127];

//=====//

void main(void)
{
    FILE *op;
    FILE *retorno;
    register int i, j;
    char nome[13];
    char imagem[13],s;
    int assin[5], result,cor=0;
    int val_1=0,val_2=0;
    int pf;
    int bs;
    unsigned long momento_x=0,momento_y=0,cont_ponto=0,cont_ponto1=0;

    graph_init();
    MaxColors = getmaxcolor() + 1; /* le o numero maximo cores*/
    MaxX = getmaxx();
    MaxY = getmaxy();

    /* Moldura com titulo- limpa a tela */
    MainWindow("==== Programa da Transformada de Hough =====");
    SAVE_VIDEO(40, 40, 450, 102);
    setcolor(15);
    setfillstyle(SOLID_FILL, 1);
    bar(41, 41, 449, 101);
    rectangle(40,40,450,102);
    result = 0;
    settextstyle(0, HORIZ_DIR, 1);
    settextjustify( LEFT_TEXT, TOP_TEXT );
    pos_x=170;
    pos_y=250;
    outtextxy( 70, 50, "OPEN:");
    le_string(70,60,nome);
    RESTORE_VIDEO(40, 40);
```

```

/* abertura de arquivo */
if((op=fopen(nome,"rb")) !=NULL) {
/* leitura do cabeçalho */
for(i=0;i<5;i++)fscanf(op,"%d",&assin[i]);
/* certificar-se se e uma figura gravada pelo programa */
if(assin[0] != 67) result++;
if(assin[1] != 69) result++;
if(assin[2] != 76) result++;
if(assin[3] != 83) result++;
if(assin[4] != 79) result++;
if(result == 0) {
    fscanf(op,"%d",&pf);
    fscanf(op,"%d",&bs);

    for(i=0;i<pf;i++)
        for(j=0;j<bs;j++) {
            fscanf(op,"%d",&cor);
            putpixel(j+20,i+20,cor); //valores putpixel(x,y,color)
        }
    SAVE_VIDEO(40,240, 450, 302);
    setcolor(15);
    setfillstyle(SOLID_FILL, 1);
    bar(41,241, 449, 301);
    rectangle(40,240,450,302);

    outtextxy( 170, 250, "SALVAR COMO:");
    le_string(70,260,imagem);
    RESTORE_VIDEO(40,240);

    if((retorno=fopen(imagem,"wb"))==NULL) exit(0);

    fprintf(retorno,"%d\n",67);
    fprintf(retorno,"%d\n",69);
    fprintf(retorno,"%d\n",76);
    fprintf(retorno,"%d\n",83);
    fprintf(retorno,"%d\n",79);

    if((op=fopen(nome,"rb")) !=NULL) {
/* leitura do cabeçalho */
        for(i=0;i<5;i++)fscanf(op,"%d",&assin[i]);
        /* certificar-se se e uma figura gravada pelo programa */
        if(assin[0] != 67) result++;
        if(assin[1] != 69) result++;
        if(assin[2] != 76) result++;
        if(assin[3] != 83) result++;
        if(assin[4] != 79) result++;
        if(result == 0) {
            fscanf(op,"%d",&pf);
            fscanf(op,"%d",&bs);

            fprintf(retorno,"%d\n",pf-1); // linhas
            fprintf(retorno,"%d\n",bs); // colunas

            for(i=0;i<pf;i++) //valores em y
                for(j=0;j<bs;j++) { //valores em x
                    fscanf(op,"%d",&cor); /*(*(MATRIZ_IMAGEM+LINHA)+COLUNA)
                    /*(*(MATRIZ_IMAGEM+i)+j)=cor;
                    if(cor<=240){ //INICIAL 200
                        putpixel(j+200,i+20,0); // 0 = cor preta = black

```

```

        cont_ponto++; //quantidade de pontos tem a imagem
        momento_x+=j;
        momento_y+=i;
    }
    else
        putpixel(j+200,i+20,15); //valores putpixel(x,y,color)
    }

}
getch();
for(i=0;i<pf-1;i++) //A ultima linha e' desprezada
    for(j=0;j<bs;j++) {
        val_1=abs(*(MATRIZ_IMAGEM+i+j)-(*(MATRIZ_IMAGEM+i+j+1)));
        val_2=abs(*(MATRIZ_IMAGEM+i+j)-(*(MATRIZ_IMAGEM+i+1+j)));
        if(val_1>30)
            val_1=0; //Quanto maior o valor menor a quantidades
        else //de pontos de borda
            val_1=1;

        if(val_2>30)
            val_2=0; //para F1 e F2 valor de 17
        else
            val_2=1;

        if(val_1==val_2)
            fprintf(retorno,"%d\n",15);
        else{
            putpixel(j+400,i+20,15); //putpixel(x,y,cor)
            fprintf(retorno,"%d\n",0); //Regiao da borda
            cont_ponto1++;
        }
    }
}
}
else{
    outtextxy(150, 66,"Arquivo");
    outtextxy(150, 72,"Irreconhecivel");
    Parada();
    SayGoodbye();
    closegraph();
    exit(1);
}

}
else {
    outtextxy(150, 66, "Arquivo nao");
    outtextxy(150, 72, "encontrado");
    Parada();
    SayGoodbye();
    closegraph();
    exit(1);
}

gprintf(&pos_x,&pos_y,"Valor da Area      = %ld",cont_ponto);
gprintf(&pos_x,&pos_y,"Valor do Perimetro   = %ld",cont_ponto1);
gprintf(&pos_x,&pos_y,"Valor do Momento_x    = %ld",momento_x);
gprintf(&pos_x,&pos_y,"Valor do Momento_y    = %ld",momento_y);
gprintf(&pos_x,&pos_y,"Coordenada do Momento_x = %ld",momento_x/cont_ponto);
gprintf(&pos_x,&pos_y,"Coordenada do Momento_y = %ld",momento_y/cont_ponto);

```



```
Beep();
Parada();
SayGoodbye();
closegraph();
fclose(op);
fclose(retorno);
exit(1);
}

//=====//
```

Anexo Q - Listagem do Programa “Captimag.C”.

```
//=====//
//
// PROGRAMA PARA CAPTURAR A IMAGENS DA FRAME-GRABBER
//
// AUTOR : Eng. ERNANY PARANAGUÁ DA SILVA
//
// NOME : CAPTIMAG.C
//
//FUNÇÕES : ROTINAS PARA CAPTURAR A IMAGEM
//=====//

#include <stdio.h>
#include <ctype.h>
#include <graph.h>
#include <conio.h>
#include <stdlib.h>
#include <io.h>
#include "isdefs.c"

/* Texto a ser escrito na tela de video */

void SAVE_VIDEO(int, int, int, int), RESTORE_VIDEO(int, int);
int buffer[32767];
void *buf;

main()
{
    register int i;
    int k, j, cor;
    int rr, rg, rb;
    char nome[15];
    short red, blue, green;
    char garb[80];
    static int oluts[3][256];
    static int ilut[256];
    FILE *sv;

    is_initialize();
    is_display(1);
    is_select_input_frame(0);
    is_select_output_frame(0);
    is_set_sync_source(1);
    is_select_olut(0);

    // Inicia uma numeracao randomica para saida de cores vermelha e azul

    rr=rand();
    rg=rand();
    rb=rand();

    //configura a tabela de saida look-up para 6-bit de niveis de cinza
    //da esquerda e os 2-bits da direita sao zerados

    for(i=0;i<256;i++)
        ilut[i]=i&~0x00000003;
```

```

//configura a tabela de saida
for(i=0;i<256;i+=4)
{
// escala de cinza para PASSTHRU WINDOW

    oluts[0][i]=i;
    oluts[1][i]=i;
    oluts[2][i]=i;

// conjunto randomico de valores de pixels (borda em torno do passthru window)

    oluts[0][i+1]=rr-rand();
    oluts[1][i+1]=rg-rand();
    oluts[2][i+1]=rb-rand();

// controle de cor para texto

    oluts[0][i+2]=0;
    oluts[1][i+2]=0;
    oluts[2][i+2]=0;

    oluts[0][i+3]=0;
    oluts[1][i+3]=0;
    oluts[2][i+3]=0;

}

// carrega tabela look-up, e estabiliza overlay/passthrough window

is_load_olut(0,&oluts[0][0],&oluts[1][0],&oluts[2][0]);
is_set_active_region(0,0,512,512);
is_set_constant(0,221);
is_set_active_region(190,200,128,128); /* linha inicial,col ini,alt,larg */
is_set_constant(0,0);
is_set_active_region(0,0,512,512);

// carrega mascara

is_load_mask(3);

// espaco no passthrough

is_passthru();

_setvideomode(18);

SAVE_VIDEO(20, 20, 450, 72);
_outtext( "SAVE AS:");

gets(nome);
if(!(sv=fopen(nome,"wb"))){
    exit(1);
}
fprintf(sv,"%d\n",67);
fprintf(sv,"%d\n",69);
fprintf(sv,"%d\n",76);
fprintf(sv,"%d\n",83);

```



```

fprintf(sv,"%d\n",79);
fprintf(sv,"%d\n",128);
fprintf(sv,"%d\n",128);
RESTORE_VIDEO(20,20);

for(k=190;k<318;k++) {
    is_get_pixel(0,k,0,460,buffer);
    for(j=200;j<328;j++) {
        cor = buffer[j];
        _setcolor(cor%16+3);
        _setpixel(j,k);
        fprintf(sv,"%d\n",cor);
    }
}
fclose(sv);
is_passthru();
getch();
_setvideomode(_DEFAULTMODE);
is_end();

}

void SAVE_VIDEO(int a, int b, int c, int d)
{
    int size;

    size = _imagesize(a,b,c,d);
    buf = (void *) malloc(size);
    _getimage(a, b, c, d, buf);
}

void RESTORE_VIDEO(int a, int b)
{
    _putimage(a, b, buf, _GPSET);
    free(buf);
}

//=====//

```

Anexo R - Listagem do Programa de Bibliotecas "Gprint1.l".

```

/=====//
//
// BIBLIOTECA
//
// AUTOR : Eng. ERNANY PARANAGUÁ DA SILVA
//
// NOME : GPRINT1.L
//
//FUNÇÕES : CONTÉM AS PRINCIPAIS ROTINAS PARA O MODO GRÁFICO
//=====//

#include <stdarg.h>
#include <stdio.h>
#include <math.h>
#include <conio.h>
#include <stdlib.h>
#include <graphics.h>
#define ESC    0x1b          // Define the escape key
//=====//

int  MaxX, MaxY;              // Valor maximo de resolucao da tela
int  MaxColors;              // Valor maximo de cores
int  pos_x,pos_y;

//=====//
void MainWindow( char *header );
void changetextstyle(int font, int direction, int charsize);
void DrawBorder(void);
void Pause(void);
void Parada(void);
void StatusLine( char *msg );
void SayGoodbye(void);
void Beep(void);
void RESTORE_VIDEO(int a, int b);
void SAVE_VIDEO(int a, int b, int c, int d);
void *buf;
void graph_init(void);

void erasestr( int xloc, int yloc, char *str );
int gprintf( int *xloc, int *yloc, char *fmt, ... );
void gprintxy(int xloc, int yloc, char *fmt,...);
float le_tela(int x, int y);
void le_string(int x,int y, char *nome);
void gprintc( int *xloc, int *yloc, char *fmt,... );

//=====//
//
//      GPRINTF: Usado como printf exeto que a saida e' especificada  //
//      por uma coordenada                                     //
//
//=====//

int gprintf( int *xloc, int *yloc, char *fmt, ... )
{
    va_list argptr;          // Lista de argumentos do pnteiro
    char str[140];           //Buffer

```

```

int cnt;

va_start( argptr, fmt );           // Inicializa funcao va_

cnt = vsprintf( str, fmt, argptr ); // imprime para o buffer
outtextxy( *xloc, *yloc, str );    // Envia para a tela grafica
*yloc += textheight( "H" ) + 2;    // Avanca para a proxima linha
if(*yloc>=445){
    Parada();
    MainWindow("===== Programa da Transformada de Hough =====");
    settextstyle(0, HORIZ_DIR, 1);
    settextjustify( LEFT_TEXT, TOP_TEXT );
    *yloc=10;
}

va_end( argptr );                  // Fecha a funcao va_

return( cnt );

}

//=====//

void gprintxy(int xloc, int yloc, char *fmt,...)
{
    va_list argptr; //lista de argumentos do ponteiro
    char str[140]; //buffer para construir a string
    struct textsettingstype textinfo;
    va_start(argptr, fmt); //inicializa funcoes va_
    vsprintf(str, fmt, argptr); //adiciona ao buffer str
    gettextsettings (&textinfo); //checa as selecoes
    erasestr(xloc,yloc,str); //apaga a area
    outtextxy(xloc, yloc, str); //escreve a string
    va_end( argptr ); //fecha funcao va_
}

//=====//
//
//      ERASESTER: Utilizada para apagar uma parte da tela antes que //
//      uma nova string seja escrita ou simplesmente para remover uma //
//      string da tela
//
//=====//

void erasestr( int xloc, int yloc, char *str )
{
    struct textsettingstype textinfo;
    int xdim, ydim;
    void *textimage; /* ponteiro da imagem */
    gettextsettings (&textinfo); /* checa as selecoes */
    switch (textinfo.direction) /* obtem as dimensoes */
    {
        case HORIZ_DIR:
            xdim = textwidth( str );
            ydim = textheight( str );
            xloc--; break;
        case VERT_DIR:
            xdim = textwidth( str );
            ydim = textheight( str );
            yloc++;
    }
}

```



```

switch ( textinfo.horiz )      /* ajuste horizontal */
{
    case LEFT_TEXT:            break;
    case CENTER_TEXT: xloc -= xdim / 2; break;
    case RIGHT_TEXT: xloc -= xdim;
}
switch ( textinfo.vert )      /* ajuste vertical */
{
    case BOTTOM_TEXT: yloc -= ydim; break;
    case CENTER_TEXT: yloc -= ydim / 2; break;
    case TOP_TEXT: ;
}
while( xloc<0 ) { xloc++; xdim--; } /* se estiver fora */
/* da tela */
while( yloc<0 ) { yloc++; ydim--; } /* move para tras */
textimage = malloc(
    imagesize( xloc, yloc, xdim, ydim ) );
getimage( xloc, yloc, xloc+xdim, yloc+ydim, textimage );
putimage( xloc, yloc, textimage, XOR_PUT );
free( textimage ); /* libera a memoria alocada */
/* finaliza EraseStr */
}

//=====================================================
//
//      LE_TELA()= Faz a leitura da tela utilizando a funcao
//      gprintf()
//
//=====================================================

float le_tela(int x,int y)
{
    int i, j, in, sinal, mat[8];
    float valor;
    char ch;

    in=0;
    i=0;
    sinal=1;
    do {
        ch=getch();
        if(ch!='\r') {
            if(ch=='\b' && in!=0) {
                in--;
                gprintxy(x+(10*i),y,"%c",' ');
                if(sinal==1 && in!=1) i=1;
                if(sinal==1) i=1;
                if(sinal==1 && in==0) gprintxy(x,y,"%c",' ');
                if(in==0) sinal=1;
            }
            if(ch=='-' && in==0) {
                sinal = -1;
                gprintxy(x,y,"%c",'-');
                in++;
            }
        }
        if(ch>='0' && ch<='9') {
            mat[i]=atoi(&ch);
            mat[i]=1*mat[i];
            gprintxy(x+(10*(i+1)),y,"%d",mat[i]);
        }
    }
}

```

```

        i++;
        in++;
    }

}

} while(ch!='\r');
valor=0;
for(j=0;j<i;j++) {
    valor += sinal*(float)mat[i-1-j]*(pow(10,j));
}

return (valor);
}

//=====
//
//      LE_STRING()= Especifica o tamanho das strings
//
//=====//

void le_string(int x,int y, char *mat)
{
    int i, in;
    char ch;

    in=0;
    i=0;
    do {
        ch=getch();
        if(ch!='\r') {
            if(ch=='\b' && in!=0) {
                i -=1;
                in--;
                gprintxy(x+(8*i),y,"%c",' ');
            }
            if(ch=='\b') {
                mat[i]=ch;
                gprintxy(x+(8*i),y,"%c",mat[i]);
                i++;
                in++;
            }
        }
    } while(ch!='\r');
    mat[i]='\0';
}

//=====
//
// GPRINTC: Utiliza da mesma forma que GPRINFT exceto pelo fato de a área onde o texto será
// escrito na tela ser restaurada para a cor de fundo
//=====//

void gprinc( int *xloc, int *yloc, char *fmt,... )
{
    va_list argptr;          /* lista de argumentos do ponteiro */
    char str[140];           /* buffer para construir a string */
    struct textsettingtype textinfo;

```

```

        int pos_adj = *xloc;          /* posicao default          */

        va_start( argptr, fmt ); /* inicializa funcoes va_          */
        vsprintf( str, fmt, argptr ); /* adiciona ao buffer str*/
        gettextsettings ( &textinfo ); /* checa as selecoes*/
        erasestr ( *xloc, *yloc, str ); /* apaga a area          */
        outtextxy ( pos_adj, *yloc, str ); /* escreve a string          */
        switch ( textinfo.direction )
        {
            /* ajuste a seguinte          */
            case HORIZ_DIR:          *yloc += textheight ( str ) + 2;
                                    break;
            case VERT_DIR:  *xloc += textheight( str ) + 2;

        }
        va_end( argptr );          /* fecha funcoes va_          */
    }                               /* finaliza GPRINTC          */

//=====//
//
//MAINWINDOW: Estabiliza a janela principal da tela grafica e mostra os codigos
//
//=====//

void MainWindow( char *header )
{
    int height;

    cleardevice();
    setcolor( 11);
    setviewport( 0, 0, MaxX, MaxY, 1 );          // abre janela grafica

    height = textheight( "H" );

    changetextstyle( DEFAULT_FONT, HORIZ_DIR, 1 );
    settextjustify( CENTER_TEXT, TOP_TEXT );
    outtextxy( MaxX/2, 2, header );
    setviewport( 0, height+4, MaxX, MaxY-(height+4), 1 );
    DrawBorder();
    setviewport( 1, height+5, MaxX-1, MaxY-(height+5), 1 );

}

//=====//
//
//          //
//          CHANGETEXTSTYLE = similar a settextstyle, entretanto checa os erros que ocorrem na //
//          leitura da fonte
//
//=====//

void changetextstyle(int font, int direction, int charsize)
{
    int ErrorCode;

    graphresult();          // limpa a area de codigo
    settextstyle(font, direction, charsize);
    ErrorCode = graphresult();          // Checa o resultado
    if( ErrorCode != grOk ){
        closegraph();
    }
}

```



```

    gprintf(&pos_x,&pos_y," Graphics System Error: %s\n", grapherrormsg( ErrorCode ) );
    exit( 1 );
}
}

//=====================================================
//
//      DRAWBORDER: Desenha a linha em torno da tela corrente
//
//=====================================================

void DrawBorder(void)
{
    struct viewporttype vp;
    setcolor( MaxColors - 1 );
    setlinestyle( SOLID_LINE, 4, NORM_WIDTH );
    getviewsettings( &vp );
    rectangle( 0, 0, vp.right-vp.left, vp.bottom-vp.top );
    setbkcolor(9);
}

//=====================================================
//
//      PAUSE: Pausa no sistema aguardando uma tecla qualquer
//      Se for ESC o programa encerra
//
//=====================================================

void Pause(void)
{
    static char msg[] = "Esc para abortar or pressione qualquer tecla...";
    int c;
    StatusLine( msg );
    c = getch();
    if( ESC == c ){
        closegraph();
        exit( 1 );
    }
    if( 0 == c ){
        c = getch();
    }
    cleardevice();
}

//=====================================================
//
//      PARADA: Aguardando uma tecla qualquer
//
//=====================================================

void Parada(void)
{
    static char      msg[] = "Pressione qualquer tecla";
    StatusLine(msg);
    while(kbhit()) getch();
    getch();
    setviewport( 0,468,639,479,1);
    clearviewport();
    setviewport( 1, 13, MaxX-1, MaxY-13, 1 );
}

```

```

}

//=====//
//
//      STATUSLINE: Mostra o status da linha
//
//=====//

void StatusLine( char *msg )
{
    int height;
    setviewport( 0, 0, MaxX, MaxY, 1 );
    setcolor( MaxColors - 1 );
    changetextstyle( DEFAULT_FONT, HORIZ_DIR, 1 );
    setttextjustify( CENTER_TEXT, TOP_TEXT );
    setlinestyle( SOLID_LINE, 0, NORM_WIDTH );
    setfillstyle( EMPTY_FILL, 0 );
    height = textheight( "H" ); // Determina a altura corrente
    bar( 0, MaxY-(height+4), MaxX, MaxY );
    rectangle( 0, MaxY-(height+4), MaxX, MaxY );
    outtextxy( MaxX/2, MaxY-(height+2), msg );
    setviewport( 1, height+5, MaxX-1, MaxY-(height+5), 1 );
}

//=====//
//
//      SAYGOODBYE : Tela de finalizacao do trabalho
//
//=====//

void SayGoodbye(void)
{
    MainWindow( "== Final ==" );
    outtextxy( MaxX/2, (MaxY-400)/2, "Trabalho de Transformada de Hough" );
    outtextxy( MaxX/2, (MaxY-200)/2, "Desenvolvido por" );
    outtextxy( MaxX/2, (MaxY)/2, "Ernany Paranaguá da Silva" );
    outtextxy( MaxX/2, (MaxY+200)/2, "EESC - USP" );
    StatusLine( "Pressione qualquer tecla para EXIT" );
    getch();

    cleardevice();           // Limpa a tela grafica
}

//=====//
//
//      BEEP : Aviso sonoro
//
//=====//

void Beep(void)
{
    sound(220);delay(100);nosound();
    delay(50);
    sound(660);delay(100);nosound();
}

```

```

}

//=====//
//
//      SAVE_VIDEO : Salva a area de video especificada
//
//=====//

void SAVE_VIDEO(int a, int b, int c, int d)
{
    int size;
    size = imagesize(a,b,c,d);
    buf = (void *) malloc(size);
    getimage(a, b, c, d, buf);
}

//=====//
//
//      RESTORE_VIDEO : Salva a area de video especificada
//
//=====//

void RESTORE_VIDEO(int a, int b)
{
    putimage(a, b, buf, COPY_PUT);
    free(buf);
}

//=====//
//
//      GRAPH_INIT: Inicializa os parametros graficos
//
//=====//
void graph_init()
{
    int gdriver = DETECT, gmode, errorcode=0;
    initgraph(&gdriver, &gmode, "");
    errorcode = graphresult();
    if (errorcode != grOk)
    {
        fprintf(&pos_x,&pos_y,"Graphics error: %s", grapherrormsg(errorcode));
        Pause();
        exit(1);
    }
}
//=====//

```


REFERÊNCIAS BIBLIOGRÁFICAS

- ACHCAR, J.A. (1995). Planejamento de Experimentos em Engenharia e Indústria. *ICMSC -USP*, São Carlos.
- ASOUDEGI, E. (1992). Quantitative automated inspection of standard parts using machine vision. *Computers and Industrial Engineering*, v.23, nº 1-4, p.361-364.
- BALLARD, D.H. (1981). Generalizing the Hough transform to detect arbitrary shapes. *Pattern Recognition* . v.13, n.2, p.111-122.
- BATCHELOR, B.G.; HILL, D.A.; HODGSON, D.C. (1985). Automated visual inspection. *IFS Publications Ltd.*, North-Holland, U.K.
- BATCHELOR, B.G.; HILL, D.A.; HODGSON, D.C. (1985). Automated visual inspection. *IFS Publications Ltd.*, North Holland, UK.
- BEN-TZVI, D.; NAQVI, A.; SANDLER, M. (1990). Synchronous multiprocessor implementation of the Hough transform. *Computer Vision, Graphics, and Image Processing*, v.52, p.437-446.
- BEN-TZVI, D.; SANDLER, M.B. (1990). A combinatorial Hough transform. *Pattern Recognition Letters*, North-Holland, v.11, p.167-174.
- BOERNER, W.M.; HO, C.M. ; FOO, B.Y. (1981). Trans. antennas propag. *IEEE*, Ap.29, n.336.
- BOIKO, R.V.; KOMROV, V.A.; KRASILENKO, V.G. (1989) Fast method for calculating moment characteristics in image processing. *Optoelectronics, Instrumentation and Data Processing*. n. 6, p.15-20.

- BUMBACA, F.; SMITH, K.C. (1987). Design and implementation of a colour vision model for computer vision applications. *Comput. Vision, Graphics Image Process.*, v.39, p.226-245.
- CARLSON, B.D.; EVANS, E.D.; WILSON, S.L. (1994). Search radar detection and track with the Hough transform: Part I: System concept. *IEEE Transactionon Aerospace and Electronic Systems*, v.30, n.1, p.102-108, jan.
- CARLSON, B.D.; EVANS, E.D.; WILSON, S.L. (1994). Search radar detection and track with the Hough transform: Part II: Detection statistics. *IEEE Transactionon Aerospace and Electronic Systems*, v.30, n.1, p.109-115, jan.
- CARLSON, B.D.; EVANS, E.D.; WILSON, S.L. (1994). Search radar detection and track with the Hough transform: Part III: Detection performance with binary integration. *IEEE Transactionon Aerospace and Electronic Systems*, v.30, n.1, p.116-125, jan.
- CHASE, R.C. et al. (1980). *Proc. Soc. Photo-Opt. Instrum. Eng.*, v.231, n.265.
- CHIN, R.T. (1986). *Handbook of Pattern Recognition and Image Processing*. Madison, Wisconsin, Academic Press. Inc. Chapter 24: Algorithms and techniques for automated visual inspection.
- CHURCHILL, D.B.; BILSLAND, D.M.; COOPER, T.M. (1992). Comparison of machine vision with human measurement of seed dimensions. *Transactions of the ASAE*, v.35, n. 1, p.61-64.

- DAVIES, E.R.; JOHNSTONE, A.I.C. (1989). Methodology for optimising cost/speed tradeoffs in real-time inspection hardware. *IEE Proceedings*, v.136, Pt.E, n.1, p.62-69, jan.
- DAVIES, L.S. (1975) A survey of edge detection techniques. *Computer Graphics and Image Processing*, v.4, n.3, p.248-270
- DEANS, S.R. (1981). Hough transform from the Randon transform. *IEEE , T-PAMI* 3, p.185-188.
- DEMPSEY, G.L.; McVERY, E.S. (1992). A Hough transform system based on Neural Networks. *IEEE Transaction on Industrial Electronics*, v.39, n.6, p.522-528, dec.
- DUDA, R.O.; HART, P.E. (1972). Use of the Hough Transformation to detect lines and curves in pictures. *Graphics and Image Processing*, California, v.15, n.1, p.11-15, jan.
- FLUSSER, J.;SUK, T. (1993). Pattern recognition by affine moment invariants. *Pattern Recognition*, v.26, n.1, p. 167-174, jan.
- FOSTER III, J.W. et al. (1990). Automated visual inspection: a tutorial. *Computers ind. Engng*, v.18, n.4, p.493-504.
- FU, K.S. (1982). Pattern Recognition for Automatic Visual Inspection. *IEEE Computer*, v.15, n.12, p.34-40, dec.
- GOMES, F.P. (1976). Iniciação à Estatística. Ed. Nobel, São Paulo, 5° edição.
- GONZALEZ, R.C.; WINTZ, P. (1987). Digital image processing. *Reading, MA: Addison-Wesley*, p.504.

- GORDON, R.; BENDER, R.; HERMAN, G.T. (1970). *J. Theor. Biol.*, v.29, n.471.
- GRAF, G.L. et al. (1981). Automatic detection of surface flaws on apples using digital image processing. *ASAE Paper*, N.81-3537. St. Joseph, MI: ASAE.
- GUNDIMEDA, D.P.; VELUVALI, S.K. (1991). An automated vision based approach for high voltage insulator testing. *Electric Power Research*, v.22, p.113-120.
- HAN, J.H.; LÁSZLÓ, T.; POSTON, T. (1994). Fuzzy Hough transform. *Pattern Recognition Letters*, v.15, p.649-658, jul.
- HOU, T.H.; LIN, L.; SCOTT, P.D. (1993). A neural network-based automated inspection system with an application to surface mount devices. *International Journal of Production Research*, v.31, n.5, p.1171-1187, may.
- HOUGH, P.V.C. (1962). *Method and means for recognizing complex patterns* U.S. Patent 3.069.654. 18 dec.
- HOY, D.E.P.; YU, F. (1991). Surface quality assessment using computer vision methods. *Elsevier Science Publishers B.V.*, p.265-274.
- HU, M.K. (1961). Pattern Recognition by Moment Invariants. *Proceedings of the IRE*, p.1428, sept.
- HU, M.K. (1962). Visual Pattern Recognition by Moment Invariants. *IRE Transactions on Information Theory*, p.179-187, feb.
- HUECKEL, M. (1973). A local visual operator which recognizes edges and lines. *J. Ass. comput.*, v.20, p.634-646, mar.

- HUNT, D.J. et al. (1990). Hough transform and signal detection theory performance for images with additive noise. *Computer Vision Graphics and Image Processing*, v.52, p.386-401.
- ILLINGWORTH, J.; KITTLER, J. (1987). The adaptive Hough transform. *IEEE Trans. Pattern Anal. Machine Intell.* v.9, n.5, p.690-695. sep.
- KASTURI, RANGACHAR et al. (1990). A system for interpretation of line drawings. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v.12, n.10, p.978-991, oct
- KENT, E.W.; SHNEIER, M.O. (1986). Eyes for automatons. *IEEE Spectrum*, p.37-45, mar.
- KIMME, C.; BALLARD, D.; SKLANSKY, J. (1975). Finding circles by an array of accumulators. *Comm. Assoc. Comput Mach.*, v.18, n.2, p.120-122.
- LECORDIER, R.; MARTIN, P. (1992). Data flow processors for automated visual inspection. *Microprocessing and Microprogramming*, v.34, p.37-40.
- LI, H.; LAVIN, M.A.; MASTER, R.J.L. (1986). Fast Hough Transform: A hierarchical approach. *Computer Vision Graphics And Image Processing*, v.36, p.139-161.
- LI, H.F.; PAO, D.; JAYAKUMAR, R. (1989). Improvements and systolic implementation of the Hough transformation for straight line detection. *Pattern Recognition*, v.22, n.6, p.697-706
- LI, Z. (1994). Stereo correspondence based on line matching in Hough space using dynamic programming. *IEEE Transactions on Systems, Man, and Cybernetics*, v.24, n.1, p.144-152, jan.

- LIN, W.; WANG, S. (1994). Note on the calculation of moments. *Pattern Recognition Letters*. v.15, n.11, p.1065-1070, nov.
- LOW, A. (1991). *Introductory Computer Vision and Image Processing*. Londres, McGraw-Hill Book Company. Cap. 6, p.71-99.
- MAEDA et al. (1990). Automated visual inspection for LSI wafer multilayer patterns by cascade pattern matching algorithm. *Systems and Computers in Japan*, v.21, n.12, p.65-77, dec.
- MERLIN, P.M.; FARBER, D.J. (1975). A parallel mechanism for detecting curves in pictures. *IEEE Trans. Comput* , v.24, p.96-98.
- MILLER, B.K.; DELWICHE, M.J. (1991). Peach defect detection with machine vision. *Transactions of the ASAE*, v.34, n.6, p.2588-2597.
- NAGATA, T.; ZHA, Hong-Bin (1991). Recognizing and locating a known object from multiple images. *IEEE Transactions on Robotics and Automation*, v.7, n.4, p.434-448, aug.
- NORTON-WAYNE, L. (1991). Inspection of lace using machine vision. In: 9th ANNUAL EUROGRAPH-ICS (UK) CONFERENCE, Sheffield, 1991. *Anais*. Sheffield ,p.113-119, apr.
- O'GORMAN, F.O.; CLOWES, M.B. (1976). Finding picture edges through collinearity of feature points. *IEEE Trans. Comput.*, v.25, p.449-456.
- OKAWA, Y.; MIZUNO, S. (1991). Automatic inspection of diode pellets. *Machine Vision and Applications*, v.4, p.131-133.
- PARADINE, C.G.; RIVETT, B.H.P. (1974). *Métodos Estatísticos para Tecnologistas*. Ed. Polígono, Ed. da USP, São Paulo.

- PIRONEN et al.(1990). Automated visual inspection of rolled metal surfaces. *Machine Vision and Applications*, New York, v.3, p.247-254.
- PREWITT, J.M.S. (1970). Object enhancement and extraction. *Picture Processing and Psychopictories*. Academic Press, New York.
- PRINCEN, J. et al.,(1990). A comparative study of Hough transform algorithms. *Internal Report, Dept. of Electronic and Electrical Engineering, University of Surrey*.
- PUTTRÉ, M. (1992). Machine vision systems struggle for definition. *Mechanical Engineering*, p.80-83, jul.
- REHKUGLER, G.E.; THROOP, J.A. (1987). Image processing algorithm for apple defect detection. *ASAE Paper*, n.87-3041.
- RIGNEY, M.P.; BRUSEWITZ, G.H.; KRANZLER, G.A. (1992). Asparagus defect inspection with machine vision. *American Society of Agricultural Engineers*, v.35, n.6, p.1873-1878, nov-dec.
- ROSENFELD, A. (1985). Introduction to machine vision. *IEEE Control Systems Magazine*, p.14-17, aug.
- ROSENFELD, A.; ORNELAS, J.; HUNG, Y. (1988). Hough transform algorithm for mesh-connected SIMD parallel processor. *Comput. Vision Graphics Image Process*, v.41, p.293-305.
- SARKAR, N.; WOLFE, R.R. (1985). Feature extraction techniques for sorting tomatoes by computer vision. *Transactions of the ASAE*, v.28. n.3, p.970-979.

- SCHALKOFF, R.J. (1989). Digital Image Processing and Computer Vision. New York, John Wiley & Sons, Inc. Cap. 4, p.145-178.
- SCHNEIER, M. (1981). Two hierarchical linear feature representations: Edge pyramids and edge quadtrees comput. *Graphics Image Process*, v.17, p.211-224.
- SCHNEIER, M. (1982). Extracting linear features from images using pyramids. *IEEE Trans. Systems Man Cybern*, SMC 14, n.4, p.569-572.
- SHAPIRO, S.D. (1975). Transformation for the computer detection of curves in noisy pictures. *Comput Graphics Image Process*, v.4, p.328-338.
- SHAPIRO, S.D. (1978). Feature space transforms for curve detection. *Pattern Recognition*, v.10, p.129-143.
- SHAPIRO, S.D. (1978). Properties of transforms for the detection of curves in noisy pictures. *Comput. Graphics Image Process*, v.8, p.219-236.
- TSUKAHARA, H.; NAKASHIMA, M.; SUGAWARA, T. (1991). Automated visual inspection system for bonded IC wires. *Fujitsu Sci. Tech.J.*, v.28, p.107-114, mar.
- WALLACE, R.S. (1985). A modified Hough transform for lines. *IEEE Conf. Computer Vision and Pattern Recognition*, San Francisco, p.665-667.
- WANG, J.; HOWARTH, P.J. (1990). Use of the Hough transform in automated lineament detection. *IEEE Transaction on Geoscience and Remote Sensing*, v. 28, n.4, p.561-566, jul.
- WASZKA, J.S. (1978). A survey of threshold selection techniques. *Computer Graphics and Image Processing*, v.7, p.239-256.

WEISS, R.; BOLDT, M. (1986). Geometric grouping applied to straight lines. *IEEE Conference Computer Vision and Pattern Recognition*, p.489-495.

WEN, W.; LOZZI, A. (1993). Recognition and inspection of manufactured parts using line moments of their boundaries. *Pattern Recognition*, v.26, n.10, p.1461-1471.